1

Introduction to statistical pattern recognition

Statistical pattern recognition is a term used to cover all stages of an investigation from problem formulation and data collection through to discrimination and classification, assessment of results and interpretation. Some of the basic concepts in classification are introduced and the key issues described. Two complementary approaches to discrimination are presented, namely a decision theory approach based on calculation of probability density functions and the use of Bayes theorem, and a discriminant function approach.

1.1 Statistical pattern recognition

1.1.1 Introduction

We live in a world where massive amounts of data are collected and recorded on nearly every aspect of human endeavour: for example, banking, purchasing (credit-card usage, point-of-sale data analysis), Internet transactions, performance monitoring (of schools, hospitals, equipment), and communications. The data come in a wide variety of diverse forms – numeric, textual (structured or unstructured), audio and video signals. Understanding and making sense of this vast and diverse collection of data (identifying patterns, trends, anomalies, providing summaries) requires some automated procedure to assist the analyst with this ‘data deluge’. A practical example of pattern recognition that is familiar to many people is classifying email messages (as spam/not spam) based upon message header, content and sender.

Approaches for analysing such data include those for signal processing, filtering, data summarisation, dimension reduction, variable selection, regression and classification and have been developed in several literatures (physics, mathematics, statistics, engineering, artificial intelligence, computer science and the social sciences, among others). The main focus of this book is on pattern recognition procedures, providing a description of basic techniques

Statistical Pattern Recognition, Third Edition. Andrew R. Webb and Keith D. Copsey.

together with case studies of practical applications of the techniques on real-world problems. A strong emphasis is placed on the statistical theory of discrimination, but clustering also receives some attention. Thus, the main subject matter of this book can be summed up in a single word: ‘classification’, both supervised (using class information to design a classifier – i.e. discrimination) and unsupervised (allocating to groups without class information – i.e. clustering). However, in recent years many complex datasets have been gathered (for example, ‘transactions’ between individuals – email traffic, purchases). Understanding these datasets requires additional tools in the pattern recognition toolbox. Therefore, we also examine developments such as methods for analysing data that may be represented as a graph.

Pattern recognition as a field of study developed significantly in the 1960s. It was very much an interdisciplinary subject. Some people entered the field with a real problem to solve. The large number of applications ranging from the classical ones such as automatic character recognition and medical diagnosis to the more recent ones in *data mining* (such as credit scoring, consumer sales analysis and credit card transaction analysis) have attracted considerable research effort with many methods developed and advances made. Other researchers were motivated by the development of machines with ‘brain-like’ performance, that in some way could operate giving human performance.

Within these areas significant progress has been made, particularly where the domain overlaps with probability and statistics, and in recent years there have been many exciting new developments, both in methodology and applications. These build on the solid foundations of earlier research and take advantage of increased computational resources readily available nowadays. These developments include, for example, kernel-based methods (including support vector machines) and Bayesian computational methods.

The topics in this book could easily have been described under the term *machine learning* that describes the study of machines that can adapt to their environment and learn from example. The machine learning emphasis is perhaps more on computationally intensive methods and less on a statistical approach, but there is strong overlap between the research areas of statistical pattern recognition and machine learning.

1.1.2 The basic model

Since many of the techniques we shall describe have been developed over a range of diverse disciplines, there is naturally a variety of sometimes contradictory terminology. We shall use the term ‘pattern’ to denote the p-dimensional data vector $\boldsymbol{x} = (x_1, \dots, x_p)^T$ of measurements (^{T} denotes vector transpose), whose components x_i are measurements of the features of an object. Thus the features are the variables specified by the investigator and thought to be important for classification. In discrimination, we assume that there exist C groups or *classes*, denoted $\omega_1, \dots, \omega_C$ and associated with each pattern $\boldsymbol{x}$ is a categorical variable z that denotes the class or group membership; that is, if $z = i$, then the pattern belongs to $\omega_i, i \in \{1, \dots, C\}$.

Examples of patterns are measurements of an acoustic waveform in a speech recognition problem; measurements on a patient made in order to identify a disease (diagnosis); measurements on patients (perhaps subjective assessments) in order to predict the likely outcome (prognosis); measurements on weather variables (for forecasting or prediction); sets of financial measurements recorded over time; and a digitised image for character recognition. Therefore, we see that the term ‘pattern’, in its technical meaning, does not necessarily refer to structure within images.

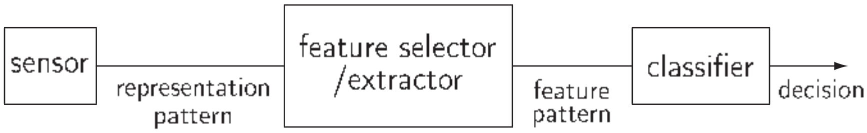

Figure 1.1 Pattern classifier.

The main topic in this book may be described by a number of terms including *pattern classifier design* or *discrimination* or *allocation rule design*. Designing the rule requires specification of the parameters of a pattern classifier, represented schematically in Figure 1.1, so that it yields the optimal (in some sense) response for a given input pattern. This response is usually an estimate of the class to which the pattern belongs. We assume that we have a set of patterns of known class $\{(\boldsymbol{x}_i, z_i), i = 1, \ldots, n\}$ (the *training* or *design* set) that we use to design the classifier (to set up its internal parameters). Once this has been done, we may estimate class membership for a pattern $\boldsymbol{x}$ for which the class label is unknown. Learning the model from a training set is the process of *induction*; applying the trained model to patterns of unknown class is the process of *deduction*.

Thus, the uses of a pattern classifier are to provide:

- A descriptive model that explains the difference between patterns of different classes in terms of features and their measurements.
- A predictive model that predicts the class of an unlabelled pattern.

However, we might ask why do we need a predictive model? Cannot the procedure that was used to assign labels to the training set measurements also be used for the test set in classifier operation? There may be several reasons for developing an automated process:

- to remove humans from the recognition process – to make the process more reliable;
- in banking, to identify good risk applicants before making a loan;
- to make a medical diagnosis without a post mortem (or to assess the state of a piece of equipment without dismantling it) – sometimes a pattern may only be labelled through intensive examination of a subject, whether person or piece of equipment;
- to reduce cost and improve speed – gathering and labelling data can be a costly and time consuming process;
- to operate in hostile environments – the operating conditions may be dangerous or harmful to humans and the training data have been gathered under controlled conditions;
- to operate remotely – to classify crops and land use remotely without labour-intensive, time consuming, surveys.

There are many classifiers that can be constructed from a given dataset. Examples include decision trees, neural networks, support vector machines and linear discriminant functions. For a classifier of a given type, we employ a learning algorithm to search through the parameter space to find the model that best describes the relationship between the measurements and class labels for the training set. The form derived for the pattern classifier depends on a number of different factors. It depends on the distribution of the training data, and the assumptions made concerning its distribution. Another important factor is the misclassification cost – the cost of making an incorrect decision. In many applications misclassification costs are hard to quantify, being combinations of several contributions such as monetary costs, time and other more subjective costs. For example, in a medical diagnosis problem, each treatment has different costs associated with it. These relate to the expense of different types of drugs, the suffering the patient is subjected to by each course of action and the risk of further complications.

Figure 1.1 grossly oversimplifies the pattern classification procedure. Data may undergo several separate transformation stages before a final outcome is reached. These transformations (sometimes termed preprocessing, feature selection or feature extraction) operate on the data in a way that, usually, reduces its dimension (reduces the number of features), removing redundant or irrelevant information, and transforms it to a form more appropriate for subsequent classification. The term *intrinsic dimensionality* refers to the minimum number of variables required to capture the structure within the data. In speech recognition, a preprocessing stage may be to transform the waveform to a frequency representation. This may be processed further to find formants (peaks in the spectrum). This is a *feature extraction* process (taking a possibly nonlinear combination of the original variables to form new variables). *Feature selection* is the process of selecting a subset of a given set of variables (see Chapter 10). In some problems, there is no automatic feature selection stage, with the feature selection being performed by the investigator who ‘knows’ (through experience, knowledge of previous studies and the problem domain) those variables that are important for classification. In many cases, however, it will be necessary to perform one or more transformations of the measured data.

In some pattern classifiers, each of the above stages may be present and identifiable as separate operations, while in others they may not be. Also, in some classifiers, the preliminary stages will tend to be problem specific, as in the speech example. In this book, we consider feature selection and extraction transformations that are not application specific. That is not to say the methods of feature transformation described will be suitable for any given application, however, but application-specific preprocessing must be left to the investigator who understands the application domain and method of data collection.

1.2 Stages in a pattern recognition problem

A pattern recognition investigation may consist of several stages enumerated below. Not all stages may be present; some may be merged together so that the distinction between two operations may not be clear, even if both are carried out; there may be some application-specific data processing that may not be regarded as one of the stages listed below. However, the points below are fairly typical.

1. Formulation of the problem: gaining a clear understanding of the aims of the investigation and planning the remaining stages.
2. Data collection: making measurements on appropriate variables and recording details of the data collection procedure (ground truth).

3. Initial examination of the data: checking the data, calculating summary statistics and producing plots in order to get a feel for the structure.

4. Feature selection or feature extraction: selecting variables from the measured set that are appropriate for the task. These new variables may be obtained by a linear or nonlinear transformation of the original set (feature extraction). To some extent, the partitioning of the data processing into separate feature extraction and classification processes is artificial, since a classifier often includes the optimisation of a feature extraction stage as part of its design.

5. Unsupervised pattern classification or clustering. This may be viewed as exploratory data analysis and it may provide a successful conclusion to a study. On the other hand, it may be a means of preprocessing the data for a supervised classification procedure.

6. Apply discrimination or regression procedures as appropriate. The classifier is designed using a training set of exemplar patterns.

7. Assessment of results. This may involve applying the trained classifier to an independent *test set* of labelled patterns. Classification performance is often summarised in the form of a confusion matrix:

		True class		
		ω_1	ω_2	ω_3
Predicted class	ω_1	e_{11}	e_{12}	e_{13}
	ω_2	e_{21}	e_{22}	e_{23}
	ω_3	e_{31}	e_{32}	e_{33}

   where e_{ij} is the number of patterns of class ω_j that are predicted to be class ω_i. The accuracy, a, is calculated from the confusion matrix as

   $$a = \frac{\sum_{i} e_{ii}}{\sum_{ij} e_{ij}}$$

   and the error rate is $1 - a$.

8. Interpretation.

The above is necessarily an iterative process: the analysis of the results may generate new hypotheses that require further data collection. The cycle may be terminated at different stages: the questions posed may be answered by an initial examination of the data or it may be discovered that the data cannot answer the initial question and the problem must be reformulated.

The emphasis of this book is on techniques for performing the steps 4, 5, 6 and 7.

1.3 Issues

The main topic that we address in this book concerns classifier design: given a training set of patterns of known class, we seek to use those examples to design a classifier that is optimal for the expected operating conditions (the test conditions).

There are a number of very important points to make about this design process.

Finite design set

We are given a *finite* design set. If the classifier is too complex (there are too many free parameters) it may model noise in the design set. This is an example of *overfitting*. If the classifier is not complex enough, then it may fail to capture structure in the data. An illustration of this is the fitting of a set of data points by a polynomial curve (Figure 1.2). If the degree of the polynomial is too high then, although the curve may pass through or close to the data points thus achieving a low fitting error, the fitting curve is very variable and models every fluctuation in the data (due to noise). If the degree of the polynomial is too low, the fitting error is large and the underlying variability of the curve is not modelled (the model *underfits* the data). Thus, achieving optimal performance on the design set (in terms of minimising some error criterion perhaps) is not required: it may be possible, in a classification problem, to achieve 100% classification accuracy on the design set but the *generalisation performance* – the expected performance on data representative of the true operating conditions (equivalently the performance on an infinite test set of which the design set is a sample) – is poorer than could be achieved by careful design. Choosing the 'right' model is an exercise in *model selection*. In practice we usually do not know what is structure and what is noise in the data. Also, training a classifier (the procedure of determining its parameters) should not be considered as a separate issue from model selection, but it often is.

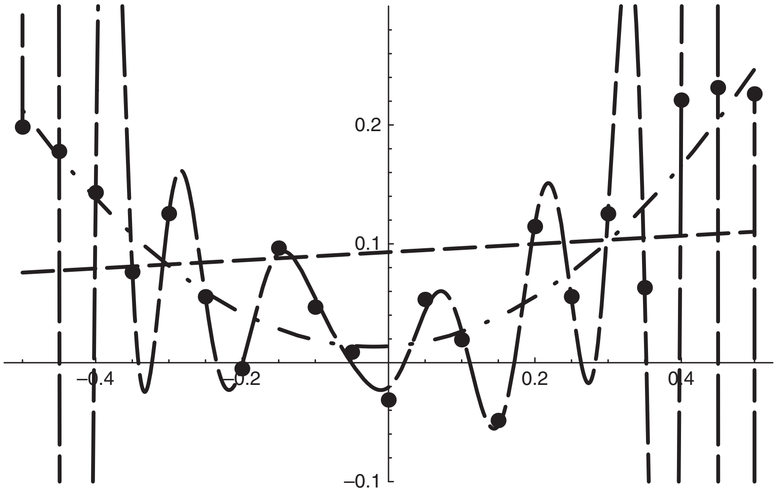

Figure 1.2 Fitting a curve to a noisy set of samples: the data samples are from a quadratic function with added noise; the fitting curves are a linear fit, a quadratic fit and a high-degree polynomial.

Optimality
A second point about the design of optimal classifiers concerns the word 'optimal'. There are several ways of measuring classifier performance, the most common being error rate, although this has severe limitations (see Chapter 9). Other measures, based on the closeness of the estimates of the probabilities of class membership to the true probabilities, may be more appropriate in many cases. However, many classifier design methods usually optimise alternative criteria since the desired ones are difficult to optimise directly. For example, a classifier may be trained by optimising a square-error measure and assessed using error rate.

Representative data
Finally, we assume that the training data are representative of the test conditions. If this is not so, perhaps because the test conditions may be subject to noise not present in the training data, or there are changes in the population from which the data are drawn (population drift), then these differences must be taken into account in the classifier design.

1.4 Approaches to statistical pattern recognition

There are two main divisions of classification: *supervised classification* (or discrimination) and *unsupervised classification* (sometimes in the statistics literature simply referred to as classification or clustering).

The problem we are addressing in this book is primarily one of *supervised pattern classification*. Given a set of measurements obtained through observation and represented as a pattern vector $\boldsymbol{x}$, we wish to assign the pattern to one of C possible classes, ω_i, $i = 1, \ldots, C$. A *decision rule* partitions the measurement space into C regions, Ω_i, $i = 1, \ldots, C$. If an observation vector is in Ω_i then it is assumed to belong to class ω_i. Each class region Ω_i may be multiply connected – that is, it may be made up of several disjoint regions. The boundaries between the regions Ω_i are the *decision boundaries* or *decision surfaces*. Generally, it is in regions close to these boundaries where the highest proportion of misclassifications occurs. In such situations, we may reject the pattern or withhold a decision until further information is available so that a classification may be made later. This option is known as the *reject option* and therefore we have $C + 1$ outcomes of a decision rule (the reject option being denoted by ω_0) in a C class problem: $\boldsymbol{x}$ belongs to ω_1 or ω_2 or ... or ω_C or withhold a decision.

In unsupervised classification, the data are not labelled and we seek to find groups in the data and the features that distinguish one group from another. Clustering techniques, described further in Chapter 11, can also be used as part of a supervised classification scheme by defining prototypes. A clustering scheme may be applied to the data for each class separately and representative samples for each group within the class (the group means for example) used as the prototypes for that class.

In the following section we introduce two approaches to discrimination that will be explored further in later chapters. The first assumes a knowledge of the underlying class-conditional probability density functions (the probability density function of the feature vectors for a given class). Of course, in many applications these will usually be unknown and must be estimated from a set of correctly classified samples termed the *design* or *training* set. Chapters 2, 3 and 4 describe techniques for estimating the probability density functions explicitly.

The second approach introduced in the next section develops decision rules that use the data to estimate the decision boundaries directly, without explicit calculation of the probability density functions. This approach is developed in Chapters 5 and 6 where specific techniques are described.

1.5 Elementary decision theory

Here we introduce an approach to discrimination based on knowledge of the probability density functions of each class. Familiarity with basic probability theory is assumed.

1.5.1 Bayes' decision rule for minimum error

Consider C classes, $\omega_1, \dots, \omega_C$, with *a priori* probabilities (the probabilities of each class occurring) $p(\omega_1), \dots, p(\omega_C)$, assumed known. If we wish to minimise the probability of making an error and we have no information regarding an object other than the class probability distribution then we would assign an object to class ω_j if

$$p(\omega_j) > p(\omega_k) \qquad k = 1, \dots, C; k \neq j$$

This classifies all objects as belonging to one class: the class with the largest prior probability. For classes with equal prior probabilities, patterns are assigned arbitrarily between those classes.

However, we do have an *observation vector* or *measurement vector* $\boldsymbol{x}$ and we wish to assign an object to one of the C classes based on the measurements $\boldsymbol{x}$. A decision rule based on probabilities is to assign $\boldsymbol{x}$ (here we refer to an object in terms of its measurement vector) to class ω_j if the probability of class ω_j given the observation $\boldsymbol{x}$, that is $p(\omega_j|\boldsymbol{x})$, is greatest over all classes $\omega_1, \dots, \omega_C$. That is, assign $\boldsymbol{x}$ to class ω_j if

$$p(\omega_j|\boldsymbol{x}) > p(\omega_k|\boldsymbol{x}) \qquad k = 1, \dots, C; k \neq j \tag{1.1}$$

This decision rule partitions the measurement space into C regions $\Omega_1, \dots, \Omega_C$ such that if $\boldsymbol{x} \in \Omega_j$ then $\boldsymbol{x}$ belongs to class ω_j. The regions Ω_j may be disconnected.

The *a posteriori* probabilities $p(\omega_j|\boldsymbol{x})$ may be expressed in terms of the *a priori* probabilities and the class conditional density functions $p(\boldsymbol{x}|\omega_i)$ using Bayes' theorem as

$$p(\omega_i|\boldsymbol{x}) = \frac{p(\boldsymbol{x}|\omega_i)p(\omega_i)}{p(\boldsymbol{x})}$$

and so the decision rule (1.1) may be written: assign $\boldsymbol{x}$ to ω_j if

$$p(\boldsymbol{x}|\omega_j)p(\omega_j) > p(\boldsymbol{x}|\omega_k)p(\omega_k) \qquad k = 1, \dots, C; k \neq j \tag{1.2}$$

This is known as Bayes' rule for *minimum error*.

For two classes, the decision rule (1.2) may be written

$$l_r(\boldsymbol{x}) = \frac{p(\boldsymbol{x}|\omega_1)}{p(\boldsymbol{x}|\omega_2)} > \frac{p(\omega_2)}{p(\omega_1)} \text{ implies } \boldsymbol{x} \in \text{class } \omega_1$$

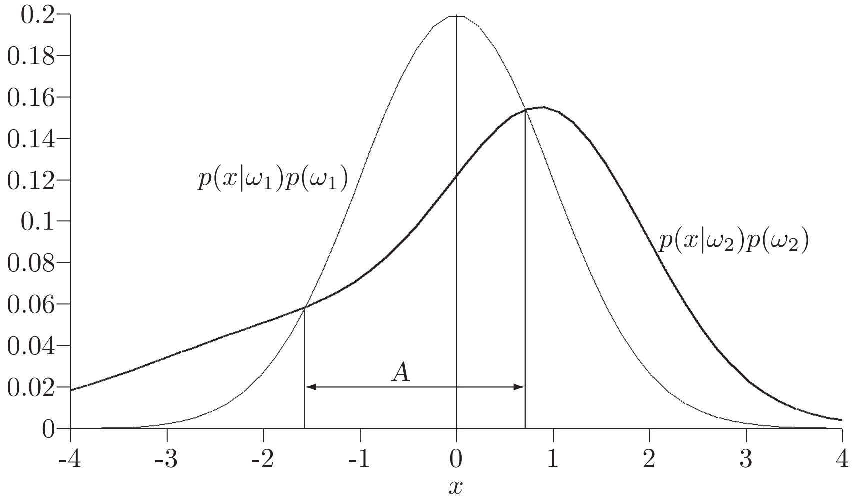

Figure 1.3 $p(x|\omega_i)p(\omega_i)$, for classes ω_1 and ω_2: for x in region A, x is assigned to class ω_1.

The function $l_r(\boldsymbol{x})$ is the *likelihood ratio*. Figures 1.3 and 1.4 give a simple illustration for a two-class discrimination problem. Class ω_1 is normally distributed with zero mean and unit variance, $p(\boldsymbol{x}|\omega_1) = N(\boldsymbol{x}; 0, 1)$. Class ω_2 is a *normal mixture* (a weighted sum of normal densities) $p(\boldsymbol{x}|\omega_2) = 0.6N(\boldsymbol{x}; 1, 1) + 0.4N(\boldsymbol{x}; -1, 2)$. Figure 1.3 plots $p(\boldsymbol{x}|\omega_i)p(\omega_i), i = 1, 2$, where the priors are taken to be $p(\omega_1) = 0.5, p(\omega_2) = 0.5$. Figure 1.4 plots the likelihood ratio $l_r(\boldsymbol{x})$ and the threshold $p(\omega_2)/p(\omega_1)$. We see from this figure that the decision rule (1.2) leads to a disconnected region for class ω_2.

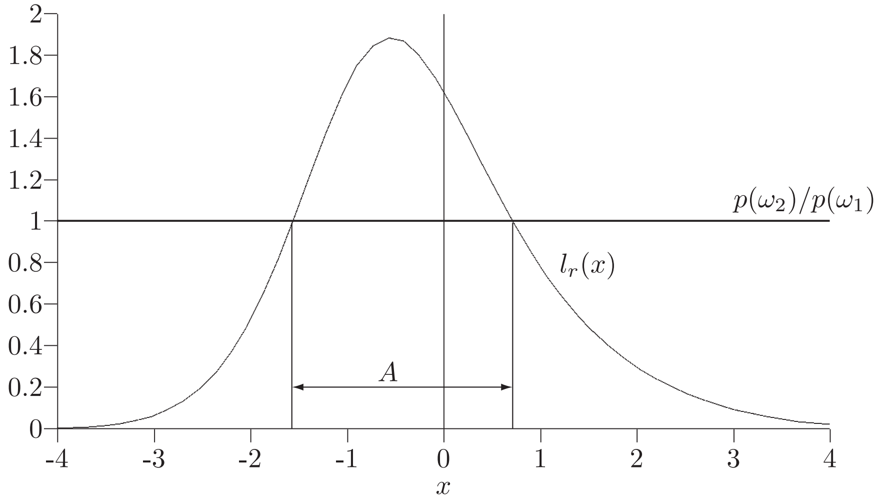

Figure 1.4 Likelihood function: for x in region A, x is assigned to class ω_1.

The fact that the decision rule (1.2) minimises the error may be seen as follows. The probability of making an error, $p(\text{error})$, may be expressed as

$$p(\text{error}) = \sum_{i=1}^{C} p(\text{error}|\omega_i)p(\omega_i) \tag{1.3}$$

where $p(\text{error}|\omega_i)$ is the probability of misclassifying patterns from class ω_i. This is given by

$$p(\text{error}|\omega_i) = \int_{\mathcal{C}[\Omega_i]} p(\boldsymbol{x}|\omega_i)d\boldsymbol{x} \tag{1.4}$$

the integral of the class-conditional density function over $\mathcal{C}[\Omega_i]$, the region of measurement space outside Ω_i ($\mathcal{C}$ is the complement operator), i.e. $\sum_{j=1, j\neq i}^{C} \Omega_j$. Therefore, we may write the probability of misclassifying a pattern as

$$\begin{aligned} p(\text{error}) &= \sum_{i=1}^{C} \int_{\mathcal{C}[\Omega_i]} p(\boldsymbol{x}|\omega_i)p(\omega_i)d\boldsymbol{x} \\ &= \sum_{i=1}^{C} p(\omega_i)\left(1 - \int_{\Omega_i} p(\boldsymbol{x}|\omega_i)d\boldsymbol{x}\right) \\ &= 1 - \sum_{i=1}^{C} p(\omega_i)\int_{\Omega_i} p(\boldsymbol{x}|\omega_i)d\boldsymbol{x} \end{aligned} \tag{1.5}$$

from which we see that minimising the probability of making an error is equivalent to maximising

$$\sum_{i=1}^{C} p(\omega_i)\int_{\Omega_i} p(\boldsymbol{x}|\omega_i)d\boldsymbol{x} \tag{1.6}$$

which is the probability of correct classification. Therefore, we wish to choose the regions Ω_i so that the integral given in (1.6) is a maximum. This is achieved by selecting Ω_i to be the region for which $p(\omega_i)p(\boldsymbol{x}|\omega_i)$ is the largest over all classes and the probability of correct classification, c, is

$$c = \int \max_i p(\omega_i)p(\boldsymbol{x}|\omega_i)d\boldsymbol{x} \tag{1.7}$$

where the integral is over the whole of the measurement space, and the Bayes' error is

$$e_B = 1 - \int \max_i p(\omega_i)p(\boldsymbol{x}|\omega_i)d\boldsymbol{x} \tag{1.8}$$

This is illustrated in Figures 1.5 and 1.6.

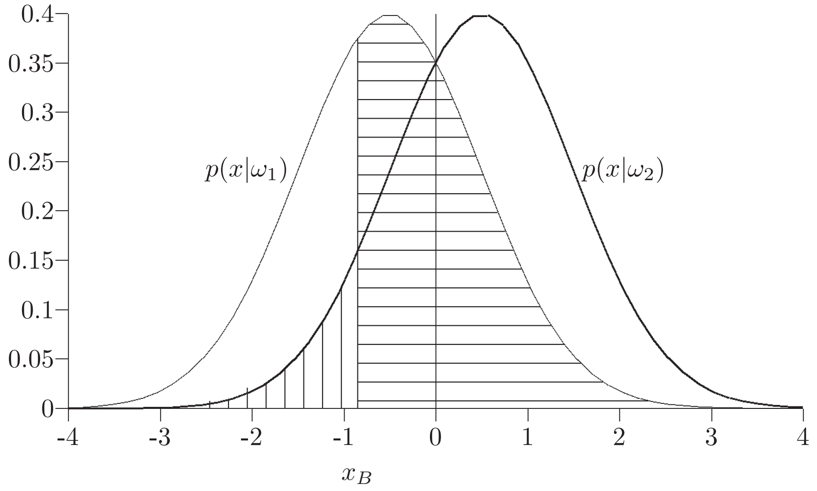

Figure 1.5 Class-conditional densities for two normal distributions.

Figure 1.5 plots the two distributions $p(\boldsymbol{x}|\omega_i)$, $i = 1, 2$ (both normal with unit variance and means ±0.5), and Figure 1.6 plots the functions $p(\boldsymbol{x}|\omega_i)p(\omega_i)$ where $p(\omega_1) = 0.3, p(\omega_2) = 0.7$. The Bayes' decision boundary defined by the point where $p(\boldsymbol{x}|\omega_1)p(\omega_1) = p(\boldsymbol{x}|\omega_2)p(\omega_2)$ (Figure 1.6) is marked with a vertical line at x_B. The areas of the hatched regions in Figure 1.5 represent the probability of error: by Equation (1.4), the area of the horizontal hatching is the probability of classifying a pattern from class 1 as a pattern from class 2 and the area of the vertical hatching the probability of classifying a pattern from class 2 as class 1. The sum of these two areas, weighted by the priors [Equation (1.5)], is the probability of making an error.

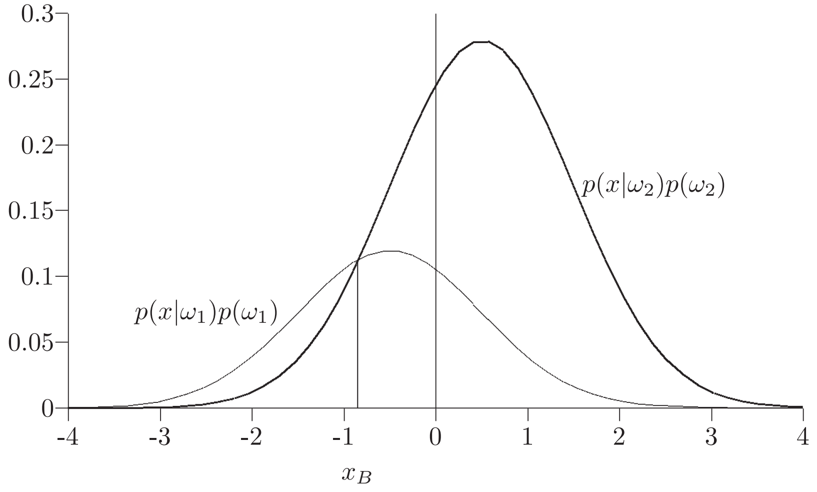

Figure 1.6 Bayes' decision boundary for two normally distributed classes with unequal priors.

1.5.2 Bayes' decision rule for minimum error – reject option

As we have stated above, an error or misrecognition occurs when the classifier assigns a pattern to one class when it actually belongs to another. In this section we consider the reject option. Usually it is the uncertain classifications (often close to the decision boundaries) that contribute mainly to the error rate. Therefore, rejecting a pattern (withholding a decision) may lead to a reduction in the error rate. This rejected pattern may be discarded, or set aside until further information allows a decision to be made. Although the option to reject may alleviate or remove the problem of a high misrecognition rate, some otherwise correct classifications are also converted into rejects. Here we consider the trade-offs between error rate and reject rate.

First, we partition the sample space into two complementary regions: R, a *reject region* and A, an *acceptance* or *classification region*. These are defined by

$$R = \left\{ \boldsymbol{x} | 1 - \max_i p(\omega_i | \boldsymbol{x}) > t \right\}$$

$$A = \left\{ \boldsymbol{x} | 1 - \max_i p(\omega_i | \boldsymbol{x}) \leq t \right\}$$

where t is a threshold. This is illustrated in Figure 1.7 using the same distributions as those in Figures 1.5 and 1.6.

The smaller the value of the threshold t then the larger is the reject region R. However, if t is chosen such that

$$1 - t \leq \frac{1}{C}$$

or equivalently,

$$t \geq \frac{C-1}{C}$$

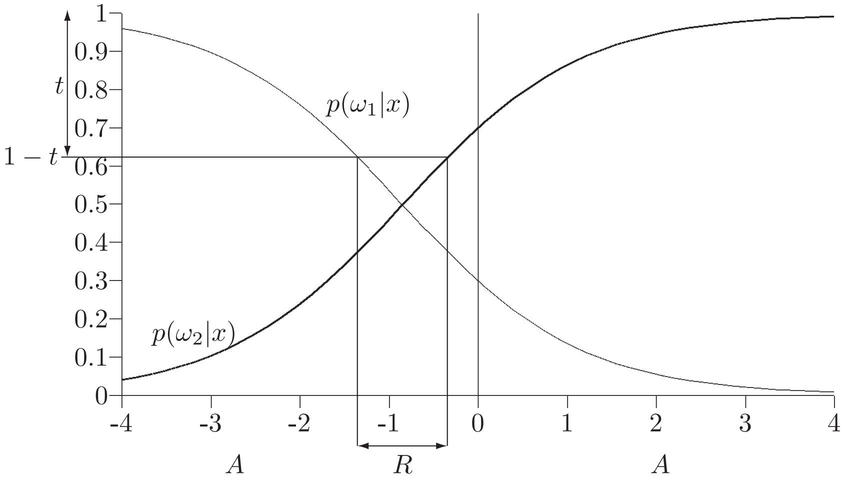

Figure 1.7 Illustration of acceptance and reject regions.

where C is the number of classes, then the reject region is empty. This is because the minimum value which $\max_i p(\omega_i|\boldsymbol{x})$ can attain is 1/C [since $1 = \sum_{i=1}^{C} p(\omega_i|\boldsymbol{x}) \leq C \max_i p(\omega_i|\boldsymbol{x})$], when all classes are equally likely. Therefore, for the reject option to be activated, we must have $t < (C-1)/C$.

Thus, if a pattern $\boldsymbol{x}$ lies in the region A, we classify it according to the Bayes' rule for minimum error [Equation (1.2)]. However, if $\boldsymbol{x}$ lies in the region R, we reject $\boldsymbol{x}$ (withhold a decision).

The probability of correct classification, $c(t)$, is a function of the threshold, t, and is given by Equation (1.7), where now the integral is over the acceptance region, A, only

$$c(t) = \int_A \max_i \left[p(\omega_i)p(\boldsymbol{x}|\omega_i)\right] d\boldsymbol{x}$$

and the unconditional probability of rejecting a measurement, r, also a function of the threshold t, is the probability that it lies in R:

$$r(t) = \int_R p(\boldsymbol{x})d\boldsymbol{x} \tag{1.9}$$

Therefore, the error rate, e (the probability of accepting a point for classification and incorrectly classifying it), is

$$\begin{aligned} e(t) &= \int_A \left(1 - \max_i p(\omega_i|\boldsymbol{x})\right) p(\boldsymbol{x})d\boldsymbol{x} \\ &= 1 - c(t) - r(t) \end{aligned}$$

Thus, the error rate and reject rate are inversely related. Chow (1970) derives a simple functional relationship between $e(t)$ and $r(t)$ which we quote here without proof. Knowing $r(t)$ over the complete range of t allows $e(t)$ to be calculated using the relationship

$$e(t) = -\int_0^t s dr(s) \tag{1.10}$$

The above result allows the error rate to be evaluated from the reject function for the Bayes' optimum classifier. The reject function can be calculated using *unlabelled* data and a practical application of the above result is to problems where labelling of gathered data is costly.

1.5.3 Bayes' decision rule for minimum risk

In the previous section, the decision rule selected the class for which the *a posteriori* probability, $p(\omega_j|\boldsymbol{x})$, was the greatest. This minimised the probability of making an error. We now consider a somewhat different rule that minimises an expected *loss* or risk. This is a very important concept since in many applications the costs associated with misclassification depend upon the true class of the pattern and the class to which it is assigned. For example, in a medical diagnosis problem in which a patient has back pain, it is far worse to classify a patient with severe spinal abnormality as healthy (or having mild back ache) than the other way round.

We make this concept more formal by introducing a loss that is a measure of the cost of making the decision that a pattern belongs to class ω_i when the true class is ω_j. We define a loss matrix $\mathbf{\Lambda}$ with components

$$\lambda_{ji} = \text{cost of assigning a pattern } \boldsymbol{x} \text{ to } \omega_i \text{ when } \boldsymbol{x} \in \omega_j$$

In practice, it may be very difficult to assign costs. In some situations, λ may be measured in monetary units that are quantifiable. However, in many situations, costs are a combination of several different factors measured in different units – money, time, quality of life. As a consequence, they are often a subjective opinion of an expert. The *conditional risk* of assigning a pattern $\boldsymbol{x}$ to class ω_i is defined as

$$l^i(\boldsymbol{x}) = \sum_{j=1}^{C} \lambda_{ji} p(\omega_j|\boldsymbol{x})$$

The average risk over region Ω_i is

$$\begin{aligned} r^i &= \int_{\Omega_i} l^i\,(\boldsymbol{x})\; p(\boldsymbol{x}) d\boldsymbol{x} \\ &= \int_{\Omega_i} \sum_{j=1}^{C} \lambda_{ij} p(\omega_i|\boldsymbol{x}) p(\boldsymbol{x}) d\boldsymbol{x} \end{aligned}$$

and the overall expected cost or *risk* is

$$r = \sum_{i=1}^{C} r^i = \sum_{i=1}^{C} \int_{\Omega_i} \sum_{j=1}^{C} \lambda_{ji} p(\omega_j|\boldsymbol{x}) p(\boldsymbol{x}) d\boldsymbol{x} \tag{1.11}$$

The above expression for the risk will be minimised if the regions Ω_i are chosen such that if

$$\sum_{j=1}^{C} \lambda_{ji} p(\omega_j|\boldsymbol{x}) p(\boldsymbol{x}) \leq \sum_{j=1}^{C} \lambda_{jk} p(\omega_j|\boldsymbol{x}) p(\boldsymbol{x}) \quad k = 1, \ldots, C \tag{1.12}$$

then $\boldsymbol{x} \in \Omega_i$. This is the *Bayes' decision rule for minimum risk*, with Bayes' risk, r^*, given by

$$r^* = \int_{\boldsymbol{x}} \min_{i=1,\ldots,C} \sum_{j=1}^{C} \lambda_{ji} p(\omega_j|\boldsymbol{x}) p(\boldsymbol{x}) d\boldsymbol{x}$$

One special case of the loss matrix $\mathbf{\Lambda}$ is the *equal cost* loss matrix for which

$$\lambda_{ij} = \begin{cases} 1 & i \neq j \\ 0 & i = j \end{cases}$$

Substituting into (1.12) gives the decision rule: assign $\boldsymbol{x}$ to class ω_i if

$$\sum_{j=1}^{C} p(\omega_j|\boldsymbol{x})p(\boldsymbol{x}) - p(\omega_i|\boldsymbol{x})p(\boldsymbol{x}) \leq \sum_{j=1}^{C} p(\omega_j|\boldsymbol{x})p(\boldsymbol{x}) - p(\omega_k|\boldsymbol{x})p(\boldsymbol{x}) \quad k = 1, \ldots, C$$

that is,

$$p(\boldsymbol{x}|\omega_i)p(\omega_i) \geq p(\boldsymbol{x}|\omega_k)p(\omega_k) \quad k = 1, \ldots, C$$

implies that $\boldsymbol{x} \in$ class ω_i; this is the Bayes' rule for minimum error.

1.5.4 Bayes' decision rule for minimum risk – reject option

As with the Bayes' rule for minimum error, we may also introduce a reject option, by which the reject region, R, is defined by

$$R = \left\{ \boldsymbol{x} \middle| \min_i l^i(\boldsymbol{x}) > t \right\}$$

where t is a threshold. The decision is to accept a pattern $\boldsymbol{x}$ and assign it to class ω_i if

$$l^i(\boldsymbol{x}) = \min_j l^j(\boldsymbol{x}) \leq t$$

and to reject $\boldsymbol{x}$ if

$$l^i(\boldsymbol{x}) = \min_j l^j(\boldsymbol{x}) > t$$

This decision is equivalent to defining a reject region Ω_0 with a constant conditional risk

$$l^0(\boldsymbol{x}) = t$$

so that the Bayes' decision rule is: assign $\boldsymbol{x}$ to class ω_i if

$$l^i(\boldsymbol{x}) \leq l^j(\boldsymbol{x}) \quad j = 0, 1, \ldots, C$$

with Bayes' risk

$$r^* = \int_R tp(\boldsymbol{x})d\boldsymbol{x} + \int_A \min_{i=1,\ldots,C} \sum_{j=1}^{C} \lambda_{ji} p(\omega_j|\boldsymbol{x})p(\boldsymbol{x})d\boldsymbol{x} \tag{1.13}$$

1.5.5 Neyman–Pearson decision rule

An alternative to the Bayes' decision rules for a two-class problem is the Neyman–Pearson test. In a two-class problem there are two possible types of error that may be made in the decision process. We may classify a pattern of class ω_1 as belonging to class ω_2 or a pattern from class ω_2 as belonging to class ω_1. Let the probability of these two errors be ϵ_1 and ϵ_2, respectively, so that

$$\epsilon_1 = \int_{\Omega_2} p(\boldsymbol{x}|\omega_1)\, d\boldsymbol{x} = \text{error probability of Type I}$$

and

$$\epsilon_2 = \int_{\Omega_1} p(\boldsymbol{x}|\omega_2)\, d\boldsymbol{x} = \text{error probability of Type II}$$

The Neyman–Pearson decision rule is to minimise the error ϵ_1 subject to ϵ_2 being equal to a constant, ϵ_0, say.

If class ω_1 is termed the positive class and class ω_2 the negative class, then ϵ_1 is referred to as the *false negative rate*: the proportion of positive samples incorrectly assigned to the negative class; ϵ_2 is the *false positive rate*: the proportion of negative samples classed as positive.

An example of the use of the Neyman–Pearson decision rule is in radar detection where the problem is to detect a signal in the presence of noise. There are two types of error that may occur; one is to mistake noise for a signal present. This is called a *false alarm*. The second type of error occurs when a signal is actually present but the decision is made that only noise is present. This is a *missed detection*. If ω_1 denotes the signal class and ω_2 denotes the noise then ϵ_2 is the probability of false alarm and ϵ_1 is the probability of missed detection. In many radar applications, a threshold is set to give a fixed probability of false alarm and therefore the Neyman–Pearson decision rule is the one usually used.

We seek the minimum of

$$r = \int_{\Omega_2} p(\boldsymbol{x}|\omega_1)\, d\boldsymbol{x} + \mu \left\{ \int_{\Omega_1} p(\boldsymbol{x}|\omega_2)\, d\boldsymbol{x} - \epsilon_0 \right\}$$

where μ is a Lagrange multiplier[1] and ϵ_0 is the specified false alarm rate. The equation may be written

$$r = (1 - \mu\epsilon_0) + \int_{\Omega_1} \{\mu p(\boldsymbol{x}|\omega_2)\, d\boldsymbol{x} - p(\boldsymbol{x}|\omega_1)\, d\boldsymbol{x}\}$$

This will be minimised if we choose Ω_1 such that the integrand is negative, i.e.

$$\text{if } \mu p(\boldsymbol{x}|\omega_2) - p(\boldsymbol{x}|\omega_1) < 0, \text{ then } \boldsymbol{x} \in \Omega_1$$

or, in terms of the likelihood ratio,

$$\text{if } \frac{p(\boldsymbol{x}|\omega_1)}{p(\boldsymbol{x}|\omega_2)} > \mu, \text{ then } \boldsymbol{x} \in \Omega_1 \tag{1.14}$$

[1] The method of Lagrange's undetermined multipliers can be found in most textbooks on mathematical methods, for example Wylie and Barrett (1995).

Thus the decision rule depends only on the within-class distributions and ignores the *a priori* probabilities.

The threshold μ is chosen so that

$$\int_{\Omega_1} p(\boldsymbol{x}|\omega_2)\, d\boldsymbol{x} = \epsilon_0,$$

the specified false alarm rate. However, in general μ cannot be determined analytically and requires numerical calculation.

Often, the performance of the decision rule is summarised in a receiver operating characteristic (ROC) curve, which plots the true positive against the false positive (that is, the probability of detection $[1 - \epsilon_1 = \int_{\Omega_1} p(\boldsymbol{x}|\omega_1)\, d\boldsymbol{x}]$ against the probability of false alarm $[\epsilon_2 = \int_{\Omega_1} p(\boldsymbol{x}|\omega_2)\, d\boldsymbol{x})]$ as the threshold μ is varied. This is illustrated in Figure 1.8 for the univariate case of two normally distributed classes of unit variance and means separated by a distance, d. All the ROC curves pass through the (0, 0) and (1, 1) points and as the separation increases the curve moves into the top left corner. Ideally, we would like 100% detection for a 0% false alarm rate and curves that are closer to this the better.

For the two-class case, the minimum risk decision [see Equation (1.12)] defines the decision rules on the basis of the likelihood ratio ($\lambda_{ii} = 0$):

$$\text{if } \frac{p(\boldsymbol{x}|\omega_1)}{p(\boldsymbol{x}|\omega_2)} > \frac{\lambda_{21} p(\omega_2)}{\lambda_{12} p(\omega_1)}, \text{ then } \boldsymbol{x} \in \Omega_1 \tag{1.15}$$

The threshold defined by the right-hand side will correspond to a particular point on the ROC curve that depends on the misclassification costs and the prior probabilities.

In practice, precise values for the misclassification costs will be unavailable and we shall need to assess the performance over a range of expected costs. The use of the ROC curve as a tool for comparing and assessing classifier performance is discussed in Chapter 9.

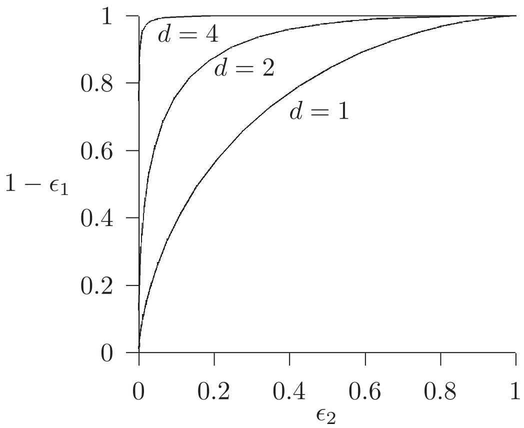

Figure 1.8 Receiver operating characteristic for two univariate normal distributions of unit variance and separation, d; $1 - \epsilon_1 = \int_{\Omega_1} p(\boldsymbol{x}|\omega_1)\, d\boldsymbol{x}$ is the true positive (the probability of detection) and $\epsilon_2 = \int_{\Omega_1} p(\boldsymbol{x}|\omega_2)\, d\boldsymbol{x}$ is the false positive (the probability of false alarm).

1.5.6 Minimax criterion

The Bayes' decision rules rely on a knowledge of both the within-class distributions and the prior class probabilities. However, situations may arise where the relative frequencies of new objects to be classified are unknown. In this situation a *minimax* procedure may be employed. The name *minimax* is used to refer to procedures for which either the maximum expected loss *or* the maximum of the error probability is a minimum. We shall limit our discussion below to the two-class problem and the minimum error probability procedure.

Consider the Bayes' rule for minimum error. The decision regions Ω_1 and Ω_2 are defined by

$$p(\boldsymbol{x}|\omega_1)p(\omega_1) > p(\boldsymbol{x}|\omega_2)p(\omega_2) \text{ implies } \boldsymbol{x} \in \Omega_1 \tag{1.16}$$

and the Bayes' minimum error, e_B, is

$$e_B = p(\omega_2)\int_{\Omega_1} p(\boldsymbol{x}|\omega_2)\,d\boldsymbol{x} + p(\omega_1)\int_{\Omega_2} p(\boldsymbol{x}|\omega_1)\,d\boldsymbol{x} \tag{1.17}$$

where $p(\omega_2) = 1 - p(\omega_1)$.

For *fixed* decision regions Ω_1 and Ω_2, e_B is a linear function of $p(\omega_1)$ (we denote this function $\tilde{e}_B$) attaining its maximum on the region [0, 1] either at $p(\omega_1) = 0$ or $p(\omega_1) = 1$. However, since the regions Ω_1 and Ω_2 are also dependent on $p(\omega_1)$ through the Bayes' decision criterion (1.16), the dependency of e_B on $p(\omega_1)$ is more complex, and not necessarily monotonic.

If Ω_1 and Ω_2 are fixed [determined according to (1.16) for some specified $p(\omega_i)$], the error given by (1.17) will only be the Bayes' minimum error for a particular value of $p(\omega_1)$, say p_1^* (Figure 1.9).

For other values of $p(\omega_1)$, the error given by (1.17) must be greater than the minimum error. Therefore, the optimum curve touches the line at a tangent at p_1^* and is concave down at that point.

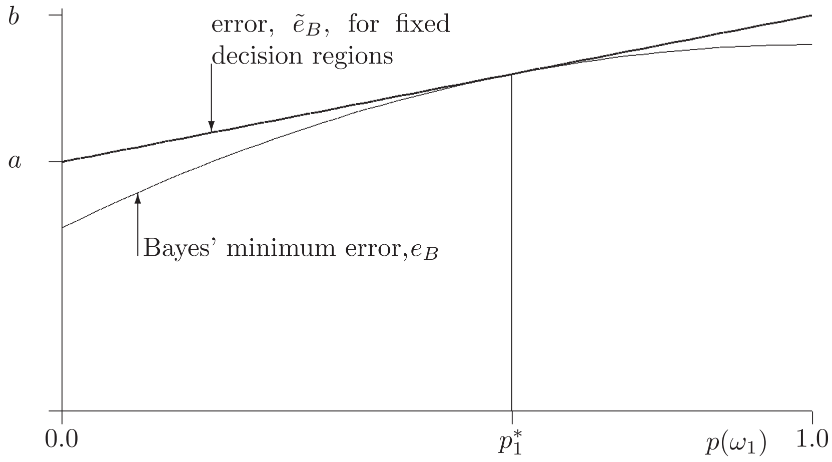

Figure 1.9 Minimax illustration.

The minimax procedure aims to choose the partition Ω_1, Ω_2, or equivalently the value of $p(\omega_1)$ so that the maximum error (on a test set in which the values of $p(\omega_i)$ are unknown) is minimised. For example, in Figure 1.9, if the partition were chosen to correspond to the value p_1^* of $p(\omega_1)$, then the maximum error which could occur would be a value of b if $p(\omega_1)$ were actually equal to unity. The minimax procedure aims to minimise this maximum value, i.e. minimise

$$\max\{\tilde{e}_B(0), \tilde{e}_B(1)\}$$

or minimise

$$\max\left\{\int_{\Omega_2} p(\boldsymbol{x}|\omega_1)\,d\boldsymbol{x}, \int_{\Omega_1} p(\boldsymbol{x}|\omega_2)\,d\boldsymbol{x}\right\}$$

This is a minimum when

$$\int_{\Omega_2} p(\boldsymbol{x}|\omega_1)\,d\boldsymbol{x} = \int_{\Omega_1} p(\boldsymbol{x}|\omega_2)\,d\boldsymbol{x} \tag{1.18}$$

which is when $a = b$ in Figure 1.9 and the line $\tilde{e}_B(p(\omega_1))$ is horizontal and touches the Bayes' minimum error curve at its peak value.

Therefore, we choose the regions Ω_1 and Ω_2 so that the probabilities of the two types of error are the same. The minimax solution may be criticised as being over-pessimistic since it is a Bayes' solution with respect to the least favourable prior distribution. The strategy may also be applied to minimising the maximum risk. In this case, the risk is

$$\begin{aligned}\int_{\Omega_1} [\lambda_{11} p(\omega_1|\boldsymbol{x}) + \lambda_{21} p(\omega_2|\boldsymbol{x})]\, p(\boldsymbol{x}) d\boldsymbol{x} + \int_{\Omega_2} [\lambda_{12} p(\omega_1|\boldsymbol{x}) + \lambda_{22} p(\omega_2|\boldsymbol{x})]\, p(\boldsymbol{x}) d\boldsymbol{x}\\ = p(\omega_1)\left[\lambda_{11} + (\lambda_{12} - \lambda_{11}) \int_{\Omega_2} p(\boldsymbol{x}|\omega_1) d\boldsymbol{x}\right]\\ + p(\omega_2)\left[\lambda_{22} + (\lambda_{21} - \lambda_{22}) \int_{\Omega_1} p(\boldsymbol{x}|\omega_2) d\boldsymbol{x}\right]\end{aligned}$$

and the boundary must therefore satisfy

$$\lambda_{11} - \lambda_{22} + (\lambda_{12} - \lambda_{11}) \int_{\Omega_2} p(\boldsymbol{x}|\omega_1)\,d\boldsymbol{x} - (\lambda_{21} - \lambda_{22}) \int_{\Omega_1} p(\boldsymbol{x}|\omega_2)\,d\boldsymbol{x} = 0$$

For $\lambda_{11} = \lambda_{22}$ and $\lambda_{21} = \lambda_{12}$, this reduces to the condition (1.18).

1.5.7 Discussion

In this section we have introduced a decision theoretic approach to classifying patterns. This divides up the measurement space into decision regions and we have looked at various strategies for obtaining the decision boundaries. The optimum rule in the sense of minimising the error is the Bayes' decision rule for minimum error. Introducing the costs of making incorrect decisions leads to the Bayes' rule for minimum risk. The theory developed assumes that the *a priori* distributions and the class-conditional distributions are known. In a real-world task, this is unlikely to be so. Therefore approximations must be made based on the data available. We consider techniques for estimating distributions in Chapters 2, 3 and 4. Two alternatives to the Bayesian decision rule have also been described, namely the Neyman–Pearson decision rule (commonly used in signal processing applications) and the minimax rule. Both require knowledge of the class-conditional probability density functions. The ROC curve characterises the performance of a rule over a range of thresholds of the likelihood ratio.

We have seen that the error rate plays an important part in decision making and classifier performance assessment. Consequently, estimation of error rates is a problem of great interest in statistical pattern recognition. For given fixed decision regions, we may calculate the probability of error using Equation (1.5). If these decision regions are chosen according to the Bayes' decision rule [Equation (1.2)], then the error is the *Bayes' error rate* or *optimal error rate*. However, regardless of how the decision regions are chosen, the error rate may be regarded as a measure of a given decision rule's performance.

Calculation of the Bayes' error rate (1.8) requires complete knowledge of the class conditional density functions. In a particular situation, these may not be known and a classifier may be designed on the basis of a training set of samples. Given this training set, we may choose to form *estimates* of the distributions (using some of the techniques discussed in Chapters 2 and 3) and thus, with these estimates, use the Bayes decision rule and estimate the error according to (1.8).

However, even with accurate estimates of the distributions, evaluation of the error requires an integral over a multidimensional space and may prove a formidable task. An alternative approach is to obtain bounds on the optimal error rate or distribution-free estimates. Further discussion of methods of error rate estimation is given in Chapter 9.

1.6 Discriminant functions

1.6.1 Introduction

In the previous section, classification was achieved by applying the Bayesian decision rule. This requires knowledge of the class-conditional density functions, $p(\boldsymbol{x}|\omega_i)$ (such as normal distributions whose parameters are estimated from the data – see Chapter 2), or nonparametric density estimation methods (such as kernel density estimation – see Chapter 4). Here, instead of making assumptions about $p(\boldsymbol{x}|\omega_i)$, we make assumptions about the forms of the *discriminant functions*.

A discriminant function is a function of the pattern $\boldsymbol{x}$ that leads to a classification rule. For example, in a two-class problem, a discriminant function $h(\boldsymbol{x})$ is a function for which

$$\begin{aligned} h(\boldsymbol{x}) > k &\Rightarrow x \in \omega_1 \\ h(\boldsymbol{x}) < k &\Rightarrow x \in \omega_2 \end{aligned} \tag{1.19}$$

for constant k. In the case of equality $[h(\boldsymbol{x}) = k]$, the pattern $\boldsymbol{x}$ may be assigned arbitrarily to one of the two classes. An optimal discriminant function for the two-class case is

$$h(\boldsymbol{x}) = \frac{p(\boldsymbol{x}|\omega_1)}{p(\boldsymbol{x}|\omega_2)}$$

with $k = p(\omega_2)/p(\omega_1)$. Discriminant functions are not unique. If f is a monotonic function then

$$g(\boldsymbol{x}) = f\ (h(x)) > k' \Rightarrow x \in \omega_1$$
$$g(\boldsymbol{x}) = f\ (h(x)) < k' \Rightarrow x \in \omega_2$$

where $k' = f(k)$ leads to the same decision as (1.19).

In the C group case we define C discriminant functions $g_i(\boldsymbol{x})$ such that

$$g_i(\boldsymbol{x}) > g_j(\boldsymbol{x}) \Rightarrow \boldsymbol{x} \in \omega_i \qquad j = 1, \ldots, C, j \neq i$$

That is, a pattern is assigned to the class with the largest discriminant. Of course, for two classes, a single discriminant function

$$h(\boldsymbol{x}) = g_1(\boldsymbol{x}) - g_2(\boldsymbol{x})$$

with $k = 0$ reduces to the two-class case given by (1.19).

Again, we may define an optimal discriminant function as

$$g_i(\boldsymbol{x}) = p(\boldsymbol{x}|\omega_i)p(\omega_i)$$

leading to the Bayes' decision rule, but as we showed for the two-class case, there are other discriminant functions that lead to the same decision.

The essential difference between the approach of the previous section and the discriminant function approach described here is that the form of the discriminant function is specified and is not imposed by the underlying distribution. The choice of discriminant function may depend on prior knowledge about the patterns to be classified or may be a particular functional form whose parameters are adjusted by a training procedure. Many different forms of discriminant function have been considered in the literature, varying in complexity from the linear discriminant function (in which g is a linear combination of the x_i) to multiparameter nonlinear functions such as the multilayer perceptron.

Discrimination may also be viewed as a problem in *regression* (see Section 1.7) in which the dependent variable, y, is a class indicator and the regressors are the pattern vectors. Many discriminant function models lead to estimates of $\mathrm{E}[y|\boldsymbol{x}]$, which is the aim of regression analysis (though in regression y is not necessarily a class indicator). Thus, many of the techniques we shall discuss for optimising discriminant functions apply equally well to regression problems. Indeed, as we find with feature extraction in Chapter 10 and also clustering in Chapter 11 similar techniques have been developed under different names in the pattern recognition and statistics literature.

1.6.2 Linear discriminant functions

First of all, let us consider the family of discriminant functions that are linear combinations of the components of $\boldsymbol{x} = (x_1, \ldots, x_p)^T$,

$$g(\boldsymbol{x}) = \boldsymbol{w}^T\boldsymbol{x} + w_0 = \sum_{i=1}^{p} w_i x_i + w_0 \qquad (1.20)$$

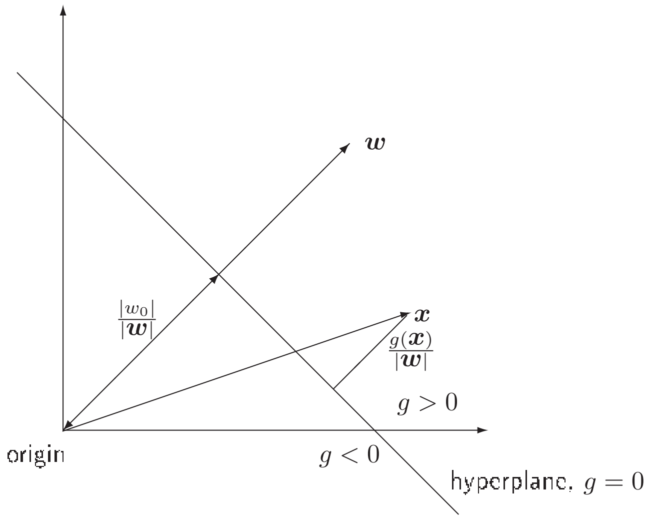
$\boldsymbol{w}$
$\frac{|w_0|}{|\boldsymbol{w}|}$
$\boldsymbol{x}$
$\frac{g(\boldsymbol{x})}{|\boldsymbol{w}|}$
$g > 0$
origin
$g < 0$
hyperplane, $g = 0$

Figure 1.10 Geometry of linear discriminant function given by Equation (1.20).

This is a *linear discriminant function*, a complete specification of which is achieved by prescribing the *weight vector* $\boldsymbol{w}$ and *threshold weight* w_0. Equation (1.20) is the equation of a hyperplane with unit normal in the direction of $\boldsymbol{w}$ and a perpendicular distance $|w_0|/|\boldsymbol{w}|$ from the origin. The value of the discriminant function for a pattern $\boldsymbol{x}$ is a measure of the perpendicular distance from the hyperplane (Figure 1.10).

A linear discriminant function can arise through assumptions of normal distributions for the class densities, with equal covariance matrices (see Chapter 2). Alternatively, without making distributional assumptions, we may impose the form of the discriminant function to be linear and determine its parameters (see Chapter 5).

A pattern classifier employing linear discriminant functions is termed a *linear machine* (Nilsson, 1965), an important special case of which is the *minimum-distance classifier*. Suppose we are given a set of prototype points $\boldsymbol{p}_1, \ldots, \boldsymbol{p}_C$, one for each of the C classes $\omega_1, \ldots, \omega_C$. The minimum-distance classifier assigns a pattern $\boldsymbol{x}$ to the class ω_i associated with the nearest point $\boldsymbol{p}_i$. For each point, the squared Euclidean distance is

$$|\boldsymbol{x} - \boldsymbol{p}_i|^2 = \boldsymbol{x}^T\boldsymbol{x} - 2\boldsymbol{x}^T\boldsymbol{p}_i + \boldsymbol{p}_i^T\boldsymbol{p}_i$$

and minimum-distance classification is achieved by comparing the expressions $\boldsymbol{x}^T\boldsymbol{p}_i - \frac{1}{2}\boldsymbol{p}_i^T\boldsymbol{p}_i$ and selecting the largest value. Thus, the linear discriminant function is

$$g_i(\boldsymbol{x}) = \boldsymbol{w}_i^T\boldsymbol{x} + w_{i0}$$

where

$$\boldsymbol{w}_i = \boldsymbol{p}_i$$
$$\boldsymbol{w}_{i0} = -\frac{1}{2}|\boldsymbol{p}_i|^2$$

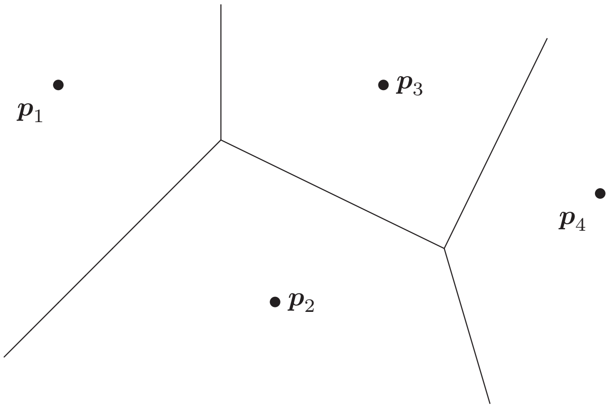

Figure 1.11 Decision regions for a minimum-distance classifier.

Therefore, the minimum-distance classifier is a linear machine. If the prototype points, $\boldsymbol{p}_i$, are the class means, then we have the nearest class mean classifier. Decision regions for a minimum-distance classifier are illustrated in Figure 1.11. Each boundary is the perpendicular bisector of the lines joining the prototype points of regions that are contiguous. Also, note from Figure 1.11 that the decision regions are convex (that is, two arbitrary points lying in the region can be joined by a straight line that lies entirely within the region). In fact, decision regions of a linear machine are always convex. Thus, the two class problems, illustrated in Figure 1.12, although separable, cannot be separated by a linear machine. Two generalisations that overcome this difficulty are piecewise linear discriminant functions and generalised linear discriminant functions.

1.6.3 Piecewise linear discriminant functions

This is a generalisation of the minimum-distance classifier to the situation in which there is more than one prototype per class. Suppose there are n_i prototypes in class ω_i, $\boldsymbol{p}_i^1, \ldots, \boldsymbol{p}_i^{n_i}$, $i = 1, \ldots, C$. We define the discriminant function for class ω_i to be

$$g_i(\boldsymbol{x}) = \max_{j=1,\ldots,n_i} g_i^j(\boldsymbol{x})$$

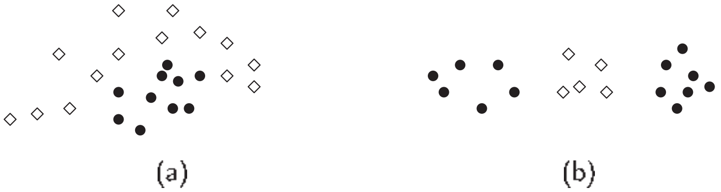

Figure 1.12 Two examples of groups not separable by a linear discriminant.

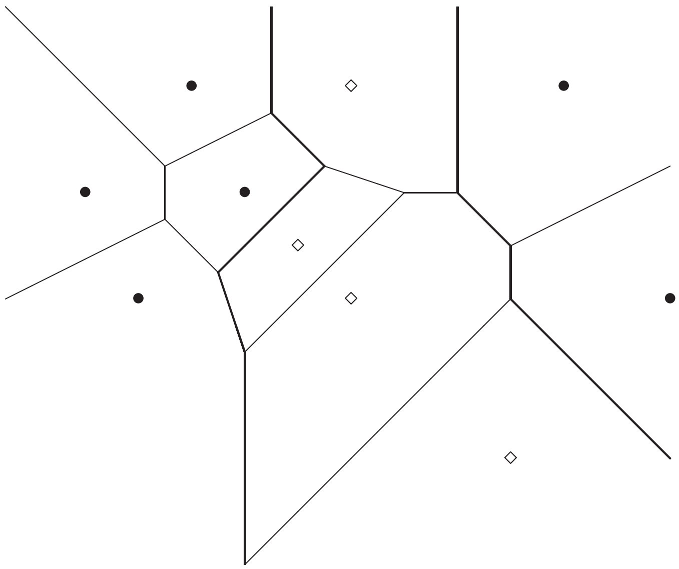

Figure 1.13 Dirichlet tessellation (comprising nearest-neighbour regions for a set of prototypes) and the decision boundary (thick lines) for two classes.

where g_i^j is a subsidiary discriminant function, which is linear and is given by

$$g_i^j(\boldsymbol{x}) = \boldsymbol{x}^T \boldsymbol{p}_i^j - \frac{1}{2}\boldsymbol{p}_i^{jT} \boldsymbol{p}_i^j \qquad j = 1, \ldots, n_i;\ i = 1, \ldots, C$$

A pattern $\boldsymbol{x}$ is assigned to the class for which $g_i(\boldsymbol{x})$ is largest; that is, to the class of the nearest prototype vector. This partitions the space into $\sum_{i=1}^{C} n_i$ regions known as the Dirichlet tessellation of the space. When each pattern in the training set is taken as a prototype vector, then we have the nearest-neighbour decision rule of Chapter 4. This discriminant function generates a piecewise linear decision boundary (Figure 1.13).

Rather than using the complete design set as prototypes, we may use a subset. Methods of reducing the number of prototype vectors (edit and condense) are described in Chapter 4, along with the nearest-neighbour algorithm. Clustering schemes may also be employed.

1.6.4 Generalised linear discriminant function

A *generalised linear discriminant function*, also termed a *phi machine* (Nilsson, 1965), is a discriminant function of the form

$$g(\boldsymbol{x}) = \boldsymbol{w}^T \boldsymbol{\phi} + w_0$$

where $\boldsymbol{\phi} = (\phi_1(\boldsymbol{x}), \ldots, \boldsymbol{\phi}_D(\boldsymbol{x}))^T$ is a vector function of $\boldsymbol{x}$. If $D = p$, the number of variables, and $\phi_i(\boldsymbol{x}) = x_i$, then we have a linear discriminant function.

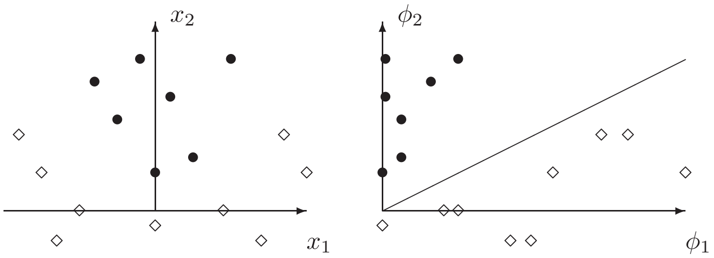

Figure 1.14 Nonlinear transformation of variables may permit linear discrimination.

The discriminant function is linear in the functions ϕ_i, not in the original measurements x_i. As an example, consider the two-class problem of Figure 1.14.

A linear discriminant function will not separate the two classes (denoted by the • and ◇ symbols in the left-hand illustration), even though they are separable. However, if we make the transformation

$$\phi_1(\boldsymbol{x}) = x_1^2$$
$$\phi_2(\boldsymbol{x}) = x_2$$

then the classes can be separated in the ϕ-space by a straight line as shown in the right-hand illustration. Similarly, disjoint classes can be transformed into a ϕ-space in which a linear discriminant function could separate the classes (provided that they are separable in the original space).

The problem, therefore, is simple. Make a good choice for the functions $\phi_i(\boldsymbol{x})$, then use a linear discriminant function to separate the classes. But, how do we choose ϕ_i? Specific examples are shown in Table 1.1.

Table 1.1 Discriminant functions, ϕ.

Discriminant function	Mathematical form, $\phi_i(\boldsymbol{x})$
Linear	$\phi_i(\boldsymbol{x}) = x_i,\ i = 1, \ldots, p$
Quadratic	$\phi_i(\boldsymbol{x}) = x_{k_1}^{l_1} x_{k_2}^{l_2},\ i = 1, \ldots, (p+1)(p+2)/2 - 1$ $l_1, l_2 = 0 \text{ or } 1;\ k_1, k_2 = 1, \ldots, p$ l_1, l_2 not both zero
νth order polynomial	$\phi_i(\boldsymbol{x}) = x_{k_1}^{l_1} \ldots x_{k_\nu}^{l_\nu},\ i = 1, \ldots, \binom{p+\nu}{\nu} - 1$ $l_1, \ldots, l_\nu = 0 \text{ or } 1;\ k_1, \ldots, k_\nu = 1, \ldots, p$ l_i not all zero
Radial basis function	$\phi_i(\boldsymbol{x}) = \phi(\lvert\boldsymbol{x} - \boldsymbol{v}_i\rvert)$ for centre $\boldsymbol{v}_i$ and function ϕ
Multilayer perceptron	$\phi_i(\boldsymbol{x}) = f(\boldsymbol{x}^T \boldsymbol{v}_i + v_{i0})$ for direction $\boldsymbol{v}_i$ and offset v_{i0}. f is the logistic function, $f(z) = 1/(1 + \exp(-z))$

Clearly there is a problem in that the more functions that are used as a basis set, then the more parameters that must be determined using the limited training set. A complete quadratic discriminant function requires $D = (p + 1)(p + 2)/2$ terms and so for C classes there are $C(p+1)(p+2)/2$ parameters to estimate. We may need to apply a constraint or to 'regularise' the model to ensure that there is no overfitting.

An alternative to having a set of different functions is to have a set of functions of the same parametric form, but which differ in the values of the parameters they take,

$$\phi_i(\boldsymbol{x}) = \phi(\boldsymbol{x}; \boldsymbol{v}_i)$$

where $\boldsymbol{v}_i$ is a set of parameters. Different models arise depending on the way the variable $\boldsymbol{x}$ and the parameters $\boldsymbol{v}$ are combined. If

$$\phi(\boldsymbol{x}; \boldsymbol{v}) = \phi(|\boldsymbol{x} - \boldsymbol{v}|)$$

that is, ϕ is a function only of the magnitude of the difference between the pattern $\boldsymbol{x}$ and the weight vector $\boldsymbol{v}$, then the resulting discriminant function is known as a *radial basis function*. On the other hand, if ϕ is a function of the scalar product of the two vectors

$$\phi(\boldsymbol{x}; \boldsymbol{v}) = \phi(\boldsymbol{x}^T\boldsymbol{v} + v_0)$$

then the discriminant function is known as a *multilayer perceptron*. It is also a model known as projection pursuit. Both the radial basis function and the multilayer perceptron models can be used in regression.

In these latter examples, the discriminant function is no longer linear in the parameters. Specific forms for ϕ for radial basis functions and for the multilayer perceptron models will be given in Chapter 6.

1.6.5 Summary

In a multiclass problem, a pattern $\boldsymbol{x}$ is assigned to the class for which the discriminant function is the largest. A linear discriminant function divides the feature space by a hyperplane whose orientation is determined by the weight vector $\boldsymbol{w}$ and distance from the origin by the weight threshold w_0. The decision regions produced by linear discriminant functions are convex.

A piecewise linear discriminant function permits nonconvex and disjoint decision regions. Special cases are the nearest-neighbour and nearest class mean classifier.

A generalized linear discriminant function, with fixed functions ϕ_i, is linear in its parameters. It permits nonconvex and multiply connected decision regions (for suitable choices of ϕ_i). Radial basis functions and multilayer perceptrons can be regarded as generalised linear discriminant functions with flexible functions ϕ_i whose parameters must be determined or specified using the training set.

The Bayes' decision rule is optimal (in the sense of minimising classification error) and with sufficient flexibility in our discriminant functions we ought to be able to achieve optimal performance in principle. However, we are limited by a finite number of training samples and also, once we start to consider parametric forms for the ϕ_i, we lose the simplicity and ease of computation of the linear functions.

1.7 Multiple regression

Many of the techniques and procedures described within this book are also relevant to problems in *regression*, the process of investigating the relationship between a dependent (or response) variable Y and predictor (also referred to as regressor, measurement and independent) variables $X_1, \dots, X_p$; a regression function expresses the expected value of Y in terms of $X_1, \dots, X_p$ and model parameters. Regression is an important part of statistical pattern recognition and, although the emphasis of the book is on discrimination, practical illustrations are sometimes given on problems of a regression nature.

The discrimination problem itself is one in which we are attempting to predict the values of one variable (the class variable) given measurements made on a set of predictor variables (the pattern vector, $\boldsymbol{x}$). In this case, the response variable is categorical.

Regression analysis is concerned with predicting the mean value of the response variable given measurements on the predictor variables and assumes a model of the form,

$$E[y|\boldsymbol{x}] \stackrel{\triangle}{=} \int y p(y|\boldsymbol{x}) dy = f(\boldsymbol{x}; \boldsymbol{\theta})$$

where f is a (possibly nonlinear) function of the measurements $\boldsymbol{x}$ and $\boldsymbol{\theta}$, a set of parameters of f. For example,

$$f(\boldsymbol{x}; \boldsymbol{\theta}) = \theta_0 + \boldsymbol{\theta}^T \boldsymbol{x},$$

where $\boldsymbol{\theta} = (\theta_1, \dots, \theta_p)^T$, is a model that is linear in the parameters and the variables. The model

$$f(\boldsymbol{x}; \boldsymbol{\theta}) = \theta_0 + \boldsymbol{\theta}^T \boldsymbol{\phi}(\boldsymbol{x}),$$

where $\boldsymbol{\theta} = (\theta_1, \dots, \theta_D)^T$ and $\boldsymbol{\phi} = (\phi_1(\boldsymbol{x}), \dots, \phi_D(\boldsymbol{x}))^T$ is a vector of nonlinear functions of $\boldsymbol{x}$, is linear in the parameters but nonlinear in the variables. *Linear regression* refers to a regression model that is linear in the parameters, but not necessarily the variables.

Figure 1.15 shows an illustrative regression summary. For each value of x, there is a population of y values that varies with x. The solid line connecting the conditional means, $E[y|x]$, is the *regression line*. The dotted lines either side represent the spread of the conditional distribution (±1 standard deviation from the mean).

It is assumed that the difference (commonly referred to as an error or residual), ϵ_i, between the measurement on the response variable and its predicted value conditional on the measurements on the predictors,

$$\epsilon_i = y_i - E[y|\boldsymbol{x}_i]$$

is an unobservable random variable. A normal model for the errors is often assumed,

$$p(\epsilon) = \frac{1}{\sqrt{(2\pi)}\sigma} \exp\left(-\frac{1}{2}\frac{\epsilon^2}{\sigma^2}\right)$$

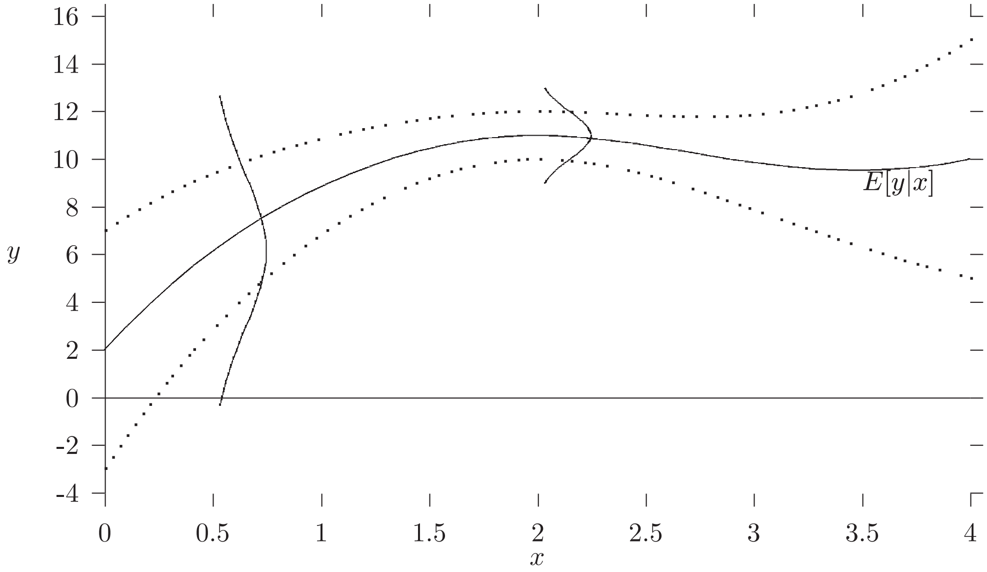

Figure 1.15 Population regression line (solid line) with representation of spread of conditional distribution (dotted lines) for normally distributed error terms, with variance depending on x.

That is,

$$p(y_i|\boldsymbol{x}_i, \boldsymbol{\theta}) = \frac{1}{\sqrt{(2\pi)}\sigma} \exp\left(-\frac{1}{2\sigma^2}(y_i - f(\boldsymbol{x}_i; \boldsymbol{\theta}))^2\right)$$

Given a set of data $\{(y_i, \boldsymbol{x}_i), i = 1, \ldots, n\}$, the maximum likelihood estimate of the model parameters (the value of the parameters for which the data are ‘most likely’), $\boldsymbol{\theta}$, is that for which

$$p(\{(y_i, \boldsymbol{x}_i)\}|\boldsymbol{\theta})$$

is a maximum. Assuming independent samples, this amounts to determining the value of $\boldsymbol{\theta}$ for which the commonly used least squares error,

$$\sum_{i=1}^{n}(y_i - f(\boldsymbol{x}_i; \boldsymbol{\theta}))^2 \tag{1.21}$$

is a minimum (see the exercises at the end of the chapter).

For the linear model, procedures for estimating the parameters are described in Chapter 5.

1.8 Outline of book

The aim of this book is to provide a comprehensive account of statistical pattern recognition techniques with emphasis on methods and algorithms for discrimination and classification. In recent years there have been many developments in multivariate analysis techniques, particularly in nonparametric methods for discrimination and classification including kernel methods, but also the use of pattern recognition techniques applied to complex datasets such as those represented by a network. These are described in this book as extensions to the basic methodology developed over the years.

This chapter has presented some basic approaches to statistical pattern recognition. Supplementary material on probability theory and data analysis can be found on the book's website.

Chapters 2, 3 and 4 describe basic approaches to supervised classification via Bayes' rule and estimation of the class-conditional densities. Chapter 2 considers normal-based models. Chapter 3 develops these models to allow for uncertainty in model parameters. Chapter 4 addresses nonparametric approaches to density estimation.

Chapters 5–7 take a discriminant function approach to supervised classification. Chapter 5 describes algorithms for linear discriminant functions. Chapter 6 considers kernel-based approaches for constructing nonlinear discriminant functions, namely radial basis functions and support vector machine methods and alternative, projection-based methods, the multilayer perceptron neural network. Chapter 7 describes approaches that result in interpretable rules, often required for some applications to provide insight into the classification process.

Chapter 8 introduces the concept of classifier combination: can improvement be achieved with an ensemble of classifiers? Some classifiers may perform well in one part of the data space, other classifiers in another part. How should they be combined?

Chapter 9 addresses the important topic of performance assessment: how good is your designed classifier and how well does it compare with competing techniques?

Chapters 10 and 11 consider techniques that may form part of an exploratory data analysis. Chapter 10 describes methods of feature selection and extraction, both linear and nonlinear. Chapter 11 addresses *unsupervised* classification or clustering. Chapter 12 considers datasets that may be represented as complex networks. Many of the techniques employed for the analysis of such datasets are part of the pattern recognition literature presented in this book.

Finally, Chapter 13 covers additional topics on pattern recognition including model selection.

1.9 Notes and references

There was a growth of interest in techniques for automatic pattern recognition in the 1960s. Many books appeared in the early 1970s, some of which are still very relevant today and have been revised and reissued. More recently, there have been books detailing developments in pattern recognition, particularly neural network methods and kernel methods.

A very good introduction is provided by the book of Hand (1981a). Perhaps a little out of date now, it provides nevertheless a very readable account of techniques for discrimination and classification written from a statistical point of view and is to be recommended. Two of the main textbooks on statistical pattern recognition are those by Fukunaga (1990) and

Devijver and Kittler (1982). Written with an engineering emphasis, Fukunaga's book provides a comprehensive account of the most important aspects of pattern recognition, with many examples, computer projects and problems. Devijver and Kittler's book covers the nearest-neighbour decision rule and feature selection and extraction in some detail, though not at the expense of other important areas of statistical pattern recognition. It contains detailed mathematical accounts of techniques and algorithms, treating some areas in depth.

Another important textbook is that by Duda *et al*. (2001). This presents a thorough account of the main topics in pattern recognition. Other books that are an important source of reference material are those by Young and Calvert (1974), Tou and Gonzales (1974) and Chen (1973). Also, good accounts are given by Andrews (1972), a more mathematical treatment, and Therrien (1989), an undergraduate text.

Books that describe the 'neural network' aspects of developments in pattern recognition and their relationship to the more traditional methods include those of Haykin (1994), who provides a comprehensive treatment of neural networks, and Bishop (1995) who provides an excellent introduction to neural network methods from a statistical pattern recognition perspective. Ripley's (1996) account provides a thorough description of pattern recognition from within a statistical framework. It includes neural network methods, approaches developed in the field of machine learning, advances in statistical techniques as well as development of more traditional pattern recognition methods and gives valuable insights into many techniques gained from practical experience. Hastie *et al*. (2001) provide a thorough description of modern techniques in pattern recognition. Other books that deserve a mention are those by Schalkoff (1992) and Pao (1989).

Bishop (2007) provides an excellent introduction to pattern recognition, particularly recent developments and details of Bayesian computational methods.

The treatment of pattern recognition by Theodoridis and Koutroumbas (2009) is a comprehensive account, with similar goals to this book but with greater emphasis on unsupervised methods. Each chapter is supported by MATLAB code [see also the books by Nabney (2001), Theodoridis *et al*. (2010) and van der Heiden *et al*. (2004)].

Hand (1997) gives a short introduction to pattern recognition techniques and the central ideas in discrimination but places greater emphasis on the comparison and assessment of classifiers.

A more specialised treatment of discriminant analysis and pattern recognition is the book by McLachlan (1992a). This is a very good book. It is not an introductory textbook, but provides a thorough account of developments in discriminant analysis. Written from a statistical perspective, the book is a valuable source of reference of theoretical and practical work on statistical pattern recognition and is to be recommended for researchers in the field.

Comparative treatments of pattern recognition techniques (statistical, neural and machine learning methods) are provided in the volume edited by Michie *et al*. (1994) who report on the outcome of the *Statlog* project. Technical descriptions of the methods are given, together with the results of applying those techniques to a wide range of problems. This volume provides the most extensive comparative study available. More than 20 different classification procedures were considered for about 20 datasets.

Books on data mining often give good treatments of pattern recognition, including both supervised and unsupervised classification (Tan *et al*., 2005; Witten and Frank, 2005; Han and Kamber, 2006).

There are many other books on pattern recognition. Some of those treating more specific parts (such as clustering) are cited in the appropriate chapters of this book. In addition, most textbooks on multivariate analysis devote some attention to discrimination and classification. These provide a valuable source of reference and are cited elsewhere in the book. There are also pattern recognition books for specialist applications, for example, medical imaging (Meyer-Baese, 2003) and forensics (Keppel *et al*., 2006).

Exercises

In some of the exercises, it will be necessary to generate samples from a multivariate normal density with mean $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$, denoted $N(\boldsymbol{\mu}, \boldsymbol{\Sigma})$. Many computer packages offer routines for this. However, it is a simple matter to generate samples from a normal distribution with unit variance and zero mean (Press *et al*., 1992). Given a vector $\boldsymbol{Y}_i$ of such samples, then the vector $\boldsymbol{U}\boldsymbol{\Lambda}^{1/2}\boldsymbol{Y}_i + \boldsymbol{\mu}$ has the required distribution, where $\boldsymbol{U}$ is the matrix of eigenvectors of the covariance matrix and $\boldsymbol{\Lambda}^{1/2}$ is a diagonal matrix whose diagonal elements are the square roots of the corresponding covariance matrix eigenvalues.

1. Consider two multivariate normally distributed classes,

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

with means $\boldsymbol{\mu}_1$ and $\boldsymbol{\mu}_2$ and equal covariance matrices, $\boldsymbol{\Sigma}_1 = \boldsymbol{\Sigma}_2 = \boldsymbol{\Sigma}$. Show that the logarithm of the likelihood ratio is linear in the feature vector $\boldsymbol{x}$. What is the equation of the decision boundary?

2. Determine the equation of the decision boundary for the more general case of $\boldsymbol{\Sigma}_1 = \alpha\boldsymbol{\Sigma}_2$, for scalar α (normally distributed classes as in Exercise 1). In particular, for two univariate distributions, $N(0, 1)$ and $N(1, 1/4)$, show that one of the decision regions is bounded and determine its extent.

3. For the distributions in Exercise 1, determine the equation of the minimum risk decision boundary for the loss matrix,

$$\boldsymbol{\Lambda} = \begin{pmatrix} 0 & 2 \\ 1 & 0 \end{pmatrix}$$

4. Consider two multivariate normally distributed classes [ω_2 with mean $(-1, 0)^T$ and ω_1 with mean $(1, 0)^T$, and identity covariance matrix]. For a given threshold μ [see Equation (1.14)] on the likelihood ratio, determine the regions Ω_1 and Ω_2 in a Neyman–Pearson rule.

5. Consider three bivariate normal distributions, $\omega_1, \omega_2, \omega_3$ with identity covariance matrices and means $(-2, 0)^T$, $(0, 0)^T$ and $(0, 2)^T$. Show that the decision boundaries are piecewise linear. Now define a class, A, being the *mixture* of ω_1 and ω_3,

$$p_A(\boldsymbol{x}) = 0.5p(\boldsymbol{x}|\omega_1) + 0.5p(\boldsymbol{x}|\omega_3)$$

and class B as bivariate normal with identity covariance matrix and mean $(a, b)^T$, for some a, b. What is the equation of the Bayes' decision boundary? Under what conditions is it piecewise linear?

6. Consider two uniform distributions with equal priors

$$p\left(x|\omega_1\right) = \begin{cases} 1 & \text{when } 0 \le x \le 1 \\ 0 & \text{otherwise} \end{cases}$$

$$p\left(x|\omega_2\right) = \begin{cases} \frac{1}{2} & \text{when } \frac{1}{2} \le x \le \frac{5}{2} \\ 0 & \text{otherwise} \end{cases}$$

Show that the reject function is given by

$$r(t) = \begin{cases} \frac{3}{8} & \text{when } 0 \le t \le \frac{1}{3} \\ 0 & \text{when } \frac{1}{3} \le t \le 1 \end{cases}$$

Hence calculate the error rate by integrating (1.10).

7. Reject option. Consider two classes, each normally distributed with means $x = 1$ and $x = -1$ and unit variances; $p(\omega_1) = p(\omega_2) = 0.5$. Generate a test set and use it (without using class labels) to estimate the reject rate as a function of the threshold t. Hence, estimate the error rate for no rejection. Compare with the estimate based on a labelled version of the test set. Comment on the use of this procedure when the true distributions are unknown and the densities have to be estimated.

8. The area of a sphere of radius r in p dimensions, S_p, is

$$S_p = \frac{2\pi^{\frac{p}{2}} r^{p-1}}{\Gamma(p/2)}$$

where Γ is the gamma function $[\Gamma(1/2) = \pi^{1/2}, \Gamma(1) = 1, \Gamma(x+1) = x\Gamma(x)]$. Show that the probability of a sample, $\boldsymbol{x}$, drawn from a zero-mean normal distribution with covariance matrix $\sigma^2\boldsymbol{I}$ ($\boldsymbol{I}$ is the identity matrix) and having $|\boldsymbol{x}| \le R$ is

$$\int_0^R S_p(r) \frac{1}{(2\pi\sigma^2)^{p/2}} \exp\left(-\frac{r^2}{2\sigma^2}\right) dr$$

Evaluate this numerically for $R = 2\sigma$ and for $p = 1, \ldots, 10$. What do the results tell you about the distribution of normal samples in high-dimensional spaces?

9. In a two-class problem, let the cost of misclassifying a class ω_1 pattern be C_1 and the cost of misclassifying a class ω_2 pattern be C_2. Show that the point on the ROC curve that minimises the risk has gradient

$$\frac{C_2 p(\omega_2)}{C_1 p(\omega_1)}$$

10. Show that under the assumption of normally distributed residuals, the maximum likelihood solution for the parameters of a linear model is equivalent to minimising the sum-square error (1.21).