Preface

In broad terms, machine learning is a specialized subfield of artificial intelligence that leverages a diverse set of techniques and draws on knowledge from multiple mathematical domains—including statistics, linear algebra, calculus, and optimization theory—to enable computers to identify patterns, make predictions, and solve complex problems that are often intractable or inefficient to address using traditional, explicitly programmed algorithmic methods. By learning from data and improving performance over time without being explicitly programmed for every specific task, machine learning systems have become fundamental to advancements in various fields such as natural language processing, computer vision, robotics, and decision-making systems.

These brief informal notes aim to concisely—but not exhaustively—illustrate some mathematical aspects of machine learning. The emphasis is on general mathematical conceptualization, attempting to provide a precise formulation of machine learning principles and concepts.

The need to write these notes arises from the fact that the fundamental concepts of the discipline are scattered across various sources—books, papers, and articles—that often employ different terminologies, notations, and modes of reasoning. This diversity can make it challenging for scientists who are not specialists in these areas to develop a unified and coherent understanding of the underlying principles. By consolidating and clarifying these core ideas in a single, accessible resource, these notes aim to facilitate interdisciplinary learning and foster a deeper comprehension of the subject.

Then, these notes aim to bridge the existing gap by offering readers a unified and concise overview of the foundational concepts. Their purpose is to provide a clear and synthetic summary that facilitates understanding and prepares readers for more advanced study. However, they are not meant to substitute authoritative papers and comprehensive textbooks, which provide more in-depth explanations, rigorous proofs, and extensive examples.

These notes do not explain the mathematical fundamentals of the discipline—such as analysis, linear algebra, probability—because it is assumed that the reader is a competent scientist with an appropriate level of mathematical knowledge. This audience may include mathematicians, physicists, engineers, or other professionals who already possess a solid foundation in these areas, allowing the material to focus on the specific arguments pertaining to the discipline.

Additionally, these notes do not address the practical implementation of mathematical techniques through specific programming languages, frameworks, or libraries. Consequently, readers seeking hands-on guidance for coding, algorithm development, or applying these mathematical concepts in real-world software environments may need to consult supplementary resources or tutorials focused on practical programming applications.

Chapter 1

Mathematical principles of machine learning

In a broad sense, statistical inference and (statistical) machine learning share the common objective of understanding, inferring, or learning about an underlying phenomenon or process that is inherently characterized by uncertainty, noise, and randomness. Both fields seek to extract meaningful insights, discover patterns, or make predictions based on information that originates from such complex systems. Typically, the only available information about the phenomenon is encapsulated in a finite set of empirical observations or samples, commonly referred to as **data**, which are generated by the stochastic mechanism governing the phenomenon.

This data-generation process is fundamentally probabilistic and can be modeled as the realization of one or more **random variables**. These variables follow an unknown **probability distribution function (PDF)**, which characterizes the likelihood of various outcomes. A central challenge in both statistical inference and machine learning is to infer properties of or make decisions related to this underlying distribution based on the observed samples. This includes estimating **parameters**, testing hypotheses, modeling dependencies, learning **predictive functions**, or uncovering **latent** structures.

Moreover, the inherent randomness and noise in the data necessitate robust methodologies that can **generalize** well beyond the observed samples to new, unseen data points. Thus, both statistical inference and machine learning rely heavily on probabilistic modeling, approximation techniques, and computational algorithms to handle uncertainty and make reliable conclusions in the face of limited and imperfect data.

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 (statistical machine learning)*** *problem consists of inferring (learning) the distribution that generated the data; that is, the probability distribution function (PDF) f that describes both the observed data and the not-yet-observed data,*

$$X_1, \ldots, X_N \sim f.$$

The last definition highlights that statistical inference and machine learning can be viewed as two sides of the same coin because, fundamentally, they share the goal of extracting knowledge from data, but they approach this goal with different emphases and methods. Statistical inference primarily focuses on understanding and quantifying relationships in the data to draw conclusions about a larger population. In contrast, machine learning emphasizes building flexible **algorithms** that can make accurate predictions on new, unseen data, often prioritizing predictive performance and **generalization**. In this chapter, we will illustrate how these two (apparently) divergent approaches can be reconciled.

The Definition 1.1 is fully general and flexible, allowing the distribution f to represent any probability distribution relevant to the problem under investigation. This distribution could correspond to the probability distribution of a single random variable X, f_X, or it could be a **joint** distribution encompassing multiple variables, or a **conditional** distribution specifying the dependencies between variables. Throughout these notes, we will examine which type of distribution is most appropriate and useful depending on the specific machine learning problem or application being addressed.

It is important to emphasize that Definition 1.1 serves to characterize the general nature of the mathematical problems encountered in machine learning. However, it does not specify the precise mathematical formulation or the methods used to solve these problems. The actual formulation and solution techniques depend on the context and objectives.

Usually, the distribution f is inferred (learned) by searching within a fixed set of probability density functions, known as a **statistical model**. The inference problem is called **parametric** if any distribution in the statistical model $f(\theta)$ 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 inference foundation

In what follows, we focus primarily on **parametric** problems and make the standard assumption that the random variables under consideration are **independent and identically distributed (i.i.d.)**, meaning each observation is drawn from the same probability distribution independently of the others. This assumption simplifies the analysis and is foundational for many theoretical results in statistical inference. Our approach is **frequentist**, which treats the parameters θ as fixed but unknown quantities. In this framework, data are viewed as random since they are subject to variability from the data generating process, but the parameters themselves do not vary; they represent the true, unknown characteristics of the underlying distribution.

The general problem outlined in Definition 1.1 can be formulated in a principled manner by defining the concept of **risk**.

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 sample space Ω. Consider the statistical model $\{f(\theta) \mid \theta \in \Theta \subseteq \mathbb{R}^d\}$. The **risk** (or **test error**) of the model distribution $f(\theta)$ with respect to the true distribution f_X is defined as*

$$R(f_X, f(\theta)) := E_X[L(f_X(X), f(X;\theta))] = \int_{X(\Omega)} L(f_X(x), f(x;\theta)) f_X(x) dx, \tag{1.1}$$

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

The risk function defined in (1.1) quantifies the expected error or loss incurred when approximating the true data distribution f_X by a model distribution $f(\theta)$. Specifically, it represents the average discrepancy between the model $f(\theta)$ and the true distribution f_X under a given loss function, measuring how well the model $f(\theta)$ performs in representing the underlying data generating process. Determining the optimal model distribution $f(\theta)$ then entails **minimizing** this risk function, thereby selecting the parameter θ that yields the smallest value.

By selecting a loss function with a logarithmic form, one can define the so-called **Kullback-Leibler (KL) divergence**, which is a fundamental measure of the difference between two probability distributions.

Definition 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 sample space Ω. Consider the statistical model $\{f(\theta) \mid \theta \in \Theta \subseteq \mathbb{R}^d\}$. The **Kullback–Leibler (KL) divergence** of the model distribution $f(\theta)$ with respect to the true distribution f_X is defined as*

$$D^{(KL)}(f_X || f(\theta)) := E_X \left[\ln \left(\frac{f_X(X)}{f(X;\theta)} \right) \right] = \int_{X(\Omega)} \ln \left(\frac{f_X(x)}{f(x;\theta)} \right) f_X(x) dx. \tag{1.2}$$

The KL divergence defined in (1.2) 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. 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 some respects, as it satisfies the following properties:

$$D^{(KL)}(f||g) \geq 0, \quad D^{(KL)}(f||g) = 0 \iff f = g. \tag{1.3}$$

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

The evident drawback of the risk defined in (1.1) is that it cannot be calculated when the true distribution f_X is unknown. An alternative measure of risk that leverages the available data is the **empirical risk**, which is defined as the average loss computed over the observed sample.

Definition 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 sample space Ω, realized by the data set $D = \{x_1 \ldots, x_N\} \subset (\mathbb{R}^n)^N$. 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 model distribution $f(\theta)$ with respect to the true distribution f_X is defined as*

$$\hat{R}_N(f_X, f(\theta))(D) := \frac{1}{N} \sum_{k=1}^{N} L(f_X(x_k), f(x_k;\theta)), \tag{1.4}$$

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

The empirical risk defined in (1.4) encapsulates information derived from the observed data set D and, importantly, is a function solely of the model parameters θ since the data set itself is considered fixed. This empirical risk acts as a practical surrogate for the true risk defined in equation (1.1), which generally involves the unknown underlying data distribution and is, therefore, intractable. Under the assumption of a suitably chosen loss function L, minimizing the empirical risk (1.4) with respect to θ yields an approximating $f(\theta)$ that represents the best possible model.

This optimization strategy is known as **empirical risk minimization (ERM)** and forms a foundational principle in machine learning. ERM provides a coherent and general framework for constructing **estimators**, which are mappings from observed data to estimated model parameters.

Before proceeding, let us formally define the notion of an estimator and further explore its role within statistical inference.

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 sample space Ω. Consider the model $\{f(\theta) \mid \theta \in \Theta \subseteq \mathbb{R}^d\}$. A **statistic** is a random variable $V_N(X_1, \ldots, X_N)$ such that $V_N : (\mathbb{R}^n)^N \to \mathbb{R}^d$ is a measurable function. If the image of V_N 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 variables that depend on the data used to train the model. More precisely, from a data set consisting of observed values $x_1, \ldots, x_N$, the estimator $\hat{\theta}_N$ gives the estimation $\hat{\theta}_N(x_1, \ldots, x_N) \in \Theta$. Because the data themselves are random, the estimator can take on different values for different realizations of the data.

Estimators play a central role in statistical machine learning, as they are the quantities we seek to infer or learn based on observed data. Through statistical machine learning algorithms, the learning process is essentially a form of **statistical inference**. These algorithms use observed data to construct estimators that approximate unknown parameters or functions characterizing the underlying data-generating process.

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 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*

$$\hat{\theta}_N^{ERM}(D) \in \arg\min_{\theta\in\Theta} \hat{R}_N(f_X, f(\theta))(D). \tag{1.5}$$

The ERM method, introduced as the first **optimization** problem in these notes, establishes a fundamental link between machine learning and **optimization theory**. ERM formalizes the learning task as an optimization problem in which the goal is to find an estimator that minimizes the empirical risk, computed over the training data. By minimizing the empirical risk, we effectively find parameters that perform well on the observed data, with the goal—under appropriate conditions of regularity and sufficient sample size—of generalizing well to new, unseen data.

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 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 (1.5), along with the* ***empirical KL divergence***

$$\hat{D}_N^{(KL)}(f_X||f(\theta))(D) := \frac{1}{N}\sum_{k=1}^{N} \ln\left(\frac{f_X(x_k)}{f(x_k;\theta)}\right) = -\frac{1}{N}\sum_{k=1}^{N} \ln f(x_k;\theta) + \text{const}, \tag{1.6}$$

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

Combining equations (1.6) and (1.5) yields the following optimization problem, which determines the so-called **maximum likelihood (ML)** estimator:

$$\hat{\theta}_N^{ML}(D) \in \arg\max_{\theta\in\Theta}\left(\sum_{k=1}^{N} \ln f(x_k;\theta)\right) = \arg\max_{\theta\in\Theta}\left(\prod_{k=1}^{N} f(x_k;\theta)\right). \tag{1.7}$$

The two objective functions in (1.7) are referred to as the **log-likelihood function (LLF)** and the **likelihood function (LF)**, respectively. The second equality in (1.7) follows from the monotonicity of the logarithm function, which ensures that the LF and the LLF attain their maximum values at the same points.

The LF can be interpreted as the probability of observing the data D given the parameters θ. When θ is unknown, a natural and widely used approach for estimation is to find the parameter values that **maximize** this probability, thus making the observed data most probable under the model. In the case of i.i.d. data, this probability is given by the likelihood function. These ideas described here underlie the estimation problem formulated in equation (1.7).

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*

$$E_{X_1,\ldots,X_N}[\hat{\theta}_N] := \int_{X_1(\Omega)\times\cdots\times X_N(\Omega)} \hat{\theta}_N(x_1, \ldots, x_N) f_X(x_1)\cdots f_X(x_N) dx_1 \cdots dx_N. \tag{1.8}$$

The joint probability distribution function (PDF) in (1.8) is the product of marginal probability distribution functions f_X because $X_1, \ldots, X_N$ are i.i.d..

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

$$BIAS(\hat{\theta}_N, \theta_*) := E_{X_1,\ldots,X_N}[\hat{\theta}_N] - \theta_*, \tag{1.9}$$

*where $\theta_* \in \Theta$ is the true value of the parameters. A point estimator is said to be **unbiased** if its bias is zero.*

The bias measures how far off, on average, the estimator's predictions are from the true parameter value it is trying to estimate. An unbiased estimator over many repeated samples hits the true parameter value on average. Conversely, an estimator with nonzero bias can overestimate or underestimate the parameter θ.

Heuristically, unbiased estimators are generally more desirable because they do not systematically deviate from the true parameter, making them more accurate on average. However, unbiasedness alone does not guarantee that an estimator is overall better. Other properties like **variance** and **mean squared error (MSE)** also matter. Sometimes, a slightly biased estimator with much lower variance can lead to more reliable estimates in practice.

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

$$\begin{aligned} VAR(\hat{\theta}_N) &:= E_{X_1,\ldots,X_N}[||\hat{\theta}_N - E_{X_1,\ldots,X_N}[\hat{\theta}_N]||^2] \\ &= \sum_{k=1}^{d} E_{X_1,\ldots,X_N}[(\hat{\theta}_k - E_{X_1,\ldots,X_N}[\hat{\theta}_k])^2] \\ &= \sum_{k=1}^{d} VAR(\hat{\theta}_k) = \operatorname{tr} COV(\hat{\theta}_N), \end{aligned} \tag{1.10}$$

*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*

$$MSE(\hat{\theta}_N, \theta_*) := E_{X_1,\ldots,X_N}[||\hat{\theta}_N - \theta_*||^2]. \tag{1.11}$$

The MSE is a widely used and reasonable criterion for measuring the performance of an estimator because it quantifies the average squared difference between the estimated value and the true parameter being estimated. If the MSE is small, it indicates that, on average, the estimator produces values close to the true value, reflecting good estimation accuracy.

One key and interesting property of the MSE is its decomposition into a bias and variance component, 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:*

$$MSE(\hat{\theta}_N, \theta_*) = ||BIAS(\hat{\theta}_N, \theta_*)||^2 + VAR(\hat{\theta}_N). \tag{1.12}$$

As we mentioned before, this result highlights why unbiased estimators are often considered good candidates in statistical estimation. However, it is crucial to emphasize that zero (or small) bias alone does not guarantee that an estimator is of high quality. Even if an estimator is unbiased, it can still have high variance. High variance implies that the estimator's predictions can vary widely depending on the particular data set used, leading to imprecise and unstable estimates. Such fluctuations make the estimator less reliable in practice, especially when working with finite or small sample sizes.

Therefore, a good estimator is one that not only has **low bias** (accuracy on average) but also exhibits **low variance** (consistency and stability). This combination ensures that the estimator reliably produces values close to the true parameter across different samples, minimizing both systematic error and random fluctuations. In the machine learning literature, an estimator with these properties is said to have good **generalization** properties. Informally, an estimator is said to **generalize** well in describing new, unseen data if it has a small mean squared error (MSE).

Further, by applying **Markov inequality** to the non-negative random variable $||\hat{\theta}_N - \theta_*||^2$, it is straightforward to show that the MSE quantifies the probability of 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 $\epsilon > 0$*

$$P(||\hat{\theta}_N - \theta_*|| > \epsilon) \leq \frac{MSE(\hat{\theta}_N, \theta_*)}{\epsilon^2}. \tag{1.13}$$

The meaning of this result is clear: if an estimator has a low MSE, then the probability that it produces an estimate far from the true parameter 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_*:*

$$\lim_{N\to\infty} P(||\hat{\theta}_N - \theta_*|| > \epsilon) = 0. \tag{1.14}$$

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 arbitrarily close to the true parameter θ_* with high probability. However, it is important to note that the converse is not true in general: consistency (convergence in probability) does not necessarily imply that the MSE converges to zero.

We conclude this section by examining the properties of the empirical risk (1.4) and exploring in which sense it serves as a good approximation of the true risk (1.1). 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 (1.1) and is defined as the **empirical mean** of the loss function $L(f_X(X), f(X;\theta))$, considered as a random variable:*

$$\hat{R}_N(f_X, f(\theta)) := \overline{L(f_X(X), f(X;\theta))}_N = \frac{1}{N}\sum_{k=1}^{N} L(f_X(X_k), f(X_k;\theta)). \tag{1.15}$$

It is evident that the empirical risk in (1.4) is a realization of the empirical risk estimator in (1.15), 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 (1.15) is an **unbiased** and **consistent** estimator of true risk (1.1):*

$$E_{X_1,\ldots,X_N}[\hat{R}_N(f_X, f(\theta))] = R(f_X, f(\theta)), \tag{1.16}$$

$$\lim_{N\to\infty} P(|\hat{R}_N(f_X, f(\theta)) - R(f_X, f(\theta))| > \epsilon) = 0. \tag{1.17}$$

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

$$\lim_{N\to\infty} P\left(\left|\frac{1}{N}\sum_{k=1}^{N} L(f_X(X_k), f(X_k;\theta)) - E_X[L(f_X(X), f(X;\theta))]\right| > \epsilon\right) = 0. \tag{1.18}$$

Consequently, the empirical risk estimator defined in (1.15) is ***consistent****.*

1.2 Statistical learning theory foundation

This section is dedicated to the perspective of **statistical learning theory** in machine learning, which is a complementary mathematical framework that analyzes how algorithms can learn predictive functions from data, instead of resolving inferential tasks. We explore this perspective specifically through the framework of **probably approximately correct (PAC)** theory.

Roughly speaking, the goal of learning theory is not to infer (learn) the distribution that generated the data in D, $f_{X,Y}$, but to use the data to learn quantitative relationships between the random variables X and Y, defined on the sample space Ω. These relations are expressed as functions (called **hypotheses**) $h : \mathcal{X} \to \mathcal{Y}$, where $\mathcal{X} \subseteq X(\Omega)$ and $\mathcal{Y} \subseteq Y(\Omega)$. It is assumed that the space of hypotheses H is a subset of all **measurable** functions from $\mathcal{X}$ to $\mathcal{Y}$, denoted by $M(\mathcal{X}, \mathcal{Y})$.

PAC learning theory formalizes the conditions under which a learning algorithm can, with high probability, find a prediction function that approximates the true function well within a specified error margin, balancing between accuracy and confidence.

In addition, 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. 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.

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 are related. Let us formalize the learning problem in the learning theory.

Definition 1.14 *Let $D = \{(x_1, y_1), \dots, (x_N, y_N)\} \subset \mathcal{X} \times \mathcal{Y}$ 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 $h(\theta)$ predicts the corresponding output $y = h(x;\theta) \in \mathcal{Y}$ 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.

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 $D = \{(x_1, y_1), \dots, (x_N, y_N)\} \in (\mathcal{X} \times \mathcal{Y})^N$, the algorithm learns the hypothesis $A_H(D) := h(\hat{\theta}_N(D)) \in H$.

Definitions 1.14 and 1.15 are somewhat vague on how to determine the best hypothesis $h(\theta)$ 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 $(X_1, Y_1), \dots, (X_N, Y_N) \sim f_{X,Y}$ 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 **test error**) of the hypothesis $h(\theta)$ is defined as*

$$R(h(\theta)) := E_{X,Y}[L(h(X;\theta), Y)] = \int_{\mathcal{X}\times\mathcal{Y}} L(h(x;\theta), y) f_{X,Y}(x, y) dx dy, \tag{1.19}$$

*where $L : \mathcal{Y}^2 \to \mathbb{R}$ is the **loss** function.*

The optimal value of $h(\theta)$ can then be determined by **minimizing** this risk:

$$h(\theta^*) \in \arg \min_{h(\theta)\in H} R(h(\theta)). \tag{1.20}$$

$$\theta^* \in \arg \min_{\theta\in\Theta} R(h(\theta)). \tag{1.21}$$

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*

$$h^* \in \arg \min_{h\in M(\mathcal{X},\mathcal{Y})} R(h). \tag{1.22}$$

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

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

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

Definition 1.18 *Let $(X_1, Y_1), \dots, (X_N, Y_N) \sim f_{X,Y}$ 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** (also called **empirical error** or **training error**) is an **estimator** of true risk (1.19) and is defined as the **empirical mean** of the loss function $L(h(X;\theta), Y)$, considered as a random variable:*

$$\hat{R}_N(h(\theta)) := \overline{L(h(X;\theta), Y)}_N = \frac{1}{N} \sum_{k=1}^{N} L(h(X_k;\theta), Y_k). \tag{1.24}$$

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. This implies that $E_{X_1,Y_1,\dots,X_N,Y_N}[\hat{R}_N(h(\theta))] = R(h(\theta))$.

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 learns the minimizing hypothesis (named* ***empirical risk hypothesis (ERH)****) as follows:*

$$ERM_H(D) := \hat{h}_N^{ERM}(D) := h(\hat{\theta}_N^{ERM}(D)) \in \arg\min_{h(\theta)\in H} \hat{R}_N(h(\theta))(D), \tag{1.25}$$

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

$$\hat{\theta}_N^{ERM}(D) \in \arg\min_{\theta\in\Theta} \hat{R}_N(h(\theta))(D). \tag{1.26}$$

The previous definition illustrates how the parameter values corresponding to the minimizing hypothesis can be interpreted as an **estimator**, because these values depend on the specific data set D and thus vary randomly with the data. However, the key subtlety here is that the estimators obtained in this way do not necessarily correspond in a straightforward way to parameters of a well-defined 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:*

$$P(R(\hat{h}_N) - R^* \leq \epsilon) \geq 1 - \delta, \tag{1.27}$$

where $\hat{h}_N := h(\hat{\theta}_N) := A_H(X_1, Y_1, \ldots X_N, Y_N)$ is the (random) hypothesis learned by A_H from H. If A_H returns $\hat{h}_N$ in a time proportional to $q_H(1/\epsilon, 1/\delta)$, then H is said to be ***efficiently agnostic PAC learnable****. The (random) quantity $\Delta(\hat{h}_N) := R(\hat{h}_N) - R^* \geq 0$ is named* ***excess risk****.*

Relation (1.27) is central to learning theory because it defines what constitutes effective learning and **generalization** with a **finite** data set D. The most important peculiarity is that it involves two accuracy parameters. The accuracy parameter ϵ 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 and generalization requirements (this corresponds to the "probably" part).

The learnability and generalization properties do not depend on distribution $f_{X,Y}$, but depend 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 and generalization, corresponding to **small** values of ϵ and **small** values of δ, imply a **greater** value of N. In other words, the PAC learnability of a hypothesis space H guarantees that the optimal learned hypothesis $\hat{h}_N \in H$ generalizes with error at most ϵ (that is $\Delta(\hat{h}_N) \leq \epsilon$) and probability at least $1-\delta$ if it is obtained from a data set D whose size $N \geq q_H(1/\epsilon, 1/\delta)$.

Proposition 1.5 *The excess risk can be bounded as*

$$\begin{aligned}\Delta(\hat{h}_N) &\leq R_{opt} + 2R_{gen} + R_{app} \\ &\leq R_{opt} + 2R'_{gen} + R_{app}\end{aligned} \tag{1.28}$$

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:*

$$P\left(\sup_{h(\theta)\in H} |R(h(\theta)) - \hat{R}_N(h(\theta))| \leq \kappa(\delta, N)\right) \geq 1 - \delta, \tag{1.29}$$

$$P\Big(|R(\hat{h}_N) - \hat{R}_N(\hat{h}_N)| \leq \kappa(\delta, N)\Big) \geq 1 - \delta. \tag{1.30}$$

The generalization error and the generalization bound are essential tools for bounding the excess risk, which is key to analyzing the **PAC learnability** of a hypothesis space and **generalization** properties of learned hypothesis $\hat{h}_N$. Indeed, from inequality (1.28), one can reduce the optimization error R_{opt} and approximation error R_{app} 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 (R_{gen} or R'_{gen}) becomes the dominant contribution to the excess risk bound. This implies that any generalization analysis can be conducted on this quantity.

From (1.30), we can estimate the sample complexity threshold $q_H(1/\epsilon, 1/\delta)$ by solving for N in the following inequality:

$$\kappa(\delta, N) \leq \epsilon. \tag{1.31}$$

In some cases, it is possible to provide an inferential interpretation of the risk minimization parameter (1.21) and its empirical counterpart (1.26) as follows.

Proposition 1.6 *The learning theory parameters $\theta \in \Theta$ can be interpreted as the parameters of a model distribution $f(x, y; \theta)$ 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 (1.19) coincides with the analogous quantity in inference theory:*

$$L(h(x;\theta), y) = L_{inf}(f_{X,Y}(x, y), f(x, y; \theta)), \tag{1.32}$$

where the model distribution $f(x, y; \theta) := F(x, y, h(\theta))$ depends on θ through the dependence on $h(\theta)$, and L_{inf} is the inferential loss function introduced in (1.1).

Then, the **risk minimization principle** applied to (1.19) with loss (1.32) connects the learning theory to the inference theory. For example, the **KL divergence** (1.2) can be obtained if we can construct a loss function as

$$L(h(x;\theta), y) = \ln f_{X,Y}(x, y) - \ln f(x, y; \theta). \tag{1.33}$$

From (1.33), we conclude that the **ML estimator** (1.7) coincides with the learning theory **ERM estimator** (1.26) if we can define empirical risk as

$$\hat{R}_N(h(\theta))(D) := \frac{1}{N}\sum_{k=1}^{N}(-\ln f(x_k, y_k; \theta)). \tag{1.34}$$

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 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**.

Using the i.i.d. data set $D = \{(x_1, y_1), \dots, (x_N, y_N)\} \subset \mathbb{R}^n \times C$, 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 $y = c(x) \in C$ for a new pattern $x \in \mathbb{R}^n$. 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 inference foundation

At first glance, the classification 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 $(X_1, Y_1), \dots, (X_N, Y_N)$, 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 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 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*** *consists of inferring (learning) the parameters θ of the statistical model*

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

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

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

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, 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*** *consists of inferring (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

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

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

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

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 $x \in \mathbb{R}^n$ is assigned to class $y \in C$ if $x \in \mathcal{R}_y$.

The previous definition allows us to observe that the classification 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 definitions 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 *The classification and the statistical model (2.4) are called* ***discriminative*** *if the goal is to model directly the conditional probability using a suitable parametric function, $f(y|x;\theta) := F(x,y;\theta)$. They are called* ***generative*** *if the goal is to model the conditional probability by applying Bayes' rule to the joint distribution $f_{X.Y}(x,y;\theta)$ as follows:*

$$f(y|x;\theta) \propto f_{X.Y}(x,y;\theta) = f_Y(y;\theta) f(x|y;\theta), \tag{2.10}$$

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 $f_{X.Y}(x,y;\theta)$ and then derive the conditional probability $f(y|x;\theta)$ via Bayes' rule, while discriminative models estimate the conditional probability $f(y|x;\theta)$ directly through a function $F(x,y;\theta)$. Generative models are called so because, once the joint distribution $f_{X.Y}(x,y;\theta)$ 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 the 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*** *consists of inferring (learning) from data in D the parameters θ of the* ***Bernoulli*** *statistical model*

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

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

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

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

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

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

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

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

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

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

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 $b(x;\theta)$. This approach to binary classification is known as **binary logistic classification** (or **binary logistic regression** in some references). In the special case where $b(x;\theta)$ 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** with a Bernoulli distribution defined in (2.11), where*

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

$$f(y^{(2)}|x;\theta) = 1 - f(x;\theta) = \frac{1}{1+\exp(b(x;\theta))}, \tag{2.18}$$

$$b(x;\theta) := \theta^{(0)} + \sum_{k=1}^{n} \theta^{(k)} x_k. \tag{2.19}$$

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

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

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

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

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

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

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

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

$$c(x;\theta) = \frac{|b(x;\theta)|}{b(x;\theta)}. \tag{2.24}$$

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 $(\theta^{(1)},\dots,\theta^{(n)}) \in \mathbb{R}^n$, 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 $b(x;\theta)$. For example, using **quadratic** functions in x, one can describe decision boundaries represented by **hyperconics** such as hyperspheres, hyperellipsoids, hyperparaboloids, and so on:

$$b(x;\theta) := \sum_{k=1}^{n}\sum_{j=1}^{n} q_{kj}(\theta)(x^{(k)} - a^{(k)})(x^{(j)} - a^{(j)}), \tag{2.25}$$

where the coefficients q_{kj} depend on the parameters θ, and $(a^{(1)},\dots,a^{(n)}) \in \mathbb{R}^n$ 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*** *with the following* ***logistic*** *statistical model:*

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

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

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

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

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

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

$$c(x;\theta) = \arg\max_{y \in C} b_y(x;\theta). \tag{2.29}$$

Let us now consider an important example of generative classification 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 defined by the following statistical model:*

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

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 distribution 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:

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

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 (QDA)**. 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 (LDA)**. The term "discriminant" can be quite confusing because the model is generative, not discriminative.

Another example of generative classification is the so-called **naive Bayes** 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 defined by the following statistical model:*

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

*where α is the normalization factor, $\Pi(y;\theta)$ 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:

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

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

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

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

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

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 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** 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 $h(\theta)$ predicts the corresponding label $y = h(x;\theta) \in \mathcal{Y}$ 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 the classification 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 follows.

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*

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

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

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

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 binary classification is that learning performance and generalization depend 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 classifying 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 classifications, 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 result 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 and generalization given the accuracy parameters ϵ, δ, and hypothesis complexity d. Furthermore, unlike inference theories that provide asymptotic guarantees on estimators, the 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.

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

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

The quantity $N_{mis}(h(\theta), D)/N$ denotes the rate of **misclassified** data points in D by the hypothesis $h(\theta)$. Then, the ERM method suggests that the best hypothesis is obtained by minimizing the 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 *Consider the classification on the hypothesis space H. 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 bound holds:*

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

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 (1.34) with $f(y|x;\theta)$ defined in (2.17) and (2.18) as estimator distribution, we obtain an alternative loss function used in binary classification, the **logistic loss**.

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

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

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

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

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

Chapter 3

Supervised regression

In this chapter, we discuss the problem of **regression**. The regression 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**, or **label**.

Using the i.i.d. data set $D = \{(x_1, y_1), \ldots, (x_N, y_N)\} \subset \mathbb{R}^n \times \mathbb{R}^m$, 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 $y = r(x) \in \mathbb{R}^m$ for new feature $x \in \mathbb{R}^n$. 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. Then, we assume that the variables X and Y are dependent, and the conditional probability density function $f(y|x)$ is the suitable model.

The following definition formalizes the regression.

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** consists of inferring (learning) the parameters θ of the statistical model*

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

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

$$r(x;\theta) := E_{Y|X=x}[Y] = \int y f(y|x;\theta) dy. \tag{3.2}$$

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** consists of inferring (learning) the parameters θ data in D, such that one assumes the following relation between X and Y:*

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

where $r(\theta) : \mathbb{R}^n \to \mathbb{R}^m$ is named ***regression function****, and ε, called* ***noise****, is a $\mathbb{R}^m$-valued random variable with the following properties ($i = 1, \dots, m$):*

$$E_{\varepsilon_i|X=x}[\varepsilon_i] = 0, \tag{3.4}$$

$$VAR(\varepsilon_i|X = x) = \sigma^2 > 0. \tag{3.5}$$

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**.

An important example of regression is the **ordinary regression**, which can be defined as follows.

Definition 3.3 *Let $D = \{(x_1, y_1), \dots, (x_N, y_N)\} \subset \mathbb{R}^n \times \mathbb{R}^m$ be an i.i.d. data set. The* ***ordinary regression*** *consists of inferring (learning) from data in D the parameters θ of the following* ***Gauss*** *statistical model:*

$$f(y|x;\theta,\sigma) := \mathcal{N}(y; r(x;\theta), \sigma^2) = \frac{1}{\sqrt{(2\pi\sigma^2)^m}} \exp\left(-\frac{||y - r(x;\theta)||^2}{2\sigma^2}\right). \tag{3.6}$$

Equivalently, the ***ordinary regression*** *is a regression problem of Definition 3.2 such that $\varepsilon \sim \mathcal{N}(0, I_{m\times m}\sigma^2)$, where the matrix $I_{m\times m}$ is the identity matrix in $\mathbb{R}^m$.*

The regression is named **linear** if the regression function is linear:

$$r(x;\theta) := \vartheta_0 + \vartheta x, \tag{3.7}$$

where $\vartheta_0 := (\theta_0^{(i)}) \in \mathbb{R}^m$, and $\vartheta := (\theta_j^{(i)}) \in \mathbb{R}^{m\times n}$. The parameter space $\Theta \ni \theta$ has dimension $d = m(n+1)$.

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

Proposition 3.1 *Consider the* ***ordinary linear regression*** *with $m = 1$, and $\dim\Theta =: d = n+1$. Define the following matrices:*

$$\bar{y} := \begin{pmatrix} y_1 \\ \vdots \\ y_N \end{pmatrix} \in \mathbb{R}^N, \tag{3.8}$$

$$\bar{\theta} := \begin{pmatrix} \theta^{(0)} \\ \vdots \\ \theta^{(d-1)} \end{pmatrix} \in \mathbb{R}^d, \tag{3.9}$$

$$\bar{x} := \begin{pmatrix} 1 & x_1^{(1)} & \cdots & x_1^{(d-1)} \\ \vdots & \vdots & \vdots & \vdots \\ 1 & x_N^{(1)} & \cdots & x_N^{(d-1)} \end{pmatrix} \in \mathbb{R}^{N\times d}. \tag{3.10}$$

Suppose that $\bar{x}^T\bar{x} \in \mathbb{R}^{d\times d}$ is invertible, that is $\bar{x}$ has full rank $n+1 = d \le N$. Then, the maximum likelihood method gives the following estimators:

$$\hat{\theta}_N = (\bar{x}^T\bar{x})^{-1}\bar{x}^T\bar{y}, \tag{3.11}$$

$$\hat{\sigma}_N^2 = \frac{1}{N}\sum_{k=1}^{N}(y_k - r(x_k;\hat{\theta}_N))^2 = \frac{1}{N}||\bar{y} - \bar{x}\hat{\theta}_N||^2. \tag{3.12}$$

Proof. To estimate the parameters of the regression function, we use the ML method and (3.6) to obtain the LLF:

$$-LLF_D(\theta,\sigma^2) = -\sum_{k=1}^{N}\ln\mathcal{N}(y_k; r(x_k;\theta),\sigma^2) = \frac{N}{2}\ln(2\pi\sigma^2) + \frac{1}{2\sigma^2}\sum_{k=1}^{N}(y_k - r(x_k;\theta))^2. \tag{3.13}$$

We reversed the sign in order to transform the maximization problem into a minimization problem. Then, the original inference problem was converted into the following optimization problems:

$$\theta(D) = \arg\min_{\theta\in\Theta} R_D(\theta), \tag{3.14}$$

$$\sigma^2(D) = \arg\min_{\theta\in\Theta} S_D(\theta,\sigma^2), \tag{3.15}$$

where the target functions are defined as follows:

$$R_D(\theta) := \sum_{k=1}^{N}(y_k - r(x_k;\theta))^2, \tag{3.16}$$

$$S_D(\theta,\sigma^2) := \frac{N}{2}\ln(2\pi\sigma^2) + \frac{1}{2\sigma^2}R_D(\theta). \tag{3.17}$$

Using definitions in (3.8-10), the function in (3.16) can be rewritten using the Euclidean norm as

$$R_D(\bar{\theta}) = ||\bar{y} - \bar{x}\bar{\theta}||^2. \tag{3.18}$$

Calculating the matrix gradient of (3.16) with respect to $\bar{\theta}$ and equating it to zero yields the following matrix equation:

$$\bar{x}^T\bar{x}\bar{\theta} = \bar{x}^T\bar{y}. \tag{3.19}$$

The invertibility of $\bar{x}^T\bar{x}$ yields (3.11), while the optimization problem (3.15) has the solution given in (3.12).
□

The target function (3.16) is defined as the sum of squared differences between the empirical observations y_k and the theoretical predictions $r(x_k;\theta)$. This is called the **squared error** because it quantifies the squared deviations incurred when using the theoretical predictions instead of the empirical ones. Additionally, the problem defined in (3.14) with (3.16) is known as the **least squares problem**, and the estimator given in (3.11) is referred to as the **ordinary least squares (OLS) estimator**. This value defines the unique minimizer because the target function in (3.18) is strictly **convex**. Indeed, the Hessian is equal to $2\bar{x}^T\bar{x}$, which is full-rank, symmetric, and positive definite under the assumptions of Proposition 3.1.

Multiplying (3.11) by $\bar{x}$, one obtains

$$\bar{x}\bar{\theta} = \bar{x}(\bar{x}^T\bar{x})^{-1}\bar{x}^T\bar{y}. \tag{3.20}$$

The geometric interpretation is significant, as the vector $\bar{x}\bar{\theta}$ 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

$$P = \bar{x}(\bar{x}^T\bar{x})^{-1}\bar{x}^T \tag{3.21}$$

is a projection, that is $P^2 = P$. The following proposition illustrates some properties of the OLS estimator.

Proposition 3.2 *The* ***OLS estimator*** *in (3.11) has the following properties:*

- *The estimator is* ***unbiased:***
$$BIAS(\hat{\theta}_N, \theta_*) = 0. \tag{3.22}$$
- *The variance is*
$$VAR(\hat{\theta}_N) = \sigma^2 E_{X_1,\dots,X_N}[\mathrm{tr}\{(\bar{x}^T\bar{x})^{-1}\}]. \tag{3.23}$$

The variance in (3.23) can be calculated explicitly if additional assumptions are made about the distribution of the data x.

Proposition 3.3 *Let $D = \{(x_1, y_1), \dots, (x_N, y_N)\} \subset \mathbb{R}^n \times \mathbb{R}^m$ be an i.i.d. data set, and consider the* ***ordinary regression****. If one assumes that data x_k, $k = 1, \dots, N$, are distributed according to $\mathcal{N}(0, I_{n\times n})$, then the random matrix $(\bar{x}^T\bar{x})^{-1}$ is distributed according to the* ***inverse-Wishart distribution****, and the variance (3.23) is given as*

$$VAR(\hat{\theta}_N) = \sigma^2\frac{d-1}{N-d} = \sigma^2\frac{n}{N-n-1}, \quad N \geq d = n+1. \tag{3.24}$$

What we illustrated in Proposition 3.1 is the first machine learning procedure presented in these notes. The outputs of this procedure are the estimators given in (3.11) and (3.12). More importantly for applications is the estimator in (3.11), which contributes to defining the regression function.

We now focus on some properties of the regression function $r(x; \theta)$ and illustrate an important relation, which is very similar to the decomposition presented in Proposition 1.1. The main idea is to note that, for any fixed value x, the quantity $r(x; \hat{\theta}_N)$ is an **estimator** that estimates the true value of the regression function $r(x)$. Thus, we can expect a bias-variance decomposition to hold for the regression function.

Proposition 3.4 *Consider the regression defined in Definition 3.1. For any fixed value $x \in \mathbb{R}^n$ the quantity $r(x; \hat{\theta}_N)$ is an* ***estimator*** *of the true value of the regression function $r(x) = E_{Y|X=x}[Y]$, corresponding to the estimator $\hat{\theta}_N$. The MSE of the estimator $r(x; \hat{\theta}_N)$ is given by*

$$\begin{aligned} MSE(r(x;\hat{\theta}_N), r(x)) &= E_D[||r(x;\hat{\theta}_N) - r(x)||^2] \\ &= ||E_D[r(x;\hat{\theta}_N)] - r(x)||^2 + E_D[||r(x;\hat{\theta}_N) - E_D[r(x;\hat{\theta}_N)]||^2] \\ &= ||BIAS(r(x;\hat{\theta}_N), r(x))||^2 + VAR(r(x;\hat{\theta}_N)), \end{aligned} \tag{3.25}$$

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

In the case of **ordinary linear regression**, it is easy to see that the variance in (3.25) is **proportional** to the variance of the estimator $\hat{\theta}_N$. Therefore, if the variance of the estimator $\hat{\theta}_N$ is high, we expect a higher variance of $r(x; \hat{\theta}_N)$ and consequently lower performance of the regression function $r(x; \hat{\theta}_N)$ when predicting on unseen data (x, y), that is, data not belonging to the training set. Recall that the estimators are random variables, and when the variance is high, they are likely to exhibit significant variability across different data sets.

Let us explain briefly 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 solve the regression problem and obtain the estimator $\hat{\theta}_N$ using formula (3.11). We might assess the quality of the estimator by comparing the prediction of the regression function $r(x'_k; \hat{\theta}_N)$ with the actual value y'_k, $k = 1 \ldots, N$; this procedure is referred to as **model validation** in the literature and is a popular method for verifying the performance of fitted models. If the variance of the regression function $r(x; \hat{\theta}_N)$ is large, we expect that the validation will reveal significant discrepancies between the predictions and the actual values of D'. This discrepancy occurs because it is unlikely that the optimal parameter $\hat{\theta}_N$, estimated from data in D, remains the correct parameter value for the new data set D'. In other words, one can have situations where $\hat{\theta}_N$ does not **generalize** well, and hence the predictions are not optimal outside of the training set.

When a trained model fits the training data very well but performs poorly on unseen data, the model is said to be **overfitted**. This situation corresponds to **low bias** and **large variance** of the estimator, and the model predictions are highly sensitive to the particular training data, causing erratic performance on new data.

Conversely, when a trained model accurately describes both the training data and unseen data, it is said to have **good generalization** power. Such a model captures the essential relationship between features and outcomes without overfitting the noise, thus **balancing bias and variance** appropriately.

Another possible case is when the trained model fails to fit the training data well and consequently performs poorly on unseen data as well. This is referred to as **underfitting**. Underfitting occurs when the model is too simple to capture the underlying structure of the data. In terms of error components, underfitting corresponds to **large bias** and **low variance**—the model makes strong assumptions that do not hold for the data, leading to systematic errors regardless of the training set. The small variance in this case is detrimental because values far from the mean—which could compensate for the large bias and be closer to the true parameter values—are unlikely.

For example, in the context of **ordinary linear regression**, underfitting arises when the assumption of linearity is insufficient to capture the true relationship between predictors and response variables. If the true relationship is **nonlinear**, a purely linear model will fail to fit the data adequately. This situation invalidates the conclusions of Proposition 3.1, as that proposition generally relies on the assumption that a linear model adequately represents the relationship at least within the data set D.

Proposition 3.3 in the context of linear regression with Gaussian features highlights how overfitting is closely related to the parameter space dimension d. Specifically, it shows that the variance of the estimator $\hat{\theta}_N$ (3.24) increases with the ratio d/N.

When $d = N$, the variance **diverges**. At this boundary case, the optimization problem (3.20) reduces to solving a standard linear system $\bar{x}\bar{\theta} = \bar{y}$. The invertibility of matrix $\bar{x}$ (recall its maximum rank) yields the exact solution $\hat{\theta}_N = \bar{x}^{-1}\bar{y}$, and a perfect fit to the training targets with $R_D(\hat{\theta}_N) = 0$. This means the model **memorizes** the training data exactly, leading to severe overfitting and poor generalization on unseen data.

The intuitive practical guideline derived is that to maintain effective linear regression performance, the number of data points N should exceed the number of parameters d, to keep the ratio d/N low. More data reduces estimator variance and increases the likelihood of capturing the true data-generating process, improving generalization beyond the training set.

To reduce the phenomenon of overfitting in regression, modified versions of regression models have been introduced in the literature that are **regularized**. Regularization techniques help control overfitting by adding a penalty term, which discourages large values of estimators. Let us define this more precisely.

Definition 3.4 *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 L_2**-regularized (ridge) ordinary regression** consists of inferring (learning) from data in D the parameters $\theta \in \Theta \subseteq \mathbb{R}^d$ of the statistical model*

$$f(y|x;\theta,\sigma) := \alpha\mathcal{N}(y; r(x;\theta), \sigma^2)g(\theta), \tag{3.26}$$

where α is an unimportant normalization constant and

$$g(\theta) := \mathcal{N}(\theta; 0, \lambda^{-1}I_{d\times d}) = \frac{1}{\sqrt{(2\pi\lambda^{-1})^d}}\exp\left(-\frac{\lambda}{2}||\theta||^2\right). \tag{3.27}$$

*In (3.27) $\lambda > 0$ is a constant termed **regularization hyperparameter**.*

The interpretation of (3.27) may seem obscure within the **frequentist** approach to inference that we have adopted, but it becomes clear in the **Bayesian** approach, where $g(\theta)$ serves as the **prior** distribution for θ. The L_2-regularized ordinary linear regression admits a closed-form solution, as illustrated by the following proposition.

Proposition 3.5 *Consider the L_2-**regularized ordinary linear regression** with $m = 1$, $\dim\Theta =: d = n+1$. Define the matrix quantities as in (3.8-10). Then, the maximum likelihood method gives the following estimators:*

$$\hat{\theta}_N^{(\lambda)} = (\bar{x}^T\bar{x} + N\lambda I_{d\times d})^{-1}\bar{x}^T\bar{y}, \tag{3.28}$$

while the estimator $\hat{\sigma}_N^{(\lambda)2}$ is the same as (3.12).

Proof. To estimate the parameters of the regression function, we use the ML method and (3.26) to obtain the LLF:

$$\begin{aligned} -LLF_D(\theta,\sigma^2) &= -\sum_{k=1}^{N}\ln\mathcal{N}(y_k; r(x_k;\theta), \sigma^2) - N\ln g(\theta) \\ &= \frac{N}{2}\ln(2\pi\sigma^2) + \frac{1}{2\sigma^2}\sum_{k=1}^{N}(y_k - r(x_k;\theta))^2 + \frac{N\lambda}{2}||\theta||^2 + const. \end{aligned} \tag{3.29}$$

We changed signs as usual. Then, the original inference problem was converted into the following optimization problems:

$$\theta(D) = \arg\min_{\theta\in\Theta} R_D(\theta), \tag{3.30}$$

$$\sigma^2(D) = \arg\min_{\theta\in\Theta} S_D(\theta,\sigma^2), \tag{3.31}$$

where the target functions are defined as follows:

$$R_D(\theta) := \sum_{k=1}^{N}(y_k - r(x_k;\theta))^2 + \frac{N\lambda}{2}||\theta||^2, \tag{3.32}$$

$$S_D(\theta,\sigma^2) := \frac{N}{2}\ln(2\pi\sigma^2) + \frac{1}{2\sigma^2}\left(R_D(\theta) - \frac{N\lambda}{2}||\theta||^2\right). \tag{3.33}$$

Using definitions in (3.8-10), the function in (3.32) can be rewritten using the Euclidean norm as follows:

$$R_D(\bar{\theta}) = ||\bar{y} - \bar{x}\bar{\theta}||^2 + \frac{N\lambda}{2}||\bar{\theta}||^2. \tag{3.34}$$

Calculating the matrix gradient of (3.34) with respect to $\bar{\theta}$ and equating it to zero yields the following matrix equation (we absorbed 1/2 in λ):

$$(\bar{x}^T\bar{x} + N\lambda I_{d\times d})\bar{\theta} = \bar{x}^T\bar{y}. \tag{3.35}$$

The non-zero value of the regularization hyperparameter makes the left-hand side matrix in (3.35) always invertible, so one obtains (3.28). The optimization problem (3.33) has the solution in (3.12). □

The resulting estimator in (3.28) is called the **L_2-regularized least squares (RLS) estimator** and it is the unique minimizer because the target function (3.32) is strictly **convex**. The following proposition illustrates some properties of the RLS estimator.

Proposition 3.6 *The **RLS estimator** in (3.28) has the following properties for all $\lambda \geq 0$:*

- *The estimator is **biased** with*

$$BIAS(\hat{\theta}_N^{(\lambda)}, \theta_*) = -N\lambda E_{X_1,\ldots,X_N}[(\bar{x}^T\bar{x} + N\lambda I_{d\times d})^{-1}]\theta_*. \tag{3.36}$$

- *The variance is*

$$VAR(\hat{\theta}_N^{(\lambda)}) = \sigma^2 E_{X_1,\ldots,X_N}[\mathrm{tr}\{\bar{x}^T\bar{x}(\bar{x}^T\bar{x} + N\lambda I_{d\times d})^{-2}\}]. \tag{3.37}$$

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

$$||\hat{\theta}_N^{(\lambda)}||^2 = \bar{y}^T\bar{x}(\bar{x}^T\bar{x} + N\lambda I_{d\times d})^{-2}\bar{x}^T\bar{y} \sim \lambda^{-2}. \tag{3.38}$$

- *The MSE is*

$$MSE(\hat{\theta}_N^{(\lambda)}) = N^2\lambda^2\theta_*^T E_{X_1,\ldots,X_N}[(\bar{x}^T\bar{x}+N\lambda I_{d\times d})^{-2}]\theta_* + \sigma^2 E_{X_1,\ldots,X_N}[\mathrm{tr}\{\bar{x}^T\bar{x}(\bar{x}^T\bar{x}+N\lambda I_{d\times d})^{-2}\}]. \tag{3.39}$$

The expression (3.38) indicates that the hyperparameter λ acts as a **penalty** term that penalizes estimators with large Euclidean norm values. This effect is commonly referred to in the literature as **shrinkage**, where the penalty encourages coefficient estimates to be closer to zero.

The MSE decomposition (3.39) indicates that **increasing** λ implies **increasing** bias and **reducing** variance. If λ is too large, the bias becomes high while the variance becomes low, which leads to poor performance on both training data and unseen data. Conversely, if λ is too small, the bias is low (resulting in good performance on training data), but the variance is high, causing overfitting and poor generalization to unseen data. Therefore, it is not possible to simultaneously reduce bias and variance. This characteristic behavior of bias and variance in relation to the estimator is known as the **bias-variance trade-off**. An effective choice of λ must balance these opposing effects to minimize the MSE of the estimator.

Another form of regularized linear regression is the **LASSO regression**.

Definition 3.5 *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 **L_1-regularized (LASSO, Least Absolute Shrinkage and Selection Operator) ordinary regression** consists of inferring (learning) from data in D the parameters $\theta \in \Theta \subseteq \mathbb{R}^d$ of the statistical model*

$$f(y|x;\theta,\sigma) := \alpha\mathcal{N}(y; r(x;\theta), \sigma^2)g(\theta), \tag{3.40}$$

*where α is an unimportant normalization constant and g has the **Laplace** form*

$$g(\theta) := \mathrm{La}(\theta; 0, \lambda^{-1}) = \frac{\lambda}{2}\exp\left(-\lambda||\theta||_1\right). \tag{3.41}$$

*In (3.41) $||\theta||_1 := \sum_{k=1}^d |\theta^{(k)}|$ is the L_1 norm, and $\lambda > 0$ is a constant termed **regularization hyperparameter**.*