CHAPTER

INTRODUCTION 1

CHAPTER OUTLINE

1.1 WHAT MACHINE LEARNING IS ABOUT

Learning through personal experience and knowledge, which propagates from generation to generation, is at the heart of human intelligence. Also, at the heart of any scientific field lies the development of models (often, they are called theories) in order to explain the available experimental evidence at each time period. In other words, we always *learn from data*. Different data and different focuses on the data give rise to different scientific disciplines.

This book is about learning from data; in particular, our intent is to detect and unveil a possible hidden structure and regularity patterns associated with their generation mechanism. This information in turn helps our analysis and understanding of the nature of the data, which can be used to make *predictions* for the future. Besides modeling the underlying structure, a major direction of significant interest in Machine Learning is to develop *efficient* algorithms for designing the models and also for analysis and prediction. The latter part is gaining importance in the dawn of what we call the *big data* era, when one has to deal with massive amounts of data, which may be represented in spaces of very large dimensionality. Analyzing data for such applications sets demands on algorithms to be computationally efficient and at the same time *robust* in their performance, because some of these data are contaminated with large noise and also, in some cases, the data may have missing values.

Such methods and techniques have been at the center of scientific research for a number of decades in various disciplines, such as Statistics and Statistical Learning, Pattern Recognition, Signal and Image Processing and Analysis, Computer Science, Data Mining, Machine Vision, Bioinformatics, Industrial Automation, and Computer-Aided Medical Diagnosis, to name a few. In spite of the different names, there is a common corpus of techniques that are used in all of them, and we will refer to such methods as Machine Learning. This name has gained popularity over the last decade or so. The name suggests the use of a machine/computer to learn in analogy to how the brain learns and predicts. In some cases, the methods are directly inspired by the way the brain works, as is the case with neural networks, covered in Chapter 18.

Machine Learning. http://dx.doi.org/10.1016/B978-0-12-801522-3.00001-X

Two problems at the heart of machine learning, which also comprise the backbone of this book, are the classification and the regression tasks.

1.1.1 CLASSIFICATION

The goal in classification is to assign an unknown *pattern* to one out of a number of classes that are considered to be known. For example, in X-ray mammography, we are given an image where a region indicates the existence of a tumor. The goal of a computer-aided diagnosis system is to predict whether this tumor corresponds to the *benign* or the *malignant* class. Optical character recognition (OCR) systems are also built around a classification system, in which the image corresponding to each letter of the alphabet has to be recognized and assigned to one of the twenty-four (for the Latin alphabet) classes; see Section 18.11, for a related case study. Another example is the prediction of the authorship of a given text. Given a text written by an unknown author, the goal of a classification system is to predict the author among a number of authors (classes); this application is treated in Section 11.15.

The first step in designing any machine learning task is to decide how to represent each pattern in the computer. This is achieved during the preprocessing stage; one has to "encode" related information that resides in the raw data (image pixels or strings of letters in the previous examples) in an efficient and information-rich way. This is usually done by transforming the raw data in a new space with each pattern represented by a vector, $\boldsymbol{x} \in \mathbb{R}^l$. This is known as the *feature vector*, and its l elements are known as the *features*. In this way, each pattern becomes a single point in an l-dimensional space, known as the *feature space* or the *input space*. We refer to this as the *feature generation* stage. Usually, one starts with some large value K of features and eventually selects the l most informative ones via an optimizing procedure known as the *feature selection* stage.

Having decided upon the input space, in which the data are represented, one has to train a classifier. This is achieved by first selecting a set of data whose class is known, which comprises the *training set*. This is a set of pairs, $(y_n, \boldsymbol{x}_n)$, $n = 1, 2, \ldots, N$, where y_n is the (output) variable denoting the class in which $\boldsymbol{x}_n$ belongs, and it is known as the corresponding *class label*; the class labels, y, take values over a *discrete* set, $\{1, 2, \ldots, M\}$, for an M-class classification task. For example, for a two-class classification task, $y_n \in \{-1, +1\}$. To keep our discussion simple, let us focus on the two-class case. Based on the training data, one then designs a function, f, which predicts the output label given the input; that is, given the measured values of the features. This function is known as the *classifier*. In general, we need to design a set of such functions.

Once the classifier has been designed, the system is ready for predictions. Given an unknown pattern, we form the corresponding feature vector, $\boldsymbol{x}$, from the raw data, and we plug this value into the classifier; depending on the value of $f(\boldsymbol{x})$ (usually on the respective sign, $\hat{y} = \text{sgn} f(\boldsymbol{x})$) the pattern is classified in one of the two classes. Figure 1.1 illustrates the classification task. Initially, we are given the set of points, each representing a pattern in the two-dimensional space (two features used, x_1, x_2). Stars belong to one class, say ω_1 and the crosses to the other, ω_2, in a two-class classification task. These are the training points. Based on these points, a classifier was learned; for our very simple case, this turned out to be a linear function,

$$f(\boldsymbol{x}) = \theta_1 x_1 + \theta_2 x_2 + \theta_0, \tag{1.1}$$

whose graph for all the points such as: $f(\boldsymbol{x}) = 0$, is the straight line shown in the figure. Then, we are given the point denoted by the red circle; this corresponds to the measured values from a pattern whose

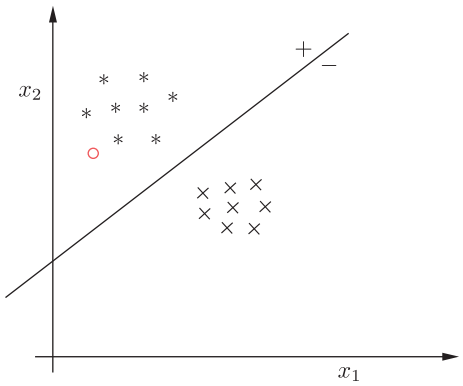

FIGURE 1.1

The classifier (linear in this simple case) has been designed in order to separate the training data into the two classes, having on its positive side the points coming from one class and on its negative side those of the other. The "red" point, whose class is unknown, is classified to the same class as the "star" points, since it lies on the positive side of the classifier.

class is unknown to us. According to the classification system, which we have designed, this belongs to the same class as the points denoted by stars. Indeed, every point on one side of the straight line will give a positive value, $f(\boldsymbol{x}) > 0$, and all the points on its other side will give a negative value, $f(\boldsymbol{x}) < 0$. The point denoted with the red circle will then result in $f(\boldsymbol{x}) > 0$, as all the star points, and it is classified in the same class, ω_1.

This type of learning is known as *supervised learning*, since a set of training data with known labels is available. Note that the training data can be seen as the available previous experience, and based on this, one builds a model to make predictions for the future. *Unsupervised/clustering* and *semisupervised* learning are not treated in this book, with the exception of the k-means algorithm, which is treated in Chapter 12. Clustering and semisupervised learning are treated in detail in the companion books [1, 2].

Note that the receiver in a digital communications system can also be viewed as a classification system. Upon receiving the transmitted data, which have been contaminated by noise and also by other transformations imposed by the transmission channel (Chapter 4), one has to reach a decision on the value of the originally transmitted symbol. However, in digital communications, the transmitted symbols come from a finite alphabet, and each symbol defines a different class, ± 1, for a binary transmitted sequence.

1.1.2 REGRESSION

The regression shares to a large extent the feature generation/selection stage, as described before; however, now the output variable, y, is *not* discrete but it takes values in an interval in the real axis or in a region in the complex numbers plane. The regression task is basically a curve fitting problem.

We are given a set of training points, $(y_n, \boldsymbol{x}_n)$, $y_n \in \mathbb{R}$, $\boldsymbol{x}_n \in \mathbb{R}^l$, $n = 1, 2, \ldots, N$, and the task is to estimate a function, f, whose graph fits the data. Once we have found such a function, when an unknown point arrives, we can predict its output value. This is shown in Figure 1.2.

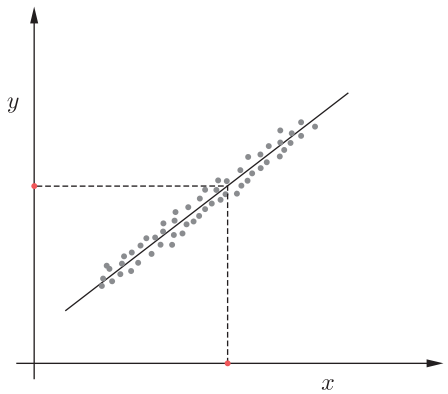

FIGURE 1.2

Once a function (linear in this case), f, has been designed, for its graph to fit the available training data set in a regression task, given a new (red) point, x, the prediction of the associated output (red) value is given by $y = f(x)$.

(a) (b)

FIGURE 1.3

(a) The blurred image, taken by a moving camera, and (b) its de-blurred estimate.

The training data in this case are the gray points. Once the curve fitting task has been completed, given a new point x (red), we are ready to predict its output value as $\hat{y} = f(\boldsymbol{x})$.

The regression task is a generic task that embraces a number of problems. For example, in financial applications one can predict tomorrow's stock market price given current market conditions and all other related information. Each piece of information is a measured value of a corresponding feature. Signal and image restoration come under this common umbrella of tasks. Signal and image de-noising can also be seen as a special type of a regression task. Figure 1.3a shows the case of a blurred image,

taken by a moving camera, and Figure 1.3b the de-blurred one (see Chapter 4). De-blurring is a typical image restoration task, where the de-blurred image is obtained as the output by feeding the blurred one as input to an appropriately designed function.

1.2 STRUCTURE AND A ROAD MAP OF THE BOOK

In the discussion above, we saw that seemingly different applications, e.g., authorship identification and channel equalization as well as financial prediction and image de-blurring can be treated in a unified framework. Many of the techniques that have been developed for machine learning are no different than techniques used in statistical signal processing or adaptive signal processing. Filtering comes under the general framework of regression (Chapter 4), and "adaptive filtering" is exactly the same as "online learning" in machine learning. As a matter of fact, as will be explained in more detail, this book can serve the needs of more than one advanced graduate or postgraduate course.

Over the years, a large number of techniques and "schools" have been developed, in the context of different applications. The two main paths are the Bayesian approach and the deterministic one. The former school considers the unknown parameters that define an unknown function, for example, $\theta_1, \theta_2, \theta_0$ in Eq. (1.1), as random variables, and the latter as having fixed, yet unknown, values. I respect both schools of thought, as I believe that there is more than one road that leads to the "truth." Each can solve some problems more efficiently than the other, and vice versa. Maybe in a few years, the scene will be more clear and more definite conclusions can be drawn. Or it may turn out, as in life, that the "truth" is in the middle. It is interesting to note that one of the most powerful learning techniques of high interest currently is the deep learning approach (covered in Chapter 18), which is an interplay of probabilistic and deterministic arguments.

In any case, every newcomer to the field has to learn the basics and the classics. That's why in this book, all major directions and methods will be discussed, in an equally balanced manner, to the greatest extent possible. Of course, the author, being human, could not avoid emphasizing the techniques with which he is most familiar. This is healthy, since writing a book is a means of sharing the author's expertise and point of view with readers. This is why I strongly believe that a new book does not come to replace previous ones, but to complement previously published points of view.

Chapter 2 is an introduction to probability and statistics. Random processes are also discussed. Readers who are familiar with such concepts can bypass this chapter. On the other hand, one can focus on different parts of this chapter. Readers who would like to focus on statistical signal processing/adaptive processing can focus more on the random processes part. Those who would like to follow a probabilistic machine learning point of view would find the part presenting the various distributions more important. In any case, the Gaussian distribution is a must for those who are not yet familiar with it.

Chapter 3 is an overview of the parameter estimation task. This is a chapter that presents an overview of the book and defines the main concepts that run across its pages. This chapter has also been written to stand alone as an introduction to machine learning. Although it is my feeling that all of it should be read and taught, depending on the focus of the course and taking into account the omnipresent time limitations, one can focus more on the parts of her or his interest. Both the deterministic as well as the probabilistic approaches are defined and discussed. In any case, the parts dealing with the definition of the inverse problems, the bias-variance trade-off, and the concepts of generalization and regularization are a must.

Chapter 4 is dedicated to the Mean-Square Error (MSE) linear estimation. For those following a statistical signal processing (SP) course, all of the chapter is important. The rest of the readers can bypass the parts related to complex-valued processing and also the part dealing with computational complexity issues, since this is only of importance if the input data are random processes. Bypassing this part will not affect reading later parts of the chapter that deal with the MSE of linear models, the Gauss-Markov theorem, and the Kalman filtering.

Chapter 5 introduces the stochastic gradient descent family of algorithms. The first part, dealing with the stochastic approximation method, is a must for every reader. The rest of the chapter, which deals with the Least-Mean-Squares (LMS) algorithm and its offsprings, is more appropriate for readers who are interested in a statistical SP course, since these families are suited for tracking time varying environments. This may not be the first priority for readers who are interested in classification and machine learning tasks with data whose statistical properties are not time varying.

Chapter 6 is dedicated to the Least-Squares (LS) cost function, which is of interest to all readers in machine learning and signal processing. The latter part dealing with the total least-squares method can be bypassed in a first reading. Emphasis is also put on ridge regression and its geometric interpretation. Ridge regression is important to the newcomer, since he/she becomes familiar with the concept of regularization; this is an important aspect in any machine learning task, tied directly with the generalization performance of the designed predictor.

I have decided to compress the part dealing with fast LS algorithms, which are appropriate when the input is a random process/signal that imposes a special structure on the involved covariance matrices, into a discussion section. It is the author's feeling that this is of no greater interest than it was a decade or two ago. Also, the main idea, that of a highly structured covariance matrix that lies behind the fast algorithms, is discussed in some detail in Chapter 4, in the context of Levinson's algorithm and its lattice and lattice-ladder by-products.

Chapter 7 is a must for any machine learning course. Courses on statistical SP can also accommodate the first part of the chapter dealing with the classical Bayesian classification—the classical Bayesian decision theory. This chapter introduces the first case study of the book and it concerns the protein folding prediction task.

The aforementioned six chapters comprise the part of the book that deals with more or less classical topics. The rest of the chapters deal with more advanced techniques and can fit with any course dealing with machine learning or statistical/adaptive signal processing, depending on the focus, the time constraints, and the background of the audience.

Chapter 8 deals with convexity, a topic that is receiving more and more attention recently. The chapter presents the basic definitions concerning convex sets and functions and the notion of projection. These are important tools used in a number of recently developed algorithms. Also, the classical projections over convex sets (POCS) algorithm and the set theoretic approach to online learning are discussed as an alternative to gradient-descent based schemes. Then, the task of optimization of non-smooth convex loss functions is discussed, and the family of proximal mapping, alternating direction method of multipliers (ADMM), and forward backward-splitting methods are presented. This is a chapter that can be used when the emphasis of the course is optimization. Employing non-smooth loss functions and/or non-smooth regularization terms, in place of the LS and its ridge regression relative, is a trend of high research and practical interest.

Chapters 9 and 10 deal with sparse modeling. The first of the two chapters introduces the main concepts and ideas and the second deals with algorithms for batch as well for as online learning scenarios. Also, in the second chapter, a case study in the context of time-frequency analysis is discussed. Depending on time constraints, the main concepts behind sparse modeling and compressed sensing can be taught in a related course. These two chapters can also be used as a specialized course on sparsity on a postgraduate level.

Chapter 11 deals with learning in reproducing kernel Hilbert spaces and nonlinear techniques. The first part of the chapter is a must for any course with an emphasis on classification. The support vector regression and support vector machines are treated in detail. Moreover, a course on statistical SP with an emphasis on nonlinear modeling can also include material and concepts from this chapter. Kernelized versions of the stochastic gradient descent rationale are treated in some detail as nonlinear versions of classical online algorithms. A case study dealing with authorship identification is discussed at the end of this chapter.

Chapters 12 and 13 deal with Bayesian learning. Thus, both chapters can become the backbone of a course on machine learning and statistical SP that intends to emphasize Bayesian methods. The former of the chapters deals with the basic principles and it is an introduction to the expectation-maximization (EM) algorithm. The use of this celebrated algorithm is demonstrated in the context of two classical applications, that of the linear regression and the Gaussian mixture modeling for probability density function estimation. The second chapter deals with approximate inference techniques, and one can use parts of it, depending on the time constraints and the background of the audience. Sparse Bayesian learning and the relevant vector machine (RVM) framework is introduced. At the end of this chapter, Gaussian processes and nonparametric Bayesian techniques are discussed, and a case study concerning hyper-spectral image unmixing is presented. Both chapters, in their full length, can be used as a specialized course on Bayesian learning.

Chapters 14 and 17 deal with Monte Carlo sampling methods. The latter chapter deals with particle filtering. Both chapters, together with the two previous ones that deal with Bayesian learning, can be combined in a course whose emphasis is on statistical methods of machine learning/statistical signal processing.

Chapters 15 and 16 deal with probabilistic graphical models. The former chapter introduces the main concepts and definitions, and at the end introduces the message passage algorithm for chains and trees. This chapter is a must for any course whose emphasis is on probabilistic graphical models. The latter of the two chapters deals with message passage algorithms on junction trees and then with approximate inference techniques. Dynamic graphical models and hidden Markov models (HMM) are introduced. The Baum-Welch and the Viterbi schemes are derived as special cases of message passage algorithms by treating HMM as a special instance of a junction tree.

Chapter 18 deals with neural networks and deep learning. This chapter is also a must in any course with an emphasis on classification. The perceptron algorithm and the backpropagation algorithms are discussed in detail, and then the discussion moves into deep architectures and their training. A case study in the context of optical character recognition is discussed.

Chapter 19 is on dimensionality reduction techniques and latent variable modeling. The methods of principle component analysis (PCA), canonical correlations analysis (CCA), and independent component analysis (ICA) are introduced. The probabilistic approach to latent variable modeling is

discussed, and the probabilistic PCA (PPCA) is presented. Then, the focus turns to dictionary learning and robust PCA. Nonlinear dimensionality reduction techniques such as kernel PCA are discussed, along with the methods of local linear embedding (LLE) and isometric mapping (ISOMAP). Finally, a case study in the context of functional magnetic resonance imaging (fMRI) data analysis, based on ICA, is presented.

Each chapter starts with the basics and moves on to cover more recent advances in the related topic. This is true mainly for the text beginning with Chapter 7, since the first six chapters cover more classical material.

In summary, we provide the following suggestions for different courses, depending on the emphasis that the instructor wants to place on various topics.

- Machine Learning with emphasis on classification:
  - Main chapters: 3, 7, 11, and 18
  - Secondary chapters: 12 and 13, and possibly the first part of 6
- Statistical Signal Processing:
  - Main chapters: 3, 4, 6, and 12
  - Secondary chapters: 5 (first part) and 13–17
- Machine Learning with emphasis on Bayesian techniques:
  - Main chapters: 3 and 12–14
  - Secondary chapters: 7, 15, and 16, and possibly the first part of 6
- Adaptive Signal Processing:
  - Main chapters: 3–6
  - Secondary chapters: 8, 9, 11, 14, and 17

I believe that the above suggestions of following various combinations of chapters is possible, since the book has been written in such a way as to make individual chapters as self-contained as possible.

At the end of most of the chapters there are Matlab exercises, mainly based on the various examples given in the text. The required Matlab code is available on the book's website, together with the solutions manual. Also, all figures of the book are available on the book's website.

REFERENCES

[1] S. Theodoridis, K. Koutroumbas, Pattern Recognition, fourth ed., Academic Press, Amsterdam, 2009.

[2] S. Theodoridis, A. Pikrakis, K. Koutroumbas, D. Cavouras, Introduction to Pattern Recognition: A MATLAB Approach, Academic Press, Amsterdam, 2010.