Machine Learning

Elvis Ren

May 20, 2020

Contents

Part I

Mathematical Background

Chapter 1

Set Theory

1.1 Axioms

Axiom 1 (Axiom of Extensionality)**.** *If X and Y have the same elements, then X = Y.*

$$\forall u(u \in X \leftrightarrow u \in Y) \to X = Y \tag{1.1}$$

Axiom 2 (Axiom of Pairing)**.** *For any a and b there exists a set $\{a,b\}$ that contains exactly a and b.*

$$\forall a \forall b \exists c \forall x(x \in c \leftrightarrow x = a \lor x = b) \tag{1.2}$$

A singleton $\{a\}$ is the set $\{a\} = \{a,a\}$. An ordered pair (a,b) is the set $(a,b) = \{\{a\},\{a,b\}\}$.

Axiom 3 (Axiom Schema of Seperation)**.** *If P is a property with parameter p, then for any X and p there exists a set $Y = \{u \in X : P(u,p)\}$ that contains all those $u \in X$ that have property P.*

$$\forall X \forall p \exists Y \forall u \big(u \in Y \leftrightarrow u \in X \land \varphi(u,p)\big) \tag{1.3}$$

If define class $C = \varphi(u,p)$, then $\forall X \exists Y(C \land X) = Y$, so a subclass of a set is a set. The empty class $\emptyset = \{u : u \neq u\}$ is a empty set.

Axiom 4 (Axiom of Union)**.** *For any X there exists a set $Y = \cup X$, the union of all elements of X.*

$$\forall X \exists Y \forall u \big(u \in Y \leftrightarrow \exists z(z \in X \land u \in z)\big) \tag{1.4}$$

Axiom 5 (Axiom of Power Set)**.** *For any X there exists a set $Y = P(X)$, the set of all subset of X.*

$$\forall X \exists Y \forall u(u \in Y \leftrightarrow u \subset X) \tag{1.5}$$

Axiom 6 (Axiom of Infinity)**.** *There exists an infinite set.*

$$\exists S \big(\emptyset \in S \land (\forall x \in S) x \cup \{x\} \in S\big) \tag{1.6}$$

A set S with above property is called inductive.

Axiom 7 (Axiom Schema of Replacement)**.** *If a class F is a function, then for any X there exists a set $Y = F(X) = \{F(x) : x \in X\}$.*

$$\forall x \forall y \forall z \big(\varphi(x,y,p) \land \varphi(x,z,p) \to y = z\big) \to \forall X \exists Y \forall y \big(y \in Y \to (\exists x \in X)\varphi(x,y,p)\big) \tag{1.7}$$

So if a class F is a function and $\mathrm{dom}(f)$ is a set, then $\mathrm{ran}(f)$ is a set.

Axiom 8 (Axiom of Regularity)**.** *Every nonempty set has an $\in$-minimal element.*

Axiom 9 (Axiom of Choice)**.** *Every family of nonempty set has a choice function.*

The Theorem 1 to Theorem 8 is the Zermelo-Fraenkel axiomatic set theory ZF. ZFC denote the ZF + AC, the axiom of choice.

Theorem 1 (Russell's Paradox)**.** *There is no set whose elements are all those sets that are not member of themselves: $S = \{X : X \notin X\}$. So the set of all set does not exist.*

Definition 1. *A binary relation f is a function if $(x,y) \in f$ and $(x,z) \in f$ implies $y = z$. For a function f from X to Y $f : X \to Y$, if $Y = \mathrm{ran}(f)$, f is onto. If $f(x) = f(y)$ implies $x = y$, f is one-to-one. The inverse image $f_{-1}(Y) = \{x : f(x) \in Y\}$. If f is one-to-one, then the inverse is $f^{-1}(y) = x$.*

Definition 2. *The restriction of a function f to a set X is:*

$$f \restriction_X = \{(x,y) \in f : x \in X\} \tag{1.8}$$

Definition 3 (class)**.** *if $\varphi(x, p_1, \dots, p_n)$ is a formula, then $C = \{x : \varphi(x, p_1, \dots, p_n)\}$ is a class. So a formula defines a class.*

Definition 4 (universe)**.** *The universe is the class of all sets: $V = \{x : x = x\}$.*

Definition 5. *A class that is not a set is a proper class.*

1.2 Ordinal Numbers

1.2.1 Well Ordering

Definition 6. *A binary relation $<$ is a partial ordering of P if :*

1. $\forall p \in \mathrm{P}(p \not< p)$.
2. $p < q \wedge q < r \rightarrow p < r$.

Definition 7. *A partial order $(\mathrm{P}, <)$ is linear ordering if $\forall p \forall q (p < q \wedge p = q \wedge q < p)$.*

Definition 8. *α is the supremum of X if α is the least upper bound of X: $\alpha = \sup\{\mathrm{X}\}$.*

Definition 9. *α is the infimum of X if α is the greatest lower bound of X: $\alpha = \inf\{\mathrm{X}\}$.*

Definition 10. *If $(\mathrm{P}, <)$ and $(\mathrm{Q}, <)$ are partially ordered sets and $f : \mathrm{P} \rightarrow \mathrm{Q}$, then f is order-preserving if $x < y \rightarrow f(x) < f(y)$. If P and Q are linearly ordered, f is called increasing.*

Definition 11. *$f : \mathrm{P} \rightarrow \mathrm{Q}$ is isomorphism of P and Q if f and f^{-1} are order-preserving. An isomorphism of P onto itself is automorphism.*

Definition 12. *A linear ordering $<$ is well-ordering if every nonempty subset of P has a least element.*

Theorem 2. *If $(\mathrm{W}, <)$ is a well-ordered set and $f : \mathrm{W} \rightarrow \mathrm{W}$ is an increasing function, then $\forall x \in \mathrm{W}\big(f(x) \geq x\big)$.*

Proof. If the set $\mathrm{X} = \big\{x \in \mathrm{W} : f(x) < x\big\}$ is nonempty, let z be its least element and $w = f(z)$. Then $f(w) = ff(z) < f(z) = w$. So $f(w) < w \rightarrow w \in \mathrm{X} \wedge w < z$. □

Theorem 3. *The only automorphism of a well-ordered set is the identity.*

Proof. $f(x) \geq x$ and $f^{-1} \geq x$. □

Theorem 4. *If two well-ordered set W_1 and W_2 are isomorphic, then the isomorphism is unique.*

Proof. construct a automorphism using two isomorphism. □

Definition 13. *Let $(\mathrm{W}, <)$ be an well-ordered set. $\alpha \in \mathrm{W}$, the initial segment W_α of W is defined as*

$$\mathrm{W}_\alpha = \{x \in \mathrm{W} : x < \alpha\} \tag{1.9}$$

Theorem 5. *no well-ordered set is isomorphic to an initial segment of itself.*

Proof. If $\mathrm{ran}(f) = \{x : x < u\}$ is an initial segment, then $f(u) < u$, contrary to Theorem 2. □

Theorem 6. *If W and V are well-ordered sets, then one of the following holds:*

1. *W is isomorphic to V.*
2. *W is isomorphic to an initial segment of V.*
3. *an initial segment of W is isomorphic to V.*

Proof. Define a set $f = \Big\{(x, y) \in \mathrm{W} \times \mathrm{V} : \mathrm{W}_x \text{ is isomorphic to } \mathrm{V}_y\Big\}$. Check the $\mathrm{dom}(f)$ and $\mathrm{ran}(f)$. □

1.2.2 Ordinal Numbers

Definition 14. *A set T is transitive if every element of T is a subset of T:*

$$a \in \mathrm{T} \rightarrow a \subset \mathrm{T} \tag{1.10}$$

or $\cup \mathrm{T} \subset \mathrm{T}$.

Definition 15. *A set is an ordinal number if it is transitive and well-ordered by $\in$. The class of all ordinals is Ord.*

Definition 16. *For two sets α and β, define a relation $<$ as $\alpha < \beta \longleftrightarrow \alpha \in \beta$.*

Theorem 7. $\emptyset \in \mathrm{Ord}$

Proof. by definition. □

Theorem 8. $\alpha \in \mathrm{Ord} \wedge \beta \in \alpha \rightarrow \beta \in \mathrm{Ord}$

Proof. $\forall x \in \beta, x \in \beta \wedge \beta \subset \alpha \to x \in \alpha \to x \subset \alpha \to x \subset \beta$. □

Theorem 9. $\alpha \in \mathrm{Ord} \wedge \beta \in \mathrm{Ord} \wedge \alpha \neq \beta \wedge \alpha \subset \beta \to \alpha \in \beta$

Proof. Let γ be the least element of $\beta - \alpha$. Since α is transitive, α is an initial segment of β_γ. So $\alpha = \{\epsilon \in \beta : \epsilon < \gamma\} = \gamma$, so $\alpha \in \beta$. □

Theorem 10. $\forall \alpha, \beta \in \mathrm{Ord} \to \alpha \subset \beta \vee \beta \subset \alpha$

Proof. Let $\gamma = \alpha \cap \beta$. γ is an ordinal. So $\gamma \subset \alpha \to \gamma \in \alpha$, and $\gamma \in \beta$, so $\gamma \in \alpha \cap \beta = \gamma$ and $\gamma \in \gamma$. □

Theorem 11. *The facts about ordinal numbers are:*

1. $\alpha = \{\beta : \beta < \alpha\}$
2. *If* C *is a nonempty class of ordinals, then* $\cap \mathrm{C}$ *and* $\cup \mathrm{C}$ *are ordinals.*
3. $\forall \alpha \in \mathrm{Ord}(\alpha \cup \{\alpha\} \in \mathrm{Ord})$ *and* $\alpha \cup \{\alpha\} = \inf\{\beta : \beta > \alpha\}$.

Definition 17. *We define* $\alpha + 1 = \alpha \cup \{\alpha\}$*, the successor of* α.

Theorem 12. *Every well-ordered set is isomorphic to a unique ordinal number.*

Definition 18. *If* $\alpha = \beta + 1$*,* α *is a successor ordinal. If* α *is not a successor ordinal, then* $\alpha = \sup\{\beta : \beta < \alpha\} = \cup\alpha$*, and is a limit ordinal.* 0 *is defined as a limit ordinal.*

Definition 19 (natural numbers)**.** *The least nonzero limit ordinal is denoted as* ω*. The ordinals less than* ω *is called finite ordinals, or natural numbers.*

Theorem 13 (Transfinite Induction)**.** *Let* C *be a class of ordinals and assume that:*

1. $0 \in \mathrm{C}$
2. $\alpha \in \mathrm{C} \to \alpha + 1 \in \mathrm{C}$
3. *If* α *is a nonzero limit ordinal and* $\forall \beta \in \alpha(\beta \in \mathrm{C}) \to \alpha \in \mathrm{C}$.

Then $\mathrm{C} = \mathrm{Ord}$.

Proof. choose the least $\alpha \notin \mathrm{C}$. □

Definition 20. *A transfinite sequence is a function that the domain is an ordinal:*

$$\langle \alpha_\xi : \xi < \alpha \rangle \tag{1.11}$$

Theorem 14 (Transfinite Recursion)**.** *Let* G *be a function on the class of transfinite sequence, then there is a unique function* F *on* Ord *that* $\forall \alpha \in \mathrm{Ord}$*:*

$$\mathrm{F}(\alpha) = \mathrm{G}(\mathrm{F} \restriction_\alpha) \tag{1.12}$$

Definition 21. *Let* $\alpha > 0$ *be a limit ordinal and* $\langle \gamma_\xi : \xi < \alpha \rangle$ *be a nondecreasing sequence of ordinals. The limit of the sequence is*

$$\lim_{\xi \to \alpha} \gamma_\xi = \sup\{\gamma_\xi : \xi < \alpha\} \tag{1.13}$$

It is possible that $\lim\limits_{\xi \to \alpha} \gamma_\xi \notin \langle \gamma_\xi : \xi < \alpha \rangle$.

Definition 22. *A sequence of ordinal* $\langle \gamma_\alpha : \alpha \in \mathrm{Ord} \rangle$ *is normal if it is increasing and continuous, that is for every limit ordinal* α*,* $\gamma_\alpha = \lim\limits_{\beta \to \alpha} \gamma_\beta$.

1.2.3 Ordinal Arithmetic

Theorem 15. *For all ordinal* α *and* β*, we have:*

1. $\alpha + 0 = \alpha$
2. $\alpha + (\beta + 1) = (\alpha + \beta) + 1$
3. $\alpha + \beta = \lim\limits_{\xi \to \beta}(\alpha + \xi)$ *for all limit ordinal* $\beta > 0$.
4. $\alpha \cdot 0 = 0$
5. $\alpha \cdot (\beta + 1) = \alpha \cdot \beta + \alpha$
6. $\alpha \cdot \beta = \lim\limits_{\xi \to \beta} \alpha \cdot \beta$ *for all limit ordinal* $\beta > 0$
7. $\alpha^0 = 1$
8. $\alpha^{\beta+1} = \alpha^\beta \cdot \alpha$
9. $\alpha^\beta = \lim\limits_{\xi \to \beta} \alpha^\xi$ *for all limit ordinal* $\beta > 0$.

So $\alpha+\beta$, $\alpha\cdot\beta$, and α^β are normal function in second variable β. Note that neither $+$ nor $\cdot$ is commutative:

$$\begin{aligned} 1+\omega=\omega\neq\omega+1 \\ 2\cdot\omega=\omega\neq\omega\cdot 2=\omega+\omega \end{aligned} \tag{1.14}$$

Theorem 16. *For all ordinal α and β, we have:*

1. $\beta<\gamma\to\alpha+\beta<\alpha+\gamma$
2. *If $\alpha<\beta$, there is a unique δ that $\alpha+\delta=\beta$.*
3. $\beta<\gamma\wedge\alpha>0\to\alpha\cdot\beta<\alpha\cdot\gamma$
4. *If $\alpha>0$, there is a unique β and $\rho<\alpha$ that $\gamma=\alpha\cdot\beta+\rho$.*
5. $\beta<\gamma\wedge\alpha>1\to\alpha^\beta<\alpha^\gamma$

Theorem 17 (Cantor's Normal Form Theorem)**.** *Every ordinal $\alpha>0$ has a unique representation:*

$$\alpha=\omega^{\beta_1}\cdot k_1+\cdots+\omega^{\beta_n}\cdot k_n \tag{1.15}$$

where $n\geq 1$, $\alpha\geq\beta_1>\cdots>\beta_n$, and k_i are nonzero natural numbers.

Proof. use induction. $\forall\alpha>0$, let β be the greatest ordinal number that $\omega^\beta\leq\alpha$. There is a unique δ and $\rho<\omega^\beta$ that $\alpha=\omega^\beta+\rho$. □

1.2.4 Well-Founded Relations

Definition 23. *A binary relation E on a set P is well-founded if every nonempty $\mathrm{X}\subset\mathrm{P}$ has a E-minimal element, that is $\forall a\in\mathrm{X}$ there is no $x\in\mathrm{X}$ that $x\mathrm{E}a$.*

Theorem 18. *If E is a well-founded relation on P, there is a unique function $\rho:\mathrm{P}\to\mathrm{Ord}$ that $\forall x\in\mathrm{P}$:*

$$\rho(x)=\sup\{\rho(y)+1:y\mathrm{E}x\} \tag{1.16}$$

The range of ρ is an initial segment of ordinals and is an ordinal number, which is the height of E.

Proof. Define a P that

$$\begin{aligned} \mathrm{P}_0&=\emptyset \\ \mathrm{P}_{\alpha+1}&=\{x\in\mathrm{P}:\forall y(y\mathrm{E}x\to y\in\mathrm{P}_\alpha)\} \\ \mathrm{P}_\alpha&=\bigcup_{\xi<\alpha}\mathrm{P}_\xi\text{ , if }\alpha\text{ is a limit ordinal} \end{aligned} \tag{1.17}$$

Let θ be the least ordinal that $\mathrm{P}_{\theta+1}=\mathrm{P}_\theta$. □

1.3 Cardinal Numbers

Chapter 2

Linear Algebra

2.1 Vector Space

2.1.1 Field

Definition 24. *For 0 and 1 of a field F, the smallest n that $\sum_{i=1}^{n} 1 = 0$ is called the characteristic of F. If no such n exists, F is called characteristic zero.* □

Definition 25. *The field Z_2 has characteristic of 2 which consists of two elements 0 and 1:*

- $0 + 0 = 0$
- $0 + 1 = 1 + 0 = 1$
- $1 + 1 = 0$
- $0 \times 0 = 0$
- $0 \times 1 = 1 \times 0 = 0$
- $1 \times 1 = 1$

2.1.2 Vector

Algebra is concerned with how to manipulate symbolic combinations of object and how to equate one with another.

Definition 26. *A vector space vector space V over a field field F has two operation $\{+, \times\}$ with $\vec{0}$ and 1.* □

Definition 27. *A subspace is a subset W of vector space V that is closed under $\{+, \times\}$. When we say a subset is a subspace of a vector space, we mean it is a vector space as well.*

Theorem 19. *$\{0\}$ is a subspace of all vector space.* □

matrix is late Latin for *womb*. The idea is that a matri is a place for holding numbers.

Definition 28. *a trace of an $n \times n$ matrix M, denoted $tr(M)$, is the sum of diagonal entries:*

$$tr(M) = \sum_{i=1}^{n} M_{ii} \tag{2.1}$$

Definition 29. *A span of a nonempty subset S of a vector space V is the set consisting of all linear combinations of the vectors in S. If $span(S) = V$, S generate (or span) V.* □

Definition 30. *The span of $\emptyset$ is $\{0\}$, not $\emptyset$.*

A span set is useful because it allow one to describe all vectors in terms of a much smaller space.

Definition 31. *A subset S of V is linearly dependent if there exist a finite number of distinct vector $u_1, u_2, \ldots, u_n$ in S and scalars $a_1, a_2, \ldots, a_n$, not all 0, that:*

$$\sum_{i=1}^{n} a_i u_i = 0 \tag{2.2}$$

S is called linearly independent if it is not linearly dependent. $\emptyset$ is linearly independent.

□

Theorem 20. *Let S be linearly independent, v is not in S. Then $S \cup v$ is linearly dependent if $v \in span(S)$.*

2.1.3 Basis

Basis tries to represent a infinite vector space using a finite set of vectors. So a complex structure could be understood using simplified structure. A linearly independent generating set has a very useful property that every vector has one and only one representation using basis.

Definition 32. *A basis β for V is a linearly independent subset of V that generate V.* □

A vector space is usually infinite. It is desirable to describe this infinite set using a finite subset, which is the role of basis.

Theorem 21. *$\emptyset$ is a basis for zero vector space $\{0\}$, so every vector space has a basis.*

Definition 33. *The standard basis for F^n is $e_1 = (1, 0, 0, \ldots, 0)$, $e_2 = (0, 1, 0, \ldots, 0)$, $e_n = (0, 0, \ldots, 1)$.*

Definition 34. *The standard basis for $P_n(F)$ is $\{1, x, x^2, \ldots, x^n\}$.*

Theorem 22. *β is a basis of V if $\forall v \in V$, v has a unique representation as a linear combination of vectors of β.*

Theorem 23. *A finite spanning set for V can be reduced to a basis.*

Theorem 24 (Replacement Theorem)**.** *Let V be generated by a set G with n vectors. Let L be a linearly independent subset of V with m vectors. Then $m < n$ and $\exists H \subset G$ with $n - m$ vectors such that $L \cup H$ generate V.* □

Theorem 25. *Let V have a finite basis. Then every basis contains the same number of vectors. This number is an intrinsic property of V and called the dimension of V.*

Theorem 26. *Let V be a vector space with dimension n:*

- *any finite generating set for V contains at least n vectors. If they contains exactly n vectors, they are a basis.*
- *any linearly independent subset of n vectors is a basis.*
- *every linearly independent subset could be extended to a basis.*

Definition 35 (Lagrange Interpolation Formula)**.** *let $c_0, c_1, \ldots, c_n$ be distinct scalars in field F. Define $n+1$ function $\{f_i\}$ as:*

$$f_i(x) = \prod_{k=0, k \neq i}^{n} \frac{x - c_k}{c_i - c_k} \tag{2.3}$$

then $\beta = \{f_i\}$ is a basis of $\mathbb{P}_n(F)$, where $\mathbb{P}_n(F)$ is a set of all polynomials over F. For $\forall g \in \mathbb{P}_n(F)$, we have

$$g = \sum_{i=0}^{n} g(c_i) f_i \tag{2.4}$$

To generate a function g of degree n that passes $n+1$ points (x_i, y_i), first use $\{x_i\}$ to generate $\{f_i\}$, then $g = \sum_{i=0}^{n} y_i f_i$.

Proof. since β is a basis of $\mathbb{P}_n(F)$, $\forall g \in \mathbb{P}_n(F)$,

$$g = \sum_{i=0}^{n} b_i f_i$$

it follows that

$$g(c_j) = \sum_{i=0}^{n} b_i f_i(c_j) = b_j$$

so $g = \sum_{i=0}^{n} g(c_i) f_i$. □

Theorem 27. *for any two subspace W_1 and W_2 of V, their dimension has a relation:*

$$dim(W_1 + W_2) = dim(W_1) + dim(W_2) - dim(W_1 \cap W_2) \tag{2.5}$$

Definition 36. *here are the definition of common terms:*

1. *square matrix: a matrix $M_{i \times j}$ that $i = j$. It is usually denoted as M, not A.*
2. *zero vector: $\vec{0}$.*
3. *transpose: $(A^\top)_{ij} = A_{ji}$.*
4. *symmetric matrix: $A^\top = A$.*
5. *diagonal matrix: for a $n \times n$ square matrix M that $M_{ij} = 0$ if $i \neq j$.*
6. *upper triangular: $A_{ij} = 0$ if $i > j$.*

□

The following text discusses the result of infinite basis.

Definition 37. *Let F be a family of sets. A member M of F is called maximal if M is contained in no member of F other than M itself.*

Definition 38. *A collection of set C is called a chain if for each pair of sets A and B in C, either $A \subseteq B$ or $B \subseteq A$.*

Theorem 28. *Let F be a family of sets. If for each chain $C \subseteq F$, there exists a member of F that contains each member of C, then F contains a maximal member.*

Proof. use axiom of choice. Note that the maximal member may not be in C. □

Definition 39. *Let* S *be a subset of a vector space* V. *A* <u>*maximal linearly independent subset*</u> *of* S *is a subset* B *of* S *that:*

1. B *is linearly independent.*
2. *The only linearly independent subset of* S *that contains* B *is* B.

Theorem 29. *If* V *has a basis* β, β *is maximal linearly independent.*

Proof. A basis is linearly independent. Because a basis generate V, nothing could be added to it and still make it linearly independent. □

Theorem 30. *Let* V *be a vector space and* S *a subset that generate* V. *If* β *is a maximal linearly independent subset of* S, *then* β *is a basis* V.

Proof. β is linearly independent, so only need to prove that β generate V. It is easy because β is maximal in S so nothing from S could be added to it. □

Theorem 31. *Let* S *be a linearly independent subset of a vector space* V. *There exists a maximal linearly independent subset of* V *that contains* S.

Proof. Let F be a family of all linearly independent subsets of V that contains S. For a chain C in F, let U be the union of all its member. This U is linearly independent and belongs to F, so it is a maximal linearly independent subset of F, which is a basis of F. □

Theorem 32. *Every vector space has a basis.*

2.2 Linear Transformation and Matrix

2.2.1 Linear Transformation

Definition 40. *A linear transformation from* V *to* W *is a function* $T : V \to W$ *that:*

1. $T(x+y) = T(x) + T(y)$
2. $T(cx) = cT(x)$

The two linear transformation verification criteria could be combined into one: prove that

$$T(cx+y) = cTx + Ty \tag{2.6}$$

The identity transformation $I_v : V \to V$ is defined as $I_v(x) = x$.
The zero transformation $T_0 : V \to W$ is defined as $T_0 = 0$.

Definition 41. *Let* $T : V \to W$ *be linear. the null space* $\mathcal{N}(T)$ *of* T *is the set* $\left\{x \in V : T(x) = 0\right\}$*. It is also called the kernel of* T*. It measures how much information is lost by the transformation* T.

Definition 42. *The range of* T *is defined as* $\mathcal{R}(T) = \left\{T(x) : x \in V\right\}$*. It measures how much information is retained by the transformation* T.

Theorem 33. *Let* $T : V \to W$ *be linear. If* $\beta = \left\{v_i\right\}$ *is a basis for* V*, then*

$$\mathcal{R}(T) = \operatorname{span}\left(T(\beta)\right) = \operatorname{span}\left(\left\{T(v_i)\right\}\right) \tag{2.7}$$

Definition 43. *Let* $T : V \to W$ *be linear. the nullity of* T *is the dimension of* $\mathcal{N}(T)$*. The rank of* T *is the dimension of* $\mathcal{R}(T)$.

Theorem 34 (Dimension Theorem)**.** *If* V *is finite dimensional,* $T : V \to W$ *is linear, then*

$$\dim\left(\mathcal{N}(T)\right) + \dim\left(\mathcal{R}(T)\right) = \dim\left(T\right) \tag{2.8}$$

Proof. expand nullity set to a basis and prove the image of extra parameters are independent. □

Theorem 35. *Let* $V : \left\{v_i\right\}$ *and* $W : \left\{w_i\right\}$ *be vector space over* F*, and their dimensions are the same. Then there exists a unique linear transformation* $T : V \to W$ *such that* $T(v_i) = w_i$.

Proof. For $x = \sum_{i=1}^{n} a_i v_i$, define $T : V \to W$ that $T(x) = \sum_{i=1}^{n} a_i w_i$. □

Theorem 35 is useful when proving two functions are the same.

Theorem 36. *Let* $T : V \to W$ *be a linear transformation.* T *is one-to-one if and only if* $\mathcal{N}(T) = \{0\}$.

2.2.2 Matrix Representation

Definition 44. *A ordered basis for* V *is a basis for* V *with a specific order.*

Definition 45. $\left\{e_1, e_2, \ldots, e_n\right\}$ *is the standard ordered basis for* F^n. $\{1, x, \ldots, x^n\}$ *is the standard ordered basis for* $P_n(F)$.

Definition 46. *Let* $\beta = \left\{u_1, u_2, \ldots, u_n\right\}$ *be an ordered basis for* V. $\forall x \in V$*, let* $\left\{a_1, a_2, \ldots, a_n\right\}$ *be the unique scalar such that*

$$x = \sum_{i=1}^{n} a_i u_i$$

the coordinate vector of x *relative to* β*, is defined as*

$$[x]_\beta = \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{bmatrix} \tag{2.9}$$

Note that $\left[u_i\right]_\beta = e_i$.

Definition 47. *Let* V *with ordered basis* $\beta = \{v_i\}$*,* W *with ordered basis* $\gamma : \{w_i\}$*,* $T : V \to W$ *be linear. There exists unique scala* $a_{ij} \in F$ *such that*

$$T(v_j) = \sum_{i=1}^{m} a_{ij} w_j \tag{2.10}$$

The $m \times n$ *matrix*[1] A *defined by* $A_{ij} = a_{ij}$ *is the matrix representation of* T *in the ordered basis* β *and* γ *and write* $A = [T]_\beta^\gamma$*. If* $V = W$ *and* $\beta = \gamma$*, we write* $A = [T]_\beta$*.* □

Note that the j-th column of A is $\left[T(v_j)\right]_\gamma$: $[T]_\beta^\gamma = \left[\ldots, \left[T(v_j)\right]_\gamma, \ldots\right]$.

Note that T is the relationship between two basis. The value of T might be the same as basis, for example when they are operators on F^n, but T and basis are different objects. It is easy to confuse them, especially on F^n.

Theorem 37. *If* $U, T : V \to W$ *are linear transformation that* $[U]_\beta^\gamma = [T]_\beta^\gamma$*, then* $U = T$*.*

Definition 48. $\mathscr{L}(V, W)$ *contains all linear transformation from* V *to* W*.*

Theorem 38. *Let* T, U *be linear transformation over* V *and* W*,*

1. $[T+U]_\beta^\gamma = [T]_\beta^\gamma + [U]_\beta^\gamma$
2. $[aT]_\beta^\gamma = a\,[T]_\beta^\gamma$ *for all scalar* a

Theorem 39. *let* $T : V \to W$ *and* $U : W \to Z$*. Then* $UT : V \to Z$ *is linear.*

Definition 49. *Let* $T : V \to W$ *and* $U : W \to Z$ *be linear transformation.* $A_{m\times n} = [U]_\alpha^\beta$ *and* $B_{n\times p} = [T]_\beta^\gamma$ *where* $\alpha = \{v_i\}$*,* $\beta = \{w_i\}$*,* $\gamma = \{z_i\}$*. Define the product of matrix* AB *as:*

$$(AB)_{ij} = \sum_{k=1}^{n} A_{ik} B_{kj} \tag{2.11}$$

then

$$[UT]_\alpha^\gamma = [U]_\beta^\gamma \, [T]_\alpha^\beta \tag{2.12}$$

Proof. For product $AB = [UT]_\alpha^\gamma$, we have

$$\begin{aligned}
(UT)(v_j) = U(T(v_j)) &= U\left(\sum_{k=1}^{m} B_{kj} w_k\right) = \sum_{k=1}^{m} B_{kj} U(w_k) \\
&= \sum_{k=1}^{m} B_{kj}\left(\sum_{i=1}^{p} A_{ik} z_i\right) = \sum_{k=1}^{m}\left(\sum_{i=1}^{p} A_{ik} B_{kj}\right) z_i \\
&= \sum_{i=1}^{p} C_{ij} z_i
\end{aligned} \tag{2.13}$$

□

Definition 50. *the Kronecker delta* δ_{ij} *is defined as*

$$\delta_{ij} = \begin{cases} 1 & , \text{if } i = j \\ 0 & , \text{if } i \neq j \end{cases} \tag{2.14}$$

Definition 51. *The* $n \times n$ *identity matrix* I_n *is defined as* $\left(I_n\right)_{ij} = \delta_{ij}$*.*

Theorem 40. *Let* u_j *and* v_j *be the jth column of* AB *and* B*, then*

1. $u_j = Av_j$: $AB = \left[Av_1, Av_2, \ldots, Av_j, \ldots, Av_p\right]$
2. $v_j = Be_j$: $B = B \times I_n$

Theorem 41. *Let* $T : V \to W$ *be linear, we have*

$$\left[T(u)\right]_\gamma = [T]_\beta^\gamma \, [u]_\beta \tag{2.15}$$

[1]The word matrix is Latin for womb which is the same root as matrimony. The idea is that a matrix is a receptacle for holding numbers.

Proof. Fix $u \in \mathrm{V}$, and define linear transformation $f : \mathrm{F} \to \mathrm{V}$ by $f(a) = au$ and $g : \mathrm{F} \to \mathrm{W}$ by $g(a) = a\mathrm{T}(u)$. Let $a = \{1\}$ be the standard basis of F. Notice that $g = \mathrm{T}f$. we have:

$$[\mathrm{T}(u)]_\gamma = [g(1)]_\gamma = [g]_\alpha^\gamma = [\mathrm{T}f]_\alpha^\gamma = [\mathrm{T}]_\beta^\gamma [f]_\alpha^\beta = [\mathrm{T}]_\beta^\gamma [f(1)]_\beta = [\mathrm{T}]_\beta^\gamma [u]_\beta \tag{2.16}$$

□

Note: in the above proof, a vector could be treated as a linear transformation from a field to vector space.

Definition 52. *Let* A *be an* $m \times n$ *matrix. The mapping* $\underline{\mathrm{L_A}}$ *that* $\mathrm{L_A} : \mathrm{F}^n \to \mathrm{F}^m$ *defined by* $\mathrm{L_A}(x) = \mathrm{A}x$ *is called* *left-multiplication transformation* . □

A linear transformation is different from matrix:

1. Matrix is finite dimensional, so it defines relation only in finite dimension space. A linear transformation could be of any dimension.
2. For a transformation, its matrix representation depends on the chosen basis.

Theorem 42.

$$\begin{cases} [\mathrm{L_A}]_\alpha^\beta & = \mathrm{A} \\ \mathrm{L}_{[\mathrm{T}]_\alpha^\beta} & = \mathrm{T} \end{cases} \tag{2.17}$$

2.2.3 Inverse

Definition 53. *Let* $\mathrm{T} : \mathrm{V} \to \mathrm{W}$ *and* $\mathrm{U} : \mathrm{W} \to \mathrm{V}$ *be linear.* U *is an* *inverse* *of* T *if* $\mathrm{TU} = \mathrm{I_W}$ *and* $\mathrm{UT} = \mathrm{I_V}$. *If* T *has an inverse,* T *is* *invertable* *, which is denoted as* T^{-1}.

Theorem 43. $(\mathrm{UT})^{-1} = \mathrm{T}^{-1}\mathrm{U}^{-1}$.

Definition 54. *Let* A *be* $n \times n$ *matrix.* A *is invertable if there is an* $n \times n$ *matrix* B *that* $\mathrm{AB} = \mathrm{BA} = \mathrm{I}$.

Theorem 44. *if* T *is invertible,*

$$\left[\mathrm{T}^{-1}\right]_\gamma^\beta = \left([\mathrm{T}]_\beta^\gamma\right)^{-1}$$

Proof.

$$\mathrm{I}_n = \left[\mathrm{I_V}\right]_\beta = \left[\mathrm{T}^{-1}\mathrm{T}\right]_\beta = \left[\mathrm{T}^{-1}\right]_\gamma^\beta [\mathrm{T}]_\beta^\gamma$$

□

Definition 55. V *is* *isomorphic* *to* W *if there exists a linear transformation* $\mathrm{T} : \mathrm{V} \to \mathrm{W}$ *that is invertible.* T *is called an* *isomorphism* *from* V *to* W.

Theorem 45. V *is isomorphic to* W *if* $\dim(\mathrm{V}) = \dim(\mathrm{W})$.

Proof. If the dimensions are the same, choose basis β of V and γ of W and create a linear mapping $\mathrm{T} : \beta \to \gamma$ by Theorem 35. □

Theorem 46. *Let* V *be a vector space over* F. *Then* V *is isomorphic to* $\mathrm{F}^n \iff \dim(\mathrm{V}) = n$.

Theorem 47. *The function* $\Phi : \mathscr{L}(\mathrm{V}, \mathrm{M}) \to \mathrm{M}_{m\times n}(\mathrm{F})$ *defined by* $\Phi(\mathrm{T}) = [\mathrm{T}]_\beta^\gamma$, *is an isomorphism. The dimension has relation that*

$$\dim\left(\mathscr{L}(\mathrm{V}, \mathrm{M})\right) = \dim(\mathrm{V}) \times \dim(\mathrm{W}) \tag{2.18}$$

2.2.4 Change of Coordinate Matrix

Theorem 48. *Let* β *and* β' *be two ordered basis of* V. *Let* $\mathrm{Q} = \left[\mathrm{I_V}\right]_{\beta'}^\beta$, *then*

1. Q *is invertible.*
2. $\forall \alpha \in \mathrm{V}$, $[\alpha]_\beta = \mathrm{Q}[\alpha]_{\beta'} = \left[\mathrm{I_V}\right]_{\beta'}^\beta [\alpha]_{\beta'}$.

$\mathrm{Q} = \left[\mathrm{I_V}\right]_{\beta'}^\beta$ *is called* *change of coordinate matrix* *that changes from* β'*-coordinates to* β*-coordinates.*

Proof. $\forall \alpha \in \mathrm{V}$, $[\alpha]_\beta = \left[\mathrm{I_V}(\alpha)\right]_\beta = \left[\mathrm{I_V}\right]_{\beta'}^\beta [\alpha]_{\beta'} = \mathrm{Q}[\alpha]_{\beta'}$. □

If Q changes β'-coordinate into β-coordinate, Q^{-1} changes β-coordinate into β'-coordinate.

Definition 56. *A* *linear operator* *is a linear transformation that map from* V *to itself.*

Theorem 49. *If* T *is a linear operator on* V*, then*

$$[\mathrm{T}]_{\beta'} = \left[\mathrm{I_V}\right]_{\beta}^{\beta'} [\mathrm{T}]_{\beta} \left[\mathrm{I_V}\right]_{\beta'}^{\beta} = \mathrm{Q}^{-1} [\mathrm{T}]_{\beta}\, \mathrm{Q} \tag{2.19}$$

Proof. $\mathrm{Q}\,[\mathrm{T}]_{\beta'} = [\mathrm{I}]_{\beta'}^{\beta} [\mathrm{T}]_{\beta'}^{\beta'} = [\mathrm{IT}]_{\beta'}^{\beta} = [\mathrm{TI}]_{\beta'}^{\beta} = [\mathrm{T}]_{\beta}^{\beta} [\mathrm{I}]_{\beta'}^{\beta} = [\mathrm{T}]_{\beta}\, \mathrm{Q}.$ □

Theorem 50. *Let* $\mathrm{A} \in \mathrm{M}_{n\times n}(\mathrm{F})$*, and* $\gamma : \{a_i\}$ *is an ordered basis for* F^n*. Then* $\left[\mathrm{L_A}\right]_\gamma = \mathrm{Q}^{-1}\mathrm{AQ}$*, where* $\mathrm{Q} = \left[a_1, a_2, \ldots, a_n\right]$.

Proof. $\left[\mathrm{L_A}\right]_{\mathrm{I}} = \mathrm{A}$, so

$$[\mathrm{L_A}]_\gamma = \left[\mathrm{I_V}\right]_{\mathrm{I}}^{\gamma} \times \left[\mathrm{L_A}\right]_{\mathrm{I}} \times \left[\mathrm{I_V}\right]_{\gamma}^{\mathrm{I}} = \left[\mathrm{I_V}\right]_{\mathrm{I}}^{\gamma} \times \mathrm{A} \times \left[\mathrm{I_V}\right]_{\gamma}^{\mathrm{I}}$$

A take aways is that Q is the change of coordinate matrix from γ to I. □

Theorem 51. *Let* $\mathrm{T} : \mathrm{V} \to \mathrm{W}$*,* β *and* β' *are ordered basis of* V*,* γ *and* γ' *are ordered basis of* W*. Then*

$$[\mathrm{T}]_{\beta'}^{\gamma'} = \left[\mathrm{I_W}\right]_{\gamma}^{\gamma'} [\mathrm{T}]_{\beta}^{\gamma} \left[\mathrm{I_V}\right]_{\beta'}^{\beta} \tag{2.20}$$

Example 1. *There is an example of the usage of change of coordinate matrix: do reflection operation* T *against a line* $y = ax$*. Let* β *be the standard basis of* R^2 *and* β' *be the standard basis of* R^2 *after the rotation of* $y = ax$*. The operation* T *has a matrix representation in* β'

$$[\mathrm{T}]_{\beta'} = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

Then calculate $[\mathrm{T}]_\beta$ *based on* $[\mathrm{T}]_{\beta'}$*.*

Definition 57. B *is* <u>*similar*</u> *to* A *if there is an invertible matrix* Q *that* $\mathrm{B} = \mathrm{Q}^{-1}\mathrm{AQ}$.

Theorem 52. *If* T *is a linear operator on finite dimension vector space* V*, and if* β *and* β' *are any ordered basis of* V*, then* $[\mathrm{T}]_{\beta'}$ *is similar to* $[\mathrm{T}]_\beta$*.*

2.2.5 Quotient Space

Definition 58. *Let subspace* $\mathrm{U} \subset \mathrm{V}$*, The* <u>*affine subset*</u> $v + \mathrm{U}$ *of* V *is defined as:*

$$v + \mathrm{U} = \{v + u : u \in \mathrm{U}\} \tag{2.21}$$

Definition 59. *Let subspace* $\mathrm{U} \subset \mathrm{V}$*. Then the* <u>*quotient space*</u> V/U *is defined as:*

$$\mathrm{V}/\mathrm{U} = \{v + \mathrm{U} : v \in \mathrm{V}\} \tag{2.22}$$

Definition 60. *Let subspace* $\mathrm{U} \subset \mathrm{V}$*. The* <u>*quotient map*</u> $\pi : \mathrm{V} \to \mathrm{V}/\mathrm{U}$ *is defined as:*

$$\pi(v) = v + \mathrm{U} \tag{2.23}$$

Theorem 53.

$$\dim\left(\mathrm{V}/\mathrm{U}\right) = \dim(\mathrm{V}) - \dim(\mathrm{U}) \tag{2.24}$$

Proof. Define $\pi : \mathrm{V} \to \mathrm{V}/\mathrm{U}$. The null space is U. □

Theorem 54. *Define* $\tilde{\mathrm{T}} : \mathrm{V}/\mathscr{N}(\mathrm{T}) \to \mathrm{W}$ *by:*

$$\tilde{\mathrm{T}}\left(v + \mathscr{N}(\mathrm{T})\right) = \mathrm{T}v$$

Then $\tilde{\mathrm{T}}$ *is an isomorphism between* $\mathrm{V}/\mathscr{N}(\mathrm{T})$ *and* T*.*

Proof. If $u + \mathscr{N}(\mathrm{T}) = v + \mathscr{N}(\mathrm{T})$, then $u - v \in \mathscr{N}(\mathrm{T})$. So $\mathrm{T}(u - v) = \mathrm{T}(u) - \mathrm{T}(v) = 0$ and $\mathrm{T}(u) = \mathrm{T}(v)$. □

2.2.6 Dual Space

Definition 61. *A linear functional is a linear transformation that map from* V *into* F.

Definition 62. *An i-th coordinate function f_i with respect to basis β is defined as $f_i(x) = a_i$ where*

$$[x]_\beta = \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{bmatrix} = \begin{bmatrix} f_1(a) \\ f_2(a) \\ \vdots \\ f_n(a) \end{bmatrix}$$

Definition 63. *The dual space of* V *is the vector space* $\mathrm{V}^* = \mathscr{L}(\mathrm{V},\mathrm{F})$. *The double dual space* V^{**} *is the dual space of* V^*.

The dimension of dual space is $\dim(\mathrm{V}^*) = \dim(\mathscr{L}(\mathrm{V},\mathrm{F})) = \dim(\mathrm{V}) \times \dim(\mathrm{F}) = \dim(\mathrm{V})$.

Definition 64. *Let $\beta = \{x_i\}$ be an ordered basis for finite dimensional vector space* V. *Define $f_i(x) = a_i$ where*

$$[x]_\beta = \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{bmatrix}$$

f_i is the i-th coordinate function with respect to basis β. let $\beta^* = \{f_i\}$. Then β^* is an ordered basis for V^*, *and $\forall f \in \mathrm{V}^*$, we have*

$$f = \sum_{i=1}^{n} f(x_i) f_i \tag{2.25}$$

β^* is called the dual basis of β.

Proof. Let $g = \sum_{i=1}^{n} f(x_i) f_i$, we have

$$g(x_j) = \left(\sum_{i=1}^{n} f(x_i) f_i\right)(x_j) = \sum_{i=1}^{n} f(x_i) f_i(x_j) = \sum_{i=1}^{n} f(x_i)\delta_{ij} = f(x_j)$$

□

Theorem 55. *Let* V *and* W *be vector space over* F *with ordered basis β and γ. For any linear transformation* $\mathrm{T} : \mathrm{V} \to \mathrm{W}$, *the mapping* $\mathrm{T}^t : \mathrm{W}^* \to \mathrm{V}^*$ *defined as* $\mathrm{T}^\top(g) = g\mathrm{T}, \forall g \in \mathrm{W}^*$ *is a linear transformation with property that* $\left[\mathrm{T}^\top\right]_{\gamma^*}^{\beta^*} = \left([\mathrm{T}]_\beta^\gamma\right)^\top$.

Proof. Let $\beta = \{x_i\}$ and $\gamma = \{y_i\}$ with dual basis $\beta^* = \{f_i\}$ and $\gamma^* = \{g_i\}$, $\mathrm{A} = [\mathrm{T}]_\beta^\gamma$. we have

$$\mathrm{T}^\top(g_j) = g_j\mathrm{T} = \sum_{s=1}^{n} (g_j\mathrm{T})(x_s) f_s$$

So the row i, column j entry of $[\mathrm{T}^\top]_{\gamma^*}^{\beta^*}$ is

$$(g_j\mathrm{T})(x_i) = g_j(\mathrm{T}(x_i)) = g_j\left(\sum_{k=1}^{m} \mathrm{A}_{kj} y_k\right) = \sum_{k=1}^{m} \mathrm{A}_{kj} g_j(y_k) = \sum_{k=1}^{m} \mathrm{A}_{kj}\delta_{kj} = \mathrm{A}_{ji}$$

Hence $\left[\mathrm{T}^\top\right]_{\gamma^*}^{\beta^*} = \mathrm{A}^\top$. □

Definition 65. *For* $\mathrm{U} \subset \mathrm{V}$, *the annihilator of* U, *denoted as* U_V^0, *is defined as*

$$\mathrm{U}_\mathrm{V}^0 = \{\phi \in \mathrm{V}^* : \phi(u) = 0, \forall u \in \mathrm{U}\}$$

So the annihilator map U *to* 0. *For vectors in* V − U, *the mapping could be any result. The annihilator is a subspace.*

Theorem 56.

$$\dim(\mathrm{U}) + \dim\left(\mathrm{U}_{\mathrm{V}}^{0}\right) = \dim(\mathrm{V}) \tag{2.26}$$

Proof. Define $i \in \mathscr{L}(\mathrm{U},\mathrm{V})$ that $i(u) = u, \forall u \in \mathrm{U}$. $i^* \in \mathscr{L}(\mathrm{V}^*,\mathrm{U}^*)$. So

$$\dim\left(\mathscr{R}(i^*)\right) + \dim\left(\mathscr{N}(i^*)\right) = \dim\left(\mathrm{V}^*\right)$$

By definition, $\mathscr{N}(i^*) = \mathrm{U}_{\mathrm{V}}^{0}$. Also $\mathscr{R}(i^*) = \mathrm{U}^*$. □

Theorem 57. *Let* V *and* W *be two finite-dimentional vector space, and* $\mathrm{T} \in \mathscr{L}(\mathrm{V},\mathrm{W})$. *Then:*

1. $\mathscr{N}(\mathrm{T}^*) = (\mathscr{R}(\mathrm{T}))^0$
2. $\mathscr{R}(\mathrm{T}^*) = (\mathscr{N}(\mathrm{T}))^0$
3. $\dim\left(\mathscr{R}(\mathrm{T}^*)\right) = \dim\left(\textit{range}\ \mathrm{T}\right)$
4. $\dim\left(\mathscr{N}(\mathrm{T}^*)\right) = \dim\left(\mathscr{N}(\mathrm{T})\right) + \dim(\mathrm{W}) - \dim(\mathrm{V})$

Proof. Suppose $\varphi \in \text{null } \mathrm{T}^*$. Then $0 = \mathrm{T}^*(\varphi) = \varphi\mathrm{T}$. Then

$$0 = (\varphi\mathrm{T})(v) = \varphi(\mathrm{T}v)$$

So $\varphi \in (\text{range } \mathrm{T})_{\mathrm{W}}^{0}$.

$$\begin{aligned}
\dim\left(\mathscr{R}(\mathrm{T}^*)\right) &= \dim\left(\mathrm{W}^*\right) - \dim\left(\mathscr{N}(\mathrm{T}^*)\right) \\
&= \dim(\mathrm{W}) - \dim\left(\mathscr{R}(\mathrm{T})^0\right) \\
&= \dim\left(\mathscr{R}(\mathrm{T})\right)
\end{aligned}$$

$$\begin{aligned}
\dim\left(\mathscr{N}(\mathrm{T}^*)\right) &= \dim\left(\mathscr{R}(\mathrm{T})^0\right) \\
&= \dim(\mathrm{W}) - \dim\left(\mathscr{R}(\mathrm{T})\right) \\
&= \dim(\mathrm{W}) - (\dim(\mathrm{V}) - \dim\left(\mathscr{N}(\mathrm{T})\right)) \\
&= \dim(\mathrm{W}) + \dim\left(\mathscr{N}(\mathrm{T})\right) - \dim(\mathrm{V})
\end{aligned}$$

□

Definition 66. *For vector* $x \in \mathrm{V}$, *define* $\hat{x} : \mathrm{V}^* \to \mathrm{F}$ *by* $\hat{x}(f) = f(x)$. $\hat{x}$ *is a linear functional on* V^*, *so* $\hat{x} \in \mathrm{V}^{**}$.

Theorem 58. *Define* $\psi : \mathrm{V} \to \mathrm{V}^{**}$ *by* $\psi(x) = \hat{\mathrm{X}}$. *Then* ψ *is an isomorphism.*

Theorem 59. *Let* V *be a finite dimension vector space with dual space* V^*. *Every ordered basis for* V^* *is the dual basis for some basis for* V.

$$\begin{tikzcd}
\mathrm{V} \arrow[rrrr, "\mathrm{T}"] \arrow[ddrr] & & & & \mathrm{W} \arrow[ddll] \\
& \mathrm{V}^* & & \mathrm{W}^* \arrow[ll, "\mathrm{T}^{\top}"'] & \\
& & \mathrm{F} & &
\end{tikzcd}$$

2.3 Linear Equations

2.3.1 Elementary Operations

Definition 67. *Let* A *be an* $m \times n$ *matrix. there are three* *elementary row operation:*

1. *interchange any two row of* A*.*
2. *multiply any row of* A *by nonzero scalar.*
3. *add any scalar multiple of a row of* A *to another row.*

Definition 68. *An* $n \times n$ *elementary matrix is a matrix obtained by performing* one *elementary operation on* I_n*.*

Definition 69. *The rank of* $\mathrm{A}_{m\times n}$*, denoted* $\mathrm{rank}(\mathrm{A})$*, is the rank*[2] *of linear transformation* $\mathrm{L_A} : \mathrm{F}^n \to \mathrm{F}^m$*.*

Theorem 60. *the rank of a matrix equals the maximum number of linearly independent columns.*

Proof. For any $\mathrm{A} \in \mathrm{M}_{m\times n}(\mathrm{F})$,

$$\begin{aligned}\mathrm{rank}(\mathrm{A}) = \mathrm{rank}(\mathrm{L_A}) = \dim\left(\mathrm{R}(\mathrm{L_A})\right) &= \mathrm{span}\left(\mathrm{L_A}(\beta)\right)\\ &= \mathrm{span}\left(\left\{\mathrm{L_A}(e_1), \mathrm{L_A}(e_2), \dots, \mathrm{L_A}(e_n)\right\}\right)\end{aligned}$$

we have $\mathrm{L_A}(e_j) = \mathrm{A}e_j = a_j$ where a_j is the jth column of A. Hence

$$\mathrm{R}(\mathrm{L_A}) = \mathrm{span}\left(\left\{a_1, a_2, \dots, a_n\right\}\right)$$

□

Theorem 61. *Let* $\mathrm{A}_{m\times n}$ *has rank* r*. Then there exist invertible matrix* $\mathrm{B}_{m\times m}$ *and* $\mathrm{C}_{n\times n}$ *that* $\mathrm{D} = \mathrm{BAC}$*, where:*

$$\mathrm{D} = \begin{bmatrix} \mathrm{I}_r & 0 \\ 0 & 0 \end{bmatrix}$$

Theorem 62. *Every invertible matrix is a product of elementary matrices.*

Definition 70. *For system* $\mathrm{A}x = b$*, the matrix* $(\mathrm{A}|b)$ *is the augmented matrix.*

Theorem 63. *If A is an invertible matrix, it is possible to transform augmented matrix* $(\mathrm{A}|\mathrm{I}_n)$ *into matrix* $(\mathrm{I}_n|\mathrm{A}^{-1})$ *by means of a finite number of elementary row operations.*

2.3.2 System of Equations

Definition 71. *A system* $\mathrm{A}_{m\times n}x = b$ *of* m *linear equation in* n *unknowns is homogeneous if* $b = 0$*. Otherwise the system is nonhomogeneous.*

Definition 72. *A system is consistent if its solution set is not empty. otherwise it is called inconsistent.*

Theorem 64. *Let* K *be the set of all solutions for* $\mathrm{A}x = 0$*. Then* $\mathrm{K} = \mathcal{N}(\mathrm{L_A})$ *has dimension of* $n - \mathrm{rank}(\mathrm{L_A}) = n - \mathrm{rank}(\mathrm{A})$*.*

Theorem 65. *if* $m < n$*, the system* $\mathrm{A}x = 0$ *has nonzero solution.*

Proof. $\mathrm{rank}(\mathrm{A}) \le m < n$, so $\mathcal{N}(\mathrm{A}) = n - \mathrm{rank}(\mathrm{A}) > 0$. □

Theorem 66. *Let* K *be the solution set of* $\mathrm{A}x = b$*,* $\mathrm{K_H}$ *be the solution set of* $\mathrm{A}x = 0$*. Then for all solution* s *to* $\mathrm{A}x = b$*,*

$$\mathrm{K} = \{s\} + \mathrm{K_H} = \left\{s + k : k \in \mathrm{K_H}\right\} \tag{2.27}$$

Theorem 67. *Let* $\mathrm{A}_{n\times n}x = b$ *be a system of equations. If* A *is invertible, the solution is* $\mathrm{A}^{-1}b$*. Conversely, if the system has exactly one solution,* A *is invertible.*

Theorem 68. *Let* $\mathrm{A}x = b$ *be a system of linear equations. the system is consistent* $\iff$ *rank*$(\mathrm{A}) =$ *rank*$(\mathrm{A}|b)$*.*

Proof. $\mathrm{R}(\mathrm{L_A}) = \mathrm{span}\left(\left\{a_1, a_2, \dots, a_n\right\}\right)$. Since $b \in \mathrm{R}(\mathrm{L_A})$, the extended span is the same. □

Definition 73. *A matrix is in reduced row echelon form if:*

[2]The rank of a linear transformation is defined in Definition (43) on page 19.

1. *any row containing a nonzero entry precedes any row in which all the entries are zero.*
2. *the first nonzero entry in each row is the only nonzero entry in its column.*
3. *the first nonzero entry in each row is* 1 *and it occurs in a column to the right of the first nonzero entry in the preceding row.*

Theorem 69. *For* $\mathrm{A}_{m\times n}$ *and* $\mathrm{B}_{n\times p}$*, we have:*

$$\operatorname{rank}(\mathrm{AB}) = \operatorname{rank}(\mathrm{B}) - \dim\left(\mathscr{N}(\mathrm{A}) \cap \mathscr{R}(\mathrm{B})\right) \tag{2.28}$$

Proof. Let β_i be the basis of $\mathscr{N}(\mathrm{A}) \cap \mathscr{R}(\mathrm{B})$, expand to the basis $\beta \cup \alpha$ of B. Prove α is a basis of $\mathscr{R}(\mathrm{AB})$. □

Theorem 70. *For* $\mathrm{A}_{m\times n}$*, we have*

1. $\operatorname{rank}(\mathrm{A}^\top\mathrm{A}) = \operatorname{rank}(\mathrm{A}) = \operatorname{rank}(\mathrm{A}\mathrm{A}^\top)$.
2. $\mathscr{R}(\mathrm{A}^\top\mathrm{A}) = \mathscr{R}(\mathrm{A}^\top)$.
3. $\mathscr{N}(\mathrm{A}^\top\mathrm{A}) = \mathscr{N}(\mathrm{A})$.

$\mathrm{A}^\top$ *could be replaced by* A^* *in* $\mathbb{C}$.

Proof. If $\exists x \neq 0\left(x \in \mathscr{N}(\mathrm{A}^\top) \cap \mathscr{R}(\mathrm{A})\right)$. Then $(\mathrm{A}^\top x = 0) \wedge \left(\exists y(x = \mathrm{A}y)\right)$. So $x^\top x = y^\top \mathrm{A}^\top x = y^\top(\mathrm{A}^\top x) = 0$ and then $x = 0$. According to Theorem 69, $\operatorname{rank}(\mathrm{A}^\top\mathrm{A}) = \operatorname{rank}(\mathrm{A}^\top) - \dim\left(\mathscr{N}(\mathrm{A}^\top) \cap \mathscr{R}(\mathrm{A})\right) = \operatorname{rank}(\mathrm{A})$. □

Theorem 71. *For a system of linear equation* $\mathrm{A}x = b$*, the associated system of* normal equations *is defined as* $n \times n$ *system*

$$\mathrm{A}^\top\mathrm{A}x = \mathrm{A}^\top b \tag{2.29}$$

$\mathrm{A}^\top\mathrm{A}x = \mathrm{A}^\top b$ *is always consistent and has unique solution when* $\operatorname{rank}(\mathrm{A}) = n$*. If* $\mathrm{A}x = b$ *is consistent, two solutions are the same.* □

2.4 Determinants

Definition 74. *Let* $\mathrm{A} \in \mathrm{M}_{n\times n}(\mathrm{F})$. *If* $n = 1$, *let* $\mathrm{A} = (\mathrm{A}_{11})$ *and we define* $det(\mathrm{A}) = \mathrm{A}_{11}$. *For* $n \geq 2$, $det(\mathrm{A})$ *(or* $|\mathrm{A}|$*) is defined as*

$$|\mathrm{A}| = \sum_{j=1}^{n} (-1)^{i+j} \mathrm{A}_{ij} \times \left|\tilde{\mathrm{A}}_{ij}\right| \tag{2.30}$$

where $\tilde{\mathrm{A}}_{ij}$ *is obtained from* A *by deleting row i and column j. This is called Laplace expansion.* □

Theorem 72. *A function* $\delta : \mathrm{M}_{n\times n}(\mathrm{F}) \to \mathrm{F}$ *is the same as* $|\mathrm{A}|$ *if it satisfies the following 3 properties:*

1. *It is n-linear function: for a scalar k,*

$$\left|\begin{bmatrix} a_1 \\ \vdots \\ u + kv \\ \vdots \\ a_n \end{bmatrix}\right| = \left|\begin{bmatrix} a_1 \\ \vdots \\ u \\ \vdots \\ a_n \end{bmatrix}\right| + k \left|\begin{bmatrix} a_1 \\ \vdots \\ v \\ \vdots \\ a_n \end{bmatrix}\right| \tag{2.31}$$

2. *It is alternating:* $\delta(\mathrm{A}) = 0$ *if any two adjacent rows are identical.*
3. $\delta(\mathrm{I}) = 1$.

The determinate is linear on each row when the remaining rows are held fixed. □

Theorem 73. *The effect of elementary row operation on the determinant of a matrix* A *is:*

1. *interchange any two rows:* $|\mathrm{B}| = -|\mathrm{A}|$.
2. *multiply a row:* $|\mathrm{B}| = k|\mathrm{A}|$.
3. *add a multiple of a row to another:* $|\mathrm{B}| = |\mathrm{A}|$.

Theorem 74. *If* $\mathrm{rank}(\mathrm{A}_{n\times n}) < n$, *then* $|\mathrm{A}| = 0$.

Proof. If $\mathrm{rank}(\mathrm{A}_{n\times n}) < n$, one row is a linear combination of all other rows. □

Theorem 75.

$$|\mathrm{AB}| = |\mathrm{A}| \times |\mathrm{B}| \tag{2.32}$$

Theorem 76. *A matrix* $\mathrm{A} \in \mathrm{M}_{n\times n}(\mathrm{F})$ *is invertible* $\Longleftrightarrow |\mathrm{A}| \neq 0$. *If it is invertible,* $\left|\mathrm{A}^{-1}\right| = \frac{1}{|\mathrm{A}|}$.

Definition 75. *The cofactor of* A *is defined as*

$$\mathrm{cof}\ \mathrm{A}_{ij} = (-1)^{i+j} \left|\tilde{\mathrm{A}}_{ij}\right| \tag{2.33}$$

□

If the determinate is calculated using cofactor operation, the performance is $n!$ multiplication. However if it is calculated using elementary row operation, the performance is $\frac{n^3 + 2n - 3}{3}$ multiplication.

Definition 76. *The adjugate of* A *is defined as*

$$\mathrm{adj}\ \mathrm{A} = (\mathrm{cof}\ \mathrm{A})^{\top} \tag{2.34}$$

Theorem 77. *The inverse of invertible square matrix A is:*

$$\mathrm{A}^{-1} = \frac{1}{|\mathrm{A}|} \mathrm{adj}\ \mathrm{A}$$

Theorem 78 (Cramer's Rule)**.** *Let* $\mathrm{A}x = b$ *be a system of n equation with n unknowns. If* $|\mathrm{A}| \neq 0$, *the system has a unique solution:*

$$x_k = \frac{\left|\mathrm{M}_k\right|}{|\mathrm{A}|} \tag{2.35}$$

where M_k *is a* $n \times n$ *matrix obtained from* A *by replacing column k of* A *by b.*

Proof. Let a_k be the kth column of A and X_k denote the matrix obtained from replacing the column k of identity matrix I_n by x. Then $\mathrm{AX}_k = \mathrm{M}_k$:

$$\begin{aligned}
\mathrm{AX}_k &= \mathrm{A}\begin{bmatrix} 1 & & & x & & \\ & 1 & & x & & \\ & & \ddots & \vdots & & \\ & & & x & & \\ & & & \vdots & \ddots & \\ & & & x & & 1 \end{bmatrix} \\
&= \left[\mathrm{A}e_1, \mathrm{A}e_2, \dots, \mathrm{A}x, \dots, \mathrm{A}e_n\right] \\
&= \left[a_1, a_2, \dots, b, \dots, a_n\right] \\
&= \mathrm{M}_k
\end{aligned}$$

Evaluate X_k by cofactor expansion along row k produces

$$\left|\mathrm{X}_k\right| = x_k \times \left|\mathrm{I}_{n-1}\right| = x_k$$

Hence

$$\left|\mathrm{M}_k\right| = \left|\mathrm{AX}_k\right| = |\mathrm{A}| \times \left|\mathrm{X}_k\right| = |\mathrm{A}| \times x_k$$

Therefore

$$x_k = \frac{\left|\mathrm{M}_k\right|}{|\mathrm{A}|}$$

□

Note: Cramer's Rule is too slow for real world calculation.

Theorem 79. *In geometry, for a square matrix* $\mathrm{A} \in \mathrm{M}_{n\times n}(\mathrm{F})$*,* $|\det \mathrm{A}|$ *is the* *n-dimensional volume* *of the parallelepiped having vector* $\mathrm{A}_{i,\cdot}$ *as adjacent sides.*

2.5 Diagonalization

There are two questions for a linear operator T:

1. Is there an ordered basis β that $[\mathrm{T}]_\beta$ is a diagonal matrix?
2. If such basis exists, how can it be found?

2.5.1 Eigenvalue and Eigenvectors

Definition 77. *A linear operator* T *on* V *is diagonalizable if there is an ordered basis β of* V *that $[\mathrm{T}]_\beta$ is a diagonal matrix. A matrix is diagonalizable if $\mathrm{L_A}$ is diagonalizable.*

If an operator T *is diagonalizable, for $\beta = \{v_i\}$, we have*

$$\mathrm{T}(v_j) = \sum_{i=1}^{n} \mathrm{D}_{ij} v_j = \mathrm{D}_{jj} v_j = \lambda_j v_j$$

So to prove a linear operator T *is diagnolizable is to find a basis $\beta = \{v_i\}$ and $\{\lambda_j\}$ that $\mathrm{T}(v_i) = \lambda_i v_i$.* □

Definition 78. *A* non-zero *vector $v \in \mathrm{V}$ is called an eigenvector of linear operator* T *if $\exists \lambda : \mathrm{T}(v) = \lambda v$. λ is called eigenvalue corresponding to eigenvector v. Eigenvector is also called characteristic vector. Eigenvalue is also called characteristic value.*

A eigenvalue could be 0*, but eigenvector could not be $\vec{0}$. An eigenvector is an invariant subspace of dimension* 1.

Theorem 80. *A linear operator* T *is diagonalizable if there exists an ordered basis consisting of eigenvectors of* T.

Theorem 81. *λ is an eigenvalue of* A $\iff \left|\mathrm{A} - \lambda \mathrm{I}_n\right| = 0$.

Proof. If λ is an eigenvalue of A, $\exists v \in \mathrm{F}^n, v \neq 0$ that $\mathrm{A}v = \lambda v$, which is $(\mathrm{A} - \lambda \mathrm{I}_n)(v) = 0$, which means $\mathrm{A} - \lambda \mathrm{I}_n$ is not invertible because $v \neq 0$, so $\left|\mathrm{A} - \lambda \mathrm{I}_n\right| = 0$. □

Theorem 82. *Every eigenvalue has at least one eigenvector.*

Proof. Since $\left|\mathrm{A} - \lambda \mathrm{I}_n\right| = 0$, $(\mathrm{A} - \lambda \mathrm{I}_n)x = 0$ is a homogeneous equation with $\dim\left(\mathrm{A} - \lambda \mathrm{I}_n\right) < n$. □

Definition 79. *For $\mathrm{A} = [\mathrm{T}]_\beta$ the polynomial $f_\mathrm{A}(t) = \left|\mathrm{A} - t\mathrm{I}_n\right|$ is called the characteristic polynomial of* A *and* T.

Theorem 83. *For all eigenvalues λ_i of* A*, define*

$$\mathrm{S}_k(\mathrm{A}) = \sum_{1 \le j_1 \le j_2 \le \cdots \le j_k} \prod_{j=1}^{k} \lambda_{i_j} \tag{2.36}$$

that is $\mathrm{S}_k(\mathrm{A})$ is the sum of the product of all k eigenvalues, which is the coefficient of characteristic polynomial of $f_\mathrm{A}(t)$:

$$f_\mathrm{A}(t) = (-1)^n t^n + (-1)^{n-1}\mathrm{S}_1(\lambda)t^{n-1} + \cdots + (-1)^{n-k}\mathrm{S}_k t^{n-1} + \cdots + \mathrm{S}_n \tag{2.37}$$

Define the sum of all[3] *principal minor of size k of* A *as $\mathrm{E}_k(\mathrm{A})$. We have*

$$\mathrm{E}_k(\mathrm{A}) = \mathrm{S}_k(\mathrm{A}) \tag{2.38}$$

So

$$tr\mathrm{A} = \sum \lambda_i \tag{2.39}$$

and

$$|\mathrm{A}| = \prod \lambda_i \tag{2.40}$$

Proof. calculate the coefficient by $\left.\dfrac{1}{k!} \dfrac{\mathrm{d}^k f_\mathrm{A}(t)}{\mathrm{d}t^k}\right|_{t=0}$ □

Theorem 84. *The choice of basis β did not change the eigenvalue of* T.

Proof.

$$\left|[\mathrm{T}]_\beta - \lambda \mathrm{I}\right| = \left|\mathrm{Q}^{-1}\left([\mathrm{T}]_\alpha - \lambda \mathrm{I}\right)\mathrm{Q}\right| = \left|\mathrm{Q}^{-1}\right| \times \left|[\mathrm{T}]_\alpha - \lambda \mathrm{I}\right| \times |\mathrm{Q}| = \left|[\mathrm{T}]_\alpha - \lambda \mathrm{I}\right|$$

□

[3] There are $\binom{n}{k}$ of them.

Theorem 85. *Similar matrices have the same characteristic function.*

Proof. Assume A is similar to B: $A = P^{-1}BP$. We have

$$f_A(\lambda) = |Ax - \lambda I| = \left|P^{-1}BP - \lambda P^{-1}P\right| = \left|P^{-1}\right| \times |B - \lambda I| \times |P| = |B - \lambda I| = f_B(\lambda)$$

□

Theorem 86. *if Q is a matrix with columns of eigenvectors of β, then according to Theorem 51 , $Q^{-1}AQ$ is a diagonal matrix with eigenvalue.*

2.5.2 Diagonalizability

Theorem 87. *Let λ_i be distinct eigenvalue of* T. *If $\left\{v_i\right\}$ are eigenvector that corresponding to λ_i, then $\left\{v_i\right\}$ is* linearly independent.

Proof. suppose it works for $k-1 \geq 1$ and we have k eigenvector $\{v_i\}$. Suppose

$$a_1v_1 + a_2v_2 + \cdots + a_kv_k = 0$$

multiply $T - \lambda_k I$ to both sides, we have

$$a_1(\lambda_1 - \lambda_k)v_1 + a_1(\lambda_2 - \lambda_k)v_2 + \cdots + a_1(\lambda_{k-1} - \lambda_k)v_{k-1} + = 0$$

because $\left\{v_1, v_2, \ldots, v_{k-1}\right\}$ are linearly independent, we have

$$a_1(\lambda_1 - \lambda_k) = a_1(\lambda_2 - \lambda_k) = a_1(\lambda_{k-1} - \lambda_k) = 0$$

because λ_i are different, we have $a_i = 0$. □

Theorem 88. *if* T *has n distinct eigenvalues, then* T *is diagonalizable. If* T *is diagonalizable, it may not have n distinct eigenvalues, for example the identity matrix I_V.*

Definition 80. *A polynomial $f(t)$ in P(F) <u>split over</u> F if there are scalars $c, a_1, \ldots, a_n$ (not necessarily distinct) in F that*

$$f(t) = c(t - a_1)(t - a_2)\ldots(t - a_n)$$

the <u>multiplicity</u> of λ is the largest positive integer k for which $(t - \lambda)^k$ is a factor of $f(t)$.

Theorem 89. *the characteristic polynomial of any diagonalizable linear operator splits.*

Proof. choose a basis β of eigenvectors. $[T]_\beta$ is a diagonal matrix D. The characteristic polynomial of T is $|D - tI|$ splits. □

Be careful that the characteristic polynomial splits does not mean the matrix is diagonalizable. The eigenvectors need to form a basis.

Definition 81. *let λ be an eigenvalue of* T. *Let $E_\lambda = \mathcal{N}(T - \lambda I_V)$. the set E_λ is called the <u>eigenspace</u> of* T *corresponding to eigenvalue λ. So is it for matrix.*

Theorem 90. *let λ be an eigenvalue of* T *having multiplicity m. then $1 \leq \dim\left(E_\lambda\right) \leq m$.*

Proof. choose ordered basis $\left\{v_1, v_2, \ldots, v_p\right\}$ for E_λ, and extend it to ordered basis $\beta = \left\{v_1, v_2, \ldots, v_p, v_{p+1}, \ldots, v_n\right\}$ for V, and let $A = [T]_\beta$. let $v_i (1 \leq i \leq q)$ be an eigenvector of T corresponding to λ, we have

$$A = \begin{pmatrix} \lambda I_p & B \\ 0 & C \end{pmatrix}$$

so

$$\begin{aligned} f(t) &= \left|A - tI_n\right| \\ &= \left|\begin{bmatrix} (\lambda - t)I_p & B \\ 0 & C - tI_{n-p} \end{bmatrix}\right| \\ &= \left|(\lambda - t)I_p\right| \times \left|C - tI_{n-p}\right| \\ &= (\lambda - t)^p g(t) \end{aligned}$$

So $(\lambda - t)^p$ is a factor of $f(t)$, and the multiplicity of λ is at least $p = \dim(E_\lambda)$, so $\dim(E_\lambda) \leq m$ □

Theorem 91. *let $\{\lambda_1, \lambda_2, \dots, \lambda_k\}$ be distinct eigenvalue of* T. *let* S_i *be a finite linearly independent subset of eigenspace* E_{λ_i}. *then* $\mathrm{S}_1 \cup \mathrm{S}_2 \cup \dots \cup \mathrm{S}_k$ *is a linearly independent subset of* V.

Theorem 92. *let* $\lambda_1, \lambda_2, \dots, \lambda_k$ *be distinct eigenvalue of* T*, then*

1. T *is diagonalizable* $\iff$ *the multiplicity of* λ_i *is equal to* $\dim\left(\mathrm{E}_{\lambda_i}\right)$ *for all i.*
2. *If* T *is diagonalizable and* β_i *is an ordered basis for* E_{λ_i} *for each i, then* $\beta = \beta_1 \cup \beta_2 \cup \dots \cup \beta_k$ *is an ordered basis for* V *consisting of eigenvectors of* T.

Theorem 93. T *is diagonalizable* $\iff$ *both of the following holds:*

1. *the characteristic polynomial of* T *splits.*
2. *for each eigenvalue* λ *of* T*, the multiplicity of* λ *equals* $n - \mathrm{rank}(\mathrm{T} - \lambda \mathrm{I})$.

Definition 82. *Let* W_i *be subspaces of a vector space* V. *The* <u>sum</u> *of these subspaces is defined as:*

$$\sum_{i=1}^{k} \mathrm{W}_i = \{v_1 + v_2 + \dots + v_k : v_i \in \mathrm{W}_i \text{ for } 1 \le i \le k\} \tag{2.41}$$

Definition 83. *let* W_i *be subspace of* V. V *is the* <u>direct sum</u> *of subspace* $\{\mathrm{W}_1, \mathrm{W}_2, \dots, \mathrm{W}_k\}$*, or* $\mathrm{V} = \mathrm{W}_1 \oplus \mathrm{W}_2 \oplus \dots \oplus \mathrm{W}_k$ *if*

$$\mathrm{V} = \sum_{i=1}^{k} \mathrm{W}_i$$

and

$$\mathrm{W}_j \cap \sum_{i \ne j} \mathrm{W}_i = \emptyset, (1 \le j \le k)$$

Theorem 94. T *is diagonalizable* $\iff$ V *is the direct sum of eigenspaces of* T.

2.5.3 Invariant Subspaces

Definition 84. *A subspace* W *of* V *is* T-<u>*invariant subspace*</u> *of* V *if* $\mathrm{T}(\mathrm{W}) \subseteq \mathrm{W}$. *Common* T*-invariant subspaces are:* $\emptyset$, V, $\mathrm{R}(\mathrm{T})$, $\mathrm{N}(\mathrm{T})$. □

Theorem 95. *A subspace* W *with basis* $\alpha = \{v_1, v_2, \dots, v_k\}$ *is* T*-invariant. Let* $\beta = \alpha \cup \gamma$ *as the expanded basis of* V. *Then*

$$[\mathrm{T}]_\beta = \begin{bmatrix} \mathrm{A}_{k \times k} & \mathrm{B} \\ 0 & \mathrm{C} \end{bmatrix} \tag{2.42}$$

The reverse is true. If $[\mathrm{T}]_\beta$ *has such representation, the first k basis of* β *is* T*-invariant.*

Definition 85. *A* T-<u>*cyclic subspace*</u> *of* V *generated by* x *is defined as* $\mathrm{W} = span\left(\{x, \mathrm{T}(x), \mathrm{T}^2(x), \dots\}\right)$.

Theorem 96. *Let* T *be a linear operator on finite-dimensional vector space* V*, and let* W *be a* T*-invariant subspace of* V. *Then the characteristic polynomial of* T_W *divides the characteristic polynomial of* T.

Proof. Choose ordered basis γ for W and expand it to β for V. Calculate $[\mathrm{T}]_\beta$ and $[\mathrm{T}]_\gamma$. □

Theorem 97. *Let* T *be a linear operator on finiate-dimensional vector space* V*, and let* W *be a* T*-cyclic subspace of* V *generated by nonzero vector* $v \in \mathrm{V}$. *Let* $k = \dim(\mathrm{W})$. *Then:*

1. $\{v, \mathrm{T}(v), \mathrm{T}^2(v), \dots, \mathrm{T}^{k-1}(v)\}$ *is a basis for* W.
2. *If* $a_0 v + a_1 \mathrm{T}(v) + a_2 \mathrm{T}^2(v) + \dots + a_{k-1}\mathrm{T}^{k-1}(v) + \mathrm{T}^k(v) = 0$*, then the characteristic polynomial of* T_W *is* $f(t) = (-1)^k \left(a_0 + a_1 t + \dots + a_{k-1} t^{k-1} + t^k\right)$.

Proof. Let $\beta = \{v, \mathrm{T}(v), \mathrm{T}^2(v), \dots, \mathrm{T}^{k-1}(v)\}$, and let a_i be the scalars that

$$a_0 v + a_1 \mathrm{T}(v) + a_2 \mathrm{T}^2(v) + \dots + a_{k-1}\mathrm{T}^{k-1}(v) + \mathrm{T}^k(v) = 0$$

Fors basis $\{v, \mathrm{T}(v), \mathrm{T}^2(v), \dots, \mathrm{T}^{k-1}(v)\}$, $\left[\mathrm{T}(v)\right]_\beta = [0, 1, \dots, 0]$, $\mathrm{T}\left(\mathrm{T}(v)\right)_\beta = [0, 0, 1, \dots, 0]$, etc, we have:

$$[\mathrm{T}_\mathrm{W}]_\beta = \begin{bmatrix} 0 & 0 & \dots & 0 & -a_0 \\ 1 & 0 & \dots & 0 & -a_1 \\ \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & \dots & 1 & -a_{k-1} \end{bmatrix}$$

which has characteristic polynomial

$$f(t) = (-1)^k (a_0 + a_1 t + \cdots + a_{k-1} t^{k-1} + t^k)$$

□

Theorem 98 (Cayley-Hamilton). *Let* T *be linear operator on a finite-dimensional vector space* V*, and let* $f(t)$ *be the characteristic polynomial of* T*. Then* $f(\mathrm{T}) = 0$.

Proof. Suppose $v \neq 0$. Let W be the T-cyclic subspace generated by v, and suppose the $\dim(\mathrm{W}) = k$. So there exists scalars $\{a_i\}$ that

$$a_0 v + a_1 \mathrm{T}(v) + a_2 \mathrm{T}^2(v) + \cdots + a_{k-1}\mathrm{T}^{k-1}(v) + \mathrm{T}^k(v) = 0$$

which implies the characteristic polynomial of $\mathrm{T_W}$ is

$$g(t) = (-1)^k \left(a_0 + a_1 t + \cdots + a_{k-1} t^{k-1} + t^k\right)$$

We have

$$g(\mathrm{T})(v) = (-1)^k \left(a_0 \mathrm{I} + a_1 \mathrm{T} + \cdots + a_{k-1}\mathrm{T}^{k-1} + \mathrm{T}^k\right)(v) = 0$$

Because $g(t)$ divides $f(t)$, $\exists q(t)$ that $f(t) = g(t)q(t)$. So

$$f(\mathrm{T})(v) = q(\mathrm{T})g(\mathrm{T})(v) = q(\mathrm{T})\left(g(\mathrm{T})(v)\right) = q(\mathrm{T})(0) = 0$$

□

Definition 86. *Let* $\mathrm{B_1} \in \mathrm{M}_{m\times m}(\mathrm{F})$*, and* $\mathrm{B_2} \in \mathrm{M}_{n\times n}(\mathrm{F})$*. The* direct sum *of* $\mathrm{B_1}$ *and* $\mathrm{B_2}$*, denoted as* $\mathrm{B_1} \oplus \mathrm{B_2}$*, as the* $(m+n)\times(m+n)$ *matrix* A *that*

$$\mathrm{A} = \begin{bmatrix} \mathrm{B_1} & 0 \\ 0 & \mathrm{B_2} \end{bmatrix}$$

Theorem 99. *Suppose* $\mathrm{V} = \mathrm{W_1} \oplus \mathrm{W_2} \oplus \cdots \oplus \mathrm{W}_k$*, where* W_i *is a* T*-invariant subspace of* V*. Suppose* $f_i(t)$ *is the characteristic polynomial of* $\mathrm{T}_{\mathrm{W}_i}$*, Then* $\prod_{i=1}^{k} f_i$ *is the characteristic polynomial of* T*. Let* β_i *be an ordered basis for* W_i*, and let* $\beta = \bigcup_{i=1}^{k} \beta_i$*. Let* $\mathrm{A} = [\mathrm{T}]_\beta$*, and* $\mathrm{B}_i = [\mathrm{T}_{\mathrm{W}_i}]_\beta$*. Then* $\mathrm{A} = \mathrm{B_1} \oplus \mathrm{B_2} \oplus \cdots \oplus \mathrm{B}_k$.

2.5.4 Limit of Markov Chain Matrix

Definition 87. *A sequence* $\{\mathrm{A_1}, \mathrm{A_2}, \dots\}$ converge *to* limit L *if* $\lim_{m\to\infty} (\mathrm{A}_m)_{ij} = \mathrm{L}_{ij}$.

Theorem 100. *If* $\mathrm{A}_i \to \mathrm{L}$*, them for any* P *and* Q*,* $\lim_{m\to\infty} \mathrm{PA}_m = \mathrm{PL}$ *and* $\lim_{m\to\infty} \mathrm{A}_m\mathrm{Q} = \mathrm{LQ}$.

Theorem 101. *Let* Q *be invertible and* $\mathrm{A}_i \to \mathrm{L}$*. Then* $\lim_{m\to\infty} (\mathrm{QAQ}^{-1})^m = \mathrm{QAQ}^{-1}$.

Definition 88. *Define a set* S *which consists of the interior of unit disk and* 1*:*

$$\mathrm{S} = \{\lambda \in \mathrm{C} : |\lambda| < 1 \vee \lambda = 1\} \tag{2.43}$$

Theorem 102. *Let* A *be square matrix in* C*.* $\lim_{m\to\infty} \mathrm{A}^m$ *exists if and only if:*

1. *Every eigenvalue of* A *is in* S*.*
2. *If* 1 *is an eigenvalue of* A*, then the dimension of its eigenspace equals its multiplicity.*

Proof. use Jordan canonical form. □

Theorem 103. *For square matrix* A *in* C*, if*

1. *Every eigenvalue of* A *is in* S*.*
2. A *is diagonalizable.*

Then $\lim_{m\to\infty} \mathrm{A}^m$ *exists.*

Proof. Since A is diagonalizable, $\exists \mathrm{Q} : \mathrm{A} = \mathrm{QDQ}^{-1}$. So $\mathrm{A}^m = \mathrm{QD}^m\mathrm{Q}^{-1}$. This is used to calculate A^m. □

Definition 89. transition matrix *or* stochastic matrix *is a square matrix* A *that* $\mathrm{A}_{ij} \geq 0 \wedge \forall j \left(\sum_i \mathrm{A}_{ij} = 1\right)$.

Definition 90. P *is a* probability vector *if its entries are all non-negative and sum to* 1.

Definition 91. $\underline{\vec{1}_n}$ *is a column vector that each coordinate is* 1.

Theorem 104. *Let* M *be a square matrix with non-negative real entries, and* v *a column vector with real non-negative coordinates. Then*

1. M *is a transition matrix if and only if* $M^\top \vec{1}_n = \vec{1}_n$.
2. v *is a probability vector if and only if* $\vec{1}_n^{\top} v = 1$.
3. *The product of two transition matrix is transition matrix.*
4. *The product of a transition matrix and probability vector is a probability vector.*

Definition 92. *A transition matrix is* $\underline{\text{regular}}$ *if some power of the matrix contains only positive entries. It may contain zero entries.*

Definition 93. *For square matrix* A*, define* $\rho_i(A) = \sum_j \left|A_{ij}\right|$ *and* $v_j(A) = \sum_i \left|A_{ij}\right|$*. The* $\underline{\text{row sum}}$ $\rho(A) = \max \rho_i$ *and* $\underline{\text{column sum}}$ $v(A) = \max v_j$.

Definition 94. *For square matrix* $A_{n\times n}$*, the* $\underline{\text{Gerschgorin disk}}$ C_i *is defined as:*

$$C_i = \left\{z \in C : \left|z - A_{ii}\right| < \rho_i(A) - \left|A_{ii}\right|\right\} \tag{2.44}$$

So the disk center is the diagonal entry, and the radius is the sum of the absolute values of all rest row entries.

Theorem 105. *Every eigenvalue of* A *is contained in a Gerschgorin disk.*

Proof. Let λ be a eigenvalue with eigenvector v. So $\sum_{j=1}^{n} A_{ij} v_j = \lambda v_i$. Assume v_k is the coordinate of v that has the largest absolute value. Then $v_k \neq 0$ because $v \neq 0$. We have

$$\left|\lambda v_k - A_{kk} v_k\right| = \left|\sum_{j=1}^{n} A_k j v_j - A_{kk} v_k\right| = \left|\sum_{j\neq k} A_{kj} v_j\right| \leq \sum_{j\neq k} \left|A_{kj}\right|\left|v_j\right| \leq \sum_{j\neq k}\left|A_{kj}\right|\left|v_k\right| = \left|v_k\right|\left(\rho_i(A) - \left|A_{kk}\right|\right)$$

So $\left|v_k\right| \times \left|\lambda - A_{kk}\right| \leq \left|v_k\right|\left(\rho_i(A) - \left|A_{kk}\right|\right)$ and $\left|\lambda - A_{kk}\right| \leq \left(\rho_i(A) - \left|A_{kk}\right|\right)$. □

Theorem 106. *Let* λ *be any eigenvalue of* A*. Then* $|\lambda| \leq \rho(A)$.

Proof. $|\lambda| = \left|(\lambda - A_{kk}) + A_{kk}\right| \leq \left|\lambda - A_{kk}\right| + \left|A_{kk}\right| \leq \rho_i(A) - \left|A_{kk}\right| + \left|A_{kk}\right| = \rho_i(A)$ □

Theorem 107. *Let* λ *be any eigenvalue of* A*. Then* $|\lambda| \leq \min\left\{\rho(A), v(A)\right\}$.

Proof. λ is an eigenvalue of $A^\top$. □

Theorem 108. *If* λ *is an eigenvalue of transition matrix, then* $|\lambda| \leq 1$.

Theorem 109. *Every transition matrix has* 1 *as eigenvalue.*

Proof. $A^\top \times \vec{1}_n = \vec{1}_n$. □

Theorem 110. *Let* A *be a matrix with positive entries, and let* λ *be an eigenvalue of* A *that* $|\lambda| = \rho(A)$*. Then* $\lambda = \rho(A)$ *and* $\vec{1}_n$ *is a basis for* E_λ.

Proof. Let v be an eigenvector for λ, and v_k is the coordinate that has the largest absolute value $b = \left|v_k\right|$. Then

$$|\lambda|\, b = \left|\lambda v_k\right| = \left|\sum_{j=1}^{n} A_{kj} v_j\right| \leq \sum_{j=1}^{n}\left|A_{kj} v_j\right| = \sum_{j=1}^{n}\left|A_{kj}\right|\left|v_j\right| \leq \sum_{j=1}^{n}\left|A_{kj}\right| b = \rho_k(A) b \leq \rho(A) b$$

Since $|\lambda| = \rho(A)$, all inequalities are equalities, so

1. $\left|\sum_{j=1}^{n} A_{kj} v_j\right| = \sum_{j=1}^{n}\left|A_{kj} v_j\right|$
2. $\left|A_{kj}\right|\left|v_j\right| = \sum_{j=1}^{n}\left|A_{kj}\right| b$
3. $\rho_k(A) \leq \rho(A)$

For Item 1 to hold, $A_{kj}v_j$ are non-negative multiples of a common complex number z. Assume $|z| = 1$. Then $\left(\exists \left\{c_j\right\} \subset \mathbb{R}^+\right)(A_{kj}v_j = c_j z)$.

For item 2, since $b = \max\left|v_j\right|$, $\left|v_j\right| = b$. So $b = \left|v_j\right| = \left|\frac{c_j}{A_{kj}}z\right| = \frac{c_j}{A_{kj}}$, and $v_j = \frac{c_j}{A_{kj}}z = bz$, and $v = bz\vec{1}_n$.

Since A and $\vec{1}_n$ are all positive, $A\vec{1}_n = \lambda\vec{1}_n$, so $\lambda > 0$. □

Theorem 111. *Let* A *be a transition matrix that each entry is positive, and let* λ *be any eigenvalue of* A *other than* 1. *Then* $|\lambda| < 1$. *Moreover, the eigenspace of eigenvalue* 1 *has dimension* 1.

Theorem 112. *Let* A *be a regular transition matrix, and* λ *be one of its eigenvalue, then*

1. $|\lambda| \le 1$.
2. *If* $|\lambda| = 1$, *then* $\lambda = 1$ *and* $\dim\left(E_\lambda\right) = 1$.

Theorem 113. *Let* A *be a disagonalizable regular transition matrix, then* $\lim_{m\to\infty} A^m$ *exists.*

Theorem 114. *Let* A *be a regular transition matrix, then*

1. *the multiplicity of eigenvalue* 1 *is* 1.
2. $\lim_{m\to\infty} A^m$ *exists.*
3. $L = \lim_{m\to\infty} A^m$ *is a transition matrix.*
4. $AL = LA = L$.
5. *The column of* L *are identical vector* v *which is the probability vector in* E_1.
6. *For any probability vector* w, $\lim_{m\to\infty} A^m w = v$.

Proof. Since $AL = L$, L are columns of eigenvector for eigenvalue 1. Let $y = \lim_{m\to\infty} A^m w = Lw$, $Ay = ALw = Lw = y$. So y is an eigenvector for eigenvalue 1, and $y = v$. □

2.6 Inner Product Space

2.6.1 Inner Product and Norm

Definition 95. *An inner product on V is a function $V \to V \to F$ (F is either C or R) that $\forall x, y, z \in V$ and $\forall c \in F$ that:*

1. $\langle x+z, y\rangle = \langle x, y\rangle + \langle z, y\rangle$
2. $\langle cx, y\rangle = c\langle x, y\rangle$
3. $\overline{\langle x, y\rangle} = \langle y, x\rangle$
4. $\langle x, x\rangle > 0$ *if* $x \neq 0$

Item (1) and (2) means the inner product is linear in first component. *Please be noted that the result of inner product could be a complex value, but the result of $\langle x, x\rangle$ is a non-negative real number.* □

Theorem 115. *properties of inner product:*

1. $\langle x, y+z\rangle = \langle x, y\rangle + \langle x, z\rangle$
2. $\langle x, cy\rangle = \bar{c}\langle x, y\rangle$
3. $\langle x, x\rangle = 0$ *if and only if* $x = 0$.
4. *If* $\langle x, y\rangle = \langle x, z\rangle$ *for all* $x \in V$, *then* $y = z$.

Item (1) and (2) means the inner product is conjugate linear in second component.

Definition 96. *the standard inner product on F^n for $x = [a_1, a_2, \ldots, a_n]$ and $y = [b_1, b_2, \ldots, b_n]$ is:*

$$\langle x, y\rangle = \sum_{i=1}^{n} a_i \overline{b_i} \tag{2.45}$$

when $F = R$, it is usually called dot product and denoted as $x \cdot y$.

Definition 97. *For $A \in M_{m\times n}(F)$, the conjugate transpose or adjoint of A is $A^* \in M_{n\times m}(F)$ that $(A^*)_{ij} = \overline{A_{ji}}$. If A is complex, $A^* = \overline{A^\top}$. If A is real, A^* is $A^\top$.*

Definition 98 (Forbenius Inner Product)**.** *Let $V = M_{n\times n}(F)$, the Forbenius Inner Product is defined as:*

$$\langle A, B\rangle = \operatorname{tr}(B^* A) \tag{2.46}$$

Theorem 116. *For square matrix $A_{n\times n}$, we have*

$$\langle A, A\rangle = \sum_{i=1}^{n}\sum_{j=1}^{n} \left|A_{ij}\right|^2 \geq 0 \tag{2.47}$$

Definition 99. *The continuous complex-valued function on interval $[0, 2\pi]$ is a inner product space H:*

$$\langle f, g\rangle = \frac{1}{2\pi}\int_0^{2\pi} f(t)\overline{g(t)}dt \tag{2.48}$$

Definition 100. *the norm or length of x is:*

$$\|x\| = \sqrt{\langle x, x\rangle} \tag{2.49}$$

Theorem 117. *the property of norm:*

- $\|cx\| = |c| \cdot \|x\|$
- $\|x\| = 0 \iff x = 0$
- *Cauchy-Schwarz Inequality* $\left|\langle x, y\rangle\right| \leq \|x\| \cdot \|y\|$
- *Triangle Inequality* $\|x+y\| \leq \|x\| + \|y\|$

Theorem 118. *If $\forall x \in C, \langle T(x), x\rangle = 0$. Then $T = 0$.*[4]

Proof.

$$\begin{aligned}\langle T(x+y), x+y\rangle = \langle T(x), y\rangle + \langle T(y), x\rangle &= 0\\ \langle T(x+iy), x+iy\rangle = \langle T(x), y\rangle - \langle T(y), x\rangle &= 0\end{aligned}$$

So $\forall y \in V$, $T(x) = 0$. So $\forall x \in V, T(x) = 0$ and $T = 0$. □

Theorem 119.

$$\|u+v\|^2 + \|u-v\|^2 = 2\left(\|u\|^2 + \|v\|^2\right) \tag{2.50}$$

[4]For it to work in all V, T needs to be self-adjoint. See Theorem 155 on page 42.

2.6.2 Orthogonal and Gram-Schmidt Process

Definition 101. *x and y are orthogonal if $\langle x,y\rangle = 0$. A subset S of V is orthogonal if any two vectors in S are orthogonal. A subset S of V is orthonormal if S is orthogonal and consists entirely of unit vectors.*

Definition 102.

$$\langle x,y\rangle = \|x\|\cdot\|y\|\, cos(\theta) \tag{2.51}$$

Definition 103. *A vector is unit vector if $\|x\| = 1$. A normalizing to non-zero x is $\frac{1}{\|x\|}x$.*

Theorem 120. *Let $f_n(t) = e^{int}$ where $0 \le t \le 2\pi$. All f_i are orthogonal.*

Proof.

$$\begin{aligned}\langle f_m,f_n\rangle &= \frac{1}{2\pi}\int_0^{2\pi} e^{imt}\overline{e^{int}}\,\mathrm{d}t \\ &= \frac{1}{2\pi}\int_0^{2\pi} e^{i(m-n)t}\,\mathrm{d}t \\ &= \frac{1}{2\pi(m-n)}e^{i(m-n)t}\Big|_0^{2\pi} \\ &= 0\end{aligned} \tag{2.52}$$

□

Theorem 121 (Pythagorean Theorem)**.** *Suppose u and v are orthogonal in V, then*

$$\|u+v\|^2 = \|u\|^2 + \|v\|^2 \tag{2.53}$$

Theorem 122. *For a finite dimensional subspace U of V, we have*

$$V = U \oplus U^{\perp} \tag{2.54}$$

Definition 104. *A orthonormal basis for V is an ordered basis that is orthonormal.*

Theorem 123. *Let $S = \{v_1, v_2, \ldots, v_k\}$ be an orthogonal subset of V consisting of non-zero vectors. If $y \in \mathrm{span}(S)$, then*

$$y = \sum_{i=1}^{k} \frac{\langle y,v_i\rangle}{\|v_i\|^2} v_i \tag{2.55}$$

Define the projection of vector a onto vector u as $\mathrm{proj}_u a = \frac{\langle a,u\rangle}{\|u\|^2}$. So

$$y = \sum_{i=1}^{k} \left(\mathrm{proj}_{v_i} y\right) v_i \tag{2.56}$$

If S is orthonormal, then

$$y = \sum_{i=1}^{k} \langle y,v_i\rangle v_i \tag{2.57}$$

Proof. let $y = \sum_{i=1}^{k} a_i v_i$. we have

$$\langle y,v_i\rangle = \left\langle \sum_{i=1}^{k} a_i v_i, v_j \right\rangle = \sum_{i=1}^{k} a_i \langle v_i,v_j\rangle = a_j \left\|v_j\right\|^2$$

So $a_j = \frac{\langle y,v_j\rangle}{\left\|v_j\right\|^2}$.

□

Theorem 124. *An orthogonal subset of V is linearly independent.*

Definition 105 (Gram-Schmidt process). *Let $S = \left\{w_1, w_2, \dots, w_n\right\}$ be linearly independent subset of V. Define $S' = \{v_1, v_2, \dots, v_n\}$, where $v_1 = w_1$ and*

$$v_k = w_k - \sum_{j=1}^{k-1} \frac{\left\langle w_k, v_j \right\rangle}{\left\| v_j \right\|^2} v_j \tag{2.58}$$

then S' is an orthogonal set of non-zero vectors that $\operatorname{span}\left(S'\right) = \operatorname{span}(S)$. The process is that for the k-th basis w_k, first project it on top of the $k-1$ orthogonal vectors $\sum_{j=1}^{k-1} \frac{\left\langle w_k, v_j \right\rangle}{\left\| v_j \right\|^2} v_j$, and calculate the reciprocal vector $w_k - \sum_{j=1}^{k-1} \frac{\left\langle w_k, v_j \right\rangle}{\left\| v_j \right\|^2} v_j$. □

Theorem 125 (QR Decomposition). *Let $A_{m\times n} = \left[a_1, a_2, \dots, a_n\right]$ with $\operatorname{rank}(A) = n$, so $\left\{a_i\right\}$ is linearly independent. Use Gram-Schmidt process to form n orthonomal basis:*

$$\begin{aligned}
u_1 &= a_1 & , e_1 &= \frac{u_1}{\|u_1\|} \\
u_2 &= a_2 - \operatorname{proj}_{u_1} a_2 & , e_2 &= \frac{u_2}{\|u_2\|} \\
&\dots \\
u_n &= a_n - \sum_{j=1}^{n-1} \operatorname{proj}_{u_j} a_n & , e_n &= \frac{u_n}{\|u_n\|}
\end{aligned}$$

Then $\forall k$, $a_k = \sum_{j=1}^{k} \left\langle a_k, e_k \right\rangle e_k$. So

$$A = QR = [e_1, e_2, \dots, e_n] \times \begin{bmatrix} \left\langle a_1, e_1 \right\rangle & \left\langle a_2, e_1 \right\rangle & \left\langle a_3, e_1 \right\rangle & \cdots & \left\langle a_n, e_1 \right\rangle \\ 0 & \left\langle a_2, e_2 \right\rangle & \left\langle a_3, e_2 \right\rangle & \cdots & \left\langle a_n, e_2 \right\rangle \\ 0 & 0 & \left\langle a_3, e_3 \right\rangle & \cdots & \left\langle a_n, e_3 \right\rangle \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & \left\langle a_n, e_n \right\rangle \end{bmatrix} \tag{2.59}$$

The Q is an orthonormal matrix. R could be calculated by:

$$R = Q^\top QR = Q^\top A \tag{2.60}$$

Theorem 126. *If V has an orthonormal basis $\beta = \left\{v_1, v_2, \dots, v_n\right\}$, then $\forall x \in V$,*

$$x = \sum_{i=1}^{n} \left\langle x, v_j \right\rangle v_i \tag{2.61}$$

Definition 106. *Let β be an orthonormal subset (not basis) of V. For $x \in V$, the Fourier coefficients of x relative to β are $\left\langle x, y_i \right\rangle$ for all $y_i \in \beta$.*

Theorem 127. *Let V with an orthonormal basis $\beta = \left\{v_1, v_2, \dots, v_n\right\}$. T is a linear operator on V and let $A = [T]_\beta$. then $A_{ij} = \left\langle T(v_j), v_i \right\rangle$.*

Proof. From Theorem 126 we have

$$T(v_j) = \sum_{i=1}^{n} \left\langle T(v_j), v_i \right\rangle v_i$$

□

Definition 107. *Let S be nonempty subset of V. The orthogonal complement of S is $S^\perp$ that $\forall x \in S, \forall y \in S^\perp, \left\langle x, y \right\rangle = 0$.*

Theorem 128. *Let* W *be a subspace of* V*. For* $y \in \mathrm{V}$*, there is unique* $u \in \mathrm{W}$ *and* $z \in \mathrm{W}^{\perp}$ *that* $y = u + z$*.* u *is the* *orthogonal projection* *of* y *on* W*. If* $\{v_1, v_2, \ldots, v_k\}$ *is an orthonormal basis of* W*, then*

$$u = \sum_{i=1}^{k} \langle y, v_i \rangle v_i \\ z = y - \sum_{i=1}^{k} \langle y, v_i \rangle v_i \tag{2.62}$$

Theorem 129. *For* $\mathrm{S} = \{v_1, v_2, \ldots, v_k\}$ *be an orthogonal subset of* V*. For* $\forall y \in \mathrm{V}$*, the orthogonal projection of* y *on* S *is* $u = \sum_{i=1}^{k} \frac{\langle y, v_i \rangle}{\|v_i\|^2} v_i$*. If* S *are orthonormal,* $u = \sum_{i=1}^{k} \langle y, v_i \rangle v_i$*. If* y *is in span of* S*, then* $y = u$*.*

Theorem 130. *Let y,u,z as defined in Theorem 128. u is the closest vector in* W *to* y *that is* $\forall x \in \mathrm{W}\left(\|y - x\| \geq \|y - u\|\right)$*.*

Proof.

$$\|y - x\|^2 = \|u + z - x\|^2 = \|(u - x) + z\|^2 = \|u - x\|^2 + \|z\|^2 \geq \|z\|^2 = \|y - u\|^2$$

□

2.6.3 Adjoint of Linear Operator

Theorem 131 (Riesz Representation Theorem)**.** *Let* $g : \mathrm{V} \to \mathrm{F}$ *be a linear transformation. Then there exist a unique* $y \in \mathrm{V}$ *that* $\forall x \in \mathrm{V}$*,* $g(x) = \langle x, y \rangle$*. The* y *is*

$$y = \sum_{i=1}^{n} \overline{g(v_i)} v_i \tag{2.63}$$

So every vector in the dual space[5] *can be represented by an inner product.*

Proof. Define $h(x) = \langle x, y \rangle$ with y defined above. So

$$h(v_j) = \langle v_j, y \rangle = \left\langle v_j, \sum_{i=1}^{n} \overline{g(v_i)} v_i \right\rangle = \sum_{i=1}^{n} \langle v_j, \overline{g(v_i)} v_i \rangle = \sum_{i=1}^{n} g(v_i) \langle v_j, v_i \rangle = g(v_j)$$

□

Theorem 132. *Let* T *be a linear operator on* V*. Then there existing a unique linear operator* $\mathrm{T}^* : \mathrm{V} \to \mathrm{V}$ *that* $\langle \mathrm{T}(x), y \rangle = \langle x, \mathrm{T}^*(y) \rangle$ *for all* $x, y \in \mathrm{V}$*.* T^* *is called the* *adjoint* *of* T*.*

Proof. For each y, $\langle \mathrm{T}(x), y \rangle$ is a linear operator from V to F, so by Theorem 131, $\exists y'$ that $\langle \mathrm{T}(x), y \rangle = \langle x, y' \rangle$. Define T^* as $\mathrm{T}^*(y) = y'$. □

Theorem 133.

$$\langle \mathrm{T}(x), y \rangle = \langle x, \mathrm{T}^*(y) \rangle \\ \langle x, \mathrm{T}(y) \rangle = \langle \mathrm{T}^*(x), y \rangle \tag{2.64}$$

So * *is added to* T *when change the location of* T*.*

Proof.

$$\langle x, \mathrm{T}(y) \rangle = \overline{\langle \mathrm{T}(y), x \rangle} = \overline{\langle y, \mathrm{T}^*(x) \rangle} = \langle \mathrm{T}^*(x), y \rangle$$

□

Theorem 134. *Let* β *be a orthonormal basis for* V*. If* T *is a linear operation on* V *then*

$$[\mathrm{T}^*]_\beta = \left([\mathrm{T}]_\beta\right)^* \tag{2.65}$$

Let A *be an* $n \times n$ *matrix. Then*

$$\mathrm{L}_{\mathrm{A}^*} = \left(\mathrm{L}_\mathrm{A}\right)^* \tag{2.66}$$

[5]Defined in Theorem 63 on page 23.

Proof. Let $A = [T]_\beta$, $B = [T^*]_\beta$, and $\beta = \{v_1, v_2, \dots, v_n\}$. Then according to Theorem 127:

$$B_{ij} = \left\langle T^*(v_j), v_i \right\rangle = \overline{\left\langle v_i, T^*(v_j) \right\rangle} = \overline{\left\langle T(v_i), v_j \right\rangle} = \overline{A_{ji}} = (A^*)_{ij}$$

□

Theorem 135. *Let T and U be linear operator on V, then*

1. $(aT + bU)^* = \overline{a}T^* + \overline{b}U^*$
2. $(UT)^* = T^*U^*$
3. $T^{**} = T$

Definition 108. *Let $T : V \to W$ be a linear transformation where V and W are finite dimensional inner product space with inner product $\langle \cdot, \cdot \rangle_V$ and $\langle \cdot, \cdot \rangle_W$. A function $T^* : W \to V$ is called adjoint of T if $\left\langle T(x), y \right\rangle_W = \left\langle x, T^*(y) \right\rangle_V$.*

Theorem 136. *Let T^* be an adjoint of $T : V \to W$. If β and γ are orthonormal basis for V and W, then*

$$[T^*]^\alpha_\beta = \left([T]^\alpha_\beta\right)^* \tag{2.67}$$

Theorem 137. *Let T^* be an adjoint of $T : V \to W$, we have:*

$$\left\langle T^*(x), y \right\rangle_V = \left\langle x, T(y) \right\rangle_W \tag{2.68}$$

Theorem 138. *If V is finite dimentional, let T be a linear operator on V, then*

$$\mathscr{R}(T^*)^\perp = \mathscr{N}(T)$$
$$\mathscr{R}(T^*) = \mathscr{N}(T)^\perp$$
$$\mathscr{R}(T)^\perp = \mathscr{N}(T^*)$$
$$\mathscr{R}(T) = \mathscr{N}(T^*)^\perp$$

So $\mathscr{R}(T^*) \perp \mathscr{N}(T)$.

Proof. If $m \in R(T^*)^\perp$, $\forall x \in V$, $0 = \left\langle m, T^*x \right\rangle = \left\langle T(m), x \right\rangle$, so $m \in N(T)$. □

2.6.4 Examples in Statistics

The following two examples show that for linear equation $Ax - y = 0$,

1. if it is consistent, that is there is solution, we want to find the solution with minimal norm.
2. If it is inconsistent, that is no solution, we want a result that has the least norm.

The same topic is discussed in pseudo inverse.

2.6.4.1 Least Square Approximation

Definition 109. *The Least Square Approximation is a problem that for* $A = \begin{bmatrix} t_1 & 1 \\ t_2 & 1 \\ \vdots & \vdots \\ t_m & 1 \end{bmatrix}$, $y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix}$, *find* $x_0 = \begin{bmatrix} c \\ d \end{bmatrix}$ *that minimize* $\left\|Ax - y\right\|$.

Definition 110. *For $x, y \in F^n$, define $\left\langle x, y \right\rangle_n = y^* \times x$.*

Theorem 139. *Let $A \in M_{m\times n}(F)$, $x \in F^n$, $y \in F^m$, then*

$$\left\langle Ax, y \right\rangle_m = \left\langle x, A^*y \right\rangle_n \tag{2.69}$$

Proof. $\left\langle Ax, y \right\rangle_m = y^* \times (Ax) = (y^* \times A)x = (A^*y)^*x = \left\langle x, A^*y \right\rangle_n$ □

Theorem 140. *Let $A \in M_{m\times n}(F)$. Then*[6]

$$\text{rank}(A^*A) = \text{rank}(A) \tag{2.70}$$

So if $\text{rank}(A) = n$, A^*A is invertible.

[6] See Theorem 70 for another proof.

Proof. For equation $A^*Ax = 0$ and $Ax = 0$. $Ax = 0$ implies that $A^*Ax = 0$. Then assume $A^*Ax = 0$, then

$$0 = \langle 0, x\rangle_n = \left\langle A^*Ax, x\right\rangle_n = \left\langle Ax, A^{**}x\right\rangle_m = \langle Ax, Ax\rangle_m$$

□

Theorem 141. *Let $A \in M_{m\times n}(F)$, $y \in F^m$. Then there exists $x_0 \in F^n$ that $(A^*A)x_0 = A^*y$ and $\forall x \in F^n$, $\left\|Ax_0 - y\right\| \leq \left\|Ax - y\right\|$. If $\operatorname{rank}(A) = n$, then $x_0 = (A^*A)^{-1}A^*y$.*

Proof. Define $W = \mathscr{R}(L_A)$. There exists a x_0 that is closest to y that $Ax_0 - y \in W^\perp$, so $\left\langle Ax, Ax_0 - y\right\rangle_m = 0$. So $\left\langle x, A^*(Ax_0 - y)\right\rangle_n = 0$, so $A^*(Ax_0 - y) = 0$ and $(A^*A)x_0 = A^*y$. □

2.6.4.2 Minimal Solution to Linear Equations

Definition 111. *A solution s is minimal solution of $Ax = b$ if $\|s\| \leq \|u\|$ for any solution u.*

Theorem 142. *Let $A \in M_{m\times n}(F)$, $y \in F^m$. Suppose $Ax = y$ is consistent. Then there exists unique minimal solution $s \in R(L_{A^*})$ of $Ax = y$. And s is the only solution in $R(L_{A^*})$. If u is a solution to $(AA^*)u = y$, then $s = A^*u$.*

Proof. By Theorem 138 define $W = R(L_{A^*})$ and $W^\perp = N(L_A)$. $\forall x$ that $Ax = y$, we have $s \in W$ and $t \in W^\perp$ that $x = s + t$. So $y = Ax = A(s + t) = As + At = As$. So s is a solution to $Ax = y$. From Theorem 66, all solution to $Ax = y$ has the form $x' = s + t'$ where $t' \in W^\perp$. And $\left\|x'\right\|^2 = \left\|s + t'\right\|^2 = \|s\|^2 + \left\|t'\right\|^2 \geq \|s\|^2$. □

2.7 Operator

2.7.1 Normal

Theorem 143. *If* T *has eigenvector, then* T^* *has eigenvector.*

Proof. $0 = \langle 0, x\rangle = \left\langle (T - \lambda I)(v), x\right\rangle = \left\langle v, (T - \lambda I)^*(x)\right\rangle = \left\langle v, (T^* - \overline{\lambda} I)(x)\right\rangle$. Since $v \neq 0$ is reciprocal to the range of $T^* - \overline{\lambda}I$, $v \notin \mathscr{R}(T^* - \overline{\lambda}I)$, so $\mathscr{N}(T^* - \overline{\lambda}I) \neq \{0\}$. □

Theorem 144 (Schur)**.** *Suppose the characteristic polynomial of* T *splits. Then there exists an orthonormal basis* β *for* V *that the* $[T]_\beta$ *is upper trianglar. Note:*

1. β *does* not *need to be eigenvectors of* T.
2. *It works in* $\mathscr{R}$ *as long as* T *splits.*

Proof. Use induction. Since T splits, it has a eigenvector. By Theorem 143 T^* has eigenvector, and make it a unit eigenvector z. Let $W = \operatorname{span}\{z\}$. Then prove $W^\perp$ is T-invariant: for $\forall y \in W^\perp$ and $x = cz \in W$:

$$\left\langle T(y), x\right\rangle = \left\langle T(y), cz\right\rangle = \left\langle y, T^*(cz)\right\rangle = \left\langle y, cT^*(z)\right\rangle = \left\langle y, c\lambda z\right\rangle = \overline{c\lambda}\left\langle y, z\right\rangle = 0$$

According to induction, $\dim\left(W^\perp\right) = n-1$ and there exists an orthonormal basis γ that $\left[T_{W^\perp}\right]_\gamma$ is upper triangular. Take $\gamma \cup \{z\}$. □

Theorem 145. *If* β *is an orthonormal basis and* $[T]_\beta$ *is a diagonal matrix,* $[T^*]_\beta = \left([T]_\beta\right)^*$ *is also a diagonal matrix.*

Theorem 146. *If an operator* T *has orthogonal eigenvectors* β *that are basis of the inner product space, then* $[T]_\beta$ *is a diagonal matrix.*

Definition 112. T *is* <u>*normal*</u> *if* $TT^* = T^*T$. *A square matrix* A *is* <u>*normal*</u> *if* $AA^* = A^*A$.

Theorem 147. T *is normal if and only of* $[T]_\beta$ *is normal under orthonormal basis* β.

Theorem 148. *Properties of normal operator* T *on* V*:*

1. $\forall x \in V$, $\left\|T(x)\right\| = \left\|T^*(x)\right\|$
2. $\forall c \in F$, $T - cI$ *is normal.*
3. *If* x *is a eigenvector of eigenvalue* λ *for* T, $T^*(x) = \overline{\lambda}x$, *so* x *is also an eigenvector of eigenvalue* $\overline{\lambda}$ *for* T^*.
4. *If* x_1 *and* x_2 *are for eigenvalues* λ_1 *and* λ_2, $\left\langle x_1, x_2\right\rangle = 0$

Proof.

$$\left\|T(x)\right\|^2 = \left\langle T(x), T(x)\right\rangle = \left\langle T^*T(x), x\right\rangle = \left\langle TT^*(x), x\right\rangle = \left\langle T^*(x), T^*(x)\right\rangle = \left\|T^*(x)^2\right\|$$

$$0 = \left\|(T - \lambda I)(x)\right\| = \left\|(T - \lambda I)^*(x)\right\| = \left\|(T^* - \overline{\lambda} I)(x)\right\|$$

$$\lambda_1\left\langle x_1, x_2\right\rangle = \left\langle \lambda x_1, x_2\right\rangle = \left\langle T(x_1), x_2\right\rangle = \left\langle x_1, T^*(x_2)\right\rangle = \left\langle x_1, \overline{\lambda_2} x_2\right\rangle = \lambda_2\left\langle x_1, x_2\right\rangle$$

So $(\lambda_1 - \lambda_2)\left\langle x_1, x_2\right\rangle = 0$. Since $\lambda_1 \neq \lambda_2$, $\left\langle x_1, x_2\right\rangle = 0$ □

Theorem 149. *If* T *is normal,* $\mathscr{N}(T) = \mathscr{N}(T^*)$ *and* $\mathscr{R}(T) = \mathscr{R}(T^*)$*. So being normal will refine Theorem 138.*

Proof. If $x \in \mathscr{N}(T)$, $\left\|T(x)\right\| = \|T^*\| = 0$, so $T^*(x) = 0$ and $x \in \mathscr{N}(T^*)$. □

Theorem 150. *In* $\mathscr{C}$*, let* V *be finite dimensional inner product space.* T *is normal if and only if there exists an orthonormal basis for* V *consisting of eigenvectors of* T.

Proof. in C the polynomial always splits. According to Theorem 144 there exists a orthonormal basis $\beta = \{v_1, v_2, \ldots, v_n\}$ that $[T]_\beta = A$ is upper triangular. v_1 is an eigenvector because $T(v_1) = A_{1,1}v_1$. Assuming $v_1, v_2, \ldots, v_{k-1}$ are eigenvector of T, we prove that v_k is also an eigenvector of T. Because A is upper triangular,

$$T(v_k) = A_{1,k}v_1 + A_{2,k}v_2 + \cdots + A_{j,k}v_j + \cdots + A_{k,k}v_k$$

Because $\forall j < k$, $A_{j,k} = \left\langle T(v_k), v_j\right\rangle = \left\langle v_k, T^*(v_j)\right\rangle = \left\langle v_k, \overline{\lambda} v_j\right\rangle = \lambda_j\left\langle v_k, v_j\right\rangle = 0$, we have $T(v_k) = A_{k,k}v_k$, so v_k is an eigenvector of T.

btw, it does not work in infinite dimensional complex inner product space. □

2.7.2 Hermitian

Definition 113. *T is self-adjoint (Hermitian) if $\mathrm{T}=\mathrm{T}^*$, or $\mathrm{A}=\mathrm{A}^*$. For real matrix, it means A is symmetric.*

Theorem 151. *Let T be a linear operator on complex inner product space. Then T is self-adjoint if and only if $\forall x\in \mathrm{V}, \left\langle \mathrm{T}(x),x\right\rangle \in \mathscr{R}$.*

Proof. If T is self-adjoint, $\overline{\left\langle \mathrm{T}(x),x\right\rangle}=\left\langle x,\mathrm{T}(x)\right\rangle=\left\langle \mathrm{T}^*(x),x\right\rangle=\left\langle \mathrm{T}(x),x\right\rangle$. So $\left\langle \mathrm{T}(x),x\right\rangle\in\mathscr{R}$.

If $\left\langle \mathrm{T}(x),x\right\rangle\in\mathscr{R}$, $\left\langle \mathrm{T}(x),x\right\rangle=\overline{\left\langle \mathrm{T}(x),x\right\rangle}=\left\langle x,\mathrm{T}(x)\right\rangle=\left\langle \mathrm{T}^*(x),x\right\rangle$. So $\forall x\in\mathrm{V}$, $\left\langle (\mathrm{T}-\mathrm{T}^*)(x),x\right\rangle=0$. According to Theorem (118), $\mathrm{T}-\mathrm{T}^*=0$. □

Theorem 152. *Let T be a self-adjoint operator on finite dimensional inner product space V. Then:*

1. *every eigenvalue is real.*
2. *If V is a real inner product space, the characteristic polynomial for T splits.*

Proof. Because T is self-adjoint, T is also normal. So according to Theorem 148 if λ is an eigenvalue of T, $\overline{\lambda}$ is an eigenvalue of T^*. So:

$$\lambda x=\mathrm{T}(x)=\mathrm{T}^*(x)=\overline{\lambda}x$$

So $\lambda=\overline{\lambda}$, and λ is real.

For a orthonormal basis β, $\mathrm{A}=[\mathrm{T}]_\beta$ is self-adjoint because $\mathrm{A}^*=([\mathrm{T}]_\beta)^*=[\mathrm{T}^*]_\beta=[\mathrm{T}]_\beta=\mathrm{A}$. Define $\mathrm{L}_\mathrm{A}(x)=\mathrm{A}x$ in $\mathscr{C}^n$. Here we create a function in $\mathscr{C}^n$ from a function in $\mathscr{R}^n$. Let γ be the standard basis for $\mathscr{C}$ which is orthonormal. $[\mathrm{L}_\mathrm{A}]_\gamma=\mathrm{A}$ is self-adjoint, so L_A is self-adjoint in $\mathscr{C}^n$. The characteristic polynomial of L_A splits. Since L_A is self-adjoint, all eigenvalues are real, so the polynomial split in $\mathscr{R}$. But L_A, A and T has the same characteristic polynomial. □

Theorem 153. *Let T be a linear operator on finite dimensional real inner product space. T is self-adjoint if and only if there exists an orthonormal basis β for V consisting of eigenvectors of T.*

Proof. By Theorem 144 there exists orthonormal basis β for V that $\mathrm{A}=[\mathrm{T}]_\beta$ is upper triangular. Because $\mathrm{A}^*=([\mathrm{T}]_\beta)^*=[\mathrm{T}^*]_\beta=[\mathrm{T}]_\beta=\mathrm{A}$, A is diagonal matrix. □

Theorem 154. *For the orthonormal basis of eigenvector T problem we have:*

1. *If T splits, we have orthonormal basis that make T upper triangular in $\mathscr{R}$ or $\mathscr{C}$. This basis may not be eigenvectors, or T may not have eigenvectors.*
2. *T is complex normal.*
3. *T is real symmetric.*

Theorem 155. *Let T be self-adjoint operator. If $\forall x\in\mathrm{V},\left\langle \mathrm{T}(x),x\right\rangle=0$. Then $\mathrm{T}=0$.*[7]

Proof. Choose orthonormal basis β that consist of eigenvector of T. For $x\in\beta$, $\mathrm{T}(x)=\lambda x$. So

$$0=\left\langle x,\mathrm{T}(x)\right\rangle=\langle x,\lambda x\rangle=\overline{\lambda}\langle x,x\rangle$$

Hence $\overline{\lambda}=0$ and $\forall x\in\beta,\mathrm{T}(x)=0$. □

2.7.3 Positive Operator

Definition 114. *An operator T is called positive operator if T is self-adjoint and $\forall x\in\mathrm{V}$:*

$$\langle \mathrm{T}x,x\rangle\geq 0 \tag{2.71}$$

Definition 115. *An Operator R is called a square root of an operator T if*

$$\mathrm{R}^2=\mathrm{T} \tag{2.72}$$

Theorem 156. *All the following are equivalent:*

1. *T is positive.*
2. *T is self-adjoint and all eigenvalue of T are non-negative.*
3. *T has positive square root.*
4. *T has self-adjoint square root.*
5. *$\exists\mathrm{R}:\mathrm{T}=\mathrm{R}^*\mathrm{R}$*

[7] Self-adjoint is not needed of $\mathrm{V}=\mathscr{C}$. See Theorem 118 on page 35.

Proof. For 2, if T is positive, $0 \le \langle \mathrm{T}v, v\rangle = \langle \lambda v, v\rangle = \lambda \langle v, v\rangle$, so $\lambda \ge 0$.

For 3, if T is self-adjoint, by Theorem 153 there are orthonormal basis $\beta = \{v_i\}$ with eigenvalue λ_i. Define $\mathrm{R}(v_i) = \sqrt{\lambda_i} v_i$. Then $\forall v_i \in \beta, \mathrm{R}^2(v_i) = \mathrm{T}(v_i)$.

For 1, $\langle \mathrm{T}v, v\rangle = \left\langle \mathrm{R}^*\mathrm{R}v, v\right\rangle = \langle \mathrm{R}v, \mathrm{R}v\rangle \ge 0$. □

Theorem 157. *A positive operator has a unique positive square root.*

Definition 116. *If* T *is a positive operator,* $\sqrt{\mathrm{T}}$ *is its positive square root.*

2.7.4 Isometry

Definition 117. *Let* T *be a linear operator on finite dimensional inner product space* V *over* F*. If* $\forall x \in \mathrm{V}$, $\left\|\mathrm{T}(x)\right\| = \|x\|$, *we call* T *unitary operator if* $\mathrm{F} = \mathscr{C}$ *or orthogonal operator if* $\mathrm{F} = \mathscr{R}$*. Unitary and orthogonal are also called isometry.*

Definition 118. *A square matrix* A *is called unitary matrix if* $\mathrm{AA}^* = \mathrm{A}^*\mathrm{A} = \mathrm{I}$ *and orthogonal matrix if* $\mathrm{AA}^\top = \mathrm{A}^\top\mathrm{A} = \mathrm{I}$.

Theorem 158. *Let* T *be an linear operator. Then the following are equivalent:*

1. $\mathrm{TT}^* = \mathrm{T}^*\mathrm{T} = \mathrm{I}$.
2. $\left\langle \mathrm{T}(x), \mathrm{T}(y)\right\rangle = \left\langle x, y\right\rangle$.
3. *If* β *is an orthonormal basis for* V*. Then* $\mathrm{T}(\beta)$ *is an orthonormal basis.*
4. $\left\|\mathrm{T}(x)\right\| = \|x\|$.

So unitary or orthogonal operator preserve inner product and norm.

Proof. $\left\langle x, y\right\rangle = \left\langle \mathrm{T}^*\mathrm{T}x, y\right\rangle = \left\langle \mathrm{T}(x), \mathrm{T}(y)\right\rangle$.

If $\beta = \{v_1, v_2, \ldots, v_n\}$ is an orthonormal basis. $\left\langle \mathrm{T}(v_i), \mathrm{T}(v_j)\right\rangle = \left\langle v_i, v_j\right\rangle = 0$.

If β and $\mathrm{T}(\beta)$ are both orthonormal basis, expand $\left\|\mathrm{T}(x)\right\|$ and $\|x\|$ to prove they are equal.

$\langle x, x\rangle = \|x\|^2 = \left\|\mathrm{T}(x)\right\|^2 = \left\langle \mathrm{T}(x), \mathrm{T}(x)\right\rangle = \left\langle x, \mathrm{T}^*\mathrm{T}x\right\rangle$. So $\forall x \in \mathrm{V}, \left\langle x, (\mathrm{I} - \mathrm{T}^*\mathrm{T})(x)\right\rangle = 0$. $\mathrm{I} - \mathrm{T}^*\mathrm{T}$ is normal, so according to Theorem 155, $\mathrm{I} - \mathrm{T}^*\mathrm{T} = 0$. □

Theorem 159. *Unitary operator is normal.*

Proof. See Theorem 158 property (1). □

Theorem 160. *Let* T *be a linear operator on* real *inner product space* V*.* V *has an orthonormal basis of eigenvectors of* T *with absolute value of all eigenvalues equal to* 1 *if and only if* T *is self-adjoint and orthogonal.*

Proof. If T is self-adjoint, there is orthonormal basis β of eigenvectors. If T is orthogonal, $\forall v_i \in \beta, \left|\lambda_i\right| \times \left\|v_i\right\| = \left\|\lambda_i v_i\right\| = \left\|\mathrm{T}(v_i)\right\| = \left\|v_i\right\|$, so $\left|\lambda_i\right| = 1$.

If V has orthonormal basis β of eigenvectors, T is self-adjoint. $\forall v_i \in \beta$, we have $\mathrm{TT}^*(v_i) = \mathrm{T}(\lambda_i v_i) = \lambda_i \mathrm{T}(v_i) = \lambda_i^2 v_i$. If $\left|\lambda_i\right| = 1$, $\mathrm{TT}^* = \mathrm{I}$. □

Theorem 161. *Let* T *be a linear operator on* complex *inner product space* V*.* V *has an orthonormal basis of eigenvectors of* T *with absolute value of all eigenvalues equal to* 1 *if and only if* T *is unitary.*

Proof. If T is unitary, it is normal, so there is orthonormal basis β of eigenvectors. If T is unitary, $\forall v_i \in \beta$, $\left|\lambda_i\right| \times \left\|v_i\right\| = \left\|\lambda_i v_i\right\| = \left\|\mathrm{T}(v_i)\right\| = \left\|v_i\right\|$, so $\left|\lambda_i\right| = 1$.

If V has orthonormal basis β of eigenvectors, T is normal. If $\left|\lambda_i\right| = 1$, $\forall v_i \in \beta, \left|\lambda_i\right| \times \left\|v_i\right\| = \left\|\lambda_i v_i\right\| = \left\|\mathrm{T}(v_i)\right\| = \left\|v_i\right\|$, so $\left\|\mathrm{T}(v_i)\right\| = \left\|v_i\right\|$ and it is unitary. □

Theorem 162. T *is isometry if* $[\mathrm{T}]_\beta$ *is isometry for a orthonormal basis* β *of* V.

Definition 119. A *is unitarily equivalent or orthogonally equivalent to* D *if and only if there exists a unitary or orthogonal matrix* P *that* $\mathrm{A} = \mathrm{P}^*\mathrm{DP}$.

Theorem 163. *Let* A *be a complex square matrix.* A *is normal if and only if it is unitarily equivalent to a diagonal matrix.*

Theorem 164. *Let* A *be a real square matrix.* A *is symmetric if and only if it is orthogonally equivalent to a diagonal matrix.*

2.7.5 Rigid motion

Definition 120. *Let* V *be real inner product space.* $f : \mathrm{V} \to \mathrm{V}$ *is a* <u>*rigid motion*</u> *if*

$$\left\|f(x) - f(y)\right\| = \left\|x - y\right\| \tag{2.73}$$

Definition 121. *Let* V *be real inner product space.* $g : \mathrm{V} \to \mathrm{V}$ *is a* <u>*translation*</u> *by* $v_0 \in \mathrm{V}$ *if*

$$\exists v_0 \forall x \in \mathrm{V}\big(g(x) = x + v_0\big) \tag{2.74}$$

Theorem 165. *A translation is a rigid motion. And a composite of rigid motion is rigid motion.*

Theorem 166. *Let* f *be a rigid motion. Then there exists a unique orthogonal operator* T *and unique translation* g *that* $f = g \circ \mathrm{T}$.

Proof. Define $\mathrm{T}(x) = f(x) - f(0)$. T is a composite of rigid motion, so it is a rigid motion. Therefore $\left\|\mathrm{T}(x)\right\| = \left\|f(x) - f(0)\right\| = \|x - 0\| = \|x\|$. Since

$$\begin{aligned}
\left\|\mathrm{T}(x) - \mathrm{T}(y)\right\|^2 &= \|x\|^2 - 2\big\langle \mathrm{T}(x), \mathrm{T}(y)\big\rangle + \left\|y\right\|^2 \\
\left\|x - y\right\|^2 &= \|x\|^2 - 2\big\langle x, y\big\rangle + \left\|y\right\|^2 \\
\left\|\mathrm{T}(x) - \mathrm{T}(y)\right\|^2 &= \left\|x - y\right\|^2
\end{aligned}$$

We have $\big\langle \mathrm{T}(x), \mathrm{T}(y)\big\rangle = \big\langle x, y\big\rangle$.

Then $\left\|\mathrm{T}(ax + y) - a\mathrm{T}(x) - \mathrm{T}(y)\right\|^2 = 0$ after expansion, T is linear. So T is an orthogonal operator. So we have unique T and g that

$$\begin{aligned}
\mathrm{T}(x) &= f(x) & -f(0) \\
g(x) &= x & +f(0)
\end{aligned} \tag{2.75}$$

□

Theorem 167. *Let* T *be an orthogonal operator on* R^2*, and let* $\mathrm{A} = [\mathrm{T}]_\beta$ *where* β *is the standard basis of* R^2*. Then one of the following is satisfied:*

1. T *is a rotation, so* $|\mathrm{T}| = 1$.
2. T *is a reflection about a line through the origin, so* $|\mathrm{T}| = -1$.

Proof. Because T is orthogonal, $\mathrm{T}(\beta) = \big\{\mathrm{T}(e_1), \mathrm{T}(e_2)\big\}$ is an orthonormal basis of R^1. Since $\mathrm{T}(e_1)$ is an unit vector, it has the form $\mathrm{T}(e_1) = (\cos\theta, \sin\theta)$. Since $\mathrm{T}(e_2)$ is orthogonal to $\mathrm{T}(e_1)$, it has the form $\mathrm{T}(e_2) = (-\sin\theta, \cos\theta)$ or $\mathrm{T}(e_2) = (\sin\theta, -\cos\theta)$. □

Theorem 168. *For expression* $f(x, y) = ax^2 + 2bxy + cy^2$*, let* $\mathrm{A} = \begin{pmatrix} a & b \\ b & c \end{pmatrix}$ *and* $\mathrm{X} = \begin{pmatrix} x \\ y \end{pmatrix}$*, the formula is* $f(\mathrm{X}) = \mathrm{X}^\top\mathrm{A}\mathrm{X} = \langle \mathrm{A}\mathrm{X}, \mathrm{X}\rangle$*. Since* A *is symmetric, there is an orthogonal matrix* P *and diagonal matrix* D *that* $\mathrm{A} = \mathrm{P}^\top\mathrm{D}\mathrm{P}$*. Define* $\mathrm{X}_0 = \begin{pmatrix} x_0 \\ y_0 \end{pmatrix}$ *that* $\mathrm{X} = \mathrm{P}\mathrm{X}_0$*. We have* $f(\mathrm{X}) = \mathrm{X}^\top\mathrm{A}\mathrm{X} = (\mathrm{P}\mathrm{X}_0)^\top\mathrm{A}(\mathrm{P}\mathrm{X}_0) = \mathrm{X}_0^\top\mathrm{D}\mathrm{X}_0 = \lambda_1 x_1^2 + \lambda_2 x_2^2$*. So the* xy *term could be removed by rotation.*

2.7.6 Spectral Theorem

Definition 122. *Let* $\mathrm{V} = \mathrm{W}_1 \oplus \mathrm{W}_2$*.* T *is a* <u>*projection*</u> *on* W_1 *along* W_2 *if* $\forall x = x_1 + x_2$ *that* $x_1 \in \mathrm{W}_1$ *and* $x_2 \in \mathrm{W}_2$*,* $\mathrm{T}(x) = x_1$.

Theorem 169. T *is a projection if and only if* $\mathrm{T}^2 = \mathrm{T}$.

Definition 123. T *is an* <u>*orthogonal projection*</u> *if* $\mathscr{R}(\mathrm{T})^\perp = \mathscr{N}(\mathrm{T})$ *and* $\mathscr{R}(\mathrm{T}) = \mathscr{N}(\mathrm{T})^\perp$[8].

Theorem 170. T *is an orthogonal projection if and only if* T *has an adjoint* T^* *that* $\mathrm{T}^2 = \mathrm{T} = \mathrm{T}^*$.

Proof. $\mathrm{T}^2 = \mathrm{T}$ because T is a projection. Let $x = x_1 + x + 2$ and $y = y_1 + y_2$ where $x_1, y_1 \in \mathscr{R}(\mathrm{T})$ and $x_2, y_2 \in \mathscr{N}(\mathrm{T})$. So

$$\begin{aligned}
\big\langle x, \mathrm{T}(y)\big\rangle &= \big\langle x_1 + x_2, y_1\big\rangle = \big\langle x_1, y_1\big\rangle \\
\big\langle \mathrm{T}(x), y\big\rangle &= \big\langle x_1, y_1 + y_2\big\rangle = \big\langle x_1, y_1\big\rangle
\end{aligned}$$

So $\mathrm{T} = \mathrm{T}^*$ and $\mathrm{T}^2 = \mathrm{T} = \mathrm{T}^*$.

For the reverse side, prove that $\mathscr{R}(\mathrm{T})^\perp = \mathscr{N}(\mathrm{T})$ and $\mathscr{R}(\mathrm{T}) = \mathscr{N}(\mathrm{T})^\perp$. □

[8] In finite dimensional space V, $\mathscr{R}(\mathrm{T})^\perp = \mathscr{N}(\mathrm{T}) \longleftrightarrow \mathscr{R}(\mathrm{T}) = \mathscr{N}(\mathrm{T})^\perp$

Theorem 171 (Spectral Theorem)**.** *Let* T *be real symmetric or complex normal with distinct eigenvalue* λ_i *and its corresponding eigenspace* W_i*. Let* T_i *be the orthogonal projection on* W_i*. We have:*

1. $\mathrm{T}_i\mathrm{T}_j = \delta_{ij}\mathrm{T}_i$
2. $\mathrm{I} = \sum_{i=1}^{k} \mathrm{T}_i$
3. $\mathrm{T} = \sum_{i=1}^{k} \lambda_i \mathrm{T}_i$

λ_i *is the* spectrum *of* T*.* I *is the resolution of the identity operator induced by T.* $\mathrm{T} = \sum_{i=1}^{k} \lambda_i \mathrm{T}_i$ *is the* spectral decomposition *of* T*.*

Proof. Let $x = \sum_{i=1}^{k} x_i$ where $x_i \in \mathrm{W}_i$. Then

$$\mathrm{T}(x) = \sum_{i=1}^{k} \mathrm{T}(x_i) = \sum_{i=1}^{k} \lambda_i x_i = \sum_{i=1}^{k} \lambda_i \mathrm{T}_i(x_i) = \sum_{i=1}^{k} \lambda_i \mathrm{T}_i(x) = \left(\sum_{i=1}^{k} \lambda_i \mathrm{T}_i\right) x$$

□

Theorem 172. *Let* $\mathrm{F} = \mathscr{C}$*.* T *is normal if and only if* $\exists g \in \mathrm{P}$*,* $\mathrm{T}^* = g(\mathrm{T})$*.*

Proof. Let $\mathrm{T} = \sum_{i=1}^{k} \lambda_i \mathrm{T}_i$ be the spectral decomposition of T. Take the adjoint of both side and we have

$$\mathrm{T}^* = \sum_{i=1}^{k} \overline{\lambda_i} \mathrm{T}_i^* \tag{2.76}$$

According to Lagrange formula[9] , $\exists g$, $g(\lambda_i) = \overline{\lambda_i}$. So $g(\mathrm{T}) = \mathrm{T}^*$. The reverse is easy to prove. □

Theorem 173. *Let* $\mathrm{F} = \mathscr{C}$*.* T *is unitary if and only if* T *is normal and* $|\lambda| = 1$ *for all eigenvalue* λ *of* T*.*

Proof. Let $\mathrm{T} = \sum_{i=1}^{k} \lambda_i \mathrm{T}_i$ be the spectral decomposition of T. We have

$$\mathrm{T}\mathrm{T}^* = \left(\sum_{i=1}^{k} \lambda_i \mathrm{T}_i\right) \times \left(\sum_{i=1}^{k} \overline{\lambda_i} \mathrm{T}_i\right) = \sum_{i=1}^{k} \left|\lambda_i\right|^2 \mathrm{T}_i^2 = \sum_{i=1}^{k} \left|\lambda_i\right|^2 \mathrm{T}_i = \sum_{i=1}^{k} \mathrm{T}_i = \mathrm{I}$$

□

Theorem 174. *Let* $\mathrm{F} = \mathscr{C}$ *and* T *normal.* T *is self-adjoint if and only if every eigenvalue of* T *is real.*

Proof. $\mathrm{T}^* = \sum_{i=1}^{k} \overline{\lambda_i} \mathrm{T}_i = \sum_{i=1}^{k} \lambda_i \mathrm{T}_i = \mathrm{T}$, so $\overline{\lambda_i} = \lambda_i$. □

2.7.7 Single Value Decomposition

Theorem 175. *Let* $\mathrm{T} : \mathrm{V} \to \mathrm{W}$ *be a linear transformation with rank* r*. Then there exists orthonormal basis* $\beta = \{v_1, v_2, \ldots, v_n\}$ *for* V *and* $\gamma = \{u_1, u_2, \ldots, u_m\}$ *for* W *and positive scalars* singular values $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r$ *such that*

$$\mathrm{T}(v_i) = \begin{cases} \sigma_i u_i & \text{if } 1 \leq i \leq r \\ 0 & \text{if } i > r \end{cases} \tag{2.77}$$

Conversely, for $1 \leq i \leq n$*,* v_i *is an eigenvector of* $\mathrm{T}^*\mathrm{T}$ *with corresponding eigenvalue* σ_i^2 *if* $1 \leq i \leq r$ *and* 0 *if* $i > r$*.*

[9] Theorem (35) on page 17.

Proof. T^*T has rank r according to Theorem 70, and positive semidefinite by Theorem 156. So there is an orthonormal basis v_i for V consisting of eigenvectors of T^*T with corresponding eigenvalues λ_i where $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_r > 0$ and $\lambda_i = 0$ for $i > r$. For $1 \leq i \leq r$, define $\sigma_i = \sqrt{\lambda_i}$ and $u_i = \frac{1}{\sigma_i}T(v_i)$. We have:

$$\left\langle u_i, u_j \right\rangle = \left\langle \frac{1}{\sigma_i}T(v_i), \frac{1}{\sigma_j}T(v_j) \right\rangle = \frac{1}{\sigma_i\sigma_j}\left\langle T^*T(v_i), v_j \right\rangle = \frac{1}{\sigma_i\sigma_j}\left\langle \lambda_i v_i, v_j \right\rangle = \frac{\sigma_i^2}{\sigma_i\sigma_j}\left\langle v_i, v_j \right\rangle = \delta_{ij}$$

So $\{u_1, u_2, \ldots, u_r\}$ are orthogonal. Because the choice of $\sqrt{\lambda_i}$, they are unitary and therefore orthonormal. Extend it to an orthonormal basis $\{u_1, u_2, \ldots, u_m\}$. □

Definition 124. *The singular values of* A *is the singular value of* L_A.

Theorem 176 (Singular Value Decomposition Theorem)**.** *Let* $A_{m\times n}$ *be of rank r with positive singular values* $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r$, *and let* $\Sigma_{m\times n}$ *be*

$$\Sigma_{ij} = \begin{cases} \sigma_i & \text{if } i = j \leq r \\ 0 \end{cases} \tag{2.78}$$

Then there exists singular value decomposition that with $U_{m\times m}$ *and* $V_{n\times n}$, *we have*

$$A = U\Sigma V^* \tag{2.79}$$

The process to find singular value decomposition is:

1. *find singular value of* A *by calculating the eigenvalue of* A^*A.
2. *sort the singular value from big to small.*
3. *for non-zero singular value* σ_i, *put* $\sqrt{\sigma_i}$ *to the i-th diagonal of* Σ.
4. *form* U *of normalized eigenvector of* A^*A.
5. *for non-zero singular value* σ_i, *calculate orthonormal vector* $u_i = \frac{1}{\sigma_i}L_A(v_i)$.
6. *expand the* u_i *to orthonormal basis and form* V.

2.7.8 Polar Decomposition

Theorem 177 (Polar Decomposition)**.** *Any square matrix* A, *there exists a Polar Decomposition using unitary matrix* W *and a positive semidefinite matrix* P *that*

$$A = WP \tag{2.80}$$

If A *is invertible, the Polar Decomposition is unique.*

Proof. Use singular value decomposition on A and we get $A = U\Sigma V^* = UV^*V\Sigma V^* = (UV^*)(V\Sigma V^*) = WP$. So let $W = UV^*$ and $P = V\Sigma V^*$. □

2.7.9 Pseudoinverse

Definition 125. *Let* $T : V \to W$ *be a linear transformation. Let* $L : \mathscr{N}(T)^\perp \to \mathscr{R}(T)$ *be a linear transformation that* $\forall x \in \mathscr{N}(T)^\perp$, $L(x) = T(x)$. *The pseudoinverse (or Moore-Penrose generalised inverse) of* T *is a unique linear transformation from* W *to* V *that*

$$T^\dagger(y) = \begin{cases} L^{-1}(y) & \text{for } y \in \mathscr{R}(T) \\ 0 & \text{for } y \in \mathscr{R}(T)^\perp \end{cases} \tag{2.81}$$

Let $\{v_1, v_2, \ldots, v_r\}$ *be a basis for* $\mathscr{N}(T)^\perp$, $\{v_{r+1}, v_{r+2}, \ldots, v_n\}$ *be a basis for* $\mathscr{N}(T)$, $\{u_1, u_2, \ldots, u_r\}$ *be basis for* $\mathscr{R}(T)$, $\{u_{r_1}, u_{r+2}, \ldots, u_m\}$ *be a basis for* $\mathscr{R}(T)^\perp$, *then:*

$$T^\dagger(u_i) = \begin{cases} \frac{1}{\sigma_i}v_i & \text{if } 1 \leq i \leq r \\ 0 \end{cases}$$

So although not all T *has inverse, the restriction* $T|_{\mathscr{N}(T)^\perp}$ *could have proper inverse.*

Theorem 178. *Let $A_{m\times n}$ be a square matrix of rank r with singular value decomposition $A = U\Sigma V^*$ and non-zero singular values $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r$. Let $\Sigma^\dagger_{m\times n}$ be a matrix that*

$$\Sigma^\dagger_{ij} = \begin{cases} \frac{1}{\sigma_i} & \text{if } i = j \leq r \\ 0 \end{cases} \tag{2.82}$$

Then $A^\dagger = V\Sigma^\dagger U^*$ is a singular value decomposition of A.

Theorem 179. *Let $T : V \to W$ be a linear transformation, then*

1. *$T^\dagger T$ is the orthogonal projection of V on $\mathcal{N}(T)^\perp$.*
2. *$TT^\dagger$ is the orthogonal projection of W on $\mathcal{R}(T)$.*

Proof. Define $L : \mathcal{N}(T)^\perp \to W$ by $L(x) = T(x)$. If $x \in \mathcal{N}(T)^\perp$, then $T^\dagger T(x) = L^{-1}L(x) = x$. If $x \in \mathcal{N}(T)$, then $T^\dagger T(x) = T^\dagger(0) = 0$. □

Theorem 180. *For a system of linear equations $Ax = b$. If $z = A^\dagger b$, then*

1. *If $Ax = b$ is consistent, then z is the unique solution with minimal norm.*
2. *If $Ax = b$ is inconsistent, then z is the best approximation: $\forall y, \|Ax - b\| \leq \|Ay - b\|$. Also if $Az = Ay$, then $\|z\| \leq \|y\|$.*

$A^\dagger b$ is the optimal solution discussed in section 2.6.4 on page 39.

Proof. Let $z = A^\dagger b$. If the equation is consistent, then $b \in \mathcal{R}(T)$, then $Az = AA^\dagger b = TT^\dagger(b) = b$ because $TT^\dagger$ is a orthogonal projection, so z is a solution to the linear system.

If y is any solution, then $T^\dagger T(y) = A^\dagger Ay = A^\dagger b = z$. So z is a orthogonal projection of y on $\mathcal{N}(T)^\perp$. So $\|z\| \leq \|y\|$.

If the equation is inconsistent, then $Az = AA^\dagger b$ is the orthogonal projection of b on $\mathcal{R}(T)$, so Az is the nearest vector to b. □

2.7.10 Conditioning

Definition 126. *For $Ax = b$, if a small change to A and b cause small change to x, the property is called well-conditioned. Otherwise the system is ill-conditioned.*

Definition 127. *The relative change in b is $\dfrac{\|db\|}{\|b\|}$ with $\|\cdot\|$ be the standard norm on $\mathscr{C}^n$.*

Definition 128. *The Euclidean norm of square matrix A is*

$$\|A\| = \max_{x\neq 0} \frac{\|Ax\|}{\|x\|} \tag{2.83}$$

Definition 129. *Let B be a self-adjoint matrix. The Rayleigh quotient for $x \neq 0$ is $R(x) = \dfrac{\langle Bx, x\rangle}{\|x\|^2}$*

Theorem 181. *For a self-adjoint matrix B, the $\max\limits_{x\neq 0} R(x)$ is the largest eigenvalue of B and $\min\limits_{x\neq 0} R(x)$ is the smallest eigenvalue of B.*

Proof. Choose the orthonormal basis v_i of B such that $Bv_i = \lambda_i v_i$ where $\lambda_1 \geq \lambda_2 \geq \lambda_n$. $\forall x \in F^n$, $\exists a_i$ that $x = \sum\limits_{i=1}^{n} a_i v_i$. So

$$R(x) = \frac{\langle Bx, x\rangle}{\|x\|^2} = \frac{\left\langle \sum\limits_{i=1}^{n} a_i\lambda_i v_i, \sum\limits_{j=1}^{n} a_j v_j \right\rangle}{\|x\|^2} = \frac{\sum_{i=1}^n \lambda_i |a_i|^2}{\|x\|^2} \leq \frac{\lambda_1 \sum_{i=1}^n |a_i|^2}{\|x\|^2} = \frac{\lambda_1\|x\|^2}{\|x\|^2} = \lambda_1$$

□

Theorem 182. $\|A\| = \sqrt{\lambda}$ *where λ is the largest eigenvalue of A^*A.*

Theorem 183. *λ is an eigenvalue of A^*A if and only if λ is an eigenvalue of AA^*.*

Theorem 184. *Let A be invertible matrix. Then $\|A^{-1}\| = \dfrac{1}{\sqrt{\lambda}}$ where λ is the smallest eigenvalue of A^*A.*

Definition 130. $\|A\| \times \|A^{-1}\|$ *is the* <u>*condition number*</u> *of* A *and denoted as* $cond(A)$*.*

Theorem 185. *For system* $Ax = b$ *where* A *is invertible and* $b \neq 0$*, we have:*

1. *For any norm* $\|\cdot\|$*, we have* $\frac{1}{cond(A)}\frac{\|db\|}{\|b\|} \leq \frac{\|dx\|}{\|x\|} \leq cond(A)\frac{\|db\|}{\|b\|}$.
2. *If* $\|\cdot\|$ *is the Euclidean norm, then* $cond(A) = \sqrt{\frac{\lambda_1}{\lambda_n}}$ *where* λ_1 *and* λ_n *are the largest and smallest eigenvalue of* A^*A.

So when $cond(b) \geq 1$*. If* $cond(b)$ *is close to* 1*, the relative error in* x *is small when relative error of* b *is small. However when* $cond(b)$ *is large, the relative error in* x *could be large or small.*

$cond(x)$ *is seldom calculated because when calculating* A^{-1} *in computer, there are rounding errors which is related to* $cond(A)$*.*

Chapter 3

Matrix

3.1 Matrix Calculus

3.1.1 Layout

There are two different layout:

- numerator layout:

$$\begin{bmatrix} \nabla f \\ \nabla g \end{bmatrix} \tag{3.1}$$

- denominator layout:

$$\begin{bmatrix} \nabla f, \nabla g \end{bmatrix} \tag{3.2}$$

numerator layout is preferred.

3.1.2 Jacobian Matrix

for $\mathrm{y}_{1\times m} = \mathrm{f}(\mathrm{x}_{1\times n})$, its Jacobian matrix is:

$$\nabla_{\mathrm{x}}\mathrm{y} = \frac{\partial \mathrm{y}}{\partial \mathrm{x}} = \begin{bmatrix} \nabla f_1(\mathrm{x}) \\ \nabla f_1(\mathrm{x}) \\ \vdots \\ \nabla f_m(\mathrm{x}) \end{bmatrix} = \begin{pmatrix} \frac{\partial f_1}{\partial x} \\ \frac{\partial f_2}{\partial x} \\ \vdots \\ \frac{\partial f_m}{\partial x} \end{pmatrix} = \begin{bmatrix} \frac{\partial f_1(\mathrm{x})}{x_1} & \frac{\partial f_1(\mathrm{x})}{x_2} & \cdots & \frac{\partial f_1(\mathrm{x})}{x_n} \\ \frac{\partial f_2(\mathrm{x})}{x_1} & \frac{\partial f_2(\mathrm{x})}{x_2} & \cdots & \frac{\partial f_2(\mathrm{x})}{x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_m(\mathrm{x})}{x_1} & \frac{\partial f_m(\mathrm{x})}{x_2} & \cdots & \frac{\partial f_m(\mathrm{x})}{x_n} \end{bmatrix} \tag{3.3}$$

3.1.3 Element-wise binary operator

for element-wise binary operator

$$\mathrm{y} = \mathrm{f}(\mathrm{w}) \bigcirc \mathrm{g}(\mathrm{x}) \tag{3.4}$$

$\bigcirc$ could be $+, -, \times$[1]$, \div, max$. The gradient is:

$$\nabla_{\mathrm{x}}\mathrm{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} f_1(\mathrm{w}) \bigcirc g_1(\mathrm{x}) \\ f_2(\mathrm{w}) \bigcirc g_2(\mathrm{x}) \\ \vdots \\ f_n(\mathrm{w}) \bigcirc g_n(\mathrm{x}) \end{bmatrix} \tag{3.5}$$

The expanded matrix could be differentiated using Jacobian matrix.

3.1.4 Vector Sum

Vector sum operation *sum* could be expressed as

$$y = \mathrm{sum}\Big(\mathrm{f}(\mathrm{x})\Big) = \sum_{i=1}^{n} f_i(\mathrm{x}) \tag{3.6}$$

$\nabla \mathrm{y}$ could be calculated as usual.

3.1.5 Chain Rules

In machine learning there are two ways of taking chain rules:

- forward differentiation: $\frac{dy}{dx} = \frac{du}{dx} \times \frac{dy}{du}$
- backward differentiation: $\frac{dy}{dx} = \frac{dy}{du} \times \frac{du}{dx}$

[1] called *hadamard product*

Backward differentiation is preferred for matrix operation.
The full expression of $\mathrm{y} = \mathrm{f}(\mathrm{g}(\mathrm{x}))$ is:

$$
\begin{aligned}
\nabla_{\mathrm{x}} f &= \frac{\partial\, \mathrm{f}(\mathrm{g}(\mathrm{x}))}{\partial \mathrm{x}} \\
&= \frac{\partial\, \mathrm{f}}{\partial \mathrm{g}} \times \frac{\partial\, \mathrm{g}}{\partial \mathrm{x}} \\
&= \begin{bmatrix}
\frac{\partial f_1(\mathrm{x})}{g_1} & \frac{\partial f_1(\mathrm{x})}{g_2} & \cdots & \frac{\partial f_1(\mathrm{x})}{g_n} \\
\frac{\partial f_2(\mathrm{x})}{g_1} & \frac{\partial f_2(\mathrm{x})}{g_2} & \cdots & \frac{\partial f_2(\mathrm{x})}{g_n} \\
\vdots & \vdots & \ddots & \vdots \\
\frac{\partial f_m(\mathrm{x})}{g_1} & \frac{\partial f_m(\mathrm{x})}{g_2} & \cdots & \frac{\partial f_m(\mathrm{x})}{g_n}
\end{bmatrix}_{m\times n}
\times
\begin{bmatrix}
\frac{\partial g_1(\mathrm{x})}{x_1} & \frac{\partial g_1(\mathrm{x})}{x_2} & \cdots & \frac{\partial g_1(\mathrm{x})}{x_r} \\
\frac{\partial g_2(\mathrm{x})}{x_1} & \frac{\partial g_2(\mathrm{x})}{x_2} & \cdots & \frac{\partial g_2(\mathrm{x})}{x_r} \\
\vdots & \vdots & \ddots & \vdots \\
\frac{\partial g_n(\mathrm{x})}{x_1} & \frac{\partial g_n(\mathrm{x})}{x_2} & \cdots & \frac{\partial g_n(\mathrm{x})}{x_r}
\end{bmatrix}_{n\times r}
\end{aligned}
\tag{3.7}
$$

Chapter 4

Probability

4.1 Random Variable

4.1.1 Events

Definition 131 (sample space). *The set of all possible outcome of an experiment is defined as* <u>*sample space*</u> S.

Definition 132 (event). *Any subset* E *of sample space* S *is defined as* <u>*event*</u>*.*

It is conventional to designate $E \cup F$ as EF.

Definition 133 (mutually exclusive). *If* $EF = \emptyset$*,* E *and* F *are said to be* <u>*mutually exclusive*</u>*.*

Definition 134. *The probabilty* P *is defined on event* E *of sample space* S *which follow the following condition:*

1. $0 \le P(E) \le 1$.
2. $P(S) = 1$.
3. *for any sequence of mutually exclusive event* $E_1, E_2, \ldots,$

$$P\left(\bigcup_{n=1}^{\infty} E_n\right) = \sum_{n=1}^{\infty} P(E_n)$$

□

If the experiment is repeated over and over again, with probability 1 the proportion of time that event E will occur is P(E).

Theorem 186 (inclusion-exclusion identity).

$$P\left(\bigcup_{i=1}^{n} E_i\right) = \sum_i P(E_i) - \sum_{i<j} P(E_iE_j) + \sum_{i<j<k} P(E_iE_jE_k) + \cdots + (-1)^{n+1} P(E_1E_2 \ldots E_n) \tag{4.1}$$

Definition 135 (conditional probability). *The* <u>*conditional probability*</u> *that* E *occurs given that* F *occurs is denoted by* $P(E|F)$ *and defined as:*

$$P(E|F) = \frac{P(EF)}{P(F)} \tag{4.2}$$

For conditional probability the sample space is changed.

□

Definition 136 (independent). *Two event* E *and* F *are* <u>*independent*</u> *if* $P(EF) = P(E)P(F)$.

Theorem 187. E *and* F *are independent if* $P(E/F) = P(E)$.

Definition 137. *The events* $\{E_i\}$ *are independent if for every subset* E_{i_j} *(*$\forall j, 1 \le j \le n$*), that*

$$P(E_{i_1}E_{i_2} \ldots E_{i_j}) = P(E_{i_1})P(E_{i_2}) \cdots P(E_{i_j}) \tag{4.3}$$

Definition 138. *For mutually exclusive event* F_i *that* $\bigcup_{i=1}^{n} F_i = S$ *.The* <u>*Bayes' formula*</u> *is defined as:*

$$P(F_j|F) = \frac{P\left\{EF_j\right\}}{\sum_{i=1}^{n} P\left\{EF_i\right\}} = \frac{P\left\{E|F_j\right\} P\left\{F_j\right\}}{\sum_{i=1}^{n} P\left\{E|F_i\right\} P\left\{F_i\right\}} \tag{4.4}$$

4.1.2 Random Variable Definition

Definition 139 (random variable). *A real value function defined on sample space is called* <u>*random variable*</u>*.*

Definition 140 (discrete random variable). *A random variable that can takes at most a countable values is said to be* <u>*discrete random variable*</u>*.*

Definition 141 (probability mass function). *The* <u>*probability mass function*</u> $p(a)$ *of discrete random variable* X *is defined as* $p(a) = P\{X = a\}$.

Definition 142 (cdf). *The* <u>*cumulative distribution function (cdf)*</u> *of random variable* X *for any* $-\infty < b < \infty$ *is defined as* $F(b) = P\{X \le b\}$

Theorem 188. *Some properties of cdf are:*

1. $F(b)$ *is nondecreasing function of* b.
2. $F(b)$ *is continuous from the right.*
3. $\lim_{b\to\infty} F(b) = F(\infty) = 1$.
4. $\lim_{b\to-\infty} F(b) = F(-\infty) = 0$.

Definition 143. $P\{X < b\}$ *is defined as* $P\{X < b\} = \lim_{h\to0^+} P\{X \le b-h\} = \lim_{h\to0^+} F(b-h)$.

$P\{X < b\} \neq F(b)$ because $F(b)$ also include the probability that X equals b.

Poisson random variable is a approximate of binomial random variable when n is large and p is small. Let $\lambda = np$, then:

$$\begin{aligned}\binom{n}{i}p^i(1-p)^{n-i} &= \frac{n!}{(n-i)!i!}\left(\frac{\lambda}{n}\right)^i\left(1-\frac{\lambda}{n}\right)^{n-i} \\ &= \frac{n(n-1)\cdots(n-i+1)}{n^i}\frac{\lambda^i}{i!}\frac{(1-\frac{\lambda}{n})^n}{(1-\frac{\lambda}{n})^i} \\ &\approx 1\times\frac{\lambda^i}{i!}\times\frac{e^{-\lambda}}{1} = e^{-\lambda}\frac{\lambda^i}{i!}\end{aligned}$$

Definition 144. $f(x)$ *is called* probability density function *if* $P\{X \in B\} = \int_B f(x)\,dx$. □

Because $P\left\{a-\frac{\varepsilon}{2} \le X \le a+\frac{\varepsilon}{2}\right\} = \int_{a-\frac{\varepsilon}{2}}^{a+\frac{\varepsilon}{2}} f(x)\,dx \approx \varepsilon f(a)$, $f(a)$ is a measure of how likely it is that the random variable will be near a within interval ε.

Sometimes when calculating the probability $P\{X = a\}$, we can first calculate $F(a)$ and then calculate $P\{X = a\} = \frac{dF(a)}{da}$.

Definition 145 (expectation). *The* expectation *of* X *is defined as:*

$$E[X] = \begin{cases}\sum xp(x) & \text{for discrete case} \\ \int_{-\infty}^{\infty} xf(x)\,dx & \text{for continuous case}\end{cases} \tag{4.5}$$

Theorem 189. *The expectation of a function* g *of a random variable* X *is:*

$$E\big[g(X)\big] = \begin{cases}\sum g(x)p(x) & \text{for discrete case} \\ \int_{-\infty}^{\infty} g(x)f(x)\,dx & \text{for continuous case}\end{cases} \tag{4.6}$$

Theorem 190.

$$E[aX+b] = a\,E[X]+b \tag{4.7}$$

Theorem 191.

$$E[aX+bY] = a\,E[X]+b\,E[Y] \tag{4.8}$$

Definition 146 (variance). *The* variance *of* X *is defined as*

$$\mathrm{Var}[X] = E\big[(X-E[X])^2\big] \tag{4.9}$$

Theorem 192.

$$\mathrm{Var}[X] = E\big[X^2\big]-E[X]^2 \tag{4.10}$$

Theorem 193. *Let* X_i *be identically independent random variable. We need to notice that*

$$\mathrm{Var}\left[\sum_{i=1}^{n} X_i\right] = n\,\mathrm{Var}[X] \neq \mathrm{Var}\big[n\times X_i\big] = n^2\,\mathrm{Var}[X]$$

4.1.3 Common Distributions

name	density	$\phi(t)$	mean	var
binomial	$\binom{n}{x}p^x(1-p)^{n-x}$	$(pe^t+q)^n$	np	npq
poisson	$e^{-\lambda}\frac{\lambda^x}{x!}$	$e^{\lambda(e^t-1)}$	λ	λ
geometric	$p(1-p)^{x-1}$	$\frac{pe^t}{1-(1-p)e^t}$	$\frac{1}{p}$	$\frac{1-p}{p^2}$
uniform	$\frac{1}{b-a}$	$\frac{e^{tb}-e^{ta}}{t(b-a)}$	$\frac{a+b}{2}$	$\frac{(b-a)^2}{12}$
exponential	$\lambda e^{-\lambda x}$	$\frac{\lambda}{\lambda-t}$	$\frac{1}{\lambda}$	$\frac{1}{\lambda^2}$
gamma	$\frac{\lambda e^{-\lambda x}(\lambda x)^{n-1}}{(n-1)!}$	$\left(\frac{\lambda}{\lambda-t}\right)^n$	$\frac{n}{\lambda}$	$\frac{n}{\lambda^2}$
normal	$\frac{1}{\sqrt{2\pi}\sigma}e^{-\frac{(x-\mu)^2}{2\sigma^2}}$	$e^{\mu t+\frac{\sigma^2t^2}{2}}$	μ	σ^2

Theorem 194. *If X is normally distributed with μ and ρ^2, then $Y = aX + b$ is a normal distribution with $a\mu + b$ and $(a\rho)^2$. So $Y = \frac{X-\mu}{\rho}$ is normally distributed with 0 and 1, which is called standard normal distribution.*

Definition 147. *A random variable X has gamma distribution with parameter α and β if the pdf is:*

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

while $\Gamma(\alpha)$ is defined as:

$$\Gamma(\alpha) = \int_0^\infty e^{-x}x^{\alpha-1}\,dx \tag{4.11}$$

exponential distribution is gamma distribution with $\alpha = 1$.

□

Theorem 195. *If X_i are independent gamma distribution with α_i and β, then the sum $\sum_{i=1}^{n} X_i$ has gamma distribution with $\sum \alpha_i$ and β.*

Theorem 196.

$$\Gamma\left(\frac{1}{1}\right) = \sqrt{\pi} \tag{4.12}$$

$$\Gamma(a+1) = a\Gamma(a) \tag{4.13}$$

$$\Gamma(n) = (n-1)! \tag{4.14}$$

□

Definition 148. *X is a beta distribution if the pdf is:*

$$f(x|\alpha,\beta) = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)}x^{\alpha-1}(1-x)^{\beta-1} \tag{4.15}$$

4.1.4 Joint Distribution

Definition 149. *The joint cumulative probability distribution function of two discrete random variable X and Y is defined as*

$$F(a,b) = P\{X \le a, Y \le b\} \tag{4.16}$$

The cumulative distribution of Y is defined as

$$F_X(a) = P\{X \le a\} = P\{X \le a, Y < \infty\} = F(a,\infty) \tag{4.17}$$

The joint probability mass function is defined as

$$p(x,y) = \mathrm{P}\left\{\mathrm{X} = x, \mathrm{Y} = y\right\} \tag{4.18}$$

The probability mass function of X *from* $p(x,y)$ *is defined as*

$$p_{\mathrm{X}}(x) = \sum_{y:p(x,y)>0} p(x,y) \tag{4.19}$$

□

Definition 150. *for continuous case,* X *and* Y *are jointly continuous if there is a function* $f(x,y)$ *that for all sets* A *and* B *we have*

$$\mathrm{P}\{\mathrm{X}\in\mathrm{A},\mathrm{Y}\in\mathrm{B}\} = \int_{\mathrm{B}}\int_{\mathrm{A}} f(x,y)\,\mathrm{d}x\,\mathrm{d}y \tag{4.20}$$

$f(x,y)$ *is called joint probability density function of* X *and* Y. *The probability density function of* X *can be obtained as*

$$\begin{aligned}\mathrm{P}\{\mathrm{X}\in\mathrm{A}\} &= \int_{\mathrm{A}}\int_{-\infty}^{\infty} f(x,y)\,\mathrm{d}x\,\mathrm{d}y \\ &= \int_{\mathrm{A}} f_{\mathrm{X}}(x)\,\mathrm{d}x\end{aligned} \tag{4.21}$$

where

$$f_{\mathrm{X}}(x) = \int_{-\infty}^{\infty} f(x,y)\,\mathrm{d}y \tag{4.22}$$

We have

$$\frac{\partial^2 \mathrm{F}(a,b)}{\partial a^1 \partial b^1} = f(a,b) \tag{4.23}$$

□

For n random variable X_i, we could construct Y_i that

$$\begin{aligned}\mathrm{Y}_1 &= g_1(\mathrm{X}_1,\mathrm{X}_2,\cdots,\mathrm{X}_n)\\ \mathrm{Y}_2 &= g_2(\mathrm{X}_1,\mathrm{X}_2,\cdots,\mathrm{X}_n)\\ &\vdots\\ \mathrm{Y}_n &= g_n(\mathrm{X}_1,\mathrm{X}_2,\cdots,\mathrm{X}_n)\end{aligned} \tag{4.24}$$

Assume its Jacobian determinant $\mathrm{J}(x_1,\ldots,x_n)\neq 0$ for all x_i. Then the probability density function of Y_i is

$$f_{\mathrm{Y}_1,\cdots,\mathrm{Y}_n}(y_1,\cdots,y_n) = \frac{1}{\left|\mathrm{J}(x_1,\cdots,x_n)\right|} f_{\mathrm{X}_1,\cdots,\mathrm{X}_n}(x_1,\cdots,x_n) \tag{4.25}$$

So the process of calculating f_{Y_i} are:

1. solve $\mathrm{X}_i = h(\mathrm{Y}_i)$ from equation (4.24).
2. calculate $\mathrm{J}(\mathrm{X}_i)$.
3. replace x_i by y_i in $f(x_i)$ and multiplicate by $\frac{1}{|\mathrm{J}|}$.

4.1.5 Independence

Definition 151. *Two random variable* X *and* Y *are independent if* $\forall a, b \in \mathrm{R}$,

$$\mathrm{P}\{\mathrm{X}\le a,\mathrm{Y}\le b\} = \mathrm{P}\{\mathrm{X}\le a\}\,\mathrm{P}\{\mathrm{X}\le b\} \tag{4.26}$$

It means

$$\mathrm{F}(a,b) = \mathrm{F}_{\mathrm{X}}(a)\mathrm{F}_{\mathrm{Y}}(b) \tag{4.27}$$

When X *and* Y *are discrete, it reduces to*

$$p(x,y) = p_{\mathrm{X}}(x)p_{\mathrm{Y}}(y) \tag{4.28}$$

When they are continuous, it reduces to

$$f(x,y) = f_{\mathrm{X}}(x)f_{\mathrm{Y}}(y) \tag{4.29}$$

□

Theorem 197. *If* X *and* Y *are indepent, for any function* h *and* g, *we have*

$$\mathrm{E}\left[g(\mathrm{X})h(\mathrm{Y})\right] = \mathrm{E}\left[g(\mathrm{X})\right]\mathrm{E}\left[h(\mathrm{Y})\right] \tag{4.30}$$

4.1.6 Covariance

Definition 152 (covariance)**.** *The covariance for* X *and* Y *is defined as*

$$\begin{aligned}\mathrm{Cov}(\mathrm{X},\mathrm{Y}) &= \mathrm{E}\left[(\mathrm{X}-\mathrm{E}[\mathrm{X}])(\mathrm{Y}-\mathrm{E}[\mathrm{Y}])\right] \\ &= \mathrm{E}[\mathrm{XY}]-\mathrm{E}[\mathrm{X}]\,\mathrm{E}[\mathrm{Y}]\end{aligned} \tag{4.31}$$

In general covariance means Y *tends to move in the same direction of* X. □

Theorem 198.

$$\mathrm{Cov}(\mathrm{X},\mathrm{Y}) = \mathrm{Var}[\mathrm{X}] \tag{4.32}$$

Theorem 199.

$$\mathrm{Cov}(\mathrm{X},\mathrm{Y}) = \mathrm{Cov}(\mathrm{Y},\mathrm{X}) \tag{4.33}$$

Theorem 200.

$$\mathrm{Cov}(c\mathrm{X},\mathrm{Y}) = c\,\mathrm{Cov}(\mathrm{X},\mathrm{Y}) \tag{4.34}$$

Theorem 201.

$$\mathrm{Cov}(\mathrm{X},\mathrm{Y}+\mathrm{Z}) = \mathrm{Cov}(\mathrm{X},\mathrm{Y}) + \mathrm{Cov}(\mathrm{X},\mathrm{Z}) \tag{4.35}$$

Theorem 202.

$$\mathrm{Cov}\left(\sum_{i=1}^{n}\mathrm{X}_i, \sum_{j=1}^{m}\mathrm{Y}_j\right) = \sum_{i=1}^{n}\sum_{j=1}^{m}\mathrm{Cov}(\mathrm{X}_i,\mathrm{Y}_j) \tag{4.36}$$

Theorem 203.

$$\begin{aligned}\mathrm{Var}\left(\sum_{i=1}^{n}\mathrm{X}_i\right) &= \mathrm{Cov}\left(\sum_{i=1}^{n}\mathrm{X}_i, \sum_{j=1}^{n}\mathrm{X}_j\right) \\ &= \sum_{i=1}^{n}\mathrm{Var}\left[\mathrm{X}_i\right] + 2\sum_{i=1}^{n}\sum_{j<i}\mathrm{Cov}\left(\mathrm{X}_i,\mathrm{X}_j\right) \\ &= \sum_{i=1}^{n}\mathrm{Var}\left[\mathrm{X}_i\right] \; \textit{(if } \mathrm{X}_i,\mathrm{X}_j \textit{ are independent)}\end{aligned} \tag{4.37}$$

This theorem is often used to calculate the variance. □

Definition 153 (sample mean)**.** *If* X_i *are independent and identically distributed, then the random variable* $\bar{\mathrm{X}} = \frac{\sum_{i=1}^{n}\mathrm{X}_i}{n}$ *is called the sample mean.*

Theorem 204.

$$\mathrm{E}\left[\bar{\mathrm{X}}\right] = \mu \tag{4.38}$$

Theorem 205.

$$\mathrm{Var}\left[\bar{\mathrm{X}}\right] = \frac{\sigma^2}{n} \tag{4.39}$$

Theorem 206.

$$\mathrm{Cov}\left(\bar{\mathrm{X}},\mathrm{X}_i-\bar{\mathrm{X}}\right) = 0 \tag{4.40}$$

Definition 154. *Function* $\mathrm{F}_{\mathrm{X+Y}}$ *is called the convolution of the distribution of* F_{X} *and* F_{Y}*, which is*

$$\begin{aligned}\mathrm{F}_{\mathrm{X+Y}}(a) &= \mathrm{P}\{\mathrm{X}+\mathrm{Y}\le a\} \\ &= \iint_{x+y\le a} f(x)g(y)\,\mathrm{d}x\,\mathrm{d}y \\ &= \int_{-\infty}^{\infty}\left(\int_{-\infty}^{a-y} f(x)\,\mathrm{d}x\right)g(y)\,\mathrm{d}y \\ &= \int_{-\infty}^{\infty}\mathrm{F}_{\mathrm{X}}(a-y)g(y)\,\mathrm{d}y\end{aligned} \tag{4.41}$$

The probability density function $f_{X+Y}(a)$ is given by

$$\begin{aligned} f_{X+Y}(a) &= \frac{dF_{X+Y}(a)}{da} \\ &= \frac{d\int_{-\infty}^{\infty} F_X(a-y)g(y)\,dy}{da} \\ &= \int_{-\infty}^{\infty} f(a-y)g(y)\,dy \end{aligned} \tag{4.42}$$

□

Definition 155. *The* <u>*moment generating function*</u> *$\phi(t)$ of the random variable X is defined by*

$$\begin{aligned} \phi(x) &= E\left[e^{tX}\right] \\ &= \begin{cases} \sum_x e^{tx}p(x) \\ \int_{-\infty}^{\infty} e^{tx}f(x)\,dx \end{cases} \end{aligned} \tag{4.43}$$

For any n random variable X_i, the <u>*joint moment generating function*</u> *$\phi(t_i)$ is defined by $\phi(t_1,\ldots,t_n) = E\left[e^{\sum_{i=1}^{n} t_i X_i}\right]$.*
□

Because $\phi^n(t) = E\left[X^n e^{tX}\right]$, we have $\phi^n(0) = E[X^n]$.

Theorem 207.

$$\phi_{X+Y}(t) = \phi_X(t)\phi_Y(t) \tag{4.44}$$

Theorem 208. *The moment generating function uniquely determine the distribution.*

4.1.7 Sample Mean and Sample Variance

Definition 156 (sample mean). *The* <u>*sample mean*</u> *is defined as $\bar{X} = \frac{\sum X_i}{n}$.* □

Definition 157 (sample variance). *Let X_i be independent and identically distributed random variable with mean μ and variance σ^2, the* <u>*sample variance*</u> *S^2 is defined by*

$$S^2 = \sum_{i=1}^{n} \frac{(X_i - \bar{X})^2}{n-1} \tag{4.45}$$

□

Theorem 209.

$$E\left[S^2\right] = \sigma^2 \tag{4.46}$$

Definition 158 (chi-squared). *If Z_i are n independent standard normal random variables, the random variable $\sum_{i=1}^{n} Z_i^2$ is called* <u>*chi-squared*</u> *random variable with n degrees of freedom.*

Theorem 210. *If X_i are independent and identically distributed normal random variable with mean μ and variance σ^2, then:*

1. *the sample mean $\bar{X}$ and sample variance S^2 are independent.*
2. *$\bar{X}$ is a normal random variable with mean μ and variance $\frac{\sigma^2}{n}$.*
3. *$\frac{(n-1)S^2}{\sigma^2}$ is a chi-squared random variable with $n-1$ degrees of freedom.*

Proof. Because X_1 are normal random varialbe, $\bar{X}$ is also a normal random variable. Since $\mathrm{Cov}\left(\bar{X}, X_i - \bar{X}\right) = 0$, normal random variable $\bar{X}$ and $X_i - \bar{X}$ are independent. Since $S^2 = \sum_{i=1}^{n} \frac{(X_i - \bar{X})^2}{n-1}$ is a function of $X_i - \bar{X}$ which is independ from $\bar{X}$, S^2 is independent from $\bar{X}$.

Because

$$\frac{(n-1)S^2}{\sigma^2}+\left(\frac{\bar{X}-\mu}{\frac{\sigma}{\sqrt{n}}}\right)^2=\sum_{i=1}^{n}\frac{(X_i-\mu)^2}{\sigma^2}$$

generate the moment generating function of $\frac{(n-1)S^2}{\sigma^2}$ from above formula and we have $E\left[e^{t\frac{(n-1)S^2}{\sigma^2}}\right]=(1-2t)^{-\frac{n-1}{2}}$ which is an chi-squared distribution with freedom $n-1$. □

4.1.8 Inequality

Theorem 211 (Markov's Inequality). *If X is a random variable that is non-negative. Then for any $a>0$ we have*

$$P\{X\geq a\}\leq\frac{E[X]}{a} \tag{4.47}$$

Proof.

$$\begin{aligned}E[X]&=\int_0^\infty xf(x)\,dx\\&=\int_0^a xf(x)\,dx+\int_a^\infty xf(x)\,dx\\&\geq\int_a^\infty xf(x)\,dx\\&\geq\int_a^\infty af(x)\,dx\\&=a\int_a^\infty f(x)\,dx\\&=P\{X\geq a\}\end{aligned}$$

□

Theorem 212 (Chebyshev's Inequality). *If X is a random variable with mean μ and variance σ^2. Then for any $k>0$ we have*

$$P\{|X-\mu|\geq k\}\leq\frac{\sigma^2}{k^2} \tag{4.48}$$

Proof. Since $(X-\mu)^2$ is nonnegative, using Markov's Inequality we have

$$P\{|(X-\mu)|\geq k\}=P\{(X-\mu)^2\geq k^2\}\leq\frac{E[(X-\mu)^2]}{k^2}$$

□

Theorem 213 (Strong Law of Large Numbers). *Let X_i be a sequence of independent random variables having a common distribution. With probability 1*

$$\frac{X_1+X_2+\cdots+X_n}{n}\to\mu \text{ as } n\to\infty \tag{4.49}$$

Theorem 214 (Central Limit Theorem). *Let X_i be a sequence of independent identical random variables with mean μ and variance σ^2, the distribution $\frac{X_1+X_2+\cdots+X_n-n\mu}{\sigma\sqrt{n}}$ tends to be standard normal distribution as $n\to\infty$.*

4.1.9 Examples

Example 2. *Suppose m coupons are selected from n different types of coupons. Each coupon is equally likely to be selected. What is the expected number of different types?*

Proof. Let X denote the number of types in m selection. We have $X=X_1+\ldots+X_n$ where

$$X_i=\begin{cases}1 & \text{, if } i \text{ occurs in selection}\\0 & \text{, otherwise}\end{cases}$$

We have

$$\mathrm{E}\left[\mathrm{X}_i\right] = 1 - \left(\frac{n-1}{n}\right)^m$$

So $\mathrm{E}[\mathrm{X}] = \mathrm{E}\left[\mathrm{X}_1\right] + \cdots + \mathrm{E}\left[\mathrm{X}_n\right] = n\left(1 - \left(\frac{n-1}{n}\right)^m\right)$ □

Example 3. *Let X_i be independent and identically distributed continuous random variable. Let $\mathrm{X}_{(i)}$ denote the ith smallest of these random variables. Then $\mathrm{X}_{(i)}$ is called <u>order statistics</u>. Note that $\mathrm{X}_{(i)} \le x$ if and only if at least i of X_j are not larger than x, so*

$$\mathrm{P}\left\{\mathrm{X}_{(i)} \le x\right\} = \sum_{k=i}^{n} \binom{n}{k} \left(\mathrm{F}(x)\right)^k \left(1 - \mathrm{F}(x)\right)^{n-k}$$

So

$$\begin{aligned} f_{\mathrm{X}_{(i)}}(x) &= \frac{\mathrm{d}\,\mathrm{P}\left\{\mathrm{X}_{(i)} \le x\right\}}{\mathrm{d}x} \\ &= \frac{n!}{(n-i)!(i-1)!} f(x) \left(\mathrm{F}(x)\right)^{i-1} \left(1 - \mathrm{F}(x)\right)^{n-i} \end{aligned}$$

The result could be read as there are $\left(\mathrm{F}(x)\right)^{i-1}$ which are less than x, $\left(1 - \mathrm{F}(x)\right)^{n-i}$ which are larger than x, and $f(x)$ which equals x. □

Example 4. *A particle moves along a circle of m + 1 nodes. Each time it has equality of moving clockwise or counterclockwise. What is the probability that node i is the last visited node?*

Proof. Suppose i is the last visited nodes and consider 2 nodes $i-1$ and $i+1$ around it. If $i-1$ is visited before $i+1$, the probability is the same as gambling without losing money when it reaches $i+1$, so the probability is the same for every nodes. So the result is $\frac{1}{m}$. □

4.2 Conditional Probability

4.2.1 Conditional Probability

Theorem 215.

$$\mathrm{P}\{\mathrm{E}\} = \begin{cases} \sum_{y} \mathrm{P}\{\mathrm{E}|\mathrm{Y}=y\}\, p(\mathrm{Y}=y) \\ \int_{-\infty}^{\infty} f_{\mathrm{Y}}(y)\,\mathrm{d}y \end{cases} \tag{4.50}$$

4.2.2 Conditional Expectation

Definition 159. *The conditional probability mass function of* X *given* $\mathrm{Y}=y$ *is*

$$p_{\mathrm{X}|\mathrm{Y}}(x|y) = \frac{p(x,y)}{p_{\mathrm{Y}}(y)} \tag{4.51}$$

The conditional probability density function of X *given* $\mathrm{Y}=y$ *(where* $f_{\mathrm{Y}}(y)>0$*) is*

$$f_{\mathrm{X}|\mathrm{Y}}(x|y) = \frac{f(x,y)}{f_{\mathrm{Y}}(y)} = \frac{f(x,y)}{\int_{x} f(x,y)\,\mathrm{d}x} \tag{4.52}$$

Definition 160. *The conditional expectation of* X *given* $\mathrm{Y}=y$ *is*

$$\mathrm{E}\left[\mathrm{X}|\mathrm{Y}=y\right] = \begin{cases} \sum_{x} x \cdot p_{\mathrm{X}|\mathrm{Y}}(x|y) \\ \int_{-\infty}^{\infty} x f_{\mathrm{X}|\mathrm{Y}}(x|y)\,\mathrm{d}s \end{cases} \tag{4.53}$$

Note: $\mathrm{E}\left[\mathrm{X}|\mathrm{Y}\right]$ is a random variable of Y. Y here may be an expression of X as the following example shows.

Example 5. *If* X *and* Y *are independent variable, calculate the conditional expectation of* X *given* $\mathrm{X}+\mathrm{Y}=n$.

Theorem 216.

$$\begin{aligned} \mathrm{E}[\mathrm{X}] &= \mathrm{E}\left[\mathrm{E}\left[\mathrm{X}|\mathrm{Y}\right]\right] \\ &= \begin{cases} \sum_{y} \mathrm{E}\left[\mathrm{X}|\mathrm{Y}=y\right] \mathrm{P}\left\{\mathrm{Y}=y\right\} \\ \int_{-\infty}^{\infty} \mathrm{E}\left[\mathrm{X}|\mathrm{Y}=y\right] f_{\mathrm{Y}}(y)\,\mathrm{d}y \end{cases} \end{aligned} \tag{4.54}$$

4.2.3 Conditional Variance

Theorem 217.

$$\begin{aligned} \mathrm{Var}\left[\mathrm{X}|\mathrm{Y}=y\right] &= \mathrm{E}\left[(\mathrm{X}-\mathrm{E}\left[\mathrm{X}|\mathrm{Y}=y\right])^2|\mathrm{Y}=y\right] \\ &= \mathrm{E}\left[\mathrm{X}^2|\mathrm{Y}=y\right] - \left(\mathrm{E}\left[\mathrm{X}|\mathrm{Y}=y\right]\right)^2 \end{aligned} \tag{4.55}$$

Theorem 218.

$$\mathrm{Var}[\mathrm{X}] = \mathrm{E}\left[\mathrm{Var}\left[\mathrm{X}|\mathrm{Y}\right]\right] + \mathrm{Var}\left[\mathrm{E}\left[\mathrm{X}|\mathrm{Y}\right]\right] \tag{4.56}$$

4.2.4 Example

Example 6. *A miner is trapped in a mine containing 3 doors. The first door needs to a tunnel that takes him to safety after 2 hours. The second door take him back after 3 hours and the third door takes him back after 5 hours. What is the expected length of time until the miner reaches out when he chooses the door equally?* □

Example 7. *n men through their hat into the room and randomly select one. They will leave the game if they have selected their own hat.*

1. *what is the probability of no match in first round?*
2. *what is the probability of k match in first round?*
3. *what is the expected match in first round?*

4. *what is the variance of match in first round?*
5. *what is the expected number of rounds?*
6. *what is the variance of rounds?*
7. *what is the expected total number of selection?*
8. *what is the expected number of wrong selection for each man?*

Proof. define these variables:

- R_n: the number of rounds necessary for n men.
- S_n: the total number of all selection.
- C_i: the number of all wrong selection by ith man.
- X_n: the number of matches in the first round.

For question (1) let E_n be the event that no match occurs in n men scenario, M be the event that the first man select its own hat. We have

$$P\{E_n\} = P\{E_n|M\}P\{M\} + P\{E_n|M^c\}P\{M^c\}$$

Because $P\{E|M\} = 0$, $P\{E_n\} = \frac{n-1}{n}P\{E_n|M^c\}$.

In M^c case, the first man selected the hat of another person. there are two possibilities on whether the other person select the hat of the first man:

1. the "another" man did not select the hat of first man. In this case, $P\{E_n|?\} = P\{E_{n-1}\}$.
2. the "another" man did select the hat of the first man. In this case, $P\{E_n|?\} = P\{E_{n-2}\}P\{?\} = \frac{1}{n-1}P\{E_{n-2}\}$.

So

$$P\{E_n\} = P\{E_{n-1}\} + \frac{1}{n-1}P\{E_{n-2}\}$$

and $P\{E_n\} = \sum_{i=2}^{n}\frac{(-1)^i}{n!}$

For question (2), for any fixed group of k men that select their own hat, the probability is

$$\frac{1}{n}\frac{1}{n-1}\cdots\frac{1}{n-(k-1)}P\{E_{n-k}\} = \frac{(n-k)!}{n!}P\{E_{n-k}\}$$

Because there are $\binom{n}{k}$ choices of k men, the result is

$$\frac{(n-k)!}{n!}P\{E_{n-k}\}\binom{n}{k} = \frac{1}{k!}\sum_{i=2}^{n-k}(-1)^{n-k}\frac{1}{i!}$$

Another way is to calculate the length C of the cycle that contains the first man. we have

$$\begin{aligned}P\{E_n\} &= \sum_{k=1}^{n}P\{E_n|C=k\}P\{C=k\}\\ &= \sum_{k=1}^{n}P\{E_{n-k}\}P\{C=k\}\end{aligned}$$

We have

$$P\{C=k\} = \frac{n-1}{n}\frac{n-2}{n-1}\cdots\frac{n-k+1}{c-k+2}\frac{1}{n-k+1} = \frac{1}{n}$$

So $P\{E_n\} = \frac{1}{n}\sum_{k=2}^{n}P\{E_{n-k}\}$.

For equestion question (3), the probability ith man select its hat is $P\{H_i\} = \frac{1}{n}$, so

$$\begin{aligned}E\left[\sum_i H_i\right] &= \sum_i E[H_i]\\ &= \sum_i \frac{1}{n}\\ &= 1\end{aligned}$$

For question (4), the $\mathrm{Var}\left[\sum_i H_i\right] = 1$

For question (5),

$$\begin{aligned}
E[R_n] &= \sum_{i=0}^{n} E[R_n | X_n = i] P\{X_n = i\} \\
&= \sum_{i=0}^{n} (1 + E[R_{n-i}]) P\{X_n = i\} \\
&= 1 + E[R_n] P\{X_n = 0\} + \sum_{i=0}^{n} E[R_{n-i}] P\{X_n = i\} \\
&= E[R_n] P\{X_n = 0\} + n(1 - P\{X_n = 0\})
\end{aligned}$$

So the only solution is $E[R_n] = n$.
For question (6), for the variance, we have

$$E[R_n | X] = 1 + E[R_{n-X}] = 1 + n - X$$

Also we have $\mathrm{Var}[R_n | X] = \mathrm{Var}[R_{n-X}]$. So we have

$$\begin{aligned}
\mathrm{Var}[R_n] &= E\left[\mathrm{Var}[R_n | X]\right] + \mathrm{Var}\left[E[R_n | X]\right] \\
&= E\left[\mathrm{Var}[R_{n-X}]\right] + \mathrm{Var}[X] \\
&= \sum_{i=0}^{n} \mathrm{Var}[R_{n-j}] P\{X = j\} + \mathrm{Var}[X] \\
&= \mathrm{Var}[R_n] P\{X = 0\} + \sum_{j=1}^{n} \mathrm{Var}[R_{n-j}] P\{X = j\} + \mathrm{Var}[X]
\end{aligned}$$

The solution is $\mathrm{Var}[R_n] = n$.
For question (7), for S_n, we have

$$E[S_n] = n + E\left[S_{n-X_n}\right]$$

And the solution is $E[S_n] = n + \frac{n^2}{2}$.

For question (8),

$$\sum_{j=1}^{n} (C_j + 1) = S_n$$

$$\begin{aligned}
E[C_j] &= E[C_j + 1] - 1 \\
&= \frac{E[S_n]}{n} - 1 \\
&= \left(1 + \frac{n}{2}\right) - 1 \\
&= \frac{n}{2}
\end{aligned}$$

□

Example 8. *what is the number of necessary trial to get k consecutive success? Each trial has probability p of being successful.*

Proof. Let N_k the number of k consecutive success. $A_{k-1,k}$ be the additional trial from $k-1$ success to k success, so

$$E[N_k] = E[N_{k-1}] + E[A_{k-1,k}]$$

We have

$$E[A_{k-1,k}] = 1 \times p + (1 + E[N_k])(1 - p) = 1 + (1 - p) E[N_k]$$

So

$$E[N_k] = \frac{1}{p} + \frac{E[N_{k-1}]}{p}$$

with

$$\mathrm{E}\left[\mathrm{N}_1\right] = \frac{1}{p}$$

So

$$\mathrm{E}\left[\mathrm{N}_k\right] = \sum_{i=1}^{k} \frac{1}{p^i}$$

□

Example 9. *Analyse the quick sort algorithm*

Proof. Suppose there are n numbers and all permutation are of equally likely. Let M_n be the expected number of comparison needed for n numbers. So

$$\begin{aligned}\mathrm{M}_n &= \sum_{j=1}^{n} \mathrm{E}\left[\text{number of comparison}|\text{pivot is the } i\text{th smallest}\right]\frac{1}{n}\\ &= \sum_{j=1}^{n}(n-1+\mathrm{M}_{j-1}+\mathrm{M}_{n-j})\frac{1}{n}\\ &= n-1+\frac{2}{n}\sum_{k=1}^{n-1}\mathrm{M}_k\end{aligned}$$

So

$$\begin{aligned}\mathrm{M}_{n+1} &= 2(n+2)\sum_{i=1}^{n}\frac{i}{(i+1)(i+2)}\\ &\approx 2(n+2)\log(n+2)\end{aligned}$$

□

Example 10 (compound random variable)**.** *Let* X_i *be identical independent random variable with mean* μ *and variance* σ^2*. Let* N *be random variable. The random variable* $\sum_{i=1}^{\mathrm{N}} \mathrm{X}_i$ *is called* <u>*compound random variable.*</u> *Its expectation is* $\mathrm{E}\left[\sum_{i=1}^{\mathrm{N}}\mathrm{X}_i\right] = \mathrm{E}[\mathrm{N}]\,\mathrm{E}[\mathrm{X}]$*, and the variance is* $\mathrm{Var}\left[\sum_{i=1}^{\mathrm{N}}\mathrm{X}_i\right] = \sigma^2\,\mathrm{E}[\mathrm{N}] + \mu^2\,\mathrm{Var}[\mathrm{N}]$

Proof. Let $\mathrm{S} = \sum_{i=1}^{\mathrm{N}}\mathrm{X}_i$ be the random variable.

$$\begin{aligned}\mathrm{E}\left[\mathrm{S}|\mathrm{N}=n\right] &= \mathrm{E}\left[\sum_{i=1}^{n}\mathrm{X}_i|\mathrm{N}=n\right]\\ &= \mathrm{E}\left[\sum_{i=1}^{n}\mathrm{X}_i\right]\\ &= n\,\mathrm{E}[\mathrm{X}]\end{aligned}$$

Thus

$$\mathrm{E}[\mathrm{S}] = \mathrm{E}\left[\mathrm{N}\,\mathrm{E}[\mathrm{X}]\right] = \mathrm{E}[\mathrm{N}]\,\mathrm{E}[\mathrm{X}]$$

$$\begin{aligned}\mathrm{Var}\left[\mathrm{S}|\mathrm{N}=n\right] &= \mathrm{Var}\left[\sum_{i=1}^{n}\mathrm{X}_i|\mathrm{N}=n\right]\\ &= \mathrm{Var}\left[\sum_{i=1}^{n}\mathrm{X}_i\right]\\ &= n\,\mathrm{Var}[\mathrm{X}]\end{aligned}$$

So $\mathrm{Var}\left[S|N\right] = N\,\mathrm{Var}\left[X\right]$, $\mathrm{E}\left[S|N\right] = N\,\mathrm{E}\left[X\right]$, and

$$\begin{aligned}\mathrm{Var}[S] &= \mathrm{E}\Big[\mathrm{Var}\big[S|N\big]\Big] + \mathrm{Var}\Big[\mathrm{E}\big[S|N\big]\Big] \\ &= \mathrm{E}[N]\,\mathrm{Var}[S] + \mathrm{Var}[N]\big(\mathrm{E}[S]\big)^2\end{aligned}$$

□

Example 11. *For n distinct value, select the biggest one using the following rules:*

1. *determine the maximum of first k value. reject all of them.*
2. *select the first one that is larger than the maximum value just found.*

What is the probability that the rule select the maximum among all n values?

Proof. Let X be the position of largest value and $P_k(\text{best})$ be the probability that the best value is elected using the rule. We have

$$\begin{aligned}P_k(\text{best}) &= \sum_{i=1}^{n} P_k(\text{best}|X=i)p(X=i) \\ &= \frac{1}{n}\sum_{i=1}^{n} P_k(\text{best}|X=i)\end{aligned}$$

Because the largest will be selected if the largest of first k is also the largest of first $i-1$, we have

$$P_k(\text{best}|X=i) = \frac{k}{i-1}$$

So

$$\begin{aligned}P_k(best) &= \frac{1}{n}\sum_{i=1}^{n} P_k(\text{best}|X=i) \\ &= \frac{k}{n}\sum_{i=k+1}^{n}\frac{1}{k-1} \\ &\approx \frac{k}{n}\int_{k}^{n-1}\frac{1}{x}\,\mathrm{d}x \\ &= \frac{k}{n}\log\frac{n-1}{k}\end{aligned}$$

The best k is $\frac{1}{e}n$. □

Example 12 (The Ballot Problem)**.** *In an election candidate A received n notes and B received m notes where $n > m$. What is the probability that A is always ahead of B?*

Proof. Let $P_{n,m}$ denote the probability. By conditioning on who receive the last vote, we have:

$$P_{n,m} = \frac{n}{n+m}P_{n-1,m} + \frac{m}{n+m}P_{n,m-1}$$

The solution is $P_{n,m} = \dfrac{n-m}{n+m}$ □

Example 13. *A coin has p of being head. What is the probability that the total number of head equals tail after* 2*n flip?*

Proof.

$$\begin{aligned}P\{\text{first time } = 2n\} &= P\left\{\text{first time } = 2n | n \text{ heads in first } 2n\right\}\binom{2n}{n}p^n(1-p)^n \\ &= P_{n,n-1}\binom{2n}{n}p^n(1-p)^n \\ &= \frac{\binom{2n}{n}p^n(1-p)^n}{2n-1}\end{aligned}$$

□

Example 14. *What is the probability that the first time there are i more heads than tails occurs after the* $2n+i$ *flip?*

Proof. it occurs $\Longleftrightarrow$ starting from the final flip and working backwards, the head is always in the lead. □

Part II

Machine Learning

Chapter 5

Classic Machine Learning

5.1 Basics

For more information, please refer to [Aur19] Part 1.

5.1.1 Classification

In supervised learning the dataset is a collection of $\left\{(X_i, y_i)\right\}_{i=1}^{N}$. X is called feature vector, and y is called label. If y is discrete, it is classification. If y is continuous, it is called regression. In unsupervised learning the dataset is a collection of $\{X\}_{i=1}^{N}$.

Some useful definitions are:

1. batch learning:
   - use all available data.
   - cannot be modified after production launch.
   - offline learning
2. online learning:
   - sequentially take data inputs in mini-batch.
   - out-of-core learning: handle huge dataset that is too big for the machine memory.
   - learning rate: the speed of learning.
     - If it is too low, the system stop improvement.
     - If it is too high, sensitive to data noise. So need to monitor system performance and do early stop.

The result of learning could be measured using utility which is a positive measure, or a cost function which is a negative measure.

5.1.2 Prepare Data

5.1.2.1 Background

Very large samples can still be non-representative if the sampling method is flawed. It is called sampling error. The most famous one is in year 1936 US presidential election between Landon and Roosevelt: the Literary Digest poll. Literary Digest use phone directory to select the candidate, which favours wealthy people. Only 25% answered, so they may not care about politics, or do not like Literary Digest.

The process of selecting relevant feature is called feature engineering, which includes feature selection and feature extraction.

If the model overfit, try to simplify the model using fewer parameters or using regularization, or increase the training data size. If the model underfit, make the model more powerful, or reduce regularization.

5.1.2.2 Select Test Data

The data will be split into three disjoint sets: training set, validation set and test set. It is common to use 80% for training and 20% for test. However if the data set is very huge, 1% for testing is ok.

Test data selection process (stratified sampling):

1. divide all data into homogeneous subgroups (strata), such as by gender, income, etc.
2. select a random percentage as test data.

5.1.2.3 Data Cleaning

After test set is selected, the training set need to be cleaned. There are few useful data cleaning methods:

1. scaling:
   (a) min-max scaling: scale to $[0, 1]$.
   (b) standardization: scale to zero mean and unit variance.
2. string handling:
   (a) ordinal encoding: convert string to enumeration.
   (b) one-hot encoder: convert string to a 0-1 matrix.
3. NAN handling: replace NAN by median.
4. polynomial change: change number using polynomial function.

The same cleaning result need to be applied to validation and test set after the training is done.

If the category is too huge, the one-hot encoder may generate a wide matrix. In this case, an embedding to low dimentional vector could be learned using representation learning.

5.1.2.4 Training with Data

The training process now has these steps:

1. run different models on training set with different hyperparameter to train parameters.
2. run these models on validation set (or called development set, or dev set). Or using cross validation:
   (a) split the training data into k chunks.
   (b) run the model k times. Each time select one chunk as validation set and the remaining as training set.
   (c) take the average of validation error as the final error.
3. select best model and its hyperparameter based on validation error.
4. run the best model on $\{\text{training set} + \text{validation set}\}$ to train its parameters.
5. calculate generalization error based on test set.

If the training error is low on training set but the generalization error is high on test set, the model is overfitting.

5.1.3 Prediction Output Format

There are 4 different output formats:

1. binary: one output that is in $\{0,1\}$. such as SVM.
2. multiclass: only one output, but could be of n different values. There are two ways to train the model:
   (a) one-versus-the-rest (OvR): train n classifiers. Each classifier run against the rest values. Preferred for most classifiers.
   (b) one-versus-one (OvO): train $\binom{n}{2}$ classifiers. Each classifier run one value against another value. SVM is slow for large data set so usually prefer OvO.
3. multilabel: there are multiple categories of binary output result.
4. multioutput: there are multiple categories of multiclass output result.

5.1.4 Gradient Descent

5.1.4.1 Batch Gradient Descent

Gradient descent is used to minimize cost function. All features need to have a similar scale in order to increase converge speed.

In GD learning, η is the learning rate, θ is model's parameter vector, f (such as MSE) is the cost function. Formula (5.1) will try to minimize cost function recursively until $|\nabla_\theta f| < \varepsilon$ (ε is called tolerance)

$$\theta \leftarrow \theta - \eta \nabla_\theta f \tag{5.1}$$

Each iteration will use all input because the cost function is the sum of error over all inputs. So it is called batch and slow on very large data set. It takes $O\left(\frac{1}{\varepsilon}\right)$ iteration to reach the ε tolerance.

5.1.4.2 Stochastic Gradient Descent

A random instance is choose at every step to compute the gradient. The SGD has the chance of finding global minimum. The learning rate needs to be reduced gradually in order to let the training converge, which is called simulated annealing process with learning schedule. Each round of learning schedule is called epoch.

If instances are chosen randomly, there is chance that some instances are never chosen. One solution is to randomly sort the test set and iterate all samples, and then sort it again.

5.1.4.3 Mini-batch Gradient Decent

Mini-batch gradient descent is in the middle between SGD and GD. Each iteration it will take a small random set of instance from training set called mini-batch.

5.1.4.4 Early Stopping

It means stop training when the validation error reaches minimum. The validation error curve for gradient descent will decrease and then goes up which means the model starts to overfit the training data. One solution is to continue run for a while and roll back to previous minimum, which means the previous minimum needs to be saved. For mini-batch and stochastic GD, the curve is not smooth so the decision would be observe for sometime and then decide whether to rollback.

5.1.5 Learning Curve

Learning curve is a plot of error against training set size for both training set and validation set.

At the beginning the error for training set is low and gradually increases, and for validation set it is high and gradually decreases. The error of training set could be 0 even for the first few test samples. The gap between two sets could be decomposed into 3 components:

1. bias: the model makes wrong assumption about the data. usually underfit the training data.
2. variance: the model is sensitive to small variation to the training data. usually overfit the training data.
3. irreducible error: data is noisy. need to clean data.

Increasing the model complexity will typically increase the variance and reduce the bias, so it is a trade-off.

5.1.6 Error Measure

There are various distance measure available:

1. l_1 norm: $\frac{1}{m}\sum_{i=1}^{m}\left|h(X^{(i)})-y^{(i)}\right|$. Also called Manhattan norm, or mean absolute error, or MSE.
2. l_2 norm: $\sqrt{\frac{1}{m}\sum_{i=1}^{m}\left(h(X^{(i)})-y^{(i)}\right)^2}$. Also called root mean square error, or RMSE.
3. In general, the l_k norm is $\left\|X-y\right\|_k$.

Higher norm focuses on large number. So l_2 is sensitive to outliers. But if outlier is rare, the l_2 is preferred.

5.1.7 Regularization

5.1.7.1 Ridge Regression

Ridge Regression also called Tikhonov regularization or l_2 . It add l_2 norm to the cost function during training (θ_0 is not added):

$$\alpha\sum_{i=1}^{n}\theta_i^2 \tag{5.2}$$

The extra cost should only be added during training and removed in evaluation. It is common to use different cost function in training and testing. The training cost function needs to be optimization friendly.

Ridge regression is very sensitive to data scale, so do the data scaling before the training.

5.1.7.2 Lasso Regression

Lasso Regression (Least absolute shrinkage and selection operator regression) will add l_1 norm to the cost function (θ_0 is not added):

$$\alpha\sum_{i=1}^{n}\left|\theta_i\right| \tag{5.3}$$

Unimportant features will have 0 weight, so it will output a sparse model. Reduce the learning rate gradually to avoid bouncing around optimal.

5.1.7.3 Elastic Net

Elastic Net is the middle between ridge regression and Lasso regression. It will add the following to cost function:

$$\frac{1-r}{2}\alpha\sum_{i=1}^{n}\theta_i^2+r\alpha\sum_{i=1}^{n}\left|\theta_i\right| \tag{5.4}$$

Choose Elastic Net or Lasso Regression over Ridge Regression.

5.1.8 Tune Model

two different model tuning methods:

1. grid search: generate a combination of parameters and search
2. randomized search

5.1.9 Performance Measure

$$\begin{array}{ccc} & \multicolumn{2}{c}{\text{predicted}} \\ & 0 & 1 \\ \text{actual} \quad 0 & \text{TN} & \text{FP} \\ \phantom{\text{actual}} \quad 1 & \text{FN} & \text{TP} \end{array}$$

Figure 5.1: confusion matrix

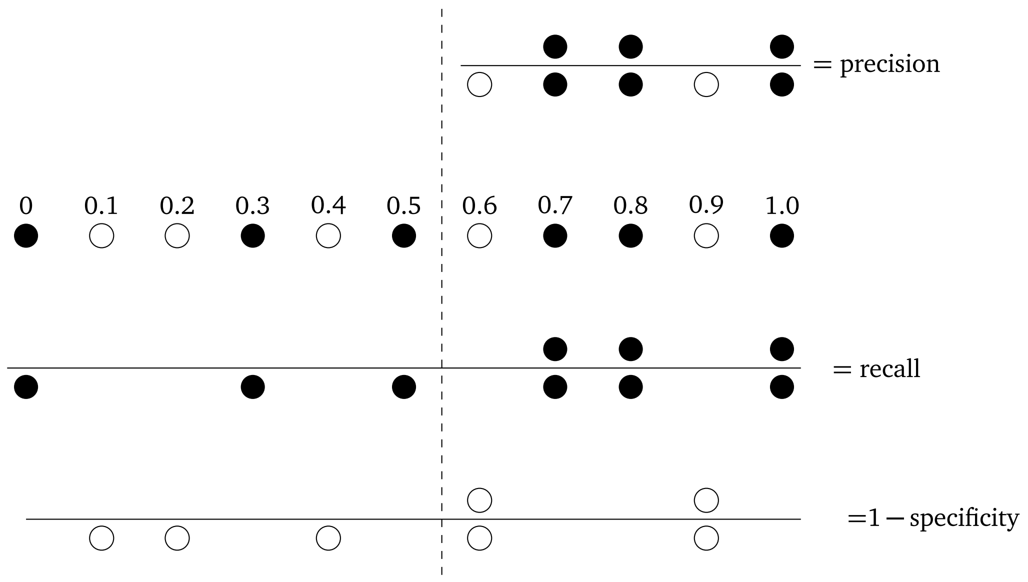

Figure 5.2: PR and ROC curve

$$\begin{aligned}
\text{precision} &= \frac{\text{TP}}{\text{TP}+\text{FP}} \\
\text{recall} &= \frac{\text{TP}}{\text{TP}+\text{FN}} \\
\text{false positive rate} &= \frac{\text{FP}}{\text{FP}+\text{TN}} \\
\text{F}_1 &= \frac{2}{\frac{1}{\text{precision}}+\frac{1}{\text{recall}}}
\end{aligned} \tag{5.5}$$

confusion matrix (PR curve) is a plot between precision and recall (precision is the y axis). There is a trade-off between precision and recall. Given a fixed model and prediction, when the threshold increases, the precision might increase (it could decrease) and the recall will always decrease. Sometimes you may care more about precision, such as classifying safe video for kids. Sometimes you care more about recall, such as the surveillance image classification. For two models, select the one that could embed the other PR curve.

receiver operating characteristic (ROC) is a plot between recall and $1-$specificity (recall is the y axis). The area under the curve (AUC) measures how good a model is. A random classifier will have AUC $= 0.5$. Compared with ROC curve, PR curve is preferred when:

1. positive rate is rare.
2. care more about false positive than false negative.

5.2 Classic Models

5.2.1 Linear Regression

The prediction is:

$$\hat{y} = \theta_0 + \sum_{i=1}^{n} \theta_n x_n = \boldsymbol{\theta}^\top \mathbf{X} \tag{5.6}$$

The cost function is:

$$\mathrm{J}(\boldsymbol{\theta}) = \mathrm{MSE} = \frac{1}{m} \sum_{i=1}^{m} \left(\boldsymbol{\theta}^\top \mathbf{X}^{(i)} - \mathbf{Y}^{(i)} \right)^2 \tag{5.7}$$

The close solution is:

$$\hat{\theta} = (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{Y} \tag{5.8}$$

The $\hat{\theta}$ is usually calculates using pseudoinverse of $\mathbf{X}$ as $\mathbf{X}^{+}\mathbf{Y}$ using SVD which has time complexity $\mathrm{O}(n^2)$.
The gradient vector of J is calculated as:

$$\begin{aligned} \frac{\partial \mathrm{J}(\theta)}{\partial \theta_j} &= \frac{2}{m} \sum_{i=1}^{m} \left(\boldsymbol{\theta}^\top \mathbf{X}^{(i)} - \mathbf{Y}^{(i)} \right) \mathbf{X}_j^{(i)} \\ \nabla_\theta \mathrm{J}(\theta) &= \frac{2}{m} \mathbf{X}^\top (\mathbf{X}\boldsymbol{\theta} - \mathbf{Y}) \end{aligned} \tag{5.9}$$

Linear regression model is convex, so it will be guaranteed to find global minimum using gradient descent.

5.2.2 Logistic Regression

logistic regression is a binary classifier. It uses the sigmoid function $\sigma(t) = \dfrac{1}{1+\exp(-t)}$ and assigns probability to $\mathbf{X}$ that $\hat{p} = \sigma(\mathbf{X}^\top \boldsymbol{\theta})$. The prediction is:

$$\hat{y} = \begin{cases} 0, \text{ if } \hat{p} < 0.5 \\ 1, \text{ if } \hat{p} \geq 0.5 \end{cases} \tag{5.10}$$

So the prediction is 1 if $\mathbf{X}^\top \boldsymbol{\theta} > 0$ and 0 if it is negative.
The log loss cost function is:

$$\mathrm{J}(\theta) = -\frac{1}{m} \sum_{i=1}^{m} \left\{ \mathbf{Y}^{(i)} \log \hat{\boldsymbol{p}}^{(i)} + \left(1 - \mathbf{Y}^{(i)}\right) \log \left(1 - \hat{\boldsymbol{p}}^{(i)}\right) \right\} \tag{5.11}$$

The derivative is:

$$\frac{\partial \mathrm{J}(\theta)}{\partial \theta_j} = \frac{1}{m} \sum_{i=1}^{m} \left(\sigma\left(\boldsymbol{\theta}^\top \mathbf{X}^{(i)} \right) - \mathbf{Y}^{(i)} \right) \mathbf{X}_j^{(i)} \tag{5.12}$$

5.2.3 Softmax Regression

Softmax Regression is a multiclass version of logistic regression. Each class has its own parameter vector $\boldsymbol{\theta}_i$. The softmax score for class k is $s_k(\mathbf{X}) = \mathbf{X}^\top \boldsymbol{\theta}_k$ and its probability is:

$$\hat{p}_k = \frac{\exp s_k(\mathbf{X})}{\sum_{i=1}^{n} \exp s_i(\mathbf{X})} \tag{5.13}$$

The prediction is:

$$\hat{y} = \underset{k}{\operatorname{argmax}}\, \mathbf{X}^\top \boldsymbol{\theta}_k \tag{5.14}$$

The cost function is measures using cross entropy function:

$$\mathrm{J}(\theta) = -\frac{1}{m} \sum_{i=1}^{m} \sum_{k=1}^{\mathrm{K}} \mathbf{Y}_k^{(i)} \log \hat{\boldsymbol{p}}_k^{(i)} \tag{5.15}$$

The gradient vector is:

$$\nabla_{\theta_k} \mathrm{J}(\theta) = \frac{1}{m} \sum_{i=1}^{m} (\hat{p}_k^{(i)} - y_k^{(i)}) \mathrm{X}^i \tag{5.16}$$

The cross entropy function is also called Kullback-Leibler divergence.

5.2.4 SVM

Support Vector Machine tries to separate the plane using two parallel lines. It maximize the distance between two lines. The points that these two lines touch is called support vector. The model is sensitive to feature scaling, so normalize the data before training. If all instances are on the opposite sides, it is called hard margin classification.

For each label i, $\mathbf{Y}^{(i)} = 1$ or $\mathbf{Y}^{(i)} = -1$.The hard margin classification is:

$$\begin{aligned} \text{minimize: } & \frac{1}{2}\mathbf{W}^\top \mathbf{W} \\ \text{subject to: } & \mathbf{Y}^{(i)}\left(\mathbf{W}^\top \mathbf{X}^{(i)} + b\right) \geq 1 \end{aligned} \tag{5.17}$$

Soft margin classification uses slack variable $\zeta^{(i)} \geq 0$ for each instance which measures how much margin violation is allowed. The optimization becomes:

$$\begin{aligned} \text{minimize: } & \frac{1}{2}\mathbf{W}^\top \mathbf{W} \\ \text{subject to: } & \mathbf{Y}^{(i)}\left(\mathbf{W}^\top \mathbf{X}^{(i)} + b\right) \geq 1 - \zeta^{(i)} \\ & \zeta^{(i)} \geq 0 \end{aligned} \tag{5.18}$$

5.2.5 Decision Trees

5.2.6 Ensemble Learning

A group of predictors is called ensemble.

5.2.6.1 Voting Classifiers

The class is chosen by majority vote, either by the majority number of votes called hard voting, or by the majority probability called soft voting.

It requires that all models are perfectly independent and have no correlated errors. It is not true because all models are trained on the same data.

5.2.6.2 Bagging and Pasting

It generate multiple classifier by using only one training algorithm but multiple training set. If each sample data is generated using replacement, it is called bagging. So a instance may appear multiple times in bagging. If each sample data is generated without replacement, it is called pasting. All predictors could be trained in parallel and good for scale up.

If it is classification, the result is chosen by majority vote. If it is regression, the result is averaged.

In bagging, it will choose $1-e^{-1} \approx 63.2\%$ samples. The remaining unchosen one is called out-of-bag instances which could be used as test test.

Bagging is more diverse so it has higher bias but lower variance. Overall bagging is better than pasting.

5.2.6.3 Random Patches and Random Subspaces

Random Patch will sample both training instances and model features. Random Subspace will run all training data but sample only model feature.

5.2.6.4 Random Forest and Extra-Trees

Random Forest is an ensemble of Decision Trees using bagging method (sometimes pasting). Extra Tree use random threshold for each Random Forest feature.

5.2.7 Boosting

Boosting trains multiple predictors sequentially, each trying to correct its predecessor. So it cannot be parallelized.

5.2.7.1 AdaBoost

Every instance is assigned a weight. The first algorithm trains on the data. The weight of misclassified instance will be increased. The second algorithm will run on the updated data set. Once all predictors are trained, the prediction is done by bagging or pasting except that the each predictor has different weight which is the average weight of its prediction.

5.2.7.2 Gradient Boosting

In gradient boosting there is a series of models F_i:

$$\begin{aligned} F_1 &\approx (X, Y) \\ F_2 &\approx \Big(X, Y - F_1(X)\Big) \\ &\dots \\ F_n &\approx \Big(X, Y - F_{n-1}(X)\Big) \end{aligned} \tag{5.19}$$

Theorem 219. *Gradient boosting is a gradient descent process.*

Proof. Suppose the loss function is:

$$\mathscr{L}(Y, F_k) = \frac{1}{2}\sum_i \Big(y_i - F_k(x_i)\Big)^2 \tag{5.20}$$

Suppose the aggregate function $\mathscr{F}_i = \sum_i F_i$, the derivative is:

$$\begin{aligned} \frac{\partial \mathscr{L}(Y, F_k)}{\partial F(x_i)} &= F_k(x_i) - y_i \\ y_i - F_k(x_i) &= -\frac{\partial \mathscr{L}(Y, F_k)}{\partial F_k(x_i)} \\ \mathscr{F}_2 &= F_1 + F_2 \\ &= F_1 + y_i - F_1 \\ &= F_1 - 1 \times \frac{\partial \mathscr{L}(Y, F_1)}{\partial F_1(x_i)} \\ &= \mathscr{F}_1 - 1 \times \frac{\partial \mathscr{L}(Y, \mathscr{F}_1)}{\partial \mathscr{F}_1(x_i)} \end{aligned} \tag{5.21}$$

□

5.2.7.3 Stacking

Instead of using soft or hard voting to aggregate the predictions, a model called blender is trained to perform the aggregation. The training set is split into two sets. The first set is used to train predictors. Next the predictors run on second set. The result of second set is used as the input to the blender.

Chapter 6

Deep Neural Network

6.1 Introduction to Artificial Neural Network

6.1.1 Autodiff

Autodiff is neither a numerical nor symbolic differentiation. It use numerical method calculation and use symbolic rules for differentiation, so it is partly symbolic and partly numerical. See [BPRS15] for more details.

One example is $f(x_1,x_2) = \log x_1 + x_1x_2 - \sin x_2$. The computation graph is:

$$\begin{tikzcd}
x_1 \arrow[r] & v_{-1} = x_1 \arrow[r] \arrow[dr] & v_1 = \log v_{-1} \arrow[r] & v_4 = v_1 + v_2 \arrow[dr] & & \\
& & v_2 = v_{-1}v_0 \arrow[ur] & & v_5 = v_4 + v_3 \arrow[r] & f(x_1,x_2) \\
x_2 \arrow[r] & v_0 = x_2 \arrow[ur] \arrow[rr] & & v_3 = \sin v_0 \arrow[ur] & &
\end{tikzcd}$$

Figure 6.1: $f(x_1,x_2) = \log x_1 + x_1x_2 - \sin x_2$

In general there are two autodiff methods: forward mode and backpropagation. In forward mode, the derivative is the intermediary nodes against all input nodes while in backpropagation the derivative is the final node against all intermediary nodes.

6.1.1.1 Forward Mode

There are $1+n$ passes in forward mode:

1. a forward evaluation of all node values. Table 6.1 shows the forward example.
2. n forward derivative calculation for all n input features. For the i-th input feature, the derivatives of all $\dot{v}$ is calculated in Table 6.2.

v_{-1}	$= x_1$	$= 2$
v_0	$= x_2$	$= 5$
v_1	$= \log v_{-1}$	$= \log 2$
v_2	$= v_{-1} \times v_0$	$= 2 \times 5$
v_3	$= \sin v_0$	$= \sin 5$
v_4	$= v_1 + v_2$	$= 0.693 + 10$
v_5	$= v_4 - v_3$	$= 10.693 + 0.959$
y	$= v_5$	$= 11.652$

Table 6.1: Forward Primal Trace

$\dot{v}_{-1}$	$= \dot{x}_1$	$= 1$
$\dot{v}_0$	$= \dot{x}_2$	$= 0$
$\dot{v}_1$	$= \frac{\dot{v}_{-1}}{v_{-1}}$	$= \frac{1}{2}$
$\dot{v}_2$	$= \dot{v}_{-1} \times v_0 + v_{-1} \times \dot{v}_0$	$= 1 \times 5 + 0 \times 2$
$\dot{v}_3$	$= \dot{v}_0 \times \cos v_0$	$= 0 \times \cos 5$
$\dot{v}_4$	$= \dot{v}_1 + \dot{v}_2$	$= 0.5 + 5$
$\dot{v}_5$	$= \dot{v}_4 - \dot{v}_3$	$= 5.5 - 0$
$\dot{y}$	$= \dot{v}_5$	$= 5.5$

Table 6.2: Forward Tangent (Derivative) Trace

Forward mode is expensive because it needs to run a pass for each input feature.

6.1.1.2 Backpropogation

Backpropagation only needs 2 passes over the calculation graph:

1. a forward evaluation of all node values. The same as forward mode. Table 6.1 shows the forward example.
2. a backward derivative calculation of all nodes in calculation graph. Table 6.3 shows the backward example.

In the backward derivative calculation, all derivatives are calculated against final result y. According to calculas, we have:

$$\frac{\partial y}{\partial v_0} = \frac{\partial y}{\partial v_2}\frac{\partial v_2}{\partial v_0} + \frac{\partial y}{\partial v_3}\frac{\partial v_3}{\partial v_0} \tag{6.1}$$

Let $\bar{v} = \frac{\partial y}{\partial v}$. We have:

$$\bar{v}_0 = \bar{v}_2\frac{\partial v_2}{\partial v_0} + \bar{v}_3\frac{\partial v_3}{\partial v_0} \tag{6.2}$$

So the final result could calculated by backward propagation. Table 6.3 gives an example.

$\bar{v}_5$	$= \bar{y}$		$= 1$
$\bar{v}_4$	$= \bar{v}_5 \frac{\partial v_5}{\partial v_4}$	$= \bar{v}_5 \times 1$	$= 1$
$\bar{v}_3$	$= \bar{v}_5 \frac{\partial v_5}{\partial v_3}$	$= \bar{v}_5 \times (-1)$	$= -1$
$\bar{v}_2$	$= \bar{v}_4 \frac{\partial v_4}{\partial v_2}$	$= \bar{v}_4 \times 1$	$= 1$
$\bar{v}_1$	$= \bar{v}_4 \frac{\partial v_4}{\partial v_1}$	$= \bar{v}_4 \times 1$	$= 1$
$\bar{v}_0$	$= \bar{v}_3 \frac{\partial v_3}{\partial v_0}$	$= \bar{v}_3 \times \cos v_0$	$= -0.284$
$\bar{v}_{-1}$	$= \bar{v}_2 \frac{\partial v_2}{\partial v_{-1}}$	$= \bar{v}_2 \times v_0$	$= 5$
$\bar{v}_0$	$= v_0 + \bar{v}_2 \frac{\partial v_2}{\partial v_0}$	$= \bar{v}_0 + \bar{v}_2 \times v_{-1}$	$= 1.716$
$\bar{v}_{-1}$	$= v_{-1} + \bar{v}_1 \frac{\partial v_1}{\partial v_{-1}}$	$= \bar{v}_{-1} + \frac{\bar{v}_1}{v_{-1}}$	$= 5.5$

Table 6.3: Reverse Adjoint (Derivative) Trace

6.2 Gradient Descent

6.2.1 Gradient Descent Classification

Gradient Descent is a numeric optimization method to calculate the minimum value of a function. For a give function $f(x)$ with gradient everywhere, if we choose α small enough, it is possible that $f(x') < f(x)$ where:

$$x' = x - \alpha \nabla_x f \tag{6.3}$$

α is called learning rate. Please be noted that the x here is a vector of all input samples.

Batch Gradient Descent calculate the gradient of the cost function for the entire training set. So it is slow, and sometimes cannot be fit into the main memory.

Stochastic Gradient Descent calculate the gradient for every single training sample. So it fluctuate.

Mini-batch Gradient Descent calculate the gradient for every batch of n samples. It could use GPU to accelerate the training, and it is less volatile than stochastic gradient descent.

So mini-batch gradient descent is the best of all choices. However it still has the following limitations:

1. The learning rate cannot be too small or too big.
2. learning rate schedule is used to gradually reduce the learning rate, which mimics the annealing. However the training data may not match the learning schedule.
3. One learning rate is applied to all data. The data could be sparse or dense.
4. Local minimum is usually not a problem. The real problem is the plateau around saddle points. The gradient is flat and takes long time to move away.

So optimization algorithms is needed for an efficient gradient descent calculation.

6.2.2 Gradient Descent Optimization Algorithms

The idea of optimizing gradient descent is to change $\alpha \nabla_x f$ in Equation (6.3). We can change all of them, such as in Momentum and NAG algorithm, or change α, such as in Adagrad, Adadelta, RMSprop, or change $\nabla_x f$, such as in Adam, AdaMax, Nadam. The popular optimization algorithms are summarized in [Rud16]. The relationship among all these optimization algorithm is in Figure 6.2.

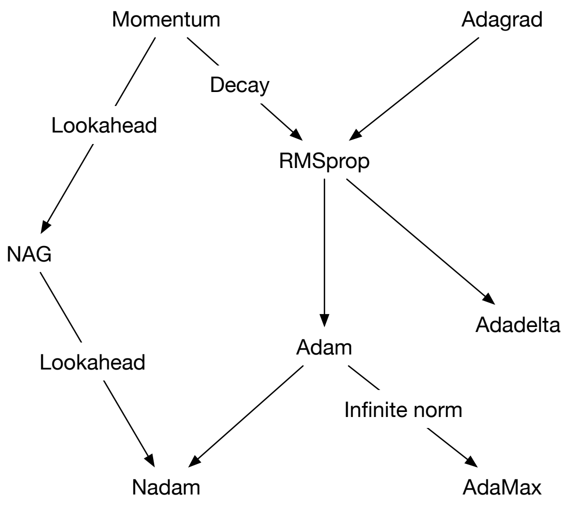

Figure 6.2: Relation among Gradient Descent Algorithms

6.2.2.1 Momentum

Momentum [Qia99] takes a weighted average of all gradient:

$$\begin{aligned}\theta_{t+1} &= \theta_t - m_t \\ m_t &= \beta m_{t-1} + (1-\beta)\nabla_{\theta_t} J(\theta_t)\end{aligned} \tag{6.4}$$

$\alpha = 0.9$ for most of the cases. The momentum term increases if gradients point in the same direction and reduces updates when gradients change direction. So it has momentum when moving towards the same direction, and it converges fast.

6.2.2.2 NAG

Nesterov Accelerated Gradient [NES83], or NAG, is a look ahead calculation. Since θ_t will be reduced by a gradient every time: $\theta_{t+2} = \theta_{t+1} - m_{t+1} = \theta_t - m_t - m_{t+1}$, it replaces $\nabla_{\theta_t} J(\theta_t)$ by $\nabla_{\theta_t} J(\theta_t - \beta m_{t-1})$:

$$\begin{aligned}\theta_{t+1} &= \theta_t - m_t \\ m_t &= \beta m_{t-1} + (1-\beta)\nabla_{\theta_t} J(\theta_t - \beta m_{t-1})\end{aligned} \tag{6.5}$$

6.2.2.3 Adagrad

Adagrad [DBW12] performs large update for infrequent and small update for frequent parameters:

$$\begin{aligned}\theta_{t+1} &= \theta_t - \frac{\alpha}{\sqrt{G_t + \epsilon}} \nabla_{\theta_t} J(\theta_t) \\ G_t &= G_{t-1} + \left(\nabla_{\theta_t} J(\theta_t)\right)^2\end{aligned} \tag{6.6}$$

ϵ is added to avoid division by zero. Usually $G_0 = 0$, $\alpha = 0.01$, $\epsilon = 10^{-8}$. Because G_t keeps accumulating, eventually it will become too big for the algorithm to be effective.

6.2.2.4 Adadelta

Adadelta [Zei12] fixes the problem of Adagrad. It is a decay version of Adagrad:

$$\begin{aligned}\theta_{t+1} &= \theta_t - \frac{\sqrt{D_{t-1} + \epsilon}}{\sqrt{G_t + \epsilon}} \nabla_{\theta_t} J(\theta_t) \\ G_t &= \beta G_{t-1} + (1-\beta)\left(\nabla_{\theta_t} J(\theta_t)\right)^2 \\ D_t &= \beta D_{t-1} + (1-\beta)(\theta_t - \theta_{t-1})^2\end{aligned} \tag{6.7}$$

Usually $G_0 = 0$, $D_0 = 0$, $\beta = 0.9$, $\epsilon = 10^{-6}$.

6.2.2.5 RMSprop

RMSprop is unpublished. It is a simplified version of Adadelta that $D_t = 0$:

$$\begin{aligned}\theta_{t+1} &= \theta_t - \frac{\alpha}{\sqrt{G_t + \epsilon}} \nabla_{\theta_t} J(\theta_t) \\ G_t &= \beta G_{t-1} + (1-\beta)\left(\nabla_{\theta_t} J(\theta_t)\right)^2\end{aligned} \tag{6.8}$$

Usually $G_0 = 0$, $\alpha = 0.001$, $\beta = 0.9$, $\epsilon = 10^{-6}$.

6.2.2.6 Adam

Adaptive Moment Estimation [KB15], or Adam, is yet another adaptive method:

$$\begin{aligned}\theta_{t+1} &= \theta_t - \frac{\alpha}{\sqrt{\frac{G_t}{1-(\beta_2)^t} + \epsilon}} \frac{m_t}{1-(\beta_1)^t} \\ m_t &= \beta_1 m_{t-1} + (1-\beta_1)\nabla_{\theta_t} J(\theta_t) \\ G_t &= \beta_2 G_{t-1} + (1-\beta_2)\left(\nabla_{\theta_t} J(\theta_t)\right)^2\end{aligned} \tag{6.9}$$

Usually $G_0 = 0$, $\alpha = 0.001$, $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 10^{-8}$.

6.2.2.7 AdaMax

AdaMax [KB15] is a infinite norm version of Adam. Adam use l_2 norm for G. For infinite norm:

$$\begin{aligned}G_t &= \beta_2^\infty G_{t-1} + (1-\beta_2^\infty)\left\|\nabla_{\theta_t} J(\theta_t)\right\|^\infty \\ &= \max\left(\beta_2 G_{t-1}, \left\|\nabla_{\theta_t} J(\theta_t)\right\|\right)\end{aligned} \tag{6.10}$$

Because $G_t > 0$ in AdaMax, ϵ is no longer useful. So the equation could be simplified:

$$\begin{aligned}\theta_{t+1} &= \theta_t - \frac{\alpha}{G_t} m_t \\ m_t &= \beta_1 m_{t-1} + (1-\beta_1)\nabla_{\theta_t} J(\theta_t) \\ G_t &= \max\left(\beta_2 G_{t-1}, \left\|\nabla_{\theta_t} J(\theta_t)\right\|\right)\end{aligned} \tag{6.11}$$

Usually $G_0 = 0$, $\alpha = 0.001$, $\beta_1 = 0.9$, $\beta_2 = 0.999$.

6.2.2.8 Nadam

Nadam [Doz16] combines Adam and NAG. Adam could be written as:

$$\theta_{t+1} = \theta_t - \frac{\alpha}{\sqrt{\frac{G_t}{1-(\beta_2)^t} + \epsilon}} \left(\beta_1 \frac{m_{t-1}}{1-(\beta_1)^{t-1}} + \frac{1-\beta_1}{1-(\beta_1)^t} \nabla_{\theta_t} J(\theta_t)\right) \tag{6.12}$$

Like NAG, Nadam changes Adam's $t-1$ version of $\frac{m_{t-1}}{1-(\beta_1)^{t-1}}$ to current t version $\frac{m_t}{1-(\beta_1)^t}$:

$$\begin{aligned}\theta_{t+1} &= \theta_t - \frac{\alpha}{\sqrt{\frac{G_t}{1-(\beta_2)^t} + \epsilon}} \left(\beta_1 \frac{m}{1-(\beta_1)^t} + \frac{1-\beta_1}{1-(\beta_1)^t} \nabla_{\theta_t} J(\theta_t)\right) \\ m_t &= \beta_1 m_{t-1} + (1-\beta_1)\nabla_{\theta_t} J(\theta_t) \\ G_t &= \beta_2 G_{t-1} + (1-\beta_2)\left(\nabla_{\theta_t} J(\theta_t)\right)^2\end{aligned} \tag{6.13}$$

Usually $G_0 = 0$, $m_0 = 0$, $\alpha = 0.002$, $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 10^{-7}$.

6.3 Training Deep Neural Network

Chapter 7

Reinforement Learning

7.1 Background

7.1.1 State

environment has state S_t^e, which receives action A_t and emits observation O_{t+1} and scalar reward R_{t+1}.

state is a function of all history, so the reinforcement learning could be a Markov process:

$$S_t = f([O_1, R_1, A_1, \dots, A_{t-1}, O_t, R_t]) \tag{7.1}$$

7.1.2 Planning and Learning

7.1.2.1 Planning

planning uses simulated experience from model. So the model is known.

There are two different planning methods:

state-space planning searches through state space.

plan-space planning searches through plans. It uses evolutionary methods. Seldom used in reinforcement learning.

7.1.2.2 Learning

learning uses real experience of environment. So the environment and policy are unknown.

7.1.3 Agent

agent has three important components:

- policy
- value function
- model

7.1.3.1 Policy

policy can be deterministic or statistics :

deterministic $a = \pi(s)$
stochastic $\pi(a|s) = \mathbb{P}[A_t = a | S_t = s]$

7.1.3.2 Model

model is anything an agent can use to predict environment response. so a model simulate environment.

distribution model explores all possibility and all probability
sample model explores only one possibility

distribution model is stronger than sample model because it can always generate sample. However in practice it is much easier to obtain sample models.

7.1.4 Evaluation and Control

evaluation tries to calculate $v(s)$ or $q(s,a)$.
control tries to calculate $v_*(s)$, $q_*(s,a)$ or π_*.

7.1.5 Exploration and Exploitation

exploration find more information about environment.
exploitation exploit known information to maximize reward.

7.1.6 Incremental Mean

If Q_{n+1} is the mean of R_i: $Q_{n+1} = \frac{1}{n}\sum_{i=1}^{n} R_i$. Q_{n+1} could be rearrange as:

$$Q_{n+1} = Q_n + \frac{1}{n}\Big(R_n - Q_n\Big) \tag{7.2}$$

If we replace $\frac{1}{n}$ by α, the formula becomes:

$$\begin{aligned} Q_{n+1} &= Q_n + \alpha\Big(R_n - Q_n\Big) \\ &= (1-\alpha)^n Q_1 + \sum_{i=1}^{n} \alpha(1-\alpha)^{n-1} R_i \end{aligned} \tag{7.3}$$

Q_n will be convergent if α follows the following formula:

$$\begin{cases} \sum\limits_{n=1}^{\infty} \alpha_n = \infty \\ \\ \sum\limits_{n=1}^{\infty} \alpha_n^2 < \infty \end{cases} \tag{7.4}$$

7.1.7 Terminology

backup uses future return to update current value.

7.2 Markov Decision Process

7.2.1 Definition

A Markov decision process (MDP) is a Markov reward process with decisions. It is an environment in which all stated are Markov. It is a tuple $\langle \mathscr{S}, \mathscr{A}, \mathscr{R}, \mathscr{P}, \gamma \rangle$:

- $\mathscr{S}$ is a finite set of states.
- $\mathscr{A}$ is a finite set of actions.
- $\mathscr{R}$ is a reward function.
- $\mathscr{P}$ is a state transition probability matrix.
- $\gamma \in [0, 1]$ is a discount factor.

7.2.2 Goals and Equations

7.2.2.1 Environment

environment has state $S_t \in \mathscr{S}$ and generate reward $R_{t+1} \in \mathbb{R}$. Anything that cannot be changed arbitrarily by the agent is part of environment. The goal of reinforcement learning is to maximize expected value of cumulative sum of received scalar reward.

The reward signal should be chosen so it will not affect how agent act.

7.2.2.2 Agent

agent has action $A_t \in \mathscr{A}(S_t)$ and observation O_t of state S_t. Agent may know everything about how the environment works but still unable to solve problem, such as the Rubik cube puzzle.

7.2.2.3 State and Reward

The probability of next state and reward is :

$$p(s', r|s, a) = \mathbb{P}\{S_t = s', R_t = r|S_{t-1} = s, A_{t-1} = a\} \tag{7.5}$$

The state-transition probabilities is :

$$p(s'|s, a) = P\{S_t = s'|S_{t-1} = s, A_{t-1} = a\} = \sum_{r \in \mathbb{R}} p(s', r|s, a) \tag{7.6}$$

Usually the state and reward probability will be treated as independent, so their formulas are:

$$\mathscr{P}^a_{s,s'} = \mathbb{P}[S_{t+1} = s'|S_t = s, A_t = a] \tag{7.7}$$

$$\mathscr{P}^\pi_{s,s'} = \sum_{a \in A(s)} \pi(a|s)\mathscr{P}^a_{s,s'} \tag{7.8}$$

$$\mathscr{R}^a_s = \mathbb{E}[R_{t+1}|S_t = s, A_t = a] \tag{7.9}$$

$$\mathscr{R}^\pi_s = \sum_{a \in A(s)} \pi(a|s)\mathscr{R}^a_s \tag{7.10}$$

The expected reward of state-action pair is:

$$r(s, a) = \mathbb{E}[R_t|S_{t-1} = s, A_{t-1} = a] = \sum_{r \in \mathbb{R}} \sum_{s' \in S} p(s', r|s, a) \tag{7.11}$$

7.2.2.4 Episode

An episode is an order sequence of $S_0, A_0, R_1, S_1, A_1, \ldots, R_n, S_n$ in MDP. So an episode will always end.

continuing task is a list of actions that never terminate.

Episode task can be converted to continuing task by appending infinite absorbing state.

7.2.2.5 Goal

The expected return is the sum of rewards in the episode:

$$G_t = R_{t+1} + R_{t+2} + R_{t+3} + \cdots + R_T \tag{7.12}$$

- terminal state: R_T
- S: All non-terminal states
- S^+: all terminal and non-terminal states

discounted return for continuing task is defined as:

$$\begin{aligned} G_t &= R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \dots \\ &= \sum_{k=0}^{\infty} \gamma^k R_{t+k_1} \\ &= R_{t+1} + \gamma G_{t+1} \end{aligned} \tag{7.13}$$

where $0 \le \gamma \le 1$.

7.2.2.6 Policy

A stochastic policy is a probability of selecting next possible action:

$$\pi(a|s) = \mathbb{P}[A_t = a | S_t = s] \tag{7.14}$$

7.2.2.7 Value Function

The state-value function V_π for policy π is:

$$\begin{aligned} V_\pi(s) &= \mathbb{E}_\pi[G_t | S_t = s] \\ &= \mathbb{E}[R_{t+1} + \gamma V_\pi(S_{t+1}) | S_t = s] \end{aligned} \tag{7.15}$$

The action-value function q_π for policy π is:

$$\begin{aligned} q_\pi(s,a) &= \mathbb{E}_\pi[G_t | S_t = s, A_t = a] \\ &= \mathbb{E}[R_{t+1} + \gamma q_\pi(S_{t+1}, A_{t+1}) | S_t = s, A_t = a] \end{aligned} \tag{7.16}$$

7.2.2.8 Bellman Equation

Bellman equation (backup process) is an expansion of value function:

$$\begin{aligned} V_\pi(s) &= \sum_{a \in \mathscr{A}(s)} \pi(a|s) q_\pi(s,a) \\ &= \sum_{a \in \mathscr{A}(s)} \pi(a|s) \left(\mathscr{R}_s^a + \gamma \sum_{s' \in \mathscr{S}} \mathscr{P}_{s,s'}^a V(S') \right) \end{aligned} \tag{7.17}$$

$$\begin{aligned} q_\pi(s,a) &= R_s^a + \gamma \sum_{s' \in \mathscr{S}} \mathscr{P}_{s,s'}^a V_\pi(s') \\ &= R_s^a + \gamma \sum_{s' \in \mathscr{S}} \mathscr{P}_{s,s'}^a \left(\sum_{a',s'} \pi(a'|s') q_\pi(s',a') \right) \end{aligned} \tag{7.18}$$

7.2.2.9 Bellman Equation Solution

Formula (7.17) is for a single state. Let

$$V_\pi = \begin{bmatrix} V_\pi(s_1) \\ \vdots \\ V_\pi(s_n) \end{bmatrix} \tag{7.19}$$

Formula (7.17) now becomes:

$$V_\pi = R^\pi + \gamma P^\pi V_\pi \tag{7.20}$$

So the solution to Bellman Equation is:

$$\mathrm{V}_\pi = (\mathrm{I} - \gamma \mathrm{P}^\pi)^{-1} \mathrm{R}^\pi \tag{7.21}$$

It is a fixed point solution to formula (7.17).

7.2.3 Optimal Policy

7.2.3.1 Policy Partial Order

$\pi \geq \pi'$ if $\forall s \in \mathcal{S}, \mathrm{V}_\pi(s) \geq \mathrm{V}_{\pi'}(s)$.

For MDP all optimal solution share the same value function V_* and include at least one deterministic policy π_*.

7.2.3.2 Optimal State-value Function

The optimal state-value function V_* is defined as:

$$\begin{aligned} \mathrm{V}_*(s) &= \max_\pi \mathrm{V}_\pi(s) \\ &= \max_a q_*(s,a) \\ &= \max_a \left(\mathcal{R}_s^a + \gamma \sum_{s' \in \mathcal{S}} \mathcal{P}_{s,s'}^a \mathrm{V}_*(s') \right) \end{aligned} \tag{7.22}$$

7.2.3.3 Optimal Action-value Function

The optimal action-value function q_* is defined as:

$$\begin{aligned} q_*(s,a) &= \max_\pi q_\pi(s,a) \\ &= \mathbb{E}[\mathrm{R}_{t+1} + \gamma \mathrm{V}_*(\mathrm{S}_{t+1}) | \mathrm{S}_t = s, \mathrm{A}_t = a] \\ &= \mathcal{R}_s^a + \gamma \sum_{s' \in \mathcal{S}} \mathcal{P}_{s,s'}^a \mathrm{V}_*(s') \\ &= \mathcal{R}_s^a + \gamma \sum_{s' \in \mathcal{S}} \mathcal{P}_{s,s'}^a \max_{a'} q_*(s',a') \end{aligned} \tag{7.23}$$

Here $\mathcal{R}_s^a$ is an expectation. In model-free learning and controlling it is the sample return from environment.

7.2.3.4 Optimal Policy from Action Value Function

Once the optimal action value function is known, we can calculate the optimal policy by:

$$\pi_*(a|s) = \begin{cases} 1 & \text{, if } a = \operatorname*{argmax}_{a \in \mathcal{A}(s)} q_*(s,a) \\ 0 & \text{, else} \end{cases} \tag{7.24}$$

So the optimal policy is a greedy algorithm.

Compared with V_*, q_* is better because it does not need to do one step lookahead.

7.2.3.5 Reason for Complex Algorithms

There is no closed form for optimal Bellman policy equation, so many iterative solution exists:

- value iteration
- policy iteration
- Q-learning
- Sarsa

7.3 Dynamic Programming

DP are effective for medium size problems (million of states).

7.3.1 Policy Evaluation (Prediction)

If P and π are known, Bellman equation (7.17) could be converted to iterative solution. All V are randomly initialized and updated using iterative policy evaluation:

$$V_{k+1}(s) = \sum_a \pi(a|s)\left(\sum_{s',r} p(s',r|s,a)\Big(r+\gamma V_k(s')\Big)\right) \tag{7.25}$$

Here $V_{t+1}(s)$ means the value of $V(s)$ in $(t+1)$ round.
There are two ways to update $V_{t+1}(s)$:

- copy $V_{t+1}(s)$ to a new array and update original array when sweeping is done
- in-place update: update $V(s)$ on the fly. updated value may be used immediately so it is faster than two array solution.

The problem is that iterative algorithm sweep through all state space, which might not be practical.
See Algorithm (7.3.1) for detail.

Algorithm 7.3.1 Iterative policy evaluation, estimate V_π

```
1:  procedure $(\pi, p, \theta)$
2:      $\forall s \in S, V(s) \leftarrow random$
3:      $V(terminal) \leftarrow 0$
4:      repeat
5:          $\Delta \leftarrow 0$
6:          for $s \in S$ do
7:              $v \leftarrow V(s)$
8:                                                        ▷ in-place update
9:              $V(s) \leftarrow \sum_a \pi(a|s)\left(\sum_{s',r} p(s',r|s,a)\big(r+\gamma V(s')\big)\right)$
10:             $\Delta \leftarrow \max(v, |v - V(s)|)$
11:         end for
12:     until $\Delta < \theta$
13: end procedure
```

7.3.2 Policy Improvement

The reason for calculating value function is to help find a better policy. A new greedy policy π' could be calculated using:

$$\begin{aligned} \pi'(s) &= \underset{a\in\mathscr{A}(s)}{\operatorname{argmax}}\, q_\pi(s,a) \\ &= \underset{a\in\mathscr{A}(s)}{\operatorname{argmax}} \sum_{s',r} p(s',r|s,a)\Big(r+\gamma V(s')\Big) \end{aligned} \tag{7.26}$$

A series of policy evaluation and improvement will converge to optimal result, and its conversion is very fast.
The drawback is that every iteration may trigger evaluation, which involves multiple sweep through all state space.
See Algorithm (7.3.2) for detail.

7.3.3 Value Iteration

In policy iteration, multiple sweep can be reduce to one by taking the best action, and calculate optimal policy using (7.24):

$$\begin{aligned} V^{old}_{k+1}(s) &= \sum_{s',r} p(s',r|s,\pi(s))\big(r+\gamma V_k(s')\big) \\ V^{new}_{k+1}(s) &= \max_a \sum_{s',r} p(s',r|s,a)\big(r+\gamma V_k(s')\big) \end{aligned} \tag{7.27}$$

Algorithm 7.3.2 Policy Iteration, estimate V_* and π_*

```
1: $\forall s \in \mathscr{S}, V(s) \leftarrow random$
2: $V(terminal) \leftarrow 0$
3: $\pi(s) \leftarrow \text{random}(\mathscr{A}(s))$

4: procedure PolicyEvaluation($\varepsilon$)
5:     repeat
6:         $\Delta \leftarrow 0$
7:         for $s \in \mathscr{S}$ do
8:             $v \leftarrow V(s)$
9:             $V(s) \leftarrow \sum_{s',r} p(s', r|s, \pi(s))\big(r + \gamma V(s')\big)$        ▷ $\pi(s)$ : use optimal policy
10:            $\Delta \leftarrow \max(\Delta, |v - V(s)|)$
11:        end for
12:    until $\Delta < \varepsilon$
13: end procedure

14: procedure PolicyImprovement
15:     stable ← TRUE
16:     for $s \in \mathscr{S}$ do
17:         $old \leftarrow \pi(s)$
18:         $\pi(s) \leftarrow \operatorname{argmax}_{a \in \mathscr{A}(s)} \sum_{s',r} P(s', r|s, a)\big(r + \gamma V(s')\big)$
19:         if $old \neq \pi(s)$ then
20:             stable ← FALSE
21:         end if
22:     end for
23:     if stable then return $(V_*, \pi_*)$
24:     else
25:         PolicyEvaluation($\varepsilon$)        ▷ update V if optimal policy has changed
26:     end if
27: end procedure
```

See Algorithm (7.3.3) for detail.

Algorithm 7.3.3 Value Iteration, estimate π_*

```
1: $\forall s \in \mathrm{S}, \mathrm{V}(s) \leftarrow random$
2: $\mathrm{V}(terminal) \leftarrow 0$

3: repeat
4:     $\Delta \leftarrow 0$
5:     for $s \in \mathrm{S}$ do
6:         $v \leftarrow \mathrm{V}(s)$
7:                                              ▷ policy evaluation and improvement
8:         $\mathrm{V}(s) \leftarrow \max_a \sum_{s',r} p(s', r|s, a)\big(r + \gamma \mathrm{V}(s')\big)$
9:         $\Delta \leftarrow \max(\Delta, |v - \mathrm{V}(s)|)$
10:    end for
11: until $\Delta < \theta$
12: return $\pi_*(s) = \operatorname{argmax}_a \sum_{s',r} p(s', r|s, a)\big(r + \gamma \mathrm{V}(s')\big)$
```

7.3.4 Generalized Policy Iteration

Generalized Policy Iteration (GPI) is a series of evaluation and improvement process. Almost all reinforcement learning methods are GPI.

7.3.5 Performance

DP is exponentially faster than direct policy space search.

DP is better than linear programming methods for large problem, but worse for small problem.

The curse of dimension is not the problem of algorithm but the problem itself.

The time complexity for v is $O(mn^2)$ and for q is $O(m^2n^2)$, where m is the number of action and n is the number of state. So DP is effective for medium size problems (million of states).

7.3.6 Extension to Sweeping

7.3.6.1 Prioritized Sweeping

backup the state with the maximum Bellman error:

$$\left| \max_{a \in \mathcal{A}} \left(\mathcal{R} + \gamma \sum_{s' \in \mathrm{S}} \mathcal{P}_{ss'}^{a} v(s') \right) - v(s) \right| \tag{7.28}$$

7.3.6.2 Realtime Dynamic Programming

Choose the state that are relevant to agent.

7.4 Monte Carlo Methods

7.4.1 Requirement

- need episode, so need to terminate
- start only when episode ends
- do not need model

Note: Monte Carlo Method is not a online method because it is not step-by-step method.

7.4.2 Prediction

7.4.2.1 First Visit MC

Monte Carlo prediction has first visit and every visit methods.

The first visit Algorithm (7.4.1) is an unbiased estimate.

Algorithm 7.4.1 First visit MC, estimate v_π

```
1: $\forall s \in \mathcal{S}, V(s) \leftarrow random$
2: Returns($s$) $\leftarrow$ []

3: loop
4:     generate an episode $\pi : S_0, A_0, R_1, S_1, A_1, R_2, \ldots, S_{T-1}, A_{T-1}, R_T$
5:     $G \leftarrow 0$
6:     for $t \leftarrow T-1, T-2, \ldots, 0$ do
7:         $G \leftarrow \gamma G + R_{t+1}$
                                                  ▷ loop backward to find first appearance
8:         if $S_t \notin \{S_0, S_1, \ldots, S_{t-1}\}$ then
9:             Returns($S_t$) $\leftarrow$ Returns($S_t$) + G
10:            $V(S_t) \leftarrow$ average(Returns($S_t$))
11:        end if
12:    end for
13: end loop
```

7.4.3 Explore All State-Action Pair

In Monte Carlo control, the policy at S_t need to explore all $q(S_t, A_t)$, which means the MC algorithm needs to cover all $\langle S_t, A_t \rangle$ pairs. There are two ways to achieve this:

- exploring start. It might not be possible to enumerate all start.
- ε-soft policy.

7.4.3.1 Exploring Starts

It tries all $\langle S_t, A_t \rangle$ pairs as the first action. See Algorithm (7.4.2) for detail.

If model is not available, q_π is preferred than V_π because it does not need transition probability when calculating optimal policy.

7.4.3.2 ε-soft Policy

In ε-soft policy, all $\langle S_t, A_t \rangle$ pairs are tried in the middle with non-zero probability:

- ε-soft : $\pi(a|s) > \dfrac{\varepsilon}{\mathcal{A}(s)}$.
- ε-greedy : see Algorithm (7.4.3) for detail.

$$\pi(a|s) = \begin{cases} \dfrac{\varepsilon}{\mathcal{A}(s)} & \text{, if } a \text{ is not the greedy choice} \\ 1 - \varepsilon + \dfrac{\varepsilon}{\mathcal{A}(s)} & \text{, if } a \text{ is the greedy choice} \end{cases}$$

Algorithm 7.4.2 first visit MCES (Exploring Starts), estimate π_*

```
1: $\pi(s) \in \mathscr{A}(s)$
2: $q(s,a) \in \mathbb{R}$
3: Returns$(s,a) \leftarrow [\,]$

4: loop
5:     choose $S_0$ and $A_0$ so all pairs will appear                ▷ exploring starts
6:     generate episode from $\langle S_0, A_0 \rangle$: $\pi : S_0, A_0, R_1, \dots, S_{T-1}, A_{T-1}, R_T$
7:     $G \leftarrow 0$
8:     for $t \leftarrow T-1, T-2, \dots, 0$ do
9:         $G \leftarrow \gamma G + R_{t+1}$
10:        if $\langle S_t, A_t \rangle \notin \{\langle S_0, A_0 \rangle, \langle S_1, A_1 \rangle, \dots, \langle S_{t-1}, A_{t-1} \rangle\}$ then
11:            Returns$(S_t, A_t) \leftarrow$ Returns$(S_t, A_t) + G$
12:            $q(S_t, A_t) \leftarrow$ average(Returns$(S_t, A_t)$)
13:            $\pi(S_t) \leftarrow \underset{a \in \mathscr{A}(S_t)}{\operatorname{argmax}}\, q(S_t, a)$
14:        end if
15:    end for
16: end loop
```

Algorithm 7.4.3 first visit MC control (ε-greedy), estimate π_*

```
1: $\pi \leftarrow$ random $\varepsilon$-greedy policy
2: $q(s,a) \in \mathbb{R}$
3: Returns$(s,a) \leftarrow [\,]$

4: loop
5:     choose $S_0$ and $A_0$                ▷ non-exploring start
6:     generate episode from $\langle S_0, A_0 \rangle$: $\pi : S_0, A_0, R_1, \dots, S_{T-1}, A_{T-1}, R_T$
7:     $G \leftarrow 0$
8:     for $t \leftarrow T-1, T-2, \dots, 0$ do
9:         $G \leftarrow \gamma G + R_{t+1}$
10:        if $\langle S_t, A_t \rangle \notin \{\langle S_0, A_0 \rangle, \langle S_1, A_1 \rangle, \dots, \langle S_{t-1}, A_{t-1} \rangle\}$ then
11:            Returns$(S_t, A_t) \leftarrow$ Returns$(S_t, A_t) + G$
12:            $q(S_t, A_t) \leftarrow$ average(Returns$(S_t, A_t)$)
13:            $A^* \leftarrow \underset{a \in \mathscr{A}(S_t)}{\operatorname{argmax}}\, q(S_t, a)$
14:            for $a \in \mathscr{A}(S_t)$ do                ▷ $\varepsilon$-greedy
15:                $$\pi(a|S_t) \leftarrow \begin{cases} 1 - \varepsilon + \dfrac{\varepsilon}{\mathscr{A}(s)} & ,\text{if } a = A^* \\ \dfrac{\varepsilon}{\mathscr{A}(s)} & ,\text{if } a \neq A^* \end{cases}$$
16:            end for
17:        end if
18:    end for
19: end loop
```

7.4.4 Off-policy prediction via Importance Sampling

7.4.4.1 Target Policy

behavior policy the policy b used to generate behavior
target policy the policy π being learned

It has the assumption of coverage:

$$\pi(a|s) > 0 \Rightarrow b(a|s) > 0 \tag{7.29}$$

7.4.4.2 Importance Sampling

The probability of $\{A_t, S_{t+1}, A_{t+1}, \dots, S_T\}$ under policy π is (using Monte Carlo property):

$$\mathbb{P}\{A_t, S_{t+1}, A_{t+1}, \dots, S_T | S_t, A_{t:T-1} \sim \pi\} = \prod_{k=t}^{T-1} \pi(A_k|S_k) p(S_{k+1}|S_k, A_k) \tag{7.30}$$

The importance-sampling ratio is:

$$\rho_{t:T-1}^{\pi/b} = \frac{\prod_{k=t}^{T-1} \pi(A_k|S_k) p(S_{k+1}|S_k, A_k)}{\prod_{k=t}^{T-1} b(A_k|S_k) p(S_{k+1}|S_k, A_k)} = \prod_{k=t}^{T-1} \frac{\pi(A_k|S_k)}{b(A_k|S_k)} \tag{7.31}$$

Let $\mathscr{T}$ denotes the set of all timestamps that s is visited, $T(t)$ is the timestamp that the episode terminate following timestamp t, G_t is the return between timestamp t and $T(t)$. There are two different importance sampling:

ordinary importance sampling

$$V(s) = \frac{\sum_{t \in \mathscr{T}} \rho_{t:T-1}^{\pi/b} G_t}{|\mathscr{T}|} \tag{7.32}$$

weighted importance sampling

$$V(s) = \frac{\sum_{t \in \mathscr{T}} \rho_{t:T-1}^{\pi/b} G_t}{\sum_{t \in \mathscr{T}} \rho_{t:T-1}^{\pi/b}} \tag{7.33}$$

For first visit method, Ordinary importance sampling is unbiased, with unlimited variance. So weighted importance sampling is preferred in practice.

For every visit method, both sampling is biased which reduces to near zero when the number of sampling increases.

In practice, every visit is preferred because it does not need to keep trace of which states have been visited.

7.4.5 Incremental Policy Evaluation

Incremental implementation need the following background. If we want to estimate

$$V_n = \frac{\sum_{k=1}^{n-1} W_k G_k}{\sum_{k=1}^{n-1} W_k}, n \geq 2 \tag{7.34}$$

V_n could be incrementally updated by:

$$V_{n+1} = V_n + \frac{W_n}{C_n}\Big(G_n - V_n\Big), n \geq 1 \tag{7.35}$$

where

$$C_{n+1} = C_n + W_{n+1}$$

Algorithm (7.4.4) implements incremental solution of weighted importance sampling:

7.4.6 Incremental Policy Control

The behavior policy b need to be ε-soft.

Algorithm (7.4.5) implements incremental control of weighted importance sampling:

Algorithm 7.4.4 off-policy MC policy evaluation, estimate q_π

```
1: $Q(s,a) \in \mathbb{R}$
2: $C(s,a) \in 0$

3: loop
4:     $b \leftarrow$ any policy with coverage of $\pi$
5:     generate episode following $b$: $S_0, A_0, R_1, \ldots, S_{T-1}, A_{T-1}, R_T$
6:     $G \leftarrow 0$
7:     $W \leftarrow 1$
8:     for $t \leftarrow T-1, T-2, \ldots, 0$ do
9:         $G \leftarrow \gamma G + R_{t+1}$
10:        $C(S_t, A_t) \leftarrow C(S_t, A_t) + W$
11:        $Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \frac{W}{C(S_t, A_t)}\Big(G - Q(S_t, A_t)\Big)$
12:        $W \leftarrow W \frac{\pi(A_t|S_t)}{b(A_t|S_t)}$
13:        if $W = 0$ then                    ▷ $\pi(A_t|S_t) = 0$, $b$ does not cover $\pi$
14:            exit For loop
15:        end if
16:    end for
17: end loop
```

Algorithm 7.4.5 off-policy MC policy control, estimate q_*

```
1: $Q(s,a) \in \mathbb{R}$
2: $C(s,a) \in 0$
3: $\pi(s) \leftarrow \underset{a}{\operatorname{argmax}} Q(s,a)$

4: loop
5:     $b \leftarrow$ any $\varepsilon$-soft policy
6:     generate episode following $b$: $S_0, A_0, R_1, \ldots, S_{T-1}, A_{T-1}, R_T$
7:     $G \leftarrow 0$
8:     $W \leftarrow 1$
9:     for $t \leftarrow T-1, T-2, \ldots, 0$ do
10:        $G \leftarrow \gamma G + R_{t+1}$
11:        $C(S_t, A_t) \leftarrow C(S_t, A_t) + W$
12:        $Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \frac{W}{C(S_t, A_t)}\Big(G - Q(S_t, A_t)\Big)$
13:        $\pi(S_t) \leftarrow \underset{a}{\operatorname{argmax}} Q(S_t, a)$
14:        if $A_t \neq \pi(S_t)$ then
15:            exit For loop
16:        end if
17:        $W \leftarrow W \frac{1}{b(A_t|S_t)}$                    ▷ $\pi(A_t|S_t) = 1$ because it is greedy
18:    end for
19: end loop
```

7.5 Temporal Difference Learning

7.5.1 Constant-α TD(0) Prediction

Suppose at time $t+1$, state becomes S_{t+1} with reward R_{t+1}. The TD error is defined as:

$$\delta_t = R_{t+1} + \gamma V(S_{t+1}) - V(S_t) \tag{7.36}$$

The simplest TD is:

$$V(S_t) \leftarrow V(S_t) + \alpha\delta_t \tag{7.37}$$

Because TD use Markov property while MC is not, it is usually more efficient.

It use $R_{t+1} + \gamma V(S_{t+1})$ to estimate G_t in formula $V(S_t) \leftarrow V(S_t) + \alpha\Big(G_t - V(S_t)\Big)$ which is an average of all G_t.

TD is sample update which explores just one condition while MC is expected update which explores all choices.

The advantage of TD:

over DP TD does not require a model of environment.

over MC TD is an online, fully incremental fashion, more efficient.

TD is sensitive to initial value (guess).

Algorithm (7.5.1) contains detail.

Algorithm 7.5.1 TD(0) policy evaluation, estimate v_π

```
 1: $\alpha \in (0,1]$
 2: $V(s) \leftarrow$ random                                    ▷ bootstrap

 3: loop
 4:     choose S
 5:     repeat
 6:         $A \leftarrow$ action given by $\pi$ for S          ▷ following the given policy $\pi$
 7:         take action A, get R and $S'$
 8:         $V(S) \leftarrow V(S) + \alpha\Big(R + \gamma V(S') - V(S)\Big)$
 9:         $S \leftarrow S'$
10:     until S is terminal
11: end loop
```

7.5.2 Sarsa: On-policy TD Algorithm

7.5.2.1 Sarsa Prediction

Sarsa is a process of $S_t, A_t, R_{t+1}, S_{t+1}, A_{t+1}$, and its formula is:

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha\Big(R_{t+1} + \gamma Q(S_{t+1}, A_{t+1}) - Q(S_t, A_t)\Big) \tag{7.38}$$

If S_{t+1} is terminal, then $Q(S_{t+1}, A_{t+1})$ is 0.

7.5.2.2 Sarsa Control

The algorithm is almost the same as Sarsa prediction. The difference is to update the policy on the fly using ε-soft or ε-greedy policy.

Algorithm (7.5.2) contains detail.

7.5.3 Expected Sarsa Learning Algorithm

expected Sarsa use expectation, rather than ε-greedy, to learn next q. It reduces variance by removing the random selection of A_{t+1} during ε-soft policy :

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha\left(R_{t+1} + \gamma\sum_a \pi(a|S_{t+1})Q(S_{t+1}, a) - Q(S_t, A_t)\right) \tag{7.39}$$

Algorithm 7.5.2 on-policy Sarsa TD control, estimate q_*

```
1: $\alpha \in (0,1]$
2: $Q(s,a) \leftarrow$ random

3: loop
4:     choose S
5:     $A \leftarrow$ action from a policy derived from Q (e.g., $\varepsilon$-greedy)    ▷ update policy
6:     repeat
7:         take action A, get R and $S'$
8:                                                    ▷ update policy after each iteration
9:         $A' \leftarrow$ action from a policy derived from Q (e.g., $\varepsilon$-greedy)
10:        $Q(S,A) \leftarrow Q(S,A) + \alpha\Big(R + \gamma Q(S',A') - Q(S,A)\Big)$
11:        $S \leftarrow S'$
12:        $A \leftarrow A'$
13:    until S is terminal
14: end loop
```

7.5.4 Q-learning: Off-policy TD Control

Q-learning is defined as:

$$Q(S_t,A_t) \leftarrow Q(S_t,A_t) + \alpha\Big(R_{t+1} + \gamma \max_a Q(S_{t+1},a) - Q(S_t,A_t)\Big) \tag{7.40}$$

Algorithm (7.5.3) contains detail.

Algorithm 7.5.3 off-policy Q-learning TD control, estimate π_*

```
1: $\alpha \in (0,1]$
2: $Q(s,a) \leftarrow$ random

3: loop
4:     choose S
5:     repeat
6:         $A \leftarrow$ action given by Q for S (e.g., $\varepsilon$-greedy)
7:         take action A, get R and $S'$
8:         $Q(S,A) \leftarrow Q(S,A) + \alpha\Big(R + \gamma \max_a Q(S',a) - Q(S,A)\Big)$
9:         $S \leftarrow S'$
10:    until S is terminal
11: end loop
```

7.5.5 Double Learning

Maximization has maximization bias. For example, if $\mathbb{E}[q(s,a)] = 0$, then $\max q(s,a) > 0$. So the selection of q is always a positive bias. One way of viewing the problem is that maximization use the same sample to determine action and estimate its value. double learning uses two q for determination and estimation.

Algorithm (7.5.4) contains detail of double q-learning TD control.

There are double learning version for Sarsa and Expected-Sarsa as well.

7.5.6 Comments

Compared with MC, TD is sensitive to initial value. It explores Monte Carlo property and is more effective.

- TD: average V using $R + \gamma V$.
- Sarsa: from V to Q.
- Expect Sarsa: remove randomness of Sarsa by average.
- Q: replace average by max.
- Double-Q: reduce max bias by using two queues.

Algorithm 7.5.4 double Q-learning TD control, estimate q_*

```
1: $\alpha \in (0,1]$
2: $Q_1(s,a) \leftarrow$ random
3: $Q_2(s,a) \leftarrow$ random

4: loop
5:     choose S
6:     repeat
7:         A ← action given by $\varepsilon$-greedy policy of $Q_1 + Q_2$ for S
8:         take action A, get R and $S'$
9:         $p \leftarrow$ random(0, 1)
10:        if $p > 0.5$ then
11:            $Q_1(S,A) \leftarrow Q_1(S,A) + \alpha\Big(R + \gamma Q_2(S', \max_a Q_1(S',a)) - Q_1(S,A)\Big)$
12:        else
13:            $Q_2(S,A) \leftarrow Q_2(S,A) + \alpha\Big(R + \gamma Q_1(S', \max_a Q_2(S',a)) - Q_2(S,A)\Big)$
14:        end if
15:        $S \leftarrow S'$
16:    until S is terminal
17: end loop
```

7.6 *n*-step Bootstrapping

7.6.1 *n*-step TD Prediction

n-step TD prediction is still TD because it changes earlier estimate.

It did not update anything for the first $n-1$ steps. If $t+n \geq T$, the missing terms are treated as 0. It is defined as:

$$G_{t:t+n} = R_{t+1} + \gamma R_{t+2} + \cdots + \gamma^{n-1} R_{t+n} + \gamma^n V_{t+n-1}(S_{t+n}) \tag{7.41}$$

The algorithm (7.6.1) contains detail.

Algorithm 7.6.1 *n*-step TD prediction, estimate v_π

```
1:  $\alpha \in (0,1]$
2:  $V(s) \leftarrow$ random
3:  $t \leftarrow 0$

4:  loop
5:      choose $S_0$
6:      $T \leftarrow \infty$
7:      while $\tau < T-1$ do
8:          if $t < T$ then
9:              take action according to $\pi(\cdot|S_t)$
10:             store $R_{t+1}$ and $S_{t+1}$
11:             if $S_{t+1}$ is terminal then
12:                 $T \leftarrow t+1$
13:             end if
14:         end if
15:         $\tau \leftarrow t-n+1$                                  ▷ $\tau$ is the pivot of update
16:         if $\tau \geq 0$ then
17:             $G \leftarrow \sum_{i=\tau+1}^{\min(\tau+n,T)} \gamma^{i-\tau-1} R_i$                  ▷ $G_{\tau:\tau+n}$
18:             if $\tau + n < T$ then
19:                 $G \leftarrow G + \gamma^n V(S_{\tau+n})$
20:             end if
21:             $V(S_\tau) \leftarrow V(S_\tau) + \alpha\Big(G - V(S_\tau)\Big)$
22:         end if
23:         $t \leftarrow t+1$
24:     end while
25: end loop
```

7.6.2 *n*-step Sarsa

It is the same as *n*-step TD prediction with q and ε-greedy.

$$G_{t:t+n} = R_{t+1} + \gamma R_{t+2} + \cdots + \gamma^{n-1} R_{t+n} + \gamma^n Q_{t+n-1}(S_{t+n}, A_{t+n}) \tag{7.42}$$

The algorithm (7.6.2) contains detail.

7.6.3 *n*-step Expected Sarsa

It is the same as *n*-step Sarsa except that it uses expectation at the last step:

$$G_{t:t+n} = R_{t+1} + \gamma R_{t+2} + \cdots + \gamma^{n-1} R_{t+n} + \gamma^n \sum_a \pi(a|S_{t+n}) Q_{t+n-1}(S_{t+n}, a) \tag{7.43}$$

7.6.4 *n*-step Off-policy Learning

note: V_{t+n} and Q_{t+n} are the result of $(t+n)$th iteration.

Algorithm 7.6.2 *n*-step Sarsa, estimate q_π or q_*

```
1: $\alpha \in (0,1]$
2: $Q(s,a) \leftarrow$ random
3: $\pi \leftarrow$ random $\varepsilon$-greedy policy or a given fixed policy
4: $t \leftarrow 0$

5: loop
6:     choose $S_0$
7:     choose action $A_0 \sim \pi(\cdot|S_0)$
8:     $T \leftarrow \infty$
9:     while $\tau < T - 1$ do
10:        if $t < T$ then
11:            take action $A_t$ and store $R_{t+1}$ and $S_{t+1}$
12:            if $S_{t+1}$ is terminal then
13:                $T \leftarrow t + 1$
14:            else
15:                choose $A_{t+1} \sim \pi(\cdot|S_{t+1})$
16:            end if
17:        end if
18:        $\tau \leftarrow t - n + 1$                                ▷ $\tau$ is the pivot of update
19:        if $\tau \geq 0$ then
20:            $G \leftarrow \sum_{i=\tau+1}^{\min(\tau+n,T)} \gamma^{i-\tau-1} R_i$
21:            if $\tau + n < T$ then
22:                $G \leftarrow G + \gamma^n Q(S_{\tau+n}, A_{\tau+n})$                ▷ $G_{\tau:\tau+n}$
23:            end if
24:            $Q(S_\tau, A_\tau) \leftarrow Q(S_\tau, A_\tau) + \alpha\Big(G - Q(S_\tau, A_\tau)\Big)$
25:            update $\pi_*$                 ▷ update as a $\varepsilon$-greedy policy if calculating $q_*$
26:        end if
27:        $t \leftarrow t + 1$
28:    end while
29: end loop
```

7.6.4.1 *n*-step Off-policy TD

for $0 \le t < T$, the update formula is:

$$V_{t+n}(S_t) = V_{t+n-1}(S_t) + \alpha \prod_{k=t}^{\min(h,T-1)} \frac{\pi(A_k|S_k)}{b(A_k|S_k)}[G_{t:t+n} - V_{t+n-1}(S_t)] \tag{7.44}$$

7.6.4.2 *n*-step Off-policy Sarsa

for $0 \le t < T$, the update formula is:

$$\begin{aligned} Q_{t+n}(S_t, A_t) =& Q_{t+n-1}(S_t, A_t) \\ &+ \alpha \prod_{k=t}^{\min(h,T-1)} \frac{\pi(A_k|S_k)}{b(A_k|S_k)}[G_{t:t+n} - Q_{t+n-1}(S_t, A_t)] \end{aligned} \tag{7.45}$$

See Algorithm (7.6.3) for detail.

Algorithm 7.6.3 Off-policy n-step Sarsa, estimate q_π or q_*

```
1:  $\alpha \in (0,1]$
2:  $Q(s,a) \leftarrow$ random
3:  $\pi \leftarrow$ random $\varepsilon$-greedy policy
4:  $t \leftarrow 0$

5:  loop
6:      choose $S_0$
7:      choose action $A_0 \sim \pi(\cdot|S_0)$
8:      $T \leftarrow \infty$
9:      while $\tau < T - 1$ do
10:         if $t < T$ then
11:             take action $A_t$ and store $R_{t+1}$ and $S_{t+1}$
12:             if $S_{t+1}$ is terminal then
13:                 $T \leftarrow t + 1$
14:             else
15:                 choose $A_{t+1} \sim \pi(\cdot|S_{t+1})$
16:             end if
17:         end if
18:         $\tau \leftarrow t - n + 1$                                   ▷ $\tau$ is the pivot of update
19:         if $\tau \ge 0$ then
20:             $\rho \leftarrow \prod_{i=\tau+1}^{\min(\tau+n-1,T-1)} \frac{\pi(A_i|S_i)}{b(A_i|S_i)}$
21:             $G \leftarrow \sum_{i=\tau+1}^{\min(\tau+n,T)} \gamma^{i-\tau-1} R_i$
22:             if $\tau + n < T$ then
23:                 $G \leftarrow G + \gamma^n Q(S_{\tau+n}, A_{\tau+n})$                 ▷ $G_{\tau:\tau+n}$
24:             end if
25:             $Q(S_\tau, A_\tau) \leftarrow Q(S_\tau, A_\tau) + \alpha\rho\Big(G - Q(S_\tau, A_\tau)\Big)$
26:             update $\pi_*$                  ▷ update as a $\varepsilon$-greedy policy if calculating $q_*$
27:         end if
28:         $t \leftarrow t + 1$
29:     end while
30: end loop
```

7.6.5 *n*-step Tree Backup Algorithm

This is an off-policy learning algorithm without importance sampling. For each step along the sampling, the non-visited notes contribute probabilistic result according to the policy. The visited node will contribute the updated bootstrapping result.

$$\begin{aligned} G_{t:t+n} = R_{t+1} \\ &+ \gamma \sum_{a \neq A_{t+1}} \pi(a|S_{t+1}) Q_{t+n-1}(S_{t+1}, a) \text{ \# other branches} \\ &+ \gamma \pi(A_{t+1}|S_{t+1}) G_{t+1:t+n} \text{ \# main sample path} \end{aligned} \tag{7.46}$$

See Algorithm (7.6.4) for detail.

Algorithm 7.6.4 n-step tree backup, estimate q_π or q_*

```
1: $\alpha \in (0, 1]$
2: $Q(s, a) \leftarrow$ random
3: $\pi \leftarrow$ random $\varepsilon$-greedy policy

4: loop
5:     choose $S_0$
6:     choose action $A_0 \sim \pi(\cdot|S_0)$
7:     $T \leftarrow \infty$
8:     $t \leftarrow 0$
9:     while $\tau < T - 1$ do
10:        if $t < T$ then
11:            take action $A_t$ and store $R_{t+1}$ and $S_{t+1}$
12:            if $S_{t+1}$ is terminal then
13:                $T \leftarrow t + 1$
14:            else
15:                choose $A_{t+1} \sim \pi(\cdot|S_{t+1})$
16:            end if
17:        end if
18:        $\tau \leftarrow t - n + 1$                                ▷ $\tau$ is the pivot of update
19:        if $\tau \geq 0$ then
20:            if $t + 1 \geq T$ then
21:                $G \leftarrow R_T$
22:            else
23:                $G \leftarrow R_{t+1} + \gamma \sum_a \pi(a|S_{t+1}) Q(S_{t+1}, a)$
24:            end if
25:                                    ▷ update G backward using tree-backup method
26:            for $k \leftarrow [\tau + 1, \ldots, \min(t, T - 1)]$ do
27:                $G \leftarrow R_k + \gamma \sum_{a \neq A_k} \pi(a|S_k) Q(S_k, a) + \gamma \pi(A_k|S_k) G$
28:            end for
29:            $Q(S_\tau, A_\tau) \leftarrow Q(S_\tau, A_\tau) + \alpha \Big( G - Q(S_\tau, A_\tau) \Big)$
30:            update $\pi_*$                ▷ update as a $\varepsilon$-greedy policy if calculating $q_*$
31:        end if
32:        $t \leftarrow t + 1$
33:    end while
34: end loop
```

7.6.6 n-step off-policy Q(σ)

Let random variable $\sigma_t \sim \text{Bern}(0,1)$ be the probability of sampling on step t, with $\sigma = 1$ means full sampling and $\sigma = 0$ means pure expectation. The formula is:

$$\begin{aligned}
G_{t:h} =& R_{t+1} + \gamma \sum_{a \neq A_{t+1}} \pi(a|S_{t+1})Q_{h-1}(S_{t+1}, a) + \gamma\pi(A_{t+1}|S_{t+1})G_{t+1:h} \\
=& R_{t+1} + \left(\gamma \sum_a \pi(a|S_{t+1})Q_{h-1}(S_{t+1}, a) - \gamma\pi(A_{t+1}|S_{t+1})Q_{h-1}(S_{t+1}, A_{t+1})\right) \\
&+ \gamma\pi(A_{t+1}|S_{t+1})G_{t+1:h} \\
=& R_{t+1} + \gamma \sum_a \pi(a|S_{t+1})Q(S_{t+1}, a) \\
&+ \gamma\pi(A_{t+1}|S_{t+1})\Big(G_{t+1:h} - Q_{h-1}(S_{t+1}, A_{t+1})\Big)
\end{aligned} \tag{7.47}$$

Replace $\pi(A_{t+1}|S_{t+1})$ by $\Big(\sigma_{t+1}\rho_{t+1} + (1-\rho_{t+1})\pi(A_{t+1}|S_{t+1})\Big)$ (ρ is the important sampling ratio defined in formula (7.31)) we have:

$$\begin{aligned}
G_{t:h} =& R_{t+1} + \gamma \sum_a \pi(a|S_{k+1})Q(S_{k+1}, a) \\
&+ \gamma\Big(\sigma_{t+1}\rho_{t+1} + (1-\rho_{t+1})\pi(A_{t+1}|S_{t+1})\Big)\Big(G_{t+1:h} - Q_{h-1}(S_{t+1}, A_{t+1})\Big)
\end{aligned} \tag{7.48}$$

$\sum_a \pi(a|S_t)Q(S_t, a)$ is called <u>expected approximate value</u> of state S_t.
See Algorithm (7.6.5) for detail.

Algorithm 7.6.5 Off-policy n-step Q(σ), estimate q_π or q_*

```
1:  $\alpha \in (0,1]$
2:  $Q(s,a) \leftarrow$ random
3:  $\pi \leftarrow$ random $\varepsilon$-greedy policy
4:  random policy $b$ that $\forall a \in \mathscr{A}, s \in \mathscr{S}, b(a|s) > 0$
5:  $t \leftarrow 0$

6:  loop
7:      choose $S_0$
8:      choose action $A_0 \sim b(\cdot|S_0)$
9:      $T \leftarrow \infty$
10:     while $\tau < T - 1$ do
11:         if $t < T$ then
12:             take action $A_t$ and store $R_{t+1}$ and $S_{t+1}$
13:             if $S_{t+1}$ is terminal then
14:                 $T \leftarrow t + 1$
15:             else
16:                 choose $A_{t+1} \sim b(\cdot|S_{t+1})$
17:                 choose $\sigma_{t+1} \in \{0,1\}$                          ▷ $\sigma$ is either 0 or 1
18:                 $\rho_{t+1} \leftarrow \frac{\pi(A_{t+1}|S_{t+1})}{b(A_{t+1}|S_{t+1})}$
19:             end if
20:         end if
21:         $\tau \leftarrow t - n + 1$                                    ▷ $\tau$ is the pivot of update
22:         if $\tau \geq 0$ then
23:             for $k \leftarrow \Big[\min(t+1, T), \dots, \tau + 1\Big]$ do
24:                 if $k = T$ then
25:                     $G \leftarrow R_T$
26:                 else
27:                     $\overline{V} \leftarrow \sum_a \pi(a|S_k)Q(S_k, a)$
28:                     $G \leftarrow R_k + \gamma\overline{V} + \gamma\Big(\sigma_k\rho_k + \big(1-\rho_k\big)\pi(A_k|S_k)\Big)\Big(G - Q(S_k, A_k)\Big)$
29:                 end if
30:             end for
31:             $G(S_\tau, A_\tau) \leftarrow G(S_\tau, A_\tau) + \alpha\Big(G - G(S_\tau, A_\tau)\Big)$
32:             update $\pi_*$                        ▷ update as a $\varepsilon$-greedy policy if calculating $q_*$
33:         end if
34:         $t \leftarrow t + 1$
35:     end while
36: end loop
```

7.7 Value Function Approximation

In function approximation, the value function v_π becomes:

$$v_\pi(s) \approx \hat{v}(s, \mathrm{W}) \ , \ \mathrm{W} \in \mathbb{R}^d \tag{7.49}$$

It is a supervised learning with data pair:

$$\begin{aligned} &\langle \mathrm{S}_1, \mathrm{R}_1 + \gamma\hat{v}(\mathrm{S}_2, \mathrm{W})\rangle \\ &\langle \mathrm{S}_2, \mathrm{R}_2 + \gamma\hat{v}(\mathrm{S}_3, \mathrm{W})\rangle \\ &\quad \cdots \end{aligned} \tag{7.50}$$

7.7.1 On-policy Prediction

7.7.1.1 Requirement

The approximate function $\hat{v}(s, \mathrm{W})$ has these requirements:

- learning needs to be online
- the learning target is non-stationary and can change over time
- differentiable of W for all $s \in \mathcal{S}$

7.7.1.2 Prediction Objective

In tabular case there is no estimation of value function quality because it will converge to the end goal. However in function approximation there is no exact value function and the quality need to be estimated.

Assume the state is distributed under $\mu(s) > 0, \sum_s \mu(s) = 1$, the mean squared value error $\overline{\mathrm{VE}}$ is defined as:

$$\overline{\mathrm{VE}} = \sum_{s \in \mathcal{S}} \mu(s)\Big(v_\pi(s) - \hat{v}(s, \mathrm{W})\Big)^2 \tag{7.51}$$

$\mu(s)$ can be chosen as the fraction of time for episodes, and stationary distribution for continuous tasks.
$\hat{v}$ is chosen to be differentiable function, which could be:

- linear combination of features
- neural network

7.7.1.3 Stochastic Gradient Descent

Assume μ is uniform distribution, the W is updated as:

$$\begin{aligned} \mathrm{W}_{t+1} &= \mathrm{W}_t - \frac{1}{2}\alpha\nabla_{\mathrm{W}}\Big(v_\pi(\mathrm{S}_t) - \hat{v}(\mathrm{S}_t, \mathrm{W}_t)\Big)^2 \\ &= \mathrm{W}_t + \alpha\Big(v_\pi(\mathrm{S}_t) - \hat{v}(\mathrm{S}_t, \mathrm{W}_t)\Big)\nabla_{\mathrm{W}}\hat{v}(\mathrm{S}_t, \mathrm{W}_t) \end{aligned} \tag{7.52}$$

where $\nabla_{\mathrm{W}} f(\mathrm{W})$ is defined as:

$$\nabla_{\mathrm{W}} f(\mathrm{W}) = \left(\frac{\partial f(\mathrm{W})}{\partial \mathrm{W}_1}, \frac{\partial f(\mathrm{W})}{\partial \mathrm{W}_2}, \ldots, \frac{\partial f(\mathrm{W})}{\partial \mathrm{W}_d}\right)^\top \tag{7.53}$$

Formula (7.52) need to follow these in order to converge to a local minimum:

- α follow equation (7.4).
- $v_\pi(\mathrm{S}_t)$ is an unbiased estimate of v.

See Algorithm (7.7.1) for detail.
In practice, the v_π in formula (7.52) is chosen as:

- for MC, the target is the return G_t:
- for TD(0), the target is $\mathrm{R}_{t+1} + \gamma\hat{v}(\mathrm{S}_{t+1}, \mathrm{W})$
- for TD(λ), the target is G_t^λ which could be forward or backward view

7.7.2 Semi-gradient methods

In formula (7.52) if $v_\pi(\mathrm{S}_t)$ depends on W, the formula is biased and will not converge as the true gradient descent methods. It is called semi-gradient methods.

Bootstrapping estimate belongs to this category.

Algorithm 7.7.1 gradient MC, estimate $\widehat{v} \approx v_\pi$

```
1: $W \leftarrow$ random

2: loop
3:     generate $S_0, A_0, R_1, \dots, R_T, S_T$ using $\pi$
4:     for $t \leftarrow \Big[0, 1, \dots, T-1\Big]$ do
5:         $W \leftarrow W + \alpha\Big(G_t - \widehat{v}(S_t, W)\Big)\nabla_W \widehat{v}(S_t, W)$
6:     end for
7: end loop
```

7.7.3 Linear Methods

Suppose $\widehat{v}$ is linear: $\widehat{v}(s, W) = W^\top X(s) = \sum_{i=1}^{d} w_i x_i(s)$. X($s$) is called feature vector represents state s. Formula (7.52) now becomes:

$$W_{t+1} = W_t + \alpha\Big(v_\pi(S_t) - \widehat{v}(S_t, W_t)\Big)X(S_t) \tag{7.54}$$

In linear case all local optimum is global optimum.

7.7.3.1 Semi-gradient Linear Methods TD(0)

semi-gradient TD(0) algorithms also converges under linear function. But it converges to a point near the local optimum, rather than global minimum.

$$\begin{aligned} W_{t+1} &= W_t + \alpha\Big(R_{t+1} + \gamma W_t^\top X_{t+1} - W_t^\top X_t\Big)X \\ &= W_t + \alpha\Big(R_{t+1}X_t + X_t(X_t - \gamma X_{t+1})^\top W_t\Big) \end{aligned} \tag{7.55}$$

The expected next weight vector could be written as:

$$\mathbb{E}[W_{t+1}|W_t] = W_t + \alpha(\mathbf{b} - AW_t) \tag{7.56}$$

where

$$\mathbf{b} = \mathbb{E}[R_{t+1}X_t] \in \mathscr{R}^d \tag{7.57}$$

and

$$A = \mathbb{E}[X_t(X_t - \gamma X_{t+1})^\top] \tag{7.58}$$

If formula (7.56) converges and is unbiased, it will converge to W_{TD} at which:

$$\begin{aligned} \mathbf{b} - AW_{TD} &= 0 \\ \mathbf{b} &= AW_{TD} \\ W_{TD} &= A^{-1}\mathbf{b} \end{aligned} \tag{7.59}$$

The solution of formula (7.59) is around global minimum:

$$\overline{VE}(W_{TD}) \leq \frac{1}{1-\gamma} \min_W \overline{VE}(W) \tag{7.60}$$

formula (7.69) applies to other on-policy bootstrapping methods as well, such as semi-gradient DP, semi-gradient action value methods.

7.7.3.2 Least-Squares TD

In LSTD, the A and b in formula (7.59) is defined as:

$$\widehat{A}_t = \sum_{k=0}^{t-1} X_k(X_k - \gamma X_{k+1})^\top + \varepsilon I \tag{7.61}$$

and

$$\widehat{b}_t = \sum_{k=0}^{t-1} \mathrm{R}_{t+1}\mathrm{X}_k \tag{7.62}$$

A small $\varepsilon > 0$ is added to ensure $\widehat{\mathrm{A}}_t$ is always invertible.
w_t is now defined as $w_t = \widehat{\mathrm{A}}_t^{-1}\widehat{b}_t$.
There is a Sherman-Morrison formula that simplify the calculation of $\widehat{\mathrm{A}}$:

$$\begin{aligned}\widehat{\mathrm{A}}_t^{-1} &= \left(\widehat{\mathrm{A}}_{t-1} + \mathrm{X}_t(\mathrm{X}_t - \gamma\mathrm{X}_{t+1})^\top\right)^{-1} \\ &= \widehat{\mathrm{A}}_{t-1}^{-1} - \frac{\widehat{\mathrm{A}}_{t-1}^{-1}\mathrm{X}_t(\mathrm{X}_t - \gamma\mathrm{X}_{t+1})^\top\widehat{\mathrm{A}}_{t-1}^{-1}}{1 + (\mathrm{X}_t - \gamma\mathrm{X}_{t+1})^\top\widehat{\mathrm{A}}_{t-1}^{-1}\mathrm{X}_t}\end{aligned} \tag{7.63}$$

with $\widehat{\mathrm{A}}_0 = \varepsilon\mathrm{I}$
LSTD does not require α, but it needs ε which has these problems:

- small ε: the inverse calculation will vary widly
- big ε: the learning is slow
- no α: it never forgets

7.7.4 On-policy Control

In approximate control, the v in formula (7.52) is changed to q:

$$\mathrm{W}_{t+1} = \mathrm{W}_t + \alpha\Big(\mathrm{U}_t - \widehat{q}(\mathrm{S}_t, \mathrm{A}_t, \mathrm{W}_t)\Big)\nabla_{\mathrm{W}}\widehat{q}(\mathrm{S}_t, \mathrm{A}_t, \mathrm{W}_t) \tag{7.64}$$

As before, the U_t could be :

- for MC, the target is the return G_t:
- for TD(0), the target is $\mathrm{R}_{t+1} + \gamma\widehat{q}(\mathrm{S}_{t+1}, \mathrm{A}_{t+1}, \mathrm{W})$
- for TD(λ), the target is G_t^λ with forward and backward view

7.8 Eligibility Traces

7.8.1 λ-return

eligibility trace is a short term memory vector that has the same dimension as W.

The forward view of TD(λ) is defined as:

$$\begin{cases} G_t^{\lambda} = (1-\lambda)\sum_{n=1}^{\infty} \lambda^{n-1} G_{t:t+n} & \text{, for continuous task} \\ G_t^{\lambda} = (1-\lambda)\sum_{n=1}^{T-t-1} \lambda^{n-1} G_{t:t+n} + \lambda^{T-t-1} G_t & \text{, for episodes} \end{cases} \tag{7.65}$$

7.8.2 TD(λ)

In backward view of TD(λ), the eligibility trace $z_t \in \mathbb{R}^d$ is defined as:

$$\begin{aligned} z_{-1} &= 0 \\ z_t &= \gamma \lambda z_{t-1} + \nabla \hat{v}(S_t, W_t) \end{aligned} \tag{7.66}$$

Of which γ is the discount rate and λ is the λ in TD(λ).
The TD error is defined as:

$$\delta_t = R_{t+1} + \gamma \hat{v}(S_{t+1}, W_t) - \hat{v}(S_t, W_t) \tag{7.67}$$

The gradient is defined as:

$$W_{t+1} = W_t + \alpha \delta_t z_t \tag{7.68}$$

Here the α is the ratio used for mean value converge calculation.
If $\lambda = 0$, TD(λ) becomes one-step semi-gradient TD.
If $\lambda = 1$, TD(λ) becomes Monte Carlo calculation.
Linear TD(λ) will converge in on-policy case if step size follows formula (7.4):

$$\overline{VE}(W_{\infty}) \leq \frac{1-\gamma\lambda}{1-\gamma} \min_{W} \overline{VE}(W) \tag{7.69}$$

In practice, do not choose $\lambda = 1$ which is the poorest choice.

7.9 Policy Gradient

7.9.1 Policy Approximation

The action $\pi(a|s,\theta) = \mathbb{P}\{\mathrm{A}_t = a|\mathrm{S}_t = s, \theta_t = \theta\}$ is a random variable.

The problem is to maximize performance measure $\mathscr{J}(\theta)$:

$$\theta_{t+1} = \theta_t + \alpha\widehat{\nabla\mathscr{J}(\theta_t)} \tag{7.70}$$

All methods that follow this schema is called policy gradient methods.

The benefit of policy gradient:

- continuous action space
- can learn stochastic policy
- no maximization cost which is slow

The disadvantage of policy gradient:

- converge to local optimum rather than global

why it is a maximization problem:

- in value prediction, the policy is fixed and the value need to converge to a theoretical result. So the error need to be minimized.
- in value control, the optimal policy is incrementally updated. However, the optimal policy is a deterministic result of updated value function.
- in policy gradient, the goal is to maximize result.

In practice, in order to ensure exploration we require the policy is never deterministic, i.e., $\pi(a|s,\theta) \in (0,1)$.

If the action space is discrete and not too large, we can use softmax for numerical preference state-action pair $e^{h(s,a,\theta)}$:

$$\pi(a|s,\theta) = \frac{e^{h(s,a,\theta)}}{\sum_b e^{h(s,a,\theta)}} \tag{7.71}$$

The softmax could be used with ε-greedy to achieve non-deterministic policy. For some problem the best approximate policy may be stochastic.

For some problem the action value function may have simpler presentation, while for some the policy gradient is simpler.

7.9.2 Policy Gradient Theorem

The policy gradient theorem says:

$$\nabla\mathscr{J}(\theta) \propto \sum_s \mu(s) \sum_a q_\pi(s,a)\nabla\pi(a|s,\theta) \tag{7.72}$$

In episodic case, the constant of proportionality is the average length of the episode. In continuous case it is 1. The μ here is the distribution of s under policy π.

The proof is:

$$\begin{aligned}
\nabla v_\pi(s) &= \nabla\left[\sum_a \pi(a|s)q_\pi(s,a)\right] \\
&= \sum_a \Big[\nabla\pi(a|s)q_\pi(s,a) + \pi(a|s)\nabla q_\pi(s,a)\Big] \\
&= \sum_a \Big[\nabla\pi(a|s)q_\pi(s,a) + \pi(a|s)\nabla\sum_{s',r} p(s',r|s,a)(r + v_\pi(s'))\Big] \\
&= \sum_a \Big[\nabla\pi(a|s)q_\pi(s,a) + \pi(a|s)\sum_{s'} p(s'|s,a)\nabla v_\pi(s')\Big] \\
&= \sum_a \Big[\nabla\pi(a|s)q_\pi(s,a) + \pi(a|s)\sum_{s'} p(s'|s,a) \\
&\quad \sum_{a'}[\nabla\pi(a'|s')q_\pi(s',a') + \pi(a'|s')\sum_{s''} p(s''|s',a')\nabla v_\pi(s'')]\Big] \\
&= \sum_{x\in\mathscr{S}}\left(\sum_{k=0}^{\infty}\mathbb{P}(s\to x,k,\pi)\right)\sum_a \nabla\pi(a|x)q_\pi(x,a)
\end{aligned} \tag{7.73}$$

where $\mathbb{P}(s \to x, k, \pi)$ is the probability of transition from state s to state x in k steps under policy π. So:

$$\begin{aligned}
\nabla \mathscr{J}(\theta) &= \nabla v_\pi(s_0) \\
&= \sum_s \left(\sum_{k=0}^{\infty} \mathbb{P}(s_0 \to x, k, \pi) \right) \sum_a \nabla \pi(a|x) q_\pi(x,a) \\
&= \sum_s \eta(s) \sum_a \nabla \pi(a|s) q_\pi(s,a) \\
&= \sum_{s'} \eta(s') \sum_s \frac{\eta(s)}{\sum_{s'} \eta s'} \sum_a \nabla \pi(a|s) q_\pi(s,a) \\
&= \sum_{s'} \eta(s') \sum_s \eta s \sum_a \nabla \pi(a|s) q_\pi(s,a) \\
&\propto \sum_s \mu(s) \sum_a \nabla \pi(a|s) q_\pi(s,a)
\end{aligned} \tag{7.74}$$

7.9.3 REINFORCE: Monte Carlo Policy Gradient

REINFORCE has actor but no critic. It use Monte Carlo method to calculate the G directly without calculating the value function.

$$\begin{aligned}
\nabla \mathscr{J}(\theta) &\propto \sum_s \mu(s) \sum_a \nabla \pi(a|s,\theta) q_\pi(s,a) \\
&= \mathbb{E}_\pi \left[\sum_a q_\pi(\mathrm{S}_t, a) \nabla \pi(a|\mathrm{S}_t, \theta) \right] \\
&= \mathbb{E}_\pi \left[\sum_a q_\pi(\mathrm{S}_t, a) \pi(a|\mathrm{S}_t, \theta) \frac{\nabla \pi(a|\mathrm{S}_t, \theta)}{\pi(a|\mathrm{S}_t, \theta)} \right] \\
&= \mathbb{E}_\pi \left[q_\pi(\mathrm{S}_t, \mathrm{A}_t) \frac{\nabla \pi(\mathrm{A}_t|\mathrm{S}_t, \theta)}{\pi(\mathrm{A}_t|\mathrm{S}_t, \theta)} \right] \\
&= \mathbb{E}_\pi \left[\mathrm{G}_t \frac{\nabla \pi(\mathrm{A}_t|\mathrm{S}_t, \theta)}{\pi(\mathrm{A}_t|\mathrm{S}_t, \theta)} \right] \\
&= \mathbb{E}_\pi \Big[\mathrm{G}_t \nabla \log \pi(\mathrm{A}_t|\mathrm{S}_t, \theta) \Big]
\end{aligned} \tag{7.75}$$

So formula (7.70) is now:

$$\begin{aligned}
\theta_{t+1} &= \theta_t + \alpha \widehat{\nabla \mathscr{J}(\theta_t)} \\
&= \theta_t + \alpha \mathrm{G}_t \nabla \log \pi(\mathrm{A}_t|\mathrm{S}_t, \theta)
\end{aligned} \tag{7.76}$$

Here G_t is the return in Monte Carlo cases.
Algorithm (7.9.1) contains detail.
Formula (7.76) has many forms:

$$\begin{aligned}
\nabla \widehat{\mathscr{J}(\theta_t)} &= \mathbb{E}_\pi \Big[\nabla \log \pi(\mathrm{A}_t|\mathrm{S}_t, \theta) \mathrm{G}_t \Big] \\
&= \mathbb{E}_\pi \Big[\nabla \log \pi(\mathrm{A}_t|\mathrm{S}_t, \theta) v_t \Big] \\
&= \mathbb{E}_\pi \Big[\nabla \log \pi(\mathrm{A}_t|\mathrm{S}_t, \theta) q_t(s,a) \Big] \\
&= \mathbb{E}_\pi \Big[\nabla \log \pi(\mathrm{A}_t|\mathrm{S}_t, \theta) \delta_t \Big]
\end{aligned} \tag{7.77}$$

7.9.4 Baseline

A baseline is an arbitrary function $b(s)$ that does not vary with a:

$$\nabla \mathscr{J}(\theta) \propto \sum_s \mu(s) \sum_a \nabla \pi(a|s,\theta) \Big(q_\pi(s,a) - b(s) \Big) \tag{7.78}$$

Algorithm 7.9.1 REINFORE: Monte Carlo control for π_*

```
1: $\theta \in \mathbb{R}^d$
2: $\gamma$: the discount rate

3: loop
4:     generate episode $S_0, A_0, R_1, \dots, S_{T-1}, A_{T-1}, R_T$ following $\pi(\cdot|\cdot, \theta)$
5:     for $t \in [0, 1, \dots, T-1]$ do
6:         $G \leftarrow \sum_{k=t+1}^{T} \gamma^{k-t-1} R_k$
7:         $\theta \leftarrow \theta + \alpha\gamma^t G \nabla \log \pi(A_t|S_t, \theta)$
8:     end for
9: end loop
```

It is ok to add baseline because it has no effect:

$$\begin{aligned}\sum_a b(s)\nabla\pi(a|s,\theta) &= b(s)\nabla\sum_a \pi(a|s,\theta) \\ &= b(s)\nabla 1 \\ &= 0\end{aligned} \tag{7.79}$$

So the REINFORCE update now becomes:

$$\theta_{t+1} = \theta_t + \alpha\Big(G_t - b(S_t)\Big)\nabla\log\pi(A_t|S_t,\theta) \tag{7.80}$$

One good choice of $b(s)$ is the state value $\widehat{v}(S_t, W)$. It is called advantage function:

$$A(s,a) = Q(s,a) - V(s) \tag{7.81}$$

For optimal A^*, the advantage function becomes:

$$\begin{aligned}A^*(s,a) &= Q^*(s,a) - V^*(s) \\ &= \begin{cases} 0 & \text{, if } a = a^* \\ < 0 & \text{, if } a \neq a^* \end{cases}\end{aligned} \tag{7.82}$$

Because REINFORCE is a Monte Carlo learning algorithm, it is natural to learn $\widehat{v}$ using Monte Carlo as well. Algorithm (7.9.2) contains detail. In algorithm, in linear case $\alpha^W = \frac{0.1}{\mathbb{E}\left[\|\nabla\widehat{v}(S_t,W)\|_\mu^2\right]}$, while the best value of α^θ depends on the problem.

Algorithm 7.9.2 REINFORE with baseline for π_*

```
1: $\theta \in \mathbb{R}^d$ such as 0
2: $\gamma$: the discount rate
3: $\alpha^\theta > 0$
4: $\alpha^W > 0$

5: loop
6:     generate episode $S_0, A_0, R_1, \dots, S_{T-1}, A_{T-1}, R_T$ following $\pi(\cdot|\cdot, \theta)$
7:     for $t \in [0, 1, \dots, T-1]$ do
8:         $G \leftarrow \sum_{k=t+1}^{T} \gamma^{k-t-1} R_k$
9:         $\delta \leftarrow G - \widehat{v}(S_t, W)$
10:        $W \leftarrow W + \alpha^W \gamma^t \delta \nabla \widehat{v}(S_t, W)$        ▷ learn $\widehat{v}$ by TD(0)
11:        $\theta \leftarrow \theta + \alpha^\theta \gamma^t G \nabla \log \pi(A_t|S_t, \theta)$
12:    end for
13: end loop
```

7.9.5 Actor-Critic Methods

In actor-critic methods, actor refers to learned policy and critic refers to learned value function, usually a state-value function:

actor update Q by policy gradient
critic update w by TD(0)

In REINFORCE with baseline case it only calculate policy, but not value function. And it is not a bootstrapping method. Bootstrapping is good because it reduce variance and accelerate learning speed, although it is biased.

actor learning alone is slow, so need the help of critic.

The change is to use $G_{t:t+1}$ instead of G_t:

$$\begin{aligned}\theta_{t+1} &= \theta_t + \alpha\Big(G_{t:t+1} - b(S_t)\Big)\nabla \log \pi(A_t|S_t,\theta) \\ &= \theta_t + \alpha\Big(R_{t+1} + \gamma\hat{v}(S_{t+1},W) - b(S_t)\Big)\nabla \log \pi(A_t|S_t,\theta)\end{aligned} \tag{7.83}$$

The forward algorithm is (7.9.3) and the backward algorithm is (7.9.4).

Algorithm 7.9.3 one-step Actor-Critic forward view for π_*

```
1: $\theta \in \mathbb{R}^d$ such as 0
2: $\gamma$: the discount rate
3: $\alpha^\theta > 0$
4: $\alpha^W > 0$

5: loop
6:     initialize S
7:     $I \leftarrow 1$
8:     repeat
9:         $A \sim \pi(\cdot|S,\theta)$
10:        take action A, observe $S', R$
11:        $\delta \leftarrow R + \gamma\hat{v}(S',W) - \hat{v}(S,W)$        ▷ $\hat{v}(S',W) = 0$ if $S'$ is terminal
12:        $W \leftarrow W + \alpha^W I\delta\nabla\hat{v}(S,W)$
13:        $\theta \leftarrow \theta + \alpha^\theta I\delta\nabla \log \pi(A|S,\theta)$
14:        $I \leftarrow \gamma I$
15:        $S \leftarrow S'$
16:    until S is terminal
17: end loop
```

Algorithm 7.9.4 Actor-Critic with eligibility trace, backward view for π_*

```
1: $\theta \in \mathbb{R}^d$ such as 0
2: $\gamma$: the discount rate
3: $\alpha^\theta > 0$
4: $\alpha^W > 0$

5: loop
6:     initialize S
7:     $I \leftarrow 1$
8:     repeat
9:         $A \sim \pi(\cdot|S,\theta)$
10:        take action A, observe $S', R$
11:        $\delta \leftarrow R + \gamma\hat{v}(S',W) - \hat{v}(S,W)$        ▷ $\hat{v}(S',W) = 0$ if $S'$ is terminal
12:        $z^W \leftarrow \gamma\lambda^\theta z^W + I\nabla\hat{v}(S,W)$
13:        $z^\theta \leftarrow \gamma\lambda^\theta z^\theta + I\nabla \log \pi(A|S,\theta)$
14:        $W \leftarrow W + \alpha^W \delta z^W$
15:        $\theta \leftarrow \theta + \alpha^\theta \delta z^\theta$
16:        $I \leftarrow \gamma I$
17:        $S \leftarrow S'$
18:    until S is terminal
19: end loop
```

7.10 Unify Planning and Learning

7.10.1 Model and Planning

Planning and learning share two basic ideas:

1. all involve computing value functions
2. compute value function by update or backup operation to simulated experiences

The difference is that planning use simulated experience generated by model, while learning use real experience generated by environment.

A model $\mathcal{M} = \langle \mathcal{P}, \mathcal{R} \rangle$ represents state and reward transition:

$$\begin{aligned} S_{t+1} &\sim \mathcal{P}(S_{t+1}|S_t, A_t) \\ R_{t+1} &= \mathcal{R}(R_{t+1}|S_t, A_t) \end{aligned} \tag{7.84}$$

typically there is an assumption that S_t and R_t are independent:

$$\mathbb{P}[S_{t+1}, R_{t+1}|S_t, A_t] = \mathbb{P}[S_{t+1}|S_t, A_t] \times \mathbb{P}[R_{t+1}|S_t, A_t] \tag{7.85}$$

the problem with model based learning is that it has two source of approximation.

The learning of model $\mathcal{M}$ is a supervised learning:

$$\begin{aligned} S_1, A_1 &\rightarrow R_2, S_2 \\ S_2, A_2 &\rightarrow R_3, S_3 \\ &\dots \\ S_{T-1}, A_{T-1} &\rightarrow R_T, S_T \end{aligned} \tag{7.86}$$

The step of planning with model:

1. use supervised learning to learn a model $\mathcal{M}$
2. use model only to generate samples S and R
3. apply model-free reinforcement learning to samples

sample based planning is often more efficient.

7.10.2 Dyna-Q

Within a planning agent, real experience has two use cases:

model-learning update model

direct reinforcement learning improve value and policy function

See Algorithm (7.10.1) for detail.

7.10.3 Prioritized Sweeping

In background model improvement it is not useful to sweep over all states. The state and transition that leads to goal states, or to state whose value has changed, are more useful. It is called backward focusing.

In backward focusing the state that has changed a lot are more likely to change. So when sweeping backward from the goal, choose the ones with the biggest change history and update them.

See Algorithm (7.10.2) for detail.

Extension to stochastic environment is done by updating the P with sampled expected value.

7.10.4 Expected and Sample Update

DP uses expected update which consider all possible events while TD uses sample update which consider a single example.

expected update is better because it can avoid sampling error. But the expectation calculation is expensive, which is roughly the time of branching factor times of sample update. In practice the branching factor is usually very high and sample update is preferred.

When computation power is limited, it is a choice between sample update for many $\langle S, A \rangle$ pair and expectation update for some $\langle S, A \rangle$. In practice the sample update error will drop along the curve $\sqrt{\frac{b-b}{bt}}$ where b is the branching factor and t is the number of performed sample update. So the converge rate is very fast by taking sample update.

Algorithm 7.10.1 Dyna-Q

```
1:  $Q(s, a) \leftarrow$ random
2:  Model$(s, a) \leftarrow$ random

3:  loop
4:      $S \leftarrow$ current state (non-terminal)
5:      $A \leftarrow \varepsilon$-greedy$(S, Q)$
6:      take A, record R and S′
7:                                                  ▷ direct Q-learning
8:      $Q(S, A) \leftarrow Q(S, A) + \alpha\Big(R + \gamma \max_a Q(S', a) - Q(S, A)\Big)$
9:      Model$(S, A) \leftarrow (R, S')$               ▷ assuming deterministic environment
10:     $i \leftarrow 0$
11:     repeat                                     ▷ indirect RL, model learning process
12:         $S \leftarrow$ random previously observed state
13:         $A \leftarrow$ random action taken in state S
14:         $R, S' \leftarrow$ Model$(S, A)$
15:         $Q(S, A) \leftarrow Q(S, A) + \alpha\Big(R + \gamma \max_a Q(S', a) - Q(S, A)\Big)$
16:         $i \leftarrow i + 1$
17:     until $i = n$
18: end loop
```

Algorithm 7.10.2 prioritized sweeping with Dyna-Q

```
1:  $Q(s, a) \leftarrow$ random
2:  Model$(s, a) \leftarrow$ random
3:  PQueue $\leftarrow [\,]$

4:  loop
5:      $S \leftarrow$ current state (non-terminal)
6:      $A \leftarrow$ policy$(S, Q)$
7:      take A, record R and S′
8:      $P \leftarrow |R + \gamma \max_a Q(S', a) - Q(S, A)|$
9:      $Model(S, A) \leftarrow (R, S')$               ▷ assuming deterministic environment
10:     $i \leftarrow 0$
11:     repeat                                     ▷ indirect RL, model learning process
12:         $\langle S, A\rangle \leftarrow$ PQueue.head()
13:         $R, S' \leftarrow$ Model$(S, A)$
14:         $Q(S, A) \leftarrow Q(S, A) + \alpha\Big(R + \gamma \max_a Q(S', a) - Q(S, A)\Big)$
15:         for $\forall \langle \overline{S}, \overline{A}\rangle$ that leads to S do
16:             $\overline{R} \leftarrow$ predicted reward from $\overline{S}, \overline{A}, S$
17:             $P \leftarrow |\overline{R} + \gamma \max_a Q(S, a) - Q(\overline{S}, \overline{A})|$
18:             if $P > \theta$ then
19:                 PQueue.add$(P \rightarrow \langle \overline{S}, \overline{A}\rangle)$
20:             end if
21:         end for
22:     until $i = n$
23: end loop
```

Chapter 8

Others

8.1 Other Notes

8.1.1 Convergence of Value Iteration

let $\mathrm{T}^\pi(v) = \mathrm{R}^\pi + \gamma \mathrm{P}^\pi v$

$$\begin{aligned} ||\mathrm{T}^\pi(u) - \mathrm{T}^\pi(v)||_\infty &= ||(\mathrm{R}^\pi + \gamma \mathrm{P}^\pi u) - (\mathrm{R}^\pi + \gamma \mathrm{P}^\pi v)||_\infty \\ &= ||\gamma \mathrm{P}^\pi (u - v)||_\infty \\ &\le ||\gamma \mathrm{P}^\pi ||u - v||_\infty||_\infty \\ &\le \gamma ||u - v||_\infty \end{aligned} \tag{8.1}$$

So T will converge to fixed point at linear rate γ according to *contraction mapping theorem*.

Bibliography

[Aur19] Aurélien Géron. *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow*. O'Reilly Media, Incorporated, 2 edition, 2019.

[BPRS15] Atilim Gunes Baydin, Barak A. Pearlmutter, Alexey Andreyevich Radul, and Jeffrey Mark Siskind. Automatic differentiation in machine learning: a survey. *J. Mach. Learn. Res.*, 18:1–43, feb 2015.

[DBW12] John C. Duchi, Peter L. Bartlett, and Martin J. Wainwright. Randomized smoothing for (parallel) stochastic optimization. In *Proc. IEEE Conf. Decis. Control*, volume 12, pages 5442–5444, 2012.

[Doz16] Timothy Dozat. Incorporating Nesterov Momentum into Adam. *ICLR Work.*, (1):2013–2016, 2016.

[KB15] Diederik P. Kingma and Jimmy Lei Ba. Adam: A method for stochastic optimization. In *3rd Int. Conf. Learn. Represent. ICLR 2015 - Conf. Track Proc.*, pages 1–15, 2015.

[NES83] Y. NESTEROV. A method for solving the convex programming problem with convergence rate O(1/k^2). *Dokl. Akad. Nauk SSSR*, 269:543–547, 1983.

[Qia99] Ning Qian. On the momentum term in gradient descent learning algorithms. *Neural Networks*, 12(1):145–151, 1999.

[Rud16] Sebastian Ruder. An overview of gradient descent optimization algorithms. pages 1–14, 2016.

[Zei12] Matthew D. Zeiler. ADADELTA: An Adaptive Learning Rate Method. 2012.

Index