CHAPTER

Feature Selection

5

5.1 INTRODUCTION

In all previous chapters, we considered the features that should be available prior to the design of the classifier. The goal of this chapter is to study methodologies related to the selection of these variables. As we pointed out very early in the book, a major problem associated with pattern recognition is the so-called curse of dimensionality (Section 2.5.6). The number of features at the disposal of the designer of a classification system is usually very large. As we will see in Chapter 7, this number can easily reach the order of a few dozens or even hundreds.

There is more than one reason to reduce the number of features to a sufficient minimum. Computational complexity is the obvious one. A related reason is that, although two features may carry good classification information when treated separately, there is little gain if they are combined into a feature vector because of a high mutual correlation. Thus, complexity increases without much gain. Another major reason is that imposed by the required generalization properties of the classifier, as discussed in Section 4.9 of Chapter 4. As we will state more formally at the end of this chapter, the higher the ratio of the number of training patterns N to the number of free classifier parameters, the better the generalization properties of the resulting classifier.

A large number of features are directly translated into a large number of classifier parameters (e.g., synaptic weights in a neural network, weights in a linear classifier). Thus, for a finite and usually limited number N of training patterns, keeping the number of features as small as possible is in line with our desire to design classifiers with good generalization capabilities. Furthermore, the ratio N/l enters the scene from another nearby corner. One important step in the design of a classification system is the performance evaluation stage, in which the classification error probability of the designed classifier is estimated. We not only need to design a classification system, but we must also assess its performance. As is pointed out in Chapter 10, the classification error estimate improves as this ratio becomes higher. In [Fine 83] it is pointed out that in some cases ratios as high as 10 to 20 were considered necessary.

The major task of this chapter can now be summarized as follows. *Given a number of features, how can one select the most important of them so as to reduce their number and at the same time retain as much as possible of their class discriminatory information*? The procedure is known as *feature selection* or *reduction*. This step is very crucial. If we selected features with little discrimination power, the subsequent design of a classifier would lead to poor performance. On the other hand, if information-rich features are selected, the design of the classifier can be greatly simplified. In a more quantitative description, we should aim to select features leading to *large between-class distance and small within-class variance* in the feature vector space. This means that features should take distant values in the different classes and closely located values in the same class. To this end, different scenarios will be adopted. One is to examine the features individually and discard those with little discriminatory capability. A better alternative is to examine them in combinations. Sometimes the application of a linear or nonlinear transformation to a feature vector may lead to a new one with better discriminatory properties. All these paths will be our touring directions in this chapter.

Finally, it must be pointed out that there is some confusion in the literature concerning the terminology of this stage. In some texts the term *feature extraction* is used, but we feel that this may be confused with the feature generation stage treated in Chapter 7. Others prefer to call it a *preprocessing stage*. We have kept the latter term to describe the processing performed on the features prior to their utilization. Such processing involves removing outliers, scaling of the features to safeguard comparable dynamic range of their respective values, treating missing data, and so forth.

5.2 PREPROCESSING

5.2.1 Outlier Removal

An *outlier* is defined as a point that lies very far from the mean of the corresponding random variable. This distance is measured with respect to a given threshold, usually a number of times the standard deviation. For a normally distributed random variable, a distance of two times the standard deviation covers 95% of the points, and a distance of three times the standard deviation covers 99% of the points. Points with values very different from the mean value produce large errors during training and may have disastrous effects. These effects are even worse when the outliers are the result of noisy measurements. If the number of outliers is very small, they are usually discarded. However, if this is not the case and they are the result of a distribution with long tails, then the designer may have to adopt cost functions that are not very sensitive in the presence of outliers. For example, the least squares criterion is very sensitive to outliers, because large errors dominate the cost function due to the squaring of the terms. A review of related techniques that attempt to address such problems is given in [Hube 81].

5.2.2 Data Normalization

In many practical situations a designer is confronted with features whose values lie within different dynamic ranges. Thus, features with large values may have a larger influence in the cost function than features with small values, although *this does not necessarily reflect their respective significance in the design of the classifier*. The problem is overcome by normalizing the features so that their values lie within similar ranges. A straightforward technique is normalization via the respective estimates of the mean and variance. For N available data of the kth feature we have

$$\bar{x}_k = \frac{1}{N}\sum_{i=1}^{N} x_{ik}, \quad k = 1, 2, \ldots, l$$

$$\sigma_k^2 = \frac{1}{N-1}\sum_{i=1}^{N}(x_{ik} - \bar{x}_k)^2$$

$$\hat{x}_{ik} = \frac{x_{ik} - \bar{x}_k}{\sigma_k}$$

In words, all the resulting normalized features will now have zero mean and unit variance. This is obviously a linear method. Other linear techniques limit the feature values in the range of $[0, 1]$ or $[-1, 1]$ by proper scaling. Besides the linear methods, nonlinear methods can also be employed in cases in which the data are not evenly distributed around the mean. In such cases transformations based on nonlinear (i.e., logarithmic or sigmoid) functions can be used to map data within specified intervals. The so-called softmax scaling is a popular candidate. It consists of two steps

$$y = \frac{x_{ik} - \bar{x}_k}{r\sigma_k}, \quad \hat{x}_{ik} = \frac{1}{1 + \exp(-y)} \tag{5.1}$$

This is basically a squashing function limiting data in the range of $[0, 1]$. Using a series expansion approximation, it is not difficult to see that for small values of y this is an approximately linear function with respect to x_{ik}. The range of values of x_{ik} that correspond to the linear section depends on the standard deviation and the factor r, which is user defined. Values away from the mean are squashed exponentially.

5.2.3 Missing Data

In practice, certain features may be missing from some feature vectors. Such incomplete-data cases are common in social sciences due, for example, to partial response in surveys. Remote sensing is another area where incomplete-data may occur when certain regions are covered by a subset of sensors. Sensor networks, which rely on a set of distributed information sources and on the data fusion from a number of sensors, is also a discipline where incomplete-data may arise.

The most traditional techniques in dealing with missing data include schemes that "complete" the missing values by (a) zeros or (b) the unconditional mean, computed from the available values of the respective feature or (c) the conditional

mean, if one has an estimate of the pdf of the missing values given the observed data. Completing the missing values in a set of data is also known as *imputation*. Another approach is to discard feature vectors with missing values. Although such a technique can be useful in cases of large data sets, in most cases it is considered a "luxury" to afford to drop available information.

Since the mid-1970s ([Rubi 76]), a large research effort has been invested to cope efficiently with the missing data task, and a number of sophisticated methods have been developed and used successfully. A popular alternative to the previously exposed, rather naive approaches is known as *imputing from a conditional distribution*. The idea here is to impute by respecting the statistical nature of the missing values. Under this rationale, missing values are not replaced by statistical means or zeros but by a random draw from a distribution. Let us denote the complete feature vector as $\boldsymbol{x}_{com}$ and assume that some of its components are missing ($\boldsymbol{x}_{mis}$) and the rest are observed ($\boldsymbol{x}_{obs}$). Then the complete feature vector is written as

$$\boldsymbol{x}_{com} = \begin{bmatrix} \boldsymbol{x}_{obs} \\ \boldsymbol{x}_{mis} \end{bmatrix}$$

Under the assumption that the probability of missing a value does not depend on the value itself (this is known as the *missing at random* (MAR) assumption), imputing from a conditional distribution means to simulate a draw from the following conditional pdf

$$p(\boldsymbol{x}_{mis}|\boldsymbol{x}_{obs};\boldsymbol{\theta}) = \frac{p(\boldsymbol{x}_{obs},\boldsymbol{x}_{mis};\boldsymbol{\theta})}{p(\boldsymbol{x}_{obs};\boldsymbol{\theta})} \tag{5.2}$$

where

$$p(\boldsymbol{x}_{obs};\boldsymbol{\theta}) = \int p(\boldsymbol{x}_{com};\boldsymbol{\theta}) d\boldsymbol{x}_{mis} \tag{5.3}$$

where $\boldsymbol{\theta}$ is an unknown set of parameters. In practice, an estimate $\hat{\boldsymbol{\theta}}$ of $\boldsymbol{\theta}$ must first be obtained from $\boldsymbol{x}_{obs}$. The celebrated EM algorithms (Chapter 2) is a popular choice for parameter estimation under the missing data setting, for example, [Ghah 94, Tsud 03].

The *multiple imputation* (MI) procedure ([Rubi 87]) is one step beyond the previous methodology, usually referred to as single imputation (SI). In MI, for each missing value, $m > 1$ samples are generated. The results are then combined appropriately so that certain statistical properties are fulfilled. MI can be justified as an attempt to overcome uncertainties associated with the estimation of the parameter $\boldsymbol{\theta}$. Hence, instead of drawing a single point from $p(\boldsymbol{x}_{mis}|\boldsymbol{x}_{obs};\hat{\boldsymbol{\theta}})$ one can use different parameters, $\hat{\boldsymbol{\theta}}_i,\ i = 1, 2, \ldots, m$, and draw the m samples from

$$p(\boldsymbol{x}_{mis}|\boldsymbol{x}_{obs};\hat{\boldsymbol{\theta}}_i),\ i = 1, 2, \ldots, m$$

A way to approach this problem is via Bayesian inference arguments (Chapter 2), where the unknown parameter vector is treated as a random one described by a posterior probability, see, for example, [Gelm 95].

In a more recent paper ([Will 07]), the missing data problem is treated in the context of logistic regression classification (Section 3.6) and explicit imputation is bypassed. This is achieved by integrating out the missing values and predicting the binary class label, y_i, of the ith pattern based on the value of

$$P(y_i|\boldsymbol{x}_{i,obs}) = \int P(y_i|\boldsymbol{x}_{i,obs}, \boldsymbol{x}_{i,mis}) p(\boldsymbol{x}_{i,mis}|\boldsymbol{x}_{i,obs}) d\boldsymbol{x}_{i,mis}$$

Under the assumption that $p(\boldsymbol{x}_{i,mis}|\boldsymbol{x}_{i,obs})$ is sufficiently modeled by a Gaussian mixture model (Section 2.5.5) the previous integration can be performed analytically. For more on the problem of missing data the interested reader may refer to the excellent review article [Scha 02]. We will return to the missing data problem in Chapter 11.

5.3 THE PEAKING PHENOMENON

As stated in the introduction of this chapter, in order to design a classifier with good generalization performance, the number of training points, N, must be large enough with respect to the number of features, l, that is, the dimensionality of the feature space. Take as an example the case of designing a linear classifier, $\boldsymbol{w}^T\boldsymbol{x} + w_0$. The number of the unknown parameters is $l + 1$. In order to get a good estimate of these parameters, the number of data points must be larger than $l + 1$. The larger the N the better the estimate, since we can filter out the effects of the noise and also minimize the effects of the outliers.

In [Trun 79], an elegant simple example has been given that reveals the interplay between the number of features and the size of the training data set and elucidates the way these two parameters influence the performance of a classifier. Consider a two-class classification task with equal prior probabilities, $P(\omega_1) = P(\omega_2) = \frac{1}{2}$, in the l-dimensional space. Both classes, ω_1, ω_2, are described by Gaussian distributions of the same covariance matrix $\Sigma = I$, where I is the identity matrix and mean values $\boldsymbol{\mu}$ and $-\boldsymbol{\mu}$, respectively, where

$$\boldsymbol{\mu} = \left[1, \frac{1}{\sqrt{2}}, \frac{1}{\sqrt{3}}, \ldots, \frac{1}{\sqrt{l}} \right]^T \tag{5.4}$$

Since the features are jointly Gaussian and $\Sigma = I$, the involved features are statistically independent (see Appendix A.9). Moreover, the optimal Bayesian rule is equivalent to the minimum Euclidean distance classifier. Given an unknown feature vector $\boldsymbol{x}$, we classify it to, say, ω_1 if

$$||\boldsymbol{x} - \boldsymbol{\mu}||^2 < ||\boldsymbol{x} + \boldsymbol{\mu}||^2$$

or after performing the algebra, if

$$z \equiv \boldsymbol{x}^T\boldsymbol{\mu} > 0$$

If $z < 0$, we decide in favor of the class ω_2. Thus, the decision relies on the value of the inner product z. In the sequel we will consider two cases.

Known Mean Value $\boldsymbol{\mu}$

The inner product z, being a linear combination of independent Gaussian variables, is also a Gaussian variable (see, e.g., [Papo 91]) with mean value $E[z] = ||\boldsymbol{\mu}||^2 = \sum_{i=1}^{l} \frac{1}{i}$ and variance $\sigma_z^2 = ||\boldsymbol{\mu}||^2$ (Problem 5.1). The probability of committing an error turns out to be equal to (Problem 5.1)

$$P_e = \int_{b_l}^{\infty} \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{z^2}{2}\right) dz \tag{5.5}$$

where

$$b_l = \sqrt{\sum_{i=1}^{l} \frac{1}{i}} \tag{5.6}$$

Note that the series in (5.6) tends to infinity as $l \longrightarrow \infty$; hence the probability of error tends to zero as the number of features increases.

Unknown mean value $\boldsymbol{\mu}$

In this case, the mean value has to be estimated from the training data set. Adopting the maximum likelihood estimate we obtain

$$\hat{\boldsymbol{\mu}} = \frac{1}{N} \sum_{k=1}^{N} s_k \boldsymbol{x}_k$$

where $s_k = 1$ if $\boldsymbol{x}_k \in \omega_1$ and $s_k = -1$ if $\boldsymbol{x}_k \in \omega_2$. Decisions are taken depending on the inner product $z = \boldsymbol{x}^T \hat{\boldsymbol{\mu}}$. However, z is no more a Gaussian variable, since $\hat{\boldsymbol{\mu}}$ is not a constant but a random vector. By the definition of the inner product, $z = \sum_{i=1}^{l} x_i \hat{\boldsymbol{\mu}}_i$ and for large enough l and the central limit theorem (Appendix A) z can be considered approximately Gaussian. Its mean value and variance are given by (Problem 5.2)

$$E[z] = \sum_{i=1}^{l} \frac{1}{i} \tag{5.7}$$

and

$$\sigma_z^2 = \left(1 + \frac{1}{N}\right) \sum_{i=1}^{l} \frac{1}{i} + \frac{l}{N} \tag{5.8}$$

The probability of error is given by (5.5) with

$$b_l = \frac{E[z]}{\sigma_z} \tag{5.9}$$

It can now be shown that $b_l \longrightarrow 0$ as $l \longrightarrow \infty$ and the probability of error tends to $\frac{1}{2}$ for any finite number N (Problem 5.2).

The above example demonstrates that:

- If for any l the corresponding pdf is known, then we can perfectly discriminate the two classes by arbitrarily increasing the number of features.
- If the pdfs are not known and the associated parameters must be estimated using a finite training set, then the arbitrary increase of the number of features leads to the maximum possible value of the error rate, that is, $P_e = 0.5$. This implies that under a limited number of training data we must try to keep the number of features to a relatively low number.

In practice, for a finite N, by increasing the number of features one obtains an initial improvement in performance, but after a critical value further increase of the number of features results in an increase of the probability of error. This phenomenon is also known as the *peaking phenomenon*. Figure 5.1 illustrates the general trend that one expects to experience in practice by playing with the number of features, l, and the size of the training data set, N. For $N_2 >> N_1$, the error values corresponding to N_2 are lower than those resulting for N_1, and the peaking phenomenon occurs for a value $l_2 > l_1$. For each value of N, the probability of error starts decreasing with increasing l till a critical value where the error starts increasing. The minimum in the curves occurs at some number $l = \frac{N}{\alpha}$, where α, usually, takes values in the range of 2 to 10. Consequently, in practice, for a small number of training data, a small number of features must be used. If a large number of training data is available, a larger number of features can be selected that yield better performance.

Although the above scenario covers a large set of "traditional" classifiers, it is not always valid. We have already seen that adopting an appropriate kernel function to design a nonlinear SVM classifier implies a mapping to a high-dimensional space,

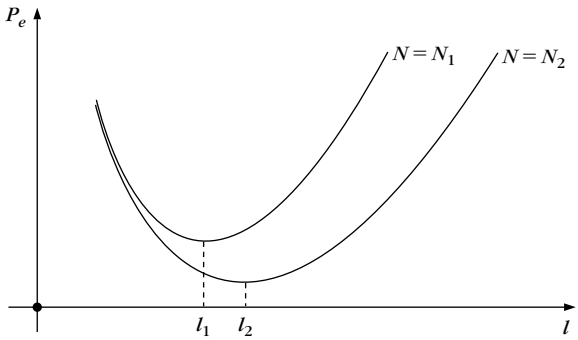

FIGURE 5.1

For a given value of N, the probability of error decreases as the number of features increases till a critical value. Further increase in the number of features forces the error probability to increase. If the number of points increases, $N_2 >> N_1$, the peaking phenomenon occurs for larger values, $l_2 > l_1$.

which can even be of infinite dimension. In spite of the fact that one now works in almost empty spaces (N is much less than the dimensionality of the space), the generalization performance of the SVM classifiers can be very good. The secret to that was disclosed to us fairly recently. It is not the number of parameters that really controls the generalization performance, under finite N, but another quantity. For some types of classifiers, this quantity is directly related to the number of parameters to be estimated and the dimensionality of the feature space. However, for some classifiers, such as the SVM, this quantity can be controlled independent of the dimensionality of the feature space. These issues are discussed at the end of this chapter. More on the peaking phenomenon and the small sample size problem can be found in for example, [Raud 80, Raud 91, Duin 00].

5.4 FEATURE SELECTION BASED ON STATISTICAL HYPOTHESIS TESTING

A first step in feature selection is to look at each of the generated features *independently* and test their discriminatory capability for the problem at hand. Although looking at the features independently is far from optimal, this procedure helps us to discard easily recognizable "bad" choices and keeps the more elaborate techniques, which we will consider next, from unnecessary computational burden.

Let x be the random variable representing a specific feature. We will try to investigate whether the values it takes for the different classes, say ω_1, ω_2, *differ significantly*. To give an answer to this question, we will formulate the problem in the context of statistical *hypothesis testing*. That is, we will try to answer which of the following hypotheses is correct:

H_1: The values of the feature differ significantly
H_0: The values of the feature do not differ significantly

H_0 is known as the *null hypothesis* and H_1 as the *alternative hypothesis*. The decision is reached on the basis of *experimental evidence* supporting the rejection or not of H_0. This is accomplished by exploiting statistical information, and obviously any decision will be taken subject to an error probability. We will approach the problem by considering the differences of the mean values corresponding to a specific feature in the various classes, and we will test whether these differences are significantly different from zero. Let us first, however, refresh our memory with some basics from the statistics related to hypothesis testing.

5.4.1 Hypothesis Testing Basics

Let x be a random variable with a probability density function, which is assumed to be known within an unknown parameter θ. As we have already seen in Chapter 2,

in the case of a Gaussian, this parameter may be the mean value or its variance. Our interest here lies in the following hypothesis test:

$$H_1 : \theta \neq \theta_0$$

$$H_0 : \theta = \theta_0$$

The decision on this test is reached in the following context. Let $x_i, i = 1, 2, \ldots, N$, be the experimental samples of the random variable x. A function $f(\cdot, \ldots, \cdot)$ is selected, depending on the specific problem, and let $q = f(x_1, x_2, \ldots, x_N)$. The function is selected so that the probability density function of q is easily parameterized in terms of the unknown θ, that is, $p_q(q; \theta)$. Let D be the interval of q in which it has a high probability of lying *under hypothesis* H_0. Let $\bar{D}$ be its complement, that is, the interval of low probability, also under hypothesis H_0. Then, if the value of q that results from the available samples, $x_i, i = 1, 2, \ldots, N$, lies in D we will accept H_0, and if it lies in $\bar{D}$ we will reject it. D is known as the *acceptance interval* and $\bar{D}$ as the *critical interval*. The variable q is known as *test statistic*. The obvious question now refers to the probability of reaching a wrong decision. Let H_0 be true. Then the probability of an error in our decision is $P(q \in \bar{D}|H_0) \equiv \rho$. This probability is obviously the integral of $p_q(q|H_0)$ ($p_q(q; \theta_o)$) over $\bar{D}$ (Figure 5.2). In practice, we preselect this value of ρ, which is known as the *significance level*, and we sometimes denote the corresponding critical (acceptance) interval as $\bar{D}_\rho$ (D_ρ). Let us now apply this procedure in the case in which the unknown parameter is the mean of x.

The Known Variance Case

Let x be a random variable and $x_i, i = 1, 2, \ldots, N$, the resulting experimental samples, which we will assume to be *mutually independent*. Let

$$E[x] = \mu$$

$$E[(x - \mu)^2] = \sigma^2$$

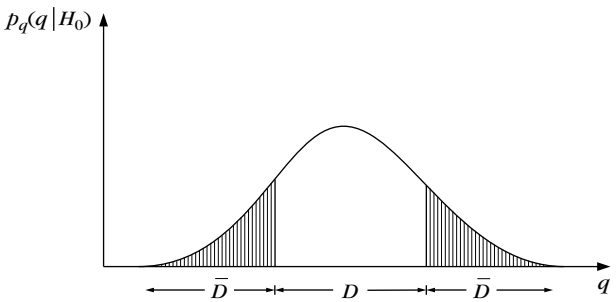

FIGURE 5.2

Acceptance and critical regions for hypothesis testing. The area of the shaded region is the probability of an erroneous decision.

A popular estimate of μ based on the known samples is the *sample mean*

$$\bar{x} = \frac{1}{N}\sum_{i=1}^{N} x_i$$

Using a different set of N samples, a different estimate will result. Thus, $\bar{x}$ is also a random variable, and it is described in terms of a probability density function $p_{\bar{x}}(\bar{x})$. The corresponding mean is

$$E[\bar{x}] = \frac{1}{N}E\left[\sum_{i=1}^{N} x_i\right] = \frac{1}{N}\sum_{i=1}^{N} E[x_i] = \mu \tag{5.10}$$

Thus, $\bar{x}$ is an *unbiased* estimate of the mean μ of x. The variance $\sigma_{\bar{x}}^2$ of $\bar{x}$ is

$$E[(\bar{x}-\mu)^2] = E\left[\left(\frac{1}{N}\sum_{i=1}^{N} x_i - \mu\right)^2\right]$$

$$= \frac{1}{N^2}\sum_{i=1}^{N} E[(x_i-\mu)^2] + \frac{1}{N^2}\sum_{i}\sum_{j\neq i} E[(x_i-\mu)(x_j-\mu)]$$

The statistical independence of the samples dictates

$$E[(x_i-\mu)(x_j-\mu)] = E[x_i-\mu]E[x_j-\mu] = 0$$

Hence

$$\sigma_{\bar{x}}^2 = \frac{1}{N}\sigma^2 \tag{5.11}$$

In words, the larger the number of measurement samples, the smaller the variance of $\bar{x}$ around the true mean μ.

Having now acquired the necessary ingredients, let us assume that we are given a value $\hat{\mu}$ and we have to decide upon

$$H_1: E[x] \neq \hat{\mu}$$

$$H_0: E[x] = \hat{\mu}$$

To this end we define the test statistic

$$q = \frac{\bar{x}-\hat{\mu}}{\sigma/\sqrt{N}} \tag{5.12}$$

Recalling the central limit theorem from Appendix A, the probability density function of $\bar{x}$ *under* H_0 (i.e., given $\hat{\mu}$) is (approximately) the Gaussian $\mathcal{N}(\hat{\mu}, \frac{\sigma^2}{N})$

$$p_{\bar{x}}(\bar{x}) = \frac{\sqrt{N}}{\sqrt{2\pi}\sigma}\exp\left(-\frac{N(\bar{x}-\hat{\mu})^2}{\sigma^2}\right)$$

Table 5.1 Acceptance Intervals $[-x_\rho, x_\rho]$ Corresponding to Various Probabilities for an $\mathcal{N}(0, 1)$ Normal Distribution

$1-\rho$	0.8	0.85	0.9	0.95	0.98	0.99	0.998	0.999
x_ρ	1.282	1.440	1.645	1.967	2.326	2.576	3.090	3.291

Hence, the probability density function of q under H_0 is approximately $\mathcal{N}(0, 1)$. For a significance level ρ the acceptance interval $D \equiv [-x_\rho, x_\rho]$, is chosen as the interval in which the random variable q lies with probability $1-\rho$ (ρ the probability of being in $\bar{D}$). This is readily provided from available tables.

An example for normally distributed $\mathcal{N}(0, 1)$ variables is given in Table 5.1. The decision on the test hypothesis can now be reached by the following steps.

- Given the N experimental samples of x, compute $\bar{x}$ and then q.
- Choose the significance level ρ.
- Compute from the corresponding tables for $\mathcal{N}(0, 1)$ the acceptance interval $D = [-x_\rho, x_\rho]$, corresponding to probability $1-\rho$.
- If $q \in D$ decide H_0, if not decide H_1.

Basically, all we say is that we *expect* the resulting value q to lie in the high-percentage $1-\rho$ interval. If it does not, then we decide that this is because the assumed mean value is not "correct." Of course, the assumed mean value may be correct, but it so happens that the resulting q lies in the least probable area because of the specific set of experimental samples available. In such a case our decision is erroneous, and the probability of committing such an error is ρ.

Example 5.1

Let us consider an experiment with a random variable x of $\sigma = 0.23$, and assume N to be equal to 16 and the resulting $\bar{x}$ equal to 1.35. Adopt the significance level $\rho = 0.05$. We will test if the hypothesis $\hat{\mu} = 1.4$ is true.

From Table 5.1 we have

$$\text{prob}\left\{-1.97 < \frac{\bar{x} - \hat{\mu}}{0.23/4} < 1.97\right\} = 0.95$$

$$\text{prob}\left\{-0.113 < \bar{x} - \hat{\mu} < 0.113\right\} = 0.95$$

Thus, since the value of $\hat{\mu}$, which we have assumed, is in the interval

$$1.237 = 1.35 - 0.113 < \hat{\mu} < 1.35 + 0.113 = 1.463$$

we accept it, as *there is no evidence at the 5% level that the mean value is not equal to* $\hat{\mu}$. The interval [1.237, 1463] is also known as the *confidence interval at the* $1-\rho = 0.95$ *level*.

The Unknown Variance Case

If the variance of x is not known, it must be estimated. The estimate

$$\hat{\sigma}^2 \equiv \frac{1}{N-1}\sum_{i=1}^{N}(x_i - \bar{x})^2 \tag{5.13}$$

is an unbiased estimate of the variance. Indeed,

$$\begin{aligned} E[\hat{\sigma}^2] &= \frac{1}{N-1}\sum_{i=1}^{N} E[(x_i - \bar{x})^2] \\ &= \frac{1}{N-1}\sum_{i=1}^{N} E\left[((x_i - \mu) - (\bar{x} - \mu))^2\right] \\ &= \frac{1}{N-1}\sum_{i=1}^{N}\left(\sigma^2 + \frac{\sigma^2}{N} - 2E[(x_i - \mu)(\bar{x} - \mu)]\right) \end{aligned}$$

Due to the independence of the experimental samples

$$E[(x_i - \mu)(\bar{x} - \mu)] = \frac{1}{N}E\big[(x_i - \mu)\big((x_1 - \mu) + \cdots + (x_N - \mu)\big)\big] = \frac{\sigma^2}{N}$$

Thus,

$$E[\hat{\sigma}^2] = \frac{N}{N-1}\frac{N-1}{N}\sigma^2 = \sigma^2$$

The test statistic is now defined as

$$q = \frac{\bar{x} - \mu}{\hat{\sigma}/\sqrt{N}} \tag{5.14}$$

However, this is no longer a Gaussian variable. Following Appendix A, and if we assume that *x is a Gaussian random variable*, then q is described by the so-called t-distribution with $N-1$ degrees of freedom. Table 5.2 shows the confidence interval $D = [-x_\rho, x_\rho]$ for various significance levels and degrees of freedom of the t-distribution.

Example 5.2

For the case of Example 5.1 let us assume that the estimate of the standard deviation $\hat{\sigma}$ is 0.23. Then, according to Table 5.2 for 15 degrees of freedom ($N = 16$) and significance level $\rho = 0.025$

$$\text{prob}\left\{-2.49 < \frac{\bar{x} - \hat{\mu}}{0.23/4} < 2.49\right\} = 0.975$$

and the confidence interval for $\hat{\mu}$ at the 0.975 level is

$$1.207 < \hat{\mu} < 1.493$$

Table 5.2 Interval Values at Various Significance Levels and Degrees of Freedom for a t-Distribution

Degrees of Freedom	$1-\rho$	0.9	0.95	0.975	0.99	0.995
10		1.81	2.23	2.63	3.17	3.58
11		1.79	2.20	2.59	3.10	3.50
12		1.78	2.18	2.56	3.05	3.43
13		1.77	2.16	2.53	3.01	3.37
14		1.76	2.15	2.51	2.98	3.33
15		1.75	2.13	2.49	2.95	3.29
16		1.75	2.12	2.47	2.92	3.25
17		1.74	2.11	2.46	2.90	3.22
18		1.73	2.10	2.44	2.88	3.20
19		1.73	2.09	2.43	2.86	3.17
20		1.72	2.09	2.42	2.84	3.15

5.4.2 Application of the t-Test in Feature Selection

We will now see how all of this is specialized for the case of feature selection in a classification problem. Our major concern now will be to test, against zero, the difference $\mu_1 - \mu_2$ between the means of the values taken by a feature in two classes. Let $x_i, i = 1, 2, \ldots, N$, be the sample values of the feature in class ω_1 with mean μ_1. Correspondingly, for the other class ω_2 we have $y_i, i = 1, 2, \ldots, N$, with mean μ_2. *Let us now assume that the variance of the feature values is the same in both classes*, $\sigma_1^2 = \sigma_2^2 = \sigma^2$. To decide about the closeness of the two mean values, we will test for the hypothesis

$$\begin{aligned} H_1 &: \Delta\mu = \mu_1 - \mu_2 \neq 0 \\ H_0 &: \Delta\mu = \mu_1 - \mu_2 = 0 \end{aligned} \tag{5.15}$$

To this end, let

$$z = x - y \tag{5.16}$$

where x, y denote the random variables corresponding to the values of the feature in the two classes ω_1, ω_2, respectively, for which *statistical independence* has been assumed. Obviously, $E[z] = \mu_1 - \mu_2$, and due to the independence assumption $\sigma_z^2 = 2\sigma^2$. Following arguments similar to those used before, we now have

$$\bar{z} = \frac{1}{N}\sum_{i=1}^{N}(x_i - y_i) = \bar{x} - \bar{y} \tag{5.17}$$

and for the known variance case $\bar{z}$ follows the normal $\mathcal{N}(\mu_1 - \mu_2, \frac{2\sigma^2}{N})$ distribution for large N. Thus, Table 5.1 can be used to decide about (5.15). If the variance is not known, then we choose the test statistic

$$q = \frac{(\bar{x} - \bar{y}) - (\mu_1 - \mu_2)}{s_z \sqrt{\frac{2}{N}}} \tag{5.18}$$

where

$$s_z^2 = \frac{1}{2N-2}\left(\sum_{i=1}^{N}(x_i - \bar{x})^2 + \sum_{i=1}^{N}(y_i - \bar{y})^2\right)$$

It can be shown that $\frac{s_z^2(2N-2)}{\sigma^2}$ follows a chi-square distribution with $2N-2$ degrees of freedom (Appendix A and Problem 5.3). As is pointed out in Appendix A, if x, y are *normally distributed* variables of the same variance σ^2, then the random variable q turns out to follow the t-distribution with $2N-2$ degrees of freedom. Thus, Table 5.2 has to be adopted for the test. When the available number of samples is not the same in all classes, a slight modification is required (Problem 5.4). Furthermore, in practice the variances may not be the same in the two classes. Sometimes this becomes the object of another hypothesis test, concerning the ratio $F = \frac{\hat{\sigma}_1^2}{\hat{\sigma}_2^2}$, to check whether it is close to unity. It can be shown that F, being the ratio of two chi-square distributed variables, follows the so-called F-distribution and the related tables should be used [Fras 58]. Finally, if the Gaussian assumption about x is not a valid one, other criteria can be used to check the equality hypothesis of the means, such as the Kruskal–Wallis statistic [Walp 78, Fine 83].

Example 5.3

The sample measurements of a feature in two classes are

class ω_1:	3.5	3.7	3.9	4.1	3.4	3.5	4.1	3.8	3.6	3.7
class ω_2:	3.2	3.6	3.1	3.4	3.0	3.4	2.8	3.1	3.3	3.6

The question is to check whether this feature is informative enough. If not, it will be discarded during the selection phase. To this end, we will test whether the values of the feature in the two classes differ significantly. We choose the significance level $\rho = 0.05$.
From the foregoing we have

$$\omega_1 : \bar{x} = 3.73 \quad \hat{\sigma}_1^2 = 0.0601$$

$$\omega_2 : \bar{y} = 3.25 \quad \hat{\sigma}_2^2 = 0.0672$$

For $N = 10$ we have

$$s_z^2 = \frac{1}{2}(\hat{\sigma}_1^2 + \hat{\sigma}_2^2)$$

$$q = \frac{(\bar{x} - \bar{y} - 0)}{s_z\sqrt{\frac{2}{N}}}$$

$$q = 4.25$$

From Table 5.2 and for $20 - 2 = 18$ degrees of freedom and significance level 0.05, we obtain $D = [-2.10, 2.10]$. Since 4.25 lies outside the interval D, we decide in favor of H_1; that is, the mean values differ significantly at the 0.05 level. *Hence, the feature is selected.*

5.5 THE RECEIVER OPERATING CHARACTERISTICS (ROC) CURVE

The hypothesis tests we have presented offer statistical evidence about the difference of the mean values of a single feature in the various classes. However, although this is useful information for *discarding* features, if the corresponding mean values are closely located, this information may not be sufficient to guarantee good discrimination properties of a feature passing the test. The mean values may differ significantly yet the spread around the means may be large enough to blur the class distinction. We will now focus on techniques providing information about the *overlap* between the classes.

Figure 5.3a illustrates an example of two overlapping probability density functions describing the distribution of a feature in two classes, together with a threshold (one pdf has been inverted for illustration purposes). We decide class ω_1 for values on the left of the threshold and class ω_2 for the values on the right. This decision is associated with an error probability, a, of reaching a wrong decision concerning class ω_1 (the probability of a correct decision is $1 - a$). This is equal to the

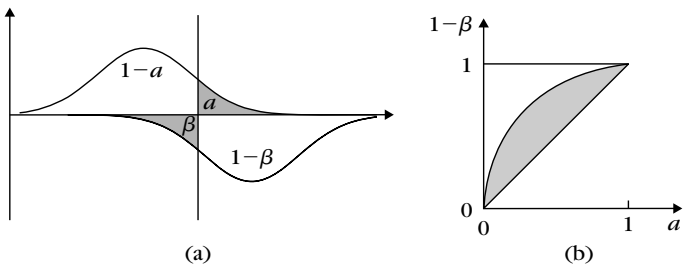

FIGURE 5.3

Example of (a) overlapping pdfs of the same feature in two classes and (b) the resulting ROC curve. The larger the shaded area in (b) the less the overlap of the respective pdfs.

shaded area under the corresponding curve. Similarly, let β ($1-\beta$) be the probability of a wrong (correct) decision concerning class ω_2. By moving the threshold over "all" possible positions, different values of a and β result. It takes little thought to realize that if the two distributions have complete overlap, then for *any* position of the threshold we get $a = 1 - \beta$. Such a case corresponds to the straight line in Figure 5.3b, where the two axes are a and $1-\beta$. As the two distributions move apart, the corresponding curve departs from the straight line, as Figure 5.3b demonstrates. Once more, a little thought reveals that the less the overlap of the classes, the larger the area between the curve and the straight line. At the other extreme of two completely separated class distributions, moving the threshold to sweep the whole range of values for a in $[0, 1]$, $1 - \beta$ remains equal to unity. Thus, the aforementioned area varies between zero, for complete overlap, and 1/2 (the area of the upper triangle), for complete separation, and *it is a measure of the class discrimination capability of the specific feature*. In practice, the ROC curve can easily be constructed by sweeping the threshold and computing percentages of wrong and correct classifications over the available training feature vectors. Other related criteria that test the overlap of the classes have also been suggested (see Problem 5.7).

More recently, the area under the receiver operating characteristics curve (AUC) has been used as an effective criterion to design classifiers. This is because larger AUC values indicate on average better classifier performance, see, for example, [Brad 97, Marr 08, Land 08].

5.6 CLASS SEPARABILITY MEASURES

The emphasis in the previous section was on techniques referring to the discrimination properties of *individual* features. However, such methods neglect to take into account the correlation that unavoidably exists among the various features and influences the classification capabilities of the feature vectors that are formed. Measuring the discrimination effectiveness of feature *vectors* will now become our major concern. This information will then be used in two ways. The first is to allow us to combine features appropriately and end up with the "best" feature vector for a given dimension l. The second is to transform the original data on the basis of an optimality criterion in order to come up with features offering high classification power. In the sequel we will first state *class separability measures*, which will be used subsequently in feature selection procedures.

5.6.1 Divergence

Let us recall our familiar Bayes rule. Given two classes ω_1 and ω_2 and a feature vector $\boldsymbol{x}$, we select ω_1 if

$$P(\omega_1|\boldsymbol{x}) > P(\omega_2|\boldsymbol{x})$$

As pointed out in Chapter 2, the classification error probability depends on the difference between $P(\omega_1|\boldsymbol{x})$ and $P(\omega_2|\boldsymbol{x})$, e.g., Eq. (2.12). Hence, the ratio $\frac{P(\omega_1|\boldsymbol{x})}{P(\omega_2|\boldsymbol{x})}$ can

convey useful information concerning the discriminatory capabilities associated with an adopted feature vector $\boldsymbol{x}$, with respect to the two classes ω_1, ω_2. Alternatively (for given values of $P(\omega_1), P(\omega_2)$), the same information resides in the ratio $\ln \frac{p(\boldsymbol{x}|\omega_1)}{p(\boldsymbol{x}|\omega_2)} \equiv D_{12}(\boldsymbol{x})$, and this can be used as a measure of the underlying discriminating information of class ω_1 with respect to ω_2. Clearly, for completely overlapped classes, we get $D_{12}(\boldsymbol{x}) = 0$. Since $\boldsymbol{x}$ takes different values, it is natural to consider the mean value over class ω_1, that is,

$$D_{12} = \int_{-\infty}^{+\infty} p(\boldsymbol{x}|\omega_1) \ln \frac{p(\boldsymbol{x}|\omega_1)}{p(\boldsymbol{x}|\omega_2)} \, d\boldsymbol{x} \tag{5.19}$$

Similar arguments hold for class ω_2, and we define

$$D_{21} = \int_{-\infty}^{+\infty} p(\boldsymbol{x}|\omega_2) \ln \frac{p(\boldsymbol{x}|\omega_2)}{p(\boldsymbol{x}|\omega_1)} \, d\boldsymbol{x} \tag{5.20}$$

The sum

$$d_{12} = D_{12} + D_{21}$$

is known as the *divergence* and can be used as a separability measure for the classes ω_1, ω_2, with respect to the adopted feature vector $\boldsymbol{x}$. For a multiclass problem, the divergence is computed for every class pair ω_i, ω_j

$$\begin{aligned} d_{ij} &= D_{ij} + D_{ji} \\ &= \int_{-\infty}^{+\infty} (p(\boldsymbol{x}|\omega_i) - p(\boldsymbol{x}|\omega_j)) \ln \frac{p(\boldsymbol{x}|\omega_i)}{p(\boldsymbol{x}|\omega_j)} \, d\boldsymbol{x} \end{aligned} \tag{5.21}$$

and the average class separability can be computed using the average divergence

$$d = \sum_{i=1}^{M} \sum_{j=1}^{M} P(\omega_i) P(\omega_j) d_{ij}$$

Divergence is basically a form of the Kullback–Leibler distance measure between density functions [Kulb 51] (Appendix A). The divergence has the following easily shown properties:

$$\begin{aligned} &d_{ij} \geq 0 \\ &d_{ij} = 0 \qquad \text{if } i = j \\ &d_{ij} = d_{ji} \end{aligned}$$

If the components of the feature vector are statistically independent, then it can be shown (Problem 5.10) that

$$d_{ij}(x_1, x_2, \ldots, x_l) = \sum_{r=1}^{l} d_{ij}(x_r)$$

Assuming now that the density functions are Gaussians $\mathcal{N}(\boldsymbol{\mu}_i, \Sigma_i)$ and $\mathcal{N}(\boldsymbol{\mu}_j, \Sigma_j)$, respectively, the computation of the divergence is simplified, and it is not difficult to show that

$$d_{ij} = \frac{1}{2}\text{trace}\{\Sigma_i^{-1}\Sigma_j + \Sigma_j^{-1}\Sigma_i - 2I\} + \frac{1}{2}(\boldsymbol{\mu}_i - \boldsymbol{\mu}_j)^T\left(\Sigma_i^{-1} + \Sigma_j^{-1}\right)(\boldsymbol{\mu}_i - \boldsymbol{\mu}_j) \tag{5.22}$$

For the one-dimensional case this becomes

$$d_{ij} = \frac{1}{2}\left(\frac{\sigma_j^2}{\sigma_i^2} + \frac{\sigma_i^2}{\sigma_j^2} - 2\right) + \frac{1}{2}(\mu_i - \mu_j)^2\left(\frac{1}{\sigma_i^2} + \frac{1}{\sigma_j^2}\right)$$

As already pointed out, a class separability measure cannot depend only on the difference of the mean values; it must also be variance dependent. Indeed, divergence does depend explicitly on both the difference of the means and the respective variances. Furthermore, d_{ij} can be large even for equal mean values, *provided the variances differ significantly*. Thus, class separation is still possible even if the class means coincide. We will come to this later on.

Let us now investigate (5.22). If the covariance matrices of the two Gaussian distributions are equal, $\Sigma_i = \Sigma_j = \Sigma$, then the divergence is further simplified to

$$d_{ij} = (\boldsymbol{\mu}_i - \boldsymbol{\mu}_j)^T\Sigma^{-1}(\boldsymbol{\mu}_i - \boldsymbol{\mu}_j)$$

which is nothing other than the Mahalanobis distance between the corresponding mean vectors. This has another interesting implication. Recalling Problem 2.9 of Chapter 2, it turns out that in this case we have a direct relation between the divergence d_{ij} and the Bayes error—that is, the minimum error we can achieve by adopting the specific feature vector. This is a most desirable property for any class separability measure. Unfortunately, such a direct relation of the divergence with the Bayes error is not possible for more general distributions. Furthermore, in [Swai 73, Rich 95] it is pointed out that the specific dependence of the divergence on the difference of the mean vectors may lead to misleading results, in the sense that small variations in the difference of the mean values can produce large changes in the divergence, which, however, are not reflected in the classification error. To overcome this, a variation of the divergence is suggested, called the *transformed divergence:*

$$\hat{d}_{ij} = 2\left(1 - \exp(-d_{ij}/8)\right)$$

In the sequel, we will try to define class separability measures with a closer relationship to the Bayes error.

5.6.2 Chernoff Bound and Bhattacharyya Distance

The minimum attainable classification error of the Bayes classifier for two classes ω_1, ω_2 can be written as:

$$P_e = \int_{-\infty}^{\infty} \min\left[P(\omega_i)p(\boldsymbol{x}|\omega_i), P(\omega_j)p(\boldsymbol{x}|\omega_j)\right] d\boldsymbol{x} \tag{5.23}$$

Analytic computation of this integral in the general case is not possible. However, an upper bound can be derived. The derivation is based on the inequality

$$\min[a, b] \leq a^s b^{1-s} \quad \text{for} \quad a, b \geq 0, \quad \text{and} \quad 0 \leq s \leq 1 \tag{5.24}$$

Combining (5.23) and (5.24), we get

$$P_e \leq P(\omega_i)^s P(\omega_j)^{1-s} \int_{-\infty}^{\infty} p(\boldsymbol{x}|\omega_i)^s p(\boldsymbol{x}|\omega_j)^{1-s}\, d\boldsymbol{x} \equiv \epsilon_{CB} \tag{5.25}$$

ϵ_{CB} is known as the *Chernoff bound*. The minimum bound can be computed by minimizing ϵ_{CB} with respect to s. A special form of the bound results for $s = 1/2$:

$$P_e \leq \epsilon_{CB} = \sqrt{P(\omega_i)P(\omega_j)} \int_{-\infty}^{\infty} \sqrt{p(\boldsymbol{x}|\omega_i)p(\boldsymbol{x}|\omega_j)}\, d\boldsymbol{x} \tag{5.26}$$

For Gaussian distributions $\mathcal{N}(\boldsymbol{\mu}_i, \Sigma_i), \mathcal{N}(\boldsymbol{\mu}_j, \Sigma_j)$ and after a bit of algebra, we obtain

$$\epsilon_{CB} = \sqrt{P(\omega_i)P(\omega_j)}\exp(-B)$$

where

$$B = \frac{1}{8}(\boldsymbol{\mu}_i - \boldsymbol{\mu}_j)^T \left(\frac{\Sigma_i + \Sigma_j}{2}\right)^{-1} (\boldsymbol{\mu}_i - \boldsymbol{\mu}_j) + \frac{1}{2}\ln \frac{|\,\frac{\Sigma_i+\Sigma_j}{2}\,|}{\sqrt{|\,\Sigma_i\,||\,\Sigma_j\,|}} \tag{5.27}$$

and $|\cdot|$ denotes the determinant of the respective matrix. The term B is known as the *Bhattacharyya distance*, and it is used as a class separability measure. It can be shown (Problem 5.11) that it corresponds to the optimum Chernoff bound when $\Sigma_i = \Sigma_j$. It is readily seen that in this case the Bhattacharyya distance becomes proportional to the Mahalanobis distance between the means. In [Lee 00] an equation that relates the optimal Bayesian error and the Bhattacharyya distance is proposed, based on an empirical study involving normal distributions. This was subsequently used for feature selection in [Choi 03].

A comparative study of various distance measures for feature selection in the context of multispectral data classification in remote sensing can be found in [Maus 90]. A more detailed treatment of the topic is given in [Fuku 90].

Example 5.4

Assume that $P(\omega_1) = P(\omega_2)$ and that the corresponding distributions are Gaussians $\mathcal{N}(\boldsymbol{\mu}, \sigma_1^2 I)$ and $\mathcal{N}(\boldsymbol{\mu}, \sigma_2^2 I)$. The Bhattacharyya distance becomes

$$B = \frac{1}{2}\ln \frac{\left(\frac{\sigma_1^2+\sigma_2^2}{2}\right)^l}{\sqrt{\sigma_1^{2l}\sigma_2^{2l}}} = \frac{1}{2}\ln\left(\frac{\sigma_1^2 + \sigma_2^2}{2\sigma_1\sigma_2}\right)^l \tag{5.28}$$

For the one-dimensional case $l = 1$ and for $\sigma_1 = 10\sigma_2$, $B = 0.8097$ and

$$P_e \leq 0.2225$$

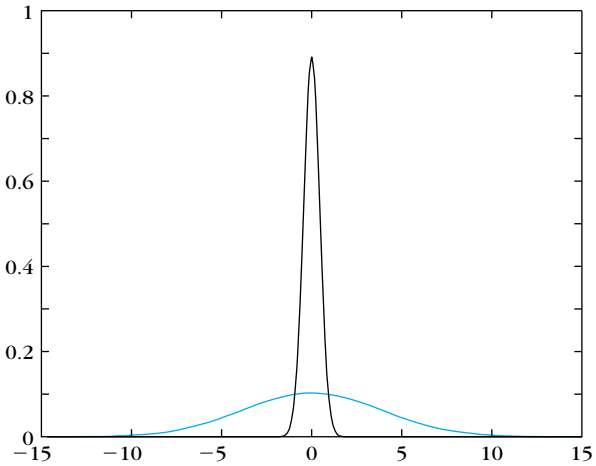

FIGURE 5.4

Gaussian pdfs with the same mean and different variances.

If $\sigma_1 = 100\sigma_2$, $B = 1.9561$ and

$$P_e \leq 0.0707$$

Thus, the greater the difference of the variances, the smaller the error bound. The decrease is bigger for higher dimensions due to the dependence on l. Figure 5.4 shows the pdfs for the same mean and $\sigma_1 = 1$, $\sigma_2 = 0.01$. The figure is self-explanatory as to how the Bayesian classifier discriminates between two classes of the same mean and significantly different variances. Furthermore, as $\sigma_2/\sigma_1 \longrightarrow 0$, the probability of error tends to zero (why?)

5.6.3 Scatter Matrices

A major disadvantage of the class separability criteria considered so far is that they are not easily computed, unless the Gaussian assumption is employed. We will now turn our attention to a set of simpler criteria, built upon information related to the way feature vector samples are scattered in the l-dimensional space. To this end, the following matrices are defined:

Within-class scatter matrix

$$S_w = \sum_{i=1}^{M} P_i \Sigma_i$$

where Σ_i is the covariance matrix for class ω_i

$$\Sigma_i = E[(\boldsymbol{x} - \boldsymbol{\mu}_i)(\boldsymbol{x} - \boldsymbol{\mu}_i)^T]$$

and P_i the *a priori* probability of class ω_i. That is, $P_i \simeq n_i/N$, where n_i is the number of samples in class ω_i, out of a total of N samples. Obviously, trace $\{S_w\}$ is a measure of the average, over all classes, variance of the features.

Between-class scatter matrix

$$S_b = \sum_{i=1}^{M} P_i(\boldsymbol{\mu}_i - \boldsymbol{\mu}_0)(\boldsymbol{\mu}_i - \boldsymbol{\mu}_0)^T$$

where $\boldsymbol{\mu}_0$ is the global mean vector

$$\boldsymbol{\mu}_0 = \sum_{i}^{M} P_i \boldsymbol{\mu}_i$$

Trace$\{S_b\}$ is a measure of the average (over all classes) distance of the mean of each individual class from the respective global value.

Mixture scatter matrix

$$S_m = E[(\boldsymbol{x} - \boldsymbol{\mu}_0)(\boldsymbol{x} - \boldsymbol{\mu}_0)^T]$$

That is, S_m is the covariance matrix of the feature vector with respect to the global mean. It is not difficult to show (Problem 5.12) that

$$S_m = S_w + S_b$$

Its trace is the sum of variances of the features around their respective global mean. From these definitions it is straightforward to see that the criterion

$$J_1 = \frac{\text{trace}\{S_m\}}{\text{trace}\{S_w\}}$$

takes large values when samples in the l-dimensional space are well clustered around their mean, within each class, and the clusters of the different classes are well separated. Sometimes S_b is used in place of S_m. An alternative criterion results if determinants are used in the place of traces. This is justified for scatter matrices that are symmetric positive definite, and thus their eigenvalues are positive (Appendix B). The trace is equal to the sum of the eigenvalues, while the determinant is equal to their product. Hence, large values of J_1 also correspond to large values of the criterion

$$J_2 = \frac{|S_m|}{|S_w|} = |S_w^{-1} S_m|$$

A variant of J_2 commonly encountered in practice is

$$J_3 = \text{trace}\{S_w^{-1} S_m\}$$

As we will see later on, criteria J_2 and J_3 have the advantage of being invariant under linear transformations, and we will adopt them to derive features in an optimal way.

In [Fuku 90] a number of different criteria are also defined by using various combinations of S_w, S_b, S_m in a "trace" or "determinant" formulation. However, whenever a determinant is used, one should be careful with S_b, since $|S_b| = 0$ for $M < l$. This is because S_b is the sum of M $l \times l$ matrices, of rank one each. In practice, all three matrices are approximated by appropriate averaging using the available data samples.

These criteria take a special form in the one-dimensional, two-class problem. In this case, it is easy to see that for equiprobable classes $|S_w|$ is proportional to $\sigma_1^2 + \sigma_2^2$ and $|S_b|$ proportional to $(\mu_1 - \mu_2)^2$. Combining S_b and S_w, the so-called *Fisher's discriminant ratio (FDR)* results

$$FDR = \frac{(\mu_1 - \mu_2)^2}{\sigma_1^2 + \sigma_2^2}$$

FDR is sometimes used to quantify the separability capabilities of individual features. It reminds us of the test statistic q appearing in the hypothesis statistical tests dealt with before. However, here the use of *FDR* is suggested in a more "primitive" fashion, independent of the underlying statistical distributions. For the multiclass case, averaging forms of *FDR* can be used. One possibility is

$$FDR_1 = \sum_{i}^{M} \sum_{j \neq i}^{M} \frac{(\mu_i - \mu_j)^2}{\sigma_i^2 + \sigma_j^2}$$

where the subscripts i, j refer to the mean and variance corresponding to the feature under investigation for the classes ω_i, ω_j, respectively.

Example 5.5

Figure 5.5 shows three cases of classes at different locations and within-class variances. The resulting values for the J_3 criterion involving the S_w and S_m matrices are 164.7, 12.5, and

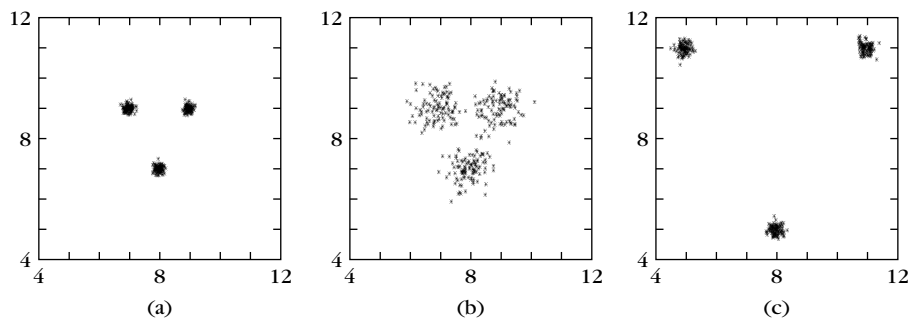

FIGURE 5.5

Classes with (a) small within-class variance and small between-class distances, (b) large within-class variance and small between-class distances, and (c) small within-class variance and large between-class distances.

620.9 for the cases in Figures 5.5a, b, and c, respectively. That is, the best is for distant well-clustered classes and the worst for the case of closely located classes with large within-class variance.

5.7 FEATURE SUBSET SELECTION

Having defined a number of criteria, measuring the classification effectiveness of individual features and/or feature vectors, we come to the heart of our problem, that is, to select a subset of l features out of m originally available. There are two major directions to follow.

5.7.1 Scalar Feature Selection

Features are treated individually. Any of the class separability measuring criteria can be adopted, for example, *ROC*, *FDR*, one-dimensional divergence, and so on. The value of the criterion $C(k)$ is computed for each of the features, $k = 1, 2, \ldots, m$. Features are then ranked in order of descending values of $C(k)$. The l features corresponding to the l best values of $C(k)$ are then selected to form the feature vector.

All the criteria we have dealt with in the previous sections measure the classification capability with respect to a two-class problem. As we have already pointed out in a couple of places, in a multiclass situation a form of average or "total" value, over all classes, is used to compute $C(k)$. However, this is not the only possibility. In [Su 94] the one-dimensional divergence d_{ij} was used and computed for every pair of classes. Then, for each of the features, the corresponding $C(k)$ was set equal to

$$C(k) = \min_{i,j} d_{ij}$$

that is, the minimum divergence value over all class pairs, instead of an average value. Thus, selecting the features with the largest $C(k)$ values is equivalent to choosing features with the best "worst-case" class separability capability, giving a "*maxmin*" flavor to the feature selection task. Such an approach may lead to more robust performance in certain cases.

The major advantage of dealing with features individually is computational simplicity. However, such approaches do not take into account existing correlations between features. Before we proceed to techniques dealing with vectors, we will comment on some *ad hoc* techniques that incorporate correlation information combined with criteria tailored for scalar features.

Let $x_{nk}, n = 1, 2, \ldots, N$ and $k = 1, 2, \ldots, m$, be the kth feature of the nth pattern. The cross-correlation coefficient between any two of them is given by

$$\rho_{ij} = \frac{\sum_{n=1}^{N} x_{ni} x_{nj}}{\sqrt{\sum_{n=1}^{N} x_{ni}^2 \sum_{n=1}^{N} x_{nj}^2}} \tag{5.29}$$

It can be shown that $|\rho_{ij}| \leq 1$ (Problem 5.13). The selection procedure evolves along the following steps:

- Select a class separability criterion C and compute its values for all the available features $x_k, k = 1, 2, \ldots, m$. Rank them in descending order and choose the one with the best C value. Let us say that this is x_{i_1}.
- To select the second feature, compute the cross-correlation coefficient defined in Eq. (5.29) between the chosen x_{i_1} and each of the remaining $m-1$ features, that is, $\rho_{i_1 j}, j \neq i_1$.
- Choose the feature x_{i_2} for which

$$i_2 = arg \max_j \left\{ \alpha_1 C(j) - \alpha_2 |\rho_{i_1 j}| \right\}, \quad \text{for all } j \neq i_1$$

  where α_1, α_2 are weighting factors that determine the relative importance we give to the two terms. In words, for the selection of the next feature, we take into account not only the class separability measure C but also the correlation with the already chosen feature. This is then generalized for the kth step

- Select $x_{i_k}, k = 3, \ldots, l$, so that

$$i_k = arg \max_j \left\{ \alpha_1 C(j) - \frac{\alpha_2}{k-1} \sum_{r=1}^{k-1} |\rho_{i_r j}| \right\} \quad \text{for } j \neq i_r, \quad r = 1, 2, \ldots, k-1$$

  That is, the average correlation with all previously selected features is taken into account.

There are variations of this procedure. For example, in [Fine 83] more than one criterion is adopted and averaged out. Hence, the best index is found by optimizing

$$\left\{ \alpha_1 C_1(j) + \alpha_2 C_2(j) - \frac{\alpha_3}{k-1} \sum_{r=1}^{k-1} |\rho_{i_r j}| \right\}$$

5.7.2 Feature Vector Selection

Treating features individually, that is, as scalars, has the advantage of computational simplicity but may not be effective for complex problems and for features with high mutual correlation. We will now focus on techniques measuring classification capabilities of feature vectors. It does not require much thought to see that computational burden is the major limiting factor of such an approach. Indeed, if we want to act according to what "optimality" suggests, we should form *all* possible vector combinations of l features out of the m originally available. According to the

type of optimality rule that one chooses to work with, the feature selection task is classified into two categories:

Filter approach. In this approach, the optimality rule for feature selection is independent of the classifier, which will be used in the classifier design stage. For each combination we should use one of the separability criteria introduced previously (e.g., Bhattacharrya distance, J_2) and select the best feature vector combination. Recalling our combinatorics basics, we obtain the total number of vectors as

$$\binom{m}{l} = \frac{m!}{l!(m-l)!} \tag{5.30}$$

This is a large number even for small values of l, m. Indeed, for $m = 20, l = 5$, the number equals 15,504. Furthermore, in many practical cases the number l is not even known *a priori*. Thus, one has to try feature combinations for different values of l and select the "best" value for it (beyond which no gain in performance is obtained) and the corresponding "best" l-dimensional feature vector.

Wrapper approach. As we will see in Chapter 10, sometimes it is desirable to base our feature selection decision not on the values of an adopted class separability criterion but on the performance of the classifier itself. That is, for each feature vector combination the classification error probability of the classifier has to be estimated and the combination resulting in the minimum error probability is selected. This approach may increase the complexity requirements even more, depending, of course, on the classifier type.

For both approaches, in order to reduce complexity, a number of efficient searching techniques have been suggested. Some of them are suboptimal and some optimal (under certain assumptions or constraints).

Suboptimal Searching Techniques

Sequential Backward Selection

We will demonstrate the method via an example. Let $m = 4$, and the originally available features are x_1, x_2, x_3, x_4. We wish to select two of them. The selection procedure consists of the following steps:

- Adopt a class separability criterion, C, and compute its value for the feature vector $[x_1, x_2, x_3, x_4]^T$.
- Eliminate one feature and for each of the possible resulting combinations, that is, $[x_1, x_2, x_3]^T$, $[x_1, x_2, x_4]^T$, $[x_1, x_3, x_4]^T$, $[x_2, x_3, x_4]^T$, compute the corresponding criterion value. Select the combination with the best value, say $[x_1, x_2, x_3]^T$.
- From the selected three-dimensional feature vector eliminate one feature and for each of the resulting combinations, $[x_1, x_2]^T$, $[x_1, x_3]^T$, $[x_2, x_3]^T$, compute the criterion value and select the one with the best value.

Thus, starting from m, at each step we drop out one feature from the "best" combination until we obtain a vector of l features. Obviously, this is a *suboptimal* searching procedure, since nobody can guarantee that the optimal two-dimensional vector has to originate from the optimal three-dimensional one. The number of combinations searched via this method is $1 + 1/2((m + 1)m - l(l + 1))$ (Problem 5.15), which is substantially less than that of the full search procedure.

Sequential Forward Selection

Here, the reverse to the preceding procedure is followed:

- Compute the criterion value for each of the features. Select the feature with the best value, say x_1.
- Form all possible two-dimensional vectors that contain the winner from the previous step, that is, $[x_1, x_2]^T, [x_1, x_3]^T, [x_1, x_4]^T$. Compute the criterion value for each of them and select the best one, say $[x_1, x_3]^T$.

If $l = 3$, then the procedure must continue. That is, we form all three-dimensional vectors springing from the two-dimensional winner, that is, $[x_1, x_3, x_2]^T, [x_1, x_3, x_4]^T$, and select the best one. For the general l, m case, it is simple algebra to show that the number of combinations searched with this procedure is $lm - l(l - 1)/2$. Thus, from a computational point of view, the backward search technique is more efficient than the forward one for l closer to m than to 1.

Floating Search Methods

The preceding two methods suffer from the so-called *nesting effect*. That is, once a feature is discarded in the backward method, there is no possibility for it to be reconsidered again. The opposite is true for the forward procedure; once a feature is chosen, there is no way for it to be discarded later on. In [Pudi 94] a technique is suggested that offers the flexibility to reconsider features previously discarded and, vice versa, to discard features previously selected. The technique is called the *floating search method*. Two schemes implement this technique. One springs from the forward selection, and the other from the backward selection rationale. We will focus on the former. We consider a set of m features, and the idea is to search for the best subset of k of them for $k = 1, 2, \ldots, l \leq m$ so that a cost criterion C is optimized. Let $X_k = \{x_1, x_2, \ldots, x_k\}$ be the set of the best combination of k of the features and Y_{m-k} the set of the remaining $m - k$ features. We also keep all the lower dimension best subsets $X_2, X_3, \ldots, X_{k-1}$ of $2, 3, \ldots, k - 1$ features, respectively. The rationale at the heart of the method is summarized as follows: At the next step the $k + 1$ best subset X_{k+1} is formed by "borrowing" an element from Y_{m-k}. Then, return to the previously selected lower dimension subsets to check whether the inclusion of this new element improves the criterion C. If it does, the new element replaces one of the

previously selected features. The steps of the algorithm, when maximization of C is required are:

- *Step I: Inclusion* $x_{k+1} = arg\max_{y \in Y_{m-k}} C(\{X_k, y\})$; that is, choose that element from Y_{m-k} which, combined with X_k, results in the best value of C. $X_{k+1} = \{X_k, x_{k+1}\}$
- *Step II: Test*
  1. $x_r = arg\max_{y \in X_{k+1}} C(X_{k+1} - \{y\})$; that is, find the feature that has the least effect on the cost when it is removed from X_{k+1}.
  2. If $r = k + 1$, change $k = k + 1$ and go to step I.
  3. If $r \neq k + 1$ AND $C(X_{k+1} - \{x_r\}) < C(X_k)$ go to step I; that is, if removal of x_r does not improve upon the cost of the previously selected best group of k, no further backward search is performed.
  4. If $k = 2$ put $X_k = X_{k+1} - \{x_r\}$ and $C(X_k) = C(X_{k+1} - \{x_r\})$; go to step I.
- *Step III: Exclusion*
  1. $X_k' = X_{k+1} - \{x_r\}$; that is, remove x_r.
  2. $x_s = arg\max_{y \in X_k'} C(X_k' - \{y\})$; that is, find the least significant feature in the new set.
  3. If $C(X_k' - \{x_s\}) < C(X_{k-1})$ then $X_k = X_k'$ and go to step I; no further backward search is performed.
  4. Put $X_{k-1}' = X_k' - \{x_s\}$ and $k = k - 1$.
  5. If $k = 2$ put $X_k = X_k'$ and $C(X_k) = C(X_k')$ and go to step I.
  6. Go to step III.1.

The algorithm is initialized by running the sequential forward algorithm to form X_2. The algorithm terminates when l features have been selected. Although the algorithm does not guarantee finding all the best feature subsets, it results in substantially improved performance compared with its sequential counterpart, at the expense of increased complexity. The backward floating search scheme operates in the reverse direction but with the same philosophy.

Optimal Searching Techniques

These techniques are applicable when the *separability criterion is monotonic*, that is,

$$C(x_1, \ldots, x_i) \leq C(x_1, \ldots, x_i, x_{i+1})$$

This property allows identifying the optimal combination but at a considerably reduced computational cost with respect to (5.30). Algorithms based on the *dynamic programming* concept (Chapter 8) offer one possibility to approaching

the problem. A computationally more efficient way is to formulate the problem as a combinatorial optimization task and employ the so-called *branch and bound* methods to obtain the optimal solution [Lawe 66, Yu 93]. These methods compute the optimal value without involving exhaustive enumeration of all possible combinations. A more detailed description of the branch and bound methods is given in Chapter 15 and can also be found in [Fuku 90]. However, the complexity of these techniques is still higher than that of the previously mentioned suboptimal techniques.

Remark

- The separability measures and feature selection techniques presented above, although they indicate the major directions followed in practice, do not cover the whole range of methods that have been suggested. For example, in [Bati 94, Kwak 02, Leiv 07] the mutual information between the input features and the classifier's outputs is used as a criterion. The features that are selected maximize the input–output mutual information. In [Sind 04] the mutual information between the class labels of the respective features and those predicted by the classifier is used as a criterion. This has the advantage that only discrete random variables are involved. The existence of bounds that relate the probability of error to the mutual information function, for example, [Erdo 03, Butz 05], could offer a theoretically pleasing flavor to the adoption of information theoretic criteria for feature selection. In [Seti 97] a feature selection technique is proposed based on a decision tree by excluding features one by one and retraining the classifier. In [Zhan 02] the tabu combinatorial optimization technique is employed for feature selection.

Comparative studies of various feature selection searching schemes can be found in [Kitt 78, Devi 82, Pudi 94, Jain 97, Brun 00, Wang 00, Guyo 03]. The task of *selection bias*, when using the wrapper approach and how to overcome it is treated in [Ambr 02]. This is an important issue, and it has to be carefully considered in practice in order to avoid biased estimates of the error probability.

5.8 OPTIMAL FEATURE GENERATION

So far, the class separability measuring criteria have been used in a rather "passive" way, that is, to measure the classification effectiveness of features generated in *some* way. In this section we will employ these measuring criteria in an "active" manner, as an integral part of the feature generation process itself. From this point of view, this section can be considered as a bridge between this chapter and the following one. The method goes back to the pioneering work of Fisher ([Fish 36]) on *linear discrimination*, and it is also known as *linear discriminant analysis (LDA)*. We will first focus on the simplest form of the method in order to get a better feeling and physical understanding of its basic rationale.

The Two-class Case

Let our data points, $\boldsymbol{x}$, be in the m-dimensional space and assume that they originate from two classes. Our goal is to generate a feature y as a linear combination of the components of $\boldsymbol{x}$. In such a way, we expect to "squeeze" the classification-related information residing in $\boldsymbol{x}$ in a smaller number (in this case only one) of features. In this section, this goal is achieved by seeking the direction $\boldsymbol{w}$ in the m-dimensional space, *along which the two classes are best separated in some way*. This is not the only possible path for generating features via linear combination of measurements, and a number of alternative techniques will be studied in the next chapter.

Given an $\boldsymbol{x} \in \mathcal{R}^m$ the scalar

$$y = \frac{\boldsymbol{w}^T \boldsymbol{x}}{||\boldsymbol{w}||} \tag{5.31}$$

is the projection of $\boldsymbol{x}$ along $\boldsymbol{w}$. Since scaling all our feature vectors by the same factor does not add any classification-related information, we will ignore the scaling factor $||\boldsymbol{w}||$. We adopt the Fisher's discriminant ratio (FDR) (Section 5.6.3)

$$FDR = \frac{(\mu_1 - \mu_2)^2}{\sigma_1^2 + \sigma_2^2} \tag{5.32}$$

where μ_1, μ_2 are the mean values and σ_1^2, σ_2^2 the variances of y in the two classes ω_1 and ω_2, respectively, after the projection along $\boldsymbol{w}$. Using the definition in (5.31) and omitting $||\boldsymbol{w}||$, it is readily seen that

$$\mu_i = \boldsymbol{w}^T \boldsymbol{\mu}_i, \ i = 1, 2 \tag{5.33}$$

where $\boldsymbol{\mu}_i, \ i = 1, 2$, is the mean value of the data in ω_i in the m-dimensional space. Assuming the classes to be equiprobable and recalling the definition of S_b in Section 5.6.3, it is easily shown that

$$(\mu_1 - \mu_2)^2 = \boldsymbol{w}^T (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)^T \boldsymbol{w} \propto \boldsymbol{w}^T S_b \boldsymbol{w} \tag{5.34}$$

where $\propto$ denotes proportionality. We now turn our attention to the denominator of (5.32). We have

$$\sigma_i^2 = E[(y - \mu_i)^2] = E[\boldsymbol{w}^T (\boldsymbol{x} - \boldsymbol{\mu}_i)(\boldsymbol{x} - \boldsymbol{\mu}_i)^T \boldsymbol{w}] = \boldsymbol{w}^T \Sigma_i \boldsymbol{w} \tag{5.35}$$

where for each $i = 1, 2$, samples $y\,(\boldsymbol{x})$ from the respective class ω_i have been used. Σ_i is the covariance matrix corresponding to the data of class ω_i in the m-dimensional space. Recalling the definition of S_w from Section 5.6.3, we get

$$\sigma_1^2 + \sigma_2^2 \propto \boldsymbol{w}^T S_w \boldsymbol{w} \tag{5.36}$$

Combining (5.36), (5.34), and (5.32), we end up that the optimal direction is obtained by maximizing Fisher's criterion

$$FDR(\boldsymbol{w}) = \frac{\boldsymbol{w}^T S_b \boldsymbol{w}}{\boldsymbol{w}^T S_w \boldsymbol{w}} \tag{5.37}$$

with respect to $\boldsymbol{w}$. This is the celebrated generalized Rayleigh quotient, which, as it is known from linear algebra (Problem 5.16), is maximized if $\boldsymbol{w}$ is chosen such that

$$S_b\boldsymbol{w} = \lambda S_w\boldsymbol{w} \tag{5.38}$$

where λ is the largest eigenvalue of $S_w^{-1}S_b$. However, for our simple case we do not have to worry about any eigen decomposition. By the definition of S_b we have that

$$\lambda S_w\boldsymbol{w} = (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)^T\boldsymbol{w} = \alpha(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)$$

where α is a scalar. Solving the previous equation with respect to $\boldsymbol{w}$, and since we are only interested in the direction of $\boldsymbol{w}$, we can write

$$\boldsymbol{w} = S_w^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2) \tag{5.39}$$

assuming, of course, that S_w is invertible. As has already been discussed, in practice, S_w and S_b are approximated by averaging using the available data samples.

Figures 5.6a and 5.6b correspond to two examples for the special case of the two-dimensional space ($m = 2$). In both cases, the classes are assumed equiprobable and have the same covariance matrix Σ. Thus $S_w = \Sigma$. In Figure 5.6a, Σ is diagonal with equal diagonal elements, and $\boldsymbol{w}$ turns out to be parallel to $\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2$. In Figure 5.6b, Σ is no more diagonal, and the data distribution does not have a spherical symmetry. In this case, the optimal direction for projection (the line on the left) is no more parallel to $\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2$, and its direction changes in order to account for the shape of the data distribution. This simple example once again demonstrates that the right choice of the features is of paramount importance. Take as an example the case of generating a feature by projecting along the direction of the line on the right in Figure 5.6b. Then, the values that this feature takes for the two classes exhibit a heavy overlap.

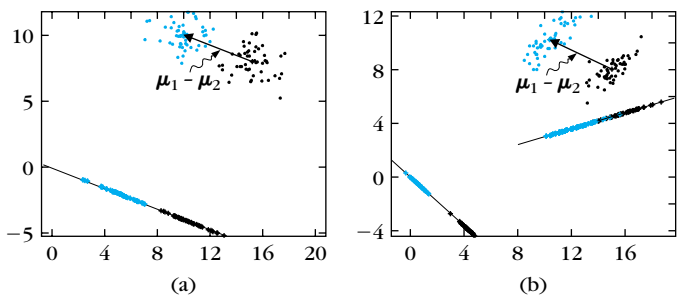

FIGURE 5.6

(a) The optimal line resulting from Fisher's criterion, for two Gaussian classes. Both classes share the same diagonal covariance matrix, with equal elements on the diagonal. The line is parallel to $\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2$. (b) The covariance matrix for both classes is nondiagonal. The optimal line is on the left. Observe that it is no more parallel to $\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2$. The line on the right is not optimal and the classes, after the projection, overlap.

Thus, we have reduced the number of features from m to 1 in an optimal way. Classification can now be performed based on y. Optimality guarantees that the class separability, with respect to y, is as high as possible, as this is measured by the FDR criterion.

In the case where both classes are described by Gaussian pdfs with equal covariance matrices, Eq. (5.39) corresponds to nothing else but the optimal Bayesian classifier with the exception of a threshold value (Problem 2.11 and Eqs. (2.44)–(2.46)). Moreover, recall from Problem 3.14 that this is also directly related to the linear MSE classifier. In other words, although our original goal was to generate a single feature (y) by linearly combining the m components of $\boldsymbol{x}$, we obtained something extra for free. Fisher's method performed feature generation and at the same time the design of a (linear) classifier; it combined the stages of feature generation and classifier design into a single one. The resulting classifier is

$$g(\boldsymbol{x}) = (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)^T S_w^{-1} \boldsymbol{x} + w_0 \tag{5.40}$$

However, Fisher's criterion does not provide a value for w_0, which has to be determined. For example, for the case of two Gaussian classes with the same covariance matrix the optimal classifier is shown to take the form (see also Problem 3.14)

$$g(\boldsymbol{x}) = (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)^T S_w^{-1} \left(\boldsymbol{x} - \frac{1}{2}(\boldsymbol{\mu}_1 + \boldsymbol{\mu}_2) \right) - \ln \frac{P(\omega_2)}{P(\omega_1)} \tag{5.41}$$

It has to be emphasized, however, that in the context of Fisher's theory the Gaussian assumption was not necessary to derive the direction of the optimal hyperplane. In practice, sometimes the rule in (5.41) is used even if we know that the data are non-Gaussian. Of course, other values of w_0 may be devised, according to the problem at hand.

Multiclass Case

The previous results, obtained for the two-class case, are readily generalized for the case of $M > 2$ classes. The multiclass LDA has been adopted as a tool for optimal feature generation in a number of applications, including biometrics and bioinformatics, where an original large number of features has to be compactly reduced. Our major task can be summarized as follows: If $\boldsymbol{x}$ is an m-dimensional vector of measurement samples, transform it into another l-dimensional vector $\boldsymbol{y}$ so that an adopted class separability criterion is optimized. We will confine ourselves to linear transformations,

$$\boldsymbol{y} = A^T \boldsymbol{x}$$

where A^T is an $l \times m$ matrix. Any of the criteria exposed so far can be used. Obviously, the degree of complexity of the optimization procedure depends heavily on the chosen criterion. We will demonstrate the method via the J_3 scattering matrix criterion, involving S_w and S_b matrices. Its optimization is straightforward,

and at the same time it has some interesting implications. Let S_{xw}, S_{xb} be the within-class and between-class scatter matrices of $\boldsymbol{x}$. From the respective definitions, the corresponding matrices of $\boldsymbol{y}$ become

$$S_{yw} = A^T S_{xw} A, \quad S_{yb} = A^T S_{xb} A$$

Thus, the J_3 criterion in the $\boldsymbol{y}$ subspace is given by

$$J_3(A) = \text{trace}\{(A^T S_{xw} A)^{-1}(A^T S_{xb} A)\}$$

Our task is to compute the elements of A so that this is maximized. Then A must necessarily satisfy

$$\frac{\partial J_3(A)}{\partial A} = 0$$

It can be shown that (Problem 5.17)

$$\begin{aligned}\frac{\partial J_3(A)}{\partial A} &= -2S_{xw} A(A^T S_{xw} A)^{-1}(A^T S_{xb} A)(A^T S_{xw} A)^{-1} + 2S_{xb} A(A^T S_{xw} A)^{-1} \\ &= 0\end{aligned}$$

or

$$(S_{xw}^{-1} S_{xb})A = A(S_{yw}^{-1} S_{yb}) \tag{5.42}$$

An experienced eye will easily identify the affinity of this with an eigenvalue problem. It suffices to simplify its formulation slightly. Recall from Appendix B that the matrices S_{yw}, S_{yb} can be diagonalized simultaneously by a linear transformation

$$B^T S_{yw} B = I, \quad B^T S_{yb} B = D \tag{5.43}$$

which are the within- and between-class scatter matrices of the transformed vector

$$\hat{\boldsymbol{y}} = B^T \boldsymbol{y} = B^T A^T \boldsymbol{x}$$

B is an $l \times l$ matrix and D an $l \times l$ diagonal matrix. Note that in going from $\boldsymbol{y}$ to $\hat{\boldsymbol{y}}$ there is no loss in the value of the cost J_3. This is because J_3 is invariant under linear transformations, within the l-dimensional subspace. Indeed,

$$\begin{aligned}J_3(\hat{\boldsymbol{y}}) = \text{trace}\{S_{\hat{y}w}^{-1} S_{\hat{y}b}\} &= \text{trace}\{(B^T S_{yw} B)^{-1}(B^T S_{yb} B)\} \\ &= \text{trace}\{B^{-1} S_{yw}^{-1} S_{yb} B\} \\ &= \text{trace}\{S_{yw}^{-1} S_{yb} B B^{-1}\} = J_3(\boldsymbol{y})\end{aligned}$$

Combining (5.42) and (5.43), we finally obtain

$$(S_{xw}^{-1}S_{xb})C = CD \tag{5.44}$$

where $C = AB$ is an $m \times l$ dimensional matrix. Equation (5.44) is a typical eigenvalue–eigenvector problem, with the diagonal matrix D having the eigenvalues of $S_{xw}^{-1}S_{xb}$ on its diagonal and C having the corresponding eigenvectors as its columns. However, $S_{xw}^{-1}S_{xb}$ is an $m \times m$ matrix, and the question is which l out of a total of m eigenvalues we must choose for the solution of (5.44). From its definition, matrix S_{xb} is of rank $M-1$, where M is the number of classes (Problem 5.18). Thus, $S_{xw}^{-1}S_{xb}$ is also of rank $M-1$ and there are $M-1$ nonzero eigenvalues. Let us focus on the two possible alternatives separately.

- $l = M - 1$: We first form matrix C so that its columns are the unit norm $M - 1$ eigenvectors of $S_{xw}^{-1}S_{xb}$. Then we form the transformed vector

$$\hat{\boldsymbol{y}} = C^T \boldsymbol{x} \tag{5.45}$$

This guarantees the maximum J_3 value. *In reducing the number of data from m to $M - 1$, there is no loss in class separability power, as this is measured by J_3.* Indeed, recalling from linear algebra that the trace of a matrix is equal to the sum of its eigenvalues, we have

$$J_{3,x} = \text{trace}\{S_{xw}^{-1}S_{xb}\} = \lambda_1 + \cdots + \lambda_{M-1} + 0 \tag{5.46}$$

Also

$$J_{3,\hat{y}} = \text{trace}\{(C^T S_{xw} C)^{-1}(C^T S_{xb} C)\} \tag{5.47}$$

Rearranging (5.44), we get

$$C^T S_{xb} C = C^T S_{xw} CD \tag{5.48}$$

Combining (5.47) and (5.48), we obtain

$$J_{3,\hat{y}} = \text{trace}\{D\} = \lambda_1 + \cdots + \lambda_{M-1} = J_{3,x} \tag{5.49}$$

It is most interesting to view this from a slightly different perspective. Let us recall the Bayesian classifier for an M class problem. Of the M conditional class probabilities, $P(\omega_i|\boldsymbol{x})$, $i = 1, 2, \ldots, M$, only $M - 1$ are independent, since they all add up to one. In general, $M - 1$ is the *minimum* number of discriminant functions needed for an M-class classification task (Problem 5.19). *The linear operation $C^T \boldsymbol{x}$, which computes the $M - 1$ components of $\hat{\boldsymbol{y}}$, can be seen as an optimal linear rule that provides $M - 1$ discriminant functions, where optimality is with respect to J_3.* This was clearly demonstrated in the two-class case, where Fisher's method was also used as a classifier (subject to an unknown threshold).

Investigating the specific form that Eq. (5.45) takes for the two-class problem, one can show that for $M = 2$ there is only one nonzero eigenvalue, and it turns out that (Problem 5.20)

$$\hat{y} = (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)^T S_{xw}^{-1} \boldsymbol{x}$$

which is our familiar Fisher's linear discriminant.

- $l < M-1$: In this case C is formed from the eigenvectors corresponding to the l largest eigenvalues of $S_{xw}^{-1} S_{xb}$. The fact that J_3 is given as the sum of the corresponding eigenvalues guarantees its maximization. Of course, in this case there is loss of the available information because now $J_{3,\hat{y}} < J_{3,x}$.

A geometric interpretation of (5.45) reveals that $\hat{\boldsymbol{y}}$ is the projection of the original vector $\boldsymbol{x}$ onto the subspace spanned by the eigenvectors $\boldsymbol{v}_i$ of $S_w^{-1} S_b$. It must be pointed out that these *are not* necessarily mutually orthogonal. Indeed, although matrices S_w, S_b (S_m) are symmetric, products of the form $S_w^{-1} S_b$ are not; thus, the eigenvectors are not mutually orthogonal (Problem 5.21). Furthermore, as we saw during the proof, once we decide on which subspace to project (by selecting the appropriate combination of eigenvectors) *the value of J_3 remains invariant under any linear transformation within this subspace*. That is, it is independent of the coordinate system, and its value depends only on the particular subspace. In general, projection of the original feature vectors onto a lower dimensional subspace is associated with some information loss. An extreme example is shown in Figure 5.7, where the two classes coincide after projection on the v_2 axis. On the other hand, from all possible projection directions, Fisher's linear discrimination rule leads to

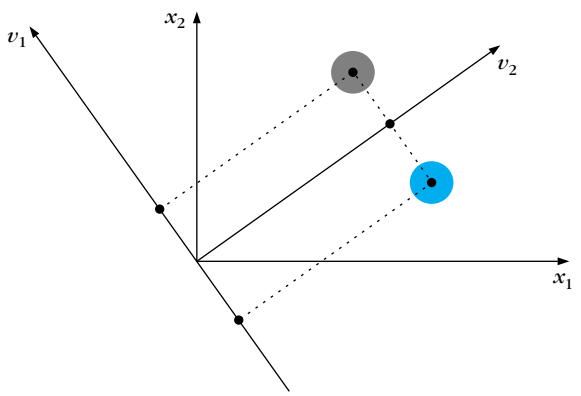

FIGURE 5.7

Geometry illustrating the loss of information associated with projections in lower dimensional subspaces. Projecting onto the direction of the principle eigenvector, $\boldsymbol{v}_1$, there is no loss of information. Projection on the orthogonal direction results in a complete class overlap.

the choice of the one-dimensional subspace v_1, which corresponds to the optimal J_3 value, that guarantees no loss of information for $l = M - 1 = 1$ (as this is measured by the J_3 criterion). Thus, this is a good choice, provided that J_3 is a good criterion for the problem of interest. Of course, this is not always the case; it depends on the specific classification task. For example, in [Hams 08] the criterion used is the probability of error for a multiclass task involving normally distributed data. A more extensive treatment of the topic, also involving other optimizing criteria, can be found in [Fuku 90].

Remarks

- If J_3 is used with another combination of matrices, such as S_w and S_m, then, in general, the rank of the corresponding matrix product involved in the trace is m and there are m nonzero eigenvalues. In such cases, the transformation matrix C is formed so that its columns are the eigenvectors corresponding to the l *largest eigenvalues*. According to (5.49), this guarantees the maximum value of J_3.
- In practice, one may encounter cases in which S_w is not invertible. This occurs in applications where the available size of the training set, N, is smaller than the dimensionality, m, of the original feature space. In such cases the resulting estimate of S_w, which is obtained as the mean of N outer vector products, has rank lower than m; hence it is singular. This is known as the small sample size (SSS) problem. Web document classification, face recognition, and disease classification based on gene-expression profiling are some examples where the small sample size problem occurs frequently in practice.

  One way to overcome this difficulty is to use the pseudoinverse S_w^+ in place of S_w^{-1} [Tian 86]. However, now, there is no guarantee that the J_3 criterion is maximized by selecting the eigenvectors of $S_w^+ S_b$ corresponding to the largest eigenvalues. An alternative route is to employ regularization techniques, in one way or another, for example, [Frie 89, Hast 95]. For example, S_w may be replaced by $S_w + \sigma\Omega$, where Ω can be any positive definite and symmetric matrix. The specific choice depends on the problem. The choice of σ is also a critical factor here. Another drawback of these techniques is that they do not scale well for problems with large dimensionality. For example, in certain tasks of face recognition, the resulting covariance matrices can be as high as a few thousand making matrix inversion a computationally thirsty task.

  Another way to deal with the small sample size problem is to adopt a two-stage approach. One such technique is the so-called PCA+LDA technique. In the first stage, principle component analysis (PCA, see Chapter 6) is performed to reduce, appropriately, the dimensionality of the feature space and linear discriminant analysis (LDA) is then performed in the low-dimensional space, for example, [Belh 97]. A drawback of this technique is that during the dimension reduction phase part of the discriminatory information may be lost.

In [Yang 02] the mixture scatter matrix, S_m, is used in the J criterion in the place of S_w. It is shown that in this case, applying first a PCA on S_m, to reduce the dimensionality to the value of the rank of S_m, followed by an LDA in the reduced space, does not lead to any loss of information. In [Chen 00] the null space of the within-class scatter matrix is brought into the game. It has been observed that the null space, of S_w contains useful discriminant information. The method first projects onto the null space and, then, in the projected space the transformation that maximizes the between-class scatter is computed. A disadvantage of this approach is that it may lose information by considering the null space instead of S_w. A second problem is that the complexity of determining the null space of S_w is very high. Computational difficulties of the method are addressed in [Cevi 05]. In [Ye 05], in the first stage, dimensionality reduction is achieved by maximizing the between-class cluster (S_b), via a QR decomposition technique. In the second stage, a refinement is achieved by focusing on the within-class scatter issue, following arguments similar to the classical LDA. A unifying treatment of a number from the previous techniques is considered in [Zhan 07].

A different approach is proposed in [Li 06]. Instead of the J_3 criterion, another criterion is introduced that involves the trace of the difference of the involved matrices, thus bypassing the need for inversions.

Besides the small sample size problem, another issue associated with the LDA is that the number of features that can be generated is at most one less than the number of classes. As we have seen, this is due to the rank of the matrix product $S_w^{-1}S_b$. For an M-class problem, there are only $M-1$ nonzero eigenvalues. All the J_3 related discriminatory information can be recovered by projecting onto the subspace generated by the eigenvectors associated with these nonzero eigenvalues. Projecting on any other direction adds no information.

Good insight into it can be gained through geometry by considering a simple example. Let us assume, for simplicity, a two-class task with classes normally distributed with covariance matrices equal to the identity matrix. Then by its definition, S_w is also an identity matrix. It is easy to show (Problem 5.20) that in this case the eigenvector corresponding to the only nonzero eigenvalue is equal to $\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2$. The (Euclidean) distance between the mean values of the projection points in the (nonzero) eigenvector direction is the same as the distance between the mean values of the classes in the original space, i.e., $||(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)||$. This can easily be deduced by visual inspection of Figure 5.7, which corresponds to a case such as is discussed our example. Projecting on the orthogonal direction adds no information since the classes coincide. All the scatter information, with respect to both classes, is obtained from a single direction.

Due to the previous drawback, there are cases where the number of classes M is small, and the resulting number of, at most, $M-1$ features is insufficient. An attempt to overcome this difficulty is given in [Loog 04]. The main

idea is to employ a different to S_b measure to quantify the between-class scatter. The Chernoff distance (closely related to the Bhattacharyya distance of Section 5.6.2) is employed. This change offers the possibility of reducing the dimensionality to any dimension l smaller than the original m. A different path is followed in [Kim 07]. From the original m features, the authors build a number of so-called composite vectors. Each vector consists of a subset of the m features. Different composite vectors are allowed to share some of the original features. LDA is then performed on this new set of feature vectors. This procedure enhances the range of the rank of the involved matrix product beyond $M-1$. In [Nena 07], the shortcomings of LDA are overcome by defining a new class-separability measure based on an information-theoretic cost inspired by the concept of mutual information.

- No doubt, scattering matrix criteria are not the only ones that can be used to compute the optimal transformation matrix. For example, [Wata 97] suggested using ta different transformation matrix for each class and optimizing with respect to the classification error. This is within the spirit of the recent trend, to optimize directly with respect to the quantity of interest, which is the classification error probability. For the optimization, smooth versions of the error rate are used to guarantee differentiability. Other ways to compute the transformation matrix will be discussed in the next chapter.

- Besides the linear nonlinear transformations can also be employed for optimal feature selection. For example, in [Samm 69] a nonlinear technique is proposed that attempts to preserve maximally all the distances between vectors. Let $\boldsymbol{x}_i, \boldsymbol{y}_i, i = 1, 2, \ldots, N$, be the feature vectors in the original m-dimensional and the transformed l-dimensional space, respectively. The transformation into the lower dimensional space is performed so as to maximize

$$J = \frac{1}{\sum_{i=1}^{N-1}\sum_{j=i+1}^{N} d^o(i,j)} \sum_{i=1}^{N-1}\sum_{j=i+1}^{N} \frac{\left(d^o(i,j) - d(i,j)\right)^2}{d^o(i,j)} \tag{5.50}$$

  where $d^o(i,j)$, $d(i,j)$ are the (Euclidean) distances between vectors $\boldsymbol{x}_i$, and $\boldsymbol{x}_j$ in the original space and $\boldsymbol{y}_i$, $\boldsymbol{y}_j$ in the transformed space, respectively.

- Another nonlinear generalization of the method consists of two (implicit) steps. First, one employs a nonlinear vector function to transform the input feature space into a higher-dimensional one, which can even be of infinite dimension. Then, the linear discriminant method is applied in this high-dimensionality space. However, the problem formulation is done so that vectors appear only via inner products. This allows the use of kernel functions to facilitate computations, as was the case with the nonlinear support vector machines presented in Chapter 4 [Baud 00, Ma 03].

5.9 NEURAL NETWORKS AND FEATURE GENERATION/SELECTION

Recently, efforts have been made to use neural networks for feature generation and selection. A possible solution is via the so-called *auto-associative networks*. A network is employed having m input and m output nodes and a single hidden layer with l nodes with linear activations. During training, the desired outputs are the same as the inputs. That is,

$$\mathcal{E}(i) = \sum_{k=1}^{m} (\hat{y}_k(i) - x_k(i))^2$$

where the notation of the previous chapter has been adopted. Such a network has a unique minimum, and the outputs of the hidden layer constitute the projection of the input m-dimensional space onto an l-dimensional subspace. In [Bour 88] it is shown that this is basically a projection onto the subspace spanned by the l principal eigenvectors of the input correlation matrix, a topic that will be our focus in the next chapter. An extension of this idea is to use three hidden layers [Kram 91]. Such a network performs a nonlinear principal component analysis. The major drawback of such an architecture is that nonlinear optimization techniques have to be used for the training. Besides the computational load, the risk of being trapped in local minima is always present.

An alternative is to use neural networks, or any other (non)linear structure, to exploit properties of the LS cost function. In Chapter 3, we saw that the outputs of a network approximate posterior probabilities, provided that the weights have been trained so that the outputs match, in the LS sense, the class labels. In [Lowe 91] it is pointed out that, besides this property, another very interesting one is also valid. A multilayer perceptron was considered with linear output nodes. The network was trained to minimize the squared error between the actual and desired responses (i.e., class labels 1 and 0). It was shown that minimizing the squared error is equivalent to maximizing the criterion

$$J = \text{trace}\{S_m^{-1} S_b\} \tag{5.51}$$

where S_m is the mixture scatter matrix of the vectors formed by the outputs of the last hidden layer nodes and S_b the corresponding between-class scatter matrix in a weighted form (Problem 5.22). If the inverse of S_m does not exist, it is replaced by its pseudoinverse. In other words, such a network can be used as a *J-optimal nonlinear* transformer of the input m-dimensional vectors into l-dimensional vectors, where l is the number of nodes in the last hidden layer.

Another approach is to employ neural networks to perform the computations associated with the optimization of various class separability criteria discussed in this chapter. Although these techniques do not necessarily provide new approaches, the incorporation of neural networks offers the capability of adaptation in case the statistics of the input data are slowly varying. In [Chat 97, Mao 95] a number of such techniques are developed. The idea behind most of these techniques is to use

a network that iteratively computes eigenvectors of correlation matrices, a step that, as we have seen, is at the heart of a number of optimality criteria.

An alternative technique has been suggested in [Lee 93, Lee 97]. They have shown that the discriminantly informative feature vectors have a component that is normal to the decision surface at least at one point on the surface. Furthermore, the less informative vectors are orthogonal to a vector normal to the decision surface at every point of the surface. This is natural, because vectors that do not have a component normal to the decision surface cannot cross it (and hence change classes) whatever their value is. Based on this observation, they estimate normal vectors to the decision boundary, using gradient approximation techniques, which are then used to formulate an appropriate eigenvalue–eigenvector problem leading to the computation of the transformation matrix.

Finally, pruning a neural network is a form of feature selection integrated into the classifier design stage. Indeed, the weights of the input nodes corresponding to less important features are expected to be small. As discussed in Chapter 4, the incorporation of appropriate regularization terms in the cost function encourages such weights to converge to zero and ultimately to be eliminated. This approach was followed, for example, in [Seti 97].

5.10 A HINT ON GENERALIZATION THEORY

So far in this book, two major issues have occupied us: the design of the classifier and its generalization capabilities. The design of the classifier involved two stages: the choice of the classifier type and the choice of the optimality criterion. The generalization capabilities led us to seek ways to reduce the feature space dimensionality. In this section we will point out some important theoretical results that relate the size N of the training data set and the generalization performance of the designed classifier.

To this end, let us summarize a few necessary basic steps and definitions.

- Let $\mathcal{F}$ be the set of all the functions f that can be realized by the adopted classifier scheme. For example, if the classifier is a multilayer perceptron with a given number of neurons, then $\mathcal{F}$ is the set of all the functions that can be realized by the specific network structure. Functions f are mappings from $\mathcal{R}^l \rightarrow \{0, 1\}$. Thus, the response is either 1 or 0; that is, the two-class problem is considered, and the mapping is either $f(\boldsymbol{x}) = 1$ or $f(\boldsymbol{x}) = 0$.
- Let $P_e^N(f)$ be the *empirical* classification error probability, based on the available input—desired output training pairs $(\boldsymbol{x}_i, y_i)$, $i = 1, 2, \ldots, N$, which are considered to be independent and identically distributed (i.i.d.). Thus, $P_e^N(f)$ is the fraction of training samples for which an error occurs, that is, $f(\boldsymbol{x}_i) \neq y_i$. Obviously, this depends on the specific function f and the size N. The optimal function that results from minimizing this empirical cost is denoted by f^* and belongs to the set $\mathcal{F}$.

- $P_e(f)$ is the true classification error probability when a function f is realized. The corresponding empirical $P_e^N(f)$ can be very small, even zero, since a classifier can be designed to classify all training feature vectors correctly. However, $P_e(f)$ is the important performance measure, because it measures error probability based on the statistical nature of the data and not on the specific training set only. *For a classifier with good generalization capabilities, we expect the empirical and the true error probabilities to be close.* $P_e(f)$ is sometimes known as the *generalization error* probability.
- P_e denotes the minimum error probability over all the functions of the set, that is, $P_e = \min_{f\in\mathcal{F}} P_e(f)$.[1] Again, in practice we would like the optimal empirical error $P_e^N(f^*)$ to be close to P_e.

The Vapnik–Chervonenkis theorem is as follows.

Theorem *Let $\mathcal{F}$ be the class of functions of the form $\mathcal{R}^l \to \{0, 1\}$. Then the empirical and true error probabilities corresponding to a function f in the class, satisfy*

$$prob\{\max_{f\in\mathcal{F}} |P_e^N(f) - P_e(f)| > \epsilon\} \leq 8\mathcal{S}(\mathcal{F}, N)\exp(-N\epsilon^2/32) \tag{5.52}$$

The term $\mathcal{S}(\mathcal{F}, N)$ is called the *shatter* coefficient of the class $\mathcal{F}$. This is defined as the *maximum* number of dichotomies of N points that can be formed by the functions in $\mathcal{F}$. From our combinatorics basics, we know that the maximum number of dichotomies on a set of N points (separating them into two distinct subsets) is 2^N. However, not all these combinations can be implemented by a function $f : \mathcal{R}^l \to \{0, 1\}$. For example, we know that, in the two-dimensional space, the set of functions realized by a perceptron (hyperplane) can form only fourteen distinct dichotomies on four points out of the $16 = 2^4$ possibilities. The two XOR combinations cannot be realized. However, the class of functions realized by the perceptron can form all possible $8 = 2^3$ dichotomies for $N = 3$ points. This leads us to the following definition

Definition 1. *The largest integer $k \geq 1$ for which $\mathcal{S}(\mathcal{F}, k) = 2^k$ is called the Vapnik–Chervonenkis, or VC dimension of the class $\mathcal{F}$, and is denoted by V_c. If $\mathcal{S}(\mathcal{F}, N) = 2^N$ for every N, then the VC dimension is infinite.*

Thus, in the two dimensional space, the VC dimension of a single perceptron is 3. In the general l-dimensional space case, the VC dimension of a perceptron is $l + 1$, as is easily verified from Section 4.13. It will not come as a surprise to say that the VC dimension and the shatter coefficient are related, because they have common origins. Indeed, this is true. It turns out that if the VC dimension is finite, then the following bound is valid

$$\mathcal{S}(\mathcal{F}, N) \leq N^{V_c} + 1 \tag{5.53}$$

[1] Strictly speaking, in this section *inf* must be used instead of *min* and *sup* instead of *max*.

That is, the shatter coefficient is either 2^N or is bounded as given in (5.53). This bound has a very important implication for the usefulness of (5.52). Indeed, for finite VC dimensions (5.53) guarantees that for large enough N the shatter coefficient is bounded by *polynomial growth*. Then the bound in (5.52) is dominated by its exponential decrease, and it tends to zero as $N \to \infty$. In words, *for large N the probability of having large differences between the empirical and the true probability errors is very small*! *Thus, the network guarantees good generalization performance for large N*. Furthermore, the theory guarantees another strong result [Devr 96]

$$\text{prob}\{P_e(f^*) - \min_{f \in \mathcal{F}} P_e(f) > \epsilon\} \leq 8\mathcal{S}(\mathcal{F}, N) \exp(-N\epsilon^2/128) \tag{5.54}$$

That is, for large N we expect with high probability the performance of the empirical error optimal classifier to be close to the performance of the optimal one, over the specific class of functions.

Let us look at these theoretical results from a more intuitive perspective. Consider two different networks with VC dimensions $V_{c1} << V_{c2}$. Then if we fix N and ϵ, we expect the first network to have better generalization performance, because the bound in (5.52) will be much tighter. Thus, the probability that the respective empirical and true errors will differ more than the predetermined quantity will be much smaller. We can think of the VC dimension as an *intrinsic capacity of a network, and only if the number of training vectors exceeds this number sufficiently can we expect good generalization performance*.

Learning theory is rich in bounds that have been derived and that relate quantities such as the empirical error, the true error probability, the number of training vectors, and the VC dimension or a VC related quantity. In his elegant theory of learning, Valiant [Vali 84] proposed to express such bounds in the flavor of statistical tests. That is, the bounds involve an error ϵ, such as in Eqs. (5.52) and (5.54), and a confidence probability level that the bound holds true. Such bounds are known as PAC bounds, which stands for Probably (the probability of the bound to fail is small) Approximately Correct (when the bound holds, the error is small). A very interesting (for our purposes) bound that can be derived refers to the minimum number of training points that guarantee, with high probability, the design of a classifier with good error performance. Let us denote this minimum number of points as $N(\epsilon, \rho)$. It can be shown that if

$$N(\epsilon, \rho) \leq \max\left(\frac{k_1 V_c}{\epsilon^2} \ln \frac{k_2 V_c}{\epsilon^2}, \frac{k_3}{\epsilon^2} \ln \frac{8}{\rho}\right) \tag{5.55}$$

then for any number of training points $N \geq N(\epsilon, \rho)$ the optimal classifier, f^*, resulting by minimizing the empirical error probability $P_e^N(f)$ satisfies the bound

$$P\{P_e(f^*) - P_e > \epsilon\} \leq \rho \tag{5.56}$$

where k_1, k_2, k_3 are constants [Devr 96]. In other words, for small values of ϵ and ρ, if $N \geq N(\epsilon, \rho)$, the performance of the optimum empirical error classifier is guaranteed,

with high probability, to be close to the optimal classifier in the class of functions $\mathcal{F}$, realized by the specific classification scheme. The number $N(\epsilon, \rho)$ is also known as *sample complexity*. Observe that the first of the two terms in the bound has a linear dependence on the VC dimension and an inverse quadratic dependence on the error ϵ. Doubling, for example, the VC dimension roughly requires that we need to double the number of training points in order to keep the same ϵ and confidence level. On the other hand, doubling the accuracy (i.e., $\epsilon/2$) requires us to quadruple the size of the training set. The confidence level ρ has a little influence on the bound, due to its logarithmic dependence. Thus, high VC dimension sets high demands on the number of training points required to guarantee, with high probability, a classifier with good performance.

Another related bound of particular interest to us that holds with a probability at least $1 - \rho$ is the following:

$$P_e(f) \leq P_e^N(f) + \phi\left(\frac{V_c}{N}\right) \tag{5.57}$$

where V_c is the VC dimension of the corresponding class and

$$\phi\left(\frac{V_c}{N}\right) \equiv \sqrt{\frac{V_c\left(\ln\left(\frac{2N}{V_c}+1\right)\right) - \ln(\rho/4)}{N}} \tag{5.58}$$

The interested reader may obtain more bounds and results concerning the Vapnik–Chervonenkis theory from [Devr 96, Vapn 95]. It will take some effort, but it is worth it! In some of the published literature, the constants in the bounds are different. This depends on the way the bounds are derived. However, this is not of major practical importance, since the essence of the theory remains the same.

Due to the importance of the VC dimension, efforts have been made to compute it for certain classes of networks. In [Baum 89] it has been shown that the VC dimension of a multilayer perceptron with hard limiting activation functions in the nodes is bounded by

$$2\left[\frac{K_n^b}{2}\right] l \leq V_c \leq 2K_w \log_2(eK_n) \tag{5.59}$$

where K_n^b is the total number of hidden layer nodes, K_n the total number of nodes, K_w the total number of weights, l the input space dimension, e the base of the natural logarithm, and $[\cdot]$ the floor operator that gives the largest integer less than its argument. The lower bound holds only for networks with a single hidden layer and full connectivity between the layers. A similar upper bound is true for RBF networks too. Looking at this more carefully, one can say that for such networks the VC dimension is roughly given by the number of weights of the network, that is, the number of its free parameters to be determined! In practice, good generalization performance is expected if the number of training samples is a few times the VC dimension. A good rule of thumb is to choose N to be of the order of 10 times the VC dimension or more [Hush 93].

Besides the Vapnik–Chervonenkis theory, the published literature is rich in results concerning aspects of designing various classifiers using a finite data set N. Although they lack the elegance of the generality of the Vapnik–Chervonenkis theory, they provide further insight into this important task. For example, in [Raud 91] asymptotic analytic results are derived for a number of classifiers (linear, quadratic, etc.) under the Gaussian assumption of the involved densities. The classification error probability of a classifier designed using a finite set of N training samples is larger, by an amount Δ_N, than the error of the same classifier designed using an infinite ($N \to \infty$) set of data. It is shown that the mean of Δ_N (over different design sets) decreases as N tends to infinity. The rate of decrease depends on the specific type of classifier, on the dimensionality of the problem, and also on the value of the asymptotic ($N \to \infty$) error. It turns out that in order to keep the mismatch Δ_N within certain limits, *the number N of design samples must be a number of times larger than the dimension l*. Depending on the type of classifier, this proportionality constant can range from low values (e.g., 1.5) up to a few hundred! Furthermore, in [Fuku 90] it is shown that keeping N constant and increasing l, beyond a point, results in an increase of the classification error. This is known as the *Hughes phenomenon*, and it was also discussed in Section 5.3.

All these theoretical results provide useful guidelines in selecting appropriate values for N and l for certain types of classifiers. Moreover, they *make crystal clear the urge to keep the number of features as small as possible with respect to N and the importance of the feature selection stage in the design of a classification system*. In the fringes of this theoretical "happening," a substantial research effort has also been devoted to experimental comparisons, involving different classifiers, with respect to their generalization capabilities; see, for example, [Mama 96] and the references therein. In practice, however, experience and empirical results retain an important part in the final decision. Engineering still has a flavor of art!

Structural Risk Minimization

In our discussion so far, we have focused on the effects of the finite size of the training data set, N, for a given class of functions, that is, a given classifier structure. Let us now touch on another important issue. If we allow N to grow indefinitely, does this luxury provide us with the means not only to have good generalization properties but also to improve our classification error so as to approach the optimal Bayesian performance? Recall that as N grows, we can expect to obtain the optimal performance with respect to all *allowable* sets of classifiers that can be implemented by the chosen network structure. However, the error rate of the corresponding optimal classifier may still be very far from that of the Bayesian classifier. Let us denote by P_B the Bayesian error probability. Then we can write

$$P_e(f^*) - P_B = \big(P_e(f^*) - P_e\big) + (P_e - P_B) \tag{5.60}$$

A diagrammatic interpretation of Eq. (5.60) is given in Figure 5.8. The right-hand side in Eq. (5.60) consists of two conflicting terms. If the class $\mathcal{F}$ is too small, then

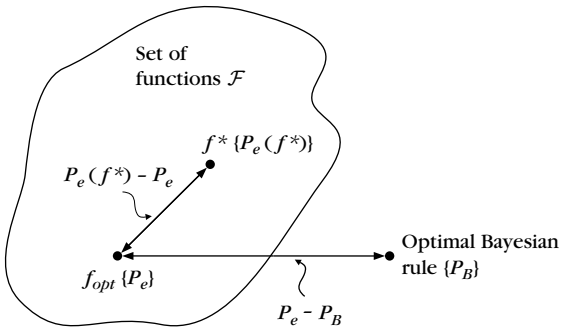

FIGURE 5.8

Diagrammatic interpretation of Eq. (5.60). The optimal function over the set $\mathcal{F}$, associated with the minimum error P_e, is denoted as f_{opt}, and f^* is the optimal function resulting from the empirical cost for a given N.

the first term is expected to be small, but the second term is likely to be large. If, on the other hand, the class of functions $\mathcal{F}$ is large, then the second term is expected to be small but the first term is probably large. This is natural, because the larger the set of functions, the higher the probability of including in the set a good approximation of the Bayesian classifier. Moreover, the smaller the class, the less the variation between its members. This reminds us of the bias–variance dilemma we discussed in Chapter 3. A little thought suffices to reveal that the two problems are basically the same, seen from a different viewpoint. Then the natural question arises once more, can we make both terms small and how? The answer is that this is possible only asymptotically, provided that *at the same time* the size of the class $\mathcal{F}$ grows appropriately. An elegant strategy to achieve this has been suggested by Vapnik and Chervonenkis [Vapn 82].

Let $\mathcal{F}^{(1)}, \mathcal{F}^{(2)}, \ldots$ be a sequence of nested classes of functions, that is,

$$\mathcal{F}^{(1)} \subset \mathcal{F}^{(2)} \subset \mathcal{F}^{(3)} \subset \cdots \tag{5.61}$$

with an increasing, yet finite, VC dimension,

$$V_{c,\mathcal{F}^{(1)}} \leq V_{c,\mathcal{F}^{(2)}} \leq V_{c,\mathcal{F}^{(3)}} \leq \cdots \tag{5.62}$$

Also let

$$\lim_{i \to \infty} \inf_{f \in \mathcal{F}^{(i)}} P_e(f) = P_B \tag{5.63}$$

For each N and class of functions $\mathcal{F}^{(i)}$, $i = 1, 2, \ldots$, compute the optimum, $f^*_{N,i}$, with respect to the *empirical* error using the N training pairs of input–output samples. Vapnik and Chervonenkis suggest choosing for each N the function f^*_N according to the *structural risk minimization principle* (SRM). This consists of the following two steps. First we select the classifier $f^*_{N,i}$ from every class $\mathcal{F}^{(i)}$ that minimizes

the corresponding empirical error over the class of functions. Then, from all these classifiers, we choose the one that minimizes the upper bound in (5.57), over all i. More precisely, form the so-called *guaranteed error bound*,

$$\tilde{P}_e(f^*_{N,i}) \equiv P_e^N(f^*_{N,i}) + \phi\left(\frac{V_{c,\mathcal{F}^{(i)}}}{N}\right) \tag{5.64}$$

and choose

$$f^*_N = arg\min_i \tilde{P}_e(f^*_{N,i}) \tag{5.65}$$

Then, as $N \to \infty$, $P_e(f^*_N)$ tends to P_B with probability one. Note that the second term in the minimized bound, $\phi\left(\frac{V_{c,\mathcal{F}^{(i)}}}{N}\right)$, is a *complexity penalty term* that increases as the network complexity increases (i.e., with the size of the class of functions and $V_{c,\mathcal{F}^{(i)}}$). If on one hand the classifier model is too simple, the penalty term is small but the empirical error term will be large in (5.64). On the other hand, if the model is complex, the empirical error is small but the penalty term large. The structural risk minimization criterion aims at achieving the best trade-off between these two terms. This is illustrated in Figure 5.9.

From this point of view, the structural risk minimization principle belongs to a more general class of approaches that try to estimate the order of a system, by considering simultaneously the model complexity and a performance index. Depending

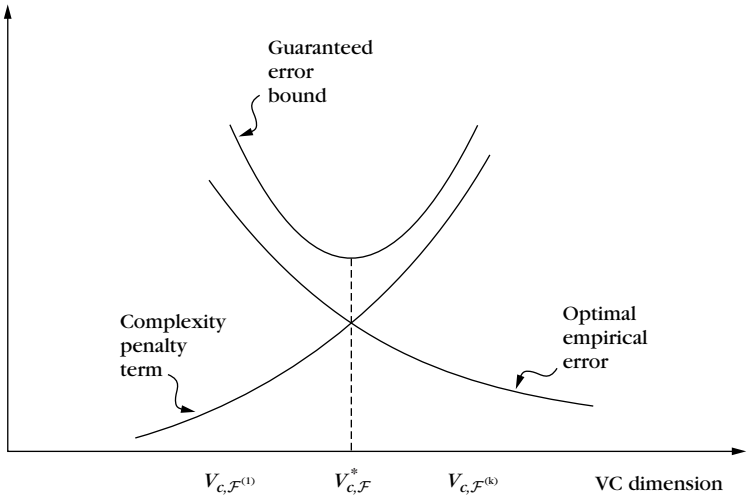

FIGURE 5.9

For a fixed N, the complexity penalty term increases and the optimal empirical error decreases as the VC dimension of the model increases. Choosing the model according to the SRM principle aims at achieving the best trade-off between these two terms that corresponds to the minimum of the guaranteed error bound. Note that, $V_{c,\mathcal{F}^{(1)}} < V^*_{c,\mathcal{F}} < V_{c,\mathcal{F}^{(k)}}$, which implies $\mathcal{F}^{(1)} \subset \mathcal{F}^* \subset \mathcal{F}^{(k)}$.

on the function used to measure the model complexity and the corresponding performance index, different criteria result. For example, in the Akaike Information Criterion [Akai 74], the place of the empirical error is taken by the value of the log-likelihood function, corresponding to the maximum likelihood estimates of the unknown parameters, and the complexity term is proportional to the number of free parameters to be estimated. See also Sections 5.11 and 16.4.1.

An alternative interpretation of the SVM cost function in Eq. (3.93) is given in [Lin 02]. It is treated as a typical regularization method with two components a data fit functional term $\left(\sum_i I(\xi_i)\right)$ and a regularization penalty term ($||\boldsymbol{w}||^2$). The latter is the complexity-related component and is used to guard against overfitting. In general, the data-fit term approaches a limiting functional as $N \to \infty$. Under some general conditions, the estimate resulting from the regularization method is the minimizer of this data-fit-limiting functional, as $N \to \infty$. It turns out that in the SVM case the minimizer of the limiting data-fit functional is the Bayes optimal rule and the SVM solution approaches it, as $N \to \infty$, provided that the kernel choice guarantees a rich enough space (RKHS) and the smoothing parameter, C, is chosen appropriately. This interesting result nicely ties the SVM and the Bayesian optimal rule. Would it be an exaggeration to say that a good theory is like a good piece of art, in the sense that both support more than one interpretation?

Remarks

- The SRM procedure provides a theoretical guideline for constructing a classifier that converges asymptotically to the optimal Bayesian one. However, the bound in (5.57), which is exploited in order to reach this limit, must not be misinterpreted. For any bound to be useful in practice, one needs an extra piece of information. Is this bound loose or tight? In general, until now, no result has provided this information. We can construct classifiers whose VC dimension is large, yet their performance can be good. A typical example is the nearest neighbor (*NN*) classifier. Its VC dimension is infinite. Indeed, since we know the class label of all the N training points, the *NN* classifier classifies correctly all training points and the corresponding shatter coefficient is 2^N. Yet, it is generally accepted that the generalization performance of this classifier can be quite good in practice. In contrast, one can build a classifier with finite VC dimension, yet whose performance is always bad ([Burg 98]). Concluding this remark, we have to keep in mind that *if two classifiers have the same empirical error, it does not, necessarily, mean that the one with the smaller VC dimension leads to better performance.*
- Observe that in all bounds given previously no assumptions have been made about the statistical distribution underlying the data set. That is, they are *distribution-free* bounds.

Support Vector Machines: A Last Touch

We have already discussed that the VC dimension of a linear classifier in the l-dimensional space is $l + 1$. However, hyperplanes that are *constrained* to leave the maximum margin between the classes *may have a smaller VC dimension*.

Let us assume that r is the radius of the *smallest* (hyper)sphere that encloses all the data (Problem 5.23), that is,

$$\|\boldsymbol{x}_i\| \leq r, \ i = 1, 2, \ldots, N$$

Then if a hyperplane satisfies the conditions in Eq. (3.73) and

$$\|\boldsymbol{w}\|^2 \leq c$$

where c is a constant, then its VC dimension, V_c, is bounded by ([Vapn 98])

$$V_c \leq \min(r^2 c, l) + 1 \tag{5.66}$$

That is, the capacity of the classifier can be *controlled independently of the dimensionality* of the feature space. This is very interesting indeed. It basically states that the capacity of a classifier may not, necessarily, be related to the number of unknown parameters! This is a more general result. To emphasize it further, note that it is possible to construct a classifier with only one free parameter, yet with infinite VC dimension; see, for example, [Burg 98]. Let us now consider a sequence of bounds

$$c_1 < c_2 < c_3 < \ldots$$

This defines the following sequence of classifiers:

$$\mathcal{F}^i : \left\{ \boldsymbol{w}^T \boldsymbol{x} + w_0 : \|\boldsymbol{w}\|^2 \leq c_i \right\} \tag{5.67}$$

where

$$\mathcal{F}^i \subset \mathcal{F}^{i+1}$$

If the classes are separable, then the empirical error is zero. Minimizing the norm $\|\boldsymbol{w}\|$ is equivalent to minimizing the VC dimension (to be fair, the upper bound of the VC dimension). Thus, we can conclude that, the design of an SVM classifier senses the spirit of the SRM principle. Hence, keeping the VC dimension minimum suggests that we can expect support vector machines to exhibit good generalization performance. More on these issues can be found in [Vapn 98, Burg 98].

The essence of all formulas and discussion in this section is that the generalization performance and accuracy of a classifier depend heavily on two parameters: the VC dimension and the number of the available feature vectors used for the training. The VC dimension may or may not be related to the number of free parameters describing the classifier. For example, in the case of the perceptron linear classifier the VC dimension coincides with the number of free parameters. However, one can

construct nonlinear classifiers whose VC dimension can be either lower or higher than the number of free parameters [Vapn 98, p. 159]. The design methodology of the SVM allows one to "play" with the VC dimension (by minimizing $\|\boldsymbol{w}\|$, Eq. (5.66)), leading to good generalization performance, although the design may be carried out in a high- (even infinite) dimensional space.

Digging this fertile ground in a slightly different direction, using tools from the PAC theory of learning one can derive a number of distribution-free and *dimension-free* bounds. These bounds bring into the surface a key property underlying the SVM design; that is, that of the *maximum margin* (SVMs are just one example of this larger family of classifiers, which are designed with an effort to maximize the margin the training points leave from the corresponding decision surface). (See also the discussion at the end of Chapter 4.) Although a more detailed treatment of this topic is beyond the scope of this book, we will provide two related bounds that reinforce this, at first surprising, property of the "emancipation" of the generalization performance from the feature space dimensionality.

Assume that all available feature vectors lie within a sphere of radius R (i.e., $\|\boldsymbol{x}\| \leq R$). Let, also, the classifier be a *linear* one, normalized so that $\|\boldsymbol{w}\| = 1$, designed using N randomly chosen training vectors. If the resulting classifier has a margin of 2γ (according to the margin definition in Section 3.7.1) and *all training vectors* lie outside the margin, the corresponding true error probability (generalization error) is no more than

$$\frac{c}{N}\left(\frac{R^2}{\gamma^2}\ln^2 N + \ln\left(\frac{1}{\rho}\right)\right) \tag{5.68}$$

where c is a constant, and this bound holds true with a probability at least $1-\rho$. Thus, adjusting the margin, as the SVM does, to be maximum we improve the bound, and this can be carried out even in an infinite dimensional space if the adopted kernel so dictates [Bart 99, Cris 00]. This result is logical. If the margin is large on a set of training points randomly chosen, this implies a classifier with large confidence, thus leading with high probability to good performance.

The bound given previously was derived under the assumption that all training points are correctly classified. Furthermore, the margin constraint implies that for all training points $y_i f(\boldsymbol{x}_i) \geq \gamma$, where $f(\boldsymbol{x})$ denotes the linear classifier (the decision is taken according to $\text{sign}(f(\boldsymbol{x}))$. A very interesting related bound refers to the more realistic case, where some of the training points are misclassified. Let k be the number of points with $y_i f(\boldsymbol{x}_i) < \gamma$. (The product $yf(\boldsymbol{x})$ is also known as the *functional* margin of the pair $(y, \boldsymbol{x})$ with respect to classifier $f(\boldsymbol{x})$) Obviously, this also allows for negative values of the product. It can be shown that with probability at least $1-\rho$ the true error probability is upper bounded by ([Bart 99, Cris 00])

$$\frac{k}{N} + \sqrt{\frac{c}{N}\left(\frac{R^2}{\gamma^2}\ln^2 N + \ln\left(\frac{1}{\rho}\right)\right)} \tag{5.69}$$

Another bound relates the error performance of the SVM classifier with the number of support vectors. It can be shown [Bart 99] that if N is the number of training vectors and N_s the number of support vectors, the corresponding true error probability is bounded by

$$\frac{1}{N - N_s}\left(N_s \log_2 \frac{eN}{N_s} + \log_2 \frac{N}{\rho}\right) \tag{5.70}$$

where e is the base of the natural logarithm and the bound holds true with a probability at least $1 - \rho$. Note that this bound is also independent of the dimension of the feature space, where the design takes place. The bound increases with N_s and this must make the user, who has designed an SVM that results in a relatively large number (with respect to N) of support vectors, cautious and "suspicious" about the performance of the resulting SVM classifier.

The previous three bounds indicate that the error performance is controlled by both N_s and γ. In practice, one may end up, for example, with a large number of support vectors and at the same time with a large margin. In such a case, the error performance could be assessed, with high confidence, depending on which of the two bounds has lower value.

5.11 THE BAYESIAN INFORMATION CRITERION

The structural risk minimization principle, discussed in the previous section, belongs to a more general class of methods that estimate the order of a system by considering, *simultaneously*, the model complexity and a performance index. Depending on the function used to measure the model complexity and the corresponding performance index, different criteria result. In this section we will focus on one of such criteria, which provides a Bayesian theory flavor to the model selection problem. Moreover, it has a structural form that resembles a number of other popular criteria that have been proposed over the years. Although the criteria of this "family" lack the elegance and generality of the SRM principle, they can be useful in a number of cases. Furthermore, they shed light on the "performance versus complexity" trade-off task from another perspective.

Let $\mathcal{D} = \{(\boldsymbol{x}_i, y_i),\ i = 1, 2, \ldots, N\}$ be the training data set. We will focus on the Bayesian classification problem, and the goal is to adopt a *parametric* model for the class posterior probabilities; that is, $P(y_i|\boldsymbol{x}; \boldsymbol{\theta})$, $y_i \in \{1, 2, \ldots, M\}$ for an M class task. The cost used for the optimal choice of the unknown parameter $\boldsymbol{\theta}$ is the log-likelihood function computed over the training set; that is, $L(\boldsymbol{\theta}) = \ln p(\mathcal{D}|\boldsymbol{\theta})$.

Let $\mathcal{M}_m$ denote one of the possible models described by the set of parameters $\boldsymbol{\theta}_m$, where m runs over all candidate models. Let us also assume that for each model we know the prior information with respect to the distribution of $\boldsymbol{\theta}_m$, that is, the pdf $p(\boldsymbol{\theta}_m|\mathcal{M}_m)$. Our goal is to choose the model for which the posterior probability

$P(\mathcal{M}_m|\mathcal{D})$ becomes maximum, over all candidate models. Using the Bayes theorem, we have

$$P(\mathcal{M}_m|\mathcal{D}) = \frac{P(\mathcal{M}_m)p(\mathcal{D}|\mathcal{M}_m)}{p(\mathcal{D})} \tag{5.71}$$

If we further assume that all models are equiprobable, $P(\mathcal{M}_m)$ can be dropped out. The joint data pdf $p(\mathcal{D})$, which is the same for all models, can also be neglected and our goal becomes to maximize

$$p(\mathcal{D}|\mathcal{M}_m) = \int p(\mathcal{D}|\boldsymbol{\theta}_m, \mathcal{M}_m)p(\boldsymbol{\theta}_m|\mathcal{M}_m)d\boldsymbol{\theta}_m \tag{5.72}$$

Employing a series of assumptions (e.g., Gaussian distribution for $\boldsymbol{\theta}_m$) and the so-called Laplacian approximation to the integral ([Schw 79, Ripl 96]), and taking the logarithm of both sides in Eq. (5.72) results in

$$\ln p(\mathcal{D}|\mathcal{M}_m) = L(\hat{\boldsymbol{\theta}}_m) - \frac{K_m}{2}\ln N \tag{5.73}$$

where $L(\hat{\boldsymbol{\theta}}_m)$ is the log-likelihood function computed at the ML estimate, $\hat{\boldsymbol{\theta}}_m$, and K_m is the number of free parameters (i.e., the dimensionality of $\boldsymbol{\theta}_m$). Equivalently, one can search for the minimum of the quantity

$$\text{BIC} = -2L(\hat{\boldsymbol{\theta}}_m) + K_m \ln N \tag{5.74}$$

The criterion is known as the Bayesian information criterion (BIC) or the Schwartz criterion. In other words, the best model indicated by this criterion depends (a) on the value of the log-likelihood function at its maximum (i.e., the adopted performance index) and (b) on a term that depends on the complexity of the model and the number of data points. If the model is too simple to describe the distribution underlying the given data set, the first term in the criterion will have a large value, since the probability of having obtained the set $\mathcal{D}$ from such a model will be small. On the other hand, if the model is complex, with a large number of free parameters that can adequately describe the data, the first term will have a small value, which however, is penalized by a large value of the second term. BIC provides a trade-off between these two terms. It can be shown that BIC is asymptotically consistent. This means that if the family of the candidate models contains the true one, then as $N \longrightarrow \infty$ the probability that BIC will select the correct model tends to one.

The Akaike information criterion (AIC) [Akai 74], though derived in a different way, has similar structure, and the only difference lies in the second term, which is $2K_m$ instead of $K_m \ln N$ (see also Section 16.4.1). In practice, it is not clear which model is to be used. It has been reported that for large values of N AIC tends to choose models that are too complex. On the other hand, for smaller values of N BIC tends to choose models that are too simple [Hast 01]. Besides the previous two criteria, a number of alternatives have also been suggested, such as [Riss 83, Mood 92, Leth 96, Wang 98]. For a review of such techniques, see, for example, [Ripl 96, Hast 01, Stoi 04a, Stoi 04b].

5.12 PROBLEMS

5.1 In Trunk's example, discussed in Section 5.3, prove that the mean value and the variance of the variable z are given by $E[z] = ||\boldsymbol{\mu}||^2 = \sum_{i=1}^{l} \frac{1}{i}$ and $\sigma_z^2 = ||\boldsymbol{\mu}||^2$ respectively. Also show that the probability of error is

$$P_e = \int_{b_l}^{\infty} \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{z^2}{2}\right) dz \tag{5.75}$$

where

$$b_l = \sqrt{\sum_{i=1}^{l} \frac{1}{i}} \tag{5.76}$$

5.2 In Trunk's example, as in Problem 5.1, show that the mean value and variance of z, in the case of unknown mean value, are given the Eqs. (5.7) and (5.8), respectively. Derive the formula for the probability of error and show that it tends to 0.5 as the number of features tends to infinity.

5.3 If x_i, y_i, $i = 1, 2, \ldots, N$ are independent samples of two Gaussian distributions of the same variance σ^2, show that the random variable $\frac{(2N-2)s_z^2}{\sigma^2}$, where

$$s_z^2 = \frac{1}{2N-2}\left(\sum_{i=1}^{N}(x_i - \bar{x})^2 + \sum_{i=1}^{N}(y_i - \bar{y})^2\right)$$

where $\bar{x}, \bar{y}$ are the respective sample mean values, is chi-square distributed with $2N - 2$ degrees of freedom.

5.4 Let N_1, N_2 be the available values of a feature in two classes, respectively. The feature is assumed to follow a Gaussian distribution with the same variance in each class. Define the test statistic

$$q = \frac{(\bar{x} - \bar{y}) - (\mu_1 - \mu_2)}{s_z\sqrt{\frac{1}{N_1} + \frac{1}{N_2}}} \tag{5.77}$$

where

$$s_z^2 = \frac{1}{N_1 + N_2 - 2}\left(\sum_{i=1}^{N_1}(x_i - \bar{x})^2 + \sum_{i=1}^{N_2}(y_i - \bar{y})^2\right)$$

and μ_1, μ_2 are the respective true mean values. Show that q follows the t-distribution with $N_1 + N_2 - 2$ degrees of freedom.

5.5 Show that the matrix

$$A = \begin{bmatrix} \frac{1}{\sqrt{n}} & \frac{1}{\sqrt{n}} & \frac{1}{\sqrt{n}} & \cdots & \frac{1}{\sqrt{n}} \\ \frac{-1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 & \cdots & 0 \\ \frac{-1}{\sqrt{6}} & \frac{-1}{\sqrt{6}} & \frac{2}{\sqrt{6}} & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \frac{-1}{\sqrt{n(n-1)}} & \frac{-1}{\sqrt{n(n-1)}} & \frac{-1}{\sqrt{n(n-1)}} & \cdots & \frac{n-1}{\sqrt{n(n-1)}} \end{bmatrix}$$

is orthogonal, that is, $AA^T = I$.

5.6 Show that if $x_i, i = 1, 2, \ldots, l$, are jointly Gaussian, then the l variables $y_i, i = 1, 2, \ldots, l$, resulting from a linear transformation of them are also jointly Gaussian. Furthermore, if x_i are mutually independent and the transformation is orthogonal, then y_i are also mutually independent and Gaussian.

5.7 Let $\omega_i, i = 1, 2, \ldots, M$, be the classes for a classification task. Divide the interval of the possible values of a feature into subintervals $\Delta_j, j = 1, 2, \ldots, K$. If $P(\Delta_j)$ is the probability of having values in the respective subinterval and $P(\omega_i|\Delta_j)$, the probability of occurrence of ω_i in this interval, show that the so-called *ambiguity function*

$$A = -\sum_i \sum_j P(\Delta_j) P(\omega_i|\Delta_j) \log_M (P(\omega_i|\Delta_j))$$

is equal to 1 for completely overlapped distributions and is equal to 0 for perfectly separated ones. For all other cases it takes intermediate values. Thus, it can be used as a distribution overlap criterion [Fine 83].

5.8 Show that if $d_{ij}(x_1, x_2, \ldots, x_m)$ is the class divergence based on m features, adding a new one x_{m+1} cannot decrease the divergence, that is,

$$d_{ij}(x_1, x_2, \ldots, x_m) \leq d_{ij}(x_1, x_2, \ldots, x_m, x_{m+1})$$

5.9 Show that if the density functions are Gaussian in both classes with the same covariance matrix Σ, then on adding a new feature x_{m+1} to the feature vector the new divergence is recursively computed by

$$d_{ij}(x_1, \ldots, x_{m+1}) = d_{ij}(x_1, \ldots, x_m) + \frac{[(\mu_i - \mu_j) - (\boldsymbol{\mu_i} - \boldsymbol{\mu_j})^T \Sigma^{-1} \boldsymbol{r}]^2}{\sigma^2 - \boldsymbol{r}^T \Sigma^{-1} \boldsymbol{r}}$$

where μ_i, μ_j are the mean values of x_{m+1} for the two classes, σ^2 is its variance, $\boldsymbol{r}$ is its cross-covariance vector with the other elements of $\boldsymbol{x}$, and $\boldsymbol{\mu}_i, \boldsymbol{\mu}_j$ are the mean vectors of $\boldsymbol{x}$ prior to x_{m+1}. If x_{m+1} is now uncorrelated with the previously selected features $x_1, \ldots, x_m$, then this becomes

$$d_{ij}(x_1, \ldots, x_{m+1}) = d_{ij}(x_1, \ldots, x_m) + \frac{(\mu_i - \mu_j)^2}{\sigma^2}$$

5.10 Show that if the features are statistically independent, then the divergence is given by

$$d_{ij}(x_1, x_2, \ldots, x_l) = \sum_{i=1}^{l} d_{ij}(x_i)$$

5.11 Show that in the case of Gaussian distributions the Chernoff bound becomes

$$\epsilon_{CB} = \exp(-b(s))$$

where

$$b(s) = \frac{s(1-s)}{2}(\boldsymbol{\mu}_i - \boldsymbol{\mu}_j)^T[s\Sigma_j + (1-s)\Sigma_i]^{-1}(\boldsymbol{\mu}_i - \boldsymbol{\mu}_j) + \frac{1}{2}\ln\frac{|\, s\Sigma_j + (1-s)\Sigma_i \,|}{|\, \Sigma_j \,|^s |\, \Sigma_i \,|^{1-s}}$$

Then take the derivative with respect to s and show that for equal covariance matrices the optimum is achieved for $s = 1/2$. Thus, in this case $b(s)$ equals the Bhattacharyya distance.

5.12 Show that the mixture scatter matrix is the sum of the within-class and between-class scatter matrices.

5.13 Show that the cross-correlation coefficient in (5.29) lies in the interval $[-1, 1]$. *Hint:* Use Schwartz's inequality $|\boldsymbol{x}^T\boldsymbol{y}| \leq \|\boldsymbol{x}\|\|\boldsymbol{y}\|$.

5.14 Show that for a two-class problem and Gaussian distributed feature vectors, with the same covariance matrix in the two classes, which are assumed equiprobable, the divergence is equal to

$$\text{trace}\{S_w^{-1}S_b\}$$

5.15 Show that the number of combinations to be searched using the backward search technique is given by

$$1 + 1/2((m+1)m - l(l+1))$$

5.16 Show that the optimal solution of the generalized Rayleigh quotient in (5.37) satisfies (5.38).

5.17 Show that

$$\frac{\partial}{\partial A}\text{trace}\{(A^TS_1A)^{-1}(A^TS_2A)\} = -2S_1A(A^TS_1A)^{-1}(A^TS_2A)(A^TS_1A)^{-1} + 2S_2A(A^TS_1A)^{-1}$$

5.18 Show that for an M-class problem the matrix S_b is of rank $M-1$. *Hint:* Recall that $\boldsymbol{\mu}_0 = \sum_i P_i\boldsymbol{\mu}_i$.

5.19 Show that if $f_i(\boldsymbol{x}), i = 1, \ldots, M$, are the discriminant functions of an M-class problem, we can construct from them $M - 1$ new functions that are, in principle, sufficient for the classification.
Hint: Consider the differences $f_i(\boldsymbol{x}) - f_j(\boldsymbol{x})$.

5.20 Show that for a two-class problem the nonzero eigenvalue of matrix $S_w^{-1}S_b$ is equal to

$$\lambda_1 = P_1P_2(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)^T S_{xw}^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)$$

and the corresponding eigenvector

$$\boldsymbol{v}_1 = S_{xw}^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)$$

where P_1, P_2 are the respective class probabilities.

5.21 Show that if matrices Σ_1, Σ_2 are two covariance matrices, then the eigenvectors of $\Sigma_1^{-1}\Sigma_2$ are orthogonal with respect to Σ_1, that is,

$$\boldsymbol{v}_i^T \Sigma_1 \boldsymbol{v}_j = \delta_{ij}$$

Hint: Use the fact that Σ_1, Σ_2 can be simultaneously diagonalized (Appendix B).

5.22 Show that in a multilayer perceptron with a linear output node, minimizing the squared error is equivalent to maximizing (5.51).
Hint: Assume the weights of the nonlinear nodes fixed and compute first the LS optimal weights driving the linear output nodes. Then substitute these values into the sum of error squares cost function.

5.23 Compute the minimal enclosure (hyper)sphere, that is, the radius as well as its center, of a set of points $\boldsymbol{x}_i$, $i = 1, 2, \ldots, N$.

MATLAB PROGRAMS AND EXERCISES

Computer Programs

5.1 *Scatter matrices.* Write a MATLAB function named *scatter_mat* that computes (a) the within-class (*Sw*), (b) the between-class (*Sb*) and the mixture (*Sm*) scatter matrices for a *c*-class classification problem, taking as inputs (a) an $l \times N$ dimensional matrix X, whose ith row is the ith data vector and (b) an N dimensional row vector y whose ith element contains the class label for the ith vector in X (the jth class is denoted by the integer $j, j = 1, \ldots, c$).

Solution

```
function [Sw,Sb,Sm]=scatter_mat(X,y)
  [l,N]=size(X);
```

```
c=max(y);
%Computation of class mean vectors, a priori prob. and
%Sw
m=[];
Sw=zeros(l);
for i=1:c
  y_temp=(y==i);
  X_temp=X(:,y_temp);
  P(i)=sum(y_temp)/N;
  m(:,i)=(mean(X_temp'))';
  Sw=Sw+P(i)*cov(X_temp');
end
%Computation of Sb
m0=(sum(((ones(l,1)*P).*m)'))';
Sb=zeros(l);
for i=1:c
  Sb=Sb+P(i)*((m(:,i)-m0)*(m(:,i)-m0)');
end
%Computation of Sm
Sm=Sw+Sb;
```

5.2 *$J3$ criterion.* Write a MATLAB function named *$J3\_comp$* that takes as inputs the within-class (*Sw*) and the mixture (*Sm*) scatter matrices and returns the value of the $J3$ criterion

Solution

```
function J3=J3_comp(Sw,Sm)
J3=trace(inv(Sw)*Sm);
```

5.3 *Best features combination.* Write a MATLAB function named *features_best_combin* that takes as inputs (a) an $l \times N$ dimensional matrix X, whose ith row is the ith data vector, (b) an N dimensional row vector y, whose ith element contains the class label for the ith vector in X (the jth class is denoted by the integer $j, j = 1, \ldots, c$), and (c) an integer q, the number of required features. It returns the best combination of q, out of the l, available features, according to the $J3$ criterion.

Solution

```
function id=features_best_combin(X,y,q)
  [l,N]=size(X);
  J3_max=0;
  id=[];
  combin=nchoosek(1:l,q);
  for j=1:size(combin,1)
```

```
    X1=X(combin(j,:),:);
    [Sw,Sb,Sm]=scatter_mat(X1,y);
    J3=J3_comp(Sw,Sm)
    if(J3>J3_max)
      J3_max=J3;
      id=combin(j,:);
    end
  end
```

5.4 *FDR criterion.* Write a MATLAB function named *FDR_comp* that returns the FDR index for a c class problem taking as inputs (a) an $l \times N$ dimensional matrix X, whose ith row is the ith data vector, (b) an N dimensional row vector y, whose ith element contains the class label for the ith vector in X (the jth class is denoted by the integer $j, j = 1, \ldots, c$), and (c) the index *ind* of the feature over which the FDR will be computed.

Solution

```
function FDR=FDR_comp(X,y,ind)
  [l,N]=size(X);
  c=max(y);
  for i=1:c
    y_temp=(y==i);
    X_temp=X(ind,y_temp);
    m(i)=mean(X_temp);
    vari(i)=var(X_temp);
  end
  a=nchoosek(1:c,2);
  q=(m(a(:,1))-m(a(:,2))).^ 2 ./ (vari(a(:,1))+vari(a(:,2)))';
  FDR=sum(q);
```

Computer Experiments

5.1 a. Generate $N1 = 100$ random numbers from the zero mean unit variance normal distribution and another $N2 = 100$ random numbers from the unit variance normal distribution with mean value equal to 2. Assume that these numbers correspond to the values a specific feature takes in the framework of a two-class problem. Use the t-test to check whether or not the hypothesis that the mean values for this feature, for the two classes, differ significantly, at a 5% significance level.

b. Repeat (a) when the mean value for the second distribution is 0.2.

c. Repeat (a) and (b) when $N1 = 150$ and $N2 = 200$.
Comment on the results.

Hint: Use the *normrnd* MATLAB function to generate the random numbers and the *ttest*2, to perform the *t*-test.

5.2 a. (i) Generate four sets, each one consisting of 100 two-dimensional vectors, from the normal distributions with mean values $[-10,\ -10]^T$, $[-10,\ 10]^T$, $[10,\ -10]^T$, $[10,\ 10]^T$ and covariance matrices equal to $0.2 * I$. These sets constitute the data set for a four-class two-dimensional classification problem (each set corresponds to a class).

a. (ii) Compute the *Sw*, *Sb*, and *Sm* scatter matrices.

a. (iii) Compute the value for the criterion $J3$.

b. Repeat (a) when the mean vectors of the normal distributions that generate the data are $[-1,\ -1]^T$, $[-1,\ 1]^T$, $[1,\ -1]^T$, $[1,\ 1]^T$.

c. Repeat (a) when the covariance matrices of the normal distributions that generate the data are equal to $3 * I$.

5.3 Generate two sets, each one consisting of 100 five-dimensional vectors, from the normal distributions with mean values $[0,\ 0,\ 0,\ 0,\ 0]^T$ and $[0,\ 2,\ 2,\ 3,\ 3]^T$ and covariance matrices equal to

$$\begin{bmatrix} 0.5 & 0 & 0 & 0 & 0 \\ 0 & 0.5 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1.5 \end{bmatrix}.$$

Their composition forms the data set for a two-class two-dimensional classification problem (each set corresponds to a class). Using the J_3 criterion find the best combination of features if:

a. they are considered individually.

b. they are considered in pairs.

c. they are considered in triples.

d. Justify the results.

5.4 a. (i) Generate two sets, each one consisting of 100 two-dimensional vectors, from the normal distributions with mean values $[2,\ 4]^T$ and $[2.5,\ 10]^T$ and covariance matrices equal to the 2×2 identity matrix I. Their composition forms the data set for a two class two dimensional classification problem (each set corresponds to a class).

a. (ii) Compute the value of the FDR index for both features.

b. Repeat (a) when the covariance matrices of the normal distributions that generate the data are both equal to $0.25 * I$.

c. Discuss the results.

REFERENCES

[Akai 74] Akaike H. "A new look at the statistical model identification," *IEEE Transactions on Automatic Control*, Vol. 19(6), pp. 716–723, 1974.

[Ambr 02] Ambroise C., McLachlan G.J. "Selection bias in gene extraction on the basis of microarray gene-expression data," *Proceedings of the National Academy of Sciences*, Vol. 99(10), pp. 6562–6566, 2002.

[Bart 99] Bartlett P., Shawe-Taylor J. "Generalization performance of support vector machines and other pattern classifiers," in *Advances in Kernel Methods: Support Vector Learning* (Schcölkopf S., Burges J.C., Smola A., eds.), MIT Press, 1999.

[Bati 94] Batiti R. "Using mutual information for selecting features in supervised neural network learning," *IEEE Transactions on Neural Networks*, Vol. 5(8), pp. 537–550, 1994.

[Baud 00] Baudat G., Anouar F. "Generalized discriminant analysis using a kernel approach," *Neural Computation*, Vol. 12(10), pp. 2385–2404, 2000.

[Baum 89] Baum E.B., Haussler D. "What size net gives valid generalization," *Neural Computation*, Vol. 1(1), pp. 151–160, 1989.

[Belh 97] Belhumeour P.N., Hespanha J.P., Kriegman D.J. "Eigenfaces vs Fisherfaces: Recognition using class specific linear projection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 19(7), pp. 711–720, 1997.

[Bish 95] Bishop C. *Neural Networks for Pattern Recognition*, Oxford University Press, 1995.

[Bour 88] Bourland H., Kamp Y. "Auto-association by multilayer perceptrons and singular value decomposition," *Biological Cybernetics*, Vol. 59, pp. 291–294, 1988.

[Brad 97] Bradley A. "The use of the area under the ROC curve in the evaluation of machine learning algorithms," *Pattern Recognition*, Vol. 30(7), pp. 1145–1159, 1997.

[Brun 00] Brunzell H., Erikcson J. "Feature reduction for classification of multidimensional data," *Pattern Recognition*, Vol. 33, pp. 1741–1748, 2000.

[Burg 98] Burges C.J.C. "A tutorial on support vector machines for pattern recognition," *Data Mining and Knowledge Discovery*, Vol. 2(2), pp. 1–47, 1998.

[Butz 05] Butz T., Thiran J.P. "From error probability to information theoretic (multi-modal) signal processing," *Signal Processing*), Vol. 85(5), pp. 875–902, 2005.

[Cevi 05] Cevikalp H., Neamtu M., Wilkes M., Barkana A. "Discriminative common vectors for face recognition," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 27(1), pp. 4–13, 2005.

[Chat 97] Chatterjee C., Roychowdhury V. "On self-organizing algorithms and networks for class-separability features," *IEEE Transactions on Neural Networks*, Vol. 8(3), pp. 663–678, 1997.

[Chen 00] Chen L.-F., Liao H.-Y.M., Ko M.-T., Lin J.-C., Yu G.-J. "A new LDA-based face recognition system which can solve the small sample size problem," *Pattern Recognition*, Vol. 33(10), pp. 1713–1726, 2000.

[Choi 03] Choi E., Lee C. "Feature extraction based on the Bhattacharyya distance," *Pattern Recognition Letters*, Vol. 36, pp. 1703–1709, 2003.

[Cris 00] Cristianini N., Shawe-Taylor J. *An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods*, Cambridge University Press, Cambridge, MA, 2000.

[Devi 82] Devijver P.A., Kittler J. *Pattern Recognition;A Statistical Approach*, Prentice Hall, 1982.

[Devr 96] Devroye L., Gyorfi L., Lugosi G. *A Probabilistic Theory of Pattern Recognition*, Springer-Verlag, 1996.

[Duin 00] Duin R.P.W. "Classifiers in almost empty spaces," *Proceedings of the 15th Int. Conference on Pattern Recognition (ICPR)*, vol. 2, Pattern Recognition and Neural Networks, IEEE Computer Society Press, 2000.

[Erdo 03] Erdogmus D., Principe J. "Lower and upper bounds for misclassification probability based on Renyi' s information," *Journal of VLSI Signal Processing*, 2003.

[Fine 83] Finette S., Bleier A., Swindel W. "Breast tissue classification using diagnostic ultrasound and pattern recognition techniques: I. Methods of pattern recognition," *Ultrasonic Imaging*, Vol. 5, pp. 55–70, 1983.

[Fish 36] Fisher R.A. "The use of multiple measurements in taxonomic problems," *Annals of Eugenics*, Vol. 7, pp. 179–188, 1936.

[Fras 58] Fraser D.A.S. *Statistics: An Introduction*, John Wiley & Sons, 1958.

[Frie 89] Friedman J.H. "Regularized discriminant analysis," *Journal of American Statistical Association*, Vol. 84, pp. 165–175, 1989.

[Fuku 90] Fukunaga K. *Introduction to Statistical Pattern Recognition*, 2nd ed., Academic Press, 1990.

[Gelm 95] Gelman A., Rubin D.B., Carlin J., Stern H. *Bayesian Data Analysis*, Chapman & Hall, London, 1995.

[Ghah 94] Ghaharamani Z., Jordan M.I. "Supervised learning from incomplete data via the EM approach," in *Advances in Neural Information Processing Systems* (Cowan J.D., Tesauro G.T., Alspector J., eds), Vol. 6, pp. 120–127, Morgan Kaufmann, San Mateo, CA, 1994.

[Guyo 03] Guyon I, Elisseeff A. "An introduction to variable and feature selection," *Journal of Machine Learning Research*, Vol. 3, pp. 1157–1182, 2003.

[Hams 08] Hamsici O. C., Martinez A. M. " Bayes optimality in LDA," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 30(4), pp. 647–657, 2008.

[Hast 95] Hastie T., Tibshirani R. "Penalized discriminant analysis," *Annals of Statistics*, Vol. 23, pp. 73–102, 1995.

[Hast 01] Hastie T., Tibshirani R., Friedman J. *The Elements of Statistical Learning*, Springer, 2001.

[Hube 81] Huber P.J. *Robust Statistics*, John Wiley & Sons, 1981.

[Hush 93] Hush D.R., Horne B.G. "Progress in supervised neural networks," *Signal Processing Magazine*, Vol. 10(1), pp. 8–39, 1993.

[Jain 97] Jain A., Zongker D. "Feature selection: Evaluation, application, and small sample performance," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 19(2), pp. 153–158, 1997.

[Kim 07] Kim C., Choi C.-H. " A discriminant analysis using composite features for classification problems," *Pattern Recognition*, Vol. 40(11), pp. 2958–2967, 2007.

[Kitt 78] Kittler J. "Feature set search algorithms," in *Pattern Recognition and Signal Processing* (Chen C.H., ed.), pp. 41–60, Sijthoff and Noordhoff, Alphen aan den Rijn, The Netherlands, 1978.

[Kram 91] Kramer M.A. "Nonlinear principal component analysis using auto-associative neural networks," *AIC Journal*, Vol. 37(2), pp. 233–243, 1991.

[Kulb 51] Kullback S., Liebler R.A. "On information and sufficiency," *Annals of Mathematical Statistics*, Vol. 22, pp. 79–86, 1951.

[Kwak 02] Kwak N., Choi C.-H. "Input feature selection for classification problems," *IEEE Transactions on Neural Networks*, Vol. 13(1), pp. 143–159, 2002.

[Land 08] Landgrebe T. C. W., Duiy R. P. W. "Efficient multiclasss ROC approximation by decomposing via confusion matrix pertubation analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 30(5), pp. 810–822, 2008.

[Lawe 66] Lawer E.L., Wood D.E. "Branch and bound methods: A survey," *Operational Research*, Vol. 149(4), 1966.

[Lee 00] Lee C., Choi E. "Bayes error evaluation of the Gaussian ML classifier," *IEEE Transactions on Geoscience Remote Sensing*, Vol. 38(3), pp. 1471–1475, 2000.

[Lee 93] Lee C., Landgrebe D.A. "Decision boundary feature extraction for nonparametric classifiers," *IEEE Transactions on Systems Man and Cybernetics*, Vol. 23, pp. 433–444, 1993.

[Lee 97] Lee C., Landgrebe D. "Decision boundary feature extraction for neural networks," *IEEE Transactions on Neural Networks*, Vol. 8(1), pp. 75–83, 1997.

[Leiv 07] Leiva-Murillo J.M., Artes-Rodriguez A. "Maximization of mutual information for supervised linear feature extraction," *IEEE Transactions on Neural Networks*, Vol. 18(5), pp. 1433–1442, 2007.

[Leth 96] Lethtokanga S.M., Saarinen J., Huuhtanen P., Kaski K. "Predictive minimum description length criterion for time series modeling with neural networks," *Neural Computation*, Vol. 8, pp. 583–593, 1996.

[Li 06] Li H., Jiang T., Zhang K. "Efficient and robust extraction by maximum margin criterion," *IEEE Transactions on Neural Networks*, Vol. 17(1), pp. 157–165, 2006.

[Lin 02] Lin Y., Wahba G., Zhang H., Lee Y. "Statistical properties and adaptive tuning of support vector machines," *Machine Learning*, Vol. 48, pp. 115–136, 2002.

[Loog 04] Loog M., Duin P.W. "Linear Dimensionality reduction via a heteroscedastic extension of LDA: The Chernoff criterion," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 26(6), pp. 732–739, 2004.

[Lowe 90] Lowe D., Webb A.R. "Exploiting prior knowledge in network optimization: An illustration from medical prognosis," *Network: Computation in Neural Systems*, Vol. 1(3), pp. 299–323, 1990.

[Lowe 91] Lowe D., Webb A.R. "Optimized feature extraction and the Bayes decision in feedforward classifier networks," *IEEE Transactions in Pattern Analysis and Machine Intelligence*, Vol. 13(4), pp. 355–364, 1991.

[Ma 03] Ma J., Jose L. S., Ahalt S. "Nonlinear multiclass discriminant analysis," *IEEE Signal Processing Letters*, Vol. 10(33), pp. 196–199, 2003.

[Mama 96] Mamamoto Y., Uchimura S., Tomita S. "On the behaviour of artificial neural network classifiers in high dimensional spaces," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 18(5), pp. 571–574, 1996.

[Mao 95] Mao J., Jain A.K. "Artificial neural networks for feature extraction and multivariate data projection," *IEEE Transactions on Neural Networks*, Vol. 6(2), pp. 296–317, 1997.

[Marr 08] Marroco C., Duin R. P. W., Tortorella F. "Maximizing the area under the ROC curve by pairwise feature combination," *Pattern Recognition*, Vol. 41, pp. 1961–1974, 2008.

[Maus 90] Mausel P.W., Kramber W.J., Lee J.K. "Optimum band selection for supervised classification of multispectra data," *Photogrammetric Engineering and Remote Sensing* Vol. 56, pp. 55–60, 1990.

[Mood 92] Moody J.E. "The effective number of parameters: An analysis of generalization and regularization in nonlinear learning systems" in *Advances in Neural Computation* (Moody J.E., Hanson S.J., Lippman R.R., eds.), pp. 847–854, Morgan Kaufman, San Mateo, CA, 1992.

[Nena 07] Nenadic Z. "Information discriminant analysis: feature extraction with an information-theoretic objective," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 29(8), pp. 1394–1408, 2007.

[Papo 91] Papoulis A. *Probability Random Variables and Stochastic Processes*, 3rd ed., McGraw-Hill, 1991.

[Pudi 94] Pudil P., Novovicova J., Kittler J. "Floating search methods in feature selection," *Pattern Recognition Letters*, Vol. 15, pp. 1119–1125, 1994.

[Raud 91] Raudys S.J., Jain A.K. "Small size effects in statistical pattern recognition: Recommendations for practitioners," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 13(3), pp. 252–264, 1991.

[Raud 80] Raudys S.J., Pikelis V. "On dimensionality, sample size, classification error, and complexity of classification algorithms in pattern recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 2(3), pp. 243–251, 1980.

[Rich 95] Richards J. *Remote Sensing Digital Image Analysis*, 2nd ed., Springer-Verlag, 1995.

[Ripl 96] Ripley B.D. *Pattern Recognition And Neural Networks*, Cambridge University Press, Cambridge, MA, 1996.

[Riss 83] Rissanen J. "A universal prior for integers and estimation by minimum description length," *The Annals of Statistics*, Vol. 11(2), pp. 416–431, 1983.

[Rubi 76] Rubin D.B. "Inference and missing data," *Biometrika*, Vol. 63, pp. 581–592, 1976.

[Rubi 87] Rubin D.B. *Multiple Imputation for Nonresponse in Surveys*, John Wiley & Sons, 1987.

[Samm 69] Sammon J.W. "A nonlinear mapping for data structure analysis," *IEEE Transactions on Computers*, Vol. 18, pp. 401–409, 1969.

[Scha 02] Schafer J., Graham J. "Missing data: Our view of the state of the art," *Psychological Methods*, vol. 7(2), pp. 67–81, 2002.

[Schw 79] Schwartz G. "Estimating the dimension of the model," *Annals of Statistics*, Vol. 6, pp. 461–464, 1978.

[Seti 97] Setiono R., Liu H. "Neural network feature selector," *IEEE Transactions on Neural Networks*, Vol. 8(3), pp. 654–662, 1997.

[Sind 04] Sindhwami V., Rakshit S., Deodhare D., Erdogmus D., Principe J.C., Niyogi P. "Feature selection in MLPs and SVMs based on maximum output information," *IEEE Transactions on Neural Networks*, Vol. 15(4), pp. 937–948, 2004.

[Stoi 04b] Stoica P., Moses R. *Spectral Analysis of Signals*, Prentice Hall, 2004.

[Stoi 04a] Stoica P., Selén Y. "A review of information criterion rules," *Signal Processing Magazine*, Vol. 21(4), pp. 36–47, 2004.

[Su 94] Su K.Y, Lee C.H. "Speech recognition using weighted HMM and subspace projection approaches," *IEEE Transactions on Speech and Audio Processing*, Vol. 2(1), pp. 69–79, 1994.

[Swai 73] Swain P.H., King R.C. "Two effective feature selection criteria for multispectral remote sensing," *Proceedings of the 1st International Conference on Pattern Recognition*, pp. 536–540, 1973.

[Tian 86] Tian Q., Marbero M., Gu Z.H., Lee S.H. "Image classification by the Folley-Sammon transform," *Optical Engineering*, Vol. 25(7), pp. 834–840, 1986.

[Tou 74] Tou J., Gonzalez R.C. *Pattern Recognition Principles*, Addison-Wesley, 1974.

[Trun 79] Trunk G.V. "A problem of dimensionality: A simple example," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 1(3), pp. 306–307, 1979.

[Tsud 03] Tsuda K., Akaho S., Asai K. "The EM algorithm for kernel matrix completion with auxiliary data," *Journal of Machine Learning Research*, Vol. 4, pp. 67–81, 2003.

[Vali 84] Valiant L. "A theory of the learnable," *Communications of the ACM*, Vol. 27(11), pp. 1134–1142, 1984.

[Vapn 82] Vapnik V.N. *Estimation of Dependencies Based on Empirical Data*, Springer-Verlag, 1982.

[Vapn 95] Vapnik V.N. *The Nature of Statistical Learning Theory*, Springer-Verlag, 1995.

[Vapn 98] Vapnik, V.N. *Statistical Learning Theory*, John Wiley & Sons, 1998.

[Walp 78] Walpole R.E., Myers R.H. *Probability and Statistics for Engineers and Scientists*, Macmillan, 1978.

[Wang 00] Wang W., Jones P., Partridge D. "A comparative study of feature salience ranking techniques," *Neural Computation*, Vol. 13(7), pp. 1603–1623, 2000.

[Wang 98] Wang Y., Adali T., Kung S.Y., Szabo Z. "Quantization and segmentation of brain tissues from MR images: A probabilistic neural network approach," *IEEE Transactions on Image Processing*, Vol. 7(8), 1998.

[Wata 97] Watanabe H., Yamaguchi T., Katagiri S. "Discriminative metric for robust pattern recognition," *IEEE Transactions on Signal Processing*, Vol. 45(11), pp. 2655–2663, 1997.

[Will 07] Williams D., Liao X., Xue Y., Carin L., Krishnapuram B. "On classification with incomplete data," *IEEE Transactions on Pattern Analysis and Machine Intelligence* Vol. 29(3), pp. 427–436, 2007.

[Yang 02] Yang J., Yang J.-Y. "Why can LDA be performed in PCA transformed space?" *Pattern Recognition*, Vol. 36, pp. 563–566, 2002.

[Ye 05] Ye J., Li Q. "A two stage linear discriminant analysis via QR decomposition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 27(6), pp. 929–941, 2005.

[Yu 93] Yu B., Yuan B. "A more efficient branch and bound algorithm for feature selection," *Pattern Recognition*, Vol. 26(6), pp. 883–889, 1993.

[Zhan 02] Zhang H., Sun G. "Feature selection using tabu search method," *Pattern Recognition*, Vol. 35, pp. 701–711, 2002.

[Zhan 07] Zhang S., Sim T. "Discriminant subspace analysis: A Fukunaga-Koontz approach," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 29(10), pp. 1732–1745, 2007.