the density function, which might be very different from the true density. Usually, however, parametric models allow the density function to be evaluated very rapidly for new values of the input vector. Non-parametric methods, by contrast, allow very general forms of density function, but suffer from the fact that the number of variables in the model grows directly with the number of training data points. This leads to models which can be very slow to evaluate for new input vectors.

In order to combine the advantages of both parametric and non-parametric methods we need to find techniques which are not restricted to specific functional forms, and yet where the size of the model only grows with the complexity of the problem being solved, and not simply with the size of the data set. This leads us to a class of models which we shall call *semi-parametric*. The price we have to pay is that the process of setting up the model using the data set (i.e. the training of the model) is computationally intensive compared to the simple procedures needed for parametric or non-parametric methods (which in some cases involve little more than evaluating a few expressions for parameter values, or even just storing the training data).

In this section we shall restrict attention to one particular form of density function, called a *mixture model*. As well as providing powerful techniques for density estimation, mixture models find important applications in the context of neural networks, for example in configuring the basis functions in radial basis function networks (Section 5.9), in techniques for conditional density estimation (Section 6.4), in the technique of soft weight sharing (Section 9.4), and in the mixture-of-experts model (Section 9.7). Here we discuss three training methods for mixture models, all of which are based on maximum likelihood, involving respectively non-linear optimization, re-estimation (leading to the EM algorithm) and stochastic sequential estimation.

In the non-parametric kernel-based approach to density estimation, the density function was represented as a linear superposition of kernel functions, with one kernel centred on each data point. Here we consider models in which the density function is again formed from a linear combination of basis functions, but where the number M of basis functions is treated as a parameter of the model and is typically much less than the number N of data points. We therefore write our model for the density as a linear combination of component densities $p(\mathbf{x}|j)$ in the form

$$p(\mathbf{x}) = \sum_{j=1}^{M} p(\mathbf{x}|j)P(j). \tag{2.71}$$

Such a representation is called a *mixture distribution* (Titterington *et al.*, 1985; McLachlan and Basford, 1988) and the coefficients $P(j)$ are called the *mixing parameters*. Notice that there is a strong similarity between (2.71) and the expression given in equation (1.22) for the unconditional density of data taken from a mixture of several classes. This similarity has been emphasized by our choice of notation. We shall call $P(j)$ the *prior* probability of the data point having been generated from component j of the mixture. These priors are chosen to satisfy the constraints

$$\sum_{j=1}^{M} P(j) = 1 \tag{2.72}$$

$$0 \leq P(j) \leq 1. \tag{2.73}$$

Similarly, the component density functions $p(\mathbf{x}|j)$ are normalized so that

$$\int p(\mathbf{x}|j)\,d\mathbf{x} = 1 \tag{2.74}$$

and hence can be regarded as class-conditional densities. To generate a data point from the probability distribution (2.71), one of the components j is first selected at random with probability $P(j)$, and then a data point is generated from the corresponding component density $p(\mathbf{x}|j)$. An important property of such mixture models is that, for many choices of component density function, they can approximate any continuous density to arbitrary accuracy provided the model has a sufficiently large number of components, and provided the parameters of the model are chosen correctly.

The key difference between the mixture model representation and a true classification problem lies in the nature of the training data, since in this case we are not provided with any 'class labels' to say which component was responsible for generating each data point. This represents an example of *incomplete data*, and we shall discuss this problem at greater length when we consider the EM algorithm in Section 2.6.2. As with any of the other density estimation techniques discussed in this chapter, the technique of mixture modelling can be applied separately to each class C_k in a true classification problem. In this case, each class-conditional density $p(\mathbf{x}|C_k)$ is represented by an independent mixture model of the form (2.71).

Having made the link with prior probabilities and conditional densities, we can introduce the corresponding *posterior* probabilities, which we can express using Bayes' theorem in the form

$$P(j|\mathbf{x}) = \frac{p(\mathbf{x}|j)P(j)}{p(\mathbf{x})} \tag{2.75}$$

where $p(\mathbf{x})$ is given by (2.71). These posterior probabilities satisfy

$$\sum_{j=1}^{M} P(j|\mathbf{x}) = 1. \tag{2.76}$$