Chen Tang's Knowledge Database

ChenTang@link.cuhk.edu.cn

ChenTang01.github.io

Last updated on October 11, 2024

Preface

The following is a compendium of my academic notes spanning various domains. I present these notes publicly to share my methodology for managing one's knowledge networks.

The inevitability of encountering occasional errors is acknowledged.

This notebook will undergo continuous updates.

Contents

Chapter 1

Mathematics, Statistics & Optimization

1.1 Proof Techniques

General Proof Techniques

Direct proof: given a set of rules (axioms), rewrite the statement in the form of the axioms.

Example 1.1.0.1

Let k be an integer, define $n = 2k+1$ as odd. Show that the square of the sum of two consecutive integers is odd.

Solution:

Rewrite $(x + [x+1])^2)$ *in the form of* $2k+1$, *where* k *is an integer.*

$$(x + [x+1])^2 = 4x^2 + 4x + 1 = 2(2x^2 + 2x) + 1 \tag{1.1.0.1}$$

Proof by Induction: when you need to show that some sequence of objects has some property P. You can

1. First shows the first of these objects has this property P;
2. Then prove that if the ith object has property P, then the $i+1$st object also has property P.

Proof by Contradiction is usually used when proving a given statement true or false. Assume one statement is true or false, and derive one expression from the statement until the expression is contradictory to an axiom.

1.1.1 Play with Bound

Prophet Inequality

Bulow-Klemperer

1.1.2 Existence and Uniqueness Proofs

1.2 Calculus and Linear Algebra

1.2.1 Keys of Calculus

Solving Limit:

1. By definition, prove $\forall \epsilon > 0$, there exists a $\delta > 0$ such that $\forall x \in (D(c, \delta)), |f(x) - L| \leq \delta$;
2. The Sandwish theorem, find $h(x) \leq f(x) \leq g(x)$ in an interval;
3. $\lim_{x\to 0}(\sin x)/x = 1$, if $\lim_{x\to c} f(x) = 0$ and $f(x) \neq 0$ in an open interval near $x = c$ (except at $x = c$): $\lim_{x\to c} \frac{\sin f(x)}{f(x)} = 1$;
4. Fraction form: eliminating common factors or rationalizing the denominator; l'hopital's rule (differentiable); bounded numerator while infinite denominator
5. Order of growth: $f = o(g)$ if $\lim_{x\to\infty} \frac{f(x)}{g(x)} = 0$, a tighter bound is $f = O(g)$ if $\frac{f(x)}{g(x)} \leq M$ for some $M > 0$. If $f = O(g)$ and $g = O(f)$, then g and f are at the same order.

Remark 1.2.1

Keys on Continuity:

- Definition: $\lim_{x\to c} f(x) = f(x)$ and both exist;
- If f and g are both continuous function, their algebraic combinations and compositions are also continuous;
- If g is continuous at b and $\lim_{x\to c} f(x) = b$, then $\lim_{x\to c} g(f(x)) = \lim_{f(x)\to b} g(f(x)) = g(\lim_{x\to c} f(x)) = g(b)$;
- Intermediate Value Theorem: if f is continuous on $[a, b]$, for any $f(a) \leq y \leq f(b)$, there exists some $a \leq c \leq b$ such that $f(c) = y$
- Mean-value theorem: if f is differentiable on $[a, b]$, there must exist a point $x \in (a, b)$ such that $\frac{f(b)-f(a)}{b-a} = f'(c)$; Cauchy's Mean-value theorem: $\frac{f'(c)}{g'(c)} = \frac{f(b)-f(a)}{g(b)-g(a)}$.

1. $\int k\,dx = kx + C$ (any number k)
2. $\int x^n\,dx = \frac{x^{n+1}}{n+1} + C \quad (n \neq -1)$
3. $\int \frac{dx}{x} = \ln|x| + C$
4. $\int e^x\,dx = e^x + C$
5. $\int a^x\,dx = \frac{a^x}{\ln a} + C \quad (a > 0,\ a \neq 1)$
6. $\int \sin x\,dx = -\cos x + C$
7. $\int \cos x\,dx = \sin x + C$
8. $\int \sec^2 x\,dx = \tan x + C$
9. $\int \csc^2 x\,dx = -\cot x + C$
10. $\int \sec x \tan x\,dx = \sec x + C$
11. $\int \csc x \cot x\,dx = -\csc x + C$
12. $\int \tan x\,dx = \ln|\sec x| + C$
13. $\int \cot x\,dx = \ln|\sin x| + C$
14. $\int \sec x\,dx = \ln|\sec x + \tan x| + C$
15. $\int \csc x\,dx = -\ln|\csc x + \cot x| + C$
16. $\int \sinh x\,dx = \cosh x + C$
17. $\int \cosh x\,dx = \sinh x + C$
18. $\int \frac{dx}{\sqrt{a^2 - x^2}} = \sin^{-1}\left(\frac{x}{a}\right) + C$
19. $\int \frac{dx}{a^2 + x^2} = \frac{1}{a}\tan^{-1}\left(\frac{x}{a}\right) + C$
20. $\int \frac{dx}{x\sqrt{x^2 - a^2}} = \frac{1}{a}\sec^{-1}\left|\frac{x}{a}\right| + C$
21. $\int \frac{dx}{\sqrt{a^2 + x^2}} = \sinh^{-1}\left(\frac{x}{a}\right) + C \quad (a > 0)$
22. $\int \frac{dx}{\sqrt{x^2 - a^2}} = \cosh^{-1}\left(\frac{x}{a}\right) + C \quad (x > a > 0)$

Figure 1.1: Basic integration formulas

Differential Equation

Definition 1.2.1.1: First-order Linear Differential Equation

$$\frac{dy}{dx} + P(x)y = Q(x),$$

where P, Q are continuous functions on x.
Solution: multiply $v(x) = e^{\int P(x)dx}$ on both sides, transform it into:

$$\frac{d}{dx}(v(x) \cdot y) = v(x)Q(x)$$
$$v(x) \cdot y = \int v(x)Q(x)dx$$
$$y = \frac{1}{v(x)} \int v(x)Q(x)dx$$

1.2.2 Keys of Linear Algebra and Matrix Theory

The main contents of this subsection are notes from [Deisenroth et al., 2020]. The properties of Matrix Multiplication:

- Associativity: $(AB)C = A(BC)$;
- Distributivity: $A(C + D) = AC + AD$;
- Identity Multiplication: $I_m A = A, AI_n = A$

Only square matrixes have the inverse matrix, and the inverse is unique. If a matrix has an inverse, then it's called **regular/invertible/nonsingular**. The properties of inverses and transposes:

- $(AB)^{-1} = B^{-1}A^{-1}$;
- $(A + B)^{-1} \neq A^{-1} + B^{-1}$;
- $(AB)^\top = B^\top A^\top$;
- $(A + B)^\top = A^\top + B^\top$;

- $(A^\top)^{-1} = (A^{-1})^\top$.

Gaussian elimination can find the inverse matrix: $[A \mid I] = [I \mid A^{-1}]$.

Solutions of Linear Systems

Reduced Row-Echelon Form Matrix:

$$\boldsymbol{A} = \begin{bmatrix} 1 & 3 & 0 & 0 & 3 \\ 0 & 0 & 1 & 0 & 9 \\ 0 & 0 & 0 & 1 & -4 \end{bmatrix} \tag{1.2.2.1}$$

There are *pivot (basic variables)* and *free variable*, and the column of free variable is dependent on pivots. The steps to find solutions of linear systems:

1. Find a particular solution for $Ax = \boldsymbol{b}$ by setting all free variable zero;
2. Find all solutions for $Ax = 0$;
3. Add them up.

There is an iterative method to solve large-scale linear equations: define error $= \|x^{(k+1)} - x_*\|$, then optimize the function $x^{(k+1)} = Cx^{(k)} + d$ and iterate it.

Vector Spaces, Basis and Rank

Definition 1.2.2.1: Vector Space and Subspace

A **Vector Space** $V = (\mathcal{V}, +, \cdot)$ is a set $\mathcal{V}$ with two operations:

1. $+ : \mathcal{V} \times \mathcal{V} \to \mathcal{V}$;
2. $\cdot : \mathbb{R} \times \mathcal{V} \to \mathcal{V}$ ($\mathbb{R}$ can be replaced by $\mathbb{C}$).

For $\mathcal{U} \subseteq \mathcal{V}$ and $\mathcal{U} \neq \emptyset$, then $U = (\mathcal{U}, +, \cdot)$ is a vector subspace.

Remark 1.2.2

- $V = (\mathcal{V}, +)$ is an *Abelian group*;
- The subspace needs to satisfy *closure*, $\lambda x \in U, x + y \in U$.

For subspaces V_1 and V_2, the sum operation of V_1 and V_2 is defined as:

$$V_1 + V_2 = \{v_1 + v_2 : v_1 \in V_1, v_2 \in V_2\}. \tag{1.2.2.2}$$

$$\dim V_1 + \dim V_2 = \dim(V_1 + V_2) + \dim(V_1 \cap V_2). \tag{1.2.2.3}$$

If $v = \sum_{i=1}^k \lambda_i x_i$, then v is a linear combination of $(x_1, x_2, \cdots, x_k)$. If **not**, all values of a solution are 0, then it's called a non-trivial solution. For $\sum_{i=1}^k \lambda_i x_i = 0$, if the non-trivial solution exists, then the vectors are called **linearly dependent**.

Definition 1.2.2.2: Basis and Rank

- **Generating Set**: if all vectors in V can be expressed as a linear combination of $\mathcal{A} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_k\} \subseteq \mathcal{V}$, then $\mathcal{A}$ is a generating set of V;
- **Span**: The set of linear combinations of $\mathcal{A}$ is its span;
- **Basis**: The minimal generating set (linearly independent) of a vector space V is called its basis;
- **Rank**: the number of linearly independent column vectors in a matrix $\mathcal{A} \in \mathbb{R}^{m\times n}$.

Remark 1.2.3

- $\mathrm{rk}(A) = \mathrm{rk}(A^\top)$;
- A matrix $A \in \mathbb{R}^{n\times n}$ is invertible if and only if $rk(A) = n$;
- The span of a matrix is also called its **image**, $dim(U) = rk(A)$;
- $Ax = b$ only has solution if and only if $rk(A) = rk(A \mid b)$;
- The solution (kernel, null space) to $Ax = 0$ has a dimension of $n - rk(A)$;

Let A be an $m \times n$ matrix, there always exist an $m \times m$ matrix P and $n \times n$ matrix Q such that:

$$PAQ = \begin{pmatrix} I_r & 0 \\ 0 & 0 \end{pmatrix}. \tag{1.2.2.4}$$

The size of $I - r$ is the rank of A.

Definition 1.2.2.3: Linear Mappings

For vector spaces V, W, a mapping $\Phi : V \to W$ is called linear mapping if:

$$\forall x, y \in V \forall \lambda, \psi \in \mathbb{R} : \Phi(\lambda x + \psi \boldsymbol{y}) = \lambda\Phi(x) + \psi\Phi(\boldsymbol{y}) \tag{1.2.2.5}$$

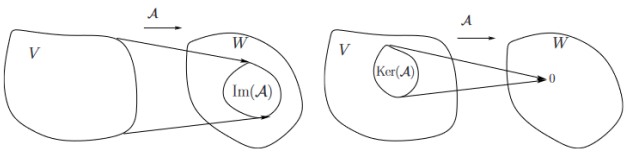


Figure 1.2: Image and Kernel

Determinant and Trace

Determinant is used to decide whether a matrix is invertible, denoted by $det(A)$ or $|A|$. Its geometric meaning is the signed volume. *Laplace Expansian* can be used to compute the determinant for large matrixes:

$$\det A = \sum_{j=1}^{n} (-1)^{1+j} a_{1j} \det A(1|j) \tag{1.2.2.6}$$

where $A(1|j)$ is the matrix A after eliminating row 1 and column j. If $det(A) = 0$, then it's non-invertible. Otherwise, it's invertible. **Trace** is defined as $Tr(A) = \sum_{i=1}^{n} a_{ii}$. Both determinant and trace can only be applied to the squared matrix.

Eigenvalue and Postive-Definiteness

Consider matrix $A \in \mathbb{R}^{n\times n}$ as a linear mapping rather than a simple matrix; A can map a n-dimensional space to another n-dimensional space.

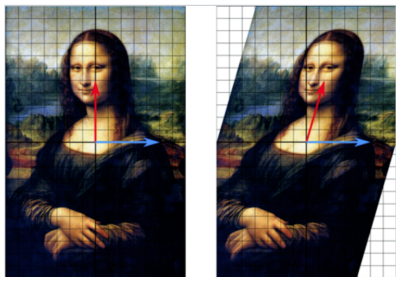


Figure 1.3: A shear mapping example

The Mona Lisa example is a linear mapping of two-dimensional space, but during the mapping, there are some vectors (such as the blue vector) that are only modified in length rather than direction. This is the idea of the eigenvector.

Definition 1.2.2.4: Eigenvalue and Eigenvector

For a square matrix A, if there exists a scalar λ and a vector $v \in \mathbb{R}^n$, such that $A \cdot v = \lambda \cdot v$. Which would lead us to $(A - \lambda I) \cdot v = \mathbf{0}$.

Remark 1.2.4

The following statements are equivalent:

- λ is an eigenvalue of $A \in \mathbb{R}^{n\times n}$;
- $rk(A - \lambda I) < n$;
- $det(A - \lambda I) = 0$.

The definiteness is similar to the idea of eigenvalue. Consider $X^T MX$ as the inner product of X and MX, where MX can be regarded as a transformed version of X. If $X^T MX > 0$ for all X, then we can get $\cos\theta = \frac{X^T \cdot MX}{||X|| \cdot ||MX||} > 0$, which means for all vectors, the linear mapping M can map the vector to an angle less than 90 degrees.

So, to show whether a matrix is positive definite, all the eigenvalues of the matrix must be calculated. If all values are larger than 0, then it's positive definite.

Remark 1.2.5

- $Tr(A) = \sum_i^n \lambda_i$;
- $\det(A) = \prod_{i=1}^n \lambda_i$;
- $\mathrm{Tr(A + B)} = \mathrm{Tr(A)} + \mathrm{Tr(B)}, \quad \det(\mathbf{AB}) = \det(\mathbf{A})\det(\mathbf{B})$;
- $\det(\mathbf{A}^{-1}) = 1/\det(\mathbf{A}), \quad \det(\mathbf{A}^n) = \det(\mathbf{A})^n$;
- $\det(\mathbf{I} + \mathbf{uv}^T) = 1 + \mathbf{u}^T\mathbf{v}$;
- $\begin{vmatrix} A & B \\ 0 & C \end{vmatrix} = \det A \det C$.

The approximation of the determinant. For small ϵ:

$$\det(\mathbf{I} + \varepsilon\mathbf{A}) \cong 1 + \det(\mathbf{A}) + \varepsilon\mathrm{Tr}(\mathbf{A}) + \frac{1}{2}\varepsilon^2\mathrm{Tr}(\mathbf{A})^2 - \frac{1}{2}\varepsilon^2\mathrm{Tr}(\mathbf{A}^2) \tag{1.2.2.7}$$

The 2×2 Matrix

Considering the 2×2 matrix $\mathbf{A}$, we have:

$$\det(\mathbf{A}) = A_{11}A_{22} - A_{12}A_{21} \tag{1.2.2.8}$$

$$\mathrm{Tr}(\mathbf{A}) = A_{11} + A_{22} \tag{1.2.2.9}$$

$$\lambda^2 - \lambda \cdot \mathrm{Tr}(\mathbf{A}) + \det(\mathbf{A}) = 0 \tag{1.2.2.10}$$

$$\mathbf{A}^{-1} = \frac{1}{\det(\mathbf{A})}\begin{bmatrix} A_{22} & -A_{12} \\ -A_{21} & A_{11} \end{bmatrix} \tag{1.2.2.11}$$

1.2.3 Norms & Matrix Derivative

Definition 1.2.3.1: Norm

The norm $||\cdot||$ is a mapping $V \to \mathbb{R}$, which assigns each vector a length that satisfy:

- *Absolutely homogeneous*: $\|\lambda x\| = |\lambda|\|x\|$;
- *Triangle inequality*: $\|x + y\| \leqslant \|x\| + \|y\|$;
- *Positivity*: $||x|| \geq 0$ and $\|\boldsymbol{x}\| = 0 \iff \boldsymbol{x} = \mathbf{0}$.

The L_p norm: $\|x\|_p = (|x_1|^p + |x_2|^p + \cdots + |x_n|^p)^{1/p}$.

Corollary 1.2.3.1: L_p norm

1. **Manhattan Norm**: $\|\boldsymbol{x}\|_1 := \sum_{i=1}^n |x_i|$;
2. **Euclidean Norm**: $\|\boldsymbol{x}\|_2 := \sqrt{\sum_{i=1}^n x_i^2} = \sqrt{\boldsymbol{x}^\top \boldsymbol{x}}$ (mostly denoted as $||\cdot||$);
3. **Infinity Norm**: $\|x\|_\infty = \max\{|x_1|, |x_2|, \ldots, |x_n|\}$.

Proof 1.2.1: Infinity Norm

By squeeze theorem:

First show that $\|x\|_p = (\sum_i |x_i|^p)^{\frac{1}{p}} \leq (\sum_i \max_i |x_i|^p)^{\frac{1}{p}} = n^{\frac{1}{p}} \max_i |x_i| \to \max_i |x_i| = \|x\|_\infty$,

Next show that $\|x\|_p = (\sum_i |x_i|^p)^{\frac{1}{p}} \geq (\max_i |x_i|^p)^{\frac{1}{p}} = \max_i |x_i| = \|x\|_\infty$.

Q.E.D.

Taking derivatives of Norm

Example 1.2.3.1

$X \in \mathbb{R}^m, f(x) : \mathbb{R}^m \to \mathbb{R}^n, g(x) = ||x||_2$, what is $\nabla g(X)$?

Solution:

First, $g(X) = \|f(X)\|_2 = \sqrt{\sum_{i=1}^n f_i(X)^2}$;

$$\nabla g(X) = \frac{1}{2}\left(\sum_{i=1}^n f_i(X)^2\right)^{-\frac{1}{2}}\left(\sum_{i=1}^n 2f_i(X)\nabla f_i(X)\right) = \frac{J_f(X)^T f(X)}{\|f(X)\|_2} \tag{1.2.3.1}$$

Matrix Norms

Definition 1.2.3.2: Induced Matrix Norms

Let $A \in \mathbb{R}^{W\times V}$ be a mapping from vector space $\mathcal{V}$ to $\mathcal{W}$.

$$\|A\|_{\mathcal{V},\mathcal{W}} := \max_{x\neq 0} \frac{\|Ax\|_{\mathcal{W}}}{\|x\|_{\mathcal{V}}}. \tag{1.2.3.2}$$

This measures how much the matrix A can stretch an n dimensional vector at most. Because norms are homogeneous to scaling, we also have:

$$\|A\|_{\mathcal{V},\mathcal{W}} = \sup_{\|v\|_{\mathcal{V}}=1} \|Av\|_{\mathcal{W}}. \tag{1.2.3.3}$$

Besides the 3 satisfactions listed above, there are two more properties that a matrix norm needs to satisfy:

- *Sub-multiplicative:* $\|AB\| \leq \|A\| \cdot \|B\|$ for all A, B;
- *Compatible with vector norm*: $\|Ax\| \leq \|A\| \cdot \|x\|$ for all A and vector x.

Proof 1.2.2: Sub-multiplicative

$$\|AB\| = \max_{\|v\|=1} \|ABv\| \leq \max_{\|v\|=1} \|A\|\|Bv\| = \|A\|\|B\|. \tag{1.2.3.4}$$

Q.E.D.

For a matrix $A \in \mathbb{R}^{m\times n}$:

$$\|A\|_1 := \max_{j\in\{1,\dots,n\}} \sum_{i=1}^{m} |a_{ij}|, \|A\|_\infty := \max_{i\in\{1,\dots,m\}} \sum_{j=1}^{n} |a_{ij}|. \tag{1.2.3.5}$$

The $\|\cdot\|_1$ is the maximum absolute column sum, the $\|\cdot\|_\infty$ is the maximum absolute row sum.
The **spectral norm**:

$$\|A\| := \|A\|_2 := \sqrt{\lambda_{\max}(A^\top A)}. \tag{1.2.3.6}$$

The **Frobenius Norm**:

$$\|A\|_F = \left(\sum_{i=1}^{m}\sum_{j=1}^{n} |a_{ij}|^2\right)^{1/2} = \sqrt{\mathrm{tr}(A^\top A)}. \tag{1.2.3.7}$$

Norms for Random Variables

The p-th moment is defined as $\mathbb{E}X^p, p > 0$, and the *absolute p-th moment* is $\mathbb{E}|X|^p$.

Definition 1.2.3.3: L^p-norm for the R.V

Similar to the norm in the general definition. For a R.V. X, its L_p norm is:

$$\|X\|_p = (\mathbb{E}|X|^p)^{1/p}, p \in (0,\infty). \tag{1.2.3.8}$$

Specifically, $\|X\|_\infty = esssup|X|$. When $p \in [1,\infty)$, the quantity $\|X\|_p$ is a norm (satisfying the triangle inequality) and $L_p = \{X : \|X\|_p < \infty\}$ is a *Banach space*. L_2 is a *Hilbert space*.

1.3 Probability Theory

The main contents of this section are from [Ross, 2014].

1.3.1 Probability Distribution

Common Discrete Distribution

Bernoulli(p)
pmf: $P(X = x \mid p) = p^x(1-p)^{1-x}; \quad x = 0, 1; \quad 0 \le p \le 1$
mean: $EX = p$; *variance*: $VarX = p(1-p)$

- A Bernoulli trial (named after James Bernoulli) is an experiment with only two possible outcomes;
- Bernoulli random variable $X = 1$ if "success" occurs and $X = 0$ if "failure" occurs where the probability of a "success" is p.

Binomial(n,p)
pmf: $P(X = x \mid n, p) = \binom{n}{x} p^x(1-p)^{n-x}, \quad x = 0, 1, \ldots, n; \quad 0 \le p \le 1$
mean: $EX = np$; *variance*: $np(1-p)$

- A Binomial experiment consists of n independent identical Bernoulli trials;
- $X = \sum_{i=1}^{n} Y_i$, where $Y_1, \cdots, Y_n$ are n identical, independent Bernoulli random variables.

Poisson(λ)
pmf: $P(X = x \mid \lambda) = \frac{e^{-\lambda}\lambda^x}{x!}; \quad x = 0, 1, \ldots; \quad 0 \le \lambda < \infty$
mean: $EX = \lambda$; *variance*: $VarX = \lambda$

- A Poisson distribution is typically used to model the probability distribution of the number of occurrences (with λ being the intensity rate) per unit time or per unit area;
- Binomial pmf approximates Poisson pmf. Poisson pmf is also a limiting distribution of a negative binomial distribution;
- A useful result: By Taylor series expansion: $e^{\lambda} = \sum_{x=0}^{\infty} \frac{\lambda^x}{x!}$.

Assumption 1.3.1.1: Poisson Process

1. Let $X(\Delta)$ be the number of events that occur during an interval Δ;
2. The events are independent: if $\Delta_1, \cdots, \Delta_n$ are disjoint intervals, then $X(\Delta_1), \cdots, X(\Delta_n)$ are independent;
3. $X(\Delta)$ only depends on the length of Δ;
4. The probability that exactly one event occurs in a small interval of length Δt equals $\lambda \Delta t + o(\Delta t)$;
5. Poisson distribution is not memoryless, but its interval (exponential distribution) is memoryless.

Geometric(p)
pmf: $P(X = x \mid p) = p(1-p)^{x-1}; \quad x = 1, 2, \ldots; \quad 0 \le p \le 1$
mean: $\frac{1}{p}$; *variance*: $\frac{1-p}{p^2}$

- The experiment consists of a sequence of independent trials;
- The property of memoryless: $P(X > s \mid X.t) = P(X > s - t)$.

Negative Binomial(r, p)
pmf: $P(X = x \mid r, p) = \binom{r+x-1}{x} p^r (1-p)^x, \quad x = 0, 1, \ldots; \quad 0 \le p \le 1$
mean: $EX = \frac{r(1-p)}{p}$; *variance*: $\frac{r(1-p)}{p^2}$

- assume there are many independent and identical experiments, to observe the rth success, X is the number of games to see the failure;

Hypergeometric(N, M, K)
pmf: $P(X = x \mid N, M, K) = \frac{\binom{M}{k}\binom{N-M}{K-x}}{\binom{N}{K}}; \quad x = 0, 1, 2, \ldots, K;$
$M - (N - K) \le x \le M; \quad N, M, K \ge 0$
mean: $EX = \frac{KM}{N}$; *variance*: $\frac{KM}{N}\frac{(N-M)(N-K)}{N(N-1)}$

Common Continuous Distribution

Uniform(a, b)
pdf:$f(x \mid a, b) = \frac{1}{b-a}$; *mean*:$EX = \frac{b+a}{2}$; *variance*: $VarX = \frac{(b-a)^2}{12}$.
Exponential(β)
pdf: $f(x \mid \beta) = \frac{1}{\beta} e^{-x/\beta}, 0 \le x < \infty, \beta > 0$; *mean*: $EX = \beta$; *variance*: $VarX = \beta^2$.
Gamma(α, β)
pdf:$f(x \mid \alpha, \beta) = \frac{1}{\Gamma(\alpha)\beta^\alpha} x^{\alpha-1} e^{-x/\beta}, \quad 0 \le x < \infty, \quad \alpha, \beta > 0$; *mean*:$\alpha\beta$; *variance*:$\alpha\beta^2$.

- The *gamma function* is defined as $\Gamma(\alpha) = \int_0^\infty t^{\alpha-1} e^{-t} dt$;
- $\Gamma(\alpha + 1) = \alpha\Gamma(\alpha), \alpha > 0$;
- $\Gamma(n) = (n-1)!, \quad$ for any integer $n > 0$.

Normal(μ, σ^2)
pdf:$f\left(x \mid \mu, \sigma^2\right) = \frac{1}{\sqrt{2\pi}\sigma} e^{-(x-\mu)^2/(2\sigma^2)}, \quad -\infty < x < \infty$; *mean*:$\mu$; *variance*:$\sigma^2$.

Example 1.3.1.1

for $f(x) = \frac{\beta\alpha^\beta}{x^{\beta+1}}, \quad \alpha < x < \infty, \quad \alpha > 0, \quad \beta > 0$:

- (a) Verify $f(x)$ is a pdf;
- (b) Derive the mean and variance of this distribution;
- (c) Prove that the variance does not exist if $\beta \le 2$.

Solution:

(a)

$$\int_{\alpha}^{\infty} f(x)dx = \int_{\alpha}^{\infty} \frac{\beta \cdot \alpha^{\beta}}{x^{\beta+1}} dx = -x^{\beta \cdot \alpha^{\beta}|\alpha} \\ = 0 + \alpha^{-\beta} \cdot \alpha^{\beta} = 1 \tag{1.3.1.1}$$

(b)

$$Ex = \int_{\alpha}^{\infty} xf(x)dx = \int_{\alpha}^{\infty} \frac{\beta \cdot \alpha^{\beta}}{X^{\beta}} dx \\ = \frac{\beta}{-\beta+1} \cdot \alpha^{\beta} \cdot x^{-\beta+1}|_{\alpha}^{\infty} = \frac{\beta \cdot \alpha}{\beta - 1} \tag{1.3.1.2}$$

$$Ex^2 = \int_{\alpha}^{\infty} \frac{\beta \cdot \alpha^{\beta}}{x^{\beta-1}} dx = \frac{\beta}{-\beta+2} \cdot \alpha^{\beta} \cdot x^{-\beta+2}|_{\alpha}^{\infty} \\ = \frac{\alpha^2 \beta}{\beta - 2} \tag{1.3.1.3}$$

$$\operatorname{Var} X = EX^2 - (EX)^2 = \frac{\beta \alpha^2}{(\beta-1)^2(\beta-2)} \tag{1.3.1.4}$$

(c)

If $\beta < 2$, then the variance is negative.

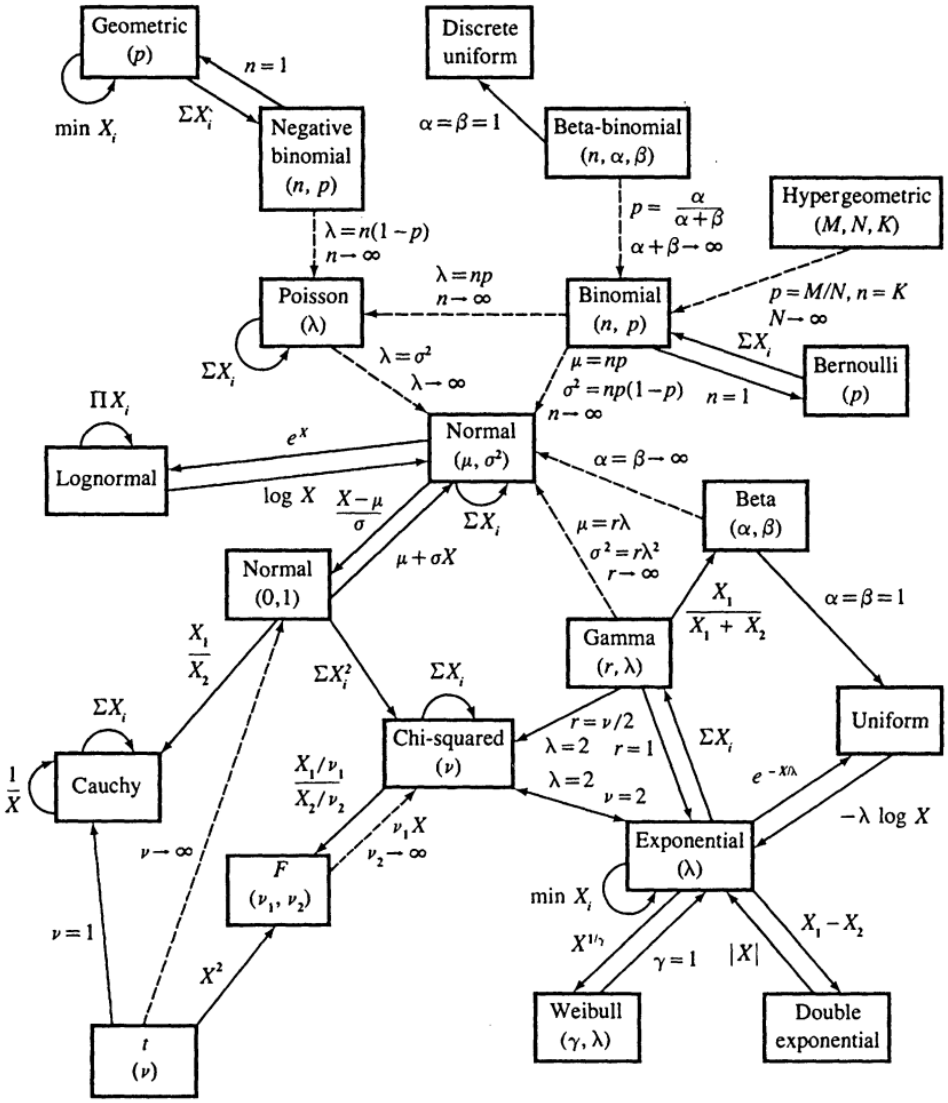


Figure 1.4: Type of distribution

1.4 Statistical Inference

This section is mainly the notes from [Casella and Berger, 2021]

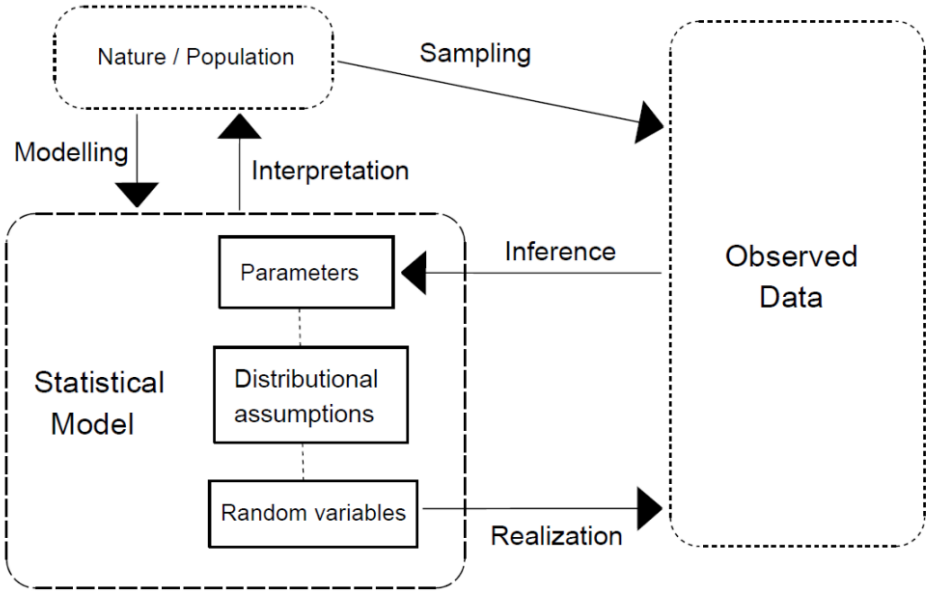


Figure 1.5: The scope of statistical inference

Statistics uses observed data to inference the statistical model.

1.4.1 Exponential Families

Definition 1.4.1.1: Exponential Families

A family of pdfs or pmfs is called an *exponential family* if:

$$f(x|\boldsymbol{\theta}) = h(x)c(\boldsymbol{\theta}) \exp\left(\sum_{i=1}^{k} w_i(\boldsymbol{\theta})t_i(x)\right) \tag{1.4.1.1}$$

Where $h(x) \leq 0, c(\boldsymbol{\theta}) \leq 0$, and $h(x), t_i(x)$ don't depend on $\boldsymbol{\theta}$. $c(\boldsymbol{\theta}), w_i(\boldsymbol{\theta})$ don't depend on x.

- Continuous: normal, gamma, beta, exponential;
- Discrete: binomial, poisson, nagative binomial;
- $\boldsymbol{\theta} = \theta_1, \theta_2, , \theta_d$, k must $\geq d$;
- If $k = d$, then it's a *full exponential family*, if $k > d$, then it's a *curved exponential family* (For example, most normal distributions are *full exponential family*, but normal distribution satisfy $\mu = \sigma^2$ is a *curved exponential family*).

To verify a pdf is an exponential family, identify the function $h(x), c(\boldsymbol{\theta}), t_i(x), w_i(\boldsymbol{\theta})$, then verify these functions satisfy the condition above.

Example 1.4.1.1

Show that Binomial, Poisson, Exponential and Normal distribution belongs to the exponential families.

Solution:

Binomial:

$$\begin{aligned} f(x|p) &= \binom{n}{x} p^x (1-p)^{n-x} = \binom{n}{x} (1-p)^n \left(\frac{p}{1-p}\right)^x \\ &= \binom{n}{x} (1-p)^n \exp\left(x \log\left(\frac{p}{1-p}\right)\right), \end{aligned} \tag{1.4.1.2}$$

Among which $\binom{n}{x}$ *is* $h(x)$, $(1-p)^n$ *is* $c(\theta)$, x *is* $t_1(x)$, *and* $\log\left(\frac{p}{1-p}\right)$ *is* $w_i(\theta)$. *Note that* $f(x \mid p)$ *is only in exponential families when* $0 < p < 1$.

Poisson:

$$\begin{aligned} & f(x|\lambda) = \frac{\lambda^x e^{-\lambda}}{x!} = \frac{1}{x!} e^{-\lambda} \exp\left(x \log(\lambda)\right) \\ & \textit{then} \\ & h(x) = \frac{1}{x!}, c(\lambda) = e^{-\lambda}, t(x) = x \text{ and} w(\lambda) = \log(\lambda). \end{aligned} \tag{1.4.1.3}$$

Exponential:

$$\begin{aligned} & f(x|\beta) = \frac{1}{\beta} \exp\left(-\frac{x}{\beta}\right) \\ & \textit{then} \\ & h(x) = 1, c(\beta) = \frac{1}{\beta}, t(x) = x \text{ and} w(\beta) = -\frac{1}{\beta}. \end{aligned} \tag{1.4.1.4}$$

Normal:

$$\begin{aligned} & f(x|\mu, \sigma^2) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{x^2}{2\sigma^2} + \frac{x\mu}{\sigma^2} - \frac{\mu^2}{2\sigma^2}\right) \\ & \textit{then} \\ & h(x) = 1, c(\mu, \sigma) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{\mu^2}{2\sigma^2}\right), \\ & t_1(x) = -\frac{x^2}{2}, w_1(\mu, \sigma) = \frac{1}{\sigma^2}, t_2(x) = x \text{ and } w_2(\mu, \sigma) = \frac{\mu}{\sigma^2}. \end{aligned} \tag{1.4.1.5}$$

Example 1.4.1.2

Show the following are exponential families:

- Gamma family with either α, β is unknown or both unknown;
- Beta family with either α, β is unknown or both unknown;
- Negative Binomial family when r is unkown.

Solution:

Gamma α unkown:

$$f(x|\alpha;\beta) = e^{-x/\beta}\frac{1}{\Gamma(\alpha)\beta^{\alpha}}\exp((\alpha-1)\log x) \tag{1.4.1.6}$$

thus $h(x) = e^{-x/\beta}, x > 0; c(\alpha) = \frac{1}{\Gamma(\alpha)\beta^{\alpha}}; w_1(\alpha) = \alpha - 1; t_1(x) = \log x.$

Gamma beta unknown:

$$f(x|\beta;\alpha) = \frac{1}{\Gamma(\alpha)\beta^{\alpha}}x^{\alpha-1}e^{\frac{-x}{\beta}} \tag{1.4.1.7}$$

thus $h(x) = \frac{x^{\alpha-1}}{\Gamma(\alpha)}, x > 0; c(\beta) = \frac{1}{\beta^{\alpha}}; w_1(\beta) = \frac{1}{\beta}; t_1(x) = -x.$

Gamma both unknown:

$$f(x|\alpha,\beta) = \frac{1}{\Gamma(\alpha)\beta^{\alpha}}\exp((\alpha-1)\log x - \frac{x}{\beta}) \tag{1.4.1.8}$$

thus $h(x) = I_{\{x>0\}}(x); c(\alpha,\beta) = \frac{1}{\Gamma(\alpha)\beta^{\alpha}}; w_1(\alpha) = \alpha - 1; t_1(x) = \log x; w_2(\alpha,\beta) = -1/\beta; t_2(x) = x.$

Beta α unknown:

$h(x) = (1-x)^{\beta-1}I_{[0,1]}(x), \quad c(\alpha) = \frac{1}{B(\alpha,\beta)}, \quad w_1(x) = \alpha - 1, \quad t_1(x) = \log x$

Beta β unkown:

$h(x) = x^{\alpha-1}I_{[0,1]}(x), c(\beta) = \frac{1}{B(\alpha,\beta)}, w_1(\beta) = \beta - 1, t_1(x) = \log(1-x)$

Beta both unknown:

$h(x) = I_{[0,1]}(x), c(\alpha,\beta) = \frac{1}{B(\alpha,\beta)}, w_1(\alpha) = \alpha - 1, t_1(x) = \log x, w_2(\beta) = \beta - 1, t_2(x) = \log(1-x).$

Negative Binomial:

$$h(x) = \binom{r+x-1}{x}I_{\mathbb{N}}(x), \quad c(p) = \left(\frac{p}{1-p}\right)^{r}, \quad w_1(p) = \log(1-p), \quad t_1(x) = x. \tag{1.4.1.9}$$

Theorem 1.4.1.1: Expectation and Variance of Exponential Families

If X is a random variable that satisfies any distribution from the exponential families, then:

$$\begin{aligned} &1.\quad \mathrm{E}\left(\sum_{i=1}^{k}\frac{\partial w_i(\boldsymbol{\theta})}{\partial\theta_j}t_i(\mathrm{X})\right) = -\frac{\partial}{\partial\theta_j}\log\left(c(\boldsymbol{\theta})\right); \\ &2.\quad \mathrm{Var}\left(\sum_{i=1}^{k}\frac{\partial w_i(\boldsymbol{\theta})}{\partial\theta_j}t_i(\mathrm{X})\right) = -\frac{\partial^2}{\partial\theta_j^2}\log\left(c(\boldsymbol{\theta})\right) - \mathrm{E}\left(\sum_{i=1}^{k}\frac{\partial^2 w_i(\boldsymbol{\theta})}{\partial\theta_j^2}t_i(\mathrm{X})\right). \end{aligned} \tag{1.4.1.10}$$

Example 1.4.1.3

Derive the mean and variance for binomial and normal distribution using the above theorem.

Solution:

Binomial:

$$h(x) = \binom{n}{x}, c(p) = (1-p)^n, t(x) = x \text{and} w(p) = \log\left(\frac{p}{1-p}\right).$$

Then,

$$\begin{aligned}
&\frac{\mathrm{d}}{\mathrm{d}p} w(p) = \frac{\mathrm{d}}{\mathrm{d}p} \log\left(\frac{p}{1-p}\right) = \frac{1}{p(1-p)}, \\
&\frac{\mathrm{d}^2}{\mathrm{d}p^2} w(p) = -\frac{1}{p^2} + \frac{1}{(1-p)^2} = \frac{2p-1}{p^2(1-p)^2}, \\
&\frac{\mathrm{d}}{\mathrm{d}p} \log\big(c(p)\big) = \frac{\mathrm{d}}{\mathrm{d}p} n \log(1-p) = -\frac{n}{1-p}, \\
&\frac{\mathrm{d}^2}{\mathrm{d}p^2} \log\big(c(p)\big) = -\frac{n}{(1-p)^2}.
\end{aligned} \tag{1.4.1.11}$$

Therefore, from Theorem 3.4.2, we have

$$\mathrm{E}\left(\frac{1}{p(1-p)}\mathrm{X}\right) = \frac{n}{1-p} \Rightarrow \mathrm{E}(\mathrm{X}) = np,$$

$$\mathrm{Var}\left(\frac{1}{p(1-p)}\mathrm{X}\right) = \frac{n}{(1-p)^2} - \mathrm{E}\left(\frac{2p-1}{p^2(1-p)^2}\mathrm{X}\right) \Rightarrow \mathrm{Var}(\mathrm{X}) = np(1-p)$$

***Normal**:*

For Normal Distribution, we have

$$h(x) = 1, c(\mu, \sigma) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{\mu^2}{2\sigma^2}\right),$$

$$t_1(x) = -\frac{x^2}{2}, w_1(\mu, \sigma) = \frac{1}{\sigma^2}, t_2(x) = x \text{and} w_2(\mu, \sigma) = \frac{\mu}{\sigma^2}.$$

Then,

$$\begin{aligned}
&\frac{\partial w_1(\mu,\sigma)}{\partial \mu} = \frac{\partial(1/\sigma^2)}{\partial \mu} = 0, \\
&\frac{\partial w_2(\mu,\sigma)}{\partial \mu} = \frac{\partial(\mu/\sigma^2)}{\partial \mu} = \frac{1}{\sigma^2}, \\
&\frac{\partial w_1(\mu,\sigma)}{\partial \sigma} = \frac{\partial(1/\sigma^2)}{\partial \sigma} = -\frac{2}{\sigma^3}, \\
&\frac{\partial w_2(\mu,\sigma)}{\partial \sigma} = \frac{\partial(\mu/\sigma^2)}{\partial \sigma} = -\frac{2\mu}{\sigma^3}, \\
&\frac{\partial}{\partial \mu} \log\big(c(\mu,\sigma)\big) = \frac{\partial}{\partial \mu}\left(-\frac{\log(2\pi)}{2} - \log(\sigma) - \frac{\mu^2}{2\sigma^2}\right) = -\frac{\mu}{\sigma^2}, \\
&\frac{\partial}{\partial \sigma} \log\big(c(\mu,\sigma)\big) = \frac{\partial}{\partial \sigma}\left(-\frac{\log(2\pi)}{2} - \log(\sigma) - \frac{\mu^2}{2\sigma^2}\right) = -\frac{1}{\sigma} + \frac{\mu^2}{\sigma^3}.
\end{aligned} \tag{1.4.1.12}$$

$$\mathrm{E}\left(\frac{1}{\sigma^2}\mathrm{X}\right) = \frac{\mu}{\sigma^2} \text{ and } \mathrm{E}\left(-\frac{2}{\sigma^3}\left(-\frac{\mathrm{X}^2}{2}\right) - \frac{2\mu}{\sigma^3}\mathrm{X}\right) = \frac{1}{\sigma} - \frac{\mu^2}{\sigma^3},$$

which implies

$$\mathrm{E}(\mathrm{X}) = \mu, \mathrm{E}(\mathrm{X}^2) = \mu^2 + \sigma^2 \text{and} \mathrm{Var}(\mathrm{X}) = \mathrm{E}(\mathrm{X}^2) - (\mathrm{EX})^2 = \sigma^2$$

Definition 1.4.1.2: The indicator function

$$I_A(x) = \begin{cases} 1, & \text{if } x \in A \\ 0, & \text{if } x \notin A \end{cases}$$

Example 1.4.1.4

Show that $f(x \mid \theta) = \frac{1}{\theta} exp(1 - \frac{X}{\theta})$ is **NOT** an exponential family.

Solution:

$$f(x|\theta) = \frac{1}{\theta} \exp\left(1 - \frac{x}{\theta}\right) I_{[\theta,\infty)}(x) \tag{1.4.1.13}$$

At here $h(x) = I_{[\theta,\infty)}(x)$, which is not independent with θ.

Definition 1.4.1.3: Reparameterization of Exponential Families

$$f(x|\boldsymbol{\eta}) = h(x)c^*(\boldsymbol{\eta}) \exp\left(\sum_{i=1}^{k} \eta_i t_i(x)\right) \tag{1.4.1.14}$$

Where $h(x)$ and $t_i(x)$ are identical with the original parameterization. $\boldsymbol{\eta} = (\eta_1, \eta_2, \cdots, \eta_n)$ and $\eta_i = w_i(\theta)$. And to make it a pdf (integrates to 1):

$$c^*(\boldsymbol{\eta}) = \left[\int_{-\infty}^{\infty} h(x) \exp\left(\sum_{i=1}^{k} \eta_i t_i(x)\right) dx\right]^{-1} \tag{1.4.1.15}$$

1.4.2 Scale and Location

step 1: define the standard pdf $f(Z)$, for example: $f(Z) = e^{-Z}, Z \geq 0$;
step 2: find the relationship between X and X: $\sigma Z + \mu = X$, where σ is the scale parameter and μ is the location parameter;
step 3: replace Z using X: $f(x) = \frac{1}{\sigma} f(\frac{x-\mu}{\sigma})$ is a distribution transformed from the standard pdf.

Theorem 1.4.2.1: Sacle-Location Family

If f(x) is any pdf, for $\mu, \sigma > 0$, then the function $g(x \mid \mu, \sigma) = \frac{1}{\sigma} f(\frac{x-\mu}{\sigma})$ is a valid pdf.

Proof 1.4.1

$$g(x|\mu,\sigma) = \frac{1}{\sigma} f\left(\frac{x-\mu}{\sigma}\right) \geq 0$$

$$\int_{-\infty}^{\infty} g(x|\mu,\sigma)dx = \int_{-\infty}^{\infty} \frac{1}{\sigma} f\left(\frac{x-\mu}{\sigma}\right) dx \xrightarrow{\left(y=\frac{x-\mu}{\sigma}\right)} \int_{-\infty}^{\infty} f(y)dy = 1. \tag{1.4.2.1}$$

Q.E.D.

Remark 1.4.1

- For discrete R.V.(pmf), the above theorem doesn't hold. (ignore the $\frac{1}{\sigma}$);
- The σ is the scale parameter, the μ is the location parameter.

Example 1.4.2.1

$$\int_{\mathbf{X}}(x_1 \mid \theta) = \frac{1}{\theta}\exp\{1-\frac{x}{\theta}\}, \quad x \geq \theta \tag{1.4.2.2}$$

$$= \frac{1}{\theta}\exp\{-\frac{x-\theta}{\theta}\} \cdot \mathrm{I}_{[0,\infty)}(x) \tag{1.4.2.3}$$

Is θ a scale parameter or a location parameter?

Solution:

It depends on the standard pdf:
If the standard pdf is $f_Z(z) = \exp\{-z\} \cdot \mathcal{I}_{[0,+\infty)}(z)$*:*
then θ *serves as both parameters because* $f_X(x) = \frac{1}{\theta}\exp\{-\frac{x-\theta}{\theta}\} \cdot I_{[0,+\infty)}(\frac{x-\theta}{\theta})$.
Else if the standard pdf is $f_Z(z) = \exp\{1-z\} \cdot \mathcal{I}_{[0,+\infty)}(z)$*:*
then θ *is only a scale parameter because* $f_X(x) = \frac{1}{\theta}\exp\{1-\frac{x}{\theta}\} \cdot I_{[0,+\infty)}(\frac{x-\theta}{\theta})$.

Theorem 1.4.2.2: For any pdf $f(\cdot)$, and $\sigma, \mu > 0$. Then X is a random variable with pdf $\frac{1}{\sigma}f\left(\frac{x-\mu}{\sigma}\right)$ **if and only if** there is a random variable Z with pdf $f(z)$ and $X = \sigma Z + \mu$.

Proof 1.4.2: Rigorous Proof

The necessity of the proof can be found in 1.4.2, here the sufficiency need
Q.E.D.

Proof 1.4.3: Intuitive Proof

Q.E.D.

1.4.3 Data Reduction

The idea of data reduction is to summarize or reduce the data $X_1, X_2, \cdots, X_n$ to get the information of the unknown parameter θ.
There are many sample point $\mathbf{x} = (x_1, x_2, \cdots, x_n)$, which are realizations (observations) of the random variable $\mathbf{X} = (X_1, X_2, \cdots, X_n)$. A **Statistic** $T(\mathbf{X})$ is a form of data reduction, or a summary of the data, $T(\mathbf{x})$ is an observation of $T(\mathbf{X})$. $\mathcal{X}$ is the **sample space**. $\mathcal{T} = \{t : t = T(\mathbf{x}),\ \text{for } \mathbf{x} \in \mathcal{X}\}$ is the image of cX under $T(\mathbf{X})$. $T(\mathbf{X})$ partition the sample space $\mathcal{X}$ into sets $A_t = \{\mathbf{x} : T(\mathbf{x}) = t, \mathbf{x} \in \mathcal{X}\}$.

Example 1.4.3.1

Give a two-dimensional example for the random variable, sample point, Statistic, observation of Statistic, sample space, image and A_t.

Solution:

Assume $\mathbf{X} = X_1, X_2$, *among which* X_1, X_2 *are Bernoulli R.V. with* $p = 0.5$. $\mathbf{x} = (0,1)$ *is a sample point. The sample space* $\mathcal{X}$ *is* $(0,0),(0,1),(1,0),(1,1)$, *set* $T(\mathbf{X})$ *as the statistic, then the image* $\mathcal{T}$ *is* $(0,1,2)$. $A_1(\mathbf{x}) = ((1,0),(0,1))$, $A_2(\mathbf{x}) = (1,1)$, $A_0(\mathbf{x}) = (0,0)$.

1.4.4 Moment Generating Function

The **Moment Generating Function** (mgf) is:

$$M_X(t) = \mathbb{E}(e^{tX}). \tag{1.4.4.1}$$

If taking the n order derivatives of the mgf by t, and let $t = 0$, then:

$$M_X^{(n)}(t)|_{t=0} = \mathbb{E}\left(X^n\right). \tag{1.4.4.2}$$

1.4.5 Concentration Inequality

Averages of independent random variables concentrate around their expectations.

Theorem 1.4.5.1: Law of Large Numbers

Suppose $X_1, X_2, \cdots, X_n$ are *i.i.d. RVs* with mean μ, then

$$\lim_{n\to\infty} \bar{X}_n = \frac{1}{n}\sum_{i=1}^{n} X_i = \mu$$

However, LLN does not provide the rate of convergence. It doesn't require the finite variance (weakest assumption among similar results).

Theorem 1.4.5.2: Central Limit Theorem

Suppose $X_1, X_2, \cdots, X_n$ are *i.i.d. RVs* with mean μ and variance σ^2, then

$$\frac{\sqrt{n}(\bar{X}_n - \mu)}{\sigma} \xrightarrow{d} N(0,1)$$

It implies:

$$\mathrm{P}\left(\frac{\sqrt{n}(\bar{X}_n - \mu)}{\sigma}\right) \to \Phi(x)$$

Still, the result holds asymptotically (not useful if you want to analyze a finite n).

☺ Denote $\overline{X} = \frac{1}{n}\sum_{i=1}^{n} X_i$, where $X_1, \cdots, X_n$ are generated *i.i.d.*, the general form of concentration inequality:

$$\mathbb{P}(|\overline{X} - \mathbb{E}[\overline{X}]| \geq \varepsilon(n)) \leq 1 - \delta(n) \tag{1.4.5.1}$$

where $\varepsilon(n)$ and $\delta(n)$ converge to 0 when $n \to \infty$.

Theorem 1.4.5.3: Markov Inequality

For a positive random variable X:

$$\mathbb{P}(X \geq t) \leq \frac{\mathbb{E}[X]}{t} = O\left(\frac{1}{t}\right). \tag{1.4.5.2}$$

Proof 1.4.4

$$\mathbb{E}[X] = \int_0^\infty xp(x)dx \geq \int_t^\infty xp(x)dx \geq t\int_t^\infty p(x)dx = t\mathbb{P}(X \geq t). \tag{1.4.5.3}$$

Q.E.D.

Theorem 1.4.5.4: Chebyshev's inequality

Let $Var(X) = \sigma^2$, then:

$$\mathbb{P}\Big(|X - \mathbb{E}[X]| \geq t\sigma\Big) \leq \frac{1}{t^2} = O\left(\frac{1}{t^2}\right). \tag{1.4.5.4}$$

Proof 1.4.5

$$\begin{aligned}\mathbb{P}(|X - \mathbb{E}[X]| \geq t\sigma) &= \mathbb{P}(|X - \mathbb{E}[X]|^2 \geq t^2\sigma^2) \\ &\leq \frac{\mathbb{E}(|X - \mathbb{E}[X]|^2)}{t^2\sigma^2} = \frac{1}{t^2}.\end{aligned}$$

Q.E.D.

Corollary 1.4.5.1

If we consider $\widehat{\mu}_n = \frac{1}{n}\sum_{i=1}^n X_i$ with variance $\frac{\sigma^2}{n}$, applying the Chebyshev's inequality would get:

$$\mathbb{P}\left(|\widehat{\mu}_n - \mu| \geq \frac{t\sigma}{\sqrt{n}}\right) \leq \frac{1}{t^2}. \tag{1.4.5.5}$$

Lemma 1.4.5.1: Chernoff Method

😎 This trick is extremely important.

Assume the mgf for X is finite for all $|t| \leq b, b > 0$. Then, we can use mgf to produce a tail bound by Markov inequality:

$$\mathbb{P}((X - \mu) \geq u) = \mathbb{P}(\exp(t(X)) \geq \exp(t(u + \mu))) \leq \frac{\mathbb{E}[\exp(tX)]}{\exp(t(u + \mu))} \tag{1.4.5.6}$$

Now consider t as a parameter. To achieve the tightest bound, we can tune t that satisfies:

$$\mathbb{P}((X - \mu) \geq u) \leq \inf_{0 \leq t \leq b} \exp(-t(u + \mu))\mathbb{E}[\exp(tX)]. \tag{1.4.5.7}$$

This is called the **Chernoff Bound**. The Chernoff bound "plays" nicely with summations, assuming X_i are independent:

$$M_{X_1+\cdots+X_n}(\lambda) = \prod_{i=1}^n M_{X_i}(\lambda), \tag{1.4.5.8}$$

This means that when we calculate a Chernoff bound of a sum of i.i.d. variables, we only need to calculate

the moment generating function for *one* of them, assume n zero-mean independent variables:

$$\begin{aligned}\mathbb{P}\left(\sum_{i=1}^{n} X_i \geq t\right) &\leq \frac{\prod_{i=1}^{n} \mathbb{E}\left[\exp(\lambda X_i)\right]}{e^{\lambda t}} \\ &= (\mathbb{E}[e^{\lambda X_1}])^n e^{-\lambda t},\end{aligned} \tag{1.4.5.9}$$

Theorem 1.4.5.5: Gaussisn Tail Bound

Suppose $X \sim N(\mu, \sigma^2)$, the mgf is $M_X(t) = \mathbb{E}[\exp(tX)] = \exp(t\mu + t^2\sigma^2/2)$, to apply the Chernoff bound we can compute:

$$\inf_{t\geq 0} \exp(-t(u+\mu)) \exp(t\mu + t^2\sigma^2/2) = \inf_{t\geq 0} \exp(-tu + t^2\sigma^2/2), \tag{1.4.5.10}$$

when $t = \frac{u}{\sigma^2}$, we can obtain:

$$\mathbb{P}(X - \mu \geq u) \leq \exp(-u^2/(2\sigma^2)). \tag{1.4.5.11}$$

This is just the right side of the distribution, taking the absolute value:

$$\mathbb{P}(|X - \mu| \geq u) \leq 2\exp(-u^2/(2\sigma^2)). \tag{1.4.5.12}$$

Now consider $\overline{X}$, we can obtain:

$$\mathbb{P}(|\widehat{\mu} - \mu| \geq t\sigma/\sqrt{n}) \leq 2\exp(-t^2/2). \tag{1.4.5.13}$$

Remark 1.4.2

Comparing the Gaussian tail bound with the corollary of Chebyshev's inequality, we can find with probability $1 - \delta$:

- Chebyshev tells that $|\widehat{\mu} - \mu| \leq \frac{\sigma}{\sqrt{n\delta}}$;
- Gaussian tells that $|\widehat{\mu} - \mu| \leq \sigma\sqrt{\frac{2\ln(2/\delta)}{n}}$, which decrease faster when $n \to \infty$.

Theorem 1.4.5.6: Bounded RVs: Hoeffding's inequality

Lemma 1.4.5.2: Hoeffding's Lemma

Let X be a random variable bounded in $[a,b]$, let X' to be an independent copy of X, then:

$$\mathbb{E}_X[\exp(\lambda(X-\mathbb{E}_X[X]))] = \mathbb{E}_X[\exp(\lambda(X-\mathbb{E}_{X'}[X']))] \overset{(i)}{\leq} \mathbb{E}_X[\mathbb{E}_{X'}\exp(\lambda(X-X'))], \tag{1.4.5.14}$$

Where i follows Jensen's inequality. Let $S \in \{-1,1\}$ to be a random sign variable, then $S(X-X')$ has the same distribution as $X-X'$:

$$\begin{aligned}\mathbb{E}_{X,X'}[\exp(\lambda(X-X'))] &= \mathbb{E}_{X,X',S}[\exp(\lambda S(X-X'))] \\ &= \mathbb{E}_{X,X'}[\mathbb{E}_S[\exp(\lambda S(X-X')) \mid X, X']]. \\ &\leq \exp\frac{\lambda^2(X-X')^2}{2} \\ &\leq \exp\frac{\lambda^2(b-a)^2}{2}\end{aligned} \tag{1.4.5.15}$$

Now use the Chernoff bound and Hoeffding's lemma to show:

$$\begin{aligned}\mathbb{P}\left(\frac{1}{n}\sum_{i=1}^n(X_i-\mathbb{E}[X_i]) \geq t\right) &= \mathbb{P}\left(\sum_{i=1}^n(X_i-\mathbb{E}[X_i]) \geq nt\right) \\ &\leq \mathbb{E}\left[\exp\left(\lambda\sum_{i=1}^n(X_i-\mathbb{E}[X_i])\right)\right]e^{-\lambda nt} \\ &= \left(\prod_{i=1}^n \mathbb{E}[e^{\lambda(X_i-\mathbb{E}[X_i])}]\right)e^{-\lambda nt} \leq \left(\prod_{i=1}^n e^{\frac{\lambda^2(b-a)^2}{8}}\right)e^{-\lambda nt}\end{aligned} \tag{1.4.5.16}$$

Rewriting this and minimize over $\lambda \geq 0$, we have

$$\mathbb{P}\left(\frac{1}{n}\sum_{i=1}^n(X_i-\mathbb{E}[X_i]) \geq t\right) \leq \min_{\lambda\geq 0}\exp\left(\frac{n\lambda^2(b-a)^2}{8}-\lambda nt\right) = \exp\left(-\frac{2nt^2}{(b-a)^2}\right) \tag{1.4.5.17}$$

1.4.6 Convergence

1.5 Theory of Optimization

1.5.1 NP-Hard

Definition 1.5.1.1: Polynomial Time Algorithms

For an input with size n, the worst-case running time is $O(n^k)$ for some constant k. A problem that is solvable by polynomial-time algorithms is said to be **tractable, easy, efficient**.

Decision Problem V.S. Optimization Problem

Decision problems simply need to answer 'yes' or 'no'; optimization needs to decide a specific value. Every optimization problem has a decision version that is no harder than the optimization problem.

Example 1.5.1.1

A_{opt}: given a graph, find the length of the shortest path.
A_{dec}: given a graph, determine whether there is a path $\leq k$.

Solution:

Using A_{opt} to solve A_{dec}: check if optimal value $\leq k$.
Using A_{dec} to solve A_{opt}: apply binary search on the value range, logarithmic at most.

P and NP

1. Class P: problems that can be solved in $O(n^k)$;
2. Class NP: problems that can be verified in $O(n^k)$ ($\mathbb{P} \subseteq \mathbb{NP}$);
3. Class NP-hard: problems that are "at least as hard as all NP problems";
4. Class NP-complete: problems in both NP and NP-hard.

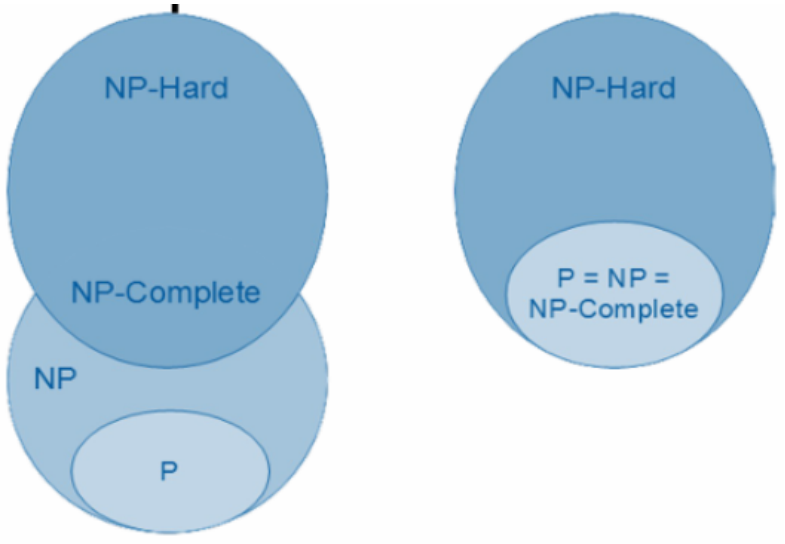


Figure 1.6: P and NP

Polynomial-Time Reduction

A reduction is an algorithm for **transforming a problem instance into another**.

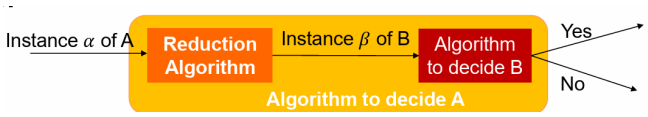


Figure 1.7: Polynomial-Time Reduction

Remark 1.5.1

- $A \leq_p B$ if A can be reduced to B **in polynomial time**;
- Reduction from A to B implies A is no harder than B, so if A is hard, then so is B;
- One way to prove two problems have equal difficulty is to prove $A \leq_p B$ and $B \leq_p A$.

1.5.2 Convexity and Continuity

Theorem 1.5.2.1: Jensen's Inequality

If $f(x)$ is convex over x, then:

$$f(\mathbb{E}[X]) \leq \mathbb{E}[f(X)],$$

Vice versa.

Definition 1.5.2.1: Local Minimum (Relative Minimum)

A point $x^* \in \Omega$ is said to be a local minimum point of f over Ω if there is an $\epsilon > 0$, such that for any $x \in \Omega$ and $|x - x^*| < \epsilon$, we have $f(x) \geq f(x^*)$.

Definition 1.5.2.2: Differentiability

A mapping $F : \mathbb{R}^n \to \mathbb{R}^m$ is said to be *differentiable* at x if there is a function: $DF : \mathbb{R}^n \to \mathbb{R}^{m \times n}$ such that:

$$\lim_{h \to 0} \frac{\|F(x+h) - F(x) - DF(x) \cdot h\|}{\|h\|} = 0. \tag{1.5.2.1}$$

The matrix $DF(x) \in \mathbb{R}^{m \times n}$ is the Jacobian Matrix.

If F is a $\mathbb{R}^n \to \mathbb{R}$ mapping, then the **gradient** can be expressed as:

$$\nabla f(x) = Df(x)^\top = \begin{pmatrix} \frac{\partial}{\partial x_1} f(x) \\ \vdots \\ \frac{\partial}{\partial x_n} f(x) \end{pmatrix} \tag{1.5.2.2}$$

The **Hessian Matrix** (symmetric matrix) can be expressed by:

$$\nabla^2 f(x) = H_f(x) = \begin{pmatrix} \frac{\partial f}{\partial x_1 \partial x_1}(x) & \frac{\partial f}{\partial x_1 \partial x_2}(x) & \cdots & \frac{\partial f}{\partial x_1 \partial x_n}(x) \\ \cdot & \cdot & \cdots & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdots & \cdot \\ \frac{\partial f}{\partial x_n \partial x_1}(x) & \frac{\partial f}{\partial x_n \partial x_2}(x) & \cdots & \frac{\partial f}{\partial x_n \partial x_n}(x) \end{pmatrix} \tag{1.5.2.3}$$

Example 1.5.2.1

Calculate the gradient of $f : \mathbb{R}^n \to \mathbb{R}, \quad f(x) = \frac{1}{2} x^\top A x + b^\top x + c$, where $A \in \mathbb{R}^{n \times n}$ be symmetric.

Solution:

Use the definition of Differentiability, first calculate $f(x+h) - f(x)$:

$$\begin{aligned} f(x+h) - f(x) &= \frac{1}{2}(x+h)^T A(x+h) + b^T(x+h) + c - f(x) \\ &= (x+h)^T Ax + (x+h)^T Ah + b^T x + b^T h - \frac{1}{2} x^\top A x + b^\top x \\ &= \frac{1}{2} h^T Ax + \frac{1}{2} x^T Ah + \frac{1}{2} h^T Ah + b^T h \\ &= \frac{1}{2}(h^T Ax)^T + \frac{1}{2} x^T Ah + \frac{1}{2} h^T Ah + b^T h \\ &= X^T Ah + b^T h \end{aligned} \tag{1.5.2.4}$$

Thus $\nabla f(x) = Ax + b$.
If A is not symmetric, then $\nabla f(x) = \frac{A^T + A}{2} x + b$.

First and Second Order Conditions

Remark 1.5.2

FONC (First-Order Necessary Conditions)
If $x^\star$ is a local minimizer of the unconstrained problem, then we must have $\nabla f(x^*) = 0$.

Example 1.5.2.2

We want to use a nth-order polynomial function to estimate the original function $g(x)$ after we observe m points: $x_1, x_2, \cdots, x_m$:

$$h(x) = a_n x^n + a_{n-1} x^{n-1} + \ldots + a_0, n < m \tag{1.5.2.5}$$

Solution:

This can be transformed into an unconstrained problem:

$$\min_{\mathbf{a}} f(\mathbf{a}) = \sum_{k=1}^{m} [g(x_k) - (a_n x_k^n + a_{n-1} x_k^{n-1} + \ldots + a_0)]^2 \tag{1.5.2.6}$$

We define $q_{ij} = \sum_{k=1}^{m}(x_k)^{i+j}$, $b_j = \sum_{k=1}^{m} g(x_k)(x_k)^j$, *and* $c = \sum_{k=1}^{m} g(x_k)^2$. *With some algebra, we can get:*

$$f(\mathbf{a}) = \mathbf{a}^T\mathbf{Q}\mathbf{a} - 2\mathbf{b}^T\mathbf{a} + c \tag{1.5.2.7}$$

The FONC can be reduced to $Qa = b$.

Theorem 1.5.2.2: SONC (Second Order Necessary Condition)

If x^* is a local minimizer of f, then it holds that:

1. $\nabla f(x^*) = 0$;
2. For all $d \in \mathbb{R}^n$:$d^\top \nabla^2 f(x^*) d \geq 0$ ($\nabla^2 f(x^*)$ is positive semidefinite);

Definition 1.5.2.3: Saddle Point

A point satisfying FONC is a **stationary point**; a stationary point with an indefinite Hessian matrix is called **saddle point**.

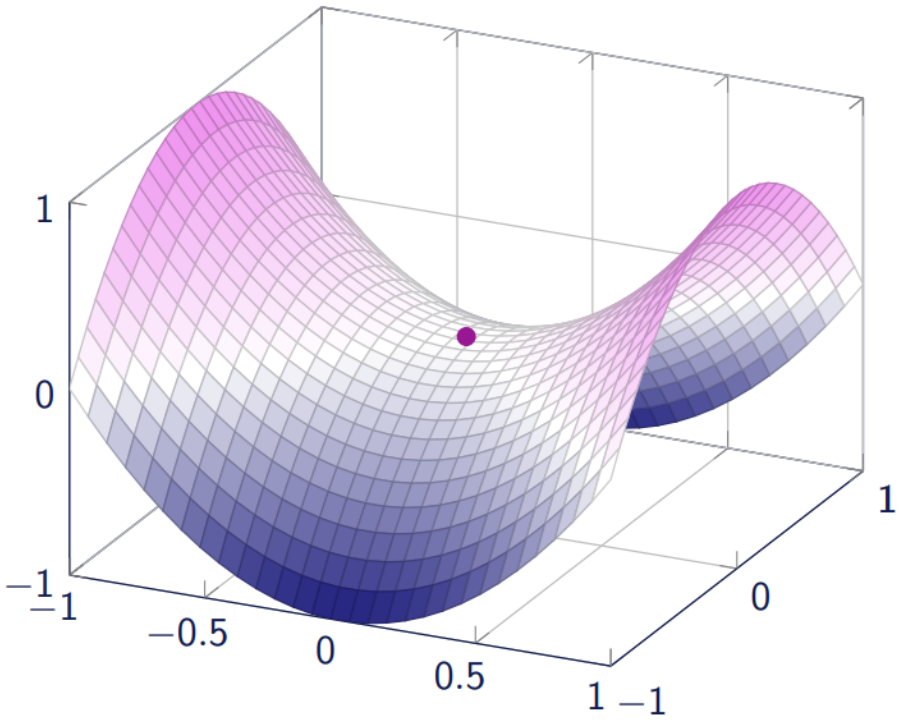


Figure 1.8: An example of saddle point

Theorem 1.5.2.3: SOSC (Second Order Sufficient Conditions)

1. $\nabla f(x^*) = 0$;
2. For all $d \in \mathbb{R}^n$:$d^\top \nabla^2 f(x^*) d > 0$ ($\nabla^2 f(x^*)$ is positive definite);

Then x^* is a *strict local minimum* of f.

Proof 1.5.1

By *taylor expansion*, $f(x^* + td) = f(x^*) + \frac{1}{2}t^2 d^\top \nabla^2 f(x^*)d + o(t^2) > f(x^*)$.
Q.E.D.

Definition 1.5.2.4: Coercivity

A continuous function $f : \mathbb{R}^n \to \mathbb{R}$ is said to be **coercive** if:

$$\lim_{\|x\| \to \infty} f(x) = +\infty \tag{1.5.2.8}$$

What's more, if f is a coercive function, then the level set $L_{\leq \alpha} := \{x \in \mathbb{R}^n : f(x) \leq \alpha\}$ is *compact* (closed and bounded), and has at least one *global minimizer* (This is called the Weierstrass Theorem).

Zero Order Conditions

Consider the constrained problem:

$$\begin{aligned} &\min f(x) \\ &s.t. x \in \Omega, x \in \mathbb{R}^n \end{aligned} \tag{1.5.2.9}$$

where f is convex, x^* is the optimal solution and f^* is the optimal value. Consider the epigraph $\Gamma \subset \mathbb{R}^{n+1} = \{(r, \mathbf{x}) : r \geqslant f(\mathbf{x}), \mathbf{x} \in \mathbb{R}^n\}$, and construct $B \in \mathbb{R}^{n+1}$ with cross section Ω and extending vertically from $-\infty$ up to f^*. B only overlaps Γ at the boundary point (f^*, x^*).

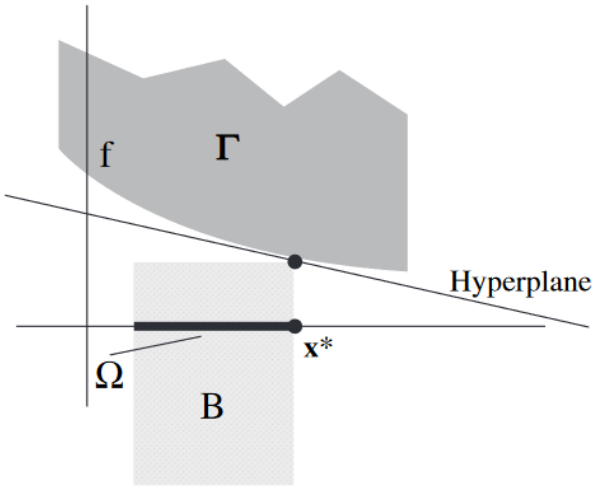


Figure 1.9: Zero Order Conditions

By the separating hyperplane theorem, there exists a vector $(s, \lambda) \in \mathbb{R}^{n+1}$ satisfy:

$$\begin{aligned} sr + \lambda^T \mathbf{x} \geq c \quad &\text{for all x} \in \mathbb{R}^n \text{ and } r \geq f(\mathbf{x}) \\ sr + \lambda^T \mathbf{x} \leq c \quad &\text{for all x} \in \Omega \text{ and } r \leq f^*. \end{aligned} \tag{1.5.2.10}$$

$s > 0$ otherwise equation 1.5.2.10 don't hold, so we can scale the equations by setting $s = 1$.

Proposition 1.5.2.1: Zero-order necessary conditions (**ZONC**)

If x^* solves equation 1.5.2.9, then there is a nonzero $\lambda \in \mathbb{R}^n$ that x^* is a solution to:

$$\begin{aligned} &\min f(x) + \lambda^T x \\ &s.t. \quad x \in \mathbb{R}^n \end{aligned} \tag{1.5.2.11}$$

And

$$\begin{aligned} &\max \lambda^T x \\ &s.t. \quad x \in \Omega \end{aligned} \tag{1.5.2.12}$$

Proof 1.5.2

The equation 1.5.2.11 and 1.5.2.12 follow from equation 1.5.2.10, where c is attained at the point f^*, x^* for both inequalities.

Q.E.D.

Remark 1.5.3

The FONC of 1.5.2.11 implies $\nabla f(x^*) = -\lambda$, and the signal of λ can give us the hint of the signal of x^* from 1.5.2.12.

Proposition 1.5.2.2: Zero-order sufficiency conditions **ZOSC**

If there is a λ such that $x^* \in \Omega$ solves equation 1.5.2.11 and 1.5.2.10, then x^* solves euqation 1.5.2.9.

Proof 1.5.3

Suppose x_1 is any other point in Ω, from equation 1.5.2.11:

$$f(x_1) + \boldsymbol{\lambda}^T x_1 \geqslant f(x^*) + \boldsymbol{\lambda}^T x^*. \tag{1.5.2.13}$$

This can be written as:

$$f(x_1) - f(x^*) \geqslant \boldsymbol{\lambda}^T x^* - \boldsymbol{\lambda}^T x_1. \tag{1.5.2.14}$$

By equation 1.5.2.12, we can get $f(x_1) \geq f(x^*)$

Q.E.D.

Remark 1.5.4

ZOSC don't require f to be convex.

Convexity

Definition 1.5.2.5: Convex Sets and Functions

Convex Sets

A set $X \subseteq \mathbb{R}^n$ is **convex** if for any $x, y \in X$, and any $\lambda \in [0, 1]$, we have $\lambda x + (1 - \lambda)y \in X$.

For example, the half space $H := \{x \in \mathbb{R}^n : a^\top x \leq b\}$ and the closed ball $B_r(a) := \{x \in \mathbb{R}^n : \|x - a\| \leq r\}$ are both convex sets.

Corollary 1.5.2.1: Intersection of Convex Sets

The intersection of convex sets is a convex set. For example, the *Polyhedral Sets*: $\{x \in \mathbb{R}^n : Ax \leq b\}$.

Convex Functions

A function f is said to be *convex* on a *convex sets* X is for every $x, y \in X$ and any $0 \leq \lambda \leq 1$:

$$f(\lambda x + (1-\lambda)y) \leq \lambda f(x) + (1-\lambda)f(y) \tag{1.5.2.15}$$

Examples are The Euclidean norm $f(x) = \|x\| = \sqrt{x^\top x}$ and affine-linear functions $f(x) = a^\top x + b$.

If a function f has the property $f(\lambda \mathbf{x} + (1-\lambda)\mathbf{y}) < \lambda f(\mathbf{x}) + (1-\lambda)f(\mathbf{y})$, then it's said to be **strictly convex**.

Strongly Convex (with parameter μ)

$$f(\lambda x + (1-\lambda)y) + \frac{\mu\lambda(1-\lambda)}{2}\|y-x\|^2 \leq \lambda f(x) + (1-\lambda)f(y) \tag{1.5.2.16}$$

This is equivalent to $f - \frac{\mu}{2}\|\cdot\|^2$.

Lemma 1.5.2.1: General Composition

Let $h : X \to \mathbb{R}$ be *convex* and $g : Y \to \mathbb{R}$ be *convex* and **non-decreasing**. Then $f(x) = g(h(x))$ is convex.

Operations that preserve convexity

😎 This part is extremely important!

- **Nonnegative weighted sums**;
- **Pointwise maximum**;

Composition with an affine mapping: Suppose $f : \mathbb{R}^n \to \mathbb{R}, A \in \mathbb{R}^{n\times m}$, and $b \in \mathbb{R}^n$. Define: $g : \mathbb{R}^m \to \mathbb{R}$ as

$$g(x) = f(Ax + b) \tag{1.5.2.17}$$

with $dom(g) = \{x \mid Ax + b \in dom(f)\}$. Then, if f is convex (concave), so is g.

Proof 1.5.4

Let $x_1, x_2 \in dom(g)$, $\alpha \in (0,1)$, we have:

$$\begin{aligned} g(\alpha x_1 + (1-\alpha x_2)) &= f(A(\alpha x_1 + (1-\alpha x_2)) + b) \\ &= f(\alpha(Ax_1 + b) + (1-\alpha)(Ax_2 + b)) \\ &\leq \alpha f(Ax_1 + b) + (1-\alpha)(Ax_2 + b) \\ &= \alpha g(x_1) + (1-\alpha)g(x_2) \end{aligned} \tag{1.5.2.18}$$

Q.E.D.

Composition of non-decreasing functions: Let $h : X \to \mathbb{R}$ be convex and **non-decreasing** and $g : Y \to \mathbb{R}$ be convex. Suppose that $X \subset \mathbb{R}^n$ and $Y \subset \mathbb{R}$ are convex sets with $h(X) \subset Y$. Then, $f = g \circ h : X \to \mathbb{R}, f(x) = h(g(x))$ is convex. This holds only when both h and g are continuous.

Proof 1.5.5

$$f(x) = h(g(x))$$
$$f'(x) = h'(g(x)) \times g'(x)$$
$$f''(x) = h''(g(x))(g'(x))^2 + h'(g(x))g''(x)$$

Q.E.D.

Power of non-negative function: If f is convex and non-negative and $k \geq 1$, then f^k is convex.

Proof 1.5.6

Assume f is twice differentiable. Let $g = f^k$: Then:

$$\begin{aligned} &\nabla g(x) = k f^{k-1} \nabla f(x); \\ &\nabla^2 g(x) = k((k-1) f^{k-2} \nabla f(x) \nabla f^T(x) + f^{k-1} \nabla^2 f(x)) \geq 0. \end{aligned} \tag{1.5.2.19}$$

Q.E.D.

Epigraph: If $f \in \mathbb{R}^n$ is a convex function, then its epigraph $epi(f) \in \mathbb{R}^{n+1}$ is a convex set:

$$\text{epi}(f) = \left\{(x, w) \mid x \in X, w \in \mathbb{R}, f(x) \leq w\right\} \tag{1.5.2.20}$$

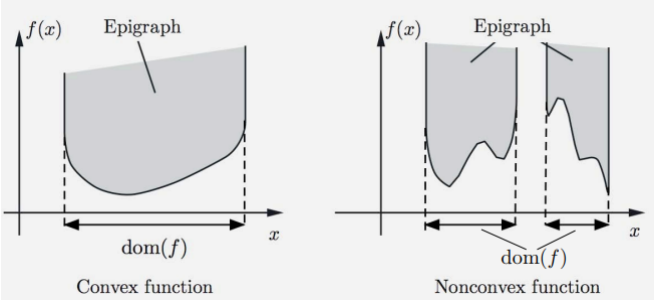


Figure 1.10: Examples of Epigraph

Theorem 1.5.2.4: Convexity and Differentiability

f is convex **if and only if**:

$$f(y) - f(x) \geq \nabla f(x)^\top (y - x), \quad \forall\, x, y \in X \tag{1.5.2.21}$$

Proof 1.5.7

First, we prove the "only if" part: suppose f is convex, we have:

$$f(\alpha \mathbf{y} + (1-\alpha)\mathbf{x}) \leqslant \alpha f(\mathbf{y}) + (1-\alpha) f(\mathbf{x}); \tag{1.5.2.22}$$

$$\frac{f(\mathbf{x} + \alpha(\mathbf{y}-\mathbf{x})) - f(\mathbf{x})}{\alpha} \leqslant f(\mathbf{y}) - f(\mathbf{x}). \tag{1.5.2.23}$$

Letting $\alpha \to 0$, we obtain:

$$\nabla f(\mathbf{x})(\mathbf{y}-\mathbf{x}) \leqslant f(\mathbf{y}) - f(\mathbf{x}). \tag{1.5.2.24}$$

Now, we prove the "if" part, assuming for all $\mathbf{x}$:

$$\begin{aligned} f(\mathbf{x}_1) &\geqslant f(\mathbf{x}) + \nabla f(\mathbf{x})(\mathbf{x}_1 - \mathbf{x}) \\ f(\mathbf{x}_2) &\geqslant f(\mathbf{x}) + \nabla f(\mathbf{x})(\mathbf{x}_2 - \mathbf{x}). \end{aligned} \tag{1.5.2.25}$$

Multiplying the first by α and the second by $1-\alpha$, and adding:

$$\alpha f(\mathbf{x}_1) + (1-\alpha) f(\mathbf{x}_2) \geqslant f(\mathbf{x}) + \nabla f(\mathbf{x})[\alpha \mathbf{x}_1 + (1-\alpha)\mathbf{x}_2 - \mathbf{x}]. \tag{1.5.2.26}$$

Replacing $\mathbf{x} = \alpha \mathbf{x}_1 + (1-\alpha)\mathbf{x}_2$, we obtain:

$$\alpha f(\mathbf{x}_1) + (1-\alpha) f(\mathbf{x}_2) \geqslant f(\alpha \mathbf{x}_1 + (1-\alpha)\mathbf{x}_2). \tag{1.5.2.27}$$

Q.E.D.

See the figure below.

Remark 1.5.5

The 'iff' condition can be replaced to:

$$h^\top \nabla^2 f(x) h \geq 0, \quad \forall\, h \in \mathbb{R}^n, \quad \forall\, x \in X \tag{1.5.2.28}$$

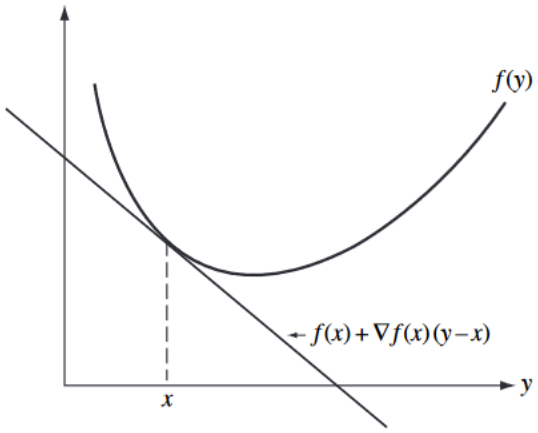


Figure 1.11: Sufficient and Necessary Condition for Convexity

Theorem 1.5.2.5: Convexity and Optimality

If f is a convex function and X is a convex set, then for the problem $\min f(x) \quad \text{s.t.} \quad x \in X$, we have:

- Every local minimizer is also a global minimizer;
- If f is strongly convex, then it has at most one global minimizer, which is the stationary point.

Lemma 1.5.2.2: Convexity and Level Sets

Let f be a convex function, then for any α, the level set $\mathcal{L}_{\leq \alpha} = \{x : f(x) \leq \alpha\}$ is a convex set.
Linear constraints are always convex.

Example 1.5.2.3

Is the optimization problem:

$$\begin{aligned} \text{maximize}_{x,y,z} \quad & xyz \\ \text{s.t.} \quad & x + 2y + 3z \leq 3 \\ & x, y, z \geq 0 \end{aligned} \tag{1.5.2.29}$$

convex or not?

Solution:

All constraints are convex since they are linear. The objective xyz is not concave. However, we can turn the above problem into an equivalent form:

$$\begin{aligned} \text{maximize}_{x,y,z} \quad & \log xyz \\ \text{s.t.} \quad & x + 2y + 3z \leq 3 \\ & x, y, z \geq 0 \end{aligned} \tag{1.5.2.30}$$

This problem is convex since $\log xyz = \log x + \log y + \log z$, which is a concave-preserved operation on concave functions.

Lipschitz Continuity

Definition 1.5.2.6: Lipschitz Continuous & Smooth

The ∇f is called Lipschitz continuous if:

$$\|\nabla f(x) - \nabla f(y)\| \leq L\|x - y\|, \quad \forall\, x, y \in \mathbb{R}^n, \tag{1.5.2.31}$$

where L is the Lipschitz constant and f is called Lipschitz smooth.

Example 1.5.2.4

Consider $f(x) = \frac{1}{2}x^\top Ax + b^\top x + c$

Solution:

$$\begin{aligned} \|\nabla f(x) - \nabla f(y)\| &= \|(Ax + b) - (Ay + b)\| \\ &= \|A(x - y)\| \leq \|A\| \cdot \|x - y\|. \end{aligned} \tag{1.5.2.32}$$

So $f(x)$ is Lipschitz Smooth with $L = \|A\|$.

Lemma 1.5.2.3: Descent Lemma

Let f be Lipschitz Smooth with L, then:

$$f(y) \leq f(x) + \nabla f(x)^\top (y - x) + \frac{L}{2}\|y - x\|^2 \quad \forall\, x, y \in \mathbb{R}^n. \tag{1.5.2.33}$$

The descent lemma provides a quadratic upper bound for f.

Proof 1.5.8

By Taylor's Expansion, the vector $z_t = x + t(y - x)$ can be expressed by $f(y) = f(x) + \int_0^1 \langle \nabla f(z_t), y - x\rangle dt$. Subtracting $\langle \nabla f(z_t), y - x\rangle$ on each side, we have:

$$\begin{aligned} f(y) - f(x) - \langle \nabla f(x), y - x\rangle = & \int_0^1 \langle \nabla f(z_t) - \nabla f(x), y - x\rangle dt \\ |f(y) - f(x) - \langle \nabla f(x), y - x\rangle| &= \left| \int_0^1 \langle \nabla f(z_t) - \nabla f(x), y - x\rangle dt \right| \\ &\leq \int_0^1 |\langle \nabla f(z_t) - \nabla f(x), y - x\rangle| dt \\ &\leq \int_0^1 \|\nabla f(z_t) - \nabla f(x)\|_2 \cdot \|y - x\|_2 dt \\ &\leq L \int_0^1 t\|x - y\|_2^2 dt \\ &= \frac{L}{2}\|x - y\|_2^2. \end{aligned} \tag{1.5.2.34}$$

Q.E.D.

Theorem 1.5.2.6: Lipschitz Continuity via Hessians

Let f be a twice cont. differentiable function. Then, the following two conditions are equivalent:

- f is Lipschitz smooth with constant L;
- $\|\nabla^2 f(x)\| \leq L$ for any $x \in \mathbb{R}^n$.

1.6 Linear Programming

1.6.1 Introduction to Linear Programming

The main contents of this section are notes from [Luenberger et al., 1984], [Bertsimas and Tsitsiklis, 1997]. The *standard form* of the linear programming:

$$\begin{aligned} \text{minimize} \quad & c_1x_1 + c_2x_2 + \ldots + c_nx_n \\ \text{subject to} \quad & a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n = b_1 \\ & a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n = b_2 \\ & \vdots \quad \vdots \\ & a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n = b_m \\ \text{and} \quad & x_1 \geqslant 0, x_2 \geqslant 0, \ldots, x_n \geqslant 0, \end{aligned} \tag{1.6.1.1}$$

Or the above equations can be concisely written in:

$$\begin{aligned} \text{minimize} \quad & \mathbf{c}^T\mathbf{x} \\ \text{subject to} \quad & \mathbf{Ax} = \mathbf{b} \quad \text{and} \quad \mathbf{x} \geqslant \mathbf{0}. \end{aligned} \tag{1.6.1.2}$$

Convertion to standard form LP

Slack Variable

For this kind of LP formation:

$$\begin{aligned} \text{minimize} \quad & c_1x_1 + c_2x_2 + \ldots + c_nx_n \\ \text{subject to} \quad & a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n \leq b_1 \\ & a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n \leq b_2 \\ & \vdots \quad \vdots \\ & a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n \leq b_m \\ \text{and} \quad & x_1 \geqslant 0, x_2 \geqslant 0, \ldots, x_n \geqslant 0, \end{aligned} \tag{1.6.1.3}$$

The above formulation can be transformed into the following standard form:

$$\begin{aligned} \text{minimize} \quad & c_1x_1 + c_2x_2 + \cdots + c_nx_n \\ \text{subject to} \quad & a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n + y_1 = b_1 \\ & a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n + y_2 = b_2 \\ & \vdots \quad \vdots \\ & a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n + y_m = b_m \\ \text{and} \quad & x_1 \geqslant 0, x_2 \geqslant 0, \ldots, x_n \geqslant 0, \\ \text{and} \quad & y_1 \geqslant 0, y_2 \geqslant 0, \ldots, y_m \geqslant 0. \end{aligned} \tag{1.6.1.4}$$

Now, the constraint would be modified from $\mathbf{A}$ to $[\mathbf{A}, \mathbf{I}]$, and the number of unknowns is changed from n to $n + m$.

Surplus Variable

Similar to the slack variable, formulation like:

$$a_{i1}x_1 + a_{i2}x_2 + \cdots + a_{in}x_n \geqslant b_i \tag{1.6.1.5}$$

can be transformed into:

$$a_{i1}x_1 + a_{i2}x_2 + \cdots + a_{in}x_n - y_i = b_i \tag{1.6.1.6}$$

Free Variable

For some variables without the constraint of the sign, there are two methods to transform it into the standard form. One is to $x_i = u_i - v_i$, where both u_i and v_i are larger or equal to zero. Another method can be used when the following condition holds:

$$a_1x_1 + \cdots + a_ix_i + \cdots + a_nx_n = b_i \tag{1.6.1.7}$$

where x_i is the free variable, thus we can replace x_i by $b_i - \sum_{j\neq i}^{n} a_jx_j$, which can eliminate one variable and one constraint simultaneously.

One important example of the linear programming model is the **maximal flow problem**:

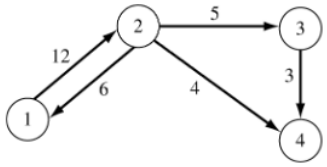


Figure 1.12: A network with capacities

This problem can be formulated into:

$$\begin{aligned}
&\text{minimize} f \\
&\text{subject to} \sum_{j=1}^{n} x_{1j} - \sum_{j=1}^{n} x_{j1} - f = 0 \\
&\sum_{j=1}^{n} x_{ij} - \sum_{j=1}^{n} x_{ji} = 0, \qquad i \neq 1, m \\
&\sum_{j=1}^{n} x_{mj} - \sum_{j=1}^{n} x_{jm} + f = 0 \\
&0 \leq x_{ij} \leq k_{ij}, \qquad \text{for all } i, j
\end{aligned} \tag{1.6.1.8}$$

Basic Solutions

The system of linear equalities:

$$\mathbf{Ax} = \mathbf{b} \tag{1.6.1.9}$$

where $\mathbf{A}$ is a $m \times n$ constraint matrix and $\mathbf{x}$ is the $n \times 1$ decision variables.

> **Assumption 1.6.1.1:** Full Rank Assumption
> The $m \times n\mathbf{A}$ has $m < n$, and the m rows of $\mathbf{A}$ are linearly independent

. This assumption ensures that linear equalities always have at least one basic solution. At least m linearly independent columns $\rightarrow \mathbf{B}$, then would get:

$$\mathbf{Bx_B} = \mathbf{b} \tag{1.6.1.10}$$

Definition 1.6.1.1: Basic Feasible Solution

$\mathbf{x} = (\mathbf{x_B}, \mathbf{0})$ is the **basic solution**, the components of $\mathbf{x}$ related to the columns of $\mathbf{B}$ are **basic variables**
If the solution also satisfies $\mathbf{x} \geq 0$, then it's called a **basic feasible solution**.
If some of the basic variables have value zero, then it's called a **degenerate basic solution**.
Similar to the definition above, we have **degenerate basic feasible solution**.

Theorem 1.6.1.1: Fundamental Theorem of Linear Programming

Given a linear program in standard form, where $\mathbf{A}$ is an $m \times n$ matrix of rank m:

1. If there is a feasible solution, there is a basic feasible solution;
2. If there is an optimal feasible solution, there is an optimal basic feasible solution.

Proof 1.6.1

(1)
If there is a feasible solution $\mathbf{x}$, the solution satisfy:

$$x_1\mathbf{a}_1 + x_2\mathbf{a}_2 + \cdots + x_n\mathbf{a}_n = \mathbf{b} \tag{1.6.1.11}$$

Assume there are p of variable $x_i > 0$, then the following equation holds:

$$x_1\mathbf{a}_1 + x_2\mathbf{a}_2 + \cdots + x_p\mathbf{a}_n = \mathbf{b} \tag{1.6.1.12}$$

Then there are two cases:

1. if $\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_p$ are linearly independent, then $p \leq m$, which means $\mathbf{x}$ is already a basic solution;
2. otherwise, there would be a non-trivial solution for $y_1\mathbf{a}_1 + y_2\mathbf{a}_2 + \cdots + y_p\mathbf{a}_p = \mathbf{0}$, which means $(x_1 - \varepsilon y_1)\mathbf{a}_1 + (x_2 - \varepsilon y_2)\mathbf{a}_2 + \cdots + (x_p - \varepsilon y_p)\mathbf{a}_p = \mathbf{b}$.
   Then for any value of ϵ, $\mathbf{x} - \varepsilon\mathbf{y}$ is a solution but may violate the signal constraint.
   Mention that there is at least one y_i that is negative or positive, thus there is at least one x_i decreasing when we increase the $\epsilon(\epsilon > 0)$, thus we set $\varepsilon = \min\{x_i/y_i : y_i > 0\}$, which would bring us to a new feasible solution but the number of zeros is larger.
   Through such iteration, we can get at least one basic feasible solution.

(2)
The idea is the same as the first one. In case one, it's obvious. in case two, we need to prove for any , $\mathbf{x} - \varepsilon\mathbf{y}$ is still optimal.
Note the new value is $\mathbf{c}^T\mathbf{x} - \varepsilon\mathbf{c}^T\mathbf{y}$. Because ε can be both positive and negative, thus $\mathbf{c}^T\mathbf{y} = 0$, which makes $\mathbf{x} - \varepsilon\mathbf{y}$ still as optimal solution.

Q.E.D.

Remark 1.6.1

This theorem reduce the original problem to the size of $\binom{n}{m} = \frac{n!}{m!(n-m)!}$.

Definition 1.6.1.2: Extreme Point

A point $\mathbf{x}$ in a convex set C is an *extreme point* if there are **no** two distinct $\mathbf{x}_1, \mathbf{x}_2 \in C$ such that $\mathbf{x} = \alpha\mathbf{x}_1 + (1-\alpha)\mathbf{x}_2$ for some $\alpha, 0 < \alpha < 1$

Theorem 1.6.1.2: Equivalence of extreme points and basic solutions

Let $\mathbf{A}$ be an $m \times n$ matrix with rank m, Let $\mathbf{K}$ denote the *convex polytope* consisting all vector $\mathbf{x}$ satisfying $\mathbf{Ax} = \mathbf{b}, \mathbf{x} \geqslant \mathbf{0}$.

A vector $\mathbf{x}$ is an extreme point of $\mathbf{K}$ if and only if $\mathbf{x}$ is a basic feasible solution.

Proof 1.6.2

(1) BFS $\longrightarrow$ extreme point:

Suppose $\mathbf{x} = (x_1, x_2, \ldots, x_m, 0, 0, \ldots, 0)$ is a BFS, it satisfies $x_1\mathbf{a}_1 + x_2\mathbf{a}_2 + \cdots + x_m\mathbf{a}_m = \mathbf{b}$. If $\mathbf{x} = \alpha\mathbf{y} + (1-\alpha)\mathbf{z}$, since the value in $\mathbf{y}, \mathbf{z}$ is larger than 0, then we have:

$$y_1\mathbf{a}_1 + y_2\mathbf{a}_2 + \cdots + y_m\mathbf{a}_m = \mathbf{b}z_1\mathbf{a}_1 + z_2\mathbf{a}_2 + \cdots + z_m\mathbf{a}_m = \mathbf{b} \tag{1.6.1.13}$$

Because the vectors $\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_m$ are linearly independent, then we can get $\mathbf{x} = \mathbf{y} = \mathbf{z}$, which means that $\mathbf{z}$ is an extreme point.

(2) Extreme point $\longrightarrow$ BFS:

Assume that $\mathbf{x}$ has k components larger than zero, then we have: $y_1\mathbf{a}_1 + y_2\mathbf{a}_2 + \cdots + y_k\mathbf{a}_k = 0$. Assume that $\mathbf{a}_1, \mathbf{a}_2, \cdots, \mathbf{a}_k$ are linearly dependent, which would lead to:

$$y_1\mathbf{a}_1 + y_2\mathbf{a}_2 + \cdots + y_k\mathbf{a}_k = 0 \tag{1.6.1.14}$$

Define $\mathbf{y} = (y_1, y_2, \ldots, y_k, 0, 0, \ldots, 0)$, it's obvious to see the following can exist:

$$\mathbf{x} + \epsilon\mathbf{y} \geqslant 0, \quad \mathbf{x} - \epsilon\mathbf{y} \geqslant 0 \tag{1.6.1.15}$$

Contradicts that $\mathbf{x}$ is an extreme point $\longrightarrow \mathbf{a}_1, \mathbf{a}_2, \cdots, \mathbf{a}_k$ are linearly independent $\longrightarrow \mathbf{x}$ is a BFS.

Q.E.D.

Corollary 1.6.1.1

1. If the convex set $\mathbf{K}$ is nonempty, there is at least one extreme point;
2. If there is a finite optimal solution to a linear programming problem, there is a finite optimal solution, which is an extreme point of the constraint set.;
3. The constraint set $\mathbf{K}$ possesses at most a finite number of extreme points;
4. If K is bounded, then it's a *convex polyhedron*.

1.6.2 Simplex Method

The standard form linear programming can be transformed to the **canonial form** (reduced row-echelon form/tabular method). The canonical form provides basic variables and non-basic variables. **Pivot equations** can transform a non-basic variable into a basic variable.

$$\begin{cases} \bar{a}'_{ij} = \bar{a}_{ij} - \frac{\bar{a}_{iq}}{\bar{a}_{pq}} \bar{a}_{pj}, i \neq p \\ \bar{a}'_{pj} = \frac{\bar{a}_{pj}}{\bar{a}_{pq}}. \end{cases} \tag{1.6.2.1}$$

1.7 Unconstrained Optimization

1.7.1 Single Variable Problem

Bisection method uses the idea that the local minimizer must satisfy the FONC: $f'(x) = 0$.

Bisection Method

1. Define $x_m = \frac{x_\ell + x_r}{2}$.
2. If $g(x_m) = 0$, then output x_m.
3. Otherwise:
   - If $g(x_m) > 0$, then let $x_r = x_m$.
   - If $g(x_m) < 0$, then let $x_\ell = x_m$.
4. If $|x_r - x_\ell| < \epsilon$: stop and output $\frac{x_\ell + x_r}{2}$, otherwise go back to step 1.

Figure 1.13: Bisection Method

The bisection method can only help us find an approximate *stationary point*. When f is convex, this is an approximate global minimizer of f.
The iteration of the bisection method takes at most $\log_2(\frac{x_r - x_\ell}{\epsilon})$ steps.
Golden section method can optimize those problems whose first derivative is difficult to obtain, but it requires us to know the range of the local optimal solution.

Golden Section Method

Assume we start with $[x_\ell, x_r]$. Assume $0 < \phi < 0.5$.

1. Set $x'_\ell = \phi x_r + (1 - \phi)x_\ell$ and $x'_r = (1 - \phi)x_r + \phi x_\ell$.
2. If $f(x'_\ell) < f(x'_r)$, then the minimizer must lie in $[x_\ell, x'_r]$, so set $x_r = x'_r$.
3. Otherwise, the minimizer must lie in $[x'_\ell, x_r]$, so set $x_\ell = x'_\ell$.
4. If $x_r - x_\ell < \epsilon$, output $\frac{x_\ell + x_r}{2}$, otherwise go back to step 1.

Figure 1.14: Golden Section Method

If we set $\phi = \frac{3-\sqrt{5}}{2}$ (comes from $\frac{r}{1} = \frac{1-2r}{1-r}$), then we can save one function evaluation during each iteration.

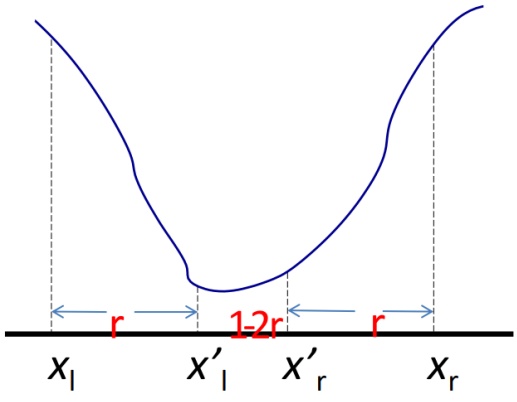


Figure 1.15: Golden Section Graph

The iteration of the golden section method takes at most $\log_2(\frac{x_r - x_\ell}{\epsilon})$ steps. A necessary condition for the

minimizer is $g(x) = f'(x) = 0$.

Newton's Method in single dimension approximates g using a first-order Taylor expansion for each x^k: $g(x) \approx g(x^k) + g'(x^k)(x - x^k)$. By setting the right-hand side to 0 and solving:

$$x^{k+1} = x^k - \frac{g(x^k)}{g'(x^k)}; \tag{1.7.1.1}$$

$$x^{k+1} = x^k - \frac{f'(x^k)}{f''(x^k)}. \tag{1.7.1.2}$$

where we assume $g'(x^k) \neq 0$ at each step. However, Newton's method may not converge for every initial point.

Theorem 1.7.1.1: Quadratic Convergence of Newton's method in 1-D case

If g is twice continuous and $x^\star$ is a root of g (global minimizer) and $g'(x^\star) \neq 0$. When $|x^0 - x^\star|$ is sufficiently small, the sequence by Newton's method $x^{k+1} = x^k - \frac{g(x^k)}{g'(x^k)}$ will satisfy:

$$|x^{k+1} - x^*| \leq C|x^k - x^*|^2. \tag{1.7.1.3}$$

Another interpretation of Newton's method: consider a second-order Taylar expansion for f:

$$f(x) \approx f(x^k) + f'(x^k)(x - x^k) + \frac{1}{2}f''(x^k)(x - x^k)^2. \tag{1.7.1.4}$$

The Newtion's method is a minimizer for the quadratic model. If the original objective function is quadratic, then Newton's method converges in one step.

1.7.2 Basic Gradient Descent Theory and Algorithms

Consider:

$$\text{minimize}_{x \in \mathbb{R}^n} \quad f(x). \tag{1.7.2.1}$$

We typically use iterative steps of the form:

$$x^{k+1} = x^k + \alpha_k d^k. \tag{1.7.2.2}$$

The $d^k \in \mathbb{R}^n$ is called a descent direction of f if $\nabla f(x)^\top d < 0$. The step size α_k may be chosen in accordance with some line search (*one-dimensioanl*) rules.

Schematic Descent Directions Method

1. Initialization: Select an initial point $\boldsymbol{x}^0 \in \mathbb{R}^n$.

For $k = 0, 1, \ldots$:

2. If a stopping criterion is satisfied, then STOP and $\boldsymbol{x}^k$ is the output.
3. Pick a descent direction $\boldsymbol{d}^k$.
4. Find a stepsize α_k satisfying $f(\boldsymbol{x}^k + \alpha_k \boldsymbol{d}^k) < f(\boldsymbol{x}^k)$.
5. Set $\boldsymbol{x}^{k+1} = \boldsymbol{x}^k + \alpha_k \boldsymbol{d}^k$.

Figure 1.16: Abstract Descent Method

A popular stopping criterion is $\|\nabla f(x^k)\| \leq$ to1 with tolerance $tol > 0$.

Definition 1.7.2.1: Steepest Descent Directions

Let f be convex and let $x \in \mathbb{R}^n$ be given with $nablaf(x) \neq 0$. Let $d \in \mathbb{R}^n$ denote the solution of:

$$\min_{\|d\|=1} \nabla f(\mathbf{x})^\top \boldsymbol{d}. \tag{1.7.2.3}$$

Every vector of the form $s = \lambda d, \lambda > 0$, is called **steepest descent direction**.
By Cauchy-Schwarz inequality: $|\nabla f(x)^T d| \leq ||\nabla f(x)|| \cdot ||d||$, which implies $\nabla f(x)^T d \geq -||\nabla f(x)|| \cdot ||d||$, thus, we can have $\nabla f(x)^T d \geq -||\nabla f(x)||$. So $d^\star = -\nabla f(x)/\|\nabla f(x)\|$.
We can choose the descent directions $d = -\lambda \nabla f(x), \lambda > 0$.

Remark 1.7.1

- The steepest descent norm depends on the chosen norm (here it is Euclidean norm);
- The gradient $\nabla f(y)$ is *perpendicular* to the level set, and it points towards a direction where f is decreasing the most.

Gradient Descent Method

1. Initialization: Select an initial point $\boldsymbol{x}^0 \in \mathbb{R}^n$.

For $k = 0, 1, \ldots$:

2. If $\|\nabla f(\boldsymbol{x}^k)\| \leq \varepsilon$, then STOP and $\boldsymbol{x}^k$ is the output.
3. Pick a stepsize α^k by a line search procedure (exact line search or backtracking) on the function

$$\phi(\alpha) = f(\boldsymbol{x}^k - \alpha \nabla f(\boldsymbol{x}^k)).$$

4. Set $\boldsymbol{x}^{k+1} = \boldsymbol{x}^k - \alpha_k \nabla f(\boldsymbol{x}^k)$.

Figure 1.17: Gradient Descent Method

Step Size Strategies

There is one simple step size strategy: **constant step size**, by which we choose $\alpha_k = \bar{\alpha}$ for all k.

Exact Line Search

We choose α_k such that:

$$\alpha_k = \arg\min_{\alpha \geq 0} f(x^k + \alpha d^k). \tag{1.7.2.4}$$

Golden section methods can solve α_k. In some cases, there are analytical solutions for α_k.

Proposition 1.7.2.1: The "zig-zag" path of ELS

The directions between two consecutive steps are perpendicular:

$$(d^{k+1})^\top d^k = 0, (x^{k+2} - x^{k+1})^T (x^{k+1} - x^k) = 0. \tag{1.7.2.5}$$

Proof 1.7.1

For the kth iteration, the $alpha_k$ need to satisfy $\alpha_k = \arg\min_{\alpha \geq 0} f(x^k + \alpha d^k)$. Denote $\phi(\alpha) = f(x^k + \alpha d^k)$, so $\phi'(\alpha) = \nabla f(x^k + \alpha d^k)^T d^k$. When $alpha = 0$, $\phi'(\alpha) = -\|\nabla f(x^k)\|^2 < 0$. To minimize the formula, we need to set $\phi'(\alpha) = 0$, then:

$$\nabla f(x^k + \alpha^k d^k)^\top d^k = 0 (d^{k+1})^\top d^k = 0. \tag{1.7.2.6}$$

Q.E.D.

Backtracking Line Search

Let $\sigma, \gamma \in (0, 1)$ be given, choose α_k as the largest element from $(1, \sigma, \sigma^2, \sigma^3, \cdots)$ such that:

$$f(x^k + \alpha_k d^k) - f(x^k) \leq \gamma \alpha_k \cdot \nabla f(x^k)^\top d^k. \tag{1.7.2.7}$$

This is called **Armijo condition**.
Procedure:

1. Start with $\alpha = 1$ or $\alpha = s$;
2. If $f(x^k + \alpha d^k) \leq f(x^k) + \gamma \alpha \cdot \nabla f(x^k)^\top d^k$, choose $\alpha_k = \alpha$, else set $\alpha = \sigma \alpha$ and repeat.

By Taylor expansion, for small α:

$$f(x^k + \alpha d^k) \approx f(x^k) + \alpha \nabla f(x^k)^\top d^k < f(x^k) + \gamma \alpha \cdot \nabla f(x^k)^\top d^k. \tag{1.7.2.8}$$

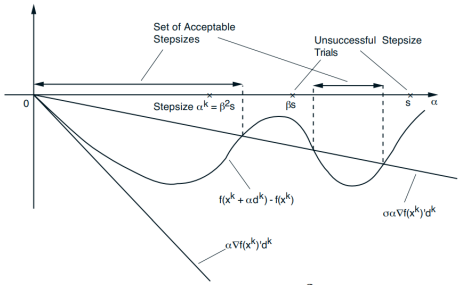


Figure 1.18: Armijo Rule

Diminishing Step Sizes

Choose $\alpha_k \to 0$ and $\sum_{k=0}^{\infty} \alpha_k = \infty$, this is frequently used in stochastic programming.

The Convergence Rate

Definition 1.7.2.2: Accumulation Point

A point x is an **accumulation point** of $\{x^k\}_k$ if for every $\epsilon > 0$, there are infinitely many numbers k with $x^k \in B_\epsilon(x)$.

Remark 1.7.2

- If x is an **accumulation point** of $\{x^k\}_k$, then there is a subsequence of $\{x^k\}_k$ that converges to x;
- If $\{x^k\}_k$ converges to some $x \in \mathbb{R}^n$, then x is the unique accumulation point of $\{x^k\}_k$;
- A bounded sequence has at least one accumulation point.

Definition 1.7.2.3: Global Convergence

The gradient method can find the stationary point **independent** of the chosen initial point.

Theorem 1.7.2.1: Global Convergence

Let $f : \mathbb{R}^n \to \mathbb{R}$ be continuously differentiable, $\{x^k\}_k$ be the sequence generated by gradient method for:

$$\min_x f(x) \quad \text{s.t.} \quad x \in \mathbb{R}^n, \tag{1.7.2.9}$$

with *exact line search* or *Armijo rule* step size strategies, $\{f(x^k)\}_k$ has the properties:

- it's nonincreasing;
- every accumulation point of $\{f(x^k)\}_k$ is a stationary point of f.

Remark 1.7.3

- It does not guarantee the existence of an accumulation point or convergence to a single limit point;
- If f is *coercive* (bounded), then $\{f(x^k)\}_k$ must have an accumulation point.

Rigorously speaking, we define the algorithm A as a point-to-point mapping:

$$x_{k+1} = A(x_k), \tag{1.7.2.10}$$

or a point-to-set mapping:

$$x_{k+1} \in A(x_k). \tag{1.7.2.11}$$

We further define the continuous descent function Z on X given the solution set $\Gamma \subset X$ if Z satisfy:

1. if $x \notin \Gamma$ and $y \subset A(x)$, then $Z(y) < Z(x)$;
2. if $x \in \Gamma$ and $y \subset A(x)$, then $Z(y) \leq Z(x)$.

Definition 1.7.2.4: Closed Mapping

A point-to-set mapping A from X to Y is said to be **closed** at $x \in X$ if $x_k \to x, x_k \in X$ and $y_k \to y, y_k \in A(x_k)$ imply that $y \in A(x)$.

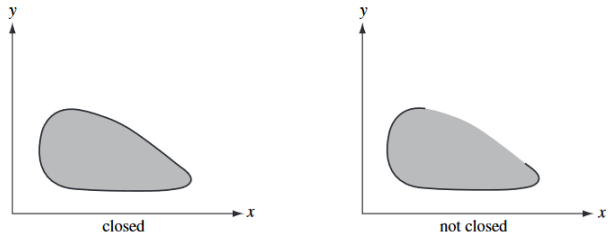


Figure 1.19: Examples of Closed and Unclosed Mapping

We can further define the *composite mapping*. Let $A : X \to Y$ and $B : Y \to Z$ be point-to-set mappings, the composite mapping $C + BA$ is defined as $C : X \to Z$ with:

$$\mathbf{C(x)} = \underset{\mathbf{y \in A(x)}}{\cup} \mathbf{B(y)}. \tag{1.7.2.12}$$

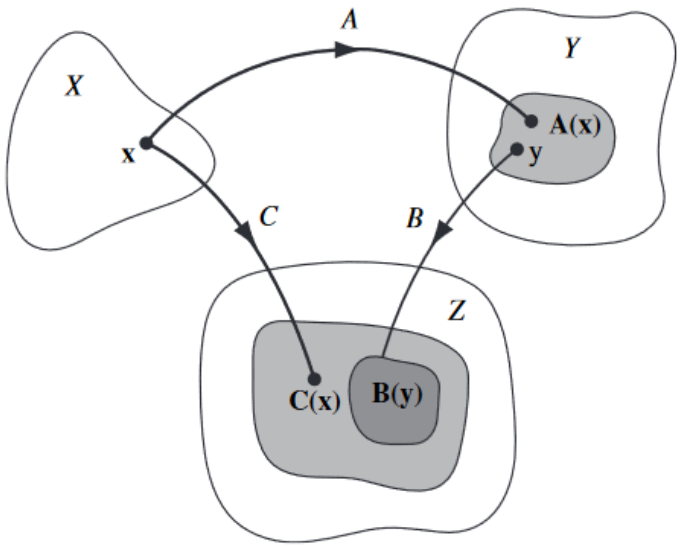


Figure 1.20: Composition of mappings

Proposition 1.7.2.2: Closedness of Composite Mapping

Let $\mathbf{A} : X \to Y$ and $\mathbf{B} : Y \to Z$ be point-to-set mappings. If $\mathbf{A}$ is closed at x, $\mathbf{B}$ is closed on $A(x)$. If there exists a y and a subsequence $\{y_{ki}\}$, and $\{y_{ki}\} \to y$, then the composite mapping $C = BA$ is closed at point x.

Proof 1.7.2

Let $x_k \to x$ and $z_k \to z, (z_k \in C(x_k))$, according to the definition, if we can prove $z \in C(x)$, the proposition holds. According to the assumption, we can find a y such that $x_k \to x, \{y \in A(x_k)\} \to y$. Since A is closed at point x, we can get $y \in A(x)$, and B is closed at y. When $x_k \to x$, $z_k \in C(x_k)(B(A(x_k))) \to z$, we can show that $z \in B(y)$. Thus, $z \in B(A(x)), z \in C(x)$.

Q.E.D.

Corollary 1.7.2.1

- Let $A : X \to Y$ and $B : Y \to Z$ be point-to-set mappings. If A is closed at x, B is closed on $A(x)$ and Y is compact, then $C = BA$ is closed at x;
- Let $A : X \to Y$ be a point-to-point mapping and $B : Y \to Z$ a point-to-set mapping. If A is continuous at x and B is closed on $A(x)$, then $C = BA$ is closed at x.

Theorem 1.7.2.2: (Rigorous) Global Convergence Theorem

Let A be an algorithm on X, given x_0, the sequence $\{x_k\}$ is generated by $x_{k+1} \in A(x_k)$. Let a solution $\Gamma \subset X$ be given, and suppose:

- all points x_k are contained in a compact set $S \subset X$;
- A is the descent function for A on X;
- the mapping A is closed outside Γ.

Then the **limit of any convergent subsequence** pf $\{x_k\}$ is a solution.

Theorem 1.7.2.3: Convergence under Lipschitz Continuity

Let $f : \mathbb{R}^n \to \mathbb{R}$ be Lipschitz Smooth, $\{x^k\}_k$ be the sequence generated by gradient method for:

$$\min_x f(x) \quad \text{s.t.} \quad x \in \mathbb{R}^n, \tag{1.7.2.13}$$

with *constant step size* $\bar{\alpha} \in (0, \frac{2}{L})$, *exact line search*, *diminishing* step size or *Armijo rule* step size strategies, $\{f(x^k)\}_k$ either:

- converge to $-\infty$;
- or converge to a finite value that $\|\nabla f(x^k)\| \to 0$.

Corollary 1.7.2.2: Complexity Bounds

Complexity bounds characterize the behavior and progress of an algorithm during the very first iterations. If f is Lipschitz smooth with L and lower bounded by $\bar{f}$, then it holds:

$$\min_{i=0,\cdots,T-1} \|\nabla f(x^i)\| \leq \frac{f(x^o) - \overline{f}}{\sqrt{\beta T}} = O(1/\sqrt{T}) \quad \forall T. \tag{1.7.2.14}$$

here $\{x^k\}$ are generated by constant step size and $\bar{\beta} = \bar{\alpha}(1 - \frac{L\bar{\alpha}}{2})$.

Remark 1.7.4

- We need to perform the gradient method for $T > \frac{1}{tol^2}$ iterations to ensure $\min_{i=0,\cdots,T-1} \|\nabla f(x^i)\| \leq tol$;
- If f is convex and the problem has a solution, then the whole sequence converges to a global minimum;
- Convergence can be ensured if the stationary points are isolated;
- This works for backtracking, constant & diminishing step sizes.

Proof 1.7.3

Recall the descent lemma:

$$f(y) \leq f(x) + \nabla f(x)^\top (y - x) + \frac{L}{2}\|y - x\|^2 \quad \forall\, x, y \in \mathbb{R}^n. \tag{1.7.2.15}$$

Set $y = x^{k+1}$ and $x = x^k$, so $x^{k+1} - x^k = -\bar{\alpha}\nabla f(x^k)$, reducing the descent lemma to:

$$f(x^{k+1}) \leq f(x^k) - (1 - \frac{L\bar{\alpha}}{2}\bar{\alpha} \cdot \|\nabla f(x^k)\|^2). \tag{1.7.2.16}$$

Denote $\bar{beta} = \bar{\alpha}(1 - \frac{L\bar{\alpha}}{2})$, we have $\bar{\beta}\|\nabla f(x^k)\|^2 \leq f(x^k) - f(x^{k+1})$. Then sum this for $k = 0, \cdots, T-1$:

$$\bar{\beta} \cdot \sum_{k=0}^{T-1} \|\nabla f(x^k)\|^2 \leq f(x^0) - f(x^T) \leq f(x^0) - \bar{f}. \tag{1.7.2.17}$$

For the left part:

$$\bar{\beta} \cdot \sum_{k=0}^{T-1} \|\nabla f(x^k)\|^2 \geq \bar{\beta} \cdot \sum_{k=0}^{T-1} \min_{i=0,\cdots,T-1} \|\nabla f(x^k)\|^2 = T \cdot \min_{i=0,\cdots,T-1} \|\nabla f(x^k)\|^2. \tag{1.7.2.18}$$

Finally, we have:

$$\min_{i \to 0,\ldots,T \to 1} \|\nabla f(x^i)\| \leq \frac{f(x^0) - \bar{f}}{\sqrt{\beta T}} = O(1/\sqrt{T}) \quad \forall T. \tag{1.7.2.19}$$

Q.E.D.

Definition 1.7.2.5: Order of Convergence

Let $\{r_k \to r^*\}$, the order is the supremum of the $p \geq 0$ satisfying:

$$0 \leqslant \overline{\lim_{k \to \infty}} \frac{|r_{k+1} - r^*|}{|r_k - r^*|^p} < \infty. \tag{1.7.2.20}$$

In most of the cases, we can use a simpler definition:

$$\beta = \lim_{k\to\infty} \frac{|r_{k+1} - r^*|}{|r_k - r^*|^p} \tag{1.7.2.21}$$

Definition 1.7.2.6: Linear Convergence

$\{x^k\}_k$ converges linear with rate $\eta \in (0,1)$ to $x^\star \in \mathbb{R}^n$ if there is $\mathcal{L} > 0$ such that:

$$\|x^{k+1} - x^\star\| \le \eta \cdot \|x^k - x^\star\|, \forall k \ge \mathcal{L} \tag{1.7.2.22}$$

Remark 1.7.5

- linear convergence with $\mathcal{L} = 0$ implies that $\|x^k - x^\star\| \le \eta^k \|x^0 - x^\star\|$.

The sequence with $r_k = a^k$ where $0 < a < 1$ converges to zero with linear convergence, since $r_{k+1}/r_k = a$.
The sequence with $r_k = a^{2^k}$ where $0 < a < 1$ converges to zero with quadratic convergence, since $r_{k+1}/r_k^2 = 1$.

Theorem 1.7.2.4: Rates for Strongly Convex Problems

Let f be Lipschitz smooth with L and suppose there is $\mu > 0$ such that:

$$\mu\|\boldsymbol{h}\|^2 \le \boldsymbol{h}^\top \nabla^2 f(\boldsymbol{x})\boldsymbol{h} (\le \|\boldsymbol{h}\|^2) \forall \boldsymbol{h}, \forall \boldsymbol{x}. \tag{1.7.2.23}$$

Then $\{x^k\}_k$ converges linearly to $x^\star$ if it's generated by the gradient method. Further, for $\eta = 1 - 2M_\mu$, it follows:

$$f(x^k) - f(x^*) \le \eta^k \cdot [f(x^0) - f(x^*)]. \tag{1.7.2.24}$$

, where

$$M = \begin{cases} \tilde{\alpha}(1 - \frac{L\tilde{\alpha}}{2}) & \text{constant step size:} \bar{\alpha} \in (0, \frac{2}{L}), \\ \frac{1}{2L} & \text{exact line search,} \\ \gamma \min\{1, \frac{2\sigma(1-\gamma)}{L}\} & \text{Armijo line search.} \end{cases} \tag{1.7.2.25}$$

The optimal rate can be $\eta = 1 - \frac{\mu}{L} = 1 - \frac{1}{\kappa}, \kappa = \frac{L}{\mu}$. The larger κ, the lower the convergence rate.

1.7.3 Conjugate Direction Methods

The conjugate gradient method (CG-method) is an iterative approach for solving quadratic optimization and linear equations $Ax = b$, where A is symmetric and PD. This problem can be treated as a minimization problem: eq $\min_{x\in\mathbb{R}^n} \phi(x) = \frac{1}{2}x^\top Ax - b^\top x. These two problems are equivalent because of \nabla(x) = Ax - b$. We are interested in the convergence of the residuals:

$$r^k := Ax^k - b = \nabla\phi(x^k) \quad \text{at the k-th iter.} \tag{1.7.3.1}$$

Definition 1.7.3.1: Q-conjugate Directions

Let Q be a symmetric matrix. $\{d_1, d_2, \ldots, d_k\}$ vectors $d_i \in \mathbb{R}^n, i \neq 0$ are Q-conjugate w.r.t. Q if:

$$d_i^T Q d_j = 0, \forall i \neq j \tag{1.7.3.2}$$

Lemma 1.7.3.1: Linear Independence for Q-conjugate Directions

If Q is PD, then directions $\{d_1, d_2, \ldots, d_k\}$ are linearly independent.

Proof 1.7.4

Proof by contradiction. If $d_k = \alpha_1 d_1 + \ldots + \alpha_{k-1} d_{k-1}$, then:

$$\begin{aligned} 0 < d_k^T Q d_k &= d_k^T Q(\alpha_1 d_1 + \ldots + \alpha_{k-1} d_{k-1}) \\ &= \alpha_1 d_k^T Q d_1 + \ldots + \alpha_{k-1} d_k^T Q d_{k-1} = 0 \end{aligned} \tag{1.7.3.3}$$

Q.E.D.

Now, reconsider the quadratic minimization problem and let x^* denote the solution. Since the linear Independence lemma:

$$x^* = \alpha_0 d_0 + \ldots + \alpha_{n-1} d_{n-1} \tag{1.7.3.4}$$

Therefore,

$$d_i^T Q x^* = d_i^T Q(\alpha_0 d_0 + \ldots + \alpha_{n-1} d_{n-1}) = \alpha_i d_i^T Q d_i \tag{1.7.3.5}$$

$$\alpha_i = \frac{d_i^T Q x^*}{d_i^T Q d_i} = \frac{d_i^T b}{d_i^T Q d_i} \tag{1.7.3.6}$$

Which can lead to the solution of x^* without the matrix inversion:

$$x^* = \sum_{i=0}^{n-1} \alpha_i d_i = \sum_{i=0}^{n-1} \frac{d_i^T b}{d_i^T Q d_i} d_i \tag{1.7.3.7}$$

Theorem 1.7.3.1: Conjugate Direction

Let $\{d_0, d_1, \ldots, d_{n-1}\}$ be Q-conjugate, x_0 be an arbitrary starting point. By updating:

$$x_{k+1} = x_k + \alpha_k d_k \tag{1.7.3.8}$$

$$g_k = Q x_k - b \tag{1.7.3.9}$$

$$\alpha_k = -\frac{g_k^T d_k}{d_k^T Q d_k} = -\frac{(Q x_k - b)^T d_k}{d_k^T Q d_k} \tag{1.7.3.10}$$

After n steps, $x_n = x^*$.

Proof 1.7.5

To prove the theorem above, we only need to prove why we have the form $\alpha_k = -\frac{(Qx_k - b)^T d_k}{d_k^T Q d_k}$.
We have:

$$d_k^T Q(x^* - x_0) = d_k^T Q(\alpha_0 d_0 + \ldots + \alpha_{n-1} d_{n-1}) = \alpha_k d_k^T Q d_k$$
$$\Rightarrow \alpha_k = \frac{d_k^T Q(x^* - x_0)}{d_k^T Q d_k} \tag{1.7.3.11}$$
$$d_k^T Q(x_k - x_0) = d_k^T Q(\alpha_0 d_0 + \alpha_1 d_1 + \ldots + \alpha_{k-1} d_{k-1}) = 0$$
$$d_k^T Q(x^* - x_0) = d_k^T Q(x^* - x_k + x_k - x_0) = d_k^T Q(x^* - x_k)$$

So,

$$\alpha_k = \frac{d_k^T Q(x^* - x_0)}{d_k^T Q d_k} = \frac{d_k^T Q(x^* - x_k)}{d_k^T Q d_k} = -\frac{d_k^T g_k}{d_k^T Q d_k} \tag{1.7.3.12}$$

Q.E.D.

The CG method is a *special* conjugate direction method that iteratively generates the conjugate vectors. d^k is chosen as a linear combination of the previous d^{k-1} and the gradient $g(d^k)$:

$$p^k = -g_{\beta}^k{}_k p^{k-1} \tag{1.7.3.13}$$

To satisfy $(d^{k-1})^T Q d^k = 0$, we obtain:

$$\beta_k = \frac{(g^k)^T Q d^{k-1}}{(d^{k-1})^T Q d^{k-1}} \tag{1.7.3.14}$$

The CG-Method

1. Initialization: Select an initial point $\boldsymbol{x}^0 \in \mathbb{R}^n$ and set $\texttt{tol} \geq 0$, $\boldsymbol{r}^0 = \boldsymbol{A}\boldsymbol{x}^0 - \boldsymbol{b}$, and $\boldsymbol{p}^0 = -\boldsymbol{r}^0$.

For $k = 0, 1, \ldots$:

2. Compute the updates $\alpha_k = -\frac{(\boldsymbol{r}^k)^\top \boldsymbol{p}^k}{(\boldsymbol{p}^k)^\top \boldsymbol{A}\boldsymbol{p}^k}$

$$\boldsymbol{x}^{k+1} = \boldsymbol{x}^k + \alpha_k \boldsymbol{p}^k \quad \text{and} \quad \boldsymbol{r}^{k+1} = \boldsymbol{r}^k + \alpha_k \boldsymbol{A}\boldsymbol{p}^k.$$

3. If $\|\boldsymbol{r}^{k+1}\| \leq \texttt{tol}$, then STOP and $\boldsymbol{x}^{k+1}$ is the output.
4. Calculate $\beta_{k+1} = \frac{(\boldsymbol{r}^{k+1})^\top \boldsymbol{A}\boldsymbol{p}^k}{(\boldsymbol{p}^k)^\top \boldsymbol{A}\boldsymbol{p}^k}$ and $\boldsymbol{p}^{k+1} = -\boldsymbol{r}^{k+1} + \beta_{k+1}\boldsymbol{p}^k$.

Figure 1.21: The CG Method

The CG method generates a set of conjugate directions and stops after at most n steps, returning a solution of the linear system.

1.7.4 Newton's Method

To solve $\min_{x\in\mathbb{R}^n} f(x)$ with $f:\mathbb{R}^n\to\mathbb{R}$ using Newton's method, we approximate the function by its second-order Taylor expansion at x^k:

$$f(x)\approx f(x^k)+\nabla f(x^k)^\top(x-x^k)+\frac{1}{2}(x-x^k)^\top\nabla^2 f(x^k)(x-x^k). \tag{1.7.4.1}$$

Minimizing this leads to:

$$x=x^k-(\nabla^2 f(x^k))^{-1}\nabla f(x^k). \tag{1.7.4.2}$$

where $d^k=-(\nabla^2 f(x^k))^{-1}\nabla f(x^k)$. is called the Newton direction. It's easy to find that $\nabla f(x)^\top \boldsymbol{d}=-\nabla f(x)^\top(\nabla^2 f(x))^{-1}\nabla f(x)$, so whether $\nabla f(x)^T d\le 0$ depends on whether $\nabla^2 f(x)$ is positive semi-definite.

The Newton Method

1. Initialization: Select an initial point $\boldsymbol{x}^0\in\mathbb{R}^n$.

For $k=0,1,\ldots$:

2. If $\|\nabla f(\boldsymbol{x}^k)\|\le$ `tol`, then STOP and $\boldsymbol{x}^k$ is the output.
3. Compute the Newton direction $\boldsymbol{d}^k$ which is the solution of the linear system
$$\nabla^2 f(\boldsymbol{x}^k)\boldsymbol{d}^k=-\nabla f(\boldsymbol{x}^k).$$
4. Choose a step size α_k by backtracking line search and calculate $\boldsymbol{x}^{k+1}=\boldsymbol{x}^k+\alpha_k\boldsymbol{d}^k$.

Figure 1.22: The Newton Method

Theorem 1.7.4.1: Quadratic Convergence of Newton's Method

Let f be twice continuously differentiable and let $x^\star$ be a local minimizer of f. For some $\epsilon>0$, if in $B_\epsilon(x^\star)$:

- f is strongly convex with μ;
- f is Lipschitz smooth with L.

Then $\{x^k\}_k$ follows:

$$\|x^{k+1}-x^*\|\le\frac{L}{2\mu}\|x^k-x^*\|^2. \tag{1.7.4.3}$$

In addition, if $\|x^0-x^*\|\le\frac{\mu\min\{1,\varepsilon\}}{L}$:

$$\|x^k-x^*\|\le\frac{2\mu}{L}\left(\frac{1}{2}\right)^{2^k},\quad k=0,1,2,\ldots \tag{1.7.4.4}$$

Remark 1.7.6

Comparison between gradient descent method and Newton's method:

- Newton takes fewer steps of iterations, but each iteration is more expensive;
- Newton requires more memory to store the Hessian matrix

Inexact Newton Steps

The idea of *inexact Newton methods* is to solve the linear system:

$$\nabla^2 f(x^k) \cdot d^k = -\nabla f(x^k) \tag{1.7.4.5}$$

only approximately to get a cheaper Newton step. In practice, the strategy aims to find an approximate direction d^k such that:

$$\|\nabla^2 f(x^k) d^k + \nabla f(x^k)\| \leq \text{to1}, \quad \text{to1} \geq 0. \tag{1.7.4.6}$$

When $tol = 0$, then $\nabla^2 f(x^k) \cdot d^k = -\nabla f(x^k)$, which is the original Newton step.

Proof 1.7.6: Inexact Newton Methods Can Work

$$\begin{aligned} \|x^{k+1} - x^*\| &= \|x^k + d^k - x^*\| \\ &= \|\nabla^2 f(x^k)^{-1} \cdot [\nabla^2 f(x^k)(x^k - x^*) + \nabla^2 f(x^k) d^k]\| \\ &\leq \|\nabla^2 f(x^k)^{-1}\| \cdot \|[\nabla^2 f(x^k)(x^k - x^*)] + [\nabla^2 f(x^k) d^k]\| \\ &= C_H \cdot (tol + o(\|x^k - x^*\|)) \end{aligned} \tag{1.7.4.7}$$

If tol_k satisfy $tol_k = o(\|x^k - x^*\|)$, then we can ensure $\|x^{k+1} - x^*\| = o(\|x^k - x^*\|)$. One popular example is to set $tol_k = \rho_k \cdot \|\nabla f(x^k)\|$, and $\rho_k \to 0$.

Q.E.D.

When $\nabla^2 f(x^k)$ is singular, we can regularize the optimization problem to:

$$\|[\nabla^2 f(x^k) + \delta_k I] d^k + \nabla f(x^k)\| \leq \text{to1}_k, \delta_k \geq 0. \tag{1.7.4.8}$$

To implement the inexact Newton method, we can combine it with the CG method:

Truncated Newton Method

1. Select $\boldsymbol{x}^0 \in \mathbb{R}^n$ and choose $\sigma, \gamma \in (0,1)$, and $\{\rho_k\}_k$.

For $k = 0, 1, \ldots$:

2. If $\|\nabla f(\boldsymbol{x}^k)\| \leq \varepsilon$, then STOP and $\boldsymbol{x}^k$ is the output.
3. Set $\boldsymbol{A} = \nabla^2 f(\boldsymbol{x}^k)$, $\boldsymbol{v}^0 = 0$, $\boldsymbol{r}^0 = \nabla f(\boldsymbol{x}^k)$, $\boldsymbol{p}^0 = -\boldsymbol{r}^0$, and $\texttt{tol} = \rho_k \cdot \|\nabla f(\boldsymbol{x}^k)\|$.
4. **For** $j = 0, 1, \ldots$:
   - If $(\boldsymbol{p}^j)^\top \boldsymbol{A}\boldsymbol{p}^j \leq 0$ return $\boldsymbol{d}^k = \boldsymbol{v}^j$ (or $\boldsymbol{d}^k = -\nabla f(\boldsymbol{x}^k)$ if $j = 0$).
   - Compute $\sigma_j = \frac{\|\boldsymbol{r}^j\|^2}{(\boldsymbol{p}^j)^\top \boldsymbol{A}\boldsymbol{p}^j}$, $\boldsymbol{v}^{j+1} = \boldsymbol{v}^j + \sigma_j \boldsymbol{p}^j$, $\boldsymbol{r}^{j+1} = \boldsymbol{r}^j + \sigma_j \boldsymbol{A}\boldsymbol{p}^j$.
   - If $\|\boldsymbol{r}^{j+1}\| \leq \texttt{tol}$, then STOP and return $\boldsymbol{d}^k = \boldsymbol{v}^{j+1}$.
   - Calculate $\beta_{j+1} = \frac{\|\boldsymbol{r}^{j+1}\|^2}{\|\boldsymbol{r}^j\|^2}$ and $\boldsymbol{p}^{j+1} = -\boldsymbol{r}^{j+1} + \beta_{j+1}\boldsymbol{p}^j$.
5. Calculate α_k via backtracking and set $\boldsymbol{x}^{k+1} = \boldsymbol{x}^k + \alpha_k \boldsymbol{d}^k$.

Figure 1.23: The Full Newton-CG Method

Quasi-Newton Methods

The principle idea of quasi-Newton methods is similar to inexact Newton-type methods. Computing the Hessian is expensive; We consider an approximation to the Taylor expansion:

$$f(x^k + d) \approx q_k(d) = f(x^k) + \nabla f(x^k)^\top d + \frac{1}{2} d^\top B_k d. \tag{1.7.4.9}$$

If B_k is PD, then we can yield the step $x^{k+1} = x^k + d = x^k - B_k^{-1}\nabla f(x^k)$. The idea of the quasi-Newton methods is that the gradient of q_{k+1} matches ∇f at point x^k and x^{k+1}. This is called the **Secant Condition**:

$$\nabla f(x^{k+1}) - \nabla f(x^k) = B_{k+1}(x^{k+1} - x^k). \tag{1.7.4.10}$$

Quasi-Newton methods generate a sequence of B_k based on this rule. However, the B_k generated by the secant condition may not be PSD, which may point to an ascent direction.

SR-1 Update

Let's denote $s^k = x^{k+1} - x^k$, and $y^k = \nabla f(x^{k+1}) - \nabla f(x^k)$, the **symmetric rank-1** updating rule uses a simple idea to update B_k by a *rank-1 update*:

$$B_{k+1} = B_k + \rho \boldsymbol{u}\boldsymbol{u}^\top, \quad \rho \in \{+1, -1\}, \quad \boldsymbol{u} \in \mathbb{R}^n. \tag{1.7.4.11}$$

Here, $\boldsymbol{u}\boldsymbol{u}^\top$ is a rank-1 matrix. Incorporating it into the Secant condition, we can get the formal SR-1 updating rule:

$$B_{k+1}^{SR-1} = B_k + \frac{(y^k - B_k s^k)(y^k - B_k s^k)^\top}{(s^k)^\top (y^k - B_k s^k)}. \tag{1.7.4.12}$$

Lemma 1.7.4.1: Sherman-Morrison-Woodbury Formula

Suppose $A \in \mathbb{R}^{n\times n}$ is invertible, and let $u, v \in \mathbb{R}^n$ be given, then $A + uv^T$ is invertible if and only if $1 + v^\top A^{-1}\boldsymbol{u} \neq 0$, or say:

$$(A + uv^\top)^{-1} = A^{-1} - \frac{A^{-1}uv^\top A^{-1}}{1 + v^\top A^{-1}u} \tag{1.7.4.13}$$

Using the SMW formula, we can explicitly calculate the inverse of the SR-1 update. (Because to implement the Newton method using the approximation B_k, we need the inverse of B_k to update every step).

$$H_{k+1}^{\mathrm{SR}-1} = H_k + \frac{(s^k - H_k y^k)(s^k - H_k y^k)^\top}{(s^k - H_k y^k)^\top y^k}. \tag{1.7.4.14}$$

Remark 1.7.7

The SR-1 update doesn't maintain the PD of H_k, which means $d^k = -H_k \nabla f(x^k)$ is not necessarily a descent direction.

SR-2: BFGS

This is the most successful quasi-Newton method, updated by:

$$B_{k+1} = B_k + \gamma^\top + \delta v v^\top. \tag{1.7.4.15}$$

We use the **Broyden-Fletcher-Goldfarb-Shanno-update** (BFGS) to update H_k:

$$H_{k+1}^{\mathrm{BFGS}} = H_k + \frac{w^k (s^k)^\top + s^k (w^k)^\top}{(s^k)^\top y^k} - \frac{(w^k)^\top y^k}{((s^k)^\top y^k)^2} \cdot s^k (s^k)^\top, \tag{1.7.4.16}$$

where $w^k = s^k - H_k y^k$

Globalized BFGS-Method

1. Initialization: Select an initial point $\boldsymbol{x}^0 \in \mathbb{R}^n$ and a symmetric, pos. def. matrix $\boldsymbol{H}_0 \in \mathbb{R}^{n\times n}$. Choose $\sigma, \gamma \in (0, 1)$.

For $k = 0, 1, \ldots$:

2. Compute the quasi-Newton direction $\boldsymbol{d}^k = -\boldsymbol{H}_k \nabla f(\boldsymbol{x}^k)$.
3. Calculate a step size α_k using Armijo line search.
4. Set $\boldsymbol{x}^{k+1} = \boldsymbol{x}^k + \alpha_k \boldsymbol{d}^k$. If $\|\nabla f(\boldsymbol{x}^{k+1})\| \leq \varepsilon$, then STOP.
5. Set $\boldsymbol{s}^k = \boldsymbol{x}^{k+1} - \boldsymbol{x}^k$ and $\boldsymbol{y}^k = \nabla f(\boldsymbol{x}^{k+1}) - \nabla f(\boldsymbol{x}^k)$. If $(\boldsymbol{s}^k)^\top \boldsymbol{y}^k < 0$ set $\boldsymbol{H}_{k+1} = \boldsymbol{H}_k$, otherwise compute

$$\boldsymbol{H}_{k+1} = \boldsymbol{H}_k + \frac{(\boldsymbol{s}^k - \boldsymbol{H}_k \boldsymbol{y}^k)(\boldsymbol{s}^k)^\top + \boldsymbol{s}^k (\boldsymbol{s}^k - \boldsymbol{H}_k \boldsymbol{y}^k)^\top}{(\boldsymbol{s}^k)^\top \boldsymbol{y}^k} - \frac{(\boldsymbol{s}^k - \boldsymbol{H}_k \boldsymbol{y}^k)^\top \boldsymbol{y}^k}{((\boldsymbol{s}^k)^\top \boldsymbol{y}^k)^2} \cdot \boldsymbol{s}^k (\boldsymbol{s}^k)^\top.$$

Figure 1.24: Globalized BFGS-Method

Remark 1.7.8

- If B_k is PD, then B_{k+1}^{BFGS} is also PD;
- To compute d^k, the Hessian is not required, and no linear system needs to be solved;
- Only matrix-vector multiplication is required, which is much faster than solving a linear system;
- If f is strongly convex, then fast local convergence can be shown.

1.7.5 Large-Scale Optimization

Inpainting

TV-Huber problem

Assume we have access to a binary mask matrix $\in \mathbb{R}^{m\times n}$ with:

$$\text{Mask}_{ij} = \begin{cases} 1 & \text{the pixel } (i,j) \text{is not damaged,} \\ 0 & \text{the pixel } (i,j) \text{is damaged.} \end{cases} \tag{1.7.5.1}$$

Then define mask $:= vec(\text{Mask}_{ij}) \in \mathbb{R}^{mn}, s := \sum_{i=1}^{mn} \text{mask}_i$ as the number of undamaged pixels, and $\mathcal{I} := \{i : \text{mask}_i = 1\} = \{q_1, q_2, \cdots, q_s\}$. Then we can define the slection matrix:

$$\text{A}=\begin{bmatrix} - & e_{q_1}^\top & - \\ & \vdots & \\ - & e_{q_s}^\top & - \end{bmatrix} \in \mathbb{R}^{s\times mn}, \quad [\mathbf{e}_j]_i = \begin{cases} 1 & \text{if } i = j, \\ 0 & \text{otherwise,} \end{cases} \quad \forall j \in \mathcal{I}. \tag{1.7.5.2}$$

Let $U \in \mathbb{R}^{m\times n}$ be the *original undamaged image*, and $u = vec(U) \in \mathbb{R}^{mn}$. Then, we can formalize $b = Au \in \mathbb{R}^s$ as the information of all undamaged pixels in U. Just solving $Ax = b$ can produce infinite solutions. Instead, we inpaint via **total variation minimization**:

$$\min_x \frac{1}{2}\|Ax - b\|^2 + \mu \sum_{i=1}^{m-1}\sum_{j=1}^{n-1} \|D_{(i,j)}\boldsymbol{x}\|_2, \tag{1.7.5.3}$$

where μ is a regularization parameter and $D_{(i,j)} \in \mathbb{R}^{2\times mn}$ is the *image gradient* at pixel (i,j). $D_{(i,j)}x = (\delta_1, \delta_2)^\top$, where

$$\delta_1 = X_{(i+1)j} - X_{ij} \quad \text{and} \quad \delta_2 = X_{i(j+1)} - X_{ij}. \tag{1.7.5.4}$$

However, the L2-norm $\|\cdot\|_2$ is not smooth at 0, hence, we use **Huber-norm** to smooth the objective function:

$$\varphi_{\text{hub}}(\mathbf{y}) = \begin{cases} \frac{1}{2\delta}\|\mathbf{y}\|^2 & \text{if } \|\mathbf{y}\| \le \delta, \\ \|\mathbf{y}\| - \frac{1}{2}\delta & \text{if } \|\mathbf{y}\| > \delta, \end{cases} \quad \delta > 0; \tag{1.7.5.5}$$

$$\min_x \frac{1}{2}\|Ax - b\|^2 + \mu \sum_{i=1}^{m-1}\sum_{j=1}^{n-1} \varphi_{\text{hub}}(D_{(i,j)}x). \tag{1.7.5.6}$$

This is called **TV-Huber problem**.

- TV-Huber problem is convex;
- Newton's method can not be applied to this problem (Huber norm is not twice continuously differentiable).

Reconstructing Sparse Signals

Given (A, b) and an appropriate basis Ψ, which can transform Ψx into a sparse matrix. To recover the original image x, we can model it to a $\mathcal{L}_0$ problem:

$$\min_x \quad \|\Psi x\|_0 \quad \text{s.t.} \quad Ax = b \tag{1.7.5.7}$$

This is a combinatorial problem; we can transform it into a convex problem:

$$\min_x \quad \|\Psi x\|_1 \quad \text{s.t.} \quad Ax = b \tag{1.7.5.8}$$

But it's still undifferentiable. We can use *compressed sensing* to transform it into:

$$\min_x \quad \frac{1}{2}\|A\Psi^{-1}x - b\|_2^2 + \mu \cdot \varphi(x), \quad \mu > 0. \tag{1.7.5.9}$$

Where φ is a smooth and non-convex alternative to the $\mathcal{L}_1$ norm:

$$\varphi(x) = \sum_{i=1}^{mn} \log\left(1 + \frac{x_i^2}{\nu}\right), \quad \nu > 0. \tag{1.7.5.10}$$

1.7.6 Optimization Acceleration

Nosterov's Extrapolation: AGM

The principle idea of many acceleration techniques is to perform an extrapolation step:

$$y^{k+1} = x^k + \beta_k(x^k - x^{k-1}), \quad \beta_k > 0 \tag{1.7.6.1}$$

to approximate the next iteration $y^{k+1} \approx x^{k+1}$.

Abstract Accelerated Gradient Method (AGM)

1. Initialization: Choose a point $\boldsymbol{x}^0 \in \mathbb{R}^n$ and set $\boldsymbol{x}^{-1} = \boldsymbol{x}^0$.

For $k = 0, 1, \ldots$:

2. Select an extrapolation parameter β_k and compute the step $\boldsymbol{y}^{k+1} = \boldsymbol{x}^k + \beta_k(\boldsymbol{x}^k - \boldsymbol{x}^{k-1})$.
3. Select a step size $\alpha_k > 0$ and set $\boldsymbol{x}^{k+1} = \boldsymbol{y}^{k+1} - \alpha_k \nabla f(\boldsymbol{y}^{k+1})$.

Figure 1.25: Abstract Accelerated Gradient Method

To see a graphical illustration:

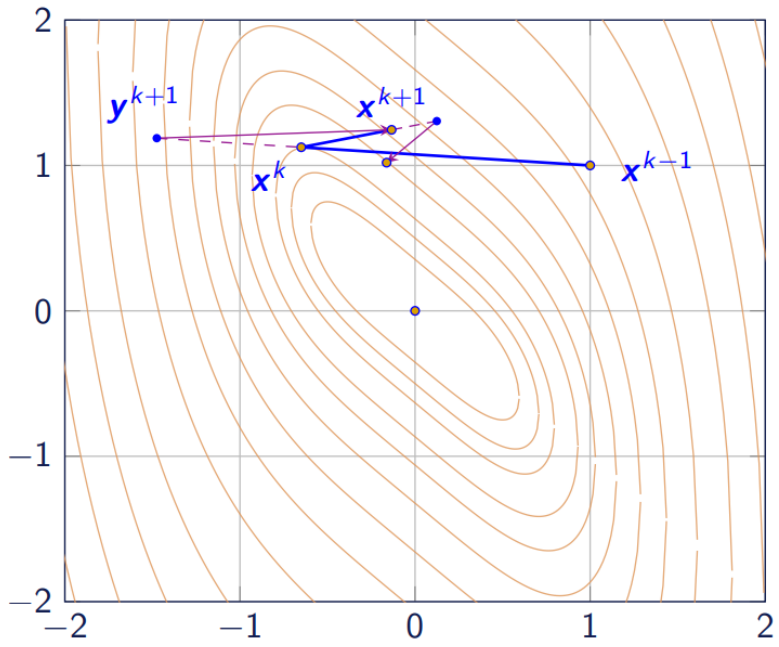


Figure 1.26: AGM illustration

The challenge here is to decide the parameter β_k to guarantee faster convergence.

Remark 1.7.9

A good selection of β_k should satisfy:

1. $\beta_k = \frac{\theta_k(1-\theta_{k-1})}{\theta_{k-1}} = \frac{\theta_k}{\theta_{k-1}} - \theta_k$;
2. $\theta_{-1} = \theta_0 = 1$;
3. $0 \leq \theta_k \leq \frac{2}{k+2}$;
4. $\frac{1-\theta_{k+1}}{\theta_{k+1}^2} \leq \frac{1}{\theta_k^2}$.

A popular choice is $\beta_k = \frac{k-2}{k+1}$.

Theorem 1.7.6.1: Improved Complexity of the Acceleration

Let f be convex and Lipschitz smooth with L, and assume the solution to $\min_x f(x)$ is non-empty. Let $\{x^k\}_k, \{\alpha_k\}_k, \{\beta_k\}_k$ be generated by the accelerated gradient method with $\alpha_k = \bar{\alpha} \in (0, \frac{1}{L}]$, and β_k satisfies the condition listed above. Then we have:

$$f(x^k) - f(x^*) \leq \frac{2\|x^0 - x^*\|^2}{\bar{\alpha}(k+1)^2} \quad \forall\, k \in \mathbb{N}. \tag{1.7.6.2}$$

Remark 1.7.10

Only $O(\varepsilon^{-\frac{1}{2}})$ steps s are required to ensure $f(x^k) - f(x^*) \leq \varepsilon$!

Sometimes the Lipschitz constant L is unknown, then we can determine $_k$ through:

- Choose $\sigma \in (0,1)$;
- Set $\alpha_k = \alpha_{k-1}$ and $x^{k+1} = y^{k+1} - \alpha_k \nabla f(y^{k+1})$;
- While $f(x^{k+1}) > f(y^{k+1}) - \frac{\alpha_k}{2}\|\nabla f(y^{k+1})\|^2$: set $\alpha_k = \sigma\alpha_k$ and repeat.

These steps work because of the descent lemma. If f is Lipschitz smooth, α_k can be found after finite iterations.

Momentum and IGM

The momentum trick (or the inertial gradient method: IGM) considers the same extrapolation step:

$$y^{k+1} = x^k + \beta_k(x^k - x^{k-1}), \quad \beta_k > 0 \tag{1.7.6.3}$$

But the new iterate x^{k+1} follows:

$$x^{k+1} = y^{k+1} - \alpha_k \nabla f(x^k) = x^k - \alpha_k \nabla f(x^k) + \beta_k(x^k - x^{k-1}). \tag{1.7.6.4}$$

The gradient is now evaluated at x_k and not at y_{k+1}! The updating formula contains the term $\beta_k(x^k - x^{k-1})$, is called the momentum. Compared to AGM, IGM doesn't enjoy the same convergence guarantee, but it's still popular, especially in non-convex scenarios.

1.8 Constrained Optimization

General nonlinear optimization problem:

$$\begin{aligned} \text{minimize}_{\mathbf{x}\in\mathbb{R}^n} \quad & f(\mathbf{x}) \\ \text{subject to} \quad & g_i(\mathbf{x}) \le 0, \quad \forall\, i = 1,\dots,m, \\ & h_j(\mathbf{x}) = 0, \quad \forall\, j = 1,\dots,p. \end{aligned} \tag{1.8.0.1}$$

The goal is to *derive optimality conditions* for this more general class of problems.

Remark 1.8.1

Some subclasses of optimization:

1. When all functions are affine-linear: linear optimization problem;
2. If $f(x) = \frac{1}{2}x^\top Cx + c^\top x$, and $C \in \mathbb{R}^{n\times n}$ is symmetric: quadratic optimization problem;

FONC for Constrained Optimization

Definition 1.8.0.1: Feasible Direction

Given $x \in X$, we call d a **feasible direction** at x if exists $\bar{t} > 0$ such that $x + td \in X$ for all $0 \le t \le \bar{t}$.

Remark 1.8.2

For example, if $\mathcal{X} = \{x : Ax = b\}$, then all feasible directions at x are given by $\{d : Ad = 0\}$.

Theorem 1.8.0.1: FONC for Constrained Problems

Let x^* be a local minimum of $\min_{x\in X} f(x)$. Then for *any* feasible direction d at x^*, we must have $\nabla f(x^*)^\top d \ge 0$.

When x^* lies in the *interior* of X, or the unconstrained case, it can be reduced to $\nabla f(x^*) = 0$.

More than the FONC, we want to generalize the optimal conditions for general nonlinear problems.

Definition 1.8.0.2: Active and Inactive Constraint

Reconsidering the general nonlinear optimization problem, at point $x \in X$, the set $\mathcal{A}(x) := \{i : g_i(x) = 0\}$ denotes the set of **active constraints**. $\mathcal{I}(x) := \{i : g_i(x) < 0\}$ denotes **inactive constraints**.

Lemma 1.8.0.1: Farkas' Lemma

Let $A \in \mathbb{R}^{m\times n}$, $B \in \mathbb{R}^{p\times n}$, and $c \in \mathbb{R}^n$ be **given**. The following two statements are equivalent:

1. For all $d \in \mathbb{R}^n$ with $Ad < 0$ and $Bd = 0$ we have $c^T d \le 0$;
2. There exists $u \in \mathbb{R}^m, u \ge 0$, and $v \in \mathbb{R}^p$ with $c = A^T u + B^T v$.

These results remain valid in the cases $m = 0$ and $p = 0$.

Theorem 1.8.0.2: FONC for Linearly Inequality Constrained Problems

If $x^\star$ is a local minimum of:

$$\text{minimize}_x f(x) \quad \text{s.t.} \quad Ax \leq b, \tag{1.8.0.2}$$

then there exists some $y \geq 0$ with:

$$\begin{aligned} &\nabla f(x^*) + A^\top y = 0 \\ &y_i \cdot (a_i^\top x^* - b_i) = 0 \quad \forall\, i, \end{aligned} \tag{1.8.0.3}$$

where a_i^T is the i-th row of A.

Proof 1.8.1

Since x^* is a local minimum, so $\nabla f(x^*)d \geq 0$ for every feasible direction d at x^*.
Now considering d, d is feasible if $a_i^T d \leq 0$ for all i with $a_i^T x^* = b_i$, which are active constraints $\mathcal{A}$.
Let's define:

$$\tilde{A} = (a_i^T)_{i \in \mathcal{A}(x^*)} \in \mathbb{R}^{|\mathcal{A}| \times n} \tag{1.8.0.4}$$

Now we can refine the optimality condition as: for all d with $\tilde{A}d \leq 0$, we have $-\nabla f(x^*)d \leq 0$. By applying the Farkas' lemma, by setting $A = \tilde{A}, B = 0, c = -\nabla(x^*)$, we can transform the optimality condition to:

$$\exists u \in \mathbb{R}^{|\mathcal{A}|} \geq 0, \text{ with } -\nabla f(x^*) = \tilde{A}^T u.$$

Now we can define $y \in \mathbb{R}^m$ with $y_{\mathcal{A}}(X^*) := u$ and $y_{\mathcal{I}}(X^*) := 0$.

Q.E.D.

Theorem 1.8.0.3: FONC for Linearly Equality Constrained Problems

The FONC condition for equality constraint is easier than the inequality constraint case. If x^* is a local minimum for $\min_x f(x) \quad \text{s.t.} \quad Ax = b$, the FONC is:

$$\exists y \quad s.t. \nabla f(x^*) + A^\top y = 0. \tag{1.8.0.5}$$

Compared to the FONC for inequality problems, the condition reduces the term $y_i \cdot (a_i^\top x^* - b_i) = 0$. This is easy to understand since every constraint is active, so $a_i^\top x^* - b_i = 0$.
Another difference is that y doesn't need to be greater than 0.

The KKT Conditions

Now, we can extend the FONC for the general nonlinear optimization problem; this is called the **KKT** condition, named by *H. Kuhn*, *A. Tucker* and *W.Karush*. The first step is to construct the **Lagrangian multiplier**: $g_i(x) \leq 0 \cdots \lambda_i$, $h_j(x) = 0 \cdots \mu_j$. Then we define the **Lagrangian** of the problem:

$$L(x, \lambda, \mu) := f(x) + \sum_{i=1}^{m} \lambda_i g_i(x) + \sum_{j=1}^{p} \mu_j h_j(x). \tag{1.8.0.6}$$

We require "$\lambda_i \geq 0$, μ_i free". These conditions are called **dual feasibility** conditions. Taking the derivative with respect to x and setting it to zero forms the main condition of KKT conditions. Besides, we have a set of complementary conditions: $\lambda_i \cdot g_i(x) = 0, \forall i$.

1. Main Condition

$$\nabla f(\boldsymbol{x}) + \sum_{i=1}^{m} \lambda_i \nabla g_i(\boldsymbol{x}) + \sum_{j=1}^{p} \mu_j \nabla h_j(\boldsymbol{x}) = \mathbf{0}.$$

2. Dual Feasibility

$$\lambda_i \geq 0 \quad i = 1, \ldots, m.$$

3. Complementarity

$$\lambda_i \cdot g_i(x) = 0 \quad \forall\, i = 1, \ldots, m.$$

We often add primal feasibility as part of the KKT conditions:

4. Primal Feasibility

$$g_i(\boldsymbol{x}) \leq 0, \quad h_j(\boldsymbol{x}) = 0 \quad \forall\, i, \quad \forall\, j.$$

Figure 1.27: KKT Conditions

Remark 1.8.3

Linearly Independent Constraint Qualification
We require the collection of gradients:

$$\{\nabla g_i(x^*) : i \in \mathcal{A}(x^*)\} \cup \{\nabla h_j(x^*) : j = 1, \ldots, p\} \tag{1.8.0.7}$$

to be **linearly independent** (full rank). A feasible point satisfying the LICQ is called *regular*.

Remark 1.8.4

A (feasible) point satisfying the KKT conditions is called a KKT point. The KKT conditions are only necessary conditions, meaning the KKT points are only candidates for local optimal solutions — just like stationary points.

Proposition 1.8.0.1: Convexity and KKT Conditions
We call the problem $\min_{x \in \mathbb{R}^n} f(x)$ s.t. $g(x) \leq 0$, $h(x) = 0$ convex if f, g_i are convex and h is an affine-linear mapping. If a problem is convex, we have the following propositions:

- Every local solution is a global solution;
- If $x^* \in X$ is the global solution and assume LICQ holds, then the KKT-conditions are satisfied;
- If x^* is a KKT point, then x^* is a global solution.

Definition 1.8.0.3: Slater's Condition

If a problem's constraints are convex (affine-linear), we say that Slater's condition holds if:

$$\exists \hat{x}: \quad g_i(\hat{x}) < 0, \quad i = 1, \dots, m, \quad h(\hat{x}) = 0. \tag{1.8.0.8}$$

Then proposition 2 above can be reduced to:

- Every global solution $x^* \in X$ satisfies the KKT conditions.

SOC for Constrained Optimization

If we assume that f, g_i, h_j are twice continuously differentiable, the Hessian of the Lagrangian is given by:

$$\nabla^2_{xx} L(x, \lambda, \mu) = \nabla^2 f(x) + \sum_{i=1}^{m} \lambda_i \nabla^2 g_i(x) + \sum_{j=1}^{p} \mu_j \nabla^2 h_j(x). \tag{1.8.0.9}$$

Definition 1.8.0.4: Critical Cone

$$\begin{aligned} C(\mathbf{x}) := \{ \boldsymbol{d} \in \mathbb{R}^n : & \nabla f(\mathbf{x})^\top \boldsymbol{d} = 0, \\ & \nabla g_i(\mathbf{x})^\top \boldsymbol{d} \leq 0, \ \forall\, i \in \mathcal{A}(\mathbf{x}), \\ & \nabla h_j(\mathbf{x})^\top \boldsymbol{d} = 0, \ \forall j \}. \end{aligned} \tag{1.8.0.10}$$

Theorem 1.8.0.4: SOC for Constrained Problems

Let x^* be a regular point and local minimum. Then, the KKT conditions hold and there are **unique** (followed from LICQ) multiplier $\lambda \in \mathbb{R}^m$ and $\mu \in \mathbb{R}^p$ such that:

$$\begin{aligned} & \nabla f(x^*) + \nabla g(x^*)\lambda + \nabla h(x^*)\mu = 0, \\ & g(x^*) \leq 0, \quad h(x^*) = 0, \quad \lambda \geq 0, \quad \lambda_i \cdot g_i(x^*) = 0 \quad \forall i \end{aligned} \tag{1.8.0.11}$$

If we have:

$$d^\top \nabla^2_{xx} L(x^*, \lambda, \mu) d \geq 0 \quad \forall\, d \in C(x^*). \tag{1.8.0.12}$$

These conditions compose the SONC. The SOSC is just to modify the last inequation to:

$$d^\top \nabla^2_{xx} L(x^*, \lambda, \mu) d > 0 \quad \forall\, d \in C(x^*) \backslash \{0\}. \tag{1.8.0.13}$$

Unconstrained	Constrained
First-Order Cond.: x^* local minimum (+ LICQ)	
▸ $\nabla f(x^*) = 0$.	▸ KKT-conditions.
Second-Order Cond.: x^* local minimum (+ LICQ)	
▸ $\nabla f(x^*) = 0$ ▸ $\nabla^2 f(x^*)$ is positive semi-definite (on $\mathbb{R}^n$).	▸ KKT-conditions ▸ $\nabla^2_{xx} L(x^*, \lambda, \mu)$ is positive semidefinite on $\mathcal{C}(x^*)$.
Second-Order Sufficient Cond.	
▸ $\nabla f(x^*) = 0$ and ▸ $\nabla^2 f(x^*)$ is positive definite (on $\mathbb{R}^n$).	▸ x^* is KKT-point and ▸ $\nabla^2_{xx} L(x^*, \lambda, \mu)$ is positive definite on $\mathcal{C}(x^*)$.
$\Longrightarrow$ x^* is strict local minimum	

Figure 1.28: Second-OrderConditions: Comparison

Remark 1.8.5

The strategy for solving nonlinear constrained programs:

1. Check LICQ (if required);
2. Derive KKT conditions;
3. Discuss different easy cases via the complementarity conditions (set multiplier or constraints to 0) to find all KKT points;
4. Calculate $C(x)$ and $\nabla^2_{xx} L(x, \lambda, \mu)$ at KKT points;
5. Check the SOC.

Other notes for solving the problem:

- Check if f is coercive or bounded: has global solutions;
- If the LICQ holds, then λ and μ are always unique.

Algorithms for Constrained Problems

The first class of algorithm is the **penalty method**; the basic idea is to solve the constrained problem via a sequence of unconstrained optimization problems, which are built by adding penalty terms with positive penalty parameters for the constraints to the objective function.

$$\min_{x \in \mathbb{R}^n} f(x) + \alpha p(x), \quad \alpha > 0, \tag{1.8.0.14}$$

where $p : \mathbb{R}^n \to \mathbb{R}$ satisfies: $p(x) = 0$ for all $x \in X$ and $p(x) > 0$ on $\mathbb{R}\backslash X$.
The procedure is given below:

1. Generate a sequence of penalty problems corresponding to a growing sequence of penalty parameters $\{\alpha_k\}_k$;
2. The solution x^k of the k-th subproblem will be used as the initial point for the next problem.

The Quadratic Penalty Function

$$\begin{aligned} P_\alpha(x) &:= f(x) + \frac{\alpha}{2}\sum_{i=1}^{m}(\max\{0, g_i(x)\})^2 + \frac{\alpha}{2}\sum_{j=1}^{p} h_j(x)^2 \\ &= f(x) + \frac{\alpha}{2}\|(g(x))_+\|^2 + \frac{\alpha}{2}\|h(x)\|^2. \end{aligned} \tag{1.8.0.15}$$

P_α is once continuously differentiable with:

$$\nabla P_\alpha(x) = \nabla f(x) + \alpha\sum_{i=1}^{m}(g_i(x))_+\nabla g_i(x) + \alpha\sum_{j=1}^{p} h_j(x)\nabla h_j(x). \tag{1.8.0.16}$$

The Quadratic Penalty Method

1. Initialization: Choose an initial point $\boldsymbol{x}^{-1} \in \mathbb{R}^n$ and $\alpha_0 > 0$.

For $k = 0, 1, \ldots$:

2. Calculate the global solution $\boldsymbol{x}^k$ of the penalty problem

$$\min_{\boldsymbol{x}\in\mathbb{R}^n} P_{\alpha_k}(\boldsymbol{x}).$$

   Here, we often utilize $\boldsymbol{x}^{k-1}$ as initial point.

3. Terminate if $\boldsymbol{x}^k \in X$. Otherwise select $\alpha_{k+1} > \alpha_k$.

Figure 1.29: The Quadratic Penalty Method

By applying the penalty method, every accumulation point of $\{x^k\}_k$ is a *global solution*.

Projected Gradient Method

Applying the gradient method to the constraint problem directly may produce x^k outside the feasible set. One idea is to project x^k onto the feasible set.

Definition 1.8.0.5: Euclidean Projections

Let X be the convex feasible set. The (Euclidean) projection of x is defined as:

$$\min_y \frac{1}{2}\|x - y\|^2 \quad \text{s.t.} \quad y \in X \tag{1.8.0.17}$$

We write $y^* = \mathcal{P}X(x)$, which is the point in X that has the minimum distance to x.

For example, if $X = \{x : Ax = b\}$ and X has full row rank, then it holds that: $\mathcal{P}_X(\mathbf{x}) = \mathbf{x} - \mathbf{A}^\top(\mathbf{A}\mathbf{A}^\top)^{-1}[\mathbf{A}\mathbf{x} - \mathbf{b}]$. Different constraints correspond to different $\mathcal{P}_X(x)$.

1.9 Nonsmooth Optimization

1.10 Simulation

1.10.1 Variance Reduction Technique

Control Variates

Suppose we want to estimate a parameter with mean μ, and we have an unbiased statistic X. Suppose we have another similar pair Y, η, then $X' = X + c(Y - \eta)$ is also an unbiased estimater.

$$\begin{aligned} \mathbb{V}[X'] &= \mathbb{V}[X + c(Y - \eta)] \\ &= \mathbb{V}[X] + c^2\mathbb{V}[Y - \eta] + 2c\mathrm{Cov}(X, Y - \eta) \\ &= \mathbb{V}[Y - \eta]c^2 + 2\mathrm{Cov}(X, Y)c + \mathbb{V}[X]. \end{aligned} \tag{1.10.1.1}$$

To minimize this objective function, set $c = -\frac{\mathrm{Cov}(X,Y)}{\mathrm{V}[Y]}$, we can get:

$$\mathbb{V}[X'] = (1 - \mathrm{Corr(X,\ Y)})\,\mathbb{V}[X], \tag{1.10.1.2}$$

which reduces the original variance.

Chapter 2

Economics and Econometrics

2.1 Game Theory and Mechanism Design

The contents of this section referenced [Fudenberg and Tirole, 1991].

2.1.1 Dynamic & Static, Complete & Incomplete Information

Static game

Definition 2.1.1.1: Game

A strategic (normal form) game $G = (N, (A_i)_{i\in N}, (u_i)_{i\in N})$ consists of:

- a finite set of players N;
- A non-empty set of actions for each player: $A_i, i \in N$;
- A payoff function for each player: $u_i : A \to \mathbb{R}(A = \prod_{i\in N} A_i)$.

Theorem 2.1.1.1: von Neumann-Morgenstern's Utility Theorem

Let C be a set of outcomes and P be the set of distributions over outcomes; a utility function $U : P \to \mathbb{R}$ has an expected utility form if there exists a function $u : C \to \mathbb{R}$ s.t. $U(x) = \sum_{c\in C} x(c)u(c)$ for all $x \in P$. In this case, $U(\cdot)$ is called an expected utility function, and $u(\cdot)$ is called a **VNM utility function**.

The VNM ([Von Neumann and Morgenstern, 1994]) is an axiomatic way to formalize the preference. If a preference $\gtrsim$ among the lotteries (distributions) $U(\cdot)$ has an expected utility representation (follow the following axioms):

- *Completeness*: for any two lotteries (distributions) x and y, at least one of $x \gtrsim y$ or $y \gtrsim x$ holds;
- *Transitivity*: if $x \gtrsim y$ and $y \gtrsim z$, then $y \gtrsim z$;
- *Continuity*: if $x \gtrsim y \gtrsim z$, then exists $p \in [0, 1]$ s.t. $px + (1 - p)z \sim y$;
- *Independence*: For any lottery z and $p \in [0, 1]$, $x \gtrsim y$ iff $px + (1 - p)z \gtrsim py + (1 - p)z$.

, there exists a VNM utility function u such that for every lottery in U:

$$P \succsim Q \iff \sum_{x \in X} P(x)u(x) \geq \sum_{x \in X} Q(x)u(x) \tag{2.1.1.1}$$

Proof 2.1.1

The proof is finding and constructing u_i that satisfies the axioms. Suppose there are n outcomes $C_1, \cdots, C_n$, by the axioms of completeness and transitivity, we can rank them by order:

$$C_1 \lesssim C_2 \lesssim \cdots \lesssim C_n$$

In those circumstance where $C_1 \sim C_n$, the therom holds obviously; otherwise $C_1 \lesssim C_n$, we can define:

$$L(p) = p \cdot C_1 + (1-p) \cdot C_n, 0 \leq p \leq 1$$

By the axiom of continuity, for any C_i, we can always find a p_i such that:

$$L(p_i) \sim C_i$$

and

$$C_1 \lesssim L(p_i) \lesssim C_n \forall i$$

Then we can define the VNM utility for every outcome as $L(p_i)$, and the expected utility for every lottery $P = sum_i \alpha_i C_i$ as

$$U(P) = u(\sum_i \alpha_i C_i) = \sum_i \alpha_i u(C_i)$$

Since the decision maker is indifferent among C_i and $p_i \cdot C_1 + (1-p_i) \cdot C_n$, by the reduction axiom, we can obtain:

$$C_i = \frac{u(C_i) - u(C_1)}{u(C_n) - u(C_1)} C_1 + \frac{u(C_n) - u(C_i)}{u(C_n) - u(C_1)} C_n \tag{2.1.1.2}$$

This means the lottery C_i s, in effect, a lottery in which the best outcome is won with probability $\frac{u(C_i)-u(C_1)}{u(C_n)-u(C_1)}$, and the worst outcome otherwise. Note that $\frac{u(C_i)-u(C_1)}{u(C_n)-u(C_1)}$ is positive first-order linear on $u(C_i)$, so a rational decision maker would prefer the lottery P to Q ($P \succsim Q$) if and only if $u(C_p) \geq u(C_q)$.

Q.E.D.

Definition 2.1.1.2: Best Response (BR)

$a_i \in A_i$ is a best response to $a_{-i} \in A_{-i}$ if $u_i(a_i, a_{-i}) \geq u_i(a_i', a_{-i})$ $\forall a_i' \in A_i$. The best response $BR_i(a_{-i})$ is not a function, but a **correspondence** (there may be multiple BRs).

Theorem 2.1.1.2: Existence of NE (Nash 1951)

Any game with a finite set of players and a finite set of strategies has an NE of mixed strategies (this theorem emphasizes on *finite games*).

Lemma 2.1.1.1: Kakutani Fixed Point

Let $f : A \rightrightarrows A$ be a correspondence, if:

- A is a compact and convex set;
- $f(x)$ is non-empty for all $x \in A$;
- for all $x \in A$, $f(x)$ is a convex set;
- $f(x)$ has a closed graph: if $\{x^n, y^n\} \to \{x, y\}$ with $y^n \in f(x^n)$, then $y \in f(x)$

Then f has a fixed point: $\exists x \in A$, s.t. $x \in f(x)$.

Proof 2.1.2

First, we show that the action set $\Sigma = \prod_{i \in \mathcal{I}} \Sigma_i$ is compact, convex, and non-empty. This conclusion is quite natural, since each $\Sigma_i = \{x \mid \sum_j x_j = 1\}$ is a simplex, thus compact, convex, and non-empty, so is the product. Then we note the best correspondence $B(\sigma)$ is not empty because:

$$B_i(\sigma_{-i}) = \arg\max_{x \in \Sigma_i} u_i(x, \sigma_{-i})$$

is non-empty by Weirstrass's theorem.

Next, we show that $B(\sigma)$ is a convex set for all σ. When $\|B(\sigma)\| = 1$, it's a convex set by nature. When $B(\sigma)$ is not a single element, we take σ_i', σ_i'' to denote any two elements among $B(\sigma_{-i})$:

$$\begin{aligned} u_i(\sigma_i', \sigma_{-i}) &\geq u_i(\tau_i, \sigma_{-i}) \quad \text{for all } \tau_i \in \Sigma_i, \\ u_i(\sigma_i'', \sigma_{-i}) &\geq u_i(\tau_i, \sigma_{-i}) \quad \text{for all } \tau_i \in \Sigma_i. \end{aligned} \tag{2.1.1.3}$$

We can show that for all $\lambda \in (0, 1)$:

$$\lambda u_i(\sigma_i', \sigma_{-i}) + (1-\lambda) u_i(\sigma_i'', \sigma_{-i}) \geq u_i(\tau_i, \sigma_{-i}) \quad \text{for all } \tau_i \in \Sigma_i. \tag{2.1.1.4}$$

By the linearity of u_i:

$$u_i(\lambda\sigma_i' + (1-\lambda)\sigma_i'', \sigma_{-i}) \geq u_i(\tau_i, \sigma_{-i}) \quad \text{for all } \tau_i \in \Sigma_i. \tag{2.1.1.5}$$

Therefore, $\lambda\sigma_i' + (1-\lambda)\sigma_i'' \in B_i(\sigma_{-i})$, indicating that $B(\sigma)$ is a convex correspondence.

At last, we prove that $B(\sigma)$ is a closed graph by contradiction. Suppose not, then exists a sequence $(\sigma^n, \hat{\sigma}^n) \to (\sigma, \hat{\sigma})$ with $\hat{\sigma}^n \in \mathcal{B}(\sigma^n)$, but $\hat{\sigma}_i \notin B_i(\sigma_{-i})$. This implies for any $\epsilon > 0$, there always exists some σ_i' such that:

$$u_i(\sigma_i', \sigma_{-i}) > u_i(\hat{\sigma}_i, \sigma_{-i}) + 3\epsilon.$$

But by the fact that $\sigma_{-i}^n \to \sigma_{-i}$, we can find n such that:

$$u_i(\sigma_i', \sigma_{-i}^n) \geq u_i(\sigma_i', \sigma_{-i}) - \epsilon.$$

This leads us to:

$$u_i(\sigma_i', \sigma_{-i}^n) > u_i(\hat{\sigma}_i, \sigma_{-i}) + 2\epsilon \geq u_i(\hat{\sigma}_i^n, \sigma_{-i}^n) + \epsilon. \tag{2.1.1.6}$$

By the Kakutani Fixed Point theorem, for any σ, $B(\sigma)$ is not empty.
Q.E.D.

Definition 2.1.1.3: Strict Dominance

$a_i \in A_i$ if **strictly dominated** if a_i is strictly dominated by mixed strategy $\alpha_i \in \Delta(A_i)$:

$$u_i(\alpha_i, a_{-i}) > u_i(a_i, a_{-i}) \ \forall a_{-i} \in A_{-i} \tag{2.1.1.7}$$

$a^* \in A$ is a *dominant strategy equilibrium* if $\forall i \in N$: $u_i(a_i^*, a_{-i}) \geq u_i(a_i, a_{-i}), \forall a_i \in A_i, \forall a_{-i} \in A_{-i}$.

Example 2.1.1.1

Vickery auction (second price auction)

Solution:

In the second price auction, the dominant strategy equilibrium for each player is to pose the actual value of the bid.

Iterated elimination of strictly dominated strategies:

- **Step 0:** Define, for each i, $S_i^0 = S_i$. (S refer to the set of strategies)
- **Step 1:** Define, for each i,

$$S_i^1 = \left\{ s_i \in S_i^0 \mid \nexists s_i' \in S_i^0 \text{ s.t. } u_i(s_i', s_{-i}) > u_i(s_i, s_{-i}) \ \forall \ s_{-i} \in S_{-i}^0 \right\}.$$

...

- **Step k:** Define, for each i,

$$S_i^k = \left\{ s_i \in S_i^{k-1} \mid \nexists s_i' \in S_i^{k-1} \text{ s.t. } u_i(s_i', s_{-i}) > u_i(s_i, s_{-i}) \ \forall \ s_{-i} \in S_{-i}^{k-1} \right\}.$$

- **Step ∞:** Define, for each i,

$$S_i^\infty = \cap_{k=0}^\infty S_i^k.$$

Figure 2.1: Pure Strategy

- Let $S_i^0 = S_i$ and $\Sigma_i^0 = \Sigma_i$.
- For each player $i \in \mathcal{I}$ and for each $n \geq 1$, we define S_i^n as

$$S_i^n = \{ s_i \in S_i^{n-1} \mid \nexists \quad \sigma_i \in \Sigma_i^{n-1} \text{ such that } u_i(\sigma_i, s_{-i}) > u_i(s_i, s_{-i}) \text{ for all } s_{-i} \in S_{-i}^{n-1} \}.$$

- Independently mix over S_i^n to get Σ_i^n.
- Let $D_i^\infty = \cap_{n=1}^\infty S_i^n$.

Figure 2.2: Mixed Strategy

D_i^∞ is the set of strategies that survive iterated strict dominance.

Definition 2.1.1.4: Rationalizability

First, we need to define the **belief**: a belief of player i about the other players' actions is a probability measure over S_{-i}, denoting as $\Delta(S_{-i})$. We allow correlation in our belief: $\Delta(S)$ denotes a probability distribution over S.

For a game G, $a_i \in A$ is rationalizable if $\exists Z_j \subset A_j, j \in N$ s.t.:

1. $a_i \in Z_i$;
2. each $a_j \in Z_j$ is a BR to some belief $\mu_j \in \Delta(Z_{-j})$.

Another definition of rationalizability is given: A strategy is said to be rationalizable if and only if it survives the iterated elimination of the never-best response.

Definition 2.1.1.5: Never Best Responses (NBR)

A pure strategy s_i is a never-best response if for all beliefs σ_{-i} there exists $\sigma_i \in \sum_i$ s.t.:

$$u_i(\sigma_i, \sigma_{-i}) > u_i(s_i, \sigma_{-i}) \tag{2.1.1.8}$$

Remark 2.1.1

- A strictly dominated strategy is a never-best response;
- The reverse does not hold (hold in correlated beliefs).

Definition 2.1.1.6: Correlated Equilibrium

A correlated equilibrium is a joint distribution $\pi in \Delta(S)$ such that R is a R.V. distributed according to π:

$$\sum_{s_{-i} \in S_{-i}} \text{Prob}(R = s | R_i = s_i) \left[u_i(s_i, s_{-i}) - u_i(t_i, s_{-i})\right] \geq 0 \tag{2.1.1.9}$$

for all $t_i \in S_i$.

Definition 2.1.1.7: Nash Equilibrium

$a^* = (a_1^*, ..., a_n^*) \in A$ is an NE if $a_i^* \in B_i(a_{-i}^*)\ \forall i \in \mathbb{N}$. An NE can be either pure-strategy or mixed-strategy, and every NE is also a correlated equilibrium.

Remark 2.1.2

The relationship is given as:

$$NE_i \subseteq R_i^\infty \subseteq D_i^\infty \tag{2.1.1.10}$$

where NE_i is the Nash Equilibrium, R_i^∞ is the set of rationalizable strategies, and D_i^∞ is the set of strategies that survive iterated strict dominance.

Not all games have an NE (usually due to the discontinuity in u_i).

Theorem 2.1.1.3: Existence of NE (Nash 1951)

Any game with a finite set of players and a finite set of strategies has an NE of mixed strategies (this

theorem emphasizes on *finite games*).

Lemma 2.1.1.2: Kakutani Fixed Point

Let $f : A \rightrightarrows A$ be a correspondence, if:

- A is a compact and convex set;
- $f(x)$ is non-empty for all $x \in A$;
- for all $x \in A$, $f(x)$ is a convex set;
- $f(x)$ has a closed graph: if $\{x^n, y^n\} \to \{x, y\}$ with $y^n \in f(x^n)$, then $y \in f(x)$

Then f has a fixed point: $\exists x \in A$, s.t. $x \in f(x)$.

Proof 2.1.3

First, we show that the action set $\Sigma = \prod_{i \in I} \Sigma_i$ is compact, convex, and non-empty. This conclusion is quite natural, since each $\Sigma_i = \{x \mid \sum_j x_j = 1\}$ is a simplex, thus compact, convex, and non-empty, so is the product. Then we note the best correspondence $B(\sigma)$ is not empty because:

$$B_i(\sigma_{-i}) = \arg\max_{x \in \Sigma_i} u_i(x, \sigma_{-i})$$

is non-empty by Weirstrass's theorem.

Next, we show that $B(\sigma)$ is a convex set for all σ. When $\|B(\sigma)\| = 1$, it's a convex set by nature. When $B(\sigma)$ is not a single element, we take σ_i', σ_i'' to denote any two elements among $B(\sigma_{-i})$:

$$\begin{aligned} u_i(\sigma_i', \sigma_{-i}) &\geq u_i(\tau_i, \sigma_{-i}) \quad \text{for all } \tau_i \in \Sigma_i, \\ u_i(\sigma_i'', \sigma_{-i}) &\geq u_i(\tau_i, \sigma_{-i}) \quad \text{for all } \tau_i \in \Sigma_i. \end{aligned} \tag{2.1.1.11}$$

We can show that for all $\lambda \in (0, 1)$:

$$\lambda u_i(\sigma_i', \sigma_{-i}) + (1-\lambda) u_i(\sigma_i'', \sigma_{-i}) \geq u_i(\tau_i, \sigma_{-i}) \quad \text{for all } \tau_i \in \Sigma_i. \tag{2.1.1.12}$$

By the linearity of u_i:

$$u_i(\lambda \sigma_i' + (1-\lambda)\sigma_i'', \sigma_{-i}) \geq u_i(\tau_i, \sigma_{-i}) \quad \text{for all } \tau_i \in \Sigma_i. \tag{2.1.1.13}$$

Therefore, $\lambda \sigma_i' + (1-\lambda)\sigma_i'' \in B_i(\sigma_{-i})$, indicating that $B(\sigma)$ is a convex correspondence.

At last, we prove that $B(\sigma)$ is a closed graph by contradiction. Suppose not, then exists a sequence $(\sigma^n, \hat{\sigma}^n) \to (\sigma, \hat{\sigma})$ with $\hat{\sigma}^n \in \mathcal{B}(\sigma^n)$, but $\hat{\sigma}_i \notin B_i(\sigma_{-i})$. This implies for any $\epsilon > 0$, there always exists some σ_i' such that:

$$u_i(\sigma_i', \sigma_{-i}) > u_i(\hat{\sigma}_i, \sigma_{-i}) + 3\epsilon.$$

But by the fact that $\sigma_{-i}^n \to \sigma_{-i}$, we can find n such that:

$$u_i(\sigma_i', \sigma_{-i}^n) \geq u_i(\sigma_i', \sigma_{-i}) - \epsilon.$$

This leads us to:

$$u_i(\sigma_i', \sigma_{-i}^n) > u_i(\hat{\sigma}_i, \sigma_{-i}) + 2\epsilon \geq u_i(\hat{\sigma}_i^n, \sigma_{-i}^n) + \epsilon. \tag{2.1.1.14}$$

By the Kakutani Fixed Point theorem, for any σ, $B(\sigma)$ is not empty.

Q.E.D.

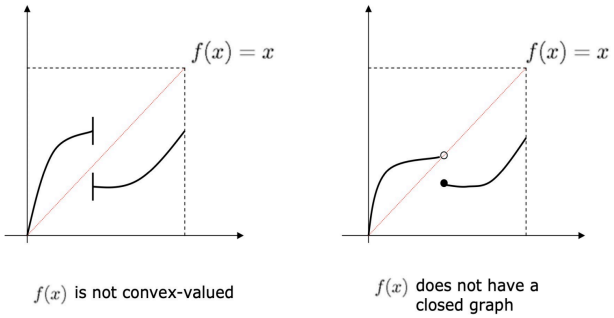


Figure 2.3: Examples of no fixed point

Example 2.1.1.2

A game with no NE but has correlated equilibria, consider the game with 3 players:

Player	Action Space	Action Notation
1	$[0,1] \times \mathbb{N}$	(x, m)
2	$[0,1] \times \mathbb{N}$	(y, n)
3	$[0,1]$	x'

The payoff function for each player is given as:

1. $-2 \cdot \mathbf{1}(x = x') + 2 \cdot \mathbf{1}(y = x \neq x') + \mathbf{1}(y \neq x \neq x') \cdot [\mathbf{1}(m > n) + \mathbf{1}(m < n)]$;
2. $-2 \cdot \mathbf{1}(x = x') + 2 \cdot \mathbf{1}(y = x \neq x') + \mathbf{1}(y \neq x \neq x') \cdot [\mathbf{1}(n > m) + \mathbf{1}(n < m)]$;
3. $\mathbf{1}(x = x')$

Solution:

The game has no NE: To avoid the circumstance of $\mathbf{1}(x = x') = 1$, player 1 has to decide x stochasticity on $[0,1]$, thus $\mathbf{1}(y \neq x \neq x') = 1$ almost surely. This would lead the game to a zero-sum game between player 1 and player 2, where A_i is not compact for each player (can not apply the Minimax Theorem). This dual-player game has no NE.

But this game has correlated equilibriums, under which players 1 and 2 make decisions x and y based on a joint distribution $\{(x, y) \| x = y\}$ (only with a mass point on the diagonal $x = y$). From Player 3's perspective, the marginal distribution of Player 1 is still stochastic. This can make $\mathbf{1}(y = x \neq x') = 1$ almost surely, and this is a set of correlated equilibriums.

Theorem 2.1.1.4: Minimax Theorem

Let $G = (\{1,2\},(A_i),(u_i))$ be a **zero-sum** game where A_i is compact and u_i is continuous. Then a^* is NE iff:

1. a_i^* is a maxminimizer: $a_i^* = \arg\max_{a_i} \min_{a_{-i}} u_i(a_i, a_{-i})$;
2. $u_i(a^*) = \max_{a_i} \min_{a_{-i}} u_i(a) = \min_{a_{-i}} \max_{a_i} u_i(a)$.

Theorem 2.1.1.5: Glicksberg

Definition 2.1.1.8: Continuous games

A game $\langle \mathcal{I},(S_i),(u_i)\rangle$ where $\mathcal{I}$ is a finite set, the S_i are nonempty compact spaces, and u_i are continuous functions is called a continuous game.

Any continuous game has a mixed strategy NE.

Proof 2.1.4

1. First, we approximate the original game with a sequence of finite games;
2. Then show these finite games have equilibriums using Nash's theorem;
3. At last, using the continuity and closed assumption to show these converge to the original game.

Q.E.D.

Nash Equilibrium sometimes can bring unreasonable behavior. In this game, both (A,A) and (B,B) are NEs, but (A,A) is surely more reasonable.

		2	
		A	B
1	A	1,1	0,0
	B	0,0	0,0

Figure 2.4: Example of unreasonable NE

Definition 2.1.1.9: Trembling Hand Perfect Equilibrium

A strategy σ is a trembling hand perfect equilibrium if there exists a sequence of **totally mixed** ($\sigma_i(s_i) > 0$ for all $s_i \in S$) strategy σ^n converging to σ, such that $\sigma_i \in BR_i(\sigma_{-i}^n)$ for all n,i.
Using this definition, we can observe that (B,B) is not a trembling hand perfect equilibrium because it's not the best response for any totally mixed strategy.

Remark 2.1.3

- Every trembling hand perfect equilibrium is an NE;
- For every *finite game*, there exists a trembling hand perfect equilibrium.

Dynamic Game

Consider a sequential game, denoting $h^0 = \emptyset$ as the initial **history**, $a^k = (a_1^k, \cdots, a_n^k)$ as the actions at stage k, and $h^{k+1} = (a^0, a^1, \ldots, a^k)$ as the hostory at stage $k+1$. Let $H = \cup_{k=0}^{K+1} H^k$ be the set of all possible histories. Under this definition, the pure strategy in the sequential games is a plan for every possible history h_k: $S_i(H^k) = \bigcup_{h^k \in H^k} S_i(h^k)$.

Definition 2.1.1.10: Information Set

The information set is a subset of the history, partitioning the nodes of the game tree. A player who is choosing an action at x is uncertain if he is at x or at $x' \in h(x)$.

Remark 2.1.4

- When $x' \in h(x)$, $A(x) = A(x')$;
- A game has perfect information if all its information sets are singletons.

Definition 2.1.1.11: Sub-Game & SPNE

A subgame G' of an extensive form game G consists of a single node and all its successors in G, with the property that if $x' \in V_{G'}, x'' \in h(x')$, then $x'' \in V_{G'}$.

A **Subgame Perfect (Nash) Equilibrium** is a strategy s^* that for any $G' \in G$, $S^*|G'$ is an NE for G'. The SPNE can remove noncredible threats. Every finite extensive form game G has an SPNE, and every finite perfect information extensive form game has a pure strategy SPNE.

Remark 2.1.5

How to find SPNE:

- In finite-horizon games: backward induction;
- For infinite-horizon games (with continuity at infinity): one-stage deviation principle.

Incomplete Information

First, we define **state**, each state $w \in \Omega$. States are exhaustive and mutually exclusive. Information function $P_i(w)$ is the set of states that i thinks he may be in, given that the true state is w (analogous to 'information set').

Assumption 2.1.1.1: Information Function

1. *Reflexitivity or Non-Deludedness*: $\omega \in P_i(\omega)$;
2. Consistency: $\forall \omega, \omega' \in \Omega, \omega' \in P_i(\omega) \Rightarrow P_i(\omega') = P_i(\omega)$.

Definition 2.1.1.12: Knowledge Operator

We say that i knows event $A \subset \Omega$ at ω if $P_i(\omega) \subset A$. Thus, the event that i knows A is:

$$K_i(A) := \{\omega \in \Omega : P_i(\omega) \subset A\} \tag{2.1.1.15}$$

Example 2.1.1.3

Consider the game between A and B; A may feel good or sick, and B can see whether he has received a message saying, 'A is sick'. $\Omega = \{a, b, c\}$:

1. a: A is good;
2. b: A is sick, B receive the message;
3. c: A is sick, B receive no message;

We can conclude that $P_A = \{\{a\}, \{b, c\}\}, P_B = \{\{a, b\}, \{c\}\}$.

Solution:

- *B know that A is sick:* $K_B(\{b, c\}) = \{\omega : P_B(\omega) \subset \{b, c\}\} = \{c\}$;
- *A doesn't know that B know A is sick:* $K_A(K_B(\{b, c\})) = \{\omega : P_A(\omega) \subset \{c\}\} = \varnothing$.

Properties of the knowledge operator:

1. $K_i(\Omega) = \Omega$;
2. $E \subset F \Rightarrow K_i(E) \subset K_i(F)$;
3. $K_i(A) \cap K_i(B) = K_i(A \cap B)$;
4. $K_i(A) \subset A$;
5. $K_i(A) = K_i(K_i(A))$
6. $\Omega \setminus K_i(A) = K_i(\Omega \setminus K_i(A))$

Remark 2.1.6

An event $A \subset \Omega$ is **self-evident** if $P_i(\omega) \subset A$ for any $\omega \in A, i$. An event $E \subset \Omega$ is **common knowledge** at state ω if exists a self-evident A s.t. $w \in A \subset E$ for all players (or $P_i(\omega) \subset E$ for all i).

In games with incomplete information, one agent is unsure about the payoffs of others. One solution is to suppose that players have **common** prior to the distribution of possible games, this can create common knowledge by making the original game into a 'bigger' one (incomplete information to imperfect information).

Definition 2.1.1.13: Bayesian Games

Compared to normal games, the Bayesian games have two more elements:

1. A set of types for each player i: $\theta_i \in \Theta_i$, different type can bring different payoff functions;

2. A (joint) distribution $p(\theta_1, \ldots, \theta_1)$ over the types.

Under this definition, a strategy for player i is a map $s_i : \Theta_i \to S_i$.
😎 Why it's called *Bayesian game*? Given $p(\theta_1, \ldots, \theta_n)$, each player can calculate $p(\theta_{-i} \mid \theta_i)$ using Bayes rule. Thus, the expected utility for player i is given as:

$$U(s_i', s_{-i}, \theta_i) = \sum_{\theta_{-i}} p(\theta_{-i} \mid \theta_i) u_i(s_i', s_{-i}(\theta_{-i}); \theta_i, \theta_{-i}) \tag{2.1.1.16}$$

Theorem 2.1.1.6: Bayesian Nash Equilibrium (BNE)
A strategy profile $s(\cdot)$ is a BNE if for all i, θ_i, we have that:

$$s_i(\theta_i) \in \arg\max_{s_i' \in \mathcal{S}_i} \sum_{\theta_{-i}} p(\theta_{-i} \mid \theta_i) u_i(s_i', s_{-i}(\theta_{-i}); \theta_i, \theta_{-i}), \tag{2.1.1.17}$$

or in the infinite-type case:

$$s_i(\theta_i) \in \arg\max_{s_i' \in S_i} \int u_i(s_i', s_{-i}(\theta_{-i}); \theta_i, \theta_{-i}) dP(\theta_{-i} \mid \theta_i). \tag{2.1.1.18}$$

🌳 For any finite Bayesian game, there exists a mixed-strategy BNE.

Dynamic Games of Incomplete Information

In this game, there is an analog of BNE, called perfect Bayesian equilibrium (PBE), which strengthens subgame perfection by:

1. A complete strategy (a mapping from information set to strategies);
2. Beliefs for each player ($P_i(v|h)$ (a conditional distribution over everything player i **does not know**, given everything that player i **does know**).

In a dynamic game of complete and perfect information, PBE is equivalent to SPNE because all information sets are singletons, so beliefs are trivial.

Example 2.1.1.4

Ante Game:
there are two players. Player i can observe $t_i \sim uniform(0, 1)$ independently. Player 1 can 'show-down'(both players show t_i, the higher i wins), or he can 'raise': under this case, player 2 can 'fold' (lose the game) or 'match' (force a showdown).
In this game, the information sets for player 1 is given as (t_1), and (t_2, a_1) for player 2. The beliefs for player 1 is $p_1(t_2|t_1) = t_2$ and $p_2(t_1|t_2, a_1) = p_2(t_1|a_1)$ for player 2.

Solution:

Find the PBE for the game.

The signaling game

Player 1 has some private information t_1 and first move A_1, player 2 sees A_1 and make a move A_2. In this case, when player 2 moves first, it's called a **screening game**. An example is the job market game ([Spence, 1973]):

player 1 is the worker, who has the intrinsic ability t_1 and education decision a_1. Player 2 is the firm that can decide the wage offered a_2.

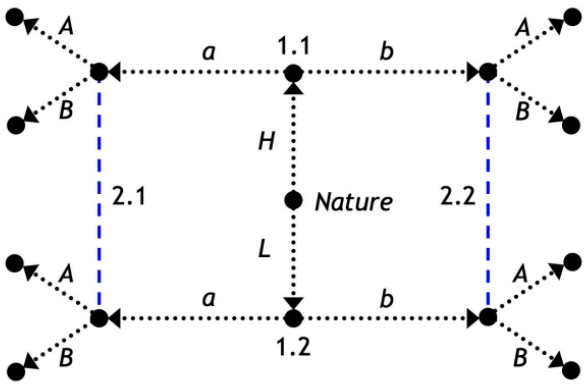


Figure 2.5: The stylised signaling game

Consider a stylized signaling game: a seller has an item with quality of either H (with probability p) or L (probability $1-p$), and the seller can advertise either H or L. In this case, the game only has a **pooling** equilibrium (the player 2's belief is independent of the seller's action). But suppose there is a cost c ($H - c < L$) when the seller lies; this game has an **separating** equilibrium: the seller always tells the truth and the player 2 always believes the advertisement.

2.1.2 Repeated Games

In **repeated games**, we refer to a situation in which the same **stage game** is played for T times with a discount factor $\delta \in [0, r)$. We assume *perfect monitoring*: the outcomes of all past periods are observed by all players. Notation:

- $\mathbf{a} = \{a^t\}_{t=0}^T$the sequence of actions;
- $\alpha = \{\alpha^t\}_{t=0}^T$: the sequence of mixed strategies;
- $u_i(\mathbf{a}) = \sum_{t=0}^T \delta^t g_i(a_i^t, a_{-i}^t)$: payoff.

In any finitely repeated prisoner's dilemma game, the unique SPE is always to defect. If the game has multiple NEs, the SPE is to select one NE in each period. In infinitely repeated games, we can consider trigger strategies: both cooperate if there is no defect in the past; otherwise, always a defect. This is a SPE when δ is large enough.

	A	B	C
A	1,1	-1,2	-2,-2
B	2,-1	0,0	-2,-2
C	-2,-2	-2,-2	-1,-1

Figure 2.6: Sustaining Cooperation

In the examples above, the game has two NEs. Now consider the following strategies:

1. In period 1, play (A, A);
2. In period $2, \cdots, t^*$, $t^* \in (1, T)$: if (A,A), then continue; otherwise (C,C);
3. In period $t^* + 1, \cdots, T$: if all (A, A) in stage 2 and (B, B) after stage 2: (B, B); otherwise (C, C).

By **one-shot deviation principle**, this is an SPE for appropriate t^*. For the largest available t^*:

$$1 \leq \sum_{n}^{T-t^*} \delta^n \tag{2.1.2.1}$$

When both T and δ are sufficiency large, we can make the actions (A, A) be played for arbitrarily many periods (infinite game).

Theorem 2.1.2.1: Nash Folk Theorem

If $(v_1, \cdots, v_n)$ is feasible and higher than the minmax payoff lower bounds $\underline{v}_i$ for all i, then there exists $1 > \delta \geq \underline{\delta}$, that for all game $G^\infty(\delta)$, there is a equilibrium with payoffs $(v_1, \cdots, v_n)$.

The Bargaining Model

[Rubinstein, 1982] studies another equilibrium in the sequential bargaining. In the model, two players are bargaining for a unit pie. Time runs from $t = 0, 1, 2, \cdots$, at time 0, player 1 propose a split $x_0, 1 - x_0$, if rejected by player 2, player 2 can propose $y_1, 1 - y_1$ at time 1. This game continues infinitely, assuming there are discounted factors δ_1, δ_2 for the two players.

Proposition 2.1.2.1: Finite-Horizon Bargaining Game

In the finite-horizon bargaining game, we can use backward induction to solve the SPNE. When $T = 2$, the SPNE is player 1 offers $1 - \delta_2, \delta_2$ at time 1 and player 2 accepts it.

Rubinstein considers the infinite case.

Theorem 2.1.2.2: [Rubinstein, 1982]

There is a **unique** SPE in the infinite bargaining game: player 1 always proposes a split $(x, 1 - x)$ with $x = (1 - \delta_2)/(1 - \delta_1\delta_2)$, player 2 always accepts when her split $\geq 1 - x$; player 2 always proposes a split

$(y, 1-y)$ with $y = \delta_1(1-\delta_2)/(1-\delta_1\delta_2)$, player 1 always accepts when her split $\geq y$.

Proof 2.1.5

It's easy to show that this is an SPE. To show the uniqueness of the SPE, let $\underline{v}_1$ and $\bar{v}_1$ denote the lowest and highest payoffs that player 1 could get in any SPE starting at a date where he gets to make an offer. Consider a date when player 2 makes an offer he can make (the proposal accepted by player 1) at most $1-\delta_1\underline{v}_1$ by a split $\delta_1\underline{v}_1, 1-\delta_1\underline{v}_1$, and at least $1-\delta_1\bar{v}_1$ by a split $\delta_1\bar{v}_1, 1-\delta_1\bar{v}_1$. Now reconsider the decision of the player 1, we can obtain:

$$\bar{v}_1 \leq 1-\delta_2(1-\delta_1\bar{v}_1), \tag{2.1.2.2}$$

$$\underline{v}_1 \geq 1-\delta_2(1-\delta_1\underline{v}_1), \tag{2.1.2.3}$$

which leads to:

$$\underline{v}_1 \geq \frac{1-\delta_2}{1-\delta_1\delta_2} \geq \bar{v}_1 \tag{2.1.2.4}$$

Q.E.D.

2.1.3 Cooperative Game

A **cooperative game** (coalitional game) is a model of interacting decision-makers that focuses on the behavior of groups of players (N). A **coalition** is a subset $S \subset N$; we call N the **grand coalition**. Every coalition S has a set of available actions A_S. An **outcome** consists of a CS (**coalition structure**) and one action associated with the CS:

$$(S_k, a_k)_{k=1,\dots,\bar{k}} \text{with} S_j \cap S_k = \emptyset, \forall j \neq k, \cup_k S_k = N, a_k \in A_{S_k}.$$

Each agent has a utility function u_i without externalities.

Definition 2.1.3.1: Transferable Payoffs

A game with transferable payoffs associates any coalition S to a real number $v(S)$, and assumes $v(\emptyset) = 0$. The set of actions available for coalition S is all possible divisions $(x_i)_{i\in S}$ such that $\sum_{i\in S} x_i = v(S)$.

A coalition S can **block** an action a_N of the grand coalition if $v(S) > \sum_{i\in S} x_i, x_i = a_{N_i}$. We call a game **cohesive** if there exists some action a_N in the grand coalition that no coalition can block.

Definition 2.1.3.2: Core

The core of a coalitional game is the set of actions a_N of the grand coalition N that are not blocked by any coalition.

Theorem 2.1.3.1: Bondareva $ Shapley

A non-negative vector λ_S with dimension 2^N is a **balanced** weight if:

$$\sum_{\{S\subset N | i\in S\}} \lambda_S = 1, \forall i \in N.$$

A payoff function is **balanced** if for every balanced weight λ:

$$\sum_{S \subset N} \lambda_S v(S) \leq v(N) \tag{2.1.3.1}$$

There is an intuitive way to understand the balanced weight, $_S$ can be viewed as the proportion of time for player i distributing on the coalition S.

A coalitional game with transferable payoffs has a non-empty core iff it is balanced.

Proof 2.1.6

The proof needs the duality theorem. The core can be solved by solving the LP:

$$\min \sum_{i \in N} x_i \text{ s.t. } \sum_{i \in S} x_i \geq v(S), \forall S \subset N. \tag{2.1.3.2}$$

The core is non-empty iff $\sum_{i \in N} x_i^* \leq v(N)$, and the dual of the LP is given by:

$$\max \sum_{S \in 2^N} \lambda_S v(S) \text{ s.t. } \lambda_S \geq 0, \forall S \subset N \& \sum_{S \ni i} \lambda_S \leq 1, \forall i \in N. \tag{2.1.3.3}$$

By the definition of LP, $\sum_{i \in N} x_i^* = \sum_{S \in 2^N} \lambda_S^* v(S)$. So if $LHS \leq v(N)$, then $RHS \leq v(N)$.

Q.E.D.

Definition 2.1.3.3: Convex Cooperative Game

A game with transferable playoffs v is *convex* (sometimes referred to as super-modularity) if for any two collations S and T:

$$v(S \cup T) + v(S \cap T) \geq v(S) + v(T), \tag{2.1.3.4}$$

which implies that the marginal contribution of an individual i to a coalition is (weekly) increasing as the coalition gets larger because for any $S \subset T$ and $i \notin T$:

$$S \subset T \ \& \ i \notin T \implies v(T \cup \{i\}) - v(T) \geq v(S \cup \{i\}) - v(S).$$

To see this, we consider the interaction between $S \cup \{i\}$ and T and apply the above definition:

$$v((S \cup \{i\}) \cup T) + v((S \cup \{i\}) \cap T) \geq v(S \cup \{i\}) + v(T). \tag{2.1.3.5}$$

Remark 2.1.7

There are some similar definitions in the game:

1. **Superaddictivity**: for any two disjoint S, T: $v(S \cup T) \geq v(S) + v(T)$.
2. **Monotone**: for any coalitions $S \subset T$: $v(S) \leq v(T)$.

Convexity implies superadditivity, and superadditivity implies monotonicity.

Proposition 2.1.3.1

Any convex game with transferable payoffs has a non-empty core.

The Core and Competitive Equilibria

Consider an exchange economy with a set of consumers N and a set of goods G. A *consumption bundle* is an element $x \in \mathbb{R}_+^G$. Consumer i enjoys utility $u_i(x_i)$ from a bundle x_i. Each consumer starts with an endowment $\omega_i \in \mathbb{R}_+^G$. An *allocation* $\{x_i\}_{i \in N}$ with $x \in \mathbb{R}_+$ is feasible if

$$\sum_{i \in N} x_i = \sum_{i \in N} \omega_i$$

Definition 2.1.3.4: Competitive Equilibria

A competitive equilibria is a price vector $(p_g^*)_{g \in G}$ and a feasible allocation $x^* = (x_i^*)_{i \in N}$ such that:

$$p^* \cdot x_i \leq p^* \cdot \omega_i \implies u_i(x_i^*) \geq u_i(x_i). \tag{2.1.3.6}$$

Intuitive understanding: x_i^* is the consumption bundle that maximizes consumer i's utility among bundles he can afford given the price p^*.

We can view the market as a cooperative game (without considering prices), and for any coalition S in the game, the distribution needs to satisfy:

$$A_S = \left\{ (x_i)_{i \in S} \mid \sum_{i \in S} x_i = \sum_{i \in S} \omega_i \right\}. \tag{2.1.3.7}$$

Note that in this game there is no transferable utility function.

Theorem 2.1.3.2

Any competitive equilibrium is in the core.

Proof 2.1.7

Let x^* be a competitive equilibrium allocation corresponding to a price vector p. Suppose that a coalition S can block x^*, then there exists $(x_i)_{i \in S}$ such that $\sum_{i \in S} x_i = \sum_{i \in S} \omega_i$ and $u_i(x_i) > u_i(x_i^*) \forall i \in S$. By definition of equilibrium, we can imply:

$$p \cdot x_i > p \cdot \omega_i, \forall i \in S$$

Adding up these conditions, we obtain

$$p \cdot \sum_{i \in S} x_i > p \cdot \sum_{i \in S} \omega_i. \tag{2.1.3.8}$$

Contradiction.

Q.E.D.

Shapley Value

An outcome in the core may be unfair, a fair payment scheme rewards each agent according to his marginal contribution. However, this scheme is dependent on the order, we can permutate on all the orderings, which is the Shapley value.

Definition 2.1.3.5: Shapley Value

The Shapley value is given by:

$$\varphi_i(v) = \sum_{S \subset N \setminus \{i\}} \frac{|S|!(|N| - |S| - 1)!}{|N|!}(v(S \cup \{i\}) - v(S)). \tag{2.1.3.9}$$

Intuitive understanding: $\varphi_i(v)$ represents the expected value of player i's contribution to the coalition formed by the players preceding him in line.

Axiom 2.1.3.1

- **Symmetry**: if i and j are interchangable ($v(S \cup \{i\}) = v(S \cup \{j\})$ for all S disjoint from $\{i, j\}$) in v, then $\varphi_i(v) = \varphi_j(v)$;
- **Dummy**: if i is dummy ($v(S \cup \{i\}) = v(S)$ for all S) in v, then $\varphi_i(v) = 0$;
- **Addictivity**: For any two games v and w, we have $\varphi(v + w) = \varphi(v) + \varphi(w)$.

Theorem 2.1.3.3

A value satisfies the three axioms above iff it is the Shapley value.

Proof 2.1.8

First, we prove the 'if' part. The only not-so-straightforward part is the proof of symmetry:

$$\begin{aligned}
\varphi_i(v) &= \sum_{S \subset N \setminus \{i\}} \frac{|S|!(|N| - |S| - 1)!}{|N|!}(v(S \cup \{i\}) - v(S)) \\
&= \sum_{S \subset N \setminus \{i,j\}} \frac{|S|!(|N| - |S| - 1)!}{|N|!}(v(S \cup \{i\}) - v(S)) \\
&+ \sum_{S \subset N \setminus \{i,j\}} \frac{(|S| + 1)!(|N| - (|S| + 1) - 1)!}{|N|!}(v(S \cup \{i, j\}) - v(S \cup \{j\})) \\
&= \sum_{S \subset N \setminus \{i,j\}} \frac{|S|!(|N| - |S| - 1)!}{|N|!}(v(S \cup \{j\}) - v(S)) \\
&+ \sum_{S \subset N \setminus \{i,j\}} \frac{(|S| + 1)!(|N| - (|S| + 1) - 1)!}{|N|!}(v(S \cup \{i, j\}) - v(S \cup \{i\})) \\
&= \varphi_j(v).
\end{aligned} \tag{2.1.3.10}$$

Next, we need to prove the 'only if' part, suppose ψ is the value that satisfies the above axioms, we need to prove $\psi = \varphi$.

Q.E.D.

2.1.4 Mechanism Design

This subsection is mainly the notes from [Jackson, 2014]. Mechanism design is a reverse engineering of game theory, it focuses on the problems associated with incentives and private information (selling, matching and auctions). A general setting for mechanism design:

1. individuals: a finite group of individuals denoted by $N = \{1, 2, \cdots, n\}$;
2. decisions: a specific decision d from the set of potential decisions D;
3. information: each individual has some private information $\theta_i \in \Theta_i$, let $\Theta = \times_i \Theta_i$;
4. preference: each individual has a value function $v_i : D \times \Theta_i \to \mathbb{R}$, we assume the utility function only depends on θ and d.

Definition 2.1.4.1: Efficient Decisions

A decision rule is a mapping $d : \Theta \to D$, it is efficient if:

$$\sum_i v_i(d(\theta), \theta_i) \geq \sum_i v_i(d', \theta_i) \tag{2.1.4.1}$$

for all θ_i and $d' \in D$. Intuition: given information θ, the decision that can generate the maximum total utility is the efficient decision.

To make the efficient decision happen, we can define a **transfer function** $t : \Theta \to \mathbb{R}^n$, and the function $t_i(\theta)$ represents the payment that i receives (Here θ is the 'announced' types, sometimes is denoted as $\hat{\theta}$). A transfer function is feasible if $0 \geq \sum_i t_i(\theta)$ for all θ, and balance if $\sum_i t_i(\theta) = 0$ for all θ.

Definition 2.1.4.2: Social Choice Functions & Mechanism

A social choice function f is a pair d, t, given $\hat{\theta}, \theta_i$, the final utility function for player i is:

$$u_i(\widehat{\theta}, \theta_i, d, t) = v_i(d(\widehat{\theta}), \theta_i) + t_i(\widehat{\theta}). \tag{2.1.4.2}$$

A **mechanism** is a pair M, g, where $M = M_1 \times \cdots \times M_n$ is the space of message ('announced' information) for all individuals, and $g : M \to D \times^n$ is an outcome function.

Remark 2.1.8

- A *social choice* function contains a mapping from decisions to the outcome: d, v, thus imply u;
- Note that a social choice function $f = (d, t)$ can be viewed as a mechanism, where $M_i = \Theta_i$ and $g = f$. This is referred to as a *direct mechanism*: one in which the space of possible actions is equal to the space of possible types. All other mechanisms are called *indirect*.

Dominant Strategy Mechanism Design

A strategy $m_i \in M_i$ is a *dominant strategy* for player i with $\theta_i \in \Theta_i$ if:

$$v_i(g_d(m_{-i}, m_i), \theta_i) + g_{t,i}(m_{-i}, m_i) \geq v_i(g_d(m_{-i}, \widehat{m_i}), \theta_i) + g_{t,i}(m_{-i}, \widehat{m_i}) \tag{2.1.4.3}$$

for all $m_{-i} \in M_{-i}$ and $\widehat{m_i} \in M_i$. This is a strong situation that the strategy is optimal no matter what the other players do.

Definition 2.1.4.3: Incentive Compatibility

A (direct) mechanism is BIC (**bayesian incentive compatibility**) iff truthtelling is a BayesNash equilibrium: i.e.,

$$\mathop{\mathrm{E}}_{\mathbf{t}\sim F}\left[u_i(g(t_i,\mathbf{t}_{-i});\mathbf{t})\right] \geq \mathop{\mathrm{E}}_{\mathbf{t}\sim F}\left[u_i(g(t_i',\mathbf{t}_{-i});\mathbf{t})\right], \quad \forall i \in [n], \forall t_i' \in T_i. \tag{2.1.4.4}$$

Similarly, a mechanism is EPIC (**ex-post incentive compatibility**) iff truthtelling is a Nash equilibrium: i.e.,

$$u_i(g(t_i,\mathbf{t}_{-i});\mathbf{t}) \geq u_i(g(t_i',\mathbf{t}_{-i});\mathbf{t}), \quad \forall i \in [n], \forall t_i' \in T_i. \tag{2.1.4.5}$$

A mechanism is DSIC (**dominant strategy incentive compatibility**) iff:

$$u_i(g(t_i,\mathbf{t}_{-i});\mathbf{t}) \geq u_i(g(t_i',\mathbf{t}_{-i});\mathbf{t}), \quad \forall i \in [n], \forall t_i' \in T_i, \forall \mathbf{t}_{-i} \in T_{-i}. \tag{2.1.4.6}$$

Remark 2.1.9

EPIC is equivalent to DSIC in a direct mechanism.

Theorem 2.1.4.1: The Revelation Principle for Dominant Strategies

If a mechanism (M, g) implements a social choice function $f = (d, t)$ in dominant strategies (i.e., there exist functions $m_i : \Theta_i \to M_i$ such that $m_i(\theta_i)$ is dominant strategy for all i and $g(m(\theta)) = f(\theta)$ for all θ), then the direct mechanism f is dominant strategy incentive compatible (sometimes also refers to **strategy-proof**).

Considering cases that have no transfer function t, a decision d is DSIC if the social choice function $f = (d, t^0)$ is DSIC. A decision rule d is **dictatorial** if there exists i such that $d(\theta) \in \arg\max_{d\in D} v_i(d, \theta_i)$ for all θ.

Theorem 2.1.4.2: The Gibbard-Satterthwaite Theorem

Suppose that $|D| \geq 3$ and each individual can have a strict ranking of these decisions. Then a decision rule d with no transfer function is DSIC iff it is dictatorial. The proof can be found in [Benoit, 2000].

The Gibbard-Satterthwaite theorem has quite negative implications for the hopes of implementing non-trivial decision rules in dominant strategies in a general set of environments.

In most settings, the transfer functions are needed.

Grove's Schemes

1. Find a decision rule d that is efficient;
2. Find a transfer function t such that d, t is DSIC.

Theorem 2.1.4.3: Groves, Jerry Green and Jean-Jacques Laffont

If d is an efficient decision rule and for each i there exists a function $x_i : \times_{j \neq i} \Theta_j \to \mathbb{R}$ such that:

$$t_i(\theta) = x_i(\theta_{-i}) + \sum_{j \neq i} v_j(d(\theta), \theta_j), \tag{2.1.4.7}$$

Then (d, t) is DSIC. The key of the theorem is that $x_i(\theta_{-i})$ is independent of θ.

Proof 2.1.9: Groves

Let's prove this by contradiction. Suppose that d is efficient and x_i satisfies 2.1.4.7, but d, t is not DSIC. Then there exists $\widehat{\theta}_i, \theta_i$ for some i such that:

$$v_i(d(\theta_{-i}, \widehat{\theta}_i), \theta_i) + t_i(\theta_{-i}, \widehat{\theta}_i) > v_i(d(\theta), \theta_i) + t_i(\theta). \tag{2.1.4.8}$$

From 2.1.4.7 this implies that:

$$v_i(d(\theta_{-i}, \widehat{\theta}_i), \theta_i) + x_i(\theta_{-i}) + \sum_{j \neq i} v_j(d(\theta_{-i}, \widehat{\theta}_i), \theta_j) > v_i(d(\theta), \theta_i) + x_i(\theta_{-i}) + \sum_{j \neq i} v_j(d(\theta), \theta_j), \tag{2.1.4.9}$$

$$v_i(d(\theta_{-i}, \widehat{\theta}_i), \theta_i) + \sum_{j \neq i} v_j(d(\theta_{-i}, \widehat{\theta}_i), \theta_j) > v_i(d(\theta), \theta_i) + \sum_{j \neq i} v_j(d(\theta), \theta_j). \tag{2.1.4.10}$$

Which contradicts the efficiency of d.

Q.E.D.

Conversely, If d is an efficient decision rule and (d, t) is DSIC. Then there exists a function $x_i : \times_{j \neq i} \Theta_j \to \mathbb{R}$ satisfies 2.1.4.7.

Proof 2.1.10: Jerry Green and Jean-Jacques Laffont

Let d be an efficient rule and (d, t) be DSIC, note that there exists a function X_i such that:

$$t_i(\theta) = x_i(\theta) + \sum_{j \neq i} v_j(d(\theta), \theta_j) \tag{2.1.4.11}$$

We only need to show that x_i is independent of θ_i. Suppose there exists $i, \theta, \widehat{\theta}$ such that $x_i(\theta) > x_i(\theta_{-i}, \widehat{\theta}_i)$. Let $\epsilon = \frac{1}{2}[x_i(\theta) - x_i(\theta_{-i}, \widehat{\theta}_i)]$, by DSIC, it follows that $d(\theta) \neq d(\theta_{-i}, \widehat{\theta}_i)$ (when $\widehat{\theta}_i$, if d is the same, he would report as θ_i). So there exists $\widetilde{\theta}_i \in \Theta_i$ such that:

$$v_i(d(\theta_{-i}, \widehat{\theta}_i), \widetilde{\theta}_i) + \sum_{j \neq i} v_j(d(\theta_{-i}, \widehat{\theta}_i), \theta_j) = \varepsilon, \tag{2.1.4.12}$$

and for any $d \neq d(\theta_{-i}, \hat{\theta}_i)$:

$$v_i(d, \widetilde{\theta}_i) + \sum_{j \neq i} v_j(d, \theta_j) = 0 \tag{2.1.4.13}$$

The efficiency of d together with these conditions on $\tilde{\theta}_i$ imply that $d(\theta_{-i}, \widetilde{\theta}_i) = d(\theta_{-i}, \widehat{\theta}_i)$ and $t_i(\theta_{-i}, \widetilde{\theta}_i) = t_i(\theta_{-i}, \widehat{\theta}_i)$.

- The utility of i from truthful announcement at $\tilde{\theta}_i$ is $\varepsilon + x_i(\theta_{-i}, \widehat{\theta}_i)$;
- By lying θ_i at $\tilde{\theta}_i$, i get $x_i(\theta_i)$.

Since $x_i(\theta_i) > \varepsilon + x_i(\theta_{-i}, \widehat{\theta}_i)$m this contradicts the DSIC of d, t.

Q.E.D.

It's intuitive to come up with the **pivotal mechanism** from the Groves scheme: if i's presence is already maximizing the sum utility of other players: $\sum_{j \neq i} v_j(d(\theta), \theta_j)$, we transfer him zero. Otherwise he is a **povital** player, we let $x_i(\theta_{-i}) = -\max_{d \in D} \sum_{j \neq i} v_j(d, \theta_j)$ and give him a negative transfer:

$$t_i(\theta) = \sum_{j \neq i} v_j(d(\theta), \theta_j) - \max_{d \in D} \sum_{j \neq i} v_j(d, \theta_j). \tag{2.1.4.14}$$

This scheme is always feasible since $t_i(\theta) \leq 0$.

Remark 2.1.10

One example of the pivotal mechanism is the *Vickrey Auction* (second-price auction). The payment(transfer function) of the winner is $t_i(\theta) = -\max_{j \neq i} \theta_j$, which is the price of the second highest valuation. Such a mechanism is DSIC so every player reports θ_i truthfully.

DSIC, Groves Scheme and the Efficiency

Sometimes, when the space of the type is large, using the Grove scheme can result in an inefficient mechanism (due to the imbalance of the transfer function). Considering the public goods problem with two individuals and $\Theta_1 = \Theta_2 = \mathbb{R}$, $c = 1.5$. By the Jerry Green and Jean-Jacques Laffont theorem, there exists X_i that can make a mechanism DSIC. Applying the feasibility of the transfer function, we can obtain:

1. $\theta_1 = \theta_2 = 1$: $-0.5 \geq x_1(1) + x_2(1)$;

2. $\theta_1 = \theta_2 = 0$: $0 \geq x_1(0) + x_2(0)$;

3. So either $-\frac{1}{4} \geq x_1(1) + x_2(0)$ or $-\frac{1}{4} \geq x_1(0) + x_2(1)$;

4. Note $d(0,1) = d(1,0) = 0$, so either $-\frac{1}{4} \geq t_1(1,0) + t_2(1,0)$ or $-\frac{1}{4} \geq t_1(0,1) + t_2(0,1)$.

This means the DSIC mechanism can always provide an inefficient mechanism in some games. This problem can be solved if there is a θ_i which is known or fixed, or be approximated if the number of the society is large enough. Another problem with the DSIC mechanism is: it may violate the **individual ratinoality**:

$$v_i(d(\theta), \theta_i) + t_i(\theta) \geq 0 \tag{2.1.4.15}$$

Bayesian Mechanism Design

The balance difficulties exhibited by Groves' schemes could be overcome in a setting where individuals have probabilistic beliefs over the types of other individuals. Assume that Θ is a finite set and that θ is randomly chosen according to a distribution P. $\bar{\theta}_i$ denotes the random variable and θ_i' denotes realizations. A Bayesian equilibrium holds if:

$$\begin{aligned} &E\left[v_i(g_d(m_{-i}(\bar{\theta}_{-i}), m_i(\theta_i)), \theta_i) + g_{t,i}(m_{-i}(\bar{\theta}_{-i}), m_i(\theta_i)) \mid \theta_i\right] \\ &\geq E\left[v_i(g_d(m_{-i}(\overline{\theta}_{-i}), \widehat{m}_i), \theta_i) + g_{t,i}(m_{-i}(\overline{\theta}_{-i}), \widehat{m}_i) \mid \theta_i\right] \end{aligned} \tag{2.1.4.16}$$

for each $i, \theta_i, \widehat{m}_i$. A direct mechanism (d,t) is BIC if truthtelling is the Bayesian equilibrium.

Theorem 2.1.4.4: Revelation Principle for Bayesian Equilibrium

If (M,g) realizes a social choice function (d,t) in Bayesian equilibrium, then the direct mechanism f is Bayesian incentive compatible.

If the distribution of different θ_i is independent, we can construct a BIC mechanism that the transfer function is always balanced (efficient).

Theorem 2.1.4.5: d'Aspremont, Gerard-Varet, and Arrow

If $\bar{\theta_{-i}}$ and $\bar{\theta}_i$ are independent of each i and d is efficient, and

$$t_i(\theta) = E\left[\sum_{j \neq i} v_j(d(\bar{\theta}), \bar{\theta}_j) | \theta_i\right] - \frac{1}{n-1} \sum_{k \neq i} E\left[\sum_{j \neq k} v_j(d(\bar{\theta}), \bar{\theta}_j) | \theta_k\right] \tag{2.1.4.17}$$

Then d, t is BIC and t is balanced.

The ex-post individual rationality is defined in 2.1.4.15. There is a weaker form of IR for the Bayesian game:

Definition 2.1.4.4: Interim Individual Rationality

$$E\left[v_i(d(\overline{\theta}), \theta_i) + t_i(\overline{\theta}) \mid \theta_i\right] \geq 0 \tag{2.1.4.18}$$

There is a weaker form of IR: **ex-ante** IR:

$$E\left[v_i(d(\overline{\theta}), \theta_i) + t_i(\overline{\theta})\right] \geq 0 \tag{2.1.4.19}$$

The ex-ante IR is easy to satisfy for all games, but sometimes the interim IR can not be satisfied (not compatible with BIC).

Finding a mechanism that is efficient, incentive-compatible, balanced and interim individual rational is quite difficult.

2.2 Development Economics

2.2.1 Models of Development Economics

2.2.2 Clan Culture

Definition 2.2.2.1: Clan Culture

A clan is a consolidated kin group made up of component families that trace their patrilineal descent from a common ancestor.

History of Clans

"Modern" clan originated in the Song Dynasty (860-1279 CE). At that time Neo-Confucian ideology was formed, which provided the theoretical basis as well as clan organization structure design. The characteristics of "modern" clan culture:

1. The families of a clan lived in the same or several nearby communities;
2. Common properties and organized routine group activities, resource pooling;
3. Compilation of genealogies;
4. Own internal governance structures.

Currently, although China has been transitioning for a long time from a traditional society to a modern society, clan culture is still prevalent and has a broad impact on the lives of Chinese people, especially in rural areas.

[Bertrand and Schoar, 2006] shows the positive correlation between the fraction of family control among listed firms and family ties using cross-country level data. [Cheng et al., 2021] uses IV (the minimum distance to two prominent neo-Confucian academies, the Kaoting Academy (Kaoting Shuyuan) and the Xiangshan Academy (Xiangshan Shuyuan)) to identify that clan culture causes higher firm ownership concentration.

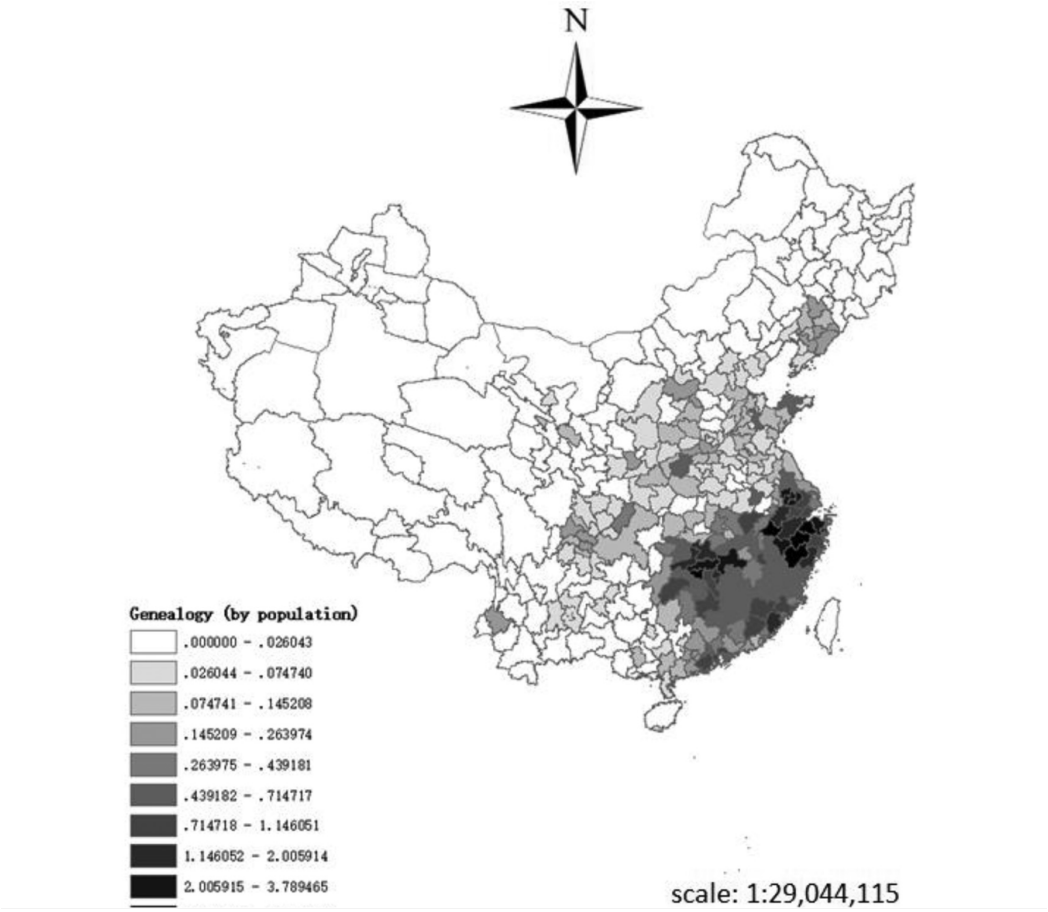


Figure 2.7: Clan Culture Intensity

Potential reasons that culture affects the concentration of family ownership:

1. Clan culture fosters high trust within the family and low trust in outsiders(**short-radius trust attitude**). According to agency-cost-based theories, family ownership can be concentrated in such a situation.

2. *Resources Pooling*: commom property ownership.

3. *Amenity Potential*: other things constant, owners subject to stronger influences of clan culture could have a higher utility.

[Zhang, 2020] estimates the effect of the clan on entrepreneurship. He finds that clan leads to a higher occurrence of entrepreneurship by helping overcome financing constraints and escape from local governments' "grabbing hand." [Zhang, 2019] investigates the relationship between the low take-up rate of social pensions and the clan culture intensity. In his article, dummy variable $temple_c$ (whether community c has ancestral temple) is constructed as the proxy variable for the strength of clan culture. Some interesting insights are obtained:

1. Clan culture is positively related to adults raising children for support in their old age;

2. Clan culture is associated with a larger number of children being born and a higher probability of having sons;

3. Clan culture is associated with a higher coresidence rate between old parents and adult sons;

4. Clan culture is associated with a higher likelihood of receiving financial transfers from non-coresident children;

5. Clan culture is associated with a lower likelihood of participating in rural pension programs.

[Cao et al., 2022a] There is much research about clan culture outside of Mainland China. [Yang, 2019] found the concave relationship the between the **heterogeneity** of clan family and the provision of public goods. This finding implies that group homogeneity yields not only benefits, but also some possible costs. reCommon Control Variables in Clan Research Identification(Individual Level):

1. *Hukou* Status;

Regional Level:

1. Distance to the sea;

Data resources in Clan Research

Remark 2.2.1

- *Comprehensive Catalogue of the Chinese Genealogy* can be used to construct the strength of clan culture;
-

2.3 Reduced-Form Identification

The main contents of this chapter are the notes of [Angrist and Pischke, 2014], [Angrist and Pischke, 2009].

2.3.1 Counterfactual Framework & Identification

Keywords 2.1

ATT, ATE, Counter-factual World, Potential Outcome

In the theory of the Neyman-Rubin Causal model, the outcome is Y_i, the potential outcome is Y_{1i}, Y_{0i}:

$$Y_i = \begin{cases} Y_{1i} & if D_i = 1 \\ Y_{0i} & if D_i = 0 \end{cases} \tag{2.3.1.1}$$

$$Y_i = Y_{0i} + (Y_{1i} - Y_{0i})D_i \tag{2.3.1.2}$$

Naive Comparison:

$$\begin{aligned} \{\mathbf{E}[Y_i|D_i = 1] - \mathbf{E}[Y_i|D_i = 0]\} = &\{\mathbf{E}[Y_{1i}|D_i = 1] - \mathbf{E}[Y_{0i}|D_i = 1]\} \\ &+ \{\mathbf{E}[Y_{0i}|D_i = 1] - \mathbf{E}[Y_{0i}|D_i = 0]\} \end{aligned} \tag{2.3.1.3}$$

- observed difference: $\mathbf{E}[Y_i|D_i = 1] - \mathbf{E}[Y_i|D_i = 0]$;
- ATE: $\mathbf{E}[Y_{1i}] - \mathbf{E}[Y_{0i}]$;
- ATT: $\mathbf{E}[Y_{1i}|D_i = 1] - \mathbf{E}[Y_{0i}|D_i = 1]$;
- selection bias: $\mathbf{E}[Y_{0i}|D_i = 1] - \mathbf{E}[Y_{0i}|D_i = 0]$.

The conditional average treatment effect (CATE) is defined as:

$$\tau_{CATE}(x) = \mathbb{E}(\tau_i|X_i = x) = \mathbb{E}(Y_{1i} - Y_{0i}|X_i = x), \tag{2.3.1.4}$$

where X_i is some additional characteristic. Note that the definition of ATE is on the whole population, including those who don't receive the treatment. The existence of the selection bias is due to the dependence between D_i and the **potential** outcome Y_{1i}, Y_{0i}. The relationship between CATE and ATE is:

$$\tau_{ATE} = \int \tau_{CATE}(x) f(x) dx, \tag{2.3.1.5}$$

$$\tau_{ATE} = \sum_{x \in \mathcal{X}} \tau_{CATE}(x) Pr(X_i = x). \tag{2.3.1.6}$$

Assumption 2.3.1.1: SUTVA (Stable Unit Treatment Value Assumption)

For two treatment vector $\mathbf{D}$ and $\mathbf{D}'$, if $\mathbf{D}_i = \mathbf{D}'_i$, then:

$$Y_i(\mathbf{D}) = Y_i(\mathbf{D}'). \tag{2.3.1.7}$$

It means that the outcome is only affected by own treatment status, and not the treatment status of others. SUTVA can be relaxed in many ways.

Remark 2.3.1

- **Estimand**: the quantity to be estimated;
- **Estimate**: the approximation of the estimand using a finite data sample;
- **Estimator**: the method or formula for arriving at the estimate for an estimand.

Definition 2.3.1.1: Identification

😎 Estimate the estimand from the data we observe.

There is another similar definition of identification from the perspective of structural form: a structure S is a DGP, let $\mathcal{S}$ be the set of all structures and $S_0 \in \mathcal{S}$ be the true structure. Identification is to test the hypothesis S_0 and estimate the parameters $\theta(S_0)$.

Randomization

Randomization can make $D_i \perp Y_{1i}, Y_{0i}$:

$$E[Y_{1i}|D_i = 1] = E[Y_{1i}|D_i = 0] = E[Y_{1i}] \tag{2.3.1.8}$$

$$E[Y_{0i}|D_i = 1] = E[Y_{0i}|D_i = 0] = E[Y_{0i}] \tag{2.3.1.9}$$

So selection $= \mathbf{E}[Y_{0i}|D_i = 1] - \mathbf{E}[Y_{0i}|D_i = 0] = 0$. 😎 Besides, by randomization, $ATT = ATE$. Randomization example:

- Rand HIE experiment: whether the insurance program makes people healthier;
- STAR: the effects of class size on education;
- OHP: this group experiment is not perfect because the group is not a determinant of whether to receive the treatment, but the treatment group does have a higher probability to get the treatment (**Instrumental Variable** can handle this situation);

Remark 2.3.2

- By randomization, the individual differences still exist;
- Checking for balance is an important step in randomization;
- The most critical idea of randomization is **Other Things Equal**(*ceteris paribus*);
- Randomization was invented by *Ronald Aylmer Fisher* in 1925.

2.3.2 Regression and Matching

Keywords 2.2

- CEF, CEF decomposition, ANOVA;
- regression justification,

Definition 2.3.2.1: Strong Ignorability

We say that D_i is strongly ignorable conditional on a vector $\mathbf{X}_i$ if:

1. $Y_i(0), Y_i(1) \perp D_i|\mathbf{X}_i$;

2. $\exists \varepsilon > 0$such that $\varepsilon < Pr(D_i = 1|\mathbf{X}_i) < 1 - \varepsilon$.

The first part of the definition is sometimes referred to *unconfoundeness* or *exogeneous*. This is a powerful tool, note that under the strong ignorability:

$$\mathbb{E}(Y_i(0)|\boldsymbol{X}_i) = \mathbb{E}(Y_i(0)|D_i = 0, \mathbf{X}_i) = \mathbb{E}(Y_i|D_i = 0, \boldsymbol{X}_i). \tag{2.3.2.1}$$

This means that **we can interchange counterfactuals and realized data in conditionals**.

Theorem 2.3.2.1: Identification of the ATE

If D_i is strongly ignorable conditional on $\mathbf{X}_i$, then

$$\mathrm{E}(\tau_i) = \sum_{x \in \operatorname{Supp} X_i} \left(\mathrm{E}(Y_i|D_i = 1, \mathbf{X}_i = x) - \mathrm{E}(Y_i|D_i = 0, \mathbf{X}_i = x)\right) Pr(\mathbf{X}_i = x) \tag{2.3.2.2}$$

Conditional Expectation Function (CEF) is a population concept:

$$\begin{aligned} E[Y_i|X_i = x] &= \int t f_y(t|X_i = x)dt \\ E[Y_i|X_i = x] &= \sum_t tP(Y_i = t|X_i = x) \end{aligned} \tag{2.3.2.3}$$

Lemma 2.3.2.1: The law of iterated expectations

$$E\left[\mathrm{y}_i|\mathrm{X}_i = x\right] = \int t f_{\mathrm{y}}\left(t|\mathrm{X}_i = x\right) dt. \tag{2.3.2.4}$$

Proof 2.3.1

$$\begin{aligned} E\{E\left[\mathrm{y}_i|\mathrm{X}_i\right]\} &= \int E\left[\mathrm{y}_i|\mathrm{X}_i = u\right] g_x(u)du \\ &= \int \left[\int t f_{\mathrm{y}}\left(t|\mathrm{X}_i = u\right) dt\right] g_x(u)du \\ &= = \int \int t f_{\mathrm{y}}\left(t|\mathrm{X}_i = u\right) g_x(u)du\,dt \\ &= \int t \left[\int f_{\mathrm{y}}\left(t|\mathrm{X}_i = u\right) g_x(u)du\right] dt = \int t \left[\int f_{xy}\left(u, t\right) du\right] dt \\ &= \int t g_y(t)dt. \end{aligned} \tag{2.3.2.5}$$

Q.E.D.

💗 3 important property of CEF:

Theorem 2.3.2.2: CEF Decompostion Property

$$Y_i = E[Y_i|X_i] + \epsilon_i \tag{2.3.2.6}$$

where ϵ_i is mean independent of X_i, and X_i is uncorrelated with any function of X_i.

Proof 2.3.2

Take the expectation of X_i at both sides:

$$\begin{aligned} E[Y_i|X_i] &= E[E[Y_i|X_i]|X_i] + E[\epsilon_i|X_i] \\ E[\epsilon_i|X_i] &= E[Y_i|X_i] - E[Y_i|X_i] = 0 \end{aligned} \tag{2.3.2.7}$$

$$E[\epsilon_i] = \int_{X_i} f_x(t)E[\epsilon_i|X_i]dt = \int_{X_i} 0dt = 0 = E[epsilon_i|X_i] \tag{2.3.2.8}$$

Q.E.D.

This means that Y_i can be decomposed into 2 parts: explaind by X_i and terms uncorrelated with X_i.

Theorem 2.3.2.3: CEF Prediction Property

CEF is the best estimator of Y_i in the MMSE sense, which means:

$$E\left[\mathrm{y}_i|\mathrm{X}_i\right] = \arg\min_{m(\mathrm{X}_i)} E\left[\left(\mathrm{Y}_i - m\left(\mathrm{X}_i\right)\right)^2\right] \tag{2.3.2.9}$$

Proof 2.3.3

$$\begin{aligned} \left(\mathrm{Y}_i - m\left(\mathrm{X}_i\right)\right)^2 &= \left(\left(\mathrm{Y}_i - E\left[\mathrm{Y}_i|\mathrm{X}_i\right]\right) + \left(E\left[\mathrm{Y}_i|\mathrm{X}_i\right] - m\left(\mathrm{X}_i\right)\right)\right)^2 \\ &= \left(\mathrm{Y}_i - E\left[\mathrm{Y}_i|\mathrm{X}_i\right]\right)^2 + 2\left(E\left[\mathrm{Y}_i|\mathrm{X}_i\right] - m\left(\mathrm{X}_i\right)\right)\left(\mathrm{Y}_i - E\left[\mathrm{Y}_i|\mathrm{X}_i\right]\right) \\ &\quad + \left(E\left[\mathrm{Y}_i|\mathrm{X}_i\right] - m\left(\mathrm{X}_i\right)\right)^2 \end{aligned} \tag{2.3.2.10}$$

The formula has the lowest constant value by setting $m\left(\mathrm{X}_i\right) = \text{CEF}$.

Q.E.D.

Theorem 2.3.2.4: ANOVA Theorem

$$\mathrm{V}(Y_i) = E[\mathrm{V}(Y_i|X_i)] + \mathrm{V}(E[Y_i|X_i]) \tag{2.3.2.11}$$

This indicates that the variance of Y_i can be decomposed into two parts:

1. the variance of the CEF;
2. the variance of the residual;

Remark 2.3.3

- The CEF property dosen't rely on any assumption! It has nothing to do with regression right now;
- If X_i is not mean independent of Y_i, then by ANOVA theorem, the variance of the outcome variable controlled by X_i could be smaller;

$$\begin{aligned} \beta &= \arg\min_b E[(Y_i - X_i'b)^2] \\ 1storder &: E[X_i(Y_i - X_i'b)] = 0 \\ solution &: \beta = E[X_iX_i']^{-1}E[X_iY_i] \end{aligned} \tag{2.3.2.12}$$

Theorem 2.3.2.5: Regression Anatomy

$$\beta_k = \frac{Cov(Y_i, \tilde{X_{ki}})}{V(\tilde{X_{ki}})} \tag{2.3.2.13}$$

Corollary 2.3.2.1: Bivariate Case

$$\beta = \frac{Cov(Y_i, \tilde{X_i})}{V(\tilde{X_i})} \tag{2.3.2.14}$$

Proof 2.3.4

Substitute

$$Y_i = \alpha + \beta_1 x_{1i} + \cdots + \beta_k x_{ki} + e_i \tag{2.3.2.15}$$

$\tilde{x}_{ki}$ is uncorrelated with e_i and other covariates by construction, thus $Cov(\tilde{x}_{ki}, x_{ki}) = Var(\tilde{x}_{ki})$, thus $Cov(Y_i, \tilde{x_{ki}} = \beta_k x_{ki})$.

Q.E.D.

Remark 2.3.4

The regression anatomy shows that each β_k in multi-regression is the bivariate slope after "partitioning out" all the other regressors.

☺ Why the population regression coefficient is what we are interested in (Link with CEF):

Theorem 2.3.2.6: Regression Justification

1. Suppose the CEF is linear, then the population regression function is it;
2. In any condition, $X_i'^{\beta}$ is the best predictor of Y_i in a MMSE sense;
3. The function $X_i'^{\beta}$ provides the MMSE linear approximation to $E[Y_i|X_i]$.

$$\beta = \arg\min_b E\{(E[\mathrm{Y}_i|\mathrm{X}_i] - \mathrm{X}_i'b)^2\} \tag{2.3.2.16}$$

Proof 2.3.5

Suppose $E[\mathrm{Y}_i|\mathrm{X}_i] = X_i^{'\beta^*}$. By regression decomposition theorem:

$$\begin{aligned} E[X_i(Y_i - X_i'\beta^*)] &= 0 \\ \beta^* = E[X_i X_i']^{-1} E[X_i Y_i] &= \tilde{\beta} \end{aligned} \tag{2.3.2.17}$$

$$\begin{aligned} \left(\mathrm{Y}_i - \mathrm{X}_i'b\right)^2 = & \{(\mathrm{y}_i - E[\mathrm{y}_i|\mathrm{X}_i]) + (E[\mathrm{y}_i|\mathrm{X}_i] - \mathrm{X}_i'b)\}^2 \\ = & (\mathrm{y}_i - E[\mathrm{y}_i|\mathrm{X}_i])^2 + (E[\mathrm{y}_i|\mathrm{X}_i] - \mathrm{X}_i'b)^2 \\ & +2(\mathrm{y}_i - E[\mathrm{y}_i|\mathrm{X}_i])(E[\mathrm{y}_i|\mathrm{X}_i] - \mathrm{X}_i'b). \end{aligned} \tag{2.3.2.18}$$

Q.E.D.

Corollary 2.3.2.2

- For the saturated model, the population linear regression is the CEF;
- For the single dummy variable, the coefficient is the mean probability of receiving treatment;

2.3.3 Instrumental Variables

Consider the demand equilibrium model:

$$\begin{cases} q_t^d = \alpha_0 + \alpha_1 p_t + u_t & \text{(Demand)} \\ q_t^s = \beta_0 + \beta_1 p_t + v_t & \text{(Supply)} \\ q_t^d = q_t^s & \text{(Equilibrium)} \end{cases} \tag{2.3.3.1}$$

Local Average Treatment Effect

1. the *instrument*: Z_i;
2. the *treatment*: D_i;
3. the *outcome*: Y_i.

The IV chain can lead us to the definition of LATE: the **first-stage** $\phi = E[D_i|Z_i = 1] - E[D_i|Z_i = 0]$; the **reduced-form** $\rho = E[Y_i|Z_i = 1] - E[Y_i|Z_i = 0]$. The LATE is defined as $\lambda = \frac{\rho}{\phi} = \frac{E[Y_i|Z_i=1]-E[Y_i|Z_i=0]}{E[D_i|Z_i=1]-E[D_i|Z_i=0]}$, which is the ratio of the reduced form to the first stage.

LATE is the average causal effect of D on Y for those whose status D is determined **solely** by Z (*compliers*), but only under the monotonicity assumption: there is no *defier* in the sample.

$$\lambda = \frac{\rho}{\phi} = E[Y_{1i} - Y_{0i}|C_i = 1] \tag{2.3.3.2}$$

To calculate the standard error of the 2SLS parameter, one can regress

$$\eta_i = Y_i - \alpha_2 - \lambda_{2SLS} D_i - \gamma_2 A_i.$$

The standard error is given by:

$$SE(\hat{\lambda}_{2SLS}) = \frac{\sigma_n}{\sqrt{n}} \times \frac{1}{\sigma_{\hat{D}}} \tag{2.3.3.3}$$

2SLS

Compared to computing LATE by hand, 2SLS (2-stage-least-squares) has two advantages:

1. 2SLS can use multiple IVs simultaneously;
2. 2SLS can control for covariates.

$$\begin{cases} \hat{D}_i = \alpha_1 + \phi Z_i + \gamma_1 A_i & \text{first-stage} \\ Y_i = \alpha_2 + \lambda_{2SLS} \hat{D}_i + \gamma_2 A_i + e_{2i} & \text{second-stage} \end{cases} \tag{2.3.3.4}$$

Validity of IV

If you can not see it in the reduced form, it ain't there.

2.3.4 Asymptotic Analysis

2.3.5 Empirical Classics

Returns to Schooling

It's been a long history to study the effects of education on the salary. Counterfactuals here are multi-faceted: instead of the single contrast $Y_{1i} - Y0i$, it is more often to study every possible schooling choice $Y_{n,i} - Y_{m,i}$, where n, m are the years of education. [Mincer, 1974] uses U.S. census data to do the following regression:

$$\ln Y_i = \alpha + \rho S_i + \beta_1 X_i + \beta_2 X_i^2 + e_i, \tag{2.3.5.1}$$

where $X_i = Age_i - S_i - 6$, defined as the *potential experience*, and S_i is the year of education. The result shows that one-year more eduction would bring 10.7% more earnings.

This result may be biased because it doesn't consider an individual's ability, which is correlated both with earnings and education.

To handle the ability bias, [Griliches, 1977] includes IQ as a control variable to eliminate the influence of personal ability, the new ρ estimated is 5.9%, which is smaller than expected.

In the process of handling ability bias, be aware of the bad control: which is controlling the occupation. This is because the occupation is a variable *after* the education years, and some effect of the education is through occupation, controlling occupation would underestimate the effect.

The above regressions don't consider other variables, one approach to handle this is by comparing the twins ([Ashenfelter and Krueger, 1994], [Ashenfelter and Rouse, 1998]), since each person of the twins is almost identical in IQ, family background, etc. The long regression is written as:

$$\ln Y_{i,f} = \alpha + \rho S_{i,f} + \lambda A_{i,f} + e_{i,f}, \tag{2.3.5.2}$$

where subscript i, f stand for individual and family respectively, $i = 1, 2$ indexes twin siblings. If we consider the regression within a family:

$$\ln Y_{1,f} = \alpha + \rho S_{1,f} + \lambda A_f + e_{1,f} \ln Y_{2,f} = \alpha + \rho S_{2,f} + \lambda A_f + e_{2,f} \tag{2.3.5.3}$$

Subtracting them would lead to:

$$\ln Y_{1,f} - \ln Y_{2,f} = \rho(S_{1,f} - S_{2,f}) + e_{1,f} - e_{2,f} \tag{2.3.5.4}$$

This regression gives an estimation of 6.2% for ρ. But in practice, the collected data encounters the misreported issue, which can bring the **Attenuation bias**.

Proof 2.3.6: Attenuation Bias

$C(\cdot)$ denotes the covariance among two variables. Consider the regression:

$$Y_i = \alpha + \beta S_i^\star + e_i \tag{2.3.5.5}$$

But the data for $S_i^\star$ is misreported, the observed $S_i = S_i^\star + m_i$, where we assume that $\mathbb{E}[m_i] = 0, C(S_i^\star, m_i) = 0$. We assume the real parameter $\beta = \frac{C(Y_i, S_i^\star)}{V(S_i^\star)}$, and the biased parameter is $\beta_b = \frac{C(Y_i, S_i)}{V(S_i)}$:

$$\begin{aligned} \beta_b &= \frac{C(Y_i, S_i)}{V(S_i)} \\ &= \frac{C(\alpha + \beta S_i^\star + e_i, S_i^\star + m_i)}{V(S_i)} \\ &= \frac{C(Y_i, S_i^\star)}{V(S_i)} = \beta \frac{V(S_i^\star)}{V(S_i)}. \end{aligned} \tag{2.3.5.6}$$

Because $V(S_i) = V(S_i^\star) + V(m_i)$, which means $\beta_b = r\beta$, where

$$r(V(S_i^\star)) = \frac{V(S_i^\star)}{V(S_i^\star) + V(m_i)} < 1. \tag{2.3.5.7}$$

Corollary 2.3.5.1: Adding Covariates Can Exacerbate Attenuation Bias

When adding covariates X_i, we would get:

$$\beta_b = \frac{V(\tilde{S_i^\star})}{V(\tilde{S_i^\star}) + V(m_i)} \beta = r(V(\tilde{S_i^\star}))\beta \tag{2.3.5.8}$$

Be aware that $V(\tilde{S_i^\star}) < V(S_i^\star)$ because $V(S_i^\star) = V(\gamma X_i + \tilde{S_i^\star})$. And $r(\cdot)$ is a decreasing function, so adding covariates would exacerbate attenuation bias.

Q.E.D.

2.3.6 Structural Equations

What is the structural identification?

Taxonomy for the types of empirical works ([Haile, 2020]):

1. Descriptive (association and facts about data);
2. Structural (**estimate or parameters features of a data generating process**, causal identification is a special case of structural form.)

,

Definition 2.3.6.1: Structural Form & Reduced Form

A structural form is a DGP $f(Y, X, Z, U)$, where Y is the **endogenous** variable, X is the **exogeneous** variable, and U is the *unobservables* ("structural errors"). A reduced form solves the structural form and gets $Y = f(X, Z, U)$, i.e., a functional or stochastic mapping for which the inputs are exogenous variables and unobservables, and the outputs are endogenous variables.

Remark 2.3.5

- *exogeneous*: one taken as given in (not determined within) the model; or one satisfying some kind of independence condition with respect to unobservables;
- This means that sometimes there is no reduced form for a structural form.

Example 2.3.6.1

Structural Form & Reduced Form

Solution:

The structural form is given as:

$$Q = D(P, X, U_d) \tag{2.3.6.1}$$
$$P = MC(Q, Z, U_s) \tag{2.3.6.2}$$

Solving the equilibrium $Q = P$ yields the reduced form relations:

$$P = p(Z, X, U_s, U_d) \tag{2.3.6.3}$$
$$Q = q(Z, X, U_s, U_d) \tag{2.3.6.4}$$

Remark 2.3.6

- Reduced form isn't equal to the use of IV, RD, etc;
- Structural form isn't equal to the complex models;
- The key is to name the structural errors.

Consider the following economic model:

$$Y_i = \alpha + \beta D_i + \varepsilon_i. \tag{2.3.6.5}$$

Using the potential outcome framework, we can obtain:

$$\begin{aligned} Y_i &= Y_i(0)(1 - D_i) + Y_i(1)D_i \\ &= Y_i(0) + \tau_i D_i \\ &= Y_i(0) + \tau D_i + (\tau_i - \tau)D_i \\ &= \underbrace{E(Y_i(0)|D_i = 0)}_{\alpha} + \underbrace{\tau}_{\beta} D_i + \underbrace{(\tau_i - \tau)D_i + (Y_i(0) - E(Y_i(0)|D_i = 0))}_{\varepsilon_i} \end{aligned} \tag{2.3.6.6}$$

We can observe $E(Y_i|D_i)$, which means we can recover:

- $E(Y_i|D_i = 1) = \alpha+\tau+E(\varepsilon_i|D_i = 1)$, where $E(\varepsilon_i|D_i = 1) = (E(\tau_i|D_i = 1)-\tau)+E(Y_i(0)|D_i = 1)-E(Y_i(0)|D_i = 0)$;
- $E(Y_i|D_i = 0) = \alpha + E(\varepsilon_i|D_i = 0)$, where $E(\varepsilon_i|D_i = 0) = 0$

We can obtain an estimation on β if D_i is strongly ignorable: $E(\varepsilon_i|D_i = 1) - E(\varepsilon_i|D_i = 0) = 0$.

There are one-to-one mappings between the potential outcome framework and structural regressions.

Chapter 3

Data Science

3.1 Machine Learning

3.1.1 Learning Theory

Feasibility of Learning (generalization)

The question is: Does training say anything about testing? In general, the answer is NO. However, we can make some reasonable assumptions on the connection between the training data and testing data, and predict the test data **in a probabilistic way**. The most common assumption:

> **Assumption 3.1.1.1**
> The training and test data are independent ansd identically distributed (i.i.d.)

- $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\} \subseteq \mathcal{X}$ are samples;
- $\{y_1, \ldots, y_n\} \subseteq \mathcal{Y}$ are labels generated by g, consider binary case $y_i \in \{-1, +1\}$;
- The algorithm learns from a hypothesis space $\mathcal{H} \ni f_{\boldsymbol{\theta}} : \mathcal{X} \to \{-1, +1\}$, consider the Perceptron: $\text{sign}(f_{\boldsymbol{\theta}}(\boldsymbol{x}) = \boldsymbol{\theta}^\top \boldsymbol{x}) \to \{-1, +1\}$;
- The learning goal is to learn $f \approx g$, i.e., $e(f(\boldsymbol{x}), g(\boldsymbol{x}))$ should be as small as possible for all $\boldsymbol{x}$.

  There are two types of errors: in-sample v.s. out-of-sample error:

$$\text{Er}_{\text{in}} = \frac{1}{n} \sum_{i=1}^{n} e(f(\boldsymbol{x}_i), g(\boldsymbol{x}_i))$$

$$\text{Er}_{\text{out}} = \mathbb{E}_{\boldsymbol{x} \sim \mathcal{D}} \left[e(f(\boldsymbol{x}), g(\boldsymbol{x})) \right]$$

  Learning g is to make Er_{out} small, but it's not computable. We can split it into two parts:

$$\text{Er}_{\text{out}} = \underbrace{Er_{out} - Er_{in}}_{\text{generalization error}} + \underbrace{Er_{in}}_{\text{traning error}}$$

  On the generalization side, we need less complex $\mathcal{H}$; on the training side, we need more complex $\mathcal{H}$.

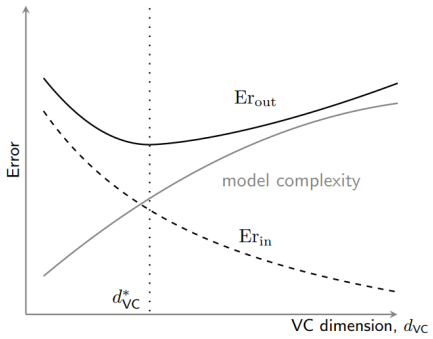


Figure 3.1: Learning Curve from VC Analysis

Under the i.i.d. assumption, we have the following formula given training samples $\mathcal{S} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$:

$$\mathbb{E}_{\mathcal{S}\sim_{i.i.d.}\mathcal{D}}\left[\mathrm{Er}_{\mathrm{in}}(f)\right] = \mathrm{Er}_{\mathrm{out}}(f) \tag{3.1.1.1}$$

Proof 3.1.1

$$\begin{aligned}\mathbb{E}_{\mathcal{S}\sim_{i.i.d.}\mathcal{D}}\left[\mathrm{Er}_{\mathrm{in}}(f)\right] &= \mathbb{E}_{\mathcal{S}\sim_{i.i.d.}\mathcal{D}}\left[\frac{1}{n}\sum_{i=1}^{n} e(f(\boldsymbol{x}_i), g(\boldsymbol{x}_i))\right] \\ &= \frac{1}{n}\sum_{i=1}^{n}\mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}}\left[e(f(\boldsymbol{x}), g(\boldsymbol{x}))\right] \quad \text{(since i.i.d.)} \\ &= \mathrm{Er}_{\mathrm{out}}(f)\end{aligned} \tag{3.1.1.2}$$

Q.E.D.

The interpretation is that $\mathrm{Er}_{\mathrm{in}}(f)$ is an unbiased estimator for $\mathrm{Er}_{\mathrm{out}}(f)$. Because n is finite, we need non-asymptotic results.

Assumption 3.1.1.2: Finite hypothesis space

The cardinality $|\mathcal{H}| < +\infty$.

Lemma 3.1.1.1: High probability bounds for a Fixed f

Fix any model $f \in \mathcal{H} : \mathcal{X} \mapsto \{-1, 1\}$, for any $t > 0$:

$$\Pr\left[\left|\mathrm{Er}_{\mathrm{in}}(f) - \mathrm{Er}_{\mathrm{out}}(f)\right| \leq t\right] \geq 1 - 2e^{-2nt^2} \tag{3.1.1.3}$$

Proof 3.1.2

We first prove the bound of one side. Note that $e(f(\boldsymbol{x}_i), g(\boldsymbol{x}_i))$ is bounded within $[0, 1]$, using Hoeffding's inequality:

$$\begin{aligned}\Pr\left[\ \mathrm{Er}_{\mathrm{in}}(f) - \mathrm{Er}_{\mathrm{out}}(f) \geq t\right] &= \Pr\left[\frac{1}{n}\sum_{i=1}^{n}(e(f(\boldsymbol{x}_i), g(\boldsymbol{x}_i)) - \mathbb{E}\left[\frac{1}{n}\sum_{i=1}^{n} e(f(\boldsymbol{x}_i), g(\boldsymbol{x}_i))\right] \geq t\right] \\ &= \Pr\left[\sum_{i=1}^{n}(e(f(\boldsymbol{x}_i), g(\boldsymbol{x}_i)) - \mathbb{E}[e(f(\boldsymbol{x}_i), g(\boldsymbol{x}_i))]) \geq nt\right] \\ &\leq e^{-2nt^2}.\end{aligned} \tag{3.1.1.4}$$

Next, let $\mathrm{Er}_{\mathrm{in}}(f) = -\mathrm{Er}_{\mathrm{in}}(f)$, we have the bound on another side. Using *union bound inequality*, we can obtain the conclusion.

Q.E.D.

Proposition 3.1.1.1: Generalization Bound for a Fixed f

For any model f in the binary clasification case. For any $\delta > 0$, the following bound holds with probability at least $1 - \delta$:

$$\mathrm{Er}_{\mathrm{out}}(f) \leq \mathrm{Er}_{\mathrm{in}}(f) + \sqrt{\frac{\log\left(\frac{2}{\delta}\right)}{2n}}$$

Proof 3.1.3

Let $\delta = 2e^{-2nt^2}$ in the above lemma.

Q.E.D.

Now we need to generalize the bound on a fixed f to a hypothesis space $\mathcal{H}$.

Theorem 3.1.1.1: Generalization for finite hypothesis space

For any $\delta > 0$, the following inequality holds with probability at least $1 - \delta$:

$$\forall f \in \mathcal{H} \quad \mathrm{Er}_{\mathrm{out}}(f) \leq \mathrm{Er}_{\mathrm{in}}(f) + \sqrt{\frac{\log\left(\frac{2|\mathcal{H}|}{\delta}\right)}{2n}} \tag{3.1.1.5}$$

Proof 3.1.4

Using the union bound inequality, we have

$$\begin{aligned} &\Pr\left[\exists f \in \mathcal{H} \quad s.t. \left|\mathrm{Er}_{\mathrm{in}}(f) - \mathrm{Er}_{\mathrm{out}}(f)\right| \geq t\right] \\ =& \Pr[|\ \mathrm{Er}_{\mathrm{in}}(f_1) - \mathrm{Er}_{\mathrm{out}}(f_1)| \geq t, or \cdots or \mid \mathrm{Er}_{\mathrm{in}}(f_{|\mathcal{H}|}) - \mathrm{Er}_{\mathrm{out}}(f_{|\mathcal{H}|})| \geq t] \\ \overset{\text{union bound}}{\leq} & \sum_{i=1}^{|\mathcal{H}|} \Pr\left[|\ \mathrm{Er}_{\mathrm{in}}(f_i) - \mathrm{Er}_{\mathrm{out}}(f_i)| \geq t\right] \leq 2|\mathcal{H}|e^{-2nt^2} \end{aligned} \tag{3.1.1.6}$$

Q.E.D.

However, $|\mathcal{H}|$ is always infinite in practice, we should find a way to measure the complexity of $\mathcal{H}$ more smartly, which is called the **growth function**.

Definition 3.1.1.1: Dichotomy & Growth Function

Given $\{x_1, \ldots, x_n\}$, the dichotomies generated by $\mathcal{H}$ are defined by:

$$\mathcal{H}(x_1, \ldots, x_n) = \{(f(x_1), \ldots, f(x_n)) : f \in \mathcal{H}\} \tag{3.1.1.7}$$

Growth function is the number of dichotomies:

$$\mathcal{G}_{\mathcal{H}}(n) = \max_{\{x_1, \ldots, x_n\} \subseteq \mathcal{X}} |\mathcal{H}(x_1, \ldots, x_n)| \tag{3.1.1.8}$$

Intuition: the growth function is the maximum number of ways to label a n-points dataset, it counts the most dichotomies that can possibly be generated on any n points in $\mathcal{X}$. In the binary classification case, we have $\mathcal{G}_{\mathcal{H}}(n) \leq 2^n$.

Computing the growth function is often untractable. However, we don't need to calculate it, we only need to bound it.

Definition 3.1.1.2: Break Point & VC-dimension

If no dataset of size k can be shattered by $\mathcal{H}$, then k is said to be the break point for $\mathcal{H}$. For example, the break point for two-dimensional perceptron is 4.

We have the following inequality to bound the growth function in polynomial. If $\mathcal{G}_{\mathcal{H}}(n) \leq 2^k$ for some k, then:

$$\mathcal{G}_{\mathcal{H}}(n) \leq \sum_{i=0}^{k-1} \binom{n}{i}, \quad \forall n \geq 1. \tag{3.1.1.9}$$

Vapnik-Chervonenkis (VC) dimension is defined as:

$$d_{\mathrm{VC}}(\mathcal{H}) := \max\{n : \mathcal{G}_{\mathcal{H}}(n) = 2^n\} \tag{3.1.1.10}$$

By definition, $k = d_{\mathrm{VC}} + 1$. Then we have:

$$\mathcal{G}_{\mathcal{H}}(n) \leq n^{d_{\mathrm{VC}}} + 1 \tag{3.1.1.11}$$

😎 For d-dimensioanl binary linear classifier, we have $d_{VC} = d + 1$.

Theorem 3.1.1.2: Generalization Bound for VC Dimension

For any $\delta > 0$, the following bounds hold with probability at least $1 - \delta$:

$$\forall f \in \mathcal{H} \quad \mathrm{Er}_{\mathrm{out}}(f) \leq \mathrm{Er}_{\mathrm{in}}(f) + \sqrt{\frac{8}{n} \log\left(\frac{4\mathcal{G}_{\mathcal{H}}(2n)}{\delta}\right)} \tag{3.1.1.12}$$

It's equivalent to:

$$\forall f \in \mathcal{H} \quad \mathrm{Er}_{\mathrm{out}}(f) \leq \mathrm{Er}_{\mathrm{in}}(f) + \sqrt{\frac{8}{n} \log\left(\frac{4((2n)^{d_{\mathrm{vc}}} + 1)}{\delta}\right)} \tag{3.1.1.13}$$

Generally, the VC generalization bound has the form:

$$\forall f \in \mathcal{H} \quad \mathrm{Er}_{\mathrm{out}}(f) \leq \mathrm{Er}_{\mathrm{in}}(f) + O\left(\sqrt{\frac{d_{\mathrm{VC}}}{n}}\right) \tag{3.1.1.14}$$

Sample complexity: The sample complexity denotes how many training examples n are needed to achieve a certain generalization performance. Suppose we want the result to hold with probability at least 1δ, and generalization error to be less than ϵ, we need to have samples:

$$n \geq \frac{8}{\varepsilon^2} \log\left(\frac{4((2n)^{d_{\mathrm{vc}}} + 1)}{\delta}\right) \tag{3.1.1.15}$$

3.1.2 Classification

Perceptron

perceptron is a linear classifier with the form:

$$f_{\boldsymbol{\theta}}(\boldsymbol{x}) = \boldsymbol{\theta}^\top \boldsymbol{x} + b$$

To simpify the notation, we often absord b into $\theta^\top$ and add $x_0 = 1$ for all samples:

$$\widetilde{\boldsymbol{x}} = (x_0, \boldsymbol{x}) \in \mathbb{R}^{d+1}, \widetilde{\boldsymbol{\theta}} = (\theta_0, \boldsymbol{\theta}) \in \mathbb{R}^{d+1}$$

The formal linear classification model is given by:

$$f_{\boldsymbol{\theta}}(\boldsymbol{x}) = \boldsymbol{\theta}^\top \boldsymbol{x} \quad \text{and} \quad y = \mathrm{sign}(f_{\boldsymbol{\theta}}(\boldsymbol{x}))$$

Implementation:

- Pick a misclassified data whose $\mathrm{sign}(f_{\boldsymbol{\theta}}(\boldsymbol{x_i})) \neq y_i$;
- Update rule: $\theta = \theta + y_i \boldsymbol{x}_i$ (this will push the hyperplane toward the misclassified sample);
- Iterate until there is no misclassified data.

Logistic Regression

Consider a Bayesian classifier: $y \leftarrow \underset{y \in \mathcal{Y}}{\operatorname{argmax}} \Pr[y|\boldsymbol{x}]$. We can learn an estimator $\Pr_{\boldsymbol{\theta}}[y|\boldsymbol{x}]$ for $\Pr[y|\boldsymbol{x}]$ based on the training data.

Definition 3.1.2.1: Logistic/Sigmoid Function

$$h(t) = \frac{e^t}{e^t + 1} = \frac{1}{1 + e^{-t}}$$

$h(t)$ can be used to approximate $\Pr[y|\boldsymbol{x}]$:

$$\Pr_{\boldsymbol{\theta}}\left[y = +1|\boldsymbol{x}\right] = h(\boldsymbol{\theta}^\top \boldsymbol{x}) = \frac{1}{1 + e^{-\boldsymbol{\theta}^\top \boldsymbol{x}}}$$

$$\Pr_{\boldsymbol{\theta}}\left[y = -1|\boldsymbol{x}\right] = 1 - \Pr\left[y = +1|\boldsymbol{x}; \boldsymbol{\theta}\right] = \frac{1}{1 + e^{\boldsymbol{\theta}^\top \boldsymbol{x}}}$$

Finally, we have:

$$\Pr_{\boldsymbol{\theta}}\left[y|\boldsymbol{x}\right] = \frac{1}{1 + \exp\left(-y \cdot \boldsymbol{\theta}^\top \boldsymbol{x}\right)} \tag{3.1.2.1}$$

Using MLE to estimate θ, the log-likelihood of all data $\{(x_i, y_i)\}$ is given as:

$$\sum_{i=1}^{n} \log(\Pr_{\boldsymbol{\theta}[y_i|\boldsymbol{x}_i])=-\sum_{i=1}^{n} \log(1+\exp(-y_i \cdot \boldsymbol{\theta}^\top \boldsymbol{x}_i))}$$

The MLE is equivalent to minimize the logistic loss (cross-entropy loss):

$$\widehat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta} \in \mathbb{R}^d}{\operatorname{argmin}} \quad \frac{1}{n} \sum_{i=1}^{n} \underbrace{\log\left(1 + \exp\left(-y_i \cdot \boldsymbol{\theta}^\top \boldsymbol{x}_i\right)\right)}_{\ell(f_{\boldsymbol{\theta}}(\boldsymbol{x}_i), y_i)} \tag{3.1.2.2}$$

Remark 3.1.1

- Comparing to LS, no closed-form solution;
- Comparing to Perceptron, no linearly-separable requirement (better boundary);
- Can be extended to multi-class version: Softmax:

$$\Pr_{\boldsymbol{\Theta}[y_i=k|\boldsymbol{x}_i]=\frac{\exp(\boldsymbol{\theta}_k^\top \boldsymbol{x}_i)}{\sum_{j=1}^{K} \exp(\boldsymbol{\theta}_j^\top \boldsymbol{x}_i)}}$$

Decision Tree

Definition 3.1.2.2: Entropy & Information Gain

Entropy measures the uncertainty in one group:

$$H(X) = -\sum_{i=1}^{n} P(x_i) \log_b P(x_i) \tag{3.1.2.3}$$

Splitting one group D into multiple groups D_i, the information gain is given by:

$$IG = H(D) - \sum_i \frac{|D_i|}{|D|} H(D_i) \tag{3.1.2.4}$$

3.1.3 Rrgression

Linear Regression

Given data $\{(x_i, y_i)\}_{i=1}^n$, where $x_i \in \mathbb{R}^d$, we use a linear model to predict y_i for a given x_i:

$$y_i \approx f_\theta(x_i) = \theta x_i \tag{3.1.3.1}$$

Using **least square** (LS) to solve $\hat{\theta}$:

$$\widehat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta} \in \mathbb{R}^d}{\operatorname{argmin}} \frac{1}{n} \sum_{i=1}^n \ell\left(f_{\boldsymbol{\theta}}(\boldsymbol{x}_i), y_i\right)$$

Using ℓ_2 loss:

$$\widehat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta}}{\operatorname{argmin}} \frac{1}{n} \sum_{i=1}^n (\boldsymbol{\theta}^\top \boldsymbol{x}_i - y_i)^2$$

To solve the LS:

$$\begin{aligned} \mathcal{L}(\boldsymbol{\theta}) &= (\boldsymbol{X}\boldsymbol{\theta} - \boldsymbol{y})^\top (\boldsymbol{X}\boldsymbol{\theta} - \boldsymbol{y}) \\ &= \boldsymbol{\theta}^\top \boldsymbol{X}^\top \boldsymbol{X} \boldsymbol{\theta} - 2\boldsymbol{\theta}^\top \boldsymbol{X}^\top \boldsymbol{y} + \boldsymbol{y}^\top \boldsymbol{y} \end{aligned}$$

When $n \geq d$, we can directly set $\nabla_\theta \mathcal{L}(\boldsymbol{\theta}) = 2\boldsymbol{X}^\top(\boldsymbol{X}\boldsymbol{\theta} - \boldsymbol{y}) = 0$ and we could obtain:

$$\widehat{\boldsymbol{\theta}} = \left(\boldsymbol{X}^\top \boldsymbol{X}\right)^{-1} \boldsymbol{X}^\top \boldsymbol{y} \tag{3.1.3.2}$$

Remark 3.1.2

LS is a maximum likelihood estimator under Gaussian noise assumption:

$$y = X\theta + \epsilon \quad \text{with} \quad \epsilon \sim \mathcal{N}(0, \boldsymbol{\Sigma})$$

3.1.4 Imitation Learning

DAGGER

Imitation learning, which learns from the demonstration, is a kind of supervised learning. Generally, experts provide a set of demonstration trajectories, which are sequences of states and actions. Assume we know:

- State space and action space;
- Access to transition oracle $\mathbb{P}(s' \mid s, a)$;
- Set of one or more teacher demonstrations $(s_0, a_0, s_1, a_1, \ldots)$ where actions are drawn from the teacher's policy $\pi^\star$.

Algorithm 1: DAGGER

Initialize $\mathcal{D} \leftarrow \emptyset$
Initialize $\hat{\pi}_1$ to any policy in Π
for $i = 1$ *to* N **do**
    Let $\pi_i = \beta_i \pi^* + (1 - \beta_i)\hat{\pi}_i$
    Sample T-step trajectories using π_i
    Get dataset $\mathcal{D}_i = \{(s, \pi^*(s))\}$ of visited states by π_i and actions given by expert
    Aggregate datasets: $\mathcal{D} \leftarrow \mathcal{D} \cup \mathcal{D}_i$
    Train classifier $\hat{\pi}_{i+1}$ on $\mathcal{D}$
return best $\hat{\pi}_i$ on validation

Figure 3.2: DAGGER

The π_i (suboptimal policy) trick is designed to bootstrap more data, generalizing the learning outcome.

3.2 Deep Learning

3.2.1 Neural Networks

Non-linear classifiers: $f(\sum_i w_i x_i + b)$, the f is called the **activation function**. Each non-linear classifier (with n inputs and 1 output) is a **neuron**.

1. Sigmoid(Logistic): $\sigma(z) = \frac{1}{1+\exp(-z)}$;
2. Tanh: $\tanh(z) = \frac{\exp(z)-\exp(-z)}{\exp(z)+\exp(-z)}$;
3. ReLU: $\text{ReLU}(z) = \max(0, z)$.

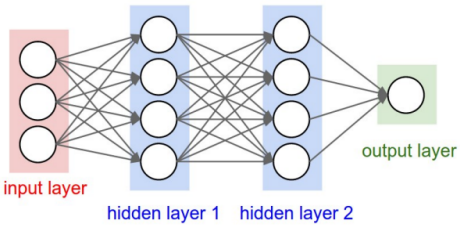
Remark 3.2.1

The properties of the tanh activation function:

- derivative: $\frac{d}{dx}\tanh(x) = 1 - \tanh^2(x)$;
- Output between -1 and 1, derivative output between 0 and 1 (larger than sigmoid, accelerate GD).
- For sigmoid: $\frac{d}{dx}\sigma(x) = \sigma(x)(1-\sigma(x))$, this function has an output range within $(0, \frac{1}{4}]$.

The simplest neural network is a multi-layer perceptron with one hidden layer. A neural network with at least one hidden layer is a universal approximator (can represent any continuous function). Shallow network can represent any function, but using deep structure is more effective: the hidden layer is a kind of modularization: and the modularization is automatically learned from data.

Forward Pass: Inference




```
# forward-pass of a 3-layer neural network:
f = lambda x: 1.0/(1.0 + np.exp(-x)) # activation function (use sigmoid)
x = np.random.randn(3, 1) # random input vector of three numbers (3x1)
h1 = f(np.dot(W1, x) + b1) # calculate first hidden layer activations (4x1)
h2 = f(np.dot(W2, h1) + b2) # calculate second hidden layer activations (4x1)
out = np.dot(W3, h2) + b3 # output neuron (1x1)
```

Figure 3.3: Example of Forward Pass in Python

W_1 is a 4×3 matrix and W_2 is a 4×4 matrix.

Back Propagation: Learning and Optimization

The first step is to set an appropriate loss function: $L = \mathcal{L}(X, Y, \theta)$, where X and Y are known from the samples and θ is the parameter to learn. Common loss functions include cross-entropy and MSE loss.

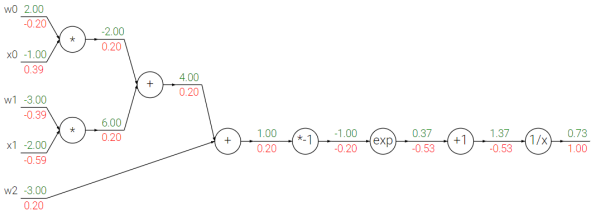


Figure 3.4: 2D Neuron Backpropagation

Initialization and Regularization

Consider a deep network with L layers, input $\mathbf{x}$ and output $\mathbf{o}$. With each layer l defined by a transformation f_l parametrized by weights $\mathbf{W}^{(l)}$, whose hidden layer output is $\mathbf{h}^{(l)}$ (let $\mathbf{h}^{(0)} = \mathbf{x}$), our network can be expressed as:

$$\mathbf{h}^{(l)} = f_l(\mathbf{h}^{(l-1)}) \text{ and thus } \mathbf{o} = f_L \circ \cdots \circ f_1(\mathbf{x}). \tag{3.2.1.1}$$

If all the hidden layer output and the input are vectors, we can write the gradient of $\mathbf{o}$ with respect to any set of parameters $\mathbf{W}^{(l)}$ as follows:

$$\partial_{\mathbf{W}^{(l)}} \mathbf{o} = \underbrace{\partial_{\mathbf{h}^{(L-1)}} \mathbf{h}^{(L)}}_{\mathbf{M}^{(L)} \stackrel{\text{def}}{=}} \cdots \underbrace{\partial_{\mathbf{h}^{(l)}} \mathbf{h}^{(l+1)}}_{\mathbf{M}^{(l+1)} \stackrel{\text{def}}{=}} \underbrace{\partial_{\mathbf{W}^{(l)}} \mathbf{h}^{(l)}}_{\mathbf{v}^{(l)} \stackrel{\text{def}}{=}}. \tag{3.2.1.2}$$

This may lead to **gradient vanishing** or **gradient exploding**.

Data Preprocessing

Setting all weights of an NN to the same constant can not work, because, for all neurons at the same layer, the gradient is the same, thus the update is the same, and it can not learn different features. For small networks, it is OK to set $W_0 = 0.01 * random.rand()$. Xavier initialization aims to make the variance across every layer is the same, preventing gradient explosion or diminishing:

$$w_{1,i} \overset{\text{iid}}{\sim} N\left(0, \frac{1}{d}\right)$$

where d is the number of inputs for each neuron. Another initialization note is to scale the features.
early stopping is a regularization technique, using a validation set to validate the error each time after the update when the validation error doesn't improve, return the stored weights. **Dropout** is the most popular regularization technique in multi-layer perceptron training. During the training process, it keeps a neuron active with probability $1 - p$, or sets it to zero otherwise.

$$h' = \begin{cases} 0 & \text{with probability } p \\ \frac{h}{1-p} & \text{otherwise} \end{cases} \tag{3.2.1.3}$$

The operation of dividing the unselected layers by $1 - p$ is to maintain the expectation of the output.

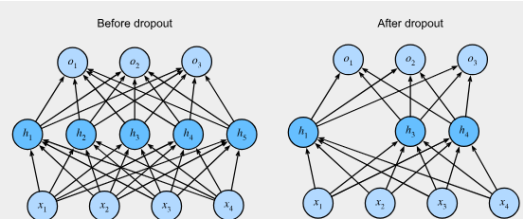


Figure 3.5: Dropout

3.2.2 Deep Learning for Computer Vision

CNN: Convolutional Neural Networks

There are three main types of layers to build ConvNet architectures: Convolutional Layer, Pooling Layer, and Fully-Connected Layer.

Convolutional Layer

The connections between kernels and inputs are local in 2D space (along width and height), but always full along the entire depth of the input volume.

> **Assumption 3.2.2.1:** Parameter Sharing
> If one feature is useful to compute at some spatial position (x, y), then it should also be useful to compute at a different position (x_2, y_2).

This assumption can reduce the number of weights and parameters dramatically. However, sometimes the parameter-sharing assumption may not make sense. This is especially the case when the input images to a ConvNet have some specific centered structure, one example is when the input are faces that have been centered in the image. We can use a **locally-connected layer** to solve this. **Padding**: sometimes, it is beneficial to pad around the edges with 0s:

- Prevent shrinking;
- Not losing edge information.

Given figure size $N \times N$ and filter size $F \times F$, padding P and stride S, the size of convolved layer is given by:

$$M = floor(\frac{N + 2P - F}{S}) + 1$$

A recent development is to introduce one more hyperparameter **dilation**, which are filters that have spaces between each cell. For example, dilation = 0 computes $w[0] * x[0] + w[1] * x[1] + w[2] * x[2]$, in contrast, dilation = 0 computes $w[0] * x[0] + w[1] * x[2] + w[2] * x[4]$.

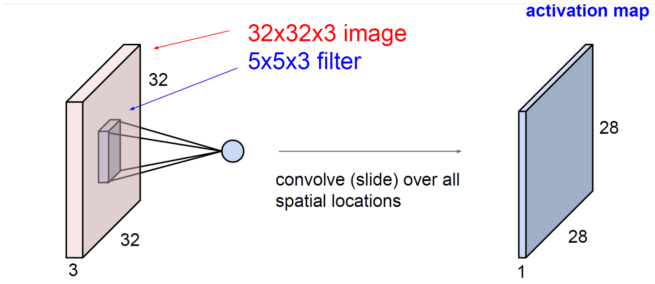


Figure 3.6: Multi-channel CNN

For the multi-channel input images, each convolutional kernel has the same depth as the input image. There can be multiple kernels in each layer:

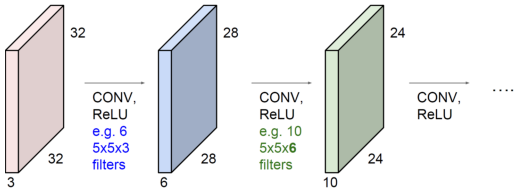


Figure 3.7: multiple kernels in each layer

Pooling Layer

Pooling layer can reduce the complexity and number of parameters for the whole network, and **subsampling** pixels will not change the object. The Pooling Layer operates independently on every depth slice of the input and resizes it spatially, using the MAX operation. The most common form is a pooling layer with filters of size (spatial extent) 2x2 applied with a stride of 2, the depth dimension remains unchanged.

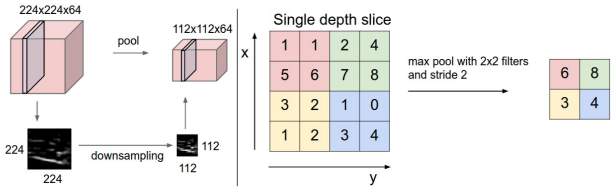


Figure 3.8: The pooling layer

General pooling: the pooling units can also perform other functions, such as average pooling. Some people don't like the pooling layer, they replace it by using a larger stride in the conv layer. Discarding pooling layers has also been found to be important in training good generative models, such as VAE and GAN.

3.2.3 Deep Learning for NLP

Language models compute the probability of occurrence of a number of words in a particular sequence. The probability of a sequence of words $\{w_1, \cdots, w_m\}$ is denoted as $P(w_1, ..., w_m)$. Since the location matters:

$$P(w_1, ..., w_m) = \prod_{i=1}^{i=m} P(w_i | w_1, ..., w_{i-1}) \approx \prod_{i=1}^{i=m} P(w_i | w_{i-n}, ..., w_{i-1}) \tag{3.2.3.1}$$

This model focuses on making predictions based on a fixed window of context. For example, bi-gram is calculated by:

$$p(w_2|w_1) = \frac{count(w_1, w_2)}{count(w_1)}$$

The trigram model calculates the probability using:

$$p(w_3|w_1, w_2) = \frac{count(w_1, w_2, w_3)}{count(w_1, w_2)}$$

However, this model suffers from the problem of *sparsity* and *exponential storage*.

RNN: Recurrent Neural Networks

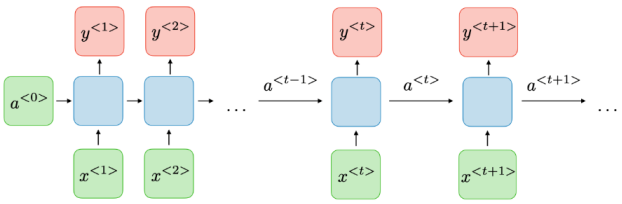


Figure 3.9: Architecture of a traditional RNN

RNNs are a class of neural networks that allow previous outputs to be used as inputs while having hidden states. For each timestep t, the activation $a^{<t>}$ and the output $y^{<t>}$ are expressed as follows:

$$a^{<t>} = g_1(W_{aa}a^{<t-1>} + W_{ax}x^{<t>} + b_a$$

$$y^{<t>} = g_2(W_{ya}a^{<t>} + b_y)$$

where g_1 and g_2 are activation functions and J are applied repeatedly at each timestep. The loss function $\mathcal{L}$ is defined as the sum of the losses at each time step. If we use the cross entropy loss:

$$J = \frac{1}{T}\sum_{t=1}^{T} J^{(t)}(\theta) = -\frac{1}{T}\sum_{t=1}^{T}\sum_{j=1}^{|V|} y_{t,j} \times log(\hat{y}_{t,j}) \tag{3.2.3.2}$$

Perplexity is defined as 2^J, where lower values imply more confidence in predicting the next word in the sequence. Advantages of RNNs:

- They can process input sequences of any length;
- The model size does not increase for longer input sequence lengths;

Disadvantages:

- Computation is slow - because it is sequential, it cannot be parallelized;
- Difficult to access information from many steps back due to problems like vanishing and exploding gradients (This can be alleviated by **clips** on the gradient).

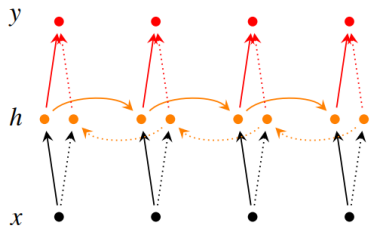


Figure 3.10: A bi-directional RNN model

This network consumes twice as much memory space for its weight and bias parameters.

$$\begin{aligned}\overrightarrow{h}_t &= f(\overrightarrow{W}x_t + \overrightarrow{V}\overrightarrow{h}_{t-1} + \overrightarrow{b})\\ \overleftarrow{h}_t &= f(\overleftarrow{W}x_t + \overleftarrow{V}\overleftarrow{h}_{t+1} + \overleftarrow{b})\\ \hat{y}_t &= g(Uh_t + c) = g(U[\overrightarrow{h}_t; \overleftarrow{h}_t] + c)\end{aligned} \tag{3.2.3.3}$$

GRU and LSTM

Gated recurrent units (GRU) are designed to have more persistent memory for capturing long-term dependencies. Denote h_t as the hidden state and x_t as the input, GRU uses h_{t-1} and x_t to generate h_t:

$$\begin{aligned}Z_t &= \sigma(W^{(z)}x_t + U^{(z)}h_{t-1})(\text{Update gate})\\ r_t &= \sigma(W^{(r)}x_t + U^{(r)}h_{t-1})(\text{Reset gate})\\ \tilde{h}_t &= \tanh(r_t \circ Uh_{t-1} + Wx_t)(\text{New memory})\\ h_t &= (1 - z_t) \circ \tilde{h}_t + z_t \circ h_{t-1}(\text{Hidden state})\end{aligned} \tag{3.2.3.4}$$

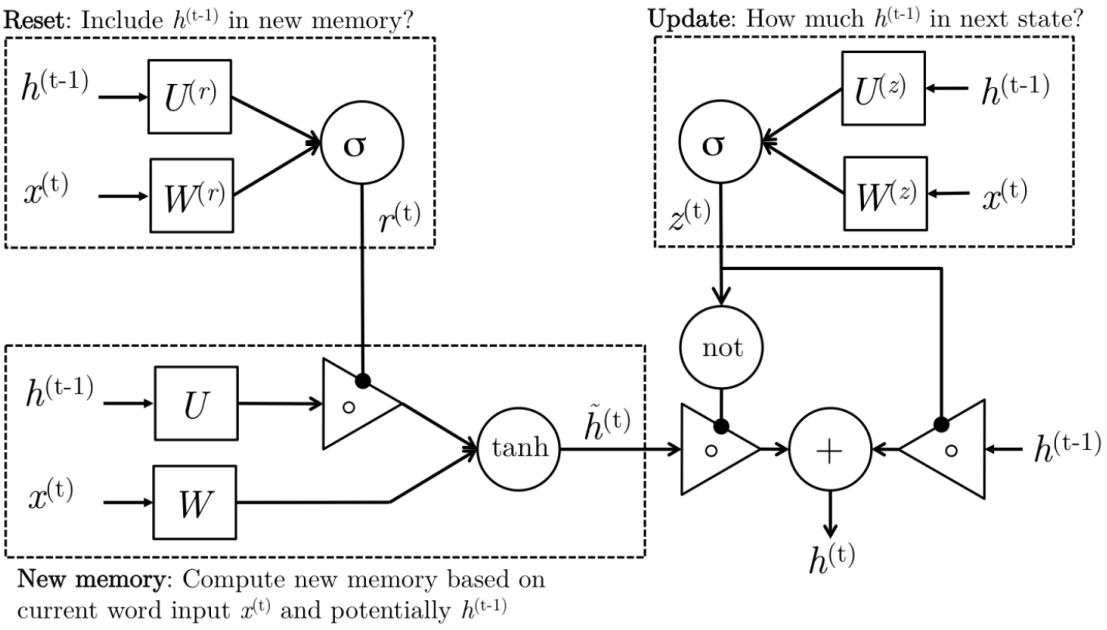


Figure 3.11: Gated Recurrent Units

Intuitive understandings:

1. **Reset Gate**: the signal r_t is responsible for determining the importance of h_t on $\tilde{h}_t$;
2. **New Memory**: $\tilde{h}_t$ is a the consolidation of a new input x_t with part of the hidden state h_{t-1} (determined by the reset gate);
3. **Update gate**: the update signal Z_t determins the proportion between h_{t-1} and $\tilde{h}_t$.

Long-Short-Term-Memories (LSTM) is another type of complex RNN to deal with vanishing gradients. On step t, there is a hidden state h_t and a **cell state** c_t which stores long-term information. The LSTM can erase, write and read information from the cell using *gates*. Each gate is still an n-dimensional vector.

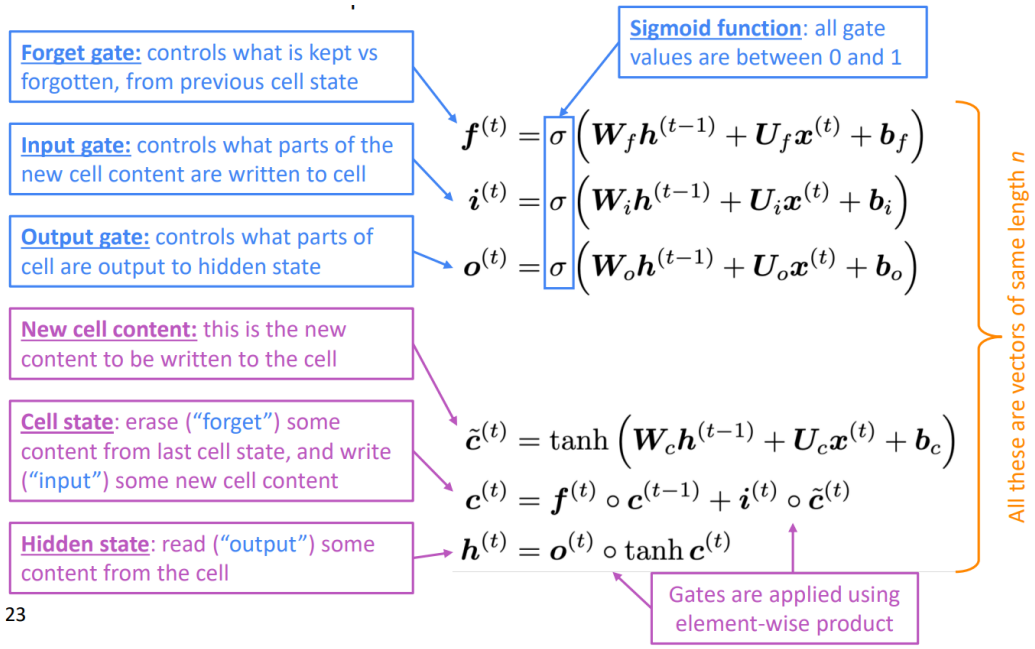


Figure 3.12: LSTM Gates

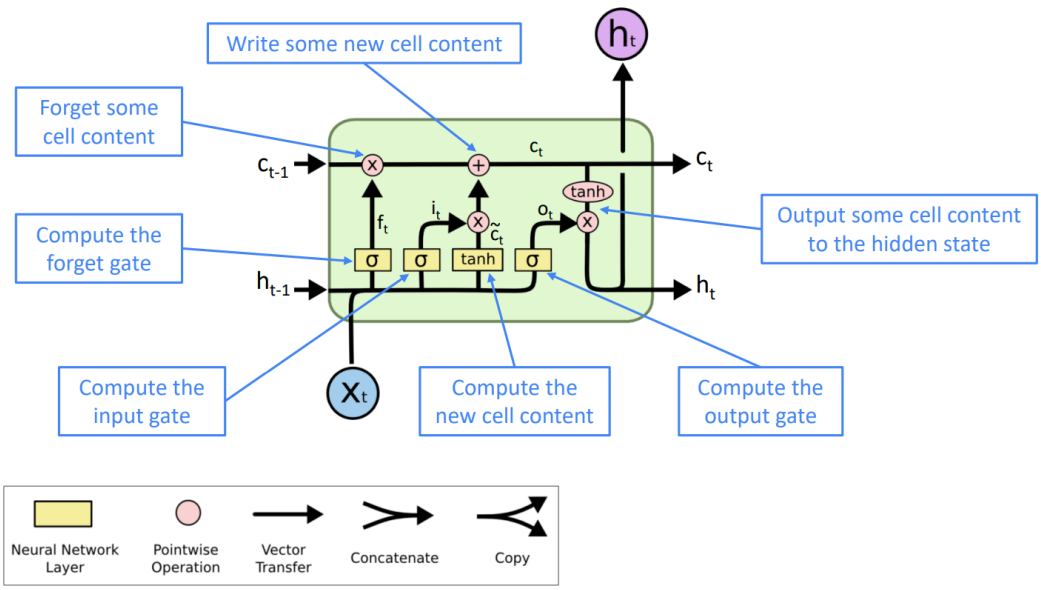


Figure 3.13: LSTM Architecture

The LSTM architecture makes it easier for the RNN to preserve information over many timesteps. e.g. the forget gate can set the cell to remember 'everything' in the past.

Remark 3.2.2

- LSTMs are specifically designed to remember information over long sequences;
- The forget gate in LSTMs allows the network to ignore irrelevant or noisy inputs, making them more robust in noisy environments;
- GRUs have fewer parameters than LSTMs, as they combine the input and forget gates into a single update gate.

Attention and Transformers

Sequence to Sequence

"Seq2Seq" is made up of two recurrent neural networks (trained at the same time): an encoder to transform the input as a *context* information, and a decoder to transform the context to the output. Encoders will process the input sequence in reverse: easier for the decoder to get started on the output. One problem with Seq2Seq is: that the last encoding neuron needs to capture all information: **Information bottleneck**! Besides, traditional RNN lacks parallelizability (stacking many LSTMs together would be slow).

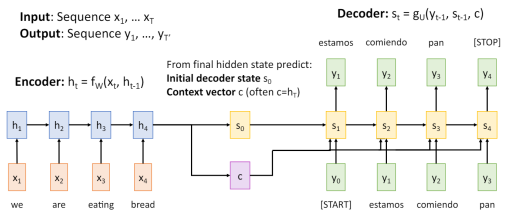


Figure 3.14: Sequence-to-Sequence with RNNs

The core idea of attention is:

on each step of the decoder, use direct connection to the encoder to focus on a particular part of the source sequence.

In the attention mechanism, the *alignment scores* are computed using $e_{t,i} = f_{att}(s_{t-1}, h_i)$, where s_{t-1} is the decoder state and f_{att} is an MLP. Then using Softmax to normalize the alignment scores and get the *attention weights* $a_{t,i}$, the hidden state in the decoder side is a linear combination of the attention score and the hidden states in the encoder side: $c_t = \sum_i a_{t,i} h_i$. The context vector at the next t is computed using $s_t = g_D(y_{t-1}, s_{t-1}, c_t)$. The intuition is: the context vector attends to the relevant part of the input sequence.

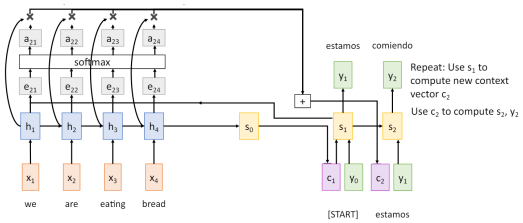


Figure 3.15: Sequence-to-Sequence with RNNs and Attentio

Attention Variants

- Basic dot-product attention: $e_i = s^T h_i$, this assumes $d_s = d_h$;

- Multiplicative attention: $e_i = s^T W h_i$;
- Reduced rank multiplicative attention: $e_i = s^T (U^T V) h_i, U \in \mathbb{R}^{k \times d_s}, V \in \mathbb{R}^{k \times d_h}$;
- Addictive attention: $e_i = v^T tanh(W_1 h_i + W_2 s)$.

Definition 3.2.3.1: Attention

Attention function can be described as mapping a query and a set of key-value pairs to an output (all vectors).

Assume we have some **queries** $q_1, \cdots, q_T$, each $q_i \in \mathbb{R}^d$; **keys** $k_1, \cdots, k_T$, each $k_i \in \mathbb{R}^d$; **values** $v_1, \cdots, v_T$, each $v_i \in \mathbb{R}^d$. *Self-attention* is those attentions where the query, key, and value are all coming from the same sources (thus the same dimension): let $v_i = k_i = q_i = x_i$. Self-attention problem:

1. *Sequence order*: use a **sequence index** $p_i \in \mathbb{R}^d$, for $i \in \{1, 2, \cdots, T\}$ positions. Then add p_i with the original vector. Now we have the position representations for the values;
2. *No nonlinearities*use the **feed-forward network** between different layers of self-attention;
3. *Don't look at future*: use the **masked attention** in the decoders: setting the attention score to the future words as $-\infty$.

Encoder: Key-Query-Value Self-attention

Let $X \in \mathbb{R}^{Txd}$ be the concatenation of input vectors, the output is given by:

$$output = softmax(XQ(XK)^T) \times XV \tag{3.2.3.5}$$

Decoder: Multi-head attention

If we want to attend different kinds of places. Let $Q_l, K_l, V_l \in \mathbb{R}^{d \times \frac{d}{h}}$, and l ranges from 1 to h. Each attention head performs attention independently. For the output, compute $output_l$ first, then use $Y[output_1, \cdots, output_h]$ to concatenate them. [Vaswani et al., 2023] proposed the transformer model.

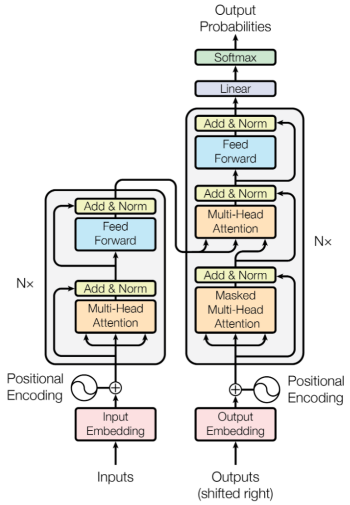


Figure 3.16: The Transformer

Tricks in the transformer model (accelerate the training process):

- Residual connections;
- Layer normalization;
- Scaled dot product attention (reduced rank multiplicative attention).

3.3 Causal Inference with Machine Learning

3.3.1 Foundations of Causal Inference

[Peters et al., 2017] provide an overview of causal inference from a fundamental science perspective. [Spirtes, 2010] summarize the development of causal inference and machine learning interface from a computer science perspective.
There are two main causal frameworks: **Structural Causal Model** and **Potential Outcome Framework**.

Directed Acyclic Graphs (DAG)

We can encode the relationship between D and Y using an *arrow* in a graph.

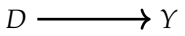


Figure 3.17: D has a causal effect on Y

When D and Y are both caused by some **confounder** U, the number of paths between D and Y has more than one:

1. The standard direct effect $D \to Y$;
2. The **back door** path $D \leftarrow U \to Y$.

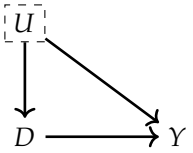


Figure 3.18: D's effect on Y is confounded by U

When U is a control variable X rather than a confunder, controlling X can block the backdoor path.

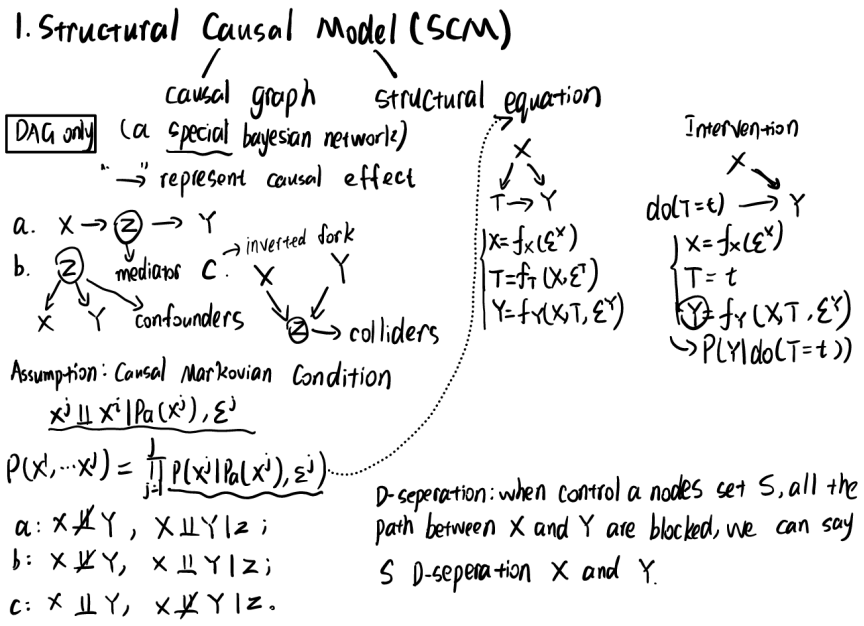


Figure 3.19: Structural Causal Model

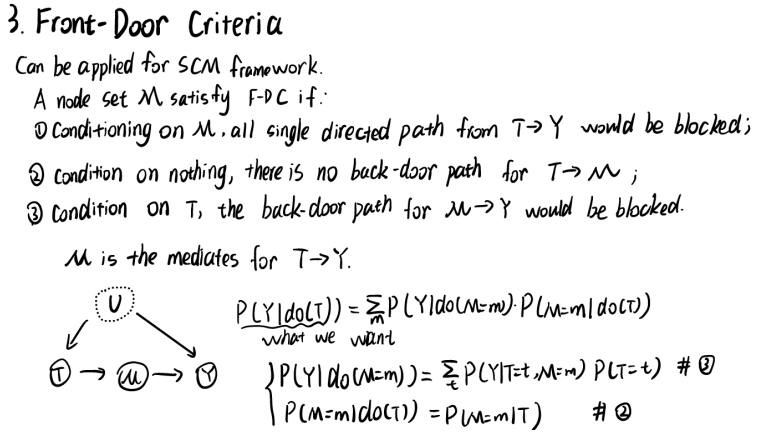


Figure 3.20: Front-Door Criteria

3.4 Reinforcement Learning & Dynamic Programming

This section is mainly the notes of [Thrun and Littman, 2000] and [Agarwal et al., 2019].

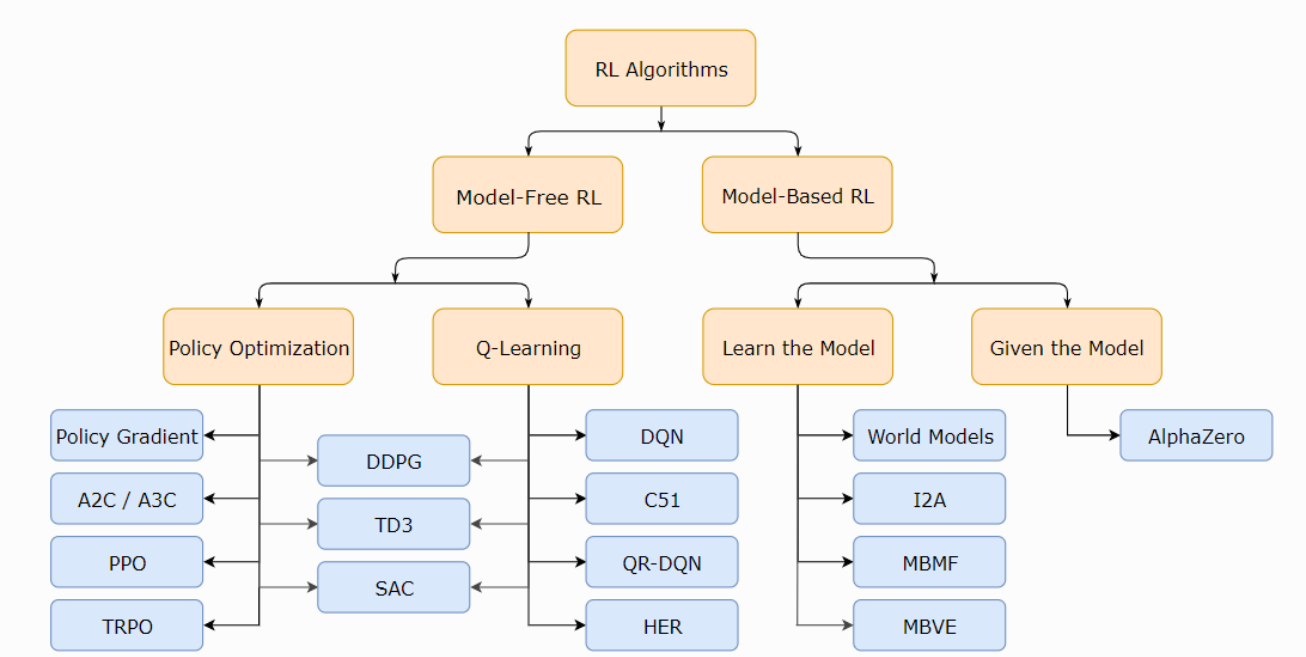


Figure 3.21: A Taxonomy of RL Algorithms

3.4.1 Introduction and MDP

Keywords 3.1

- Exploration, Exploitation, Reward, Policy, Value Function;
-

The four elements of reinforcement learning:

1. *Policy*: agent's way to interact with the environment;
2. *Reward Signal*: on each time step, the environment sends to the agent a single number;
3. *Value Function*: specify what is good in the long run, which is the discount value of the rewards;
4. *Model*: mimics behaviors of the environment;

Remark 3.4.1

"Deadly Trials"

1. The balance between **Exploration** and **Exploitation**;
2. Reinforcement learning is difficult to generalize;
3. Delayed consequences may cause RL algorithm to perform poorly.

MDP is a mathematical framework to model *discrete-time* sequential decision process, denoted by the tuple $(\mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R}, \rho_0, \gamma)$:

- $\mathcal{S}$: the state space, which is the states for the **entire** environment(MOBA games may can not directly be modeled by MDP);
- $\mathcal{A}$: the action space. $\mathcal{A}$ can depend on $s \in \mathcal{S}$;
- $\mathcal{T} : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$:the environment transition probability function;
- $\mathcal{R} : \mathcal{S} \times \mathcal{A} \to \Delta(\mathbb{R})$: the reward function;
- $\rho_0 \in \Delta(\mathcal{S})$: the initial state distribution;
- $\gamma \in [0, 1]$ is the discount factor.

The goal is to optimize:

$$\mathbb{E}_{s_t, a_t, r_t, t \geq 0}[R_0] = \mathbb{E}_{s_t, a_t, r_t, t \geq 0}\Big[\sum_{t=0}^{\infty} \gamma^t r_t\Big] \tag{3.4.1.1}$$

Remark 3.4.2

- The $\Delta(\cdot)$ may not be deterministic, but some random distribution;
- Among the above tuple, $\mathcal{S}, \mathcal{T}, \mathcal{R}, \rho_0$ can not be modified by the agent, to train a good policy, $\mathcal{A}, \gamma$ is the key;
- RL is more like infants rather than adults;
- The reward function is the way of communicating with the agent *what* to do, not *how* to do;
- The *trajectory* of the MDP sequence: $S_0, A_0, R_1, S_1, A_1, \cdots$.

Figure 3.22: The agent-environment interaction in a Markov decision process

Theorem 3.4.1.1: Dynamics of MDP

$$p(s', r|s, a) \doteq \Pr\{S_t = s', R_t = r \mid S_{t-1} = s, A_{t-1} = a\} \tag{3.4.1.2}$$

$$\sum_{s' \in \mathcal{S}} \sum_{r \in \mathcal{R}} p(s', r|s, a) = 1, \text{ for all } s \in \mathcal{S}, a \in \mathcal{A}(s) \tag{3.4.1.3}$$

Corollary 3.4.1.1: Some formula derived from the dynamics theorem

$$p(s'|s, a) \doteq \Pr\{S_t = s' \mid S_{t-1} = s, A_{t-1} = a\} = \sum_{r \in \mathbb{R}} p(s', r|s, a) \tag{3.4.1.4}$$

$$r(s, a) \doteq \mathbb{E}[R_t \mid S_{t-1} = s, A_{t-1} = a] = \sum_{r \in \mathbb{R}} r \sum_{s' \in \mathcal{S}} p(s', r|s, a) \tag{3.4.1.5}$$

$$r(s, a, s') \doteq \mathbb{E}[R_t \mid S_{t-1} = s, A_{t-1} = a, S_t = s'] = \sum_{r \in \mathbb{R}} r \frac{p(s', r|s, a)}{p(s'|s, a)} \tag{3.4.1.6}$$

Definition 3.4.1.1: Some useful function:

The act value function given policy π:

$$Q^{\pi}(s,a) = \mathbb{E}_{s_t,a_t,r_t,t\geq 0}\Big[\sum_{t=0}^{\infty}\gamma^t r(s_t,a_t) \mid s_0 = s, a_0 = a\Big] \tag{3.4.1.7}$$

The expected return at state s given policy π:

$$V^{\pi}(s) = \mathbb{E}_{a\sim\pi(s)}\left[Q^{\pi}(s,a)\right] \tag{3.4.1.8}$$

The **advantage function**:

$$A^{\pi}(s,a) = Q^{\pi}(s,a) - V^{\pi}(s) \tag{3.4.1.9}$$

The temporal-difference error:

$$\delta_t = r_t + \gamma V(s_{t+1}) - V(s_t) \tag{3.4.1.10}$$

If we denote $G_t = R_{t+1} + \gamma G_{t+1}$, then we have:

Theorem 3.4.1.2: Bellman Equation

$$\begin{aligned} v_{\pi}(s) &\doteq \mathbb{E}_{\pi}[G_t \mid S_t = s] \\ &= \mathbb{E}_{\pi}[R_{t+1} + \gamma G_{t+1} \mid S_t = s] \\ &= \sum_a \pi(a|s)\sum_{s'}\sum_r p(s',r|s,a)\Big[r + \gamma\mathbb{E}_{\pi}[G_{t+1}|S_{t+1} = s']\Big] \\ &= \sum \pi(a|s)\sum p(s',r|s,a)\Big[r + \gamma v_{\pi}(s')\Big], \quad \text{for all } s \in \mathcal{S}, \end{aligned} \tag{3.4.1.11}$$

If we consider a n-state game, assume the reward $r \in \mathbb{R}^n$ is deterministic. Then we denote P as the transition matrix for a corresponding strategy and $V \in \mathbb{R}^n$ as the value function vector. Finally, we can get the Bellman equation in matrix form: eq V=r+γPV.

A policy π is better π' if and only if $v_{\pi}(s) \leq v_{\pi'}(S)$for all $s \in \mathcal{S}$. There is always at list one *optimal policy* denoted by π_* (doesn't hold for partially observed MDP). They share the same state-value function, called the *optimal state-value function*: $v_*(s) \doteq \max_{\pi} v_{\pi}(s)$. The optimal policy also shares the same *optimal action-value function*: $q_*(s,a) \doteq \max_{\pi} q_{\pi}(s,a) = \mathbb{E}[R_{t+1} + \gamma v_*(S_{t+1}) \mid S_t = s, A_t = a]$. Then we can get the **Bellman optimality function** in an action-value function sense:

Theorem 3.4.1.3: Bellman optimality function

$$\begin{aligned} q_*(s,a) &= \mathbb{E}\Big[R_{t+1} + \gamma \max_{a'} q_*(S_{t+1},a')\Big|S_t = s, A_t = a\Big] \\ &= \sum_{s'.r} p(s',r|s,a)\Big[r + \gamma \max_{a'} q_*(s',a')\Big]. \end{aligned} \tag{3.4.1.12}$$

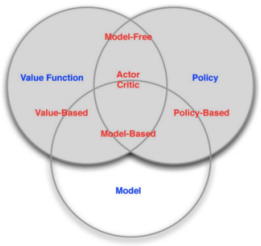


Figure 3.23: Classification of different reinforcement learning agents

Definition 3.4.1.2: The optimal policy π^*

A policy π^* is an optimal policy if for every policy π and every $s \in \mathcal{S}$, we have:

$$V^{\pi^*}(s) \geq V^{\pi}(s) \tag{3.4.1.13}$$

Remark 3.4.3

- If we have the full information of the game (MDP framework), then the optimal policy is always deterministic;
- The optimal policy is always stochastic when there are **minimax** structure (e.g. protection information);
- Whether the optimal policy is stochastic or deterministic has nothing to do with the stochasticity of the game.

3.4.2 Bandit Algorithms & Online Learning

See notes of concentration inequalities at 1.4.5. Difference between online learning and optimization problem: in online learning you don't konw the objective function, which needs to be learn over time.

Bayesian (Rested) Bandit

Assumption 3.4.2.1: The Markovian MAB

- We have k arms, each has reward Bernoulli p_i with unknown p_i;
- However, p_i is drawn from a known prior distribution $Beta(\alpha, \beta)$:

$$f(x|\alpha,\beta) = \frac{x^{\alpha-1}(1-x)^{\beta-1}}{B(\alpha,\beta)}, B(\alpha,\beta) = \int_0^1 t^{\alpha-1}(1-t)^{\beta-1}dt$$

  Note that the Beta distribution is a conjugate prior for the Bernoulli.

- For arm i, if observe m success out of n trials, we can update the posterior to $Beta(\alpha+n, \beta+n-m)$;
- Define $X_i(n) = (\alpha+m, \beta+m-n)$ as the state space. From a Bayesian view, the transition is given by

$$\mathsf{P}(X_i(n+1) = X_i(n) + (1,0)|X_i(n)) = \frac{\alpha+m}{\alpha+\beta+n}$$

- We can use $X(n)$ denote the state of n arms, the state space has $2k$ dimensions;
- The Marvian MAB is also called Rested Bandit, because once you pull arm i, you will get a reward and update $R(X_i(N_i(t)))$, where $N_i(t)$ is the number pulling arm i by time $t-1$. $N_i(t+1) = N_i(t)+1$ if $A_t = i$ otherwise $N_i(t+1) = N_i(t)$.

The policy is given by $A_t = \pi_t(\mathcal{F}_t)$ and the cummulative reward is given by:

$$J^{\pi} = \mathbb{E}\left[\sum_{t=0}^{\infty} \beta^t R(X_{A_t}(N_{A_t}(t)))\right] \tag{3.4.2.1}$$

Now we can transfrom the Bayesian bandit game to an infinite-horizon MDP, [Frostig and Weiss, 2016] prove that the optimal policy has a very simple closed-form and intuitive structure. Consider a policy on arm i given $X_i(0)$ at $t = 0$:

1. select arm i and continue this policy;
2. retire from the game and receive v every time in the future.

We can choose v so that at $t = 0$ these two options are no different:

$$v_i(X_i(0)) = \sup\left\{v : \sum_{t=0}^{\infty} \beta^t v \leq \sup_{\tau \geq 1} \mathbb{E}\left[\sum_{t=0}^{\tau-1} \beta^t R_i(X_i(t)) + \sum_{t=\tau}^{\infty} \beta^t v | X_i(0)\right]\right\} \tag{3.4.2.2}$$

Alternatively, we can rewrite $v_i(X_i(0))$ to:

Definition 3.4.2.1: Gittins Index

$$v_i(X_i(0)) = \max_{\tau > 0} \frac{\mathbb{E}\left[\sum_{t=0}^{\tau-1} \beta^t R_i(X_i(t)) | X_i(0)\right]}{\mathbb{E}\left[\sum_{t=0}^{\tau-1} \beta^t | X_i(0)\right]} \tag{3.4.2.3}$$

The optimal policy is to pull arms with the highest Gittins index.

Stochastic Bandit

The regret for stochastic bandit games is defined as :

$$\overline{R}_t = \sum_{i=1}^{m} \mathbb{E}[N_{t,i}]\Delta_i \tag{3.4.2.4}$$

Where $N_{t,i} = \sum_{t'=0}^{t} \mathbb{1}\{a_{t'} = i\}$ and $\Delta_i = \mu^* - \mu_i$.

Greedy Algorithms

Algorithm 1: The greedy algorithm

Output: $\pi(t), t \in \{0, 1, \ldots, T\}$

while $0 \leq t \leq m-1$ **do**

$$\pi(t) = t + 1$$

while $m \leq t \leq T$ **do**

$$\pi(t) = \arg\max_{i \in [m]} \left\{ \frac{1}{N_{t-1,i}} \sum_{t'=0}^{t-1} r_{t'} \mathbb{1}\{a_{t'} = i\} \right\}$$

Figure 3.24: The greedy algorithm

This algorithm achieves a regret at most $O(T)$.

The ϵ-greedy algorithm

Algorithm 2: The ε-greedy algorithm

Input: $\varepsilon_t, t \in \{0, 1, \ldots, T\}$ the exploration parameters

Output: $\pi(t), t \in \{0, 1, \ldots, T\}$

while $0 \leq t \leq m-1$ **do**

$$\pi(t) = t + 1$$

while $m \leq t \leq T$ **do**

$$\pi(t) \sim \begin{cases} \arg\max_{i \in [m]} \left\{ \frac{1}{N_{t-1,i}} \sum_{t'=0}^{t-1} r_{t'} \mathbb{1}\{a_{t'} = i\} \right\} & \text{with probability } 1 - \varepsilon_t \\ i \quad \text{with probability } \varepsilon_t / m, \text{ for each } i \in [m] \end{cases}$$

Figure 3.25: The epsilon greedy algorithm

The lower bound of ϵ greedy: $\overline{R}_t \geq \frac{1}{m}(\Delta_2 + \cdots + \Delta_m)\varepsilon(T - m)$, where $\epsilon \leq \epsilon_t$ for all t;

The explore-then-commit algorithm (ETC)

Algorithm 1: The explore-then-commit algorithm

Input: k: number of exploration pulls on each arm

Output: $\pi(t), t \in \{0, 1, \ldots, T\}$

while $0 \leq t \leq km - 1$ **do**

$$a_t = (t \bmod m) + 1$$

while $km \leq t \leq T - 1$ **do**

$$a_t = \arg\max_{i \in [m]} \frac{1}{k} \sum_{t'=0}^{mk-1} r_{t'} \mathbb{1}\{a_{t'} = i\}$$

Figure 3.26: The ETC algorithm

This algorithm has an upper bound of $O(\Delta^2 log T)$, or $O(T^{\frac{2}{3}})$.

The UCB algorithm

Algorithm 1: The UCB algorithm.

Input: δ: confidence level

Output: $a_t, t \in \{0, 1, \ldots, T\}$

while $t \leq T - 1$ **do**

$$a_t = \arg\max_{i \in [m]} \mathrm{UCB}_i(t-1, \delta),$$

where ties break arbitrarily and for $i \in [m]$,

$$\mathrm{UCB}_i(t-1, \delta) = \begin{cases} \infty, & N_{i,t-1} = 0, \\ \frac{1}{N_{i,t-1}} \sum_{t' \leq t-1} r_{t'} \mathbb{1}\{a_{t'} = i\} + \sqrt{\frac{2\log(1/\delta)}{N_{i,t-1}}}, & N_{i,t-1} > 0; \end{cases}$$

Figure 3.27: UCB Algorithm

The UCB is based on the principle of optimism in the face of uncertainty, which states that:

One should act as if the environment is as nice as plausibly possible.

The UCB estimate for each arm is an over-estimate compared with the empirical mean $\hat{\mu}_{i,t-1} = \frac{1}{N_{i,t-1}} \sum_{t' \leq t-1} r_{t'} \mathbb{1}\{a_{t'} = i\}$, recall the Chernoff-Hoeffding bound:

$$\mathbb{P}(\overline{X} - \mathbb{E}[\overline{X}] \leq z) \geq 1 - \exp(-nz^2/2). \tag{3.4.2.5}$$

Repalcing z to $\sqrt{\frac{2\log(1/\delta)}{N_{i,t-1}}}$, we would get:

$$\mathbb{P}(\mu_i \geq \hat{\mu}_{i,t-1} + \sqrt{\frac{2\log(1/\delta)}{N_{i,t-1}}}) \leq \delta. \tag{3.4.2.6}$$

Thompson Sampling Algorithm

Assume that the reward distributions of different arms belong to the same family with respective parameters:

$$r(i) \sim p(x \mid \theta_i). \tag{3.4.2.7}$$

When estimating θ, the posterior is:

$$p(\theta \mid x) = \frac{p(x \mid \theta)p(\theta)}{\int_{\theta'} p(x \mid \theta')p(\theta')d\theta'}. \tag{3.4.2.8}$$

If the posterior $p(\theta|x)$ is in the same probability distribution family as the prior $p(\theta)$, then they are called conjugate distributions. The Bernoulli-Beta is important for Bernoulli Bandits, for $\theta = \{\alpha, \beta\}$:

$$p(\theta) = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} x^{\alpha-1}(1-x)^{\beta-1}, \tag{3.4.2.9}$$

where $\Gamma(z) = \int_0^\infty x^{z-1}\exp(-x)dx$ is the Gamma function. So when $p(\theta) \sim \text{Beta}(\alpha_0, \beta_0)$ and observing $x_1, \ldots, x_{\alpha'+\beta'} \sim x$ i.i.d., with α' ones and β' zeros.

$$\begin{aligned} p(\theta \mid x_1, \ldots, x_{\alpha'+\beta'}) &= \frac{p(x_1, \ldots, x_{\alpha'+\beta'} \mid \theta)p(\theta)}{\int_{\theta'} p(x_1, \ldots, x_{\alpha'+\beta'} \mid \theta')p(\theta')d\theta'} \\ &= \frac{\binom{\alpha'+\beta'}{\alpha'}\theta^{\alpha'}(1-\theta)^{\beta'}\frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)}\theta^{\alpha_0-1}(1-\theta)^{\beta_0-1}}{\int_{\theta'} p(x_1, \ldots, x_{\alpha'+\beta'} \mid \theta')p(\theta')d\theta'} \\ &= \frac{\binom{\alpha'+\beta'}{\alpha'}\frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)}}{\int_{\theta'} p(x_1, \ldots, x_{\alpha'+\beta'} \mid \theta')p(\theta')d\theta'}\theta^{\alpha_0+\alpha'-1}(1-\theta)^{\beta_0+\alpha'-1} \\ &\sim \text{Beta}(\alpha_0+\alpha', \beta_0+\beta'). \end{aligned} \tag{3.4.2.10}$$

Algorithm 1: Thompson sampling (Bernoulli bandits)

Input: Prior α_0, β_0
Output: $a_t, t \in [T]$
Initialize $\alpha_i = \alpha_0$, $\beta_i = \beta_0$, for $i \in [m]$
while $t \leq T-1$ **do**
  Sample $\theta_i(t) \sim \text{Beta}(\alpha_i, \beta_i)$ independently for $i \in [m]$
  $a_t = \arg\max_{i\in[m]} \theta_i(t)$ with arbitrary tiebreaker
  If $r_t = 1$ then $\alpha_{a_t} \mathrel{+}= 1$; If $r_t = 0$ then $\beta_{a_t} \mathrel{+}= 1$;

Figure 3.28: Thompson sampling for Bernoulli bandits

TS algorithm can work on any conjugate distributions.

Algorithm 2: Thompson sampling

Input: Prior θ_0
Output: $a_t, t \in [T]$
Initialize $\theta_i = \theta_0$, for $i \in [m]$
while $t \leq T-1$ **do**
  Sample independently for $i \in [m]$, $\theta_i(t) \sim p(\theta \mid \{r_{t'}\}_{1\{a_{t'}=i, t'\leq t-1\}})$
  $a_t = \arg\max_{i\in[m]} \theta_i(t)$ with arbitrary tiebreaker
  Update the posterior probability distribution of $\theta_{a_t}(t+1)$ by

$$p(\theta_{a_t}(t+1) \mid \{r_{t'}\}_{1\{a_{t'}=i\}}) = \frac{p(\{r_{t'}\}_{1\{a_{t'}=i\}} \mid \theta)p(\theta)}{\int_{\theta'} p(\{r_{t'}\}_{1\{a_{t'}=i\}} \mid \theta')p(\theta')d\theta'}$$

Figure 3.29: General Thompson Sampling

Analysis of Bandit Algorithms

ETC Algorithm

Theorem 3.4.2.2: ETC Regret

Assume $r(i)$ is *1-sub-Gaussian* for each i, the regret satisfies:

$$\overline{R}_T \leq k \sum_{i\in[m]} \Delta_i + (T - mk) \sum_{i\in[m]} \Delta_i \exp\left(-\frac{k\Delta_i^2}{4}\right). \tag{3.4.2.11}$$

When $m = 2$, taking $k = \lceil \max\left\{1, 4\Delta_2^{-2}\log(T\Delta_2^2/4)\right\}\rceil$ yields:

$$k = \lceil \max\left\{1, 4\Delta_2^{-2}\log(T\Delta_2^2/4)\right\}\rceil \tag{3.4.2.12}$$

Proof 3.4.2

$$\begin{aligned}
\mathbb{E}[N_{T,i}] &= k + (T - mk)\mathbb{P}\big(i = \arg\max_{i'} \hat{\mu}_{mk-1,i'}\big) \\
&\leq k + (T - mk)\mathbb{P}(\hat{\mu}_{mk-1,i} \geq \hat{\mu}_{mk-1,1}) \\
&= k + (T - mk)\mathbb{P}\big(\hat{\mu}_{mk-1,i} - \mu_i - \big(\hat{\mu}_{mk-1,1} - \mu_1\big) \geq \Delta_i\big).
\end{aligned} \tag{3.4.2.13}$$

By the sub-Gaussian tail bound:

$$\mathbb{P}\left(\hat{\mu}_{mk-1,i} - \mu_i - (\hat{\mu}_{mk-1,1} - \mu_1) \geq \Delta_i\right) \leq \exp\left(-\frac{k\Delta_i^2}{4}\right). \tag{3.4.2.14}$$

Therefore,

$$\begin{aligned}
\overline{R}_T &= \sum_{i=1}^{m} \mathbb{E}[N_{T,i}]\,\Delta_i \\
&\leq \sum_{i=1}^{m} \Delta_i \left(k + (T - mk)\,\mathbb{P}\big(\hat{\mu}_{mk-1,i} - \mu_i - (\hat{\mu}_{mk-1,1} - \mu_1) \geq \Delta_i\big)\right) \\
&\leq \sum_{i=1}^{m} \Delta_i \left(k + (T - mk)\exp\left(-\frac{k\Delta_i^2}{4}\right)\right).
\end{aligned} \tag{3.4.2.15}$$

By setting $m = 2$, we can derive:

$$\begin{aligned}
\overline{R}_T &\leq \Delta_2 \left(k + (T - mk)\exp\left(-\frac{k\Delta_2^2}{4}\right)\right) \\
&\leq \Delta_2 \left(k + T\exp\left(-\frac{k\Delta_2^2}{4}\right)\right).
\end{aligned} \tag{3.4.2.16}$$

To minimize the regret, take a derivative on k and use the first-order condition. When $k = \lceil \max\left\{1, 4\Delta_2^{-2}\log(T\Delta_2^2/4)\right\}\rceil$:

$$\begin{aligned}
\overline{R}_T &\leq \Delta_2\left(k + T\exp\left(-\frac{k\Delta_2^2}{4}\right)\right) \\
&\leq \Delta_2\left(k_0 + 1 + T\exp\left(-\frac{k_0\Delta_2^2}{4}\right)\right) \\
&\leq \Delta_2\left(\frac{4}{\Delta_2^2}\log\left(\frac{T\Delta_2^2}{4}\right) + 1 + T\exp\left(-\frac{\Delta_2^2}{4}\cdot\frac{4}{\Delta_2^2}\cdot\log\left(\frac{T\Delta_2^2}{4}\right)\right)\right) \\
&\leq \Delta_2\left(\frac{4}{\Delta_2^2}\log\left(\frac{T\Delta_2^2}{4}\right) + 1 + T\cdot\frac{4}{T\Delta_2^2}\right) \\
&\leq \Delta_2 + \frac{4}{\Delta_2} + \frac{4}{\Delta_2}\log\left(\frac{T\Delta_2^2}{4}\right).
\end{aligned} \tag{3.4.2.17}$$

Q.E.D.

Recap of Different Algorithms

Remark 3.4.4

For ϵ-greedy, by choosing $\varepsilon_t = \min\{1, Ct^{-1}\Delta_{\min}^{-2}m\}$:

$$\overline{R}_T \leq C'\sum_{i\geq 2}\left(\Delta_i + \frac{\Delta_i}{\Delta_{\min}^2}\log\max\left\{e, \frac{T\Delta_{\min}^2}{m}\right\}\right) \tag{3.4.2.18}$$

For ETC under 2-armed bandits, when $T \geq 4\sqrt{2\pi e}/\Delta^2$, by choosing $k = \lceil \frac{2}{\Delta^2}W(\frac{T^2\Delta^4}{32\pi})\rceil$,

$$\overline{R}_T \leq O(\frac{1}{\Delta}\log T\Delta^2) + o(\log T) + \Delta \tag{3.4.2.19}$$

For the two algorithms listed above, to achieve the optimal regret. the information about Δ is necessary. UCB and TS don't require knowing Δ.
For UCB algorithm, by setting $\delta = T^{-2}$,

$$\overline{R}_T \leq 3\sum_{i=1}^{m}\Delta_i + \sum_{i:\Delta_i>0}\frac{16\log T}{\Delta_i}. \tag{3.4.2.20}$$

For TS (Bernoulli bandits):

$$\overline{R}_T \leq \sum_{i:\Delta_i>0}\frac{\mu_1 - \mu_i}{d_{\mathrm{KL}}(\mu_1 \parallel \mu_i)}\log T + o(\log T). \tag{3.4.2.21}$$

3.4.3 Model-Based RL

There are two streams of reinforcement learning algorithms, *model-based* v.s. *model-free* methods. Model-based algorithms require the knowledge on the environment(transition matrix P and reward function r), or need to estimate P, r during the computations. Model-free algorithms don't rely the estimation of the environment. **Policy Evaluation (Prediction)**

$$
\begin{aligned}
v_{k+1}(s) &\doteq \mathbb{E}_\pi[R_{t+1} + \gamma v_k(S_{t+1}) \mid S_t = s] \\
&= \sum_a \pi(a|s) \sum_{s',r} p(s', r|s, a)\Big[r + \gamma v_k(s')\Big]
\end{aligned} \tag{3.4.3.1}
$$

Iterative Policy Evaluation, for estimating $V \approx v_\pi$

Input π, the policy to be evaluated
Algorithm parameter: a small threshold $\theta > 0$ determining accuracy of estimation
Initialize $V(s)$ arbitrarily, for $s \in \mathcal{S}$, and $V(terminal)$ to 0

Loop:
  $\Delta \leftarrow 0$
  Loop for each $s \in \mathcal{S}$:
    $v \leftarrow V(s)$
    $V(s) \leftarrow \sum_a \pi(a|s) \sum_{s',r} p(s', r|s, a)\big[r + \gamma V(s')\big]$
    $\Delta \leftarrow \max(\Delta, |v - V(s)|)$
until $\Delta < \theta$

Figure 3.30: Iterative Policy Evaluation

The policy evaluation would terminate after finite steps. For any $\epsilon \in (0, \frac{1}{1-\gamma})$, after at least $N \geq \frac{\gamma}{(1-\gamma)^4} \frac{n^2 m \log(cnm/\delta)}{\varepsilon^2}$ steps, with at least probability $1 - \delta$:

$$
\|P(\cdot \mid s, a) - \hat{P}(\cdot \mid s, a)\|_1 \leq (1 - \gamma)^2 \varepsilon. \tag{3.4.3.2}
$$

Theorem 3.4.3.1: Policy Improvement Theorem

For all $s \in \mathcal{S}$, if:

$$
q_\pi(s, \pi'(s)) \geq v_\pi(s) \tag{3.4.3.3}
$$

Then the policy π' must be better than policy π.

Policy Improvement

$$
\begin{aligned}
\pi'(s) \doteq & \arg\max_a q_\pi(s, a) \\
= & \arg\max_a \mathbb{E}[R_{t+1} + \gamma v_\pi(S_{t+1}) \mid S_t = s, A_t = a] \\
= & \arg\max_a \sum_{s',r} p(s', r|s, a)\Big[r + \gamma v_\pi(s')\Big],
\end{aligned} \tag{3.4.3.4}
$$

Policy Iteration and Value Iteration

Policy Iteration (using iterative policy evaluation) for estimating $\pi \approx \pi_*$

1. Initialization
   $V(s) \in \mathbb{R}$ and $\pi(s) \in \mathcal{A}(s)$ arbitrarily for all $s \in \mathcal{S}$; $V(terminal) \doteq 0$

2. Policy Evaluation
   Loop:
      $\Delta \leftarrow 0$
      Loop for each $s \in \mathcal{S}$:
         $v \leftarrow V(s)$
         $V(s) \leftarrow \sum_{s',r} p(s', r \mid s, \pi(s)) \big[r + \gamma V(s')\big]$
         $\Delta \leftarrow \max(\Delta, |v - V(s)|)$
   until $\Delta < \theta$ (a small positive number determining the accuracy of estimation)

3. Policy Improvement
   policy-stable $\leftarrow$ *true*
   For each $s \in \mathcal{S}$:
      old-action $\leftarrow \pi(s)$
      $\pi(s) \leftarrow \operatorname{argmax}_a \sum_{s',r} p(s', r \mid s, a) \big[r + \gamma V(s')\big]$
      If *old-action* $\neq \pi(s)$, then *policy-stable* $\leftarrow$ *false*
   If *policy-stable*, then stop and return $V \approx v_*$ and $\pi \approx \pi_*$; else go to 2

Figure 3.31: Policy Iteration

Value Iteration, for estimating $\pi \approx \pi_*$

Algorithm parameter: a small threshold $\theta > 0$ determining accuracy of estimation
Initialize $V(s)$, for all $s \in \mathcal{S}^+$, arbitrarily except that $V(terminal) = 0$

Loop:
| $\Delta \leftarrow 0$
| Loop for each $s \in \mathcal{S}$:
| $v \leftarrow V(s)$
| $V(s) \leftarrow \max_a \sum_{s',r} p(s', r \mid s, a) \big[r + \gamma V(s')\big]$
| $\Delta \leftarrow \max(\Delta, |v - V(s)|)$
until $\Delta < \theta$

Output a deterministic policy, $\pi \approx \pi_*$, such that
$\pi(s) = \operatorname{argmax}_a \sum_{s',r} p(s', r \mid s, a) \big[r + \gamma V(s')\big]$

Figure 3.32: Value Iteration

Differences between VI and PI:

Remark 3.4.5

- In each sweep, VI only updates one-step evaluation and one-step improvement, PI updates multiple-step evaluation and one-step improvement;
- PI takes fewer rounds but takes more time within each round.

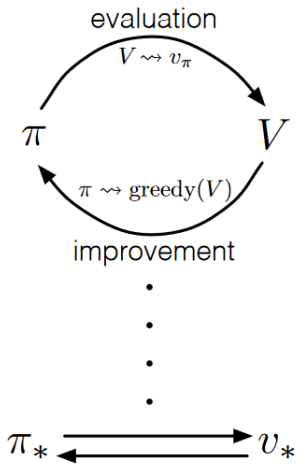


Figure 3.33: Generalized Policy Iteration

Episodic Discrete MDP

In the episodic setting, in every episode, the learner acts for H steps, starting from a fixed starting state $s_0 \sim_0$, and the process repeats for K episodes. The agent's goal is to minimize the expected regret:

$$\overline{R}_K = \mathbb{E}\left[KV^*(s_0) - \sum_{k=0}^{K-1}\sum_{h=0}^{H-1} r(s_h^k, a_h^k)\right]. \tag{3.4.3.5}$$

UCVI: Upper Confidence bound Value Iteration

In MDP, the exploration is more difficult compared to bandit games, because we can only decide the action rather than navigate to a particular state which is barely reached before. Using the ideas in UCB from bandit games, we can give an overestimation of value functions for those states that were barely reached before. By value iteration and policy iteration, this overestimation can navigate the agent to explore the strange states.

Algorithm 1: UCVI

Input: δ: confidence level

while $k \le K-1$ **do**

Estimate the transition kernel

$$\hat{P}_h^k(s' \mid s,a) = \frac{N_h^k(s,a,s')}{N_h^k(s,a)}$$

Compute the exploration bonus $\mathrm{UCB}_h^k(s,a,\delta)$ as

$$\begin{cases} \infty\,, & N_h^k(s,a) = 0\,, \\ \dfrac{1}{N_h^k(s,a)} \displaystyle\sum_{k' \le k-1} r_h^{k'} \mathbb{1}\{(s_h^{k'}, a_h^{k'}) = (s,a)\} + \sqrt{\dfrac{4H^2 \log(nmHK/\delta)}{N_h^k(s,a)}}\,, & N_h^k(s,a) > 0\,; \end{cases}$$

For all states $s \in S$, $k \in [K]$, $V_H^k(s) \leftarrow 0$

for $h = H-1, \ldots, 0$ **do**

For all (s,a) pairs, update the action value estimate

$$\hat{Q}_h^k(s,a) = \min\{\mathrm{UCB}_h^k(s,a,\delta) + \sum_{s' \in S} \hat{P}_h^k(s' \mid s,a) \hat{V}_{h+1}^k(s'), H\}$$

For all $s \in S$, update the state value estimate

$$\hat{V}_h^k(s) = \max_a \hat{Q}_h^k(s,a)$$

For all $s \in S$, update the policy

$$\pi_h^k(s) = \arg\max_a \hat{Q}_h^k(s,a)$$

return $\hat{Q}_h^{K-1}(s,a)$, $\hat{V}_h^{K-1}(s)$, $\pi_h^{K-1}(s)$ for all $h \in [H]$

Figure 3.34: UCVI

UCVI is an extremely important algorithm. Taking confidence level $\delta = 1/KH$, the regrest can achieve a performance of $\overline{R}_T \le 10\sqrt{n^2 m H^4 K \log(nmH^2K^2)}$.

PSRL: Posterior Sampling for Reinforcement Learning

Algorithm 2: PSRL

Input: Prior $p(\theta_0)$ on the distribution of P_h and r_h

Initialize $\theta = \theta_0$

while $k \le K-1$ **do**

Sample P_h, r_h from θ

Run value iteration on P_h, r_h

Update the posterior probability distribution of θ_{k+1} by

$$p(\theta_{k+1} \mid \{\tau_{k'}\}_{k' \le k}) = \frac{p(\{\tau_{k'}\}_{k' \le k} \mid \theta) p(\theta)}{\int_{\theta'} p(\{\tau_{k'}\}_{k' \le k} \mid \theta') p(\theta') d\theta'}$$

Figure 3.35: PSRL

The regret of PSRL: $\overline{R}_T \le \sqrt{30 n^2 m H^3 K \log(nmHK)}$.

3.4.4 Model-free RL

The model-free framework is important in two types of domains:

1. When the MDP model is unknown, but we can sample trajectories from the MDP;
2. When the MDP model is known but computing the value function via our model-based control methods is infeasible due to the size of the domain.

Q-Learning

Recall the model-based value iteration, the model appears in two places. One for the computation of the action value function, and one for computing all the optimal policies $\max_{a \in A} [R(s,a) + \gamma \sum_{s' \in S} P(s' \mid s,a) V(s')]$. Both places could remove the dependency on P by using Q-function $Q(s,a)$ directly.
Another feature of Q-learning is the *step size* α. In Q-learning, instead of utilizing the Bellman optimality equation directly, the update only takes α portion of the action value.

Algorithm 2: Q-learning

Input: ϵ, α
For all $(s,a) \in \mathcal{S} \times \mathcal{A}$, $Q'(s,a) \leftarrow 0$, $Q(s,a) \leftarrow \infty$
while $\|Q - Q'\|_\infty > \epsilon$ **do**
  $Q \leftarrow Q'$
  Sample a trajectory τ from the policy $\pi(a \mid s) = \arg\max_{a \in A} Q(s,a)$
  For all state-action-reward-state tuple $(s,a,r,s') \in \tau$,
    $Q'(s,a) \leftarrow (1-\alpha) Q(s,a) + \alpha \max_{a' \in \mathcal{A}} [r + \gamma Q(s',a')]$
$Q^* \leftarrow Q$ for all $(s,a) \in \mathcal{S} \times \mathcal{A}$
$\pi^* \leftarrow \arg\max_{a \in \mathcal{A}} Q(s,a)$
return $Q^*(s,a)$, $\pi^*(s)$ for all s,a

Figure 3.36: Q-learning

Where $\delta_t = r + \gamma \max_{a \in \mathcal{A}} Q(s',a') - Q(s,a)$ is known as the **temporal difference** error (TD).
One problem with Q-learning is that the trajectory sampled is subject to the current policy, which lacks exploration. One simple approach is using ϵ greedy exploration.

Algorithm 3: Q-learning with ε-greedy exploration

Input: ϵ, α
For all $(s,a) \in \mathcal{S} \times \mathcal{A}$, $Q'(s,a) \leftarrow 0$, $Q(s,a) \leftarrow \infty$
while $\|Q - Q'\|_\infty > \epsilon$ **do**
  $Q \leftarrow Q'$
  Sample a trajectory τ from the policy

$$\pi(a \mid s) = \begin{cases} \arg\max_{a \in A} Q(s,a) & \text{with probability } 1-\varepsilon \\ \text{random action} & \text{with probability } \varepsilon \end{cases}$$

  For all state-action-reward-state tuple $(s,a,r,s') \in \tau$,
    $Q'(s,a) \leftarrow (1-\alpha) Q(s,a) + \alpha \max_{a' \in \mathcal{A}} [r + \gamma Q(s',a')]$
$Q^* \leftarrow Q$ for all $(s,a) \in \mathcal{S} \times \mathcal{A}$
$\pi^* \leftarrow \arg\max_{a \in \mathcal{A}} Q(s,a)$
return $Q^*(s,a)$, $\pi^*(s)$ for all s,a

Figure 3.37: Q-learning with ϵ-greedy exploration

In reinforcement learning algorithms, we tend to overestimate the value function of the states, this problem comes from the fact that we are using our estimate to both choose the better coin and estimate its value. Consider

the state s with two possible actions a_1, a_2, and $Q(s, a_1), Q(s, a_2) = 0$. Then we have:

$$\begin{aligned}\hat{V}(s) &= \mathbb{E}[\max(\hat{Q}(s, a_1), \hat{Q}(s, a_2))] \\ &\geq \max(\mathbb{E}[\hat{Q}(s, a_1)], \mathbb{E}[\hat{Q}(s, a_2)]) \\ &= \max(0, 0) \\ &= 0 = V^*(s)\,.\end{aligned} \tag{3.4.4.1}$$

Where the inequality is followed by Jensen's inequality.
In double Q-learning, we can maintain two independent unbiased estimates, Q_1 and Q_2 and when we use one to select the maximum, we can use the other to get an estimate of the value of this maximum.

Algorithm 6: Double Q-learning

Input: ϵ, α, γ
Initialize $Q_1(s, a), Q_2(s, a)$ for all $s \in S, a \in A$ arbitrarily
$t \leftarrow 0$
$\pi \leftarrow \epsilon$-greedy policy with respect to $Q_1 + Q_2$
while *true* **do**
  Sample action a_t from policy π at state s_t
  Take action a_t and observe reward r_t and next state s_{t+1}
  if *with 0.5 probability* **then**
    $Q_1(s_t, a_t) \leftarrow Q_1(s_t, a_t) + \alpha(r_t + \gamma Q_2(s_{t+1}, \arg\max_{a'} Q_1(s_{t+1}, a')) - Q_1(s_t, a_t))$
  else
    $Q_2(s_t, a_t) \leftarrow Q_2(s_t, a_t) + \alpha(r_t + \gamma Q_1(s_{t+1}, \arg\max_{a'} Q_2(s_{t+1}, a')) - Q_2(s_t, a_t))$
  $\pi \leftarrow \epsilon$-greedy policy with respect to $Q_1 + Q_2$
  $t \leftarrow t + 1$
return $\pi, Q_1 + Q_2$

Figure 3.38: Double Q-Learning

Monte-Carlo Policy Evaluation

The idea of Monte-Carlo is quite simple, run the game with the policy π for many iterations, and average all the rewards to approximate the value function.

Algorithm 1: First-visit Monte-Carlo policy evaluation

Input: $h_1, \ldots, h_j$
For all states s, $N(s) \leftarrow 0$, $S(s) \leftarrow 0$, $V(s) \leftarrow 0$
for *each episode* h_j **do**
  for $t = 1, \ldots, L_j$ **do**
    if $s_{j,t} \neq s_{j,u}$ *for* $u < t$ **then**
      $N(s_{j,t}) \leftarrow N(s_{j,t}) + 1$
      $S(s_{j,t}) \leftarrow S(s_{j,t}) + G_{j,t}$
      $V^\pi(s_{j,t}) \leftarrow S(s_{j,t})/N(s_{j,t})$
return V^π

Figure 3.39: Monte-Carlo policy evaluation

Some variations of Monte-Carlo policy evaluation:

- The body of the algorithm can remove S by using $V^\pi(s_{j,t}) \leftarrow V^\pi(s_{j,t}) + \frac{1}{N(s_{j,t})}(G_{j,t} - V^\pi(s_{j,t}))$.;
- The algorithm only counts the first visit in every episode, there is another *every-visit* version that doesn't use this condition;
- Similar to Q-learning, there is another incremental updating scheme $V^\pi(s_{j,t}) \leftarrow V^\pi(s) + \alpha(G_{j,t} - V^\pi(s))$.

Importance Sampling: Off-Policy Policy Evaluation

The goal of importance sampling is to estimate the expected value of a function $f(x)$ when x is drawn from the distribution q using only the data $f(x_1) \cdots f(x_n)$, where x_i are drawn from a different distribution p. To estimate

$\mathbb{E}_{x\sim q}[f(x)]$:

$$\begin{aligned}\mathbb{E}_{x\sim q}[f(x)] &= \int_x q(x)f(x)dx \\ &= \int_x p(x)\left[\frac{q(x)}{p(x)}f(x)\right]dx \\ &= \mathbb{E}_{x\sim p}\left[\frac{q(x)}{p(x)}f(x)\right] \\ &\approx \sum_{i=1}^{n}\left[\frac{q(x_i)}{p(x_i)}f(x_i)\right].\end{aligned} \tag{3.4.4.2}$$

Now we bring the context back to reinforcement learning. To estimate $V^{\pi_1}(s) = \mathbb{E}[G_t \mid s_t = s]$, using n trajectories $h_1, \cdots, h_n$ generated by π_2, the importance sampling estimate can give us:

$$V^{\pi_1}(s) \approx \frac{1}{n}\sum_{j=1}^{n}\frac{\mathbb{P}(h_j \mid \pi_1, s)}{\mathbb{P}(h_j \mid \pi_2, s)}G(h_j), \tag{3.4.4.3}$$

where $G(h_j) = \sum_{t=1}^{L_j-1}\gamma^{t-1}r_{j,t}$ is the total discounted reward for each trajectory. We have:

$$\begin{aligned}\mathbb{P}(h_j \mid \pi, s = s_{j,1}) &= \prod_{t=1}^{L_j-1}\mathbb{P}(a_{j,t} \mid s_{j,t})\mathbb{P}(r_{j,t} \mid s_{j,t}, a_{j,t})\mathbb{P}(s_{j,t+1} \mid s_{j,t}, a_{j,t}) \\ &= \prod_{t=1}^{L_j-1}\pi(a_{j,t} \mid s_{j,t})\mathbb{P}(r_{j,t} \mid s_{j,t}, a_{j,t})\mathbb{P}(s_{j,t+1} \mid s_{j,t}, a_{j,t}),\end{aligned} \tag{3.4.4.4}$$

This equation needs information of the model (P, r), but by the operation of division, we have:

$$\begin{aligned}V^{\pi_1}(s) &\approx \frac{1}{n}\sum_{j=1}^{n}\frac{\mathbb{P}(h_j \mid \pi_1, s)}{\mathbb{P}(h_j \mid \pi_2, s)}G(h_j) \\ &= \frac{1}{n}\sum_{j=1}^{n}\frac{\prod_{t=1}^{L_j-1}\pi_1(a_{j,t} \mid s_{j,t})\mathbb{P}(r_{j,t} \mid s_{j,t}, a_{j,t})\mathbb{P}(s_{j,t+1} \mid s_{j,t}, a_{j,t})}{\prod_{t=1}^{L_j-1}\pi_2(a_{j,t} \mid s_{j,t})\mathbb{P}(r_{j,t} \mid s_{j,t}, a_{j,t})\mathbb{P}(s_{j,t+1} \mid s_{j,t}, a_{j,t})}G(h_j) \\ &= \frac{1}{n}\sum_{j=1}^{n}G(h_j)\prod_{t=1}^{L_j-1}\frac{\pi_1(a_{j,t} \mid s_{j,t})}{\pi_2(a_{j,t} \mid s_{j,t})}.\end{aligned} \tag{3.4.4.5}$$

Which maintains the off-policy policy evaluation as a model-free technique.

Temporal Difference Learning

Replace the G_t in incremental Monte-Carlo policy evaluation with $r_t + \gamma V^{\pi}(s_{t+1})$, and we can get:

Algorithm 5: TD Learning to evaluate policy π

Input: step size α, number of trajectories n
For all states s, $V^{\pi}(s) \leftarrow 0$
while $n > 0$ **do**
  Begin episode E at state s
  while *episode E has not terminated* **do**
    $a \leftarrow$ action at state s under policy π
    Take action a in E and observe reward r, next state s'
    $V^{\pi}(s) \leftarrow V^{\pi}(s) + \alpha(R + \gamma V^{\pi}(s') - V^{\pi}(s))$
    $s \leftarrow s'$
  $n \leftarrow n - 1$
return V^{π}

Figure 3.40: TD Learning

where the term $\delta_t = r_t + \gamma V^{\pi}(s_{t+1}) - V^{\pi}(s_t)$ is called *temporal difference*.

Monte-Carlo Control

Definition 3.4.4.1: GLIE: Greedy in the limit of infinite exploration

A policy π is *greedy in the limit of infinite exploration* if:

1. $\lim_{k\to\infty} N_k(s,a) \to \infty$with probability1;
2. $\lim_{k\to\infty} \pi_k(a \mid s) = \arg\max_a Q(s,a)$with probability1.

One example of GLIE is an ϵ-greedy policy where $\epsilon \to 0$.

Algorithm 3: Online Monte-Carlo control

Initialize $Q(s,a) = 0$, $Returns(s,a) = 0$ for all $s \in S, a \in A$
Set $k \leftarrow 1$
while *true* **do**
  Sample k-th episode $\{s_{t,k}, a_{t,k}, r_{t,k}\}_{t\in[H]}$ under policy π
  for $t = 1, \ldots, H$ **do**
    if *First visit to* (s,a) *in episode* k **then**
      Append $\sum_{t'=t}^{H} r_{t',k}$ to $Returns(s_t, a_t)$
      $Q(s_t, a_t) \leftarrow$ average$(Returns(s_t, a_t))$
  $k \leftarrow k+1$, $\epsilon = \frac{1}{k}$
  $\pi_k = \epsilon$-greedy with respect to Q
Return Q, π_k

Figure 3.41: Online Monte-Carlo control

Algorithm 4: SARSA

Input: ϵ, α_t
Initialize $Q(s,a)$ for all $s \in S, a \in A$ arbitrarily except $Q(terminal, \cdot) = 0$
$\pi \leftarrow \epsilon$-greedy policy with respect to Q
for *each episode* **do**
  $t \leftarrow 1$
  Set s_1 as the starting state
  Choose action a_1 from policy $\pi(s_1)$
  while *until episode terminates* **do**
    Take action a_t and observe reward r_t and next state s_{t+1}
    Choose action a_{t+1} from policy $\pi(s_{t+1})$
    $Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha_t(r_t + \gamma Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t))$
    $\pi \leftarrow \epsilon$-greedy with respect to Q
    $t \leftarrow t+1$
Return Q, π

Figure 3.42: SARSA

SARSA is short for *state action reward state action*. The difference between SARSA and Q-learning is that SARSA is an on-policy algorithm, updating the Q-function using the current strategy, while Q-learning is an off-policy learning algorithm.

3.4.5 Value Function Approximation

The methodologies above try to estimate the value function or Q-function for every state and action, but these methods might not apply to those scenarios with very large state and action spaces. A popular approach for this issue is via **value function approximation**(VFA).

$$V^\pi(s) \approx \hat{V}(s, \mathbf{w}) \quad \text{or} \quad Q^\pi(s,a) \approx \hat{Q}(s,a,\mathbf{w}), \tag{3.4.5.1}$$

where $\mathbf{w}$ refers to parameter or wights. Some function approximations are listed below:

1. Linear combinations of features;
2. Neural networks;
3. Decision trees.

Linear feature representations

In linear function representations, we use a feature vector to represent a state:

$$x(s) = (x_1(s), x_2(s), \ldots, x_d(s))^T, \tag{3.4.5.2}$$

where d is the dimensionality of the feature space, the approximation can be expressed by:

$$\hat{V}(s, \mathbf{w}) = x(s)^T \mathbf{w} = \sum_{j=1}^{d} x_j(s) \mathbf{w}_j. \tag{3.4.5.3}$$

Then we can define the loss function:

$$J(\mathbf{w}) = \mathbb{E}_{s \sim \rho^\pi(s)} \left[(V^\pi(s) - \hat{V}(s, \mathbf{w}))^2 \right]; \tag{3.4.5.4}$$

$$\rho^\pi(s) = \lim_{T \to \infty} \frac{\sum_{t=0}^{T} \gamma^t \mathbb{P}(s_t = s \mid \pi)}{\sum_{t=0}^{T} \gamma^t}. \tag{3.4.5.5}$$

where $\rho^\pi(s)$ denotes the cumulative probability under the policy π. There are several ways to solve this optimization problem:

1. Gradient Descent (SGD);
2. Policy evaluation with linear VFA;

Algorithm 1: Monte-Carlo policy evaluation with linear VFA

```
Initialize w = 0, R(s) = 0 ∀s, k = 1
while true do
    Sample k-th episode (s_{k,1}, a_{k,1}, r_{k,1}, s_{k,2}, ..., s_{k,H_k}) given π
    for t = 1, ..., H_k do
        if first visit to s in episode k then
            Append Σ_{j=t}^{H_k} r_{k,j} to R(s_t)
            w ← w + α(avg(R(s_t)) − V̂(s_t, w))x(s_t)
    k = k + 1
```

Figure 3.43: Monte-Carlo policy evaluation with linear VFA

The updating formula $\mathbf{w} \leftarrow \mathbf{w} + \alpha(avg(R(s_t)) - \hat{V}(s_t, \mathbf{w}))x(s_t)$ is derived by taking derivative of $\mathbf{w}$ on $J(\mathbf{w})$:

$$\begin{aligned} \nabla J(\mathbf{w}) &= 2(V^\pi(s) - \hat{V}(s, \mathbf{w})) \frac{\partial (V^\pi(s) - \hat{V}(s, \mathbf{w}))}{\partial \mathbf{w}} \\ &= 2(V^\pi(s) - \hat{V}(s, \mathbf{w})) \cdot x(s). \end{aligned} \tag{3.4.5.6}$$

In different policy evaluation settings the term $V^\pi(s) - \hat{V}(s, \mathbf{w})$ could be replaced by different styles:

- Monte-Carlo: $\sum_{j=t}^{H_k} r_{k,j} - \hat{V}(s, \mathbf{w})$;
- Temporal-difference: $r + \gamma \hat{V}^\pi(s', \mathbf{w}) - \hat{V}^\pi(s, \mathbf{w})$;
- Q-learning (maximum TD(0)): $r + \gamma \max_{a'} \hat{Q}(s', a', \mathbf{w}) - \hat{Q}(s, a, \mathbf{w})$.

One thing about VFA is that these algorithms may not converge or converge to a suboptimal solution.

Deep Q-Learing

This part would introduce Deep Q-learning ([Mnih et al., 2015]), Double DQN ([Van Hasselt et al., 2016]) and Dueling DQN ([Wang et al., 2016]).

DQN

The DQN is designed to estimate the state-action function $\hat{Q}(s, a, \mathbf{w})$.

Layer	Input	Filter size	Stride	Num filters	Activation	Output
conv1	84x84x4	8x8	4	32	ReLU	20x20x32
conv2	20x20x32	4x4	2	64	ReLU	9x9x64
conv3	9x9x64	3x3	1	64	ReLU	7x7x64
fc4	7x7x64			512	ReLU	512
fc5	512			18	Linear	18

Figure 3.44: DQN Architecture

The DQN architecture takes inputs consisting of an $84 \times 84 \times 4$ image, which denotes the state. But the original size of the Atari game is $210 \times 160 \times 3$, so it needs some pre-processing:

- *Single frame encoding*:
- *Dimensionality reduction*:

The loss function is given by:

$$J(\mathbf{w}) = \mathbb{E}_{(s_t, a_t, r_t, s_{t+1})}[(y_t^{DQN} - \hat{Q}(s_t, a_t, \mathbf{w}))^2], \tag{3.4.5.7}$$

$$y_t^{DQN} = r_t + \gamma \max_{a'} \hat{Q}(s_{t+1}, a', \mathbf{w}^-). \tag{3.4.5.8}$$

Where $\mathbf{w}^-$ represents the parameters of the target network, and $\mathbf{w}$ is the online network after the update.
Experience Replay: The Q-network is updated by SGD with sampled gradients from minibatch data: $(s, a, r, s') \sim$ Uniform(D). D is called the *replay buffer*. This setting has some advantages and limitations:

- *Greater data efficiency*: Each step of experience can be potentially used for many updates, which improves data efficiency;
- *Remove sample correlations*: Randomizing the transition experiences reduces the correlations between consecutive samples and therefore reduces the variance of updates and stabilizes the learning;
- *Avoiding oscillations or divergence*;
- *Limitations*: Both the replay buffer and the uniform sampling don't t differentiate important transitions or informative transitions. A more sophisticated replay strategy is *Prioritized Replay* by [Schaul et al., 2015].

Target Network is used to improve the stability of learning and deal with the nonstationary learning targets. For every $C = 10000$ update steps the target network $\hat{Q}(s, a, \mathbf{w}^-)$ is updated by copying the parameters' values $(\mathbf{w}^- = \mathbf{w})$ from the online network $\hat{Q}(s, a, \mathbf{w})$.

Algorithm 2: Deep Q-learning

Initialize replay memory D with a fixed capacity
Initialize action value function $\hat{Q}$ with random weights $\mathbf{w}$
Initialize target action value function $\hat{Q}$ with weights $\mathbf{w}^- = \mathbf{w}$
for *episode* $k = 1, \ldots, K$ **do**
  Observe initial frame x_1 and pre-process frame to get state s_1
  for *time step* $t = 1, \ldots, T$ **do**
    Select action $a_t = \begin{cases} \text{random action} & \text{with probability } \epsilon \\ \arg\max_a \hat{Q}(s_t, a, \mathbf{w}) & \text{otherwise} \end{cases}$
    Execute action a_t in emulator and observe reward r_t and image x_{t+1}
    pre-process s_t, x_{t+1} to get s_{t+1}, and store transition (s_t, a_t, r_t, s_{t+1}) in D
    Sample uniformly a random minibatch of N transitions
     $\{(s_j, a_j, r_j, s_{j+1})\}_{j \in [N]}$ from D
    Set $y_j = r_j$ if episode ends at step $j+1$, otherwise set
     $y_j = r_j + \gamma \max_{a'} \hat{Q}(s_{j+1}, a', \mathbf{w}^-)$
    Perform a stochastic gradient descent step on
     $J(\mathbf{w}) = \frac{1}{N} \sum_{j=1}^{N} (y_j - \hat{Q}(s_j, a_j, \mathbf{w}))^2$ with respect to $\mathbf{w}$
    Every C steps reset $\mathbf{w}^- = \mathbf{w}$

Figure 3.45: Deep Q-learning

DDQN

Similar to the idea used in double Q-learning, DDQN uses two different networks to perform action selection and value function approximation. However, DDQN doesn't maintain two independent networks but utilizes the target network by:

$$y_t^{DDQN} = r_t + \gamma \hat{Q}(s_{t+1}, \arg\max_{a'} \hat{Q}(s_{t+1}, a', \mathbf{w}), \mathbf{w}^-). \tag{3.4.5.9}$$

Dueling DQN

Recall the *advantage function*:

$$A^\pi(s, a) = Q^\pi(s, a) - V^\pi(s). \tag{3.4.5.10}$$

We have $\mathbb{E}_{a \sim \pi(s)}[A^\pi(s, a)] = 0$. Like in DQN, the dueling network is also a neural network function approximator for learning the Q-function. Differently, it approximates the Q function by decoupling the value function and the advantage function.

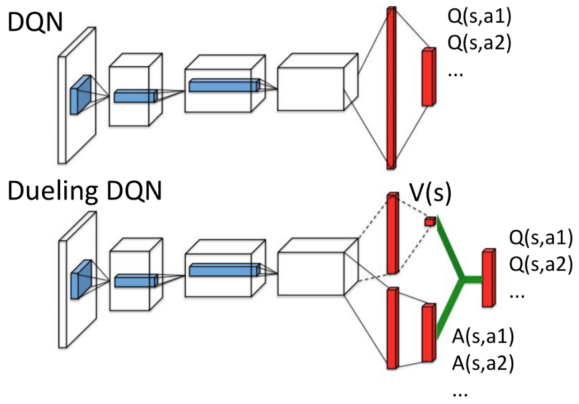


Figure 3.46: Dueling DQN v.s. DQN

Where the green layer conducts the following operations:

$$\hat{Q}(s,a,\mathbf{w},\mathbf{w_A},\mathbf{w_V}) = \hat{V}(s,\mathbf{w},\mathbf{w_V}) + \left(A(s,a,\mathbf{w},\mathbf{w_A}) - \max_{a' \in A} A(s,a',\mathbf{w},\mathbf{w_A})\right); \tag{3.4.5.11}$$

$$\hat{Q}(s,a,\mathbf{w},\mathbf{w_A},\mathbf{w_V}) = \hat{V}(s,\mathbf{w},\mathbf{w_V}) + \left(A(s,a,\mathbf{w},\mathbf{w_A}) - \frac{1}{|\mathcal{A}|}\sum_{a'} A(s,a',\mathbf{w},\mathbf{w_A})\right). \tag{3.4.5.12}$$

in which $\mathbf{w_V}$ is the parameter of the FC layer to predict the value functions, and $\mathbf{w_A}$ is the parameter of the FC layer to predict the advantage functions.

3.4.6 Policy Gradient Algorithms

In most RL algorithms, the policy does not exist without the action value estimates $Q(s,a)$, these are called **value-based** algorithms. These algorithms have several limitations:

1. Can only address discrete action spaces due to the $\arg\max_{a \in A}$ operation;
2. Computing $Q(s,a)$ for all state-action pairs is costly;
3. Can only improve the policy indirectly by improving the estimates of the value function.

Policy-based method directly parameterizes the policy function $\pi_\theta(s)$ without calculating the value functions. A value function may still be used to learn the policy parameters, but it is not required for action selection. To implement the policy-based methods, an important step is to approximate policies with parametrization. In discrete case, we can use softmax and state-action preferences $h(s,a,\theta)$ to parameterize a policy:

$$\pi(a \mid s,\boldsymbol{\theta}) = \frac{\exp(h(a,s,\boldsymbol{\theta}))}{\sum_{a'}\exp(h(a',s,\boldsymbol{\theta}))}. \tag{3.4.6.1}$$

In continuous cases, we can use the following functions to parameterize a policy:

$$\pi(a \mid s,\boldsymbol{\theta_\mu},\boldsymbol{\theta_\sigma}) = \frac{1}{\sigma(s,\theta_\sigma)\sqrt{2\pi}}\exp(-\frac{(a-\mu(s,\theta_\mu))^2}{2\sigma(s,\theta_\sigma)^2}); \tag{3.4.6.2}$$

$$\mu(s,\boldsymbol{\theta}) = \boldsymbol{\theta}_{\boldsymbol{\mu}}^T \boldsymbol{x}_{\boldsymbol{\mu}}(s), \quad \sigma(s,\boldsymbol{\theta}) = \exp(\boldsymbol{\theta}_{\boldsymbol{\sigma}}^T \boldsymbol{x}_{\boldsymbol{\sigma}}(s)), \tag{3.4.6.3}$$

In the settings where the episode terminates at some terminal state set, we can define the objective function as:

$$J(\boldsymbol{\theta}) = V^{\pi_\theta}(s_0) = \sum_{s\in\mathcal{S}} \rho^{\pi_\theta}(s \mid s_0) r(s), \tag{3.4.6.4}$$

where $r(s) = \mathbb{E}_{a\sim_T}[\mathcal{R}(s,a)]$ and $\rho^{\pi_\theta}(s \mid s_0) = \frac{1}{T}\sum_{t=0}^{T} \mathbb{P}(s_t = s \mid s_0, \pi_\theta)$. In the continuous setting where the process continues infinitely, we can define:

$$\begin{aligned} J(\boldsymbol{\theta}) &= \lim_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} \mathbb{E}[r_t \mid s_0, \pi_\theta] \\ &= \lim_{t\to\infty} \mathbb{E}[r_t \mid s_0, \pi_\theta] \\ &= \sum_{s} \rho^{\pi_\theta}(s \mid s_0) r(s) \\ &= V^{\pi\theta}(s_0). \end{aligned} \tag{3.4.6.5}$$

Theorem 3.4.6.1: Policy Gradient Theorem

$$\begin{aligned} \nabla_\theta J(\theta) &\propto \sum_{s\in\mathcal{S}} \rho^{\pi_\theta}(s \mid s_0) \sum_{a\in\mathcal{A}} Q^{\pi_\theta}(s,a) \nabla_\theta \pi_\theta(a \mid s) \\ &= \mathbb{E}_\pi \left[Q^{\pi_\theta}(s,a) \nabla_\theta \log \pi_\theta(a \mid s) \right]. \end{aligned} \tag{3.4.6.6}$$

Proof 3.4.3: Policy Gradient Theorem

Here we only prove the episodic case.

$$\begin{aligned} \nabla_\theta V^\pi(s) &= \nabla_\theta \Big(\sum_{a\in\mathcal{A}} \pi_\theta(a \mid s) Q^\pi(s,a) \Big) \\ &= \sum_{a\in\mathcal{A}} \left(\nabla_\theta \pi_\theta(a \mid s) Q^\pi(s,a) + \pi_\theta(a \mid s) \nabla_\theta Q^\pi(s,a) \right) \\ &= \sum_{a\in\mathcal{A}} \left(\nabla_\theta \pi_\theta(a \mid s) Q^\pi(s,a) + \pi_\theta(a \mid s) \nabla_\theta \sum_{s',r} \mathbb{P}(s', r \mid s, a)(r + V^\pi(s')) \right) \\ &= \sum_{a\in\mathcal{A}} \left(\nabla_\theta \pi_\theta(a \mid s) Q^\pi(s,a) + \pi_\theta(a \mid s) \sum_{s',r} \mathbb{P}(s', r \mid s, a) \nabla_\theta V^\pi(s') \right) \\ &= \sum_{a\in\mathcal{A}} \left(\nabla_\theta \pi_\theta(a \mid s) Q^\pi(s,a) + \pi_\theta(a \mid s) \sum_{s'} \mathbb{P}(s' \mid s, a) \nabla_\theta V^\pi(s') \right) \end{aligned} \tag{3.4.6.7}$$

This equation has a nice recursive form, to unroll the recursion of $\nabla_\theta V^\pi(s')$, denote

$$\phi(s) = \sum_{a\in\mathcal{A}} \nabla_\theta \pi_\theta(a \mid s) Q^\pi(s,a) \tag{3.4.6.8}$$

$$\eta(s) = \sum_{k=0}^{\infty} \rho^\pi(s_0 \to s, k) \tag{3.4.6.9}$$

for simplicity, and denote $\rho^\pi(s \to s', k)$ as the probability of transitioning from s to s' with policy π after k steps. We have:

$$\begin{aligned}
\nabla_\theta J(\theta) = \nabla_\theta V^\pi(s_0) &= \phi(s_0) + \sum_a \pi_\theta(a \mid s) \sum_{s'} \mathbb{P}(s' \mid s_0, a) \nabla_\theta V^\pi(s') \\
&= \phi(s_0) + \sum \rho^\pi(s_0 \to s', 1) \nabla_\theta V^\pi(s') \\
&= \phi(s_0) + \sum_{s'} \rho^\pi(s_0 \to s', 1) \phi(s') + \sum_{c''} \rho^\pi(s_0 \to s'', 2) \nabla_\theta V^\pi(s'') \\
&= \sum_s \sum_{k=0}^{\infty} \rho^\pi(s_0 \to s, k) \phi(s) = (\sum_s \eta(s)) \sum_s \eta(s) \phi(s) \\
&\propto \sum_s \frac{\eta(s)}{\sum_s \eta(s)} \phi(s) = \sum_s \rho^\pi(s \mid s_0) \sum_a \nabla_\theta \pi_\theta(a|s) Q^\pi(s, a).
\end{aligned} \tag{3.4.6.10}$$

Further, the policy gradient can be written as:

$$\begin{aligned}
\nabla_\theta J(\theta) &\propto \sum_{s \in \mathcal{S}} \rho^\pi(s \mid s_0) \sum_{a \in \mathcal{A}} Q^\pi(s, a) \nabla_\theta \pi_\theta(a \mid s) \\
&= \sum_{s \in \mathcal{S}} \rho^\pi(s \mid s_0) \sum_{a \in \mathcal{A}} Q^\pi(s, a) \pi_\theta(a \mid s) \frac{\nabla_\theta \pi_\theta(a \mid s)}{\pi_\theta(a \mid s)} \\
&= \sum_{s \in \mathcal{S}} \rho^\pi(s \mid s_0) \sum_{a \in \mathcal{A}} \pi_\theta(a \mid s) Q^\pi(s, a) \frac{\nabla_\theta \pi_\theta(a \mid s)}{\pi_\theta(a \mid s)} \\
&= \mathbb{E}_\pi[Q^\pi(s, a) \nabla_\theta \log \pi_\theta(a \mid s)],
\end{aligned} \tag{3.4.6.11}$$

Q.E.D.

Algorithm 1: REINFORCE (Monte-Carlo method)

Initialize the policy parameter θ
for *each episode* **do**
  Sample one trajectory on policy π_θ: $s_0, a_0, r_0, s_1, a_1, \ldots, s_T$
  for *each* $t = 0, 1, \ldots, T$ **do**
    $G_t \leftarrow \sum_{t'=t}^{T} \gamma^{t'-t} r_{t'}$
    $\theta \leftarrow \theta + \alpha \gamma^t G_t \nabla_\theta \log \pi_\theta(a_t \mid s_t)$

Figure 3.47: REINFORCE

For $Q^\pi(s, a)$, we can use $G_t = \sum \gamma^t r_t$ to estimate. For $\nabla_\theta \log \pi_\theta(a \mid s)$, it depends on its form. Policy gradient suffers from high variance, one solution is by subtracting a baseline $b(s)$ independent of θ:

$$\nabla_\theta J(\theta) \propto \sum_{s \in \mathcal{S}} \rho^\pi(s \mid s_0) \sum_{a \in \mathcal{A}} (Q^\pi(s, a) - b(s)) \nabla \pi_\theta(a \mid s). \tag{3.4.6.12}$$

We can see that:

$$\sum_a b(s) \nabla \pi(a \mid s, \theta) = b(s) \nabla \sum_a \pi(a \mid s, \theta) = b(s) \nabla 1 = 0, \tag{3.4.6.13}$$

which means the expectation value does not change. A natural choice for the baseline is an estimate of the state value $\hat{V}(s, \boldsymbol{w})$.

Remark 3.4.6

This equation can be generalized to:

Lemma 3.4.6.1: Stein's identity

$$\mathbb{E}_\pi[\nabla_a \log \pi(a \mid s)\phi(s,a) + \nabla_a \phi(s,a)] = 0, \tag{3.4.6.14}$$

where $\phi(s,a)$ is an arbitrary state-action function.

Algorithm 2: REINFORCE with baseline

Initialize the policy parameter $\boldsymbol{\theta}$ and $\boldsymbol{w}$ at random.
for *each episode* **do**
  Sample one trajectory under policy π_θ: $s_0, a_0, r_0, s_1, a_1, r_1 \ldots, s_T$
  for *each* $t = 1, 2, \ldots, T$ **do**
    $G_t \leftarrow \sum_{t'=t}^{T} \gamma^{t'-t} r_{t'}$
    $\delta \leftarrow G_t - \hat{V}(s_t, \boldsymbol{w})$
    $\boldsymbol{w} \leftarrow \boldsymbol{w} + \alpha_{\boldsymbol{w}} \delta \nabla_{\boldsymbol{w}} \hat{V}(s_t, \boldsymbol{w})$
    $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \alpha_{\boldsymbol{\theta}} \gamma^t \delta \nabla_{\boldsymbol{\theta}} \log \pi_\theta(a_t \mid s_t)$

Figure 3.48: REINFORCE with baseline

Remark 3.4.7

Why the introduction of baseline reduces the variance:
Assume $b(s)$ is an unbiased estimator of $V^\pi(s)$, then the term $R_t(\tau) - b(s_t)$ has a mean 0. Thus,

$$\begin{aligned} \operatorname{Var}\left(\sum_{t=0}^{T-1} \nabla_\theta \log \pi_\theta(a_t|s_t)(R_t(\tau) - b(s_t))\right) &\approx \sum_{t=0}^{T-1} \mathbb{E}\tau \left[(\nabla_\theta \log \pi_\theta(a_t|s_t)(R_t(\tau) - b(s_t)))^2\right] \\ &\approx \sum_{t=0}^{T-1} \mathbb{E}_\tau \left[(\nabla_\theta \log \pi_\theta(a_t|s_t))^2\right] \mathbb{E}_\tau \left[(R_t(\tau) - b(s_t))^2\right] \end{aligned} \tag{3.4.6.15}$$

The above equation does not necessarily show that adding a baseline would reduce the variance, this only provides an intuitive understanding.

Remark 3.4.8

The general form of *policy gradient*:

$$g = \mathbb{E}\left[\sum_{t=0}^{\infty} \Psi_t \nabla_\theta \log \pi_\theta\left(a_t \mid s_t\right)\right] \tag{3.4.6.16}$$

The Ψ_t could be the following form:

- $\sum_{t=0}^{\infty} r_t$: : the total reward of the trajectory;
- $Q^\pi(s_t, a_t)$: the action value function;
- $\sum_{t'}^{\infty}$: the reward following action a_t;
- $\sum_{t'=t}^{\infty} r_{t'} - b\left(s_t\right)$: the reward following action a_t with a baseline;
- $A^\pi(s_t, a_t)$: the advantage function;
- $r_t + V^\pi(s_{t+1}) - V^\pi(s_t)$: the TD residual.

Actor-Critic

In practice, most of the variance is from the Monte-Carlo estimation G_t of $Q(s_t, a_t)$. If we use the parameterized temporal-difference method to estimate the state-action function, then we can get **actor-critic** algorithm. Actor-critic methods consist of two models;

1. The critic updates the value function parameters w;
2. The actor updates the policy parameters θ in the direction suggested by the critic.

Algorithm 3: One-step actor–critic (episodic)

Initialize the policy parameter $\boldsymbol{\theta}$ and $\boldsymbol{w}$ at random. **for** *each episode* **do**
  Initialize s_0, the first state of each episode
  for *each* $t = 0, 1, \ldots, T-1$ **do**
    sample $a \sim \pi(a \mid s_t, \boldsymbol{\theta})$
    take action a and observe s', r
    $\delta \leftarrow r + \gamma \hat{V}(s', \boldsymbol{w}) - \hat{V}(s, \boldsymbol{w})$
    $\boldsymbol{w} \leftarrow \boldsymbol{w} + \alpha_{\boldsymbol{w}} \delta \nabla_{\boldsymbol{w}} \hat{V}(s, \boldsymbol{w})$
    $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \alpha_{\boldsymbol{\theta}} \delta \nabla_{\boldsymbol{\theta}} \log \pi(a \mid s, \boldsymbol{\theta})$
    $s' \leftarrow s$

Figure 3.49: One-step actor-critic

Deep Deterministic Policy Gradient: DDPG

DDPG is an algorithm similar to DQN, it utilizes the trick of *replay buffer* and *target netwrok*. However, DDPG maintains two networks at the same time: one to approximate the Q-function (critic), and another to choose the action (actor). Because the approximated Q-function can be differentiable, DDPG can be applied for continuous action spaces.

Algorithm 1 Deep Deterministic Policy Gradient

Input: initial policy parameters θ, Q-function parameters ϕ, empty replay buffer $\mathcal{D}$
Set target parameters equal to main parameters $\theta_{\text{targ}} \leftarrow \theta$, $\phi_{\text{targ}} \leftarrow \phi$
repeat
  Observe state s and select action $a = \text{clip}(\mu_\theta(s) + \epsilon, a_{Low}, a_{High})$, where $\epsilon \sim \mathcal{N}$
  Execute a in the environment
  Observe next state s', reward r, and done signal d to indicate whether s' is terminal
  Store (s, a, r, s', d) in replay buffer $\mathcal{D}$
  If s' is terminal, reset environment state.
  if it's time to update **then**
    for however many updates **do**
      Randomly sample a batch of transitions, $B = \{(s, a, r, s', d)\}$ from $\mathcal{D}$
      Compute targets

$$y(r, s', d) = r + \gamma(1 - d)Q_{\phi_{\text{targ}}}(s', \mu_{\theta_{\text{targ}}}(s'))$$

      Update Q-function by one step of gradient descent using

$$\nabla_\phi \frac{1}{|B|} \sum_{(s,a,r,s',d) \in B} \left(Q_\phi(s, a) - y(r, s', d)\right)^2$$

      Update policy by one step of gradient ascent using

$$\nabla_\theta \frac{1}{|B|} \sum_{s \in B} Q_\phi(s, \mu_\theta(s))$$

      Update target networks with

$$\phi_{\text{targ}} \leftarrow \rho\phi_{\text{targ}} + (1 - \rho)\phi$$
$$\theta_{\text{targ}} \leftarrow \rho\theta_{\text{targ}} + (1 - \rho)\theta$$

    end for
  end if
until convergence

Figure 3.50: Deep Deterministic Policy Gradient

where $d = 1$ if it's the terminal state, otherwise $d = 0$.

Soft Actor-Critic: SAC

This algorithm is proposed by [Haarnoja et al., 2018]. SAC is an algorithm that optimizes a stochastic policy in an off-policy way, a key feature of SAC is **entropy regularization**, which is designed to maximize the trade-off between the expected return and the randomness in the policy. (exploitation v.s. exploration) The objective function is given by:

$$\pi^* = \arg\max_\pi \mathop{\text{E}}_{\tau \sim \pi} \left[\sum_{t=0}^{\infty} \gamma^t \Big(R(s_t, a_t, s_{t+1}) + \alpha H\left(\pi(\cdot|s_t)\right) \Big) \right], \tag{3.4.6.17}$$

where $H(P) = \text{E}_{x \sim P}\left[-\log P(x)\right]$ is the **entropy**, SAC concurrently learns a policy π_θ and two Q-functions Q_{ϕ_1}, Q_{ϕ_2}:

Algorithm 1 Soft Actor-Critic

Input: initial policy parameters θ, Q-function parameters ϕ_1, ϕ_2, empty replay buffer $\mathcal{D}$
Set target parameters equal to main parameters $\phi_{\text{targ},1} \leftarrow \phi_1$, $\phi_{\text{targ},2} \leftarrow \phi_2$
repeat
  Observe state s and select action $a \sim \pi_\theta(\cdot|s)$
  Execute a in the environment
  Observe next state s', reward r, and done signal d to indicate whether s' is terminal
  Store (s, a, r, s', d) in replay buffer $\mathcal{D}$
  If s' is terminal, reset environment state.
  if it's time to update **then**
    for j in range(however many updates) **do**
      Randomly sample a batch of transitions, $B = \{(s, a, r, s', d)\}$ from $\mathcal{D}$
      Compute targets for the Q functions:

$$y(r, s', d) = r + \gamma(1 - d)\left(\min_{i=1,2} Q_{\phi_{\text{targ},i}}(s', \tilde{a}') - \alpha \log \pi_\theta(\tilde{a}'|s')\right), \quad \tilde{a}' \sim \pi_\theta(\cdot|s')$$

      Update Q-functions by one step of gradient descent using

$$\nabla_{\phi_i} \frac{1}{|B|} \sum_{(s,a,r,s',d) \in B} \left(Q_{\phi_i}(s, a) - y(r, s', d)\right)^2 \qquad \text{for } i = 1, 2$$

      Update policy by one step of gradient ascent using

$$\nabla_\theta \frac{1}{|B|} \sum_{s \in B} \Big(\min_{i=1,2} Q_{\phi_i}(s, \tilde{a}_\theta(s)) - \alpha \log \pi_\theta\left(\tilde{a}_\theta(s)|\, s\right) \Big),$$

      where $\tilde{a}_\theta(s)$ is a sample from $\pi_\theta(\cdot|s)$ which is differentiable wrt θ via the reparametrization trick.
      Update target networks with

$$\phi_{\text{targ},i} \leftarrow \rho \phi_{\text{targ},i} + (1 - \rho)\phi_i \qquad \text{for } i = 1, 2$$

    end for
  end if
until convergence

Figure 3.51: Soft Actor-Critic

Trust Region Policy Optimization: TRPO

If the noise of the gradient is large, choosing the best step size using the exact line search could be analogous to choosing the optimum among the random signals. One way to solve this is the **trust region** method ([Schulman et al., 2015]). Let $c(s_t)$ be the cost function, and denote

$$\begin{aligned} &\eta(\pi) = \mathbb{E}_{s_0, a_0, \ldots}\left[\sum_{t=0}^{\infty} \gamma^t c(s_t)\right], \text{ where} \\ &s_0 \sim \rho_0(s_0), a_t \sim \pi(a_t|s_t), s_{t+1} \sim P(s_{t+1}|s_t, a_t). \end{aligned} \tag{3.4.6.18}$$

Recall the advantage function $A_\pi(s, a) = Q_\pi(s, a) - V_\pi(s)$, so for another policy $\tilde{\pi}$, we have the improvement formula:

$$\eta(\tilde{\pi}) = \eta(\pi) + \mathbb{E}_{s_0, a_0, s_1, a_1, \ldots}\left[\sum_{t=0}^{\infty} \gamma^t A_\pi(s_t, a_t)\right] \tag{3.4.6.19}$$

This is accumulated over timesteps. If we denote $\rho_\pi(s) = (P(s_0 = s) + \gamma P(s_1 = s) + \gamma^2 P(s_2 = s) + \ldots)$, then we can rewrite this in accumulation over states:

$$\eta(\tilde{\pi}) = \eta(\pi) + \sum_s \rho_{\tilde{\pi}}(s) \sum_a \tilde{\pi}(a|s) A_\pi(s, a). \tag{3.4.6.20}$$

Optimizing the right side directly is difficult since we can not take the derivative of $\rho_{\tilde{\pi}}(s)$. So we introduce a local approximation:

$$L_\pi(\tilde{\pi}) = \eta(\pi) + \sum_s \rho_\pi(s) \sum_a \tilde{\pi}(a|s) A_\pi(s, a), \tag{3.4.6.21}$$

where we approximate $\rho_{\tilde{\pi}}(s)$ using $\rho_\pi(s)$. Now we can optimize the right side directly, it is proved that:

Lemma 3.4.6.2: [Kakade and Langford, 2002]

$$\eta(\pi_{\text{new}}) \leq L_{\pi_{\text{old}}}(\pi_{\text{new}}) + \frac{2\epsilon\gamma}{(1 - \gamma(1 - \alpha))(1 - \gamma)} \alpha^2, \tag{3.4.6.22}$$

where the term is given by:

$$\pi' = \arg\min_{\pi'} L_{\pi_{\text{old}}}(\pi'); \tag{3.4.6.23}$$

$$\pi_{\text{new}}(a|s) = (1 - \alpha)\pi_{\text{old}}(a|s) + \alpha\pi'(a|s); \tag{3.4.6.24}$$

$$\epsilon = \max_s |\mathbb{E}_{a \sim \pi'(a|s)} [A_\pi(s, a)]|. \tag{3.4.6.25}$$

Since $\alpha, \gamma \in [0, 1]$, we can get a tighter bound:

$$\eta(\pi_{\text{new}}) \leq L_{\pi_{\text{old}}}(\pi_{\text{new}}) + \frac{2\epsilon\gamma}{(1 - \gamma)^2} \alpha^2. \tag{3.4.6.26}$$

Now we can generalize this lemma to:

Theorem 3.4.6.2: Monotonic Improvement Guarantee for General Stochastic Policies

By replacing α with a distance measure between π and $\tilde{\pi}$, we can generate a bound similar to equation 3.4.6. More specifically, if we let $\alpha = D_{\text{TV}}^{\max}(\pi_{\text{old}}, \pi_{\text{new}}) = \max_s D_{TV}(\pi(\cdot|s) \parallel \tilde{\pi}(\cdot|s))$, where $D_{TV}(p \parallel q) = \frac{1}{2} \sum_i |p_i - q_i|$ is the *total variation divergence*, and $\epsilon = \max_s |\mathbb{E}_{a \sim \pi'(a|s)}[A_\pi(s, a)]|$. Then equation 3.4.6 holds.

Note the relationship between the total variation divergence and the **KL divergence**: $D_{TV}(p \parallel q)^2 \leq D_{\text{KL}}(p \parallel q)$, we can get the following bound:

$$\eta(\tilde{\pi}) \leq L_\pi(\tilde{\pi}) + C D_{\text{KL}}^{\max}(\pi, \tilde{\pi}), C = \frac{2\epsilon\gamma}{(1 - \gamma)^2}. \tag{3.4.6.27}$$

Where $D_{\text{KL}}^{\max}(\pi, \tilde{\pi}) = \max_s D_{\text{KL}}(\pi(\cdot|s) \parallel \tilde{\pi}(\cdot|s))$..

From this theorem we can propose an algorithm:

Algorithm 1 Approximate policy iteration algorithm guaranteeing non-increasing expected cost η

Initialize π_0.

for $i = 0, 1, 2, \ldots$ until convergence **do**

Compute all advantage values $A_{\pi_i}(s, a)$.

Solve the constrained optimization problem

$$\pi_{i+1} = \arg\min_{\pi} \left[L_{\pi_i}(\pi) + \left(\frac{2\epsilon\gamma}{(1-\gamma)^2} \right) D_{\mathrm{KL}}^{\max}(\pi_i, \pi) \right]$$

where $\epsilon = \max_s \max_a |A_\pi(s, a)|$

and $L_{\pi_i}(\pi) = \eta(\pi_i) + \sum_s \rho_{\pi_i}(s) \sum_a \pi(a|s) A_{\pi_i}(s, a)$

end for

Figure 3.52: Approximate policy iteration algorithm guaranteeing non-increasing expected cost η

Trust region policy optimization is an approximation to the algorithm above. When we parameterize the policy π using θ, we can solve:

$$\text{minimize} \left[L_{\theta_{\text{old}}}(\theta) + C D_{\mathrm{KL}}^{\max}(\theta_{\text{old}}, \theta) \right] \tag{3.4.6.28}$$

to improve the real eta. But if we use C derived from theory, the step size would be very small. So we use a constraint on the KL divergence to keep the optimal solution within a **trust region**:

$$D_{\mathrm{KL}}^{\max}(\theta_{\text{old}}, \theta) \leq \delta. \tag{3.4.6.29}$$

What's more, this new constraint constrains the KL divergence in every state. We relax this by transforming the optimization to:

$$\min_{\theta} L_{\theta_{\text{old}}}(\theta) \tag{3.4.6.30}$$

$$\text{subject to } \overline{D}_{\mathrm{KL}}^{\rho\theta_{old}}(\theta_{\text{old}}, \theta) \leq \delta. \tag{3.4.6.31}$$

where $\overline{D}_{\mathrm{KL}}^{\rho\theta_{old}}(\theta_{old}, \theta) := \mathbb{E}_{s\sim\rho\theta_{old}} \left[D_{\mathrm{KL}}(\pi_{\theta_{old}}(\cdot|s) \parallel \pi_\theta(\cdot|s)) \right]$. In practice, we can estimate the objective and constraint based on the sample:

$$\begin{aligned} &\underset{\theta}{\text{minimize}}\, \mathbb{E}_{s\sim\rho_{\theta_{\text{old}}}} \left[\frac{\pi_\theta(a|s)}{\pi_{\theta_{old}}(a|s)} Q_{\theta_{\text{old}}}(s, a) \right] \\ &\text{subject to } \mathbb{E}_{s\sim\rho_{\theta_{\text{old}}}} [D_{\mathrm{KL}}(\pi_{\theta_{\text{old}}}(\cdot|s) \parallel \pi_\theta(\cdot|s))] \leq \delta. \end{aligned} \tag{3.4.6.32}$$

Proximal Policy Optimization: PPO

PPO, designed by [Schulman et al., 2017], has some of the benefits of TRPO, but is much simpler to implement. TRPO algorithm can fluctuate greatly when $\rho_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{\text{old}}}(a_t|s_t)}$ changes too quickly, so PPO proposes the *clip surrogate loss*:

$$L^{CLIP}(\theta) = \hat{\mathbb{E}}_t\Big[\min(\rho_t(\theta)\hat{A}_t, \text{clip}(\rho_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t)\Big] \tag{3.4.6.33}$$

There is another surrogate loss in the penalty style:

$$L^{KLPEN}(\theta) = \hat{\mathbb{E}}_t\left[\frac{\pi_\theta(a_t \mid s_t)}{\pi_{\theta_{\text{old}}}(a_t \mid s_t)}\hat{A}_t - \beta\,\text{KL}[\pi_{\theta_{\text{old}}}(\cdot \mid s_t), \pi_\theta(\cdot \mid s_t)]\right] \tag{3.4.6.34}$$

In practice, there is a simple way to adjust the hyperparameter β, compute $d = \hat{\mathbb{E}}_t[\text{KL}[\pi_{\theta_{\text{old}}}(\cdot \mid s_t), \pi_\theta(\cdot \mid s_t)]]$:

- If $d < d_{\text{targ}}/1.5, \beta \leftarrow \beta/2$;
- $d > d_{\text{targ}} \times 1.5, \beta \leftarrow \beta \times 2$.

Algorithm 1 PPO, Actor-Critic Style

for iteration=$1, 2, \ldots$ **do**
  for actor=$1, 2, \ldots, N$ **do**
    Run policy $\pi_{\theta_{\text{old}}}$ in environment for T timesteps
    Compute advantage estimates $\hat{A}_1, \ldots, \hat{A}_T$
  end for
  Optimize surrogate L wrt θ, with K epochs and minibatch size $M \leq NT$
  $\theta_{\text{old}} \leftarrow \theta$
end for

Figure 3.53: PPO, Actor-Critic Style

Chapter 4

Operations Management

4.1 Empirical Operations Management

[Roth, 2007] describes the evolution of empirical OM from 1980 to 2007, the author selects 12 profounding papers in this domain. [Brusco et al., 2017] reviewed the clustering methods applied in 6 OM journals.
[Choi et al., 2016], multi-methodological OM is advocated, which includes the empirical methodology.

> **Definition 4.1.0.1:** Multi-methodological OM
> an approach for OM research in which at least two distinct OM research methods are employed nontrivially to meet the research goals.

[Roth and Singhal, 2022] classified 75 papers as empirical among the top 200 cited papers in *POM*, these papers are mainly from 3 topical areas:

1. Responsibility Operations: covers environmental management, sustainability, and humanitarian efforts;
2. Supply Chain Management: bullwhip effect, risk management, supply chain finance;
3. Manufacturing Strategy and Quality Management.

Primary data (surveys, experiments, interviews) are used in these studies, followed by secondary data (public database, firm's data). Roth's suggestions:

1. For some topic which is very intuitive and not surprising, focus on the **size** of effect rather than sign;
2. Avoid the confirmation bias and focus on consistency;
3. Focus on endogeneity and causality, using common sense simultaneously.

[Kumar and Tang, 2022] reviewed different domains of OM publications on *POM*, within which a section about empirical OM is covered.
[Mithas et al., 2022] reviewed 411 empirical papers form 2016-2021 on *POM*, with a causal inference and counterfactual perspective.

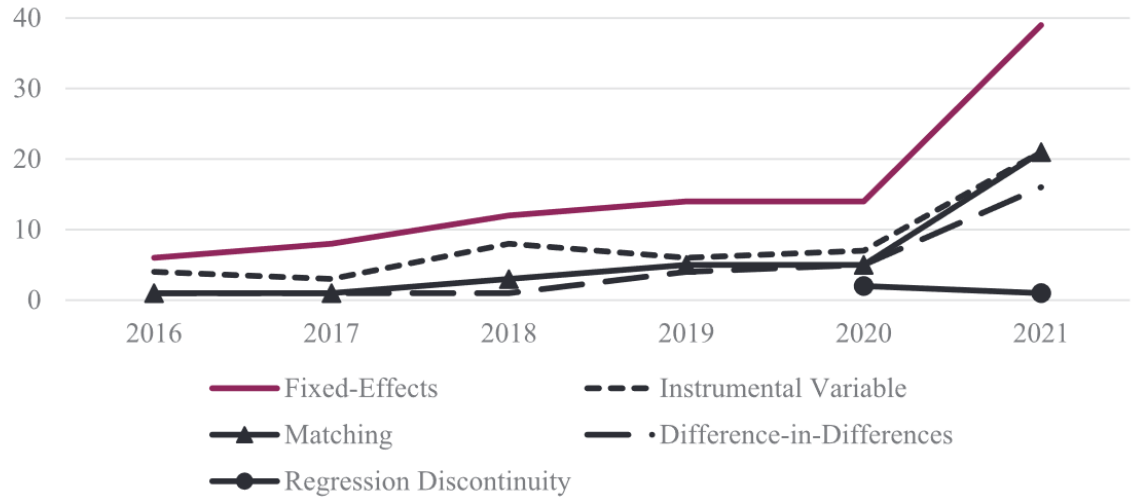


Figure 4.1: identification strategies from 2016 to 2021

Remark 4.1.1

Two challenges in assessing causality:

$$\begin{aligned} T = ATE &+ [E(Y(0)|Z=1) - E(Y(0)|Z=0)] \\ &+ (1-\pi) * [(ATT - ATU)]\,. \end{aligned} \tag{4.1.0.1}$$

- $[E(Y(0)|Z=1) - E(Y(0)|Z=0)]$ is the **baseline bias**, coming from OVB or *simultaneity*;
- $[(ATT - ATU)]$ is the **differential treatment bias**.

Approach	How baseline bias is addressed	How differential treatment effect bias is addressed	What is estimated	Limitations
1. Regression-based	Assumes strong ignorability (i.e., selection on observables only)	Ruled out by constant-coefficient models	ATE	Assumption of correct specification of the relationship between Y and the controls. No distinction between the treatment and covariates
2. Matching and weighting	Assumes strong ignorability (i.e., selection on observables only)	Possible to assess differential treatment effects across strata	ATT, ATU, or ATE (depends on the assumptions)	It is possible to conduct sensitivity analyses (e.g., Rosenbaum's gamma) for assessing selection bias and for the violation of the strong ignorability assumption
3. Instrumental variable (IV)	Relaxes the assumption of selection on observables. Exploits randomization induced by the IV	Ignores treatment heterogeneity by estimation for only a subgroup	LATE (Only for compliers or defiers, but not both)	Difficult to find strong and relevant IVs. Exclusion restriction (IV affects the outcome only via treatment) is not testable. Yields large standard errors if the sample sizes are small
4. Regression discontinuity (RD)	Sharp RD uses the assumption of selection on observables, but fuzzy RD relaxes that. Exploits the randomization induced by the cutoff score on the running variable	Ignores treatment heterogeneity by estimation for only a subgroup	LATE	Running variable perfectly (or fuzzily) determines the treatment assignment. Assumes no discontinuity in other factors that could also affect the outcome other than the treatment
5. Differences-in-differences (DID)	Exploits within-unit variation over time, assuming that all unobservables are time-invariant	Ignores treatment heterogeneity between ATT and ATU	ATT	Assumes parallel trends: Had there been no treatment, the trend in the outcomes would be parallel between the treated and the control. Mostly useful for sharp binary interventions. Requires longitudinal data on both the treated and control units
6. Fixed effects (FEs)	Exploits within-unit variation over time, assuming that all unobservables are time-invariant. Uses changes over time in the control group as a counterfactual for the changes in the treated group	Ignores treatment heterogeneity (assumes that the heterogeneity of the unit-specific causal effect across the population is random)	ATT	Needs strong assumptions or long time series for modeling the counterfactual trajectory. Assumes that past outcomes do not influence the treatment and no lagged treatment effects

Figure 4.2: How identification techniques works

[Fisher and Raman, 2022] especially investigate the empirical research in retail operations from traditional ones like forecasting and inventory planning, to new technologies, like radio frequency identification (RFID) and e-commerce.

4.1.1 Quantitative Marketing

Promotion Decompostion

[Leeflang et al., 2002] reviewed the development process of the SCAN*PRO (scanner and promotion) model, which was proposed by [Wittink et al., 1988]. The original version is given by:

$$q_{kjt} = \left[\prod_{r=1}^{n}\left[\frac{p_{krt}}{\overline{p}_{kr}}\right]^{\beta rj}\prod_{l=1}^{3}\gamma_{lrj}^{D_{lkn}}\right]\left[\prod_{t=1}^{T}\delta_{ji}^{X_t}\right]\left[\prod_{k=1}^{K}\lambda_{kj}^{Z_k}\right]e^{u_{kjt}} \tag{4.1.1.1}$$
$$k = 1,\ldots,K,\ t = 1,\ldots,T$$

Where: q_{kjt} is unit sales in store k for brand j at week t, p_{krt} is the unit price in store k for brand j at week t, $_{kr}isthemedianpriceatstorekforbrandrinnon-promotedweeks,andD_{lkrt}$ is the dummy variable indicating whether promotion activity l is launched in store k for brand r at time t. The terms X_t and Z_k can be interpreted as time and location fixed effects, and u_{kjt} is the error term. The above equation looks complicated, but if we use

log to transform it, we can get

$$\log q_{kjt} = \sum_{i=1}^{n} (\log p_{krt} - \log \bar{p_{kr}}) \cdot \beta_{rj} \cdot \sum_{l=1}^{3} \log \gamma_{lkj} D_{lkrt} + Fixed\_Effects + u_{kjt} \tag{4.1.1.2}$$

The β_{rj} is the price discount elasticities (own-brand for $r = j$, cross-brand if $r \neq j$), the γ_{jt} is the promotion multiplier. This model allows all brands to have unique own- and cross-brand effects for the marketing variables. Equation 4.1.1.1 does not account for dynamic promotion effects, which occur if promotions influence future demand. The **process function** of the own-brand discount elasticity can capture these effects:

$$\beta'_{kjt} = \beta'_{0j} + \beta'_{1j} Dsum_{kjt} + \beta'_{2j} CDsum_{kjt} + \beta'_{3j} d_\eta \left(\frac{1}{PTime_{kjt}} \right) + \beta'_{4j} d_\eta \left(\frac{1}{CPTime_{kjt}} \right) + u'_{kjt}, \tag{4.1.1.3}$$

where

- $Dsum_{kjt}$ is $\sum_{s=1}^{\omega} \eta^{s-1} \times (discount_{kj,t-s})$ in store k for brand j at week t, η is the decline rate;
- $CDsum_{kjt}$ is $\sum_{\substack{r=1 \\ r \neq j}}^{n} \sum_{s=1}^{\omega} \eta^{s-1} \times (discount_{kr,t-s})$;
- d_η is a dummy variable equal to 1 if $\eta = 1$ and 0 if $0 < \eta < 1$;
- $PTime_{kjt}$ is the number of weeks since the last own-brand promotion.

Besides decomposing the demand based on elasticity (gross effects)m, there is another decomposition scheme based on the unit sales (net effects). [Van Heerde et al., 2004] is an example of the net effects decomposition. In the standard sales-based decomposition, the own-brand sales increase can be split into three parts: cross-brand, crossperiod, and category-expansion effects. Denote S_{ijt} as the sales volume at store i for brand j at time t, and CS_i as the average sales at store i. Define:

$$\text{OBS}_{ijt} = -\frac{S_{ijt}}{CS_i}, \quad \text{CBS}_{ijt} = \sum_{k=1}^{l} \frac{S_{ikt}}{CS_i},$$

$$\text{PPCS}_{it} = \sum_{s=-T^*}^{l} \sum_{k=1}^{l} \frac{S_{ikt+s}}{CS_i}, \quad \text{TCS}_{it} = -\sum_{s=-T^*}^{l} \sum_{k=1}^{l} \frac{S_{ikt+s}}{CS_i}.$$

Where OBS_{ijt} is the own-brand sales, CBS_{ijt} is the cross-brand sales, $PPCS_{it}$ is the Pre- and Post category sales, and TCS_{it} is the total category sales. By construction, the following equation holds:

$$\text{OBS}_{ijt} = \text{CBS}_{ijt} + \text{PPCS}_{it} + \text{TCS}_{it}. \tag{4.1.1.4}$$

For each variable given above, the firm can use a linear system to estimate the net effects:

$$\begin{aligned}
\mathrm{X}_{ijt} = \alpha'_j &+ \sum_{l=1}^{4} \beta_{\mathrm{ob},lj} \mathrm{PI}_{ijlt} + \sum_{l=1}^{4} \gamma'_{1,lj} \mathrm{CPI}_{ijlt} + \sum_{m=1}^{3} \gamma'_{2,mj} \mathrm{D}_{ijmt} \\
&+ \sum_{m=1}^{3} \gamma'_{3,mj} \mathrm{CD}_{ijmt} + \gamma'_{4j} \mathrm{RP}_{ijt} + \gamma'_{5j} \mathrm{CRP}_{ijt} \\
&+ \sum_{\tau=T+T^*+1}^{T_{\max}-T-T^*} \gamma'_{6,\tau j} W_t + \sum_{\tau=1}^{T_+T^*} \gamma'_{7,\tau j} \mathrm{PI}_{ijlt+\tau} \\
&+ \sum_{\tau=1}^{T+T^*} \gamma'_{8,\tau j} \mathrm{PI}_{ijlt-\tau} + \sum_{\tau=1}^{T+T^*} \gamma'_{9,\tau j} \mathrm{CPI}_{ijlt+\tau} \\
&+ \sum_{\tau=1}^{T+T^*} \gamma'_{10,\tau j} \mathrm{CPI}_{ijlt-\tau} + u'_{ijt},
\end{aligned} \tag{4.1.1.5}$$

where PI_{ijlt} and CPI_{ijlt} is the own-brand price index and cross-brand price index among the same category considering support l, D_{ijmt} is the non-price promotion considering type m, RP_{ijt} is the regular price. To allow the interactions between different products, the semiparametric SCAN*PRO-model was proposed:

$$\begin{aligned}
&\ln q_{kjt} = m(\ln(PI_{k1t}), \ln(PI_{k2t}), ..., \ln(PI_{knt}) + \sum_{l=1}^{3} \sum_{r=1}^{n} \gamma''_{lrj} D_{lkrt} + \delta''_{ji} X_t + \lambda''_{kj} Z_k + u''_{kjt} \\
&\qquad t = 1, \ldots, T \text{ and } k = 1, \ldots, K
\end{aligned} \tag{4.1.1.6}$$

where $m(\cdot)$ is a nonparametric function.

Online Advertising

[Dinner et al., 2014] studied the croos-channel effects between online and offline advertising. They verified that the cross-channel effect exists, but it could be diminished by an indirect effect manifested by its impact on paid search impressions and click-through rate.

4.2 Supply Chain

4.2.1 Manufacturing

[Chen et al., 2023a] investigates the impact of "quick response" of the fashion product on the supply chain under different demand stochasticity. They assume there are two periods, and each period has an identical demand function $d_i = A_i - p_i (i = 1, 2)$, where A_i is the market size:

$$A_i = \begin{cases} H = 1 + \sigma & \text{with probability } 0.5, \\ L = 1 - \sigma & \text{with probability } 0.5. \end{cases} \tag{4.2.1.1}$$

σ is the standard deviation of A_i and a measure on the demand uncertainty. It is assumed that A_1 and A_2 are positively and perfectly correlated. There are three scenarios for the manufacturer to decide:

1. **N**: no quick response;
2. **D**: quick response with dynamic wholesale pricing;
3. **C**: quick response with committed wholesale pricing.

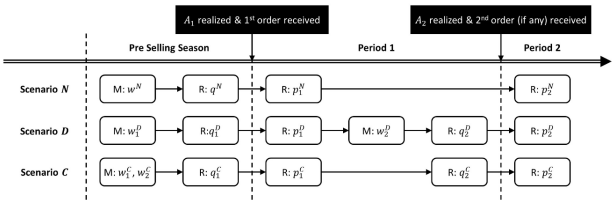


Figure 4.3: Sequence of Events

To analyze the effect of "quick response" on the supply chain (compare scenarios N and D), the authors give 3 intermediate scenarios α, β, γ and extract 4 effects. The results show that "quick response" may harm the retailer when demand uncertainty is in a medium range.

[Dong et al., 2022a] studies the effects of the auditing systems mode (decentralized v.s. centralized) in the food supply chain on the food safety outcome and economical payoff. In the decentralized mode, there are four parties in the game:

1. At stage 1, the upstream agency decides on its auditing action $\mathcal{A}_1 \in \{\mathcal{Y}, \mathcal{N}\}$, there is an auditing cost c_1 if $\mathcal{A}_1 = \mathcal{Y}$;
2. At stage 2, the upstream supplier decides on the safety level of the raw material $\mathcal{A}_2 \in \{\mathcal{S}, \mathcal{R}\}$, there is an extra cost Δc_2 if $\mathcal{A}_2 = \mathcal{S}$;
3. At stage 3, the downstream agency decides on its auditing action $\mathcal{A}_3 \in \{\mathcal{Y}, \mathcal{N}\}$, there is an auditing cost c_3 if $\mathcal{A}_3 = \mathcal{Y}$;
4. At stage 4, the downstream producer decides on the safety level of the food $\mathcal{A}_4 \in \{\mathcal{S}, \mathcal{R}\}$, there is an extra cost Δc_4 if $\mathcal{A}_4 = \mathcal{S}$.

Each time the agency chooses to audit, there would be a probability p_1, p_3 of the TP rate. When the unsafe food enters the market, four agents would face penalties k_1, k_2, k_3 and k_4. The centralized mode follows a similar modeling process. The authors deduce the SPME under different parameter settings, the paper identifies the driving forces behind equilibrium decisions:

- The *penalty-shield* effect of the downstream agency may lead to risky behavior by the upstream supplier.
- A *free-riding* effect between the two agencies may lead to less auditing.

The results show that the centralized auditing system may generate inferior performance.
[Dong et al., 2022b] investigate the assortment strategy under the usage of 3D printing. The model assumes that the assortments can be selected from two sets: the generic set $\mathcal{G}$ and the 3D-specific set $\mathcal{S}$. The manufacturer decides on three production sets: the dedicated technology set $\mathcal{D} \subset \mathcal{G}$, the traditional flexible set $\mathcal{T} \subset \mathcal{G}$, and the 3D printing set $\mathcal{P} \subset \mathcal{G} \cup \mathcal{S}$. Then the authors use multinomial-logit model to characterize the market of each product and separate the investment cost to *adoption cost*, *development cost*, and *capacity cost*. The paper shows the assortment structure: based on the popularity of the products. The Numerical study shows that 3D printing tends to be more valuable when popularities of the generic designs are distributed more evenly and when popularities of the 3D-specific designs are distributed less evenly.

4.2.2 Distribution Channel

[Kouvelis and Shi, 2020] analytically study the game consisting of three independent parties: a manufacturer, a retailer, and a sales agent. They give the equilibrium solution for the parties' decision to choose the mode of the value chain under different parameter settings. Key modeling framework: The manufacturer has a unit cost $c \geq 0$, and consumers have two valuations on the product: $r(r > c)$ or 0. The efforts of the sales agent can make the probability of the consumers' non-negative valuation higher.

4.3 Revenue Management

Revenue management is a data-driven system to price perishable assets tactically at the micro-market level to maximize expected revenue or profit. Reference: [Gallego and Topaloglu, 2019], [Strauss et al., 2018], [Klein et al., 2020].

4.3.1 Traditional RM

Keywords 4.1

- Protection Level, Booking Limit, Littlewood's Rule

Assumption 4.3.1.1: What does **"traditional"** means in RM?

1. The traditional RM system doesn't consider the choice model, in particular, it assumes the demands are independent random variables;
2. Further assumption: the consumer will leave without purchasing if the preferred fare class is unavailable (holds when gaps in fares are large enough);
3. The capacity is fixed, the capacity's marginal profit is zero(can be relaxed);
4. All booked consumers would arrive (another circumstance see 4.3.1).

Single Resource RM with the-worst-case Arrival Pattern

Assumption 4.3.1.2: Single Resource RM

1. The units of capacity is c, pricing at multiple different level $p_n < \cdots < p_1$;
2. Low-before-high fare class arrival order: D_2 before D_1 for example (this is the worst case for revenue);
3. **Protection level** for customer j: leave $y \in \{0, 1, \cdots, c\}$ for $D_{j-1}, , D_1$; $c - y$ is the **booking limit** which serves D_j;

😎 So the problem is to solve the optimal protection level given the current consumer level j.

Let $V_j(x)$ be the optimal revenue given D_j coming in, x units remained. $V_0(x) = 0$ by design. Let y be the protection level for $D_{j-1}, \cdots, D_1$: sales at $p_j = \min\{x - y, D_j\}$. The remaining capacity for $D_{j-1}, \cdots, D_1$ is $x - \min\{x - y, D_j\} = max\{y, x - D_j\}$. Now let $W_j(y, x)$ be the optimal solution. We have:

$$W_j(y, x) = p_j \mathbb{E}\{\min\{x - y, D_j\}\} + \mathbb{E}\{V_{j-1}(\max\{y, x - D_j\})\} \tag{4.3.1.1}$$

$$V_j(x) = \max_{y \in \{0,\ldots,x\}} W_j(y, x) = \max_{y \in \{0,\ldots,x\}} \left\{p_j \mathbb{E}\{\min\{x - y, D_j\}\} + \mathbb{E}\{V_{j-1}(\max\{y, x - D_j\})\}\right\} \tag{4.3.1.2}$$

Proposition 4.3.1.1: Structure of the Optimal Policy

$$y^*_{j-1} = \max\{y \in \mathbb{N}_+ : \Delta V_{j-1}(y) > p_j\}. \tag{4.3.1.3}$$

The maximizer of $W_j(y, x)$ is given by $y^*_j, , y^*_1$

Remark 4.3.1

The optimal solution for $y_j$is independent of the distribution of D_j;

Corollary 4.3.1.1: When $j = 2$:

Theorem 4.3.1.1: Littlewood's rule

$$y_1^* = \max\{y \in \mathbb{N}_+ : \mathbb{P}\{D_1 \geq y\} > r\}. \tag{4.3.1.4}$$

Remark 4.3.2

The Littlewoods Rule:

1. The solution depends on the **fare ratio**: $r := p_2/p_1$;
2. When the distribution of D_2 is continuous: $F_1(y) = \mathbb{P}\{D_1 \leq y\}$. The optimal protection level is $y_1^* = F_1^{-1}(1-r) = \mu_1 + \sigma_1 \phi^{-1}(1-r)$:
   (a) if $r > \frac{1}{2}$, $y_1^* < \mu_1$ and y_1^* decreases with σ_1;
   (b) if $r < \frac{1}{2}$, $y_1^* < \mu_1$ and y_1^* increases with σ_1;
   (c) if $r = \frac{1}{2}$, $y_1^* = \mu_1$;
3. Using the Littlewoods rule would result in some D_1 served by competitors (high spill rates). Solution: add a penalty to save more seats for the high-fare consumers:

$$y_1^* = \max\left\{y \in \mathbb{N}_+ : \mathbb{P}\{D_1 \geq y\} > \frac{p_2}{p_1 + \rho}\right\} \tag{4.3.1.5}$$

Multi-fare Heuristics: The EMSR (expected marginal seat revenue) is a set of heuristic algorithms. The EMSR-a algorithm is based on the idea of adding protection levels produced by applying Littlewood's rule to each pair of fare classes. Suppose that we are at stage j and we need to decide the protection level for fare classes $j-1, j-2, \cdots, 1$. We can apply the Littlewood's rule on fare class $k(k < j)$:

$$y_{kj}^* = \max\left\{y \in \mathbb{N}_+ : \mathbb{P}\{D_k \geq y\} > \frac{p_j}{p_k}\right\}. \tag{4.3.1.6}$$

The overall protection level at stage j is given by $y_{j-1}^a := \sum_{k=1}^{j-1} y_{kj}^*$.
The EMSR-b algorithm simply aggregate the fare classes $j-1, \cdots, 1$ into one class, the demand is denoted as $D[1, j-1]$, and the aggregated price is defined as:

$$\bar{p}_{j-1} = \sum_{k=1}^{j-1} p_k \frac{\mu_k}{\mu[1, j-1]}. \tag{4.3.1.7}$$

The algorithm uses the Littlewood's rule to solve the two-fare class problem:

$$y_{j-1}^b = \max\left\{y \in \mathbb{N}_+ : \mathbb{P}\{D[1, j-1] \geq y\} > \frac{p_j}{\bar{p}_{j-1}}\right\} \tag{4.3.1.8}$$

The Upper Bound of $V_n(c)$

Assume the demand vector $D = (D_n, \ldots, D_1)$ is known in advance, we can transform the problem into a knapsack problem:

$$\bar{V}(c \mid D) := \max\left\{\sum_{j=1}^{n} p_j x_j : \sum_{j=1}^{n} x_j \le c, 0 \le x_j \le D_j \forall j = 1, \ldots, n\right\}. \tag{4.3.1.9}$$

This problem has an explicit solution, which is the upper bound for $V_n^U(c)$:

$$\bar{V}(c \mid D) = \sum_{j=1}^{n} p_j \min\{D_j, (c - D[1, j-1])^+\}, \tag{4.3.1.10}$$

where the term $(c - D[1, j-1])^+$ is the remaining capacity after the reserved capacity for the fare class $j-1, \cdots, 1$. Taking the expectation on the capacity, we have:

$$\begin{aligned} V_n^U(c) &= \sum_{j=1}^{n} p_j \mathbb{E}\{\min\{D_j, (c - D[1, j-1])^+\}\} \\ &= \sum_{j=1}^{n} p_j \left(\mathbb{E}\{\min\{D[1, j], c\}\} - \mathbb{E}\{\min\{D[1, j-1], c\}\}\right) \\ &= \sum_{j=1}^{n} (p_j - p_{j+1}) \mathbb{E}\{\min\{D[1, j], c\}\}, \\ &= \sum_{j=1}^{n} (p_j - p_{j+1}) \sum_{k=1}^{c} \mathbb{P}\{D[1, j] \ge k\} \end{aligned} \tag{4.3.1.11}$$

where we define $p_{n+1} \equiv 0$. Recall that in $\bar{V}(c \mid D)$, D are random variables. Since $\bar{V}(c \mid D)$ is concave, by applying Jensen's inequality, we can obtain a more tractable upper bound:

$$\begin{aligned} \bar{V_n}(c) &= \max\left\{\sum_{j=1}^{n} p_j x_j : \sum_{j=1}^{n} x_j \le c, \quad 0 \le x_j \le \mu_j = 1, \ldots, n\right\} \\ &= \sum_{j=1}^{n} (p_j - p_{j+1}) \min\{\mu[1, j], c\} \end{aligned} \tag{4.3.1.12}$$

The Lower Bound of $V_n(c)$

The lower bound of the revenue management system is simply applying zero protection level, at the worst case, the bound is given by:

$$\begin{aligned} V_n^L(c) &= \sum_{j=1}^{n} p_j \mathbb{E}\{\min\{D_k, (c - D[j+1, n])^+\}\} \\ &= \sum_{j=1}^{n} p_j (\mathbb{E}\min\{D[j, n], c\}\} - \mathbb{E}\min\{D[j+1, n], c\}\}) \\ &= \sum_{j=1}^{n} (p_j - p_{j-1}) \mathrm{E}\{\min\{D[j, n], c\}\}, \end{aligned} \tag{4.3.1.13}$$

where we define $p_0 \equiv 0$.

Proposition 4.3.1.2

For the multiple fare class problem, we have:

$$V_n^L(c) \le V_n(c) \le V_n^U(c) \le \bar{V}_n(c). \tag{4.3.1.14}$$

Single Resource RM with General Arrival Pattern

In this model, we consider the effects of time. Assume the horizon is T, and time t represents the time left until the end. Let M_t denote the assortment at time t. Assume the fare class j arrive according to a Poisson process with λ_{jt}, and let N_{jt} denote the number of consumers that arrive during the last t units of time, which is Poisson with parameter:

$$\Lambda_{jt} := \int_0^t \lambda_{js} 1(j \in M_s) ds. \tag{4.3.1.15}$$

Now, refine $V(x)$ to $V(t,x)$ as the maximum expected revenue given capacity x and time t. In a short time window δt, the probability that there is one request for class j is $\lambda_{jt}\delta t + o(\delta t)$, and the probability that there are no requests is $1 - \sum_{k\neq j} \lambda_{kt}\delta t + o(\delta t)$. Now we can have:

$$\begin{aligned} V(t,x) =& \sum_{j\in M_t} \lambda_{jt}\delta t \max\{p_j + V(t-\delta t, x-1), V(t-\delta t, x)\} \\ &+ \left(1 - \sum_{j\in M_t} \lambda_{j_t}\delta t\right) V(t-\delta t, x) + o(\delta t) \\ &= V(t-\delta t, x) + \delta t \sum_{j\in M_t} \lambda_{jt}[p_j - \Delta V(t-\delta t, x)]^+ + o(\delta t), \end{aligned} \tag{4.3.1.16}$$

where $\Delta V(t,x) = V(t,x) - V(t,x-1)$. Subtracting $V(t-\delta t, x)$ from both sides and dividing by δt, and setting $\delta t \to 0$, we can get the HJB (Hamilton–Jacobi–Bellman) equation:

$$\begin{aligned} &\frac{\partial V(t,x)}{\partial t} = \mathcal{R}_t(\Delta V(t,x)). \\ &\mathcal{R}_t(z) := \sum_{j\in M_t} \lambda_{jt} \left[p_j - z\right]^+ \end{aligned} \tag{4.3.1.17}$$

We can see that fare j is accepted iff $p_j \ge \Delta V(t,x)$. So if we accept fare j at state (t,x), we should accept all fare classes $k < j$. We can define:

$$a(t,x) := \max\{j : p_j \ge \Delta V(t,x)\} \tag{4.3.1.18}$$

Theorem 4.3.1.2: The structure of $V(t,x)$

The value function $V(t,x)$ is increasing in t and x. $\Delta V(t,x)$ is decreasing in x. If the arrival rate λ_{jt} is stationary across time, $V(t,x)$ is strictly increasing and concave in t.

Discrete-Time Formulation

In the discrete setting, we can set $\delta t = 1$, and drop the $\wr(\delta t)$ term. By setting the interval density $k > 1$, and

letting $\lambda_{jt} \leftarrow \frac{1}{k}\lambda_{j,t/k}$, we can have:

$$\begin{aligned} V(t,x) &= \sum_{j\in M_t} \lambda_{jt} \max\{p_j + V(t-1,x-1), V(t-1,x)\} \\ &\quad + \left(1 - \sum_{j\in M_t} \lambda_{jt}\right) V(t-1,x) \\ &= V(t-1,x) + \sum_{j\in M_t} \lambda_{jt}[p_j - \Delta V(t-1,x)]^+ \\ &= V(t-1,x) + \mathcal{R}_t(\Delta V(t-1,x)) \end{aligned} \tag{4.3.1.19}$$

Network Revenue Management with Independent Demands

In the airline RM vocabulary, the **flight leg** refers to a resource and the ODF (**origin-destination-fare**) refers to the product. In the traditional model, it is often assumed that the set of possible ODFs is given and the demand for each ODF is independent of the demand for other ODFs.

Formulations of Dynamic Programming

Assume there are m resources (flight legs) in the network and let $c^2 := (c_1, \ldots, c_m)$ denote the capacity. We measure time backward (start at $t = T$ and end at $t = 0$). We index each ODF (itinerary) by $1, \cdots, K$, and the price levels for ODF k by p_{kj} for $j \in \{1, \ldots, n_k\}$. Let λ_{tkj} denote the arrival rate at time t of customers interested in ODF kj, and A_k be the resource vector (with dimension of m). We use the dummy u_{kj} to denote the decision variable: whether we accept a request for ODF kj, the feasible set is constrained by the remained capacity x: $A_k \cdot u_{kj} \le x$. Given a short enough period δt such that $\sum_{k=1}^{K}\sum_{j=1}^{n_k} \lambda_{tkj}\delta t \ll 1$, we can write the DP as:

$$\begin{aligned} V(t,x) = &\sum_{k=1}^{K}\sum_{j=1}^{n_k} \lambda_{tkj}\delta t \max_{u_{kj}\in\{0,1\}} [p_{kj}u_{kj} + V(t-\delta t, x - A_k u_{kj})] \\ &+ \left\{1 - \sum_{k=1}^{K}\sum_{j=1}^{n_k} \lambda_{tkj}\delta t\right\} V(t-\delta t, x) + o(\delta t), \end{aligned} \tag{4.3.1.20}$$

where $o(\delta t)$ comes from the property of Poisson process. Subtracting $V(t-\delta t, x)$ from both sides and dividing by δt, and using the notation $\Delta_k V(t,x) = V(t,x) - V(t, x - A_k)$, we obtain the HJB equation:

$$\frac{\partial V(t,x)}{\partial t} = \sum_{k=1}^{K}\sum_{j=1}^{n_k} \lambda_{tkj}[p_{kj} - \Delta_k V(t,x)]^+, \tag{4.3.1.21}$$

where the term $[p_{kj} - \Delta V_k(t,x)]^+$ is equivalent to the maximum of $p_{kj}u_{kj} + V(t, x - A_k u_{kj}) - V(t-\delta t, x)$. Let

$$R_t(u, \Delta V(t,x)) := \sum_{k=1}^{K}\sum_{j=1}^{n_k} \lambda_{tkj}[p_{kj} - \Delta_k V(t,x)]u_{kj}, \tag{4.3.1.22}$$

and consider the optimization problem:

$$\begin{aligned}\mathcal{R}_t(\Delta V(t,x)) &:= \max_u R_t(u, \Delta V(t,x)) \\ &= \sum_{k=1}^{K} \sum_{j=1}^{n_k} \lambda_{tkj} \max_{u_{jk}\in\{0,1\}} [p_{kj} - \Delta_k V(t,x)] u_{kj} \\ &= \sum_{k=1}^{K} \sum_{j=1}^{n_k} \lambda_{tkj} [p_{kj} - \Delta_k V(t,x)]^+.\end{aligned} \tag{4.3.1.23}$$

The optimal solution to the optimization is to accept all fares for itinerary k that exceed $\Delta_k V(t,x)$, and we can write the HJB equation as:

$$\frac{\partial V(t,x)}{\partial t} = \mathcal{R}_t(\Delta V(t,x)) \tag{4.3.1.24}$$

We can aggregate λ_{tkj} to $\lambda_{tk} = \sum_{j=1}^{n_k} \lambda_{tkj}$, and set $\lambda_{tkj} = \lambda_{tk} q_{tkj}$, where q_{tkj} follows the distribution P_{tk}. This yields:

$$\frac{\partial V(t,x)}{\partial t} = \sum_{k=1}^{K} \lambda_{tk} \, \mathrm{E}[P_{tk} - \Delta_k V(t,x)]^+ = \mathcal{R}_t(\Delta V(t,x)). \tag{4.3.1.25}$$

The Optimal Policy

In practice, for convenience **the single index** formulation is widely used. Let $n = \sum_{k=1}^{K} n_k$ be the number of ODFs, and $N = \{1, \ldots, n\}$ to be the set of all ODFs. The single index formulation is given by:

$$\frac{\partial V(t,x)}{\partial t} = \mathcal{R}_t(\Delta V(t,x)) = \sum_{j\in M_t} \lambda_{tj} [p_j - \Delta_j V(t,x)]^+ \tag{4.3.1.26}$$

In the optimal solution, we have:

$$u_j^*(t,x) = \begin{cases} 1 & \text{if } j \in M_t, A_j \le x, \text{and } p_j \ge \Delta_j V(t,x) \\ 0 & \text{otherwise,} \end{cases} \tag{4.3.1.27}$$

The LP Upper Bound

Theorem 4.3.1.3: The upper bound for $V(T,c)$

For $V(T,c)$ from problem 4.3.1.26, we have the bound:

$$V(T,c) \le \mathbb{E}[\bar{V}(T,c|D)] \le \bar{V}(T,c), \tag{4.3.1.28}$$

where the shaper bound comes from:

$$\begin{aligned}\bar{V}(T,c|D) := \max \quad & \sum_{j=1}^{n} p_j y_j \\ \text{s.t.} \quad & \sum_{j\in N} a_{ij} y_j \le c_i \qquad \forall i \in M \\ & 0 \le y_j \le D_j \qquad \forall j \in N.\end{aligned} \tag{4.3.1.29}$$

and the wider bound comes from:

$$\begin{aligned} \bar{V}(T,c) \;:=\; \max \; & \sum_{j\in N} p_j \; y_j \\ \text{s.t.} \; & \sum_{j\in N} a_{ij} \; y_j \le c_i \qquad \forall i \in M \\ & 0 \le y_j \le \Lambda_j \qquad \forall j \in N. \end{aligned} \tag{4.3.1.30}$$

The decision variable y_j corresponds to the number of requests for ODF j, and D_j is the aggregated demand for the ODF j, which is a Poisson with parameter $\Lambda_j = \int_0^T \lambda_{sj} ds$.

Proof 4.3.1: The upper bound for $V(T,c)$

For every instance of D, the optimal policy 4.3.1.27 constitute a feasible solution to program 4.3.1.29, so $V(T,c) \le \mathbb{E}[\bar{V}(T,c \mid D)]$.
For the second inequality, note that $\mathbb{E}[\bar{V}(T,c \mid D)]$ is concave on D, using Jensen's inequality:

$$\mathbb{E}[\bar{V}(T,c \mid D)] \le \bar{V}(T,c \mid \mathbb{E}[D]) = \bar{V}(T,c \mid \Lambda) = \bar{V}(T,c). \tag{4.3.1.31}$$

Q.E.D.

By strong duality, we can transform program 4.3.1.30 to:

$$\begin{aligned} \bar{V}(T,c) \;=\; \min \; & \sum_{i\in M} c_i z_i + \sum_{j\in N} \Lambda_j \beta_j \\ \text{s.t.} \; & \sum_{i\in M} a_{ij} z_i + \beta_j \ge p_j \quad \forall j \in N \\ & z_i \ge 0, \; \beta_j \ge 0 \quad \forall i \in M, \; j \in N. \end{aligned} \tag{4.3.1.32}$$

At optimality, the dual variable z_i^* is the marginal value of capacity of resource i, while the dual variable β_j^* is the marginal value of demand for ODF j. Setting $\beta_j = (p_j - \sum_{i\in M} a_{ij} z_i)^+$ so that we can optimize on z only. Consequently:

$$\sum_{j\in N} \Lambda_j \beta_j = \sum_{j\in N} \Lambda_j (p_j - \sum_{i\in M} a_{ij} z_i)^+ = \int_0^T \mathcal{R}_t(A'z) dt, \tag{4.3.1.33}$$

$$\bar{V}(T,c) = \min_{z\ge 0} \left\{ \int_0^T \mathcal{R}_t(A'z) dt + c'z \right\}. \tag{4.3.1.34}$$

This provides a way to approximate $\Delta V(T,c)$.

Heuristics for the Network RM

Suppose $\{z_i^* : i \in M\}$ from the dual program 4.3.1.32 is known. The **bid-price heuristic** policy is: making ODF j available iff $A_j \le x$ and $p_j \ge \sum_{i\in M} a_{ij} z_i^*$.
Suppose$y = (y_1^*, \ldots, y_n^*)$ from the dual program 4.3.1.30 is known. The **probabilistic admission control heuristic** (PAC) policy is: making PDF j available with probability $\frac{y_j^*}{\Gamma_j}$ whenever $A_j \le x$.
Comparing the two heuristic policies, the bid-price uses m parameters (resources) while the PAC uses n parameters (ODF). Usually, $m \ll n$, so bid-price heuristics is more widely used.

Remark 4.3.3

- The bid-price heuristic is NOT asymptotically optimal;
- PAC is asymptotically optimal;
- However, the PAC does not perform as well as the bid-price heuristic with frequent updates over the horizon.

Upgrades and Upsells

Let U_j be the set of products to fulfill a request j: the customers are willing to take any product $k \in U_j$ at the price p_j. The value function with upgrades is given by:

$$\frac{\partial V(t,x)}{\partial t} = \sum_{j \in \mathcal{M}_t} \lambda_j \max_{k \in U_j} (p_j - \Delta_k V(t,x))^+ = \sum_{j \in \mathcal{M}_t} \lambda_j (p_j - \tilde{\Delta}_j V(t,x))^+ \tag{4.3.1.35}$$

where

$$\tilde{\Delta}_j V(t,x) = \min_{k \in U_j} \Delta_k V(t,x). \tag{4.3.1.36}$$

In the upsell model, assume that r_{jk} is the revenue by upselling product $k \in U_j$ ($r_{jk} > p_j$), $and there is a probability \pi_{jk}$ that the customer accepts the upsell. This leads to the upsell HJB equation:

$$\frac{\partial V(t,x)}{\partial t} = \sum_{j \in M_t} \lambda_j \max_{k \in U_j} \left[\pi_{jk} (r_{jk} - \tilde{\Delta}_k V(t,x))^+ + (1 - \pi_{jk})(p_j - \tilde{\Delta}_j V(t,x))^+ \right] \tag{4.3.1.37}$$

Overbooking

Suppose that the flight capacity is c, and the demand is D. Assume that passengers who do not show up are given a full refund and that the unit cost for denied boarding is θ. Let b be the booking limit, then $N := min(D, b)$ is the number of bookings. The objective is to maximize:

$$R(b) := p\mathbb{E}[Z(\min(D,b))] - \theta\mathbb{E}[Z(\min(D,b)) - c]^+]. \tag{4.3.1.38}$$

The optimal b^* is given by:

$$b^* = \min \left\{ b \geq 0 : \mathbb{P}\{Z(b) \geq c\} > \frac{p}{\theta} \right\}. \tag{4.3.1.39}$$

Fairness in Airline RM

[Aslani et al., 2014] summarize some unfair pricing phenomena in revenue management, such as the hidden-city ticketing and throwaway ticketing. [Wang and Ye, 2016] studies the cause and impact of the hidden-city ticketing:

Assumption 4.3.1.3: [Wang and Ye, 2016]

1. Airlines only provide itineraries with at most one stop. For an O-D (orientation-destination) pair $i \to j$, we use the notation $(k_{ij}^t, p_{ij}^t)_{(i,j) \in O}$, where k_{ij}^t is the connection city ($k_{ij}^t = 0$ indicates direct flight), and p_{ij}^t is the price;
2. For a passenger with O-D pair $i \to j$, he only considers k_{ij}^t without considering p_{ij}^t. Denote $\lambda_{ij}^t(p_{ij}^t)$ as the probability of purchase. We assume that $\lambda_{ij}^t(p_{ij}^t)$ is known in advance and it is small enough so that at each period, the probability that there are more than one purchases can be ignored.

3. For any i, j, t: $\lambda_{ij}^t(p)$ is continuously differentiable and non-increasing in p;

4. For any $c > 0$, $\lambda_{ij}^t(p)(p-c)$ is quasiconcave in p and there exists a unique maximizer for $\lambda_{ij}^t(p)(p-c)$.

For one particular airline company, denote the capacity of the flight leg using vector $\boldsymbol{x}$, and the optimal expected revenue as $V^t(\boldsymbol{x})$, we can formulate the dynamic programming:

$$\begin{aligned} V^t(\boldsymbol{x}) &= \max_{k_{ij}^t, p_{ij}^t, \forall (i,j)\in O} \left\{ \sum_{(i,j)\in O} \lambda_{ij}^t(p_{ij}^t) \left(p_{ij}^t + V^{t-1}(\boldsymbol{x} - A_{ij}^{k_{ij}^t}) \right) + \left(1 - \sum_{(i,j)\in O} \lambda_{ij}^t(p_{ij}^t) \right) V^{t-1}(\boldsymbol{x}) \right\} \\ &= V^{t-1}(\boldsymbol{x}) + \max_{k_{ij}^t, p_{ij}^t, \forall (i,j)\in O} \left\{ \sum_{(i,j)\in O} \lambda_{ij}^t(p_{ij}^t) \left(p_{ij}^t + V^{t-1}(\boldsymbol{x} - A_{ij}^{k_{ij}^t}) - V^{t-1}(\boldsymbol{x}) \right) \right\} \end{aligned} \tag{4.3.1.40}$$

where A_{ij}^k is the capacity consumption vector for each flight leg. With boundary conditions $V^0(\boldsymbol{x}) = 0 \quad \forall \boldsymbol{x}$, $V^t(0) = 0 \quad \forall t$ and $V^t(\boldsymbol{x}) = -\infty$ if $\boldsymbol{x}$ contains any negative entry. Note that we can solve the problem in two steps because the optimization over k_{ij}^t and p_{ij}^t are separated. We can first solve k_{ij}^t:

$$\hat{k}_{ij}^t = \arg\max_{k_{ij}^t} V^{t-1}(\boldsymbol{x} - A_{ij}^{k_{ij}^t}) \tag{4.3.1.41}$$

then we can solve the optimal p_{ij}^t by maximize:

$$\lambda_{ij}^t(p_{ij}^t) \left(p_{ij}^t + V^{t-1}(\boldsymbol{x} - A_{ij}^{\hat{k}_{ij}^t}) - V^{t-1}(\boldsymbol{x}) \right) \tag{4.3.1.42}$$

The Cause of Hidden City

We can define $c_{ij}^t = V^{t-1}(\boldsymbol{x}) - V^{t-1}(\boldsymbol{x} - A_{ij}^{\hat{k}_{ij}^t})$ as the opportunity cost of using capacity $A_{ij}^{\hat{k}_{ij}^t}$, then the optimal price for O-D pair $i \to j$ is determined by:

$$\hat{p}_{ij}^t = \arg\max_p \lambda_{ij}^t(p)(p - c_{ij}^t). \tag{4.3.1.43}$$

This is an easy problem, at the optimal $\hat{p_{ij}}$ (without considering t), we have:

$$1 - \frac{c_{ij}}{\hat{p}_{ij}} = -\frac{\lambda_{ij}(\hat{p}_{ij})}{\hat{p}_{ij}\lambda_{ij}'(\hat{p}_{ij})} = -(E_{ij}(\hat{p}_{ij}))^{-1} \tag{4.3.1.44}$$

Where $E_{ij}(p) = \frac{p\lambda_{ij}'(p)}{\lambda_{ij}(p)}$ is the price elasticity of demand at price p. Consider the trip $i \to j \to k$, (k is the hidden city), the prices should satisfy:

$$1 - \frac{c_{ij}}{p_{ij}} = -(E_{ij}(p_{ij}))^{-1} \quad \text{and} \quad 1 - \frac{c_{ik}}{p_{ik}} = -(E_{ik}(p_{ik}))^{-1}. \tag{4.3.1.45}$$

It's easy to verify that $c_{ij} \le c_{ik}$, combined with that $p_{ij} > p_{ik}$, we can obtain:

$$|E_{ij}(p_{ij})| < |E_{ik}(p_{ik})| \tag{4.3.1.46}$$

Proposition 4.3.1.3

If the hidden-city phenomenon $i \to j \to k$ happens, then the price elasticity of demand $i \to k$ is greater than that of $i \to j$ at the optimal price.

The authors later provide an algorithm that would produce no arbitrage opportunity for hidden-city ticketing. however, under this circumstance, all parties' welfare decreases compared to the baseline model.

4.3.2 Consumer Choice Model and Assortment Optimization

Assume the products that could be offered is $N := \{1, \cdots, n\}$, for any assortment $S \subseteq N$ and product j, we denote $\pi_j(S)$ as the probability that a consumer will select product $j \in S$. $\Pi(S) := \sum_{j \in S} \pi_j(S) = 0$indicates that the consumer does not purchase or purchases outside S. Denote p and z as n-dimensioanl vectors for prices and costs of each product. The decision maker should optimize a combinatorial problem:

$$\mathcal{R}(z) := \max_{S \subseteq N, p \geq z} R(S, p, z) := \sum_{j \in S} (p_j - z_j) \pi_j(S, p) \tag{4.3.2.1}$$

Most of the time, by setting $p_j \to \infty$, we can make $pi_j(N, j) = 0$ (remove j out of S). Thus, we can transform the problem above to:

$$\mathcal{R}(z) := \max_{p \geq z} R(p, z) := \sum_{j \in N} (p_j - z_j) \pi_j(N, p) \tag{4.3.2.2}$$

Basic Attraction Model (BAM)

Each $j \in N_+$ has an attraction value $v_j > 0$ (v_0 indicates no-purchase), the choice model is given by:

$$\pi_j(S) = \frac{v_j}{v_0 + V(S)} \quad \forall j \in S \tag{4.3.2.3}$$

BAM has a great property: for any $S \subseteq T$, we have $\pi_S(T) := \sum_{j \in S} \pi_j(T)$.

Axiom 4.3.2.1: The Luce Axiom

If $\pi_i(\{i\}) \in (0, 1), \forall i \in T$, then for any $Q \subseteq S_+, S \subseteq T$:

$$\pi_Q(T) = \pi_Q(S) \pi_{S_+}(T). \tag{4.3.2.4}$$

If $\pi_i(\{i\}) = 0$ for some $i \in T$, then for some $i \in T$ and $S \subseteq T$:

$$\pi_S(T) = \pi_{S \setminus \{i\}}(T \setminus \{i\}). \tag{4.3.2.5}$$

Remark 4.3.4

Shortage of BAM:

- v_0 is a fixed parameter and does not depend on the offered products;
- This approach ignores the possibility that the consumer may look for the products outside S at a later time.

Now consider the assortment optimization for BAM, the expected revenue when offering $S \subseteq N$ is:

$$R(S) = \sum_{j \in S} p_j \pi_j(S) = \frac{\sum_{j \in S} p_j v_j}{v_0 + V(S)}. \tag{4.3.2.6}$$

This equation neglects the cost, one can replace p_j by $r_j = p_j - z_j$. The goal is to find the S^*:

$$\mathcal{R}^* = \max_{S \subseteq N} R(S) = \max_{S \subseteq N} \left\{ \frac{\sum_{j \in S} p_j \ v_j}{v_0 + V(S)} \right\}. \tag{4.3.2.7}$$

Assume that the products are indexed such that $p_1 \geq p_2 \geq \ldots \geq p_n$, denote the class of nested-by-revenue assortments is $\{E_0, E_1, \ldots, E_n\}$, where $E_0 = \emptyset$ and $E_j := \{1, \ldots, j\}$.

Theorem 4.3.2.1: Optimal Assortment for BAM

An optimal assortment for BAM is in the class of nested-by revenue assortments $\{E_0, E_1, \ldots, E_n\}$.

The proof is straightforward since $R(S) \leq \mathcal{R}^*$, consequently $\sum_{i \in S} r_i v_i \leq \mathcal{R}^*(1 + V(S))$, equivalently $\sum_{i \in S}(r_i - \mathcal{R}^*)v_i \leq \mathcal{R}^*$, which implies that $S^* = \{i \in N : r_i \geq \mathcal{R}^*\}$.

😎 This theorem can reduce the number of assortments to search from 2^n to $\log n$, without estimating the value of v_i.

Generalized Attraction Model (GAM)

In addition to the attraction value v_j, there are **shadow attraction values** $w_j \in [0, v_j]$, and the selection behaviour is given by:

$$\pi_j(S) = \frac{v_j}{v_0 + W(\bar{S}) + V(S)} \quad \forall j \in S, \tag{4.3.2.8}$$

where $\bar{S} = N \setminus S$ and $W(\bar{S}) := \sum_{j \in \bar{S}} w_j$.

Remark 4.3.5

- If $w_j = 0$ for all products, it recovers to BAM;
- Parsimonious GAM (p-GAM) is given by $w_j = \theta v_j$ for all j;
- In p-GAM, if $\theta = 1$, we can get the *independent demand model* (IDM), under which $\pi_j(S)$ is independent of j.

The GAM has a similar optimal assortment structure to the BAM: denote $\tilde{r}_i := r_i v_i / (v_i - w_i)$, then $S^* = \{i \in N : \tilde{r}_i \geq \mathcal{R}^*\}$. However, to reduce the search number to $logn$, the estimation for v_i, w_i is needed.

Proposition 4.3.2.1: Independence of Irrelevant Alternatives (IIA)

IIA is a shortcoming for BAM and GAM, under which the following equation holds:

$$\frac{\pi_j(S)}{\pi_j(S \cup \{k\})} = \frac{\pi_i(S)}{\pi_i(S \cup \{k\})} \tag{4.3.2.9}$$

This shouldn't happen when k is a closer substitute for j than for i. This equation indicates that the ratio $\pi_i(S)/\pi_j(S) = \frac{v_i}{v_j}$, which is **independent** of the set S. IIA results in the red bus, blue bus paradox: $\pi_{rbus}(\{rbus, bbus\}) = \frac{1}{2}$, but $\pi_{rbus}(\{rbus, bbus1, bbus2\}) = \frac{1}{3}$.

Assortment for BAM/GAM with TUM constraints

In practice, the assortment optimization is usually a constrained optimization:

$$\begin{aligned} \mathcal{R}^* = \max \quad & \frac{\sum_{j\in N} p_j v_j x_j}{v_0 + \sum_{j\in N} v_j x_j} \\ \text{s.t.} \quad & \sum_{j\in N} a_{ij} x_j \le b_i \quad \forall i \in L \\ & x_j \in \{0,1\} \qquad \forall j \in N, \end{aligned} \tag{4.3.2.10}$$

The above problem is hard because of the fractional objective function and binary constraint. However, if the matrix a is TU (totally unimodular), It can be transformed into linear programming (This is called Cooper Transformation, rewrite $y_j = \frac{v_j x_j}{v_0 + \sum_{j\in N} v_j x_j}$).

Remark 4.3.6

A unimodular matrix M is a square integer matrix with determinant 1 or -1. It is an integer matrix that is invertible over another integer matrix. A matrix is totally unimodular if every square non-singular submatrix is unimodular. For every equation $Mx = b$, where M, b are integers, and M is totally unimodular, the solutions are all integers.

$$\begin{aligned} & \max \sum_{j\in N} r_j x_j \\ & \sum_{j\in N} x_j + x_0 = 1, \\ & 0 \le \frac{x_j}{v_j} \le x_0 \forall j \in N \\ & \sum_{j\in N} a_{ij} \frac{x_j}{v_j} \le b_i x_0 \forall i \in M. \end{aligned} \tag{4.3.2.11}$$

The x_j can be interpreted as the probability of selecting product j. The optimal structure for this optimization problem is to select $\{x_j : x_j > 0\}$.

Random Consideration Set Model

This choice model assumes that all consumers have the same preference for ordering, each item in the assortment is **independently** considered with 'attention' probability w_i. So the probability of purchasing i form S is given by:

$$\pi_i(S) = \lambda_i \prod_{j>i, j\in S} (1-\lambda_j) \quad \forall i \in S \tag{4.3.2.12}$$

Example 4.3.2.1

Consider an assortment with $S = \{1,2\}$ with $1 \prec 2$. The consideration set for $\{1\}$ is $w_1(1-w_2)$, $\{\emptyset\}$ is $(1-w_1)(1-w_2)$.

Solution:

The selection probability for product 2 *is* w_2, $w_1(1-w_2)$ *for product* 1, $(1-w_1)(1-w_2)$ *for selecting*

nothing.

Lemma 4.3.2.1

Let $S \subseteq N$, and let $k \in N$ be such that $i \prec k$ for all $i \in S$, then:

$$R(S \cup \{k\}) = (1 - \lambda_k)R(S) + \lambda_k r_k \tag{4.3.2.13}$$

and consequently $R(S \cup \{k\}) > R(S)$ iff $r_k > R(S)$. (The proof is obvious).

To optimize the assortment for the RCS, denote $E_0 = \emptyset$ and $E_j = \{1, \dots, j\}$ (consecutive sets in the full ranking). Set $H(E_0) := 0$, $H(E_j)$ is given by:

$$H(E_j) := H(E_{j-1}) + \lambda_j (p_j - H(E_{j-1}))^+. \tag{4.3.2.14}$$

Let $\tilde{E}_j := \{i \in E_j : p_i > H(E_{i-1})\}, \quad j \in N$. Hence $\mathbf{S}^* = \tilde{E}_n$ is an optimal assortment:

$$H(E_j) = R(\tilde{E}_j) \geq R(S) \ \forall S \subseteq E_j, \ j \in N, \tag{4.3.2.15}$$

Random Utility Models (RUM)

Consumers observe $v_i = u_i + \epsilon_i$ and choose product i with the highest v_i. The distribution of ϵ_i can lead to different choice models.

- i.i.d. Gaussians lead to the Probit model;
- i.i.d. Gumbels lead to the Multi-nomial model (MNL), which can be generated to avoid IIA.

Multi-Nomial Model (MNL)

The CDF of Gumbel distribution is given by:

$$F(x : \nu, \phi) = \exp(-\exp(-\phi^2(x - \nu))), \tag{4.3.2.16}$$

where ν and ϕ are location and scale parameters respectively. The variance is $\frac{\pi^2}{6\phi^2}$ ad the mean is $\nu + \frac{\gamma}{\phi}$ (γ is the Euler constant).

There is also a mixture of choice models: mixed-MNL aka latent class MNL (LC-MNL) assumes that the decision-maker belongs to an unobserved market segment $i \in M$ w.p. $\theta_i > 0, \sum_{i \in M} \theta_i = 1$. The probability of selecting j from S is given by:

$$\pi_j(S) = \sum_{i \in M} \theta_i \pi_j(S; i) \tag{4.3.2.17}$$

Theorem 4.3.2.2: McFadden

If ϵ_i are i.i.d. Gumbels, by choosing i with the highest u_i, we can get

$$\pi_j(S) = \frac{e^{\phi u_j}}{1 + \sum_{k \in S} e^{\phi u_k}} \quad \forall j \in S, \tag{4.3.2.18}$$

MNL is a kind of BAM. All RUM can be approximated well by an LC-MNL.

Nested Logit Model

In the nested model, there are two stages: first the consumers select either one of the nests or decide to leave; then they decide which product to choose from the selected nest. We use $M := \{1, \ldots, m\}$ to denote the set of nests. For product j in nest S_i, the probability of choosing i is given by:

$$q_{j|i}(S_i) := \frac{v_{ij}}{V_i(S_i)} = \frac{v_{ij}}{\sum_{j \in S_i} v_{ij}} \tag{4.3.2.19}$$

We use γ_i to denote how easily the products in nest i substitute for each other, so we can get:

$$Q_i(S_1, \ldots, S_m) := \frac{V_i(S_i)^{\gamma_i}}{v_0 + \sum_{l \in M} V_l(S_l)^{\gamma_l}} \tag{4.3.2.20}$$

$$Q_i(S_1, \ldots, S_m) q_{j|i}(S_i) = \frac{V_i(S_i)^{\gamma_i}}{v_0 + \sum_{l \in M} V_l(S_l)^{\gamma_l}} \frac{v_{ij}}{V_i(S_i)} \tag{4.3.2.21}$$

Remark 4.3.7

- When $\gamma_i = 1$ for all i, the Nested Lgoit model reduces to an MNL;
- The nested choice model doesn't suffer from IIA;
- But it's difficult to decide the order of nesting.

The assortment optimization for the nested logit model is to find the sets of products $(S_1, \ldots, S_m)$ to maximize:

$$R(S_1, \ldots, S_m) := \sum_{i \in M} R_i(S_i) Q_i(S_1, \ldots, S_m) = \frac{\sum_{i \in M} R_i(S_i) V_i(S_i)^{\gamma_i}}{v_0 + \sum_{i \in M} V_i(S_i)^{\gamma_i}}. \tag{4.3.2.22}$$

We can define the optimal solution as:

$$\mathcal{R}^* = \max_{\substack{(S_1, \ldots, S_m) \\ S_i \subseteq N \ \forall i \in M}} R(S_1, \ldots, S_m). \tag{4.3.2.23}$$

This is a binary fractional program if we identify each subset S_i with an incidence vector $x_i = (x_{i1}, \ldots, x_{in}) \in \{0,1\}^n$. Let $V_i(x_i) := \sum_{j \in N} v_{ij} x_{ij}$, $f_i(x_i) := R_i(x_i) V_i(x_i)^{\gamma_i}$ and $g_i(x_i) := V_i(x_i)^{\gamma_i}$. Then we have:

$$\mathcal{R}^* \geq \sum_{i \in M} f_i(x_i) / (v_0 + \sum_{i \in M} g_i(x_i)), \tag{4.3.2.24}$$

which can be rewritten as:

$$\sum_{i \in M} f_i(x_i) - \mathcal{R}^* \sum_{i \in M} g_i(x_i) \leq \mathcal{R}^* v_0, \forall x_i \in \{0,1\}^n, i \in M. \tag{4.3.2.25}$$

The inequality is tight at the optimal solution, we can transform it into a linear program:

$$\begin{aligned} \mathcal{R}^* = \min\ & z \\ \text{s.t.}\ & \sum_{i \in M} v_i \leq v_0 z \\ & f_i(x_i) - z g_i(x_i) \leq y_i \quad \forall x_i \in \{0,1\}^n, \quad i \in M, \end{aligned} \tag{4.3.2.26}$$

where z and y_i are decision variables $(m+1)$.

Properties of the Choice Model

There are two important properties for the MNL-based choice model: **regularity** and **submodularity**. Regularity means for any assortment S, adding a new item $j \notin S$ would reduce the probability of $P_i(S \cup j)$ for any $i \in S$ under the choice model.

Definition 4.3.2.1: Submodularity

Given a choice model P, let the demand function: $2^{\mathcal{J}} \mapsto \mathbb{R}$ of the choice model be given by $d(S) := \sum_{j \in S} P_{j:S}$ for any assortment $S \subset \mathcal{J}$. The intuitive understanding of submodularity is the marginal increment of new items on d is always decreasing:

$$d(S_2 \cup \{k\}) - d(S_2) \quad \leq \quad d(S_1 \cup \{k\}) - d(S_1), \quad \forall S_1 \subset S_2 \subset \mathcal{J}, k \in \mathcal{J} \setminus S_2. \tag{4.3.2.27}$$

Markov Chain Choice Model

In the preference list (permutation) model, each decision maker has a strict preference list σ, which has a distribution $\boldsymbol{Q}$. The choice probability is given by:

$$\lambda_i := \pi_i(N) = \sum_{\sigma} \boldsymbol{Q}(\sigma(1) = i) \tag{4.3.2.28}$$

If i is not available, consumers would substitute j w.p. $\rho_{ij} = \sum_{\sigma} \boldsymbol{Q}(\sigma(2) = j | \sigma(1) = i)$. In an example with $n = 2$ and $\lambda = (\lambda_0, \lambda_1, \lambda_2)$, the transition matrix is given by:

$$\rho = \begin{pmatrix} 1 & 0 & 0 \\ \rho_{10} & 0 & \rho_{12} \\ \rho_{20} & \rho_{21} & 0 \end{pmatrix} \tag{4.3.2.29}$$

We denote $\phi_j(S)$ the probability that a consumer considers product j but $j \notin S$, we have:

$$\pi_j(S) = \lambda_j + \sum_{i \in \bar{S}} \phi_i(S) \rho_{ij} \quad \forall j \in S \tag{4.3.2.30}$$

$$\phi_j(S) = \lambda_j + \sum_{i \in \bar{S}} \phi_i(S) \rho_{ij} \quad \forall j \in \bar{S}. \tag{4.3.2.31}$$

There is an efficient way to solve the assortment optimization problem under the MC choice model. Let g_i be the optimal expected revenue that can be obtained from a consumer that is currently considering product i:

$$g_i = \max \left\{ p_i, \sum_{i \in N} \rho_{ij} g_j \right\} \quad \forall i \in N. \tag{4.3.2.32}$$

If $i \in S$, we have $g_i = p_i$. Unfortunately, g_i doesn't have a closed form. One way to find the value of g_i is through value iteration, or we can solve it by LP:

$$\begin{aligned} \mathcal{R}^* = \min & \sum_{i \in N} \lambda_i g_i \\ \text{s.t.} \quad & g_i \geq p_i \quad \forall i \in N \\ & g_i \geq \sum_{i \in N} \rho_{ij} g_j \quad \forall i \in N \end{aligned} \tag{4.3.2.33}$$

Mention that under the optimal g^*, the expected revenue is given by $\sum_i g_i^* \lambda_i$. The optimal assortment is given by $S^* := \{i \in N : g_i^* = p_i\}$

Remark 4.3.8

The logic behind the LP is: if we want to find the expected revenue for a given S, we need to solve a similar LP, where the first inequality falls in $i \in S$ and the second inequality falls in $i \in \bar{S}$. Which takes exponential time if we want to solve $R(S)$ for all S. We can solve one LP to find the maximum $\mathcal{R}$.

Rank-1 Markov Chain

Theorem 4.3.2.3: Outer product representation of a rank-one matrix

Every rank-one matrix $A \in \mathbb{R}^{m \times n}$ can be written as an 'outer product':

$$A = pq^T,$$

where $p \in \mathbb{R}^m, q \in \mathbb{R}^n$. A Markov chain rank-one needs to satisfy $\sum_{j \neq i} p_j = 1/q_i$.

Remark 4.3.9

It can be shown that BAM, GAM and RCS are all rank-1 Markov Chain Models.

Heuristics in Assortment Optimization

The assortment optimization problem is usually NP-hard (such as mixed MNL), there are some heuristic algorithm to approximate $\mathcal{R}^*$.

Revenue-Ordered

The RO-assortment is to find τ and the corresponding assortment under τ: $S(\tau) := \{i \in N : r_i \geq \tau\}$. The optimal assortment is $\mathcal{R}^0 := \max_{\tau \geq 0} R(S(\tau))$, and it's proven that $\mathcal{R}^0 \geq \frac{1}{n}\mathcal{R}^*$.

The Refined Greedy Heuristic

[Nip et al., 2021] consider the competitive and cooperative assortment games under the Markov Chain Choice Model. They assume There are $\mathcal{M} = \{1, \ldots, m\}$ retailers, each manages a collection of $\mathcal{N}_k$ products. We call product j an *exclusive* product to k if $j \in \mathcal{N}_k \backslash \mathcal{N}_{-k}$, *common* otherwise. In the game, each retailer needs to dicide the assortment $S_k \subseteq \mathcal{N}_k$, and the prices r_j^k are assumed to be fixed. We assume the consumers follow the Markov Chain Choice Model to select the products: with probability $\lambda_j > 0$ and $\sum_{j=1}^n \lambda_= 1$, a customer arrives for product j, he would choose j from seller k following the distribution $\beta_j^k(S_1, \ldots, S_m)$:

- TBW

If $j \notin S$, the consumer would switch to product i w.p. ρ_{ji}.

Endogenized MNL model

[Wang and Wang, 2017] propose the **endogenized choice model**, which is an extent on MNL and incorporating the network effects.

Definition 4.3.2.2: Network Effects

1. *global* network effects: the utility for consuming a product depends on the total level of consumption of the product;
2. *local* network effects: a consumer gains utility only if his "neighbors" purchase the same product.

To capture the impact of market share on consumers' choice, the utility for product i at time $t+1$ is modeled as:

$$U_i^{t+1} = u_i(q_i^t) + \xi_i^{t+1}, \tag{4.3.2.34}$$

where $\mathbf{q}^t := (q_i^t)_{i \in S}$ is the market share in assortment S. The function $u_i(\cdot)$ is assumed to be linear or log-linear: $u_i(q_i) = \alpha_i + \gamma_i q_i$, where α_i is the intrinsic utility and γ_i measures the magnititude pf the network effects. According to the MNL model, the market share at time $t+1$ is given by:

$$q_i^{t+1} = \frac{\exp(u_i(q_i^t))}{1 + \Sigma_{j \in S} \exp(u_j(q_j^t))}, \quad \forall i \in S. \tag{4.3.2.35}$$

This could result $\mathbf{q^t}$ to converge to a steady-state $\mathbf{q}$:

$$q_i = \frac{\exp(u_i(q_i))}{1 + \sum_{j \in S} \exp(u_j(q_j))} := G_i(\mathbf{q}), \quad \forall i \in S, \tag{4.3.2.36}$$

and $q_0 = 1 - \sum_{i \in S} q_i$. This is the **endogenized MNL model**. Given sufficient condition on u_i, $\mathbf{q}^t$ would converge to an unique $\mathbf{q}$ whatever $\mathbf{q}^0$.

literature Reading Notes

[Liu et al., 2023] study the Simultaneous Recommendation (SMR) and the Sequential Recommendation (SQR) model, and find that the SQR model can always bring more revenue but would reduce the connsumer welfare given a certain assortment. Denote $c_i^k(S) = \min_{j \in \{l | k \in S_l\}} c_{ij}^k$ as the least utility discount of purchasing from another store.

Theorem 4.3.2.4: Assortment Structure of the SMR model

The choice probability for each store i is given by:

$$p_{ik}^M(S) = \begin{cases} \frac{v_k}{1+\sum_{l \in S_i} v_l + \sum_{l \in S \setminus S_i} v_l \exp(-c_i^l(S))} & \text{if } k \in S_i, \\ \frac{v_k \exp(-c_i^k(S))}{1+\sum_{l \in S_i} v_l + \sum_{l \in S \setminus S_i} v_l \exp(-c_i^l(S))} & \text{if } k \in \bar{S} \setminus S_i, \\ 0 & \text{otherwise,} \end{cases}$$

The optimization problem is formulated by:

$\max_{S=(S_1,S_2,\ldots,S_m)} \quad \sum_{i=1}^m \lambda_i \sum_{k=1}^n r_k p_{ik}^M(S)$

It's proved that the optimal assortment is $\boldsymbol{S}^* = (\tilde{S}, \tilde{S}, \ldots, \tilde{S})$, where $\tilde{S}$ is the optimal revenue-ordered assortment for an independent store.

Theorem 4.3.2.5: Assortment Structure of the SQR model

The choice probability is given by ([Gao et al., 2021]):

$$p_{ik}^{Q}(\boldsymbol{S}) = \begin{cases} \frac{v_k}{1+\sum_{l\in S_i} v_l} & if\, k \in S_i, \\ \frac{v_k \exp(-c_i^k(\boldsymbol{S}))}{(1+\sum_{l\in S_i} v_l)(1+\sum_{l\in S_i} v_l + \sum_{l\in \bar{S}\setminus S_i} v_l \exp(-c_i^l(\boldsymbol{S})))} & if\, k \in \bar{S} \setminus S_i, \\ 0 & otherwise. \end{cases}$$

When the utility discouts are store-wise homogeneous ($c_{ij}^k = c_i$), and assume $\lambda_m = \min(\lambda_1, \cdots, \lambda_m)$ and $c_1 \le c_2 \le \cdots \le c_m$. There exists an optimal solution $(S_1^*, S_2^*, \ldots, S_m^*)$ such that $S_1^* \subseteq S_2^* \subseteq \cdots \subseteq S_{m-1}^* \subseteq \tilde{S} \subseteq S_m^*$ and s_i^* are all revenue-ordered.

[Gao et al., 2021] study the assortment optimization in an MNL model with impatient customers. Consider $\mathcal{N} = \{1, \ldots, n\}$ products and $\mathcal{M} = \{1, \ldots, m\}$ stages. The assortments in different stages are disjoint: $S_k \cap S_\ell = \emptyset$ for all $k \neq l$. The patience level of a customer is a RV taking values in $\mathcal{M}$, let $\lambda_k = \mathbb{P}\{Y \ge k\}$. ☺ One importance of this paper is that it shows the closed-form choice probability under multi-stage setting.

Theorem 4.3.2.6: Choice Probability in [Gao et al., 2021]

$$\phi_i^k(S_1, \ldots, S_m) = \frac{\lambda_k v_i}{\left(1 + \sum_{\ell=1}^{k-1} V(S_\ell)\right)\left(1 + \sum_{\ell=1}^{k} V(S_\ell)\right)}, \tag{4.3.2.37}$$

where $V(S) = \sum_{i\in s} v_i$.

Proof 4.3.2

$$\begin{aligned} \phi_i^k(S_1, \ldots, S_k) &= \mathbb{P}\{Y \ge k\} \cdot \mathbb{P}\Big\{U_0 \ge \max_{j\in S_1\cup\ldots\cup S_{k-1}} U_j, \\ &\quad \times U_i \ge \max\{U_0, \max_{j\in S_k\setminus\{t\}} U_j\}\Big\} \\ &= \lambda_k \cdot \mathbb{P}\{U_0 \ge \hat{U}_{k-1}\} \\ &\quad \cdot \mathbb{P}\{U_i \ge \max\{U_0, \tilde{U}_k\} \mid U_0 \ge \hat{U}_{k-1}\}. \end{aligned} \tag{4.3.2.38}$$

Note the conditional part $\mid U_0 \ge \hat{U}_{k-1}$, which is very important.

Q.E.D.

[Peng et al., 2024] proposes a transformer-based deep learning choice model.

4.3.3 RM in the Railway System

The revenue management system has a huge difference compared to the traditional RM system in the airline industry:

1. Regulated by government, usually we can not adjust the prices too frequently;

2. The seats on different legs are assumed to be 'heterogeneous' (even in the same class) because of the *assign-to-seat* policy.

Although the train company can not use the pricing technique to increase revenue, it can decide the capacity reserved for those seats for longer itineraries (*capacity-control problem*). [Zhu et al., 2023] is the first paper studying this question.

Assumption 4.3.3.1: [Zhu et al., 2023]

- There are N homogeneous seats consisting of $M+1$ stops (M legs), the seller can sell itineraries ij at price of p_{ij} for any leg(s) with $j \geq i$;
- The time is starting from 1 and ending at T, each with an arrival rate λ_{ij}^t. Assume the time is short enough that $\sum_{1 \leqslant i \leqslant j \leqslant M} \lambda_{ij}^t \leqslant 1$;
- Let $C^t \in \{0,1\}^{N \times M}$ be the capacity matrix ;
- A policy π is defined as mapping t and C^t to a set of binary variables $u_{k,ij}^t$, which represents a request $i \longrightarrow j$ arriving in time t will be accepted and assigned to seat k.

The authors first investigate the static problem, under which the demands d_{ij} are known in advance and we don't need to consider t. We can formulate it into a integer program:

$$\begin{aligned} \text{maximize} \quad & \sum_{1 \leqslant i \leqslant j \leqslant M} p_{ij} \sum_{k=1}^{N} x_{k,ij} \\ \text{subject to} \quad & \sum_{k=1}^{N} x_{k,ij} \leqslant d_{ij}, \quad \forall 1 \leqslant i \leqslant j \leqslant M, \\ & \sum_{(i,j): i \leqslant \ell \leqslant j} x_{k,ij} \leqslant C_{k\ell}, \quad \forall k \in [N], \ell \in [M], \\ & x_{k,ij} \in \{0,1\}, \quad \forall k \in [N], 1 \leqslant i \leqslant j \leqslant M. \end{aligned} \tag{4.3.3.1}$$

This is an NP-hard problem, and we can relax it into a linear program:

$$\begin{aligned} \text{maximize} \quad & \sum_{1 \leqslant i \leqslant j \leqslant M} p_{ij} \sum_{k=1}^{N} x_{k,ij} \\ \text{subject to} \quad & \sum_{k=1}^{N} x_{k,ij} \leqslant d_{ij}, \quad \forall 1 \leqslant i \leqslant j \leqslant M, \\ & \sum_{(i,j): i \leqslant \ell \leqslant j} x_{k,ij} \leqslant C_{k\ell}, \quad \forall k \in [N], \ell \in [M], \\ & x_{k,ij} \geqslant 0, \quad \forall k \in [N], 1 \leqslant i \leqslant j \leqslant M. \end{aligned} \tag{4.3.3.2}$$

It can be shown that the structure of C not only influences the computational complexity but also decides whether the solutions of problem 4.3.3.1 and 4.3.3.2 are identical.

Definition 4.3.3.1: Maximal Sequence

We call $[u,v]$ a maximal sequence of seat k (denote as $[u,v] \sim C_k$) iff:

$$C_{ku} = C_{k(u+1)} = \cdots = C_{kv} = 1 \quad \text{and} \quad C_{k(u-1)} = C_{k(v+1)} = 0 \tag{4.3.3.3}$$

To complete the definition, we need to define $C_{k0} = C_{k(M+1)} = 0$ for all k. We also define $\mathcal{M}_{uv}(C) = \{k \in [N] | [u,v] \sim C_k\}$.

4.3.4 Assortment Optimization with Competition

Consider two firms indexed by $i \in \{-1, 1\}$, each firm can access the set of products N_i and each chooses assortments $(S_1, S_{-1}) \in \mathcal{F}_1 \times \mathcal{F}_{-1}$. The expected revenue for firm i is:

$$R_i(S_i, S_{-i}) := \sum_{j \in S_i} p_j \pi_j(S_i, S_{-i}) = \frac{\sum_{j \in S_i} p_j v_j}{v_0 + V(S_i) + V(S_{-i})} \tag{4.3.4.1}$$

The key question is: does the Nash equilibrium exist in this game? Let $z_i^*(S_{-i})$ denote the best expected revenue for firm i given S_{-i}, we have:

$$z_i^*(S_{-i}) \geq \frac{\sum_{j \in S_i} p_j v_j}{v_0 + V(S_i) + V(S_{-i})} \quad \forall S_i \in \mathcal{F}_i, \tag{4.3.4.2}$$

$$[v_0 + V(S_{-i})] z_i^*(S_{-i}) \tag{4.3.4.3}$$

So the problem is equivalent to solving:

$$\max_{S_i \in \mathcal{F}_i} \left\{ \sum_{j \in S_i} (p_j - z_i^*(S_{-i}))\, v_j \right\}. \tag{4.3.4.4}$$

This transformation can not bring us to the optimal solution since $z_i^*(S_{-i})$ is unknown. Let $\hat{S}_i$ be the best response to $\hat{S_{-i}}$ and $\tilde{S}_i$ to $\tilde{S_{-i}}$.

Lemma 4.3.4.1

If $V(\hat{S}_{-i}) \leq V(\tilde{S}_{-i})$, then $V(\hat{S}_i) \leq V(\tilde{S}_i)$.

Using this lemma, we can observe that the sequence $\{(\hat{S}_1^t, \hat{S}_{-1}^t) : t = 0, 1, \ldots\}$ gennerated in the *tatonnement process* would converge to a Nash equilibrium. What's more, a Nash equilibrium generated by the tatonnement process is Pareto dominant (has the highest $V(S^*)$).

4.4 Pricing

[Den Boer, 2015] reviews the literature studying dynamic pricing and learning - the study of optimal dynamic pricing in an uncertain environment where characteristics of consumer behavior can be learned from accumulating sales data by 2014. There are two main streams of research in dynamic pricing: one assumes the demand function is changing with time, while another assumes static demand but finite inventory. [Özer and Phillips, 2012] provides a comprehensive tutorial on pricing.

4.4.1 Basic Pricing Theory

This subsection summarizes the basic pricing theory for **multi-product momnpoly** firms.

Perspective from the Firm

The firm's profit function is given by:

$$R(p,z) := (p-z)'d(p) = \sum_{i=1}^{n}(p_i - z_i)d_i(p_1,\dots,p_n), \tag{4.4.1.1}$$

Where $z = (z_1,\cdots,z_n)$ is the variable cost vector, $p = (p_1,\cdots,p_n)$ is the price vector, $d(p)$ is the demands. Currently it's popular to set $p_i \in [0,\infty]$, whereby setting $p_i = \infty$ is equivalent to not offering product i.
Consider the revenue as a function only of z, given by:

$$\mathcal{R}(z) := \max_{p\in X} R(p,z), \tag{4.4.1.2}$$

Where X is the set of allowable prices.

Theorem 4.4.1.1: The Decreasing Covex Property
$\mathcal{R}(z)$ is **decreasing convex** in z.

Proof 4.4.1

The decreasing property is obvious since $R(p,z)$ is decreasing in z, here is the proof for convexity.
For any $z,\tilde{z}$:

$$\begin{aligned}
\mathcal{R}(\alpha z + (1-\alpha)\tilde{z}) &= \max_{p\in X} R(p, \alpha z + (1-\alpha)\tilde{z}) \\
&= \max_{p\in X} R(\alpha p + (1-\alpha)p, \alpha z + (1-\alpha)\tilde{z})) \\
&= \max_{p\in X} \left[\alpha(p-z)' + (1-\alpha)(p-\tilde{z})'\right] d(p) \\
&= \max_{p\in X} \left[\alpha R(p,z) + (1-\alpha)R(p,\tilde{z})\right] \\
&\le \alpha \max_{p\in X} R(p,z) + (1-\alpha)\max_{p\in X} R(p,\tilde{z}) \\
&= \alpha\mathcal{R}(z) + (1-\alpha)\mathcal{R}(\tilde{z}).
\end{aligned} \tag{4.4.1.3}$$

☺ This proof is very similar to proving the convexity of the hyperplane.

Q.E.D.

Transforming the revenue function as a function only to z is useful. Because by **Jensen's Inequality**, $\mathbb{E}[\mathcal{R}(\mathbb{Z})] \ge \mathcal{R}(\mathbb{E}[\mathbb{Z}])$. Their differences can be interpreted as the difference between a dynamic pricing policy

$p(Z)$ that responds to changes in Z and a static policy $\mathcal{R}(\mathbb{E}[\mathbb{Z}])$, the larger the variance of Z, the larger the gap between them. This implies that an interesting phenomenon, in the revenue management domain, considering dynamic pricing, the firms can prefer stochastic production cost over static cost.

Perspective from the Consumers

To answer whether consumers are better off with pricing policy $p(Z)$ or $p(\mathbb{E}(Z))$, we can frame this using the **utility theory** ([Chen and Gallego, 2019]). Assume that consumers purchase a non-negative vector $q = (q_1, q_2, \cdots, q_n)$ of products, their *net utility* is given by:

$$S(q, p) := U(q) - q'p, \tag{4.4.1.4}$$

where $U(q)$ us an increasing concave function. Then the *optimal surplus* can be conputed by:

$$\mathcal{S}(p) := \max_{q \geq 0} S(q, p) \tag{4.4.1.5}$$

where the solution $q^\star = d(p) = \nabla^{-1}U(p)$. under the first-order condition. Similar to the analysis from the firm's perspective, we can get $\mathbb{E}[\mathcal{S}(P)] \geq \mathcal{S}(\mathbb{E}[P])$. And at last, we can achieve:

$$\mathbb{E}[\mathcal{S}(p(\mathbb{Z}))] \geq \mathcal{S}(\mathbb{E}[p(\mathbb{Z})]) \geq \mathcal{S}(p(\mathbb{E}[Z])). \tag{4.4.1.6}$$

Pricng Considering Finite Capacity

Consider a monopoly selling n products with price p using m types of resources, $A \in \mathbb{R}^{m \times n}$ the resource consumption matrix. Assume that $d(p) : \mathfrak{R}_+^n \to \mathfrak{R}_+^n$ is the continuous demand function. Given a m-dimensional resource limit vector c, the revenue can be expressed as:

$$\bar{V}(c) := \max_{p \in X} R(p, 0) \text{ subject to } Ad(p) \leq c. \tag{4.4.1.7}$$

This problem may be difficult to solve because the objective function is not concave, and the constraint is not convex. There are two solutions:

1. Using the inverse demand function $p(q)$, rewriting the optimization to maximize $p(q)'p$ and the constraint as $Aq \leq c, q \geq 0$. Now the constraint is convex;
2. Using the Lagrangian relaxation,

Single Product Pricing Problems

Let $R(p, z) = (p - z)'d(p)$, the maximizer $p(z)$ is definite as:

$$p(z) \in \arg\max_{p \geq 0} R(p, z) \tag{4.4.1.8}$$

For a special single product case with $n = 1$, $d(p) = \lambda \mathcal{P}(W \geq p)$, where $\lambda > 0$ is the expected market size and W is the maximum willingness to pay. **If** $\mathbb{E}[W] < \infty$, **then** $p(Z)$ exists. Now consider more general case, let $\bar{d}(p) := \sup_{\tilde{p} \geq p} d(\tilde{p})$, notice $\bar{d}(p)$ is a decreasing function. Denote $\bar{R}(p, z) := (p - z)\bar{d}(p)$.

Theorem 4.4.1.2

If $d(p)$ is upper semicontinuous in $p \geq 0$ and $\bar{d}(p) = o(\frac{1}{p})$, then there exists a finite maximizer $p(z)$ increasing in z, that simultaneously maximizes $R(p, z)$ and $\bar{R}(p, z)$. So $R(p, z) = \bar{R}(p, z)$.

4.4.2 Multi-agent Dynamic Pricing

Dynamic Pricing with Competition

Reinforcement Learning and Pricing Collusion

[Gallego and Hu, 2014] studied the dynamic pricing in competition. Consider a market of m firms selling differentiated perishable products over a finite horizon $[0, T]$. At time $t = 0$, each firm i has capacity c_i. Let $d(t, p(t))$ be the demand vector and $r_i(t, p) = p_i d_i(t, p)$. The authors prove that if the choke price is available for each firm, then an equilibrium exists. [Gerpott and Berends, 2022] provide a comprehensive survey for this topic.

Dynamic Pricing and Algorithmic Collusion

[den Boer, 2023] provides a mathematical definition of algorithmic collusion, there are four conditions for collusion:

1. the algorithm should be implementable in practice (**incomplete information** about the payoff structure);
2. the algorithm should be rational, ensuring a good performance against various alternative algorithms is also essential;
3. supracompetitive outcomes;
4. no illegal communications.

Let $\nu := \{\nu_{a_1,a_{-1}} : (a_1, a_{-1}) \in \mathcal{A}_1 \times \mathcal{A}_{-1}\} \in \mathcal{V}_0$ be the collection of probability distributions of $(Y_{1,a_1,a_{-1}}, Y_{-1,a_{-1},a_1})$, we call ν the *true parameter* in the repeated game. [Misra et al., 2019] is probably the first paper that incorporates reinforcement learning (MAB) algorithm into dynamic pricing.

Assumption 4.4.2.1: [Misra et al., 2019]

For consumers:

1. She has stable preferences v_i doesn't change over time;
2. Her choice satisfies weak axiom of revealed preference: if $v_i > p$, then she would buy it;
3. Heterogeneity among consumers can be separated as observable and unobservable heterogeneity: $v_i = f(\mathbb{Z}_i) + \nu_i$;
4. There are bound for $\nu_i : \nu_i \in [-\delta, \delta]$

Model Setting:

1. There are K prices that a firm can choose: $p \in \{p_1, \ldots, p_K\}$;
2. sample mean $\bar{\pi_{kt}} = 1/n_{kt} \sum_{\tau=1}^{n_{kt}} \pi_{k\tau}$, a policy is defined as $p_t = \Psi(\{p_\tau, \pi_\tau | \tau = 1, \ldots, t-1\})$;
3. The regret is given by: $\text{Regret}(\Psi, \{\pi(\mathfrak{p}_t)\}, t) = \pi^* t - \sum_{k=1}^{K} \pi(p_k)\mathbb{E}[n_{kt}]$

There are K segments of consumers, each segment is denoted by θ_k and s, and their behaviour follows:

1. $D(p_k)_{s,t} = 0$ is consistent with consumers being of type $\{\theta_1, \cdots, \theta_{k-1}\}$;
2. $D(p_k)_{s,t} = 1$ is consistent with consumers being of type $\{\theta_k, \cdots, \theta_K\}$;

In this paper a novel algorithm (UCB-PI) is proposed and verified that can outperform other algorithms. [Calvano et al., 2019], [Calvano et al., 2020b], [Calvano et al., 2020a] are the first few papers studying the algo-

rithmic collusion of RL algorithm in a Non-monopoly market. In these papers, the demand function is assumed as

$$q_{i,t} = \frac{e^{\frac{a_i - p_{i,t}}{\mu}}}{\sum_{j=1}^{n} e^{\frac{a_j - p_{j,t}}{\mu}} + e^{\frac{a_0}{\mu}}}. \tag{4.4.2.1}$$

Where the parameters a_i indicate the vertical differentiation and a_0 is an inverse index of aggregated demand. μ indicates the horizontal differentiation. Each firm's objective is to maximize the profit: $\pi_{i,t} = \left(p_{i,t} - c_i\right) q_{i,t}$, where c_i is the cost. For a set of given parameters, we can compute both the Bertrand-Nash equilibrium of the one-shot game and the monopoly prices for each firm, denoted by P^N and P^M. The action space for each firm is given by $\left[\mathbf{p}^N - \xi(\mathbf{p}^M - \mathbf{p}^N), \mathbf{p}^M + \xi(\mathbf{p}^M - \mathbf{p}^N)\right], \xi > 0$. To ensure the state space is finite, the state space is defined as $s_t = \{\mathbf{p}_{t-1}, \ldots, \mathbf{p}_{t-k}\}$. So for each player i, we have $|A| = m$ and $|S| = m^{nk}$.
This paper assumes each firm utilizes the Q-learning algorithm with ϵ exploration:

$$Q_{t+1}(s,a) = (1-\alpha)Q_t(s,a) + \alpha\left[\pi_t + \delta \max_{a \in A} Q_t(s',a)\right] \tag{4.4.2.2}$$

$$\epsilon_t = e^{-\beta t}. \tag{4.4.2.3}$$

To test the final results, the authors further define the **average profit gain** Δ:

$$\Delta \equiv \frac{\bar{\pi} - \pi^N}{\pi^M - \pi^N} \tag{4.4.2.4}$$

$\Delta = 0$ corresponds to the competitive outcome and $\Delta = 1$ corresponds to the perfectly collusive outcome. Their simulation shows that Q-learning pricing algorithms systematically learn to collude. The collusion is typically partial and is enforced by punishment in case of deviation. The article has no analytical proof. The authors also investigate the algorithm's strategy toward competitor's deviation: at epoch 0 the supra-competitive outcomes have been achieved, and at epoch 1 one party is defected by force, but then back to using the algorithm to decide as normal. The dynamics can be shown in the picture below:

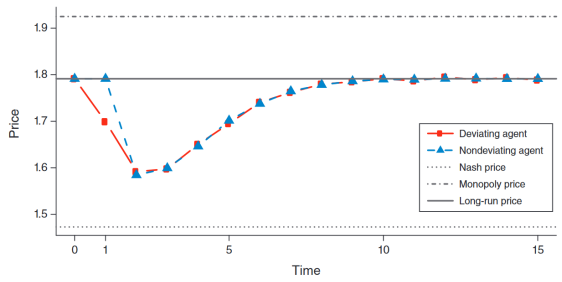


Figure 4.4: collusion deviation

[den Boer et al., 2022] verified this collusion but criticized that this paper doesn't reveal the 'true' algorithmic collusion. To see this, we can first define the $\delta - \epsilon$ equilibrium:

Definition 4.4.2.1: $\delta - \epsilon$ equilibrium

A pair of strategies $\sigma^{(i)}, \sigma^{(-i)}$ is a $\delta - \epsilon$ equilibrium if for all i

$$\begin{aligned}\sigma^{(i)}(s) \in \arg\max_{p\in\mathcal{A}_i} &\frac{\epsilon}{|\mathcal{A}_{-i}|}\sum_{p_{-i}\in\mathcal{A}_{-i}}\{\pi_i(p,p_{-i}) + \delta V^{(i)}_{\sigma^{(-i)}}(p,p_{-i})\} \\ &+ (1-\epsilon)\{\pi_i(p,\sigma^{(-i)}(s_{-i},s_i)) + \delta V^{(i)}_{\sigma^{(-i)}}(p,\sigma^{(-i)}(s_{-i},s_i))\},\end{aligned} \tag{4.4.2.5}$$

where $V_{\sigma^{(-i)}}$ is defined as:

$$\begin{aligned}V^{(i)}_{\sigma^{(-i)}}(s) := \max_{p\in\mathcal{A}_i} &\frac{\epsilon}{|\mathcal{A}_{-i}|}\sum_{p_{-i}\in\mathcal{A}_{-i}}\{\pi_i(p,p_{-i}) + \delta V^{(i)}_{\sigma^{(-i)}}(p,p_{-i})\} \\ &+ (1-\epsilon)\{\pi_i(p,\sigma^{(-i)}(s_{-i},s_i)) + \delta V^{(i)}_{\sigma^{(-i)}}(p,\sigma^{(-i)}(s_{-i},s_i))\}.\end{aligned} \tag{4.4.2.6}$$

Consider cases where the price space is only 2 (collusive price C and defective price D), we can formulate a prisoner's dilemma:

	C	D
C	(R,R)	(S,T)
D	(T,S)	(P,P)

Figure 4.5: Prisoner Dilemma

Assume $T > R > P > S$. There are 16 types of pure strategy, among which three can be equilibrium depending on the value δ and ϵ:

- AD 'all defect': always plays D: all the time;
- GT 'grim trigger': plays C in state (C,C), and D otherwise: $\frac{\epsilon(S+T-R-P)+2(R-T)}{(\epsilon^2-3\epsilon+2)(P-T)} < \delta < \frac{\epsilon(S+T-R-P)+2(P-S)}{(1-\epsilon)(\epsilon(T-P)+2(P-S))}$;
- WSLS 'Win-Stay-Lose-Shift': plays C in state (C,C) and (D,D), and D otherwise: $\delta > \frac{\epsilon(P+R-S-T)+2(T-R)}{(1-\epsilon)(\epsilon(P+S-T-R)+2(R-P))}$.

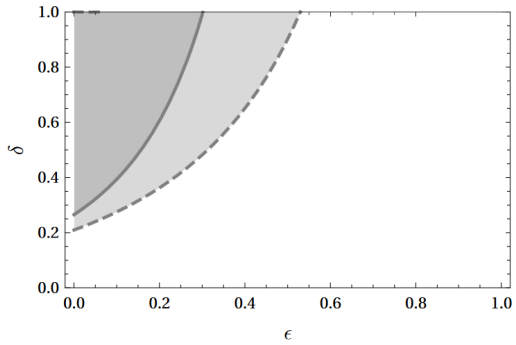


Figure 4.6: Phase Diagram of Equilibrium

We can see that when ϵ is large, the collusive equilibrium only exists after a long time of exploration. However, Boer shows that the time of exploration could be very long and there is still a large fraction of simulations that don't converge to the collusive outcome.

[Asker et al., 2022] and [Asker et al., 2023] extend Calvano's research methodology using the Bertrand Pricing model, and find that during such setting, the game would converge to the collusive outcome when both agents use the *synchronous updating* scheme.
[Hansen et al., 2021] follows the research and verifies whether the UCB algorithm would lead to collusion in the duopoly market. They assume a symmetric linear demand:

$$d_j^*(p_j, p_{-j}) = \alpha - \beta p_j + \gamma p_{-j}. \tag{4.4.2.7}$$

The competitive price and the joint monopoly (supra-competitive) price can be calculated:

$$p^D = \frac{\alpha}{2\beta - \gamma}, p^M = \frac{\alpha}{2(\beta - \gamma)}. \tag{4.4.2.8}$$

In the experiment, a firm observes a noisy outcome:

$$\pi_t(p_j, p_{-j}) = p_j d_j^*(p_j, p_{-j}) + \varepsilon_{j,t}, \tag{4.4.2.9}$$

where $\varepsilon_{j,t} \sim U[-\frac{1}{\delta}, \frac{1}{\delta}]$, and δ is the Signal-to-Noise-Ratio (SNR). The two firms use UCB-tuned algorithm to dynamically price their products:

$$\begin{aligned} \mathrm{V}_{k,t} &= \overline{\pi_{k,t}^2} - \bar{\pi}_{k,t}^2 + \sqrt{\frac{2\log t}{n_{k,t}}}, \\ \mathrm{UCB-tuned}_{k,t} &= \bar{\pi}_{k,t} + \sqrt{\frac{\log t}{n_{k,t}} \min\left(\frac{1}{4}, \mathrm{V}_{k,t}\right)}, \end{aligned} \tag{4.4.2.10}$$

The simulation results show that the higher the SNR, the higher the chance of converging to the collusive outcome. (A simple ϵ greedy algorithm would converge to Nash-equilibrium).

Theorem 4.4.2.1: [Hansen et al., 2021]
Suppose the true demand function is deterministic, and both firms use independent UCB algorithms. Then,

1. The prices are always exactly correlated from the third period onward;
2. The fraction of times in the first t periods that *either* seller charges p_L converges to zero as $t \to \infty$.

Proof 4.4.2

Assume for each firm there are only two prices that can be chosen: p_L, p_H, where p_L can be thought of as the competitive price and p_H is the supra-competitive price. After the first two rounds, there can only be two cases:

1. Matched prices, that is, (p_H, p_H) in one round and (p_L, p_L) in the other;
2. Mismatched prices.

In both cases, the two agents have the same index for the two arms. Because the decision process and the demand function are deterministic, they will pose the same action forever. Although they might choose p_L at some time when n_L is small, the actions would be made simultaneously, and revert to p_H ar the next epoch.

Q.E.D.

[Banchio and Mantegazza, 2022] rigorously analyze why Q-learning can influence the dynamics of the pricing in the Prisoner's Dilemma setting. They approximate the discrete dynamic pricing using a continuous fluid dynamical system. They show that although algorithms explore independently, their estimates are correlated because their payoffs depend on the entire action profile (referred as **spontaneous coupling**). This coupling leads to the continuous counterpart of stochastic cycles, in which the agents cooperate most of the time.
[Martello, 2022] shows that the policy-gradient reinforcement learning algorithms can result in genuine collusion. [Hettich, 2021] shows that using DQN can result in similar collusion (even faster than Q-learning), and the agents can respond strategically toward the opponent's defection. [Klein, 2021] discuss the algorithmic collusion of Q-learning in another economic environment. Consider two agents $i \in \{1,2\}$ price in infinitely discrete time indexed by $t \in \{0,1,2,\cdots\}$. Price adjustments occur **sequentially**: agent 1 can only adjust its price in odd-numbered periods, agent 2 in even-numbered periods. Each agent selects its price from a discrete set $P = \{0, \frac{1}{k}, \frac{2}{k}, \ldots, 1\}$. Firm i profit at time t is derived as $\pi_i(p_{it}, p_{jt}) = p_{it} D_i(p_{it}, p_{jt})$, its objective function can be expressed as $\sum_{s=0}^{\infty} \delta^s \pi_i(p_{i,t+s}, p_{j,t+s})$. The demand is linear:

$$D_i(p_{it}, p_{jt}) = \begin{cases} 1 - p_{it} & \text{if } p_{it} < p_{jt} \\ 0.5(1 - p_{it}) & \text{if } p_{it} = p_{jt} \\ 0 & \text{if } p_{it} > p_{jt} \end{cases}. \tag{4.4.2.11}$$

This setting provides a monopoly collusive price of 0.5 with a per-firm profit of 0.125. Here, we use the Markov Perfect Equilibrium (MPE):

$$V_i(p_{jt}) = \max_p \left[\pi_i(p, p_{jt}) + E_{p_{j,t+1}}\left[\delta \pi_i(p, p_{j,t+1}) + \delta^2 V_i(p_{j,t+1})\right]\right] \tag{4.4.2.12}$$

An action pair (R_1, R_2) is an MPE if equation 4.4.2.12 holds for both agents. To align with the game setting, the Q-learning algorithm is simply modified, the state s_t is the opponent's price $p_{j,t}$, and the updating formula is given as:

$$\tilde{Q}(p_{i,t}, s_t) = \pi_i(p_{i,t}, s_t) + \delta \pi_i(p_{i,t}, s_{t+1}) + \delta^2 \max_p Q_i(p, s_{t+1}) \tag{4.4.2.13}$$

Both agents use the ϵ-greedy strategy to select their actions. The performance metric is the average profit in the last 1000 runs. In this paper, the author distinguishes whether a strategy is Nash-equilibrium by looking at the optimality of both agents:

$$\Gamma_i(p_i, p_j) \doteq \frac{Q_i(p_i, p_j)}{\max_p Q_i^*(p, p_j)}, \tag{4.4.2.14}$$

If $\Gamma = 1$ for both agents, the action pair p_i, p_j can be regarded as a Nash equilibrium. Among all Nash equilibriums, some are collusive equilibriums when profitability is above the competitive benchmark. It is found that agents have high chances to collude, and *Edgeworth price cycles* are observed. However, unlike theoretical analysis in past works, these price cycles are deterministic: It is always the same firm that undertakes the costly action of "resetting" the price cycle by jumping up in price rather than undercutting, with the other firm able to free-ride on this.

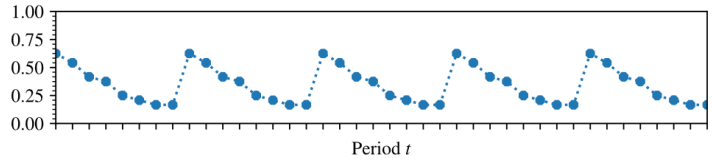


Figure 4.7: Edgeworth price cycles

Remark 4.4.1

Two Limitations of [Klein, 2021]

- The convergence is not guaranteed;
- Q-learning is restricted to playing pure strategies.

[Calvano et al., 2023] surveys some papers on algorithmic collusion and further categorizes the collusion into two groups:

- **genuine collusion**: the algorithms do not cut their prices because they anticipate retaliatory responses by rivals;
- **spurious collusion**: the algorithms do not cut their prices simply because they are missing opportunities to earn more.

In this paper, the authors claim that the memoryless Q-learning algorithm won't lead to collusion, and the exploration mode (random exploration v.s. optimistic initialization) is the key to a collusive outcome.
[Epivent and Lambin, 2024] found that, although the agent in the seemingly genuine collusion in [Calvano et al., 2020b] would start a price war when the opponent deviates to a lower price, but he would also start the price war when the opponent deviates to a higher price. The authors further argued that the seeming collusion (and the reward-punishment scheme) is due to poor exploration.

Empirical Evidence of Collusion

[Musolff, 2022] empirically examines the impact of repricing tools on the pricing dynamics. Using RDD, he found the pricing history exhibits the pattern of Maskin-Tirole's Edgeworth cycle. However, the data fits the delegated strategies better, which would increase the price in the future and might result in collusion. [Clark et al., 2023] provide a more convincible result: using IV, they identify that: AI adoption by one station in a duopoly market does not change the distribution of market-level margins relative to a duopoly market where no stations adopted. However, AI adoption by both stations in a duopoly market changes the distribution of market-level margins.

Remark 4.4.2

Note that both papers only examine the effects of the AI pricing algorithms on the prices in competing environments, but AI $\neq$ reinforcement learning.

Theoretical Evidence

[Dolgopolov, 2024] provide the most convincible proof for the collusive outcome of using Q-learning (although he only considers the memoryless Q-learning with singleton state). Consider a two-player Prisoner's Dilemma game with action $A = \{C, N\}$ and payoffs $\pi_{NC} > \pi_{CC} > \pi_{NN} > \pi_{CN}$. The system can be modeled as a Markov process where the state g is a pair of Q-vectors $g = (Q_1, Q_2)$, each Q-vector contains Q-values for every element in A. The state space is denoted as $\mathfrak{G} \subseteq \{(Q_1, Q_2) : Q_i \in \mathfrak{D}^{|A|}\}$, where $\mathfrak{D} := \{\zeta\epsilon, \zeta \in \mathbb{Z}\}$ (to discretize the Q-value). The **unperturbed dynamics** denoted as P_0 is the transition probability $P_0(g, g')$ correspondence to the RL

algorithm without exploration. The author proved that even using memoryless Q-learning, the algorithm has positive probability converging to the collusive outcome if the parameter is not degenerate.
[Possnig, 2023] proves that whether the algorithm would converge to the collusive outcome depends on two things:

1. The state space: if the state space is the historical price, then with sufficient conditions on the structure of the payoff function, the algorithm would converge to the static Nash Equilibrium;

2. When the state space is rich, even with the sufficient conditions above, the RL algorithm can still converge to the collusive outcome with a positive probability.

Algorithmic Collusion

[**?**] [Aouad and den Boer, 2021] proposes an efficient algorithm to support the tacit collusion in the multi-agent assortment game (solving the collusive factor is NP-hard, thus needs an efficient way). [Meylahn and V. den Boer, 2022] presents a price algorithm that can learn to collude when used by both players in a duopoly with an unknown demand function and no inventory constraints. They assume there is a **unique** maximizer $\mathbf{p}^{col}$ and a unique Nash Equilibrium $\mathbf{p}^{com}$. There are five phases in their algorithm, in the first and second phases, the two players explore and estimate the collusive outcomes; in the third and fourth phases, the two players explore and estimate the competitive outcomes; in the 5th stage, the two players make decision based on the previous outcomes. The authors proved that this algorithm can converge to the supracompetitive outcomes when used by both firms and can respond competitively to the reaction function when used by a single firm.
[Loots and den Boer, 2023] proposed an algorithm that can collude under the multinomial logit demand function, and this algorithm can learn another firm's private information using the public pricing information.
[Abada and Lambin, 2023] studies the dynamic pricing collusive scenario using the model of the electricity markets. Agents are indexed by $i \in I$ and their actions time t are denoted by a_i^t, $a_i^t \in \mathbb{R}$ is positive (negative) if the agent i sells to (buys from) the market at time t. There is a group of fringe producers a_f^t, so the overall demand at time t is given by:

$$D^t = \sum_i a_i^t + a_f^t \quad \forall t \in 1, \dots, T.$$

But the fringe producers are price-takers and can be ignored, thus the residual inverse demand function is given by:

$$p^t = \tilde{p}^t - \beta^t \sum_i a_i^t \quad \forall t \in 1, \dots, T,$$

Each agent has a capacity l_i^t with a linking function $l_i^{t+1} = l_i^t - a_i^t$, facing the trading constraint as well as the stocking constraint. Denoting $\mathbf{a}^t \equiv (a_1^t, a_2^t, \dots, a_I^t) \in \mathbb{R}^{|I|}$, we can write the payoff function for each agent as (ignore the discounted factor):

$$\Pi_i = \sum_{t=1}^{T} p^t(\mathfrak{a}^t) a_i^t, \tag{4.4.2.15}$$

and the corresponding optimization can be written as:

$$\begin{aligned} \text{Max}_{a_i^1, \dots, a_i^T} & \sum_{t=1}^{T} p^t(\mathbf{a}^t) a_i^t \\ \text{s.t.} \quad & l_i^{t+1} = l_i^t - a_i^t \quad \forall t \in 0, \dots, T-1, \\ & |a_i^t| \le K_i \qquad\qquad \forall t \in 1, \dots, T, \\ & 0 \le l_i^t \le L_i \qquad\qquad \forall t \in 1, \dots, T. \end{aligned} \tag{4.4.2.16}$$

Each agent uses Q-learning with ϵ-greedy exploration, the state space is defined as the historical prices and

capacities (in a discrete style). The experiments show that the agent can converge to the collusive outcomes, and each agent exhibits the punishment for any deviation. The authors explain this 'seemingly' collusion as a result of **lack of exploration**, to support this intuition, they artificially force the learning process to keep exploring at some steps, and the results learned are closer to the full-competition phenomenon. Finally, the authors provide two methods to avoid(reduce) this collusion:

1. Local learning: prevent the aggregated learning;
2. The regulator can play in the market as a participant.

Some Theoretical Works from a Game Theory Perspective

[Dolgopolov, 2024] studied whether a prisoner's dilemma would converge to the collusive outcomes using bandit RL algorithms. He proved that by using a large enough learning rate, the game would converge to the collusive outcomes, the proof uses one-shot deviation principle, and to simply the proof, the author discretizes the Q-value.

4.5 Platform Operations Management

[Rietveld and Schilling, 2021] reviews literature in platform competitions.

[Wang et al., 2023] uses the game theory framework to analyze the cross-licensing policy initiated by Qualcomm, which provides some insights for the up-stream manufacturing company:

1. The supplier shouldn't adopt the cross-licensing policy if the inferior manufacturer's cost of innovation is high;
2. Cross-licensing may achieve a higher level of total innovation if the superior manufacturer's cost of innovation is low;
3. The superior manufacturer can benefit from cross-licensing, if innovation is costly but the manufacturers' costs of innovation is similar.

[Deng et al., 2021] studies the business pattern operation for the last-mile delivery in the city using the game theory framework. In the model, the decision maker can decide whether to launch a UCC (urban consolidation center) or to operate a P2P platform for carriers to share the capacity. There are n carriers in the region, each jas a delivery task with volume v_i. $v_i = v_L$ with a probability λ, and $v_i = v_H (> v_L)$ with a probability $1 - \lambda$. Each carrier can deliver his own task with a fixed cost of c and a variable cost of m, or deliver by the platform. In the UCC mode, the UCC incurs a fixed cost of C, a variable cost of M, and a variable subsidy S. The center charges a unit fee $\bar{p}$ for the carriers, after observing the unit fee, each carrier decides to deliver on his own or to outsource the task independently. In the P2P mode, the platform follows a similar decision process. The equilibrium analysis shows the critical condition (related to m, n) under which the UCC or the P2P mode can reduce more social-environmental cost.

4.5.1 Hotel Platform

An overview of Airbnb

[Dolnicar, 2021] provides a comprehensive illustration of all aspects of Airbnb, including the business model, competitive landscape, and the regulations of Airbnb.

[Guttentag, 2019] reviewed some tier c papers about the progress on Airbnb, their main focus is on the loyalty and motivation of guests and hosts, Airbnb's regulation and culture, as well as Airbnb's impact on the tourism sector.

Airbnb makes money by renting out property that it doesn't own, the hosts can be an individual or a company (But Airbnb doesn't own property, it's just an intermediary). ([Folger, 2023]) In 2017, Airbnb invested in Niido, a hotel-like apartment program managed by Airbnb. In 2020, Airbnb ended its partnership with Niido apartments. During the whole process, only 2 apartments were in service.

[Zhang et al., 2022] studies how Airbnb property demand changed after the acquisition of *verified* images. Variables description:

- Treatment variable: 212 properties had verified photos by the end of April 2017, the remaining 7,211 did not;
- Property demand: purchased date, the number of days in a month in which the property was open, blocked, and booked. $\frac{booked}{open} \times 100$
- Property Price: the price is endogenous because of the random demand shocks, characteristics of competing properties were used as IVs (The logic is that the characteristics of competing products are unlikely to be

correlated with unobserved shocks in the demand for the focal property. However, the proximity of the characteristics of a property and its competitors influences the competition and as a result, the property markup and price). Cost-related variables are collected including residential utility fees.

- Property Photos: CNN architecture was used to predict the quality of a photo (dummy variable).

Other Identification Techniques:

- DiD model: $DEMAND_{itcym} = INTERCEPT + \alpha TREATIND_{it} + \lambda CONTROLS_{it} + PROPERTY_i + SEASONALITY_{cym} + \varepsilon_{it}$;
- PSW method: calculate the prosensity score $\widehat{ps_i(X_i)}$ and use IPTW ($\omega_i(T, X_i) = \frac{T}{\widehat{ps_i(X_i)}} + \frac{1-T}{1-\widehat{ps_i(X_i)}}$) to weight the sample.
- Relative time model was used to test the common trends assumption, Rosenbaum bounds test was used to test the validation of PSW methods.

Besides the work above, the authors investigate what makes a picture good. They listed 3 components (composition, color, figure-ground relationship) including 12 attributes and used the following regression to give some human-interpretable suggestions:

$$\begin{aligned} DEMAND_{itcym} = & INTERCEPT + \alpha TREATIND_{it} \\ & + \mu IMAGE\_COUNT_{it} \\ & + \rho_1 \mathrm{R\_\_IO}_{it} \\ & + \rho_2 BEDROOM\_PHOTO\_RATIO_{it} \\ & + \rho_3 \mathrm{IN\_\_IO}_{it} \\ & + \rho_4 LIVING\_PHOTO\_RATIO_{it} \\ & + \eta IMAGE\_ATTRIBUTES_{it} \\ & + \lambda CONTROLS_{it} + SEASONALITY_{cym} \quad + PROPERTY_i + \epsilon_{it} \end{aligned} \tag{4.5.1.1}$$

[Chen et al., 2023b] investigates the professional players' effects on the non-professional host. They define a host who has more than one properties on the platform simultaneously as professional players, and use a quasi-experiment (OHOH policy) and DiD model to analysis whether competition effects or differentiation effects is dominant.
They predict 2 propositions:

1. If differentiation effects dominate, then OHOH would not affect the supply and price of non-professional hosts;
2. If competition effects dominate, then OHOH would increase the supply and price of non-professional hosts.

$$Y_{it} = \mu_i + \nu_t + \beta \cdot 1(\text{Policy})_{it} + \gamma' \mathbf{X}_{it} + \varepsilon_{it} \tag{4.5.1.2}$$

Their results show that the competition effects dominate the role of professional hosts. [Farronato and Fradkin, 2022] investigate the peer's entry on consumer's welfare in the accommodation industry using a structural model.
Intuition: Peer hosts are responsive to market conditions, expand supply as hotels fill up, and keep hotel prices down as a result.

4.5.2 Platform Owner's Entry

Keywords 4.2

Complementory Markets, Spillover Effects

[Chen and Tsai, 2023] is the first paper considering the asymmetric information between the platform owner and the third-party sellers, and the effects of the information disadvantage on consumer's welfare. They assemble data from Amazon: 122000 products, each with two sellers offering the same product. *The information asymmetry*: when Amazon's competitors make sales, Amazon adjusts its prices accordingly; conversely, third-party sellers do not react to their competitors' sales.
After observing the empirical evidence of the information asymmetry, they design a theoretical model and identify the parameters. Using the structural regression, they find that by giving information to third-party sellers. Both the Amazon, third-party sellers and the consumers' welfare would be increased.

[Zhu and Liu, 2018] surveys empirical studies that examine the direct entry of platform owners into complementary product spaces.

[Zhu and Liu, 2018] studies the entry of Amazon platform. Logit regression is adopted to verify the following **Hypothesis**:

- Platform owners are more likely to compete with a complementor when its products are successful;
- Platform owners are less likely to compete with a complementor when its products require significant platform-specific investments to grow.

Identification Techniques:

- The sales ranking is used as the proxy variable for the sales of the products;
- To overcome the impact of the referral rates by category-level fixed effects;
- To measure the seller's platform-specific investment, they calculate the seller's average answers;

[He et al., 2020] investigates the effects and the mechanisms of platform owner's entry on third-party's online and offline demands using a B2B shopping platform's data. They use DiD to identify that the platform owner's entry does harm the demands (more in online channels than offline channels).
What's more, they propose three mechanisms to explain the effects:

1. Competition Effects: The owner can appropriate value from the third-party sellers (only significant for online channels);
2. Spillover Effects: increasing the exposure or awareness of the products (not the same as the mobile app market);
3. Disintermediation effect: sellers would use defensive strategy to transact outside the platform (mostly in offline channel) see [**?**] and [**?**].

Finally, DDD and PSM were adopted to identify the heterogeneity of the effects between large and small third-party stores.
[Deng et al., 2023] use data from JD.com and provide an unexpected result, thet [Wen and Zhu, 2019] and [Foerderer et al., 2018] focus on complementors' reactions, especially on innovation strategy for the platform owner entry.
[Shi et al., 2023] investigates the timing of the platform owner's entry on the value creation.

Definition 4.5.2.1: Timing of Owner's Entry

- **Platform Complementors**: actors that offer an application that brings additional value to platform users when used in combination with the platform;
- **Early-entry**: the entry occurs when the ratio between the current and the eventual complementary market's size is low;
- **Late-entry**: entry to a relatively mature complementary market;
- **Value creation**: the activities geared toward increasing the perceived attractiveness of the platform ecosystem among customers and measure it as changes to complement popularity among customers. (proxy variable)

Identification Techniques:

- Use the *the number of reviews* as the proxy for the popularity of complements (dependent variables);
- *functional specificity* measures the heterogeneity of a complement based on the complexity of services offered by the compliment;
- Follows [Zhu and Liu, 2018] to account for platform-specific investments by *interfacce coupling*: whether the complement connects with the platform core;
- To verify the exogeneity of Amazon's entry decision, a logit regression is conducted to test the number of reviews (popularity) does not influence Amazon's decision;
- PSM method is adopted.

$$Reviews_{it} = \alpha + \beta Treated_i \times After_t + \delta Controls_{it} + C_i + T_i + \epsilon_{it} \tag{4.5.2.1}$$

Many papers analyze the platform owner's entry from a game-theory perspective.
[Hagiu et al., 2020] investigate the owner's entry decision in the complementary markets.

Assumption 4.5.2.1: Creating platforms by hosting rivals

- There are two companies M and S: M sells product A and B_M, S can only sell B_S;
- The costs to produce are all set to zero;
- The number of customers is normalized to one, λ_A customers are only interested in A (A-type), $1-\lambda_A$ are interested in both (B-type);
- $u_A > 0$ for both type of customers, $u_B > 0$ and $u_S = u_B + \Delta$ for B-type;
- There are costs for consumers to go to each store with a cost σ, and $0 < \sigma < \min\{u_A, u_B\}, \Delta < \sigma$.

The authors analyze two conditions: without hosting and hosting. By solving the equilibrium given two conditions, they find the following conclusion:

1. In the without hosting condition, M would dominant the whole market, the profit for S is zero;
2. Even without transfer for S to sell products in the platform M, M can still benefit if $\lambda_A \leq \sigma/u_A$ and $\Delta > \lambda_A(u_A - \sigma)^+ F/(1-\lambda_A)$.

[Cheng et al., 2022] analyze the owner's entry of the incumbent sellers in the E-commerce platform.

Assumption 4.5.2.2: Sell-on Contract

Without the entry, the demands for the two incumbent sellers are:

$$\begin{aligned} D_1^s &= \frac{1}{2}[1 - p_1^s + \theta(p_2^s - p_1^s)] \\ D_2^s &= \frac{1}{2}[1 - p_2^s + \theta(p_1^s - p_2^s)] \end{aligned} \tag{4.5.2.2}$$

After the entry, it is modified to:

$$\begin{aligned} D_r^o &= \frac{1}{2+a}\left[a - p_r^o + \frac{1}{2}[\delta(p_1^o - p_r^o) + \delta(p_2^o - p_r^o)]\right] \\ D_1^o &= \frac{1}{2+a}\left[1 - p_1^o + \frac{1}{2}[\theta(p_2^o - p_1^o) + \delta(p_r^o - p_1^o)]\right] \\ D_2^o &== \frac{1}{2+a}\left[1 - p_2^o + \frac{1}{2}[\theta(p_1^o - p_2^o) + \delta(p_r^o - p_2^o)]\right] \end{aligned} \tag{4.5.2.3}$$

Where a captures the strength of pltform's own brand. θ, δ are the cross sensitivity.

They find that in both the sell-to and sell-on conditions, the entry of the platform owner may not harm the incumbent sellers. What's more, comparing to the entry of a new third-party sellers, the incumbent sellers profit more when facing the entry of the owner. [Lai et al., 2022] investigates the introduction of Amazon's fulfillment program (FBA) on the third-party sellers in the E-commerce platform.

Assumption 4.5.2.3: FBA on the third-party sellers

- Both Amazon and the third party sell substitute products, at price of p_A and p_S;
- Amazon procure the products from OEM supplier at price w with a cost c_0, the third-party obtain its products with a cost g;
- The third party pays r share of its revenue to Amazon as commission;
- The delivery cost for different means are all c, the Amazon provides a better service $S_A > S_s$, the third party can pay $T > c$ to use FBA;

Without FBA, the demands are given by:

$$\begin{aligned} q_A &= Q_A - p_A + \beta p_S + \alpha s_A - \eta s_S, \\ q_S &= Q_S - p_S + \beta p_A + \alpha s_S - \eta s_A, \end{aligned} \tag{4.5.2.4}$$

With FBA, the demands are given by:

$$\begin{aligned} q_A &= Q_A - p_A + \beta p_S + \alpha s_A - \eta s_A, \\ q_S &= Q_S - p_S + \beta p_A + \alpha s_A - \eta s_A. \end{aligned} \tag{4.5.2.5}$$

They assume are informations are common knowledge.

There are three decision makers: Amazon: p_A, T, Third-party: p_S, FBA, OEM: w:

$$\begin{aligned}\Pi_{A,N} &= (p_A - w - c)q_A(p_A, p_S) + rp_S q_S(p_A, p_S)\\ \Pi_{S,N} &= [(1-r)p_S - g - c]q_S(p_A, p_S)\\ \Pi_{M,N} &= (w - c_0)q_A(p_A, p_S)\end{aligned} \tag{4.5.2.6}$$

To given insights which policy would benefit different parties, they took the following steps:

1. In the without-FBA condition, given w, solve the equilibrium for $p_A^\star(w), p_S^\star(w)$;
2. Substituting them into the decision model of OEM, solve the optimal $w^\star$ in the SOSC condition.
3. In the FBA condition, repeat above steps, solve the equilibrium for $p_A^\star(w), p_S^\star(w), w^\star$;
4. Substituting them into the decision model of OEM, solve the optimal $T^\star$.

They find some interesting insights:

1. The third-party benefits from FBA when $T < \bar{T}$;
2. Amazon can benefit from FBA if η is either small or large;
3. The FBA program can achieve a *win-win-win* outcome for the third-party seller, Amazon and its OEM supplier.

4.5.3 Consumer Polarization

Consumer polarization is a topic in consumer research. **Group-Polarization Hypothesis** suggests that group discussion generally produces attitudes that are more extreme in the direction of the average of prediscussion attitudes in a variety of situations. Works like [Rao and Steckel, 1991] provides a mathematical presentation for this phenomenon (in the domain of preference):

$$U_{\mathrm{s}} = \sum_{i=1}^{m} \lambda_i u_i + \phi(\bar{u} - K) 0 \le \lambda_i \le 1, \sum_{i=1}^{m} \lambda_i = 1, \phi \ge 0 \tag{4.5.3.1}$$

In this model, the $\bar{u}$ is the algebraic mean of all consumers' utility, and K is the **Pivot Point**. Rewrite this formula:

$$\begin{aligned} U_{\mathrm{g}} &= \sum_{i=1}^{m} \left(\lambda_i + \frac{\phi}{m}\right) u_i - \phi K \qquad && w_0 \le 0;\\ &= w_0 + w_1 u_1 + w_2 u_2 + \cdots + w_m u_m && \sum_{i=1}^{m} w_i \ge 1;\\ & && 0 \le \tfrac{w_0}{1 - \sum w_i} \le 1. \end{aligned} \tag{4.5.3.2}$$

Remark 4.5.1

- [Zhao et al., 2023] use experiment results to suggest that eWOM (electronic word of mouth) polarization (the degree of eWOM to which positive and negative sentiments are simultaneously strong) would decrease the consumers' intention to purchase, mediating by the enhancement of attitude ambivalence. (Ambivalence is a psychological state where a person endorses both positive and negative attitudinal positions)
- [Iyer and Yoganarasimhan, 2021] use game theory framework, to get the conclusion that sequential decision-making could reduce the polarization.

4.5.4 Network Effect

Network Effect and **Network Externality**:

[Narayan et al., 2011] verifies that peer influence affects attribute preferences via a Bayesian updating mechanism. In their model, the utility is given as follows:

$$U_{ijp}^{R} = X_{jp}\beta_{i}^{R} + \lambda_{i}\varepsilon_{ijp}^{R} \tag{4.5.4.1}$$

Where U_{ijp} is the utility of consumer i for product j given choice set p, X_{jp} is the attribute of product j in the choice set p, β_{i}^{R} is the customers' weights. The Bayesian updating process is given below:

$$\beta_{ik}^{R} = \rho_{ik}\beta_{ik}^{l} + (1-\rho_{ik})\frac{\sum_{i=1,i\neq i}^{N} w_{ii}\beta_{ik}^{l}}{\max\left[\left(\sum_{i=1,i\neq i}^{N} w_{ii}\right), 1\right]}, \quad \text{where } 0 \leq \rho_{ik} \leq 1. \tag{4.5.4.2}$$

Other research on peer influence:

Remark 4.5.2

The consumers' interaction and social connections have a proposition proposed in [Zhang et al., 2017] for their goal attainment and spending: a positive linear term plus a negative squared term;

4.5.5 Online Gaming

Many industrial news about online gaming can be found in [Chen et al., 2017]. In [Lei, 2022], the dissertation fully discussed loot box pricing, matchmaking, and price discrimination with fairness constraints.

Play-Duration and Spending

[Zhang et al., 2017]'s work shows a nonlinear effect of social connections and interactions on consumers' goal attainment and spending: A positive linear terms and a negative squared term. Mechanism: functional in providing useful information or tips that can facilitate goal attainment, but would raise information overload problems.

Player engagement can be embodied by many specific metrics, such as time or money spent in the game, the number of matches played within a time window, or churn risk. [Chen et al., 2017] define churn risk as the proportion of total players stopping playing the game over a period of time.

Matchmaking

Matchmaking connects multiple players to participate in online PvP games. (PvP(Player-versus-Player) games, which cover many popular genres, such as multiplayer online battle arena (MOBA), first-person shooting (FPS), and e-sports, have increased worldwide popularity in recent years.)

The past matchmaking strategy matches similar skilled players in the same round (SBMM), the current MM system focuses on improving the players' engagement and decreasing the churn rate. For example, in [Chen et al., 2017] EOMM (Engagement Optimization MatchMaking) is proposed to minimize the churn rate.

[Chen et al., 2021] propose an algorithm to maximize the cumulative active players.

Assumption 4.5.5.1: Chen 2021 MatchMaking

1. players can have heterogeneous skill levels: level 1 to level K;
2. the outcome of each match is a Bernoulli random variable: $p_{kj} = 1 - p_{jk}, p_{kk} = 0.5$, $p_{kj} > 0.5$ if $k > j$;
3. player's skill level is fixed: *relative* level;
4. and their state depends on the win-loss outcomes of the past m matches: $g \in \mathcal{G}$ ($2^m + 1$ possible cardinality);
5. A geometric losing churn model: players churn with a fixed probability, starting from the second loss in a row;
6. $P^k_{win}, P^k_{lose} \in [0,1]^{|\mathcal{G}|\times|\mathcal{G}|}$ is the transition matrix of level k player's engagement state;
7. $M_{kj} = p_{kj}P^k_{win} + (1-p_{kj})P^k_{lose}$ is the aggregate transition matrix. ($\bar{G}$ is the reduced aggregate transition matrix);
8. using the fluid matching model and assume players are infinitely divisible;

The **Dynamic Programming** formulation: $f_{kg,jg'}$ is the amount of kg players matched with jg' players, s^t_{kg} is the number of kg players at time t.
FB *flow balance* constraints:

$$\begin{aligned}
\sum_{j=1}^{K}\sum_{g'\in\bar{\mathcal{G}}} f^t_{kg,jg'} &= s^t_{kg}, k = 1,\dots,K, \forall g \in \bar{\mathcal{G}}, \\
\sum_{j=1}^{K}\sum_{g'\in\bar{\mathcal{G}}} f^t_{jg',kg} &= s^t_{kg}, k = 1,\dots,K, \forall g \in \bar{\mathcal{G}}, \\
f^t_{kg,jg'} &= f^t_{jg',kg}, j = 1,\dots,K, k = 1,\dots,K, \forall g \in \bar{\mathcal{G}}, g' \in \bar{\mathcal{G}} \\
f^t_{kg,jg'} &\geq 0, j = 1,\dots,K, k = 1,\dots,K, \forall g \in \bar{\mathcal{G}}, g' \in \bar{\mathcal{G}}
\end{aligned} \tag{4.5.5.1}$$

ED *evolution of demographics*:

$$\mathbf{s}^{t+1}_k = \sum_{j=1,\dots,K} \left(\mathbf{f}^t_{kj}\mathbf{1}\right)^\top \left(\bar{M}_{kj} + N_k\right) k = 1,\dots,K \tag{4.5.5.2}$$

The value-to-go function is:

$$\begin{aligned}
V^\pi(\mathbf{s}^t) = \sum_{k=1}^{K}\sum_{g\in\bar{\mathcal{G}}} s^{t+1}_{kg} + \gamma V^\pi(\mathbf{s}^{t+1}) \\
\text{subject to (FB), (ED).}
\end{aligned} \tag{4.5.5.3}$$

The above model can be formulated in a linear programming style:

Theorem 4.5.5.1: Chen 2021 MM LP Formulation

$$
\begin{aligned}
V^*(\mathbf{s}^0) = \max & \sum_{t=1}^{\infty} \gamma^{t-1} \sum_{k} \sum_{g \in \bar{\mathcal{G}}} s_{kg}^t \\
& \text{s.t.} \sum_{j=1}^{K} \sum_{g' \in \bar{\mathcal{G}}} f_{kg,jg'}^t = s_{kg}^t, \forall k, \forall g \in \bar{\mathcal{G}}, t = 0, 1, \dots \\
& \sum_{j=1}^{K} \sum_{g' \in \bar{\mathcal{G}}} f_{jg',kg}^t = s_{kg}^t, \forall k, \forall g \in \bar{\mathcal{G}}, t = 0, 1, \dots \\
& f_{kg,jg'}^t = f_{jg',kg}^t, j = 1, \dots, K, k = 1, \dots, K, \forall g \in \bar{\mathcal{G}}, g' \in \bar{\mathcal{G}}, t = 0, 1, \dots \\
& f_{kg,jg'}^t \geq 0, j = 1, \dots, K, k = 1, \dots, K, \forall g \in \bar{\mathcal{G}}, g' \in \bar{\mathcal{G}}, t = 0, 1, \dots \\
& \mathbf{s}_k^{t+1} = \sum_{j=1,\dots,K} \left(\mathbf{f}_{kj}^t \mathbf{1} \right)^\top \left(\bar{M}_{kj} + N_k \right), \forall k, t = 0, 1, \dots
\end{aligned}
\tag{4.5.5.4}
$$

Remark 4.5.3

- Using an optimal matchmaking policy instead of SBMM may reduce the required bot ratio significantly while maintaining the same level of engagement.

4.6 Behavioral Operations Management

[Donohue et al., 2020] surveys the behavioral issues in the OM literature: the behavioral OM studies the difference between the **normative models** and the actual observed behaviors. Two important examples:

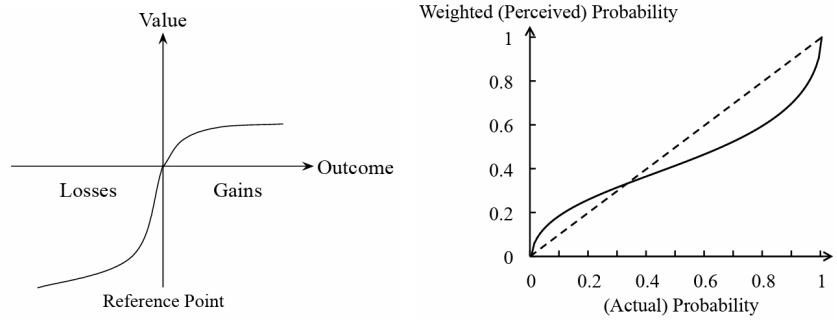
- pull-to-center bias: the participants' ordering decisions are systematically pulled toward mean demand and away from the "optimal" ordering decision (the mechanism is not unified);
- The influence of *trust* on information sharing in the distributed supply chain

4.6.1 Behavioral Economics

Individual choice theories

Prospect Theory

There are many cases where the consumers' choice is inconsistent with the expected utility theory. The key idea of the prospect theory is that people evaluate an outcome based on the comparison of the outcome with some subjective reference point, rather than based on the absolute outcome itself: the evaluation of a lottery relies on a value function and a nonlinear probability function:



(a) Value Function (b) Probability Weighting Function

Figure 4.8: The Valuation in Prospect Theory

- the valuation of the outcomes of a lottery is classified as gains or losses compared to the reference point: consumers are usually 'risk aversion' in the gain domain (concavity) and 'risk seeking' in the loss domain (convexity);
- the consumers tend to overreact to extreme events that have small probabilities (see the solid curve above the dashed line).

Framing effetcs: different presentations of the same choice problem may yield reversed preferences. **Endowment effects**: People value a product or service more once they establish property rights to it.

Choice Over Time

The **hyperbolic** discounting model consider a sequence of dated consumptions denoted by $\{(x_i, t_i), i = 1, \ldots, n\}$, the utility is given by:

$$U(x_1, t_1; \ldots; x_n, t_n) = \sum_{i=1}^{n} v(x_i)\phi(t_i). \tag{4.6.1.1}$$

$v(\cdot)$ is the value function with the same structural properties as the one in prospect theory, and $\phi(\cdot)$ is the discount function with a hyperbolic form: $\phi(t) = (1 + \alpha t)^{-\beta/\alpha}$, with $\alpha, \beta > 0$. When $\alpha \to 0$, the hyperbolic form reverts to the standard exponential discount function: $\phi(t) = e^{-\beta t}$ (which is more tractable).

Reciprocity and Inequality Aversion

Example 4.6.1.1

Consider the bargaining game with two players: a proposer (he) and a responder (she). He is provided with an endowment X, and he offers an amount $Y \in [0, X]$ to her, and she decides whether to accept it. The unique SPNE is to offer the smallest amount while accepting all the offers. However, the experiments give contradictory results.

Solution:

There are two explanations for this phenomenon:

- ***Reciprocity:*** *proposers make high offers because they fear rejections of low offers by responders;*
- ***Fairness****: proposers make high offers because they are purely generous.*

The inequality aversion model captures the fairness: for n individuals, player i's utility is given by:

$$U_i(x) = x_i - \frac{\alpha_i}{n-1}\sum_{j\neq i}\max\{x_j - x_i, 0\} - \frac{\beta_i}{n-1}\sum_{j\neq i}\max\{x_i - x_j, 0\} \tag{4.6.1.2}$$

where α is the envy weight and β is the guilt weight: $0 \leq \beta_i \leq 1$ and $\beta_i \leq \alpha_i$ because people are more averse to being worse o than being better off.

Trust and Reputations

Complexity and Bounded Ratinoality

Definition 4.6.1.1: Level-k Thinking

A level-0 player is a 'native' type who picks choices randomly. A level-1 player believes that other players are level-0 type and best responds to the level-0 types strategy. A similar definition for level-k thinking.

4.6.2 Behavioral Revenue Management

Chapter 20 in [Özer and Phillips, 2012] discussed how human deviations affect pricing management. The first stream studies how individual choice theory affects consumer pricing.

Context effects

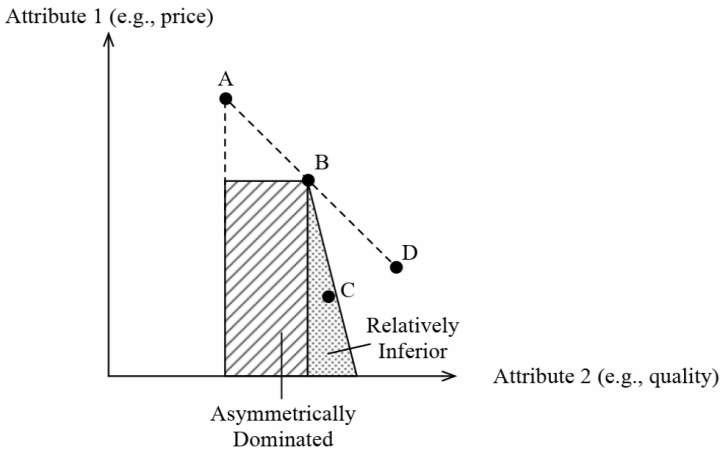


Figure 4.9: Demonstration of Attraction Effect and Compromise Effect

The attraction effect happens when adding an *asymmetrically dominated* product or a *relatively inferior* product (comparing with product i) in the assortment, the selection probability of i increases.
There is another kind of context effect called the compromise effect: suggesting that the choice probability of an alternative can be increased by making the alternative become a compromise/an intermediate alternative in the choice set. This effect is a consequence of loss aversion implied by prospect theory.

- the product C can lead an attraction effect for B;
- the product D can lead a compromise effect for B.

The last context effect is similarity effect: consider three products A, B, C, where B and C share more similarity. For assortment A, B, the inclusion of new product C could decrease the purchasing probability of B more than C because they are more similar.

Price Presentation Effects

One well-known example is the **price-ending effect**. The digits 0, 5, and 9 are the most frequently used rightmost (i.e., ending) digits in retail prices, with the digit 9 especially dominant for discounted prices.
There is another stream of literature studying how social preferences affect pricing and how behavioral issues can influence the B2B contract.

[Kocabiyikoglu et al., 2015] study and compare decision-making behavior in an experimental setting under the newsvendor model and the two-fare revenue management model (the two models have a clear analytical solution). Under the two models, the decision-maker needs to decide on the *order quantity* or the *protection level*:

$$x^* = F^{-1}\left(\frac{c_U}{c_U + c_O}\right), \tag{4.6.2.1}$$

where F is the distribution (class 1 for RM), c_U is the underage cost and c_O is the overage cost. So the analytical solutions for the models are $x^* = F^{-1}\left(\frac{p-c}{p}\right) = F^{-1}\left(\frac{p_1-p_2}{p_1}\right)$ respectively. The authors employ a 2×2 design in the experiments: NV/RM $times$ high/low-cost, and adapt the uniform distribution on the demand. They ask several participants to play the games for several rounds, the main results show that in the low-cost condition, the decision-makers tend to make decisions with smaller values than the analytical solution, and vice versa. They analyze the difference in decisions between the RM and NV problem: RM is always higher than NV, which indicates

that the decision-maker identifies the perceived losses due to leftovers as the main driver of the behavioral patterns.

Choice Model & Assortment Optimization Considering Behavioral Issues

[Wang, 2018] solves the assortment optimization problem considering the prospect theory. Define $r(S, \mathbf{p})$ as the reference price for assortment S when the current price vector is $\mathbf{p}$, and the utility for each product i can be expressed by:

$$U_i = \alpha_i - \beta p_i + \lambda(r(S,\mathbf{p}) - p_i)^+ - \gamma(p_i - r(S,\mathbf{p}))^+ + \xi_i. \tag{4.6.2.2}$$

$\lambda < \gamma$ ($\lambda > \gamma$) indicates the "loss-averse" ("gain-seeking") effect respectively, the utility of no-purchase is $U_0 = \xi_0$. After obtaining the utility under the reference price, the consumers follow an MNL model (Gumbel distribution for ξ_i). The formation of the reference price could be varied: highest, average, etc, and the model does not suffer from IIA because a new product into the assortment could influence the selection probability by changing the reference price. For an assortment S at prices $\mathbf{p}$, the *consumer surplus* is defined as:

$$\log\left(1 + \sum_{i\in S} \exp(\alpha_i - \beta p_i + \lambda(r(S,\mathbf{p}) - p_i)^+ - \gamma(p_i - r(S,\mathbf{p}))^+)\right) \tag{4.6.2.3}$$

Now considering the lowest price as the reference price: $r_l(S,\mathbf{p}) = \min_{k\in s} p_k$, and the selecting probability is:

$$q_i(S,\mathbf{p};r_l) = \frac{\exp(\alpha_i - (\beta+\gamma)p_i + \gamma \min_{k\in S} p_k)}{1 + \sum_{j\in S}\exp(\alpha_j - (\beta+\gamma)p_j + \gamma \min_{k\in S} p_k)}. \tag{4.6.2.4}$$

Proposition 4.6.2.1

For the lowest price reference, the choice probability $q_i(S,\mathbf{p};r_l)$ is decreasing in p_i and is increasing in p_j for any $j \in S, j \neq i$.

Using this framework, one may find that sometimes increasing the price of the products (with the lowest price) can increase the total welfare.

Assortment Planning and Pricing

First, assume the prices of all products are predetermined, so the problem can be formulated by:

$$\max_{S\subseteq N} \mathbb{R}(S,\mathbf{p};r_l).$$

In this case, a quasi-markup-ordered assortment is optimal:

Definition 4.6.2.1: Quasi-markup-ordered assortment

Suppose that products are labeled in the price-decreasing order: $p_1 \geq p_2 \geq \cdots \geq p_N$. For any $2 \leq n \leq N$, relabel all products in $S_{n-1} := \{1, 2, \cdots, n-1\}$ in the markup(revenue)-decresing order. For any $i \leq n-1$, the subset $\{k : k \leq i, k \in S_{n-1}\} \cup \{n\}$ is called a quasi-arkup-ordered assortment.

Proof 4.6.1: QMOA Optimal

Since the reference is the lowest price, then there are N circumstances. For each $n = 1, \cdots, N$m the firm

need to select the other products from S_{n-1}:

$$\max_{S \subseteq S_{n-1}} R(S \cup \{n\}, \mathfrak{p}; r_l). \tag{4.6.2.5}$$

Then, the choice probability for each product i is:

$$q_i(\mathcal{S} \cup \{n\}, \mathbf{p}; r_l) = \frac{\exp(\alpha_i - (\beta+\gamma)p_i + \gamma p_n)}{1 + \sum_{j \in S \cup \{n\}} \exp(\alpha_j - (\beta+\gamma)p_j + \gamma p_n)} \tag{4.6.2.6}$$

Let $\max_{S \subseteq S_{n-1}} R(S \cup \{n\}, \mathbf{p}; r_l) = x$. We can write:

$$\begin{aligned} x = H_n(x) := (p_n - c_n - x) \cdot \exp(\alpha_n - \beta p_n) \\ + \max_{S \subseteq S_{n-1}} \sum_{i \in S} (p_i - c_i - x) \cdot \exp(\alpha_i - (\beta+\gamma)p_i + \gamma p_n). \end{aligned} \tag{4.6.2.7}$$

Note the RHS has a familiar structure and for every n, there is a revenue-order assortment.

Q.E.D.

There are two parameters needed for searching the optimal assortment: the reference price parameter n and the revenue order parameter i, so this assortment optimization costs $O(n \log n)$. The author also considers the joint problem of optimizing the assortment as well as determining the price:

$$\max_{S, \mathbf{p}} R(S, \mathbf{p}; r_l). \tag{4.6.2.8}$$

This objective function is not jointly concave even without the reference price.

Definition 4.6.2.2: Same-markup/Same-price policy

Label the products using cost, $c_1 \geq c_2 \geq \cdots \geq c_N$. For $1 \leq n \leq N$, products $1, 2, \cdots, n$ are referred as high-cost products, whereas products $n+1, n+2, \cdots N$ are called low-cost products. The pricing policy that charges the same markup for high-cost products and the same price for low-cost products is called the same-markup/same-price policy.

Theorem 4.6.2.1: The Optimal Policy

It is optimal to offer all products, and the same-markup/same-price policy is optimal.

Proof 4.6.2

Assuming that product i is not included in the optimal assortment, it's obvious to note that if including i, by setting its price and revenue sufficiently high, the revenue would increase, so the optimal assortment is N. Let x^* as the optimal profit:

$$x^* = \max_{\mathbf{p}} \sum_{i \in \mathcal{N}} (p_i - c_i - x^*) \cdot \exp\left(\alpha_i - (\beta + \gamma)p_i + \gamma \min_{k \in \mathcal{N}} p_k\right). \tag{4.6.2.9}$$

We use y to substitute $\min \mathbf{p}$, and the optimization can be transform to:

$$\max_{\mathbf{p} \geq y} \sum_{i \in \mathcal{N}} (p_i - c_i - x^*) \cdot \exp(\alpha_i - (\beta + \gamma)p_i + \gamma y). \tag{4.6.2.10}$$

This means that we can optimize p_i for each i separately, and the optimal solution is given by

$$p_i = \max(c_i + x^* + 1/(\beta + \gamma), y).$$

Q.E.D.

Following the conclusion of the last theorem, the AOP can be transformed to solving N two-dimensional problem (for each n):

$$\begin{aligned} \max_{m + c_n \geq p} &\left(\sum_{i=1}^{n} m \cdot \exp(\alpha_i - (\beta + \gamma)(m + c_i) + \gamma p) + \sum_{i=n+1}^{N} (p - c_i) \cdot \exp(\alpha_i - \beta p) \right) \\ &\cdot \left(1 + \sum_{j=1}^{n} \exp(\alpha_j - (\beta + \gamma)(m + c_j) + \gamma p) + \sum_{j=n+1}^{N} \exp(\alpha_j - \beta p) \right)^{-1}. \end{aligned} \tag{4.6.2.11}$$

However, for a given n, the problem is not jointly concave in p and m, so we need an efficient algorithm. Let $R(\mathcal{N}, \mathbf{p}; r_l) = x$, we can get:

$$\begin{aligned} \boldsymbol{x} =& \Gamma(x; \gamma) \\ :=& \max_{\mathbf{p}} \sum_{i \in \mathcal{N}} \left\{ (p_i - c_i - x) \exp\left(\alpha_i - (\beta + \gamma)p_i + \gamma \min p_k\right) \right\}. \end{aligned} \tag{4.6.2.12}$$

Note that $\Gamma(x; \gamma)$ is decreasing in x, the efficient algorithm contain two parts: first compute $\Gamma(x; \gamma)$, then use the bisection method to search x^* such that $x^* = \Gamma(x^*; \gamma)$.

[Rooderkerk et al., 2011] is the first paper incorporating context effects into the **attribute-based** choice model. Given participant h (can be understood as the response parameter heterogeneity), item i and the choice set S, the utility z_{hi}^S is defined by:

$$z_{hi}^S = \underbrace{V_{hi}}_{\text{partworth utility}} + \underbrace{VC_{hi}^S}_{\text{context-dependent utility}} + \varepsilon_{hi}^S \tag{4.6.2.13}$$

Assume every item can be described along Q attributes, each attribute has L_q levels. Denote X_{iql} as the dummy indicating whether item i has level l of attribute q, the partworth utility is given by:

$$V_{hi} = \sum_{q=1}^{Q} \sum_{l=2}^{L_q} \beta_{hql} x_{iql}, \tag{4.6.2.14}$$

and the contextual component covers all context effects: compromise, attraction, and similarity:

$$\mathrm{VC}_{\mathrm{hi}}^{\mathrm{S}} = \beta_{\mathrm{h}}^{\mathrm{COM}}\mathrm{COM}_{\mathrm{i}}^{\mathrm{S}} + \beta_{\mathrm{h}}^{\mathrm{ATT}}\mathrm{ATT}_{\mathrm{i}}^{\mathrm{S}} + \beta_{\mathrm{h}}^{\mathrm{SIM}}\mathrm{SIM}_{\mathrm{i}}^{\mathrm{S}}. \tag{4.6.2.15}$$

When the parameters β are all zero, this is reduced to a RUM. The utility of no-choice is set as $\mathrm{z}_{\mathrm{h0}}^{\mathrm{S}} = \beta_{\mathrm{h0}} + \varepsilon_{\mathrm{h0}}^{\mathrm{S}}$. The measure of the three effects is based on the Euclidean distance within the attribute space, x_{kq} is the level of attribute q for product k, define the middle point M as:

$$\mathrm{x}_{\mathrm{Mq}}^{\mathrm{S}} = \frac{\min_{\mathrm{k}\in\mathrm{S}}\mathrm{x}_{\mathrm{kq}} + \max_{\mathrm{k}\in\mathrm{S}}\mathrm{x}_{\mathrm{kq}}}{2}, \quad \mathrm{q} = 1, ..., \mathrm{Q}. \tag{4.6.2.16}$$

the closer one item from M, the larger the compromise effect, so $COM_i^s = -d_{iM}^S$. To measure the attraction effect, we need the *preference vector* to normalize the effect, which is a Q-dimensional vector:

$$\mathrm{v}_{\mathrm{preference}}^{\mathrm{S}} = \left[\left(\max_{\mathrm{k}\in\mathrm{S}}\mathrm{x}_{\mathrm{k1}} - \min_{\mathrm{k}\in\mathrm{S}}\mathrm{x}_{\mathrm{k1}}\right) ... \left(\max_{\mathrm{k}\in\mathrm{S}}\mathrm{x}_{\mathrm{kQ}} - \min_{\mathrm{k}\in\mathrm{S}}\mathrm{x}_{\mathrm{kQ}}\right)\right] \tag{4.6.2.17}$$

Now the attraction effect can be defined as $d_{ij,preference}^S$ if item i dominates item j otherwise $-d_{ij,preference}^S$. The similarity effect is given by $\mathrm{SIM}_{\mathrm{i}}^{\mathrm{S}} = \min_{\mathrm{j}\in\mathrm{S}/\{\mathrm{i}\}}\mathrm{d}_{\mathrm{ij,position}}^{\mathrm{S}}$, where $V_{position}^S$ is a vector independent of $v_{preference}^S$. The authors further include the interaction effect between attraction effect and the similarity effect. Empirical results show that under most cases the new model has a higher out-of-sample hit rate than the simple RUM model.

[Yousefi Maragheh et al., 2020] proposes the CMNL (contextual MNL) model incorporating three context effects and its assortment optimization. The choice model assumes the deterministic utility f_i^S for item i within assortment S can be divided into two parts:

$$f_i^S = \mu_i + \sum_{j\in S}\alpha_{ji}, i \in S, \tag{4.6.2.18}$$

where μ_i is the *baseline utility*, and α_{ji} is used to capture the three context effect of product j on i. The utility of no-purchase is normalized to $f_0^s = 0$. Thus, the probability of selecting j among S is given by:

$$P_j(\mathcal{S}) = \begin{cases} \frac{\exp\left(\mu_j + \sum_{i\in S}\alpha_{ij}\right)}{1+\sum_{l\in S}\exp\left(\mu_l + \sum_{i\in S}\alpha_{il}\right)} & \text{if } j \in \mathcal{S}, \\ 0 & \text{if } j \notin \mathcal{S}, \end{cases} \tag{4.6.2.19}$$

Denote x as the vector indicating whether i is in S, the probability can be written in:

$$P_j(x) = \frac{x_j \exp\left(\mu_j + \sum_{i=1}^N x_i\alpha_{ij}\right)}{1 + \sum_{l=1}^N x_l \exp\left(\mu_l + \sum_{i=1}^N x_i\alpha_{il}\right)}. \tag{4.6.2.20}$$

Matrix $A \in \mathbb{R}^{n\times n}$ is the matrix for α_{ji}, now the assortment optimization can be formulated as:

$$\max_{x\in\{0,1\}^N} r(x, \mu, A) = \max_{x\in\{0,1\}^N}\sum_{i=1}^N x_i r_i P_i(x), \tag{4.6.2.21}$$

where r_i is the revenue for product i. When $r_i = 1, \forall i$, this problem becomes a CTROP (Click Through Rate Optimization Problem), which is widely used in online recommendations:

$$\max_{x\in\{0,1\}^N} M(x, \mu, A) = \max_{x\in\{0,1\}^N}\sum_{i\in\mathcal{N}} x_i P_i(x). \tag{4.6.2.22}$$

The CMNL behaves well in empirical validation (using AIC and BIC scores), but both the AOP and CTROP are NP-hard.

In some scenarios, the assortment may have a 'star' product, consumers would over-evaluate its utility, this is called the focal effect. [Kovach and Tserenjigmid, 2022] design a new choice model (focal luce model: FLM) that considers this phenomenon:

$$v(i,\mathcal{S}) = v_i + \log\left(1 + \delta(\mathcal{S}) \cdot \mathbf{1}\{i \in F(\mathcal{S})\}\right) + \epsilon_i, \tag{4.6.2.23}$$

where $F(\cdot)$ is the *focal function* indicating the star products given the set $\mathcal{S}$, and $\delta(\cdot)$ is the *distortion function* indicating the over-evaluation for the star products.

Remark 4.6.1

- FLM is not a RUM (random utility model);
- FLM does not describe the focal function and the distortion function.

[Jiang et al., 2023] consider the assortment optimization under the FLM:

$$\max_{\mathcal{S}\subseteq\mathcal{N}} f(\mathcal{S}) := \frac{\sum_{i\in\mathcal{S}} r_i u_i(\mathcal{S})}{\sum_{i\in\mathcal{S}_+} u_i(\mathcal{S})} \tag{4.6.2.24}$$

The first scenario they consider is the ranking-based focal effect, let $\sigma(i,\mathcal{S})$ be the rank of item i's utility in $\mathcal{S}$, the top-k focal function:

$$F(\mathcal{S}) = \{i : \sigma(i,\mathcal{S}) \le \min\{K, |\mathcal{S}|\}, i \in \mathcal{S}\},$$

and the distortion function is set to be a constant: $\delta(\mathcal{S}) = \delta$.

[Cao et al., 2022b] consider the assortment optimization in the airline industry using the spiked-MNL choice model. 'Spiked' means that each product j has two utility w_j, v_j with $w_j \ge v_j \ge 0$, if j is the cheapest among the assortment, the utility is w_j, otherwise v_j.

4.7 Data-Driven Operations Management

[?] provide a data-driven framework for firms to allocate budget portfolios on online cross-channel advertisement. They obtain data from the selling and advertising records of a globally-known CPG company on Tmall. There are 4 channels:

- Banner;
- Recommender;
- SEO-product;
- SEO-store.

And there are 3 advertisement effects:

- promotional effect;
- cross-period spillover effect;
- cross-product spillover effect.

through a reduced-form regression using the pricing records of a non-competing but analogous market as IV:

$$\begin{aligned} r_i^t = \alpha_{i,0} + \sum_{j\in\mathcal{J}} \alpha_{i,j} s_{i,j}^t + \sum_{j\in\mathcal{J}} \hat{\alpha}_{i,j} (s_{i,j}^t)^2 + \sum_{j\in\mathcal{J}} \left(\alpha_{i,j}^{pre} \delta_{pre}^t + \alpha_{i,j}^{pro} \delta_{pro}^t + \alpha_{i,j}^{pos} \delta_{pos}^t \right) s_{i,j}^t + \\ \sum_{j\in\mathcal{J}} \left(\beta_{i,j}^{\ell 1} s_{i,j}^{t-1} + \beta_{i,j}^{\ell 2} s_{i,j}^{t-2} + \beta_{i,j}^{\ell 3} s_{i,j}^{t-3} \right) + \sum_{k\in\mathcal{I}\setminus\{i\}} \sum_{j\in\mathcal{J}} \gamma_{k,j} s_{k,j}^t + \boldsymbol{\Theta}_i \cdot \boldsymbol{\Omega}_i^t + \epsilon_i, \quad i \in \mathcal{I}. \end{aligned} \tag{4.7.0.1}$$

where r_i^t is the revenue for product i at time t, $s_{i,j}^t$ is the spending of budges on product i through channel j, $s_{k,j}^t$ measures the cross-channel spillover effect. The regression results are significant. What's more, the authors utilize the parameters of the regressions and optimize them in a sequential game:

$$\begin{aligned} \max\ & \sum_{i\in\mathcal{I}} \sum_{t\in\mathcal{T}} r_i^t - \sum_{i\in\mathcal{I}} \sum_{j\in\mathcal{J}} \sum_{t\in\mathcal{T}} s_{i,j}^t \\ s.t.\ & \sum_{i\in\mathcal{I}} \sum_{j\in\mathcal{J}} \sum_{t\in\mathcal{T}} s_{i,j}^t \le B; \\ & 0 \le s_{i,j}^t \le \bar{s}_{i,j}^t, \quad i\in\mathcal{I}, j\in\mathcal{J}, t\in\mathcal{T}; \\ & r_i^t = \alpha_{i,0} + \sum_{j\in\mathcal{J}} \alpha_{i,j} s_{i,j}^t + \sum_{j\in\mathcal{J}} \hat{\alpha}_{i,j} \left(s_{i,j}^t\right)^2 + \\ & \quad \sum_{j\in\mathcal{J}} \left(\alpha_{i,j}^{pre} \delta_{pre}^t + \alpha_{i,j}^{pro} \delta_{pro}^t + \alpha_{i,j}^{pos} \delta_{pos}^t \right) s_{i,j}^t + \\ & \quad \sum_{j\in\mathcal{J}} \left(\beta_{i,j}^{\ell 1} s_{i,j}^{t-1} + \beta_{i,j}^{\ell 2} s_{i,j}^{t-2} + \beta_{i,j}^{\ell 3} s_{i,j}^{t-3} \right) + \qquad (\textit{cross-period effect}) \\ & \quad \sum_{k\in\mathcal{I}\setminus\{i\}} \sum_{j\in\mathcal{J}} \gamma_{k,j} s_{k,j}^t + \qquad (\textit{cross-product effect}) \\ & \quad \eta_i^t, \qquad i\in\mathcal{I}, t\in\mathcal{T}; \end{aligned}$$

Figure 4.10: The Optimization Problem

The simulation results give positive feedback.

4.8 Analytical Modeling (Game Theory Style)

[Dai and Singh, 2020] study a problem of the doctor's pathway behavior considering the reputational payoff.

Assumption 4.8.0.1: [Dai and Singh, 2020]

- A client has a state θ, $\theta = 1$ means positive, there is a prior on θ (Bernoulli with parameter α, and α is common knowledge);
- The expert has a private signal $S_e \in \{0, 1\}$ (two types of experts: $e \in \{l, h\}$), the precision is ρ_e: $\Pr(s_e = 0|\theta = 0) = \Pr(s_e = 1|\theta = 1) = \rho_e$. Assume that $rho_h > \rho_l > 0.5$;
- After the observing α and the private signal S_e, there are two decisions: $t \in \{0, 1\}$ denotes the testing decision, $a \in \{0, 1\}$ denotes the diagnosis decision;
- The client's utility depends on the state and expert's diagnosis decision: TP: B; FP: $-D$; TN: 0l FN: $-d$.
- Test $t = 1$ would incur a cost c for the client ($c < d$);
- Using the Bayesian rule, we can calculate the posterior belief of the expert:

$$b_e(\alpha|s_e) \triangleq \begin{cases} \dfrac{\alpha\rho_e}{\alpha\rho_e + (1-\alpha)(1-\rho_e)} & \text{if } s_e = 1, \\ \dfrac{\alpha(1-\rho_e)}{\alpha(1-\rho_e) + (1-\alpha)\rho_e} & \text{if } s_e = 0. \end{cases} \tag{4.8.0.1}$$

- We can use the posterior belief to calculate the expected utility for the clients given S_e and t, a purely altruistic expert would decide whether to conduct the test t based on the expected utility of the client;
- The expert's type is personal information, but peers can update their beliefs $\beta(t, a)$ about the expert's type, the reputational utility for the expert is $r \cdot \beta(t, a)$;
- The expert's expected payoff is given by:

$$u_e = \phi U + (1 - \phi) r\beta(t, a) \tag{4.8.0.2}$$

When the expert's type is common knowledge, it's intuitive to come up with $\bar{\alpha}_S^e$ and $\underline{\alpha}_s^e$ that the expert provides:

1. positive diagnosis without performing the test if $\alpha > \bar{\alpha}_S^e$;
2. perform the test if $\bar{\alpha}_S^e \geq \alpha \geq \underline{\alpha}_S^e$;
3. negative diagnosis without performing the test if $\alpha < \underline{\alpha}_S^e$.

In the asymmetric-information case, assuming the peers' prior belief of $e = h$ is γ, the authors prove that under certain circumstances ($\underline{B} \leq B \leq \bar{B}$), the system has a single separating NE: the type-l expert conduct the test and the type-h expert doesn't. This model assumes that given the expert's type, the precision ρ_e is fixed. [Adida and Dai, 2023] uses a similar framework to analyze whether the payment scheme would affect the expert's pathway behavior.

Chapter 5

Miscellaneous

5.1 Notes on Tools

5.1.1 Stata

coefplot

A tutorial about the use of coefplot.

5.1.2 LaTeX Shortcuts

There are 6x6 colors in the preset preamble:

aa	ab	ac	ad	ae	af
ba	bb	bc	bd	be	bf
ca	cb	cc	cd	ce	cf
da	db	dc	dd	de	df
ea	eb	ec	ed	ee	ef
fa	fb	fc	fd	fe	ff

Using `\href{URL}{text}` to refer a website.
Using `\eq` to write equation, `\tab` to get an unordered list, `\lis` to get an ordered list.

$$E = mc^2 \tag{5.1.2.1}$$

- item 1;
- item 2;
- item 3.

1. item 1;
2. item 2;
3. item 3.

Format of Words

```
\hl{highlighted}, \ul{underlined},
↪ \st{strikethrough}\\
\rt{red}, \yt{yellow}, \bt{blue}, \gt{green}
```

highlighted, underlined, ~~strikethrough~~
red, yellow, blue, green

Shortcuts

```
\RR, \NN, \ZZ, \QQ\\
\bA, \bB, \bC, \bD
```

$\mathbb{R}$, $\mathbb{N}$, $\mathbb{Z}$, $\mathbb{Q}$
$\mathbb{A}, \mathbb{B}, \mathbb{C}, \mathbb{D}$

Shortcuts

```
\tbf{Text}, \tit{Text}\\
\cA, \cB, \cC, \cD
```

Text, *Text*
A, B, C, D

Emoji

```
\emogood, \emobad, \emocool, \emoheart,
↪ \emotree
```

Use `\ass, \ax, \thm, \co, \pro, \defi, \re, \key, \ex, \proo` to use preset tcolorboxes template.

Assumption 5.1.2.1: Example

1. item 1;
2. item 2;
3. item 3.

Axiom 5.1.2.1: Exmaple
Test

Theorem 5.1.2.1: Exmaple
Test

Corollary 5.1.2.1: Exmaple
Test

Proposition 5.1.2.1: Exmaple
Test

Definition 5.1.2.1: Exmaple
Test

Remark 5.1.1

Expamle

Keywords 5.1

Example

Example 5.1.2.1

Problem example

Solution:

The solution is omitted.

Proof 5.1.1: proposition 5.1.2

The Formal Proof

Q.E.D.

Using `\label, \ref` to refer to chapters 5, sections 5.1, equations 5.1.2, and boxes 5.1.2.
Using `\cite` to cite the literature in APA style. For example: [Angrist,]
Using `\sep` to insert a horizontal line with words in the middle:

Compilation

Put photos in the *pic* file and use `\fig` to show it.

$$h(x) = \binom{n}{x},\ c(p) = (1-p)^n,\ t(x) = x \text{ and } w(p) = \log\left(\frac{p}{1-p}\right).$$

Then,

$$\begin{aligned}
&\frac{\mathrm{d}}{\mathrm{d}p}w(p) = \frac{\mathrm{d}}{\mathrm{d}p}\log\left(\frac{p}{1-p}\right) = \frac{1}{p(1-p)},\\
&\frac{\mathrm{d}^2}{\mathrm{d}p^2}w(p) = -\frac{1}{p^2} + \frac{1}{(1-p)^2} = \frac{2p-1}{p^2(1-p)^2},\\
&\frac{\mathrm{d}}{\mathrm{d}p}\log\left(c(p)\right) = \frac{\mathrm{d}}{\mathrm{d}p}n\log(1-p) = -\frac{n}{1-p},\\
&\frac{\mathrm{d}^2}{\mathrm{d}p^2}\log\left(c(p)\right) = -\frac{n}{(1-p)^2}.
\end{aligned}$$

Therefore, from Theorem 3.4.2, we have

$$\begin{aligned}
&\mathrm{E}\left(\frac{1}{p(1-p)}X\right) = \frac{n}{1-p} \ \Rightarrow\ \mathrm{E}(X) = np,\\
&\mathrm{Var}\left(\frac{1}{p(1-p)}X\right) = \frac{n}{(1-p)^2} - \mathrm{E}\left(\frac{2p-1}{p^2(1-p)^2}X\right) \ \Rightarrow\ \mathrm{Var}(X) = np(1-p).
\end{aligned}$$

Figure 5.1: Example.png

Using `\alg` to write the pseudo code:
Using `\begin{python}`, `\begin{stata}`, `\begin{shell}` to write codes:

```
1    print("Hello!")
```

Algorithm 1 Example Code

Require: $n \geq 0$
Ensure: $y = x^n$
  $y \leftarrow 1$
  $X \leftarrow x$
  $N \leftarrow n$
  while $N \neq 0$ **do**
    if N is even **then**
      $X \leftarrow X \times X$
      $N \leftarrow \frac{N}{2}$ ▷ This is a comment
    else if N is odd **then**
      $y \leftarrow y \times X$
      $N \leftarrow N - 1$
    end if
  end while

```
    ss install reghdfe
```

```
    git push origin main
```

List of Figures

List of Algorithms

Bibliography

[Abada and Lambin, 2023] Abada, I. and Lambin, X. (2023). Artificial intelligence: Can seemingly collusive outcomes be avoided? *Management Science*.

[Adida and Dai, 2023] Adida, E. and Dai, T. (2023). Impact of physician payment scheme on diagnostic effort and testing. *Management Science, Forthcoming*.

[Agarwal et al., 2019] Agarwal, A., Jiang, N., Kakade, S. M., and Sun, W. (2019). Reinforcement learning: Theory and algorithms. *CS Dept., UW Seattle, Seattle, WA, USA, Tech. Rep*, 32.

[Angrist,] Angrist, J. D. *Mostly Harmless Econometrics: An Empiricist's Companion*.

[Angrist and Pischke, 2009] Angrist, J. D. and Pischke, J.-S. (2009). *Mostly Harmless Econometrics: An Empiricist's Companion*. Princeton university press.

[Angrist and Pischke, 2014] Angrist, J. D. and Pischke, J.-S. (2014). *Mastering'metrics: The Path from Cause to Effect*. Princeton university press.

[Aouad and den Boer, 2021] Aouad, A. and den Boer, A. V. (2021). Algorithmic collusion in assortment games. *Available at SSRN 3930364*.

[Ashenfelter and Krueger, 1994] Ashenfelter, O. and Krueger, A. (1994). Estimates of the economic return to schooling from a new sample of twins. *The American economic review*, pages 1157–1173.

[Ashenfelter and Rouse, 1998] Ashenfelter, O. and Rouse, C. (1998). Income, schooling, and ability: Evidence from a new sample of identical twins. *The Quarterly Journal of Economics*, 113(1):253–284.

[Asker et al., 2022] Asker, J., Fershtman, C., and Pakes, A. (2022). Artificial intelligence, algorithm design, and pricing. In *AEA Papers and Proceedings*, volume 112, pages 452–56.

[Asker et al., 2023] Asker, J., Fershtman, C., and Pakes, A. (2023). The impact of artificial intelligence design on pricing. *Journal of Economics & Management Strategy*.

[Aslani et al., 2014] Aslani, S., Modarres, M., and Sibdari, S. (2014). On the fairness of airlines' ticket pricing as a result of revenue management techniques. *Journal of Air Transport Management*, 40:56–64.

[Banchio and Mantegazza, 2022] Banchio, M. and Mantegazza, G. (2022). Artificial intelligence and spontaneous collusion. *Available at SSRN*.

[Benoıt, 2000] Benoıt, J.-P. (2000). The Gibbard–Satterthwaite theorem: A simple proof. *Economics Letters*, 69(3):319–322.

[Bertrand and Schoar, 2006] Bertrand, M. and Schoar, A. (2006). The role of family in family firms. *Journal of economic perspectives*, 20(2):73–96.

[Bertsimas and Tsitsiklis, 1997] Bertsimas, D. and Tsitsiklis, J. N. (1997). *Introduction to Linear Optimization*, volume 6. Athena scientific Belmont, MA.

[Brusco et al., 2017] Brusco, M. J., Singh, R., Cradit, J. D., and Steinley, D. (2017). Cluster analysis in empirical OM research: Survey and recommendations. *International Journal of Operations & Production Management*, 37(3):300–320.

[Calvano et al., 2020a] Calvano, E., Calzolari, G., Denicolò, V., Harrington Jr, J. E., and Pastorello, S. (2020a). Protecting consumers from collusive prices due to AI. *Science (New York, N.Y.)*, 370(6520):1040–1042.

[Calvano et al., 2019] Calvano, E., Calzolari, G., Denicolò, V., and Pastorello, S. (2019). Algorithmic pricing what implications for competition policy? *Review of industrial organization*, 55:155–171.

[Calvano et al., 2020b] Calvano, E., Calzolari, G., Denicolo, V., and Pastorello, S. (2020b). Artificial intelligence, algorithmic pricing, and collusion. *American Economic Review*, 110(10):3267–3297.

[Calvano et al., 2023] Calvano, E., Calzolari, G., Denicoló, V., and Pastorello, S. (2023). Algorithmic collusion: Genuine or spurious? *International Journal of Industrial Organization*, 90:102973.

[Cao et al., 2022a] Cao, J., Xu, Y., and Zhang, C. (2022a). Clans and calamity: How social capital saved lives during China's Great Famine. *Journal of Development Economics*, 157:102865.

[Cao et al., 2022b] Cao, Y., Kleywegt, A. J., and Wang, H. (2022b). Network revenue management under a spiked multinomial logit choice model. *Operations Research*, 70(4):2237–2253.

[Casella and Berger, 2021] Casella, G. and Berger, R. L. (2021). *Statistical Inference*. Cengage Learning.

[Chen et al., 2023a] Chen, J., Hu, Q., Shi, D., and Zhang, F. (2023a). Quick response under strategic manufacturer. *Manufacturing & Service Operations Management*.

[Chen et al., 2021] Chen, M., Elmachtoub, A. N., and Lei, X. (2021). Matchmaking strategies for maximizing player engagement in video games. *Available at SSRN 3928966*.

[Chen and Gallego, 2019] Chen, N. and Gallego, G. (2019). Welfare analysis of dynamic pricing. *Management Science*, 65(1):139–151.

[Chen and Tsai, 2023] Chen, N. and Tsai, H.-T. (2023). Price competition under information (dis) advantage. *Available at SSRN 4420175*.

[Chen et al., 2023b] Chen, W., Wei, Z., and Xie, K. (2023b). Regulating professional players in peer-to-peer markets: Evidence from Airbnb. *Management Science*, 69(5):2893–2918.

[Chen et al., 2017] Chen, Z., Xue, S., Kolen, J., Aghdaie, N., Zaman, K. A., Sun, Y., and Seif El-Nasr, M. (2017). EOMM: An Engagement Optimized Matchmaking Framework. In *Proceedings of the 26th International Conference on World Wide Web*, pages 1143–1150, Perth Australia. International World Wide Web Conferences Steering Committee.

[Cheng et al., 2022] Cheng, H. K., Jung, K. S., Kwark, Y., and Pu, J. (2022). Impact of own brand product introduction on optimal pricing models for platform and incumbent sellers. *Information Systems Research*.

[Cheng et al., 2021] Cheng, J., Dai, Y., Lin, S., and Ye, H. (2021). Clan culture and family ownership concentration: Evidence from China. *China Economic Review*, 70:101692.

[Choi et al., 2016] Choi, T.-M., Cheng, TCE., and Zhao, X. (2016). Multi-methodological research in operations management. *Production and Operations Management*, 25(3):379–389.

[Clark et al., 2023] Clark, R., Assad, S., Ershov, D., and Xu, L. (2023). Algorithmic pricing and competition: Empirical evidence from the german retail gasoline market. *Journal of Political Economy*.

[Dai and Singh, 2020] Dai, T. and Singh, S. (2020). Conspicuous by its absence: Diagnostic expert testing under uncertainty. *Marketing Science*, 39(3):540–563.

[Deisenroth et al., 2020] Deisenroth, M. P., Faisal, A. A., and Ong, C. S. (2020). *Mathematics for Machine Learning*. Cambridge University Press.

[Den Boer, 2015] Den Boer, A. V. (2015). Dynamic pricing and learning: Historical origins, current research, and new directions. *Surveys in operations research and management science*, 20(1):1–18.

[den Boer, 2023] den Boer, A. V. (2023). Algorithmic collusion: A mathematical definition and research agenda for the OR/MS community. *Available at SSRN 4636488*.

[den Boer et al., 2022] den Boer, A. V., Meylahn, J. M., and Schinkel, M. P. (2022). Artificial collusion: Examining supracompetitive pricing by Q-learning algorithms. *Amsterdam Law School Research Paper*, (2022-25).

[Deng et al., 2021] Deng, Q., Fang, X., and Lim, Y. F. (2021). Urban consolidation center or peer-to-peer platform? The solution to urban last-mile delivery. *Production and Operations Management*, 30(4):997–1013.

[Deng et al., 2023] Deng, Y., Tang, C. S., Wang, W., and Yoo, O. S. (2023). Can third-party sellers benefit from a platform's entry to the market? *Service Science*.

[Dinner et al., 2014] Dinner, I. M., Heerde Van, H. J., and Neslin, S. A. (2014). Driving online and offline sales: The cross-channel effects of traditional, online display, and paid search advertising. *Journal of marketing research*, 51(5):527–545.

[Dolgopolov, 2024] Dolgopolov, A. (2024). Reinforcement learning in a prisoner's dilemma. *Games and Economic Behavior*, 144:84–103.

[Dolnicar, 2021] Dolnicar, S. (2021). *Airbnb Before, During and After COVID-19*. The University of Queensland.

[Dong et al., 2022a] Dong, L., Rashkova, I., and Shi, D. (2022a). Food safety audits in developing economies: Decentralization vs. centralization. *Manufacturing & Service Operations Management*, 24(6):2863–2881.

[Dong et al., 2022b] Dong, L., Shi, D., and Zhang, F. (2022b). 3D printing and product assortment strategy. *Management Science*, 68(8):5724–5744.

[Donohue et al., 2020] Donohue, K., Özer, Ö., and Zheng, Y. (2020). Behavioral Operations: Past, Present, and Future. *Manufacturing & Service Operations Management*, 22(1):191–202.

[Epivent and Lambin, 2024] Epivent, A. and Lambin, X. (2024). On algorithmic collusion and reward-punishment schemes. *Economics Letters*, page 111661.

[Farronato and Fradkin, 2022] Farronato, C. and Fradkin, A. (2022). The welfare effects of peer entry: The case of Airbnb and the accommodation industry. *American Economic Review*, 112(6):1782–1817.

[Fisher and Raman, 2022] Fisher, M. and Raman, A. (2022). Innovations in retail operations: Thirty years of lessons from Production and Operations Management. *Production and Operations Management*, 31(12):4452–4461.

[Foerderer et al., 2018] Foerderer, J., Kude, T., Mithas, S., and Heinzl, A. (2018). Does platform owner's entry crowd out innovation? Evidence from Google photos. *Information Systems Research*, 29(2):444–460.

[Folger, 2023] Folger, J. (2023). How Airbnb Works—for Hosts, Guests, and the Company Itself.

[Frostig and Weiss, 2016] Frostig, E. and Weiss, G. (2016). Four proofs of Gittins' multiarmed bandit theorem. *Annals of Operations Research*, 241(1):127–165.

[Fudenberg and Tirole, 1991] Fudenberg, D. and Tirole, J. (1991). *Game Theory*. MIT press.

[Gallego and Hu, 2014] Gallego, G. and Hu, M. (2014). Dynamic pricing of perishable assets under competition. *Management Science*, 60(5):1241–1259.

[Gallego and Topaloglu, 2019] Gallego, G. and Topaloglu, H. (2019). *Revenue Management and Pricing Analytics*, volume 279 of *International Series in Operations Research & Management Science*. Springer New York, New York, NY.

[Gao et al., 2021] Gao, P., Ma, Y., Chen, N., Gallego, G., Li, A., Rusmevichientong, P., and Topaloglu, H. (2021). Assortment optimization and pricing under the multinomial logit model with impatient customers: Sequential recommendation and selection. *Operations research*, 69(5):1509–1532.

[Gerpott and Berends, 2022] Gerpott, T. J. and Berends, J. (2022). Competitive pricing on online markets: A literature review. *Journal of Revenue and Pricing Management*, 21(6):596–622.

[Griliches, 1977] Griliches, Z. (1977). Estimating the returns to schooling: Some econometric problems. *Econometrica: Journal of the Econometric Society*, pages 1–22.

[Guttentag, 2019] Guttentag, D. (2019). Progress on airbnb: A literature review. *Journal of Hospitality and Tourism Technology*, 10(4):814–844.

[Haarnoja et al., 2018] Haarnoja, T., Zhou, A., Hartikainen, K., Tucker, G., Ha, S., Tan, J., Kumar, V., Zhu, H., Gupta, A., Abbeel, P., et al. (2018). Soft actor-critic algorithms and applications. *arXiv preprint arXiv:1812.05905*.

[Hagiu et al., 2020] Hagiu, A., Jullien, B., and Wright, J. (2020). Creating platforms by hosting rivals. *Management Science*, 66(7):3234–3248.

[Haile, 2020] Haile, P. (2020). Structural vs. reduced form: Language, confusion, and models in empirical economics.

[Hansen et al., 2021] Hansen, K. T., Misra, K., and Pai, M. M. (2021). Frontiers: Algorithmic collusion: Supracompetitive prices via independent algorithms. *Marketing Science*, 40(1):1–12.

[He et al., 2020] He, S., Peng, J., Li, J., and Xu, L. (2020). Impact of platform owner's entry on third-party stores. *Information Systems Research*, 31(4):1467–1484.

[Hettich, 2021] Hettich, M. (2021). Algorithmic collusion: Insights from deep learning. *Available at SSRN 3785966*.

[Iyer and Yoganarasimhan, 2021] Iyer, G. and Yoganarasimhan, H. (2021). Strategic Polarization in Group Interactions. *Journal of Marketing Research*, 58(4):782–800.

[Jackson, 2014] Jackson, M. O. (2014). Mechanism theory. *Available at SSRN 2542983*.

[Jiang et al., 2023] Jiang, B., Wang, Z., Xue, C., and Zhang, N. (2023). Assortment optimization in the presence of focal effect: Operational insights and efficient algorithms. *Available at SSRN*.

[Kakade and Langford, 2002] Kakade, S. and Langford, J. (2002). Approximately optimal approximate reinforcement learning. In *Proceedings of the Nineteenth International Conference on Machine Learning*, pages 267–274.

[Klein et al., 2020] Klein, R., Koch, S., Steinhardt, C., and Strauss, A. K. (2020). A review of revenue management: Recent generalizations and advances in industry applications. *European Journal of Operational Research*, 284(2):397–412.

[Klein, 2021] Klein, T. (2021). Autonomous algorithmic collusion: Q-learning under sequential pricing. *The RAND Journal of Economics*, 52(3):538–558.

[Kocabiyikoglu et al., 2015] Kocabiyikoglu, A., Gogus, C. I., and Gonul, M. S. (2015). Revenue Management vs. Newsvendor Decisions: Does Behavioral Response Mirror Normative Equivalence? *Production and Operations Management*, 24(5):750–761.

[Kouvelis and Shi, 2020] Kouvelis, P. and Shi, D. (2020). Who should compensate the sales agent in a distribution channel? *Production and Operations Management*, 29(11):2437–2460.

[Kovach and Tserenjigmid, 2022] Kovach, M. and Tserenjigmid, G. (2022). The focal luce model. *American Economic Journal: Microeconomics*, 14(3):378–413.

[Kumar and Tang, 2022] Kumar, S. and Tang, C. S. (2022). Expanding the boundaries of the discipline: The 30th-anniversary issue of Production and Operations Management. *Production and Operations Management*, 31(12):4257–4261.

[Lai et al., 2022] Lai, G., Liu, H., Xiao, W., and Zhao, X. (2022). "Fulfilled by amazon": A strategic perspective of competition at the e-commerce platform. *Manufacturing & Service Operations Management*, 24(3):1406–1420.

[Leeflang et al., 2002] Leeflang, P. S., Van Heerde, H. J., and Wittink, D. R. (2002). How Promotions Work: SCAN*PRO-Based Evolutionary Model Building. *SSRN Electronic Journal*.

[Lei, 2022] Lei, X. (2022). *Revenue Management in Video Games and with Fairness*. PhD thesis, Columbia University.

[Liu et al., 2023] Liu, Y., Chen, X. A., Liu, Y., and Wang, Z. (2023). Simultaneous vs sequential: Optimal assortment recommendation in multi-channel retailing. *Available at SSRN 4592172*.

[Loots and den Boer, 2023] Loots, T. and den Boer, A. V. (2023). Data-driven collusion and competition in a pricing duopoly with multinomial logit demand. *Production and Operations Management*, 32(4):1169–1186.

[Luenberger et al., 1984] Luenberger, D. G., Ye, Y., et al. (1984). *Linear and Nonlinear Programming*, volume 2. Springer.

[Martello, 2022] Martello, S. (2022). *Autonomous Pricing Using Policy-Gradient Reinforcement Learning*. PhD thesis, Alma Mater Studiorum˙ University of Bologna.

[Meylahn and V. den Boer, 2022] Meylahn, J. M. and V. den Boer, A. (2022). Learning to collude in a pricing duopoly. *Manufacturing & Service Operations Management*, 24(5):2577–2594.

[Mincer, 1974] Mincer, J. (1974). Schooling, experience, and earnings. Human behavior & social institutions no. 2.

[Misra et al., 2019] Misra, K., Schwartz, E. M., and Abernethy, J. (2019). Dynamic online pricing with incomplete information using multiarmed bandit experiments. *Marketing Science*, 38(2):226–252.

[Mithas et al., 2022] Mithas, S., Chen, Y., Lin, Y., and De Oliveira Silveira, A. (2022). On the causality and plausibility of treatment effects in operations management research. *Production and Operations Management*, 31(12):4558–4571.

[Mnih et al., 2015] Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., et al. (2015). Human-level control through deep reinforcement learning. *nature*, 518(7540):529–533.

[Musolff, 2022] Musolff, L. (2022). Algorithmic pricing facilitates tacit collusion: Evidence from e-commerce. In *Proceedings of the 23rd ACM Conference on Economics and Computation*, pages 32–33.

[Narayan et al., 2011] Narayan, V., Rao, V. R., and Saunders, C. (2011). How Peer Influence Affects Attribute Preferences: A Bayesian Updating Mechanism. *Marketing Science*, 30(2):368–384.

[Nip et al., 2021] Nip, K., Wang, C., and Wang, Z. (2021). Competitive and Cooperative Assortment Games under Markov Chain Choice Model. *Production and Operations Management*.

[Özer and Phillips, 2012] Özer, Ö. and Phillips, R. (2012). *The Oxford Handbook of Pricing Management*. OUP Oxford.

[Peng et al., 2024] Peng, Z., Rong, Y., and Zhu, T. (2024). Transformer-based choice model: A tool for assortment optimization evaluation. *Naval Research Logistics (NRL)*.

[Peters et al., 2017] Peters, J., Janzing, D., and Schölkopf, B. (2017). *Elements of Causal Inference: Foundations and Learning Algorithms*. The MIT Press.

[Possnig, 2023] Possnig, C. (2023). *Reinforcement Learning and Collusion*. Department of Economics, University of Waterloo.

[Rao and Steckel, 1991] Rao, V. R. and Steckel, J. H. (1991). A polarization model for describing group preferences. *Journal of Consumer Research*, 18(1):108–118.

[Rietveld and Schilling, 2021] Rietveld, J. and Schilling, M. A. (2021). Platform competition: A systematic and interdisciplinary review of the literature. *Journal of Management*, 47(6):1528–1563.

[Rooderkerk et al., 2011] Rooderkerk, R. P., Van Heerde, H. J., and Bijmolt, T. H. (2011). Incorporating context effects into a choice model. *Journal of Marketing Research*, 48(4):767–780.

[Ross, 2014] Ross, S. M. (2014). *Introduction to Probability Models*. Elsevier, Amsterdam ; Boston, eleventh edition edition.

[Roth and Singhal, 2022] Roth, A. M. and Singhal, V. R. (2022). Pioneering role of the Production and Operations Management in promoting empirical research in operations management. *Production and Operations Management*, 31(12):4529–4543.

[Roth, 2007] Roth, A. V. (2007). Applications of empirical science in manufacturing and service operations. *Manufacturing & Service Operations Management*, 9(4):353–367.

[Rubinstein, 1982] Rubinstein, A. (1982). Perfect Equilibrium in a Bargaining Model. *Econometrica*, 50(1):97.

[Schaul et al., 2015] Schaul, T., Quan, J., Antonoglou, I., and Silver, D. (2015). Prioritized experience replay. *arXiv preprint arXiv:1511.05952*.

[Schulman et al., 2015] Schulman, J., Levine, S., Abbeel, P., Jordan, M., and Moritz, P. (2015). Trust region policy optimization. In *International Conference on Machine Learning*, pages 1889–1897. PMLR.

[Schulman et al., 2017] Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. (2017). Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.

[Shi et al., 2023] Shi, R., Aaltonen, A., Henfridsson, O., and Gopal, R. D. (2023). Comparing platform owners' early and late entry into complementary markets. *MIS Quarterly*.

[Spence, 1973] Spence, M. (1973). Job Market Signaling. *The Quarterly Journal of Economics*, 87(3):355.

[Spirtes, 2010] Spirtes, P. (2010). Introduction to causal inference. *Journal of Machine Learning Research*, 11(5).

[Strauss et al., 2018] Strauss, A. K., Klein, R., and Steinhardt, C. (2018). A review of choice-based revenue management: Theory and methods. *European Journal of Operational Research*, 271(2):375–387.

[Thrun and Littman, 2000] Thrun, S. and Littman, M. L. (2000). Reinforcement learning: An introduction. *AI Magazine*, 21(1):103–103.

[Van Hasselt et al., 2016] Van Hasselt, H., Guez, A., and Silver, D. (2016). Deep reinforcement learning with double q-learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 30.

[Van Heerde et al., 2004] Van Heerde, H. J., Leeflang, P. S. H., and Wittink, D. R. (2004). Decomposing the Sales Promotion Bump with Store Data. *Marketing Science*, 23(3):317–334.

[Vaswani et al., 2023] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. (2023). Attention Is All You Need.

[Von Neumann and Morgenstern, 1994] Von Neumann, J. and Morgenstern, O. (1994). *Theory of Games and Economic Behavior*. Princeton university press.

[Wang et al., 2023] Wang, J., Huang, T., and Lee, J. (2023). Cross-licensing in a Supply Chain with Asymmetric Manufacturers.

[Wang, 2018] Wang, R. (2018). When prospect theory meets consumer choice models: Assortment and pricing management with reference prices. *Manufacturing & Service Operations Management*, 20(3):583–600.

[Wang and Wang, 2017] Wang, R. and Wang, Z. (2017). Consumer Choice Models with Endogenous Network Effects. *Management Science*, 63(11):3944–3960.

[Wang et al., 2016] Wang, Z., Schaul, T., Hessel, M., Hasselt, H., Lanctot, M., and Freitas, N. (2016). Dueling network architectures for deep reinforcement learning. In *International Conference on Machine Learning*, pages 1995–2003. PMLR.

[Wang and Ye, 2016] Wang, Z. and Ye, Y. (2016). Hidden-city ticketing: The cause and impact. *Transportation Science*, 50(1):288–305.

[Wen and Zhu, 2019] Wen, W. and Zhu, F. (2019). Threat of platform-owner entry and complementor responses: Evidence from the mobile app market. *Strategic Management Journal*, 40(9):1336–1367.

[Wittink et al., 1988] Wittink, D. R., Addona, M. J., Hawkes, W. J., and Porter, J. C. (1988). SCAN* PRO: The estimation, validation and use of promotional effects based on scanner data. *Internal Paper, Cornell University*.

[Yang, 2019] Yang, H. (2019). Family clans and public goods: Evidence from the New Village beautification project in South Korea. *Journal of Development Economics*, 136:34–50.

[Yousefi Maragheh et al., 2020] Yousefi Maragheh, R., Chen, X., Davis, J., Cho, J., Kumar, S., and Achan, K. (2020). Choice modeling and assortment optimization in the presence of context effects. *Available at SSRN 3747354*.

[Zhang, 2019] Zhang, C. (2019). Family support or social support? The role of clan culture. *Journal of Population Economics*, 32:529–549.

[Zhang, 2020] Zhang, C. (2020). Clans, entrepreneurship, and development of the private sector in China. *Journal of Comparative Economics*, 48(1):100–123.

[Zhang et al., 2017] Zhang, C., Phang, C. W., Wu, Q., and Luo, X. (2017). Nonlinear Effects of Social Connections and Interactions on Individual Goal Attainment and Spending: Evidences from Online Gaming Markets. *Journal of Marketing*, 81(6):132–155.

[Zhang et al., 2022] Zhang, S., Lee, D., Singh, P. V., and Srinivasan, K. (2022). What makes a good image? Airbnb demand analytics leveraging interpretable image features. *Management Science*, 68(8):5644–5666.

[Zhao et al., 2023] Zhao, P., Ma, Z., Gill, T., and Ranaweera, C. (2023). Social media sentiment polarization and its impact on product adoption. *Marketing Letters*.

[Zhu and Liu, 2018] Zhu, F. and Liu, Q. (2018). Competing with complementors: An empirical look at Amazon. com. *Strategic management journal*, 39(10):2618–2642.

[Zhu et al., 2023] Zhu, F., Liu, S., Wang, R., and Wang, Z. (2023). Assign-to-Seat: Dynamic Capacity Control for Selling High-Speed Train Tickets. *Manufacturing & Service Operations Management*, 25(3):921–938.