CHAPTER

LEARNING IN PARAMETRIC MODELING: BASIC CONCEPTS AND DIRECTIONS

3

CHAPTER OUTLINE

3.1 INTRODUCTION

Parametric modeling is a theme that runs across the spine of this book. A number of chapters focus on different aspects of this important problem. This chapter provides basic definitions and concepts related to the task of learning when parametric models are mobilized to describe the available data.

Machine Learning. http://dx.doi.org/10.1016/B978-0-12-801522-3.00003-3

© 2015 Elsevier Ltd. All rights reserved.

As it has already been pointed out in the introductory chapter, a large class of machine learning problems ends up in being equivalent to a function estimation/approximation task. The function is "learned" during the learning/training phase by digging in the information that resides in the available training data set. This function relates the so-called input variables to the output variable(s). Once this functional relationship is established, one can in turn exploit it to predict the value(s) of the output(s), based on measurements obtained from the respective input variables; these predictions can then be used to proceed to the decision-making phase.

In parametric modeling, the aforementioned functional dependence is defined via a set of unknown parameters, whose number is *fixed*. In contrast, in the so-called *nonparametric* methods, unknown parameters may still be involved, yet their number depends on the size of the data set. Nonparametric methods will also be treated in this book. However, the emphasis in this chapter lies in the former ones.

In parametric modeling, there are two possible paths to deal with the uncertainty imposed by the unknown values of the parameters. According to the first one, specific values are obtained and assigned to the unknown parameters. In the other approach, which has a stronger statistical flavor, parametric models are adopted in order to describe the underlying probability distributions, which describe the input and output variables, without it being necessary to obtain specific values for the unknown parameters.

Two of the major machine learning tasks, namely the regression and the classification, are presented and the main directions in dealing with these problems are exposed. Various issues that are related to the parameter estimation task, such as estimator efficiency, bias-variance dilemma, overfitting, curse of dimensionality are introduced and discussed. The chapter can also be considered as a road map to the rest of the book. However, instead of just presenting the main ideas and directions in a rather "dry" way, we chose to deal and work with the involved tasks by adopting simple models and techniques, so that the reader gets a better feeling of the topic. An effort was made to pay more attention to the scientific notions than to algebraic manipulations and mathematical details, which will, unavoidably, be used to a larger extent while "embroidering" the chapters to follow.

The Least-Squares (LS), the Maximum Likelihood (ML), the Regularization as well as the Bayesian Inference techniques are presented and discussed. An effort has been made to assist the reader to grasp an informative view of the big picture conveyed by the book. Thus, this chapter could also be used as an overview introduction to the parametric modeling task in the realm of machine learning.

3.2 PARAMETER ESTIMATION: THE DETERMINISTIC POINT OF VIEW

The task of estimating the value of an unknown parameter vector, $\boldsymbol{\theta}$, has been at the center of interest in a number of application areas. For example, in the early years in the University, one of the very first tasks any student has to study is the so-called curve fitting problem. Given a set of data points, one must find a curve or a surface that "fits" the data. The usual path to follow is to *adopt* a functional form, such as a linear function or a quadratic one, and try to estimate the associated unknown coefficients so that the graph of the function "passes through" the data and follows their deployment in space as close as possible. Figures 3.1a and b are two such examples. The data lie in the $\mathbb{R}^2$ space and are given to us as a set of points (y_n, x_n), $n = 1, 2, \ldots, N$. The adopted functional form for the curve corresponding to Figure 3.1a is

$$y = f_{\boldsymbol{\theta}}(x) = \theta_0 + \theta_1 x, \tag{3.1}$$

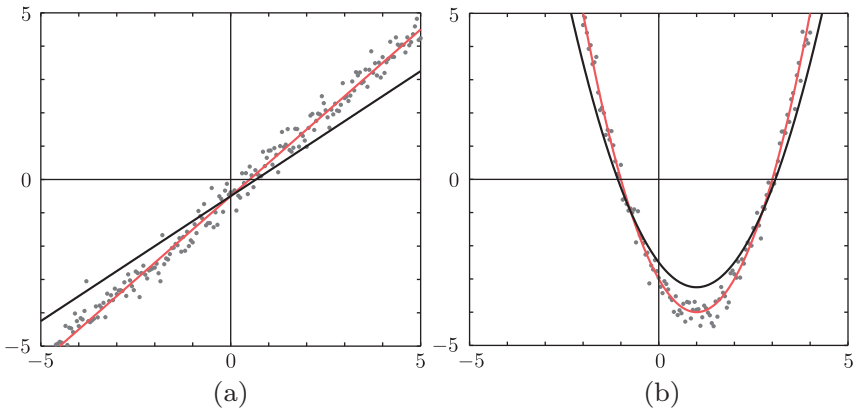

FIGURE 3.1

Fitting (a) a linear function and (b) a quadratic one. The red lines are the optimized ones.

and for the case of Figure 3.1b

$$y = f_{\boldsymbol{\theta}}(x) = \theta_0 + \theta_1 x + \theta_2 x^2. \tag{3.2}$$

The unknown parameter vectors are $\boldsymbol{\theta} = [\theta_0, \theta_1]^T$ and $\boldsymbol{\theta} = [\theta_0, \theta_1, \theta_2]^T$, respectively. In both cases, the parameter values, which define the curves drawn by the red lines, provide a much better fit compared to the values associated with the black ones. In both cases, the task comprises two steps: (a) first adopt a specific *parametric functional* form, which we reckon to be more appropriate for the data at hand and (b) estimate the values of the unknown parameters in order to obtain a "good" fit.

In the more general and formal setting, the task can be defined as follows. Given a set of data points, $(y_n, \boldsymbol{x}_n)$, $y_n \in \mathbb{R}$, $\boldsymbol{x}_n \in \mathbb{R}^l$, $n = 1, 2, \ldots, N$, and a parametric set of functions,

$$\mathcal{F} := \left\{ f_{\boldsymbol{\theta}}(\cdot) : \ \boldsymbol{\theta} \in \mathcal{A} \subseteq \mathbb{R}^K \right\}, \tag{3.3}$$

find a function in $\mathcal{F}$, which will be denoted as $f(\cdot) := f_{\boldsymbol{\theta}_*}(\cdot)$, such that given a value of $\boldsymbol{x} \in \mathbb{R}^l$, $f(\boldsymbol{x})$ best approximates the corresponding value $y \in \mathbb{R}$. We start our discussion by considering y to be a real variable, $y \in \mathbb{R}$, and as we move on and understand better the various "secrets," we will allow it to move to higher dimensional Euclidean spaces. The value $\boldsymbol{\theta}_*$ is the value that results from the estimation procedure. The values of $\boldsymbol{\theta}_*$ that define the red line curves in Figures 3.1a and b are

$$\boldsymbol{\theta}_* = [-0.5, 1]^T, \quad \boldsymbol{\theta}_* = [-3, -2, 1]^T, \tag{3.4}$$

respectively.

To reach a decision with respect to the choice of $\mathcal{F}$ is not an easy task. For the case of the data in Figure 3.1, we were a bit "lucky." First, the data live in the two-dimensional space, where we have the luxury of visualization. Second, the data were scattered along curves whose shape is pretty familiar to us; hence, a simple inspection suggested the proper family of functions, for each one of the two cases. Obviously, real life is hardly as generous as that and in the majority of practical applications,

the data reside in high-dimensional spaces and/or the shape of the surface (hypersurface, for spaces of dimensionality higher than three) can be quite complex. Hence, the choice of $\mathcal{F}$, which dictates the functional form (e.g., linear, quadratic, etc.) is not easy. In practice, one has to use as much a priori information as possible concerning the physical mechanism that underlies the generation of the data, and most often use different families of functions and finally keep the one that results in the best performance, according to a chosen criterion.

Having adopted a parametric family of functions, $\mathcal{F}$, one has to get an estimate for the unknown set of parameters. To this end, a measure of fitness has to be adopted. The more classical approach is to adopt a *loss* function, which quantifies the deviation/error between the measured value of y and the predicted one using the corresponding measurements $\boldsymbol{x}$, as in $f_{\boldsymbol{\theta}}(\boldsymbol{x})$. In a more formal way, we adopt a *nonnegative* (loss) function,

$$\mathcal{L}(\cdot,\cdot) : \mathbb{R} \times \mathbb{R} \longmapsto [0, \infty),$$

and compute $\boldsymbol{\theta}_*$ so as to minimize the total loss, or as we say the *cost*, over all the data points, or

$$f(\cdot) := f_{\boldsymbol{\theta}_*}(\cdot) : \ \boldsymbol{\theta}_* = \arg\min_{\boldsymbol{\theta} \in \mathcal{A}} J(\boldsymbol{\theta}), \tag{3.5}$$

where

$$J(\boldsymbol{\theta}) := \sum_{n=1}^{N} \mathcal{L}\big(y_n, f_{\boldsymbol{\theta}}(\boldsymbol{x}_n)\big), \tag{3.6}$$

assuming that a minimum exists. Note that, in general, there may be more than one optimal values $\boldsymbol{\theta}_*$, depending on the shape of $J(\boldsymbol{\theta})$.

As the book evolves, we are going to see different loss functions and different parametric families of functions. For the sake of simplicity, for the rest of this chapter we will adhere to the LS loss function,

$$\mathcal{L}\big(y, f_{\boldsymbol{\theta}}(\boldsymbol{x})\big) = \big(y - f_{\boldsymbol{\theta}}(\boldsymbol{x})\big)^2,$$

and to the linear class of functions.

The LS loss function is credited to the great mathematician Carl Frederich Gauss, who proposed the fundamentals of the LS method in 1795 at the age of eighteen. However, it was Adrien-Marie Legendre who first published the method in 1805, working independently. Gauss published it in 1809. The strength of the method was demonstrated when it was used to predict the location of the asteroid Ceres. Since then, the LS loss function has "haunted" all scientific fields, and even if it is not used directly, it is, most often, used as the standard against which the performance of more modern alternatives are compared. This success is due to some nice properties that this loss criterion has, which will be explored as we move on in this book.

The combined choice of linearity with the LS loss function turns out to simplify the algebra and hence becomes very pedagogic for introducing the newcomer to the various "secrets" that underlie the area of parameter estimation. Moreover, understanding linearity is very important. Treating nonlinear tasks, most often, turns out to finally resort to a linear problem. Take, for example, the nonlinear model in Eq. (3.2) and consider the transformation

$$\mathbb{R} \ni x \longmapsto \boldsymbol{\phi}(x) := \begin{bmatrix} x \\ x^2 \end{bmatrix} \in \mathbb{R}^2. \tag{3.7}$$

Then, Eq. (3.2) becomes

$$y = \theta_0 + \theta_1\phi_1(x) + \theta_2\phi_2(x). \tag{3.8}$$

That is, the model is now linear with respect to the components $\phi_k(x)$, $k = 1, 2$, of the two-dimension image, $\boldsymbol{\phi}(x)$, of x. As a matter of fact, this simple trick is at the heart of a number of nonlinear methods that will be treated later on in the book. No doubt, the procedure can be generalized to any number, K, of functions, $\phi_k(\boldsymbol{x})$, $k = 1, 2, \ldots, K$, and besides monomials, other types of nonlinear functions can be used such as exponentials, splines, wavelets, to name a few. In spite of the nonlinear nature of the input-output dependence modeling, we still consider this model to be linear, because it retains its linearity with respect to the involved unknown parameters, θ_k, $k = 1, 2, \ldots, K$. Although for the rest of the chapter we will adhere to linear functions, in order to keep our discussion simpler, everything that will be said applies to nonlinear ones. All that is needed is to replace $\boldsymbol{x}$ with $\boldsymbol{\phi}(\boldsymbol{x}) := [\phi_1(\boldsymbol{x}), \ldots, \phi_K(\boldsymbol{x})]^T \in \mathbb{R}^K$.

In the sequel, we will present two examples in order to demonstrate the use of parametric modeling. These examples are generic and can represent a wide class of problems.

3.3 LINEAR REGRESSION

In statistics, the term *regression* was coined to define the task of modeling the relationship of a *dependent* random variable, y, which is considered to be the response of a system, when this is activated by a set of random variables, $\mathrm{x}_1, \mathrm{x}_2, \ldots, \mathrm{x}_l$, which will be represented as the components of an equivalent random vector $\mathbf{x}$. The relationship is modeled via an additive disturbance or noise term, η. The block diagram of the process, which relates the involved variables, is given in Figure 3.2. The noise variable, η, is an *unobserved* random variable. The goal of the regression task is to estimate the parameter vector, $\boldsymbol{\theta}$, given a set of measurements, $(y_n, \boldsymbol{x}_n)$, $n = 1, 2, \ldots, N$, that we have at our disposal. This is also known as the *training data set*, or the *observations*. The dependent variable is usually known as the *output* variable and the vector $\mathbf{x}$ as the *input* vector or the *regressor*. If we model the system as a linear combiner, the dependence relationship is written as

$$\mathrm{y} = \theta_0 + \theta_1\mathrm{x}_1 + \cdots + \theta_l\mathrm{x}_l + \eta = \theta_0 + \boldsymbol{\theta}^T\mathbf{x} + \eta. \tag{3.9}$$

The parameter θ_0 is known as the *bias* or the *intercept*. Usually, this term is absorbed by the parameter vector $\boldsymbol{\theta}$ with a simultaneous increase of the dimension of $\mathbf{x}$ by adding the constant 1 as its last element. Indeed, we can write

$$\theta_0 + \boldsymbol{\theta}^T\mathbf{x} + \eta = [\boldsymbol{\theta}^T, \theta_0]\begin{bmatrix}\mathbf{x}\\ 1\end{bmatrix} + \eta.$$

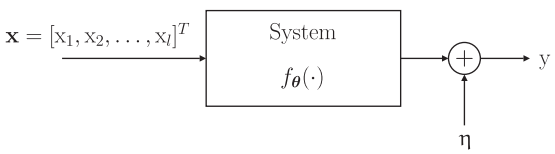

FIGURE 3.2

Block diagram showing the input-output relation in a regression model.

From now on, the regression model will be written as

$$\mathrm{y} = \boldsymbol{\theta}^T \mathbf{x} + \eta, \tag{3.10}$$

and, unless otherwise stated, this notation means that the bias term has been absorbed by $\boldsymbol{\theta}$ and $\mathbf{x}$ has been extended by adding 1 as an extra component. Because the noise variable is unobserved, we need a model to be able to *predict* the output value of y, given the value $\mathbf{x}$.

In linear regression, we adopt the following prediction model:

$$\hat{y} = \hat{\theta}_0 + \hat{\theta}_1 x_1 + \cdots + \hat{\theta}_l x_l := \hat{\boldsymbol{\theta}}^T \boldsymbol{x}. \tag{3.11}$$

Using the LS loss function, the estimate $\hat{\boldsymbol{\theta}}$ is set equal to $\boldsymbol{\theta}_*$, which minimizes the square difference between $\hat{y}_n$ and y_n, over the set of the available observations; that is, by minimizing, with respect to $\boldsymbol{\theta}$, the cost function

$$J(\boldsymbol{\theta}) = \sum_{n=1}^{N} (y_n - \boldsymbol{\theta}^T \boldsymbol{x}_n)^2. \tag{3.12}$$

Taking the derivative (gradient) with respect to $\boldsymbol{\theta}$ and equating to the zero vector, $\mathbf{0}$, we obtain

$$\left(\sum_{n=1}^{N} \boldsymbol{x}_n \boldsymbol{x}_n^T \right) \hat{\boldsymbol{\theta}} = \sum_{n=1}^{N} \boldsymbol{x}_n y_n. \tag{3.13}$$

Another more popular way to write the previously obtained relation is via the so-called input matrix, X, defined as the $N \times (l+1)$ matrix, which has as rows the (extended) regressor vectors, $\boldsymbol{x}_n^T$, $n = 1, 2, \ldots, N$, expressed as

$$X := \begin{bmatrix} \boldsymbol{x}_1^T \\ \boldsymbol{x}_2^T \\ \vdots \\ \boldsymbol{x}_N^T \end{bmatrix} = \begin{bmatrix} x_{11} & \ldots & x_{1l} & 1 \\ x_{21} & \ldots & x_{2l} & 1 \\ \vdots & \ddots & \vdots & \\ x_{N1} & \ldots & x_{Nl} & 1 \end{bmatrix}. \tag{3.14}$$

Then, it is straightforward to see that Eq. (3.13) can be written as

$$X^T X \hat{\boldsymbol{\theta}} = X^T \boldsymbol{y}, \tag{3.15}$$

where

$$\boldsymbol{y} := [y_1, y_2, \ldots, y_N]^T, \tag{3.16}$$

and the LS estimate is given by

$$\boxed{\hat{\boldsymbol{\theta}} = (X^T X)^{-1} X^T \boldsymbol{y} : \quad \text{The LS Estimate,}} \tag{3.17}$$

assuming, of course, that $(X^T X)^{-1}$ exists.

In other words, the obtained estimate of the parameter vector is given by a *linear set of equations*. This is a major advantage of the LS loss function, when applied to a linear model. Moreover, this solution is *unique*, provided that the $(l+1) \times (l+1)$ matrix $X^T X$ is invertible. The uniqueness is due to the parabolic shape of the graph of the LS cost function. This is illustrated in Figure 3.3 for the

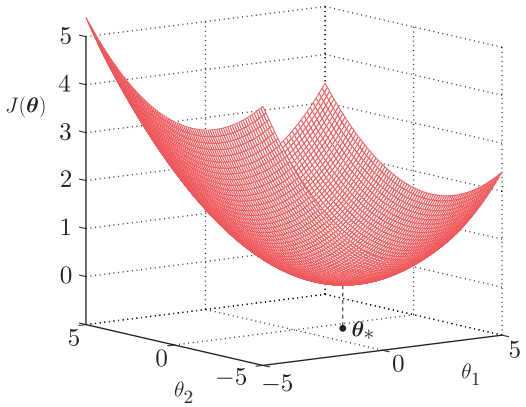

FIGURE 3.3

The least-squares loss function has a unique minimum at the point $\boldsymbol{\theta}_*$.

two-dimensional space. It is readily observed that the graph has a unique minimum. This is a consequence of the fact the LS cost function is a strictly convex one. Issues related to convexity of loss functions will be treated in more detail in Chapter 8.

Example 3.1. Consider the system that is described by the following model:

$$\mathrm{y} = \theta_0 + \theta_1 \mathrm{x}_1 + \theta_2 \mathrm{x}_2 + \eta := [0.25, -0.25, 0.25] \begin{bmatrix} \mathrm{x}_1 \\ \mathrm{x}_2 \\ 1 \end{bmatrix} + \eta, \tag{3.18}$$

where η is a Gaussian random variable of zero mean and variance $\sigma^2 = 1$. The random variables x_1 and x_2 are assumed to be mutually independent, and uniformly distributed over the interval $[0, 10]$. Generate $N = 50$ points for each one of the three random variables. For each triplet, use Eq. (3.18) to generate the corresponding value, y, of y. In this way, the points $(y_n, \boldsymbol{x}_n)$, $n = 1, 2, \ldots, 50$, are generated, where each observation, $\boldsymbol{x}_n$ of $\mathbf{x}$, lies in $\mathbb{R}^3$, after extending it by adding one as its last element. These are used as the training points to obtain the LS estimates of the coefficients of the linear model

$$\hat{y} = \hat{\theta}_0 + \hat{\theta}_1 x_1 + \hat{\theta}_2 x_2.$$

Repeat the experiments with $\sigma^2 = 10$.

The values of the LS optimal estimates are obtained by solving a 3×3 linear system of equations and they are

$$\text{(a) } \hat{\theta}_0 = 0.6642,\ \hat{\theta}_1 = 0.2471,\ \hat{\theta}_2 = -0.3413,$$
$$\text{(b) } \hat{\theta}_0 = 1.5598,\ \hat{\theta}_1 = 0.2408,\ \hat{\theta}_2 = -0.5386,$$

for the two cases, respectively. Figures 3.4a and b show the recovered planes. Observe that in the case of Figure 3.4a, corresponding to a noise variable of small variance, the obtained plane follows the data points much closer, compared to that of Figure 3.4b.

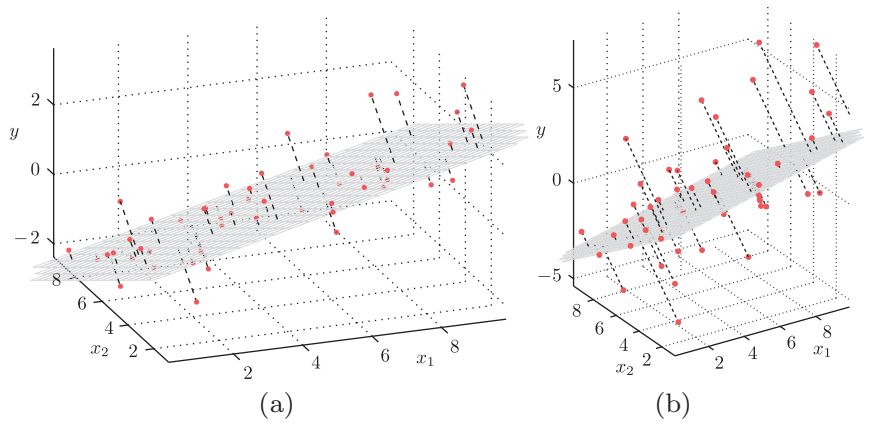

FIGURE 3.4

Fitting a plane using the LS method for (a) a low variance and (b) a high variance noise case. Note that when the noise variance in the regression model is low, a much better fit to the data set is obtained.

Remarks 3.1.

- The set of points $(\hat{y}_n, x_{n1}, \ldots, x_{nl})$, $n = 1, 2, \ldots, N$, lie on a *hyperplane* in the $\mathbb{R}^{l+1}$ space. Equivalently, they lie on a hyperplane that crosses the origin and, thus, it is a linear subspace in the extended space $\mathbb{R}^{l+2}$, when one absorbs θ_0 in $\boldsymbol{\theta}$, as explained previously.
- Notice that the prediction model in Eq. (3.11) could still be used, even if the true systems' structure does not obey the linear model in Eq. (3.9). For example, the true dependence between y and $\mathbf{x}$ may be a nonlinear one. Well, in such a case, the predictions of the y's, based on the model in Eq. (3.11), may not be satisfactory. It all depends on the deviation of our adopted model from the true structure of the system that generates the data.
- The prediction performance of the model depends, also, on the statistical properties of the noise variable. This is an important issue. We will see later on that, depending on the statistical properties of the noise variable, some loss functions and methods may be more suitable than others.
- The two previous remarks suggest that in order to quantify the performance of an estimator some related criteria are necessary. In Section 3.9, we will present some theoretical touches that shed light on certain aspects related to the performance of an estimator.

3.4 CLASSIFICATION

Classification is the task of predicting the class to which an object, known as *pattern*, belongs. The pattern is assumed to belong to one and only one among a number of a priori known classes. Each pattern is *uniquely* represented by a set of measurements, known as *features*. One of the early stages in designing a classification system is to select an appropriate set of feature variables. These should "encode" as much class-discriminatory information, so that, by measuring their value for a given pattern, to be able to predict, with high enough probability, the class of the pattern. Selecting the appropriate set of features,

for each problem is not an easy task and it comprises one of the most important areas within the field of *Pattern Recognition* (e.g., [11, 35]). Having selected, say, l feature (random) variables, $\mathrm{x}_1, \mathrm{x}_2, \ldots, \mathrm{x}_l$, we stack them as the components of the so-called *feature vector*, $\mathbf{x} \in \mathbb{R}^l$. The goal is to design a *classifier*, such as a function[1] $f(\boldsymbol{x})$, or equivalently a *decision surface*, $f(\boldsymbol{x}) = 0$, in $\mathbb{R}^l$, so that given the values in a feature vector, $\boldsymbol{x}$, which corresponds to a pattern, we will be able to *predict* the class to which the pattern belongs.

To formulate the task in mathematical terms, each class is represented by the *class label* variable, y. For the simple two-class classification task, this can take either of two values, depending on the class, e.g., $1, -1$, or $1, 0$, etc. Then, given the value of $\boldsymbol{x}$, corresponding to a specific pattern, its class label is predicted according to the rule,

$$\hat{y} = \phi(f(\boldsymbol{x})),$$

where $\phi(\cdot)$ is a nonlinear function that indicates on which side of the decision surface, $f(\boldsymbol{x}) = 0$, $\boldsymbol{x}$ lies. For example, if the class labels are ± 1, the nonlinear function is chosen to be the sign function, or $\phi(\cdot) = \mathrm{sgn}(\cdot)$. It is now clear that what we have said so far in the previous section can be transferred here and the task becomes that of estimating a function $f(\cdot)$, based on a set of training points $(y_n, \boldsymbol{x}_n) \in D \times \mathbb{R}^l$, $n = 1, 2, \ldots, N$, where D denotes the discrete set in which y lies. Function $f(\cdot)$ is selected so as to belong in a specific parametric class of functions, $\mathcal{F}$, and the goal is, once more, to estimate the parameters so that the deviation between the true class labels, y_n, and the predicted ones, $\hat{y}_n$, is minimum according to a preselected criterion. So, is the classification any different from the regression task?

The answer to the previous question is that they are similar, yet different. Note that in a classification task, the dependent variables are of a *discrete* nature, in contrast to the regression, where they lie in an interval. This suggests that, in general, different techniques have to be adopted to optimize the parameters. For example, the most obvious choice for a criterion in a classification task is the probability of error. However, in a number of cases, one can attack both tasks using the same type of loss functions, as we will do in this section; even if such an approach is adopted, in spite of the similarities in their mathematical formalism, the goals of the two tasks remain different.

In the regression task, the function $f(\cdot)$ has to "explain" the data generation mechanism. The corresponding surface in the $(y, \boldsymbol{x})$ space $\mathbb{R}^{l+1}$ should develop so as to follow the spread of the data in the space, as close as possible. In contrast, in classification, the goal is to place the corresponding surface $f(\boldsymbol{x}) = 0$, in $\mathbb{R}^l$, so as to separate the data that belong to different classes as much as possible. The goal of a classifier is to *partition* the space where the features vectors lie into regions and associate each region with a class. Figure 3.5 illustrates two cases of classification tasks. The first one is an example of two linearly separable classes, where a straight line can separate the two classes, and the second one of two nonlinearly separable classes, where the use of a linear classifier would have failed to separate the two classes.

Let us now make what we have said, so far, more concrete. We are given a set of training patterns, $\boldsymbol{x}_n \in \mathbb{R}^l$, $n = 1, 2, \ldots, N$, that belong to either of two classes, say ω_1 and ω_2. The goal is to design a hyperplane

$$\begin{aligned} f(\boldsymbol{x}) &= \theta_0 + \theta_1 x_1 + \cdots + \theta_l x_l \\ &= \boldsymbol{\theta}^T \boldsymbol{x} = 0, \end{aligned}$$

[1] In the more general case, a set of functions.

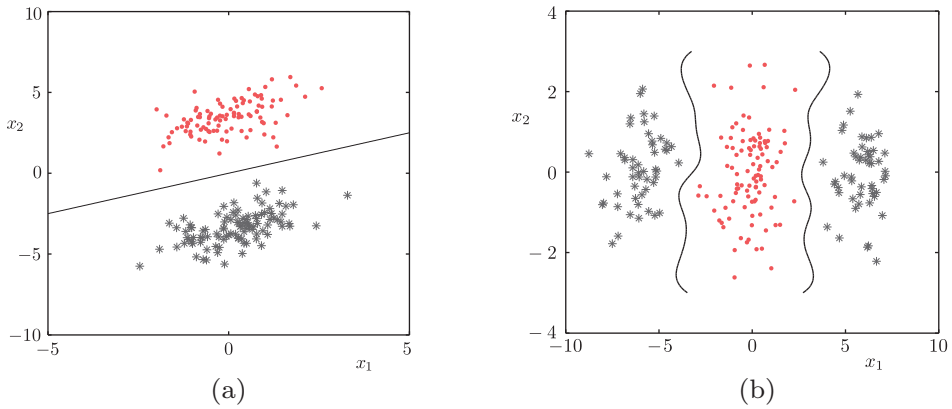

FIGURE 3.5

Examples of two-class classification tasks. (a) A linearly separable and (b) a nonlinearly separable one. The goal of a classifier is to divide the space into regions and associate each region with a class.

where we have absorbed the bias θ_0 in $\boldsymbol{\theta}$ and extend the dimension of $\boldsymbol{x}$, as it has already been explained before. Our aim is to place this hyperplane in between the two classes. Obviously, any point lying on this hyperplane scores a zero, $f(\boldsymbol{x}) = 0$, and the points lying on either side of the hyperplane score either a positive ($f(\boldsymbol{x}) > 0$) or a negative value ($f(\boldsymbol{x}) < 0$), depending on which side of the hyperplane they lie. We, therefore, should train our classifier so that the points from one class score a positive value and the points of the other a negative one. This can be done, for example, by labeling all the points from class, say ω_1, with $y_n = 1$, $\forall n : \boldsymbol{x}_n \in \omega_1$, and all the points from class ω_2 with $y_n = -1$, $\forall n : \boldsymbol{x}_n \in \omega_2$. Then the LS loss is mobilized to compute $\boldsymbol{\theta}$ so as to minimize the cost

$$J(\boldsymbol{\theta}) = \sum_{n=1}^{N} \left(y_n - \boldsymbol{\theta}^T \boldsymbol{x}_n\right)^2.$$

The solution is exactly the same as Eq. (3.13). Figure 3.6 shows the resulting LS classifiers for two cases of data. Observe that in the case of Figure 3.6b, the resulting classifier cannot classify correctly all the data points. Our desire to place all the data, which originate from one class, on one side and the rest on the other cannot be satisfied. All that our LS classifier can do is to place the hyperplane so that the sum of squared errors, between the desired (true) values of the labels, y_n, and the predicted outputs, $\boldsymbol{\theta}^T \boldsymbol{x}_n$, are a minimum. It is mainly for cases such as overlapping classes, which are usually encountered in practice, where one has to look for an alternative to the LS criteria and methods, in order to serve better the needs and the goals of the classification task. For example, a reasonable optimality criterion would be to minimize the probability of error; that is, the percentage of points for which the true labels, y_n, and the predicted by the classifier ones, $\hat{y}_n$, are different. Chapter 7 presents methods and loss functions appropriate for the classification task. In Chapter 11, support vector machines are discussed and in Chapter 18, neural networks and deep learning methods are presented, which are currently among the most powerful techniques for classification problems.

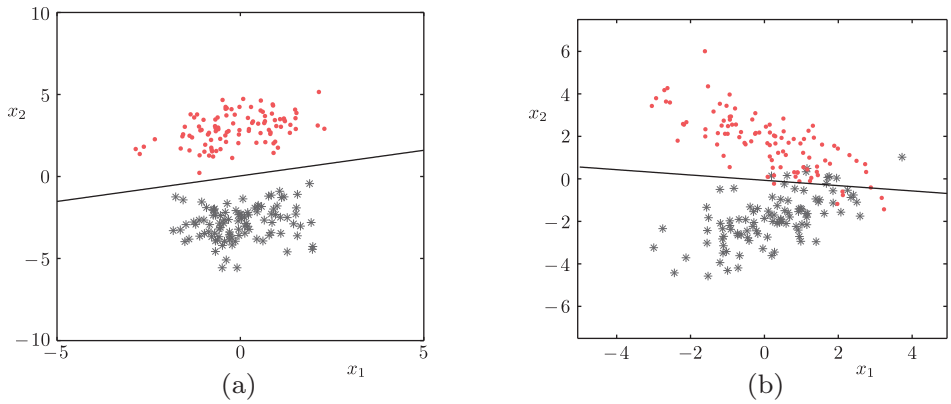

FIGURE 3.6

Design of a linear classifier, $\theta_0 + \theta_1 x_1 + \theta_2 x_2 = 0$, based on the LS loss function. (a) The case of two linearly separable classes and (b) the case of nonseparable classes. In the latter case, the classifier cannot separate fully the two classes. All it can do is to place the separating (decision) line so as to minimize the deviation between the true labels and the predicted output values in the LS sense.

Generative versus discriminative learning

The path that we have taken in order to introduce the classification task was to consider a functional dependence between the output variable (label), y, and the input variables (features), $\mathbf{x}$. The involved parameters were optimized with respect to a cost function. This path of modeling is also known as *discriminative learning*. We were not concerned with the statistical nature of the dependence that ties these two sets of variables together. In a more general setting, the term discriminative learning is also used to cover methods that model directly the posterior probability of a class, represented by its label y, given the feature vector $\boldsymbol{x}$, as in $P(y|\boldsymbol{x})$. The common characteristic of all these methods is that they bypass the need of modeling the input data distribution explicitly. From a statistical point of view, discriminative learning is justified as follows.

Using the product rule for probabilities, the joint distribution between the input data and their respective labels can be written as

$$p(y,\boldsymbol{x}) = P(y|\boldsymbol{x})p(\boldsymbol{x}).$$

In the discriminative learning, only the first of the two terms in the product is considered; a functional form is adopted and parameterized appropriately, as $P(y|\boldsymbol{x};\boldsymbol{\theta})$. Parameters are then estimated via the use of a cost. The distribution of the input data is ignored. Such an approach has the advantage that simpler models can be used, especially if the input data are described by pdfs of a complex form. The disadvantage is that the input data distribution is ignored, although it can carry important information, which could be exploited to the benefit of the overall performance.

In contrast, the alternative path, known as *generative learning*, exploits the input data distribution. Once more, employing the product rule, we have

$$p(y,\boldsymbol{x}) = p(\boldsymbol{x}|y)P(y).$$

$P(y)$ is the probability concerning the classes and $p(\boldsymbol{x}|y)$ is the distribution of the input given the class label. For such an approach, we end up with one distribution per class, which has to be learned. In parametric modeling, a set of parameters is associated with each one of these conditional distributions. Once the joint distribution has been learned, the prediction of the class label of an unknown pattern, $\boldsymbol{x}$, is performed based on the a posteriori probability,

$$P(y|\boldsymbol{x}) = \frac{p(y,\boldsymbol{x})}{p(x)} = \frac{p(y,\boldsymbol{x})}{\sum_y p(y,\boldsymbol{x})}.$$

We will return to these issues in more detail in Chapter 7.

Supervised, semisupervised, and unsupervised learning

The way both the regression as well as the classification tasks were introduced relied on a given set of training data. In other words, the values of the dependent variables, y, are known over the available training set of the regressors and feature vectors/patterns, respectively. For this reason, such tasks belong to the family of problems known as *supervised learning*. However, there are learning problems where the dependent variable is not known, or it may be known only for a small percentage of the available data. In such cases, we refer to *clustering* and *semisupervised learning*, respectively. In this book, our main concern is on the supervised learning. For the other types of learning, the interested reader can consult, for example, [7, 35].

3.5 BIASED VERSUS UNBIASED ESTIMATION

In supervised learning, we are given a set of training points, $(y_n, \boldsymbol{x}_n)$, $n = 1, 2, \ldots, N$, and we return an estimate of the unknown parameter vector, say $\hat{\boldsymbol{\theta}}$. However, the training points themselves are random variables. If we are given another set of N observations of the *same* random variables, these are going to be different, and obviously the resulting estimate will also be different. In other words, by changing our training data different estimates result. Hence, we can assume that the resulting estimate, of a fixed yet unknown parameter, is itself a random variable. This, in turn, poses questions on how good an estimator is. No doubt, each time, the obtained estimate is optimal with respect to the adopted loss function *and* the specific training set used. However, who guarantees that the resulting estimates are "close" to the true value, assuming that there is one? In this section, we will try to address this task and to illuminate some related theoretical aspects. Note that we have already used the term *estimator* in place of the term *estimate*. Let us elaborate a bit on their difference, before presenting more details.

An estimate, such as $\hat{\boldsymbol{\theta}}$, has a specific value, which is the result of a function acting on a set of observations, on which our chosen estimate depends (see Eq. (3.17)). In general, we could generalize Eq. (3.17) and write that

$$\hat{\boldsymbol{\theta}} = f(\mathbf{y}, X).$$

However, once we allow the set of observations to change randomly, and the estimate becomes itself a random variable, we write the previous equation in terms of the corresponding random variables,

$$\hat{\theta} = f(\mathbf{y}, \mathrm{X}),$$

and we refer to this functional dependence as the *estimator* of the unknown vector $\boldsymbol{\theta}$.

In order to simplify the analysis and focus on the insight behind the methods, we will assume that our parameter space is that of real numbers, $\mathbb{R}$. We will also assume that the model (i.e., the set of functions $\mathcal{F}$), which we have adopted for modeling our data, is the correct one and the (unknown to us) value of the associated true parameter is equal to[2] θ_o. Let $\hat{\theta}$ denote the random variable of the associated estimator. Adopting the squared error loss function to quantify deviations, a reasonable criterion to measure the performance of an estimator is the *mean-square error* (MSE),

$$\text{MSE} = \mathbb{E}\left[(\hat{\theta} - \theta_o)^2\right], \tag{3.19}$$

where the mean $\mathbb{E}$ is taken over *all* possible training data sets of size N. If the MSE is small, then we expect that, on average, the resulting estimates to be close to the true value. However, this simple and "natural" looking criterion hides some interesting surprises for us. Let us insert the mean value $\mathbb{E}[\hat{\theta}]$ of $\hat{\theta}$ in Eq. (3.19) to get

$$\begin{aligned}\text{MSE} &= \mathbb{E}\left[\left(\left(\hat{\theta} - \mathbb{E}[\hat{\theta}]\right) + \left(\mathbb{E}[\hat{\theta}] - \theta_o\right)\right)^2\right] \\ &= \underbrace{\mathbb{E}\left[\left(\hat{\theta} - \mathbb{E}[\hat{\theta}]\right)^2\right]}_{\text{Variance}} + \underbrace{\left(\mathbb{E}[\hat{\theta}] - \theta_o\right)^2}_{\text{Bias}^2},\end{aligned} \tag{3.20}$$

where, for the second equality, we have taken into account that the mean value of the product of the two involved terms turns out to be zero, as it is readily seen. What Eq. (3.20) suggests is that the MSE consists of two terms. The first one is the variance around the mean value and the second one is due to the bias; that is, the deviation of the mean value of the estimator from the true one.

3.5.1 BIASED OR UNBIASED ESTIMATION?

One may naively think that choosing an estimator that is *unbiased*, as is $\mathbb{E}[\hat{\theta}] = \theta_o$, such that the second term in Eq. (3.20) becomes zero, is a reasonable choice. Adopting an unbiased estimator may also be appealing from the following point of view. Assume that we have L different training sets, each comprising N points. Let us denote each data set by $\mathcal{D}_i$, $i = 1, 2, \ldots, L$. For each one, an estimate $\hat{\theta}_i$, $i = 1, 2, \ldots, L$, will result. Then, form the new estimator by taking the average value,

$$\hat{\theta}^{(L)} := \frac{1}{L}\sum_{i=1}^{L}\hat{\theta}_i.$$

This is also an unbiased estimator, because

$$\mathbb{E}[\hat{\theta}^{(L)}] = \frac{1}{L}\sum_{i=1}^{L}\mathbb{E}[\hat{\theta}_i] = \theta_o.$$

Moreover, assuming that the involved estimators are mutually uncorrelated,

$$\mathbb{E}\left[(\hat{\theta}_i - \theta_o)(\hat{\theta}_j - \theta_o)\right] = 0,$$

[2] Not to be confused with the intercept; the subscript here is "o" and not "0."

and of the same variance, σ^2, then the variance of the new estimator is now much smaller (Problem 3.1),

$$\sigma^2_{\hat{\theta}^{(L)}} = \mathbb{E}\left[\left(\hat{\theta}^{(L)} - \theta_o\right)^2\right] = \frac{\sigma^2}{L}.$$

Hence, by averaging a large number of such unbiased estimators, we expect to get an estimate close to the true value. However, in practice, data is a commodity that is not always abundant. As a matter of fact, very often the opposite is true and one has to be very careful about how to exploit it. In such cases, where one cannot afford to obtain and average a large number of estimators, an unbiased estimator may not necessarily be the best choice. Going back to Eq. (3.20), there is no reason to suggest that by making the second term equal to zero, the MSE (which, after all, is the quantity of interest to us) becomes minimum. Indeed, let us look at Eq. (3.20) from a slightly different view. Instead of computing the MSE for a given estimator, let us replace $\hat{\theta}$ with θ in Eq. (3.20) and compute an estimator that will minimize the MSE with respect to θ, directly. In this case, focusing on unbiased estimators, or $\mathbb{E}[\theta] = \theta_o$, introduces a constraint to the task of minimizing the MSE, and it is well-known that an unconstrained minimization problem always results in loss function values that are less than or equal to any value generated by a constrained counterpart,

$$\min_{\theta} \text{MSE}(\theta) \leq \min_{\theta:\, \mathbb{E}[\theta]=\theta_o} \text{MSE}(\theta), \tag{3.21}$$

where the dependence of MSE on the estimator θ in Eq. (3.21) is explicitly denoted.

Let us denote by $\hat{\theta}_{\text{MVU}}$ a solution of the task $\min_{\theta:\, \mathbb{E}[\theta]=\theta_o} \text{MSE}(\theta)$. It can be readily verified by Eq. (3.20) that $\hat{\theta}_{\text{MVU}}$ is an unbiased estimator of minimum variance. Such an estimator is known as the *minimum variance unbiased estimator* (MVU) and we assume that such an estimator exists. An MVU does not always exist ([20], Problem 3.2). Moreover, if it exists it is unique (Problem 3.3). Motivated by Eq. (3.21), our next goal is to search for a *biased* estimator, which results, hopefully, in a smaller MSE. Let us denote this estimator as $\hat{\theta}_b$. For the sake of illustration, and in order to limit our search for $\hat{\theta}_b$, we consider here only $\hat{\theta}_b$s that are scalar multiples of $\hat{\theta}_{\text{MVU}}$, so that

$$\hat{\theta}_b = (1+\alpha)\hat{\theta}_{\text{MVU}}, \tag{3.22}$$

where $\alpha \in \mathbb{R}$ is a free parameter. Notice that $\mathbb{E}[\hat{\theta}_b] = (1+\alpha)\theta_o$. By substituting Eq. (3.22) into Eq. (3.20) and after some simple algebra we obtain

$$\text{MSE}(\hat{\theta}_b) = (1+\alpha)^2 \text{MSE}(\hat{\theta}_{\text{MVU}}) + \alpha^2\theta_o^2. \tag{3.23}$$

In order to get $\text{MSE}(\hat{\theta}_b) < \text{MSE}(\hat{\theta}_{\text{MVU}})$, α must be in the range (Problem 3.4)

$$-\frac{2\text{MSE}(\hat{\theta}_{\text{MVU}})}{\text{MSE}(\hat{\theta}_{\text{MVU}}) + \theta_o^2} < \alpha < 0. \tag{3.24}$$

It is easy to verify that the previous range implies that $|1+\alpha| < 1$. Hence, $|\hat{\theta}_b| = |(1+\alpha)\hat{\theta}_{\text{MVU}}| < |\hat{\theta}_{\text{MVU}}|$. We can go a step further and try to compute the optimum value of α, which corresponds to the minimum MSE. By taking the derivative of $\text{MSE}(\hat{\theta}_b)$ in Eq. (3.23) with respect to α, it turns out (Problem 3.5) that this occurs for

$$\alpha_* = -\frac{\text{MSE}(\hat{\theta}_{\text{MVU}})}{\text{MSE}(\hat{\theta}_{\text{MVU}}) + \theta_o^2} = -\frac{1}{1 + \frac{\theta_o^2}{\text{MSE}(\hat{\theta}_{\text{MVU}})}}. \tag{3.25}$$

Therefore, we have found a way to obtain the optimum estimator, among those in the set $\{\hat{\theta}_b = (1+\alpha)\hat{\theta}_{\text{MVU}} : \alpha \in \mathbb{R}\}$, which results in minimum MSE. This is true, but as many nice things in life, this is not, in general, realizable. The optimal value for α is given in terms of the unknown, θ_o! However, Eq. (3.25) is useful in a number of other ways. First, there are cases where the MSE is proportional to θ_o^2; hence, this formula can be used. Also, for certain cases, it can be used to provide useful bounds [19]. Moreover, as far as we are concerned in this book, it says something very important. If we want to do better than the MVU, then, looking at the text after Eq. (3.24), a possible way is to *shrink* the norm of the MVU estimator. Shrinking the norm is a way of introducing bias into an estimator. We will discuss ways to achieve this in Section 3.8 and later on in Chapters 6 and 11.

Note that what we have said so far is readily generalized to parameter vectors. An unbiased parameter vector satisfies

$$\mathbb{E}[\boldsymbol{\theta}] = \boldsymbol{\theta}_o,$$

and the MSE around the true value, $\boldsymbol{\theta}_o$, is defined as

$$\text{MSE} = \mathbb{E}\left[(\boldsymbol{\theta} - \boldsymbol{\theta}_o)^T(\boldsymbol{\theta} - \boldsymbol{\theta}_o)\right].$$

Looking carefully at the previous definition reveals that the MSE for a parameter vector is the sum of the MSEs of the components, $\theta_i,\ i = 1, 2, \ldots, l$, around the corresponding true values θ_{oi}.

3.6 THE CRAMÉR-RAO LOWER BOUND

In the previous sections, we saw how one can improve upon the performance of the MVU estimator, provided that this exists and it is also known. However, how can one know that an unbiased estimator, that has been obtained, is also of minimum variance? The goal of this section is to introduce a criterion that can provide such information.

The *Cramér-Rao lower bound* [8, 31] is an elegant theorem and one of the most well-known techniques used in statistics. It provides a lower bound on the variance of *any* unbiased estimator. This is very important because (a) it offers the means to assert whether an unbiased estimator has minimum variance, which, of course, in this case coincides with the corresponding MSE in Eq. (3.20), and (b) if this is not the case, it can be used to indicate how far away the performance of an unbiased estimator is from the optimal one, and finally (c) it provides the designer with a tool to know the best possible performance that can be achieved by an unbiased estimator. Because our main purpose here is to focus on the insight and physical interpretation of the method, we will deal with the simple case where our unknown parameter is a real number. The general form of the theorem, involving vectors, is given in Appendix B.

We are looking for a bound of the variance of an unbiased estimator, whose randomness is due to the randomness of the training data, as we change from one set to another. Thus, it does not come as a surprise that the bound involves the joint pdf of the data, parameterized in terms of the unknown parameter, θ. Let $\mathcal{X} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N\}$ denote the set of N observations, corresponding to a random vector,[3] $\mathbf{x}$, that depends on the unknown parameter. Also, let the respective joint pdf of the observations be denoted as $p(\mathcal{X}; \theta)$.

[3] Note, here, that $\mathbf{x}$ is treated as a random quantity in a general setting, and not necessarily in the context of the regression/classification tasks.

Theorem 3.1. *It is assumed that the joint pdf satisfies the following regularity condition:*

$$\mathbb{E}\left[\frac{\partial \ln p(\mathcal{X};\theta)}{\partial \theta}\right] = 0, \quad \forall \theta. \tag{3.26}$$

This regularity condition is a weak one and holds for most of the cases in practice (Problem 3.6). Then, the variance of any unbiased estimator, $\hat{\theta}$, must satisfy the following inequality:

$$\boxed{\sigma_{\hat{\theta}}^2 \geq \frac{1}{I(\theta)} : \quad \textit{Cramér-Rao Lower Bound},} \tag{3.27}$$

where

$$I(\theta) := -\mathbb{E}\left[\frac{\partial^2 \ln p(\mathcal{X};\theta)}{\partial \theta^2}\right]. \tag{3.28}$$

Moreover, the necessary and sufficient condition for obtaining an unbiased estimator that achieves the bound is the existence of a function $g(\cdot)$ such that for all possible values of θ,

$$\frac{\partial \ln p(\mathcal{X};\theta)}{\partial \theta} = I(\theta)\left(g(\mathcal{X}) - \theta\right). \tag{3.29}$$

The MVU estimate is then given by

$$\hat{\theta} = g(\mathcal{X}) := g(\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N), \tag{3.30}$$

and the variance of the respective estimator is equal to $1/I(\theta)$.

When an MVU estimator attains the Cramér-Rao bound, we say that it is *efficient*. All the expectations before are taken with respect to $p(\mathcal{X};\theta)$. The interested reader may find more on the topic in more specialized books on statistics [20, 27, 34].

Example 3.2. Let us consider the simplified version of the linear regression model in Eq. (3.10), where the regressor is real valued and the bias term is zero,

$$y_n = \theta x + \eta_n, \tag{3.31}$$

where we have explicitly denoted the dependence on n, which runs over the number of available observations. Note that in order to further simplify the discussion, we have assumed that our N observations are the result of different realizations of the noise variable only, and that we have kept the value of the input, x, constant, which can be considered to be equal to one, without harming generality; that is, our task degenerates to that of estimating a parameter from its noisy measurements. Thus, for this case, the observations are the scalar outputs, y_n, $n = 1, 2, \ldots, N$, which we consider to be the components of a vector, $\boldsymbol{y} \in \mathbb{R}^N$. We further assume that η_n are samples of a Gaussian white noise with zero mean and variance equal to σ_η^2. Then, the joint pdf of the output observations is given by

$$p(\mathbf{y};\theta) = \prod_{n=1}^{N} \frac{1}{\sqrt{2\pi\sigma_\eta^2}} \exp\left(-\frac{(y_n - \theta)^2}{2\sigma_\eta^2}\right), \tag{3.32}$$

or

$$\ln p(\boldsymbol{y};\theta) = -\frac{N}{2}\ln(2\pi\sigma_\eta^2) - \frac{1}{2\sigma_\eta^2}\sum_{n=1}^{N}(y_n-\theta)^2. \tag{3.33}$$

We will derive the corresponding Cramér-Rao bound. Taking the derivative of the logarithm with respect to θ we have

$$\frac{\partial \ln p(\boldsymbol{y};\theta)}{\partial\theta} = \frac{1}{\sigma_\eta^2}\sum_{n=1}^{N}(y_n-\theta) = \frac{N}{\sigma_\eta^2}(\bar{y}-\theta), \tag{3.34}$$

where

$$\bar{y} := \frac{1}{N}\sum_{n=1}^{N} y_n,$$

that is, the sample mean of the measurements. The second derivative, as required by the theorem, is given by

$$\frac{\partial^2 \ln p(\boldsymbol{y};\theta)}{\partial\theta^2} = -\frac{N}{\sigma_\eta^2},$$

and hence,

$$I(\theta) = \frac{N}{\sigma_\eta^2}. \tag{3.35}$$

Equation (3.34) is in the form of Eq. (3.29), with $g(\boldsymbol{y}) = \bar{y}$; thus, an efficient estimator can be obtained and the lower bound of the variance of any unbiased estimator, for our data model of Eq. (3.31), is

$$\sigma_{\hat{\theta}}^2 \geq \frac{\sigma_\eta^2}{N}. \tag{3.36}$$

We can easily verify that the corresponding estimator, $\bar{\mathrm{y}}$ is indeed an unbiased one under the adopted model of Eq. (3.31),

$$\mathbb{E}[\bar{\mathrm{y}}] = \frac{1}{N}\sum_{n=1}^{N}\mathbb{E}[\mathrm{y}_n] = \frac{1}{N}\sum_{n=1}^{N}\mathbb{E}[\theta+\eta_n] = \theta.$$

Moreover, the previous formula, combined with Eq. (3.34), also establishes the regularity condition, as it is required by the Cramér-Rao theorem.

The bound in Eq. (3.36) is a very natural result. The Cramér-Rao lower bound depends on the variance of the noise source. The higher this is, and therefore the higher the uncertainty of each measurement with respect to the value of the true parameter is, the higher the minimum variance of an estimator is expected to be. On the other hand, as the number of observations increases and more "information" is disclosed to us, the uncertainty decreases and we expect the variance of our estimator to decrease.

Having obtained the lower bound for our task, let us return our attention to the LS estimator for the specific regression model of Eq. (3.31). This results from Eq. (3.13), by setting $x_n = 1$ and a

simple inspection shows that the LS estimate is nothing but the sample mean, $\bar{y}$, of the observations. Furthermore, the variance of the corresponding estimator is given by

$$\begin{aligned}\sigma_{\bar{y}}^2 &= \mathbb{E}\left[(\bar{\mathrm{y}}-\theta)^2\right] = \mathbb{E}\left[\frac{1}{N^2}\left(\sum_{n=1}^{N}(\mathrm{y}_n-\theta)\right)^2\right] \\ &= \frac{1}{N^2}\,\mathbb{E}\left[\left(\sum_{n=1}^{N}\eta_n\right)^2\right] = \frac{1}{N^2}\,\mathbb{E}\left[\sum_{i=1}^{N}\eta_i\sum_{j=1}^{N}\eta_j\right] \\ &= \frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N}\mathbb{E}[\eta_i\eta_j] = \frac{\sigma_\eta^2}{N},\end{aligned}$$

which coincides with our previous finding via the use of the Cramér-Rao theorem. In other words, for this particular task and having assumed that the noise is Gaussian, the LS estimator $\bar{\mathrm{y}}$ is an MVU estimator and it attains the Cramér-Rao bound. However, if the input is not fixed, but it also varies from experiment to experiment and the training data become (y_n, x_n), then the LS estimator attains the Cramér-Rao bound only asymptotically, for large values of N (Problem 3.7). Moreover, it has to be pointed out that if the assumptions for the noise being Gaussian *and* white are not valid, then the LS estimator is not efficient anymore.

It turns out that this result, which has been obtained for the real axis case, is also true for the general regression model given in Eq. (3.10) (Problem 3.8). We will return to the properties of the LS estimator in more detail in Chapter 6.

Remarks 3.2.

- The Cramér-Rao bound is not the only one that is available in the literature. For example, the Bhattacharya bound makes use of higher order derivatives of the pdf. It turns out that in cases where an efficient estimator does not exist, then the Bhattacharya bound is tighter compared to the Cramér-Rao one, with respect to the variance of the MVU estimator [27]. Other bounds also exist [21]; however, the Cramér-Rao bound is the easiest to determine.

3.7 SUFFICIENT STATISTIC

If an efficient estimator does not exist, this does not necessarily mean that the MVU estimator cannot be determined. It may exist, but it will not be an efficient one, in the sense that it does not satisfy the Cramér-Rao bound. In such cases, the notion of *sufficient statistic* and the Rao-Blackwell theorem come into the picture.[4] Although these are beyond the focus of this book, they are mentioned here in order to provide a more complete picture of the topic.

The notion of sufficient statistic is due to Sir Ronald Aylmer Fisher (1890-1962). Fisher was an English statistician and biologist who made a number of fundamental contributions that laid out many of the foundations of modern statistics. Besides statistics, he made important contributions in genetics.

[4] It must be pointed out that the use of sufficient statistic in statistics extends much beyond the search for MVUs.

In short, given a random vector, $\mathbf{x}$, which depends on a parameter θ, a sufficient statistic for the unknown parameter is a function

$$T(\mathcal{X}) := T(\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N),$$

of the respective observations, which contains *all* information about θ. From a mathematical point of view, a statistic $T(\mathcal{X})$ is said to be sufficient for the parameter θ if the conditional joint pdf

$$p\left(\mathcal{X}|T(\mathcal{X});\theta\right),$$

does not depend on θ. In such a case, it becomes apparent that $T(\mathcal{X})$ must provide *all* information about θ, which is contained in the set $\mathcal{X}$. Once $T(\mathcal{X})$ is known, $\mathcal{X}$ is no longer needed, because no further information can be extracted from it; this justifies the name of "sufficient statistic." The concept of sufficient statistic is also generalized to parameter vectors $\boldsymbol{\theta}$. In such a case, the sufficient statistic may be a *set* of functions, called a *jointly sufficient statistic*. Typically, there are as many functions as there are parameters; in a slight abuse of notation, we will still write $T(\mathcal{X})$ to denote this set (vector of) functions.

A very important theorem, which facilitates the search for a sufficient statistic in practice, is the following [27].

Theorem 3.2 (Factorization Theorem)**.** *A statistic $T(\mathcal{X})$ is sufficient if and only if the respective joint pdf can be factored as*

$$p(\mathcal{X};\boldsymbol{\theta}) = h(\mathcal{X})g\left(T(\mathcal{X}),\boldsymbol{\theta}\right).$$

That is, the joint pdf is factored into two parts: one part that depends only on the statistic and the parameters and a second part that is independent of the parameters. The theorem is also known as the Fisher-Neyman factorization theorem.

Once a sufficient statistic has been found and under certain conditions related to the statistic, the Rao-Blackwell theorem determines the MVU estimator (MVUE) by taking the expectation conditioned on $T(\mathcal{X})$. A by-product of this theorem is that if an unbiased estimator is expressed *solely* in terms of the sufficient statistic, then it is necessarily the unique MVUE [23]. The interested reader can obtain more on these issues from [20, 21, 27].

Example 3.3. Let x be a Gaussian, $\mathcal{N}(\mu, \sigma^2)$, random variable and let the set of observations be $\mathcal{X} = \{x_1, x_2, \ldots, x_N\}$. Assume μ to be the unknown parameter. Show that

$$S_\mu = \frac{1}{N}\sum_{n=1}^{N} x_n,$$

is a sufficient statistic for the parameter μ.

The joint pdf is given by

$$p(\mathcal{X};\mu) = \frac{1}{(2\pi\sigma^2)^{\frac{N}{2}}}\exp\left(-\frac{1}{2\sigma^2}\sum_{n=1}^{N}(x_n-\mu)^2\right).$$

Plugging the obvious identity,

$$\sum_{n=1}^{N}(x_n-\mu)^2 = \sum_{n=1}^{N}(x_n-S_\mu)^2 + N(S_\mu-\mu)^2,$$

into the joint pdf, we obtain

$$p(\mathcal{X};\mu)=\frac{1}{(2\pi\sigma^2)^{\frac{N}{2}}}\exp\left(-\frac{1}{2\sigma^2}\sum_{n=1}^{N}(x_n-S_\mu)^2\right)\exp\left(-\frac{N}{2\sigma^2}(S_\mu-\mu)^2\right),$$

which, according to the factorization theorem, proves the claim.

In a similar way, one can prove (Problem 3.9) that if the unknown parameter is the variance σ^2, then $\bar{S}_{\sigma^2}:=\frac{1}{N}\sum_{n=1}^{N}(x_n-\mu)^2$ is a sufficient statistic, and if both μ and σ^2 are unknown then a sufficient statistic is the set (S_μ, S_{σ^2}), where

$$S_{\sigma^2}=\frac{1}{N}\sum_{n=1}^{N}(x_n-S_\mu)^2.$$

That is, in this case, all information concerning the unknown set of parameters that can be possibly extracted from the available N observations, can be fully recovered by considering only the sum of the observations and the sum of their squares.

3.8 REGULARIZATION

We have already seen that the LS estimator is a minimum variance unbiased estimator, under the assumptions of linearity of the regression model and in the presence of a Gaussian white noise source. We also know that one can improve the performance by shrinking the norm of the MVU estimator. There are additional ways to achieve this goal and they will be discussed later on in this book. In this section, we focus on one possibility. Moreover, we will see that trying to keep the norm of the solution small serves important needs in the context of machine learning.

Regularization is a mathematical tool to impose a priori information on the structure of the solution, which comes as the outcome of an optimization task. Regularization was first suggested by the great Russian mathematician Andrey Nikolayevich Tychonoff (sometimes spelled Tikhonov) for the solution of integral equations. Sometimes, it is also referred as Tychonoff-Phillips regularization, to honor David Phillips as well, who developed the method independently [29, 37].

In the context of our task and in order to shrink the norm of the parameter vector estimate, we can reformulate the LS minimization task, given in Eq. (3.12), as

$$\text{minimize:}\qquad J(\boldsymbol{\theta})=\sum_{n=1}^{N}\left(y_n-\boldsymbol{\theta}^T\boldsymbol{x}_n\right)^2, \tag{3.37}$$

$$\text{subject to:}\qquad \|\boldsymbol{\theta}\|^2\leq\rho, \tag{3.38}$$

where $\|\cdot\|$ stands for the Euclidean norm of a vector. In this way, we do not allow the LS criterion to be completely "free" to reach a solution, but we *limit* the space in which to search for it. Obviously, using different values of ρ, we can achieve different levels of shrinkage. As we have already discussed, the optimal value of ρ cannot be analytically obtained, and one has to experiment in order to select an estimator that results in a good performance. For the LS loss function and the constraint used before, the optimization task can equivalently be written as [5, 6]

$$\boxed{\text{minimize:} \qquad L(\boldsymbol{\theta}, \lambda) = \sum_{n=1}^{N} \left(y_n - \boldsymbol{\theta}^T \boldsymbol{x}_n\right)^2 + \lambda \|\boldsymbol{\theta}\|^2 : \quad \text{Ridge Regression.}} \tag{3.39}$$

It turns out that, for specific choices of $\lambda \geq 0$ and ρ, the two tasks are equivalent. Note that this new cost function, $L(\boldsymbol{\theta}, \lambda)$, involves one term that measures the model *misfit* and a second one that quantifies the *size* of the norm of the parameter vector. It is straightforward to see that taking the gradient of L in Eq. (3.39) with respect to $\boldsymbol{\theta}$ and equating to zero, we obtain the *regularized LS* solution for the linear regression task of Eq. (3.13)

$$\left(\sum_{n=1}^{N} \boldsymbol{x}_n \boldsymbol{x}_n^T + \lambda I\right) \hat{\boldsymbol{\theta}} = \sum_{n=1}^{N} y_n \boldsymbol{x}_n, \tag{3.40}$$

where I is the identity matrix of appropriate dimensions. The presence of λ biases the new solution away from that which would have been obtained from the unregularized LS formulation. The task is also known as *ridge regression*. Ridge regression attempts to reduce the norm of the estimated vector and *at the same time* tries to keep the sum of squared errors small; in order to achieve this *combined* goal, the vector components, θ_i, are modified in such a way so that the contribution in the misfit measuring term, from the less informative directions in the input space, is minimized. We will return to this in more detail in Chapter 6. Ridge regression was first introduced in [18].

It has to be emphasized that in practice, the bias parameter, θ_0, is left out from the norm in the regularization term; penalization of the bias would make the procedure dependent on the origin chosen for y. Indeed, it is easily checked out that adding a constant term to each one of the output values, y_n, in the cost function, would not result in just a shift of the predictions by the same constant, if the bias term is included in the norm. Hence, usually, ridge regression is formulated as

$$\text{minimize} \quad L(\boldsymbol{\theta}, \lambda) = \sum_{n=1}^{N} \left(y_n - \theta_0 - \sum_{i=1}^{l} \theta_i x_{ni}\right)^2 + \lambda \sum_{i=1}^{l} |\theta_i|^2. \tag{3.41}$$

It turns out (Problem 3.10) that minimizing Eq. (3.41) with respect to $\theta_i, i = 1, 2, \ldots, l$, is equivalent with minimizing Eq. (3.39) using *centered* data and neglecting the intercept. That is, one solves the task

$$\text{minimize} \quad L(\boldsymbol{\theta}, \lambda) = \sum_{n=1}^{N} \left((y_n - \bar{y}) - \sum_{i=1}^{l} \theta_i (x_{ni} - \bar{x}_i)\right)^2 + \lambda \sum_{i=1}^{l} |\theta_i|^2, \tag{3.42}$$

and the estimate of θ_0 in Eq. (3.41) is given in terms of the obtained estimates, $\hat{\theta}_i$,

$$\hat{\theta}_0 = \bar{y} - \sum_{i=1}^{l} \hat{\theta}_i \bar{x}_i,$$

where

$$\bar{y} = \frac{1}{N} \sum_{n=1}^{N} y_n \quad \text{and} \quad \bar{x}_i = \frac{1}{N} \sum_{n=1}^{N} x_{ni}, \ i = 1, 2, \ldots, l.$$

In other words, $\hat{\theta}_0$ compensates for the differences between the sample means of the output and input variables. Note that similar arguments hold true if the Euclidean norm, used in Eq. (3.40) as a regularizer, is replaced by other norms, such as the ℓ_1 or in general ℓ_p, $p > 1$ norms, Chapter 9.

From a different viewpoint, reducing the norm can be considered as an attempt to "simplify" the structure of the estimator, because a smaller number of components of the regressor now have an important say. This viewpoint becomes more clear if one considers nonlinear models, as was discussed in Section 3.2. In this case, the existence of the norm of the respective parameter vector in Eq. (3.39) forces the model to get rid of the less important terms in the nonlinear expansion, $\sum_{k=1}^{K} \theta_k \phi_k(\boldsymbol{x})$, and effectively pushes K to lower values.

Although in the current context, the complexity issue emerges in a rather disguised form, one can make it a major player in the game by choosing to use different functions and norms for the regularization term; and there are many reasons that justify such choices.

Inverse problems: Ill-conditioning and overfitting

Most tasks in machine learning belong to the so-called *inverse problems*. The latter term encompasses all the problems where one has to infer/predict/estimate the values of a model based on a set of available output/input observations-training data. In a less mathematical terminology, in inverse problems one has to unravel unknown causes from known effects; in other words, to reverse the cause-effect relations. Inverse problems are typically *ill-posed*, as opposed to the *well-posed* ones. Well-posed problems are characterized by (a) the existence of a solution, (b) the uniqueness of the solution, and (c) the stability of the solution. The latter condition is usually violated in machine learning problems. This means that the obtained solution may be very sensitive to changes of the training set. *Ill conditioning* is another term used to describe this sensitivity. The reason for this behavior is that the model used to describe the data can be complex, in the sense that the number of the unknown free parameters is large with respect to the number of data points. The "face" with which this problem manifests itself in machine learning is known as *overfitting*. This means that during training, the estimated parameters of the unknown model learn too much about the idiosyncrasies of the specific training data set, and the model performs badly when it deals with another set of data, other than that used for the training. As a matter of fact, the MSE criterion discussed in Section 3.5 attempts to quantify exactly this way; that is, the mean deviation of the obtained estimates from the true value by changing the training sets.

When the number of training samples is small with respect to the number of the unknown parameters, the available information is not enough to "reveal" a sufficiently good model that fits the data, and it can be misleading due to the presence of the noise and possible outliers. Regularization is an elegant and efficient tool to cope with the complexity of the model; that is, to make it less complex, more smooth. There are different ways to achieve this. One way is by constraining the norm of the unknown vector, as ridge regression does. When dealing with more complex, compared to linear, models, one can use constraints on the smoothness of the involved nonlinear function; for example, by involving derivatives of the model function in the regularization term. Also, regularization can help when the adopted model and the number of training points are such that no solution is possible. For example, in our LS linear regression task of Eq. (3.13), if the number, N, of the training points is less than the dimension of the regressors $\boldsymbol{x}_n$, then the $l \times l$ matrix, $\bar{\Sigma} = \sum_n \boldsymbol{x}_n \boldsymbol{x}_n^T$, is not invertible. Indeed, each term in the summation is the outer product of a vector with itself and hence it is a matrix of rank one.

Thus, as we know from linear algebra, we need at least l linearly independent terms of such matrices to guarantee that the sum is of full rank, hence invertible. However, in ridge regression, this can be bypassed, because the presence of λI in Eq. (3.40) guarantees that the left-hand matrix is invertible. Furthermore, the presence of λI can also help when $\bar{\Sigma}$ is invertible but it is ill-conditioned. Usually in such cases, the resulting LS solution has a very large norm and, thus, it is meaningless. Regularization helps to replace the original ill-conditioned problem with a "nearby" one, which is well-conditioned and whose solution approximates the target one.

Another example where regularization can help to obtain a solution, and, more important, a unique solution to an otherwise unsolvable problem, is when the model's order is large compared to the number of data, albeit we know that it is sparse. That is, only a very small percentage of the model's parameters are nonzero. For such a task, a standard LS linear regression approach has no solution. However, regularizing the LS loss function using the ℓ_1 norm of the parameters' vector can lead to a unique solution; the ℓ_1 norm of a vector comprises the sum of the absolute values of its components. This problem will be considered in Chapters 9 and 10.

Regularization is closely related to the task of using priors in Bayesian learning, as we will discuss in Section 3.11. Finally, note that regularization is not a panacea for facing the problem of overfitting. As a matter of fact, selecting the right set of functions $\mathcal{F}$ in Eq. (3.3) is the first crucial step. The issue of the complexity of an estimator and the consequences on its "average" performance, as this is measured over all possible data sets, is discussed in Section 3.9.

Example 3.4. The goal of this example is to demonstrate that the estimator obtained via the ridge regression can score a better MSE performance compared to the unconstrained LS solution. Let us consider, once again, the model exposed in Example 3.2, and assume that the data are generated according to

$$y_n = \theta_o + \eta_n, \quad n = 1, 2, \ldots, N,$$

where, for simplicity, we have assumed that the regressors $x_n \equiv 1$, and η_n, $n = 1, 2, \ldots, N$, are i.i.d. zero-mean Gaussian noise samples of variance σ_η^2.

We have already seen in Example 3.2 that the solution to the LS parameter estimation task is the sample mean $\hat{\theta}_{\text{MVU}} = \frac{1}{N}\sum_{n=1}^{N} y_n$. We have shown also that this solution scores an MSE of σ_η^2/N and under the Gaussian assumption for the noise it achieves the Cramér-Rao bound. The question now is whether a biased estimator, $\hat{\theta}_b$, which corresponds to the solution of the associated ridge regression task, can achieve an MSE lower than $\text{MSE}(\hat{\theta}_{\text{MVU}})$.

It can be readily verified that Eq. (3.40), adapted to the needs of the current linear regression scenario, results in

$$\hat{\theta}_b(\lambda) = \frac{1}{N+\lambda}\sum_{n=1}^{N} y_n = \frac{N}{N+\lambda}\hat{\theta}_{\text{MVU}},$$

where we have explicitly expressed the dependence of the estimate $\hat{\theta}_b$ on the regularization parameter λ. Notice that for the associated estimator we have, $\mathbb{E}[\hat{\theta}_b(\lambda)] = \frac{N}{N+\lambda}\theta_o$.

A simple inspection of the previous relation takes us back to the discussion related to Eq. (3.22). Indeed, by following a sequence of similar steps to Section 3.5.1, one can verify (see Problem 3.11) that the minimum value of $\text{MSE}(\hat{\theta}_b)$ is

$$\mathrm{MSE}(\hat{\theta}_b(\lambda_*)) = \frac{\frac{\sigma_\eta^2}{N}}{1 + \frac{\sigma_\eta^2}{N\theta_o^2}} < \frac{\sigma_\eta^2}{N} = \mathrm{MSE}(\hat{\theta}_{\mathrm{MVU}}), \tag{3.43}$$

attained at $\lambda_* = \sigma_\eta^2/\theta_o^2$. The answer to the question whether the ridge regression estimate offers an improvement to the MSE performance is therefore positive in the current context. As a matter of fact, there *always* exists a $\lambda > 0$ such that the ridge regression estimate, which solves the general task of Eq. (3.39), achieves an MSE lower than the one corresponding to the MVU estimate [4, Section 8.4].

We will now demonstrate the previous theoretical findings via some simulations. To this end, the true value of the model was chosen to be $\theta_o = 10^{-2}$. The noise was Gaussian of zero mean value and variance $\sigma_\eta^2 = 0.1$. The number of generated samples was $N = 100$. Note that this is quite large, compared to a single parameter we have to estimate. The previous values imply that $\theta_o^2 < \sigma_\eta^2/N$. Then, it can be shown that, for any value of $\lambda > 0$, we can obtain a value for $\mathrm{MSE}(\hat{\theta}_b(\lambda))$, which is smaller than that of $\mathrm{MSE}(\hat{\theta}_{\mathrm{MVU}})$ (see Problem 3.11). This is verified by the values shown in Table 3.1. To compute the MSE values in the table, the expectation operation in the definition in Eq. (3.19) was approximated by the respective sample mean. To this end, the experiment was repeated L times and the MSE was computed as

$$\mathrm{MSE} \approx \frac{1}{L}\sum_{i=1}^{L}(\hat{\theta}_i - \theta_o)^2.$$

To get accurate results, we perform $L = 10^6$ trials. The corresponding MSE value for the unconstrained LS task is equal to $\mathrm{MSE}(\hat{\theta}_{\mathrm{MVU}}) = 1.00108 \times 10^{-3}$. Observe that substantial improvements can be attained when using regularization, in spite of the relatively large number of training data.

However, the percentage of performance improvement depends heavily on the specific values that define the model, as Eq. (3.43) suggests. For example, if $\theta_o = 0.1$, the obtained values from the experiments were $\mathrm{MSE}(\hat{\theta}_{\mathrm{MVU}}) = 1.00061 \times 10^{-3}$ and $\mathrm{MSE}(\hat{\theta}_b(\lambda_*)) = 9.99578 \times 10^{-4}$. The theoretical ones, as computed from Eq. (3.43), are 1×10^{-3} and 9.99001×10^{-4}, respectively. The improvement obtained by using the ridge regression is now rather insignificant.

Table 3.1 Attained Values of MSE for Ridge Regression and Different Values of the Regularization Parameter

λ	$\mathrm{MSE}(\hat{\theta}_b(\lambda))$
0.1	9.99082×10^{-4}
1.0	9.79790×10^{-4}
100.0	2.74811×10^{-4}
$\lambda_* = 10^3$	9.09671×10^{-5}

The attained MSE for the unconstrained LS estimate was $MSE(\hat{\theta}_{MVU}) = 1.00108 \times 10^{-3}$.

3.9 THE BIAS-VARIANCE DILEMMA

This section goes one step beyond Section 3.5. There, the MSE criterion was used to quantify the performance with respect to the unknown parameter. Such a setting was useful, in order to help us understand some trends and also better digest the notions of "biased" versus "unbiased" estimation. Here, although the criterion will be the same, it will be used in a more general setting. To this end, we shift our interest from the unknown parameter to the dependent variable and our goal becomes to obtain an estimator of the value y, given a measurement of the regressor vector, $\mathbf{x} = \boldsymbol{x}$. Let us first consider the more general form of regression,

$$\mathrm{y} = g(\mathbf{x}) + \eta, \tag{3.44}$$

where, once more, we have assumed that the dependent variable takes values in the real axis, $\mathrm{y} \in \mathbb{R}$, for simplicity and without harm of the generality. The first question to be addressed is whether there exists an estimator that guarantees minimum MSE performance.

3.9.1 MEAN-SQUARE ERROR ESTIMATION

Our goal is to obtain an estimate $\hat{g}(\boldsymbol{x})$ of the unknown (nonlinear in general) function $g(\boldsymbol{x})$. This problem can be cast in the context of the more general estimation task setting.

Let the *jointly distributed* random variables, y, $\mathbf{x}$. Then, given a set of observations, $\mathbf{x} = \boldsymbol{x} \in \mathbb{R}^l$, the task is to obtain a function $\hat{y} := \hat{g}(\boldsymbol{x}) \in \mathbb{R}$, such that

$$\hat{g}(\boldsymbol{x}) = \arg\min_{f:\mathbb{R}^l \to \mathbb{R}} \mathbb{E}\left[(\mathrm{y} - f(\boldsymbol{x}))^2\right], \tag{3.45}$$

where the expectation is taken with respect to the conditional probability of y given the value of $\boldsymbol{x}$; in other words, $p(y|\boldsymbol{x})$.

We will show that the optimal estimate is the mean value of y, or

$$\boxed{\hat{g}(\boldsymbol{x}) = \mathbb{E}\left[\mathrm{y}|\boldsymbol{x}\right] := \int_{-\infty}^{+\infty} y p(y|\boldsymbol{x})\, \mathrm{d}y : \quad \text{Optimal MSE Estimate.}} \tag{3.46}$$

Proof. We have that

$$\begin{aligned}
\mathbb{E}\left[\left(\mathrm{y} - f(\boldsymbol{x})\right)^2\right] &= \mathbb{E}\left[\left(\mathrm{y} - \mathbb{E}[\mathrm{y}|\boldsymbol{x}] + \mathbb{E}[\mathrm{y}|\boldsymbol{x}] - f(\boldsymbol{x})\right)^2\right] \\
&= \mathbb{E}\left[\left(\mathrm{y} - \mathbb{E}[\mathrm{y}|\boldsymbol{x}]\right)^2\right] + \mathbb{E}\left[\left(\mathbb{E}[\mathrm{y}|\boldsymbol{x}] - f(\boldsymbol{x})\right)^2\right] \\
&\quad + 2\,\mathbb{E}\left[\left(\mathrm{y} - \mathbb{E}[\mathrm{y}|\boldsymbol{x}]\right)\left(\mathbb{E}[\mathrm{y}|\boldsymbol{x}] - f(\boldsymbol{x})\right)\right],
\end{aligned}$$

where the dependence of the expectation on $\boldsymbol{x}$ has been suppressed for notational convenience. It is readily seen that the last (product) term on the right-hand side is zero, hence, we are left with the following:

$$\mathbb{E}\left[\left(\mathrm{y} - f(\boldsymbol{x})\right)^2\right] = \mathbb{E}\left[\left(\mathrm{y} - \mathbb{E}[\mathrm{y}|\boldsymbol{x}]\right)^2\right] + \left(\mathbb{E}[\mathrm{y}|\boldsymbol{x}] - f(\boldsymbol{x})\right)^2, \tag{3.47}$$

where we have taken into account that, for fixed $\boldsymbol{x}$, the terms $\mathbb{E}[y|\boldsymbol{x}]$ and $f(\boldsymbol{x})$ are not random variables. From Eq. (3.47) we finally obtain our claim,

$$\mathbb{E}\left[\left(\mathrm{y} - f(\boldsymbol{x})\right)^2\right] \geq \mathbb{E}\left[\left(\mathrm{y} - \mathbb{E}[\mathrm{y}|\boldsymbol{x}]\right)^2\right]. \tag{3.48}$$

□

Note that this is a very elegant result. The optimal estimate, in the MSE sense, of the unknown function is given as $\hat{g}(\boldsymbol{x}) = \mathbb{E}[\mathrm{y}|\boldsymbol{x}]$. Sometimes, the latter is also known as the *regression of* y *conditioned on* $\mathbf{x} = \boldsymbol{x}$. This is, in general, a nonlinear function. It can be shown that if $(\mathrm{y}, \mathbf{x})$ take values in $\mathbb{R} \times \mathbb{R}^l$ and are jointly Gaussian, then the optimal MSE estimator $\mathbb{E}[\mathrm{y}|\boldsymbol{x}]$ is a linear (affine) function of $\boldsymbol{x}$.

The previous results generalize to the case where $\mathbf{y}$ is a random vector that takes values in $\mathbb{R}^k$. The optimal MSE estimate, given the values of $\mathbf{x} = \boldsymbol{x}$, is equal to

$$\hat{\boldsymbol{g}}(\boldsymbol{x}) = \mathbb{E}[\mathbf{y}|\boldsymbol{x}],$$

where now $\hat{\boldsymbol{g}}(\boldsymbol{x}) \in \mathbb{R}^k$ (Problem 3.13). Moreover, if $(\mathbf{y}, \mathbf{x})$ are jointly Gaussian random vectors, the MSE optimal estimate is also an affine function of $\boldsymbol{x}$ (Problem 3.14).

The findings of this subsection can be fully justified by physical reasoning. Assume, for simplicity, that the noise source in Eq. (3.44) is of zero mean. Then, for a fixed value $\mathbf{x} = \boldsymbol{x}$, we have that $\mathbb{E}[\mathrm{y}|\boldsymbol{x}] = g(\boldsymbol{x})$ and the respective MSE is equal to

$$\mathrm{MSE} = \mathbb{E}\left[(\mathrm{y} - \mathbb{E}[\mathrm{y}|\boldsymbol{x}])^2\right] = \sigma_\eta^2. \tag{3.49}$$

No other function of $\boldsymbol{x}$ can do better, because the optimal one achieves an MSE equal to the noise variance, which is irreducible; it represents the intrinsic uncertainty of the system. As Eq. (3.47) suggests, any other function, $f(\boldsymbol{x})$, will result in an MSE larger by the factor $(\mathbb{E}[\mathrm{y}|\boldsymbol{x}] - f(\boldsymbol{x}))^2$, which corresponds to the deviation from the optimal one.

3.9.2 BIAS-VARIANCE TRADEOFF

We have just seen that the optimal estimate, in the MSE sense, of the dependent variable in a regression task is given by the conditional expectation $\mathbb{E}[\mathrm{y}|\boldsymbol{x}]$. In practice, any estimator is computed based on a specific training data set, say $\mathcal{D}$. Let us make the dependence on the training set explicit and express the estimate as a function of $\boldsymbol{x}$ parameterized on $\mathcal{D}$, or $f(\boldsymbol{x}; \mathcal{D})$. A reasonable measure to quantify the performance of an estimator is its mean-square deviation from the optimal one, expressed by $\mathbb{E}_{\mathcal{D}}[(f(\boldsymbol{x}; \mathcal{D}) - \mathbb{E}[\mathrm{y}|\boldsymbol{x}])^2]$, where the mean is taken with respect to all possible training sets, because each one results in a different estimate. Following a similar path as for Eq. (3.20), we obtain

$$\boxed{\begin{aligned}\mathbb{E}_{\mathcal{D}}\left[\left(f(\boldsymbol{x}; \mathcal{D}) - \mathbb{E}[\mathrm{y}|\boldsymbol{x}]\right)^2\right] = &\underbrace{\mathbb{E}_{\mathcal{D}}\left[\left(f(\boldsymbol{x}; \mathcal{D}) - \mathbb{E}_{\mathcal{D}}[f(\boldsymbol{x}; \mathcal{D})]\right)^2\right]}_{\text{Variance}} + \\ &\underbrace{\left(\mathbb{E}_{\mathcal{D}}[f(\boldsymbol{x}; \mathcal{D})] - \mathbb{E}[\mathrm{y}|\boldsymbol{x}]\right)^2}_{\text{Bias}^2}.\end{aligned}} \tag{3.50}$$

As was the case for the MSE parameter estimation task when changing from one training set to another, the mean-square deviation from the optimal estimator comprises two terms. The first one is contributed from the variance of the estimator around its own mean value and the second one from the difference of the mean from the optimal estimate; in other words, the bias. It turns out that one *cannot* make *both* terms small simultaneously. For a fixed number of training points, N, in the data sets $\mathcal{D}$, trying to minimize the

variance term results in an increase of the bias term and vice versa. This is because, in order to reduce the bias term, one has to increase the complexity (more free parameters) of the adopted estimator $f(\cdot; \mathcal{D})$. This, in turn, results in higher variance as we change the training sets. This is a manifestation of the overfitting issue that we have already discussed. The only way to reduce both terms simultaneously is to increase the number of the training data points, N, and at the same time increase the complexity of the model *carefully*, so as to achieve the aforementioned goal. If one increases the number of training points and at the same time increases the model complexity excessively, the overall MSE may increase. This is known as the *bias-variance dilemma* or *tradeoff*. This is an issue that is omnipresent in any estimation task. Usually, we refer to it as *Occam's razor* rule.

Occam was a logician and a nominalist scholastic medieval philosopher who expressed this law of parsimony: "Plurality must never be posited without necessity." The great physicist Paul Dirac expressed the same statement from an aesthetics point of view, which underlies mathematical theories: "A theory with a mathematical beauty is more likely to be correct than an ugly one that fits the data." In our context of model selection, it is understood that one has to select the simplest model that can "explain" the data. Although this is not a scientifically proven result, it underlies the rationale behind a number of developed model selection techniques [1, 32, 33, 38] and [35, Chapter 5], which trade off complexity with accuracy.

Next, we present a simplistic, yet pedagogic, example to demonstrate this tradeoff between bias and variance. We are given the training points plotted in Figure 3.7 in the (x, y) plane. The points have been generated according to a regression model of the form

$$\mathrm{y} = g(\mathrm{x}) + \eta. \tag{3.51}$$

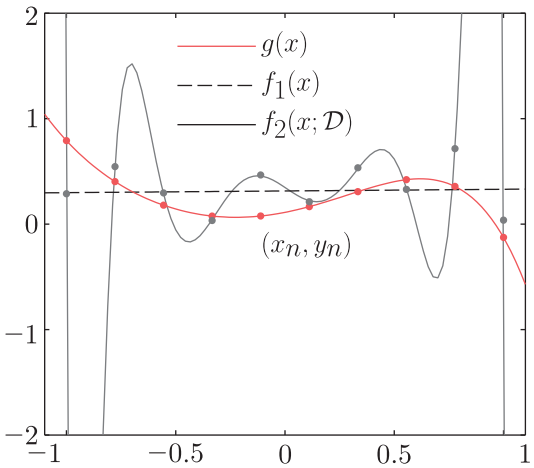

FIGURE 3.7

The observed data are the points denoted as gray dots. These are the result of adding noise to the red points, which lie on the red curve associated with the unknown $g(\cdot)$. Fitting the data by a low degree polynomial, $f_1(x)$, results in high bias; observe that most of the data points lie outside the straight line. On the other hand, the variance of the estimator will be low. In contrast, fitting a high-degree polynomial, $f_2(x; \mathcal{D})$, results in low bias, because the corresponding curve goes through all the data points; however, the respective variance will be high.

The graph of $g(x)$ is shown in Figure 3.7. First, we are going to be very naive and very cautious in spending computational resources, so we have chosen a fixed linear model to fit the data,

$$\hat{y} = f_1(x) = \theta_0 + \theta_1 x,$$

where the values θ_1 and θ_0 have been chosen arbitrarily, irrespective of the training data. The graph of this straight line is shown in Figure 3.7. Because no training was involved and the model parameters are fixed, there is no variation as we change the training sets and $\mathbb{E}_{\mathcal{D}}[f_1(x)] = f_1(x)$, with the variance term being equal to zero. On the other hand, the square of the bias, which is equal to $(f_1(x) - \mathbb{E}[\mathrm{y}|x])^2$, is expected to be large because the choice of the model was arbitrary, without paying attention to the training data. In the sequel, we go to the other extreme. We choose a complex class of functions, such as a very high-order polynomial, $f_2(\cdot; \mathcal{D})$. Then, the corresponding graph of the model is expected always to go through the training points. One such curve is illustrated in Figure 3.7. Generate different data sets $\mathcal{D}_i$ as

$$\mathcal{D}_i = \left\{(g(x_n) + \eta_{i,n}, x_n) : n = 1, 2, \ldots, N\right\}, \quad i = 1, 2, \ldots,$$

where $\eta_{i,n}$ denotes different noise samples, drawn from a white noise process. In other words, all training points have the same x-coordinate and the change in the training sets is due to the different values of the noise. For such an experimental setup, the bias term at each point, x_n, $n = 1, 2, \ldots, N$, is zero, because

$$\mathbb{E}_{\mathcal{D}}[f_2(x_n; \mathcal{D})] = \mathbb{E}_{\mathcal{D}}[g(x_n) + \eta] = g(x_n) = \mathbb{E}_{\mathcal{D}}[\mathrm{y}|x_n].$$

On the other hand, the variance term at the points x_n, $n = 1, 2, \ldots, N$, is expected to be large, because

$$\mathbb{E}_{\mathcal{D}}\left[(f_2(x_n; \mathcal{D}) - g(x_n))^2\right] = \mathbb{E}_{\mathcal{D}}\left[(g(x_n) + \eta - g(x_n))^2\right] = \sigma_\eta^2.$$

Assuming that the functions $f_2(\cdot)$ and $g(\cdot)$ are continuous and smooth enough and the points x_n are dense enough to cover the interval of interest in the real axis, we expect similar behavior at all the points $x \neq x_n$.

A more realistic example is illustrated in Figure 3.8. Consider the model in Eq. (3.51), where $g(\cdot)$ is a fifth-order polynomial. We select a number of points across the respective curve and add noise to them; these comprise the training data set. We run two sets of experiments. The first one attempts to fit in the noisy data a high-order polynomial of degree equal to ten and the second one a low second-order polynomial. For each one of the two setups, we repeat the experiment 1000 times, each time adding a different noise realization to the originally selected points. Figures 3.8a and c show ten (for visibility reasons, out of the 1000) of the resulting curves for the high- and low-order polynomials, respectively. The substantially higher variance for the case of the high-order polynomial is readily noticed. Figures 3.8b and d show the corresponding curves, which result from averaging over the 1000 performed experiments, together with the graph of our "unknown" function. The high-order polynomial results in an excellent fit of very low bias. The opposite is true for the case of the second-order polynomial. The reader may find more information on the bias-variance dilemma problem in [16].

Finally, note that the left-hand side of Eq. (3.50) is the mean, with respect to $\mathcal{D}$, of the second term in Eq. (3.47). It is easy to see that, by reconsidering Eq. (3.47) and taking the expectation on both y and $\mathcal{D}$, given the value of $\mathbf{x} = \boldsymbol{x}$, the resulting MSE becomes (try it, following similar arguments as for Eq. (3.50))

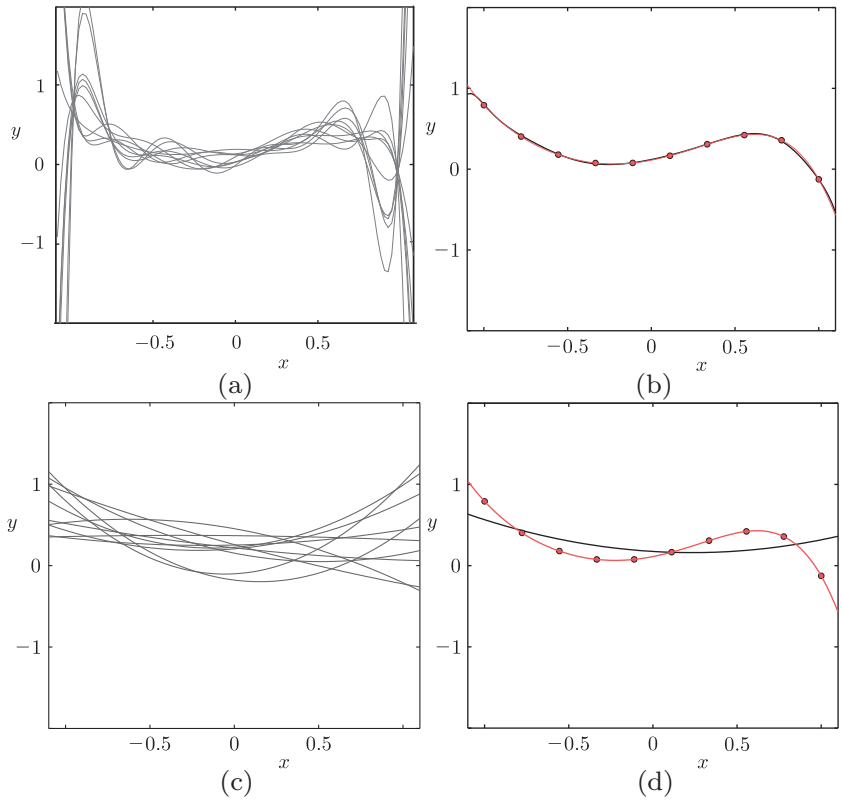

FIGURE 3.8

(a) Ten of the resulting curves from fitting a tenth-order polynomial and (b) the corresponding average over 1000 different experiments, together with the red curve of the unknown polynomial. The dots indicate the points that give birth to the training data, as described in the text. (c) and (d) illustrate the results from fitting a second-order polynomial. Observe the bias-variance tradeoff as a function of the complexity of the fitted model.

$$\begin{aligned}\text{MSE}(\boldsymbol{x}) &= \mathbb{E}_{\text{y}|\boldsymbol{x}}\,\mathbb{E}_{\mathcal{D}}\left[\left(\text{y} - f(\boldsymbol{x};\mathcal{D})\right)^2\right] \\ &= \sigma_\eta^2 + \mathbb{E}_{\mathcal{D}}\left[\left(f(\boldsymbol{x};\mathcal{D}) - \mathbb{E}_{\mathcal{D}}\left[f(\boldsymbol{x};\mathcal{D})\right]\right)^2\right] \\ &\quad + \left(\mathbb{E}_{\mathcal{D}}\left[f(\boldsymbol{x};\mathcal{D})\right] - \mathbb{E}[\text{y}|\boldsymbol{x}]\right)^2, \end{aligned} \tag{3.52}$$

where Eq. (3.49) has been used and the product rule, as stated in Chapter 2, has been exploited. In the sequel, one can take the mean over $\boldsymbol{x}$. The resulting MSE is also known as the *test* or *generalization error* and it is a measure of the performance of the respective adopted model. Note that the generalization error in Eq. (3.52) involves averaging over (theoretically) all possible training data sets of certain size N. In contrast, the so-called *training error* is computed over a single data set, the one used for the training, and this results in an overoptimistic estimate of the error. We will come back to this important issue in Section 3.13.

3.10 MAXIMUM LIKELIHOOD METHOD

So far, we have approached the estimation problem as an optimization task around a set of training examples, without paying any attention to the underlying statistics that generates these points. We only used statistics in order to check under which conditions the estimators were efficient. However, the optimization step did not involve any statistical information. For the rest of the chapter, we are going to involve statistics more and more. In this section, the ML method is introduced. It is not an exaggeration to say that ML and LS are two of the major pillars on which parameter estimation is based and new methods are inspired from. The ML method was suggested by Sir Ronald Aylmer Fisher.

Once more, we will first formulate the method in a general setting, independent of the regression/classification tasks. We are given a set of say, N, observations, $\mathcal{X} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N\}$, drawn from a probability distribution. We assume that the joint pdf of these N observations is of a known parametric functional type, denoted as $p(\mathcal{X}; \boldsymbol{\theta})$, where the parameter vector $\boldsymbol{\theta} \in \mathbb{R}^K$ is unknown and the task is to estimate its value. This is known as the *likelihood function* of $\boldsymbol{\theta}$ with respect to the given set of observations, $\mathcal{X}$. According to the ML method, the estimate is provided by

$$\boxed{\hat{\boldsymbol{\theta}}_{\text{ML}} := \arg\max_{\boldsymbol{\theta} \in \mathcal{A} \subset \mathbb{R}^K} p(\mathcal{X}; \boldsymbol{\theta}) : \quad \text{Maximum Likelihood Estimate.}} \tag{3.53}$$

For simplicity, we will assume that the parameter space $\mathcal{A} = \mathbb{R}^K$, and that the parameterized family $\{p(\mathcal{X}; \boldsymbol{\theta}) : \boldsymbol{\theta} \in \mathbb{R}^K\}$ enjoys a unique minimizer with respect to the parameter $\boldsymbol{\theta}$. This is illustrated in Figure 3.9. In other words, given the set of observations $\mathcal{X} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N\}$, one selects the unknown parameter vector so as to make this joint event the most likely one to happen.

Because the logarithmic function, $\ln(\cdot)$, is monotone and increasing, one can instead search for the maximum of the *log-likelihood function*,

$$\left.\frac{\partial \ln p(\mathcal{X}; \boldsymbol{\theta})}{\partial \boldsymbol{\theta}}\right|_{\boldsymbol{\theta} = \hat{\boldsymbol{\theta}}_{\text{ML}}} = \mathbf{0}. \tag{3.54}$$

Assuming the observations to be i.i.d., the ML estimator has some very attractive properties, namely:

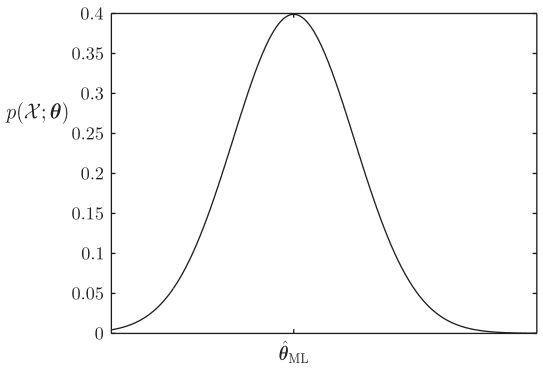

FIGURE 3.9

According to the maximum likelihood method, we assume that, given the set of observations, the estimate of the unknown parameter is the value that maximizes the corresponding likelihood function.

- The ML estimator is asymptotically unbiased; that is, assuming that the model of the pdf, which we have adopted, is correct and there exists a true parameter $\boldsymbol{\theta}_o$, then

$$\lim_{N\to\infty} \mathbb{E}[\hat{\theta}_{\mathrm{ML}}] = \boldsymbol{\theta}_o. \tag{3.55}$$

- The ML estimate is asymptotically *consistent* so that given any value of $\epsilon > 0$,

$$\lim_{N\to\infty} \mathrm{Prob}\left\{\left|\hat{\theta}_{\mathrm{ML}} - \boldsymbol{\theta}_o\right| > \epsilon\right\} = 0, \tag{3.56}$$

  that is, for large values of N, we expect the ML estimate to be very close to the true value with high probability.
- The ML estimator is asymptotically efficient; that is, it achieves the Cramér-Rao lower bound.
- If there exists a sufficient statistic, $T(\mathcal{X})$, for an unknown parameter, then only $T(\mathcal{X})$ suffices to express the respective ML estimate (Problem 3.18).
- Moreover, assuming that an efficient estimator does exist, then this estimator is optimal in the ML sense (Problem 3.19).

Example 3.5. Let $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N$, be the observation vectors stemming from a normal distribution with known covariance matrix and unknown mean; that is,

$$p(\boldsymbol{x}_n; \boldsymbol{\mu}) = \frac{1}{(2\pi)^{l/2}|\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(\boldsymbol{x}_n - \boldsymbol{\mu})^T \Sigma^{-1}(\boldsymbol{x}_n - \boldsymbol{\mu})\right).$$

Assume that the observations are mutually independent. Obtain the ML estimate of the unknown mean vector.

For the N statistically independent observations, the joint log-likelihood function is given by

$$L(\boldsymbol{\mu}) = \ln \prod_{n=1}^{N} p(\boldsymbol{x}_n; \boldsymbol{\mu}) = -\frac{N}{2}\ln\left((2\pi)^l|\Sigma|\right) - \frac{1}{2}\sum_{n=1}^{N}(\boldsymbol{x}_n - \boldsymbol{\mu})^T \Sigma^{-1}(\boldsymbol{x}_n - \boldsymbol{\mu}).$$

Taking the gradient with respect to $\boldsymbol{\mu}$, we obtain[5]

$$\frac{\partial L(\boldsymbol{\mu})}{\partial \boldsymbol{\mu}} := \begin{bmatrix} \dfrac{\partial L}{\partial \mu_1} \\ \dfrac{\partial L}{\partial \mu_2} \\ \vdots \\ \dfrac{\partial L}{\partial \mu_l} \end{bmatrix} = \sum_{n=1}^{N} \Sigma^{-1}(\boldsymbol{x}_n - \boldsymbol{\mu}),$$

and equating to $\mathbf{0}$ leads to

$$\hat{\boldsymbol{\mu}}_{\mathrm{ML}} = \frac{1}{N}\sum_{n=1}^{N} \boldsymbol{x}_n.$$

In other words, for Gaussian distributed data, the ML estimate of the mean is the sample mean. Moreover, note that the ML estimate is expressed in terms of its sufficient statistic, (see Section 3.7).

[5] Recall from matrix algebra that $\frac{\partial(\boldsymbol{x}^T\boldsymbol{b})}{\partial \boldsymbol{x}} = \boldsymbol{b}$ and $\frac{\partial(\boldsymbol{x}^T A\boldsymbol{x})}{\partial \boldsymbol{x}} = 2A\boldsymbol{x}$, if A is symmetric.

3.10.1 LINEAR REGRESSION: THE NONWHITE GAUSSIAN NOISE CASE

Consider the linear regression model

$$\mathrm{y} = \boldsymbol{\theta}^T \mathbf{x} + \eta.$$

We are given N training data points $(y_n, \boldsymbol{x}_n)$, $n = 1, 2, \ldots, N$. The corresponding (unobserved) noise samples, η_n, $n = 1, \ldots, N$, are assumed to follow a jointly Gaussian distribution with zero mean and covariance matrix equal to Σ_η. Our goal is to obtain the ML estimate of the parameters $\boldsymbol{\theta}$.

The joint log-likelihood function of $\boldsymbol{\theta}$, with respect to the training set, is given by

$$L(\boldsymbol{\theta}) = -\frac{N}{2}\ln(2\pi) - \frac{1}{2}\ln|\Sigma_\eta| - \frac{1}{2}(\boldsymbol{y} - X\boldsymbol{\theta})^T \Sigma_\eta^{-1}(\boldsymbol{y} - X\boldsymbol{\theta}), \tag{3.57}$$

where $\boldsymbol{y} := [y_1, y_2, \ldots, y_N]^T$, and $X := [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N]^T$ stands for the input matrix. Taking the gradient with respect to $\boldsymbol{\theta}$, we get

$$\frac{\partial L(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} = X^T \Sigma_\eta^{-1}(\boldsymbol{y} - X\boldsymbol{\theta}), \tag{3.58}$$

and equating to the zero vector, we obtain

$$\hat{\boldsymbol{\theta}}_{\mathrm{ML}} = \left(X^T \Sigma_\eta^{-1} X\right)^{-1} X^T \Sigma_\eta^{-1} \boldsymbol{y}. \tag{3.59}$$

Remarks 3.3.

- Compare Eq. (3.59) with the LS solution given in Eq. (3.17). They are different, unless the covariance matrix of the successive noise samples, Σ_η, is diagonal and of the form $\sigma_\eta^2 I$; that is, if the noise is Gaussian as well as white. In this case, the LS and the ML solutions coincide. However, if the noise sequence is nonwhite, the two estimates differ. Moreover, it can be shown (Problem 3.8) that, *in this case of colored Gaussian noise, the ML estimate is an efficient one and it attains the Cramér-Rao bound, even if N is finite*.

3.11 BAYESIAN INFERENCE

In our discussion, so far, we have assumed that the parameter associated with the functional form of the adopted model is a deterministic constant, whose value is unknown to us. In this section, we will follow a different rationale. The unknown parameter will be treated as a random variable. Hence, whenever our goal is to estimate its value, this is conceived as an effort to estimate the value of a *specific* realization that corresponds to the observed data. A more detailed discussion concerning the Bayesian inference rationale is provided in Chapter 12. As the name Bayesian suggests, the heart of the method beats around the celebrated Bayes theorem. Given two jointly distributed random vectors, say, $\mathbf{x}$, θ, Bayes theorem states that

$$p(\boldsymbol{x}, \boldsymbol{\theta}) = p(\boldsymbol{x}|\boldsymbol{\theta})p(\boldsymbol{\theta}) = p(\boldsymbol{\theta}|\boldsymbol{x})p(\boldsymbol{x}). \tag{3.60}$$

David Bayes (1702-1761) was an English mathematician and a Presbyterian minister who first developed the basics of the theory. However, it was Pierre-Simon Laplace (1749-1827), the great French mathematician, who further developed and popularized it.

Assume that $\mathbf{x}, \Theta$ are two statistically dependent random vectors. Let $\mathcal{X} = \{\boldsymbol{x}_n \in \mathbb{R}^l, n = 1, 2, \ldots, N\}$, be the set of the observations resulting from N successive experiments. Then, Bayes theorem gives

$$p(\boldsymbol{\theta}|\mathcal{X}) = \frac{p(\mathcal{X}|\boldsymbol{\theta})p(\boldsymbol{\theta})}{p(\mathcal{X})} = \frac{p(\mathcal{X}|\boldsymbol{\theta})p(\boldsymbol{\theta})}{\int p(\mathcal{X}|\boldsymbol{\theta})p(\boldsymbol{\theta})\,\mathrm{d}\boldsymbol{\theta}}. \tag{3.61}$$

Obviously, if the observations are i.i.d., then we can write

$$p(\mathcal{X}|\boldsymbol{\theta}) = \prod_{n=1}^{N} p(\boldsymbol{x}_n|\boldsymbol{\theta}).$$

In the previous formulas, $p(\boldsymbol{\theta})$ is the a priori pdf concerning the statistical distribution of Θ, and $p(\boldsymbol{\theta}|\mathcal{X})$ is the conditional or a posteriori pdf, formed after the set of N observations has been obtained. The prior probability density, $p(\boldsymbol{\theta})$, can be considered as a constraint that *encapsulates our prior knowledge* about Θ. No doubt, our uncertainty about Θ is modified after the observations have been received, because more information is now disclosed to us. If the adopted assumptions about the underlying models are sensible, we expect the posterior pdf to be a more accurate one to describe the statistical nature of Θ. We will refer to the process of approximating the pdf of a random quantity, based on a set of training data, as *inference*, to differentiate it from the process of estimation, that returns a single value for each parameter/variable. So, according to the inference approach, one attempts to draw conclusions about the nature of the randomness that underlies the variables of interest. This information can in turn be used to make predictions and to take decisions.

We will exploit Eq. (3.61) in two ways. The first refers to our familiar goal of obtaining an estimate of the parameter vector Θ, which "controls" the model that describes the generation mechanism of our observations, $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N$. Because $\mathbf{x}$ and Θ are two statistically dependent random vectors, we know from Section 3.9 that the MSE optimal estimate of the value of Θ, given $\mathcal{X}$, is

$$\hat{\boldsymbol{\theta}} = \mathbb{E}[\Theta|\mathcal{X}] = \int \boldsymbol{\theta} p(\boldsymbol{\theta}|\mathcal{X})\,\mathrm{d}\boldsymbol{\theta}. \tag{3.62}$$

Another direction along which one can exploit the Bayes theorem, in the context of statistical inference, is to obtain an estimate of the pdf of $\mathbf{x}$ given the observations $\mathcal{X}$. This can be done by *marginalizing* over a distribution, using the equation

$$p(\boldsymbol{x}|\mathcal{X}) = \int p(\boldsymbol{x}|\boldsymbol{\theta})p(\boldsymbol{\theta}|\mathcal{X})\,\mathrm{d}\boldsymbol{\theta}, \tag{3.63}$$

where the conditional independence of $\mathbf{x}$ on $\mathcal{X}$, given the value $\Theta = \boldsymbol{\theta}$, expressed as $p(\boldsymbol{x}|\mathcal{X}, \boldsymbol{\theta}) = p(\boldsymbol{x}|\boldsymbol{\theta})$, has been used. Equation (3.63) provides an estimate of the unknown pdf, by exploiting the information that resides in the obtained observations as well as in the adopted functional dependence on the parameters Θ. Note that, in contrast to what we did in the case of the ML method, where we used the observations to obtain an estimate of the parameter vector, here we assume the parameters to be random variables, we provide our prior knowledge about Θ via $p(\boldsymbol{\theta})$ and integrate the joint pdf, $p(\boldsymbol{x}, \boldsymbol{\theta}|\mathcal{X})$, over $\boldsymbol{\theta}$.

Once $p(\boldsymbol{x}|\mathcal{X})$ is available, it can be used for prediction. Assuming that we have obtained the observations $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N$, our estimate about the next value, $\boldsymbol{x}_{N+1}$, to occur can be determined via $p(\boldsymbol{x}_{N+1}|\mathcal{X})$. Obviously, the form of $p(\boldsymbol{x}|\mathcal{X})$ is, in general, changing as new observations are obtained,

because each time an observation becomes available, part of our uncertainty about the underlying randomness is removed.

Example 3.6. Consider the simplified linear regression task of Eq. (3.31) and assume $x = 1$. As we have already said, this problem is that of estimating the value of a constant buried in noise. Our methodology will follow the Bayesian philosophy. Assume that the noise samples are i.i.d. drawn from a Gaussian pdf of zero mean and variance σ_η^2. However, we impose our a priori knowledge concerning the unknown θ, via the prior distribution

$$p(\theta) = \mathcal{N}(\theta_0, \sigma_0^2). \tag{3.64}$$

That is, we assume that we know that the values of θ lie around θ_0, and σ_0^2 quantifies our degree of uncertainty about this prior knowledge. Our goals are first to obtain the a posteriori pdf, given the set of measurements $\boldsymbol{y} = [y_1, \ldots, y_N]^T$, and then to obtain $\mathbb{E}[\theta|\boldsymbol{y}]$, according to Eqs. (3.61) and (3.62) and adapting them to our current notational needs. We have that

$$\begin{aligned} p(\theta|\boldsymbol{y}) &= \frac{p(\boldsymbol{y}|\theta)p(\theta)}{p(\boldsymbol{y})} = \frac{1}{p(\boldsymbol{y})}\left(\prod_{n=1}^{N} p(y_n|\theta)\right)p(\theta) \\ &= \frac{1}{p(\boldsymbol{y})}\left(\prod_{n=1}^{N} \frac{1}{\sqrt{2\pi}\sigma_\eta}\exp\left(-\frac{(y_n-\theta)^2}{2\sigma_\eta^2}\right)\right) \\ &\quad \times \frac{1}{\sqrt{2\pi}\sigma_0}\exp\left(-\frac{(\theta-\theta_0)^2}{2\sigma_0^2}\right). \end{aligned} \tag{3.65}$$

After some algebraic manipulations on Eq. (3.65) (Problem 3.23), one ends up in the following:

$$p(\theta|\boldsymbol{y}) = \frac{1}{\sqrt{2\pi}\sigma_N}\exp\left(-\frac{(\theta-\bar{\theta}_N)^2}{2\sigma_N^2}\right), \tag{3.66}$$

where

$$\bar{\theta}_N = \frac{N\sigma_0^2\bar{y}_N + \sigma_\eta^2\theta_0}{N\sigma_0^2 + \sigma_\eta^2}, \tag{3.67}$$

with $\bar{y}_N = \frac{1}{N}\sum_{n=1}^{N} y_n$ being the sample mean of the observations and

$$\sigma_N^2 = \frac{\sigma_\eta^2\sigma_0^2}{N\sigma_0^2 + \sigma_\eta^2}. \tag{3.68}$$

In other words, if the prior and the conditional pdfs are Gaussians, then the posterior is also Gaussian. Moreover, the mean and the variance of the posterior are given by Eqs. (3.67) and (3.68), respectively.

Observe that as the number of observations increases, $\bar{\theta}_N$ tends to the sample mean of the observations; recall that the latter is the estimate that results from the ML method. Also, note that the variance keeps decreasing as the number of observations increases, which is in line with common sense, because more observations mean less uncertainty. Figure 3.10 illustrates the previous discussion. Data samples, y_n, were generated using a Gaussian pseudorandom generator with mean equal to $\theta = 1$ and variance equal to $\sigma_\eta^2 = 0.1$. So the true value of our constant is equal to 1. We used a Gaussian

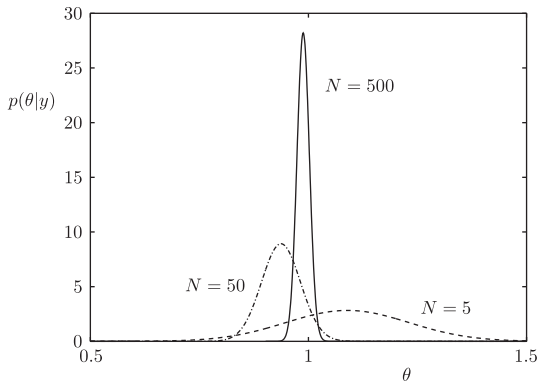

FIGURE 3.10

In the Bayesian inference approach, note that as the number of observations increases, our uncertainty about the true value of the unknown parameter is reduced and the mean of the posterior pdf tends to the true value and the variance tends to zero.

prior pdf with mean value equal to $\theta_0 = 2$ and variance $\sigma_0^2 = 6$. We observe that as N increases, the posterior pdf gets narrower and its mean tends to the true value of 1.

It should be pointed out that in the case of this example, both the ML and LS estimates become identical, or

$$\hat{\theta} = \frac{1}{N}\sum_{n=1}^{N} y_n = \bar{y}_N.$$

This will also be the case for the mean value in Eq. (3.67), if we set σ_0^2 very large, as might happen if we have no confidence in our initial estimate of θ_0 and we assign a very large value to σ_0^2. In effect, this is equivalent to using no prior information.

Let us now investigate what happens if our prior knowledge about θ_0 is "embedded" in the LS criterion in the form of a constraint. This can be done by modifying the constraint in Eq. (3.38), such that

$$(\theta - \theta_0)^2 \leq \rho, \tag{3.69}$$

which leads to the minimization of the following Lagrangian

$$\text{minimize} \quad L(\theta, \lambda) = \sum_{n=1}^{N}(y_n - \theta)^2 + \lambda\left((\theta - \theta_0)^2 - \rho\right). \tag{3.70}$$

Taking the derivative with respect to θ and equating to zero, we obtain

$$\hat{\theta} = \frac{N\bar{y}_N + \lambda\theta_0}{N + \lambda},$$

which, for $\lambda = \sigma_\eta^2/\sigma_0^2$, becomes identical to Eq. (3.67). The world is small after all! This has happened only because we used Gaussians both for the conditional as well as the prior pdfs. For different forms

of pdfs, this would not be the case. However, this example shows that a close relationship ties priors and constraints. They both attempt to impose prior information. Each method, in its own unique way, is associated with the respective pros and cons. In Chapters 12 and 13, where a more extended treatment of the Bayesian inference task is provided, we will see that the very essence of regularization, which is a means against overfitting, lies at the heart of the Bayesian approach.

Finally, one may wonder if the Bayesian inference has offered us any more information, compared to the deterministic parameter estimation path. After all, when the aim is to obtain a specific value for the unknown parameter, then taking the mean of the Gaussian posterior comes to the same solution, which results from the regularized LS approach. Well, even for this simple case, the Bayesian inference readily provides a piece of extra information; this is an estimate of the variance around the mean, which is very valuable in order to assess our trust of the recovered estimate. Of course, all these are valid provided that the adopted pdfs offer a good description of the statistical nature of the process at hand [24].

Finally, it can be shown, Problem 3.24, that the previously obtained results are generalized for the more general linear regression model, of nonwhite Gaussian noise, which was considered in Section 3.10, as shown by

$$\boldsymbol{y} = X\boldsymbol{\theta} + \boldsymbol{\eta}.$$

It turns out that the posterior pdf is also Gaussian with mean value equal to

$$\mathbb{E}[\theta|\boldsymbol{y}] = \boldsymbol{\theta}_0 + \left(\Sigma_0^{-1} + X^T\Sigma_\eta^{-1}X\right)^{-1} X^T\Sigma_\eta^{-1}\left(\boldsymbol{y} - X\boldsymbol{\theta}_0\right), \tag{3.71}$$

and

$$\Sigma_{\theta|y} = \left(\Sigma_0^{-1} + X^T\Sigma_\eta^{-1}X\right)^{-1}. \tag{3.72}$$

3.11.1 THE MAXIMUM A POSTERIORI PROBABILITY ESTIMATION METHOD

The Maximum A Posteriori Probability Estimation technique, usually denoted as MAP, is based on the Bayesian theorem, but it does not go as far as the Bayesian philosophy allows. The goal becomes that of obtaining an estimate which maximizes Eq. (3.61); in other words,

$$\boxed{\hat{\boldsymbol{\theta}}_{\text{MAP}} = \arg\max_{\boldsymbol{\theta}} p(\boldsymbol{\theta}|\mathcal{X}): \quad \text{MAP Estimate},} \tag{3.73}$$

and because $p(\mathcal{X})$ is independent of $\boldsymbol{\theta}$, this leads to

$$\begin{aligned}\hat{\boldsymbol{\theta}}_{\text{MAP}} &= \arg\max_{\boldsymbol{\theta}} p(\mathcal{X}|\boldsymbol{\theta})p(\boldsymbol{\theta}) \\ &= \arg\max_{\boldsymbol{\theta}} \left\{\ln p(\mathcal{X}|\boldsymbol{\theta}) + \ln p(\boldsymbol{\theta})\right\}.\end{aligned} \tag{3.74}$$

If we consider Example 3.6, it is a matter of simple exercise to obtain the MAP estimate and show that

$$\hat{\theta}_{\text{MAP}} = \frac{N\bar{y}_N + \dfrac{\sigma_\eta^2}{\sigma_0^2}\theta_0}{N + \dfrac{\sigma_\eta^2}{\sigma_0^2}} = \bar{\theta}_N. \tag{3.75}$$

Note that for this case, the MAP estimate coincides with the regularized LS solution, for $\lambda = \sigma_\eta^2/\sigma_0^2$. Once more, we verify that adopting a prior pdf for the unknown parameter acts as a regularizer, which embeds into the problem the available prior information.

Remarks 3.4.

- Observe that for the case of the Example 3.6, all three estimators, namely ML, MAP and the Bayesian (taking the mean), result *asymptotically*, as N increases, in the same estimate. This is a more general result and it is true for other pdfs as well as for the case of parameter vectors. As the number of observations increases, our uncertainty is reduced and $p(\mathcal{X}|\boldsymbol{\theta})$, $p(\boldsymbol{\theta}|\mathcal{X})$ peak sharply around a value of θ. This forces all the methods to result in similar estimates. However, the obtained estimates are different for finite values of N. Recently, as we will see in Chapters 12 and 13, Bayesian methods have become very popular, and seem to be the choice, among the three methods, for a number of practical problems.
- The choice of the prior pdf in the Bayesian methods is not an innocent task. In Example 3.6, we chose the conditional pdf (likelihood function) as well as the prior pdf to be Gaussians. We saw that the posterior pdf was also Gaussian. The advantage of such a choice was that we could come to closed form solutions. This is not always the case, and then the computation of the posterior pdf needs sampling methods or other approximate techniques. We will come to that in Chapters 12 and 14. However, the family of Gaussians is not the only one with this nice property of leading to closed form solutions. In probability theory, if the posterior is of the same form as the prior, we say that $p(\boldsymbol{\theta})$ is a *conjugate prior* of the likelihood function $p(\mathcal{X}|\boldsymbol{\theta})$ and then the involved integrations can be carried out in closed form, see, e.g., [15, 30] and Chapter 12. Hence, the Gaussian pdf is a conjugate of itself.
- Just for the sake of pedagogical purposes, it is interesting to recapitulate some of the nice properties that the Gaussian pdf possesses. We have met these properties in various sections and problems in the book, so far: (a) it is a conjugate of itself; (b) if two random variables (vectors) are jointly Gaussian, then their marginal pdfs are also Gaussian and the posterior pdf of one w.r.t. the other is also Gaussian; (c) moreover, the linear combination of jointly Gaussian variables turns out to be Gaussian; (d) as a by-product, it turns out that the sum of statistically independent Gaussian random variables is also a Gaussian one; and finally (e) the central limit theorem states that the sum of a large number of independent random variables tends to be Gaussian, as the number of the summands increases.

3.12 CURSE OF DIMENSIONALITY

In a number of places in this chapter, we mentioned the need of having a large number of training points. In Section 3.9.2, while discussing the bias-variance tradeoff, it was stated that in order to end up with a low overall MSE, the complexity (number of parameters) of the model should be small enough with respect to the number of training points. In Section 3.8, overfitting was discussed and it was pointed out that, if the number of training points is small with respect to the number of parameters, overfitting occurs.

The question that is now raised is how big a data set should be, in order to be more relaxed concerning the performance of the designed predictor. The answer to the previous question depends largely on the

dimensionality of the input space. It turns out that, the larger the dimension of the input space the more data points are needed. This is related to the so-called *curse of dimensionality*, a term coined for the first time in [3].

Let us assume that we are given the same number of points, N, thrown randomly in a unit cube (hypercube) in two different spaces, one being of low and the other of very high dimension. Then, the average distance of the points in the latter case will be much larger than that in the low-dimensional space case. As a matter of fact, the average distance shows a dependence that is analogous to the exponential term ($N^{-1/l}$), where l is the dimensionality of the space [14, 35]. For example, the average distance between two out of 10^{10} points in the 2-dimensional space is 10^{-5} and in the 40-dimensional space is equal to 1.83. Figure 3.11 shows two cases, each one consisting of 100 points. The red points lie on a (one-dimensional) line segment of length equal to one and were generated according to the uniform distribution. Gray points cover a (two-dimensional) square region of unit area, which were also generated by a two-dimensional uniform distribution. Observe that, the square area is more sparsely populated compared to the line segment. This is the general trend and high-dimensional spaces are sparsely populated; thus, many more data points are needed in order to fill in the space with enough data. Fitting a model in a parameter space, one must have enough data covering sufficiently well all regions in the space, in order to be able to learn well enough the input-output functional dependence, (Problem 3.26).

There are various ways to cope with the curse of dimensionality and try to exploit the available data set in the best possible way. A popular direction is to resort to suboptimal solutions by projecting the input/feature vectors in a lower dimensional subspace or manifold. Very often, such an approach leads to small performance losses, because the original training data, although they are generated in a high-dimensional space, in fact they may "live" in a lower-dimensional subspace or manifold, due to physical

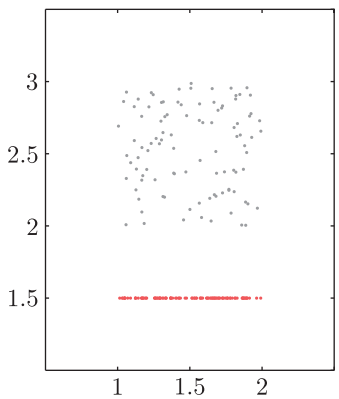

FIGURE 3.11

A simple experiment, which demonstrates the curse of dimensionality. A number of 100 points are generated randomly, drawn from a uniform distribution, in order to fill the 1-d segment of length equal to one ($[1, 2] \times \{1.5\}$) (red points), and the two-dimensional rectangular region of unit area, $[1, 2] \times [2, 3]$ (gray points). Observe that, although the number of points in both cases is the same, the rectangular region is sparsely populated compared to the densely populated line segment.

dependencies that restrict the number of free parameters. Take as an example a case where the data are 3-dimensional vectors, but they lie around a straight line, which is a one-dimensional linear manifold (affine set or subspace if it crosses the origin) or around a circle (one-dimensional nonlinear manifold) embedded in the 3-dimensional space. That is, the true number of free parameters, in this case, is equal to one; this is because one free parameter suffices to describe the location of a point on a circle or on a straight line. The true number of free parameters is also known as the *intrinsic dimensionality* of the problem. The challenge, now, becomes that of learning the subspace/manifold onto which to project. These issues will be considered in more detail in Chapter 19.

Finally, it has to be noted that the dimensionality of the input space is not always the crucial issue. In pattern recognition, it has been shown that the critical factor is the so-called *VC-dimension* of a classifier. In a number of classifiers, such as (generalized) linear classifiers or neural networks (to be considered in Chapter 18), the VC-dimension is directly related to the dimensionality of the input space. However, one can design classifiers, such as the support vector machines (Chapter 11), whose performance is not directly related to the input space and they can be efficiently designed in spaces of very high (of even infinite) dimensionality [35, 38].

3.13 VALIDATION

From previous sections, we already know that what is a "good" estimate according to one set of training points, it is not necessarily a good one for other data sets. This is an important aspect in any machine learning task; the performance of a method may vary with the random choice of the training set. A major phase, in any machine learning task, is to quantify/predict the performance that the designed (prediction) model is expected to exhibit in practice. It will not come as a surprise to state that "measuring" the performance against the training data set would lead to an "optimistic" value of the performance index, because this is computed on the same set on which the estimate was optimized; this trend has been known since the early 1930s [22]. For example, if the model is complex enough, with a large number of free parameters, the training error may even become zero, since a perfect fit to the data can be achieved. What is more meaningful and fair is to look for the so-called *generalization* performance of an estimator; that is, its average performance computed over *different* data sets, which *did not* participate in the training (see, Section 3.9.2).

Figure 3.12 shows a typical performance that is expected to result in practice. The error measured on the (single) training data set is shown together with the (average) test/generalization error as the model complexity varies. If one tries to fit a complex model, with respect to the size of the available training set, then the error measured on the training set will be overoptimistic. On the contrary, the true error, as this is represented by the test error, takes large values; in the case where the performance index is the MSE, this is mainly contributed by the variance term (Section 3.9.2). On the other hand, if the model is too simple it leads the test error also to large values; for the MSE case, this time the contribution is mainly due to the bias term. The idea is to have a model complexity that corresponds to the minimum of the respective curve. As a matter of fact, this is the point that various model selection techniques try to predict.

For some simple cases and under certain assumptions concerning the underlying models, we are able to have analytical formulas that quantify the average performance as we change data sets. However, in practice, this is hardly the case, and one must have a way to test the performance of an obtained

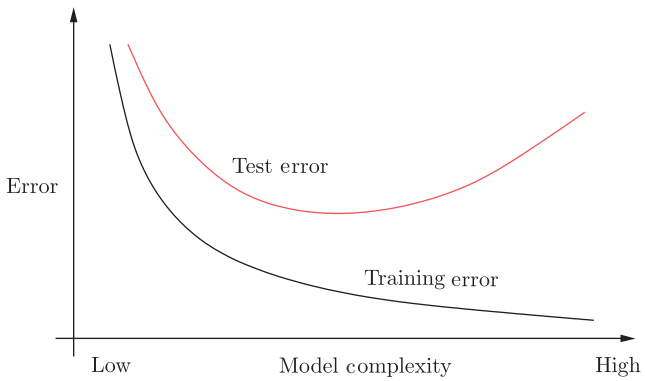

FIGURE 3.12

The training error tends to zero as the model complexity increases; for complex enough models with a large number of free parameters, a perfect fit to the training data is possible. However, the test error initially decreases, because more complex models "learn" the data better, to a point. After that point of complexity, the test error increases.

classifier/predictor using different data sets. The process is known as *validation* and there are a number of alternatives that one can resort to.

Assuming that enough data are at the designer's disposal, one can split the data into one part, to be used for training, and another part for testing the performance. For example, the probability of error is computed over the test data set for the case of a classifier, or the MSE for the case of a regression task; other measures of fit can also been used. If this path is taken, one has to make sure that both the size of the training set as well as the size of the test set are large enough, with respect to the model complexity; a large test data set is required in order to provide a statistically sound result on the test error. Especially if different methods are compared, the smaller the difference in their comparative performance is expected to be, the larger the size of the test set must be made, in order to guarantee reliable conclusions [35, Chapter 10].

Cross-validation

In practice, very often the size of the available data is not sufficient and one cannot afford to "lose" part of it from the training set for the sake of testing. *Cross-validation* is a very common technique that is usually employed. Cross-validation has been rediscovered a number of times; however, to our knowledge, the first published description can be traced back to [25]. According to this method, the data set is split into, say K, roughly equal-sized parts. We repeat training K times, each time selecting one (different each time) part of the data for testing and the remaining $K-1$ parts for training. This gives us the advantage of testing with a part of the data that has not been involved in the training, hence it can be considered as being independent, and at the same time using, eventually, all the data both for training and testing. Once we finish, we can (a) combine the obtained K estimates by averaging or via another more advanced way and (b) combine the test errors to get a better estimate of the generalization error that our estimator is expected to exhibit in real-life applications. The method is known as K-fold cross-validation. An extreme case is when we use $K = N$, so that each time one sample is left for testing.

This is sometimes referred to as the *leave-one-out* (LOO) cross-validation method. The price one pays for K-fold cross-validation is the complexity of training K times. In practice, the value of K depends very much on the application, but typical values are of the order of 5 to 10.

The cross-validation estimator of the generalization error is very nearly unbiased. The reason for the slight bias is that the training set in cross-validation is slightly smaller than the actual data set. The effect of this bias will be conservative in the sense that the estimated fit will be slightly biased in the direction suggesting a poorer fit. In practice, this bias is rarely a concern, especially in the LOO case, where each time only one sample is left out. The variance, however, of the cross-validation estimator can be large, and this has to be taken into account when comparing different methods. In [12], the use of *bootstrap* techniques is suggested in order to reduce the variance of the obtained error predictions by the cross-validation method.

Moreover, besides complexity and high variance, cross-validation schemes are not beyond criticisms. Unfortunately, the overlap among the training sets introduces unknowable dependencies between runs, making the use of formal statistical tests difficult [10]. All this discussion reveals that the validation task is far from innocent. Ideally, one should have at her/his disposal large data sets and divide them in several *nonoverlapping* training sets, of whatever size is appropriate, along with separate test sets (or a single one) that are (is) large enough. More on different validation schemes and their properties can be found in, e.g., [2, 11, 17, 35] and an insightful related discussion in [26].

3.14 EXPECTED AND EMPIRICAL LOSS FUNCTIONS

What was said before in our discussion concerning the generalization and the training set-based performance of an estimator, can be given a more formal statement via the notion of *expected loss*. Adopting a loss function, $\mathcal{L}(\cdot,\cdot)$, in order to quantify the deviation between the predicted value, $\hat{y} = f(\boldsymbol{x})$, and the respective true one, y, the corresponding expected loss is defined as

$$J(f) := \mathbb{E}\left[\mathcal{L}\left(\mathrm{y}, f(\mathbf{x})\right)\right], \tag{3.76}$$

or more explicitly

$$\boxed{J(f) = \int \ldots \int \mathcal{L}\big(y, f(\boldsymbol{x})\big) p(y, \boldsymbol{x}) dy d\boldsymbol{x} : \quad \text{Expected Loss Function,}} \tag{3.77}$$

where the integration is replaced by summation whenever the respective variables are discrete. As a matter of fact, this is the ideal cost function one would like to optimize with respect to $f(\cdot)$, in order to get the optimal estimator over *all* possible values of the input-output pairs. However, such an optimization would in general be a very hard task, even if one knew the functional form of the joint distribution. Thus, in practice, one has to be content with two approximations. First, the functions to be searched are constrained within a certain family, $\mathcal{F}$, (in this chapter, we focused on parametrically described families of functions). Second, because the joint distribution is either unknown and/or the integration may not be analytically tractable, the expected loss is approximated by the so-called *empirical loss* version, defined as

$$\boxed{J_N(f) = \frac{1}{N} \sum_{n=1}^{N} \mathcal{L}\big(y_n, f(\boldsymbol{x}_n)\big) : \quad \text{Empirical Loss Function.}} \tag{3.78}$$

As an example, the MSE function, discussed earlier, is the expected loss associated with the squared error loss function and the LS cost is the respective empirical version. For large enough values of N and provided that the family of functions is restricted enough,[6] we expect that the outcome from optimizing J_N to be close to that which would be obtained by optimizing J [38].

From the validation point of view, given any prediction function, $f(\cdot)$, what we called generalization error corresponds to the corresponding value of J in Eq. (3.77) and the training error to that of J_N in Eq. (3.78).

We can now take the discussion a little further, which will reveal some more secrets concerning the accuracy-complexity tradeoff in machine learning. Let f_* be the function that optimizes the expected loss,

$$f_* := \arg\min_f J(f), \tag{3.79}$$

and $f_{\mathcal{F}}$ the optimal after *constraining* the task within the family of functions $\mathcal{F}$,

$$f_{\mathcal{F}} := \arg\min_{f\in\mathcal{F}} J(f). \tag{3.80}$$

Let us also define

$$f_N := \arg\min_{f\in\mathcal{F}} J_N(f). \tag{3.81}$$

Then, we can readily write that

$$\mathbb{E}\left[J(f_N) - J(f_*)\right] = \underbrace{\mathbb{E}\left[J(f_{\mathcal{F}}) - J(f_*)\right]}_{\text{approximation error}} + \underbrace{\mathbb{E}\left[J(f_N) - J(f_{\mathcal{F}})\right]}_{\text{estimation error}}. \tag{3.82}$$

The *approximation error* measures the deviation in the generalization error, if instead of the overall optimal function one uses the optimal obtained within a certain family of functions. The *estimation error* measures the deviation due to optimizing the empirical instead of the expected loss. If one chooses the family of functions to be very large, then it is expected that the approximation error will be small, because there is high probability f_* will be close to one of the members of the family. However, the estimation error is expected to be large, because for a fixed number of data points, N, fitting a complex function is likely to lead to overfitting. For example, if the family of functions is the class of polynomials of a very large order, a very large number of parameters are to be estimated and overfitting will occur. The opposite is true if the class of functions is a small one. In parametric modeling, complexity of a family of functions is related to the number of free parameters. However, this is not the whole story. As a matter of fact, complexity is really measured by the so-called *capacity* of the associated set of functions. The VC-dimension mentioned in Section 3.12 is directly related to the capacity of the family of the considered classifiers. More concerning the theoretical treatment of these issues can be obtained from [9, 38, 39].

[6] That is, the family of functions is not very large. To keep the discussion simple, take the example of quadratic class of functions. This is larger than that of the linear ones, because the latter is a special case (subset) of the former.

3.15 NONPARAMETRIC MODELING AND ESTIMATION

The focus of this chapter is on the task of parameter estimation and on techniques that spring from the idea of parametric functional modeling of an input-output dependence. To put the final touches on this chapter, we shift our attention to the alternative philosophy that runs across the field of statistical estimation; that of *nonparametric modeling*. In contrast to parametric modeling, either no parameters are involved or if parameters pop in, their number is not fixed but grows with the number of training samples. We will treat such models in the context of reproducing kernel Hilbert spaces (RKHS) in Chapter 11. There, instead of parameterizing the family of functions, in which one constrains the search for finding the prediction model, the candidate solution is constrained to lie within a specific functional space.

In this section, the nonparametric modeling rationale is demonstrated in the framework of approximating an unknown pdf. Although such techniques are very old, they can still be used and they are also the focus of more recent research efforts [13].

Our kick off point is the classical *histogram* approximation of an unknown pdf. Let us assume that we are given a set of points, $x_n \in \mathbb{R}$, $n = 1, 2, \ldots, N$, which have been independently drawn from an unknown distribution. Figure 3.13a illustrates a pdf approximation using the histogram technique. The

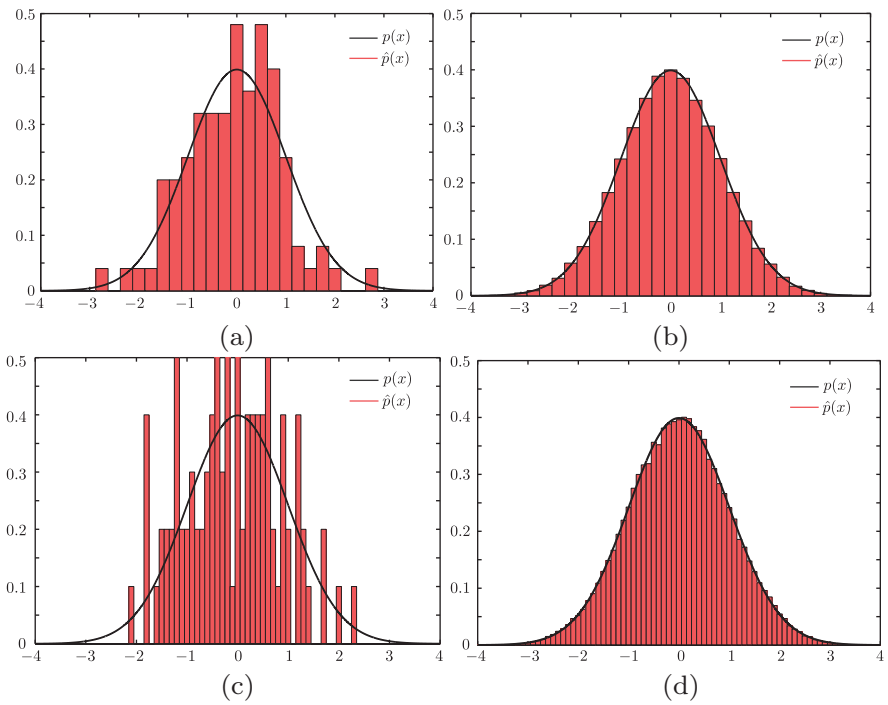

FIGURE 3.13

The gray curve corresponds to the true pdf. The red curves correspond to the histogram approximation method, for various values of the pair (h, N). (a) $h = 0.25$ and $N = 100$, (b) $h = 0.25$ and $N = 10^5$, (c) $h = 0.1$ and $N = 100$, (d) $h = 0.1$ and $N = 10^5$. The larger the data size and the smaller the size of the bin the better the approximation becomes.

real axis is divided into a number of successive interval bins, each of length h, which is a user-defined constant. Let us focus on one of these interval bins and denote its middle point as $\hat{x}$. Count the number of observations that lie inside this bin, say, k_N. Then, the pdf approximation for all the points within this specific interval is given by

$$\hat{p}(x) = \frac{1}{h}\frac{k_N}{N}, \quad \text{if } |x - \hat{x}| \leq \frac{h}{2}. \tag{3.83}$$

This results to a "staircase" approximation of the continuous pdf function. It turns out that this very simple rule converges to $p(x)$, provided that $h \to 0$, $k_N \to \infty$ and $k_N/N \to 0$. That is, when the length of the bins become small, the number of observations in each bin is large enough to guarantee that the *frequency ratio* k_N/N is a good estimate of the probability of a point lying in the bin, and the number of observations tends to infinity faster than the number of points in the bins. Figure 3.13a corresponds to a (relatively) large value of h and a (relatively) small number of points. Figure 3.13d corresponds to a (relatively) small value of h and a relatively large number of points. Observe that, in the latter case, the approximation of the pdf is smoother and closer to the true curve. The training points were generated by a Gaussian of variance equal to one.

To apply the histogram approximation method, in practice, we first select h. Then, given a point x, we count the number of the observations that lie within the interval $[x - h/2, x + h/2]$, and we make use of the ratio in Eq. (3.83) to obtain the estimate $\hat{p}(x)$. To dress up what we have just described with a mathematical formalism, define

$$\phi(x) := \begin{cases} 1, & \text{if } |x| \leq 1/2, \\ 0, & \text{otherwise.} \end{cases} \tag{3.84}$$

Then, it is easily checked out that the histogram estimate at x is given by

$$\hat{p}(x) = \frac{1}{h}\frac{1}{N}\sum_{n=1}^{N}\phi\left(\frac{x - x_n}{h}\right). \tag{3.85}$$

Indeed, the summation is equal to the number of observations that lie within the interval $[x - h/2, x + h/2]$. An alternative way to view Eq. (3.85) is as an expansion over a set of functions, each one centered at an observation point. However, although such an expansion converges to the true value, it is a bit unorthodox, because it attempts to approximate a continuous function in terms of discontinuous ones. As a matter of fact, this is a reason that the convergence of histogram methods is slow, in the sense that many points, N, are required for a reasonably good approximation. This can also be verified from Figure 3.13, where in spite of the fact that 10^5 points have been used, the approximation is still not very good. Note that what we have said so far can be generalized to any Euclidean space $\mathbb{R}^l$.

Parzen [28] in order to bypass the drawback that we have just stated, proved that the approximation is still possible if one replaces the discontinuous function $\phi(\cdot)$, by a smooth one. Such functions are known as *kernels, potential functions*, or *Parzen windows*, and must satisfy the following conditions:

$$\phi(\boldsymbol{x}) \geq 0 \text{ and} \tag{3.86}$$

$$\int \phi(\boldsymbol{x})\, d\boldsymbol{x} = 1, \tag{3.87}$$

where the more general case of a Euclidean space, $\mathbb{R}^l$, has been chosen as the data space. The Gaussian pdf is, obviously, such a function. For such a choice, one can write the Parzen approximation of an unknown pdf as

$$\hat{p}(\boldsymbol{x}) = \frac{1}{N}\sum_{n=1}^{N}\frac{1}{(2\pi)^{l/2}h^l}\exp\left(-\frac{(\boldsymbol{x}-\boldsymbol{x}_n)^T(\boldsymbol{x}-\boldsymbol{x}_n)}{2h^2}\right). \tag{3.88}$$

In words, according to the Parzen approximation, a kernel function (the Gaussian in this case) is centered at each one of the observations and we take their sum. Such types of expansions will be a popular theme in this book. An interesting issue is to search for ways to reduce the number of points that contribute into the summation, by selecting the most important ones. That is, we will try to make such expansions more *sparse*. As we will see, besides issues related to the computational load, reducing the number of terms is in line with our effort to be more robust against overfitting; Occam's razor rule once again.

The way we have approached the task of pdf function approximation, so far in this section, was to select a bin of *fixed size*, h, centered at the point of interest x. In higher-dimensional spaces, the interval bin becomes a square of length h, in the two-dimensional case, a cube in three dimensions, and a hypercube in higher dimensions. The other alternative is to fix the number of points, k, and try to increase the volume of the hypercube around $\boldsymbol{x}$, so that k points are included. Because the approximation depends on the ratio $\frac{1}{N}\frac{k}{V}$, this is also a good idea. In dense (high values of pdf) areas, k points will be clustered within regions of small volume, and in less dense (low values of pdf) areas they will fill in regions of larger volume. Moreover, one can now consider alternatives to hypercube shapes, such as hyperspheres, hyperellipsoids, and so on. The algorithmic procedure is simple. Search for the k-nearest neighbors of $\boldsymbol{x}$, among the available observations, and compute the volume in the space within which they are located. Then, estimate the value of the pdf at $\boldsymbol{x}$, using the previously stated ratio. This is known as the *k-nearest neighbor density estimation*. More on the topic can be found in, e.g., [35, 36].

The previous technique of the k-nearest neighbors can be further relaxed and be emancipated from the idea of estimating pdfs. Then, it gives birth to one of the most widely known and used methods for classification, known as the k-nearest neighbor classification rule, which is discussed in Chapter 7.

PROBLEMS

3.1 Let $\hat{\theta}_i$, $i = 1, 2, \ldots, m$, be unbiased estimators of a parameter vector $\boldsymbol{\theta}$, so that $\mathbb{E}[\hat{\theta}_i] = \boldsymbol{\theta}$, $i = 1, \ldots, m$. Moreover, assume that the respective estimators are uncorrelated to each other and that all have the same (total) variance, $\sigma^2 = \mathbb{E}[(\theta_i - \boldsymbol{\theta})^T(\theta_i - \boldsymbol{\theta})]$. Show that by averaging the estimates, e.g.,

$$\hat{\theta} = \frac{1}{m}\sum_{i=1}^{m}\hat{\theta}_i,$$

the new estimator has total variance $\sigma_c^2 := \mathbb{E}[(\hat{\theta} - \boldsymbol{\theta})^T(\hat{\theta} - \boldsymbol{\theta})] = \frac{1}{m}\sigma^2$.

3.2 Let a random variable x being described by a uniform pdf in the interval $[0, \frac{1}{\theta}]$, $\theta > 0$. Assume a function[7] g, which defines an estimator $\hat{\theta} := g(\mathrm{x})$ of θ. Then, for such an estimator to be unbiased, the following must hold:

$$\int_0^{\frac{1}{\theta}} g(x)\,\mathrm{d}x = 1.$$

However, such a function g does not exist.

3.3 A family $\{p(\mathcal{D};\boldsymbol{\theta}) : \boldsymbol{\theta} \in \mathcal{A}\}$ is called *complete* if, for any vector function $\boldsymbol{h}(\mathcal{D})$ such that $\mathbb{E}_{\mathcal{D}}[\boldsymbol{h}(\mathcal{D})] = \mathbf{0}$, $\forall \boldsymbol{\theta}$, then $\boldsymbol{h} = \mathbf{0}$.

Show that if $\{p(\mathcal{D};\boldsymbol{\theta}) : \boldsymbol{\theta} \in \mathcal{A}\}$ is complete, and there exists an MVU estimator, then this estimator is unique.

3.4 Let $\hat{\theta}_u$ be an unbiased estimator, so that $\mathbb{E}[\hat{\theta}_u] = \theta_o$. Define a biased one by $\hat{\theta}_b = (1+\alpha)\hat{\theta}_u$. Show that the range of α where the MSE of $\hat{\theta}_b$ is smaller than that of $\hat{\theta}_u$ is

$$-2 < -\frac{2\mathrm{MSE}(\hat{\theta}_u)}{\mathrm{MSE}(\hat{\theta}_u) + \theta_o^2} < \alpha < 0.$$

3.5 Show that for the setting of the Problem 3.4, the optimal value of α is equal to

$$\alpha_* = -\frac{1}{1 + \frac{\theta_o^2}{\mathrm{var}(\hat{\theta}_u)}},$$

where, of course, the variance of the unbiased estimator is equal to the corresponding MSE.

3.6 Show that the regularity condition for the Cramér-Rao bound holds true if the order of integration and differentiation can be interchanged.

3.7 Derive the Cramér-Rao bound for the LS estimator, when the training data result from the linear model

$$y_n = \theta x_n + \eta_n, \quad n = 1, 2, \ldots,$$

where x_n and η_n are i.i.d. samples of a zero mean random variable, with variance σ_x^2, and a Gaussian one with zero mean and variance σ_η^2, respectively. Assume, also, that x and η are independent. Then, show that the LS estimator achieves the CR bound only asymptotically.

3.8 Let us consider the regression model

$$y_n = \boldsymbol{\theta}^T \boldsymbol{x}_n + \eta_n, \quad n = 1, 2, \ldots, N,$$

where the noise samples $\boldsymbol{\eta} = [\eta_1, \ldots, \eta_N]^T$ come from a zero mean Gaussian random vector, with covariance matrix Σ_η. If $X = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N]^T$ stands for the input matrix, and $\boldsymbol{y} = [y_1, \ldots, y_N]^T$, then show that,

$$\hat{\boldsymbol{\theta}} = \left(X^T \Sigma_\eta^{-1} X\right)^{-1} X^T \Sigma_\eta^{-1} \boldsymbol{y},$$

is an efficient estimate.

[7] To avoid any confusion, let g be Lebesgue integrable on intervals of $\mathbb{R}$.

Notice, here, that the previous estimate coincides with the ML one. Moreover, bear in mind that in the case where $\Sigma_\eta = \sigma^2 I$ then the ML estimate becomes equal to the LS one.

3.9 Assume a set of i.i.d. $\mathcal{X} = \{x_1, x_2, \ldots, x_N\}$ samples of a random variable, with mean μ and variance σ^2. Define also the quantities

$$S_\mu := \frac{1}{N}\sum_{n=1}^{N} x_n, \quad S_{\sigma^2} := \frac{1}{N}\sum_{n=1}^{N}(x_n - S_\mu)^2,$$
$$\bar{S}_{\sigma^2} := \frac{1}{N}\sum_{n=1}^{N}(x_n - \mu)^2.$$

Show that if μ is considered to be known, a sufficient statistic for σ^2 is $\bar{S}_{\sigma^2}$. Moreover, in the case where both (μ, σ^2) are unknown, then a sufficient statistic is the pair (S_μ, S_{σ^2}).

3.10 Show that solving the task

$$\text{minimize} \quad L(\boldsymbol{\theta}, \lambda) = \sum_{n=1}^{N}\left(y_n - \theta_0 - \sum_{i=1}^{l}\theta_i x_{ni}\right)^2 + \lambda\sum_{i=1}^{l}|\theta_i|^2,$$

is equivalent with minimizing

$$\text{minimize} \quad L(\boldsymbol{\theta}, \lambda) = \sum_{n=1}^{N}\left((y_n - \bar{y}) - \sum_{i=1}^{l}\theta_i(x_{ni} - \bar{x}_i)\right)^2 + \lambda\sum_{i=1}^{l}|\theta_i|^2,$$

and the estimate of θ_0 is given by

$$\hat{\theta}_0 = \bar{y} - \sum_{i=1}^{l}\hat{\theta}_i\bar{x}_i.$$

3.11 This problem refers to Example 3.4, where a linear regression task with a real valued unknown parameter θ_o is considered. Show that $\text{MSE}(\hat{\theta}_b(\lambda)) < \text{MSE}(\hat{\theta}_{\text{MVU}})$ or the ridge regression estimate shows a lower MSE performance than the one for the MVU estimate, if

$$\begin{cases} \lambda \in (0, \infty), & \theta_o^2 \leq \frac{\sigma_\eta^2}{N}, \\ \lambda \in \left(0, \dfrac{2\sigma_\eta^2}{\theta_o^2 - \dfrac{\sigma_\eta^2}{N}}\right), & \theta_o^2 > \dfrac{\sigma_\eta^2}{N}. \end{cases}$$

Moreover, the minimum MSE performance for the ridge regression estimate is attained at $\lambda_* = \sigma_\eta^2/\theta_o^2$.

3.12 Assume that the model that generates the data is

$$y_n = A\sin\left(\frac{2\pi}{N}kn + \phi\right) + \eta_n,$$

where $A > 0$, and $k \in \{1, 2, \ldots, N-1\}$. Assume that η_n are i.i.d. samples from a Gaussian noise, of variance σ_η^2. Show that there is no unbiased estimator for the phase, ϕ, based on N measurement points, $y_n, n = 0, 1, \ldots N-1$, that attains the Cramér-Rao bound.

3.13 Show that if $(\mathbf{y}, \mathbf{x})$ are two jointly distributed random vectors, with values in $\mathbb{R}^k \times \mathbb{R}^l$, then the MSE optimal estimator of $\mathbf{y}$ given the value $\mathbf{x} = \boldsymbol{x}$ is the regression of $\mathbf{y}$ conditioned on $\boldsymbol{x}$, or $\mathbb{E}[\mathbf{y}|\boldsymbol{x}]$.

3.14 Assume that $\mathbf{x}$, $\mathbf{y}$ are jointly Gaussian random vectors, with covariance matrix

$$\Sigma := \mathbb{E}\left[\begin{bmatrix}\mathbf{x}-\boldsymbol{\mu}_x\\ \mathbf{y}-\boldsymbol{\mu}_y\end{bmatrix}\left[(\mathbf{x}-\boldsymbol{\mu}_x)^T, (\mathbf{y}-\boldsymbol{\mu}_y)^T\right]\right] = \begin{bmatrix}\Sigma_x & \Sigma_{xy}\\ \Sigma_{yx} & \Sigma_y\end{bmatrix}.$$

Assuming also that the matrices Σ_x and $\bar{\Sigma} := \Sigma_y - \Sigma_{yx}\Sigma_x^{-1}\Sigma_{xy}$ are nonsingular, then show that the optimal MSE estimator $\mathbb{E}[\mathbf{y}|\boldsymbol{x}]$ takes the following form

$$\mathbb{E}[\mathbf{y}|\boldsymbol{x}] = \mathbb{E}[\mathbf{y}] + \Sigma_{yx}\Sigma_x^{-1}(\boldsymbol{x} - \boldsymbol{\mu}_x).$$

Notice that $\mathbb{E}[\mathbf{y}|\boldsymbol{x}]$ is an affine function of $\boldsymbol{x}$. In other words, for the case where $\mathbf{x}$ and $\mathbf{y}$ are jointly Gaussian, the optimal estimator of $\mathbf{y}$, in the MSE sense, which is in general a nonlinear function, becomes an affine function of $\boldsymbol{x}$.

In the special case where x, y are scalar random variables, then

$$\mathbb{E}[\mathrm{y}|x] = \mu_y + \frac{\alpha\sigma_y}{\sigma_x}(x - \mu_x),$$

where α stands for the *correlation coefficient*, defined as

$$\alpha := \frac{\mathbb{E}\left[(\mathrm{x}-\mu_x)(\mathrm{y}-\mu_y)\right]}{\sigma_x\sigma_y},$$

with $|\alpha| \leq 1$. Notice, also, that the previous assumption on the nonsingularity of Σ_x and $\bar{\Sigma}$ translates, in this special case, to $\sigma_x \neq 0 \neq \sigma_y$, and $|\alpha| < 1$.

Hint: Use the matrix inversion lemma from Appendix A, in terms of the Schur complement $\bar{\Sigma}$ of Σ_x in Σ and the fact that $\det(\Sigma) = \det(\Sigma_y)\det(\bar{\Sigma})$.

3.15 Assume a number l of jointly Gaussian random variables $\{\mathrm{x}_1, \mathrm{x}_2, \ldots, \mathrm{x}_l\}$, and a nonsingular matrix $A \in \mathbb{R}^{l\times l}$. If $\mathbf{x} := [\mathrm{x}_1, \mathrm{x}_2, \ldots, \mathrm{x}_l]^T$, then show that the components of the vector $\mathbf{y}$, obtained by $\mathbf{y} = A\mathbf{x}$, are also jointly Gaussian random variables.

A direct consequence of this result is that any linear combination of jointly Gaussian variables is also Gaussian.

3.16 Let $\mathbf{x}$ be a vector of jointly Gaussian random variables of covariance matrix Σ_x. Consider the general linear regression model

$$\mathbf{y} = \Theta\mathbf{x} + \boldsymbol{\eta},$$

where $\Theta \in \mathbb{R}^{k\times l}$ is a parameter matrix and $\boldsymbol{\eta}$ is the noise vector which is considered to be Gaussian, with zero mean, and with covariance matrix Σ_η, independent of $\mathbf{x}$. Then show that $\mathbf{y}$ and $\mathbf{x}$ are jointly Gaussian, with covariance matrix given by

$$\Sigma = \begin{bmatrix}\Theta\Sigma_x\Theta^T + \Sigma_\eta & \Theta\Sigma_x\\ \Sigma_x\Theta^T & \Sigma_x\end{bmatrix}.$$

3.17 Show that a linear combination of Gaussian independent variables is also Gaussian.

3.18 Show that if a sufficient statistic $T(\mathcal{X})$ for a parameter estimation problem exists, then $T(\mathcal{X})$ suffices to express the respective ML estimate.

3.19 Show that if an efficient estimator exists, then it is also optimal in the ML sense.

3.20 Let the observations resulting from an experiment be x_n, $n = 1, 2, \ldots, N$. Assume that they are independent and that they originate from a Gaussian pdf $\mathcal{N}(\mu, \sigma^2)$. Both, the mean and the variance, are unknown. Prove that the ML estimates of these quantities are given by

$$\hat{\mu}_{\text{ML}} = \frac{1}{N}\sum_{n=1}^{N} x_n, \quad \hat{\sigma}^2_{\text{ML}} = \frac{1}{N}\sum_{n=1}^{N}(x_n - \hat{\mu}_{\text{ML}})^2.$$

3.21 Let the observations $x_n, n = 1, 2, \ldots, N$, come from the uniform distribution

$$p(x;\theta) = \begin{cases} \dfrac{1}{\theta}, & 0 \le x \le \theta, \\ 0, & \text{otherwise.} \end{cases}$$

Obtain the ML estimate of θ.

3.22 Obtain the ML estimate of the parameter $\lambda > 0$ of the exponential distribution

$$p(x) = \begin{cases} \lambda \exp(-\lambda x), & x \ge 0, \\ 0, & x < 0, \end{cases}$$

based on a set of measurements, x_n, $n = 1, 2, \ldots, N$.

3.23 Assume an $\mu \sim \mathcal{N}(\mu_0, \sigma_0^2)$, and a stochastic process $\{x_n\}_{n=-\infty}^{\infty}$, consisting of i.i.d. random variables, such that $p(x_n|\mu) = \mathcal{N}(\mu, \sigma^2)$. Consider a number of N members of the process $\{x_n\}_{n=-\infty}^{\infty}$, so that $\mathcal{X} := \{x_1, x_2, \ldots, x_N\}$, and prove that the posterior $p(x|\mathcal{X})$, of any $x = x_{n_0}$ conditioned on $\mathcal{X}$, turns out to be Gaussian with mean μ_N and variance $\sigma^2 + \sigma_N^2$, where

$$\mu_N := \frac{N\sigma_0^2\bar{x} + \sigma^2\mu_0}{N\sigma_0^2 + \sigma^2}, \quad \sigma_N^2 := \frac{\sigma^2\sigma_0^2}{N\sigma_0^2 + \sigma^2}.$$

3.24 Show that for the linear regression model,

$$\boldsymbol{y} = X\boldsymbol{\theta} + \boldsymbol{\eta},$$

the a posteriori probability $p(\boldsymbol{\theta}|\boldsymbol{y})$ is a Gaussian one, if the prior distribution probability is given by $p(\boldsymbol{\theta}) = N(\boldsymbol{\theta}_0, \Sigma_0)$, and the noise samples follow the multivariate Gaussian distribution $p(\boldsymbol{\eta}) = N(\mathbf{0}, \Sigma_\eta)$. Compute the mean vector and the covariance matrix of the posterior distribution.

3.25 Assume that x_n, $n = 1, 2, \ldots, N$, are i.i.d. observations from a Gaussian $\mathcal{N}(\mu, \sigma^2)$. Obtain the MAP estimate of μ, if the prior follows the exponential distribution

$$p(\mu) = \lambda \exp(-\lambda\mu), \quad \lambda > 0, \mu \ge 0.$$

3.26 Consider, once more, the same regression model as that of Problem 3.8, but with $\Sigma_\eta = I_N$. Compute the MSE of the predictions $\mathbb{E}[(\mathrm{y} - \hat{\mathrm{y}})^2]$, where y is the true response and $\hat{\mathrm{y}}$ is the predicted value, given a test point $\boldsymbol{x}$ and using the LS estimator,

$$\hat{\theta} = \left(X^T X\right)^{-1} X^T \mathbf{y}.$$

The LS estimator has been obtained via a set of N measurements, collected in the (fixed) input matrix X and $\mathbf{y}$, where the notation has been introduced previously in this chapter. The expectation $\mathbb{E}[\cdot]$ is taken with respect to y, the training data, $\mathcal{D}$ and the test points $\mathbf{x}$. Observe the dependence of the MSE on the dimensionality of the space.

Hint. Consider, first, the MSE, given the value of a test point $\boldsymbol{x}$, and then take the average over all the test points.

REFERENCES

[1] H. Akaike, A new look at the statistical model identification, IEEE Trans. Autom. Control 19 (6) (1970) 716-723.

[2] S. Arlot, A. Celisse, A survey of cross-validation procedures for model selection, Stat. Surv. 4 (2010) 40-79.

[3] R.E. Bellman, Dynamic Programming, Princeton University Press, Princeton, 1957.

[4] A. Ben-Israel, T.N.E. Greville, Generalized Inverses: Theory and Applications, second ed., Springer-Verlag, New York, 2003.

[5] D. Bertsekas, A. Nedic, O. Ozdaglar, Convex Analysis and Optimization, Athena Scientific, Belmont, MA, 2003.

[6] S. Boyd, L. Vandenberghe, Convex Optimization, Cambridge University Press, Cambridge, 2004.

[7] O. Chapelle, B. Scholkopf, A. Zien, Semisupervised Learning, MIT Press, Cambridge, 2006.

[8] H. Cramer, Mathematical Methods of Statistics, Princeton University Press, Princeton, 1946.

[9] L. Devroy, L. Györfi, G. Lugosi, A Probabilistic Theory of Pattern Recognition, Springer, New York, 1991.

[10] T.G. Dietterich, Approximate statistical tests for comparing supervised classification learning algorithms, Neural Comput. 10 (1998) 1895-1923.

[11] R. Duda, P. Hart, D. Stork, Pattern Classification, second ed., Wiley, New York, 2000.

[12] A. Efron, R. Tibshirani, Improvements on cross-validation: the .632+ bootstrap method, J. Am. Stat. Assoc. 92 (438) (1997) 548-560.

[13] D. Erdogmus, J.C. Principe, From linear adaptive filtering to nonlinear information processing, IEEE Signal Process. Mag. 23 (6) (2006) 14-33.

[14] J.H. Friedman, Regularized discriminant analysis, J. Am. Stat. Assoc. 84 (1989) 165-175.

[15] A. Gelman, J.B. Carlin, H.S. Stern, D.B. Rubin, Bayesian Data Analysis, second ed., CRC Press, Boca Raton, FL, 2003.

[16] S. Geman, E. Bienenstock, R. Doursat, Neural networks and the bias-variance dilemma, Neural Comput. 4 (1992) 1-58.

[17] T. Hastie, R. Tibshirani, J. Friedman, The Elements of Statistical Learning, second ed., Springer, New York, 2009.

[18] A.E. Hoerl, R.W. Kennard, Ridge regression: biased estimation for nonorthogonal problems, Technometrics 12 (1) (1970) 55-67.

[19] S. Kay, Y. Eldar, Rethinking biased estimation, IEEE Signal Process. Mag. 25 (6) (2008) 133-136.

[20] S. Kay, Statistical Signal Processing, Prentice Hall, Upper Saddle River, NJ, 1993.

[21] M. Kendall, A. Stuart, The Advanced Theory of Statistics, vol. 2, MacMillan, New York, 1979.

[22] S.C. Larson, The shrinkage of the coefficient of multiple correlation, J. Educ. Psychol. 22 (1931) 45-55.

[23] E.L. Lehmann, H. Scheffe, Completeness, similar regions, and unbiased estimation: Part II, Sankhyā 15 (3) (1955) 219-236.

[24] D. McKay, Probable networks and plausible predictions—a review of practical Bayesian methods for supervised neural networks, Netw. Comput. Neural Syst. 6 (1995) 169-505.

[25] F. Mosteller, J.W. Tukey, Handbook of Social Psychology, Chap. Data Analysis, Including Statistics, Addison-Wesley, Reading, MA, 1954.
[26] R.M. Neal, Assessing relevance determination methods using DELVE, in: C.M. Bishop (Ed.), Neural Networks and Machine Learning, Springer-Verlag, New York, 1998, pp. 97-129.
[27] A. Papoulis, P. Unnikrishna, Probability, Random Variables, and Stochastic Processes, fourth ed., McGraw Hill, New York, NY, 2002.
[28] E. Parzen, On the estimation of a probability density function and mode, Ann. Math. Stat., 33 (1962) 1065-1076.
[29] D.L. Phillips, A technique for the numerical solution of certain integral equations of the first kind, J. Assoc. Comput. Mach. 9 (1962) 84-97.
[30] H. Raiffa, R. Schlaifer, Applied Statistical Decision Theory, Division of Research, Graduate School of Business Administration, Harvard University, Boston, 1961.
[31] R.C. Rao, Information and the accuracy attainable in the estimation of statistical parameters, Bull. Calcutta Math. Soc. 37 (1945) 81-89.
[32] J. Rissanen, A universal prior for integers and estimation by minimum description length, Ann. Stat. 11 (2) (1983) 416-431.
[33] G. Schwartz, Estimating the dimension of the model, Ann. Stat. 6 (1978) 461-464.
[34] J. Shao, Mathematical Statistics, Springer, New York, 1998.
[35] S. Theodoridis, K. Koutroumbas, Pattern Recognition, fourth ed., Academic Press, New York, 2009.
[36] S. Theodoridis, A. Pikrakis, K. Koutroumbas, D. Cavouras, An Introduction to Pattern Recognition: A MATLAB Approach, Academic Press, New York, 2010.
[37] A.N. Tychonoff, V.Y. Arsenin, Solution of Ill-posed Problems, Winston & Sons, Washington, 1977.
[38] V.N. Vapnik, The Nature of Statistical Learning Theory, Springer-Verlag, New York, 1995.
[39] V.N. Vapnik, Statistical Learning Theory, John Wiley & Sons, New York, 1998.