LEARNING FROM DATA

Concepts, Theory, and Methods

Second Edition

VLADIMIR CHERKASSKY
FILIP MULIER

IEEE PRESS

WILEY-INTERSCIENCE
A JOHN WILEY & SONS, INC., PUBLICATION

LEARNING FROM DATA

THE WILEY BICENTENNIAL–KNOWLEDGE FOR GENERATIONS

Each generation has its unique needs and aspirations. When Charles Wiley first opened his small printing shop in lower Manhattan in 1807, it was a generation of boundless potential searching for an identity. And we were there, helping to define a new American literary tradition. Over half a century later, in the midst of the Second Industrial Revolution, it was a generation focused on building the future. Once again, we were there, supplying the critical scientific, technical, and engineering knowledge that helped frame the world. Throughout the 20th Century, and into the new millennium, nations began to reach out beyond their own borders and a new international community was born. Wiley was there, expanding its operations around the world to enable a global exchange of ideas, opinions, and know-how.

For 200 years, Wiley has been an integral part of each generation's journey, enabling the flow of information and understanding necessary to meet their needs and fulfill their aspirations. Today, bold new technologies are changing the way we live and learn. Wiley will be there, providing you the must-have knowledge you need to imagine new worlds, new possibilities, and new opportunities.

Generations come and go, but you can always count on Wiley to provide you the knowledge you need, when and where you need it!

WILLIAM J. PESCE
PRESIDENT AND CHIEF EXECUTIVE OFFICER

PETER BOOTH WILEY
CHAIRMAN OF THE BOARD

LEARNING FROM DATA

Concepts, Theory, and Methods

Second Edition

VLADIMIR CHERKASSKY
FILIP MULIER

IEEE PRESS

WILEY-INTERSCIENCE
A JOHN WILEY & SONS, INC., PUBLICATION

Published by John Wiley & Sons, Inc., Hoboken, New Jersey
Published simultaneously in Canada

Library of Congress Cataloging-in-Publication Data:

Cherkassky, Vladimir S.
Learning from data : concepts, theory, and methods / by Vladimir Cherkassky, Filip Mulier. – 2nd ed.
p. cm.
ISBN 978-0-471-68182-3 (cloth)
1. Adaptive signal processing. 2. Machine learning. 3. Neural networks (Computer science) 4. Fuzzy systems. I. Mulier, Filip. II. Title.
TK5102.9.C475 2007
006.3′1–dc22 2006038736

Printed in the United States of America
10 9 8 7 6 5 4 3 2 1

CONTENTS

PREFACE

There are two problems in modern science:

- too many people use different terminology to solve the same problems;
- even more people use the same terminology to address completely different issues.

Anonymous

In recent years, there has been an explosive growth of methods for learning (or estimating dependencies) from data. This is not surprising given the proliferation of

- low-cost computers (for implementing such methods in software)
- low-cost sensors and database technology (for collecting and storing data)
- highly computer-literate application experts (who can pose "interesting" application problems)

A learning method is an algorithm (usually implemented in software) that estimates an unknown mapping (dependency) between a system's inputs and outputs from the available data, namely from known (input, output) samples. Once such a dependency has been accurately estimated, it can be used for prediction of future system outputs from the known input values. This book provides a unified description of principles and methods for learning dependencies from data.

Methods for estimating dependencies from data have been traditionally explored in diverse fields such as statistics (multivariate regression and classification), engineering (pattern recognition), and computer science (artificial intelligence, machine

learning, and, more recently, data mining). Recent interest in learning from data has resulted in the development of biologically motivated methodologies, such as artificial neural networks, fuzzy systems, and wavelets.

Unfortunately, developments in each field are seldom related to other fields, despite the apparent commonality of issues and methods. The mere fact that hundreds of ''new'' methods are being proposed each year at various conferences and in numerous journals suggests a certain lack of understanding of the basic issues common to all such methods.

The premise of this book is that there are just a handful of important principles and issues in the field of learning dependencies from data. Any researcher or practitioner in this field needs to be aware of these issues in order to successfully apply a particular methodology, understand a method's limitations, or develop new techniques.

This book is an attempt to present and discuss such issues and principles (common to all methods) and then describe representative popular methods originating from statistics, neural networks, and pattern recognition. Often methods developed in different fields can be related to a common conceptual framework. This approach enables better understanding of a method's properties, and it has methodological advantages over traditional ''cookbook'' descriptions of various learning algorithms.

Many aspects of learning methods can be addressed under a traditional statistical framework. At the same time, many popular learning algorithms and learning methodologies have been developed outside classical statistics. This happened for several reasons:

1. Traditionally, the statistician's role has been to analyze the inferential limitations of the structural model constructed (proposed) by the application-domain expert. Consequently, the conceptual approach (adopted in statistics) is parameter estimation for model identification. For many real-life problems that require flexible estimation with finite samples, the statistical approach is fundamentally flawed. As shown in this book, learning with finite samples should be based on the framework known as risk minimization, rather than density estimation.
2. Statisticians have been late to recognize and appreciate the importance of computer-intensive approaches to data analysis. The growing use of computers has fundamentally changed the traditional boundaries between a statistician (data modeler) and a user (application expert). Nowadays, engineers and computer scientists successfully use sophisticated empirical data-modeling techniques (i.e., neural nets) to estimate complex nonlinear dependencies from the data.
3. Statistics (being part of mathematics) has developed into a closed discipline, with its own scientific jargon and academic objectives that favor analytic proofs rather than practical methods for learning from data.

Historically, we can identify three stages in the development of predictive learning methods. First, in 1985–1992 classical statistics gave way to neural networks (and other empirical methods, such as fuzzy systems) due to an early enthusiasm and naive claims that biologically inspired methods (i.e., neural nets) can achieve model-free learning not subject to statistical limitations. Even though such claims later proved to be false, this stage had a positive impact by showing the power and usefulness of flexible nonlinear modeling based on the risk minimization approach. Then in 1992–1996 came the return of statistics as the researchers and practitioners of neural networks became aware of their statistical limitations, initiating a trend toward interpretation of learning methods using a classical statistical framework. Finally, the third stage, from 1997 to present, is dominated by the wide popularity of support vector machines (SVMs) and similar margin-based approaches (such as boosting), and the growing interest in the Vapnik–Chervonenkis (VC) theoretical framework for predictive learning.

This book is intended for readers with varying interests, including researchers/practitioners in data modeling with a classical statistics background, researchers/practitioners in data modeling with a neural network background, and graduate students in engineering or computer science.

The presentation does not assume a special math background beyond a good working knowledge of probability, linear algebra, and calculus on an undergraduate level. Useful background material on optimization and linear algebra is included in Appendixes A and B, respectively. We do not provide mathematical proofs, but, whenever possible, in place of proofs we provide intuitive explanations and arguments. Likewise, mathematical formulation and discussion of the major concepts and results are provided as needed. The goal is to provide a unified treatment of diverse methodologies (i.e., statistics and neural networks), and to that end we carefully define the terminology used throughout the book. This book is not easy reading because it describes fairly complex concepts and mathematical models for solving inherently difficult (ill-posed) problems of learning with finite data. To aid the reader, each chapter starts with a brief overview of its contents. Also, each chapter is concluded with a summary containing an overview of open research issues and pointers to other (relevant) chapters.

Book chapters are conceptually organized into three parts:

- *Part I: Concepts and Theory* (Chapters 1–4). Following an introduction and motivation given in Chapter 1, we present formal specification of the inductive learning problem in Chapter 2 that also introduces major concepts and issues in learning from data. In particular, it describes an important concept called an *inductive principle*. Chapter 3 describes the regularization (or penalization) framework adopted in statistics. Chapter 4 describes Vapnik's statistical learning theory (SLT), which provides the theoretical basis for predictive learning with finite data. SLT, aka VC theory, is important for understanding various learning methods developed in neural networks, statistics, and pattern recognition, and for developing new approaches, such as SVMs

(described in Chapter 9) and noninductive learning settings (described in Chapter 10).

- *Part II: Constructive Learning Methods* (Chapters 5–8). This part describes learning methods for regression, classification, and density approximation problems. The objective is to show conceptual similarity of methods originating from statistics, neural networks, and signal processing and to discuss their relative advantages and limitations. Whenever possible, we relate constructive learning methods to the conceptual framework of Part I. Chapter 5 describes nonlinear optimization strategies commonly used in various methods. Chapter 6 describes methods for density approximation, which include statistical, neural network, and signal processing techniques for data reduction and dimensionality reduction. Chapter 7 provides descriptions of statistical and neural network methods for regression. Chapter 8 describes methods for classification.
- *Part III: VC-Based Learning Methodologies* (Chapters 9 and 10). Here we describe constructive learning approaches that originate in VC theory. These include SVMs (or margin-based methods) for several inductive learning problems (in Chapter 9) and various noninductive learning formulations (described in Chapter 10).

The chapters should be followed in a sequential order, as the description of constructive learning methods is related to the conceptual framework developed in the first part of the book. A shortened sequence of Chapters 1–3 followed by Chapters 5, 6, 7 and 8 is recommended for the beginning readers who are interested only in the description of statistical and neural network methods. This sequence omits the mathematically and conceptually challenging Chapters 4 and 9. Alternatively, more advanced readers who are primarily interested in SLT and SVM methodology may adopt the sequence of Chapters 2, 3, 4, 9, and 10.

In the course of writing this book, our understanding of the field has changed. We started with the currently prevailing view of learning methods as a collection of tricks. Statisticians have their own bag of tricks (and terminology), neural networks have a different set of tricks, and so on. However, in the process of writing this book, we realized that it is possible to understand the various heuristic methods (tricks) by a sound general conceptual framework. Such a framework is provided by SLT developed mainly by Vapnik over the past 35 years. This theory combines fundamental concepts and principles related to learning with finite data, well-defined problem formulations, and rigorous mathematical theory. Although SLT is well known for its *mathematical* aspects, its *conceptual* contributions are not fully appreciated. As shown in our book, the conceptual framework provided by SLT can be used for improved understanding of various learning methods even where its mathematical results cannot be directly applied. Modern learning methods (i.e., flexible approaches using finite data) have slowly drifted away from the original problem statements posed in classical statistical decision and estimation theory. A major conceptual contribution of SLT is in revisiting the problem

statement appropriate for modern data mining applications. On the very basic level, SLT makes a clear distinction between the problem formulation and a solution approach (aka inductive principle) used to solve a problem. Although this distinction appears trivial on the surface, it leads to a fundamentally new understanding of the learning problem not explained by classical theory. Although it is tempting to skip directly to constructive solutions, this book devotes enough attention to the learning problem formulation and important concepts *before* describing actual learning methods.

Over the past 10 years (since the first edition of this book), we have witnessed considerable growth of interest in SVM-related methods. Nowadays, SVM (aka kernel) methods are commonly used in data mining, statistics, signal processing, pattern recognition, genomics, and so on. In spite of such an overwhelming success and wide recognition of SVM methodology, many important VC theoretical concepts responsible for good generalization of SVMs (such as margin, VC dimension) remain rather poorly understood. For example, many recent monographs and research papers refer to SVMs as a ''special case of regularization.'' So in this second edition, we made a special effort to emphasize the conceptual aspects of VC theory and to contrast the VC theoretical approach to learning (i.e., *system imitation*) versus the classical statistical and function approximation approach (i.e., *system identification*). Accurate interpretation of VC theoretical concepts is important for improved understanding of inductive learning algorithms, as well as for developing emerging state-of-the-art approaches based on noninductive learning settings (as discussed in Chapter 10). In this edition, we emphasize the philosophical interpretation of predictive learning, in general, and of several VC theoretical concepts, in particular. These philosophical connections appear to be quite useful for understanding recent advanced learning methods and for motivating new noninductive types of inference. Moreover, philosophical aspects of predictive learning can be immediately related to epistemology (understanding of human knowledge), as discussed in Chapter 11.

Many people have contributed directly and indirectly to this book. First and foremost, we are greatly indebted to Vladimir Vapnik of NEC Labs for his fundamental contributions to SLT and for his patience in explaining this theory to us. We would like to acknowledge many people whose constructive feedback helped improve the quality of the second edition, including Ella Bingham, John Boik, Olivier Chapelle, David Hand, Nicol Schraudolph, Simon Haykin, David Musicant, Erinija Pranckeviciene, and D. Solomatine—all of whom provided many useful comments.

This book was used in the graduate course ''Predictive Learning from Data'' at the University of Minnesota over the past 10 years, and we would like to thank students who took this course for their valuable feedback. In particular, we acknowledge former graduate students X. Shao, Y. Ma, T. Xiong, L. Liang, H Gao, M. Ramani, R. Singh, and Y. Kim whose research contributions are incorporated in this book in the form of several fine figures and empirical

comparisons. Finally, we would like to thank our families for their patience and support.

Vladimir Cherkassky
Filip Mulier

Minneapolis, Minnesota
March 2007

NOTATION

The following uniform notation is used throughout the book. Scalars are indicated by script letters such as a. Vectors are indicated by lowercase bold letters such as $\mathbf{w}$. Matrices are given using uppercase bold letters $\mathbf{V}$. When elements of a matrix are accessed individually, we use the corresponding lowercase script letter. For example, the (i,j) element of the matrix $\mathbf{V}$ is v_{ij}. Common notation for all chapters is as follows:

Data

n	Number of samples
d	Number of input variables
$\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_n]$	Matrix of input samples
$\mathbf{y} = [y_1, \ldots, y_n]$	Vector of output samples
$\mathbf{Z} = [\mathbf{X}, \mathbf{y}]$	Combined input–output training data or
$\mathbf{Z} = [\mathbf{z}_1, \ldots, \mathbf{z}_n]$	Representation of data points in a feature space

Distribution

P	Probability
$F(\mathbf{x})$	Cumulative probability distribution function (cdf)
$p(\mathbf{x})$	Probability density function (pdf)
$p(\mathbf{x}, y)$	Joint probability density function
$p(\mathbf{x}; \omega)$	Probability density function, which is parameterized
$p(y\|\mathbf{x})$	Conditional density
$t(\mathbf{x})$	Target function

Approximating Functions

$f(\mathbf{x}, \omega)$	A class of approximating functions indexed by abstract parameter ω (ω can be a scalar, vector, or matrix). Interpretation of $f(\mathbf{x}, \omega)$ depends on the particular learning problem

$f(\mathbf{x}, \omega_0)$	The function that minimizes the expected risk (optimal solution)
$f(\mathbf{x}, \omega^*)$	Estimate of the optimal solution obtained from finite data
$f(\mathbf{x}, \mathbf{w}, \mathbf{V}) = \sum_{i=1}^{m} w_i g_i(\mathbf{x}, \mathbf{v}_i) + b$	Basis function expansion of approximating functions with bias term
$g_i(\mathbf{x}, \mathbf{v})$	Basis function in a basis function expansion
$w, \mathbf{w}, \mathbf{W}$	Parameters of approximating function
$v, \mathbf{v}, \mathbf{V}$	Basis function parameters
m	Number of basis functions
Ω	Set of parameters, as in $\mathbf{w} \in \Omega$
Δ	Margin distance
$t(\mathbf{x})$	Target function
ξ	Error between the target function and the approximating function, or error between model estimate and time output

Risk Functionals

$L(y, f(\mathbf{x}, \omega))$	Discrepancy measure or loss function
L_2	Squared discrepancy measure
$Q(\omega)$	A set of loss functions
R	Risk or average loss
$R(\omega)$	Expected risk as a function of parameters
$R_{\text{emp}}(\omega)$	Empirical risk as a function of parameters

Kernel Functions

$K(\mathbf{x}, \mathbf{x}')$	General kernel function (for kernel smothing)
$S(\mathbf{x}, \mathbf{x}')$	Equivalent kernel of a linear estimator
$H(\mathbf{x}, \mathbf{x}')$	Inner product kernel

Miscellaneous

$(\mathbf{a} \cdot \mathbf{b})$	Inner (dot) product of two vectors
$I()$	Indicator function of a Boolean argument that takes the value 1 if its argument is true and 0 otherwise. By convention, for a real-valued argument, $I(x) = 1$ for $x > 0$, and $I(x) = 0$ for $x \leq 0$
$\phi[f(\mathbf{x}, \omega)]$	Penalty functional
λ	Regularization parameter
h	VC dimension
γ_k	Learning rate for stochastic approximation at iteration step k
$[a]_+$	Positive argument, equals max $(a, 0)$
L	Lagrangian

In addition to the above notation used throughout the book, there is chapter-specific notation, which will be introduced locally in each chapter.

1

INTRODUCTION

Where observation is concerned, chance favors only the prepared mind.
Louis Pasteur

This chapter describes the motivation and reasons for the growing interest in methods for learning (or estimation of empirical dependencies) from data and introduces informally some relevant terminology.

Section 1.1 points out that the problem of learning from data is just one part of the general experimental procedure used in different fields of science and engineering. This procedure is described in detail, with emphasis on the importance of other steps (preceding learning) for overall success. Two distinct goals of learning from data, predictive accuracy (generalization) and interpretation (explanation), are also discussed.

Section 1.2 discusses the relationship between statistical dependency and the notion of causality. It is pointed out that causality cannot be inferred from data analysis alone, but must be demonstrated by arguments outside the statistical analysis. Several examples are presented to support this point.

Section 1.3 describes different types of variables for representing the inputs and outputs of a learning system. These variable types are numeric, categorical, periodic, and ordinal.

Section 1.4 overviews several approaches for describing uncertainty. These include traditional (frequentist) probability corresponding to measurable frequencies,

Learning From Data: Concepts, Theory, and Methods, Second Edition
By Vladimir Cherkassky and Filip Mulier

Bayesian probability quantifying subjective belief, and fuzzy sets for characterization of event ambiguity. The distinction and similarity between these approaches are discussed. The difference between the probability as characterization of event randomness and fuzziness as characterization of the ambiguity of deterministic events is explained and illustrated by examples.

This book is mainly concerned with estimation of *predictive* models from data. This framework, called Predictive Learning, is formally introduced in Chapter 2. However, in many applications data-driven modeling pursues different goals (other than prediction). Several major data analytic methodologies are described and contrasted to Predictive Learning in Section 1.5.

1.1 LEARNING AND STATISTICAL ESTIMATION

Modern science and engineering are based on using *first-principle* models to describe physical, biological, and social systems. Such an approach starts with a basic scientific model (e.g., Newton's laws of mechanics or Maxwell's theory of electromagnetism) and then builds upon them various applications in mechanical engineering or electrical engineering. Under this approach, experimental data (measurements) are used to verify the underlying first-principle models and to estimate some of the model parameters that are difficult to measure directly. However, in many applications the underlying first principles are unknown or the systems under study are too complex to be mathematically described. Fortunately, with the growing use of computers and low-cost sensors for data collection, there is a great amount of data being generated by such systems. In the absence of first-principle models, such readily available data can be used to derive models by estimating useful relationships between a system's variables (i.e., unknown input–output dependencies). Thus, there is currently a paradigm shift from the classical modeling based on first principles to developing models from data.

The need for understanding large, complex, information-rich data sets is common to virtually all fields of business, science, and engineering. Some examples include medical diagnosis, handwritten character recognition, and time series prediction. In the business world, corporate and customer data are becoming recognized as a strategic asset. The ability to extract useful knowledge hidden in these data and to act on that knowledge is becoming increasingly important in today's competitive world.

Many recent approaches to developing models from data have been inspired by the learning capabilities of biological systems and, in particular, those of humans. In fact, biological systems learn to cope with the unknown statistical nature of the environment in a data-driven fashion. Babies are not aware of the laws of mechanics when they learn how to walk, and most adults drive a car without knowledge of the underlying laws of physics. Humans as well as animals also have superior pattern recognition capabilities for tasks such as face, voice, or smell recognition. People are not born with such capabilities, but learn them through

data-driven interaction with the environment. Usually humans cannot articulate the rules they use to recognize, for example, a face in a complex picture. The field of pattern recognition has a goal of building artificial pattern recognition systems that imitate human recognition capabilities. Pattern recognition systems are based on the principles of engineering and statistics rather than biology. There always has been an appeal to build pattern recognition systems that imitate human (or animal) brains. In the mid-1980s, this led to great enthusiasm about the so-called (artificial) neural networks. Even though most neural network models and applications have little in common with biological systems and are used for standard pattern recognition tasks, the biological terminology still remains, sometimes causing considerable confusion for newcomers from other fields. More recently, in the early 1990s, another biologically inspired group of learning methods known as fuzzy systems became popular. The focus of fuzzy systems is on highly interpretable representation of human application-domain knowledge based on the assertion that human reasoning is ''naturally'' performed using fuzzy rules. On the contrary, neural networks are mainly concerned with data-driven learning for good generalization. These two goals are combined in the so-called neurofuzzy systems.

The authors of this book do not think that biological analogy and terminology are of major significance for artificial learning systems. Instead, the book concentrates on using a statistical framework to describe modern methods for learning from data. In statistics, the task of predictive learning (from samples) is called statistical estimation. It amounts to estimating properties of some (unknown) statistical distribution from known samples or training data. Information contained in the training data (past experience) can be used to answer questions about future samples. Thus, we distinguish two stages in the operation of a learning system:

1. Learning/estimation (from training samples)
2. Operation/prediction, when predictions are made for future or test samples

This description assumes that both the training and test data are from the *same* underlying statistical distribution. In other words, this (unknown) distribution is fixed. Specific learning tasks include the following:

- Classification (pattern recognition) or estimation of class decision boundaries
- Regression: estimation of unknown real-valued function
- Probability density estimation (from samples)

A precise mathematical formulation of the learning problem is given in Chapter 2.

There are two common types of the learning problems discussed in this book, known as supervised learning and unsupervised learning. *Supervised* learning is used to estimate an unknown (input, output) mapping from known (input, output) samples. Classification and regression tasks fall into this group. The term ''supervised'' denotes the fact that output values for training samples are known (i.e., provided by a ''teacher'' or a system being modeled). Under the *unsupervised*

learning scheme, only input samples are given to a learning system, and there is no notion of the output during learning. The goal of unsupervised learning may be to approximate the probability distribution of the inputs or to discover ''natural'' structure (i.e., clusters) in the input data. In biological systems, low-level perception and recognition tasks are learned via unsupervised learning, whereas higher-level capabilities are usually acquired through supervised learning. For example, babies learn to recognize (''cluster'') familiar faces long before they can understand human speech. On the contrary, reading and writing skills cannot be acquired in unsupervised manner; they need to be taught. This observation suggests that biological unsupervised learning schemes are based on powerful internal structures (for optimal representation and processing of sensory data) developed through the years of evolution, in the process of adapting to the statistical nature of the environment. Hence, it may be beneficial to use biologically inspired structures for unsupervised learning in artificial learning systems. In fact, a well-known example of such an approach is the popular method known as the self-organizing map for unsupervised learning described in Chapter 6. Finally, it is worth noting here that the distinction between supervised and unsupervised learning is on the level of problem statement only. In fact, methods originally developed for supervised learning can be adapted for unsupervised learning tasks, and vice versa. Examples are given throughout the book.

It is important to realize that the problem of learning/estimation of dependencies from samples is only one part of the general experimental procedure used by scientists, engineers, medical doctors, social scientists, and others who apply statistical (neural network, machine learning, fuzzy, etc.) methods to draw conclusions from the data. The general experimental procedure adopted in classical statistics involves the following steps, adapted from Dowdy and Wearden (1991):

1. State the problem
2. Formulate the hypothesis
3. Design the experiment/generate the data
4. Collect the data and perform preprocessing
5. Estimate the model
6. Interpret the model/draw the conclusions

Even though the focus of this book is on step 5, it is just one step in the procedure. Good understanding of the whole procedure is important for any successful application. No matter how powerful the learning method used in step 5 is, the resulting model would not be valid if the data are not informative (i.e., gathered incorrectly) or the problem formulation is not (statistically) meaningful. For example, poor choice of the input and output variables (steps 1 and 2) and improperly chosen encoding/feature selection (step 4) may adversely affect learning/inference from data (step 5), or even make it impossible. Also, the type of inference procedure used in step 5 may be indirectly affected by the problem formulation in step 2, experiment design in step 3, and data collection/preprocessing in step 4.

Next, we briefly discuss each step in the above general procedure.

Step 1: Statement of the problem. Most data modeling studies are performed in a particular application domain. Hence, domain-specific knowledge and experience are usually necessary in order to come up with a meaningful problem statement. Unfortunately, many recent application studies tend to focus on the learning methods used (i.e., a neural network) at the expense of a clear problem statement.

Step 2: Hypothesis formulation. The hypothesis in this step specifies an unknown dependency, which is to be estimated from experimental data. At this step, a modeler usually specifies a set of input and output variables for the unknown dependency and (if possible) a general form of this dependency. There may be several hypotheses formulated for a single problem. Step 2 requires combined expertise of an application domain and of statistical modeling. In practice, it usually means close interaction between a modeler and application experts.

Step 3: Data generation/experiment design. This step is concerned with how the data are generated. There are two distinct possibilities. The first is when the data generation process is under control of a modeler—it is known as the *designed experiment* setting in statistics. The second is when the modeler cannot influence the data generation process—this is known as the *observational* setting. An observational setting, namely random data generation, is assumed in this book. We will also refer to a random distribution used to generate data (inputs) as a *sampling distribution*. Typically, the sampling distribution is not completely unknown and is implicit in the data collection procedure. It is important to understand how the data collection affects the sampling distribution because such a priori knowledge can be very useful for modeling and interpretation of modeling results. Further, it is important to make sure that past (training) data used for model estimation, and the future data used for prediction, come from the same (unknown) sampling distribution. If this is not the case, then (in most cases) predictive models estimated from the training data alone cannot be used for prediction with the future data.

Step 4: Data collection and preprocessing. This step has to do with both data collection and the subsequent preprocessing of data. In the observational setting, data are usually ''collected'' from the existing databases. Data preprocessing includes (at least) two common tasks: outlier detection/removal and data preprocessing/encoding/feature selection.

Outliers are unusual data values that are not consistent with most observations. Commonly, outliers are due to gross measurement errors, coding/recording errors, and abnormal cases. Such nonrepresentative samples can seriously affect the model produced later in step 5. There are two strategies for dealing with outliers: outlier detection and removal as a part of preprocessing, and development of robust modeling methods that are (by design) insensitive to outliers. Such robust statistical methods (Huber 1981)

are not discussed in this book. Note that there is a close connection between outlier detection (in step 4) and modeling (in step 5).

Data preprocessing includes several steps such as variable scaling and different types of encoding techniques. Such application-domain-specific encoding methods usually achieve dimensionality reduction by providing a small number of informative features for subsequent data modeling. Once again, preprocessing steps should not be considered completely independent from modeling (in step 5): There is usually a close connection between the two. For example, consider the task of variable scaling. The problem of scaling is due to the fact that different input variables have different natural scales, namely their own units of measurement. For some modeling methods (e.g., classification trees) this does not cause a problem, but other methods (e.g., distance-based methods) are very sensitive to the chosen scale of input variables. With such methods, a variable characterizing weight would have much larger influence when expressed in milligrams rather than in pounds. Hence, each input variable needs to be rescaled. Commonly, such rescaling is done independently for each variable; that is, each variable may be scaled by the standard deviation of its values. However, independent scaling of variables can lead to suboptimal representation for many learning methods.

Preprocessing/encoding step often includes selection of a small number of informative features from a high-dimensional data. This is known as *feature selection* in pattern recognition. It may be argued that good preprocessing/data encoding is the most important part in the whole procedure because it provides a small number of informative features, thus making the task of estimating dependency much simpler. Indeed, the success of many application studies is usually due to a clever preprocessing/data encoding scheme rather than to the learning method used. Generally, a good preprocessing method provides an optimal representation for a learning problem, by incorporating a priori knowledge in the form of application-specific encoding and feature selection.

Step 5: Model estimation. Each hypothesis in step 2 corresponds to unknown dependency between the input and output features representing appropriately encoded variables. These dependencies are quantified using available data and a priori knowledge about the problem. The main goal is to construct models for accurate prediction of future outputs from the (known) input values. The goal of predictive accuracy is also known as *generalization* capability in biologically inspired methods (i.e., neural networks). Traditional statistical methods typically use fixed parametric functions (usually *linear in parameters*) for modeling the dependencies. In contrast, more recent methods described in this book are based on much more flexible modeling assumptions that, in principle, enable estimating nonlinear dependencies of an arbitrary form.

Step 6: Interpretation of the model and drawing conclusions. In many cases, predictive models developed in step 5 need to be used for (human) decision making. Hence, such models need to be interpretable in order to be useful

because humans are not likely to base their decisions on complex ''black-box'' models. Note that the goals of accurate prediction and interpretation are rather different because interpretable models would be (necessarily) simple but accurate predictive models may be quite complex. The traditional statistical approach to this dilemma is to use highly interpretable (structured) parametric models for estimation in step 5. In contrast, modern approaches favor methods providing high prediction accuracy, and then view interpretation as a separate task.

Most of this book is on formal methods for estimating dependencies from data (i.e., step 5). However, other steps are equally important for an overall application success. Note that the steps preceding model estimation strongly depend on the application-domain knowledge. Hence, practical applications of learning methods require a *combination* of modeling expertise with application-domain knowledge. These issues are further explored in Section 2.3.4.

As steps 1–4 preceding model estimation are application domain dependent, they cannot be easily formalized, and they are beyond the scope of this book. For this reason, most examples in this book use simulated data sets, rather than real-life data.

Notwithstanding the goal of an accurate predictive model (step 5), most scientific research and practical applications of predictive learning also result in gaining better *understanding* of unknown dependencies (step 6). Such understanding can be useful for

- Gaining insights about the unknown system
- Understanding the limits of applicability of a given modeling method
- Identifying the most important (relevant) input variables that are responsible for the most variation of the output
- Making decisions based on the interpretation of the model.

It should be clear that for real-life applications, meaningful interpretation of the predictive learning model usually requires a good understanding of the issues and choices in steps 1–4 (preceding to the learning itself).

Finally, the interpretation formalism adopted in step 6 often depends on the target audience. For example, standard interpretation methods in statistics (i.e., analysis of variance decomposition) may not be familiar to an engineer who may instead prefer to use fuzzy rules for interpretation.

1.2 STATISTICAL DEPENDENCY AND CAUSALITY

Statistical inference and learning systems are concerned with estimating unknown dependencies hidden in the data, as shown in Fig. 1.1. This procedure corresponds to step 5 in the general procedure described in Section 1.1, but the input and output variables denote preprocessed features of step 4. The goal of predictive learning is

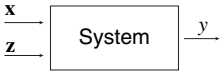

FIGURE 1.1 Real systems often have unobserved inputs **z**.

to estimate unknown dependency between the input $(\mathbf{x})$ and output (y) variables, from a set of past observations of $(\mathbf{x}, y)$ values. In Fig. 1.1, the other set of variables labeled **z** denotes all other factors that affect the outputs but whose values are not observed or controlled. For example, in manufacturing process control, the quality of the final product (output y) can be affected by nonobserved factors such as variations in the temperature/humidity of the environment or small variations in (human) operator actions. In the case of economic modeling based on the analysis of (past) economic data, nonobserved and noncontrolled variables include, for example, the black market economy, as well as quantities that are inherently difficult to measure, such as software productivity. Hence, the knowledge of observed input values $(\mathbf{x})$ does not uniquely specify the outputs (y). This uncertainty in the outputs reflects the lack of knowledge of the unobserved factors $(\mathbf{z})$, and it results in *statistical dependency* between the observed inputs and output(s). The effect of unobserved inputs can be characterized by a conditional probability distribution $p(y|\mathbf{x})$, which denotes the probability that y will occur given the input $\mathbf{x}$.

Sometimes the existence of statistical dependencies between system inputs and outputs (see Fig 1.1) is (erroneously) used to demonstrate cause-and-effect relationship between variables of interest. Such misinterpretation is especially common in social studies and political arguments. We will discuss the difference between statistical dependency and causality and show some examples. The main point is that *causality* cannot be inferred from data analysis *alone*; instead, it must be assumed or *demonstrated* by an argument outside the statistical analysis.

For example, consider (x, y) samples shown in Fig. 1.2. It is possible to interpret these data in a number of ways:

- Variables (x, y) are correlated
- Variable x statistically depends on y, that is, $x = g(y) + \text{error}$

Each formulation is based on different assumptions (about the nature of the data), and each would require different methods for dependency estimation. However,

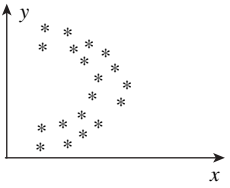

FIGURE 1.2 Scatterplot of two variables that have a statistical dependency.

statistical dependency does not imply causality. In fact, causality is not necessary for accurate estimation of the input–output dependency in either formulation. Meaningful interpretation of the input and output variables, in general, and specific assumptions about causality, in particular, should be made in step 1 or 2 of the general procedure discussed in Section 1.1. In some cases, these assumptions can be *supported* by the data, but they should never be deduced from the data alone.

Next, we consider several common instances of the learning problem shown in Fig. 1.1 along with their application-specific interpretation. For example, in manufacturing process control the causal relationship between controlled input variables and the output quality of the final product is based on understanding of the physical nature of the process. However, it does not make sense to claim causal relationship between person's height and weight, even though statistical dependency (correlation) between height and weight can be easily demonstrated from data. Similarly, it is well known that people in Florida are older (on average) than those in the rest of the United States. This observation does not imply, however, that the climate of Florida causes people to live longer (people just move there when they retire).

The next example is from a real-life study based on the statistical analysis of life expectancy for married versus single men. Results of this study can be summarized as follows: Married men live longer than single men. Does it imply that marriage is (causally) good for one's health; that is, does marriage increase life expectancy? Most likely not. It can be argued that males with physical problems and/or socially deviant patterns of behavior are less likely to get married, and this explains why married men live longer. If this explanation is true, the observed statistical dependency between the input (person's marriage status) and the output (life expectancy) is due to other (unobserved) factors such as person's health and social habits.

Another interesting example is medical diagnosis. Here the observed symptoms and/or test results (inputs $\mathbf{x}$) are used to diagnose (predict) the disease (output y). The predictive model in Fig. 1.1 gives the *inverse* causal relationship: It is the output (disease) that causes particular observed symptoms (input values).

We conclude that the task of learning/estimation of statistical dependency between (observed) inputs and outputs can occur in the following situations:

- Outputs causally depend on the (observed) inputs
- Inputs causally depend on the output(s)
- Input–output dependency is caused by other (unobserved) factors
- Input–output correlation is noncausal
- Any combination of them

Nevertheless, each possibility is specified by the arguments *outside* the data.

The preceding discussion has a negative bearing on naive approaches by some proponents of automatic data mining and knowledge discovery in databases. These approaches advocate the use of automatic tools for discovery of meaningful associations (dependencies) between variables in large databases. However, meaningful dependencies can be extracted from data only if the problem formulation is

meaningful, namely if it reflects a priori knowledge about the application domain. Such commonsense knowledge cannot be easily incorporated into general-purpose automatic knowledge discovery tools.

One situation when a causal relationship can be inferred from the data is when all relevant input factors (affecting the outputs) are observed and controlled in the formulation shown in Fig. 1.1. This is a rare situation for most applications of predictive learning and data mining. As a hypothetical example, consider again the life expectancy study. Let us assume that we can (magically) conduct a controlled experiment where the life expectancy is observed for the two groups of people identical in every (physical and social) respect, except that men in one group get married, and in the other stay single. Then, any different life expectancy in the two groups can be used to infer causality. Needless to say, such controlled experiments cannot be conducted for most social systems or physical systems of practical interest.

1.3 CHARACTERIZATION OF VARIABLES

Each of the input and output variables (or features) in Fig. 1.1 can be of several different types. The two most common types are *numeric* and *categorical*. Numeric type includes real-valued or integer variables (age, speed, length, etc.). A numeric feature has two important properties: Its values have an *order relation* and a *distance relation* defined for any two feature values. In contrast, categorical (or symbolic) variables have neither their order nor distance relation defined. The two values of a categorical variable can be either equal or unequal. Examples include eye color, sex, or country of citizenship. Categorical outputs in Fig. 1.1 occur quite often and represent a class of problems known as pattern recognition, classification, or discriminant analysis. Numeric (real-valued) outputs correspond to regression or (continuous) function estimation problems. Mathematical formulation for classification and regression problems is given in Chapter 2, and much of the book deals with approaches for solving these problems.

A categorical variable with two values can be converted, in principle, to a numeric binary variable with two values (0 or 1). A categorical variable with J values can be converted into J binary numeric variables, namely one binary variable for each categorical value. Representing a categorical variable by several binary variables is known as ''dummy variables'' encoding in statistics. In the neural network literature this method is known as 1-of-J encoding, indicating that each of the J binary variables encodes one feature value.

There are two other (less common) types of variables: periodic and ordinal. A *periodic* variable is a numeric variable for which the distance relation exists, but there is no order relation. Examples are day of the week, month, or year. An *ordinal* variable is a categorical variable for which an order relation is defined but no distance relation. Examples are gold, silver, and bronze medal positions in a sport competition or student ranking within a class. Typically, ordinal variables encode (map) a numeric variable onto a small set of *overlapping* intervals corresponding to

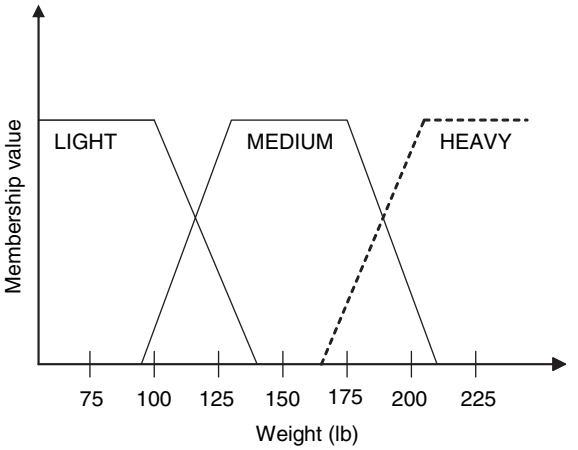

FIGURE 1.3 Membership functions corresponding to different fuzzy sets for the feature *weight*.

the values (labels) of an ordinal variable. Ordinal variables are closely related to linguistic or fuzzy variables commonly used in spoken English, for example, AGE (with values young, middle-aged, and old) and INCOME (with values low, middle-class, upper-middle-class, and rich). There are two reasons why the distance relation for the ordinal or fuzzy values is not defined. First, these values are often subjectively defined by humans in a particular context (hence known as linguistic values). For example, in a recent poll caused by the debate over changes in the U.S. tax code, families with an annual income between $40,000 and $50,000 classified incomes over $100,000 as rich, whereas families with an income of $100,000 defined themselves as middle-class. The second reason is that (even in a fixed context) there is usually no crisp boundary (distinction) between the two closest values. Instead, ordinal values denote overlapping sets. Figure 1.3 shows possible reasonable assignment values for an ordinal feature weight where, for example, the weight of 120 pounds can be encoded as both medium and light weight but with a different degree of membership. In other words, a single (numeric) input value can belong (simultaneously) to *several* values of an ordinal or fuzzy variable.

1.4 CHARACTERIZATION OF UNCERTAINTY

The main formalism adopted in this book (and most other sources) for describing uncertainty is based on the notions of probability and statistical distribution. Standard interpretation/definition of probability is given in terms of (measurable) frequencies, that is, a probability denotes the relative frequency of a random experiment with K possible outcomes, when the number of trials is very large (infinite). This traditional view is known as a *frequentist* interpretation. The $(\mathbf{x}, y)$ observations in the system shown in Fig. 1.1 are sampled from some (unknown) statistical

distribution, under the frequentist interpretation. Then, learning amounts to estimating parameters and/or structure of the unknown input–output dependency (usually related to the conditional probability $p(y|\mathbf{x})$) from the available data. This approach is introduced in Chapter 2, and most of the book describes concepts, theory, and methods based on this formulation. In this section, we briefly mention two other (alternative) ways of describing uncertainty.

Sometimes the frequentist interpretation does not make sense. For example, an economist predicting 80 percent chance of an interest rate cut in the near future does not really have in mind a random experiment repeated, say, 1000 times. In this case, the term probability is used to express a measure of *subjective degree of belief* in a particular outcome by an observer. Assuming events with disjoint outcomes (as in the frequentist interpretation), it is natural to encode subjective beliefs as real numbers between 0 and 1. The value of 1 indicates complete certainty that an event will occur, and 0 denotes complete certainty that an event will not occur. Then, such degrees of belief (provided they satisfy some natural consistency properties) can be viewed as conventional probabilities. This is known as the *Bayesian* interpretation of probabilities. The Bayesian interpretation is often used in statistical inference for specifying a priori knowledge (in the form of subjective prior probabilities) and combining this knowledge with available data via the Bayes theorem. The prior probability encodes our knowledge about the system before the data are known. This knowledge is encoded in the form of a prior probability distribution. The Bayes formula then provides a rule for updating prior probabilities after the data are known. This is known as Bayesian inference or the Bayesian inductive principle (discussed later in Section 2.3.3).

Note that probability is used to measure uncertainty in the *event outcome*. However, an event A itself can either occur or not. This is reflected in the probability identities:

$$P(A) + P(A^c) = 1, \qquad P(AA^c) = 0,$$

where A^c denotes a complement of A, namely $A^c = \text{not } A$, and $P(A)$ denotes the probability that event A will occur.

These properties hold for both the frequentist and Bayesian views of probability. This view of uncertainty is applicable if an observer is capable of unambiguously recognizing occurrence of an event. For example, an ''interest rate cut'' is an unambiguous event. However, in many situations the events themselves occur to a certain subjective degree, and (useful) characterization of uncertainty amounts to specifying a degree of such partial occurrence. For example, consider a feature *weight* whose values light, medium, and heavy correspond to overlapping intervals as shown in Fig. 1.3. Then, it is possible to describe uncertainty of a statement like

Person weighing x pounds is HEAVY

by a number (between 0 and 1), and denoted as $\mu_H(x)$. This is known as a *fuzzy membership function*, and it is used to quantify the degree of subjective belief that the above statement is true, that a person belongs to a (fuzzy) set HEAVY. Ordinal values LIGHT, MEDIUM, and HEAVY are examples of the

fuzzy sets (values), and the membership function is used to specify the degree of partial membership (i.e., of a person weighing x pounds in a fuzzy set HEAVY). As the membership functions corresponding to different fuzzy sets can overlap (see Fig. 1.3), a person weighing 170 pounds belongs to two fuzzy sets, H(eavy) and M(edium), and the sum of the two membership functions does not have to add up to 1. Moreover, a person weighing 170 pounds can belong *simultaneously* to fuzzy set HEAVY and to its complement *not* HEAVY. This type of uncertainty cannot be properly handled using probabilistic characterization of uncertainty, where a person cannot be HEAVY and *not* HEAVY at the same time. A description of uncertainty related to partial membership is provided by fuzzy logic (Zadeh 1965; Zimmerman 1996).

A continuous fuzzy set (linguistic variable) A is specified by the fuzzy membership function $\mu_A(x)$ that gives partial degree of membership of an object x in A. The fuzzy membership function, by definition, has values in the interval $[0, 1]$, to denote partial membership. The value $\mu_A(x) = 0$ means that an object x is not a member of the set A, and the value 1 indicates that x entirely belongs to A.

It is usually assumed that an object is (uniquely) characterized by a scalar feature x, so the fuzzy membership function $\mu_A(x)$ effectively represents a univariate function such that $0 \leq \mu_A(x) \leq 1$. Figure 1.4 illustrates the difference between the fuzzy set (or partial membership) and the traditional "crisp" set membership using different ways to define the concept "boiling temperature" as a function of the water temperature. Note that ordinary (crisp) sets can be viewed as a special case of fuzzy sets with only two (allowed) membership values $\mu_A(x) = 1$ or $\mu_A(x) = 0$.

There are numerous proponents and opponents of the Bayesian and fuzzy characterization of uncertainty. As both the frequentist view and (subjective) Bayesian view of uncertainty can be described by the same axioms of probability, it has lead to the view (common among statisticians) that any type of uncertainty can be fully described by probability. That is, according to Lindley (1987), "probability is the

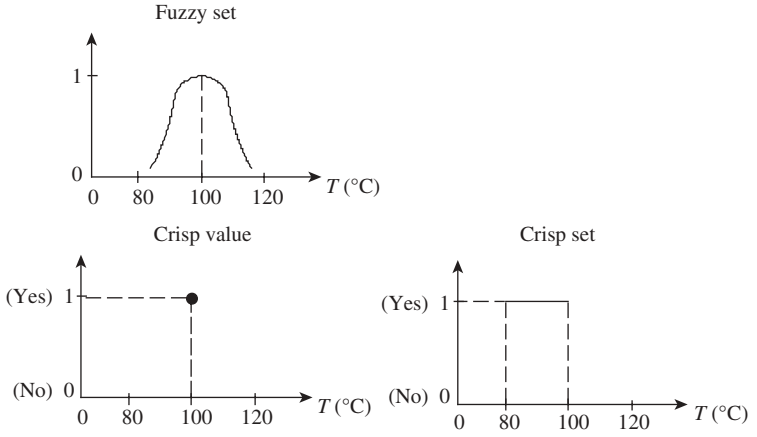

FIGURE 1.4 Fuzzy versus crisp definition of a boiling temperature.

only sensible description of uncertainty and is adequate for all problems involving uncertainty. All other methods are inadequate.'' However, probability describes *randomness*, that is, uncertainty of event occurrence. Fuzziness describes uncertainty related to event *ambiguity*, that is, the subjective degree to which an event occurs. This is an important distinction. Moreover, there are recent claims that probability theory is a special case of fuzzy theory (Kosko 1993).

In the practical context of learning systems, both Bayesian and fuzzy approaches are useful for specification of a priori knowledge about the unknown system. However, both approaches provide *subjective* (i.e., observer-dependent) characterization of uncertainty. Also, there are practical situations where multiple types of uncertainty (frequentist probability, Bayesian probability, and fuzzy) can be combined. For example, a statement ''there is an 80 percent chance of a happy marriage'' describes a (Bayesian) probability of a fuzzy event.

Finally, note that mathematical tools for describing uncertainty (i.e., probability theory and fuzzy logic) have been developed fairly recently, even though humans have dealt with uncertainty for thousands of years. In practice, uncertainty cannot be separated from the notion of *risk* and *risk taking*. In a way, predictive learning methods described in this book can be viewed as a general framework for *risk management*, using empirical models estimated from past data. This view is presented in the last chapter of this book.

1.5 PREDICTIVE LEARNING VERSUS OTHER DATA ANALYTICAL METHODOLOGIES

The growing uses of computers and database technology have resulted in the explosive growth of methods for learning (or estimating) useful models from data. Hence, a number of diverse methodologies have emerged to address this problem. These include approaches developed in classical statistics (multivariate regression/classification, Bayesian methods), engineering (statistical pattern recognition), signal processing, computer science (AI and machine learning), as well as many biologically inspired developments such as artificial neural networks, fuzzy logic, and genetic algorithms. Even though all these approaches often address similar problems, there is little agreement on the fundamental issues involved, and it leads to many heuristic techniques aimed at solving specific applications. In this section, we identify and contrast major methodologies for empirical learning that are often obscured by terminology and minor (technical) details in the implementation of learning algorithms.

At the present time, there are three distinct methodologies for estimating (learning) empirical models from data:

- *Statistical model estimation*, based on extending a classical statistical and function approximation framework (rooted in a density estimation approach) to developing flexible (adaptive) learning algorithms (Ripley 1995; Hastie et al. 2001).

- *Predictive learning*: This approach has originally been developed by practitioners in the field of artificial neural networks in the late 1980s (with no particular theoretical justification). Under this approach, the main focus is on estimating models with good generalization capability, as opposed to estimating ''true'' models under a statistical model estimation methodology. The theoretical framework for predictive learning called Statistical Learning Theory or Vapnik–Chervonenkis (VC) theory (Vapnik 1982) has been relatively unknown until the wide acceptance of its practical methodology called Support Vector Machines (SVMs) in late 1990s (Vapnik 1995). In this book, we use the terms VC theory and predictive learning interchangeably, to denote a methodology for estimating models from data.
- *Data mining*: This is a new practical methodology developed at the intersection of computer science (database technology), information retrieval, and statistics. The goal of data mining is sometimes stated generically as estimating ''useful'' models from data, and this includes, of course, predictive learning and statistical model estimation. However, in a more narrow sense, many data mining algorithms attempt to extract a subset of data samples (from a given large data set) with useful (or interesting) properties. This goal is conceptually similar to *exploratory data analysis* in statistics (Hand 1998; Hand et al. 2001), even though the practical issues are quite different due to huge data size that prevents manual exploration of data (commonly used by statisticians). There seems to be no generally accepted theoretical framework for data mining, so data mining algorithms are initially introduced (by practitioners) and then ''justified'' using formal arguments from statistics, predictive learning, and information retrieval.

There is a significant overlap between these methodologies, and many learning algorithms (developed in one field) have been universally accepted by practitioners in other fields. For example, classification and regression trees (CART) developed in statistics later became very popular in data mining. Likewise, SVMs, originally developed under the predictive learning framework (in VC theory), have been later used (and reformulated) under the statistical estimation framework, and also used in data mining applications. This may give a (misleading) impression that there are only superficial (terminological) differences between these methodologies. In order to understand their differences, we focus on the main assumptions underlying each approach.

Let us relate the three methodologies (statistical model estimation, predictive learning, and data mining) to the general experimental procedure for estimating empirical dependencies from data discussed in Section 1.1. The goal of any data-driven methodology is to estimate (learn) a *useful model* of the unknown system (see Fig. 1.1) from *available data*. We can clearly identify three distinct concepts that help to differentiate between learning methodologies:

1. ''*Useful*'' *model*: There are several commonly used criteria for ''usefulness.'' The first is the prediction accuracy (aka generalization), related to the

capability of the model (obtained using available or training data) to provide accurate estimates (predictions) for future data (from the same statistical population). The second criterion is accurate estimation of the "true" underlying model for data generation, that is, system identification (in Fig. 1.1). Note that correct system identification always implies accurate prediction (but the opposite is not true). The third criterion of the model's "usefulness" relates to its explanatory capabilities; that is, its ability to describe available data in a manner leading to better understanding or interpretation of available data. Note that the goal of obtaining good "descriptive" models is usually quite subjective, whereas the quality of "predictive" models (i.e., generalization) can be objectively evaluated, in principle, using independent (test) data. In the machine learning and neural network literature, predictive methods are also known as "supervised learning" because a predictive model has a unique "response" variable (being predicted by the model). In contrast, descriptive models are referred to as "unsupervised learning" because there is no predefined variable central to the model.

2. *Data set* (used for model estimation): Here we distinguish between the two possibilities. In predictive learning and statistical model estimation, the data set is given explicitly. In data mining, the data set (used for obtaining a useful model) often is not given but must be extracted from a large (given) data set. The term "data mining" suggests that one should search for this data set (with useful properties), which is hidden somewhere in available data.
3. *Formal problem statement* providing (assumed) statistical model for data generation and the goal of estimation (learning). Here we may have two possibilities. That is, when the problem statement is formally well defined and given a priori (i.e., *independent* of the learning algorithm). In predictive learning and statistical model estimation, the goal of learning can be formally stated, that is, there exist mathematical formulations of the learning problem (e.g., see Section 2.1). On the contrary, the field of data mining does not seem to have a single clearly defined formal problem statement because it is mainly concerned with exploratory data analysis.

The existence of the learning problem statement *separate* from the solution approach is critical for meaningful (scientific) comparisons between different learning methodologies. (It is impossible to rigorously compare the performance of methods if each is solving a different problem.) In the case of data mining, the lack of formal problem statement does not suggest that such methods are "inferior" to other approaches. On the contrary, successful applications of data mining to a specific problem may imply that existing learning problem formulations (adopted in predictive learning and statistical model estimation) may not be appropriate for certain data mining applications.

Next, we describe the three methodologies (statistical model estimation, predictive learning, and data mining), in terms of their learning problem statement and solution approaches.

Statistical model estimation is the use of a subset of a population (called a sample) to estimate an underlying statistical model, in order to make conclusions about the entire population (Petrucelli et al. 1999). Classical statistics assumes that the data are generated from some distribution with *known* parametric form, and the goal is to estimate certain properties (of this distribution) useful for specific applications (*problem setting*). Frequently, this goal is stated as density estimation. This goal is achieved by estimating parameters (of unknown distributions) using available data. This goal (probability density estimation) is achieved by maximum-likelihood methods (*solution approach*). The theoretical analysis underlying statistical inference relies heavily on parametric assumptions and asymptotic arguments (i.e., statistically ''optimal'' properties are proved in an asymptotic case when the sample size is large). For example, applying the maximum-likelihood approach to linear regression with normal independent and identically distributed (iid) noise leads to parameter estimation via least squares. In many applications, however, the goal of learning can be stated as obtaining models with good prediction (generalization) capabilities (for future samples). In this case, the approach based on density estimation/function approximation may be suboptimal because it may be possible to obtain good predictive models (reflecting certain properties of the unknown distributions), even when accurate estimation of densities is impossible (due to having only a finite amount of data). Unfortunately, the statistical methodology remains deeply rooted in density estimation/function approximation theoretical framework, which interprets the goal of learning as accurate estimation of the unknown system (in Fig. 1.1), or accurate estimation of the unknown statistical model for data generation, even when application requirements dictate a predictive learning setting. It may be argued that system identification or density estimation is not as prevalent today, because the ''system'' itself is too complex to be identified, and the data are often collected (recorded) automatically for purposes other than system identification. In such real-life applications, often the only meaningful goal is the prediction accuracy for future samples. This may be contrasted to a classical statistical setting where the data are manually collected on a one-time basis, typically under experimental design setting, and the goal is accurate estimation of a given prespecified parametric model.

Predictive learning methodology also has a goal of estimating a useful model using *available training data*. So the problem formulation is often similar to the one used under the statistical model estimation approach. However, the *goal of learning* is explicitly stated as obtaining a model with good prediction (generalization) capabilities for future (test) data. It can be easily shown that estimating a good predictive model is not equivalent to the problem of density estimation (with finite samples). Most practical implementations of predictive learning are based on the idea of obtaining a good predictive model via fitting a set of possible models (given a priori) to available (training) data, aka minimization of empirical risk. This approach has been theoretically described in VC learning theory, which provides general conditions under which various estimators (implementing empirical risk minimization) can generalize well. As noted earlier, VC theory is, in fact, a mathematical theory formally describing the predictive learning methodology.

Historically, many practical predictive learning algorithms (such as neural networks) have been originally introduced by practitioners, but later have been "explained" or "justified" by researchers using statistical model estimation (i.e., density estimation) arguments. Often this leads to certain confusion because such an interpretation creates a (false) impression that the methodology itself (the goal of learning) is based on statistical model estimation. Note that by choosing a simpler but more appropriate problem statement (i.e., estimating relevant properties of unknown distributions under the predictive learning approach), it is possible to make some gains on the inherent stumbling blocks of statistical model estimation (curse of dimensionality, dealing with finite samples, etc.). Bayesian approaches in statistical model estimation can be viewed as an alternative approach to this issue because they try to fix statistical model estimation by including information outside of the data to improve on these stumbling blocks.

Data mining methodology is a diverse field that includes many methods developed under statistical model estimation and predictive learning. There exist two classes of data mining techniques, that is, methods aimed at building "global" models (describing all available data) and "local" models describing some (unspecified) portion of available data (Hand 1998, 1999). According to this taxonomy, "global" data mining methods are (conceptually) identical to methods developed under predictive learning or statistical model estimation. On the contrary, methods for obtaining "local" models aim at discovering "interesting" models for (unspecified) subsets of available data. This is clearly an ill-posed problem, and any meaningful solution will require either (1) exact specification of the portion of the data for which a model is sought or (2) specification of the model that describes the (unknown) subset of available data. Of course, the former leads again to the predictive learning or the statistical model estimation paradigm, and only the latter represents a new learning paradigm. Hence, the data mining paradigm amounts to selecting a portion of data samples (from a given data set) that have certain predefined properties. This paradigm covers a wide range of problems (i.e., data segmentation), and it can also be related to information retrieval, where the "useful" information is specified by its "predefined properties."

This book describes learning (estimation) methods using mainly the predictive learning methodology following concepts developed in VC learning theory. Detailed comparisons between the predictive learning and statistical model estimation paradigms are presented in Sections 3.4.5, 4.5 and 9.9.

2

PROBLEM STATEMENT, CLASSICAL APPROACHES, AND ADAPTIVE LEARNING

All models are wrong, but some are useful.
George Box

Chapter 2 starts with mathematical formulation of the inductive learning problem in Section 2.1. Several important instances of this problem, such as classification, regression, density estimation, and vector quantization, are also presented. An important point is made that with finite samples, it is always better to solve a particular

Learning From Data: Concepts, Theory, and Methods, Second Edition
By Vladimir Cherkassky and Filip Mulier

instance of the learning problem *directly*, rather than trying to solve a more general (and much more difficult) problem of joint (*input*, *output*) density estimation.

Section 2.2 presents an overview and gives representative examples of the classical statistical approaches to estimation (learning) from samples. These include parametric modeling based on the maximum likelihood (ML) and Empirical Risk Minimization (ERM) inductive principles and nonparametric methods for density estimation. It is noted that the classical methods may not be suitable for many applications because parametric modeling (with finite samples) imposes very rigid assumptions about the unknown dependency; that is, it specifies its parametric form. This tends to introduce large modeling bias, namely the discrepancy between the assumed parametric model and the (unknown) truth. Likewise, classical nonparametric methods work only in an asymptotic case (very large sample size), and we never have enough samples to satisfy these asymptotic conditions with high-dimensional data.

The limitations of classical approaches provide motivation for adaptive (or flexible) methods. Section 2.3 provides the philosophical interpretation of learning and defines major concepts and issues necessary for understanding various adaptive methods (presented in later chapters). The formulation for predictive learning (given in Section 2.1) is naturally related to the philosophical notions of induction and deduction. The role of a priori assumptions (i.e., knowledge outside the data) in learning is also examined. Adaptive methods achieve greater flexibility by specifying a wider class of approximating functions (than parametric methods). The predictive model is then selected from this wide class of functions. The main problem becomes choosing the model of optimal complexity (flexibility) for the finite data at hand. Such a choice is usually achieved by introducing constraints (in the form of a priori knowledge) on the selection of functions from this wide class of potential solutions (functions). This brings immediately several concerns:

- How to incorporate a priori assumptions (constraints) into learning?
- How to measure model complexity (i.e., flexibility to fit the training data)?
- How to find an optimal balance between the data and a priori knowledge?

These issues are common to all methods for learning from samples. Even though there are thousands of known methods, there are just a handful of fundamental issues. Frequently, they are hidden in the details of a method. Section 2.3 presents a general framework for dealing with such important issues by introducing distinct concepts such as a priori knowledge, inductive principle (type of inference), and learning methods. Section 2.3 concludes with description of major inductive principles and discussion of their advantages and limitations.

Even though standard inductive learning tasks (described in Section 2.1) are commonly used for many applications, Section 2.3.4 takes a broader view, arguing that an appropriate learning formulation should reflect application-domain requirements, which often leads to ''non-standard'' formulations.

Section 2.4 presents the summary.

2.1 FORMULATION OF THE LEARNING PROBLEM

Learning is the process of estimating an unknown (input, output) dependency or structure of a System using a limited number of observations. The general learning scenario involves three components (Fig. 2.1): a *Generator* of random input vectors, a *System* that returns an output for a given input vector, and the *Learning Machine* that estimates an unknown (input, output) mapping of the System from the observed (input, output) samples. This formulation is very general and describes many practical learning problems found in engineering and statistics, such as interpolation, regression, classification, clustering, and density estimation. Before we look at the learning machine in detail, let us clearly describe the roles of each component in mathematical terms:

Generator: The generator (or sampling distribution) produces random vectors $\mathbf{x} \in \Re^d$ drawn independently from a fixed probability density $p(\mathbf{x})$, which is unknown. In statistical terminology, this situation is called observational. It differs from the designed experiment setting, which involves creating a deterministic sampling scheme optimal for a specific analysis according to experiment design theory. In this book, the observational setting is usually assumed; that is, a modeler (learning machine) has had no control over which input values were supplied to the System.

System: The system produces an output value y for every input vector $\mathbf{x}$ according to the fixed conditional density $p(y|\mathbf{x})$, which is also unknown. Note that this description includes the specific case of a deterministic system, where $y = t(\mathbf{x})$, as well as the regression formulation of $y = t(\mathbf{x}) + \xi$, where ξ is random noise with zero mean. Real systems rarely have truly random outputs; however, they often have unmeasured inputs (Fig. 1.1). Statistically, the effect of these changing unobserved inputs on the output of the System can be characterized as random and represented as a probability distribution.

Learning Machine: In the most general case, the Learning Machine is capable of implementing a set of functions $f(\mathbf{x}, \omega)$, $\omega \in \Omega$, where Ω is a set of abstract

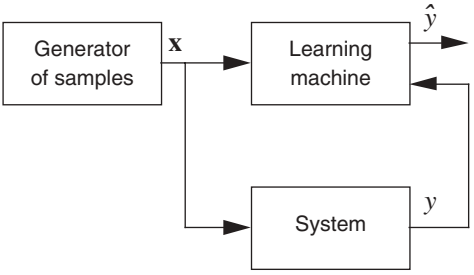

FIGURE 2.1 A Learning Machine using observations of the System to form an approximation of its output.

parameters used only to index the set of functions. In this formulation, the set of functions implemented by the Learning Machine can be any set of functions, chosen a priori, before the formal inference (learning) process has begun. Let us look at some simple examples of Learning Machines and how they fit this formal description. The examples chosen are all solutions to the regression problem, which is only one of the four most common learning tasks (Section 2.1.2). The examples illustrate the notion of a set of functions (of a Learning Machine) and not the mechanism by which the Learning Machine chooses the best approximating function from this set.

Example 2.1: Parametric regression (fixed-degree polynomial)

In this example, the set of functions is specified as a polynomial of fixed degree and the training data have a single predictor variable ($x \in \Re^1$). The set of functions implemented by the Learning Machine is

$$f(x, \mathbf{w}) = \sum_{i=0}^{M-1} w_i x^i, \tag{2.1}$$

where the set of parameters Ω takes the form of vectors $\mathbf{w} = [w_0, \ldots, w_{M-1}]$ of fixed length M.

Example 2.2: Semiparametric regression (polynomial of arbitrary degree)

One way to provide a wider class of functions for the Learning Machine is to remove the restriction of fixed polynomial degree. The degree of the polynomial now becomes another parameter that indexes the set of functions

$$f_m(x, \mathbf{w}_m) = \sum_{i=0}^{m-1} w_i x^i. \tag{2.2}$$

Here the set of parameters Ω takes the form of vectors $\mathbf{w}_m = [w_0, \ldots, w_{m-1}]$, which have an arbitrary length m.

Example 2.3: Nonparametric regression (kernel smoothing)

Additional flexibility can also be achieved by using a nonparametric approach like kernel averaging to define the set of functions supported by the Learning Machine. Here the set of functions is

$$f_\alpha(x, \mathbf{w}_n | \mathbf{x}_n) = \frac{\sum_{i=1}^{n} w_i K_\alpha(x, x_i)}{\sum_{i=1}^{n} K_\alpha(x, x_i)}, \tag{2.3}$$

where n is the number of samples and $K_\alpha(x, x')$ is called the *kernel* function with bandwidth α. For the general case $\mathbf{x} \in \Re^d$, the kernel function $K(\mathbf{x}, \mathbf{x}')$ obeys the following properties:

1. $K(\mathbf{x}, \mathbf{x}')$ takes on its maximum value when $\mathbf{x}' = \mathbf{x}$
2. $|K(\mathbf{x}, \mathbf{x}')|$ decreases with $|\mathbf{x} - \mathbf{x}'|$
3. $K(\mathbf{x}, \mathbf{x}')$ is in general a symmetric function of $2d$ variables

Usually, the kernel function is chosen to be radially symmetric, making it a function of one variable $K(\eta)$, where η is the scaled distance between $\mathbf{x}$ and $\mathbf{x}'$:

$$\eta = \frac{|\mathbf{x} - \mathbf{x}'|}{s(\mathbf{x})}.$$

The scale factor $s(\mathbf{x})$ defines the size (or width) of the region around $\mathbf{x}$ for which K is large. It is common to set the scale factor to a constant value $s(\mathbf{x}) = \alpha$, which is the form of the kernel used in our example equation (2.3). An example of a typical kernel function is the Gaussian

$$K_\alpha(x, x') = \exp\left(-\frac{(x - x')^2}{2\alpha^2}\right). \tag{2.4}$$

In this Learning Machine, the set of parameters Ω takes the form of vectors $[\alpha, w_1, \ldots, w_n]$ of a fixed length that depends on the number of samples n. In this example, it is assumed that the input samples $\mathbf{x}_n = [x_1, \ldots, x_n]$ are used in the specification of the set of approximating functions of the Learning Machine. This is formally stated in (2.3) by having the set of approximating functions conditioned on the given vector of predictor sample values. The previous two examples did not use input samples for specifying the set of functions.

Choice of approximating functions: Ideally, the choice of a set of approximating functions reflects a priori knowledge about the System (unknown dependency). However, in practice, due to complex and often informal nature of a priori knowledge, such specification of approximating functions may be difficult or impossible. Hence, there may be a need to incorporate a priori knowledge into the learning method with an already given set of approximating functions. These issues are discussed in more detail in Section 2.3. There is also an important distinction between two types of approximating functions: linear in parameters or nonlinear in parameters. Throughout this book, learning (estimation) procedures using the former are also referred to as *linear*, whereas those using the latter are called *nonlinear*. We point out that the notion of linearity is with respect to parameters rather than input variables. For example, polynomial regression (2.2) is a linear method. Another example of a linear class of approximating functions (for regression) is the trigonometric expansion

$$f_m(x, \mathbf{v}_m, \mathbf{w}_m) = \sum_{j=1}^{m-1} (v_j \sin(jx) + w_j \cos(jx)) + w_0.$$

On the contrary, multilayer networks of the form

$$f_m(\mathbf{x}, \mathbf{w}, V) = w_0 + \sum_{j=1}^{m} w_j g\left(v_{0j} + \sum_{i=1}^{d} x_i v_{ij}\right)$$

provide an example of nonlinear parameterization because it depends nonlinearly on parameters V via nonlinear basis function g (usually taken as the so-called sigmoid activation function).

The distinction between linear and nonlinear methods is important in practice because learning (estimation) of model parameters amounts to solving a linear or nonlinear optimization problem, respectively.

2.1.1 Objective of Learning

As noted in Section 1.5, there may be two distinct interpretations of the goal of learning for generic system shown in Fig. 2.1. Under statistical model estimation framework, the goal of learning is accurate *identification* of the unknown system, whereas under *predictive learning* the goal is accurate *imitation* (of a system's output). It should be clear that the goal of system identification is more demanding than the goal of system imitation. For instance, accurate system identification does not depend on the distribution of input samples, whereas good predictive model is usually conditional upon this (unknown) distribution. Hence, an accurate model (in the sense of system's identification) would certainly provide good generalization (in the predictive sense), but the opposite may not be true. The mathematical treatment of system identification leads to the function approximation framework and to fundamental problems of estimating multivariate functions known as the curse of dimensionality (see Chapter 3). On the contrary, the goal of predictive learning leads to Vapnik–Chervonenkis (VC) learning theory described later in Chapter 4. This book advocates the setting of predictive learning, which formally defines the notion of accurate system imitation (via minimization of prediction risk) as described in this section. We contrast the function approximation approach versus predictive learning throughout the book, in particular, using empirical comparisons in Section 3.4.5.

The problem encountered by the Learning Machine is to select a function (from the set of functions it supports) that best approximates the System's response. The Learning Machine is limited to observing a finite number (n) of examples in order to make this selection. These training data as produced by the Generator and System will be independent and identically distributed (iid) according to the joint probability density function (pdf)

$$p(\mathbf{x}, y) = p(\mathbf{x})p(y|\mathbf{x}). \tag{2.5}$$

The finite sample (training data) from this distribution is denoted by

$$(\mathbf{x}_i, y_i), \qquad (i = 1, \ldots, n). \tag{2.6}$$

The quality of an approximation produced by the Learning Machine is measured by the loss $L(y,f(\mathbf{x},\omega))$ or discrepancy between the output produced by the System and the Learning Machine for a given input $\mathbf{x}$. By convention, the loss takes on non-negative values, so that large positive values correspond to poor approximation. The expected value of the loss is called the *risk functional*:

$$R(\omega) = \int L(y,f(\mathbf{x},\omega))\, p(\mathbf{x},y)\mathrm{d}\mathbf{x}\mathrm{d}y. \tag{2.7}$$

Learning is the process of estimating the function $f(\mathbf{x},\omega_0)$, which minimizes the risk functional over the set of functions supported by the Learning Machine using only the training data ($p(\mathbf{x},y)$ is not known). With finite data we cannot expect to find $f(\mathbf{x},\omega_0)$ exactly, so we denote $f(\mathbf{x},\omega^*)$ as the estimate of the optimal solution obtained with finite training data using some learning procedure. It is clear that any learning task (regression, classification, etc.) can be solved by minimizing (2.7) if the density $p(\mathbf{x},y)$ is known. This means that density estimation is the most general (and hence most difficult) type of learning problem. The problem of learning (estimation) from finite data alone is inherently ill posed. To obtain a useful (unique) solution, the learning process needs to incorporate a priori knowledge in addition to data. Let us assume that a priori knowledge is reflected in the set of approximating functions of a Learning Machine (as discussed earlier in this section). Then the next issue is: How should a Learning Machine use training data? The answer is given by the concept known as an *inductive principle.* An inductive principle is a general prescription for obtaining an estimate $f(\mathbf{x},\omega^*)$ of the ''true dependency'' in the class of approximating functions from the available (finite) training data. An inductive principle tells us *what* to do with the data, whereas the learning method specifies *how* to obtain an estimate. Hence, a learning method (or algorithm) is a constructive implementation of an inductive principle for selecting an estimate $f(\mathbf{x},\omega^*)$ from a particular set of functions $f(\mathbf{x},\omega)$. For a given inductive principle, there are many learning methods corresponding to a different set of functions of a learning machine. The distinction between inductive principles and learning methods is further discussed in Section 2.3.

2.1.2 Common Learning Tasks

The generic learning problem can be subdivided into four classes of common problems: classification, regression, density estimation, and clustering/vector quantization. For each of these problems, the nature of the loss function and the output (y) differ. However, the goal of minimizing the risk functional based only on training data is common to all learning problems.

Classification

In a (two-class) classification problem, the output of the system takes on only two (symbolic) values $y = \{0,1\}$ corresponding to two classes (as discussed in Section 1.3). Hence, the output of the Learning Machine needs to only take on

two values as well, so the set of functions $f(\mathbf{x}, \omega)$, $\omega \in \Omega$, becomes a set of *indicator* functions. A commonly used loss function for this problem measures the classification error

$$L(y,f(\mathbf{x}, \omega)) = \begin{cases} 0, & \text{if } y = f(\mathbf{x}, \omega), \\ 1, & \text{if } y \neq f(\mathbf{x}, \omega). \end{cases} \tag{2.8}$$

Using this loss function, the risk functional

$$R(\omega) = \int L(y,f(\mathbf{x}, \omega))p(\mathbf{x}, y)\mathrm{d}\mathbf{x}\mathrm{d}y \tag{2.9}$$

quantifies the probability of misclassification. Learning then becomes the problem of estimating the indicator function $f(\mathbf{x}, \omega_0)$ (classifier) that minimizes the probability of misclassification (2.9) using only the training data.

Regression

Regression is the process of estimating a real-valued function based on a finite set of noisy samples. The output of the System in regression problems is a random variable that takes on real values and can be interpreted as the sum of a deterministic function and a random error with zero mean:

$$y = t(\mathbf{x}) + \xi, \tag{2.10}$$

where the deterministic function is the mean of the output conditional probability

$$t(\mathbf{x}) = \int yp(y|\mathbf{x})\mathrm{d}y. \tag{2.11}$$

The set of functions $f(\mathbf{x}, \omega)$, $\omega \in \Omega$, supported by the Learning Machine may or may not contain the regression function (2.11). A common loss function for regression is the squared error

$$L(y,f(\mathbf{x}, \omega)) = (y - f(\mathbf{x}, \omega))^2. \tag{2.12}$$

Learning then becomes the problem of finding the function $f(\mathbf{x}, \omega_0)$ (regressor) that minimizes the risk functional

$$R(\omega) = \int (y - f(\mathbf{x}, \omega))^2 p(\mathbf{x}, y)\mathrm{d}\mathbf{x}\mathrm{d}y \tag{2.13}$$

using only the training data. This risk functional measures the accuracy of the Learning Machine's *predictions* of the System output. Under the assumption that

the noise is zero mean, this risk can also be written in terms of the Learning Machine's accuracy of approximation of the function $t(\mathbf{x})$, as detailed next. The risk is

$$\begin{aligned} R(\omega) &= \int (y - t(\mathbf{x}) + t(\mathbf{x}) - f(\mathbf{x}, \omega))^2 p(\mathbf{x}, y) \mathrm{d}\mathbf{x}\mathrm{d}y \\ &= \int (y - t(\mathbf{x}))^2 p(\mathbf{x}, y) \mathrm{d}\mathbf{x}\mathrm{d}y + \int (f(\mathbf{x}, \omega) - t(\mathbf{x}))^2 p(\mathbf{x}) \mathrm{d}\mathbf{x} \\ &\quad + 2 \int (y - t(\mathbf{x}))(t(\mathbf{x}) - f(\mathbf{x}, \omega)) p(\mathbf{x}, y) \mathrm{d}\mathbf{x}\mathrm{d}y. \end{aligned} \tag{2.14}$$

Assuming that the noise has zero mean, the last summand in (2.14) is

$$\begin{aligned} &\int (y - t(\mathbf{x}))(t(\mathbf{x}) - f(\mathbf{x}, \omega)) p(\mathbf{x}, y) \mathrm{d}\mathbf{x}\mathrm{d}y \\ &\quad = \int \xi (t(\mathbf{x}) - f(\mathbf{x}, \omega)) p(y|\mathbf{x}) p(\mathbf{x}) \mathrm{d}\mathbf{x}\mathrm{d}y \\ &\quad = \int (t(\mathbf{x}) - f(\mathbf{x}, \omega)) \left[\int \xi \, p(y|\mathbf{x}) \mathrm{d}y \right] p(\mathbf{x}) \mathrm{d}\mathbf{x} \\ &\quad = \int (t(\mathbf{x}) - f(\mathbf{x}, \omega)) E_\xi(\xi|\mathbf{x}) p(\mathbf{x}) \mathrm{d}\mathbf{x} = 0. \end{aligned} \tag{2.15}$$

Therefore, the risk can be written as

$$R(\omega) = \int (y - t(\mathbf{x}))^2 p(\mathbf{x}, y) \mathrm{d}\mathbf{x}\mathrm{d}y + \int (f(\mathbf{x}, \omega) - t(\mathbf{x}))^2 p(\mathbf{x}) \mathrm{d}\mathbf{x}. \tag{2.16}$$

The first summand does not depend on the approximating function $f(\mathbf{x}, \omega)$ and can be written in terms of the noise variance

$$\begin{aligned} \int (y - t(\mathbf{x}))^2 p(\mathbf{x}, y) \mathrm{d}\mathbf{x}\mathrm{d}y &= \int \xi^2 p(y|\mathbf{x}) p(\mathbf{x}) \mathrm{d}\mathbf{x}\mathrm{d}y \\ &= \int \left[\int \xi^2 p(y|\mathbf{x}) \mathrm{d}y \right] p(\mathbf{x}) \mathrm{d}\mathbf{x} \\ &= \int E_\xi(\xi^2|x) p(\mathbf{x}) \mathrm{d}\mathbf{x}. \end{aligned} \tag{2.17}$$

Substituting (2.17) into (2.16) gives an equation for the risk

$$R(\omega) = \int E_\xi(\xi^2|x) p(\mathbf{x}) \mathrm{d}\mathbf{x} + \int (f(\mathbf{x}, \omega) - t(\mathbf{x}))^2 p(\mathbf{x}) \mathrm{d}\mathbf{x}. \tag{2.18}$$

Therefore, the risk for the regression problem (assuming L_2 loss and zero mean noise) has a contribution due to the noise variance and a contribution

due to function approximation accuracy. As the noise variance does not depend on ω, minimizing just the second term in (2.18) would be equivalent to minimizing (2.13); that is, the goal of obtaining smallest prediction risk is equivalent to the most accurate estimation of the unknown function $t(\mathbf{x})$ by a Learning Machine.

Density Estimation

For estimating the density of $\mathbf{x}$, the output of the System is not used. The output of the Learning Machine now represents density, so $f(\mathbf{x}, \omega)$, $\omega \in \Omega$, becomes a set of densities. For this problem, the natural criterion is ML, or equivalently, minimization of the negative log-likelihood. Using the loss function

$$L(f(\mathbf{x}, \omega)) = -\ln\ f(\mathbf{x}, \omega) \tag{2.19}$$

in the risk functional (2.7) gives

$$R(\omega) = \int -\ln f(\mathbf{x}, \omega) p(\mathbf{x}) \mathrm{d}\mathbf{x}, \tag{2.20}$$

which is a common risk functional used for density estimation. Minimizing (2.20) using only the training data $\mathbf{x}_1, \ldots, \mathbf{x}_n$ leads to the density estimate $f(\mathbf{x}, \omega_0)$.

Clustering and Vector Quantization

Say, the goal is optimal partitioning of the unknown distribution in $\mathbf{x}$-space into a *prespecified number* of regions (clusters) so that future samples drawn from a particular region can be approximated by a single point (cluster center or local prototype). Here the set of vector-valued functions $\mathbf{f}(\mathbf{x}, \omega)$, $\omega \in \Omega$, are vector quantizers. A vector quantizer provides the mapping

$$\mathbf{x} \xrightarrow{\mathbf{f}(\mathbf{x}, \omega)} \mathbf{c}(\mathbf{x}), \tag{2.21}$$

where $\mathbf{c}(\mathbf{x})$ denotes the cluster center coordinates. In this way, continuous inputs $\mathbf{x}$ are mapped onto a discrete number of centers in $\mathbf{x}$-space. The vector quantizer is completely described by the cluster center coordinates and the partitioning of the input vector space. A common loss function in this case would be the *squared error distortion*

$$L(\mathbf{f}(\mathbf{x}, \omega)) = (\mathbf{x} - \mathbf{f}(\mathbf{x}, \omega)) \cdot (\mathbf{x} - \mathbf{f}(\mathbf{x}, \omega)), \tag{2.22}$$

where $\cdot$ denotes the inner product. Minimizing the risk functional

$$R(\omega) = \int (\mathbf{x} - \mathbf{f}(\mathbf{x}, \omega)) \cdot (\mathbf{x} - \mathbf{f}(\mathbf{x}, \omega)) p(\mathbf{x}) \mathrm{d}\mathbf{x} \tag{2.23}$$

would give an optimal vector quantizer based on the observed data. Note that the vector quantizer minimizing this risk functional is designed to optimally quantize future data generated from a density $p(\mathbf{x})$. In this context, vector quantization is a learning problem. This objective differs from another common objective of optimally quantizing (compressing) a given finite data set. Vector quantization has a goal of *data reduction.* Another important problem (discussed in this book) is *dimensionality reduction.* The problem of dimensionality reduction is that of finding low-dimensional mappings of a high-dimensional distribution. These low-dimensional mappings are often used as features for other learning tasks.

2.1.3 Scope of the Learning Problem Formulation

The mathematical formulation of the learning problem may give the unintended impression that learning algorithms do not require human intervention, but this is clearly not the case. Even though available research literature (and most descriptions in this book) is concerned with formal description of learning methods, there is an equally important *informal part* of any practical learning system. This part involves practical issues such as selection of the input and output variables, data encoding/representation, and incorporating a priori domain knowledge into the design of a learning system. As discussed in Section 1.1, this (informal) part is often more critical for an overall success than the design of a learning machine itself. Indeed, if the wrong (uninformative) input variables are used in modeling, then no learning method can provide an accurate prediction. Thus, one must keep in mind the conceptual range of the formal learning model and the role of the human participant during an informal stage.

There are also many practical situations that do not fit the inductive learning formulation because they violate the assumptions imposed on the generator distribution. Recall that the generator is assumed to produce independently drawn samples from a fixed probability distribution. For example, in the problem of time series prediction, samples are assumed to be generated by a dynamic system, and so they are not independent. This does not make time series prediction a completely different problem. Many of the learning approaches in this book have been used for practical applications of time series prediction with good results. Another assumption that may not hold for practical problems is that of an unchanging generator distribution. One simple practical example that violates this assumption is when designed experiment data are used to train a Learning Machine for predicting future observational data. Another example is the design of a classifier using data that do not reflect future prior probabilities. More complicated issues arise when the Generator distribution is modified by the Learning Machine. This would occur in problems of pedagogical pattern selection (Cachin 1994), where the Learning Machine actively explores the input space. These practical learning problems present open theoretical issues, yet good practical solutions can be achieved using heuristics and clever engineering.

2.2 CLASSICAL APPROACHES

The classical approach, as proposed by Fisher (1952), divides the learning problem into two parts: specification and estimation. *Specification* consists in determining the parametric form of the unknown underlying distributions, whereas *estimation* is the process of determining parameters of these distributions. Classical theory focuses on the problem of estimation and sidesteps the issue of specification.

Classical approaches to the learning problem depend on much stricter assumptions than those posed in the general learning formulation because they assume that functions are specified up to a fixed number of parameters. The two inductive principles that are most commonly used in the classical learning process are Empirical Risk Minimization (ERM) and Maximum Likelihood (ML). ML is a specific form of the more general ERM principle obtained when using particular loss functions. These two inductive principles will be described using the classical solutions for the common learning tasks presented in Section 2.1.2.

2.2.1 Density Estimation

The classical approach for density estimation restricts the class of density functions supported by the learning machine to a parametric set. That is, $p(\mathbf{x};\mathbf{w})$, $\mathbf{w} \in \Omega$, is a set of densities, where $\mathbf{w}$ is an M-dimensional vector (Ω is contained in $\Re^M$, M is fixed). Let us assume that the unknown density $p(\mathbf{x};\mathbf{w}_0)$ belongs to this class. Given a set of iid training data $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_n]$, the probability of seeing this particular data set as a function of $\mathbf{w}$ is

$$P(\mathbf{X}|\mathbf{w}) = \prod_{i=1}^{n} p(\mathbf{x}_i;\mathbf{w}), \tag{2.24}$$

and this is called the *likelihood function.* The ML inductive principle states that we should choose the parameters $\mathbf{w}$ that maximize the likelihood function. This corresponds to choosing a $\mathbf{w}^*$, and therefore the distribution model $p(\mathbf{x};\mathbf{w}^*)$, which is most likely to generate the observed data. To make the problem more tractable, the *log-likelihood function* is maximized. This is equivalent to minimizing the ML risk functional

$$R_{\mathrm{ML}}(\mathbf{w}) = -\sum_{i=1}^{n} \ln p(\mathbf{x}_i;\mathbf{w}). \tag{2.25}$$

On the contrary, using the ERM inductive principle, one empirically estimates the risk function using the training data. The empirical risk is the *average* risk for the training data. This estimate, called the *empirical risk*, is then minimized by choosing the appropriate parameters. For density estimation, the expected risk is given by

$$R(\mathbf{w}) = \int L(p(\mathbf{x};\mathbf{w}))p(\mathbf{x})\mathrm{d}\mathbf{x}.$$

This expectation is estimated by taking an average of the risk over the training data:

$$R_{\text{emp}}(\mathbf{w}) = \frac{1}{n}\sum_{i=1}^{n} L(p(\mathbf{x}_i; \mathbf{w})). \tag{2.26}$$

Then the optimum parameter values $\mathbf{w}^*$ are found by minimizing the empirical risk (2.26) with respect to $\mathbf{w}$. Notice that ERM is a more general inductive principle than the ML principle because it does not specify the particular form of the loss function. If the loss function is

$$L(p(\mathbf{x}; \mathbf{w})) = -\ln p(\mathbf{x}; \mathbf{w}), \tag{2.27}$$

then the ERM inductive principle is equivalent to the ML inductive principle for density estimation. Let us now look at two examples of classical density estimation.

Example 2.4: Estimating the parameters of the normal distribution using finite data

We have observed n samples of x, denoted by $x_1, \ldots, x_n$, that were generated according to the normal distribution

$$p(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{ -\frac{(x-\mu)^2}{2\sigma^2} \right\}, \tag{2.28}$$

where the mean μ and variance σ^2 are the two unknown parameters. The log-likelihood function for this problem is

$$P(\mathbf{X}|\mu, \sigma^2) = -\frac{1}{2} n \ln(2\pi) - n \ln(\sigma) - \frac{1}{2\sigma^2} \sum_{i=1}^{n} (x_i - \mu)^2. \tag{2.29}$$

This can be maximized by taking partial derivatives, leading to the estimates

$$\begin{aligned} \hat{\mu} &= \frac{1}{n}\sum_{i=1}^{n} x_i, \\ \hat{\sigma}^2 &= \frac{1}{n}\sum_{i=1}^{n} (x_i - \hat{\mu})^2. \end{aligned} \tag{2.30}$$

Example 2.5: Mixture of normals (Vapnik 1995)

Now, let us perform the estimation for a more complicated density. Let n samples of x, denoted by $x_1, \ldots, x_n$, be generated according to the distribution

$$p(x) = \frac{1}{2\sqrt{2\pi\sigma^2}} \exp\left\{ -\frac{(x-\mu)^2}{2\sigma^2} \right\} + \frac{1}{2\sqrt{2\pi}} \exp\left\{ -\frac{x^2}{2} \right\}. \tag{2.31}$$

In this case, only the parameters μ and σ^2 of the first density are unknown. The log-likelihood function for this problem is

$$P(\mathbf{X}|\mu,\sigma^2) = \sum_{i=1}^{n} \ln\left(\frac{1}{2\sqrt{2\pi\sigma^2}}\exp\left\{-\frac{(x_i-\mu)^2}{2\sigma^2}\right\} + \frac{1}{2\sqrt{2\pi}}\exp\left\{-\frac{x_i^2}{2}\right\}\right). \quad (2.32)$$

The ML inductive principle tells us that we should find values of μ and σ^2 that maximize (2.32). We can show that for certain values of μ and σ^2 there does not exist a global maximum, indicating that the ML procedure fails to provide a definite solution. Specifically, if μ is set to the value of any training data point, then there is no value of σ^2 that gives a global maximum. Let us attempt to evaluate the likelihood for the choice $\mu = x_1$:

$$\begin{aligned} P(\mathbf{X}|\mu = x_1,\sigma^2) = {} & \ln\left(\frac{1}{2\sqrt{2\pi\sigma^2}} + \frac{1}{2\sqrt{2\pi}}\exp\left\{-\frac{x_1^2}{2}\right\}\right) \\ & + \sum_{i=2}^{n} \ln\left(\frac{1}{2\sqrt{2\pi\sigma^2}}\exp\left\{-\frac{(x_i-x_1)^2}{2\sigma^2}\right\} + \frac{1}{2\sqrt{2\pi}}\exp\left\{-\frac{x_i^2}{2}\right\}\right). \end{aligned} \quad (2.33)$$

Because we would like to maximize this quantity, we consider a lower bound by assuming that some of the terms take on their minimum values:

$$\begin{aligned} P(\mathbf{X}|\mu = x_1,\sigma^2) &> \ln\left(\frac{1}{2\sqrt{2\pi\sigma^2}} + 0\right) + \sum_{i=2}^{n} \ln\left(0 + \frac{1}{2\sqrt{2\pi}}\exp\left\{-\frac{x_i^2}{2}\right\}\right), \\ P(\mathbf{X}|\mu = x_1,\sigma^2) &> -\ln\sigma - \sum_{i=2}^{n}\frac{x_i^2}{2} - n\ln(2\sqrt{2\pi}). \end{aligned} \quad (2.34)$$

The lower bound of the likelihood continues to increase for decreasing σ, which means that a global maximum does not exist. Note that this argument applies for choosing μ equal to any of the training data points x_i. This example shows how the ML inductive principle can fail to provide a solution for estimation of fairly simple densities (mixture of Gaussians).

2.2.2 Classification

The classical classification problem is a special case of the general classification problem, introduced in Section 2.1.2, based on the following restricted learning model: The conditional densities for each class $p(\mathbf{x}|y=0)$ and $p(\mathbf{x}|y=1)$ are estimated via classical (parametric) density estimation and the ML inductive principle. These estimates will be denoted as $p_0(\mathbf{x},\alpha^*)$ and $p_1(\mathbf{x},\beta^*)$, respectively, to indicate that they are parametric functions with parameters chosen via ML. The probability

of occurrence of each class, called *prior* probabilities, $P(y = 0)$ and $P(y = 1)$, is assumed to be known or estimated, namely as a fraction of samples from a particular class in the training set. Using Bayes theorem, it is possible with these quantities to determine for a given observation $\mathbf{x}$ the probability of that observation belonging to each class. These probabilities, called *posterior* probabilities, can be used to construct a discriminant rule that describes how an observation $\mathbf{x}$ should be classified so as to minimize the probability of error. This rule chooses the output class that has the maximum posterior probability. First, Bayes rule is used to calculate the posterior probabilities for each class:

$$\begin{aligned} P(y = 0|\mathbf{x}) &= \frac{p_0(\mathbf{x}, \alpha^*)P(y = 0)}{p(\mathbf{x})}, \\ P(y = 1|\mathbf{x}) &= \frac{p_1(\mathbf{x}, \beta^*)P(y = 1)}{p(\mathbf{x})}. \end{aligned} \tag{2.35}$$

The denominator of these equations is a normalizing constant, which can be expressed in terms of the prior probabilities and class conditional densities as

$$p(\mathbf{x}) = p_0(\mathbf{x}, \alpha^*)P(y = 0) + p_1(\mathbf{x}, \beta^*)P(y = 1). \tag{2.36}$$

Note that there is usually no need to compute this normalizing constant because the decision rule is a comparison of the relative magnitudes of the posterior probabilities. Once the posterior probabilities are determined, the following decision rule is used to classify $\mathbf{x}$:

$$f(\mathbf{x}) = \begin{cases} 0, & \text{if } p_0(\mathbf{x}, \alpha^*)P(y = 0) > p_1(\mathbf{x}, \beta^*)P(y = 1), \\ 1, & \text{otherwise.} \end{cases} \tag{2.37}$$

Equivalently, the rule can be written as

$$f(\mathbf{x}) = I\left\{ \ln p_1(\mathbf{x}, \beta^*) - \ln p_0(\mathbf{x}, \alpha^*) + \ln \frac{P(y = 1)}{P(y = 0)} > 0 \right\}, \tag{2.38}$$

where $I(\ )$ is the indicator function that takes the value 1 if its argument is true and 0 otherwise. Note that in the above expressions, the class labels are denoted by $\{0, 1\}$. Sometimes, for notational convenience, the class labels $\{-1, +1\}$ are used. In order to determine this rule using the classical approach for classification, the conditional class densities need to be estimated. This approach corresponds to determining the parameters α^* and β^* using the ML or ERM inductive principles. Therefore, we apply the ERM inductive principle *indirectly* to first estimate the densities and then use them to formulate the decision rule. This differs from applying the ERM inductive principle *directly* to minimize the empirical risk

$$R_{\text{emp}}(\mathbf{w}) = \frac{1}{n} \sum_{i=1}^{n} I(y_i = f(\mathbf{x}_i, \mathbf{w})) \tag{2.39}$$

by estimating the expected risk functional for classification (2.9) using average of the empirical risk (2.39).

2.2.3 Regression

In the classical formulation of the regression problem, we seek to estimate a vector of parameters of an unknown function $f(\mathbf{x}, \mathbf{w}_0)$ by making measurements of the function with error at any point $\mathbf{x}_k$:

$$y_k = f(\mathbf{x}_k, \mathbf{w}_0) + \xi_k, \tag{2.40}$$

where the error is independent of $\mathbf{x}$ and is distributed according to a known density $p_\xi(\xi)$. Based on the observation of data $\mathcal{Z} = \{(\mathbf{x}_i, y_i),\ i = 1, \ldots, n\}$, the likelihood is given by

$$P(\mathcal{Z}|\mathbf{w}) = \sum_{i=1}^{n} \ln\, p_\xi(y_i - f(\mathbf{x}_i, \mathbf{w})). \tag{2.41}$$

Assuming that the error is normally distributed with zero mean and fixed variance σ, the likelihood is given by

$$P(\mathcal{Z}|\mathbf{w}) = -\frac{1}{2\sigma^2} \sum_{i=1}^{n} (y_i - f(\mathbf{x}_i, \mathbf{w}))^2 - n \ln (\sqrt{2\pi}\sigma). \tag{2.42}$$

Maximizing the likelihood in this form (2.42) is equivalent to minimizing the functional

$$R_{\text{emp}}(\mathbf{w}) = \frac{1}{n} \sum_{i=1}^{n} (y_i - f(\mathbf{x}_i, \mathbf{w}))^2, \tag{2.43}$$

which is in fact the risk functional obtained by using the ERM inductive principle for the squared loss function.

Note that the squared loss function is, strictly speaking, appropriate only for Gaussian noise. However, it is often used in practical applications where the noise is not Gaussian.

2.2.4 Solving Problems with Finite Data

When solving a problem based on finite information, one should keep in mind the following general commonsense principle: *Do not attempt to solve a specified problem by indirectly solving a harder general problem as an intermediate step.* In Section 2.1.1, we saw that density estimation is the universal solution to the learning problem. This means that once the density is known (or accurately estimated), all specific learning tasks can be solved using that density. However, being the most

general learning problem, density estimation requires a larger number of samples than a problem-specific formulation (i.e., regression, classification). As we are ultimately interested in solving a specific task, we should solve it directly. Conceptually, this means that instead of estimating the joint pdf (2.5) fully, we should only estimate those features of the density that are critical for solving our particular problem. Posing the problem directly will then require fewer observations for the specified level of solution accuracy. The following is an example with finite samples that shows how better results can be achieved by solving a simpler more direct problem.

Example 2.6: Discriminant analysis

We wish to build a two-class classifier from data, where it is known that the data are generated according to the multivariate normal probability distributions $N(\mu_0, \sum_0)$ and $N(\mu_1, \sum_1)$. In the classical procedure, the parameters of the densities $\mu_0, \mu_1, \sum_0$, and $\sum_1$ are estimated using the ML based on the training data. The densities are then used to construct a decision rule. For two *known* multivariate normal distributions, the optimal decision rule is a polynomial of degree 2 (Fukunaga 1990):

$$f(\mathbf{x}) = I\left\{\tfrac{1}{2}(\mathbf{x}-\mu_0)^{\mathrm{T}}\Sigma_0^{-1}(\mathbf{x}-\mu_0) - \tfrac{1}{2}(\mathbf{x}-\mu_1)^{\mathrm{T}}\Sigma_1^{-1}(\mathbf{x}-\mu_1) + c > 0\right\}, \quad (2.44)$$

where

$$c = \ln\frac{\det(\sum_0)}{\det(\sum_1)} - \ln\frac{P(y=0)}{P(y=1)}. \quad (2.45)$$

The boundary of this decision rule is a paraboloid. To produce a good decision rule, we must estimate the two $d \times d$ covariance matrices accurately because it is their inverses that are used in the decision rule. In practical problems, there are often not enough data to provide accurate estimates, and this leads to a poor decision rule. One solution to this problem is to impose the following artificial constraint: $\sum_0 = \sum_1 = \sum$, which leads to the linear decision rule

$$f(\mathbf{x}) = I\left\{(\mu_0-\mu_1)^{\mathrm{T}}\Sigma^{-1}\mathbf{x} + \frac{1}{2}\left(\mu_1^{\mathrm{T}}\Sigma^{-1}\mu_1\right) - \frac{1}{2}\left(\mu_0^{\mathrm{T}}\Sigma^{-1}\mu_0\right) - \ln\frac{P(y=0)}{P(y=1)} > 0\right\}. \quad (2.46)$$

This decision rule requires estimation of two means μ_0 and μ_1 and only one covariance matrix $\sum$. In practice, the simpler linear decision rule often performs better than the quadratic decision rule, even when it is known that $\sum_0 \neq \sum_1$. To

demonstrate this phenomenon, consider 20 data samples (10 per class) generated according to the following two class distributions:

$$\text{Class 0} \qquad \text{Class 1}$$
$$N\left([0,0], \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}\right) \qquad N\left([2,0], \begin{bmatrix} 1 & 0.5 \\ 0.5 & 1 \end{bmatrix}\right)$$

Assume that it is known that class densities are Gaussian, but that the means and covariance matrices are unknown. These data will be separated using both the quadratic decision rule and the linear decision rule. Note that the linear decision rule, which assumes equal covariances, does not match the underlying class distributions. However, the first-order model provides the lowest classification error (Fig. 2.2).

2.2.5 Nonparametric Methods

The development of nonparametric methods was an attempt to deal with the main shortcoming of classical techniques: that of having to specify the parametric form of the unknown distributions and dependencies. Nonparametric techniques require few assumptions for developing estimates; however, this is at the expense of requiring a large number of samples. First, nonparametric methods for density estimation are developed. From these, nonparametric regression and classification approaches can be constructed.

Nonparametric Density Estimation

The most commonly used nonparametric estimator of density is the histogram. The histogram is obtained by dividing the sample space into bins of constant width and determining the number of samples that fall into each bin (Fig. 2.3). One of the drawbacks of this approach is that the resulting density is discontinuous. A more sophisticated approach is to use a sliding window kernel function to bin the data, which results in a smooth estimate.

The general principle behind nonparametric density estimation is that of solving the integral equation defining the density:

$$\int_{-\infty}^{x} p(u)\mathrm{d}u = F(x), \tag{2.47}$$

where $F(x)$ is the cumulative distribution function (cdf). As the cdf is unknown, the right-hand side of (2.47) is approximated by the empirical cdf estimated from the training data:

$$F_n(x) = \sum_{i=1}^{n} I(x \geq x_i), \tag{2.48}$$