Volume 1 / **Fundamental Analysis**

THE ART OF THEORETICAL RESEARCH

FIRST EDITION

Fujimiya Amane

TAOTR, Edinburgh, UK

Contents

Introduction and Preface

Well, nothing here. For now at least.

1 The Double Descent and Model Complexity

An expository analysis

F. Amane

The study of **double descent**, if anything, is not so straight forward. When we are investigating something without clear guarantee, and clear foundation, it is easy for any one to step out, and face the wall of unsolicited and unfounded theoretical ground work together. To know what this means, however, few observations is needed. Let's have a look.

1.1 Notation

We denotes named components, for example in later section, *supervisor*, *generator* and *learner (model)* by a 3-fold notation, for example, $\mathsf{Sup}(\cdot)\{\cdot,\cdot\}[\cdot]$, with each "$\cdot$" an unknown parameter, components, or set. $(\cdot)$ is the receiver (input), $[\cdot]$ is the output transmitter, and the $\{\cdot,\dots\}$ is the descriptor of the model itself. Of course, the formation can be changed, as the order is not strictly necessary, and the braces themselves covers for clarification, but for readability, it is typically arranged so.

We also clarify certain word. A concept (supervisor) is learnable if its behaviour can be expressed, contained. More often, it is subjective to the model given. A model is learnable, if there exists mutation inside the behaviour configuration of it. Historically, this is modelled to be the custom **weight** of the model, which gives it the learnability. The more detail definition follows in the text.

1.2 Introduction

Machine learning theory facilitates the *learning process*, or rather, its formal algorithmic form. this theory's goal in mind is to inspect, evaluate, and control the process of creating a in-out 'model' - or rather, an actor, M, that correctly simulate a behaviour of some concepts, state, expressed by a function $c \in \mathcal{C}$ of such function space, contained by $\mathsf{Sup}(\cdot)[Y]\{\mathcal{C},\cdot\}$. By [V. Vapnik, 1996], we call this a *supervisor*. Emulating this 'fitting' process is expressed in the principle of empirical risk minimization.

This, is the algorithm of **empirical learning**. Aside from the algorithmic expression, its formulation in closed form, is actually the expression:

Algorithm 1: ERM

Data: $i = 1 \to \infty, n \neq \infty, \epsilon > 0$

$A_i \leftarrow \text{rand}(\{A_i^{(j)}\}_j)$;

$X \leftarrow I(\mathbb{R}) \in \mathcal{D}$;

Supervisor $= S(\cdot)$;

while $\nabla(M, S) > \epsilon$ *or* $\nabla(M, S) = b > \epsilon$ *if* $i = 1$ **do**

  $M_i(A, I) \leftarrow \mathcal{T}(A_i, I)$;

  $\nabla(M, S) \leftarrow \mathop{\mathbb{E}}_{X} \ell_n(\mathcal{T} \in M, S(X)) + k\lambda$;

  $A_{i+1} = \mathsf{Update}(\nabla(M, S), M)$;

  $i \leftarrow i + 1$;

return $\arg\min_{A_k \in \{A_j\}_j^\infty} \mathop{\mathbb{E}}_{X \in \mathcal{D}} \ell_n(\mathcal{T} \in M, S(X)) + k\lambda$;

$$\begin{aligned} \text{ERM}(\mathsf{Gen}, M, \mathsf{Sup}) &= \arg\min_{M \in \mathbb{M}} \mathop{\mathbb{E}}_{X \in \mathcal{D}} \left[\ell_n(\mathcal{T} \in M, S(X)) + \frac{1}{k} Q(\lambda) \right] \\ &= \arg\min_{A_k \in \{A_j\}_j^\infty} R(M, \mathsf{Sup}) \end{aligned} \tag{1.1}$$

The general idea is that the algorithm is the thing that facilitates the learning process, such that given a supervisor $\mathsf{Sup}(\cdot)[\cdot]$ of correct respond, a random model system $M(A, I)$ with the model's state A, such that for any $X \in \mathcal{D}$ of the dataset of the **test environment**, after successive iteration i, for $\epsilon > 0$, the behaviour of M and Sup matches. This algorithm stops only when such result is fulfilled, and is indeed, called the empirical risk minimization paradigm. Empirical in this context is because there is the difference between the exposure availability, and the actual environment domain. Thereby, it is empirical under limited data assumption.

Learning is then pretty simple in principle. Assume that the supervisor is always right, we have the setting up environment X, and we observes different behaviours from both the supervisor and the newly initialized model. We then have to find the update rule such that it improves model M's respond to match Sup, within the error margin ϵ. Such update rules have a lot of jargon in it, but it have to takes on certain properties of the resultant evaluation ∇ that compares the two, and ensure that a convergence happens, in specific circumstances. We have quite a lot of assumptions, conventions, and thereof, so let's formalize it into some definitions.

Definition 1.2.1 (Learning Model)**.** *A model M is called a **learnable model**, or **learning model** if it satisfies the following condition:*

1. *M is an reactive model, such that it is an end-to-end process (IO process).*
2. *M is an k-tuple of $(I, p_I, A, m, u_A, f, O)$ for I the input (or receiver), O the process output, p_I is the data received transformer (or preprocessing, in more usual term), A is the set inner operation algorithm, m the*

memory unit, u_A the non-flow updating, and f is the unit's flow of operation. Configuration of such setting aside from I and O is called the model's ***state****, denoted* Γ.

Normally, with our notation, we write this as:

$$M(I)\{A, m, u_A, p_I, f\}[O] = M(I)\{\Gamma\}[O] \tag{1.2}$$

The template for a model is as such, and the more detail configuration of said model is more complicated, but generally, it is enough for the current problem setting. Then, the supervisor is stated otherwise.

Definition 1.2.2 (Supervisor)**.** *A model is called a* ***supervisor*** $\mathsf{Sup}(X)[Y]$ *if it's of the same configuration as M, without u_A, treated as* ***ground truth****, that is, $\forall x \in X$, if $M(x) \neq \mathsf{Sup}(x)$, then $\mathrm{Tr}(M(x)) = F$; and its scope is* ***none****.*

The definition of the scope is pretty simple - it is similar to the amount of brain exposed to the outside observer, in this case, an arbitrary supervisor.

Definition 1.2.3 (Scope)**.** *A* ***scope*** *of a model is the amount of exposure it emit, for any given outside observer $\mathcal{O} \neq M$. A scope is* ***none*** *if it is* ***entirely hidden*** *from the observer, and* ***clear*** *if it is entirely transparent.*

Here lies the difference between supervisor and model. Even though its name gave away the mystery, it is indeed, that the supervisor is considered to be entire true. In some cases, however, we incorporate some sort of **noise** into the supervisor's response, and takes a portion of the supervisor's response as false. This is employed to test the model during learning session, such that to facilitate 'confusion of decision' into the play. Nevertheless, supervisor is the ideal model of behaviour - we takes on the supervisor's response as the ideal version to be learned from. But because the inner structure is not exposed, then the supervisor itself can be expressed as a formal dataform Y contains only its response to particular $x \in X$ of the generator. The generic dataset $\mathcal{D}$ is then a 2-tuple of both $(\mathsf{Gen}[X], \mathsf{Sup}(X))$, shortly as $\{X, Y\}$.

But can the model M correctly learn the behaviour of Sup? We takes it as doubtfully probable. There are quite a few aspects on this. The first thing is the *time-probable learning* criteria, which is, perhaps expressed by PAC-learning in chapter 3, which again, establish the time-restricted probable learning guarantee for finite time, and measurable polynomial time complexity. The second aspect is the *model complexity* criteria, of which is the ability of the model to mutate its behaviour, to a wide range of configuration. Such configuration is hard to define, and more fuzzy definition is adopted. For example, for a linear model $\mathsf{Lin}(X)\{\boldsymbol{w}, b\}[Y]$, the configuration space spans the entire $\mathbb{R}^n$ space, for a given n-plane (or hyperplane). The difference between model expression hence, is more than just the number of parameters, or certain just saying non-linearity into it. But, for now, we would adopt an assumption of forth, to generally ignore the probable of unlearnable.

Assumption 1.2.0.1 (Model learnability)**.** Any model can **approximate** the supervisor's response. If the approximation is within $\epsilon > 0$ of arbitrary value, then it is totally learnable.

Assumption 1.2.1 (Totally learnable)**.** For all supervisor Sup, in the space of model configuration space $\mathcal{M}[\{\mathbb{M}\}]$, there exists a configuration $\mathbb{M}$ such that there exists a model M learnable of the supervisor. or $R(M, \mathsf{Sup}) < \epsilon$.

Of course, not all concept (we often interchange the word concept and supervisor with each other, since the other is quite a mouthful to speak of) can be learned, as least within efficient time-complexity, or within plausible model approximation error. We then only consider, and only test on such concept that can be well-approximated, or totally learnable. Interestingly, a supervisor is learnable, if it is not **trivial**. Hence, if the model figured to learn patterns and analyze structure of the non-structured, trivial and unlearnable supervisor, we can theorize it being **overfitted** - a general concept related to the inability to *generalize* of the model. This idea is exhibited in the **Rademacher complexity** concept, which use the same logic. However, such topic is reserved for later on, so for now, we only care about the regime of learnable concepts.

We define, then, the **learning process** as a procedure $\mathcal{T}$ that takes $\mathsf{Gen}[X]$, giving it to $M(X)[Y']$ and evaluate the metric of correspondence $\mathcal{R} : \ell$ with respect to $\mathsf{Sup}(X)[Y]$. This procedure then includes the updating process

$$\mathsf{Update}(\nabla(M, S), M) \to A_{i+1}$$

, where ∇ is usually the mathematical expression of closeness - it measures the nessesary change required, under certain formulation of its form, for the sequence

$$M\{\Gamma_1\}, M\{\Gamma_2\}, \ldots, M\{\Gamma_n\}$$

converges to Sup. All of this, is compressed to a few learning problem generally conceived. Those includes *generalization learning*, *approximation learning* or fitness learning, *efficient learning*, and perhaps, *adaptive learning*. Those problem, then raised the following questions that learning theory needs to address [Vapnik 1996][1]:

1. What are (necessary and sufficient) conditions for consistency of a learning process based on the ERM principle?
2. How fast is the rate of convergence of the learning process?
3. How can one control the rate of converggence (the generalization ability) of the learning process?
4. How can one construct algorithms that can control the generalization ability?

and several more operational questions.

This then must be raised, with its arguments in it. There are quite a few things I want to talk about the adaptive learning problem and others, but for now, that is appropriately enough, I guess.

[1] Vladimir N. Vapnik, The Nature of Statistical Learning Theory (Second Edition), Springer-Verlag

1.3 Analytical View

We adopt, then a more detail viewport of those problems and situations. What we have done above is the conceptual descriptions, the underlying form, mathematical representation is only listed, not concretely defined under certain situation of interest. But they are important for our argument later on. For now, the general view would be targeted of provision.

Taking some respective comment above on the topic of model analysis, we would take on a more formal, less conceptual approach to subjects like scole and exposure. Furthermore, we also formally define a lot of important concept, such as generalization and efficient learning. All of this, would eventually helps later analysis on double descent, and *potentially*, n-fold descent.

1.3.1 What is, then, double descent

The phenomenon dubbed **double descent** is defined more empirically than not, and is rather novel in its analysis as a whole. Its basis branched out from the subject of **error analysis**, where analysis is taken for a correction system like machine learning models, within the space of error evaluator. There, for such to be succeed, the model in question must be an open system, with formattble output to be evaluated, and there exists a ground truth for it to reference to. The complexity of the model, including its procedure, memories, representation, everything, is considered a portion of such analysis, simply because unlike stationary system, a dynamic system like learning model changes its representation over time. This representation and the computational procedure, which are mutable, are responsible for the beheviour of the error term. Thereby, the axis of complexity-error is a reasonable consideration in such view. The classical theory theorized that in the form of bias-variance tradeoff - a deeper breakdown of the term, by using two proxies that measure the instability of the model (error fluctuation), and the relative expressiveness of such model (complexity), with tradeoff, hinted at a sweetspot in between.

Double descent breaks this sweetspot theorem. By the sweetspot, we theorized that for complexity go to extreme, the variance term shoot to infinity, or rather, the instability of the system, i.e. the error, goes to infinity, and hence failed the learning task. Double descent breaks this by subjects the complexity axis to convergence, or rather:

Definition 1.3.1 (Double descent)**.** *For a given model $M(I)\{\dots\}[O]$, with its complexity $\mathsf{Comp}(M) > 0$, then:*

1. *The model's error analytical form $R(M, \mathsf{Sup})$ always **converges** to 0.*
2. *There exists two anomaly a_1, a_2 such that $\max_{M \in \mathbb{M}} R(M, \mathsf{Sup}) = \{a_1, a_2\}$.*

The depth of the region between two peaks is trivial, since for either way, they are the highest point of the landscape of error. Double descent remove the restriction of a infinite variance wall on the right side, while keeping the left side relatively intact - for simply too minimal complexity, under given complex concept c, the error term shoots to infinity.

A question. What if bias-variance is not about error, but about stability and time-dependent response structure?

We haven't talked about one thing though. Why all of this matters? What does the error term represents?

2 Graph Theory

Analysis in depth

F. Amane, K. Kenzaki

Definition 2.0.1 (Graph)**.** *A **simple graph** is a pair (V, E), where V is a finite set, and where E is a subset of $\mathcal{P}_2(V)$, with $\mathcal{P}_2(V)$ is the set of all 2-element subsets of V.*

A quick way to refers to certain elements of the graph is as followed.

Definition 2.0.2 (Notations)**.** *Let $G = (V, E)$ be a simple graph.*

1. *The set V is the vertext set of G, denoted $\mathrm{V}(G)$. The element of V are called the **vertices** (or **nodes**) of G.*
2. *The set E is called the **edge set** of G, denoted $\mathrm{E}(G)$. The element of E are called the **edges** of G. We denoted uv for $\{u, v\}$ as connections between u adn v.*

Certain properties of such simple graph case follows:

Definition 2.0.3 (Connectedness)**.** *If the two graphs are $G_1 = (V(G_1), E(G_1))$ and $G_2 = (V(G_2), E(G_2))$ where $V(G_1)$ and $V(G_2)$ are disjoint, then their **union** $G_1 \cup G_2$ is the graph with vertex set $V(G_1) \cup V(G_2)$ and edge family $E(G_1) \cup E(G_2)$.*

Definition 2.0.4 (Adjacency)**.** *Two vertices u, v are said to be **adjacent** to each other if $wv \in E$. uv is then said to **join** u with v. The **neighbour** of a vertex $v \in V$ is then the set of all $u \in V$ such that $uv \in E$.*

Definition 2.0.5 (Graph Isomorphism)**.** *Let G and H be two simple graphs. A **graph isomorphism** (or **isomorphism**) from $G \to H$ is the bijection $\phi : \mathrm{V}(G) \to \mathrm{V}(H)$ that preserves edges, i.e., for any two vertices u and v of G, we have*

$$(uv \in \mathrm{E}(G)) \iff (\phi(u)\phi(v) \in \mathrm{E}(H)) \tag{2.1}$$

We say that G and H are isomorphic (denoted $G \cong H$) if there exists a graph isomorphism from G to H.

We also list several types of graphs, specifically, the **subgraph**, null /complete graph, path /cycle graph, Kneser graphs, diagraphs and multigraph (or general graph).

Amane

What if, for a general graph, there are **multiple non-equal edges and connections**, such that each node function more as a small subgraph, or either an itermediate I/O node?

Definition 2.0.6 (Subgraph). *A subgraph of a graph X is a graph Y such that $\mathrm{V}(Y) \subseteq \mathrm{V}(X)$ and $\mathrm{E}(Y) = \mathrm{E}(X)$. If $\mathrm{V}(Y) = \mathrm{V}(X)$ then Y is a spanning subgraph of X. A subgraph Y of X is an induced subgraph if two vertices of $\mathrm{V}(Y)$ are adjacent in Y if and only if they are adjacent in X.*

One of the important definition for subgraph, especially for those of recommendation system, and graph construction network, is the notion of **induced subgraph**.

Definition 2.0.7 (Induced subgraph). *Let $S \subset V$. The induced subgraph of G on the set S denotes the subgraph*

$$(S, E \cap \mathcal{P}_2(S)) \tag{2.2}$$

of G. In other words, it denotes the subgraph of G whose vertices are the elements of S, and whose edges are precisely those edges of G whose both endpoints belong to S.

Definition 2.0.8 (Null graph). *A graph whose edge-set $E = \varnothing$ is a **null graph**. we denote the null graph on n vertices by $N_n = (V, \varnothing)$.*

Definition 2.0.9 (Complete graph). *A simple graph $G = (V, E)$ in which each pair of distinct vertices are adjacent is a **complete** graph. We denote this by K_n. For graph K with $\mathrm{V}(G) = n$, we have:*

$$V(K_n) = \frac{n(n-1)}{2} \tag{2.3}$$

Definition 2.0.10 (Path graph). *For each $n \in \mathbb{N}$, we define the n-th **path graph** P_n to be the simple graph*

$$P_n = (\{1, 2, \ldots, n \{i(i+1) \mid 1 \leq i < n\}) \tag{2.4}$$

This graph has n vertices and $n-1$ edges (unless $n = 0$)

Definition 2.0.11 (Cycle graphs). *For each $n > 1$, we define the n-th **cycle graph** C_n to be the simple graph*

$$C_n = (\{1, \ldots, n\}, \{i(i+1) \mid 1 \leq i < n\} \cup \{1n\}) \tag{2.5}$$

This graph has n vertices and n edges, unless $n = 1$, then it has only 1 edge. For multigraph, however, the C_2 graph can have two edges.

2.1 Algebraic representation

The tool of computation is still linear algebra, for any large computational system under the neural network system. Hence, there are several useful way to represent such graph on linear algebraical notation and system.

Definition 2.1.1 (Adjacency matrix). *Let G be a simple graph with $V(G) = \{1, \ldots, n\}$ and $E(G) = \{e_1, \ldots, e_m\}$. The adjacency matrix of G, denoted $A(G)$, is the $n \times n$ matrix defined as*

$$A_{dj} = A \in \mathbb{R}^{n \times n} : \quad A_{ij} = \begin{cases} 1 & v_i v_j \in E, i \neq j \\ 0 & \textit{otherwise} \end{cases} \tag{2.6}$$

The modification (somewhat familiar) of the adjacency matrix is by the notion of **incidence matrix**, which takes into account the case of directed graph.

Definition 2.1.2 (Incident matrix)**.** *Let G be a graph with $\mathrm{V}(G) = \{1, \dots, n\}$ and $\mathrm{E}(G) = \{e_1, \dots, e_m\}$. Suppose each edge $e \in \mathrm{E}(G)$ is an orientation, which is arbitrary but fixed. The (vertex-edge) **incidence matrix** of G, denoted $Q(G)$, is the matrix of size $n \times m$, defined as followed:*

$$Q(G) \in \mathbb{R}^{n \times m} : \quad Q_{ij} = \begin{cases} 1 & (i, e_j) \in G, e_j \text{ originates at } i \\ -1 & (i, e_j) \in G, e_j \text{ terminates at } i \\ 0 & \text{otherwise (both)} \end{cases} \tag{2.7}$$

For undirected graph, the matrix is defined as

$$Q(G) \in \mathbb{R}^{n \times m} : \quad Q_{ij} = \begin{cases} 1 & \exists k, e_j = \{v_i, v_k\} \\ 0 & \text{otherwise} \end{cases} \tag{2.8}$$

For multidirected graph (or multigraph), we use the **Laplacian matrix** to represent such graph in matrix form. Notice that of the coming definition, the Laplacian matrix does not encode incidences in its matrix, as it is not the main focus of the definition.

Definition 2.1.3 (Laplacian matrix)**.** *Let G be a graph with $\mathrm{V}(G) = \{1, \dots, n\}$ and $\mathrm{E}(G) = \{e_1, \dots, e_m\}$. The **Laplacian matrix** of G, denoted $L(G)$, is the $n \times n$ matrix defined as followed:*

$$L(G) \in \mathbb{R}^{n \times m} : L_{ij} = \begin{cases} 0 & i \neq j \\ deg(i) & i = j \end{cases} \tag{2.9}$$

The Laplacian matrix and the incidences matrix can be related, with the fixed orientation assumption, by

$$L(G) = Q(G)Q'(G) \tag{2.10}$$

where $Q'(G)$ is the transpose of Q. This identity suggests that the Laplacian might depends on the orientation, although it is evident from the definition that the Laplacian is independent of orientation.

We present more subsidiary theory on graph theory.

Theorem 2.1.1 (Handshaking Lemma)**.** *In any graph G, the sum of all the vertex-degree is an even number. Or, the vertex degree add up to twice the number of edges:*

$$\sum_{v \in V(G)} \deg_G(v) = 2|E(G)|$$

Proof by induction. The base case here is $m = 0$. The graph with no edge then $\deg(v) = 0$.

Assume that the degree sum formula holds for all $(m-1)$-edge graphs. Let G be a graph with $m \geq 1$ edges, and let xy be any edge of G. We can

apply the inductive hypothesis to $G - xy$ (the graph we get from **deleting** xy from G), a graph with $m - 1$ edges.

Both x and y have one extra incident edge in G they don't have in $G - xy$ - the edge xy itself. Hence

$$\deg_G(x) = 1 + \deg_{G-xy}(x)$$

and

$$\deg_G(y) = 1 + \deg_{G-xy}(y)$$

For any other vertex v, $G - xy$ and G have the same number of edges, so we have

$$\deg_G(v) = \deg_{G-xy}(v)$$

Furthermore, $G - xy$ and G have the same set of vertices. So if we add up the vertex degrees in $G - xy$ and G, the result is that

$$\sum_{v \in V(G)} \deg_G(v) = 2 + \sum_{V \in V(G-xy)} \deg_{G-xy}(v)$$

Applying the inductive hypothesis, we get that the degree sum in $G - xy$ is $2(m - 1)$, so the degree sum in G is $2(m - 1) + 2 = 2m$. By induction, the degree sum formula holds all graphs. □

2.2 Permutation invariance and equivariance

In mathematical terms, any function f that takes an adjacency matrix $\mathbf{A}$ as input should ideally satisfy one of the two following properties:

$$f(\mathbf{PAP})^\top = f(\mathbf{A}) \quad \text{(Permutation Invariances)} \tag{2.11}$$

$$f(\mathbf{PAP}^\top) = \mathbf{P}f(\mathbf{A}) \quad \text{(Permutation Equivariance)} \tag{2.12}$$

where $\mathbf{P}$ is a permutation matrix. Permutation invariance means that the functin does not depend on arbitrary ordering of the rows/coulmns in the adjacency matrix, while permutation equivariance means that the output of f is permuted in a consistent way when we permute the adjacency matrix.

2.3 Graph Fourier Transform

In graph theory, especially for the **spectral graph theory**, we present an encoding standard, called the **graph Laplacian**. Then we will focus on the Fourier analysis of such grpah representation, such as the *graph Fourier transform*. For the purpose, however, we believe that the spectral graph theory alongside its analysis can get us to the layered encoding appropriated of analysing the aggregation contour that the graph is taken into.

Graph Laplacian

Given the adjacency matrix $A \in \{0, 1\}^{n \times n}$ and degree matrix $D \in \mathbb{Z}^{n \times n}$, for a graph $G = (V, E)$, we define the graph Laplacian as the matrix L such that:

$$L := D - A \tag{2.13}$$

Intuitively, it is the analog for the Laplacian operator $\Delta f(x) = \nabla \cdot \nabla f(x)$. In this form, it is quite different from the above Laplacian matrix, but the same idea applies. In this form, it is perhaps better for analysis, though.

We list several conclusive properties of the Laplacian matrix:

1. L is a real symmetric matrix, there by $\lambda_k > 0, \forall \lambda \in \{\lambda_k\}_{k=0}^{N-1}$ with associated orthornomal eigenvectors $\{\varphi_k\}_{k=0}^{N-1}$.

2. If G is finite and connected, then

$$0 = \lambda_0 < \lambda_1 < \cdots < \lambda_{N-1} \tag{2.14}$$

3. The spectrum of the Laplacian $\sigma(L)$ is fixed but one's choice of eigenvectors $\{\varphi_k\}_{k=0}^{N-1}$ can vary. Throughout the paper, we assume that the choice are fixed.

4. Let Ψ denote the $N \times N$ matrix where the k-th column is precisely the vector φ_k, then we can easily show that $\varphi_0 = 1/\sqrt{N}$.

The Fourier Transform

Let $G(V, E)$ be a weighted graph, $\mathbf{L}$ be its corresponding graph Laplacian, and $f : V \to \mathbb{R}$ a function defined on the vertices of G. The **Graph Fourier Transform** (GFT) is defined as

$$\mathcal{GF}[f](\lambda_l) = \hat{f}(\lambda_l) = \langle f, \varphi_l \rangle = \sum_{i=1}^{n} f(i)\varphi_l(i) \tag{2.15}$$

The inverse Fourier transform is then given by

$$f(n) = \sum_{i=0}^{N-1} \hat{f}(\lambda_l)\varphi_l(n) \tag{2.16}$$

This also leads it to the graph convolution (used in graph convolutional network) as

$$f * g(n) = \sum_{l=0}^{N-1} \hat{f}(\lambda_l)\hat{g}(\lambda_l)\varphi_l(n) \tag{2.17}$$

for $f, g : V \to \mathbb{R}$ as functions on the continuous domain.

3 PAC Learning Theory

A rigorous treatment of machine learning theory

FUJIMIYA AMANE

Probably Approximately Correct learning (or PAC-learning), is a way of analytically, without direct description, decipher the coming ability of a learning model, and its guarantee of success in finite space and time, of plausible complexity and constraint. Hence, PAC-learning is more of a **guarantee guide** and feasibility evaluator, for such acting model on its own. This short section will introduce the concept of such, under the *concept-hypothesis framework*, to see how it works and further implications.

3.1 The PAC learning framework

3.1.1 Setting

Statistical learning incorporates a formal learning structure to the standard machine learning procedure. This is accomplished by formalizing the notion of concepts, hypothesis, inference spaces and inspection methods for a well-formed system[MRT18], SLT$(\mathcal{X}, \mathcal{Y}, \mathcal{C})$.

We denote $\mathcal{X}$ as the input space, $\mathcal{Y}$ as the target space. For the purpose of this example, we let $\mathcal{Y} = \{0, 1\}$. A concept $c : \mathcal{X} \to \mathcal{Y}$ is a mapping from the input space to target space, and C denotes the concept class of all concepts. The sample space containing all examples, S is assumed to be an i.i.d. (independently identically distributed) dataset.

The learning problem is then formulated as below:

1. Provided a learner $\mathcal{L}$, hypothesis set of possible concepts $\mathcal{H}$ such that $\mathcal{H} \setminus C \neq \varnothing$.

2. The learner received a dataset $S = \{x_1, x_2, \ldots, x_m\}$ drawn from a distribution D which is unknown, and is i.i.d.

3. The learner received a label set $\{c(x_1), c(x_2), \ldots, c(x_m)\}$ which are based on a specific target concept $c \in C$.

The task of such learning agent is then to choose a hypothesis $h_S \in H$ such that it has a small generalization error with respect to the concept c, using the dataset S and the labels. Note that the class concept is hidden, as always since it is the learning objective. In such formalization, we have two parameters of which is important, as put as objective: generalization and empirical error.

Definition 3.1.1 (Generalization error). *Given a hypothesis $h \in H$, a target concept $c \in C$, and an underlying distribution D, the generalization error or risk of h is defined as:*

$$R(h) = \Pr_{x \sim D}[h(X) \neq c(x)] = \mathop{\mathrm{E}}_{x \sim D}[1_{h(x) \neq c(x)}]$$

Where 1_w is the indicator function of the event w.

This generalization error is not directly accessible to the learner since both the distribution $\mathcal{D}$ and the target concept c are unknown.

Definition 3.1.2 (Empirical error). *Given a hypothesis $h \in H$, a target concept $c \in C$, and a sample $S = (x_1, \ldots, x_m)$, the empirical error or empirical risk of h is defined by*

$$\hat{R}(h) = \frac{1}{m} \sum_{i=1}^{m} 1_{h(x_i \neq c(x_i))}$$

Thus, the empirical error of $h \in \mathcal{H}$ is its average error over the sample S, while the generalization error is its expected error based on the distribution $\mathcal{D}$. One cannot know the generalization error as above-mentioned, but the empirical error can be founded accordingly, as it is limited into the range of sample S. In fact, we might call it as a narrow-training error, because of it's narrow properties when it comes to universal concept class.

This naturally leads to the algorithm of *empirical risk minimization*, of which we learn h_S by attempting to minimize $\hat{R}(h_S)$ as a surrogate for $R(h_S)$.

3.1.2 PAC learning

We denote by $O(n)$ an upper bound on the cost of the computational representation of any element $x \in \mathcal{X}$ and by $size(c)$ the maximal cost of the computational representation of $c \in C$. For example, x may be a vector in $\mathbb{R}^n$, for which the cost of an array-based representation would be in $O(n)$.

Definition 3.1.3 (PAC-learning). *A concept class C is said to be PAC-learnable if there exists an algorithm $\mathcal{A}$ and a polynomial function $poly(\cdot, \cdot, \cdot, \cdot)$ such that for any $\epsilon > 0$ and $1 > \delta > 0$, for all distribution D on $\mathcal{X}$ and for any target concept $c \in C$, the following holds for any sample size $m \geq poly(1/\epsilon, 1/\delta, n, size(c))$:*

$$\Pr_{S \sim D^m}[R(h_S) \leq \epsilon] \geq 1 - \delta \tag{3.1}$$

If $\mathcal{A}$ further run in $poly(1/\epsilon, 1/\delta, n, size(c))$, then C is said to be efficiently PAC-learnable. When such algorithm $\mathcal{A}$ exists, it is called a PAC-learning algorithm for C.

From such definition, one can say that $poly(1/\epsilon, 1/\delta, n, size(c))$ is the sample size, and we say that algorithm $\mathcal{A}$ learns class C in the consistency model if given any set of labeled example S, the algorithm produces a concept $c \in C$ consistent with S if such exists.

Remark 3.1.1. The algorithm runs in polynomial time. Note that if the running time of the algorithm in $1/\epsilon$ and $1/\delta$, then the sample size m must also be polynomial. The assumption here is that the algorithm must read in data, thus it is $\Omega(m)$, of which $m \geq poly(1/\epsilon, 1/\delta, n, size(c))$. If m is not a polynomial, the algorithm cannot be runs in polynomial time as well.

Regarding the consistency of the algorithm $\mathcal{A}$, it needs only

$$\frac{1}{\epsilon}\left(\ln|C| + \ln\frac{1}{\delta}\right)$$

examples to output an $h \in H$ for error at most, ϵ, and probability of at least $1-\delta$. This applies so long as this quantity is in polynomial with $size(c)$ and n.

A concept classes C is thus PAC-learnable if the hypothesis returned by the algorithm after observing a number of points polynomial in $1/\epsilon$, and $1/\delta$ is approximately correct, with error being at most ϵ, and high probability of such correction:

$$\underset{h\in\mathcal{H}}{\operatorname{argmin}} \Pr_{S\sim D^m}[R(h_S) \leq \epsilon] = 1-\delta$$

which justifies the PAC terminology, $\delta > 0$ is used to define the confidence $1-\delta$, and $\epsilon > 0$ defines the accuracy $1-\epsilon$. If we let the parameter δ stand alone, then it is referred to as the inverse confidence, or probability of non-representative hypothesis. More clearly speaking, ϵ is the error, and δ represents the probability that $R(h) \geq \epsilon$, or the generalization error o the distribution $\mathcal{D}$ is more than the accepted error.

3.2 Generalization Bounds

To understand generalization bound, we first need to introduce the categorization of the hypothesis based of one criterion: consistency.

3.2.1 Consistent Learning

We'll define the notion of a consistent learning algorithm, or consistent learner[Kan17], for a concept class C and hypothesis h.

Definition 3.2.1 (Consistent Learner)**.** *We say that an algorithm is a consistent learner for a concept class C using hypothesis class $\mathcal{H}$, if for all n, for all $c \in C_n$ and for all m, given*

$$S = \{(x_1, c(x_1)), (x_2, c(x_2)), \ldots, (x_m, c(x_m))\}$$

as input, $x_i \in X_n$ outputs, then the algorithm L outputs a hypothesis $h \in \mathcal{H}_n$ such that

$$h(x_i) = c(x_i), i = 1, \ldots, m$$

Then, a consistent hypothesis is the hypothesis that is consistent with all the training data provided to it from the concept c. The following section contains the debate between consistent hypothesis (i.e., for example, admitting no error on training data, almost perfect - maybe even so), and inconsistent hypothesis, of which have general defects from the actual concept, or even different by a margin.

3.2.2 Finite H, consistent hypothesis

We will consider the general sample complexity bound, or equivalently, a generalization bound for consistent hypothesis in the case where the cardinality $|\mathcal{H}|$ is finite. We will assume, as such, that the target concept c is in $\mathcal{H}$.

Theorem 3.2.1 (Learning bound - finite $\mathcal{H}$, consistent case)**.** *Let H be a finite set of functions mapping from $\mathcal{X} \to \mathcal{Y}$. Let $\mathcal{A}$ be an algorithm that for any target concept $c \in H$ and i.i.d. samples S returns a consistent hypothesis H_S, such that $\hat{R}(h_S) = 0$. Then for any $\epsilon, \delta > 0$, the inequality $\Pr_{S \sim D^m}[R(h_S) \leq \epsilon] \geq 1 - \delta$ holds if*

$$m \geq \frac{1}{\epsilon}\left(\log |H| + \log \frac{1}{\delta}\right)$$

This sample complexity result admits the following equivalent statement as a generation bound: for any $\epsilon, \delta > 0$, with probability at least $1 - \delta$,

$$R(h_S) \leq \frac{1}{m}\left(\log |\mathcal{H}| + \log \frac{1}{\delta}\right) \tag{3.2}$$

Proof

Fix $\epsilon > 0$. We do not know which consistent hypothesis $h_S \in H$ is selected by the algorithm $\mathcal{A}$. This depends on S. Therefore, we need to give a uniform convergence bounds, that is, a bound that holds for the set of all consistent hypothesis. Thus, we will bound the probability that some $h \in H$ would be consistent and have error more than ϵ, denoted by:

$$\Pr[\exists h \in H : \hat{R}(h) = 0 \wedge R(h) > \epsilon]$$

This is equal to:

$$\Pr(Q) = \Pr[(h_1 \in H, \hat{R}(h_1) = 0 \wedge R(h_1) > \epsilon) \vee (h_2 \in H, \hat{R}(h_2) = 0 \wedge R(h_2) > \epsilon) \vee \dots]$$

For shorthand notation, we set $\mathcal{H}_\epsilon = \{h \in \mathcal{H} : R(h) > \epsilon\}$. Hence,

$$\Pr(Q) = \mathbb{P}[\exists h \in \mathcal{H}_\epsilon : \hat{R}_S(h) = 0]$$

We can see that

$$\Pr(Q) \leq \sum_{h \in H} \Pr[\hat{R}(h) = 0 \wedge R(h) > \epsilon]$$

by union bound, and by conditional probability,

$$\Pr(Q) \leq \sum_{h \in H} \Pr[\hat{R}(h) = 0 \mid R(h) > \epsilon]$$

Now, consider any $h \in H$ with $R(h) > \epsilon$. Then the probability that h is consistent on training sample S without error, can be bounded as:

$$\Pr[\hat{R}(h) = 0 \mid R(h) > \epsilon] \leq (1 - \epsilon)^m$$

with m the number of samples. This implies that:

$$\Pr[\exists h \in H : \hat{R}(h) = 0 \wedge R(h) > \epsilon] \leq |H|(1-\epsilon)^m \leq |H|e^{-m\epsilon}$$

Setting the right-hand side to equal to δ, we gain the above statement. Hence, proved.

The theorem shows that when the hypothesis set $\mathcal{H}$ is finite, a consistent algorithm $\mathcal{A}$ is a PAC-learning algorithm, since the sample complexity is dominated by a polynomial in $1/\epsilon$ and $1/\delta$. The generalization error of consistent hypothesis is upper bounded by a term that decrease w.r.t m. The decreases rate of $O(1/m)$ is guaranteed by this theorem.

3.2.3 Examples

Boolean Conjunction

Consider the concept class $\mathcal{C}_n$ of conjunction of at most n Boolean literals $x_1, \ldots, x_n$.

A Boolean literal is either a variable $x_i, i \in [n]$ or its negation $\bar{x}_i$. For $n = 4$, this might be $x_1 \wedge \bar{x}_2 \wedge x_4$.

A simple algorithm for finding a consistent hypothesis is thus based on positive examples and consists of the following: For each positive example $(b_1, .., b_n), i \in [n]$, if $b_i = 1$ then $\bar{x}_i$ is ruled out as a possible literal in the concept class and if $b_i = 0$, hen x_i is ruled out. The conjunction of all of the literal not ruled out is a hypothesis consistent with the target.

We have $|\mathcal{H}| = |\mathcal{C}_n| = 3^n$ since each literal can be either 1,0 or not chosen. Plugging this into the complexity bound for $\epsilon > 0$ and $\delta > 0$,

$$m \geq \frac{1}{\epsilon}\left(n \log(3) + \log\left(\frac{1}{\delta}\right)\right)$$

Thus, the class of conjunction of at most n Boolean literals is PAC-learnable.

Universal Concept Classes

Consider the set $\mathcal{X} = \{0,1\}^n$ of all Boolean vectors with n components, and let $\mathcal{U}_n$ be the concept class formed by all subsets of $\mathcal{X}$. Is this concept PAC-learnable? To guarantee a consistent hypothesis, the hypothesis class must include the concept class, thus $|\mathcal{H}| \geq |\mathcal{U}_n| = 2^{(2^n)}$. We are given then

$$m \geq \frac{1}{\epsilon}\left(2^n \log 2 + \log\frac{1}{\delta}\right) \geq O\left(\log|\mathcal{H}| + \log\frac{1}{\delta}\right)$$

Hence, it is not guaranteed by the theorem that it is PAC-learnable. In fact, recalling the definition of PAC-learning, recall that for a concept class $\mathcal{U}_n$ to be PAC-learnable, it needs that

$$\Pr_{S \sim D^m}[R(\mathcal{U}_n) \leq \epsilon] \geq 1 - \delta, \quad m \geq \frac{1}{\epsilon}\left(\log|\mathcal{C}| + \log\frac{1}{\delta}\right), \mathcal{U}_n \in \mathcal{C}$$

But here, assuming that the error is set for $\epsilon > 0$ and $\delta > 0$, the polynomial exceeds the bound, hence it is not PAC-learnable.

3.2.4 Finite hypothesis sets H - inconsistent case

In the most general case, there may be no hypothesis in $\mathcal{H}$ consistent with the labeled training sample. This, in fact is the typical case in practice, where the learning problems may be somewhat difficult or the concept classes more complex than the hypothesis set used by the learning algorithm.

To derive the learning guarantees in the more general setting, we would use the following corollary, of which relates the generalization error and empirical error of a single hypothesis.

Corollary 3.2.2. *Fix $\epsilon > 0$. Then, for any hypothesis $h : X \to \{0, 1\}$, the following inequalities hold:*

$$\mathop{\mathbb{P}}_{S \sim \mathcal{D}^m}\left[\hat{R}_S(h) - R(h) \geq \epsilon\right] \leq \exp\left(-2m\epsilon^2\right) \tag{3.3}$$

$$\mathop{\mathbb{P}}_{S \sim \mathcal{D}^m}\left[\hat{R}_S(h) - R(h) \leq -\epsilon\right] \leq -\exp\left(-2m\epsilon^2\right) \tag{3.4}$$

By the union bound, this implies the following two-sided inequality:

$$\mathop{\mathbb{P}}_{S \sim \mathcal{D}^m}\left[|\hat{R}_S(h) - R(h)| \leq \epsilon\right] \leq 2\exp\left(-2m\epsilon^2\right) \tag{3.5}$$

This result is proved using a derivation from Hoeffding's inequality. Setting the right-hand side of (5) to be equal to δ and solving this for ϵ yields immediately the following bound for a single hypothesis.

Corollary 3.2.3 (Generalization bound - single hypothesis)**.** *Fix a hypothesis $h : \mathcal{X} \to \{0, 1\}$. Then, for any $\delta > 0$, the following inequality holds with probability at least $1 - \delta$:*

$$R(h) \leq \hat{R}_S(h) + \sqrt{\frac{\log 2/\delta}{2m}} \tag{3.6}$$

Can we use this corollary to bound the generalization error of the hypothesis h_S returned by a learning algorithm when training on a sample S? No, since h_S is a random variable depends on the training set S, rather than being fixed. Unlike the case of a fixed hypothesis for which the expectation $\mathbb{E}[\hat{R}_S(h_S)]$ is the generalization error, the generalization error $R(h_S)$ is a random variable and in general distinct from the expectation, which is a constant.

Thus, as in the proof for the consistent case, we need a uniform convergence bound, that holds with high probability for all hypotheses $h \in \mathcal{H}$.

Theorem 3.2.4 (Learning bound - finite $\mathcal{H}$, inconsistent case)**.** *Let $\mathcal{H}$ be a finite hypothesis set. Then, for any $\delta > 0$, with probability at least $1 - \delta$, the following inequality holds:*

$$\forall h \in \mathcal{H}, \quad R(h) \leq \hat{R}_S(h) + \sqrt{\frac{\log |\mathcal{H}| + \log 2/\delta}{2m}} \tag{3.7}$$

Thus, for a finite hypothesis set $\mathcal{H}$,

$$R(h) \leq \hat{R}_S(h) + O\left(\sqrt{\frac{\log_2 |\mathcal{H}|}{m}}\right)$$

The term $\log |\mathcal{H}|$ can be interpreted as the number of bit needed to represent $\mathcal{H}$. A larger sample size m guarantees better generalization, and the bound increases with $|\mathcal{H}|$, but only logarithmically.

Note that the bound suggests seeking a trade-off between reducing the empirical error versus controlling the size of the hypothesis set: a larger hypothesis set is penalized by the second term but could help reduce the empirical error, that is the first term. But for a similar empirical error, it suggests using a smaller hypothesis set. This can be viewed as an instance of the so-called *Occam's Razor*, of which can be said as: *Plurality should not be posited without necessity*, or, the simplest explanation is best.[MRT18]

4 Phase Transition in Networks

Seems weird, and definitely useful somewhat.

FUJIMIYA AMANE

This section analyse and capture the general idea, as well as application of such in personal setting, of the paper *Zeroth, first and second-order phase transition in deep neural networks* [Liu23]. Well, sort of. The paper is entirely ubiquituous to me, and to be honest, it shows relevant interests of forth.

4.1 General

In general idea, the paper takes in two ideas: phase transition in physics, and entropic behaviour, still is physics, but it gets better overtime. THe claim here is that for similar phenomenology in statistical physics, applying to both deep and shallow neural network, as it is that the phase transition behaviour is not **unique** to any kind, then phase transitions is the cause for the fluctuation and the odd behaviour of model complexity, and prediction error. In other words - something akin to the hypothesis on the dual complexity-error theorem, but explained using statistical physics interpretation.

Here, they investigate the loss landscape (as always) of DNN, and prove that there exists phase transitions with a striking (their word, not mine) similarity to classical physics. The proposed model is interesting - multilayer linear network of stochastic neurons, and L_2 regularization, and from this, they work it to

1. They identified an order parameter and effective landscape that describes the phase transition between a trivial phase and a feature learning phase.
2. Proved that:
   (a) depth-0 nets (linear regression) do not have a phase transition.
   (b) depth-1 nets have the second-order phase transitions.
   (c) depth-D nets have the first-order phase transition for $D > 1$
   (d) infinite-depth nets have the zeroth-order phase transition.
3. Such networks approximate commonly used nonlinear networks of the same depth and connectivity structure.

The final third big point is oddly similar to the claim made by Cybenko's universal approximation theorem. The general idea, however, is valid. You might want to see why I thought it to be so, but overall, it is nice nonetheless. But let's see if the claim is valid, and how do they prove it to be such. Are these another misinterpretation?

> You might think it's weird why there is statistical physics, or overall physics in the field of machine learning. Those two are definitely distinct, right? Well, you might be wrong, quite wrong in fact. Almost **everything** in machine learning is dynamic. Unlike older models, and general mathematical modelling, machine learning model stays in a mutable state of limited functionality, i.e. its learning capability exists, albeit limited by certain aspect of the model itself. Thereby, we can, indeed, exploit this view and treat a stable model, in a more general setting such that the overall investigation focus on the dynamic of the system as a whole. This is why thermodynamic is so useful in machine learning - it emulates, or rather, permits the analysis of the total working environment of the learning model, which already is both stochastic (somewhat), procedural, algorithmic in nature, and statistically (un)stable during operation.

4.2 Setting

We assume the following framework

$$\begin{aligned} \ell(w,\gamma) &= \mathbb{E}_x[\ell_0(w,x)] + \gamma R(w) \\ L(\gamma) &= \min_w \ell(w,\gamma) \\ w_* &= \arg\min_w \ell(w,\gamma) \end{aligned} \tag{4.1}$$

The hyperparameter γ affects the model such that, by the vague definition of well-behaved, the model is well-behaved, if γ is easy to tune. Here, they claim that phase transition is responsible for the case there γ is harder to tune. This is similar to the treatment of zero-temperature theory "that ignores the stochastic effects" due to the noise in the stochastic gradient Langevin dynamics.[1]

Here, γ is described similar to the nontemperature macroscopic thermodynamic variable, akin to pressure. In such view, optimization of the objective thus involves balancing the prediction error and model complexity.

4.2.1 Clarification I

We have several things to clarify in this section. In fact, quite a lot.

[1]The zero-temperature theory here is analogous to the theory of absolute zero temperature in physics, that characterizes a specific system state, typically at its lowest potential energy, where no reaction or spontaneous process can further convert its internal energy into thermal energy. The implication and direct anecdotally representative meaning of this, in this case, is rather unknown. All mentioned points to the stochastic effects of random fluctuation, since there is nothing more, and the system is entirely "in its lowest energy point". I really need to clarify on this point

Stochastic Process

A stochastic process is a family of random variables $\{X_\theta\}$ is a family of random variables $\{X_\theta\}$ indexed by a parameter θ wehere θ belongs to some index set Θ. in most cases, Θ is time, hence $\{X_\theta\}$ represents the time process, either discrete or continuous. But to be fair, in life you can only have the discrete version of such, at scale.

For processes in time, a less formal definition is that a stochastic process is simply a process that develops in time according to probabilistic rules. The direct antonym word to such is **stationary processes** where, the probabilistic rules do not change with time. Exhibition of such concept varies, however, from the picture. For gradient descent, adopting stochasticity means processing less total information in one go, and instead iteratively go through all sample points in the sample space, update accordingly, and repeat until it's done.

Evolution of Neural System

Given a loss function ϵ, define the *traning energy* $E(W) = \sum_{i=1}^{n} \epsilon(X_i \mid W)$. Training the neural network f with weight $W \in \mathbb{R}^d$ using full batch gradient descent amounts to allowing the weights W to evolve according to the gradient flow of $E(W)$:

$$\frac{d\phi}{dt} = -\frac{\partial L}{\partial \phi}. \tag{4.2}$$

or in this context

$$\frac{\partial W}{\partial t} = -\nabla_W E(W) \tag{4.3}$$

Langevin equation evolution

The paper *Statistical mechanics of Learning from Examples*, [Seung 1992] consider a generalization of the evolution of neural system, using the Langevin equation, that is

$$\frac{\partial W}{\partial t} = -\nabla_W E(W) - \nabla_W V(W) + \eta(t) \tag{4.4}$$

where $\eta(t)$ is the white noise , t is the *temperature* of the dynamical system, $V(W)$ represents "possible constraints" on the range of weights, and does not depends on $\mathcal{D}$.

The thermodynamic system interpretation is quite difficult to follows, since every done here is just overall, somewhat applicable, but it's not entirely statistical and thermodynamic. What does even temperature mean here?

Ehrenfest model

The Ehrenfest model of second law of dynamic is mentioned, albeit in the form of a framework. The model considers N particles in two containers. Particles independently change container at a rate λ. If $X(t) = i$ is defined to be the number of particles in one container at time t, then it is a birth-death process with transition rates

1. $q_{i,i-1} = i\lambda$ for $i = 1, 2, \ldots, N$
2. $q_{i,i+1} = (N - i)\lambda$ for $i = 0, 1, \ldots, N - 1$

and equilibrium distribution $\pi_i = 2^{-N}\binom{N}{i}$.

5 Philosophy of Artificial Intelligence

Reviewing of old people's arguing about themselves

Fujimiya Amane

5.1 The concept of intelligence

Despite a long history or research and debate, there is still no standard definition of intelligence, of even the ancient time. Of ancient philosophy, there's a few prominent proposal at hand of which belongs to the construct of intelligence as per **nous**, from the Greek word νους, of which equates to intelligence or intellect (in rare time), as a concept for the faculty of the human mind necessary for understanding what is true, or real. But a lot of progresses regarding the actionable concept of intelligence is not done until of the later, of which then the philosophy has matured enough for debates on the conception of intelligence to happen.

The direct antecedents of the word 'intelligence' lie in the Latin *intelligentia*, meaning the action or faculty of understanding, itself derived from *intellegere*, meaning to understand. Using only the etymological evolution of such word, naively, intelligence can be said to be, the overall actionable section of a living being to interpret knowledge, and acts on such knowledge. Fortunately, the world intelligence itself do not have much problems with the mystique effect, like in the other word of "mind" and consciousness. Rather, the opposite happens that it has been there for a long time being loaded with negative connotations, because it appears to us to describe something very mechanical, something that has little to do with feelings and value, while feelings and value are two things regard as what made a living being, a living being, at least in the narrow view of the human being. It is also, unfortunately, universally acknowledged in certain way that tests that seek to evaluate a human beings intelligence tell us nothing about that person's worth.

However, such consideration is left for the ethics and others faculty of philosophy. The fact that intelligence is still, quite not understood correctly is a problem in the stepping way of defining what is the main goal for at least, this discussion - on what can be regarded as artificial intelligence, and its philosophical argument of such. We would like to in certain light, define of which what can be regard as intelligence, and what can be done with it.

5.1.1 Previous attempts

It seems, as far as history can be traced back, the definition of intelligence is lightly regarded. Hence, the following section would mostly consist of modern take, containing statements on intelligence from psychology and artificial intelligent scientist, both of which apply certain definition and standard onto the notion of intelligence, to for the former one helping with discovering the mechanism of the mind and its functions, and one to define the foundational requirement sufficient to support the framework of what to be an artificial intelligent being.

Of the ancient time, Plato (400-300BC), regard a few ideas on human intelligence, at most. In the ancient definition, for human, intelligence is the faculty for reasoning, of the ability of the man to reason and making decisions. He differentiated between two ways of using reason: discursive reason, and intuitive reason. Discursive reason is exercised in the confrontation of theses. Thesis A is opposed to thesis B, and concludes with thesis C. Plato called it a dialectic exercise, of which is the type of reasoning used in mathematical proofs and logic exercises. it is a slow use of reason, which makes all the step explicit before reaching a conclusion. Instead, intuitive reason is a quick use of reason. It starts from the premises and reaches the conclusion without going through the whole deductive process. Intuition is the simplified form of deductive reasoning. Of this, Plato considered intuitive reason as the highest form of human intelligence. he spoke of intuition as a direct contemplation of the truth. These two faculties are two ways of using reasons. As reason is only human, Plato related these two modes to human beings alone.

Aristotle (Nueva Biblioteca Filosofica, 2017, Historia Analium), regard every living being have a "soul". Each type of soul is entrusted with particular functions of which makes of its own. Regarding the classification, Aristotle distinguished between three types of souls in the natural world. First, the vegetative soul, was typical of plants. Its functions were growth and nutrition. Second, the sensitive soul was typical of animals capable of locomotion. Along with the previous functions, it was capable of locomotion and sense perception. While some people may argue that plants can also “sense”, it has to be clear that Aristotle was referring to the ability of receiving information via complex sense organs. Third, the intellectual soul corresponded to humans. Along with the previous functions, it had the ability to reason. For Aristotle, the “soul” was not a spirit that survived the death of the body, as some understand it today. “Soul” only meant “anima”: the principle of animation or the principle of life. Any living being had a soul. Having a soul was synonymous with having life. So all organisms in biology have a “soul” in Aristotelian terms. In fact, this is the origin of the world “animal”: an organism that has an “anima” or soul. Aristotle, like Plato and all later Roman and medieval philosophers, only assigned the rational soul to human beings. For them, intelligence and reason were almost synonymous. But even then, intelligence and reason are not the same as sensory perception and imagination. These faculties fall within Aristotle’s sensitive soul. He assigned these abilities to other creatures besides human beings. This is consistent with recent scientific findings in animal memory, imagination and learning skills. But in this case, we are no longer speaking of intelligence in the strict sense, which implies

reason, but about different cognitive abilities that do not imply reason.

A rather light approach on the concept of intelligence is discussed by ancient Chinese philosophy, specifically, Confucianism and Taoism. For Confucians, intelligence is a matter of the ability to make the right moral judgement and to defend the validity of that judgement. It therefore comes as no surprise that Confucius said that "the intelligent mean is a person without perplexity" (The analects, IX.29, XIV.28), meaning that an intelligent person ought not to be perplexed in his or her judgements about right and wrong, and should be totally indifferent to the pursuit of self-interest. The right moral judgement always derives from benevolence.

Taoism (Lao Tzu, Chuang Tzu, and Graham - 1990) on the other hand, suggest a different concept, of which considered the human intelligence of interest. An intelligent person in the Taoist tradition is one who knows the *Tao* - 'the true greatness' - and can put this understanding into practice. With full knowledge of his/her internal as well as external conditions, s/he has ability to act intelligently, and is also able to conceal all strengths inside and to behave humbly. Being free from conventional standards of judgement, s/he is perceptive and responsive to changes in immediate circumstances. Acting according to these ideas, all his/her actions become as spontaneous as those of the natural world, and s/he accomplishes self-preservation by merging him/herself with the *Tao*.

5.1.2 Modern collective definition on intelligence

Linda Gottfriedson (1994) offers a definition of such intelligence problem:

> Intelligence is a very general mental capability that, among other things, involves the ability to reason, plan, solve problems, think abstractly, comprehend complex ideas, learn quickly and learn from experience. It is not merely book learning, a narrow academic skill, or test-taking smarts. Rather, it reflects a broader and deeper capability for comprehending our surroundings—"catching on," "making sense" of things, or "figuring out" what to do. [**Linda**; **1994**]

This definition is hardly can be considered clear, per context of it in the Wall Street Journal, "Mainstream Science on Intelligence". We do not know what the mental capability is, since it is conjugated to the philosophy of the mind, that itself is controversial. The actions involving such also contains several problems of which cannot be comprehended as if it is being discrete, or rather a byproducts of certain thought processes resides in the mind of a living subject, assume it has the capacity to do so in the first place. Some of the idea align well with the of psychology on which that intelligence comprises the mental abilities needed for adaptation to the environment (Sternberd & Detterman, 1986).

A. Anastasi also conjectured that:

> Intelligence is not a single unitary ability, but rather a composite of several functions. The term denotes that combination of abilities

> required for survival and advancement within a particular culture.
> [**Anastasi**; **1992**]

In conclusion, most of the definition is rather lackluster in term of its logical deduction, as well as the not-so-well-informed nature of those attempts in trying to create a starting point upon which the empirical analysis starts to work on, rather than defining it from there. A more exhaustive list can found in [LH07, Shane Legg and Marcus Hutter (2007)]

5.1.3 The problem, and the concept

Of the classical and ancient philosophical definition of intelligence, most of the proposal concentrated on defining the actionable part - reasoning pay few attentions to the underlying cause of intelligence. Ancient Chinese philosophy leans toward ethics more than else, of which concerns with a narrow sense, human intelligence as an inquiry of the truth, and the ability for a human being to be intelligent.

The modern definition fared no better. Coincidentally, as well as surprisingly, most of the modern philosophical arguments on what can be considered as intelligence are of psychologist and more or less, artificial intelligence researcher. All of them share several view on the ability for a living being to 'learn', not rigorously defined what is learning in such sense, sometimes using foggy resolution as the ability of an organism to solve new problems. AI scientist refers to it as "the ability of a system to act appropriately in an uncertain environment, where appropriate action is that which increases the probability of success, and success is the achievement of behavioral subgoals that support the system's ultimate goal" (J. S. Albus). However, all of which seems to only suggest a partial factor of which might give rise to intelligence. For example, taking J. S. Albus definition, how can you know what can be regarded as "appropriate action" and success for any given particular system? Because of the subjectivity of the world, and as we can conjecture from ourselves, the notion is success can vary wildly with certain consideration at hand. Furthermore, it also assumes the state of being programmed, that is, by providing the intelligence objectives and goals.

But some observation can be taken from this. From the view point of Plato and Aristotle, we can theorize partially, that intelligence can be taken into two wide category: natural intelligence, including human intelligence and non-human intelligence, biologically speaking, and artificial intelligence, of which is a manipulation of certain mechanism to manifest the concept of natural intelligence upon such system. The approach of such thinking includes the moderately controversial viewpoint of human being the pinnacle of intelligence, but as far as empirical evidences are suggested, it might be ruled out as agreed upon. Furthermore, under the analysis of Aristotle, and recent discovery in biology, we can say that intelligence is a component of a living being, separated from the conception of sensory, memory, and imagination. This allow us to somewhat formulate and formalize the intelligence of the mind, into a component of its own, of which become rather machine-like, but discrete and actionable.

However, until a good enough framework is conceptualized, the lack of good definition of intelligence in the field of philosophy, psychology and even AI, holds sufficient consideration when it comes to the respective research. Even though we can see a lot of observations being taken from those naive approach and attempts to do a make-shift definition, the fact that there's still the lack of any good argument on intelligence hinders the foundation of which intelligence and artificial intelligence can be practiced upon. The problem is, it is regard widely as useless or no one particularly want to test it. Hans Eysenc (1988) acknowledged that it was the most controversial concept in psychology. Many have argued that intelligence is an empty construction, and efforts to measure it thus useless, obviously disregarding its existence as a scientific concept analogous to gravitation or mass. For philosopher, intelligence has traditionally appeared a less interesting concept than consciousness. Philosopher of mind have occupied themselves with the mind-body problem, mental states and the "hard problem" of consciousness6 , but the development of AI has caused new philosophical interest in the concept of intelligence, though seldom appropriately decoupled from its closely related phenomenon consciousness. In the field of AI, there's no good definition either, as discussed and demonstrated above, but there's also the extremism like in psychology. Pei Wang (2008) specifically investigated this matter, and concluded that there existed two widespread opinions on the concept of intelligence. One view was that there exists a natural definition of intelligence, and that different understandings in various schools of artificial intelligence merely deal with different aspects, while they approach the same subject. The other common attitude was to view intelligence as a concept that escapes definition, and that it does not matter what researchers think about this as long as they produce results. Both positions end up concluding that attempts at reaching a shared definition is pointless.

5.2 The Artificial Intelligence

What is the definition of artificial intelligence then? The answer is, loosely speaking, we do not know. It in itself is a deep philosophical question, and attempts to systematically answer it fall within the foundations of artificial intelligence for analysis and debate.

Although there is no clear definition of artificial intelligence, there can be a lot of attempts in trying to capture the notion from different angle of interpretation. Margaret T. Boden wrote that an artificial intelligence could have general powers of "reasoning and perception - plus language, creativity and emotion". However, she does not forget to add that "that's easier said than done".

This same question of "What is AI?" has one of its answer pretty straightforward. In original Dartmouth conference, among others, Newell (1973), one of the grandfathers of modern-day AI (recall that he attended the 1956 conference) expressed that:

This same question of "What is AI?" has one of its answer pretty straightforward. In original Dartmouth conference, among others, Newell (1973), one of the grandfathers of modern-day AI (recall that he attended the 1956

conference) expressed that:

> AI is the field devoted to building artifacts that are intelligent, where 'intelligent' is operationalized through intelligence tests (such as the Wechsler Adult Intelligence Scale), and other tests of mental ability (including, e.g., tests of mechanical ability, creativity, and so on). [**Newell**; **1973**]

In the paper "The Philosophical Foundations of Artificial Intelligence", Selmer Bringsjord and Konstantine Arkoudas give a provisional answer to the introduction:

> AI is the field devoted to building artifacts capable of displaying, in controlled, well-understood environments, and over sustained periods of time, behaviours that we consider to be intelligent, or more generally, behaviour that we take to be at the heart of what it is to have a mind. [**Selmer Bringsjord and Konstantine Arkoudas**; **2007**]

The provisional definition here seems to itself to provide even more confusion and question, regarding the further analysis of each assumptions and components of which the definition relies on to gauge it's criteria to be specified as artificially created intelligence.

Russel and Norvig (2009) in their *AIMA* text, provide a set of possible answers to the question

	Human-based	Ideal Rationally
Reasoning-based	Systems that think like humans	Systems that think rationally
Behavior-Based	Systems that act like humans	Systems that act rationally

of which presents candidates for four possible goals for AI. This quarter of possibilities does reflect the relevant literature, by John Haugeland (1985), Luger and Stubblefield (1993) and Winston (1992). Regarding this definition, one can say that it's by far the most actionable classification out there, considering the applicational approach to such definition as per action-based criteria, even though some of the category might be ambiguous (what does it mean to think rationally?).

Of Russell and Norvig themselves, they falls into the category of the Ideal/Act. They lay out all of AI in terms of intelligent agents, which are systems that act in accordance with various ideal standards for rationality. Russell sees AI as the field devoted to building intelligent agents, which are functions taking as input tuples of percepts from the external environment, and producing behavior (actions) on the basis of these percepts.

The behaviour of the agent in the environment E, from a class $\mathbf{E}$ of environments, produces a sequence of states. A performance measure U evaluates

this sequence. Let $V(f, \mathbf{E}, U)$ denote the expected utility according to U of the agent function f while operating on $\mathbf{E}$. Then a perfectly rational agent would be required to be the function f_{opt}, such that:

$$f_{opt} = \text{argmax}_f V(f, \mathbf{E}, U)$$

According to the above equation, a perfectly rational agent can be taken to be the function f_{opt} which produces the maximum expected utility in the environment under consideration.

Russell himself created another entire brand, called **bounded optimality**, of which given a certain setting, optimize the agent in such system. To understand Russell's view, first we follow him in introducing a distinction: We say that agents have two components: a program, and a machine upon which the program runs. Denoting $Agent(P, M)$ as the agent implemented on machine M, $\mathcal{P}(M)$ the set of all programs running on M, then the bounded optimal program $P_{\text{opt},M}$ is defined as

$$P_{\text{opt},M} = \underset{P \in \mathcal{P}(M)}{\text{argmax}}\, V(Agent(P, M), \mathbf{E}, U)$$

In other words, for a given machine M, or architecture, the goal is to find the optimal program with such constraint, and with criteria that determines the optimality of being the "best" program. This might be computational complexities, risk factors, errors analysis results, vice versa.

5.2.1 The Descartes Argument

In part five of the book **Discourse on Method**, Rene Descartes discusses the condition for a robot to be an intelligent being.

> (I)f someone touched it (= the machine) in a particular place, it would ask what one wishes to say to it, or if it were touched somewhere else, it would cry out that it was being hurt, and so on. But it could not arrange words in different ways to reply to the meaning of everything that is said in its presence, as even the most unintelligent human beings can do.

Here, Descartes argues that in order for human-like robots to acquire intelligence, they have to gain a universal capability to accurately react to any unknown situation that may happen in the environment. However, what machines can do is no more than to respond to a single situation one-on-one via a specific organ, hence, they cannot be considered to have a universal capability that even unintelligent human beings can enjoy.

Continuing, Descartes argues that those machines do not act on their knowledge, but the disposition of organs.

> For whereas reason is a universal instrument that can be used in all kinds of situations, these organs need a specific disposition for every particular action. It follows that it is morally impossible for a machine to have enough different dispositions to make it act in every human situation in the same way as our reason makes us act.

The argument is quite clear. Human is universal of the environment. Whereas machine is no more than a combination of abilities that are applicable only to certain situation that the creator could imagine when they built the automated machine.

5.2.2 The flavour - "Strong" versus "Weak"

Boden's concept of artificial general intelligence resembles John Searl's "strong AI".

"Weak" AI can be defined as the form of AI that aims at a system able to pass not just the Turing Test (again, abbreviated as TT), but the Total Turing Test (Harnad 1991). In TTT, a machine must muster more than linguistic indistinguishability: it must pass for a human in all behaviors - throwing a baseball, eating, teaching a class, etc. According to Searl, "weak AI" is a computer that can behave as if it were thinking wisely.

"Strong AI" is then differently defined as a computer that actually thinks like humans. For a quote:

> According to strong AI, ... the appropriately programmed computer really is a mind, in the sense that computers given the right programs can be literally said to understand and have other cognitive states. (Searl, I)

The theme of strong AI was frequently discussed in the late 20th century - however, it became clear that in order for a computer to be a strong AI, it must resolve various difficult problems. The most difficult philosophical problem was **the frame problem**.

The **frame problem** is the problem that an AI cannot autonomously distinguish important factors from unimportant ones when it tries to cope with somethin in a certain situation. The problem arises, for example, when we let AI robots operate in the real world. This problem was proposed by John McCarthy and Patrick J. Hayes in 1969. This is considered a philosophical problem that cannot be merely reduced to a technical problems.

For now, the problem is unresolved, per Boden and specialists. Although there is no consensus about the definition of the frame problem, we could say that this is a problem centered around the question of how we can make an AI memorize the 'tacit knowledge' that almost all human adult can have in a given context.

Take an example of the normal cashier. Theoretically, and perhaps realistically speaking, a high schooler can be trained in a rather hasty fashion to do the job. While doing such job, there are many factors we take for granted. For example, during the process of using the computer to input the amount of cash required per transaction making, there might be a few terminal difference between different interface. The cashier knows that, and adapt to it. If she encounter an angry or hurry, the cashier can also acts accordingly, without the consideration for the performance. "If the customer is in a hurry, then maybe I should use this or that or skip this". Or even "If I pushes this button, then the trading screen appears". In fact, there are too many knowledge to be involved in such normal and particularly easy job. However, we do not have to input the knowledge that the stock market will affects the customer's

money that you are indeed doing transactions, or the fact that if the customer comes in with a bag of money, then in some cases, there will be missing bills, simply because it is not concerned of such.

Considering this, it becomes clear that there is an infinite amount of knowledge the robot must memorize. Who can make such a list of knowledge, and how is it possible to make the robot memorize them? The reason why this happens is that, when a robot encounters a new situation that it has never experienced, it cannot autonomously judge what kind of coping would be important to itself and what kind of coping would not, and therefore it cannot adequately solve the problem it faces. It is interesting that humans seem to be able to solve this kind of problem.

According to Dreyfus, for traditional AI to have the capacity to solve the frame problem, and become a true AI, it must become the Heideggerian AI, which incorporates Vorhandenheit (presence-at-hand) and Zuhandenheit (readiness-to-hand). Examining Rodney Brooks' robot architecture, he said that the robot respond only to fixed features of the environment, not to context or changing significance. But the robot 'continually referring to its sensors rather than to an internal world model'. Then, would the act of choosing, implementing a good enough sensor which have in itself the impendent sensory power in need, and most importantly have the best data coverage be enough to mitigate this? Since even for human, the sensory power is limited, since our limbs, nervous system and cognitive realization systems are finite, but perhaps we have a major sensor of which brought us the power of solving the frame problem locally, as well as responding to 'context or changing significance'? There might be, but then the question is quite straightforward: What (part) of human exhibits such trait?

5.3 The Chinese Room Argument

One of the most prominent critique for the philosophy of "strong AI" is the Chinese Room Argument (CRA, Searle 1980). Accordingly,

> CRA is based on a thought-experiment in which Searle himself stars. He is inside a room; outside the room are native Chinese speakers who don't know that Searle is inside it. Searle-in-the-box, like Searle-in-real-life, doesn't know any Chinese, but is fluent in English. The Chinese speakers send cards into the room through a slot; on these cards are written questions in Chinese. The box, courtesy of Searle's secret work therein, returns cards to the native Chinese speakers as output. Searle's output is produced by consulting a rulebook: this book is a lookup table that tells him what Chinese to produce based on what is sent in. To Searle, the Chinese is all just a bunch of – to use Searle's language – squiggle-squoggles.

We denotes O the observers (Chinese speaker), i, o being the input, output accordingly, and the rulebook P.

The argument of this experiment is simple - the Searle (in the box) is supposed to be everything a computer can be, and because he doesn't under-

stand Chinese, no computer could have such understanding. Searle is mindlessly moving around, and according to the argument, that's all computers do, fundamentally.

Nowadays, CRA, among AI practitioners, is generally rejected. Among these practitioners, there is John Pollock, who writes:

> Once (my intelligent system) OSCAR is fully functional, the argument from analogy will lead us inexorably to attribute thoughts and feelings to OSCAR with precisely the same credentials with which we attribute them to human beings. Philosophical arguments to the contrary will be passé. (Pollock 1995, p. 6)

Still, despite such argument, the relevance of CRA is actually more apparent than ever. The brute fact is that deeply semantic NLP is rarely even pursued, hence the proponents of CRA are certainly not the ones feeling some discomfort in the light of the current state of AI. Searle would rightly point to any of the success stories of AI, including the Watson system we have discussed, and still proclaim that understanding is nowhere to be found - and he would be well within his philosophical rights in saying this.

5.3.1 The Gödelian Argument

In 1961, J. R. Lucas presents the Gödelian argument against the existence of a "strong" AI. His proof is based on Gödel theorem, which is stated as followed:

> In any consistent system which is strong enough to produce simple arithmetic there are formulae which cannot be proved-in-the-system, but which we can see to be true. Essentially, we consider the formula which says, in effect, "This formula is unprovable-in-the-system". If this formula were provable-in-the-system, we should have a contradiction: for if it were provable-in-the-system, then it would not be unprovable-inthe-system, so that "This formula is unprovable-in-the-system" would be false: equally, if it were provable-in-the-system, then it would not be false, but would be true, since in any consistent system nothing false can be proved-in-the-system, but only truths. (Lucas, I)

This theorem holds for all formal systems which are consistent, adequate for simple arithmetics, and shows that those formal systems are incomplete, with some fact being true, but unprovable.

Lucas argues that the theorem must apply to cybernetical machines (or now computers) because

> It is of the essence of being a machine, that it should be a concrete instantiation of a formal system. It follows that given any machine which is consistent and capable of doing simple arithmetic, there is a formula which is incapable of producing as being true.

Further argued, he then comes to such conclusion that no machine can be a complete or adequate model of the mind, since "the mind are essentially different from machines".

Lucas's defenders, Roger Penrose, also state in his *Shawdow of the Mind* (1994). A human mathematician, if presented with a sound formal sytem F, could argues as followed:

> Though I don't know that I necessarily am F, I conclude that if I were, then the system F would have to be sound and, more to the point, F' would have to be sound, where F' is F supplemented by the further assertion "I am F"[1]. I perceive that it follows from the assumption that I am F that the Gödel argument $G(F')$ would have to be true and, furthermore, that it would not be a consequence of F'. But I have just perceived that "If I happen to be F, then $G(F')$ would have to be true", and perceptions of this nature would be precisely what F' is supposed to achieve. Since I am therefore capable of perceiving something beyond the powers of F', I deduce that I cannot be F after all. Moreover, this applies to any other system, inplace of F. (Penrose I, 1996, 3.2)

But in the end, it seems like the talk is more or less unfinished, as there are little evidence for both side on the subjective matters of the provability of reasoning.

5.4 Conclusion

In the end, to really know precisely what AI is, or requires you to do, you need to dive into it. Of certain time in the past such dive might be manageable. Nowadays, with extended knowledge and the constitutional framework being unclear, the dive might be more demanding than the beginning of the mid-century.

[1]The phrase "I am F" is merely a shorthand for "F encapsulates all the humanly accessible methods of mathematical proof"

6 The Possibility and Probability

A stupid take on probability

FUJIMIYA AMANE

6.1 Preliminaries

While researching and discovering probability, a rather strange question popped up in mind aside from the existence of probability: What is impossible, and possible in probability? The question seems vaguely trivial, yet not quite so. The notion of what is possible and what is impossible root deeply into the philosophical ground, as also the theory of probability that remains so. Furthermore, some of it contains a more direct reference to a more sophisticated theory, possibility theory, of which concerns that is possible, or not.

6.2 Possible or Impossible

> "The probability for this event is zero, which doesn't mean it's impossible." Another professor came in super happy one day, saying he had watched a football match the evening before, where the coin tossed at the beginning had landed on the edge, an event that is commonly assigned probability 0, but obviously not impossible.

What is possible and impossible? Loosely speaking, we say that something is possible, when it can indeed happen, with a certain "sizable" amount of probability. Or rather, if it is observed to occur. Hence, impossible can also be translated to not having observed, and with no probability to be assigned.

Of course, such definition is flaw in its interpretation. What is probability in this case? What are the observed values? What is used to observe? If a blind person cannot see a plane, would it mean it does not exist? Under which type of observant that the possibility remains valid? One such way at first to define this would be in probability.

In probability theory, we denote $P(E) = 1$ for a "will definitely occur" event, and $P(E) = 0$ for a "never going to happen" event.

But then if during statement, E is with $P(E) = 1$, does it mean that it will always be true? Of probability theory, we call E as being true with probability of 1, rather than saying it's true. Why is this? This implies that within certain consideration, if something is true, then it is true with

probability of one. But something with probability of one is not necessarily true. What does this mean?

6.3 Trueness

What is being true, then? A very simple question yet it brings up many more question. But to avoid such complication, maybe it is better to start with an attempt to understand the notion of being true.

Being true, implies that for a statement S to be tested, the consequence of such statement, E, will always occur, no matter what. This implies a statement S always has the following: a condition C, and a consequence E. This means that $P(E \mid C) = 1$, under our law of probability, if we can settle it right. We will ignore the "something with probability of one is not necessarily true", as we might figure it our ourselves.

But there's something that can perhaps break the uniqueness of this event. In reality, and as well as for any given event, under a statement S, we always have, or even as the definition of the consequence which allows for probability suggest, a possibility space Ω. What can this space of possibility be? Assuming the context of throwing a coin. The possibility space would be $\Omega = \{1, 0\}$. As we know it, there are no "quantum state" for such coin, so we can personally, in this case, remove the case of it being both 1 and 0 simultaneously. But this does, only concern with the condition C being throwing the coin only. In fact, if we expand the condition to being throwing a coin on a table with ridges, there is a non-zero chance for it being stuck in the middle, thus letting the result undetermined by the previous space Ω. Then, we said that $\Omega \subset \Omega'$, which is the new space of $\Omega' = \{0, 1, NaN\}$. Further expand this even more, we get even more sample space. Maybe the coin will be titled by a certain degree, hence we can have an event of $1(45°)$ if it leans to one specific side 45 degree more, and so on. This, include with the fact that potentially, without certainty, there are any combination of condition possible, of which the "scale" can either increase, or decreases, says something about the probability theory that elementally, we are using.

Don't say that if we cannot "see" it anytime soon, it won't happen. Just like with the coin - you did not see it does not mean it won't be ever happen. It just does not happen without certain condition C.

A might trivial realization, but worth considering at this point is that, there are no events without condition, that is, for any event E, $P(E) = P(E \mid C)$ given that the event E is described in the context, or condition C.

So what does it mean to be true? Would the statement that an atom is normally made of a combination of proton, neutron and electron being wrong (note that the statement is normally)? No, as far as we can see. So what does it being true means? In consideration of the last statement of atom, we can say that it is a convention of an object. If an object description, or an event E, as it being atom made of those subatomic particles, which is then being a subset of a bigger possibility pool (maybe we might see another atom

being made of another more particles), then it is always true, if $P(E) \neq 0$. Of course, even though we say that without observation does not mean it is not true, as there are uncertainties in both way, an atom cannot be made out of cars. This will eventually be brought up the idea of classifying which type of inference can a statement or an event be in our empirical-theoretical theory, but that is for another section. Continuing the line of logic, if we restrict the event E, with its component condition C, then $P(E \mid C) = 1$ will be said to be "true with a probability of 1 within condition C", if it always occurs, given such condition. Hence, implying $\neg P(E \mid C) = 0$, for a metric $[0, 1]$ being absoluteness. This also implies that, virtually, *there are no absolute truth universally*.

One interesting notion of which we are talking here about, is that conditional probability can really apply in this case, as it being recursively defined is a pattern in mathematics. Why? Because in certain way, C is an event, too. And the recursive nature of mathematics is very interesting such that, there might be even another article on it entirely.

6.4 Probability is fake

To understand the section title, which is rather a bold claim on probability is objectively, 'fake', we might want to take a look at the thought process which leads to the arising question of such theory.

1. What does probability really means, and what they actually measure?
2. What is the relation of it to the truth, and why there's even the term "true with probability of 1"?
3. How do we even know that probability guarantee the interpretation that it is interpreted all along?
4. Does probability exists directly as a "feature", or just an estimation, or quantification in the absent of determinism?

Those are four main questions of which concretely, create and formulate the statement "probability is fake", either objectively or subjectively. Still, to either prove it, or disprove it, much is done with answering every questions.

6.4.1 Probability in classical theory

The classic - standing probability theory focus on reasoning of events that are inherently random. The definition of probability in such framework is rather difficult, especially the reference frame which probability resides in, and the real-life existence in respect of probability.

Albeit difficult, there are certain "ways" to define probability. One of which is via repeatable experiments (random). By experiment means statistical experiments, of which an action might have multiple of different outcomes, all of which can be specified using numerical values. The outcomes can be somewhat "predicted", but when, where the particular outcomes will happen is unknown because of the random assumption of the whole experimental

subject. This definition is not self-referential, in any sense, which is one of its inherent advantage.

In this particular expression of probability, certain assumptions already arise: the process is random, of which the word "random" needs clarification and definition of itself. The experiments is repeatable, which helps averaging the observation. Regarding this, a proposal is to define it mathematically as degree of belief. Although making sense, the definition in this sense is also very unpredictive in the nature of it.

As of this attempt in defining probability, probability can be loosely defined as a quantification of any events, of which outcomes can be known, but when, where, and will it occurs is unknown. Probability hence provide a metric of certainty for such event.

6.4.2 On De Finetti's treatise on the theory of probability

The main goal of this article, is to explore the notion of which probability does not exists. But what does this sentence really mean?

By De Finetti, probability does not exists in an objective sense. Rather, probability exists only subjectively within the minds of individuals. He defined subjective probabilities in term of the rates at which individuals are willing to bet money on events, even thought in principle such betting rates could depend on state-dependent marginal utility for money as well as on belief.

> n the conception we follow and sustain here, only subjective probabilities exist - i.e., the degree of belief in the occurrence of an event attributed by a given person at a given instant and with a given set of information (pp. 3-4) (I)

The subjective theory of probability was proposed with the same behavioristic definition of probability, namely that it is a rate at which an individual is willing to bet on the occurrence of the event. Why betting? betting rates are the primitive measurements that reveal your probabilities or someone else's probabilities which are the only probabilities that really exists. This inverts the objectivistic theory of gambling, in which probabilities are taken to be intrinsic properties of events, and personal betting rates are later derived from them.

7 Introduction to Category Theory

Fujimiya Amane

Our examples frequently involve sequence, of which is a finite list of elements of a certain type, denoted $[a_0, \ldots, a_{n-1}]$. The set of sequences over A is denoted $Seq(A)$. Further operations are:

1. tips $= a \mapsto [a] : A \to Seq(A)$.
2. cons (Notationally $\_ : \_$) $= (a, [a_0, a_1, \ldots, a_{n-1}]) \mapsto [a, a_0, \ldots, a_{n-1}]$. Formally, $A \times Seq(A) \to Seq(A)$.
3. joins $= ([a_0, \ldots, a_{m-1}], [a_m, \ldots, a_{n-1}]) \mapsto [a_0, \ldots, a_{n-1}]$. Formally, $Seq(A) \times Seq(A) \to Seq(A)$.
4. $Seq(f)$ (function mapping) $= [a_0, \ldots, a_{n-1}] \mapsto [f(a_0), \ldots, f(a_{n-1})]$. Formally, $Seq(A) \to Seq(B)$ whenever $f : A \to B$ is a morphism.
5. $\bigoplus$ / (direct sum) $= [a_0, \ldots, a_{n-1}] \mapsto a_0 \oplus \cdots \oplus a_{n-1}$. Formally, $Seq(A) \to A$ whenever $\bigoplus : A \times A \to A$. and $\bigoplus$ is associative and has a neutral element.

7.1 Introduction to Functions

Category theory can be said as the generalization of mathematical structures, and their morphism, or functions in specific case, between elements of underlying mathematical objects, and between structures themselves.

Generally, it is realized, certainly, that the subject should have named itself arbitrary function analysis instead, because of the formal system and methodology used in the study of category theory.

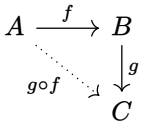
A function, or mapping, is a kind of relation between elements in one 'object' to 'another object' of similar or different type. Often, in elementary mathematics, it would be different type of sets. For example, in the context of analysis, it would be $\mathbb{R} \longrightarrow \mathbb{R}$, or with $\mathbb{C}$. Given two objects, A and B, the mapping from $A \longrightarrow B$ is denoted as $f : A \longrightarrow B$. The operation is unary (with single operand, and single output), thus we call function a unary operation on objects. To specify the input and output, we have the convention:

$$A = src(f) = dom(f) \quad B = tgt(f) = range(f)$$

A successive sequence of function, for example, between three objects (A, B, C) expressed as

$$A \underset{f}{\longrightarrow} B \underset{f}{\longrightarrow} C$$

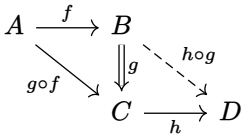
can have a composite function, $g \circ f$ (here we have g before f, since in the commutative diagram, $range(f) = dom(g)$, hence g 'wraps' above f):



This function, $g \circ f : A \longrightarrow B$ is written as

$$g \circ f(a) = g(f(a))$$

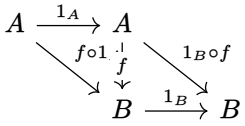
Further operation chain including the quadruplet (A, B, C, D), with $h : C \longrightarrow D$ with such



Here, the formation of $h \circ g$ and $g \circ f$ let us compare $(h \circ g) \circ f$ and $h \circ (g \circ f)$ as indicated in the diagram. The result is quite surprising, $(h \circ g) \circ f = h \circ (g \circ f)$. This is because accordingly,

$$((h \circ g) \circ f)(a) = h(g(f(a))) = (h \circ (g \circ f))(a)$$

For any two functions to be "equal", for every argument, their value must be the same. Under the observation of commutative diagram as above, it is analogous to having the same endpoint. The observation above is also applied for identity function, $1_A : A \longrightarrow A$ given by $1_A(a) = a$. Theoretically, those function acts as units for the composition operation: $f \circ 1_A = f = 1_A \circ f$. The commutative diagram for identity of the tuple (A, B) is



7.2 Category and Precategory

Definition 7.2.1. *A category is a system of:*

Objects *denoted by* $A, B, C, \ldots$.

Morphism *denoted by* $f, g, h, \ldots$.

Relation on Morphism/Objects *called typing of the morphisms. We say that for* $f : A \to B$*, then* $A \to B$ *is the type of* f*, and* f *is a morphism from* A *to* B*. Then* $\mathrm{src}(f) = A$ *and* $\mathrm{tgt}(f) = B$.

Composition *A binary partial operation denoted as* $f \circ g = gf = f; g$.

Identity *For each object* A *a distinguished morphism, called* ***identity*** *on* A. *By default, we denote it as* id_A, *denotes the identity on object* A.

This category definition defines the categorical language in which properties of the category can be stated, of which can be built from normal logical connectives and quantification and equality. Of course, category does have axioms, as for all formalism of mathematical system, needs axioms for its description of the mathematical structure.

A precategory is a category of which the (I) requirement is dropped. By simple trick (as is stated by the original author), we can construct a category $\mathcal{B}$ from a pre-category $\mathcal{A}$. Suppose that $\mathcal{B}$ is defined by:

- An object in $\mathcal{B}$ is an object in $\mathcal{A}$.
- A morphism in $\mathcal{B}$ is a triple (A, f, B) with $f : A \longrightarrow_{\mathcal{A}} B$.
- $f : A \longrightarrow_{\mathcal{B}} B \equiv A = A', B = B'$ where $f = (A', f', B')$. $(*)$
- $f \circ_{\mathcal{B}} g = (A, f' \circ_{\mathcal{A}} g', C)$ where $(A, f', B) = f$ and $(B, g', C) = g$.
- $id_{\mathcal{B},A} = (A, id_{\mathcal{A},A}, A)$.

Then, we can deduce or prove that $\mathcal{B}$ is a category. Our best concern of this proof will be the starred $(*)$ statement of which validate the notion of such. Using the notation, we see that a morphism $f = (A, f', B)$ of which $f' : A' \to_{\mathcal{A}} B'$. Then, of which we can see that for $f_{\mathcal{B}} = f'_{\mathcal{A}}$, then the morphism $f : A \to_{\mathcal{B}} B$ of which is defined on $\mathcal{B}$, will be also given of such definition, that is: $f : A \to_{\mathcal{B}} B = (A, f', B)$, but $f_{\mathcal{B}} = f'_{\mathcal{A}}$, then both object of which also, again, $A, B \in \mathcal{B} \wedge A, B \in \mathcal{A} = T$ (both of them is in $\mathcal{A}$ and $\mathcal{B}$), then it must be said that $\mathcal{B}$ is a category since this satisfy the axiom (I).

The axioms on the morphisms and composition are very weak, so that many mathematical structures can be rendered as a category. By imposing extra axioms, still in the categorical language, the categories might have more of the properties of interest. A Cartesian closed category (CCC) is a category in which the extra properties make the morphisms behave like real functions, that is, there exists the notion of currying and of applying a curried morphism. In brief, currying is a process of transforming an operation on two variables into an operation on one variable that returns a function taking the second variable as an argument. For example, for a function $f : X \times Y \to Z$, currying will create a function $\hat{f} : X \to Z^Y$ of which $\hat{f}(x) = (y \mapsto f(x, y))$. One can also think about this as being $X \times Y \to Z \equiv X \to [Y \to Z]$. There is a close relationship between this type of category and typed λ-calculi.

Further into notation, given two objects A and B in an arbitrary category $\mathcal{C}$, then the collection of morphisms is denoted $\mathcal{C}[A, B]$ or $\mathcal{C}(A, B)$. This is called a hom-set, and another frequent notation is $\mathrm{Hom}_{\mathcal{C}}[A, B]$.

7.3 Functors

Further increment in the scope of our problems, we are interested in morphisms between categories. A homomorphism of categories is called a functor.

Definition 7.3.1. *A functor*

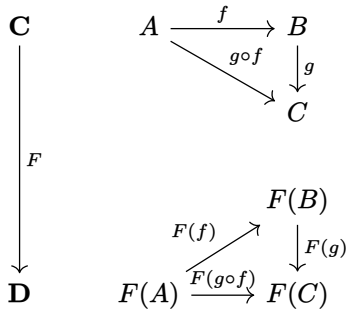
$$F : \mathbf{C} \longrightarrow \mathbf{D}$$

between category $\mathbf{C}$ *and* $\mathbf{D}$ *is a mapping of objects to objects and morphisms to morphisms such that*

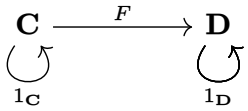
1. $F(f : A \longrightarrow B) = F(f) : F(A) \longrightarrow F(B)$
2. $F(1_A) = 1_{F(A)}$
3. $F(g \circ f) = F(g) \circ F(f)$

The formula $F : A \longrightarrow B$ means that F is a functor

Retrospectively, F preserves domains and codomains, identity morphism, and binary composition. The functor $F : \mathbf{C} \longrightarrow \mathbf{D}$ thus gives a sort of "picture" of $\mathbf{C}$ and $\mathbf{D}$. Commutative diagram is shown as below:



One can clearly see that each category have the identity morphism $1_{\mathbf{C}} : \mathbf{C} \longrightarrow \mathbf{C}$. Regarding this, considering only two categories, we have



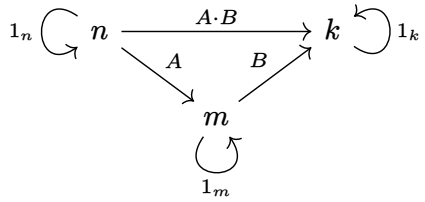
Hence, the components for a category between categories is sufficed. We call this **Cat** category to represent the category of all categories and functors.

7.4 Examples of Category

There are many examples of categories, which we will discuss as a practice of defining mathematical system in categorical language.

7.4.1 Unital Ring

For a unital ring R, the category Mat_R is category whose objects are positive integers and in which the set of morphisms from $n \to m$ is the set of $m \times n$ matrices with value in R. Composition is by matrix multiplication



7.4.2 Group and Monoid

A group or generally, a monoid defines the category BG with a single object. The group elements are its morphisms, each group elements represent a distinct endomorphism (as we will discuss later) with composition given by multiplication. The identity element $e \in G$ acts the identity morphism for the unique object of the category.

Generally, monoid is a set M equipped with a binary operation $M \times M \longrightarrow M$ and a distinguished element $u \in M$ such that:

$$\forall (x, y, z) \in M \quad \begin{cases} u \cdot x = x = x \cdot u \\ x \cdot (y \cdot z) = (x \cdot y) \cdot z \end{cases}$$

Equivalently, a monoid is a category with just one object. The arrow of the category are the elements of the monoid. In particular, the identity arrow is the unit element u.

7.5 Investigation on Morphism

In modern mathematics, we have different way of formalising the mathematical foundation. While listing the object is interesting, all the axiom focus on sorting objects fitting to different categories, based on their behaviour within such system. For example, for the monoid, it is a tuple $(M, e, \times)$ that satisfies the above notion.

Similar, our main investigation would be mostly morphism and functors, since from investigating their behaviour, can we achieve understanding the underlying abstraction of object.

7.5.1 Total Morphism

A total morphism is a general morphism, or a bare morphism. It is similar to the aforemention morphism, which takes the form $M_{\mathcal{B}} : (O_1, \ldots, O_n) \longrightarrow (O'_1, \ldots O'_n$. Usually, it takes the form of a mapping size $n \to m$. In the context of linear algebra, such morphism can be said as a linear mapping of $M_{\mathbf{Lin}} : (\mathbb{F}_1, \ldots, \mathbb{F}_n) \longrightarrow (\mathbb{F}'_1, \ldots, \mathbb{F}'_m)$

7.5.2 Homomorphism

In a restrictive sense, a homomorphism is a function between two algebras that preserves its algebraic structure. It makes sense for category of algebraic type to be called a structure set, like with the case of monoid and groups. Then, a

homomorphism is a synonym for morphism (remember that total morphism takes an arbitrary domain space and returns also an arbitrary domain space) when structure sets are generalized to arbitrary object.

Traditionally, a homomorphism between two magmas (structured set equipped with binary operations on itself) A and B is a function

$$\phi : A \longrightarrow B$$

That satisfy the underlying binary operation. Specifically, for all $a_1, a_2 \in A$, we have

$$\phi(a_1 \times a_2) = \phi(a_1) \times \phi(a_2)$$

This definition gives us the correct notion of **magma, semigroup** and **group** homomorphism, but not for monoid homomorphism and ring homomorphism.

Definition 7.5.1 (monoid/ring homomorphism)**.** *A homomorphism between two monoids A and B is a semigroup homomorphism, $\theta : A \longrightarrow B$ of the underlying semigroups that preserves identity element, that is:*

$$\theta(1_A) = 1_B$$

A ring homomorphism is a function between rings that is a homomorphism for both the additive group and the multiplicative monoid.

More generally, a homomorphism between sets equipped with any algebraic structure is a map preserving this structure. This can be made precise using Lawvere theories, but generally, structure-preserving can be made clear using intuitional explanation.

There are many structures in category theory. Suppose with a triplet (x, y, z) where, $x, y, z \in X$, of which on X addition is defined. This means that some triple (x, y, z) of the elements of X are special so far as $x + y = z$ is true. Then we can say that the triplet is a structure that conforms onto elements of X, and we denote it as $(x, y, z) \in plus$. Continuing with the assumption for the second set Y of which satisfy certain requirements as X, essentially the same construct, and with the second structure named $plus'$.

From X to Y, we define a morphism $\phi : X \longrightarrow Y$. We said that ϕ is a structure preserving morphism, or homomorphism, if

$$(x_1, x_2, x_3) \in plus \underset{\phi}{\longrightarrow} (\phi(x_1), \phi(x_2), \phi(x_3)) \in plus'$$

7.5.3 Isomorphism

Isomorphism is a somewhat expansion from homomomorphism, as far as the definition is concerned of.

By the standard group theory definition, we can say about isomorphism as followed.

Definition 7.5.2 (group isomorphism)**.** *Let G and H be groups. An isomorphism $f : G \longrightarrow H$ is a bijection $f : G \longrightarrow H$ such that for all $g_1, g_2 \in G$,*

$$f(g_1 g_2) = f(g_1) f(g_2)$$

From the definition of group isomorphism, isomorphism can be thought as the bijective version of homomorphism, that is, as the previous intuitive explanation of homomorphism structure preservation property, then

$$(x_1, x_2, x_3) \in plus \underset{\phi}{\iff} (\phi(x_1), \phi(x_2), \phi(x_3)) \in plus'$$

for an isomorphism $\phi : A \longrightarrow B$.

For a category-theoretic definition, we have the following:

Definition 7.5.3 (categorical isomorphism)**.** *In any category* $\mathbf{C}$*, an arrow* $f : A \to B$ *is called an isomorphism if there is an arrow* $g : B \to A$ *in* $\mathbf{C}$ *such that:*

$$g \circ f = 1_A \quad \textit{and} \quad f \circ g = 1_B$$

effectively, this means that the arrow f has its inverse. Since inverse are unique, we write $g = f^{-1}$.

8 Introduction to Mathematics

Really, a not so painful introductory.

Fujimiya Amane, H. Miharu

Mathematics is strange because sometimes it's very intuitive. The logical argument of naive consideration is enough, perhaps most of the time to go by in life. However, if we want to dive deeper into mathematics, formalism and hence, a foundation must be settled, as preliminaries setting to work on, and to understand the ground of what is considered to be mathematical method.

This section will be myself, as per someone who wrote this book (kind of), to threaten myself, and hence establish myself a somewhat formal groundwork - of at least the starting point on then the elementary foundation is settled. Regarding this though, is that the foundation setting of mathematics itself is a topic of debate, and also a pretty much hassle to even define such. So for now, this part will cover eagerly proof, the elementary logic system, and several elementary treatments of numbers as the first analytical work.

8.1 Proofs and Rigours

By definition, let's define an informal view on proof and logics.

Definition 8.1.1 (Proof, I)**.** *A **proof** is a sequence of true statements, without logical gaps, that is a logical argument establishing some conclusion.*

This definition captures the notion of proofs, but however, lacks the actionable components.

Definition 8.1.2 (Proof, II)**.** *A proof is a sequence of logical statements, one implying another, which gives an explanation of why a given statement is true. Mathematical proof is absolute.*

To prove things, we need to start from assumptions, or axioms. By non-rigorous manner, axioms can be taken as the formal ground for which the work can be done on the above layer. The axiom can be reasonable or not, so is the truth assumption for that axiom. It is like saying "If the axiom is true, then this is what I told you". Others axiom might still yield different results, as per mathematicians like. The key thing is to define, and at least draw out a general assumption that your axiom takes form, that is on first hand, actionable. Then the rigorous system will be built upon that.

Per attention, we also care about the little "statement" in our definition, too.

Definition 8.1.3 (Statements, I)**.** *A statement is a sentence that can have a true value. That is, it is a logical unit with truth value being assigned to its truth.*

Therefore, we can also say statements declare or assert truths of certain subject.

Those statements can be either false or true. Often, we will want to concern about statements with truth value, not as the statement of there are infinitely many primes of the form n^2+1, because it is not probably true, but none has a clue. So it might be call (?) a conjecture instead. There are also times when statements can have no justifiable truth value, of which then we will need to put on such convention on the statement specific properties.

Definition 8.1.4 (Statements, II)**.** *A statement is a declarative sentence, otherwise called a declaration that is discrete in its properties of true or false, but not both. Formally, this is written for a statement $P(n)$ as:*

$$P(n) = \{P : P = T \text{ or } P = F\} \tag{8.1}$$

In some cases, it is not immediately clear if a statement is true or false. So even with the indication slightly contradictory to the above notion, for a sentence to be a statement, its capacity of assigning truth is a requirement, but we do not need to know its exact propensity, or its actual truth.
[1]

A sentence that satisfies or not the condition to be a statement, but contains in its description of the declarative part of variables or elements of prescribed sets, which is called the domain of such variable, then is called **open sentence**. If the open sentence $P(x)$ with $x \in S$, then we say $P(x)$ is an **open sentence on the domain** S. Then, $P(x)$ would be a statement for each $x \in S$.

Thereby, from observations of the elementary system, the foundational blocks for proofs are from those sentences, and its characteristic sentence of open sentences. In interest and necessity of creation of statement, (open) sentences must be converted to statements. Only then the higher properties of statement, which would be discussed later, shall be applied onto the process of making proofs. To do this, we might want to take a look at the logical system of which enables the elementary form of logical argument, leading to the proof of requirements.

[1] In some texts (Advanced Calculus, Sternberg, 1990), open sentence is called as a **statement frame**, of which then statement is obtained from such frame. This is accordance to the coverage of each open sentence (frame), since it applies the rules onto different elements of the domain.

8.2 The Propositional Logics

Working with logic, traditionally, is to deal with absolute truth. There's the notion between true, and false. So, rationality and statements in logics is figured in terms of discrete truth. We denote true as T, false as F for convenience.

What constitutes to our consideration of what is true or false? Specifically, what do we mean by truth? We use the above notion of a statement for such. Statements is the foremost piece of 'equipment' we have in logic, aside from (open) sentences. Informally, they carry truth value, of which meaning their

description is realized, and not vague in terms of such realization. Simply speak, if a statement is false, then it is indeed, false, under any consideration of inspection, given the system of formalism that it is based off.

8.2.1 Logical connectives

Logical operations use statements and act on them, producing new logical statements. Their truth value (either true or false) is presented by using truth table, essentially a table-based organization for the combination of truth value. So in essence, they are the tool for statements to acts of each other, based of the properties of relations, and what is connected to each others.

In a perspective, we can say that logical system is constructed using logical operation, as a mean to connect logical statements, or transforming them. Because of this, they are sometimes referred to as **logical connectives**. Formally speaking, for a statement $P(x)$, logical operation is of the form $\text{LO} : S^n \to T^m$, where n and m indicate the domain involvement of the operation, and the domain resultant of such operation. An **unary** operation takes the form of $n, m = 1$, since they take one and produce (transforms) one. Binary operation would be $n = 2, m = 1$ since they took two argument, and produce one. All of this would be reflected in the truth table, as per configuration of the mathematicians.

Let P and Q be statements. Then $P \wedge Q$ is the "and" operator, and $P \vee Q$ is the "or" operator. Formally, they are called conjunction and disjunction, respectively. Their role is to combine specific statements (2 statements) under the landscape of some rules: Either of them is correct yields correct statements, or a bit more lax, only one of them need to be such, at bare minimum. So for the conjunction operation, combining two statements, A, B, will have the following truth table

A	B	$A \wedge B$
T	T	T
T	F	F
F	T	F
F	F	F

Here we can see how truth table is presented: Statements with their truth in specific columns, and the operation's results of logical realization on the other side. Sometimes, because for every operation, even nested or so, complex or such, must then come out to be a single truth value at the end of its chain of logical actions, it is convenient to list out the possible truth value as a table of such.

Of course, increasing the amount of statements increases the possible permutation that such complex operations can bring, but that is of the concern later on, and still will result in one single truth value.

The truth table of the disjunction operation is presented as

A	B	$A \vee B$
T	T	T
T	F	T
F	T	T
F	F	F

Similarly, $P \implies Q$ is implication operation. The only case for which implication is false is when P is true, but Q is false. Why is this the case? Suppose that P is false and Q is true. In this scenario, for example, the student in a test did not get an A on his exam, but when he receives his final grades he learned that his final grade was an A. How could this happen? The only argument about this is that the whole ordeal is a mistake, and need to check back, why? Because the instructor did not lie, so do the grade. So there must be a mistake somewhere, hence, it is indeed false.

$P \iff Q$ is the second binary operation aside from the standard operations above. It stands for "if and only if", and is called a **biconditional** logical operator. It is much stricter than the implication operation: similar to and, it requires for the truth to be evaluated as T, if only the two-way implication is true. Specifically,

$$(P \implies Q) \wedge (Q \implies P) = T$$

This would also be the time when you are introduced to what exactly are we doing, that we are using notations and formal language of such form we are now. Such form is called as **atomic**, of which statements and operations can be constructed from other components, such as the **and** operator and the implication operator, as you saw. In case of biconditional operator, formally, we often state it as:

$$P \text{ is equivalent to } Q$$

or as

$$P \text{ is necessary and sufficient for } Q$$

Their truth table is as followed:

P	Q	$P \implies Q$	$P \iff Q$
T	T	T	T
T	F	F	F
F	T	T	F
F	F	T	T

So far, we have only seen binary operation, taking two arguments (statements) and combine them. An example of logical operation, specifically unary, is the operation of negation $\neg$. For a statement A, negation has its truth table as such:

A	$\neg A$
T	F
F	T

Negation specifically means "not", of which negates the original statement.

Overall, the truth table is as followed for two statement A and B.

A	B	$A \vee B$	$\neg A$	$\neg B$	$\neg(A \vee B)$	$\neg A \wedge \neg B$
T	T	T	F	F	F	F
T	F	T	F	T	F	F
F	T	T	T	F	F	F
F	F	F	T	T	T	T

8.2.2 Tautology and Contradiction

We have seen single use of operation, up to this point. However, we can use the logical connectives above to form more intricate statements, which can be nested, sequential of many statements and connectives. More generally, a **compound statement** is a statement composed of one or more given statements (also called **component statements** in this context), and at least one logical connectives.

The compound statement below, which is combined of the logical *or*, and logical *or*, is expressed as

A	$\neg A$	$A \vee \neg A$
T	F	T
F	T	T

This is the statement $A \vee \neg A$ for any given statement A. The resultant is always true as we have observed. When such case happens, we call the statement a **tautology**. The concept of tautology is pretty important, even in this elementary stage of treatment on logic, specifically because it is static that it can be utilized in several proofs as a "constant" form. Specifically, tautology means that the compound statement S being classified as such, would always be true for all possible combination of truth values of the component statements that comprise S. Hence, $A \vee (\neg A)$ is a tautology.

An example shall be given. For statement P and Q, the compound statement $(\neg Q) \vee (P \implies Q)$ is a tautology, based of its truth value. (You can check it yourself - or just me). Still this statement, if we let P being '3 is odd', and '57 is prime', we just get

> 57 is not a prime, or 57 is prime if 4 is odd

This is true regardless of which statement P or Q is considered.

On the other hand, a compound statement S is called a contradiction if it is false for all possible combinations of truth values of the component statements that are used to form S. The statement $P \wedge (\neg P)$ is a contradiction, as being shown of its truth table:

P	$\neg P$	$P \wedge (\neg P)$
T	F	**F**
F	T	**F**

8.2.3 Logical equivalence

Certain logical proposition are equivalent, which we denote $\equiv$. Two logical statements are called logically equivalent if the truth tables (all possible assignments of truth value for the logical variables) are the same. Formally, this is defined as logical equivalence.

Specifically, let R and S be two compound statements involving the same component statements. Then R and S are called **logically equivalent** if R and S have the same truth values for all combinations of truth values of their component statements.

We have the following definition.

Definition 8.2.1 (Logical equivalence)**.** *Let A and B being any (compound) statements or open sentences with domain specified. Then A and B is logically equivalent if for any configuration of truth in A, B matches its truth value to A, and reverse.*

Logical equivalence is especially useful, in case we want to compare certain complex compound statements and its truth value as consequences. This includes also, confirming for example, if we somewhat want to 'shorten' down certain compound statement to be a logical operation itself, for example, bidirectional, we can pretty much confirming them both. This works for tautology, per purpose of what we want to do. The key thing, simple down, is that it offers us a way to look at multiple interpretation of a single logical system, of which then the equivalence notion is defined.

There's several things to note from this. Firstly, is that for two logical statements to be compared, then every compartment of their component logical truth must be told. Second, there will be time when one statement uses certain component, that the other compound statements have none. In such case, it is hard to tell aside from testing if the behaviours result from such statement can be equivalent or not, since there is 'external' factor to the truth evaluation. However, it is a minor concern, as after testing of truth value then comparing it is still a better choice.

Setting that aside, there are some theorems we need to know of this elementary section.

Theorem 8.2.1 (Equivalencing)**.** *Let P and Q be two statements. Then*

$$P \implies Q \equiv (\neg P) \vee Q$$

Theorem 8.2.2 (Equivalent Law)**.** *Let P, Q and R be statements. Then,*

1. *$P \vee Q \equiv Q \vee P$, and $P \wedge Q \equiv Q \wedge P$ (Commutativity)*
2. *Associativity:*
$$P \vee (Q \vee R) \equiv (P \vee Q) \vee R$$
similarly,
$$P \wedge (Q \wedge R) \equiv (P \wedge Q) \wedge R$$
3. *Distributivity:*
$$P \vee (Q \wedge R) \equiv (P \vee Q) \wedge (P \vee R)$$
similarly,
$$P \wedge (Q \vee R) \equiv (P \wedge Q) \vee (P \wedge R)$$

Finally, we have De Morgan's Laws:

Theorem 8.2.3 (De Morgan)**.** *Let P and Q be two statements. Then*

$$\neg(P \vee Q) \equiv (\neg P) \wedge (\neg Q) \tag{8.2}$$
$$\neg(P \wedge Q) \equiv (\neg P) \vee (\neg Q) \tag{8.3}$$

All the above theorem can be deduced and verify by means of truth tables.

8.2.4 Quantifiers

In logic, there are certain lexical components we call as quantifiers, which is to 'quantify' the scope of application, for logical arguments. $\forall x, P(x)$ means $P(x)$ is true for all x, and $\exists x, P(x)$ there exists such x that $P(x)$ is true. These two notions can be understood intuitively: One guarantee *universal correctness*, while the other guarantee *existential correctness* - there will always exist such fact, at the minimal counting unit we can find. The quantifiers are usually bounded, as per their definition in the logical unit. Formally, $\forall$ is the universal quantifier, and $\exists$ is called the existential quantifier. These quantifiers are used to assert certain open sentence, of which then produce statements. We will discuss this in the next section.

Negations of quantifiers are as the following table:

$$\neg(\forall x)P(x) \equiv (\exists x)(\neg P(x)) \tag{8.4}$$
$$\neg(\exists x)P(x) \equiv (\forall x)(\neg P(x)) \tag{8.5}$$

8.3 From frames to statements

We have mentioned that if $P(x)$ is an open sentence over a domain S, then $P(x)$ is a statement of each $x \in S$. This implies the need for the domain of S to be specified for the open sentence $P(x)$. Or rather, we can break down a statement as it is formed by:

$$\text{Statements} = P(x) + \text{Dom}(x) \tag{8.6}$$

of which $\text{Dom(x)} = S$, is the domain of the variable x. There are a few ways to convert open sentences into statements, of which we will make use of the above logical system for such task.

8.3.1 Quantified(-cation of) statements

One of such way to convert an open sentence into a statement, is by the mean of quantification. If $P(x)$ is a statement frame over some domain S, obtaining statements from this frame can be done by attaching quantifiers onto it. This asserts the frame $P(x)$, for specific x compartment. Such statement is then called as **quantified statement**. Formally, for a statement $P(x)$, then the quantified form of such statement is taken in the form of $(\forall x)P(x)$ or $(\exists x)P(x)$. One frequently presents sentences containing (multiple) variables as being always true without explicitly writing the universal quantifier, however. So instead of

$$(\forall x)(\forall y)(\forall z)[x + (y + z) = (x + y) + z]$$

we can just write

$$x + (y + z) = (x + y) + z$$

for the ease of writing the quantifiers. Note that the shortened form of this quantified statement is of the form $(\forall x)(\forall y)(\forall z)P(x, y, z)$.

For existential quantifier, an existentially quantified statement only guarantee 'sometimes' true quantification to such frame, hence it must not be absent from the formal writing.

The statement $(\forall x)(x < 4)$ still contains the variable 'x', but it is no longer allowed to take on any values, and is called a bound variable. Roughly speaking, quantified statements contains quantified variables, which are bound, while unquantified variables are free. The notation $P(x)$ is hence very specific - it is only used when the variable x is free in the sentence being discussed.

Order of quantifiers

One of the simple fact for quantifiers and bounds has their order, is that they are not commutative, if the quantifiers' types are different. So $(\forall x)(\exists y)P(x,y)$ is inherently different from $(\forall y)(\exists x)P(x,y)$. Why would this happen?

First, we revise what the *quantifiers* actually means. Quantifier, is analogous to the domain, for specific statement frame (we use this word for better representation than open sentence). It enforces the operational scope of finding the statement - for example, $\forall x \exists y$ means exactly that, there exists an y in the scope of the domain of x, such that it becomes the minimal existential guarantee. Thereby, only one exists would give the statement generated off the statement frame, to be true. However, if we reverse it, then the scope changes: $\exists x \forall y$ means any y will have at least one x of such property as in $P(x,y)$. The scope has changed - now it must be correct for every element possible of y, and each of them need at least 1 example. If we can interpret it differently, it's the extremal of minimal bound, and the minimal of the extremal bound, so to speak.

On the other hand, if they are of the same types, then it is fine. Among a group of quantifiers of the same type, the order does not affect the meaning. Thus $(\forall x)(\forall y)$ and $(\forall y)(\forall x)$ has the same meaning. This also mean we can sometime abbreviate it as $(\forall x, y)$, if we wish to reduce the amount of notation needed.

Lemma 8.3.1. *If quantifiers of both types are used, the order of which they are written affects the meaning of the statement, and hence they are not commutative. Quantifiers of the same types do not have such effects.*

Theorem 8.3.2. *Denoting truth as numbers, $\{0,1\}$. We have the logical statement as a function $L : \prod_{\times, i \in I}(B_i) \to \{0,1\}$, of which $\{B_i\}_{i=1}$ is the set of all 'bounds' that each variable constitute in such. Then, we can interpret the two existential and universal as bound such that:*

$$B_\forall = \{x_i\}_{i=1}^{m>1} \tag{8.7}$$

$$B_\exists = \{y_1\} \tag{8.8}$$

Thus, existential order can be thought as, for binary case,

$$L_B : B_\forall \times B_\exists \to \{0,1\} \tag{8.9}$$

to relies on its truth value for the domain region of $B_\forall$ as a stronger bound to guarantee that the statement is then true.

Similarly, the statement

$$L_B : B_\exists \times B_\forall \to \{0, 1\} \tag{8.10}$$

relies on the domain $B_\exists$ to dominate the strong bound of true evaluation.

The above theorem only offer a "naive" but systematic way of justifying the relationship between two quantifiers, under a elementary setting of propositional logic.

Of all, when using quantifiers, we need to keep track of our order, regarding all the quantifiers being used. If not, mathematicians' career would be in shamble since they used them all wrong, if ever, until they realize it right now. Because it is a very convenient tool, that is why we must be careful on how to use it. Additionally, there are several times we should reduce the amount we use, and instead work on it with an actionable bound, instead of an absolute minimal/maximal bound of such.

On a side note, the idea of strong bound truth validation is a very interesting idea, and might meet its analogue somewhere else in the mathematical system.

8.3.2 Characterization

Suppose that some concept (or object) is expressed in a statement frame $P(x)$ over a domain S, and $Q(x)$ is another frame over the domain S concerning this concept. We say this concept is characterized by $Q(x)$ if $\forall x \in S, P(x) \iff Q(x)$, i.e. they maintain a bidirectional relationship, is a true statement. The statement above is then called a characterization of this concept.

Let's take an example.

Example 8.3.1. Suppose that

$$P(x) : x = -3 \text{ and } Q(x) : |x| = 3$$

where $x \in \mathbb{R}$. Then the biconditional $P(x) \iff Q(x)$ can be expressed as "$x = -3$ is necessary and sufficient for $|x| = 3$", or perhaps better, $x = -3$ is a necessary and sufficient condition for $|x| = 3$.

Now, consider the quantified statement $\forall x \in \mathbb{R}, P(x) \iff Q(x)$. This statement is false because $P(3) \iff Q(3)$ is false.

Note that sometimes, we arguably misuse characterization, and definition. Supposedly, Let A being a triangle. Then, A is equilateral if it has three equal sides. This is the definition. For the notion introduced in this section, we have the following characterization:

$$A \text{ is equilateral} \iff A \text{ has three equal angles}$$

Notice that the definition is of three sides, and the characteristic of such concept, is that it must then, consequentially, have three equal angles. So the definition is indeed true, but not a characterization, because that is what a definition is, even though the bidirectional relationship is true.

This sounds like more of a rehearsal, but in fact, there are many definitions that use bidirectional conditions to sustain itself of a concrete definition.

That is because it offers something that is analogous to equality in general mathematics, but stronger - a binding between the concept and what is its representation. That is why we must define and differentiate between two 'objects' that is created from such bidirectional statement, definition, and the latter of this section's name. Otherwise, our foundation of definitions, for at least the starter point of mathematics, would be in troubles.

8.4 Informal (Basic) Set Theory

To 'reinforce' our understanding of mathematics, a basic informal discussion on set theory would be required. In fact, a lot of our notions from the above section, comes off from a rough treatment of set theory, without rigorous consideration. The introduction will include naive set theory, a few elementary set-theoretic rigours, and obviously, in a short manner to fit the section.

A set is (informally) a collection of things, without regards to order. The collection itself is regarded as a single object, which means in cases, we can have collection of collections. Elements in a set are only counted once. So for a set, we call it as A, then we have elements inside it, for example, t. We write $t \in A$ to reflect that t is an element of A. If we have more than two elements in A, then $t \neq t'$ is to indicate that they are unique, or different.

To describe a set, we use some way of listing the set: Either by listing, or identification. The latter one, however, is more general, and waste less ink for the printer to work on (I don't have enough money to replace all of that).

Example for listing the elements includes:

$$\mathbb{N} = \{0, 1, 2, \cdots\}$$

of natural numbers set $\mathbb{N}$. For integer $\mathbb{Z}$, then

$$\mathbb{Z} = \{0, \pm 1, \pm 2, \cdots\}$$

and for rational number:

$$\mathbb{Q} = \left\{ r = \frac{m}{n} : m, n \in \mathbb{Z}, n \neq 0 \right\}$$

Identifying though, is easier. If a set E includes elements with some types of properties, we write

$$E = \{x : T(x)\}$$

For example, the set P of all even numbers is

$$E = \{x : x = 2k, k \in \mathbb{Z}\}$$

This is explained as for all even number, there is the property that $x = 2k$ for all $k \in \mathbb{Z}$, that is, x must have an even factor of 2.

"Actionably", we want something to do with sets. For two collection, what do we want first? Perhaps, to compare them. Then, for two sets A and B, A is equal to B written as $A = B$ if,

$$(\forall x) x \in A \iff x \in B$$

i.e. two sets are equal if they have the same element. Formalizing this notion gives us the following principle (rather than definition).

Theorem 8.4.1 (Principle of Extensionality, I)**.** *If two sets have exactly the same members, then they are equal.*

We can state things more concisely and less ambiguously, here and elsewhere, by utilizing a modest amount of symbolic notation. Unless it becomes the abuse of notation, it is always a good way to use them, for the best of reducing unnecessary words and ink to the writer (including me). Thus we have a restatement:

Theorem 8.4.2 (Principle of Extensionality, II)**.** *If A and B are sets, such that for every $t \in A, B$,*

$$t \in A \iff t \in B$$

then $A = B$, or two sets are equal.

The principle of extensionality gives us a concern, over the size of a set. What if there are no elements to compare two sets with? For set theory, we have the notion of a small set, which is a set with only the 'zero' element, i.e. $\{0\}$. An even smaller set would be the empty set $\varnothing$. This set has no members at all. Furthermore, empty set is **unique**, since extensionality tells us that any two such sets must coincide.

A **subset** is as we can say, a set that is inside, or dominated by other sets. It can be such that the subset is a partition of the set that contains it, or else. For any set X, given another set A. A is the subset of X if $x \in A$ then $x \in X$, denoted

$$A \subseteq X$$

or

$$X \supseteq A$$

The first notation is the subset notation, the second one establish a reverse relation, called superset, that is X is a superset of A. If $A \subseteq X$, and we have $y \in X$ but $y \notin A$, then A is a proper subset of X, denoted

$$A \subset X$$

If we want to check for subset, then we have the following definition.

Definition 8.4.1. *A is a subset of B written as $A \subseteq B$ or $A \subset B$ if all elements in A are in B. Generally,*

$$(\forall x) x \in A \iff x \in B$$

From this, we can have that

$$(A = B) \iff (A \subseteq B \wedge B \subseteq A)$$

Suppose X is a set and P is the property of some elements in X, then we can write $\{x \in X : P(x)\}$ for the subset of X containing of the elements of which $P(x)$ is true for $x \in X$. This then can be used for others operations between sets:

Definition 8.4.2. *Given two set A and B, we have the following:*

1. *Intersection:* $A \cap B = \{x : x \in A \wedge x \in B\}$.
2. *Union:* $A \cup B = \{x : x \in A \vee x \in B\}$.
3. *Set difference:* $A \setminus B = \{x \in A : x \notin B\}$.
4. *Symmetric difference:* $A \triangle B = \{x : x \in A \oplus x \in B\}$ *($\oplus$ is the exclusive or notation) i.e. the elements in exactly one of the two sets.*
5. *Power set:* $\mathcal{P}(A) = \{X : X \subseteq A\}$*, i.e. the set of all subsets.*

Theorem 8.4.3 (Proposition of operations)**.** *For A, B, C are sets, we have:*

1. $(A \cap B) \cap C = A \cap (B \cap C)$
2. $(A \cup B) \cup C = A \cup (B \cup C)$
3. $A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$

Finally, of the elementary mean, we might want to introduce you the De Morgan Theorem:

Theorem 8.4.4 (De Morgan)**.** *Given a set X and a collection of set, denoted by $\{A_i\}_{i \in I}$, we have:*

$$X \setminus \left(\bigcup_{i \in I} A_i \right) = \bigcap_{i \in I} (X \setminus A) \tag{8.11}$$

and

$$X \setminus \left(\bigcap_{i \in I} A_i \right) = \bigcup_{i \in I} (X \setminus A_i) \tag{8.12}$$

Under n sets operations, it is better to not write them all down as such, and so similarly to summation $\sum$, and product $\prod$, we have ourselves a set general notation. If A_α are sets for all $\alpha \in I$, then

$$\bigcap_{\alpha \in I} A_\alpha = \{x : (\forall \alpha \in I)\, x \in A_\alpha\}$$

and

$$\bigcup_{\alpha \in I} A_\alpha = \{x : (\exists \alpha \in I)\, x \in A_\alpha\}$$

8.4.1 Abstractions, and the hierarchical structures of sets

In set theory, there's the notion of container. This would come in handy later, and I think it's just at the right time for this discussion, after we at least have an idea of how set operate in mundane sense.

A container is different from its element, but as we have said above, there can even be the collection of collections. A famous example is the empty set. When we say that $A = \varnothing$, we did not say that it has nothing, because, A in this case, is the collection. We instead can write it as: $A = \{\varnothing\}$, specifically refers to the fact that it's a set, without any member. $\varnothing$ hence can be interpreted,

and defined as a state of set - the state of being empty for any set. So, then, we have $\{\varnothing\} \neq \varnothing$, simply because one is a state of objects, while the other one refers to a collection, with that state of object. Intuitively, it's between a man with nothing, and a man with an empty bottle of water - at least he has an empty bottle.

So, thinking about it, ∞ and $\varnothing$ is weird. It can be a thing of it own, i.e. an object being analogous to $\varnothing$ might mean that it is void; on the other hand, an 'infinite' object might refer to its domain, or else. But when you package it into a collection, it becomes 'representative properties' of the collection itself (or the set itself). $\{\infty\}$ can be understood as the infinite collection, or a collection of infinite membership, for being distinct. So there's quite a thing about collection, and just plain resort of objects.

This problem is actually the reason why we begin with Russell's paradox [1] in the first place. We would like to take a detour, and confront Russell's paradox for the moment (Gerstein). We start with a property P and assume that the property can be used to define a set, $\{x \mid P(x)\}$. Consider the set

$$S = \{A \mid A \text{ is a set and } A \notin A\}$$

Notice that some sets are not elements of themselves. The set of integers $\mathbb{Z}$ does not include the set itself. We obtain the paradox when we consider 'the set of all sets which are not member of themselves', or

$$R = \{\text{Sets } A \mid A \notin A\}$$

The question is, is R a member of itself? R cannot be a member of itself, but it must be, since it contains everything. This is a contradiction, hence R cannot be a set. But this explanation is lackluster.

The argument of Russell's paradox is concerned of the set $\{x \mid x \notin x\}$. Is it a member of itself? We think the answer is no. What do you mean by $x \notin x$? What is x in this case? It seems that such thing does not even exist. Why? Because an apple cannot be justified so that it is not itself. Even when we regard that we can have collection of collections, the narrow view when you look at a collection, instead of later scale, is that now the collections inside the collection, is now called element instead. You cannot have an element to not belong to itself, simply because the statement does not make sense - you need to have a collection at the far side of the operation. This means that the whole statement is simply false, hence even S does not exist. Instead, we say we have two things, x and $\{x\}$. Inherently, $x \notin \{x\}$. The notation change - now $\{x\}$ is the collection of x, not just x itself. Then the formula $x \notin x$ is simply rejected, because it is false in interpretation. This indeed, surprisingly, leads to the theory of types, of which Russell himself postulated such. This creates slicing, of which divides things into set of elements, set of sets of elements, and more. In other word, an orderly fashion of types abstraction. The statement $x \notin \{x\}$ though, is wrong, since we now reduce the 'typing' down to element, and its container. One is a set, one is the container of such set. It is obviously wrong, because it's similar to asking if your apple in the bag, is not in the bad that contain such apple.

[1] Also called Russell-Zermelo paradox

On the other hand, if we still accept the notion of the statement, then for $S = \{x \mid x \notin x\}$ is indeed true, and exists, because of the law of scaling and typing. This holds for the next case, $S \in S$?, and the answer is no, since it cannot contain itself, validly, within the typing of scale. So there is no universal set.

What we have done is to reject the existence of even S, thus invalidating the question itself; also, to prove that there is no universal set available. But there are several ways to do this, instead. One example is the treatment of such, so that we cannot create such arbitrary set. New sets can only be created via the above operations on old sets, plus replacement, which says that you can replace an element of a set with another element. This is an example of the treatment of set theory, following ZFC (Zermelo-Fraenkel) set theory, which was formed to counter the existence of Russell's paradox. Another argument, *The von Neumann-Bernays alternative*, also proved to be effective against such paradox, but retains the ability to have a universal set - in this case, is called as a 'proper' class. [2]

8.5 Functions and Relations

8.5.1 Functions

We'll get this section quick, not as much as other sections, becasue the topic itself is too large, and hence, we'll also cover it just as fast as it is large. We begin with the definition of function.

Definition 8.5.1. *A function or map $f : A \to B$ is a rule that assigns for each $a \in A$ precisely one element $f(a) \in B$. We can write $a \mapsto f(a)$. A and B are called the domain and co-domain respectively.*

Formally speaking, we can define it to be a subset $f \subseteq A \times B$ such that for any $a \in A$, there exists a unique $b \in B$ such that $(a, b) \in f$. We then think of $(a, b) \in f$ as saying $f(a) = b$.

A shorter definition can take place:

Definition 8.5.2 (Function, II)**.** *A map f from X to Y, denoted $f : X \longrightarrow Y$ or $X \longrightarrow^f Y$ is a rule that bind each $x \in X$ with exactly one $y = f(x) \in Y$.*

Now, how do we know what types of function to deal with? Yes, function can have a lot of types, but there are the mains, and obviously characteristically only 3.

Definition 8.5.3 (Types)**.** *There are three types of function, which is injective, bijective and surjective. We define those as followed.*

1. *Injective: $f : X \to Y$ is said to be injective if it "hits" everything at most, once:*
$$(\forall x, y \in X) f(x) = f(y) \implies x = y$$

[2] For more information on this debate, see the relevant information in *Herbert B. Enderton, Elements of Set Theory*, Gerstein's argument of the paradox, Plato Stanford's articles on this topic, as well as several introductory literature regarding the same problems.

2. *Surjective: $f : X \to Y$ is said to be surjective if it "hits" everything at least once:*
$$(\forall y \in Y)(\exists x \in X) f(x) = y$$
If we can call it with the later notation, it is analogous to $Imf = Y$.

3. *Bijective: A function is bijective if it is both injective and surjective (hits everything exactly once). Generally,*
$$\forall y \in Y, \exists! x \in X \ \ s.t. \ \ y = f(x)$$

Composition of injective mappings are an injective mapping, so do surjective, thus we also have bijection mappings to have bijective compositions

For the following part, we will discover certain interesting proposition on such function operation.

Definition 8.5.4 (Composition)**.** *The composition of two functions is a function that is made from applying one after another. Function composition is associative.*

Given two maps $f : X \longrightarrow Y$ and $g : Y \longrightarrow Z$, the product or composition of two maps f and g is $g \circ f : X \longrightarrow Z$ and is defined as
$$g \circ f = g(f(x)), \forall x \in X$$

Definition 8.5.5 (Image)**.** *If $f : A \to B$ and $U \subseteq A$, then $f(U) = \{f(u) : u \in U\}$. Then $f(A)$ is the image of A. We denote this by $Imf = f(X)$, of which the set Imf is called image of mapping f.*

By definition, f is surjective iff $f(A) = B$, meaning exactly that "it hits all the domain".

Definition 8.5.6 (Identity)**.** *The identity map $id_A : A \to A$ is defined as the map $a \mapsto a$.*

Suppose $f : X \longrightarrow Y$ is a bijective mapping from X to Y. Then for each $y \in Y$, there exists one and only $x \in X$ so that $f(x) = y$. Then we have a mapping $g : Y \longrightarrow X$ defined as
$$\forall y \in Y, x \in X : g(y) = x, f(x) = y \Longrightarrow g \circ f = i_X, f \circ g = i_Y$$

This notion is generalized to be called **inverse mapping/function**, of which we will break down even, into left and right inverse.

Definition 8.5.7 (Right and left inverse)**.** *Given $f : A \to B$, a left inverse of f is a function $g : B \to A$ such that $g \circ f = id_A$. A right inverse of f is then a function $g : B \to A$ such that $f \circ g = id_B$.*

Theorem 8.5.1. *The left inverse of f exists iff f is injective.*

Proof. If the left inverse g exists, then $\forall a, a' \in A$, we have $f(a) = f(a')$ implies that $g(f(a)) = g(f(a')) \implies a = a'$ therefore f is injective.

if f is injective, we then construct g as
$$g : \begin{cases} g(b) = a & \text{if } b \in f(A), f(a) = b \\ g(b) = \text{anything} & \text{otherwise} \end{cases}$$
Then g is a left inverse of f. □

Finally, we have the concept of equivalent function .

Definition 8.5.8. *Two sets X and Y is called **equivalent**, that is $X \sim Y$ if there exists a bijective relation $f : X \longrightarrow Y$.*

8.5.2 Relations

We begin discussing of relation, with its variety of definition. Note that, though, relation is larger, and more free than function, with respect to their flexibility and transformation constraint.

Definition 8.5.9 (Relations, I)**.** *Let X, Y be sets. A relation $R = R(x, y)$ is a logical formula for which x takes the range of X and y takes the range of Y, sometimes called a relation from X to Y. If $R(x, y)$ is true, we say taht x is related to y by R, and we write xRy to indicate that x is related to y by R.*

Inherently, relation is much larger in term of functionalities, and its dimension of definition, such that there are little restriction on such. Hence, you can even think of the functin f as being a relation restricted by certain criteria, which is normal as being unidirectional simultaneity relation, that is, there's no two y for one x, roughly speaking. Why is this the case for the definition of function that we took for granted, questions are easy to arise from.

Example 8.5.1. Let $f : X \to Y$ be a function. In fact, this is a relation R such that $R(x, y) \equiv (f(x) = y)$

Another type of this is as followed. Defining $X = Y = \mathbb{Z}$.

In another formalism, we define a relation a bit differently:

Definition 8.5.10 (Relations, II)**.** *Let r be a relation. Then for r being a **binary relation**, that is, taking in two arbitrary component, on set A and B, the result is a subset*

$$R \subseteq A \times B$$

*being a set of 2-tuple (a, b). For $(a, b) \in R$, we say that a is related to b by R. A is the **domain** of R, and B is the **codomain** of R. For $A = B$, then r is the binary relation on the set A.*

From this definition, there're a few questions.

1. Of structures, and sets, how do we compare their "size"?
2. How do we measure the operational space of certain structures?
3. Are there the n-relation of which results in certain configuration of the I/O set?

Those questions would be left out for later view, since it's quite irrelecant to the follow-up section, but is good to keep in mind. Also, as you might seen, we can also take into account the operational space of the relation, as well as the narrower version of it, the functions.

For relations, again, we also have a few "universal properties" specified specifically under such pretext:

Definition 8.5.11 (Universal)**.** *Let X be a set, and let $\sim$ be a relation on X.*

- *We say that $\sim$ is reflexive if $\forall x \in X, x \sim x$*
- *We say that $\sim$ is symmetric if $\forall x, y \in X, x \sim y \implies y \sim x$.*
- *We say that $\sim$ is antisymmetric if $\forall x, y \in X, x \sim y \wedge y \sim x \implies x = y$.*
- *We say that $\sim$ is transitive if $\forall x, y, z \in X, x \sim y \wedge y \sim z \implies x \sim z$.*

For this, we might prepare an example.

Example 8.5.2. Let $X = \mathbb{R}$, and define a relation $\leq$ as standard. Consider the above properties:

1. Relexivity: Yes, since $x \leq x$ is always true.
2. Symmetry: No. It does not make sense that $x \leq y \implies yleqx$

8.6 Concluding Remark

This chapter serves as the introductory "topic" on mathematics, as far as the viewpoint into such field is concerned. This includes, in rather fashion, the beginning of mathematical logic, proofs, and the way of doing maths, the further treatment of elementary set theory, and functions, plus relations, two of the foundational operatives that we have.

The only thing perhaps is lacking, is a bit into number theory, and Euclidean geometry as to complete the foundational topic, but I think perhaps I should not bore myself of doing such menial task alone, while giving little space for expansion of ideas (that is, until when I look back and change otherwise). Still, it is nice to remind us of the lesson learned, and the point received from writing such section.

9 Graph Neural Network An introduction

Fujimiya Amane, H. Miharu

A very much **informal** introduction to such practicality flow, we introduce a more 'flesh out' and weirdly interesting view on the neural network and data representation aspect - not as *continuous stream of data*, but *discrete, ordered* or *non-ordered* data and processes. This is encapsulated in its data form, i.e., the **Graph Neural Network** (GNN).

One interesting point about graph neural network, is that it assumes local/neighbourhood responsibility, or rather, **data dependencies**. This differs from the usual i.i.d. assumption, or rather, put it in different dimensions of *independently*, *identically* sampled data. Hence, we can also say that graph network and graph data form has its intrinsic advantage of being more **data-expressive** and data-driven in true sense, than in most cases.

9.1 The graph representation

A **graph** G is a 2-tuple (V, E) where V is the set of **vertices** (or nodes) and E is the set of **edges**. The set $ne[n]$ stands for the neighbour of vertex n, while $co[n]$ is the set of edges that have n as vertex. Edge is often denoted by (u, v) for vertices u and v. We said that (u, v) joins u and v, and it can be directed or undirected. A graph is called a *directed graph* if all edges are direct or *undirected graph* if all edges are *undirected*. The degree of vertices v, denoted by $d(v)$, is t number of edges connected with v. The graph data can be loosely (not specifically in cases) as followed. Let $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{W})$ be a graph, where $\mathcal{V} = \{1, \dots, n\}$ is the set of nodes, $\mathcal{E} = \{1, \dots, M\} \subseteq \mathcal{N} \times \mathcal{N}$ is the set of edges, and $\mathcal{W} : \mathcal{E} \to \mathbb{R}$ is the edge's weight function. There is also the optional weight $\mathcal{W}' : \mathcal{V} \to \mathbb{R}$ for each vertices. In this case, it is for certain problem like TVP, where each cities have certain properties for the path. We say that a data sample $\mathbf{x}$ is a graph data, if its entries are related through the graph $\mathcal{G}$. Below is a typical undirected graph with nodes and edges. Note that in this case, there is no direction indicated.

9.1.1 Usage of Graph

Now, the question is, what is the use of such data? What is the purpose for graphs and its proponents? First, **graph**, as it might be, is **relational**. Almost everything, even the learning pattern is relational. In fact, because of the limitation of continuous data given in reality, we might even can treat regression as the problem of finding an effective **generator** that generates news

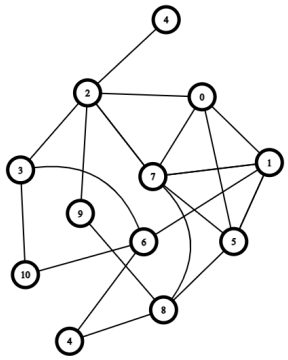


Figure 9.1: A typical **undirected** graph with nodes and edges

nodes and edges according to certain rules and laws that describe the relational property, alongside the already embedded relational property - here is the **order relation** between vertices. But usually, data can be naturally represented by **graph structures**, more in several application areas. Typically, those are:

1. **Natural Language Processing** (NLP) - Information in NLP usually comes in shape of continuous sequential, arbitrary length string of words and numbers. Between words, numbers, and the data, there is the structure of such natural language - the **grammar**, then the contextual underlying - the **context** of such sentence/paragraph/documents. This is very much likely to be represented in term of graph - in fact, the relational diagram often seen in loose presentation of NLP is exactly a type of graph theoretical *graph* - it is because of the permutation invariance property of graphs.

2. **Conditional representation** - Sometimes, we want to model relationship between objects, persons, systems' properties, causal relationship, etc. For this, graph neural network is advisable. This is also a case for combinatorial representation - for example, the Travelling Salesman Problem (TSP) that utilize multiple combinatorial choice from one city - A to another - B. Those choice can be represented as a space of encoding space D of distances, and the graph $G(V, E)$ itself.

3. **Relational formation**: Those with innate relations would be able to be formed using graph. The job now, though, is to formulate the *relational strength* and properties between them. This is extended as relational formation when it is equipped with varied sensory system. The job now is to form a relational understanding from simply data alone.

4. **Knowledge representations**: Most of the time, again, knowledge can also be represented as a graph. This puts us to the task of **knowledge relations** and **knowledge-filling reconstruction**, that is, finding missing knowledge in the graphical system, and patch it in.

Overall, quite much. To be fair, it is quite natural to think of it as this way, since it makes sense for higher data representation than just a latent space with only point-element. The spirit is quite done, now, what do we want to extract from graph?

9.1.2 The Graph Problem

TO partially answer the above question, a learning scenario, or simply the learning problem must be addressed.

Definition 9.1.1 (Graph-theoretical learning)**.** *Given a graph $G(V, E, \psi_E)$ for ψ_E the edge proprietary classification, there exists an encoding space $\mathsf{ENC}(G)$ that the graph lives in, and a function $\phi : (E, \psi_E) \to (E', \psi'_E)$ such that to change the configuration of the connector space. The **learning problem** is for a learner $L(H, M, C)$ to learn a function ϕ that is appropriate of the intended use case, such that for any given data $D(V^*, \odot)$, either:*

1. *Assign E^* and transform to ψ^*_E.*
2. *For existing E^*, transform to ψ^*_E.*

All within marginable error evaluation of L.

The learner, $L(H, M, C)$, consists of the hypothesis - or rather, pattern memorial H contains the learner's perception of the problem, the **memory** M consists of experiences (in a sense, somewhat similar to recurrent actions) with accordance from the neightbourhood, and C is the various *configurations.* available.

Question 9.1.1 (Data classes). For any given learner L, can there exists any given any class of graph that it is not able to learn?

Question 9.1.2 (Efficiency). For a learner L, assume the coherent representability of the graph pattern H. How many counter M there are, or iterative sequence that is needed, so that for $\epsilon > 0$, and $\delta > 0$, then

$$\Pr_{x,y \in D}[(L(x) \neq y) < 1 - \epsilon] < 1 - \delta \tag{9.1}$$

assuming the usual inference pair $(x, y) \in D$?

Those question and several more are reserved for later analysis. But for now, they are a reasonable set of question directed toward the statistical analysis of GNN, since we abandoned the i.i.d. question in mind.

9.2 An analytical view on GNN

GNN in its form creates its operational space upon graph data, for its decision template to be dealt with. This is enabled, using the first property of graph and relational graph: *every* node with edges to any other node, is **influenced** by its directly connected node. Given a graph $G(V, E)$, then the neighbourhood $neigh(v_i)$ for $v_i \in V$ is defined by

$$neigh(v_i) = \{v_i, v_j\}_j : (v_i, v_j) \in E$$

We adopt a more friendly notation (this is most taken from [Franco et al., 2009]). The label attached to node $n \in V$, and edge (n_1, n_2) is denoted $\mathbf{l}_n \in \mathbb{R}^{l_n}$, and $\mathbf{l}_{(n_1,n_2)} \in \mathbb{R}^{l_E}$. Let $\mathbf{l}$ denote the vector obtained by stacking together all the labels of the graph.

The domain considered as **supervised learning** for such dataset of the learning algorithm is defined as follow:

$$\mathcal{L} = \{(\mathbf{G}_i, n_{i,j}, \mathbf{t}_{i,j}) \mid \mathbf{G}_i = (V_i, E_i) \in \mathcal{G}; n_{i,j} \in V_i; \mathbf{t}_{i,j} \in \mathbb{R}^m, \\ 1 \leq i \leq p, 1 \leq j \leq q_i\} \quad (9.2)$$

Where $\mathbf{t}_{i,j}$ is the desired target associated with $n_{i,j}$, $p \leq |\mathcal{G}|$ and $q_i \leq |V_i|$. All the graphs of the learning set can be combined into a unique disconnected graph, and therefore the learning set can be represented as the 2-tuple $\mathcal{L}(\mathbf{G}, \mathcal{T})$ instead.

The **model**'s intuitive idea is that nodes in a graph represent **objects** or **concepts**, and edges represents their relationships. Each concept is naturally defined by its feature and the related concepts. Thus, we can attach a *state* $x_n \in \mathbb{R}$ to each node n that is based on the information contained in the neighbourhood of n. The state x_n contained a representation of the concept denoted by n and can be used to produce an output o_n i.e. a decision about the concept.

Let f_w be a parametric function called the *local transition function*, that expresses the dependence of a node n on its neighbourhood and let g_w be the local output function that describe how the output is produced. Then x_n and o_n are defined as:

$$\begin{aligned} x_n &= f_w(\mathbf{l}_n, \mathbf{l}_{co[n]}, x_{ne[n]}, \mathbf{l}_{ne[n]}) \\ o_n &= g_w(x_n, \mathbf{l}_n) \end{aligned} \quad (9.3)$$

where $l_n, l_{co[n]}, x_{ne[n]}$ and $l_{ne[n]}$ are the label of n, the label of its edges, the state and labels of the nodes in the neighbourhood of n respectively.

Different notions of neighbourhood can be adopted. In general, the equation could be simplified into several different **data embedding**, for example, containing everything in $x_{ne[n]}$, or just x_n. Furthermore, the above notion is optimized for undirected graphs. When dealing with undirected graph, the function f_w can also accept as input a representation of the direction of the arc (in a sense, we can implement it as another data section, ψ_E like the model above).

Let x, o, l and l_N be the vector constructed by stacking all the states, the outputs, the labels, and all the node labels respectively. Then we can rewrite (1) in compact form as

$$x = F_w(x, l), \quad o = G_w(x, l_N) \quad (9.4)$$

where F_w is the *global transition function* and G_w being the *global output function*, stacked version of $|N|$ instances of f_w and g_w, respectively.

> Depends on the structure of the graph, here we are interested in the case where x, o are uniquely defined (what do you mean by *uniquely defined*?) and (2) defines a map $\varphi_w : \mathcal{D} \to \mathbb{R}^m$ which

takes a graph as input and return an output o_n for each node. The *Banach fixed point theorem* provides a sufficient condition for the existence and uniqueness of the solution of a system of equations, and according to such, (2) has a unique solution provided that F_w is a contraction map with respect to the **state**, that is, the exists $0 \leq \mu < 1$ such that

$$||F_w(x, l) - F_w(y, l)|| \leq \mu ||x - y||$$

for any x, y on the vectoral norm $||\cdot||$. [Sca09]

9.2.1 GNN versus MLP

We have jumped quite far from the usual standard of machine learning theory. Put asides the questions of philosophical **intelligence** (and artificial alike), the main architecture that enables such advancement (if one want to call such probabilistic approach as advancement), is the **neural network** model of itself. Usually, this is established within the basis of a Multilayer Perceptron (MLP) structure. Here, however, we takes more of the form of graph theory, the theory of nodes and edges. Intrinsically, however, they works under the same model of neural unit. Before jumping in, there are perspective to look upon.

The difference of procedure

GNN and MLP is very different. On one side, you have the system of MLP, being represented of

$$\mathsf{MLP}(n) = (I, L_n^{(k)}, O)$$

with a single flux direction - i.e. the normal I/O procedure. There is no communication between $\{L_j^{(i)}\}_{i \leq k}$ or the **neural processing unit** between layers. Most of the time, neural network can be seen as a system of directed edge, *bipartite graph* between successive layers, and every neuron is a subgraph containing the following procedure

$$\mathsf{NEU}(I_m, \cdot, O_1) = (I_m, \Sigma, \phi, u_t, O_1)$$

Where I_m, Σ, ϕ and u_t are, accordingly, the m-input handler, the input signal processing, ϕ is the interpretable unit of the neuron, and O_1 is the single output directive afterward. Control injection starts with $I_m, \mathbf{W}_I$ for a series of weight accordingly, within the assumption of "change of behaviours through input confidence", as we have seen. In fact, the entire structure of a single neuron is done as

$$O_1^t = \mathop{\mathbb{E}}_{t \to 0, I \in \mathcal{D}} \phi[\Sigma(I_m \odot u_{t-1}, \mathbf{W}_I, \mathbf{b})]$$

the $\mathbb{E}$ is there for averaging purpose expression only. And the entire process is a bunch of nested function, of which then **backpropagation** algorithm is used.

For GNN, backpropagation is still used, but generally, there are a few (if not a lot) differences. First of all, GNN generally do not have *directions*. Or

rather, a mono-directed viewport. There is indeed I/O procedure for such network, but the graph $G(V, E)$ usually have no property of directed node for $e_i \in E$. The graph itself, $G(V, E)$, is not the operating structure, in most cases of usage. Rather, it is instead, the figurative layer-top aggregation and mask (for example, convolutional pooling mask on top of everything). Hence, for example, the task of graph reconstruction, it is such that we are given

$$\mathcal{D} = \{\mathcal{G}, \mathcal{G}'\} \to \{(V, 0), (V, E)\}$$

in which the given data scenario, and the "true" data mask. This gives plenty flexibility, for every encoded $\mathbf{H}_v \in \mathbb{R}^d$, of vertex $v \in V$, the only thing that is indeed, mutable for control, is the edges. Comparing this to neural network, where the operating system inherently have too many "complexity" in control, it is in a sense, that the restricting behaviours of the communication method - the edges, is nice enough for graph neural network.

The data

We can not state a lot of things about the data, however there is. But we can have first, one partial question:

> Can MLP interface, be interpreted in the form of graph?

In one way or another, the answer might be yes. Recalling that MLP is a restricted form of graph GNN, i.e. the bipartite, single-directed, convergent graph tube (the entire structure is indeed a tube). This is inherently similar to GNN. However, the way we treat the data of the structure is different.

Remember again, that MLP itself is a strict I/O process. This means it cannot utilize its internal state. In fact, MLP has the **hidden state** exactly for that reason. It uses the representation itself, in cohesion with the single directed graph, to calculate and push everything to the output node at the end. However, GNN took the over-the-top approach. You are, inherently exposed to the graph structure, or half of it, by the vertices. There, you have all of the data presented, living in its metric space. There, you then extract the latent space of each vertex, according to one's own agenda of choice, by aggregation - or rather, **method of local estimation**. From this, essentially, you are acting on the structure of the data on its own. Strictly on the side of functional estimation (MLP representation) and GNN, MLP is in fact, looking only at the inherent mutation of controllers, while GNN use an over-the-top latent space to dictate the structure of the predicted value.

10 Elementary Number Theory On Euclidean's algorithm and their uses

FUJIMIYA AMANE, H. MIHARU

Computation is hard. Indeed, for the old timer's fundamental arithmetic operations, there are addition, subtraction, multiplication, then divisions. And there is the greatest common divisors calculation, which is the main focus on this chapter, for what it is, and what have been done in my own time reading about it - including the efficient way of calculating such divisors in time-ly manner. Of course, no one wants to wait too much, isn't it?

10.1 Preliminaries

We have several definitions that must be settled before the algorithm is stated.

Definition 10.1.1 (Factor of integers)**.** *Given $a, b \in Z$, we says that a divides b, a is a factor of b or $a \mid b$ if $(\exists c \in \mathbb{Z})b = ac$. For any $b, \pm 1$ and $\pm b$ are always factors of b. The other factors are called proper factors.*

Theorem 10.1.1 (Remainder theorem)**.** *Given $a, b \in \mathbb{Z}, b \neq 0$, there are always unique $q, r \in \mathbb{Z}$ with $a = qb + r$ and $0 \leq r < b$.*

Proof. Choose $q = \max\{q : qb \leq a\}$. This maximum exists because the set of all q such that $qb \leq a$ is finite. We then have:

$$r = a - qb$$

Then $0 \leq r \leq b$ and thus q and r are found. To show that they are unique, we suppose that

$$a = qb + r = q'b + r'$$

Then

$$(q - q')b = (r' - r)$$

Since $0 \leq r, r' \leq b$, we have $-b < r - r' < b$. However, since $r - r'$ is a multiple of b, thus $q - q' = r' - r = 0$ Consequently, $q = q'$ and $r = r'$.

□

Definition 10.1.2 (Common factor)**.** *A common factor of a and b is a number $c \in \mathbb{Z}$ such that $c \mid a$ and $c \mid b$.*

There, we then define the **greatest common divisors**

Definition 10.1.3 (GCD)**.** *The highest common factor or greatest common divisor of two numbers $a, b \in \mathbb{N}$ is a number $d \in \mathbb{N}$ such that d is a common factor of a and b, and if c is also a common factor, $c \mid d$. We denote*

$$d = \text{hcf}(a, b) = \gcd(a, b) = (a, b)$$

Clearly, if the hcf exists, it must be the largest common factor, since all other common factors divide it, and thus necessarily unique.

Proposition 10.1.2. *If $c \mid a$ and $c \mid b$, then $c \mid (ua + vb)$ for all $u, v \in \mathbb{Z}$.*

By definition, we have $a = kc$ and $b = lc$. Then $ua + vb = ukc + vlc = (uk + vl)c$. Then $c \mid (ua + vb)$.

Theorem 10.1.3. *Let $a, b \in \mathbb{N}$. Then $(a, b) = \gcd(a, b)$ exists.*

Proof. Let $S = \{ua = +vb : u, v \in \mathbb{Z}\}$ be the set of all linear combination of a, b. Let d be the smallest positive member of S, say, $d = xa + yb$. Hence if $c \mid a$ and $c \mid b$, then $c \mid d$. We need then to show that $d \mid a$ and $d \mid b$, thus $d = (a, b)$.

By the division algorithm ,there exists a number $q, r \in \mathbb{Z}$ such that $a = qd + r$. Then $r = a - qd = a(1 - qx) - qyb$. Therefore r is a linear combination of a and b. Since d is the smallest positive member of S and $0 \leq r \leq d$, then $r = 0$ and thus $d \mid a$. Similarly, $d \mid b$. □

Corollary 10.1.4. *Let $d = (a, b)$. Then d is the smallest positive linear combination of a and b.*

Lemma 10.1.5 (Bézout's identity)**.** *Let $a, b \in \mathbb{N}$ and $c \in \mathbb{Z}$. Then there exists $u, v \in \mathbb{Z}$ with $c = ua + vb$ iff $(a, b) \mid c$.*

Proof. Let $d = (a, b)$. If c is a linear combination of a and b, then $d \mid c$. Suppose that $d \mid c$, then for $d = xa + yb, c = kd$, we have: $c = (kx)a + (ky)b$ which is a linear combination of a, b. □

Some supplementary pieceworks left, required as detailed description only in this case.

Lemma 10.1.6. *Let $d = \gcd(a, b)$. Then*

$$\gcd\left(\frac{a}{d}, \frac{b}{d}\right)$$

Proof. Let $a = a'd$ and $b = b'd$. Suppose that $e \mid a/d$ and $e \mid b/d$. Then $a/d = ex$ and $b./d = ey$ for some $x, y \in \mathbb{N}$. Thus $a = exd$ and $b = eyd$. Observe that $ed \mid a$ and $ed \mid b$. But d us the gcd, hence $e = 1$, as required. □

10.2 The Euclid's Algorithm

Computing the greatest common divisor from scratch, for large number, is incredibily hard. In fact, not so much people want to do it, because it is so time-consuming and error terms because of the substantially large amount of

factorization needed, and testing for those numbers. One way to optimize this, though, is thought up in ancient mathematics, of a way to connect the lower greatest common divisor to the greater one. What if the larger problem and the smaller problem belong to the same *basis*? This is expressed in the following lemma.

Lemma 10.2.1 (GCD basis)**.** *Let $a, b \in \mathbb{N}$, with $b \neq 0$. Let $a = bq + r$, $0 \leq r < b$. Then:*

$$\gcd(a, b) = \gcd(b, r) \tag{10.1}$$

Proof. Let d be a common divisor of a and b. Since $a = bq + r$, we have that d is a divisor of r. It follows that any divisor of a and b is also a divisor of b and r.

Now, let $d = \gcd(b, r)$. Since $a = bq + r$, we have that $d \mid a$. Thus any divisor of b and r is a divisor of a and b.

It follows that the set of common divisors of a and b is the same as the set of common divisor of b and r. Thus $\gcd(a, b) = \gcd(b, r)$ □

The point of this lemma is to connects the dot, such that $b < a$ and $r < b$. So why wasting time calculating the big number when you can do sucessively, a chain of algorithmic calculation?

Proposition 10.2.2. *If we continuously break down a and b by the following procedure:*

$$\begin{aligned} a &= q_1 b + r_1 \\ b &= q_2 r_1 + r_2 \\ r_1 &= q_3 r_2 + r_3 \\ &\vdots \\ r_{n-2} &= q_n r_{n-1} \end{aligned} \tag{10.2}$$

Then the highest common factor is r_{n-1}.

Proof. We have $(a, b) = (b, r_1) = (r_1, r_2) = \cdots = (\text{factor of } r_{n-1})$. Hence the algorithm converges. □

The algorithmic procedure is as followed.

The extended version of Euclid's algorithm, which works it way from the bottom up with the assumption of the gcd in the form $xa+yb$, is useful, somewhat, in the particular application in solving the Bezout identity (theorem) such that

$$\gcd(a, b) = xa + yb \tag{10.3}$$

Lemma 10.2.3. *$c \mid a$ and $c \mid b$ iff $c \mid \gcd(\mathrm{a}, \mathrm{b})$*

Proof. Let $d = \gcd(a, b)$. Suppose that $c \mid a$ and $c \mid b$. We can find integers x and y such that $d = xa + yb$. It follows immediately that $c \mid d$. Conversely, suppose that $c \mid d$. By definition, $d \mid a$ and $d \mid b$ so it is immediate that $c \mid a$ and $c \mid b$. □

10.2.1 Euclid's element

Euclid's algorithm turns out to be of fundamental importance in modern cryptography. It is therefore perhaps surprising that it is 2,300 years old.

We know virtually nothing about Euclid himself although by the close scrutiny of ancient texts scholars have deduced that he lived around 300BC, that he was probably educated in Athens, and that his working life was spent in Alexandria.

Despite the obscurity of his life, he is famous because of the book he wrote known in English as the Elements from the Greek Stoicheia. This book is the single most influential maths book ever written, arguably the most influential science book ever written, and one of the most influential books — period, as the Americans would say — ever written.

11 Bias-Variance Tradeoff

Including the overfitting/underfitting jargon inside

Fujimiya Amane, H. Miharu

The **bias-variance tradeoff** is the basic tradeoff in *error analysis* (the analysis of performance of the learning procedural), which is the highest echelon of metric and analytical form. This, in turn, creates the "bias-variance" theory of statistical inference, of which double descent is concerned.

The topic of double descent can be said to be the branching behaviour from this supposed bias-variance rule, of which introduces new **interpolation threshold**, new hypothesis of error form, new hypothesis on the generalize model complexity, and more.

This section aims to introduce the basis for analysis of the tradeoff - and the classical theory of error analysis surrounding such hypothesis. This includes

1. Definition of **bias** and **variance** - their relations, and the bias-variance tradeoff. Its behaviours and decomposition, relations to other measure of model, and hypothesis of such.

2. A discussion on the fallacy of bias-variance tradeoff - this actually has been notice since a long time now back when the first paper representing the tradeoff was presented.

3. Bias-variance decomposition on some of the standard measure of error upon the value space of Euclidean measure, and finite dimensional space.

4. Correctness range of bias-variance tradeoff, in comparison to the new modern "regime" of observation that clearly even separate the tradeoff's descent phase into n-descent.

5. A few discussion and example of the classical theory, and where it fails in even the simplest case - polynomial regression of n-degree polynomial $p(x, n)$.

Those are the main topics, subsequently introduced in this section alone. Certain notions that bears similarity and accordance will also be introduced, for example, the **approximation-estimation tradeoff**, and else. However, we notice that the "scope" of functionality is important in guiding which kind of tradeoff is similar in nature to the according phenomena.

Furthermore, because our main goal, as per this section, is to diving deeply into the central theory of bias-variance tradeoff and its outlier - namely the

double desent phenomena, it is imperative that we should focus on several formal definition of the setting, including:

1. A formal, and analytical definition on bias and variance, and its bias-variance tradeoff.
2. A formal analysis on the error space and its representative manifold (if ever).
3. A formal definition on the notion of **descent**, and the notion of double descent in either imperative, or parameter controlling behaviour. Either way, a formal definition for experiments, and a formal definition for analytical viewport.

11.1 Underfitting and Overfitting

Before jumping in, we ought to get a sense of the theory behind the classification of the two events called **underfitting** and **overfitting**. They are both important, and related to the notion of bias and variance, as well as the direct anectote of which both terms point to - the model complexity and its correctness.

11.1.1 Setting

We again start with the setting of which we would have our model being examined. Let the sample distribution be $p(x, y)$. The data available for appearance, $\mathcal{D}$, for simulating limited generalization, is partitioned into n-fold, such that:

$$\mathcal{D} = \{x_i, y_i\}_{i \leq m} = \bigsqcup_{i=1}^{n} \mathcal{D}_i \quad s.c \quad \mathcal{D}_1 \left(\bigsqcup_{j=2}^{n} \mathcal{D}_j \right)^{-1} > 1 \tag{11.1}$$

The first partition, $\mathcal{D}_1$, is called the **training data set** of the dataset. For $n = 2$, then $\mathcal{D}_2$ is called the **test data set**; and for $n = 3$, $\mathcal{D}_3$ is called the **validation data set**, and so on. However, the ratio ensures that the ratio between the training main set, and all other partition, remains unequal - more on the training sample if any. All of those data are drawn from the distribution, such that $(x_i, y_i) \sim p(x, y)$ for all $1 \leq i \leq n_t r$. Temporarily, we would denote the training set as $\mathcal{D}_{tr}$, and hence, $sizeof(\mathcal{D}_{tr}) = n_{tr}$.

We train a classifier $h \in \mathcal{H}$ on this training set using some form of ERM. Here, we use the regularized version of such operation:

Definition 11.1.1 (RERM-train)**.** *Given a model $h \in \mathcal{H}$, and its target concept c with sample point $\{x_i, y_i\}_n$ per the training data set, then the **regularized empirical risk minimization** (RERM) on the training set $\mathcal{D}_{tr}$ is defined as:*

$$\hat{h} = \arg\min_{h \in \mathcal{H}} \sum_{i=1}^{n_{tr}} \mathcal{L}(y_i, h(x_i)) + \lambda \Omega(h) \tag{11.2}$$

where $\mathcal{L}(y_i, h(x_i))$ is the error function and $\lambda(h)$ is the regularizer.

Before continue, let's smooth out the notation a bit. We would like to use the set-theoretic notation so that we are explicit about what data set we are averaging over:

$$\sum_{i=1}^{n_{tr}} \mathcal{L}(y_i, h(x_i)) = \sum_{(x,y)\in\mathcal{D}_{tr}} \mathcal{L}(y_i, h(x_i)) \tag{11.3}$$

so that the training set is emphasized, and also, it simplifies the notation.

Secondly, we can re-interpret that ERM process as optimnize possible model, for the best accuracy measurement possible. For this, we define an accuracy function $\mathcal{A}$ that is antitonely related to $\mathcal{L}$, such that the hypothesis criterion is written as

$$\hat{h} = \arg\max_{h\in\mathcal{H}} \sum_{(x,y)\in\mathcal{D}_{tr}} \mathcal{A}(y_i, h(x_i)) - \lambda\Omega(h) \tag{11.4}$$

while still prefering smaller model by the term λ.

Now, once we have $\hat{h}$, we wish to use it, or rather, **evaluate** it in reality (or at least a fraction of such). In such case, the other partition of the dataset is used, practically, but why we have to segmenting into such computational process, has something to do with the ideal form of that evaluation itself. In an ieal world, we will evaluate $\hat{h}$ over the entire sample distribution $p(x, y)$, provided there exists a consistent distribution, and is apparent to use. This would mean we will have to compute

$$\text{ac}(\hat{h}) = \mathop{\mathbb{E}}_{\mathcal{D}\sim p(x,y)}[\mathcal{A}(Y, \hat{h}(X))] = \int_{x,y} p(x,y)\mathcal{A}(y, \hat{h}(x)) \tag{11.5}$$

Unfortunately, we do not have access to the actual distribution $p(x, y)$. In reality, we only assume its existence, but the distribution itself is unknown. Even if we know it, the computational process is still impossible, because there are infinite sample point in such regard. Even with the central limit theorem, density sampling would still be unstable in certain regard.

Thus, we instead turns to the notion of set acccuracy, of which compute the accuracy $\text{ac}_{\mathcal{D}}(\hat{h})$ on any given data set $\mathcal{D}$ as

$$\text{ac}_{\mathcal{D}}(\hat{h}) = \frac{1}{|\mathcal{D}|} \sum_{(x,y)\in\mathcal{D}} \mathcal{A}(y_j, \hat{h}(x_j)) \tag{11.6}$$

As a law, we have that

$$\lim_{n\to\infty} \text{ac}_{\mathcal{D}_n}(\hat{h}) = \text{ac}(\hat{h}) \tag{11.7}$$

by the law of large number. Thus we can also denote the training and test as $\text{ac}_{\mathcal{D}_{tr}}(\hat{h})$ and the test set as $\text{ac}_{\mathcal{D}_{te}}(\hat{h})$

11.1.2 Overfitting

After defining the metric for accuracy, we have assumed the proxy of which to be used evaluating the model. We say that a learnt hypothesis $\hat{h}$ overfits the training samples $\mathcal{D}_{tr}$ if there exists some other hypothesis h' that perform worse than $\hat{h}$ in training, but better in testing.

Definition 11.1.2 (Overfitting, I). *We say that $\hat{h}$ overfits $\mathcal{D}_{tr}$ if there exists $h' \in \mathcal{H}$ such that:*

$$ac_{\mathcal{D}_{tr}}(\hat{h}) > ac_{\mathcal{D}_{tr}}(h'), \quad ac_{\mathcal{D}_{te}}(\hat{h}) < ac_{\mathcal{D}_{te}}(h') \tag{11.8}$$

This definition is consistent with multitude of different definition used widely, for example, [Tom Michael, 1997]:

Definition 11.1.3 (Overfitting, Tom, 1997). *Given a hypothesis space $\mathcal{H}$, a hypothesis $h \in \mathcal{H}$ is said to overfit the training data if there exists some alternative hypothesis $h' \in \mathcal{H}$ such that h has smaller error than h' over the training example, but h' has a smalelr overall error than h over the entire distribution (or data set) of instance.*

However, how do we tell if such hypothesis overfits on the training data, without actually verifying the existence of the hypothesis h'? In other word,

Question 11.1.1. Given the measure $ac_{\mathcal{D}}(\hat{h})$ for any given hypothesis of certain operation, for a stochastics process, how do you tell there exists certain "best possible of the hypothesis class $\mathcal{H}$" such that $\hat{h} \neq h'$?

The classical answer is with the complexity of the hypothesis, and the extend of the data set n.

11.1.3 Underfitting

Under similar consideration, we define the underfitting situation as below, where there exists certain hypothesis that is the opposite of the overfitting case.

Definition 11.1.4 (Underfitting). *We say that $h \in \mathcal{H}$ **underfits** the training data $\mathcal{D}_{tr}$ if there exists $h'' \in \mathcal{H}$ such that*

$$ac_{\mathcal{D}_{tr}}(\hat{h}) < ac_{\mathcal{D}_{tr}}(h''), \quad ac_{\mathcal{D}_{te}} < ac_{\mathcal{D}_{te}}(h'') \tag{11.9}$$

Both of such definition is then concerned of the "generalization" problem, of which should be formulated, yet for certain time, it is not well-defined or understood. In usual jargon of complete definition, it is loosely referred to as the ability to estimate, and predict unseen data, but no actual or at least good approximation of such definition is in place. We note the several following points:

1. The notion of underfitting and overfitting falls in line, to the action of optimization analysis. This itself, is a measure proxy, of which maps the model-operator dual space to the proxy accuracy space. Hence, it is reasonable to said that all the above operations and measure of fits is in the accuracy measure space, such then analytical studies should be considered. However, the "changing" factor that give rises to such measure space forms and its analytical properties is not understood - however, there's a hint of which it can be done - through reasonable **domain and procedural analysis**, breaking down everything into certain (non)closed-form analytical expression, or else, a given algorithmic approach. Again, though, it is better to justify such analysis in term of not so "strictly" sense - such that the procedural can be understooded in a more process-like manner.

2. Generalization is typically the concern for such accuracy space - the proxy itself has its function is to identify certain measure of such generalization ability of the model. The definition, in such form, should be created accordingly.

3. The notion of model complexity $MC(\mathcal{M})$ is also typical of concern, too. Usually, this type of consideration often means counting the number of parameters expressible by the kernel of the model, or by specifying the computational level and depth such model used to take onto, given the sense of thing. However, the true definition of such is not defined yet, as we shall see.

Then moving on, before jumping next at hand, and assume the prelimiary knowledge, we know that there are certainly three things that we have to face it sooner or later: the **model complexity**, the **generalization formulation**, and the **measure space of accuracy** definition.

Now, we will change to the second and third measure similar to such form, and pretty much related to the accuracy space - bias and variance.

11.2 Bias

In such manner as accuracy being defined as above, the notion of bias is more direct - that is, it does not depends that much on the "way of segregation" on which we ought to "test out generalization. Rather, it is a measure of the model as a whole, throughout its operation.

In a loosely defined fashion, the **bias** of any estimator/hypothesis model measures the **central tendency** of such model, to the true function of f. Of such, we have the first definition:

Definition 11.2.1 (Bias, I)**.** *Given a model $M[h, S]$, where $h \in \mathcal{H}$ and S is the associated data, the **bias** of h to the true concept $c \supset \{S\}$ is defined as the measure of estimation of the central tendency of the hypothesis to the true concept:*

$$\text{Bias}(h_S) = \underset{x \sim \mathcal{P}}{\delta} \{\mathbb{E}[h_S(x)], f(x)\} \tag{11.10}$$

where $\delta(\cdot, \cdot)$ is the difference measure, of certain measure space associated with the hypothesis and concept.

From such definition, we can change it, for example, to the more typical regression task, then

$$\text{Bias}(h_S) = \underset{x \sim \mathcal{P}}{\mathbb{E}}[h_S(x)] - f(x) \tag{11.11}$$

Notice that we are comparing the total function, that is perhaps every actionable data point, to the hypothesis. So, in fact, a better formulation given the praticality, under regression setting (this of course can be generalized later) should be:

$$\text{Bias}[h_S(x)] = (m^{-1}) \sum_{j=1, x_j \sim \mathcal{P}}^{m} \left\{ \left[(n^{-1}) \sum_{i=1}^{n} h_{S,i}(x_j) \right] - f(x_j) \right\} \tag{11.12}$$

where m, n is the total amount of sample point avaliable for experiments (if we count letting the model run and operational as testing and experimenting), and the total times that sample is repeatedly experimented upon (or the total runs within the same sample set), respectively. There, the association index also lies accordingly, for example, x_j being the j-th points in the dataset, coressponding to "example counts", simply speak; while $h_{S,i}$ means the i-th iteration, of the sample S associated.

Within such formulation, we have the iterative definition of bias:

Definition 11.2.2 (Bias, Iterative)**.** *Suppose a model with the form $M(h, S)$, where $S \subset \mathbb{R}^{n+m}$, and a **logging set** contaning previous iteration of the model in the procedure $\mathcal{T}(M, L)$, of such that $S \subset L$ and $L = \mathbb{R}^{m+p}$, where m is the index of the number of example, and p is the index of the number of ensemble (for each iteration in the past). Then, the bias at the t-th iteration is (assuming $p = t - 1$) is defined as*

$$\text{Bias}[h_S(x), t] = (m^{-1}) \sum_{j=1, x_j \sim \mathcal{P}}^{p=t-1} \left\{ \left[(n^{-1}) \sum_{i=1}^{m} h_{S,i}(x_j) \right] - f(x_j) \right\} \tag{11.13}$$

In the more considerate approach, we would want to add non-negativity as a normal metric, hence we can square the term out as a way of "positive scaling", and then normalize it back. Coincidentally, this way of normalizing and treatment sounds, and feel simular to the notion of **standard deviation** in normal statistical analysis. This is expressible as

$$\text{Bias}[h_S(x), t] = (m^{-1}) \sqrt{\sum_{j=1, x_j \sim \mathcal{P}}^{p=t-1} \left\{ \left\| \left[(n^{-1}) \sum_{i=1}^{m} h_{S,i}(x_j) \right] - f(x_j) \right\|_2 \right\}} \tag{11.14}$$

where $|| \cdot ||_2$ is the 2-norm, or the Euclidean norm. So in one way or another, we can think of bias, as the de-facto standard deviation of machine learning model.

Overall, such definition enables us to inspect such bias with reasonable iterative behaviour. We should also note that, this can be generally generalized for the case of overfitting and underfitting. First, however, notice that in case of such fitness is observed, we have:

$$\text{Bias}[h(\mathcal{D})] = \frac{\text{Bias}[h(\mathcal{D}_{tr})] + \text{Bias}[h(\mathcal{D}_{te})]}{2} \tag{11.15}$$

This applies, generally, to others partitioning. However, the total score should also take into account of the **parition size** for each partitioning, especially in the case of uneven corss-validation testing, of which then such is prominent.

Definition 11.2.3 (Overfitting, Bias-term)**.** *Given the partition $\mathcal{D} \to \{\mathcal{D}_{tr}, \mathcal{D}_{te}\}$, then the model $M(h, \mathcal{D})$ is said to overfit if:*

$$\lim_{|\mathcal{D}| \to \mathcal{D}_{max}} ret(M_{tr}, M_{te}) = \lim_{|\mathcal{D}| \to \mathcal{D}_{max}} \frac{\text{Bias}[h_{\mathcal{D}_{te}}(x)]}{\text{Bias}[h_{\mathcal{D}_{tr}}(x)]} = 0 \tag{11.16}$$

or, given an assumption:

$$ret(M_{tr}, M_{te}) \in \mathcal{O}(x^2) \tag{11.17}$$

In similar fashion, underfitting can also be definded

Definition 11.2.4 (Underfitting, Bias-term). *Given the partition $\mathcal{D} \to \{\mathcal{D}_{tr}, \mathcal{D}_{te}\}$, then the model $M(h, \mathcal{D})$ is said to underfit if:*

$$\lim_{|\mathcal{D}| \to \mathcal{D}_{max}} ret(M_{tr}, M_{te}) = \lim_{|\mathcal{D}| \to \mathcal{D}_{max}} \frac{\text{Bias}[h_{\mathcal{D}_{tr}}(x)]}{\text{Bias}[h_{\mathcal{D}_{te}}(x)]} = 0 \tag{11.18}$$

or, given an assumption:

$$ret(M_{te}, M_{tr}) \in \mathcal{O}(x^2) \tag{11.19}$$

by just changing the fractional term.

This *should be*, enough for our definition as of now. Note that, however, that bias term is really close to accuracy, in most sense. So it puts us at quite a postion - just how **variance** fits into the play?

11.3 Variance

In a somewhat acceptable and identical manner, we can still, loosely define variance. But the spirit of the variance term is simple. Informally, it is a measure of fluctuation of a learner around its central tendency (again, expectation value), where, the fluctuations result from different sampling of the training set [B. Neal, 2019]. So how much of this is true? We first go for the formal definition of such:

Definition 11.3.1 (Variance, I). *Given a model $M[h, S]$, where $h \in \mathcal{H}$ and S is the associated data, the **variance** of h to the true concept $c \supset \{S\}$ is defined as the measure of **fluctuations** of the hypothesis (learner) around the central tendency to the true concept:*

$$\text{Var}(h_S) = \mathop{\mathbb{E}}_{x \sim \mathcal{P}} \left[(h_S(x) - \mathbb{E}[h_S(x)])^2 \right] \tag{11.20}$$

This definition is weird - we haven't seen the squared term at all, even of which we have encountered the normalized version of bias. What if we do not use the term square after all? Furthermore, if we can think of bias as a form of standard deviation, we notice that we actuall define standard deviation, from the variance. So, what gives?

We actually turn to the general idea, and also, the general kind of *distinction* between two case. Specifically, itis the question of "normally, and now in learning theory - which **estimator** was used?", and "How many active estimator (and their classes) are there?". All of which, again, would be brought up again after this defining section is partially complete, that is.

In similar fashion to the bias term, we can also define the variance term in a more iterative manner, of such to fits the task of monitoring such terms, with increment model operation time.

First, notice that term now includes

$$(h_S(x) - \mathbb{E}[h_S(x)])^2$$

of which the moving term would be the second term concerned of the mean average of the hypothesis. In fact, we can move variance to a more generalized definition, that is,

Definition 11.3.2 (Variance, II)**.** *Given a model $M[h, S]$, where $h \in \mathcal{H}$ and S is the associated data, the **variance** of h to the true concept $c \supset \{S\}$ is defined as the measure of **fluctuations** of the hypothesis (learner) around the central tendency to the true concept:*

$$\text{Var}(h_S) = \mathop{\mathbb{E}}_{x \sim \mathcal{P}} d_M(h_S, \langle h_S \rangle) \tag{11.21}$$

where the notation just shift to $\langle h_S \rangle = \mathbb{E}(h_S)$. The iterative form then should be concerned of the main decomposition component, hence the equating term should be quite different.

Definition 11.3.3 (Variance, Iterative, I)**.** *Suppose a model with the form $M(h, S)$, where $S \subset \mathbb{R}^{n+m}$, and a **logging set** contaning previous iteration of the model in the procedure $\mathcal{T}(M, L)$, of such that $S \subset L$ and $L = \mathbb{R}^{m+p}$, where m is the index of the number of example, and p is the index of the number of ensemble (for each iteration in the past). Then, the **variance** at the t-th iteration is (assuming $p = t - 1$) is defined as*

$$\text{Var}(h_S, t) = \frac{1}{n} \sum_{j=1}^{n} \left[h_S(x_j) - \frac{1}{p} \left(\sum_{i=1}^{p} h_S(x_{i,j}) \right) \right]^2 \tag{11.22}$$

For $x_{i,j}$ is the ith iteration that the model is applied on such sample point, and jth is the index position of that sample point in the dataset.

The connection between variance and over/underfitting is not too apparent at first, but, let's us make an observation, because somehow, variance and bias still form the tradeoff termed *bias-variance tradeoff*, which indicate clearly that certain relation of the form must be included. . bias-variance tradeoff

On a side note, the variance term, again, can be combined, such as

$$\text{Var}(h(\mathcal{D})) = \frac{\text{Var}(h(\mathcal{D}_{tr})) + \text{Var}(h(\mathcal{D}_{te}))}{2} \tag{11.23}$$

for the usual case of two partitions.

11.4 Bias, Variance and fitting

After defining and investigating certain fixture of bias and variance term, we now then want to understand "what the heck we just defined and looked at". Simply put, the understanding of which all such terms are, including their relation to the simple error term.

In all, *bias* can be interpreted as the **capacity of expression** of the model, that is, the ability for the model to fits the description of a model. There are several way to do such, including **generative** and **simulation comparison**, but usually, the bias term is concerned of such. Hence, it is more or less directly related to *model complexity*.

On the other hand, *variance* concerns of the **stability** of the model, as it said to the fluctuation between iterations and versions of the hypothesis. If the model varies too much, it indicates certain overfitting behaviour, since the avilable context parameters are too much, and a conjecture is that the

more available the context parameters are, the more sensitive the entire model is, when operating under iterative training algorithm, or either, loosely call - jumping from this context frame (of data) to another context frame of different data.

Now, how does these two connect to the phenomena of overfitting and underfitting? First, we note on the nature of over/underfitting:

1. They relies on the comparison between partitions - for example, accuracy between the training set and test set. What about such notion, outward, and instead of 2-paritioned, we go for n-partitioned?

2. They relies on the proxy of accuracy, however, such proxy is not **expressive** enough, given the case where the factor of model complexity is mention, it is not clear in where the formulation takes into account of said factor. In fact, the notion of overfitting and underfitting is not generalized, and often being very case-specific for it to gain effects.

The bias term is the only one that, intuitively, go close in with overfitting and underfitting, since it directly measure between the exposed concept samples, and the hypothesis given of the model, the difference over the entirety of the data set. Variance, on the other hand, is a bit tricky - it measures the fluctuation between the hypothesis class itself. To how such fluctuation is interpreted is a problem, but the general consensus and status quo seems to indicate that the more stable the model is, the **better** is is from preventing itself from overfitting. However, if in the long-run, the stability is always at minimal, there are good chance that the model is underfitting - because it cannot do that much things to change its shape, or representation to fits what is is observing, then the statbility would stay low.

This interpretation leads us to inspect the epoch evolution - naturally - of the bias and variance term, to figure out how to interpreted of such. To do such, we need a *time-series* and iterative formula for both term, hence why we made those formulation above. The next step, though, would be to investigate aforementioned terms, with a specific model, specific loss function, specific accuracy metric and threshold, and analyse the situation of it. And, per previous researchers' works, they indicate and point out a conjecture - exactly the **bias-variance tradeoff**. This term would be important for this whole chapter.

Another interpretation of the bias and variance is in the form of **statistical bias** and **statistical variance** - mostly concerned of statistics. The bias of a learning algorithm A for a given learning problem and a fixed size m for training sets, is the persistent or *systematic error* that the learning algorithm is expected to make when trained on training sets of size m.

Statistical bias captures the idea of a *systematic error* for a given sample size. For example, if the true function is a sine wave $f(x) = \sin(x)$, and the learning algorithm fits lines $f(x) = ax + b$, then there will be systematic error at each bump in the sine wave, no matter what fit there can be. The statistical variance is then defined similarly, as we thought, the expected value of the squared difference between any particular hypothesis f_S and the averaged hypothesis, taken with respect to all training samples S of size m. The variance captures random variation in the algorithm from one training set S

to another. This variation can result from variation in the training sample, from random noise ϵ of which is irreducible, or from random behaviour in the learning algorithm ifself, for example, the random weight initialization.

Of this setting, **machine learning bias** (ML bias) can be described in terms of absolute bias (certain hypothesis are entirely eliminated from the hypothesis space) and relative bias (certain hypothesis are preferred over others). This type of bias for a learning algorithm - if can be formulated explicitly - provides a specification for the desired behaviour of the algorithm and clarifies the design and implementation of machine learning algorithms.

On any particular problem, an absolute bias can be characterized as appropriate or not. Here, notice that the definition indicates it as **controlling directive operators**, such that the bias itself is mutable, and acts on the hypothesis class itself. Statistical bias is more an over-the-top measure of empirical mean, not as a directive approach.

11.5 'Alternative' to bias and variance

Are there any particular alternative, or representation differs from the norm of such it was made to be? One candidate of theoretical exhibition we would call as closely resemble said bias and variance, as expressed through a decomposition (more likely to be a tradeoff), is through the *approximation-estimation error.*

Given a set of candidate classifiers $\mathcal{H}$, we can decompose the generalization risk $R(\hat{h}_n)$ as:

$$R(\hat{h}_n) = R(\hat{h}_n) - \inf_{h \in \mathcal{H}} R(h) + \inf_{h \in \mathcal{H}} R(h) - R(h^*) + R(h^*) \tag{11.24}$$

where the first two term is the estimation error,

$$Est(\hat{h}_n) = R(\hat{h}_n) - \inf_{h \in \mathcal{H}} R(h) \tag{11.25}$$

and the second one is the approximation error,

$$Approx(\hat{h}_n) = \inf_{h \in \mathcal{H}} R(h) - R(h^*) \tag{11.26}$$

The first part compares the performance of the classifier $\hat{h}_n$ against the best possible classifier within $\mathcal{H}$, while the second one is a statement about the power of the class itself - how close we can get to the Bayes classifier if we stay within $\mathcal{H}$. We can reduce the estimation error by making $\mathcal{H}$ smaller, but then the approximation error increases, as per older theory. This makes up the typical situation as above, similar to bias and variance, and to the tradeoff we soon engage in. In fact, approximation-estimation relations is closely related to bias-variance itself, so it is good to have it in mind as the Bayes classifier is inherently the goal of learning algorithm.

11.6 The bias-variance tradeoff

11.7 History of bias-variance tradeoff

Per [Bradly Neal, 2019], "a universal bias-variance tradeoff, without qualitifications, is a mere hypothesis". The history of such formalization to a tradeoff has much more to do with machine learning, and a very long standing intuition in mind.

Bibliography

[] *(PDF) Intelligence: A Brief History*. URL: https://www.researchgate.net/publication/242083221_Intelligence_A_Brief_History (visited on 07/17/2024).

[] *Daniel Estrada, Rethinking machines: artificial intelligence beyond the philosophy of mind - PhilPapers*. URL: https://philpapers.org/rec/ESTRMA (visited on 07/17/2024).

[Aso18] Fatai Asodun. "THE PROBLEMATIC SEARCH FOR THE NATURE OF INTELLIGENCE: PHILOSOPHICAL REACTIONS AND PROJECTIONS". en. In: 4 (2018).

[Bel+19] Mikhail Belkin et al. "Reconciling modern machine-learning practice and the classical bias–variance trade-off". In: *Proceedings of the National Academy of Sciences* 116.32 (July 2019), pp. 15849–15854. ISSN: 1091-6490. DOI: 10.1073/pnas.1903070116. URL: http://dx.doi.org/10.1073/pnas.1903070116.

[BG24] Selmer Bringsjord and Naveen Sundar Govindarajulu. "Artificial Intelligence". In: *The Stanford Encyclopedia of Philosophy*. Ed. by Edward N. Zalta and Uri Nodelman. Summer 2024. Metaphysics Research Lab, Stanford University, 2024. URL: https://plato.stanford.edu/archives/sum2024/entries/artificial-intelligence/ (visited on 07/17/2024).

[Bir23] Dresp-Langley Birgitta. "Artificial Consciousness: Misconception(s) of a Self-Fulfilling Prophecy". In: *Queios*. Ed. by Birgitta Dresp. 2023.

[CD21] Herman Cappelen and Josh Dever. *Making AI Intelligible: Philosophical Foundations*. arXiv:2406.08134 [cs]. Apr. 2021. DOI: 10.1093/oso/9780192894724.001.0001. URL: http://arxiv.org/abs/2406.08134 (visited on 07/17/2024).

[Con16] Vincent Conitzer. *Philosophy in the Face of Artificial Intelligence*. arXiv:1605.06048 [cs]. May 2016. DOI: 10.48550/arXiv.1605.06048. URL: http://arxiv.org/abs/1605.06048 (visited on 07/17/2024).

[Cos24] Christian Coseru. "Artificial Minds and the Dilemma of Personal Identity". In: *Philosophy East and West* 74.2 (2024), pp. 281–297. DOI: 10.1353/pew.2024.a925193.

[GBD92] Stuart Geman, Elie Bienenstock, and René Doursat. "Neural Networks and the Bias/Variance Dilemma". In: *Neural Computation* 4.1 (1992), pp. 1–58. DOI: 10.1162/neco.1992.4.1.1.

[GS24] Gilles E. Gignac and Eva T. Szodorai. "Defining intelligence: Bridging the gap between human and artificial perspectives". In: *Intelligence* 104 (May 2024), p. 101832. ISSN: 0160-2896. DOI: 10.1016/j.intell.2024.101832. URL: https://www.sciencedirect.com/science/article/pii/S0160289624000266 (visited on 07/17/2024).

[Hoi+] Fabian Hoitsma et al. "Mitigating Implicit and Explicit Bias in Structured Data Without Sacrificing Accuracy in Pattern Classification". In: *AI and Society* (), pp. 1–20. DOI: 10.1007/s00146-024-02003-0.

[Kan17] Varun Kanade. "Consistent Learner and Occam's Razor". In: *Lecture notes in Computational Learning Theory* (Nov. 12, 2017).

[Kas21] Atoosa Kasirzadeh. *Reasons, Values, Stakeholders: A Philosophical Framework for Explainable Artificial Intelligence*. en. arXiv:2103.00752 [cs]. Feb. 2021. URL: http://arxiv.org/abs/2103.00752 (visited on 07/17/2024).

[Ken+] Ryan Kennedy et al. "Net Versus Relative Impacts in Public Policy Automation: A Conjoint Analysis of Attitudes of Black Americans". In: *AI and Society* (), pp. 1–13. DOI: 10.1007/s00146-024-01975-3.

[Lan00] Peter Lanz. "The Concept of Intelligence in Psychology and Philosophy". en. In: *Prerational Intelligence: Adaptive Behavior and Intelligent Systems Without Symbols and Logic, Volume 1, Volume 2 Prerational Intelligence: Interdisciplinary Perspectives on the Behavior of Natural and Artificial Systems, Volume 3*. Ed. by Holk Cruse, Jeffrey Dean, and Helge Ritter. Dordrecht: Springer Netherlands, 2000, pp. 19–30. ISBN: 978-94-010-0870-9. DOI: 10.1007/978-94-010-0870-9_3. URL: https://doi.org/10.1007/978-94-010-0870-9_3 (visited on 07/17/2024).

[LH07] Shane Legg and Marcus Hutter. *A Collection of Definitions of Intelligence*. en. arXiv:0706.3639 [cs]. June 2007. URL: http://arxiv.org/abs/0706.3639 (visited on 07/17/2024).

[Liu23] Masahito Ueda Liu Ziyin. "Zeroth, first, and second-order phase transitions in deep neural networks". In: (Dec. 2023). DOI: 10.1103/PhysRevResearch.5.043243. URL: https://journals.aps.org/prresearch/abstract/10.1103/PhysRevResearch.5.043243.

[LM] Harvey Lederman and Kyle Mahowald. "Are Language Models More Like Libraries or Like Librarians? Bibliotechnism, the Novel Reference Problem, and the Attitudes of Llms". In: *Transactions of the Association for Computational Linguistics* ().

[Mon+20] Dagmar Monett et al. "Special Issue "On Defining Artificial Intelligence"—Commentaries and Author's Response". en. In: *Journal of Artificial General Intelligence* 11.2 (Feb. 2020), pp. 1–100. ISSN: 1946-0163. DOI: 10.2478/jagi-2020-0003. URL: https://www.sciendo.com/article/10.2478/jagi-2020-0003 (visited on 07/17/2024).

[Mor] Masahiro Morioka. "Artificial Intelligence and Contemporary Philosophy". en. In: ().

[MRT18] Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. *Foundations of Machine Learning*. 2nd ed. Adaptive Computation and Machine Learning. Cambridge, MA: MIT Press, 2018. 504 pp. ISBN: 978-0-262-03940-6.

[MT23] Piero Molino and Jacopo Tagliabue. *Witgenstein's influence on artificial intelligence*. en. arXiv:2302.01570 [cs]. Feb. 2023. URL: http://arxiv.org/abs/2302.01570 (visited on 07/17/2024).

[PSC21] Nicolas Palanca-Castan, Beatriz Sánchez Tajadura, and Rodrigo Cofré. "Towards an interdisciplinary framework about intelligence". In: *Heliyon* 7.2 (Feb. 2021), e06268. ISSN: 2405-8440. DOI: 10.1016/j.heliyon.2021.e06268. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7902546/ (visited on 07/17/2024).

[Sca09] Franco Scarselli. "The graph neural network model". In: *IEEE Transactions on Neural Networks* (2009). URL: https://ieeexplore.ieee.org/document/4700287.

[SIT17] SITNFlash. *The History of Artificial Intelligence*. en-US. Aug. 2017. URL: https://sitn.hms.harvard.edu/flash/2017/history-artificial-intelligence/ (visited on 07/17/2024).

[SK21] Michael T. Stuart and Markus Https://Orcidorg Kneer. "Guilty Artificial Minds: Folk Attributions of Mens Rea and Culpability to Artificially Intelligent Agents". In: *Proceedings of the ACM on Human-Computer Interaction* 5.CSCW2 (2021). Publisher: ACM Digital library.

[Sri19] Suchitra Srivastava. "Evolution Of Concept Of Intelligence". In: 6 (Oct. 2019), pp. 56–69.

[YL22] Yue Yu and Pavel Loskot. *Polynomial Distributions and Transformations*. 2022. arXiv: 2212.04865 [cs.IT]. URL: https://arxiv.org/abs/2212.04865.

[Zha23] Youheng Zhang. "A Historical Interaction between Artificial Intelligence and Philosophy". en. In: *Theory of Science* 1.1 (Oct. 2023). ISSN: 1804-6347, 1210-0250. DOI: 10.46938/tv.2023.579. URL: https://teorievedy.flu.cas.cz/index.php/tv/article/view/579 (visited on 07/17/2024).