Section 2.3

and then adding a small level of random noise (governed by a Gaussian distribution) to each such point to obtain the corresponding target value t_n. By generating data in this way, we are capturing an important property of many real-world data sets, namely that they possess an underlying regularity, which we wish to learn, but that individual observations are corrupted by random noise. This noise might arise from intrinsically *stochastic* (i.e., random) processes such as radioactive decay but more typically is due to there being sources of variability that are themselves unobserved.

In this tutorial example we know the true process that generated the data, namely the sinusoidal function. In a practical application of machine learning, our goal is to discover the underlying trends in the data given the finite training set. Knowing the process that generated the data, however, allows us to illustrate important concepts in machine learning.

1.2.2 Linear models

Our goal is to exploit this training set to predict the value $\widehat{t}$ of the target variable for some new value $\widehat{x}$ of the input variable. As we will see later, this involves implicitly trying to discover the underlying function $\sin(2\pi x)$. This is intrinsically a difficult problem as we have to generalize from a finite data set to an entire function. Furthermore, the observed data is corrupted with noise, and so for a given $\widehat{x}$ there is uncertainty as to the appropriate value for $\widehat{t}$. *Probability theory* provides a framework for expressing such uncertainty in a precise and quantitative manner, whereas *decision theory* allows us to exploit this probabilistic representation to make predictions that are optimal according to appropriate criteria. Learning probabilities from data lies at the heart of machine learning and will be explored in great detail in this book.

Chapter 2

Chapter 5

To start with, however, we will proceed rather informally and consider a simple approach based on curve fitting. In particular, we will fit the data using a polynomial function of the form

$$y(x, \mathbf{w}) = w_0 + w_1 x + w_2 x^2 + \ldots + w_M x^M = \sum_{j=0}^{M} w_j x^j \tag{1.1}$$

where M is the *order* of the polynomial, and x^j denotes x raised to the power of j. The polynomial coefficients $w_0, \ldots, w_M$ are collectively denoted by the vector $\mathbf{w}$. Note that, although the polynomial function $y(x, \mathbf{w})$ is a nonlinear function of x, it is a linear function of the coefficients $\mathbf{w}$. Functions, such as this polynomial, that are linear in the unknown parameters have important properties, as well as significant limitations, and are called *linear models*.

Chapter 4

1.2.3 Error function

The values of the coefficients will be determined by fitting the polynomial to the training data. This can be done by minimizing an *error function* that measures the misfit between the function $y(x, \mathbf{w})$, for any given value of $\mathbf{w}$, and the training set data points. One simple choice of error function, which is widely used, is the sum of