Introduction

These brief informal notes aim to concisely—but not exhaustively—illustrate some mathematical aspects of machine learning. The emphasis is on general conceptualization and the connections between this discipline and other mathematical fields, such as statistical inference, computational statistics, statistical learning, statistical decision theory, optimization theory, information theory, and others.

In broad terms, machine learning is a specialized subfield of artificial intelligence that draws on various techniques and knowledge from multiple mathematical domains to enable computers to solve problems that are intractable using traditional algorithmic methods. Unfortunately, these disciplines often use different terminologies, notations, and modes of reasoning, which can make it challenging for scientists who are not specialists in these areas to develop a unified understanding of the underlying principles.

These notes seek to bridge this gap by providing readers with a unified and synthetic overview of the general prerequisites and foundational concepts. However, they are not intended to replace authoritative papers and textbooks, where more thorough explanations and detailed examples can be found. Additionally, these notes do not cover the practical implementation of mathematical techniques using specific programming languages, frameworks, or libraries.

Chapter 1

Mathematical principles of machine learning

In a broad sense, statistical inference and statistical machine learning share the common goal of understanding (inferring or learning) a certain phenomenon or process characterized by uncertainty, noisiness, and randomness. Typically, the unique information about the phenomenon is represented by a limited amount of empirical sample **data** generated by the mechanism underlying the phenomenon. This generation process is inherently random, producing data that can be viewed as realizations of appropriate **random variables**, distributed according to a specific probability distribution function (PDF) that is often unknown.

Ultimately, the objective is to reverse-engineer the process and infer general properties that apply to both observed and unobserved data. Thus, the most general problem in statistical inference and machine learning is to estimate, from observed data, the PDF that generates all data—both observed and unobserved—related to the phenomenon. Let us now provide a more precise definition of these concepts:

Definition 1.1 *Let $X_1, \ldots, X_N$ be a sample of random variables whose values represent observed **data**. The **statistical inference** (or **statistical machine learning**) consists of inferring (or learning) the distribution that generated the data; that is, the probability distribution function (PDF) f that describes both the observed data and potentially the not-yet-observed data:*

$$X_1, \ldots, X_N \sim f. \tag{1.1}$$

The last definition is completely general, and the distribution f can be any distribution relevant to the problem under investigation; it may be the distribution of a single variable X (denoted by f_X), or a joint, marginal, or conditional distribution. In these notes, we will explore which distribution is most appropriate depending on the problem at hand. It is important to emphasize that Definition 1.1 only describes the general nature of the mathematical problem in machine learning, but does not specify how it is mathematically formulated or solved.

Usually, the distribution f is inferred (learned) by searching within a fixed set of probability density functions, which is called a **statistical model**. The inference problem is called **parametric** if any distribution in the statistical model (f_θ) can be parameterized by quantities $\theta \in \Theta \subseteq \mathbb{R}^d$, which are referred to as **parameters** or **weights**; otherwise, it is called **non-parametric**. The process of performing inference from data, implemented algorithmically in computational contexts, is also known as **learning** (or **fitting**, or **training**) in the machine learning literature.

1.1 Statistical foundations

In what follows, we focus primarily on **parametric** problems and assume that the random variables are **independent and identically distributed** (i.i.d.). We will adopt a **frequentist** approach, in which the parameters θ are fixed but unknown quantities.

The general problem outlined in Definition 1.1 can be formulated in a principled manner by employing concepts from **statistical decision theory**:

Definition 1.2 *Let $X_1, \dots, X_N \sim f_X$ be an i.i.d. sample of $\mathbb{R}^n$-valued random variables defined on the same sample space Ω. Consider the statistical model $\{f_\theta \mid \theta \in \Theta \subseteq \mathbb{R}^d\}$. The **risk** (or **error**) of the estimator distribution f_θ with respect to the true distribution f_X is defined as*

$$R(f_X, f_\theta) := E_X[L(f_X(X), f_\theta(X))] = \int_{X(\Omega)} L(f_X(x), f_\theta(x)) f_X(x) dx, \tag{1.2}$$

*where $L : [0,1]^2 \to \mathbb{R}$ is the **loss** function, and its expression $L(f_X(X), f_\theta(X))$ is a random variable.*

The risk (1.2) quantifies the error incurred when approximating the true distribution f_X with the model distribution f_θ. The optimal value of f_θ can then be determined by **minimizing** this risk.

By selecting a loss function with a logarithmic form, one can define the so-called Kullback-Leibler divergence:

Definition 1.3 *Let $X_1, \dots, X_N \sim f_X$ be an i.i.d. sample of $\mathbb{R}^n$-valued random variables defined on the same sample space Ω. Consider the statistical model $\{f_\theta \mid \theta \in \Theta \subseteq \mathbb{R}^d\}$. The **Kullback–Leibler (KL) divergence** of the estimator distribution f_θ with respect to the true distribution f_X is defined as the risk with a logarithmic loss function:*

$$D_{KL}(f_X || f_\theta) := E_X\left[\ln\left(\frac{f_X(X)}{f_\theta(X)}\right)\right] = \int_{X(\Omega)} \ln\left(\frac{f_X(x)}{f_\theta(x)}\right) f_X(x) dx. \tag{1.3}$$

The quantity (1.3) is one of the most important concepts in statistical inference and machine learning because it provides a fundamental way to measure how one probability distribution diverges from another one. This measure quantifies the expected information lost when using an approximate distribution instead of the true distribution and is widely used to assess differences between probability distributions. The KL divergence behaves like a distance in the sense that it satisfies the following properties:

$$D_{KL}(f||g) \geq 0, \quad D_{KL}(f||g) = 0 \iff f = g. \tag{1.4}$$

However, in general, the triangle inequality is not satisfied, so the KL divergence is not a metric in the formal sense of metric spaces.

Unfortunately, the risk (1.2) cannot be calculated if f is unknown. An alternative measure of risk that utilizes the available data is the **empirical risk**, defined as follows:

Definition 1.4 *Let $X_1, \dots, X_N \sim f_X$ be an i.i.d. sample of $\mathbb{R}^n$-valued random variables defined on the same sample space Ω, realized by the data set $D = (x_1 \dots, x_N)$. Consider the statistical model $\{f_\theta \mid \theta \in \Theta \subseteq \mathbb{R}^d\}$. The **empirical risk (ER)** (also called **empirical error**, or **training error**) of the estimator distribution f_θ with respect to the true distribution f_X is defined as*

$$\hat{R}_N(f_X, f_\theta)(D) := \frac{1}{N} \sum_{k=1}^{N} L(f_X(x_k), f_\theta(x_k)), \tag{1.5}$$

*where $L : [0,1]^2 \to \mathbb{R}$ is the **loss** function.*

The empirical risk (1.5) encodes information about the observed data set and is clearly a function solely of the parameters θ because the data set is fixed. Moreover, it serves as a "good" approximation for (1.2) and, given a suitable loss function, can be minimized with respect to the parameters θ to find the best estimator distribution f_θ. This procedure, known as **empirical risk minimization (ERM)**, is a principled and general method for obtaining what are called **estimators** in statistical inference. Let us now define this concept:

Definition 1.5 *Let $X_1, \ldots, X_N$ be an i.i.d. sample of $\mathbb{R}^n$-valued random variables defined on the same sample space Ω. Consider the model $\{f_\theta \mid \theta \in \Theta \subseteq \mathbb{R}^d\}$. A **statistic** is a random variable $V_N = v(X_1, \ldots, X_N)$ such that $v : (\mathbb{R}^n)^N \to \mathbb{R}^d$ is a measurable function. If the image of v is (a subset of) Θ, then V_N is named point **estimator** for $\theta \in \Theta$. In this latter case V_N is indicated as $\hat{\theta}_N$.*

We have introduced a fundamental concept in machine learning: **estimators**. Estimators are random quantities that depend on the data used to train the model. Specifically, when the data are realized as a set of values $x_1, \ldots, x_N$, the estimator $\hat{\theta}_N$ takes the value $\hat{\theta}_N(x_1, \ldots, x_N) \in \Theta$. These estimators are the central quantities that are **estimated** (or **learned**) from data through the statistical machine learning algorithms that drive the inference (learning) process, as mentioned in Definition 1.1. Therefore, statistical machine learning algorithms perform statistical estimation to accomplish their tasks. It is important to emphasize that an estimator $\hat{\theta}_N$ is a random variable depending on data. Therefore, for every realization of the sample $X_1, \ldots, X_N$, the estimator may take a different value.

As we mentioned before, empirical risk minimization provides a fundamental method to find estimators, once a suitable loss function has been chosen:

Definition 1.6 *Let $X_1, \ldots, X_N \sim f_X$ be an i.i.d. sample of $\mathbb{R}^n$-valued random variables defined on the same sample space Ω, realized by the data set $D = (x_1 \ldots, x_N)$. Consider the statistical model $\{f_\theta \mid \theta \in \Theta \subseteq \mathbb{R}^d\}$. The **empirical risk minimization (ERM)** method gives the **ERM estimator** as follows:*

$$\hat{\theta}_N^{ERM}(D) \in \arg\min_{\theta \in \Theta} \hat{R}_N(f_X, f_\theta)(D). \tag{1.6}$$

The ERM method is the first optimization problem introduced in these notes, highlighting a fundamental connection between statistical machine learning and optimization theory. If one uses the empirical risk minimization method together with the empirical KL divergence, one obtains the so-called **maximum likelihood** method:

Definition 1.7 *Let $X_1, \ldots, X_N \sim f_X$ be an i.i.d. sample of $\mathbb{R}^n$-valued random variables defined on the same sample space Ω, realized by the data set $D = (x_1 \ldots, x_N)$. Consider the statistical model $\{f_\theta \mid \theta \in \Theta \subseteq \mathbb{R}^d\}$. The ERM method (1.6), along with the **empirical KL divergence***

$$\hat{D}_{KL}(f_X || f_\theta)_N(D) := \frac{1}{N} \sum_{k=1}^{N} \ln\left(\frac{f_X(x_k)}{f_\theta(x_k)}\right) = -\frac{1}{N} \sum_{k=1}^{N} \ln f_\theta(x_k) + \text{const}, \tag{1.7}$$

*is named the **maximum likelihood (ML)** method.*

Combining equations (1.7) and (1.6) yields the following optimization problem, which determines the so-called **maximum likelihood (ML)** estimator:

$$\hat{\theta}_N^{ML}(D) \in \arg\max_{\theta \in \Theta} \left(\sum_{k=1}^{N} \ln f_\theta(x_k)\right) = \arg\max_{\theta \in \Theta} \left(\prod_{k=1}^{N} f_\theta(x_k)\right). \tag{1.8}$$

The two objective functions in (1.8) are referred to as the **log-likelihood function (LLF)** and the **likelihood function (LF)**, respectively. The second equality in (1.8) follows from the monotonicity of the logarithm function, which ensures that the LF and the LLF attain their maximum values at the same point.

The likelihood can be interpreted as the probability of observing the data D given the parameters θ. When θ is unknown, a natural approach is to estimate it by selecting the values that **maximize the probability of the observed data**. In the case of i.i.d. data, this probability is given by the likelihood function. These ideas underly the problem stated in (1.8).

Let's introduce some key concepts related to estimators.

Definition 1.8 *The **mean** (or **expected value**) of the point estimator $\hat{\theta}_N$ is defined as*

$$E_{X_1,\dots,X_N}[\hat{\theta}_N] := \int_{X_1(\Omega)\times\cdots\times X_N(\Omega)} \hat{\theta}_N(x_1,\dots,x_N) f_X(x_1)\cdots f_X(x_N) dx_1\cdots dx_N. \tag{1.9}$$

The joint probability distribution function (PDF) in (1.9) is the product of marginal probability distribution functions because $X_1,\dots,X_N$ are i.i.d..

Definition 1.9 *The **bias** of the point estimator $\hat{\theta}_N$ is defined as*

$$BIAS(\hat{\theta}_N,\theta) := E_{X_1,\dots,X_N}[\hat{\theta}_N] - \theta. \tag{1.10}$$

*A point estimator is said to be **unbiased** if its bias is zero.*

The bias quantifies how far the estimate $\hat{\theta}_N$ is from θ on average. Heuristically, unbiased estimators are generally more desirable than biased ones because they are, on average, more accurate.

Definition 1.10 *The **variance** of the point estimator $\hat{\theta}_N = (\hat{\theta}_1,\dots,\hat{\theta}_d)$ of $\theta \in \Theta \subseteq \mathbb{R}^d$ is defined as*

$$\begin{aligned} VAR(\hat{\theta}_N) &:= E_{X_1,\dots,X_N}[||\hat{\theta}_N - E_{X_1,\dots,X_N}[\hat{\theta}_N]||^2] \\ &= \sum_{k=1}^{d} E_{X_1,\dots,X_N}[(\hat{\theta}_k - E_{X_1,\dots,X_N}[\hat{\theta}_k])^2] \\ &= \sum_{k=1}^{d} VAR(\hat{\theta}_k) = \operatorname{tr} COV(\hat{\theta}_N), \end{aligned} \tag{1.11}$$

*where $\operatorname{tr} COV(\hat{\theta}_N)$ denotes the trace of the **covariance matrix** of $\hat{\theta}_N$.*

Definition 1.11 *The **mean squared error (MSE)** of the point estimator $\hat{\theta}_N$ is defined as*

$$MSE(\hat{\theta}_N,\theta) := E_{X_1,\dots,X_N}[||\hat{\theta}_N - \theta||^2]. \tag{1.12}$$

The mean squared error (MSE) is a reasonable criterion for measuring the performance of an estimator. If the MSE is small, we expect that, on average, the resulting estimates are close to the true value. Moreover, the MSE has the interesting property of being decomposable into a sum of bias and variance, as illustrated by the following proposition.

Proposition 1.1 *The MSE of an estimator $\hat{\theta}_N$ can be decomposed as a sum of its bias and its variance:*

$$MSE(\hat{\theta}_N,\theta) = ||BIAS(\hat{\theta}_N,\theta)||^2 + VAR(\hat{\theta}_N). \tag{1.13}$$

This result makes unbiased estimators good candidates due to their zero bias. However, it is important to emphasize that having zero (or small) bias alone is not sufficient to assess an estimator's quality; the variance in (1.13) also plays a crucial role in evaluating the MSE. A low-bias estimator (i.e., accurate on average) with high variance may produce imprecise predictions and fluctuate significantly depending on the data set $x_1,\dots,x_N$. A good estimator is therefore one that is unbiased and exhibits low variance.

The MSE is a good criterion for evaluating estimator performance because it quantifies the deviation of the estimator $\hat{\theta}_N$ from the true value θ, as expressed in the following proposition:

Proposition 1.2 *Let $\hat{\theta}_N$ be an estimator with finite variance, $VAR(\hat{\theta}_N) < \infty$. Then, for all $\epsilon > 0$*

$$P(||\hat{\theta}_N - \theta|| > \epsilon) \leq \frac{MSE(\hat{\theta}_N, \theta)}{\epsilon^2}. \tag{1.14}$$

The meaning of this result is clear: if an estimator has a low mean squared error (MSE), then the probability that it produces a value far from the true value is low. Moreover, if we assume that the MSE approaches zero as $N \to \infty$, it follows that the estimator is **consistent**:

Definition 1.12 *The point estimator $\hat{\theta}_N$ is called **consistent** if its MSE vanishes as $N \to \infty$, that is, it converges in probability to θ:*

$$\lim_{N\to\infty} P(||\hat{\theta}_N - \theta|| > \epsilon) = 0. \tag{1.15}$$

Consistency is a fundamental property that describes the **asymptotic** behavior of an estimator. It ensures that, as the sample size increases, the estimator produces values that get closer and closer to the parameter θ. Note that the convergence of the mean squared error (MSE) to zero implies consistency; however, the converse is not true in general.

We conclude this section by examining the properties of the empirical risk (1.5) and exploring in which sense it serves as a good approximation of the true risk (1.2). Some general concepts of random variables and estimators will be useful. Let us begin by defining the empirical risk as an estimator of the true risk:

Definition 1.13 *Let $X_1, \ldots, X_N \sim f_X$ be an i.i.d. sample of $\mathbb{R}^n$-valued random variables defined on the same sample space Ω. Consider the statistical model $\{f_\theta \mid \theta \in \Theta \subseteq \mathbb{R}^d\}$. The **empirical risk** is an **estimator** of true risk (1.2) and is defined as the **empirical mean** of the loss function $L(f_X(X), f_\theta(X))$, considered as a random variable:*

$$\hat{R}_N(f_X, f_\theta) := \overline{L(f_X(X), f_\theta(X))}_N = \frac{1}{N}\sum_{k=1}^{N} L(f_X(X_k), f_\theta(X_k)). \tag{1.16}$$

It is evident that the empirical risk in (1.5) is a realization of the empirical risk estimator in (1.16), and the latter quantity is a random variable depending on the data set D.

The empirical risk is useful for efficiently approximating the true risk because it is an **unbiased** and **consistent** estimator of the true risk. These properties rely heavily on the assumption that the data set is i.i.d.. Let us now formalize these concepts:

Proposition 1.3 *Let $X_1, \ldots, X_N \sim f_X$ be an i.i.d. sample of $\mathbb{R}^n$-valued random variables defined on the same sample space. The empirical risk (1.16) is an **unbiased** and **consistent** estimator of true risk (1.2):*

$$E_{X_1,\ldots,X_N}[\hat{R}_N(f_X, f_\theta)] = R(f_X, f_\theta), \tag{1.17}$$

$$\lim_{N\to\infty} P(|\hat{R}_N(f_X, f_\theta) - R(f_X, f_\theta)| > \epsilon) = 0. \tag{1.18}$$

It is interesting to note that the consistency of the empirical risk is related to the **law of large numbers**, as stated in the following proposition:

Proposition 1.4 *Let $X_1, \ldots, X_N \sim f_X$ be an i.i.d. sample of $\mathbb{R}^n$-valued random variables defined on the same sample space Ω. Consider the loss function $L(f_X(X), f_\theta(X))$ as a random variable. Therefore, the sequence $L(f_X(X_1), f_\theta(X_1)), \ldots, L(f_X(X_N), f_\theta(X_N))$ is also i.i.d. and satisfies the **law of large numbers**:*

$$\lim_{N\to\infty} P\left(\left|\frac{1}{N}\sum_{k=1}^{N} L(f_X(X_k), f_\theta(X_k)) - E_X[L(f_X(X), f_\theta(X))]\right| > \epsilon\right) = 0. \tag{1.19}$$

Consequently, the empirical risk estimator defined in (1.16) is consistent.

1.2 Statistical learning theory foundations

This section is dedicated to the **statistical learning** theory perspective of machine learning. It is explored through the framework of **probably approximately correct (PAC)** learning. Our goal is to illustrate the fundamental principles of this theory and to compare the inferential approach with the learning theory approach, highlighting how these two frameworks complement each other.

Learning theory is a complementary paradigm that aims to solve **predictive** tasks rather than inferential ones. Roughly speaking, its main goal is not to infer (or learn) the distribution that generated the data in D, but to use the data to learn quantitative relationships between the random variables X and Y (defined on Ω). These relations are expressed as functions (called **hypotheses**) $h : \mathcal{X} \to \mathcal{Y}$, where $\mathcal{X} \subseteq X(\Omega)$ and $\mathcal{Y} \subseteq Y(\Omega)$. The hypotheses are assumed to belong to the set H that is a subset of all **measurable** functions from $\mathcal{X}$ to $\mathcal{Y}$, denoted by $M(\mathcal{X}, \mathcal{Y})$.

The learning theory paradigm addresses questions beyond those tackled by statistical inference, doing so in a way that does not depend on specific statistical models or data distributions. For instance, it studies the number of samples in a data set D—known as the **sample complexity**—that are necessary to achieve effective learning, irrespective of the underlying distribution generating D. According to the theory, this sample complexity is closely connected to the **complexity** of the hypothesis—often measured by the number of parameters describing it—and this relationship affects both the efficiency of the learning process and the model's ability to **generalize** to unseen data during training.

These questions are addressed within a probabilistic framework that differs from the inferential one we used, as it essentially leaves **unspecified** the particular statistical inference applied to the data set.

Let us formalize the learning problem in this framework:

Definition 1.14 *Let $D = \{(x_1, y_1), \ldots, (x_N, y_N)\} \subset \mathcal{X} \times \mathcal{Y}$ be an i.i.d. data set. The **(supervised) learning problem** consists of finding parameters $\theta \in \Theta \subseteq \mathbb{R}^d$ that determine a **hypothesis** $h_\theta : \mathcal{X} \to \mathcal{Y}$, where the hypothesis space is $H = \{h_\theta : \mathcal{X} \to \mathcal{Y} \mid \theta \in \Theta\}$, such that h_θ predicts the corresponding output $y \in \mathcal{Y}$ from input $x \in \mathcal{X}$ with a desired level of accuracy.*

The term **supervised** means that learning the parameters requires using the complete set of data (x, y) to guide the learning process. As mentioned before, the emphasis in learning theory is primarily on prediction through the hypothesis h_θ. Moreover, it is important to emphasize that the data set D is still generated by a distribution $f_{X,Y}$.

A fundamental concept in learning theory is the **learning algorithm**, which represents the procedure for obtaining the best hypothesis from a data set D. The notion of a learning algorithm can be formalized as follows:

Definition 1.15 *Consider the hypothesis space $H = \{h_\theta : \mathcal{X} \to \mathcal{Y} \mid \theta \in \Theta \subseteq \mathbb{R}^d\}$. A **learning algorithm** is a function*

$$A_H : \bigcup_{N \in \mathbb{N}} (\mathcal{X} \times \mathcal{Y})^N \to H.$$

Given the i.i.d. data set $D = \{(x_1, y_1), \ldots, (x_N, y_N)\} \in (\mathcal{X} \times \mathcal{Y})^N$, the algorithm selects the hypothesis $A_H(D) := h_{\hat{\theta}(D)} \in H$.

Definitions 1.14 and 1.15 are somewhat vague on how to determine the best hypothesis h_θ from the data set D. To formulate a principled method, we use an approach similar to the one employed for the formulation of inference in the previous section. Then, we define the **risk** in the context of learning theory:

Definition 1.16 *Let $(X_1, Y_1), \ldots, (X_N, Y_N) \sim f_{X,Y}$ be an i.i.d. sample of $(\mathcal{X} \times \mathcal{Y})$-valued random variables defined on the same sample space Ω. Consider the hypothesis space $H = \{h_\theta : \mathcal{X} \to \mathcal{Y} \mid \theta \in \Theta \subseteq \mathbb{R}^d\}$. The **risk** (or **error**) of the hypothesis h_θ is defined as*

$$R(h_\theta) := E_{X,Y}[L(h_\theta(X), Y)] = \int_{\mathcal{X} \times \mathcal{Y}} L(h_\theta(x), y) f_{X,Y}(x, y) dx dy, \tag{1.20}$$

*where $L : \mathcal{Y}^2 \to \mathbb{R}$ is the **loss** function, and its expression $L(h_\theta(X), Y)$ is a random variable.*

The optimal value of h_θ can then be determined by **minimizing** this risk:

$$h_\theta^* := h_{\theta^*} \in \arg\min_{h_\theta \in H} R(h_\theta). \tag{1.21}$$

$$\theta^* \in \arg\min_{\theta \in \Theta} R(h_\theta). \tag{1.22}$$

More generally, if $H = M(\mathcal{X}, \mathcal{Y})$, we can define the **Bayes optimal hypothesis** and **Bayes error**:

Definition 1.17 *The **Bayes optimal hypothesis** is defined as:*

$$h^* \in \arg\min_{h \in M(\mathcal{X}, \mathcal{Y})} R(h). \tag{1.23}$$

*The value of risk calculated with h^* is called **Bayes error**:*

$$R^* := \min_{h \in M(\mathcal{X}, \mathcal{Y})} R(h) = R(h^*). \tag{1.24}$$

Since the true distribution $f_{X,Y}$ is generally unknown, the **empirical risk** is introduced as a proxy for the true risk:

Definition 1.18 *Let $(X_1, Y_1), \ldots, (X_N, Y_N) \sim f_{X,Y}$ be an i.i.d. sample of $(\mathcal{X} \times \mathcal{Y})$-valued random variables defined on the same sample space Ω. Consider the hypothesis space $H = \{h_\theta : \mathcal{X} \to \mathcal{Y} \mid \theta \in \Theta \subseteq \mathbb{R}^d\}$. The **empirical risk** is an **estimator** of true risk (1.19) and is defined as the **empirical mean** of the loss function $L(h_\theta(X), Y)$, considered as a random variable:*

$$\hat{R}_N(h_\theta) := \overline{L(h_\theta(X), Y)}_N = \frac{1}{N} \sum_{k=1}^{N} L(h_\theta(X_k), Y_k). \tag{1.25}$$

It is easy to see that the empirical risk in learning theory is an **unbiased** and **consistent** estimator of the true risk, similar to what we established in the inference framework.

The **empirical risk minimization (ERM)** method is an important type of learning algorithm within the framework of learning theory. This algorithm leads to the definition of the ERM estimator and the corresponding minimizing hypothesis:

Definition 1.19 *Let $(X_1, Y_1), \ldots, (X_N, Y_N) \sim f_{X,Y}$ be an i.i.d. sample of $(\mathcal{X} \times \mathcal{Y})$-valued random variables defined on the same sample space Ω, realized by the data set $D = \{(x_1, y_1) \ldots, (x_N, y_N)\}$. Consider the hypothesis space $H = \{h_\theta : \mathcal{X} \to \mathcal{Y} \mid \theta \in \Theta \subseteq \mathbb{R}^d\}$. The **empirical risk minimization (ERM)** algorithm ERM_H selects the minimizing hypothesis (named **empirical risk hypothesis (ERH)**) as follows:*

$$ERM_H(D) := \hat{h}_N^{ERM}(D) := h_{\hat{\theta}_N^{ERM}(D)} \in \arg\min_{h_\theta \in H} \hat{R}_N(h_\theta)(D), \tag{1.26}$$

*where the **ERM estimator** is defined as follows:*

$$\hat{\theta}_N^{ERM}(D) \in \arg\min_{\theta \in \Theta} \hat{R}_N(h_\theta)(D). \tag{1.27}$$

The previous definition illustrates how the parameter values corresponding to the minimizing hypothesis can be considered an **estimator**, since their values vary randomly with the data set D. However, in this case, the estimated value does not correspond to any obvious parameter of a probability density function (PDF).

The main concept of learning theory is the **agnostic PAC learnability** of a hypothesis space, which is defined as follows:

Definition 1.20 *The hypothesis space $H = \{h_\theta : \mathcal{X} \to \mathcal{Y}\}$ is **agnostic probably approximately correct (PAC) learnable** with respect to the **loss function** $L : \mathcal{Y}^2 \to \mathbb{R}$ if there exists a **learning algorithm** A_H and a **polynomial** function $q_H : (0,1)^2 \to \mathbb{R}$ such that for any $\epsilon > 0, \delta \in (0,1)$, for all **distributions** $f_{X,Y}$ on $\mathcal{X} \times \mathcal{Y}$, and for any i.i.d. **data set** D that realizes the sequence of i.i.d. random variables $(X_1, Y_1), \ldots, (X_N, Y_N) \sim f_{X,Y}$ with **sample complexity** $N \geq q_H(1/\epsilon, 1/\delta)$, the following holds:*

$$P(R(\hat{h}_N) - R^* \leq \epsilon) \geq 1 - \delta, \tag{1.28}$$

*where $\hat{h}_N := h_{\hat{\theta}_N} := A_H(X_1, Y_1, \ldots X_N, Y_N)$ is the (random) hypothesis selected by A_H from H. If A_H returns $h_{\hat{\theta}_N}$ in a time proportional to $q_H(1/\epsilon, 1/\delta)$, then H is said to be **efficiently agnostic PAC learnable**. The quantity $R(\hat{h}_N) - R^*$ is named **excess risk**.*

The Definition 1.20 involves two accuracy parameters. The accuracy parameter ϵ, which specifies how close the output hypothesis must be to the optimal hypothesis (this corresponds to the "approximately correct" part), while the confidence parameter δ specifies the probability that the hypothesis meets this accuracy requirement (this corresponds to the "probably" part). This learnability property does not depend on distribution $f_{X,Y}$, but depends on the number of samples N, which must be greater than a threshold $q_H(1/\epsilon, 1/\delta)$, which determines the sample complexity of learning H. This threshold depends on the accuracy parameters ϵ, δ. Since q_H is polynomial, optimal learning, corresponding to **small** values of ϵ and **small** values of δ, implies a **greater** value of N to guarantee learnability.

Proposition 1.5 *The excess risk can be bounded as*

$$\begin{aligned} R(\hat{h}_N) - R^* &\leq R_{opt} + 2R_{gen} + R_{app} \\ &\leq R_{opt} + 2R'_{gen} + R_{app} \end{aligned} \tag{1.29}$$

where

- $R_{opt} := \hat{R}_N(\hat{h}_N) - \hat{R}_N(\hat{h}_N^{ERM})$ *is named **optimization error***
- $R_{gen} := |R(\hat{h}_N) - \hat{R}_N(\hat{h}_N)| \leq R'_{gen} := \sup_{h_\theta \in H} |R(h_\theta) - \hat{R}_N(h_\theta)|$ *are named both **generalization error***
- $R_{app} := R(h_\theta^*) - R^*$ *is named **approximation error***

The generalization error is very important because it is related to the **generalization bound**, which is defined as follows:

Definition 1.21 *Let $H = \{h_\theta : \mathcal{X} \to \mathcal{Y}\}$ be a hypothesis space and $L : \mathcal{Y}^2 \to \mathbb{R}$ a loss function. Let $\kappa : (0,1) \times \mathbb{N} \to (0, \infty)$ be such that for all $\delta \in (0,1)$ holds $\kappa(\delta, N) \to 0$ for $N \to \infty$. We call κ the **generalization bound** for H if for all **distributions** $f_{X,Y}$ on $\mathcal{X} \times \mathcal{Y}$, all $N \in \mathbb{N}$, and all $\delta \in (0,1)$, the following holds:*

$$P\left(\sup_{h_\theta \in H} |R(h_\theta) - \hat{R}_N(h_\theta)| \leq \kappa(\delta, N)\right) \geq 1 - \delta, \tag{1.30}$$

$$P\left(|R(\hat{h}_N) - \hat{R}_N(\hat{h}_N)| \leq \kappa(\delta, N)\right) \geq 1 - \delta. \tag{1.31}$$

The generalization error and generalization bound are essential tools for bounding the excess risk, which is key to analyzing the PAC learnability of a hypothesis space. Indeed, from inequality (1.29), one can reduce the optimization error and approximation error by choosing the empirical risk minimizer $\hat{h}_N = \hat{h}_N^{ERM}$ and by enlarging the hypothesis space H within $M(\mathcal{X}, \mathcal{Y})$. Consequently, the generalization error (in whichever form) becomes the dominant contribution to the excess risk bound.

From (1.31), we can estimate the sample complexity threshold $q_H(1/\epsilon, 1/\delta)$ by solving for N in the following inequality:

$$\kappa(\delta, N) \leq \epsilon. \tag{1.32}$$

In some cases, it is possible to provide an inferential interpretation of the risk minimization parameter (1.22) and its empirical counterpart (1.27) as follows:

Proposition 1.6 *The learning theory parameters $\theta \in \Theta$ can be interpreted as the parameters of a chosen estimator distribution $f_\theta(x, y)$ of the true distribution $f_{X,Y}$ if there exists a suitable loss function $L : \mathcal{Y}^2 \to \mathbb{R}$ and a space of hypotheses $h_\theta : \mathcal{X} \to \mathcal{Y}$, such that the risk in (1.20) coincides with the analogous quantity in inference theory:*

$$L(h_\theta(x), y) = L_{inf}(f_{X,Y}(x, y), f_\theta(x, y)), \tag{1.33}$$

where the estimator distribution $f_\theta(x, y) := F(x, y, h_\theta)$ depends on θ through the dependence on h_θ, and L_{inf} is the inferential loss.

Then, the **risk minimization principle** applied to (1.20) with loss (1.33) connects the learning theory to the inference theory. For example, the **KL divergence** (1.3) can be obtained if we can construct a loss function as:

$$L(h_\theta(x), y) := \ln f_{X,Y}(x, y) - \ln f_\theta(x, y). \tag{1.34}$$

From (1.34), we conclude that the **ML estimator** (1.8) coincides with the learning theory **ERM estimator** (1.27) if we can define empirical risk as:

$$\hat{R}_N(h_\theta)(D) := \frac{1}{N} \sum_{k=1}^{N} (- \ln f_\theta(x_k, y_k)). \tag{1.35}$$

Chapter 2

Supervised classification

In this chapter, we apply the general statistical concepts described in the previous chapter to discuss the problem of **classification** (also called **pattern recognition**). The classification problem aims to model the quantitative relationship between an $\mathbb{R}^n$-valued random variable X, called **feature**, or **pattern**, and a random variable Y that takes values in a finite set C, called the set of **classes**. The values of Y are also called **labels** or **ground truth**.

Using the i.i.d. data set $D = \{(x_1, y_1), \ldots, (x_N, y_N)\}$, the objective is to obtain a **classification function** (also called a **classifier** or a **hypothesis**) $c : \mathbb{R}^n \to C$ that allows us to predict the class y for a new pattern x. The inference (learning) process is called **supervised** since it is guided by utilizing the data set D, which is formed by complete pairs of patterns and labels.

2.1 Statistical foundation

At first glance, the classification problem appears simply as a **prediction** problem that does not seem directly related to any inference problem, as defined in Definition 1.1. In the machine learning literature, often only the predictive aspects are emphasized—that is, the determination of the classification function from data—while the inference principles are left in the background. In contrast, our mathematical discussion begins with the guiding principle illustrated in Definition 1.1, with the goal of providing an inferential definition of classification.

Then, one must find a statistical model and a suitable set of random variables (assumed i.i.d.) for the chosen model. The random variables are clearly $(X_1, Y_1), \ldots, (X_N, Y_N)$, while the choice of model is less obvious. To understand how to build a reasonable model, the concept of **statistical dependence** among random variables can be very helpful:

Definition 2.1 *Two random variables X and Y, defined on the same sample space Ω, are **statistically dependent** if for all $x \in X(\Omega)$ and $y \in Y(\Omega)$,*

$$f_{X,Y}(x, y) \neq f_X(x) f_Y(y), \tag{2.1}$$

*where $f_{X,Y}$ is the **joint** PDF, and f_X, f_Y are the **marginal** PDFs of X and Y, respectively.*

Using the definition of conditional probability, one can show that (2.1) is equivalent to the following relations:

$$f(y|x) \neq f_Y(y), \tag{2.2}$$

$$f(x|y) \neq f_X(x). \tag{2.3}$$

To perform classification, it is reasonable to assume that the variables X and Y are dependent; otherwise, there would be no correlation between them, rendering the classification problem meaningless. Therefore, we assume dependence between X and Y, and consider the conditional probability

density function $f(y|x)$, which captures the dependence of Y on X, as a suitable model candidate. The following definition formalizes the classification problem based on the previous reasoning:

Definition 2.2 *Let $D = \{(x_1, y_1), \ldots, (x_N, y_N)\} \subset \mathbb{R}^n \times C$ be an i.i.d. data set, where C is the finite set of **classes**. The **classification problem** consists of inferring (or learning) the parameters θ of the statistical model*

$$\{f_\theta(y|x) \mid \theta \in \Theta \subseteq \mathbb{R}^d\} \tag{2.4}$$

*from the data in D. After the learning process, the **classification function** $c : \mathbb{R}^n \to C$ is calculated as*

$$c_\theta(x) := \arg\max_{y \in C} f_\theta(y|x) = \arg\max_{y \in C} \ln f_\theta(y|x). \tag{2.5}$$

The classification function has an intuitive interpretation: the class of a pattern x is the most probable value of y conditioned on x. Thus, the classification problem, when viewed as a decision problem, is inherently probabilistic in nature. The probabilistic rule in (2.5) is named **Bayesian classification rule** or **Bayes classifier**.

We provide a complementary geometric definition of classification:

Definition 2.3 *Let $D = \{(x_1, y_1), \ldots, (x_N, y_N)\} \subset \mathbb{R}^n \times C$ be an i.i.d. data set, where C is the finite set of **classes**. The **classification problem** consists of inferring (or learning) from the data in D the parameters $\theta \in \Theta \subseteq \mathbb{R}^d$ of the family of regions of $\mathbb{R}^n$ defined as follows:*

$$\mathcal{R}_C(\theta) := \{\mathcal{R}_y(\theta) \subset \mathbb{R}^n \mid y \in C\}, \tag{2.6}$$

where the region

$$\mathcal{R}_y(\theta) := \{x \in \mathbb{R}^n \mid f_\theta(y|x) > f_\theta(y'|x), \forall y' \neq y\} \subset \mathbb{R}^n \tag{2.7}$$

*is named the **decision rule** for the class $y \in C$. The **decision boundary** between classes y and $y' \neq y$ is defined as:*

$$\mathcal{D}_{yy'}(\theta) := \{x \in \mathbb{R}^n \mid f_\theta(y|x) = f_\theta(y'|x)\}. \tag{2.8}$$

Decision rules and decision boundaries satisfy the following relations:

$$\bigcup_{y \in C} \mathcal{R}_y \cup \bigcup_{y' \neq y} \mathcal{D}_{yy'} = \mathbb{R}^n, \quad \mathcal{R}_y \cap \mathcal{R}_{y'} = \emptyset \quad \forall y \neq y'. \tag{2.9}$$

After the learning process, the pattern x is assigned to class y if $x \in \mathcal{R}_y$.

The previous definition allows us to observe that the classification problem is equivalent to the geometric problem of finding a family of regions $\mathcal{R}_y(\theta)$, where each region corresponds to the set of patterns $x \in \mathbb{R}^n$ belonging to class $y \in C$. Moreover, these regions, together with their decision boundaries, form a **partition** of the pattern space (see (2.9)), since any pattern x can belong to only one class y.

It is important to emphasize that the decision rules in (2.7) depend on parameters θ, which are random variables learned from data, as discussed in the previous chapter. This implies that the decision rules can be viewed as a type of **random set**.

2.2 Further developments

The Definition 2.2 and 2.3 are general because we have not specified the explicit form of the statistical model. To make further progress, we need to specify a suitable form of the statistical model (2.4) and perform inference (learning) on its parameters. This specification can be made in two ways, which are termed **discriminative** and **generative**. Let us formalize these concepts:

Definition 2.4 *A classification problem and its statistical model (2.4), as defined in Definition 2.2, are called* ***discriminative*** *if the goal is to model directly the conditional probability using a suitable parametric function. They are called* ***generative*** *if the goal is to model the conditional probability by applying Bayes' rule to the joint distribution $f_\theta(x, y)$ as follows:*

$$f_\theta(y|x) \propto f_\theta(x, y) = f_\theta(y) f_\theta(x|y), \tag{2.10}$$

where the normalization constant in the denominator is ignored because it is independent of y.

In other words, generative statistical models estimate the joint distribution $f_\theta(x, y)$ and then derive the conditional probability $f_\theta(y|x)$ via Bayes' rule, while discriminative models estimate the conditional probability $f_\theta(y|x)$ directly. Generative models are called so because, once the joint distribution is learned, one can **generate** new data points (x, y) by sampling from this distribution. This simple mathematical principle, implemented in more sophisticated mathematical forms, is the driving force behind so-called **generative AI**, which generates new data, such as images, text, or audio, by sampling from learned probability distributions.

If the set of classes contains only two elements, the classification is called **binary**, and we provide the following discriminative definition:

Definition 2.5 *Let $D = \{(x_1, y_1), \ldots, (x_N, y_N)\} \subset \mathbb{R}^n \times C$ be an i.i.d. data set, where $C = \{y^{(1)}, y^{(2)}\}$ is the finite set of* ***classes****. The* ***(discriminative) binary classification problem*** *consists of inferring (or learning) from data in D the parameters θ of the statistical model in (2.4), where the distributions have the* ***Bernoulli*** *form*

$$f_\theta(y|x) := \mathrm{Be}(y; f_\theta(x)) = \begin{cases} f_\theta(x) & y = y^{(1)} \\ 1 - f_\theta(x) & y = y^{(2)} \end{cases}, \tag{2.11}$$

and $0 \le f_\theta(x) \le 1$ for all x and θ. The ***binary classification function*** *$c : \mathbb{R}^n \to C$ is calculated as*

$$c_\theta(x) := \begin{cases} y^{(1)} & f_\theta(x) > 1/2 \\ y^{(2)} & f_\theta(x) < 1/2 \end{cases}. \tag{2.12}$$

The ***decision rules*** *for the classes in C are*

$$\mathcal{R}_{y^{(1)}}(\theta) := \{x \in \mathbb{R}^n \mid f_\theta(y^{(1)}|x) > f_\theta(y^{(2)}|x)\} = \{x \in \mathbb{R}^n \mid f_\theta(x) > 1/2\}, \tag{2.13}$$

$$\mathcal{R}_{y^{(2)}}(\theta) := \{x \in \mathbb{R}^n \mid f_\theta(y^{(2)}|x) > f_\theta(y^{(1)}|x)\} = \{x \in \mathbb{R}^n \mid f_\theta(x) < 1/2\}. \tag{2.14}$$

The ***decision boundary*** *is*

$$\mathcal{D}_{y^{(1)}y^{(2)}}(\theta) := \{x \in \mathbb{R}^n \mid f_\theta(y^{(1)}|x) = f_\theta(y^{(2)}|x) = f_\theta(x) = 1/2\}. \tag{2.15}$$

For the discriminative binary classification problems, a typical choice for the conditional probability is the **logistic (sigmoid)** function:

$$f_\theta(y^{(1)}|x) = f_\theta(x) := \frac{\exp(b_\theta(x))}{1 + \exp(b_\theta(x))}. \tag{2.16}$$

The logistic function is a natural choice for probability modeling because its output is constrained to the interval $(0, 1) \subset \mathbb{R}$ for every argument $b_\theta(x)$. This approach to binary classification is known as **binary logistic classification** (or **binary logistic regression** in some references). In the special case where $b_\theta(x)$ has a linear form, the model corresponds to **half-space** decision rules separated by a **linear** decision boundary, that is described by a linear function (a **hyperplane**). Data separated by linear decision boundaries are called **linearly separable**, and the decision rules and decision boundaries are called **linear discriminant functions**. Let us formalize these results:

Proposition 2.1 *Consider the* ***binary linear logistic classification problem*** *with a Bernoulli distribution defined in (2.11), where*

$$f_\theta(y^{(1)}|x) = f_\theta(x) := \frac{\exp(b_\theta(x))}{1+\exp(b_\theta(x))} = \frac{1}{1+\exp(-b_\theta(x))}, \tag{2.17}$$

$$f_\theta(y^{(2)}|x) = 1 - f_\theta(x) := \frac{1}{1+\exp(b_\theta(x))}, \tag{2.18}$$

$$b_\theta(x) := \theta_0 + \sum_{k=1}^{n} \theta_k x_k. \tag{2.19}$$

The corresponding ***decision rules*** *are the* ***half-spaces****:*

$$\mathcal{R}_{y^{(1)}}(\theta) = \{x \in \mathbb{R}^n \mid b_\theta(x) > 0\}, \tag{2.20}$$

$$\mathcal{R}_{y^{(2)}}(\theta) = \{x \in \mathbb{R}^n \mid b_\theta(x) < 0\}. \tag{2.21}$$

The ***decision boundary*** *is the hyperplane defined by the Cartesian equation:*

$$b_\theta(x) = 0. \tag{2.22}$$

Finally, the ***classification function*** *is given by:*

$$c_\theta(x) = \begin{cases} y^{(1)} & b_\theta(x) > 0 \\ y^{(2)} & b_\theta(x) < 0 \end{cases}. \tag{2.23}$$

If $y^{(1)} = -y^{(2)} = 1$, the classification function can be compactly represented as:

$$c_\theta(x) = \frac{|b_\theta(x)|}{b_\theta(x)}. \tag{2.24}$$

The geometric interpretation of (2.23) and (2.24) is straightforward: a pattern x belongs to class $y^{(1)}(=1)$ if it lies in the "upper" half-space, or to class $y^{(2)}(=-1)$ if it lies in the "lower" half-space, relative to the direction of the vector $(\theta_1, \ldots, \theta_n) \in \mathbb{R}^n$, which is orthogonal to the separating hyperplane.

The linear binary logistic classification described earlier can be generalized to handle **nonlinear** decision boundaries by incorporating nonlinear functions $b_\theta(x)$. For example, using **quadratic** functions in x, one can describe decision boundaries represented by **hyperconics** such as hyperspheres, hyperellipsoids, hyperparaboloids, and so on:

$$b_\theta(x) := \sum_{k=1}^{n}\sum_{j=1}^{n} q_{kj}(\theta)(x_k - a_k)(x_j - a_j), \tag{2.25}$$

where the coefficients q_{kj} depend on the parameters θ, and $(a_1, \ldots, a_n) \in \mathbb{R}^n$ represent the center of the quadratic form.

For completeness, it is worth giving the properties of the general logistic classification for more than two classes:

Proposition 2.2 *Consider the* ***multiclass classification problem*** *with the following* ***logistic*** *statistical model:*

$$f_\theta(y|x) := \frac{\exp(b_\theta^{(y)}(x))}{\sum_{y' \in C} \exp(b_\theta^{(y')}(x))}, \quad y \in C. \tag{2.26}$$

The corresponding ***decision rule*** *for the class $y \in C$ is:*

$$\mathcal{R}_y(\theta) = \{x \in \mathbb{R}^n \mid b_\theta^{(y)}(x) > b_\theta^{(y')}(x), \forall y' \neq y\}. \tag{2.27}$$

*The **decision boundary** between classes y and $y' \neq y$ is:*

$$\mathcal{D}_{yy'}(\theta) = \{x \in \mathbb{R}^n \mid b_\theta^{(y)}(x) = b_\theta^{(y')}(x)\}. \tag{2.28}$$

*Finally, the **classification function** is given by:*

$$c_\theta(x) = \arg\max_{y \in C} b_\theta^{(y)}(x). \tag{2.29}$$

Let us now consider an important example of the generative classification problem that is based on the Gaussian distribution, which is called **Gaussian discriminant analysis**, and explore its properties.

Definition 2.6 *Let $D = \{(x_1, y_1), \ldots, (x_N, y_N)\} \subset \mathbb{R}^n \times C$ be an i.i.d. data set, where C is the finite set of **classes**. The **Gaussian discriminant analysis (GDA)** is the **generative** classification problem defined by the following statistical model:*

$$f_\theta(y|x) := \alpha \Pi(y) \mathcal{N}(x; \mu(y), \Sigma(y)) = \frac{\alpha \Pi(y)}{\sqrt{(2\pi)^N \det \Sigma(y)}} \exp\left[-\frac{1}{2}(x - \mu(y))^T \Sigma^{-1}(y)(x - \mu(y))\right], \tag{2.30}$$

*where α is the normalization factor, $\Pi(y)$ is the **prior** distribution of y, and $\mu(y)$ and $\Sigma(y)$ are the **mean** vector and the **covariance** matrix of the Gaussian associated with class y, respectively. Therefore, the parameters of the model are $\theta = (\Pi(y), \mu(y), \Sigma(y))$, for $y \in C$.*

Comparing (2.30) with (2.10), it is easy to make the following identifications for the GDA:

$$f_\theta(y) := \Pi(y), \quad f_\theta(x|y) := \mathcal{N}(x; \mu(y), \Sigma(y)). \tag{2.31}$$

The following proposition illustrates the properties of the decision boundaries for the GDA.

Proposition 2.3 *For any pair of labels y and $y' \neq y$, the **decision boundary** of the GDA is given by the following Cartesian equation:*

$$\frac{1}{2} x^T [\Sigma^{-1}(y) - \Sigma^{-1}(y')]x + [\mu(y')\Sigma^{-1}(y') - \mu(y)\Sigma^{-1}(y)]x + const = 0, \tag{2.32}$$

where const denotes the remaining terms that do not depend on the pattern x.

The relation (2.32) illustrates that decision boundaries are **quadratic**, and for this reason, the GDA is called **quadratic discriminant analysis**. In the case where covariance matrices do not depend on class y, the decision boundary is **linear** and the GDA is called **linear discriminant analysis**. The term "discriminant" can be quite confusing because the model is generative, not discriminative.

Another example of a generative classification problem is the so-called **naive Bayes** problem defined as follows:

Definition 2.7 *Let $D = \{(x_1, y_1), \ldots, (x_N, y_N)\} \subset \mathbb{R}^n \times C$ be an i.i.d. data set, where C is the finite set of **classes**. The **naive Bayes** problem is the **generative** classification problem defined by the following statistical model:*

$$f_\theta(y|x) := \alpha \Pi_\theta(y) f_\theta(x|y) = \alpha \Pi_\theta(y) \prod_{k=1}^{n} f_\theta(x^{(k)}|y), \tag{2.33}$$

*where α is the normalization factor, $\Pi_\theta(y)$ is the **prior** distribution of y, and it is assumed that the components of the pattern vector $x = (x^{(1)}, \ldots, x^{(n)}) \in \mathbb{R}^n$ are **conditionally independent** given the class y. This simplifying assumption is known as the **naive Bayes hypothesis**, and permits the product decomposition on the right-hand side in (2.33).*

For the general problem of classification, the standard maximum likelihood method can be used to derive estimators for the model parameters:

$$\hat{\theta}_N^{ML}(D) \in \arg\max_{\theta\in\Theta} \left(\sum_{k=1}^{N} \ln f_\theta(y_k|x_k) \right). \tag{2.34}$$

The expression of the **log-likelihood function** in (2.34) depends on the adopted model. In the case of a **binary logistic classification problem**, expression (2.34) can be rewritten using (2.17), (2.18), and the definition $\eta(y^{(1)}) = -\eta(y^{(2)}) = 1$ as

$$\hat{\theta}_N^{ML}(D) \in \arg\max_{\theta\in\Theta} \left(-\sum_{k=1}^{N} \ln(1 + \exp(-\eta(y_k) b_\theta(x_k)) \right). \tag{2.35}$$

In the case of **multiclass logistic classification**, (2.35) is generalized as:

$$\hat{\theta}_N^{ML}(D) \in \arg\max_{\theta\in\Theta} \left(\sum_{k=1}^{N} \left[b_\theta^{(y_k)}(x_k) - \ln \sum_{y\in C} \exp(b_\theta^{(y)}(x_k)) \right] \right). \tag{2.36}$$

2.3 Learning theory perspectives

The section is devoted to the **learning theory** perspective on **binary** classification, which is developed through the integration of two theories: **probably approximately correct (PAC)** learning and **Vapnik–Chervonenkis (VC)** theory. This approach aims to deepen the understanding of binary classification by showing the synergy between the PAC learning framework—focusing on guarantees of generalization under **sample complexity** constraints—and the VC theory—providing key tools such as **VC dimension** that characterize **hypothesis space complexity**.

Let us formalize the (binary) classification problem within this framework; the definition is a specialization of Definition 1.14:

Definition 2.8 *Let $\mathcal{X} \subseteq \mathbb{R}^n$ and $\mathcal{Y} := \{y^{(1)}, y^{(2)}\}$ be the set of **classes**. Let $D = \{(x_1, y_1), \ldots, (x_N, y_N)\} \subset \mathcal{X} \times \mathcal{Y}$ be an i.i.d. data set. The **supervised binary classification problem** consists of finding parameters $\theta \in \Theta \subseteq \mathbb{R}^d$ that determine a **hypothesis** (or **classifier**) $h_\theta : \mathcal{X} \to \mathcal{Y}$, where the hypothesis space is $H = \{h_\theta : \mathcal{X} \to \mathcal{Y} \mid \theta \in \Theta\}$, such that h_θ predicts the corresponding label $y \in \mathcal{Y}$ from pattern $x \in \mathcal{X}$ with a desired level of accuracy.*

The hypothesis space in the previous definition is described as a space of functions. More broadly, inspired by Definition 2.3, we can define the hypothesis space for a classification problem as the space of decision rules or decision boundaries.

Following the principles of learning theory, one can define the risk using a suitable loss function. In binary classification, a popular choice of loss is the **0-1 loss**, which is defined as:

Definition 2.9 *Given a binary classifier $h_\theta : \mathcal{X} \to \mathcal{Y}$, the **0-1 loss** function $L : \mathcal{Y}^2 \to \mathbb{R}$ is defined such that:*

$$L(h_\theta(x), y) := 1(h_\theta(x) \neq y) = \begin{cases} 1 & h_\theta(x) \neq y \\ 0 & otherwise \end{cases}. \tag{2.37}$$

Combining (2.37) with the definition of risk (1.19), one obtains the **0-1 risk**:

$$R^{(0-1)}(h_\theta) := E_{X,Y}[1(h_\theta(X) \neq Y)] = P(h_\theta(X) \neq Y). \tag{2.38}$$

The relation (2.38) expresses the fact that, in order to reduce the risk, one has to find the hypothesis that minimizes the probability of **misclassification**. This criterion has a remarkable theoretical importance because its **Bayes optimal hypothesis** has a known form given by the following proposition:

Proposition 2.4 *The **Bayes optimal hypothesis** of the **0-1 risk** given in (2.38) is given by*

$$h^*(x) = \begin{cases} y^{(1)} & f(y^{(1)}|x) > 1/2 \\ y^{(2)} & otherwise \end{cases}. \tag{2.39}$$

The importance of result (2.39) lies in the fact that it provides a theoretical justification for the classification function (2.12), specialized for binary classification, establishing a connection between the learning theory approach and the inferential one.

The main idea of learning theory applied to classification is that learning performance depends on a quantity that describes the **complexity** of the hypothesis space. For binary classification it can be quantified by introducing the **Vapnik–Chervonenkis dimension**:

Definition 2.10 *The **Vapnik–Chervonenkis dimension (VC dimension)** of a hypothesis space H of binary classifiers, denoted by $VC\dim H$, is the cardinality of the largest set of points that can be **shattered** by H. By shattering a set of points, we mean that for every possible way of labeling those points as positive or negative, there exists a hypothesis in H that can perfectly classify them.*

The VC dimension is a purely **combinatorial** notion, which is independent of any probabilistic models of data. For binary linear classification problems, one can quantify the VC dimension of the hyperplane hypothesis space using the following proposition:

Proposition 2.5 *Let H be the hypothesis space of hyperplanes in $\mathbb{R}^n$. Then, $VC\dim H = n + 1$.*

For general hypothesis spaces, not only does the VC dimension characterize PAC learnability; it even determines the sample complexity:

Proposition 2.6 *Let H be a hypothesis space of functions $h : \mathcal{X} \to \{0, 1\}$, and consider the **0-1 loss** function. Assume that $VC\dim H = d < \infty$. Then, there exists a constant $C > 0$ such that H is **agnostic PAC learnable** with **sample complexity***

$$N \geq C\frac{d + \ln(1/\delta)}{\epsilon^2}. \tag{2.40}$$

The previous theorem is a key result in learning theory for binary classification, addressing a type of practical question that is often neglected in inference theory — namely, how many data samples (sample complexity) are required to guarantee effective learning given the accuracy parameters ϵ, δ, and hypothesis complexity d. Furthermore, unlike inference theories that provide asymptotic guarantees on estimators, this result (2.40) is clearly non-asymptotic, making it applicable to practical scenarios involving **finite** data sets. However, the major drawback of bounds like (2.40) is that they can be extremely **loose** for most problems due to their distribution-independence.

It is worth reflecting on the dependence of the right-hand side of (2.40) on d. This linear dependence explains that the greater the complexity of the hypothesis space, the greater the sample complexity should be to guarantee effective learning. In the case of linear classification with a hypothesis space of hyperplanes, the sample complexity is related to the **number of parameters** of the model, which is $n + 1$ (Proposition 2.5). Therefore, the more features the pattern vectors have, the larger the number of training samples required; this claim has great practical value.

We now consider the empirical 0-1 risk, which is given by:

$$\hat{R}_N^{(0-1)}(h_\theta)(D) := \frac{1}{N}\sum_{k=1}^{N} 1(h_\theta(x_k) \neq y_k) = \frac{N_{misc}(D)}{N}. \tag{2.41}$$

The quantity $N_{misc}(D)/N$ denotes the rate of **misclassified** data points in D by the hypothesis h_θ. Then, the ERM method suggests that the best hypothesis is obtained by minimizing quantity (2.41). In some special cases, one can find a particular hypothesis for which the number of misclassifications is exactly **zero**. This is the case of an **separable data set**, which was mentioned previously for linear classification.

Definition 2.11 *The i.i.d. data set* $D = \{(x_1, y_1), \ldots, (x_N, y_N)\} \subset \mathcal{X} \times \mathcal{Y}$ *is* ***separable*** *if there exists a hypothesis* $h_\theta \in H$ *such that* $\hat{R}_N^{(0-1)}(h_\theta)(D) = 0$.

Now we establish an important excess risk bound that describes generalization:

Proposition 2.7 *Let* H *be a hypothesis space of binary classifiers taking values in* $\{1, -1\}$*, having* $VC\dim H = d < \infty$*. Then, for any* $\delta \in (0, 1)$*, for any* $h \in H$*, over an i.i.d. sample* D *of data with size* N *drawn according to* $f_{X,Y}$*, the following bounds hold:*

$$P\left(R(h) - \hat{R}_N(h) \leq O\left(\sqrt{\frac{\ln(N/d)}{N/d}}\right)\right) \geq 1 - \delta. \tag{2.42}$$

The generalization bound (2.42) is very expressive because it states that, in order to have effective generalization, the number of data points N should be greater than the VC dimension of H. This result is consistent with (2.40).

Using (1.35) with $f_\theta(y|x)$ defined in (2.17) and (2.18) as estimator distribution, we obtain an alternative loss function used in the binary classification problem, the **logistic loss**:

Definition 2.12 *Consider a binary classification problem with classes* $y = \pm 1$ *and a hypothesis space consisting of decision rules represented by curves in* $\mathbb{R}^n$ *defined by equations* $h_\theta(x) = 0$*. The* ***logistic loss*** *is defined as:*

$$L(h_\theta(x), y) := \ln(1 + \exp(-y h_\theta(x)). \tag{2.43}$$

Equation (2.43) suggests that binary logistic classification with classes $y = \pm 1$ can be formulated as a binary classification problem in learning theory, which is solved by determining the **ERM estimator**:

$$\hat{\theta}_N^{ERM}(D) \in \arg\min_{\theta \in \Theta}\left(\frac{1}{N}\sum_{k=1}^{N} \ln(1 + \exp(-y_k h_\theta(x_k))\right), \tag{2.44}$$

which is equivalent to **ML estimator** (2.35).

Chapter 3

Supervised regression

In this chapter, we discuss the problem of **regression**. The regression problem aims to model the quantitative relationship between a $\mathbb{R}^n$-valued random variable X, which is named **covariate**, **predictor**, or **feature**, and a $\mathbb{R}^m$-valued random variable Y, named the **response**, **ground truth**, or **labels**.

Using the i.i.d. data set $D = \{(x_1, y_1), \ldots, (x_N, y_N)\}$, the objective is to obtain a **regression function** (also called a **hypothesis**) $r : \mathbb{R}^n \to \mathbb{R}^m$ that allows us to predict the value $y \approx r(x)$ for new feature x. If r is a linear function, the problem is called **linear regression**; otherwise, it is called **nonlinear regression**. The inference (learning) process is called **supervised** since it is guided by utilizing the data set D, which is formed by complete pairs of features and labels.

3.1 Statistical foundation

Similarly to classification, the problem of regression is formulated according to the guiding principle illustrated in Definition 1.1, with the objective of providing an inferential definition of regression.

Then, one must find a statistical model and a suitable set of random variables (assumed i.i.d.) for the chosen model. The random variables are clearly $(X_1, Y_1), \ldots, (X_N, Y_N)$, while the choice of model can be borrowed from the classification problem. Then, we assume that the variable X and Y are dependent, and the conditional probability density function $f(y|x)$ is the suitable model.

The following definition formalizes the regression problem:

Definition 3.1 *Let $D = \{(x_1, y_1), \ldots, (x_N, y_N)\} \subset \mathbb{R}^n \times \mathbb{R}^m$ be an i.i.d. data set. The **regression problem** consists of inferring (or learning) the parameters θ of the statistical model*

$$\{f_\theta(y|x) \mid \theta \in \Theta \subseteq \mathbb{R}^d\} \tag{3.1}$$

*from the data in D. After the learning process, the **regression function** $r : \mathbb{R}^n \to \mathbb{R}^m$ is calculated as*

$$r_\theta(x) := E_\theta(Y|X = x) = \int y f_\theta(y|x) dy. \tag{3.2}$$

In the last definition, the data x is regarded as **fixed** before inference, this fact is called **fixed design**. In applications, the final objective is to infer the regression function from data. To explain this related problem, we illustrate a complementary definition of regression:

Definition 3.2 *Let $D = \{(x_1, y_1), \ldots, (x_N, y_N\}) \subset \mathbb{R}^n \times \mathbb{R}^m$ be an i.i.d. data set. The **regression problem** consists of inferring (or learning) the parameters θ data in D, such that one assumes the following relation between X and Y:*

$$Y = r_\theta(X) + \varepsilon, \tag{3.3}$$

*where ε, called **noise**, is a $\mathbb{R}^m$-valued random variable with the following properties ($i = 1, \ldots, m$):*

$$E(\varepsilon_i | X = x) = 0, \tag{3.4}$$

$$VAR(\varepsilon_i | X = x) = \sigma^2. \tag{3.5}$$

The random variable ε represents the fluctuations that cannot be captured by the model, and it is independent of X and θ. These fluctuations correspond to the intrinsic "error" committed when assuming relation (3.3). Relation (3.5) implies that all components ε_i have the same variance; this property is named **homoscedasticity**.

Definition 3.3 *The simple linear regression problem*

The following proposition illustrates how the linear regression problem has a closed solution under certain hypotheses.

Proposition 3.1 *Consider the linear regression problem defined in Definition 2.3 with $m = 1$, $\epsilon \sim \mathcal{N}(0, \sigma^2)$ and regression function $r_\theta(x)$. Define the following matrices:*

$$\bar{y} = \begin{pmatrix} y^{(1)} \\ \vdots \\ y^{(N)} \end{pmatrix} \in \mathbb{R}^N, \tag{3.6}$$

$$\hat{\theta} = \begin{pmatrix} \hat{\theta}_0 \\ \vdots \\ \hat{\theta}_{d-1} \end{pmatrix} \in \mathbb{R}^d, \tag{3.7}$$

$$\bar{x} = \begin{pmatrix} 1 & x_1^{(1)} & \cdots & x_{d-1}^{(1)} \\ \vdots & \vdots & \vdots & \vdots \\ 1 & x_1^{(N)} & \cdots & x_{d-1}^{(N)} \end{pmatrix} \in \mathbb{R}^{N \times d}. \tag{3.8}$$

Suppose that $\bar{x}^T \bar{x} \in \mathbb{R}^{d \times d}$ is invertible (i.e., $\bar{x}$ has full rank $d \leq N$). Then, the maximum likelihood method gives the following estimators:

$$\hat{\theta}_N = (\bar{x}^T \bar{x})^{-1} \bar{x}^T \bar{y}, \tag{3.9}$$

$$\hat{\sigma}_N^2 = \frac{1}{N} \sum_{k=1}^{N} (y_k - r_{\hat{\theta}_N}(x_k))^2 = \frac{1}{N} ||\bar{y} - \bar{x}\hat{\theta}_N||^1. \tag{3.10}$$

Proof. Suppose $\epsilon \sim \mathcal{N}(0, \sigma^2)$. Then, the distribution in (2.4) has a **Gaussian** form with mean μ depending on θ and x (because, as stated in (2.5), the mean is the regression function), and variance σ^2 as another independent parameter that adds up with θ (recall (1.15)):

$$f_\theta(y | x; \sigma^2) = \mathcal{N}(y; \mu_\theta(x), \sigma^2) = \mathcal{N}(y; r_\theta(x), \sigma^2). \tag{3.11}$$

In other words, one has $Y | X = x \sim \mathcal{N}(r_\theta(x), \sigma^2)$, with the variance in (2.15) the same as ϵ. To estimate the parameters of the regression function, we use the ML method to obtain the LLF (recall (1.12) and (1.13)):

$$-LLF_D(\theta, \sigma^2) = -\sum_{k=1}^{N} \ln \mathcal{N}(y_k; r_\theta(x_k), \sigma^2) = \frac{N}{2} \ln(2\pi\sigma^2) + \frac{1}{2\sigma^2} \sum_{k=1}^{N} (y_k - r_\theta(x_k))^1. \tag{3.12}$$

We changed sign because we want to convert a maximization problem into a minimization one. Using the MML, the original inference problem was converted into the following function optimization problems:

$$\theta(D) = \arg\min_{\theta\in\Theta} R_D(\theta), \tag{3.13}$$

$$\sigma^2(D) = \arg\min_{\theta\in\Theta} S_D(\sigma^2), \tag{3.14}$$

where the target functions are defined as follows:

$$R_D(\theta) = \sum_{k=1}^{N}(y_k - r_\theta(x_k))^2, \tag{3.15}$$

$$S_D(\sigma^2) = \frac{N}{2}\ln(2\pi\sigma^2) + \frac{1}{2\sigma^2}R_D(\theta). \tag{3.16}$$

Using definitions in (2.10), (2.11), (2.12), the function in (2.19) can be rewritten using the Euclidean norm as follows:

$$R_D(\theta) = ||\bar{y} - \bar{x}\hat{\theta}||^1. \tag{3.17}$$

Calculating the matrix gradient of (2.21) with respect to $\hat{\theta}$ and equating it to zero yields the following matrix equation:

$$\bar{x}^T\bar{x}\hat{\theta} = \bar{x}^T\bar{y}. \tag{3.18}$$

The invertibility of $\bar{x}^T\bar{x}$ yields (2.13), while the optimization problem (2.18), (2.20) with (2.13) has the solution in (2.14).
□

The target function (2.19) is defined as a sum of squared differences between the empirical prediction y_k and the theoretical prediction $r_\theta(x_k)$. This is referred to as the **squared error** because it quantifies the squared errors incurred when using the theoretical predictions rather than the empirical ones. Additionally, the problem defined in (2.17) and (2.19) is known as the **least squares problem**, and the resulting estimator in (2.13) is called the **ordinary least squared (OLS) estimator**. The equation (2.13) defines the unique minimizer because the target function (2.21) is strictly **convex**. Indeed, the Hessian of (2.21) is equal to $\bar{x}^T\bar{x}$ that is a full-rank, symmetric and positive-definite matrix under the hypotheses of Proposition 2.1.

Multiplying (2.13) by $\bar{x}$, one obtains that

$$\bar{x}\hat{\theta} = \bar{x}(\bar{x}^T\bar{x})^{-1}\bar{x}^T\bar{y}. \tag{3.19}$$

The geometric interpretation of (2.23) is significant, as the vector $\bar{x}\hat{\theta}$ is the **orthogonal projection** of N-dimensional vector $\bar{y}$ in the d-dimensional column space of $\bar{x}$. The reader can prove that the matrix

$$P = \bar{x}(\bar{x}^T\bar{x})^{-1}\bar{x}^T \tag{3.20}$$

is a projection, i.e., $P^2 = P$.

The following proposition illustrates some properties of estimators (2.13).

Proposition 3.2 *The estimator in (2.13) has the following properties:*

- *The estimator is unbiased and distributed according to $\mathcal{N}(\theta, COV(\hat{\theta}_N))$, where*

$$COV(\hat{\theta}_N) = \hat{\sigma}_N^2(\bar{x}^T\bar{x})^{-1}. \tag{3.21}$$

- *The estimator has variance*

$$VAR(\hat{\theta}_N) = \hat{\sigma}_N^2\mathrm{tr}((\bar{x}^T\bar{x})^{-1}) \sim d/N. \tag{3.22}$$

- *The estimator is consistent:* $MSE(\hat{\theta}_N) = VAR(\hat{\theta}_N) \to 0$ *if* $N \to \infty$.

What we have illustrated in Proposition 2.1 is the first machine learning procedure of these notes. The outputs of the procedure are the estimators in (2.13) and (2.14). But more important for applications is the estimator (2.13) that contributes to the definition of the regression function through (2.6).

We now focus on some properties of the regression function $r_\theta(x)$ and illustrate an important relation which is very similar to the decomposition that we illustrated in Proposition 1.1. The main idea is noting that for any fixed value x the quantity $r_{\hat{\theta}_N}(x)$ is a random variable that **estimates** the true value of the regression function, that we denote as $r(x)$. So, we can expect that there exists a similar relation to (1.8) that is applied to the regression function.

Proposition 3.3 *Let* $D = ((x^{(1)}, y^{(1)}), \ldots, (x^{(N)}, y^{(N)}))$ *be an i.i.d. data set and consider the regression problem of Definition 2.2. For any fixed value* x *the quantity* $r_{\hat{\theta}_N}(x)$ *is an estimator of the true value of the regression function* $r(x) = E(Y|X = x)$*, corresponding to the estimator* $\hat{\theta}_N$*. The MSE of the estimator* $r_{\hat{\theta}_N}(x)$ *is given by:*

$$\begin{aligned} MSE(r_{\hat{\theta}_N}(x), r(x)) &= E_D(||r_{\hat{\theta}_N}(x) - r(x)||^2) \\ &= ||E_D(r_{\hat{\theta}_N}(x)) - r(x)||^2 + E_D(||E_D(r_{\hat{\theta}_N}(x)) - r_{\hat{\theta}_N}(x)||^2) \\ &= ||BIAS(r_{\hat{\theta}_N}(x), r(x))||^2 + VAR(r_{\hat{\theta}_N}(x)), \end{aligned} \tag{3.23}$$

where $E_D = E_{X_1, Y_1, \ldots, X_N, Y_N}$.

It is easy to see that the variance in (2.27) is proportional to the variance of $\hat{\theta}_N$ in Proposition 2.1. Then, if the variance of $\hat{\theta}_N$ is high, we expect a higher variance in (2.27) and lower performance of the regression function $r_{\hat{\theta}_N}(x)$ when predicting **unseen** data (x, y), i.e., data not belonging to the training set. Recall that the estimators are random variables, and in the case of high variance, it is probable that they exhibit significant variability across different data sets.

Let us explain this concept in more depth, as it is crucial for the practical application of (linear) regression. Consider two data sets $D = ((x^{(1)}, y^{(1)}), \ldots, (x^{(N)}, y^{(N)}))$, and $D' = ((x'^{(1)}, y'^{(1)}), \ldots, (x'^{(N)}, y'^{(N)}))$. The first data set D is used to resolve the regression problem and obtain the estimator $\hat{\theta}_N$ using formula (2.13). We could think to assess the quality of the estimator $\hat{\theta}_N$ by comparing the prediction of the regression function $r_{\hat{\theta}_N}(x'^{(i)})$ with the actual value $y'^{(i)}$ for $i = 1, \ldots, N$; this procedure is referred to as **model validation** in the literature, and it is a popular method for verifying the performance of fitted models. If the variance of $r_{\hat{\theta}_N}(x)$ is high, we expect that the validation will reveal significant discrepancies between the predictions and the actual values of D'. This occurs because it is "unlikely" that the correct value of the parameters θ describing data D' is still $\hat{\theta}_N$ which were obtained from D; therefore, the predictions are not optimal for data outside of the training set.

When the trained model can accurately describe the training data but predicts poorly for unseen data, the model is said to be **overfitted**, and this case corresponds to low bias and high variance. On the other side, when the trained model can accurately describe both the training data and unseen data, the model is said to have good **generalization** power. Another possible situation is that the trained model cannot accurately describe the training data and possibly can describe correctly unseen data (this is not strange as it can appear): the model is said to be **underfitted**, and this case corresponds to a high bias and low variance of the estimator. In the case of linear regression, one can have underfitting if the linearity of the regression function is not enough to capture the relations between observed data; this implies that the hypotheses of Definition 2.3 and the conclusions of Proposition 2.2 are not valid anymore. Indeed, these last two results are valid if one can assume that the linear form is suitable **at least** for data in the data set D.

The Proposition 2.2 facilitates a deeper understanding of how overfitting manifests in the context of linear regression. Specifically, it illustrates how the variance of the estimator $\hat{\theta}_N$ in (2.13) depends on the ratio d/N, where d is the dimension of the parameter space and N is the size of the data set.

Then, it is clear that if d is equal to data set size N, the variance reaches its maximum (recall that $d \leq N$ according to Proposition 2.1). Indeed, in this case, the optimization problem in (2.21) becomes an ordinary linear system problem that can be solved through algebraic methods if the matrix $\bar{x}$ is invertible, giving the exact solution $\hat{\theta}_N = \bar{x}^{-1}y$ and a vanishing value for target, $R(\hat{\theta}_N) = 0$; therefore, the process of training **memorizes** exactly the data, and this leads to severe overfitting and unoptimal generalization on unseen data. Then, the practical recipe to have an effective linear regression function is to have more data than parameters, in order to reduce the ratio d/N. This is clearly intuitive: the more data that are available, the more the probability of learning a complete view of the data generation process.

In the general case when $d > N$, the matrix $\bar{x}^T\bar{x}$ is not invertible and the calculation of the estimator must be done numerically from (2.22). Non-invertibility implies that row vectors in (2.12) are **linearly dependent (or collinear)**. The thumb rule about d/N can be applied also to this case and suggests that overfitting is related to collinearity of feature vectors.

In a more general context, the rule of thumb derived from the ratio d/N is generalized by introducing the concept of **complexity** (or **expressiveness**) of the regression function (or hypothesis) r. If the complexity of r is too high (e.g. the function r is not linear but rather a high-degree polynomial or it is a highly non-linear function, as one encounters in **deep learning** with **neural networks**), the phenomenon of overfitting can occur when the data number N is much less than this complexity. Unfortunately, in these more complex contexts, the complexity of a model is not directly related to the number of parameters and it is quite challenging to estimate the complexity of more intricate inference problems, such as those associated with deep learning. The complexity can also depend on **algorithms** used to obtain estimators.

A general principle that characterizes the behavior of bias and variance of estimators is called the **bias-variance trade-off** and it is quantitatively expressed in Proposition 2.3: according to this principle, an overfitted model corresponds to low-bias and high-variance estimators, while an underfitted model corresponds to high-bias and low-variance estimators. Effective models must be able to handle these two opposite effects in order to reduce the MSE of estimators.

To reduce the phenomenon of overfitting in regression, in the literature were introduced modified versions of regression that are **regularized**. Let us define this more precisely.

Definition 3.4 *Let $D = ((x^{(1)}, y^{(1)}), \dots, (x^{(N)}, y^{(N)}))$ be an i.i.d. data set. The regression problem defined in Definition 2.3 is called **regularized (or L^2-regularized, or ridge)** if one assumes the following form for the probability distribution in (2.4):*

$$f_\theta(y|x;\sigma^2) := \alpha\mathcal{N}(y; r_\theta(x), \sigma^2)f(\theta, \sigma^2), \tag{3.24}$$

where α is an unimportant normalization constant and f is defined as follows:

$$f(\theta, \sigma^2) := \mathcal{N}(\theta; 0, \lambda^{-1}I_{d\times d}) = \frac{1}{\sqrt{2\pi\lambda^{-d}}}\exp(-||\theta||^2\lambda/2). \tag{3.25}$$

*In (2.29) $\lambda > 0$ is a constant termed **regularization hyperparameter**, and $I_{d\times d}$ is the $d \times d$ identity matrix, where d is the dimension of parameter space $\Theta \ni \theta$.*

The interpretation of (2.28) may seem obscure within the frequentist approach to inference that we have adopted, but it is clear in **Bayesian** approach where (2.28) serves as the **prior** of θ and σ^2. The regularized linear regression problem, as defined, has a closed-form solution, which is illustrated by the following proposition.

Proposition 3.4 *Consider the regularized regression problem defined in Definition 2.4 with $m = 1$, and with a linear regression function as defined in (2.6) and (2.9). Define the matrix quantities as in (2.10), (2.11), and (2.12). Then, the maximum likelihood method gives the following estimator for $\theta \in \Theta \subseteq \mathbb{R}^d$:*

$$\hat{\theta}_N^{(\lambda)} = (\bar{x}^T\bar{x} + N\lambda I_{d\times d})^{-1}\bar{x}^T\bar{y}, \tag{3.26}$$

while the estimator for $\hat{\sigma}_N^{(\lambda)2}$ is the same as (2.14).

Proof. Assume (2.28) and (2.29). To estimate the parameters of the regression function, we use the usual ML method to obtain the LLF (recall (1.12) and (1.13)):

$$\begin{aligned} -LLF_D(\theta,\sigma^2) &= -\sum_{k=1}^{N} \ln \mathcal{N}(y_k; r_\theta(x_k), \sigma^2) - N \ln f(\theta,\sigma^2) \\ &= \frac{N}{2}\ln(2\pi\sigma^2) + \frac{1}{2\sigma^2}\sum_{k=1}^{N}(y_k - r_\theta(x_k))^2 - \frac{N}{2}\ln(\frac{\lambda^d}{2\pi}) + \frac{N\lambda}{2}||\theta||^1. \end{aligned} \tag{3.27}$$

We changed sign as usual and removed the constant α because it is unimportant. Using the MML, the original inference problem was converted into the following function optimization problems

$$\theta(D) = \arg\min_{\theta\in\Theta} R_D(\theta), \tag{3.28}$$

$$\sigma^2(D) = \arg\min_{\theta\in\Theta} S_D(\sigma^2), \tag{3.29}$$

where the target functions are defined as follows:

$$R_D(\theta) = \sum_{k=1}^{N}(y_k - r_\theta(x_k))^2 + \frac{N\lambda}{2}||\theta||^2, \tag{3.30}$$

$$S_D(\sigma^2) = \frac{N}{2}\ln(2\pi\sigma^2) + \frac{1}{2\sigma^2}(R_D(\theta) - N\lambda||\theta||^2/2). \tag{3.31}$$

Using definitions in (2.10), (2.11), (2.12), the function in (2.34) can be rewritten using the Euclidean norm as follows:

$$R_D(\theta) = ||\bar{y} - \bar{x}\hat{\theta}||^2 + N\lambda||\hat{\theta}||^2/1. \tag{3.32}$$

Calculating the matrix gradient of (2.36) with respect to $\hat{\theta}$ and equating it to zero yields the following matrix equation (we absorbed $1/2$ in λ):

$$(\bar{x}^T\bar{x} + N\lambda I_{d\times d})\hat{\theta} = \bar{x}^T\bar{y}. \tag{3.33}$$

The regularization hyperparameter makes the left-hand side matrix in (2.37) always invertible, then one obtains (2.30), while the optimization problem (2.34), (2.35) with (2.30) has the solution in (2.14). □

The resulting estimator in (2.30) is called the **regularized least squared (RLS) estimator** and it is the unique minimizer because the target function (2.34) is strictly **convex**. The following proposition illustrates some properties of the regularized estimator (2.30).

Proposition 3.5 *The estimator in (2.30) has the following properties for all $\lambda \geq 0$:*

- *The estimator is distributed according to $\mathcal{N}(\hat{\theta}_N^{(\lambda)}, COV(\hat{\theta}_N^{(\lambda)}))$, where*

$$E(\hat{\theta}_N^{(\lambda)}) = (\bar{x}^T\bar{x} + N\lambda I_{d\times d})^{-1}\bar{x}^T\bar{x}\theta = \theta - N\lambda(\bar{x}^T\bar{x} + N\lambda I_{d\times d})^{-1}\theta, \tag{3.34}$$

$$COV(\hat{\theta}_N^{(\lambda)}) = \hat{\sigma}_N^{(\lambda)2}(\bar{x}^T\bar{x} + N\lambda I_{d\times d})^{-1}\bar{x}^T\bar{x}(\bar{x}^T\bar{x} + N\lambda I_{d\times d})^{-1}. \tag{3.35}$$

- *The estimator is biased with*

$$BIAS(\hat{\theta}_N^{(\lambda)}) = -N\lambda(\bar{x}^T\bar{x} + N\lambda I_{d\times d})^{-1}\theta \geq BIAS(\hat{\theta}_N) = 0. \tag{3.36}$$

- *The estimator has variance:*

$$VAR(\hat{\theta}_N^{(\lambda)}) = \mathrm{tr}(COV(\hat{\theta}_N^{(\lambda)})) \sim \frac{d}{N(\lambda+1)^2} \leq VAR(\hat{\theta}_N). \tag{3.37}$$

- *The estimator has a **shrinking** Euclidean norm when $\lambda \to \infty$:*

$$||\hat{\theta}_N^{(\lambda)}|| \sim \lambda^{-1}. \tag{3.38}$$

- *The MSE of the estimator has the following qualitative behavior:*

$$MSE(\hat{\theta}_N^{(\lambda)}) \sim \lambda^4 + \frac{d}{N(\lambda+1)^2}. \tag{3.39}$$

The relation (2.42) is related to the form of (2.34). The term proportional to λ acts as a **penalty** term that penalizes estimators with high values of the Euclidean norm. In the literature, this property is called **shrinkage**. While the relations (2.40) and (2.41) illustrate that the bias and the variance are always **not less** and **not greater** that the equivalent for OLS estimator. Moreover, the relation (2.43) is a lucid realization of the **bias-variance trade-off** illustrated before: if λ is too high, the bias is high (low performance on training data) and the variance is low (good performance on unseen data); otherwise, the bias is low (good performance on training data) and the variance is high (low performance on unseen data).