AI Applications Lecture 19
Learning Theory and Generalization Gap

SUZUKI, Atsushi
Jing WANG

2025-11-21

Contents

1 Introduction

Machine learning using parametric functions, including generative AI, is the problem of determining good parameters to solve a given task based on training data. For this to be possible, information about the task must be sufficiently provided through the training data sequence. A practical problem, then, is specifically how much training data should be provided.

In principle, the greater the expressive power of the parametric function, the larger the number of training data required tends to be. Conversely, if the number of parameters to be changed is small and the expressive power of the parametric function is low, as in PEFT, the required number of training data tends to be small. In this lecture, we will theoretically examine the relationship between the expressive power of parametric functions and the number of training data required for learning.

Such relationships are typical themes of **learning theory**, which quantitatively treats the relationship between the number of parameters/size of the function space, the number of training data, and performance in the real environment from the perspectives of probability theory, statistics, and information theory. In this lecture, we introduce **expected risk** and **empirical risk** as indicators for measuring generalization performance, and discuss the **generalization gap**, which is the difference between the two. Furthermore, we introduce **Rademacher complexity** and **covering number** as measures representing the ”size” of the function space, and outline the relationship between these and the generalization gap. Finally, using **Fano’s inequality**, we provide fundamental performance limits that do not depend on any specific learning algorithm or model.

1.1 Learning Outcomes

The Learning Outcomes of this lecture are explicitly stated below. By the end of this lecture, students should be able to:

- Explain the difference between empirical risk and expected risk.
- Explain the trade-off between empirical risk and the generalization gap caused by model size in machine learning.
- Explain that when the set of candidate hypotheses in machine learning is large, there exist fundamental performance limits independent of any algorithm or model.

In the following, we restate the necessary mathematical preparation to achieve these, formulate generalization loss and generalization gap, and describe their quantitative evaluation and limits.

2 Preparation: Restatement of Mathematical Notations

- **Definition:**

  - $(\mathrm{LHS}) \coloneqq (\mathrm{RHS})$: Indicates that the left-hand side is defined by the right-hand side. For example, $a \coloneqq b$ indicates that a is defined by b.

- **Set:**
  - Sets are often denoted by uppercase calligraphic letters. Example: $\mathcal{A}$.
  - $x \in \mathcal{A}$: Indicates that element x belongs to set $\mathcal{A}$.
  - $\{\}$: Empty set.
  - $\{a, b, c\}$: Set consisting of elements a, b, c (extensional notation of a set).
  - $\{x \in \mathcal{A} \mid P(x)\}$: Set consisting of elements of set $\mathcal{A}$ for which proposition $P(x)$ is true (intensional notation of a set).
  - $\mathbb{R}$: Set of all real numbers.
  - $\mathbb{R}_{>0}$: Set of all positive real numbers.
  - $\mathbb{R}_{\geq 0}$: Set of all non-negative real numbers.
  - $\mathbb{Z}$: Set of all integers.
  - $\mathbb{Z}_{>0}$: Set of all positive integers.
  - $\mathbb{Z}_{\geq 0}$: Set of all non-negative integers.
  - $[1, k]_{\mathbb{Z}} \coloneqq \{1, 2, \ldots, k\}$: Set of integers from 1 to k for a positive integer k.

- **Function:**
  - $f : \mathcal{X} \to \mathcal{Y}$: Indicates that function f is a map that takes an element of set $\mathcal{X}$ as input and outputs an element of set $\mathcal{Y}$.
  - $y = f(x)$: Indicates that the output is $y \in \mathcal{Y}$ when $x \in \mathcal{X}$ is input to function f.

- **Vector:**
  - In this course, a vector refers to a column of numbers.
  - Vectors are denoted by bold italic lowercase letters. Example: $\boldsymbol{v}$.
  - $\boldsymbol{v} \in \mathbb{R}^n$: Indicates that vector $\boldsymbol{v}$ is an n-dimensional real number vector.
  - The i-th element of vector $\boldsymbol{v}$ is denoted as v_i.

$$\boldsymbol{v} = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix}. \tag{1}$$

- **Matrix:**
  - Matrices are denoted by bold italic uppercase letters. Example: $\boldsymbol{A}$.

- $\boldsymbol{A} \in \mathbb{R}^{m,n}$: Indicates that matrix $\boldsymbol{A}$ is a real matrix with m rows and n columns.
- The element at the i-th row and j-th column of matrix $\boldsymbol{A}$ is denoted as $a_{i,j}$.

$$\boldsymbol{A} = \begin{bmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,n} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m,1} & a_{m,2} & \cdots & a_{m,n} \end{bmatrix}. \tag{2}$$

- The transpose of matrix $\boldsymbol{A}$ is denoted as $\boldsymbol{A}^\top$. If $\boldsymbol{A} \in \mathbb{R}^{m,n}$, then $\boldsymbol{A}^\top \in \mathbb{R}^{n,m}$, and

$$\boldsymbol{A}^\top = \begin{bmatrix} a_{1,1} & a_{2,1} & \cdots & a_{m,1} \\ a_{1,2} & a_{2,2} & \cdots & a_{m,2} \\ \vdots & \vdots & \ddots & \vdots \\ a_{1,n} & a_{2,n} & \cdots & a_{m,n} \end{bmatrix}. \tag{3}$$

- A vector is also a matrix with 1 column, and its transpose can also be defined.

$$\boldsymbol{v}^\top = \begin{bmatrix} v_1 & v_2 & \cdots & v_n \end{bmatrix} \in \mathbb{R}^{1,n}. \tag{4}$$

- **Tensor:**
  - In this lecture, the term tensor simply refers to a multidimensional array. A vector can be regarded as a 1st-order tensor, and a matrix as a 2nd-order tensor. Tensors of 3rd order or higher are denoted by underlined bold italic uppercase letters, such as $\underline{\boldsymbol{A}}$.
  - Students who have already learned about abstract tensors in mathematics or physics may find it resistant to call simple multidimensional arrays tensors. There is (some) consistency in terminology if one assumes that the basis is always fixed to the standard basis and the tensor in the mathematical sense is identified with its component representation (which becomes a multidimensional array).

3 Generalization Loss and Generalization Gap

In this section, we formulate the learning problem abstractly and rigorously define expected risk, empirical risk, and the generalization gap.

3.1 Abstract Formulation of Learning Problems

First, we provide a general framework for learning problems consisting of three elements: data, function space, and loss function.

Definition 3.1 (General Learning Problem Framework**).**

- Let the data space be $\mathcal{Z}$. An element $z \in \mathcal{Z}$ of $\mathcal{Z}$ is a single data point.

- Let the function space (hypothesis set) be $\mathcal{F}$. $\mathcal{F}$ is a set of collected models (hypotheses) f that are candidates for execution.

- Let the loss function be
$$\ell : \mathcal{F} \times \mathcal{Z} \to \mathbb{R}. \tag{5}$$
For $f \in \mathcal{F}$ and $z \in \mathcal{Z}$, $\ell(f, z)$ is a scalar measuring "how bad f is for z".

Remark 3.1. In Definition 3.1, the first argument of ℓ is the function f itself, and the second argument is the data point z. In practice, z is often an input-output pair (x, y), and $\ell(f, z)$ is a function measuring the distance between $f(x)$ and y. However, in order to treat it more abstractly without limiting it to such a structure, we first formulate it as a function for elements of $\mathcal{Z}$.

Next, we confirm that the framework of a typical prediction problem is included as a special case of the abstract formulation above.

Example 3.1 (Prediction Problem Framework**).** Let the data space be $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$, where the input space is $\mathcal{X}$ and the output space is $\mathcal{Y}$. A single data point is $z = (x, y) \in \mathcal{X} \times \mathcal{Y}$.
Let the predictive function space be $\mathcal{F} \subset \mathcal{Y}^{\mathcal{X}}$. That is, $\mathcal{F}$ is a set of functions f that take an input $x \in \mathcal{X}$ and return a prediction $\hat{y} = f(x) \in \mathcal{Y}$.
Write the loss function regarding prediction $\ell_{\mathrm{pred}} : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ as

$$\ell_{\mathrm{pred}}(\hat{y}, y). \tag{6}$$

For example, in the case of regression, it is the squared loss $\ell_{\mathrm{pred}}(\hat{y}, y) = (\hat{y} - y)^2$, and in the case of classification, it is the 0 − 1 loss $\ell_{\mathrm{pred}}(\hat{y}, y) = \mathbf{1}\{\hat{y} \neq y\}$, etc. Here, $\mathbf{1}\{\cdot\}$ is an indicator function that returns 1 if the event is true and 0 if it is false.
At this time, if we define the overall loss function $\ell : \mathcal{F} \times \mathcal{Z} \to \mathbb{R}$ as

$$\ell(f, (x, y)) \coloneqq \ell_{\mathrm{pred}}(f(x), y), \tag{7}$$

it becomes a special case within the framework of Definition 3.1.
On a computer, the function space $\mathcal{F}$ is often represented as a **parametric family**. That is, introducing a parameter space $\mathit{\Theta} \subset \mathbb{R}^d$, we prepare a map

$$f_{(\cdot)} : \mathit{\Theta} \to \mathcal{Y}^{\mathcal{X}}, \qquad \boldsymbol{\theta} \mapsto f_{\boldsymbol{\theta}}. \tag{8}$$

At this time, the function space can be written as

$$\mathcal{F} = f_{\mathit{\Theta}} \coloneqq \{f_{\boldsymbol{\theta}} \mid \boldsymbol{\theta} \in \mathit{\Theta}\}. \tag{9}$$

3.2 Expected Risk and Empirical Risk

Next, we define the expected risk, which is the performance indicator in the real environment, and the empirical risk, which is calculable on a finite number of training data. Hereinafter, the notation of probability distributions and expected values follows standard learning theory textbooks [1].

Definition 3.2 (Expected Risk)**.** Let the probability distribution on the data space $\mathcal{Z}$ be P. That is, $Z \sim \mathrm{P}$ is a random data point in the environment. For loss function ℓ and function $f \in \mathcal{F}$, we define the **expected risk** as

$$\mathrm{ExpRisk}_{\ell,\mathrm{P}}(f) \coloneqq \mathbb{E}_{Z\sim\mathrm{P}}\left[\ell(f, Z)\right]. \tag{10}$$

Remark 3.2. P is interpreted as a probability distribution representing the data-generating process in the real environment. Therefore, $\mathrm{ExpRisk}_{\ell,\mathrm{P}}(f)$ is ”the expected loss when using f in the real environment”, and is the quantity we ultimately want to minimize. Formally written, ideally we want to solve

$$\underset{f\in\mathcal{F}}{\text{Minimize}}\ \mathrm{ExpRisk}_{\ell,\mathrm{P}}(f). \tag{11}$$

When described by a parametric family $f_{\boldsymbol{\theta}}$, this can be written as

$$\underset{\boldsymbol{\theta}\in\Theta}{\text{Minimize}}\ \mathrm{ExpRisk}_{\ell,\mathrm{P}}(f_{\boldsymbol{\theta}}). \tag{12}$$

However, in machine learning including actual generative AI, we almost never know the explicit form of P in advance. Therefore, we minimize the empirical risk instead of the expected risk using a finite training data sequence actually observed.

Definition 3.3 (Empirical Risk)**.** Let the data sequence (training dataset) be

$$\boldsymbol{z} = (z_1, z_2, \ldots, z_m) \in \mathcal{Z}^m. \tag{13}$$

At this time, we define the **empirical risk** as

$$\mathrm{EmpRisk}_{\ell,\boldsymbol{z}}(f) \coloneqq \frac{1}{m}\sum_{i=1}^{m} \ell(f, z_i). \tag{14}$$

Remark 3.3. The empirical risk $\mathrm{EmpRisk}_{\ell,\boldsymbol{z}}(f)$ is calculable on the training data, and its gradient

$$\nabla_{\boldsymbol{\theta}}\mathrm{EmpRisk}_{\ell,\boldsymbol{z}}(f_{\boldsymbol{\theta}}) \tag{15}$$

can also be efficiently calculated by backpropagation if it is a neural network. Therefore, actual learning algorithms often solve the minimization problem of empirical risk instead of expected risk:

$$\underset{\boldsymbol{\theta}\in\Theta}{\text{Minimize}}\ \mathrm{EmpRisk}_{\ell,\boldsymbol{z}}(f_{\boldsymbol{\theta}}). \tag{16}$$

This framework is called **Empirical Risk Minimization** [1].

3.3 Optimization Algorithm and Generalization Gap

There exists a difference between the expected risk and the empirical risk, and this is the **generalization gap**. To discuss the generalization gap, we formally define the optimization algorithm.

Definition 3.4 (Learning Algorithm)**.** Let $\mathcal{Z}^* \coloneqq \bigcup_{m=0}^{\infty} \mathcal{Z}^m$ be the set of data sequences of all lengths. We define the **learning algorithm** as

$$\mathfrak{A} : \mathcal{Z}^* \to \mathcal{F}. \tag{17}$$

$\mathfrak{A}(\boldsymbol{z})$ is a map that takes a data sequence $\boldsymbol{z}$ as input and returns a single function f from the hypothesis set $\mathcal{F}$.

When a training data sequence $\boldsymbol{z}$ is given, $\mathfrak{A}(\boldsymbol{z})$ is a model selected based on some objective (usually reduction of empirical risk). At this time, what we are interested in is the expected risk of $\mathfrak{A}(\boldsymbol{z})$ in the real environment:

$$\mathrm{ExpRisk}_{\ell,\mathrm{P}}\left(\mathfrak{A}(\boldsymbol{z})\right). \tag{18}$$

On the other hand, the algorithm $\mathfrak{A}$ is usually designed to minimize the empirical risk:

$$\mathrm{EmpRisk}_{\ell,\boldsymbol{z}}\left(\mathfrak{A}(\boldsymbol{z})\right). \tag{19}$$

Definition 3.5 (Generalization Gap)**.** Assume that data generating distribution P, loss function ℓ, data sequence $\boldsymbol{z}$, and learning algorithm $\mathfrak{A}$ are given. At this time, we define the **generalization gap** as

$$\mathrm{GenGap}_{\ell,\mathrm{P},\boldsymbol{z}}(\mathfrak{A}) \coloneqq \mathrm{ExpRisk}_{\ell,\mathrm{P}}\left(\mathfrak{A}(\boldsymbol{z})\right) - \mathrm{EmpRisk}_{\ell,\boldsymbol{z}}\left(\mathfrak{A}(\boldsymbol{z})\right). \tag{20}$$

Remark 3.4. Using equation (20), we can decompose as follows:

$$\begin{aligned}\mathrm{ExpRisk}_{\ell,\mathrm{P}}\left(\mathfrak{A}(\boldsymbol{z})\right) &= \mathrm{EmpRisk}_{\ell,\boldsymbol{z}}\left(\mathfrak{A}(\boldsymbol{z})\right) + \Big(\mathrm{ExpRisk}_{\ell,\mathrm{P}}\left(\mathfrak{A}(\boldsymbol{z})\right) - \mathrm{EmpRisk}_{\ell,\boldsymbol{z}}\left(\mathfrak{A}(\boldsymbol{z})\right)\Big) && (21)\\ &= \mathrm{EmpRisk}_{\ell,\boldsymbol{z}}\left(\mathfrak{A}(\boldsymbol{z})\right) + \mathrm{GenGap}_{\ell,\mathrm{P},\boldsymbol{z}}(\mathfrak{A}). && (22)\end{aligned}$$

The first term on the right-hand side is observable, and the second term is the generalization gap. If the generalization gap is small, it can be said that it ”generalizes well” in the sense that the performance on the training data is close to the performance in the real environment as it is.

3.4 Qualitative Discussion of Generalization Gap

Qualitatively, the following intuition holds:

- If there is a sufficient amount of training data to approximate the data generating distribution P, the empirical risk $\mathrm{EmpRisk}_{\ell,\boldsymbol{z}}(f)$ becomes a good approximation of the expected risk $\mathrm{ExpRisk}_{\ell,\mathrm{P}}(f)$, and the generalization gap tends to be small.
- Even with the same amount of data, if the function space $\mathcal{F}$ is large (high expressive power), it means the search range of the optimization algorithm $\mathfrak{A}$ is wide, so $\mathfrak{A}(\boldsymbol{z})$ selected dependent only on training data is likely to deviate from the original purpose of expected risk minimization. As a result, the generalization gap tends to be large.

For the above reasons, given the size of the function space, we want to quantitatively discuss:

- ”How much data is needed to suppress the generalization gap below a certain threshold?”

In the next section, we introduce Rademacher complexity and covering number for this purpose, and provide an upper bound evaluation of the generalization gap.

4 Quantitative Evaluation of Generalization Gap

In this section, we quantitatively evaluate the generalization gap when the function space $\mathcal{F}$ is given, using **Rademacher complexity** and **covering number**.

4.1 Definition of Rademacher Complexity

First, we define Rademacher complexity. This is a measure that gauges how ”flexibly” a function space can adapt to a random sign sequence [1].

Definition 4.1 (Rademacher Random Variables)**.** Fix $m \in \mathbb{Z}_{>0}$. **Rademacher random variables** are a sequence of independent and identically distributed $\{-1, +1\}$-valued random variables

$$\boldsymbol{\sigma} = (\sigma_1, \ldots, \sigma_m) \tag{23}$$

where each satisfies

$$\mathbb{P}(\sigma_i = 1) = \mathbb{P}(\sigma_i = -1) = \frac{1}{2}. \tag{24}$$

Definition 4.2 (Empirical Rademacher Complexity of Function Class)**.** Assume a data sequence $\boldsymbol{z} = (z_1, \ldots, z_m) \in \mathcal{Z}^m$ and a real-valued function class

$$\mathcal{G} \subset \mathbb{R}^{\mathcal{Z}} \tag{25}$$

are given. At this time, we define the **empirical Rademacher complexity** as

$$\widehat{\mathfrak{R}}_{z}(\mathcal{G}) := \mathbb{E}_{\boldsymbol{\sigma}}\left[\sup_{g\in\mathcal{G}} \frac{1}{m}\sum_{i=1}^{m} \sigma_i g(z_i)\right]. \tag{26}$$

Here, the expectation is with respect to the Rademacher random variable sequence $\boldsymbol{\sigma}$ in Definition 4.1.

Definition 4.3 (Expected Rademacher Complexity)**.** Let the data generating distribution be P, and sample m data sequences $\boldsymbol{Z} = (Z_1, \ldots, Z_m) \sim \mathrm{P}^m$ independently and identically. At this time, we define the **expected Rademacher complexity** as

$$\mathfrak{R}_{\mathrm{P},m}(\mathcal{G}) := \mathbb{E}_{\boldsymbol{Z}\sim\mathrm{P}^m}\left[\widehat{\mathfrak{R}}_{\boldsymbol{Z}}(\mathcal{G})\right]. \tag{27}$$

We are interested in the Rademacher complexity for the loss function ℓ and the hypothesis class $\mathcal{F}$. Therefore, we consider the function class composed of the loss function and the hypothesis class:

$$\ell \circ \mathcal{F} := \{\, g_f : \mathcal{Z} \to \mathbb{R} \mid f \in \mathcal{F},\; g_f(z) := \ell(f, z)\,\}. \tag{28}$$

Definition 4.4 (Rademacher Complexity of Hypothesis Class with Loss)**.** For loss function ℓ and hypothesis class $\mathcal{F}$, we define the **empirical Rademacher complexity** on data sequence z as

$$\mathrm{EmpRad}_{\ell,z}(\mathcal{F}) := \widehat{\mathfrak{R}}_{z}(\ell \circ \mathcal{F}), \tag{29}$$

and the **expected Rademacher complexity** on data length m as

$$\mathrm{ExpRad}_{\ell,\mathrm{P},m}(\mathcal{F}) := \mathfrak{R}_{\mathrm{P},m}(\ell \circ \mathcal{F}). \tag{30}$$

Remark 4.1. Rademacher complexity measures the ability to select a function g from the function class $\mathcal{G}$ that maximizes the inner product

$$\frac{1}{m}\sum_{i=1}^{m} \sigma_i g(z_i) \tag{31}$$

with the random label, when regarding the random sequence $\boldsymbol{\sigma}$ as a ”random label”. In other words, it is an indicator of ”how well random noise can be fitted”, representing the flexibility (complexity) of the function class.
Also, for the inclusion relationship of function classes $\mathcal{G} \subset \mathcal{G}'$,

$$\mathfrak{R}_{\mathrm{P},m}(\mathcal{G}) \leq \mathfrak{R}_{\mathrm{P},m}(\mathcal{G}') \tag{32}$$

holds [1, Proposition 26.9]. Therefore, $\mathrm{ExpRad}_{\ell,\mathrm{P},m}(\mathcal{F})$ is, as the name suggests, a measure of the ”size” or **complexity** of the function space.

4.2 Evaluation of Generalization Gap by Rademacher Complexity

Using Rademacher complexity, the generalization gap can be bounded from above with high probability. The following theorem is based on Shalev-Shwartz and Ben-David's textbook [1, Theorem 26.5].

Theorem 4.1 (Generalization Gap Upper Bound by Rademacher Complexity)**.** Let the data generating distribution be P, and sample $\boldsymbol{Z} = (Z_1, \ldots, Z_m) \sim \mathrm{P}^m$ independently and identically. For loss function ℓ and hypothesis class $\mathcal{F}$, assume that

$$\ell(f, z) \in [0, 1] \tag{33}$$

holds for all $f \in \mathcal{F}$ and $z \in \mathcal{Z}$ (i.e., the loss is bounded in the interval $[0, 1]$).
At this time, for any $\delta \in (0, 1)$, with probability at least $1 - \delta$,

$$\sup_{f \in \mathcal{F}} \Big(\mathrm{ExpRisk}_{\ell, \mathrm{P}}(f) - \mathrm{EmpRisk}_{\ell, \boldsymbol{Z}}(f) \Big) \le 2\, \mathrm{EmpRad}_{\ell, \boldsymbol{Z}}(\mathcal{F}) + 3\sqrt{\frac{\log(2/\delta)}{2m}} \tag{34}$$

holds.

Remark 4.2. Theorem 4.1 indicates the following points:

- The left-hand side is the maximum value of the generalization gap

  $$\mathrm{ExpRisk}_{\ell, \mathrm{P}}(f) - \mathrm{EmpRisk}_{\ell, \boldsymbol{Z}}(f) \tag{35}$$

  for all functions $f \in \mathcal{F}$. Therefore, whatever $f = \mathfrak{A}(\boldsymbol{Z})$ any learning algorithm $\mathfrak{A}$ chooses, its generalization gap is bounded from above by the right-hand side.

- The first term on the right-hand side $\mathrm{EmpRad}_{\ell, \boldsymbol{Z}}(\mathcal{F})$ is the complexity of the function class, and the second term is a term dependent only on the sample size m and confidence δ. Therefore, **the larger the function class** (the larger the Rademacher complexity), the larger the upper bound of the generalization gap tends to be.

- On the other hand, to reduce the empirical risk $\mathrm{EmpRisk}_{\ell, \boldsymbol{Z}}(f)$, it is often advantageous to allow a function class with higher ”expressive power”. That is,

  - If the function class $\mathcal{F}$ is enlarged, the minimum empirical risk decreases, but the upper bound of the generalization gap increases.
  - If the function class $\mathcal{F}$ is made smaller, the upper bound of the generalization gap decreases, but a function close to the optimal value of the expected risk might be outside $\mathcal{F}$ in the first place.

- Therefore, to **reduce both empirical risk and generalization gap**, it is necessary to appropriately adjust the size of the function space (model capacity), and there exists a trade-off here. This provides the theoretical background when comparing large-scale models with many parameters and models with limited effective degrees of freedom like PEFT.

4.3 Covering Number and Dudley Integral

Directly evaluating Rademacher complexity itself is often difficult. Therefore, we introduce the **covering number** as another measure of the ”size” of the function class, and use this to evaluate Rademacher complexity [2, 3].

Definition 4.5 (Covering Number)**.** Let the universal set be $\mathcal{S}$, and its subset be $\mathcal{A} \subset \mathcal{S}$. Let the distance (or pseudometric) on $\mathcal{S}$ be

$$d : \mathcal{S} \times \mathcal{S} \to \mathbb{R}_{\geq 0}. \tag{36}$$

For $\epsilon \geq 0$, an **ϵ-cover** of $\mathcal{A}$ is a finite set $C \subset \mathcal{S}$ such that

$$\forall a \in \mathcal{A} \; \exists c \in C \text{ such that } d(a, c) \leq \epsilon \tag{37}$$

is satisfied. That is, each point in $\mathcal{A}$ is within distance ϵ from some point in C.
We define the ϵ-covering number of $\mathcal{A}$ as

$$\mathrm{CN}_{\mathcal{A},d}(\epsilon) := \min\{\, |C| \mid C \subset \mathcal{S} \text{ is an } \epsilon\text{-cover of } \mathcal{A} \,\}. \tag{38}$$

Remark 4.3. The covering number $\mathrm{CN}_{\mathcal{A},d}(\epsilon)$ represents the ”size” of set $\mathcal{A}$ at distance scale ϵ. Intuitively, it counts ”how many balls of radius ϵ are needed to cover $\mathcal{A}$”.
In particular, if $\mathcal{A} \subset \mathcal{A}' \subset \mathcal{S}$, then for any $\epsilon \geq 0$,

$$\mathrm{CN}_{\mathcal{A},d}(\epsilon) \leq \mathrm{CN}_{\mathcal{A}',d}(\epsilon) \tag{39}$$

holds. This is obvious from the fact that the number of balls needed to cover $\mathcal{A}'$ must be greater than or equal to the number needed to cover its subset $\mathcal{A}$. Therefore, the covering number is also a measure of the complexity (size) of the set.

Next, we describe the **Dudley entropy integral** type evaluation which evaluates Rademacher complexity from the covering number. The following theorem is an example of a Dudley-type evaluation for Rademacher processes, based on textbooks on concentration inequalities such as Boucheron, Lugosi, Massart [3, Theorem 13.2] (constants may vary slightly depending on the literature).

Theorem 4.2 (Evaluation of Rademacher Complexity by Dudley Integral)**.** Consider a data sequence $z = (z_1, \dots, z_m) \in \mathcal{Z}^m$ and a real-valued function class $\mathcal{G} \subset \mathbb{R}^{\mathcal{Z}}$. Define the pseudometric $d_{2,z}$ on $\mathcal{G}$ as

$$d_{2,z}(g, g') := \sqrt{\frac{1}{m} \sum_{i=1}^{m} \left(g(z_i) - g'(z_i)\right)^2}. \tag{40}$$

Also, let the diameter of $\mathcal{G}$ be

$$\mathrm{diam}_{d_{2,z}}(\mathcal{G}) := \sup_{g,g' \in \mathcal{G}} d_{2,z}(g, g'). \tag{41}$$

At this time, there exists an absolute constant $C > 0$ such that

$$\widehat{\mathfrak{R}}_z(\mathcal{G}) \leq \frac{C}{\sqrt{m}} \int_0^{\mathrm{diam}_{d_{2,z}}(\mathcal{G})} \sqrt{\log \mathrm{CN}_{\mathcal{G},d_{2,z}}(\epsilon)}\, d\epsilon \tag{42}$$

holds.

Remark 4.4. In the right-hand side of Theorem 4.2, the covering number $\mathrm{CN}_{\mathcal{G},d_{2,z}}(\epsilon)$ at each scale ϵ of the function class $\mathcal{G}$ appears. If the function class is very large and many balls are needed at every scale, the integral value becomes large, and the Rademacher complexity also becomes large. Conversely, if the function class is relatively small (or smooth), the covering number becomes small, and the Rademacher complexity is also kept small.
By combining this theorem with Theorem 4.1, we can mathematically support the qualitative trend that:

- If the covering number of the function space is small (i.e., the function space is ”small” in a certain sense), the Rademacher complexity becomes small, and as a result, the generalization gap also becomes small.

5 Performance Limits of Machine Learning by Fano’s Inequality

In the previous chapter, we gave an evaluation method for the **upper bound** of the generalization gap when a function set included in parametric functions is given. This can be regarded as a **sufficient condition** for generalization in the sense that ”if certain conditions (e.g., upper bound of Rademacher complexity) are met, generalization works well”.

In this chapter, conversely, we provide a **necessary condition** that ”better performance cannot be achieved” no matter what algorithm or model is used. Specifically, we show the performance limits of machine learning using **Fano’s inequality** in information theory [4, 5].

5.1 Basic Definitions of Information Theory

First, we define the basic concepts of information theory necessary to state Fano's inequality. The notation hereinafter follows the textbook by Cover and Thomas [4].

Definition 5.1 (Shannon Entropy)**.** Let a random variable V on a finite set $\mathcal{V}$ have a probability mass function

$$p_V(v) := \mathbb{P}(V = v), \quad v \in \mathcal{V}. \tag{43}$$

We define **Shannon entropy** as

$$H(V) := -\sum_{v\in\mathcal{V}} p_V(v) \log p_V(v). \tag{44}$$

Here, the base of the logarithm is 2, and the unit is bits.

Definition 5.2 (Conditional Entropy and Mutual Information)**.** For random variables V, Y on finite sets $\mathcal{V}, \mathcal{Y}$, we define **conditional entropy** as

$$H(V \mid Y) := \mathbb{E}_Y\big[H(V \mid Y = y)\big] = -\sum_{y\in\mathcal{Y}} p_Y(y) \sum_{v\in\mathcal{V}} p_{V|Y}(v \mid y) \log p_{V|Y}(v \mid y). \tag{45}$$

Also, we define **mutual information** as

$$I(V;Y) := H(V) - H(V \mid Y). \tag{46}$$

Definition 5.3 (Kullback–Leibler Divergence)**.** For two probability distributions P and Q on the same finite set $\mathcal{Y}$, we define **Kullback–Leibler divergence** as

$$D(P\|Q) := \sum_{y\in\mathcal{Y}} P(y) \log \frac{P(y)}{Q(y)}. \tag{47}$$

Definition 5.4 (Binary Entropy Function)**.** For $p \in [0, 1]$, we define the **binary entropy function** as

$$h_2(p) := -p \log p - (1 - p) \log(1 - p) \tag{48}$$

(in the case of $0 \log 0$, it is considered 0).

5.2 Fano's Inequality

Fano's inequality is an inequality that bounds the error probability from below by mutual information in the problem of estimating a discrete parameter V from observation Y [4, Theorem 2.10.1].

Theorem 5.1 (Fano's Inequality)**.** Consider a random variable V on a finite set $\mathcal{V}$ and an observation Y. For any estimator $\hat{V} = \hat{V}(Y)$ that estimates the value of V from Y, consider the error probability

$$P_{\mathrm{e}} := \mathbb{P}(\hat{V} \neq V). \tag{49}$$

At this time, the following holds:

$$H(V \mid Y) \leq h_2(P_{\mathrm{e}}) + P_{\mathrm{e}} \log(|\mathcal{V}| - 1). \tag{50}$$

Especially when $|\mathcal{V}| \geq 2$, as a coarser upper bound,

$$H(V \mid Y) \leq 1 + P_{\mathrm{e}} \log |\mathcal{V}| \tag{51}$$

holds.
Furthermore, when V follows a uniform distribution

$$\mathbb{P}(V = v) = \frac{1}{|\mathcal{V}|}, \quad v \in \mathcal{V}, \tag{52}$$

the error probability satisfies

$$P_{\mathrm{e}} \geq 1 - \frac{I(V;Y) + \log 2}{\log |\mathcal{V}|}. \tag{53}$$

Remark 5.1. Equation (50) shows that ”the uncertainty of V, $H(V \mid Y)$, is constrained from below by the error probability P_{e} and the number of states $|\mathcal{V}|$”. Intuitively,

- If P_{e} becomes very small, $H(V \mid Y)$ must also become small (that is, the uncertainty of V is small when Y is known).
- Conversely, if $H(V \mid Y)$ is still large (V is not well known even looking at Y), P_{e} must be large to some extent.

Equation (53) for the special case of uniform distribution shows that when the number of possible states of V, $|\mathcal{V}|$, is large, **unless the mutual information $I(V;Y)$ is sufficiently large, the error probability P_{e} cannot be made small**. This mathematically expresses the basic intuition in machine learning that

- ”The larger the number of hypotheses to distinguish (number of model/parameter candidates), the more information is required to distinguish them.”

5.3 Fano’s Inequality and Minimax Risk

Fano’s inequality is used to give a lower bound for **minimax risk** in statistical estimation problems. The following explanation follows [5].

First, we define minimax risk.

Definition 5.5 (Minimax Risk)**.** Let the parameter space be Θ, and assume that for each $\theta \in \Theta$, a distribution P_θ^n of observation vector Y^n is given (e.g., product distribution of n

independent observations). Let the loss function

$$\ell : \Theta \times \Theta \to \mathbb{R}_{\geq 0} \tag{54}$$

be the loss between the true parameter θ and the estimated value $\hat{\theta}$.
Let the estimator be

$$\hat{\theta} : \mathcal{Y}^n \to \Theta, \tag{55}$$

and define the risk under θ as

$$R_n(\theta, \hat{\theta}) := \mathbb{E}_{Y^n \sim P_\theta^n}\left[\ell\big(\theta, \hat{\theta}(Y^n)\big)\right]. \tag{56}$$

At this time, we define the **minimax risk** as

$$M_n(\Theta, \ell) := \inf_{\hat{\theta}} \sup_{\theta \in \Theta} R_n(\theta, \hat{\theta}). \tag{57}$$

Remark 5.2. If we regard Θ as ”a set of candidate generative models”, P_θ^n as ”data distribution observed under parameter θ”, and $\ell(\theta, \hat{\theta})$ as ”distance between true model and learned model”, then $M_n(\Theta, \ell)$ represents the performance limit in the sense that ”the risk cannot be reduced below this for the worst true model no matter what learning algorithm is used”.

Using Fano’s inequality, we can provide a lower bound for this minimax risk [5, Corollary 1].

Theorem 5.2 (Local Minimax Risk Lower Bound Based on Fano’s Inequality [5, Corollary 1])**.** Assume the setting of Definition 5.5. Fix a finite set of parameters

$$\{\theta_1, \ldots, \theta_M\} \subset \Theta, \tag{58}$$

and let the distance (or pseudometric) between θ be $d : \Theta \times \Theta \to \mathbb{R}_{\geq 0}$. Also, assume there exists a non-decreasing function

$$\Phi : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0} \tag{59}$$

representing the relationship between loss function ℓ and distance d such that

$$\forall \theta, \theta' \in \Theta,\ \ell(\theta, \theta') \geq \Phi\big(d(\theta, \theta')\big) \tag{60}$$

is satisfied.
For arbitrary $\varepsilon \geq 0$, assume that

$$d(\theta_v, \theta_{v'}) > \varepsilon \quad (v \neq v',\ v, v' \in \{1, \ldots, M\}) \tag{61}$$

holds for $\{\theta_1, \ldots, \theta_M\}$ (i.e., ε-packing condition). Furthermore, let V be a uniform distribution on $\{1, \dots, M\}$, and generate observation Y^n according to the Markov chain

$$V \to \theta_V \to Y^n. \tag{62}$$

Consider the Kullback–Leibler divergence between $P^n_{\theta_v}$ and an arbitrary auxiliary distribution Q_n

$$D\left(P^n_{\theta_v} \| Q_n\right). \tag{63}$$

At this time, the minimax risk satisfies

$$M_n(\Theta, \ell) \geq \Phi\left(\frac{\varepsilon}{2}\right)\left(1 - \frac{\min\limits_{v=1,\dots,M} D\left(P^n_{\theta_v} \| Q_n\right) + \log 2}{\log M}\right). \tag{64}$$

Furthermore, $\min_v D(P^n_{\theta_v} \| Q_n)$ appearing in the numerator of the right-hand side can be replaced by average or maximum value to satisfy a similar lower bound (refer to the original paper [5, Corollary 1] for details).

Remark 5.3. Theorem 5.2 corresponds to the following intuition:

- The set $\{\theta_1, \dots, \theta_M\}$ is sufficiently separated from each other in the sense of distance d (packing condition (61)), and much information is needed to distinguish them.
- On the other hand, if the data distribution $P^n_{\theta_v}$ under each θ_v is not very far from a certain auxiliary distribution Q_n (Kullback–Leibler divergence is small), it is difficult to identify the index v from the observation Y^n.
- By Fano's inequality, the error probability of index identification is bounded from below, and as a result, a lower bound is derived that the estimation error in the sense of distance d or loss ℓ cannot be reduced below a certain level.

$\log M$ appearing in the right-hand side of equation (64) represents ”the number of candidates to be identified”, and $D(P^n_{\theta_v} \| Q_n)$ appearing in the numerator represents ”the difficulty of distinguishing between candidates”. Therefore,

- The larger the number of candidates M (the larger the model class Θ), the larger the lower bound of the minimax risk tends to be.
- On the other hand, if each $P^n_{\theta_v}$ becomes distant from each other (KL divergence is large), identification becomes easier, and the lower bound becomes smaller.

From the perspective of machine learning, this can be interpreted as providing a fundamental performance limit that:

- ”If the candidate model class is too huge and the number of observations n is limited, errors above a certain level cannot be avoided no matter what learning algorithm is used.”

6 Summary and Next Lecture Preview

6.1 Answers to Learning Outcomes

Finally, we briefly review the main points corresponding to each of the Learning Outcomes stated in this lecture.

- **Difference between empirical risk and expected risk**

  Expected risk is the expected value of loss with respect to the data generating distribution and represents the average performance in the real environment. On the other hand, empirical risk is the average loss on a finite number of training data and is a quantity that can be actually calculated and optimized. The difference between the two is the generalization gap, and the smaller this is, the more the performance on the training data generalizes to the real environment.

- **Trade-off between empirical risk and generalization gap**

  The larger the function space, the smaller the minimum value of empirical risk tends to be, but on the other hand, the upper bound of the generalization gap also becomes larger. This is one of the reasons why methods like PEFT that reduce the number of parameters can be advantageous from the perspective of data efficiency and generalization performance.

- **Fundamental performance limits of machine learning** When the set of candidate probability distributions is a large set, no matter what estimator (learning algorithm) is used, risk above a certain level cannot be avoided. This suggests that trying to cover all cases is theoretically unreasonable.

6.2 Next Lecture Preview

Next time, we will learn about the basic framework of **learning from rewards**, that is, **Reinforcement Learning**.

References

[1] S. Shalev-Shwartz and S. Ben-David, Understanding Machine Learning: From Theory to Algorithms. Cambridge: Cambridge University Press, 2014.

[2] M. Mohri, A. Rostamizadeh, and A. Talwalkar, Foundations of Machine Learning. Cambridge, MA: MIT Press, 2 ed., 2018.

[3] S. Boucheron, G. Lugosi, and P. Massart, Concentration Inequalities: A Nonasymptotic Theory of Independence. Oxford: Oxford University Press, 2013.

[4] T. M. Cover and J. A. Thomas, Elements of Information Theory. Hoboken, NJ: Wiley-Interscience, 2 ed., 2006.

[5] J. Scarlett and V. Cevher, “An introductory guide to fano’s inequality with applications in statistical estimation.” arXiv preprint, 2019.