3. Performance and the Generalisation Error

Generalisation (test set) and empirical (training-set) classification errors are meaningful characteristics of any pattern classification system. Generally, one needs to know both these error rates and their relationship with the training-set sizes, the number of features, and the type of the classification rule. This knowledge can help one to choose a classifier of the proper complexity, with an optimal number of features, and to determine a sufficient number of training vectors. While training the non-linear SLP, one initially begins with the Euclidean distance classifier and then moves dynamically towards six increasingly complex statistical classifiers. Therefore, utilisation of theoretical generalisation error results obtained for these seven statistical classifiers becomes a guide for analysing the small sample properties of neural net generated classification algorithms.

The generalisation error depends on the (unknown) characteristics of the data. To design the classifier we make also certain assumptions about the data. In applications, this hypothetical model can be violated. Thus, in order to derive formulae for the generalisation and asymptotic errors, we need to differentiate between

- assumptions about the distribution densities used to *design* the classifier; and
- assumptions about the data model for which this particular classifier is *applied*.

In this chapter, after a more precise presentation of the main types of classification errors and a short overview of competing methods for analysing the small sample behaviour, we consider a number of parametric linear and non-linear classification rules. An exceptional peculiarity of our consideration is the asymptotic analysis where both the number of inputs n and the training-set size N are increasing at a constant rate. For the simplest classifiers, we derive expressions for the generalisation error and compare the approximate generalisation error results with exact ones obtained from complex formulae. We also review asymptotic investigations for other more sophisticated classification rules. We then show the deterioration of the plug-in based classifiers when the number of training vectors in each category is very diverse and demonstrate that parameters of distribution densities that are common for the two pattern classes have a small influence on the generalisation error. Finally, we show that it is usually the intrinsic dimensionality of the data, and not the number of features, that most often determines the sensitivity of a pattern recognition classifier to the training-set size.

3.1 Bayes, Conditional, Expected and Asymptotic Probabilities of Misclassification

There are several types of probabilities of misclassification (PMC) that are helpful in characterising the quality of a pattern recognition system (an algorithm). Among these are probabilities ε_{ij} of incorrect classification of an object from class ω_i as ω_j, a general sum of probabilities of misclassification, the loss, a rejection rate (refuse from decision), etc. In many applications, however, the general sum probability of correct (or incorrect) decisions is the most important. In Chapter 1, we already introduced the *Bayes*, *conditional*, *expected* and *asymptotic* probabilities of misclassification. We now define these probabilities more precisely.

3.1.1 The Bayes Probability of Misclassification

Suppose we know the conditional probability density functions (PDF) $p_1(\boldsymbol{X})$, $p_2(\boldsymbol{X})$ and the prior probabilities P_1, P_2 = 1-P_1 of the pattern classes. Then, we can design the optimal Bayes classifier B which, in classifying all possible vectors from classes ω_1 and ω_2, results in the minimal probability of misclassification. This PMC is called the *Bayes error* and is denoted by ε_B.

The Bayes PMC is a function of only the true probability density functions of the classes $p_1(\boldsymbol{X})$, $p_2(\boldsymbol{X})$ and the prior probabilities P_1, P_2. One can represent the Bayes PMC as

$$\varepsilon_B = \int_{\Omega} \min \{P_1\, p_1(\boldsymbol{X}),\, P_2\, p_2(\boldsymbol{X}))\}\, d\, \boldsymbol{X}. \tag{3.1}$$

In Figure 1.6, the Bayes error corresponds to two dashed areas.

3.1.2 The Conditional Probability of Misclassification

In each real classification problem, we use the training data to design an appropriate classifier. Usually, we need to choose one particular classifier design algorithm. In Chapter 2 we presented several of the algorithms: the EDC, the Fisher, the minimum empirical error classifiers, etc. The probability of misclassification of a classifier designed from one particular training-set using the algorithm A is conditioned on the particular classifier method and the particular training-set. The error rate for classifying the pattern vectors from the general population is called the *conditional probability of misclassification* and is denoted by $\varepsilon_N^{\mathsf{A}}$. The index A indicates that the classifier design method A was used, and the index N symbolizes that the training-set is fixed, composed of $N = N_1 + N_2$ observation vectors. The word *conditional* indicates that the classifier, and its probability of misclassification, are conditioned on one particular training-set. Thus, for $\mathsf{A} \equiv$ EDC we will obtain one value for the conditional PMC ($\varepsilon_N^{\mathrm{E}}$) and for $\mathsf{A} \equiv$ F (the Fisher classifier) we, with rare exceptions, obtain another value ($\varepsilon_N^{\mathrm{F}}$). Use of a new training-set composed from N_1, N_2 new observations will result in a new pair of conditional probabilities ($\varepsilon_N^{\mathrm{E*}}$ and $\varepsilon_N^{\mathrm{F*}}$). In order to simplify analytical formulae in this chapter, we will often assume N_2=N_1= $\overline{N}$ = N/2. In the artificial neural networks literature, this error rate is sometimes referred to as the *conditional generalisation error*.

3.1.3 The Expected Probability of Misclassification

In statistical theory, one analyses sets of randomly sampled observations. Therefore, classifier's parameters and the classifier's probability of misclassification can be analysed as *random* variables. In probability theory, a random variable is characterised by its probability density function. Let $f(\varepsilon_N^{\mathsf{A}})$ be the probability density function of the random variable $\varepsilon_N^{\mathsf{A}}$ and let $\overline{\varepsilon}_N^{\mathsf{A}}$ be its expectation over all possible random training sets of size N_1 and N_2. This expectation is called the *expected probability of misclassification.*

In the artificial neural net literature, both the conditional and expected errors are often called the *generalisation error.* Often little attention is given to the differences between them. In this book, we make a distinction between these two types of error rate: the conditional PMC is referred to as the *conditional* generalisation error and the expected PMC $\overline{\varepsilon}_N^{\mathsf{A}}$ is called the *expected (mean)* generalisation error.

3.1.4 The Asymptotic Probability of Misclassification

The parameters of a classification rule are determined more and more precisely as the training sample sizes, N_1 and N_2, increase. In particular, the mean value of $\varepsilon_N^{\mathsf{A}}$, namely $\overline{\varepsilon}_N^{\mathsf{A}}$, tends to some fixed value, and the variance $V\varepsilon_N^{\mathsf{A}}$ tends to zero. At the limit, when N_1, $N_2 \to \infty$, the probability density function $f(\varepsilon_N^{\mathsf{A}})$ becomes similar to a delta function (a constant value $\varepsilon_\infty^{\mathsf{A}}$). This limit is called the *asymptotic probability of misclassification*:

$$\varepsilon_\infty^{\mathsf{A}} = \lim_{N_1\to\infty, N_2\to\infty} \overline{\varepsilon}_N^{\mathsf{A}}.$$

The asymptotic PMC $\varepsilon_{\infty}^{\mathsf{A}}$ depends on A, the method used to design the classifier, the prior probabilities P_1, P_2, and on the true pattern-class densities $p_1(\boldsymbol{X})$, $p_2(\boldsymbol{X})$. We note that the asymptotic PMC $\varepsilon_{\infty}^{\mathsf{A}}$ does not depend on the training-set or its size.

Example 1. To demonstrate the differences between the four types of classification errors we depict the probability density function of the random variable $\varepsilon_{N}^{\mathsf{A}}$ for three values of the training set size $\bar{N} = N/2= N_1=N_2$ in Figure 3.1.

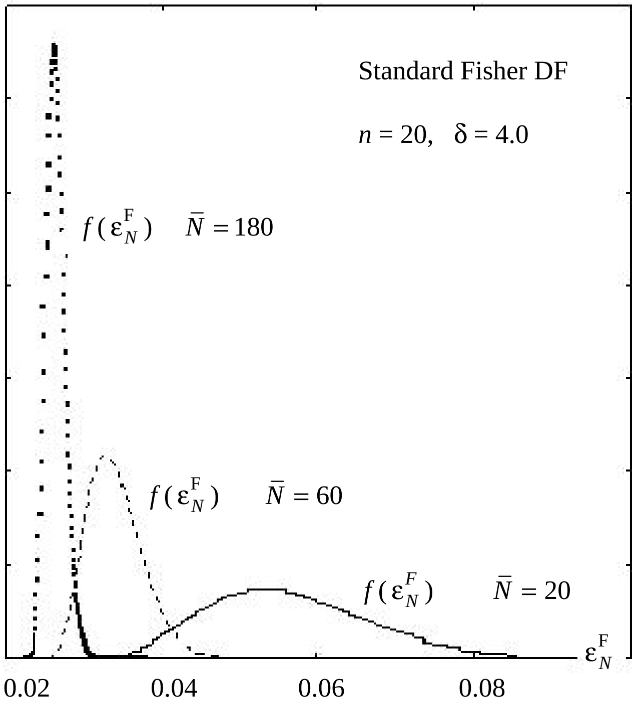

Fig. 3.1. The histograms of the conditional probabilities of misclassification of the standard Fisher linear DF for training-set sizes $\bar{N}$ = 20, 60 and 180.

The probability density function $f(\varepsilon_{N}^{\mathsf{A}})$ of the random variable $\varepsilon_{N}^{\mathsf{A}}$ is always to right of the Bayes PMC ε_B. The non optimality of the classifier design procedure causes $\varepsilon_{N}^{\mathsf{A}} < \varepsilon_{\infty}^{\mathsf{A}}$ for certain training sets. However, we always have that $\varepsilon_{N}^{\mathsf{A}} \geq \varepsilon_B$ and $\bar{\varepsilon}_{N}^{\mathsf{A}} \geq \varepsilon_{\infty}^{\mathsf{A}}$.

3.1.5. Learning Curves: An Overview of Different Analysis Methods

In this chapter, we analyse the learning curves that decrease in the generalisation error $\bar{\varepsilon}_N^{A}$ as N increases. A number of approaches to studying the generalisation error in the finite training-set size case have been proposed. In our analysis, we compare the multivariate statistics approach with other approaches. For this comparison, we need certain definitions that are utilised in these other approaches. Therefore, we will present a very brief overview of these competing approaches. We give more details about these approaches in the last section of this chapter.

Above we have already stressed the importance of distinguishing between the assumptions about the distribution densities used to *design* the classifier and the assumptions about the data model for which a particular classifier *is applied*. To denote this difference, some researchers have stated that when the distribution density assumptions utilised to design the classifier coincide with the data model for which this particular classifier is applied, we have a *faithful* case. When the designer's assumptions differ from reality, we have an *unfaithful* case.

Vapnik and Chervonenkis (1968, 1974) and Vapnik (1982) developed the Cover (1965) capacity concept and obtained a number of upper bounds for the *conditional* generalisation error ε_N^{A} for classifiers that minimise the empirical classification error. In Sections 3.6.2 and 3.6.6, while discussing the minimum empirical error and maximal margin classifiers, we will present several of their error bounds. Amari, Fujita and Shinomoto (1992) did not specify a particular type of distribution densities. They have shown that the *expected* generalisation error $\bar{\varepsilon}_N^{A}$ behaves asymptotically (when $N \rightarrow \infty$) as

$$\bar{\varepsilon}_N^{A} \sim h/N,$$

when the network is deterministic, the teacher signal is noiseless, and the network giving correct classification is uniquely specified by the h-dimensional weight vector $\boldsymbol{V}$. In the case with an empty zone between the pattern classes ($\varepsilon_\infty^{A} = 0$), we have much better small sample behaviour, namely:

$$\bar{\varepsilon}_N^{A} \sim c/N^2,$$

for a unique deterministic network trained by the noisy teacher

$$\bar{\varepsilon}_N^{A} \sim c/N^{1/2},$$

and for the stochastic network

$$\bar{\varepsilon}_N^{A} \sim \varepsilon_\infty^{A} + c_1/N,$$

where c and c_1 are unknown constants.

Amari and Murata (1993) have proved fundamental universal convergence theorems for the average generalisation and training errors measured as the predictive entropic loss $\overline{H}_N^{\mathrm{A}}$ (stochastic complexity) evaluated by the expectation of $-\log p(o| \boldsymbol{V}, \boldsymbol{X})$ for an input–output pair ($\boldsymbol{X}$, o). For the weights estimated by the maximum likelihood method or by the Bayes posterior distribution approach, they proved that an average generalisation entropic loss of the stochastic network is

$$\overline{H}_N^{\mathrm{A}} = H_\infty^{\mathrm{A}} + h^* /(2N),$$

where h^* shows a complexity of the network.

For the faithful (realisable) network $h^*= h$ and for the unfaithful (unrealisable) network:

$$h^*= \mathrm{tr}\mathbf{K}^{-1}\mathbf{G},$$

where **K** is the Hessian matrix, and **G** is the Fisher information matrix.

For a deterministic dichotomy network, Amari (1993) showed that

$$\overline{H}_N^{\mathrm{A}} = h/ N.$$

We will demonstrate that these general asymptotic ($N \rightarrow \infty$) results agree with the results obtained with the classical statistical framework considered in this book when the training-set size N is very large, i.e. when the expected errors are close to asymptotic ones. In finite (small) training-set cases, however, an exact statistical analysis yields a very different result.

A characteristic property of the *statistical−mechanic approach* is the so-called *thermodynamic limit*, where one examines the generalisation error as $N \rightarrow \infty$ and as the complexity $h \rightarrow \infty$ at some fixed rate. This allows us to meaningfully investigate an asymptotic generalisation error when the number of examples is half the number of parameters, twice the number of parameters, 10 times the number of parameters, and so on (Haussler *et al.,* 1994). This approach uses other mathematical methods from statistical mechanics such as the replica symmetry technique and the annealed approximation. There, the mean value of the ratio of two random variables is approximated by the ratio of the mean values of these two random variables. The validity of this approximation is still questionable. For some models the statistical–mechanic approach succeeds in obtaining the expected generalisation errors and its “phase transitions” (sudden drops in the generalisation error). For the deterministic dichotomy network, for example, a strong rigorous result was proved (Gyorgyi and Tishby, 1990; Opper and Haussler, 1991):

$$\overline{H}_N^{\mathrm{A}} = 0.62\, h/N.$$

In certain cases, a different power law other than $1/N$ or $1/N^{1/2}$ was obtained (Haussler *et al,* 1994; Seung *et al,* 1992).

The statistical approach adopted in this book has the following positive characteristic. Provided the theoretical analysis is performed correctly, we can obtain exact expressions that are valid for a given theoretical data model. Here we do not have unknown constants or unrealistic error bounds. A negative characteristic is that we have to know the type of the probability density functions of the data. Thus, practical utilisation of the theoretical results becomes problematic. Another difficulty consists in the complexity of the exact expressions for the generalisation error and the necessity to use asymptotic expansions if the exact formulae become too complicated. Fortunately, the double asymptotics approach (when both N and h are increasing) works sufficiently well. Exact learning curves, $\overline{\varepsilon}_N^{\mathrm{A}} = f(N)$, obtained for some theoretical models allow one to better understand theoretical questions about the small sample behaviour of distinct statistical classifiers and to find conditions such that statistical methods can be utilised successfully in small training-set situations.

3.1.6 Error Estimation

The Bayes, asymptotic, conditional and expected probabilities of misclassification are only *abstractions* which depend on unknown distributions $f_1(\boldsymbol{X})$ and $f_2(\boldsymbol{X})$. These probabilities of misclassification can be evaluated by means of special experiments. The conditional PMC of a newly-designed classifier can be estimated using an additional test data set (or the validation set) composed of pattern vectors that did not participate in the training process. We call this estimate a *test set estimate of the classification error*. To estimate the expected PMC, one needs to have a number of independent random training sets of the same size. The experimental estimation of the asymptotic and Bayes PMC is difficult.

To test a classifier we sometimes use the same pattern vectors which were used for training the classifier. Utilisation of the training-set to estimate PMC results in a *training-set error* or the resubstitution estimate. This error is often called an *empirical error* and sometimes it is called the *apparent error*. The term *apparent error* arose from the fact that in the small training-set size case, this error estimation method results in overly optimistic error estimates. That is, the classifier seems to perform better than it does in reality. We shall discuss classification error estimation methods further in Chapter 6.

3.2 The Generalisation Error of the Euclidean Distance Classifier

3.2.1 The Classification Algorithm

The discriminant function of the Euclidean distance classifier

$$h^{\mathrm{E}}(\boldsymbol{X}) = \boldsymbol{X}^T\boldsymbol{V}^{\mathrm{E}} + v_0 = \boldsymbol{X}^T(\hat{\boldsymbol{M}}_1 - \hat{\boldsymbol{M}}_2) - ½\ (\hat{\boldsymbol{M}}_1 + \hat{\boldsymbol{M}}_2)^T(\hat{\boldsymbol{M}}_1 - \hat{\boldsymbol{M}}_2) \qquad (3.2)$$

is a function both of the random vector to be classified, $\boldsymbol{X}$, and the sample means, $\hat{\boldsymbol{M}}_1$ and $\hat{\boldsymbol{M}}_2$. If $\bar{N} \rightarrow \infty$, EDC approaches the Bayes classifier for the model of two spherically Gaussian populations sharing the common covariance matrix. In Section 2.4.2 it was demonstrated that, from the point of view of the Bayesian predictive approach, for the Gaussian priors of the vector $\Delta\boldsymbol{M} = \boldsymbol{M}_1 - \boldsymbol{M}_2$, the sample-based EDC is an ideal discriminant function to classify two spherical Gaussian populations. In the SG case, the *ideal* EDC discriminant function has a Gaussian distribution with mean

$$E[h\,(\boldsymbol{X}) \mid \omega_i] = [(\boldsymbol{M}_i - ½\,(\boldsymbol{M}_1 + \boldsymbol{M}_2)]^T(\boldsymbol{M}_1 - \boldsymbol{M}_2\,),$$

and variance

$$V[h\,(\boldsymbol{X}) \mid \omega_i] = \sigma^2(\boldsymbol{M}_1 - \boldsymbol{M}_2)^T\,(\boldsymbol{M}_1 - \boldsymbol{M}_2).$$

Thus the asymptotic and Bayes probabilities of misclassification are given by

$$\varepsilon_\infty^{\mathrm{F}} = P_1 Prob\,\{\,h(\boldsymbol{X}) < 0 \mid \omega_1\} + P_2 Prob\,\{h(\boldsymbol{X}) \geq 0 \mid \omega_2\} = \Phi\{-\delta\,/2\} = \varepsilon_B,$$

where $\delta^2 = \sigma^{-2}(\boldsymbol{M}_1 - \boldsymbol{M}_2\,)^T\,(\boldsymbol{M}_1 - \boldsymbol{M}_2\,)$ is squared Euclidean distance scaled by σ^{-2}.

3.2.2 Double Asymptotics in the Error Analysis

In order to derive analytical expressions for the generalisation error we must find the probability that a value of the *sample discriminant function* is negative (when $\boldsymbol{X} \in \omega_1$) or positive (when $\boldsymbol{X} \in \omega_2$). The exact analytical expression for the generalisation error for the case when the true data distributions are Gaussian with a common CM (GCCM data model) requires integration of a complex analytical expression (Raudys, 1967, Raudys and Pikelis, 1980). In order to obtain a simple and easily interpreted expression, we notice that the sample-based discriminant function is the sum of n random components. Thus, one can approximate the unknown distribution of the discriminant function $h(\boldsymbol{X})$ by a Gaussian distribution. The key tool for providing this approximate distribution of $h(\boldsymbol{X})$ is the application of a central limit theorem. We assume that the number of features, n, is very large. In order to have a realistic situation we also assume that the training-set size N is large.

Thus, a simple asymptotic method for analysing distributions of complex multivariate statistics is as follows: we analyse the case when N, the training-set size, and n, the number of dimensions, are increasing simultaneously at a constant rate. This technique was first utilised by Raudys (1967), and is now widely applied in investigating the generalisation error of the statistical classification algorithms and artificial neural networks. In theoretical physics, this asymptotic approximation method is called the “thermodynamic limit” approach. In comparison with conventional asymptotic analysis, where only the sample size N

approaches infinity (see e.g. Chapter 5 in Fukunaga, 1990), this approach allows one to obtain very accurate estimates when the sample size N is comparable with the number of features n, even when N is relatively small. The *double asymptotics method* is the main technique utilised to obtain the generalisation error formulae in this chapter.

In Section 1.4, we have presented the asymptotic expression for the expected probability of misclassification for the theoretical model when the true classes covariance matrix $\overline{\Sigma}$ is proportional to the identity matrix, i.e. $\overline{\Sigma} = \sigma^2\mathbf{I}$, and $N_2=N_1$. In practice, EDC is used to classify non-spherical and even non-Gaussian pattern classes. Therefore, in this section, we do not assume that the common covariance matrix is $\overline{\Sigma} = \sigma^2\mathbf{I}$. In order to present a simple and easily understandable derivation of the generalisation error formula, we initially assume $N_2=N_1=\overline{N}$ and $P_2=P_1=½$.

In order to stress that the training vectors are considered as random vectors, one commonly denotes them by capital letters, i.e. $\boldsymbol{X}_1^{(1)}, \boldsymbol{X}_2^{(1)},\ldots, \boldsymbol{X}_N^{(2)}$. For the two Gaussian pattern classes model, $N_X(\boldsymbol{M}_1, \overline{\Sigma})$, $N_X(\boldsymbol{M}_2, \overline{\Sigma})$, the sample mean vectors are Gaussian: $\hat{\boldsymbol{M}}_1 \sim N_M(\boldsymbol{M}_1, 1/\overline{N}\,\overline{\Sigma})$, $\hat{\boldsymbol{M}}_2 \sim N_M(\boldsymbol{M}_2, 1/\overline{N}\,\overline{\Sigma})$. Consequently, we need to consider the DF $h^{\mathrm{E}}(\boldsymbol{X}, \hat{\boldsymbol{M}}_1, \hat{\boldsymbol{M}}_2)$ as a random variable which depends on three independent n-dimensional random vectors $\boldsymbol{X}$, $\hat{\boldsymbol{M}}_1$ and $\hat{\boldsymbol{M}}_2$. The expected PMC can be written as the sum of two conditional probabilities:

$$\overline{\varepsilon}_N^{\mathrm{E}} = P_1\, Prob\{h^{\mathrm{E}}(\boldsymbol{X}, \hat{\boldsymbol{M}}_1, \hat{\boldsymbol{M}}_2) < 0 \mid \boldsymbol{X} \in \omega_1\} + P_2\, Prob\{h^{\mathrm{E}}(\boldsymbol{X}, \hat{\boldsymbol{M}}_1, \hat{\boldsymbol{M}}_2) \geq 0 \mid \boldsymbol{X} \in \omega_2\}.$$

Suppose both the dimensionality, n, and training-set size, N, are large. Then, according to the central limit theorem, the distribution of

$$h(\boldsymbol{X}, \hat{\boldsymbol{M}}_1, \hat{\boldsymbol{M}}_2) = \sum_{s=1}^{n} \left(x_s\,(\hat{m}_{1s} - \hat{m}_{2s}) - ½\,(\hat{m}_{1s} + \hat{m}_{2s})(\hat{m}_{1s} - \hat{m}_{2s})\right)$$

is approximately Gaussian. In the above equation, subscript s indicates the components of random vectors $\boldsymbol{X}$, $\hat{\boldsymbol{M}}_1$, and $\hat{\boldsymbol{M}}_2$. Thus, asymptotically, when n and N are increasing, the expected probability of misclassification (the expected generalisation error) is

$$\overline{\varepsilon}_N^{\mathrm{E}} = P_1\,\Phi\{-\frac{E\left[h(\boldsymbol{X}, \hat{\boldsymbol{M}}_1, \hat{\boldsymbol{M}}_2) \middle| \boldsymbol{X} \in \omega_1\right]}{\sqrt{V\left[h(\boldsymbol{X}, \hat{\boldsymbol{M}}_1, \hat{\boldsymbol{M}}_2) \middle| \boldsymbol{X} \in \omega_1\right]}}\} + P_2\,\Phi\{\frac{E\left[h(\boldsymbol{X}, \hat{\boldsymbol{M}}_1, \hat{\boldsymbol{M}}_2) \middle| \boldsymbol{X} \in \omega_2\right]}{\sqrt{V\left[h(\boldsymbol{X}, \hat{\boldsymbol{M}}_1, \hat{\boldsymbol{M}}_2) \middle| \boldsymbol{X} \in \omega_2\right]}}\}, \tag{3.4}$$

where E and V denote the expectation and variance with respect to three independent random vectors $\boldsymbol{X}$, $\hat{\boldsymbol{M}}_1$, and $\hat{\boldsymbol{M}}_2$.

Without loss of generality, we may assume $\boldsymbol{M}_1 + \boldsymbol{M}_2 = \mathbf{0}$. For $N_2 = N_1 = \overline{N}$ the sum $\boldsymbol{Z}_1 = \hat{\boldsymbol{M}}_1 + \hat{\boldsymbol{M}}_2$ and the difference $\boldsymbol{Z}_2 = \hat{\boldsymbol{M}}_1 - \hat{\boldsymbol{M}}_2$ are independently distributed as $\boldsymbol{Z}_1 \sim N_{Z_1}(0,\ 4/N\ \overline{\Sigma})$, and $\boldsymbol{Z}_2 \sim N_Y(\boldsymbol{M},\ 4/N\,\overline{\Sigma})$, where $\boldsymbol{M}_1 - \boldsymbol{M}_2 = \boldsymbol{M}$. Consequently, for $\boldsymbol{X} \in \omega_i$, we have that $\boldsymbol{Z} = \boldsymbol{X} - ½\,(\hat{\boldsymbol{M}}_1 + \hat{\boldsymbol{M}}_2) \sim N_Z\big((-1)^{i+1}\,½\boldsymbol{M}, \overline{\Sigma}\,(1 + 1/N)\big)$ and is independent of $\boldsymbol{Z}_2 = \hat{\boldsymbol{M}}_1 - \hat{\boldsymbol{M}}_2$. Then, $h^{\mathrm{E}}(\boldsymbol{X},\ \hat{\boldsymbol{M}}_1,\ \hat{\boldsymbol{M}}_2) = \boldsymbol{Z}^T \boldsymbol{Z}_2$.

Recognising that $(\boldsymbol{Z}^T\boldsymbol{U})^2 = \mathrm{tr}(\boldsymbol{Z}^T\boldsymbol{U}\,\boldsymbol{Z}^T\boldsymbol{U}) = \mathrm{tr}(\boldsymbol{U}\boldsymbol{U}^T\boldsymbol{Z}\boldsymbol{Z}^T)$, one can readily find that

$$E\left[h(\boldsymbol{X}, \hat{\boldsymbol{M}}_1, \hat{\boldsymbol{M}}_2)\middle|\boldsymbol{X} \in \omega_i\right] = E\left[\boldsymbol{Z}^T\boldsymbol{Z}_2)\middle|\boldsymbol{X} \in \omega_i\right] = (-1)^{i+1}\,½\,\boldsymbol{M}^T\boldsymbol{M}, \qquad (3.5a)$$

and

$$V\left[h(\boldsymbol{X}, \hat{\boldsymbol{M}}_1, \hat{\boldsymbol{M}}_2)\middle|\boldsymbol{X} \in \omega_i\right] = \boldsymbol{M}^T\,\overline{\Sigma}\,\boldsymbol{M}(1 + 2/N) + \mathrm{tr}(\overline{\Sigma}^{\,2})4/N\,(1 + 1/N). \qquad (3.5b)$$

An expression for the expected PMC follows directly from (3.4) and (3.5 ab):

$$\overline{\varepsilon}_N^{\mathrm{E}} \approx \Phi\{-\frac{\boldsymbol{M}^T\boldsymbol{M}}{2\sqrt{\boldsymbol{M}^T\overline{\Sigma}\boldsymbol{M}(1+\frac{2}{N}) + \mathrm{tr}\Sigma^2\,\frac{4}{N}(1+\frac{1}{N})}}\}. \qquad (3.6)$$

For large n and N, ignoring terms of order $1/N$ and n/N^2, we obtain the very simple expression

$$\overline{\varepsilon}_N^{\mathrm{E}} \approx \Phi\{-\frac{\delta^*}{2}\frac{1}{\sqrt{T_{\boldsymbol{M}}}}\}, \qquad (3.7)$$

where $\delta^* = \dfrac{\boldsymbol{M}^T\boldsymbol{M}}{\sqrt{\boldsymbol{M}^T\overline{\Sigma}\boldsymbol{M}}}$, $T_{\boldsymbol{M}} = 1 + \dfrac{4n^*}{\delta^{*2}N}$, $n^* = \dfrac{(\boldsymbol{M}^T\boldsymbol{M})^2(tr\,\overline{\Sigma}^2)}{(\boldsymbol{M}^T\overline{\Sigma}\boldsymbol{M})^2}$.

3.2.3 The Spherical Gaussian Case

3.2.3.1 The Case $N_2 = N_1$

Now, assume that $\overline{\Sigma} = \mathbf{I}\sigma^2$. Then $\delta^* = \delta$, and $n^* = n$, i.e., $T_{\boldsymbol{M}} = 1 + 2n/(\delta^2\,\overline{N})$. We recall that n represents the number of features (dimensionality) and $N = 2\,\overline{N}$ represents the training-set size. Equation (3.7) shows that the increase in the expected generalisation error depends on the asymptotic error rate and is proportional to n/N, the ratio of the dimensionality to the sample size. Therefore, for fixed N and δ, the generalisation error increases as n increases.

An important characteristic while designing the classification rules is the *learning quantity* $\kappa = \bar{\varepsilon}_N^{\mathrm{E}} / \varepsilon_\infty^{\mathrm{E}}$ – the relative increase in the mean generalisation error. It indicates the number of times that the generalisation error can be reduced by increasing the training-set size. E.g., if κ = 1.55, then one can expect that while increasing the training-set size, one can reduce the mean generalisation error 1.55 times. Therefore, theoretical values of this characteristic can be used for a rough estimation of the training-set size. The learning quantity of the EDC, κ_{EDC}, depends mainly on the ratio *N*/*n* and depends, to a lesser extent, on the asymptotic PMC.

For data model $\bar{\Sigma} = \mathbf{I}\sigma^2$ in Table 3.1 we present *exact theoretical values* for $\kappa = \bar{\varepsilon}_N^{\mathrm{A}} / \varepsilon_\infty^{\mathrm{E}}$ of the Euclidean distance classifier, the standard Fisher linear DF and the quadratic classifier obtained from the analytical formulae by integration (from Raudys and Pikelis, 1980). The reader can compare the asymptotic equation (3.7) with the exact values in the table and see that the asymptotic approximations are highly accurate. This evidence once more indicates that the double asymptotic analysis (when $\overline{N} \to \infty$, $n \to \infty$) is very useful tool, even in the very small training-set size case.

Table 3.1. Learning quantity, ratio $\kappa = \bar{\varepsilon}_N^{\mathrm{A}} / \varepsilon_\infty^{\mathrm{E}}$ of the Euclidean distance, the standard Fisher and standard quadratic classifiers versus *N*, the training-set size, for dimensionality *n*=50 and five values of distance δ and asymptotic error ε_∞ (Reprinted from Raudys and Pikelis, On dimensionality, sample size, classification error and complexity of classification algorithm in pattern recognition, *Pattern Analysis and Machine Intelligence* 2, 1980, © IEEE).

EDC					Fisher LDF					QDF					$\overline{N}$
1.82	2.34	3.09	3.66	4.22											8
1.70	2.03	2.41	2.65	2.87											12
1.54	1.70	1.84	1.92	1.99											20
1.43	1.50	1.55	1.58	1.61	2.05	3.39	8.40	19.7	52.0						30
1.30	1.32	1.33	1.34	1.35	1.62	2.15	3.61	5.95	10.6	2.21	3.25	7.87	18.3	40.6	50*
1.18	1.17	1.16	1.16	1.17	1.33	1.51	1.93	2.47	3.27	2.13	3.12	7.10	13.1	25.1	100
1.08	1.07	1.06	1.06	1.06	1.14	1.19	1.31	1.44	1.61	1.81	2.35	3.23	4.03	5.05	250
1.04	1.03	1.03	1.03	1.03	1.07	1.09	1.15	1.20	1.27	1.58	1.78	2.01	2.18	2.35	500
1.02	1.02	1.02	1.02	1.02	1.04	1.05	1.07	1.10	1.13	1.37	1.42	1.47	1.51	1.56	1000
1.01	1.01	1.01	1.01	1.01	1.01	1.02	1.03	1.04	1.05	1.18	1.16	1.18	1.18	1.20	2500
1.68	2.56	3.76	4.65	5.50	1.68	2.56	3.76	4.65	5.50	1.68	2.56	3.76	4.65	5.50	δ
0.2	0.1	0.03	0.01	.003	0.2	0.1	0.03	0.01	.003	0.2	0.1	0.03	0.01	.003	$\varepsilon_\infty^{\mathrm{E}}$

* 80 for QDF

We see that, in the spherical Gaussian case, the EDC can be trained with reasonably small training sets. Thus, in principle, the single layer perceptron can also be trained with small training sets. In Section 3.3 we show that in the non-spherical case (when $\bar{\Sigma} \neq \mathbf{I}\sigma^2$) we can encounter a more complex behaviour.

3.2.3.2 The Case $N_2 \neq N_1$

An important conclusion about the *non-optimality* of the sample-based plug-in Euclidean distance classifier follows from analysis of the case when $N_2 \neq N_1$. Deev (1970, 1972) considered the discriminant function of the form

$$h^* (\boldsymbol{X}) = \boldsymbol{X}^T \boldsymbol{V}^{(\mathrm{E})} + v_o + c$$

where an additional bias term c is introduced. He showed that asymptotically as N_1, N_2 and n, the dimensionality, increase, the expected PMC tends to

$$P_1 \Phi\{ -\frac{1}{2} \frac{\delta^2 - n/N_1 + n/N_2 - 2c}{\sqrt{\delta^2 + n/N_1 + n/N_2}} \} +$$

$$P_2 \Phi\{ -\frac{1}{2} \frac{\delta^2 + n/N_1 - n/N_2 + 2c}{\sqrt{\delta^2 + n/N_1 + n/N_2}} \}. \qquad (3.8)$$

When $n/N_1 << \delta^2$ and $n/N_2 << \delta^2$, the function $\Phi\{d\}$ on every side of $d = -\delta/2$ is almost linear. For $\bar{\Sigma} = \mathbf{I}\sigma^2$ ($\delta^* = \delta$, and $n^* = n$) we have no difference between (3.7) and (3.8). However, when these two conditions are not fulfilled the non-linear character of function $\Phi\{d\}$ causes an increase in the classification error. The following “unbiased” discriminant function can be recommended for case $N_2 \neq N_1$

$$h^{unbiased} (\boldsymbol{X}) = \boldsymbol{X}^T \boldsymbol{V}^{\mathrm{E}} + v_o + bias^{\mathrm{E}}(n, N_1, N_2),$$

where for the Euclidean distance classifier

$$bias^{\mathrm{E}}(n, N_1, N_2) = ½\, (n/N_2 - n/N_1) = c. \qquad (3.9)$$

A similar bias correcting term has to be used to improve the small sample properties of the standard Fisher linear DF. The non-optimality of the plug-in rule (when $N_2 \neq N_1$) is particularly high when the standard quadratic DF is used (Section 3.5.3).

3.3 Most Favourable and Least Favourable Distributions of the Data

One important aspect of our analysis is the investigation of cases where the statistical classifier is constructed according to one model of the multivariate density function and is applied to classify data described by another statistical model. It is a typical situation in practice. Equation (3.7) shows that, in principle, we can obtain both very small and very large increases in the generalisation error. Thus, in principle, the Euclidean distance classifier can be trained with particularly short training sets. This conclusion is very important for the preceptrons' analysis. It explains that, regardless of the formal dimensionality of the data, in some situations the single layer perceptron can also be trained with extremely short training sets.

3.3.1 The Non-Spherical Gaussian Case

From Equation (3.7) it follows that asymptotically, as the training-set size increases ($N \rightarrow \infty$), the expected generalisation error of the EDC tends to its asymptotic value $\varepsilon_{\infty}^{E} = \Phi\{-\delta^{*}/2\}$. In the general case when $\overline{\Sigma} \neq \mathbf{I}\sigma^2$, we note that $\delta^* \leq \delta$. Thus the asymptotic error ε_{∞}^{E} is larger than the asymptotic PMC of the standard Fisher DF $\varepsilon_{\infty}^{F} = \Phi\{-\delta/2\}$. Such a situation is illustrated in Figure 1.2.

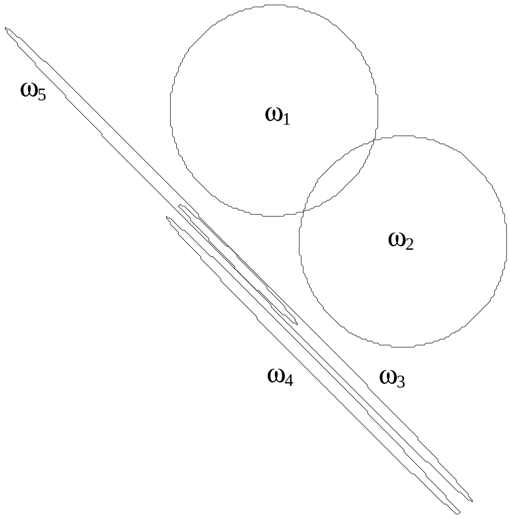

Fig. 3.2. Effect of the covariance matrix on the effective dimensionality *n**: for the classes ω_1 and ω_2, $n^* = n$; for ω_3, ω_4, $n^* >> n$; and for ω_3, ω_5, $n^* << n$ (Reprinted from *Neural Networks*, 11:297-313, Raudys, Evolution and generalization of a single neurone, 1998, with permission from Elsevier Science).

For the 20-variate Gaussian data model **C** mentioned in Section 1.5, we have that $\delta^* = 3.76$, $\delta = 4.65$, $\varepsilon_{\infty}^{E} = 0.03$ and $\varepsilon_{\infty}^{F} = 0.01$. There exists situations where the features are correlated but $\varepsilon_{\infty}^{E} = \varepsilon_{\infty}^{F}$. Two such examples are presented in Figure 3.2 (pairs of the classes ω_3, ω_4 and ω_3, ω_5). We see that the parameter δ^* controls the asymptotic PMC. Therefore, it is called the *effective Mahalanobis distance*. In the general case, $\bar{\Sigma} \neq \mathbf{I}\sigma^2$ and $1 \leq n^* \leq \infty$.

In Equation (3.7), the parameter n^* plays the role of the number of features and controls the sensitivity of the EDC to the training-set size. Therefore, it is called the *effective dimensionality*. We see that hypothetically there exist situations where the Euclidean distance classifier is either very *insensitive* to the number of training observations or, on the contrary, very *sensitive* to the training-set size.

3.3.2 The Most Favourable Distributions of the Data

When $n^* = 1$, we call the model the *most favourable distribution* of Gaussian pattern classes for the EDC. An example of densities of this type have been presented in Figure 3.2 (the pairs of the classes ω_3 and ω_5). In this example, the pattern classes are situated almost on a straight line. Here, the true (intrinsic) dimensionality of the data is close to 1.

Another example with $n^* \approx 1$ is two 100-variate Gaussian classes with common covariance matrix (GCCM): $\boldsymbol{M}_1 = -\boldsymbol{M}_2 = 1.042 \times (1, 1, \ldots, 1)^T$; unit variances; and the correlation between all variables $\rho = 0.3$. We call this data model **D**. In order to ensure $\delta^* = \delta = 3.76$ ($\varepsilon_\infty^{\mathrm{E}} = \varepsilon_\infty^{\mathrm{F}} = 0.03$), we have chosen the means, $\boldsymbol{M}_1$, $\boldsymbol{M}_2$, in a special manner. In this case, the formal dimensionality is $n = 100$, however the effective dimensionality, n^*, is $n^* \approx 1.05$, which is close to 1. Due to the low effective dimensionality for this choice of parameters, we can train the EDC with very small training sets. In a series of 10 experiments with training sets containing only *five* 100-variate vectors from each class ($N = 10$), we have obtained a very low generalisation error. The EDC yielded on average an error rate of 0.039 with the standard deviation of 0.009. In spite of the high number of variables ($n = 100$) for this very favourable distribution case, five vectors per class are sufficient to train the classifier adequately.

3.3.3 The Least Favourable Distributions of the Data

Hypothetically, there exist data models where the effective dimensionality is very large. We call such models where $n^* \rightarrow \infty$ *least favourable densities* of the pattern classes for the GCCM model. The pair classes ω_3 and ω_4 in Figure 3.2 are a perfect example of such distributions. Another example of the least favourable distribution is the following 100-variate ($n = 100$) GCCM data model:

$\boldsymbol{M}_1 = -\boldsymbol{M}_2 = 0.0018805 \times (1, 1, \ldots, 1)^T$; unit variances; all correlations $\rho = -0.0101$.

We call this data model **E**. For the data model **E** we have that $\delta^* = \delta = 3.76$ and $\varepsilon_\infty^{\mathrm{E}} = \varepsilon_\infty^{\mathrm{F}} = 0.03$. However, for these GCCM classes, the effective dimensionality is $n^* \approx 10^8$! In several experiments performed with Gaussian data model **E** and different training sets chosen at random ($\bar{N}$ = 200), we always obtained the expected error for the EDC of $\hat{\varepsilon}_N^{\mathrm{E}}$ = 0.497.

High sensitivity of the generalisation error of the EDC to the size of the training-set can be easily explained. In the above examples with high n^*, an insignificant deviation in the sample means $\hat{\boldsymbol{M}}_1$, $\hat{\boldsymbol{M}}_2$ causes a critical rotation of the decision boundary, and, thus, a crucial increase in the generalisation error.

Therefore, we can conclude that in extremely unfavourable cases, it is not expedient to utilise the EDC. For “almost singular” data, a more adequate type of the classifier should be used. For example, for the 100-variate negatively-correlated data model **E**, the more complex standard Fisher linear DF is an asymptotically optimal decision rule. Use of this classifier in the finite training-set case ($\bar{N}$ = 200), yielded a “reasonable” error of 0.058, i.e., only 1.93 times higher than the asymptotic error $\varepsilon_\infty^{\mathrm{F}}$ = 0.03. This result corresponds to the theoretical Equation (1.26) for δ=3.76 and $\bar{N}$ =2n; see also Table 3.1.

3.3.4 Intrinsic Dimensionality

Consider the GCCM data model $N_X(\boldsymbol{M}_1, \bar{\Sigma})$ and $N_X(\boldsymbol{M}_2, \bar{\Sigma})$. The matrix $\bar{\Sigma}$ can be represented as $\bar{\Sigma} = \boldsymbol{\Phi\Lambda\Phi}^T$, where $\boldsymbol{\Phi}$ is an $n \times n$ orthonormal matrix of eigenvectors of $\bar{\Sigma}$ and Λ is $n \times n$ diagonal matrix of eigenvalues. Let the diagonal matrix

$$\Lambda = \begin{bmatrix} \Lambda_r & \mathbf{0} \\ \mathbf{0} & \varepsilon \mathbf{I}_{n-r} \end{bmatrix},$$

where:
Λ_r is an $r \times r$ diagonal matrix ($r < n$);
$\mathbf{I}_{p-r}$ is an $(n-r) \times (n-r)$ identity matrix;
ε is a small positive constant, such that $(n-r)\varepsilon << 1$.

Also, let $(\boldsymbol{M}_1 - \boldsymbol{M}_2)^T\boldsymbol{\Phi} = (\boldsymbol{m}^T, \boldsymbol{m}_2{}^T)$, where absolute values of components m_{2j} of the $(n-r)$-variate vector $\boldsymbol{m}_2$ are very small, $m_{2j} << \varepsilon$, and can be ignored. In the data described by this model, the random pattern-class vectors $\boldsymbol{X}$ lie in a subspace of dimensionality r. We say that such data has the *intrinsic dimensionality*, r. The effective dimensionality of such data is

$$n^* = \frac{(\boldsymbol{M}^T\boldsymbol{M})^2(\mathrm{tr}\,\bar{\Sigma}^2)}{(\boldsymbol{M}^T\,\bar{\Sigma}\boldsymbol{M})^2} = \frac{(\boldsymbol{M}^T\boldsymbol{\Phi\Phi}^T\boldsymbol{M})^2\,\mathrm{tr}\,(\boldsymbol{\Phi\Lambda\Phi}^T\boldsymbol{\Phi\Lambda\Phi}^T)}{(\boldsymbol{M}^T\boldsymbol{\Phi\Lambda\Phi}^T\boldsymbol{M})^2} = \frac{(\boldsymbol{m}^T\boldsymbol{m})^2\,\mathrm{tr}(\Lambda_r^2)}{(\boldsymbol{m}^T\Lambda_r\boldsymbol{m})^2}\,.$$

Let Λ_r be an $r \times r$ identity matrix. Then for this data model we also have that $n^*=r$. The fact that that $n^* = r$ means that the increase in the generalisation error of the EDC does not depend on the formal dimensionality n. It depends on r, the dimensionality of the subspace. The intrinsic dimensionality of the multivariate data model with $n^* \approx 1$, discussed above in this subsection, is close to 1. In real world pattern classification problems, the pattern vectors often lie in non-linear subspaces of lower dimensionality. However, the variability of the data in the other n-r dimensions is not extremely small, i.e. the condition $(n\text{-}r)\ \varepsilon << 1$ is not fulfilled. For this configuration we have the intermediate cases. We note also that for certain distributions of components of the diagonal matrix Λ_r and the vector $\boldsymbol{m}$, we can have high values of the effective dimensionality: $n^* >> r$. Thus, the effective and intrinsic dimensionalities have *different interpretations*.

As a general conclusion one can say that the sensitivity of the Euclidean distance classifier to the training-set size strongly depends on the data. In principle, the sensitivity can be very low as well as extremely high. In practice, however, we seldom have distributions similar to the least favourable or to the most favourable distributions described in this section.

3.4 Generalisation Errors for Modifications of the Standard Linear Classifier

3.4.1 The Standard Fisher Linear DF

The weight vector of the standard Fisher linear DF has been defined in (1.3). When $\bar{N} \rightarrow \infty$ the Fisher LDF approaches the Bayes classifier for the model of two Gaussian classes sharing common covariance matrix (GCCM). For this model, the Bayes linear discriminant function has a Gaussian distribution with mean

$$E[h(\boldsymbol{X}) \mid \omega_i] = [(\boldsymbol{M}_i - ½\,(\boldsymbol{M}_1 + \boldsymbol{M}_2)]^T\, \bar{\Sigma}^{-1}(\boldsymbol{M}_1 - \boldsymbol{M}_2\,),$$

and variance

$$V[h(\boldsymbol{X}) \mid \omega_i] = (\boldsymbol{M}_1 - \boldsymbol{M}_2)^T\, \bar{\Sigma}^{-1}\, \bar{\Sigma}\ \ \bar{\Sigma}^{-1}(\boldsymbol{M}_1 - \boldsymbol{M}_2) = (\boldsymbol{M}_1 - \boldsymbol{M}_2)^T\, \bar{\Sigma}^{-1}(\boldsymbol{M}_1\text{-}\boldsymbol{M}_2).$$

Thus the asymptotic and the Bayes probabilities of misclassification are given by

$$\varepsilon_\infty^{\mathrm{F}} = P_1 Prob\ \{\ h^{\mathrm{F}}(\boldsymbol{X}) < 0 \mid \omega_1\} + P_2 Prob\ \{h^{\mathrm{F}}(\boldsymbol{X}) \geq 0 \mid \omega_2\} = \Phi\{-\delta\,/2\} = \varepsilon_B,$$

where $\delta^2 = (\boldsymbol{M}_1 - \boldsymbol{M}_2\,)^T\ \bar{\Sigma}^{-1}(\boldsymbol{M}_1 - \boldsymbol{M}_2\,)$ is a squared Mahalanobis distance.

3.4.2 The Double Asymptotics for the Expected Error

In the standard Fisher linear DF, the number of parameters to be estimated from the training-set is much larger than that for the EDC. In the two-category case, for the EDC we need to estimate $2n$ components of the mean vectors. For the Fisher linear DF we, need to estimate the additional $n(n+1)/2$ components of the covariance matrix. For the GCCM model with $N_2=N_1=\bar{N}$, the sample mean vectors are Gaussian, such that $\hat{\boldsymbol{M}}_1 \sim N(\boldsymbol{M}_1, \frac{1}{\bar{N}}\bar{\Sigma})$ and $\hat{\boldsymbol{M}}_2 \sim N(\boldsymbol{M}_2, \frac{1}{\bar{N}}\bar{\Sigma})$. The scaled covariance matrix is Wishart such that $(N\text{-}2)\hat{\Sigma} \sim \mathrm{W}(\bar{\Sigma}_n, N\text{-}2)$. Without proof, we present a final double asymptotic expression for the expected generalisation error,

$$\bar{\varepsilon}_N^{\mathrm{F}} \approx \Phi\{-\frac{\delta}{2}\frac{1}{\sqrt{T_{\boldsymbol{M}}T_{\bar{\Sigma}}}}\}, \qquad (3.10)$$

where the term $T_{\boldsymbol{M}} = 1+\frac{4n}{\delta^2 N}$ arises from the inexact sample estimation of the mean vectors of the classes and the term $T_{\bar{\Sigma}} = 1+\frac{n}{N-n}$ arises from the inexact sample estimation of the covariance matrix.

Table 3.1 and the terms $T_{\boldsymbol{M}}$ and $T_{\bar{\Sigma}}$ allow one to evaluate the *cost for estimation* of components of the n-variate vectors $\hat{\boldsymbol{M}}_1$, $\hat{\boldsymbol{M}}_2$ and the $n \times n$ matrix $\bar{\Sigma}$ in terms of the increase in the generalisation error or in terms of the training-set size.

Example 2. Let $\delta = 5.5$ ($\varepsilon_\infty^{\mathrm{E}} = \varepsilon_\infty^{\mathrm{F}} = 0.003$) and $n = 50$. In order to have the mean generalisation error smaller than 0.005 ($\kappa \le 1.61$) to estimate the mean vectors (EDC) we need $N = 30+30$ training vectors. To estimate the covariance matrix (the Fisher classifier) we need an additional N=220+220 training vectors (Table 3.1).

3.4.3 The Conditional Probability of Misclassification

For the linear classifier, let the weights estimated from the training-set be $\boldsymbol{V}$, v_0. Then the generalisation error of the linear classifier is

$$\varepsilon_N^{\mathrm{F}} = P_1\,\Phi\{-\frac{v_0+\boldsymbol{V}^T\boldsymbol{M}_1}{\sqrt{\boldsymbol{V}^T\bar{\Sigma}^{-1}\boldsymbol{V}}}\} + P_2\,\Phi\{\frac{v_0+\boldsymbol{V}^T\boldsymbol{M}_2}{\sqrt{\boldsymbol{V}^T\bar{\Sigma}^{-1}\boldsymbol{V}}}\}.$$

The weights $\boldsymbol{V}$, v_0 are functions of the two sample mean vectors $\hat{\boldsymbol{M}}_1$, $\hat{\boldsymbol{M}}_2$ and the sample covariance matrix $\hat{\Sigma}$. This fact implies that the weights are random variables. Consequently the conditional PMC is also a random variable. Asymptotically, as $N \to \infty$, the increase in the conditional classification error $\Delta\varepsilon_N^{\mathrm{F}} = \varepsilon_N^{\mathrm{F}} - \varepsilon_\infty^{\mathrm{F}}$ is distributed as

$$\frac{\phi(\delta/2)}{\delta N}\left(\frac{\delta^2}{4}\chi_1^2 + (1+\frac{\delta^2}{4})\chi_{n-1}^2\right).$$

We have already used this representation to depict the distribution densities of the conditional classification error in Figure 3.1. Utilising the fact that $E(\chi_r^2) = r$, and $\mathrm{Var}(\chi_r^2) = 2r$ we obtain one more expansion for the expected classification error

$$\bar{\varepsilon}_N^{\mathrm{F}} \approx \varepsilon_\infty^{\mathrm{F}} + \frac{\phi(\delta/2)}{\delta N}\left(\frac{\delta^2}{4} + (1+\frac{\delta^2}{4})(n-1)\right) . \tag{3.11}$$

3.4.4 A Standard Deviation of the Conditional Error

The representation of $\Delta\varepsilon_N^{\mathrm{F}}$ as the function of two chi-square random variables leads to a new result concerning the standard deviation of the conditional error, which is

$$\sqrt{V\varepsilon_N^{\mathrm{F}}} \approx \frac{\sqrt{2}\ \phi(\delta/2)}{\delta N}\sqrt{\frac{\delta^4}{16} + (1+\frac{\delta^2}{4})^2(n-1)} . \tag{3.12}$$

Formula (3.12) shows an interesting behaviour for the conditional error. When both N and n are increasing simultaneously and proportionally, the expected PMC tends to a constant value. The standard deviation, however, tends to zero**!** This fact suggests that, regardless of which randomly selected training-set we use in the high-dimensional case, we will obtain approximately the same conditional error. This is an important conclusion for practitioners. In the high-dimensional cases, we need to pay attention only to the expected PMC and need not be concerned about the variance of the conditional classification error.

The data in Table 3.1 can be used to verify the accuracy of expressions (3.10) and (3.11). When n is close to N, equation (3.10) is much more accurate than (3.11). Expression (3.11) shows that the increase in the expected generalisation error is proportional to ratio n/N. However, Equation (3.10) shows that this phenomenon is true only when $N >> n$. When N is close to n, the generalisation error becomes very large and then we have more complex dependence.

Equations (3.7) and (3.10) can be used to demonstrate the scissors and peaking effects discussed in Chapter 1. When the training-set size is comparatively small, instead of using the Fisher rule, one needs to use a simple decision rule – the EDC. An alternative to using a simple classifier is to use the Fisher rule and reduce the number of the features.

3.4.5 Favourable and Unfavourable Distributions

While analysing the EDC we have seen that there exist favourable and unfavourable pattern-class distributions. A similar situation is characteristic for the Fisher classifier. To design the standard Fisher classifier one must estimate the covariance matrix. In theory, one must have the training-set size at least equal to the number of the features. However, in practical cases, one needs a much larger training-set size. More precise evaluations follow from analysing the terms $T_{\boldsymbol{M}}$ and $T_{\overline{\Sigma}}$ in the Equation (3.10).

In principle, for non-Gaussian data models one can create special artificial data models for which the Fisher classifier can be perfectly trained using $n + 1$ training vectors. To create such an example, we leave this as an exercise for the reader. A similar conclusion is valid for the least favourable case – one can create models for which the Fisher classifier is very sensitive to the number of training vectors.

Example 3. In Figure 3.3, we have two bimodal uni-variate densities. For such data the Fisher classifier requires a tremendous number of training examples. In this case, an insignificant deviation in the sample means causes an enormous increase in the generalisation error. This data model is also unfavourable for the EDC as well as for the RDA classifier.

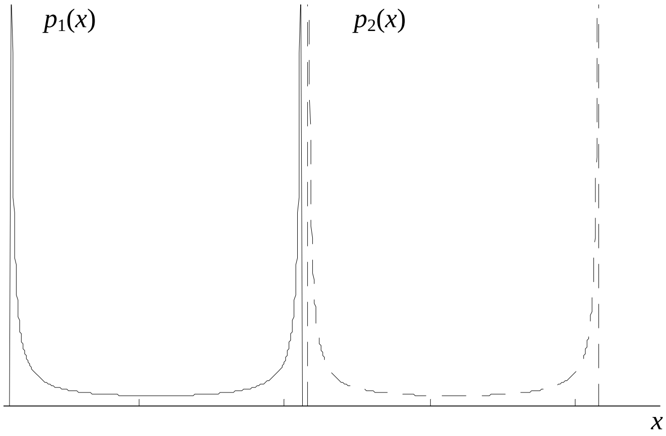

Fig. 3.3. Unfavourable distributions for the Fisher linear DF. Two bimodal densities, p_1 (x) – a bold line and p_2 (x) – a dashed line.

3.4.6 Theory and Real-World Problems

For real data problems one never deals with the most favourable case nor with the most unfavourable case. Among several dozens of real world-pattern recognition problems solved by the author of this book, no extremely bizarre cases involving unfavourable pattern-class distributions has been met**.** Analysis of training-set and test set error rates presented in numerous research papers in the literature also leads one to think that very unfavourable distributions are not encountered in real world problems. Most often the theoretical estimates obtained for the Gaussian model approximately characterises the learning quantity estimated experimentally. Notable exceptions are visibly multimodal data sets where the linear or quadratic classifiers should not be used. Rather, one should use nonparametric classifiers such as Parzen window, *k-NN* rule, or piecewise-linear classifiers, etc.

In Table 3.2 we present theoretical estimates (upper rows), and experimental estimates (lower rows) of the test set errors evaluated by using four real-world data sets. Each estimate is a mean value obtained from 50 repetitions of the experiment with different, randomly-chosen training sets. The theoretical estimates are calculated for a very idealised data model – multivariate spherical Gaussian distributions. Visual analysis of histograms of the data set features showed that all data sets were more or less unimodal but some features were highly asymmetrical. In spite of notable differences of the real distributions from the idealised Gaussian distributions, the theoretical estimates of κ are tolerably close to the experimental values. Therefore, the author recommends that one use the values of the learning quantity $\kappa = \bar{\varepsilon}_N^{\mathrm{A}} \,/\, \varepsilon_\infty^{\mathrm{A}}$ found for the spherical Gaussian model (Table 3.1, Equations (3.10) and (3.7) with $n^* = n$) as a rule of thumb.

Table 3.2. Comparison of the theoretical and empirical values of ratio $\kappa = \bar{\varepsilon}_N^{\mathrm{A}} \,/\, \varepsilon_\infty^{\mathrm{A}}$ (Reprinted from Raudys and Pikelis, On dimensionality, sample size, classification error and complexity of classification algorithm in pattern recognition, *Pattern Analysis and Machine Intelligence* 2, 1980, © IEEE).

	Data set 1	Data set 2	Data set 3	Data set 4
	$n = 5$	$n = 5$	$n = 32$	$n = 6$
	$Nt = 500$	$Nt = 600$	$Nt = 300$	$Nt = 600$
$\bar{N}$	12 20 50	12 20 50	20 50	12 20 50
E theory	1.14 1.09 1.04	1.19 1.101.04	1.18 1.08	1.13 1.09 1.04
experiment	1.12 1.09 1.04	1.18 1.14 1.10	1.11 1.00	0.98 1.01 1.00
F theory	1.67 1.30 1.12	1.70 1.32 1.12	4.51 2.55	1.32 1.13 1.07
experiment	1.44 1.32 1.09	2.01 1.63 1.32	13.6 6.62	1.23 1.20 1.10

3.4.7 The Linear Classifier D for the Diagonal CM

One modification of the linear Fisher DF, discussed in Section 2.3.1, is the classifier

$$\hat{h}^{\mathrm{D}}(\boldsymbol{X}) = (\boldsymbol{X} - ½\,(\hat{\boldsymbol{M}}_1 + \hat{\boldsymbol{M}}_2))^T\, \hat{\boldsymbol{D}}^{-1}(\hat{\boldsymbol{M}}_1 - \hat{\boldsymbol{M}}_2), \tag{3.13}$$

where $\hat{\boldsymbol{D}}$ is a sample estimate of the diagonal variance matrix composed from diagonal elements of $\hat{\boldsymbol{\Sigma}}$, the pooled sample estimate of the covariance matrix.

The discriminant function (3.13) is a function of two random vectors and random diagonal matrix

$$h^{\mathrm{D}}(\boldsymbol{X},\, \hat{\boldsymbol{M}}_1,\, \hat{\boldsymbol{M}}_2,\, \hat{\boldsymbol{D}}) = \boldsymbol{Z}^T\, \hat{\boldsymbol{D}}^{-1}\boldsymbol{Z}_2,$$

where $\boldsymbol{Z}$ and $\boldsymbol{Z}_2$ have been defined in Section 3.2.2.

Let the *true distributions* of the pattern classes be spherical Gaussian $N_X(\boldsymbol{M}_r\,,\,\mathbf{I}\sigma^2)$. We assume that the classifier designer does not know that $\bar{\Sigma} = \mathbf{I}\sigma^2$. In order to design the classifier (3.13) he utilises the variance matrix $\boldsymbol{D}$ estimated from the training-set. The matrix $\boldsymbol{D}$ is composed from n estimates $d_1, d_2, \dots, d_n$ of the variances of all n features. Let $N_2 = N_1 = \bar{N} = N/2$, and $P_2 = P_1 = ½$. Then each single element d_j of diagonal matrix $\hat{\boldsymbol{D}}$ is a scaled chi-square random variable of the form $(N\text{-}2)^{-1}\chi^2_{N-2}$.

The first two statistical moments of an inverse of a chi-square random variable are

$$E(\chi_r^2)^{-1} = \frac{1}{r-2}\,,\ \text{and}\quad E(\chi_r^2)^{-2} = \frac{1}{(r-2)(r-4)}\,.$$

Thus, $E(d_j^{-1}) = \dfrac{N-2}{N-4}$ and $E(d_j^{-2}) = \dfrac{(N-2)^2}{(N-4)(N-6)}$. As in Section 3.2.2, we have that

$$E\left[h(\boldsymbol{X}, \hat{\boldsymbol{M}}_1, \hat{\boldsymbol{M}}_2, \hat{\boldsymbol{D}}) \mid \boldsymbol{X} \in \omega_i\right] = (\text{-}1)^{i+1}\,\frac{1}{2}\,\frac{N-2}{N-4}\ \boldsymbol{M}^T\boldsymbol{M} = (\text{-}1)^{i+1}\frac{1}{2}\,\frac{N-2}{N-4}\,\delta^2 \quad (3.14a)$$

and $V\left[h(\boldsymbol{X}, \hat{\boldsymbol{M}}_1, \hat{\boldsymbol{M}}_2, \hat{\boldsymbol{D}}) \mid \boldsymbol{X} \in \omega_i\right] =$

$$\left(\frac{N-2}{N-4}\right)^2 \sum_{j=1}^{n}\left(\frac{N-4}{N-6}(m_j^2 + \frac{4}{N})(\frac{m_j^2}{4} + 1 + \frac{1}{N}) + \frac{m_j^4}{4}\right), \qquad (3.14b)$$

where $m_1, m_2, \dots, m_n$ are the components of vector $\boldsymbol{M} = \boldsymbol{M}_1 - \boldsymbol{M}_2$, $\delta^2 = \boldsymbol{M}^T\boldsymbol{M}$, and we denote $\delta_{(4)} = \dfrac{1}{\delta^2}\sum_{j=1}^{n} m_j^4$.

An expression for the expected PMC follows directly from (3.4) and (3.14*ab*):

$$\bar{\varepsilon}_N^{\mathrm{D}} \approx \Phi\{ -\frac{\delta}{2\sqrt{(1+\dfrac{2}{N-6})(1+\dfrac{2}{N}\,(1+\dfrac{2n}{\delta^2}) + \dfrac{4n}{\delta^2 N^2}) + \dfrac{\delta_{(4)}}{2(N-6)}}}\ \}. \qquad (3.15)$$

For large n and N, ignoring the terms of order $\frac{1}{N}$, and $\frac{n}{N^2}$, we have the approximation

$$\bar{\varepsilon}_N^{\mathrm{D}} \approx \Phi\{-\frac{\delta}{2}\frac{1}{\sqrt{T_{\boldsymbol{M}}}}\}, \tag{3.16}$$

where the term $T_{\boldsymbol{M}} = 1 + \frac{2n}{\delta^2 \overline{N}}$ corresponds to the estimation of the mean vectors.

Analytical formulae for the generalisation error of EDC and the "diagonal" classifiers, (3.7) and (3.16), indicate that both expressions are asymptotically identical. Therefore estimation of n common variances asymptotically (as $N \to \infty$ and $n \to \infty$) does not affect the increase in the generalisation error! This fact suggests that the estimation of parameters that are common to both pattern classes is less important than the estimation of differing parameters. We will consider this problem more in Section 3.5.

3.4.8 The Pseudo-Fisher Classifier

In the Pseudo-Fisher classifier, instead of the conventional inverse of the sample covariance matrix $\hat{\Sigma}$ we use the pseudo-inverse (Section 2.5.1). Here instead of classifying the original vectors $\boldsymbol{X}$ we classify the new vectors $\boldsymbol{X}_{new} = \boldsymbol{\Phi}_1^T \boldsymbol{X}$ in an r-dimensional subspace corresponding to the non-zero eigenvalues of the covariance matrix $\hat{\Sigma}$ ($r = N$-2 is the rank of the sample covariance matrix $\hat{\Sigma}$). Recall, that in the "diagonal" classifier design paradigm, one assumes that the covariance matrix of the vector to be classified is a diagonal matrix composed of the diagonal elements of the sample covariance matrix $\hat{\Sigma}$. In the new space, the Pseudo-Fisher classifier's covariance matrix is a diagonal matrix Λ composed of the variances of the vector $\boldsymbol{X}_{new}$ – the r non-zero eigenvalues $\lambda_1, \lambda_2, ..., \lambda_r$ of $\hat{\Sigma}$.

The generalisation error of the Pseudo-Fisher classifier can be viewed as follows. In the pseudo-inverse approach, the feature space is rotated by means of a certain (random) orthogonal transformation $\boldsymbol{X}_{new} = \boldsymbol{\Phi}_1^T \boldsymbol{X}$ and afterwards, unlabeled data are classified by the "diagonal" classifier in the new r-variate feature space corresponding to the r non-zero eigenvalues of the sample covariance matrix $\hat{\Sigma}$. The Mahalanobis distance in the r-variate space increases with $r = N$-2, however, the accuracy of determination of the eigenvalues $\lambda_1, \lambda_2, ..., \lambda_r$ decreases. If both n and N are large ($N < n$), the expected error of the Pseudo-Fisher classifier is

$$\bar{\varepsilon}_N^{\mathrm{PF}} \approx \Phi\left\{-\frac{\delta\sqrt{r/n}}{2}\frac{1}{\sqrt{(1+\gamma^2)T_M + \gamma^2\frac{3\delta^2}{4n}}}\right\}, \tag{3.17}$$

where $\gamma = \sqrt{V_d} \,/\, E_d$; E_d, V_d are respectively mean and variance of a random variable $1/\lambda_r$ and λ_r is a randomly chosen eigenvalue of matrix $\hat{\Sigma}$ having Wishart W($\mathbf{I}_n$, N-2) distribution.

Equation (3.17) is similar to (3.7). Analysis of equation (3.17) shows that the terms *r/n* and γ are increasing as the training-set size *N* increases from 1 up to *n*. E.g., for dimensionality *n* =20 we have:

$\gamma = 0.32$ when $N = 3$;
$\gamma = 1.03$ when N=11; and
$\gamma = 9.75$ when N =21.

Thus, as *N* increases the term *r/n* in the nominator of (3.17) causes a decrease in the generalisation error while the term γ in the denominator of (3.17) causes an increase the generalisation error.

The differing behaviours of the terms *r/n* and γ produce an interesting and unexpected behaviour of the generalisation error: with an increase in the training-set size, $N = 2\overline{N}$, from 1 to *n*, the generalisation error decreases at first, reaches the minimum, and afterwards begins increasing (see curve 3 in Figure 3.4).

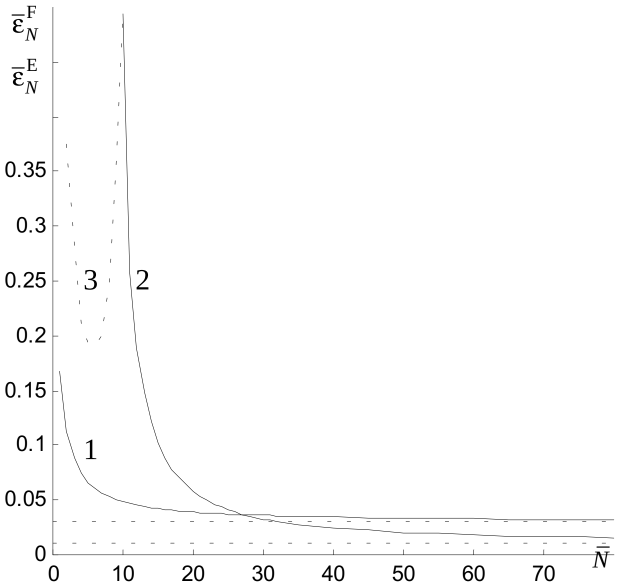

Fig. 3.4. The scissors effect: the generalisation error versus $\overline{N}$: 1 – EDC (for *n** = 20), 2, 3 – the Pseudo-Fisher and Fisher classifiers. For $\overline{N}$ < 30 the EDC is preferred.

The minimal error is obtained for $\overline{N}$ = *n*/2 and the maximal error is obtained for *N* = *n*. This complex and “strange” behaviour is a consequence of non-optimality of the plug-in Pseudo-Fisher classifier. If the training-set size is *N* > *n*, then we obtain the Fisher linear DF and the expected error decreases regularly as *N* increases. In Chapter 4 we will see that the non-linear single layer perceptron can also exhibit such peaking behaviour.

3.4.9 The Regularised Discriminant Analysis

RDA is an intermediate classifier between the EDC and the Fisher linear DF. In RDA while calculating the weights of the linear discriminant function, one uses the ridge estimate, $\hat{\Sigma}^{\mathrm{RDA}} = \hat{\Sigma} + \lambda\mathbf{I}$, rather than the conventional pooled sample estimate $\hat{\Sigma}$. Thus, while changing the parameter λ, we obtain a sequence of classifiers ranging from the EDC ($\lambda = 0$) to the Fisher linear DF ($\lambda \to \infty$). We can obtain the RDA classifier in the SLP training, too.

Positive values of λ added to each diagonal element of the sample covariance matrix help to invert the covariance matrix and act as regularisers. At the same time, the addition of $\lambda\mathbf{I}$ to the true covariance matrix distorts the weights vector and, consequently, increases the classification error. We can evaluate this effect numerically.

When the term $\lambda\mathbf{I}$ is added to the true covariance matrix Σ, we have the following discriminant function

$$h^{\mathrm{RDA}}(\boldsymbol{X}) = \boldsymbol{X}^T(\hat{\Sigma} + \lambda\mathbf{I})^{-1}(\boldsymbol{M}_1 - \boldsymbol{M}_2) - ½\,(\boldsymbol{M}_1 + \boldsymbol{M}_2)^T(\hat{\Sigma} + \lambda\mathbf{I})^{-1}(\boldsymbol{M}_1 - \boldsymbol{M}_2).$$

For the GCCM model, this linear discriminant function has a Gaussian distribution with mean

$$E[h^{\mathrm{RDA}}(\boldsymbol{X}) \mid \omega_i] = [(\boldsymbol{M}_i - ½\,(\boldsymbol{M}_1 + \boldsymbol{M}_2)]^T(\bar{\Sigma} + \lambda\mathbf{I})^{-1}(\boldsymbol{M}_1 - \boldsymbol{M}_2),$$

and variance

$$V[h^{\mathrm{RDA}}(\boldsymbol{X}) \mid \omega_i] = (\boldsymbol{M}_1 - \boldsymbol{M}_2)^T(\bar{\Sigma} + \lambda\mathbf{I})^{-1}\,\bar{\Sigma}\,(\bar{\Sigma} + \lambda\mathbf{I})^{-1}(\boldsymbol{M}_1 - \boldsymbol{M}_2).$$

Assuming equal prior probabilities of the classes, we have that the asymptotic probability of misclassification is

$$\varepsilon_\infty^{\mathrm{RDA}} = ½\,Prob\{h^{\mathrm{RDA}}(\boldsymbol{X})) < 0 \mid \omega_1\} + ½\,Prob\{h^{\mathrm{RDA}}(\boldsymbol{X}) \geq 0 \mid \omega_2\} = \Phi\{-½\,\delta_\lambda\} \geq \varepsilon_B,$$

where $\delta_\lambda^2 = \dfrac{(\boldsymbol{M}^{\boldsymbol{T}}(\bar{\Sigma} + \lambda\mathbf{I})^{-1}\boldsymbol{M})^2}{\boldsymbol{M}^{\boldsymbol{T}}(\bar{\Sigma} + \lambda\mathbf{I})^{-1}\bar{\Sigma}(\bar{\Sigma} + \lambda\mathbf{I})^{-1}\boldsymbol{M}}$ is the squared modified Mahalanobis distance for the RDA classifier and $\boldsymbol{M} = \boldsymbol{M}_1 - \boldsymbol{M}_2$.

Analysis of expression for $\varepsilon_\infty^{\mathrm{RDA}}$ shows that growth in the regularisation parameter λ increases the asymptotic error.

Example 4. For the data model depicted in Figure 1.2 for $\lambda = 0$, we have that the asymptotic and Bayes errors are $\varepsilon_\infty^{\mathrm{F}} = \varepsilon_B = 0.0401$. As $\lambda \to \infty$, this error rate gradually increases until $\varepsilon_\infty^{\mathrm{E}} = 0.0912$. For the 20-variate correlated GCCM data **C**

discussed in Section 1.4, if we let $\lambda = 0$, the asymptotic and the Bayes errors are $\varepsilon_{\infty}^{\mathrm{RDA}} = \varepsilon_B = 0.01$. When $\lambda \rightarrow \infty$ we find that $\varepsilon_{\infty}^{\mathrm{RDA}} = \varepsilon_{\infty}^{\mathrm{E}} = 0.03$.

For very small values of λ the expected (generalisation) error can be represented by the following asymptotic formula:

$$\bar{\varepsilon}_N^{\mathrm{RDA}} \rightarrow \Phi\{-\frac{\delta_\lambda}{2}\frac{\sqrt{1+T_\lambda}}{\sqrt{T_{\boldsymbol{M}}\, T_{\bar{\Sigma}}}}\}, \tag{3.18}$$

where T_λ is a certain function of $\bar{\Sigma}$, $\boldsymbol{M}$ and λ. Expression (3.18) is a very complex function of $\boldsymbol{M}$, $\bar{\Sigma}$ and is accurate only for very small λ. Therefore, we do not discuss the implications of Expression (3.18) in detail here. However, (3.18) explains that the positive term T_λ reduces the negative influence of $T_{\bar{\Sigma}}$. This term increases with λ and, to some extent, it compensates for inaccuracies ("noise") caused by the inexact estimation of the covariance matrix. When $\lambda \rightarrow 0$, the term $T_\lambda \rightarrow 0$.

When the training-set size $N \rightarrow \infty$, then $\bar{\varepsilon}_N^{\mathrm{RDA}} \rightarrow \varepsilon_{\infty}^{\mathrm{RDA}} = \Phi\{-\delta_\lambda/2\}$. Contrary to the asymptotic error, the learning quantity $\kappa = \bar{\varepsilon}_N^{\mathrm{RDA}} / \varepsilon_{\infty}^{\mathrm{RDA}}$ – the relative increase in the mean generalisation error – *decreases* with increasing λ. For the data model **C** and training set size $N = 2n = 40$ for $\lambda = 0$ (the Fisher classifier), we have $\kappa = 2.47$ When $\lambda \rightarrow \infty$ (EDC) this value decreases until $\kappa = 1.16$

Thus, we have the following dual behaviour: the growth of λ increases the asymptotic error, but improves the classifier's small sample properties (reduces the ratio $\kappa = \bar{\varepsilon}_N^{\mathrm{RDA}} / \varepsilon_{\infty}^{\mathrm{RDA}}$). As a result, in the finite training-set size case, as λ increases, most often the generalisation error decreases at first, arrives at the minimum and then increases again. This phenomenon is known as the *peaking effect*, similar to the peaking that arises with an increase in the number of the features (see Section 1.5).

The value of λ where the generalisation error is minimised is called the *optimal value of the regularisation parameter*. In Figure 3.5, we have three experimental graphs (1-1, 1-2, 1-3) of the generalisation error against λ, determined for the artificial 20-variate GCCM data model **C** and three randomly selected training sets of size $\bar{N} = 80$. We also present two estimates of the mean generalisation error – the average of the conditional error over 25 random training-sets of sizes $\bar{N} = 80$ (graph 2) and $\bar{N} = 16$ (graph 3). The minima of the graphs are denoted by "×".

Note that the optimal value of λ_{opt} depends on each particular data type (the dimensionality, a configuration of the data points distribution, etc.) and the training-set size. In general, λ_{opt} decreases with the increase in the training-set size. For the GCCM pattern classes we have $\lambda_{opt} \rightarrow 0$ as $N \rightarrow \infty$. However, we have noted in related experiments that λ_{opt} fluctuates with each particular training-set.

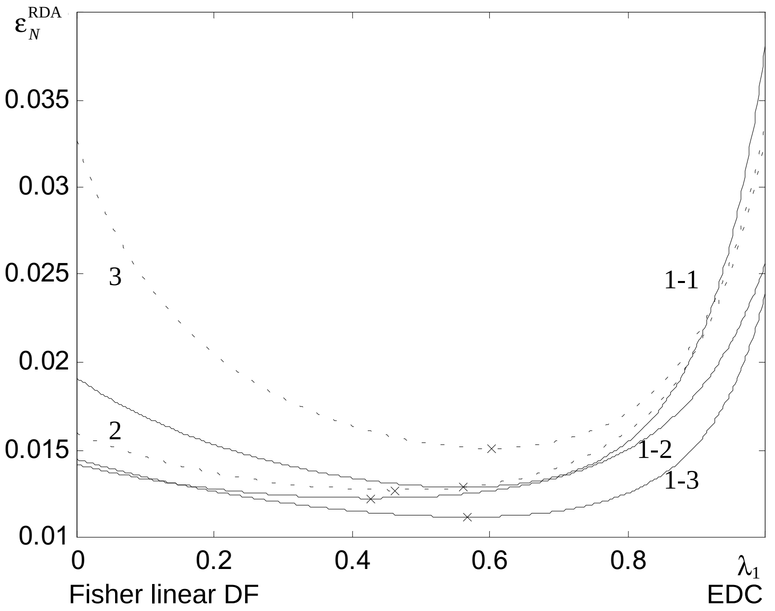

Fig. 3.5. The peaking effect: the generalisation error as a function of the regularisation parameter $\lambda_1=\lambda/(1+\lambda)$. 20-variate correlated Gaussian data: 1-1, 1-2, 1-3 give the generalisation errors for three randomly selected training-sets of size $\bar{N}$ = 80, 2 gives the generalisation error averaged over 25 random training-sets of size $\bar{N}$ = 80, 3 gives the generalisation error averaged over 25 random training-sets, $\bar{N}$ = 32.

3.5 Common Parameters in Different Competing Pattern Classes

While designing several parametric statistical classification algorithms, we note that a portion of the distribution density parameters are common to both pattern classes. For instance, in the standard Fisher linear DF, we assume that both populations share the same covariance matrix $\bar{\Sigma}$. In the "diagonal classifier" D, we assume that n variances $d_1, d_2, \ldots, d_n$ are common in both populations. In both cases, we assume that the mean vectors $\boldsymbol{M}_1$, $\boldsymbol{M}_2$ are different. An important issue discussed in this section is the fact that for certain conditions, the parameters of the multivariate distribution density functions common for all pattern classes have little influence on the generalisation error when compared to the parameters that differ.

In the multi-layer perceptron and the radial basic function neural networks, the weights of the hidden layer are typically connected with all outputs. This relationship suggests that they are common for all pattern classes. Thus, according to this argument, the hidden layer weights should have less influence on classifier performance than the output layer weights.

3.5.1 The Generalisation Error of the Quadratic DF

The classical statistical classification method, where one assumes that the covariance matrices and means are different, is the standard quadratic DF, which is given in (2.11). The number of parameters to be estimated from the training-set is $n(n+3)/2$ per class. The double asymptotic analysis shows that in the case of GCCM classes, the expected PMC of this classifier can be asymptotically approximated as

$$\bar{\varepsilon}_N^{\mathrm{Q}} \approx \Phi\{ -\frac{\delta}{2} \frac{1}{\sqrt{T_{\Sigma\mathrm{A}} T_{\boldsymbol{M}} + T_{\Sigma\mathrm{B}}}} \}, \tag{3.19}$$

where $T_{\Sigma\mathrm{A}} = 1 + \dfrac{n}{\bar{N} - n}$ and $T_{\Sigma\mathrm{B}} = \dfrac{\delta^4 / 2 + n(2 + n / \delta^2)}{\bar{N} - n}$.

We can see differences and similarities between (3.10) and (3.19). For both discriminant functions we have the term $T_{\boldsymbol{M}}$ multiplied by the term T_Σ ($T_{\Sigma\mathrm{A}}$). In the term $T_{\Sigma\mathrm{A}}$, however, we use $\bar{N}$, the sample-size of one single class, instead of $N = 2\bar{N}$ in the term T_Σ for the linear Fisher DF. This difference is caused by the fact that to estimate the covariance matrix of each single class we use only half of the training vectors. In expression (3.19) for the quadratic discriminant function, we have *additional term*, $T_{\Sigma\mathrm{B}}$. This term reflects the situation that the covariance matrices are *different* in both classes and thus, indicates that for small δ or large n, the increase in the generalisation error can become proportional to n^2/N. This term can be comparatively small for highly overlapping classes (when δ, the distance between the pattern classes, is small) with low dimensions. The influence of this term increases as the dimensionality n and the distance δ increase. Thus, in the high-dimensional case or when the classification error is small, the quadratic discriminant function becomes very sensitive to the training-set size. The SLP, the Anderson–Bahadur linear DF, the modified RDA (2.39) and the MLP can become good competitors to the standard quadratic discriminant function in this case.

3.5.2 The Effect of Common Parameters in Two Competing Classes

In order to design the EDC we need to estimate n components of the mean vector $\boldsymbol{M}_1$ and n components of the mean vector $\boldsymbol{M}_2$. In the "diagonal" classifier we estimate the mean vectors $\boldsymbol{M}_1$, $\boldsymbol{M}_2$, and n non-zero elements of the diagonal variance matrix $\boldsymbol{D}$. Analytical formulae for the generalisation error of both classifiers, (3.7) and (3.16), respectively, indicate that the expressions are asymptotically identical. Thus, we have the following result:

> estimation of n common variances asymptotically does not affect the increase in the generalisation error.

Formula (3.16) is to some extent general and can be used for the case when we use the Fisher classifier with the constrained “the first order tree type” estimate of the covariance matrix containing 2*n*-1 different components. In this covariance matrix model, 2*n*-1 parameters are common for both pattern classes. Thus, asymptotically (as $N \rightarrow \infty$ and $n \rightarrow \infty$), the generalisation error for this classifier is the same as for the EDC. This formula is also valid for the number of other models with constrained covariance matrices that are discussed in Chapter 2.

A similar formula is valid for the linear statistical classifier (2.16) designed for the block-structured covariance matrix model (2.15). Here we have

$$\bar{\varepsilon}_N^{\mathrm{BD}} = P_1 \Phi\left(- \frac{\sum_{j=1}^{H} \frac{N_1+N_2}{N_1+N_2-n_j}\left(\delta_j^2 - \lambda_{1j} + \lambda_{2j}\right) - 2c}{2\sqrt{\sum_{j=1}^{H}\left(\delta_j^2 + \lambda_{1j} + \lambda_{2j}\right)}} \right) +$$

$$P_2 \Phi\left(- \frac{\sum_{j=1}^{H} \frac{N_1+N_2}{N_1+N_2-n_j}\left(\delta_j^2 + \lambda_{1j} - \lambda_{2j}\right) + 2c}{2\sqrt{\sum_{j=1}^{H}\left(\delta_j^2 + \lambda_{1j} + \lambda_{2j}\right)}} \right) \qquad (3.20)$$

where δ_j^2 is the squared Mahalanobis distance associated with *j*-th n_j-dimensional block, *H* is a number of independent blocks, $\lambda_{ij} = n_j / N_i$, and c is a constant.

In order to better understand the above equation assume N_2=N_1= $\bar{N}$, P_2=P_1= ½, c = 0, and all blocks are of equal size, i.e. $n_1 = n_2 =$, …, $= n_H = n/H$. Then, (3.20) can be written as

$$\bar{\varepsilon}_N^{\mathrm{BD}} \approx \Phi\{ -\frac{\delta}{2}\frac{1}{\sqrt{T_M T_\Sigma^{BD}}} \}, \qquad (3.21)$$

where δ^2 is the squared Mahalanobis distance, the term $T_{\boldsymbol{M}} = 1 + \frac{4n}{\delta^2 N}$ arises from estimation of the mean vectors of the classes and is defined in (3.10). The term

$$T_\Sigma^{BD} = 1 + \frac{n_H}{N - n_H} \qquad (3.22)$$

is caused by the sample estimates of the separate blocks of the covariance matrix. It differs from the similar term T_Σ in (3.10). When we have many blocks and when n is close to N, we have an obvious gain from the block-wise structuralisation of the covariance matrix. When $n \rightarrow \infty$, $N \rightarrow \infty$, and the size of each block $n_H = n/H$ does not change, the term $T_{\Sigma}^{BD} \rightarrow 1$. We see that asymptotically, estimation of $H(n_H + 1)n_H$ parameters for the common blocks does not affect the increase in the generalisation error.

Generalisation of the above results for the diagonal and block-diagonal classifiers was accomplished by Meshalkin and Serdobolskij (1978). They analysed the plug-in classification rules where the h_d characteristics of the multivariate distribution density functions are different in each of two pattern classes and that h_c characteristics are common. They showed that

> asymptotically when both the number of parameters, h_d and h_c, and the sample size N are increasing proportionally, and the distance between pattern classes does not change, the increase in the generalisation error is determined by the number of differing parameters.

This result is of fundamental significance for understanding the classification performance of the multi-layer perceptrons and radial basic function neural networks. For artificial neural networks, the weights of the hidden layer typically are connected with all outputs. Thus, they are common for all pattern classes. Therefore, one can guess that, the hidden layer weights will have less influence than the weights of the output layer.

The general asymptotic theorems, however, hide some terms that are important in the realistic case of pattern recognition problems with finite dimensions and finite training sets. For example, the term $\delta_{(4)} /(4(N - 3))$ in Equation (3.15) shows that, in principle, situations can exist where the conclusion about minor influence of common parameters can be incorrect. For some non-spherical data, the term $\delta_{(4)} /(4(N - 3))$ can increase as the number of the features n increases. Consequently, one can get an increase in the generalisation error. We have already discussed a similar situation in Section 3.4, where we have analysed the Pseudo-Fisher classifier in the small training-set case.

3.5.3 Unequal Sample Sizes in Plug-In Classifiers

In the plug-in approach to classifier design, we use unbiased estimates $\hat{\mathbf{Y}}_i$ of the parameter vector $\mathbf{Y}_i$ of the pattern-class densities. In spite of the fact that the estimates $\hat{\mathbf{Y}}_i$ are unbiased, the plug-in density estimates $p_i(\boldsymbol{X} \mid \hat{\mathbf{Y}}_i)$ are biased. The sample plug-in discriminant function is

$$h(\boldsymbol{X}) = \log p_1(\boldsymbol{X} \mid \hat{\mathbf{Y}}_1, \hat{\mathbf{Y}}) - \log p_1(\boldsymbol{X} \mid \hat{\mathbf{Y}}_2, \hat{\mathbf{Y}}) + \log \frac{P_1}{P_2} ,$$

where $\hat{\mathbf{Y}}$ is a sample estimate of the parameter vector common to both pattern classes, and $\hat{\mathbf{Y}}_1, \hat{\mathbf{Y}}_2$ are parameter vectors that are unique to each pattern class.

This discriminant function is biased, too. The bias increases as the difference between N_1 and N_2 increases, causing an unnecessary increase in the generalisation error (see e.g. Section 3.2.3.2). Therefore, the bias-correcting term (3.9) was suggested. Equation (3.20) indicates a way for obtaining a bias-correcting term *c* for the standard Fisher linear DF and its variant with block covariance matrix (2.16). For the linear Fisher DF the bias term is

$$bias^{\mathrm{F}}(n, N_1, N_2) = ½T_{\Sigma}(n/N_2 - n/N_1) = c. \qquad (3.23)$$

For the quadratic DF, however, the sample covariance matrices are assumed to be *different* in both pattern classes. Therefore term $T_{\Sigma\mathrm{B}}$ in Equation (3.19) appears. Consequently, the bias is much more significant than for the linear classification rule. For the quadratic DF, we obtain the following rather unexpected effect: an increase in the training-set size of one of the pattern classes can increase the expected classification (generalisation) error!

Example 5. For two 40-dimensional GCCM classes, δ=2.56 ($\varepsilon_{\infty}^{\mathrm{Q}} = \varepsilon_B$ = 0.1), the expected error of the standard quadratic DF is

for $N_1 = N_2 = 200$ $\quad \bar{\varepsilon}_N^{\mathrm{Q}} \approx 0.229$,
for N_1=200, N_2 =2000 $\quad \bar{\varepsilon}_N^{\mathrm{Q}} \approx 0.265$, and
for N_1=200, N_2 =20000 $\quad \bar{\varepsilon}_N^{\mathrm{Q}} \approx 0.274$.

We see, an increase in N_2, the number of training vectors from category ω_2, notably increases the generalisation error!

Use of an unbiased estimate of the multivariate Gaussian density or the Bayes predictive approach does not help to reduce the generalisation error. This effect is due to the assumption that the prior distributions of the mean vectors and the covariance matrices are flat. In obtaining the Bayes predictive decision rule we may be averaging over an unduly dispersed prior parameter distribution.

An introduction of the bias-correcting term to the classical quadratic DF helps to obtain an unbiased quadratic DF that reduces the generalisation error. We suggest the following unbiased quadratic DF:

$$\hat{g}^{\mathrm{Q}}_{unbiased}(\boldsymbol{X}) = k_2(\boldsymbol{X} - \hat{\boldsymbol{M}}_2)^T \hat{\Sigma}_2^{-1}(\boldsymbol{X} - \hat{\boldsymbol{M}}_2) - k_1(\boldsymbol{X} - \hat{\boldsymbol{M}}_1)^T \hat{\Sigma}_1^{-1}(\boldsymbol{X} - \hat{\boldsymbol{M}}_1) +$$

$$\log|\hat{\Sigma}_2|/|\hat{\Sigma}_1| + \log P_1/P_2 + bias^{\mathrm{Q}}(n, N_1, N_2), \qquad (3.24)$$

where

$k_i = 1 - n/N_i$;

$bias^{\mathrm{Q}}(n, N_1, N_2) = \sum_{j=1}^{n}(\Psi((N_2 - j)/2) - \Psi((N_1 - j)/2) + n\log(N_2/N_1)$;

$\Psi(r)$ is an Euler psi function: $\Psi(r+1) = -\mathbf{C} + \sum_{s=1}^{r} 1/s$;

$\Psi(r+1/2) = -\mathbf{C}-2\log 2+ 2\sum_{s=0}^{r-1}(1-2s)^{-1}$, and $\mathbf{C}$ is a constant.

For the new “unbiased” quadratic DF we calculate:

$$\bar{\varepsilon}_N^{Q} \approx 0.229, \text{ when } N_1 = N_2 = 200;$$
$$\bar{\varepsilon}_N^{Q} \approx 0.187, \text{ when } N_1=200, N_2 = 2000; \text{ and}$$
$$\bar{\varepsilon}_N^{Q} \approx 0.183, \text{ when } N_1=200, N_2 = 20000.$$

For N_2 = 20000, the mean generalisation error diminished from $\bar{\varepsilon}_N^{Q} \approx 0.274$ to 0.183. For the unequal CM case ($\Sigma_2 = 2\,\Sigma_1$) ε_∞^{Q} =0.34, n = 40, N_1=200. An increase in N_2 from 200 to 20000 diminishes the mean generalisation error from $\bar{\varepsilon}_N^{Q} \approx 0.170$ to 0.084.

These examples help one to understand the deficiency of the plug-in approach. Thus, one should diminish the bias of the quadratic DF. While training the SLP classifier we attempt to minimise the empirical probability of misclassification and do not suffer such a marked effect due to bias of the classifier. It is an advantage of the neural net approach.

3.6 Minimum Empirical Error and Maximal Margin Classifiers

There are a number of classification design algorithms that yield the minimum empirical error classifiers. While training the SLP classifier with an increase in the weights’ magnitude, the SLP approaches the minimum empirical error classifier. If the number of empirical errors is zero, then in further training, the non-linear single-layer perceptron goes towards the maximal margin classifier.

In the nonparametric (structural) approach, no assumptions on the type of the multivariate distribution density are made. This is the main difference between the nonparametric (structural) minimum empirical error classifier and the parametric linear statistical classifiers. The parameters of the linear MEE classifier are searched directly from the training vectors. We avoid estimation of a great number of “unnecessary” parameters of the multivariate distribution density function. In this section, we consider consequences of this difference by analysing the generalisation error.

3.6.1 Favourable Distributions of the Pattern Classes

We have observed cardinal differences in values of the generalisation error of the Euclidean distance classifier calculated for the favourable and unfavourable distributions of the pattern classes. A similar situation arises in the analysis of the minimum empirical error classifier. We show that in the *favourable case*, one vector per class can be sufficient to train the classifier.

Example 6. Let the pattern vectors have support on a straight line in the multivariate feature space. Let the vectors from class 1 have support on the interval (A_1 A_2) on this line, and the vectors from class 2 have support on the interval (B_1 B_2). The support of each class does not overlap. We have chosen that $|A_1, A_2| < |A_2, B_1|$ and $|B_1, B_2| < |A_2, B_1|$ (Figure 3.6).

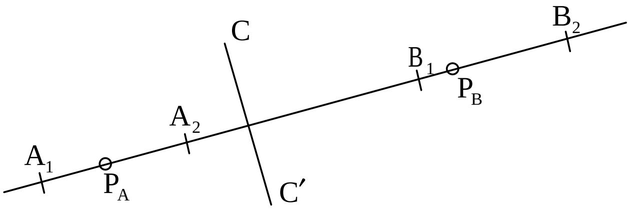

Fig. 3.6. The "most favourable" distribution of two pattern classes in two-variate feature space (Reprinted from *Neural Networks*, 11:297-313, Raudys, Evolution and generalization of a single neurone, 1998, with permission from Elsevier Science).

Suppose now, that only one observation per class is available for training – P_A and P_B. Let us design a linear classifier with maximal margin between the discriminant hyperplane and the two training vectors. Obviously, the linear decision boundary C-C', $\boldsymbol{V}^T\boldsymbol{X} + v_0 = 0$, crosses the interval ($A_2$, B_1). For this pattern class model the generalisation error will be zero. This example is one of the *most favourable* distributions of the pattern classes and is very favourable for the Euclidean distance classifier too. The pattern-class configuration is similar to the pattern-class configuration of the classes ω_3 and ω_5, depicted in Figure 3.2.

In *unfavourable* pattern-class distribution cases, in order to obtain a low generalisation error one needs many more training examples. For the linear Fisher classifier one of the unfavourable cases was presented in Figure 3.3; for the EDC it was presented in Figure 3.2 (the classes ω_3 and ω_4). The model of the pattern classes ω_3 and ω_4 can also become difficult to train for the minimum empirical error classifier.

3.6.2 VC Bounds for the Conditional Generalisation Error

The absolute and the relative increases in the generalisation error depend on the data model and its parameters **Y**. Here we have a dilemma: in order to be able to determine the increase in the generalisation error "one must know almost everything” (Vapnik, 1995). This requirement is an important deficiency of the statistical analysis approach. To overcome this difficulty Vapnik and Chervonenkis, (1974) devised a new viewpoint to analyse the classification, prediction and function approximation algorithms based on an empirical risk minimisation. Instead of characterising the *complexity of the data,* Vapnik and Chervonenkis (1968) searched for a characteristic of the *complexity of a set of classifiers.* Following the Cover’s capacity concept, they succeeded in formulating a single complexity characteristic for a fraction of algorithms that minimise the empirical error that later became known as the Vapnik – Chervonenkis (VC) dimension

The VC dimension of a set F of classifiers is the maximum number of vectors h that can be divided into two classes in all 2^h possible ways using any classifier from the set F. For *linear classifiers* in an n-dimensional feature space $h = n + 1$. For multilayer perceptrons, the VCD is bounded by a number of weights. However, an exact expression does not exist yet.

For linear classifiers that minimise the empirical classification error it has been shown that the following inequality holds with probability at least 1-η (Vapnik, 1982, Equation 11.2)

$$\varepsilon_N^{\mathrm{MEE}} \le \varepsilon_{bound}^{\mathrm{MEE}} = \hat{\varepsilon}_N^{\mathrm{MEE}} + \frac{n(\ln\frac{N}{n}+1)-\log\eta}{2N}(1+\sqrt{1+\frac{4N\ \hat{\varepsilon}_N^{\mathrm{MEE}}}{n(\ln\frac{N}{n}+1)-\log\eta}})\ , \quad (3.25a)$$

where:

$\varepsilon_N^{\mathrm{MEE}}$ is the conditional probability of misclassification (the generalisation error) of the minimum empirical error (MEE) classifier;

$\hat{\varepsilon}_N^{\mathrm{MEE}} = N_{errors} / N$ is the empirical probability of misclassification (apparent error);

N_{errors} is the number of errors in the training-set and

$N = N_1 + N_2$ is the number of training examples.

The estimate (3.25*a*) is obtained for the worst possible type of pattern-class distributions. The last term on the right side of Expression (3.25*a*) gives an upper bound for the difference between the empirical error $\hat{\varepsilon}_N^{\mathrm{MEE}}$ and the conditional probability of misclassification $\varepsilon_N^{\mathrm{MEE}}$. Thus, for any other pattern-class distributions we should have a smaller difference between $\varepsilon_N^{\mathrm{MEE}}$ and $\hat{\varepsilon}_N^{\mathrm{MEE}}$. The bound (3.25*a*) indicates that the increase in the generalisation error is proportional to *n/N,* the dimensionality – training-set size ratio. This bound also indicates that, in theory, very “problematic” distributions of the pattern classes can occur. For such unfavourable cases, the number of training samples required to design the classifier is very large.

Example 7. For dimensionality $n = 100$, $\eta = 0.5$, and zero empirical error ($\hat{\varepsilon}_N^{\mathrm{MEE}} = 0$), the bound is

for $N = 200$ $\varepsilon_{bound}^{\mathrm{MEE}} = 0.85$; for $N = 2{,}000$ $\varepsilon_{bound}^{\mathrm{MEE}} = 0.2$;

and only for very large training-sets is the bound is comparatively reasonable: e.g.

$$\text{for } N = 20{,}000 \qquad \varepsilon_{bound}^{\mathrm{MEE}} = \;\; 0.03.$$

The bound (3.25*a*) is derived for the worst-case distributions. Therefore, a number of modifications to this bound has been suggested. E.g., Cherkassky and Mullier (1998, Section 4.3.1) propose the bound

$$\varepsilon_{bound}^{\mathrm{MEE}} = \frac{a_1 n(\log a_2 \frac{N}{n} + 1) - \ln \eta}{2N}\left(1 + \sqrt{1 + \frac{4N\hat{\varepsilon}_N^{\mathrm{MEE}}}{a_1 n(\log a_2 \frac{N}{n} + 1) - \ln \eta}}\right), \qquad (3.25b)$$

where constants a_1, a_2 must be in the range $0 < a_1 \leq 4$, $0 < a_2 \leq 2$.

Unfortunately, “for classification problems, good empirical values for a_1 and a_2 are unknown”. The bounds become tighter when the training-set size N is large. Both bounds mentioned agree with conclusions from asymptotic analysis of the parametric rules: the increase in the generalisation error depends on the ratio of dimensionality to sample size, n/N. For more details concerning these error bounds the reader is directed to books by Vapnik (1982, 1995), Vidyasagar (1997), Cherkassky and Mulier (1998).

The Vapnik and Chervonenkis approach allows one to obtain a number of other error bounds for the conditional generalisation error of the classifiers within this group and to suggest a strategy for the selection of a classifier with optimal complexity. According to this strategy, called the *structural risk minimisation*,

one needs to select a rule with the minimal estimate of the error bound.

The structural risk minimisation approach is very simple to use as it does not require one to know the number of unknown characteristics of the data that are difficult or even impossible to estimate. Therefore, this approach has become very popular and has played a great role in explaining finite design set effects both in statistical pattern recognition and ANN theory. Unfortunately characterisation of the classifier’s complexity by one single characteristic requires one to analyse the “worst-case functions”. Then the error bounds “become too loose to be practically useful”.

“These bounds become more tight (accurate) only when $N \to \infty$, an empirical risk is very close to the true risk” (Cherkassky and Mulier, 1998, Chapter 4). In comparison with the mean generalisation error values for the Euclidean distance, Fisher and zero empirical error classifiers reported above, the error bounds are incomparably large. In order to have the luxury of ignoring everything about the data except the number of dimensions, *n*, we have to sacrifice accuracy of the estimates. Thus, “we cannot expect this approach to provide immediate solutions to practical problem sets” (Cherkassky and Mulier, 1998).

3.6.3 Unfavourable Distributions for the Euclidean Distance and Minimum Empirical Error Classifiers

In spite of the fact that the error bounds (3.25*ab*) result in very pessimistic values, in principle, for the Euclidean distance classifier E we can obtain even higher generalisation error values. In Section 3.3, we have shown that for the EDC the effective dimensionality, n^*, is unbounded in principle. Thus, models of the pattern classes can be created where any number of the training vectors will be insufficient to design parametric classifiers, such as the Euclidean distance classifier, the Fisher linear DF, the regularised DA.

Example 8. For the unfavourable 100-variate negatively correlated data **E** (see Section 3.3.3) with $\varepsilon_\infty^{\mathrm{E}} = \varepsilon_\infty^{\mathrm{F}} = 0.03$ and $n^*=10^8$, when $N = 400$ ($\bar{N} = 200$) from (3.7) we obtain $\bar{\varepsilon}_N^{\mathrm{E}}$ =0.497. In ten experiments with the EDC and ten independently chosen random training sets, we obtained the conditional PMC 0.497 (the same value, 0.497, in all 10 experiments; see subsection 3.3.3 for more details).

When $\bar{N} = 1000$ ($N = 2000$) from Equation (3.7) we have $\bar{\varepsilon}_N^{\mathrm{E}}$ =0.494, and
when $\bar{N} = 10000$ ($N = 20000$) then $\bar{\varepsilon}_N^{\mathrm{E}}$ =0.480.

These values exceed the error bound (3.25*a*) for the least favourable case of the minimum empirical error classifier. Note for such highly correlated data models the EDC should not be used. Here the standard Fisher linear DF performs much better.

3.6.4 Generalisation Error in the Spherical Gaussian Case

In the most favourable case, one training vector per class can be sufficient to design the maximum margin classifier. From (3.25*a*) it follows that, in unfavourable cases, we need many training vectors in order to obtain the reliable classifier. In practice, we never have the most favourable nor the most unfavourable distributions for the pattern classes. In Table 3.2, we have presented experimental evaluations of the learning quantity $\kappa = \bar{\varepsilon}_N^{\mathrm{A}} / \varepsilon_\infty^{\mathrm{A}}$ of the Euclidean distance classifier and the standard Fisher linear DF obtained on four real-world data sets. The differences between the experimental results and the theoretical estimates obtained for the spherical Gaussian model are not large. Thus, in order to have some rule of thumb we will analyse the generalisation errors for the spherical Gaussian model.

We analyse one particular example of the minimum empirical error classifier – a *zero empirical error classifier* ($\hat{\varepsilon}_N^{\mathrm{MEE}} = 0$) – in the bench-mark model case, i.e. when the true pattern classes are spherical Gaussian $N_X(\boldsymbol{M}_1, \mathbf{I})$, $N_X(\boldsymbol{M}_2, \mathbf{I})$ and the prior probabilities of the classes are equal to ½. We will find the expected generalisation error of the zero empirical error (ZEE) linear classifier and the margin classifier trained by a hypothetical “random search” optimisation procedure.

The random search *(Monte Carlo) optimisation procedure* generates many (say, M) random discriminant hyperplanes according to a certain *prior* distribution of the weights, determined by *a priori* density $p^{prior}(\boldsymbol{V}, v_0)$, and selects those that classify the training sets LS^1 and LS^2, each of size $\bar{N}$, without error, as well as satisfying the following set of $N = 2\bar{N}$ conditions, S_Δ:

$$\text{for all } \bar{N} \text{ vectors } \boldsymbol{X}_{1j} \text{ from } LS^1, \qquad \frac{\boldsymbol{V}^T\boldsymbol{X}_{1j} + v_0}{\sqrt{\boldsymbol{V}^T\boldsymbol{V}}} > \Delta, \tag{3.26a}$$

$$\text{for all } \bar{N} \text{ vectors } \boldsymbol{X}_{2j} \text{ from } LS^2, \qquad \frac{\boldsymbol{V}^T\boldsymbol{X}_{2j} + v_0}{\sqrt{\boldsymbol{V}^T\boldsymbol{V}}} < -\Delta, \tag{3.26b}$$

where Δ is a bound for the margin.

The optimisation theory approach advocates that if the prior distribution $p^{prior}(\boldsymbol{V}, v_0)$ incorporates the optimal weight, then with an increase in the number of repetitions M of the random search, this procedure will find the minimum of the cost function with a desired accuracy. Thus, in the case that the number M tends to infinity and the training sets of opposite classes are linearly separable, then, at least theoretically, we should always succeed in finding one or more solutions that satisfy conditions (3.26*ab*). Note that the above random search training procedure will never be used in practice. The gradient training algorithms used in ANN design as well as many other algorithms mentioned in Chapter 2 allow one to estimate the weights $v_0, v_1, \dots, v_n$ much quicker and more effectively. For the random search optimisation, however, one can obtain analytical results that lead to quantitative results. The analytical expressions can serve as estimates for the expected classification errors of the linear classifiers trained by more sophisticated optimisation techniques.

We will find the *mean expected probability of misclassification* for pattern vectors that did not participate in training. This expectation is taken *both* with respect to the N random training vectors, and with respect to random character of the $(n+1)$ - variate weight vector determined by the priori density $p^{prior}(\boldsymbol{V}, v_0)$.

At first we consider the prior density $p^{prior}(\boldsymbol{V}, v_0)$ of the $(n+1)$-variate weight vector $(\boldsymbol{V}, v_0)^T$ to be spherical Gaussian $N_V(\boldsymbol{0}, \mathbf{I})$. Assume $\boldsymbol{M}_1 = -\boldsymbol{M}_2$. In this case, only vague *a priori* information on the weights $v_0, v_1, \dots, v_n$ is used to design the classification rule. To derive the formula for the expected probability of misclassification $\bar{\varepsilon}_N^{\text{ZEE}}$ we use the predictive Bayes approach discussed in Chapter 2:

$$\bar{\varepsilon}_N^{\text{ZEE}} = \int\int P(error|\boldsymbol{V}, v_0)\, p^{\,apost}(\boldsymbol{V}, v_0)\, d\,\boldsymbol{V}\, d\,v_0, \tag{3.27}$$

where:

$P(error|\boldsymbol{V}, v_0)$ is a conditional probability of misclassification, given $(\boldsymbol{V}^T, v_0)^T$;

$P(error|\boldsymbol{V}, v_0) = \frac{1}{2}\, Prob\{\boldsymbol{V}^T\boldsymbol{X}+ v_0 < 0\ |\boldsymbol{X}\in\ \omega_1\} + \frac{1}{2}\, Prob\{\boldsymbol{V}^T\boldsymbol{X}+ v_0 \geq 0\ |\boldsymbol{X}\in\ \omega_2\} =$

$$\tfrac{1}{2}\,\Phi\{-\frac{\boldsymbol{V}^T\boldsymbol{M}_1+v_0}{\sqrt{\boldsymbol{V}^T\boldsymbol{V}}}\}+\tfrac{1}{2}\,\Phi\{\frac{\boldsymbol{V}^T\boldsymbol{M}_2+v_0}{\sqrt{\boldsymbol{V}^T\boldsymbol{V}}}\}; \tag{3.28}$$

$p^{\,apost}(\boldsymbol{V}, v_0)$ is *a posteriori* probability density function of $(\boldsymbol{V}^T, v_0)^T$;

$$p^{\,apost}(\boldsymbol{V}, v_0) = \frac{P(\mathrm{S}_\Delta, \boldsymbol{V}, v_0)}{P(\mathrm{S}_\Delta)} \propto P(\mathrm{S}_\Delta \mid \boldsymbol{V}, v_0)\, p^{\,prior}(\boldsymbol{V}, v_0); \tag{3.29}$$

and $P(\mathrm{S}_\Delta \mid \boldsymbol{V}, v_0)$ is the conditional probability of the N events S_Δ.

The conditional probability of event (3.26*a*) is $1- \Phi\{-\frac{\boldsymbol{V}^T\boldsymbol{M}_1+v_0}{\sqrt{\boldsymbol{V}^T\boldsymbol{V}}} + \Delta\}$. Thus, for N independent training vectors, we can write

$$P(\,\mathrm{S}_\Delta \mid V, v_0) = [1- \Phi\{-\frac{\boldsymbol{V}^T\boldsymbol{M}_1+v_0}{\sqrt{\boldsymbol{V}^T\boldsymbol{V}}} + \Delta\}]^{\bar{N}}\ [1- \Phi\{\frac{\boldsymbol{V}^T\boldsymbol{M}_2+v_0}{\sqrt{\boldsymbol{V}^T\boldsymbol{V}}} + \Delta\}]^{\bar{N}}\ . \tag{3.30}$$

Probabilities (3.28) and (3.30) are conditioned on the vector $(\boldsymbol{V}^T, v_0)^T$. Now we will show that, provided the distributions of vectors $\boldsymbol{X}$ and $(\boldsymbol{V}^T, v_0)^T$ are spherical, these probabilities depend on only two independent scalar random variables. Let $\mathbf{T}$ be an $n \times n$ orthonormal matrix with the first row vector

$$\boldsymbol{t}_1 = \delta^{-1}\boldsymbol{M}^T,$$

where $\boldsymbol{M} = \boldsymbol{M}_1 - \boldsymbol{M}_2$ and $\delta^2 = \boldsymbol{M}^T\boldsymbol{M}$ is the squared Mahalanobis distance. Then

$\mathbf{T}\,\boldsymbol{M} = (\delta, 0, 0,..., 0)^T$,

$$\frac{\boldsymbol{V}^T\boldsymbol{M}_1+v_0}{\sqrt{\boldsymbol{V}^T\boldsymbol{V}}} = \frac{(\mathbf{T}\boldsymbol{V})^T(\mathbf{T}\boldsymbol{M}\frac{1}{2}+\mathbf{T}(\boldsymbol{M}_1+\boldsymbol{M}_2)\frac{1}{2})+v_0}{\sqrt{(\mathbf{T}\boldsymbol{V})^T\mathbf{T}\boldsymbol{V}}} = \frac{\vartheta_1\delta/2+v_0}{\sqrt{\vartheta_1^2+\ \sum_{i=2}^n\vartheta_i^2}}\,,$$

$$\frac{\boldsymbol{V}^T\boldsymbol{M}_2+v_0}{\sqrt{\boldsymbol{V}^T\boldsymbol{V}}} = -\,\frac{\vartheta_1\delta/2-v_0}{\sqrt{\vartheta_1^2+\ \sum_{i=2}^n\vartheta_i^2}}\,,$$

where $\boldsymbol{V}^T\mathbf{T}^T = (\,\vartheta_1\,, \dots,\ \vartheta_n\,)$ and we used the assumption

In order to obtain a generalisation error expression suitable for numerical evaluation, we must define the prior distribution of the weights.

The spherical Gaussian prior distribution of the weights. From this assumption it follows that the Gaussian N(0,1) random variables ϑ_1, v_0 and the chi-squared random variable $\Sigma_{i=2}^{n} \vartheta_i^2 = \chi_{n-1}^2$ are mutually independent. Then

$$\frac{\boldsymbol{V}^T\boldsymbol{M}_1 + v_0}{\sqrt{\boldsymbol{V}^T\boldsymbol{V}}} = \frac{\vartheta_1\delta/2 + v_0}{\sqrt{\vartheta_1^2 + \ \Sigma_{i=2}^{n}\vartheta_i^2}} = \nu\frac{\delta}{2} + \omega\,,$$

where $\nu = \dfrac{\vartheta_1}{\sqrt{\vartheta_1^2 + \chi_{n-1}^2}}$ and $\omega = \dfrac{v_0}{\sqrt{\vartheta_1^2 + \chi_{n-1}^2}}$.

By using standard statistical transformation theory one can readily show that ν and ω are mutually independent: ν is Beta and ω is Student random variables:

$$p(\nu,\,\omega) = \frac{n-1}{2\pi}\,\frac{(1-\nu^2)^{(n\,-\,3)\,/\,2}}{(1+\omega^2)^{(n\,+\,1)\,/\,2}} \equiv p^{\,prior}(\boldsymbol{V},\,v_0).$$

Consequently, the conditional probabilities (3.28) and (3.30) can be represented as functions of two independent scalar random variables:

$$P(error \mid v,\,\omega,\,\delta) = P(error \mid \nu,\,\omega) = ½\;\Phi\{-\,\nu\delta/2 + \omega\,\} + ½\;\Phi\{-\,\nu\delta/2 - \omega\,\}, \quad (3.31)$$

$$P(\,\mathrm{S}_\Delta \mid v,\,\omega,\;\delta,\,\Delta) = P(\mathrm{S}_\Delta \mid \nu,\,\omega) =$$

$$[1\text{-}\Phi\{-\,\nu\delta/2 + \omega + \Delta\}]^{\overline{N}}\;\;[1\text{-}\Phi\{-\,\nu\delta/2 - \omega\; + \Delta\}]^{\overline{N}}\;. \qquad (3.32)$$

We eventually obtain a final expression for the expected generalisation error

$$\bar{\varepsilon}_N^{\mathrm{ZEE}} \;=\; \int\int P(error \mid v,\,\omega,\;\delta)\,p^{\,apost}(v,\,\omega|\,\delta,\,\Delta)\,d\,\nu\,d\,\omega =$$

$$\int\int P(error|\{v,\,\omega,\;\delta)\;P(\mathrm{S}_\Delta \mid v,\,\omega,\;\delta,\,\Delta)\,p^{\,prior}(\nu,\,\omega)\,d\,\nu\,d\,\omega. \quad (3.33)$$

Representation (3.33) can be used to calculate the mean expected error. For fixed Mahalanobis distance δ the deviation of $\bar{\varepsilon}_N^{\mathrm{ZEE}}$ from the Bayes error ε_B depends mainly on $\overline{N}/n$, the training set size / dimensionality ratio. Here we have a similarity with parametric classification. In Table 3.3, for $n = 50$, five Mahalanobis distances, the boundary for the margin $\Delta = 0$ and different $\overline{N}$ values we present values of the learning quantity $\kappa = \bar{\varepsilon}_N^{\mathrm{ZEE}}/\varepsilon_B$ (left columns: ZEE with random priors). For comparison we present some κ values for the standard linear Fisher classifier (from Table 3.1).

Table 3.3. Values for the ratio $\kappa = \bar{\varepsilon}_N^{\mathrm{ZEE}} / \varepsilon_B$ for the ZEE classifier (with random and “Euclidean” prior weights) versus $\bar{N}$, the training-set size, for dimensionality n = 50 and five values of the distance δ along with the Bayes error ε_B (Reprinted from Raudys, On dimensionality, sample size, classification error of nonparametric linear classification algorithms, *Pattern Analysis and Machine Intelligence* 19, 1997, © IEEE).

ZEE with random priors					ZEE with Euclidean priors					Fisher LDF					$\bar{N}$
2.15	3.75	9.99	25.0	70.9	1.63	1.99	2.70	3.47	4.42						8
2.06	3.43	8.56	20.6	56.7	1.48	1.69	2.12	2.57	3.08						12
1.90	2.95	6.83	15.7	41.6	1.29	1.40	1.66	1.91	2.16						20
1.72	2.57	5.62	12.4	32.0	1.17	1.25	1.43	1.60	1.77	2.05	3.39	8.40	19.7	52.0	30
1.56	2.16	4.34	9.13	22.5	1.08	1.13	1.26	1.37	1.48	1.62	2.15	3.61	5.95	10.6	50
1.35	1.73	3.09	6.04	14.1	1.03	1.06	1.13	1.21	1.27	1.33	1.51	1.93	2.47	3.27	100
1.16	1.32	2.06	3.59	7.68	1.01	1.02	1.07	1.10	1.14	1.14	1.19	1.31	1.44	1.61	250
1.68	2.56	3.76	4.65	5.50	1.68	2.56	3.76	4.65	5.50	1.68	2.56	3.76	4.65	5.50	δ
0.2	0.1	0.03	0.01	.003	0.2	0.1	0.03	0.01	.003	0.2	0.1	0.03	0.01	.003	ε_B

Our main concern is to understand small training-set peculiarities that arise while training the perceptrons in a neural net classifier. The assumption of a random Gaussian prior weight vector describes an unrealistic situation for the perceptron training. This model gives *pessimistic estimates* of the expected generalisation error. To obtain *optimistic estimates* we assume that additional prior information is available in order to generate the prior weights in an alternative way.

Second choice of prior distribution. This choice is motivated by a fact already discussed in the Section 1.3: if the conditions E1 − E4 are satisfied, then the first iteration of the back propagation training of the SLP produces the weight vector of the EDC with $v_0^{\mathrm{E}} = 0$ and $\boldsymbol{V}^{\mathrm{E}} = \eta\ (\hat{\boldsymbol{M}}_1 - \hat{\boldsymbol{M}}_2)$. In the EDC determination of v_0^{E} and $\boldsymbol{V}^{\mathrm{E}}$, we use the first statistical moments of the training sets LS^1 and LS^2 , while in order to obtain the ZEE classifier we use the highest order statistical moments. Thus, we may guess that the starting (after the first iteration) and final (after the ZEE classifier is obtained) perceptron weights are correlated weakly.

Thus, it is reasonable to assume prior distribution $p^{prior}(\boldsymbol{V}, v_0)$ to be determined by the weights of EDC. In addition to the authentic training sets LS^1 and LS^2, let there exist an infinite number of extra training sets LS_1, LS_2, LS_3, ... each composed from N=2$\bar{N}$ vectors. From each set, we estimate the mean values $\hat{\boldsymbol{M}}_1$, $\hat{\boldsymbol{M}}_2$ and calculate the weights, $\boldsymbol{V}^{\mathrm{E}}$ and v_0^{E}. The distribution of such vectors implies a prior distribution $p^{prior}(\boldsymbol{V}, v_0)$. The additional training sets LS_1, LS_2, ... provide a considerable amount of prior information to determine the classifier’s weights. One may guess that this prior weight’s generation model leads to optimistically biased error estimates.

The difference between the error rates calculated for the optimistic and pessimistic prior distributions for the perceptron weights is clearly seen in Table 3.3. We emphasise that the ZEE classifier can be used even when the dimensionality *n* exceeds the number of training examples *N*. This conclusion is valid both for the optimistic and the pessimistic evaluations. In Chapter 4, we will see the generalisation error of the non-linear SLP classifier depends on the weights initialisation and generally lies between the optimistic and pessimistic estimates just considered.

3.6.5 Intrinsic Dimensionality

Consider the GCCM data model $N_X(\boldsymbol{M}_1, \bar{\Sigma})$, $N_X(\boldsymbol{M}_2, \bar{\Sigma})$ with intrinsic dimensionality $r < n$. We have already considered this data model in Section 3.3.4. For this data model

$$Prob\{\ \boldsymbol{V}^T\boldsymbol{X}+ v_0< 0 \mid \boldsymbol{X}\in \omega_1\} = \Phi\{-\frac{\boldsymbol{V}^T\boldsymbol{M}_1 + v_0}{\sqrt{\boldsymbol{V}^T\bar{\Sigma}\boldsymbol{V}}}\} =$$

$$= \Phi\{-\frac{\boldsymbol{V}^T\Phi\Phi^T\boldsymbol{M}_1 + v_0}{\sqrt{\boldsymbol{V}^T\Phi\Phi^T\bar{\Sigma}\Phi\Phi^T\boldsymbol{V}}}\} = \Phi\{-\frac{\boldsymbol{v}_1{}^T\boldsymbol{m}/2 + v_0}{\sqrt{\boldsymbol{v}_1{}^T\boldsymbol{v}_1}}\},$$

$$Prob\{\ \boldsymbol{V}^T\boldsymbol{X}+ v_0 \geq 0 \mid \boldsymbol{X}\in \omega_2\} = \Phi\{-\frac{\boldsymbol{v}_1{}^T\boldsymbol{m}/2 - v_0}{\sqrt{\boldsymbol{v}_1{}^T\boldsymbol{v}_1}}\}, \tag{3.34}$$

where $\boldsymbol{v}_1 = \phi_1\boldsymbol{V}$ is *r*-variate subvector, a part of vector $\boldsymbol{\Phi}^T\boldsymbol{V}=\begin{bmatrix}\boldsymbol{v}_1\\ \boldsymbol{v}_2\end{bmatrix}$, and $\boldsymbol{\Phi}^T = \begin{bmatrix}\phi_1\\ \phi_2\end{bmatrix}$.

The above representations indicate that for this almost singular data model with intrinsic dimensionality equal to *r*, the small training-set properties of the zero empirical error classifier can be analysed in an *r*-variate space. In this space, the *r*-variate vector $\boldsymbol{X}_r = \phi_1\boldsymbol{X}$ is Gaussian with distribution $N_X(\boldsymbol{m}/2, \mathrm{I}_r)$, or $N_X(-\boldsymbol{m}/2, \mathrm{I}_r)$. Thus, we can use the formulae derived for the multivariate spherical Gaussian data model, as well as Table 3.3.

3.6.6 The Influence of the Margin

We have the margin classifier in the random search (Monte Carlo) optimisation procedure (3.26*ab*), when the bound for the margin $\Delta > 0$. The representations (3.32), (3.33) in (3.28) and (3.30) can be used to calculate the mean expected generalisation error for each particular Δ value. Numerical calculations indicate that an increase in Δ, on average diminishes the mean expected generalisation error $\bar{\varepsilon}_N^{\mathrm{ZEE}\ \&\Delta}$. In practice we observe overtraining. This fact suggests that the classification performance is conditioned by the particular training-set. The training sets that lead to larger margins, on average, result in a smaller generalisation error.

This conclusion agrees with an error bound derived by Vapnik (1976). In cases where there is an empty zone between the training sets ($\hat{\varepsilon}_N^{\text{MEE}} = 0$), for the maximal margin classifier (MMC) instead of bound (3.25*a*) one can use a tighter bound obtained for a schema of the empirical risk minimisation calculated for an unlabeled validation set vectors of size *N*

$$\varepsilon_N^{\text{MC}} \leq \frac{d(\ln\frac{2N}{d}+1)-\log\eta}{N},$$

where *d* is the smaller value of two quantities, n and $\frac{H^2}{\rho^2}+1$,

H is a diameter of labeled training and unlabeled validation sets,

ρ is a minimal distance between vectors of opposite classes in training and validation sets.

The minimal distance ρ matches up with the margin width and indicates that the bound for the generalization error diminishes as the margin Δ increases. While contemplating the values of $\bar{\varepsilon}_N^{\text{ZEE}\,\&\Delta}$, analytically calculated for different Δ, one needs to remember the following fact. According to the random search “training” conditions (3.26*ab*), the mean generalisation error $\bar{\varepsilon}_N^{\text{ZEE}\,\&\Delta}$ is an average value calculated for the training sets of the same size and for a number of weight vectors weights generated according to density $p^{prior}(\boldsymbol{V}, v_0)$. Thus, this analytical approach does not allow one to investigate the influence of the margin’s width on the generalisation performance for one particular training-set and one particular training session. This approach allows analysis only of the *average behaviou*r defined by $p^{prior}(\boldsymbol{V}, v_0)$. Analogous conclusion is valid for the bound $\varepsilon_N^{\text{MM}}$.

Example 9. The SLP was used to demonstrate the influence of the margin width on the conditional PMC. In this example, we had one particular linearly separable training-set. Spherical Gaussian data was used. In Figure 3.7 we have three graphs that are typical for the Gaussian patterns: the conditional classification error $\varepsilon_N^{\text{SLP}}$ versus the margin width M for three different initialisations. In all three training experiments, we translated the data centre $\hat{\boldsymbol{M}} = ½\,(\hat{\boldsymbol{M}}_1 + \hat{\boldsymbol{M}}_2)$ into the zero point. We used the sigmoid activation function and trained the perceptron in batch mode with the standard back-propagation algorithm using the targets 1 and 0. In order to obtain a large margin quickly we increased the learning step η progressively with each iteration number *t*, η=0.0005*1.1^t. The progressive increase of the learning step prevents the gradient of the SLP cost function from converging to zero and allows one to obtain the full range of statistical classifiers during the perceptron training process (we will have more details in Chapter 4).

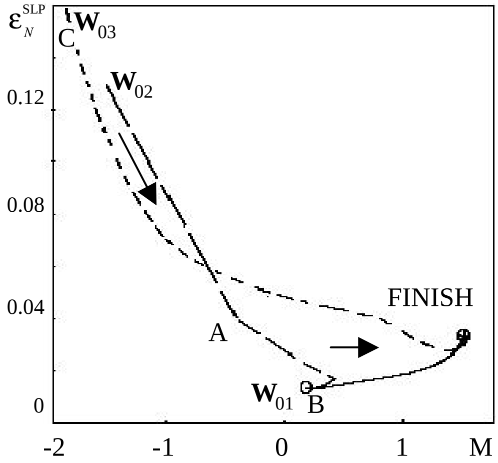

Fig. 3.7. The generalisation error $\varepsilon_N^{\mathrm{SLP}}$ versus margin M: $\mathbf{W}_{01}\rightarrow$ FINISH – zero initialisation; $\mathbf{W}_{03}\rightarrow$ C $\rightarrow$ FINISH – random N(0, 0.4) initialisation; $\mathbf{W}_{02}\rightarrow$ A $\rightarrow$ B $\rightarrow$ FINISH – an intermediate case (Copy from Raudys, On dimensionality, sample size, classification error of nonparametric linear classification algorithms, *Pattern Analysis and Machine Intelligence* 19, 1997, © IEEE).

When $\boldsymbol{V}_{(t=0)} = \mathbf{0}$ and $N_2 = N_1$, the first training iteration yields the EDC, a rule which is an optimal sample-based classifier for the spherical Gaussian pattern vectors (see Section 2.4.2). Therefore, further training can only increase the generalisation error (curve $\mathbf{W}_{01}\rightarrow$ FINISH in Figure 3.7). When the initial weights vary widely, the SLP uses either no or vague prior information. In this case, the training process gradually reduces the generalisation error; see learning curve $\mathbf{W}_{03}\rightarrow$ C $\rightarrow$ FINISH which corresponds to the initial weight vector generated according to $N_V(\mathbf{0},\ \sigma_{in}^2\,\mathbf{I}_{n+1})$ with $\sigma_{in}^2 = 0.4$. The curve $\mathbf{W}_{03}\rightarrow$ C $\rightarrow$ FINISH in Figure 3.7 shows only 81 to 850 iterations. The information in the figure indicates that in the case of unsuccessful initialisation, an increase in the margin width decreases the generalisation error. We will return to the initialisation problem in Section 4.5.2.

The behaviour of SLP in the spherically Gaussian case agrees with the theoretical conclusion: the maximal margin classifier is different from the EDC. Therefore, the MMC is not the best classifier for the spherical Gaussian data model. On average, for such data any non-EDC decision rule will perform worse.

3.6.7 Characteristics of the Learning Curves

The universal learning curves (the decrease in the generalisation error $\bar{\varepsilon}_N^{\mathrm{A}}$ with an increase in the training-set size N) advocate that in the case when the classes (general populations, the densities $p_1(\boldsymbol{X})$ and $p_2(\boldsymbol{X})$) overlap, the increase in the generalisation error $\bar{\varepsilon}_N^{A}$ - ε_∞^{A} diminishes as 1/*N* with *N*. Unfortunately, the universal curves are obtained in the traditional asymptotic method only where *N* is increasing. Therefore, the universal curves say nothing about the behaviour of the generalisation error when *N* is comparatively small and the generalisation error $\bar{\varepsilon}_N^{A}$ notably exceeds its asymptotic value ε_∞^{A}. In fact, the latter situation is the most interesting one for practitioners. Exact and asymptotic generalisation error values of the ZEE, standard Fisher linear, quadratic, and the EDC classifiers confirm the universal behaviour of the learning curves when *N* is very large. The characteristics of the learning curves, however, are different in the case of small training sets.

In Table 3.4, we have the relative increase in the generalisation error $\kappa = \bar{\varepsilon}_N^{\mathrm{ZEE}} / \varepsilon_\infty^{\mathrm{ZEE}}$ as a function of $\bar{N}/n$, the ratio of the training-set size to dimensionality, calculated for 50-variate spherically Gaussian data model with $\delta = 3.76$ ($\varepsilon_B = 0.03$). In addition, for each three subsequent κ values, we calculated the order coefficient τ in an approximation $\kappa = 1 + \alpha_\tau \bar{N}^{-\tau}$.

Table 3.4. The relative increase in the generalisation error $\kappa = \bar{\varepsilon}_N^{\mathrm{ZEE}} / \varepsilon_\infty^{\mathrm{ZEE}}$ and the order parameter *r* versus $\bar{N}/n$, the ratio of training-set size to dimensionality.

N/n	.16	.24	0.4	0.6	1	2	5	10	20	50	100	200	500	1000
κ	10	8.58	6.77	5.58	4.34	3.09	2.06	1.59	1.34	1.15	1.077	1.039	1.016	1.008
τ		.05	0.3	0.4	0.5	0.6	0.6	0.7	0.7	0.8	0.9	1	1	

Table 3.4 indicates that for very small training-set sizes, the generalisation error $\bar{\varepsilon}_N^{\mathrm{ZEE}}$ decreases at a very small rate: $\tau = 0.1$. With an increase in the training-set size τ, the rate coefficient, increases until $\tau = ½$, and gradually approaches the asymptotic value $\tau = 1$ which follows from the traditional (only the sample size is increasing) asymptotics, VC bounds, computational learning theory, statistical mechanics and information–theoretic approaches.

Contrary to the results for the ZEE classifier, for very small training-set sizes the generalisation error of the Fisher classifier $\bar{\varepsilon}_N^{\mathrm{F}}$ decreases quickly: $\tau = 2 \div 3$ (Table 3.5). In the upper part of this table, we have κ and τ values for $\delta = 2.56$ ($\varepsilon_B = 0.1$); in the lower part, $\delta = 3.76$ ($\varepsilon_B = 0.03$).

Table 3.5. The relative increase in the generalisation error $\kappa = \bar{\varepsilon}_N^{\mathrm{F}} / \varepsilon_\infty^{\mathrm{F}}$ and the order parameter τ, versus the ratio of training-set size to dimensionality, *n/N*.

N/n	0.6	0.8	1	2	5	10	20	50	100	200	500	1000
κ	3.34	2.52	2.14	1.51	1.19	1.092	1.046	1.018	1.0091	1.0045	1.0018	1.0009
τ		2	1.4	1.2	1.1	1	1	1	1	1	1	
κ	8.15	4.80	3.55	1.93	1.31	1.145	1.074	1.028	1.014	1.0069	1.0027	1.0014
τ		3	2	1.6	1.3	1.2	1.1	1	1	1	1	

With an increase in N the rate coefficient τ diminishes and gradually approaches the “universal” value $\tau = 1$. We note that the characteristics of the learning curves also depend on the distance between the pattern classes δ. For the simple parametric classifier EDC and the ZEE classifier with the Euclidean initialisation, the expected error probability decreases at the rate n/N.

For fixed N/n, the ratio $\kappa = \bar{\hat{\varepsilon}}_N / \hat{\varepsilon}_\infty$ for both the parametric and nonparametric classifiers depend on the asymptotic error rate $\hat{\varepsilon}_\infty$: κ increases as $\hat{\varepsilon}_\infty$ decreases. In contrast, the difference $\Delta_{error} = \bar{\hat{\varepsilon}}_N - \hat{\varepsilon}_\infty$ increases with $\hat{\varepsilon}_\infty$. For Gaussian patterns, the ZEE classifier is preferable to the Fisher classifier when the dimensionality n is close to the training-set size N. The use of prior information about the weights makes the ZEE classifier preferable in almost all situations. Thus, we see that while training the SLP classifier in the proper way (when the conditions E1 − E4 are satisfied and we obtain EDC just after the first iteration), we may reasonably hope to obtain good small training-set properties for the perceptron.

3.7 Parzen Window Classifier

The well known Radial basis-functions (RBF) network has much in common with the Parzen window spherical-kernel based classifier. In this section, we define an intrinsic dimensionality concept. We also advocate that the intrinsic dimensionality, not a formal number of inputs n, plays the main role in determining the small training-set size properties of the PW and RBF classification rules.

3.7.1 The Decision Boundary of the PW Classifier with Spherical Kernels

When the training-set sizes N_1 and N_2 are increasing without bound and the smoothing parameter of the PW classifier (2.49) λ tends to zero, the decision boundary of the Parzen window classifier approaches that of the optimal Bayes classifier. It is a very favourable property of the Parzen window classifier. In order to profit from this property we need to collect and to store a very large training-set. Moreover, the computer time required to perform necessary multiplications and summations is also very large. In practice, we deal with training sets of finite length and, therefore, we can rarely utilise this favourable property of the Parzen window classifier.

The complexity of the Parzen window classifier depends on the value of the smoothing parameter λ. When $\lambda \to \infty$ the PW classifier (2.49) with $\boldsymbol{D} = \mathbf{I}$ produces the linear decision boundary, the hyperplane. This hyperplane is parallel to the discriminant hyperplane of the simplest statistical classifier − the Euclidean distance classifier. Equation (2.51) in Section 2.6.3 indicates that the distance between these two hyperplanes depends on tr($\hat{\Sigma}_2 - \hat{\Sigma}_1$), the trace of a difference of two sample CM, $\hat{\Sigma}_2 - \hat{\Sigma}_1$. We see that when the sample covariance matrices are different we will not obtain EDC. In the finite training-set case, the distance mentioned will fluctuate with $\hat{\Sigma}_1$ and $\hat{\Sigma}_2$. When $\lambda \rightarrow 0$, the PW classifier with Gaussian kernel approaches the *k-NN* rule with $k = 1$.

In Section 3.2, it was shown that the increase in the generalisation error of the EDC depends on sample size/dimensionality ratio (Equation (3.7), Table 3.1). Thus, one can expect that for spherical Gaussian pattern classes the PW classifier with large λ can be relatively insensitive to the training-set size. However, for pattern classes with high effective dimensionality n^* one can expect that the PW classifier with large λ will be sensitive to the training-set size.

Example 10. In Figure 3.8, we present a scatter diagram of 100 bi-variate Gaussian vectors from two pattern classes with highly correlated features. The means and the covariance matrix of this data:

$$\boldsymbol{M}_1 = -\boldsymbol{M}_2 = (0.1,\ 0.1)^T,\ \bar{\Sigma} = \begin{bmatrix} 1 & -0.99 \\ -0.99 & 1 \end{bmatrix}.$$

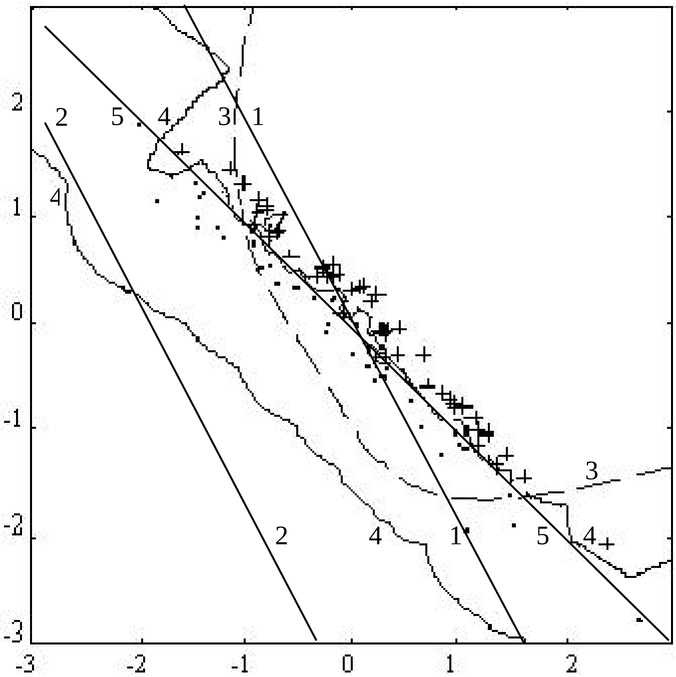

Fig. 3.8. An influence of the effective dimensionality and the smoothing parameter λ^2 on complexity of the Parzen window classifier: 1 – the decision boundary of the EDC, 2 – the PW classifier, λ= 1000, 3 – λ = 2, 4 – λ = 0.001, 5 – the Fisher linear DF.

For this data model the EDC is asymptotically optimal: the asymptotic error of the EDC and the Fisher classifier $\varepsilon_{\infty}^{\mathrm{F}} = \varepsilon_{\infty}^{\mathrm{E}} = 0.0786 = \varepsilon_B$ (the Mahalanobis distance δ=2.8284). The training-set size/dimensionality ratio is high: $N / n = 100/2 = 50$. Therefore, the Fisher linear classifier results in a rather exact decision boundary (line 5 in Figure 3.8). Contrary, for EDC the effective dimensionality $n^* = 39602$. Therefore, ratio N/n^* is very low: $N/n^* = 100/39602 = 0.0025$. Thus, the decision boundary of the EDC (line 1) is far from the optimal one. For large λ the PW classifier results in the linear discriminant boundary (straight line 2). It is parallel to that of EDC. Due to the random character of $\hat{\Sigma}_1$ and $\hat{\Sigma}_2$ (see Equation (2.51)), with each new randomly chosen training-set this linear decision boundary fluctuates to side of boundary 1. For small λ we obtain a much better boundary (two meandering curves, 4).

3.7.2 The Generalisation Error

In this subsection, we analyse the generalisation error of the PW classifier with the Gaussian kernel function (Equation (2.49) with $\boldsymbol{D} = \mathbf{I}$) in the limit case, when the smoothing parameter λ tends to zero. Then the PW classifier approaches the *k-NN* rule with $k = 1$. Again we use one of our bench-mark models – the multivariate Gaussian with common covariance matrix. We used this data model already while analysing the small sample properties of the parametric statistical classifiers.

At the fixed point $\boldsymbol{x}$ of the multivariate feature space, Ω, a value of the Parzen window distribution density estimate

$$\hat{p}_i(\boldsymbol{x}) = \frac{1}{N_i}\sum_{j=1}^{N_i} N_{\boldsymbol{x}}(\boldsymbol{X}_j^{(i)}, \mathbf{I}\,\lambda). \tag{3.35}$$

depends on N_i vectors of the training-set $\boldsymbol{X}_1^{(i)}, \boldsymbol{X}_2^{(i)}, \dots, \boldsymbol{X}_{N_i}^{(i)}$ considered here as random ones. Therefore, the density (3.35) can be analysed as a random variable. We are making the classification errors in a case where the multivariate vector $\boldsymbol{x}$ belongs to ω_i and $\hat{p}_i(\boldsymbol{x}) < \hat{p}_{3-i}(\boldsymbol{x})$. Therefore, the probability of misclassification conditioned on the particular vector $\boldsymbol{x}$ is determined by

$$P(misclassification \mid \boldsymbol{x}, \boldsymbol{x} \in \omega_i) = P(\hat{p}_i(\boldsymbol{x}) < \hat{p}_{3-i}(\boldsymbol{x})),\ \ (i = 1, 2). \tag{3.36}$$

According to the central limit theorem when $N_i \to \infty$, the distribution of the sum (3.35) of random terms $N_{\boldsymbol{x}}(\boldsymbol{X}_j^{(i)}, \mathbf{I}\lambda)$ tends to the Gaussian one. Thus, the mean generalisation error is determined conditionally by means E and variances V of the estimates $\hat{p}_1(\boldsymbol{x})$ and $\hat{p}_2(\boldsymbol{x})$ at one particular point $\boldsymbol{x}$

$$P(misclassification \mid \boldsymbol{x}, \boldsymbol{x} \in \omega_i) \approx \Phi\{\frac{E\hat{p}_1(\boldsymbol{x}) - E\hat{p}_2(\boldsymbol{x})}{\sqrt{V\hat{p}_1(\boldsymbol{x}) + V\hat{p}_1(\boldsymbol{x})}}\,(-1)^i\}. \tag{3.37}$$

The conditional mean of the nonparametric density estimate (conditioned at fixed point $\boldsymbol{x}$) with respect to all possible training sets consisting of N_i observations, is

$$E\,\hat{p}_i(\boldsymbol{x}) = \frac{1}{N_i}\sum_{j=1}^{N_i} \int N_{\boldsymbol{x}}(\boldsymbol{X}_j^{(i)}, \mathbf{I}\lambda) N_{\boldsymbol{x}}(\boldsymbol{M}_i, \overline{\Sigma})\, d\boldsymbol{X}_j^{(i)} = N_{\boldsymbol{x}}(\boldsymbol{M}_i, \overline{\Sigma} + \mathbf{I}\lambda). \tag{3.38}$$

From (3.38) we see that the nonparametric Parzen window density is a biased estimator. Its expectation is $N_{\boldsymbol{x}}(\boldsymbol{M}_i, \bar{\Sigma} + \mathbf{I}\lambda)$ which is not equal to the true density value $N_x(\boldsymbol{M}_i, \bar{\Sigma})$ at the point $\boldsymbol{X}$. The bias, which is the difference $\Delta(\boldsymbol{X}) = N_x(\boldsymbol{M}_i, \bar{\Sigma} + \mathbf{I}\lambda) - N_x(\boldsymbol{M}_i, \bar{\Sigma})$, changes its sign depending on the distance $(\boldsymbol{X} - \boldsymbol{M}_i)^T \bar{\Sigma}^{-1}(\boldsymbol{X} - \boldsymbol{M}_i)$. In general, the bias increases as the smoothing parameter λ increases. However, for very small λ, we have a minor bias. Note that the bias does not depend on the number of the training examples N_i. For the above data model at the fixed point $\boldsymbol{x}$, the variance of the PW density estimator is

$$V\ \hat{p}_i(\boldsymbol{x}) = \frac{1}{N_i}\left[\frac{|2\bar{\Sigma}+\mathbf{I}\lambda|^{1/2}}{\lambda^{n/2}}\,(N_{\boldsymbol{x}}(\boldsymbol{M}_i, 2\bar{\Sigma} + \mathbf{I}\lambda))^2 - (E\ \hat{p}_i(\boldsymbol{x}))^2\right]. \tag{3.39}$$

Consider the following data model. Let $\mathbf{\Phi}$ be an $n \times n$ orthonormal matrix such that $\bar{\Sigma} = \mathbf{\Phi\Lambda\Phi}^T$ (Λ is a diagonal matrix of the eigenvalues with elements $\lambda_1, \lambda_2, \ldots, \lambda_n$). This model has been discussed in Section 3.3.4. Then

$$V\ \hat{p}_i(\boldsymbol{x}) = \frac{1}{N_i}\left[\prod_{j=1}^{n}\sqrt{1+\frac{2\lambda_j}{\lambda}}\,(N_{\boldsymbol{x}}(\boldsymbol{M}_i, 2\bar{\Sigma} + \mathbf{I}\lambda))^2 - (E\ \hat{p}_i(\boldsymbol{x}))^2\right]. \tag{3.40}$$

For very small λ, the variance of the PW estimator is controlled mainly by the term

$$\frac{1}{N_i}\ \prod_{j=1}^{n}\sqrt{1+\frac{2\lambda_j}{\lambda}}\ . \tag{3.41}$$

This term diminishes as λ, the smoothing parameter, and N_i, the training-set size, are increasing. Let the eigenvalues of the CM Σ be equal: $\lambda_1 = \lambda_2 \ldots = \lambda_n = d$ and let the number of features n be increased. Then for small λ we can conclude that in order to keep the variance (3.40) constant, the number of training vectors N_i should increase exponentially as a function of the dimensionality n at the rate:

$$N_i \equiv \left(1+\frac{2d}{\lambda}\right)^{n/2}. \tag{3.42}$$

3.7.3 Intrinsic Dimensionality

Let n-r eigenvalues of the covariance matrix $\bar{\Sigma}$ be very small:

$$\lambda_1 = \lambda_2 \ldots = \lambda_r = d, \qquad \lambda_{r+1} = \lambda_{r+2} = \ldots = \lambda_n = \varepsilon \to 0.$$

We call r the *intrinsic dimensionality* of the data. For the above data model we have

$$N_i \equiv \left(1+\frac{2d}{\lambda}\right)^{r/2} . \tag{3.43}$$

This result means that the small sample size properties of the nonparametric Parzen window density estimator with the spherical kernel (3.35) are not determined by the formal dimensionality of the data, n, but by the true *intrinsic dimensionality*, r. Therefore, the number of training vectors required to design this classifier should increase exponentially with the intrinsic dimensionality.

3.7.4 Optimal Value of the Smoothing Parameter

Three remarks follow from the above theoretical analysis:

1. The conclusion about the intrinsic dimensionality is valid only for the PW estimator classification rule using *spherical kernels* and a *common* smoothing parameter λ for all n variables.

2. In real-world problems, the intrinsic dimensionality of the data can be different in various parts of the multivariate feature space Ω. Therefore, the sensitivity of the PW classifier to the finiteness of the number of the training examples can also be different in various parts of the feature space Ω.

3. It is important to know that for small λ the estimator (3.35) has a large positive asymmetry. Consequently, for small λ, one will have difficulty in deriving exact analytical formulae that relate the generalisation error, the asymptotic error, the dimensionality, the smoothing parameter and the training-set size.

In Table 3.6 we depict values of learning quantity $\kappa = \bar{\varepsilon}_N^{\mathrm{PW}}(\lambda , N)/\varepsilon_\infty^{\mathrm{PW}}(\lambda)$ obtained by simulation for the spherical Gaussian pattern classes with Mahalanobis distances $\delta = 2.56$ and $\delta = 4.65$. The data in Table 3.6 indicates that a characteristic of the dependence of the expected error rate on the number of training examples essentially relies on the value of smoothing parameter λ.

As predicted by Equation (3.42) the ratio (sample-size/dimensionality) found from Table 3.6 has the following approximate structure: $N = \alpha \times \beta^n$ where α and β are two positive scalars. In Section 3.6.7 we have that for the parametric classifiers EDC, Fisher DF, and quadratic DF the relative increase in the expected PMC $\kappa = \bar{\varepsilon}_N^{\mathrm{A}} / \varepsilon_\infty^{\mathrm{A}}$ is proportional to $\kappa = 1 + \alpha_\tau N^{-\tau}$. For small N the degree τ exceeds 1 and τ approaches 1 as the training-set size increases. For the PW classifier this dependency is confirmed only for large values of λ. For large values of λ, the PW classifier is similar in behaviour to the Euclidean distance classifier and τ is close to 1. However, for small λ, τ = ½ or even τ = 1/3. Thus, the PW classifier behaves similarly to the structural nonparametric ZEE classifier.

Table 3.6. Values of the learning quantity – the ratio $\kappa = \bar{\varepsilon}_N^{\mathrm{PW}} (\lambda , N)/ \ \varepsilon_{\infty}^{\mathrm{PW}} (\lambda)$ – of the Parzen window classifier as a function of number of training examples *N* and dimensionality *n* for two values of the smoothing parameter λ.

$\gamma = \bar{N}/n$	$n = 3$	$n = 5$	$n=8$	$n=3$	$n=5$	$n=8$
λ=0.01		δ = 2.56			δ = 4.65	
0.6	1.97	2.15	2.28	3.53	3.55	4.29
1.0	1.90	1.98	2.13	2.92	3.24	3.48
2.0	1.78	1.87	1.95	2.61	3.07	3.18
5.0	1.64	1.71	1.91	2.32	2.56	2.72
10.0	1.50	1.66	1.84	2.15	2.16	2.28
50.0	1.39	1.62	1.81	1.53	1.86	2.14
λ=0.64		δ = 2.56			δ = 4.65	
0.6	1.96	2.12	2.26	3.51	3.53	4.21
1.0	1.80	1.94	2.05	2.86	3.15	3.41
2.0	1.68	1.80	2.90	2.51	2.87	3.08
5.0	1.46	1.58	1.76	2.18	2.38	2.46
10.0	1.23	1.44	1.65	1.71	1.90	2.10
50.0	1.06	1.12	1.29	1.20	1.27	1.54

The classification of unlabeled data into two spherical multivariate Gaussian populations with equal identity covariance matrices is not a representative data model of real-world problems. In real-world classification problems, we prefer to use the PW classifier to classify populations with complex multimodal distribution density functions. The bias of the Parzen window estimator (see Equation (3.38)) and its asymptotic PMC both increase as the smoothing parameter λ increases. However, the learning quantity κ of the PW classifier usually decreases with increasing λ. Thus, typically the function $\bar{\varepsilon}_N^{\mathrm{PW}} = f(\lambda)$ has a minimum. For complex distributions of the pattern classes and complex decision boundaries it can have more than one minimum.

Example 11. The peaking phenomenon is illustrated by Figure 3.9 where we have three graphs of the dependence of the generalisation error $\varepsilon_N^{\mathrm{PW}}$ on the smoothing parameter λ. The results are for artificial bi-variate data with a complex shaped distribution (see the scatter diagrams in Figures 2.6 and 2.12). The graphs correspond to three randomly chosen training sets of different sizes. For this data model all three graphs exhibit two minima.

We see a sophisticated behaviour for the generalisation error as a function of the smoothing parameter. We call the value of λ where $\varepsilon_N^{\mathrm{PW}} = f(\lambda)$ has a minimum, an *optimal smoothing value* and denote it by $\lambda_{opt}^{\mathrm{PW}}$. Configurations of curves $\varepsilon_N^{\mathrm{PW}} = f(\lambda)$ essentially depend on the number of training pattern vectors and the true pattern-class distributions.

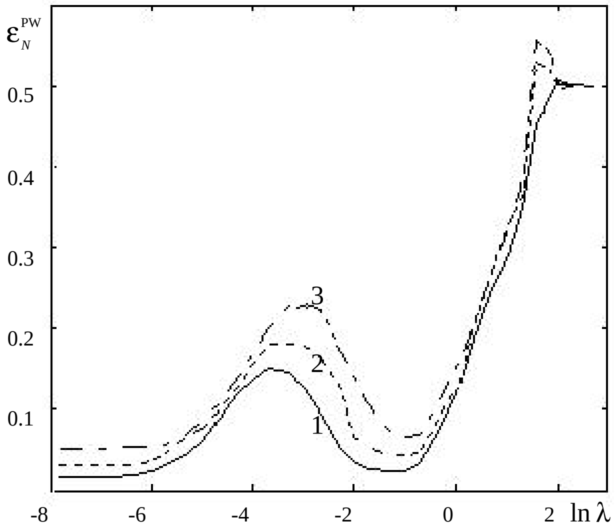

Fig. 3.9. Dependence of the generalisation error on the smoothing parameter λ. For three randomly chosen training sets: 1 – N = 2000, 2 – N = 500, 3 – N = 200.

When we use the PW classifier with a single value for the smoothing parameter for all n features, e.g., estimators (3.35), we recommend that one "normalise" the data in advance by transforming it so that the variances of all n features are equal to 1. The bias of the PW estimates depends on the value of the smoothing parameter λ. Therefore, in order to avoid the discriminant function's bias we recommend using the same λ value for all pattern classes.

Determination of λ_{opt}^{PW}, the optimal value of the smoothing parameter, is very important for solving real-world pattern recognition tasks. However, in evaluating λ_{opt}, theoretical estimates are practically useless because we use the PW classifier to solve complex non-linear real-world problems where multivariate densities are unknown. For that reason, the appropriate decision boundary will almost certainly be non-linear. In order to find λ_{opt}^{PW} for a particular pattern recognition problem, we recommend that one evaluate the classifiers' performance for several values of λ, then choose the value that provides the best generalisation performance.

Recommendations. We make the following suggestions for applying the Parzen window classifier to real-world pattern classification problems:

1. We advise that one normalise the data in order to have the variances of all n features equal to 1. Then the optimal value $\lambda_{optimal}^{PW}$ will usually fall in the interval (0.0001, 100). For most practical problems the "valley" (an interval where the minimum can be found) of the curve $\varepsilon_N^{PW} = f(\lambda)$ is rather flat. Thus, a set of ten

values: 10^{-8}, 10^{-6}, 10^{-4}, 10^{-2}, 10^{-1}, 1, 10 10^{2}, 10^{3}, 10^{4} is often sufficient to evaluate the curve $\varepsilon_N^{\mathrm{PW}} = f(\lambda)$ and to determine $\lambda_{optimal}^{\mathrm{PW}}$.

2. In high-dimensional feature space, evaluation of the classification errors of the Parzen window classifiers for ten different values of $\lambda_{optimal}^{\mathrm{PW}}$ becomes computationally demanding. Most of the computation time is spent on calculating the distances $H(\boldsymbol{X}, \boldsymbol{X}_j^{(i)})$ between the validation set pattern $\boldsymbol{X}$ and $\boldsymbol{X}_j^{(i)}$, the jth training pattern from ith class:

$$H(\boldsymbol{X}, \boldsymbol{X}_j^{(i)}) = (\boldsymbol{X} - \boldsymbol{X}_j^{(i)})^T(\boldsymbol{X} - \boldsymbol{X}_j^{(i)}) = \sum_{s=1}^{n} (x_s - x_{js}^{(i)})^2.$$

To conserve computer processing time, we recommend that the error rates for all values of λ be estimated *simultaneously*. For each single training vector $\boldsymbol{X}_j^{(i)}$ we recommend that one find the distance $H(\boldsymbol{X}, \boldsymbol{X}_j^{(i)})$ and calculate ten terms $h_m^{ij} = \dfrac{H(\boldsymbol{X}, \boldsymbol{X}_j^{(i)})}{\lambda_m}$, $m = 1, \dots, 10$ corresponding to ten values of λ_m. We use an $N \times 10$ table of estimates h_m^{ij} to evaluate the multivariate densities ten times, and classify each single unknown vector $\boldsymbol{X}$ ten times for each value of λ_m. In the high-dimensional case, the computer time required to obtain ten simultaneous estimates of error rates is considerably smaller than the time for ten independent experiments.

3. In most practical problems, the shape of the kernel function $\kappa\{H(\boldsymbol{X}, \boldsymbol{X}_j^{(i)})\}$ does not essentially affect the generalisation error. The value of the smoothing parameter λ is much more important. However, in some problems, hyper-rectangular windows perform worse than smooth “bell-shaped” windows (Skurichina, 1991). The reason for such behaviour is not completely clear.

3.7.5 The *k-NN* rule

In this rule, the class membership of the input pattern vector $\boldsymbol{X}$ is chosen to be the class of a majority of its k nearest neighbours. Frequently the Euclidean distance is used for distance calculations in the original or the “normalised” n-variate feature space. However, the Mahalanobis metric can sometimes lead to better performance.

The *k-NN* and the Parzen window classifiers have many similar characteristics. They both have a varying width of spherical hard-limiting hyper-rectangular window function in the n-dimensional feature space Ω. Both classifiers allow us to obtain complex non-linear decision boundaries. The curvature of the decision boundary depends on the values of the smoothing parameters, k and λ. When $k = 1$ or $\lambda \to 0$, the curvature is maximal and diminishes with an increase in k or λ.

In the finite training-set case, the performance of the *k-NN* classifier essentially depends on *k*, the number of nearest neighbours. Therefore, the optimal number of neighbours should be selected in accordance with the number of training pattern vectors and a configuration of the optimal decision boundary. In practice, the configuration is unknown. For that reason, we recommend estimating the classification error for several values of *k* simultaneously to determine the optimal number of neighbours, *k*.

The *k-NN* classifier is faster than the Parzen window classifier. Numerous studies with large scale real-world data have shown that the *k-NN* classifier often outperforms the MLP-based classification rules.

3.8 Multinomial Classifier

The multinomial classifier M is a nonparametric algorithm for classifying the objects described by categorical variables. This classifier has much in common with decision tree classifiers. In designing co-operative neural networks, one uses the class numbers of decisions of the partner networks (local experts). Thus, one deals with the categorical variables in order to make a final decision. This model is useful in designing co-operative neural networks and analysing decision tree classifiers and genetic learning algorithms. In this section, we consider small sample properties of the multinomial classifier in the two category case.

The multinomial classifier M uses the probabilities of the cells $P_j^{(i)}$ (i = 1, 2; j = 1, 2, … , m) to allocate the multivariate vector $\boldsymbol{X}$ with discrete components to one of the pattern classes. The asymptotic and the Bayes probabilities of misclassification of the multinomial classifier are equal to

$$\varepsilon_\infty^{\mathrm{M}} = \varepsilon_B = \sum_{j=1}^{m} min\{ P_1 P_j^{(1)}, P_2 P_j^{(2)} \}. \tag{3.44}$$

The sample based multinomial classifier M uses the frequencies estimated from the training set: $\hat{P}_j^{(i)} = N_j^{(i)} / N_i$. Here, $\hat{P}_j^{(i)}$ is a sample estimate of the true probability $P_j^{(i)}$ and $N_j^{(i)}$ is the number of observations chosen from N_i observations in the training-set sampled from the i-th class with state s_j (Section 2.9). The generalisation error of the sample based decision rule exceeds $\varepsilon_\infty^{\mathrm{M}} = \varepsilon_B$. A classification error occurs if the multivariate vector $\boldsymbol{X}$ having state s_j actually belongs to ω_i, but $P_i \hat{P}_j^{(i)} < P_{3\text{-}i} \hat{P}_j^{(3-i)}$. Therefore, the expected PMC can be written as

$$\bar{\varepsilon}_N^{\mathrm{M}} = \sum_{j=1}^{m} P\{ P_1 \hat{P}_j^{(1)} < P_2 \hat{P}_j^{(2)} \} P_1 P_j^{(1)} + \sum_{j=1}^{m} P\{ P_1 \hat{P}_j^{(1)} > P_2 \hat{P}_j^{(2)} \} P_2 P_j^{(2)} +$$

$$\sum_{j=1}^{m} P\{ P_1 \hat{P}_j^{(1)} = P_2 \hat{P}_j^{(2)} \}(P_1 P_j^{(1)} + P_2 P_j^{(2)})/2. \tag{3.45}$$

For the case $P_2 = P_1 = ½$ and $N_2 = N_1 = \bar{N}$, we use simple combinatorial results to obtain

$$\bar{\varepsilon}_N^{\mathrm{M}} =$$

$$\frac{1}{2}\sum_{j=1}^{m}\Big[\sum_{t=r+1}^{\bar{N}} \sum_{r=0}^{\bar{N}-1} \frac{\bar{N}!}{r!(\bar{N}-r)!}(P_j^{(1)})^r(1-P_j^{(1)})^{\bar{N}-r}\,\frac{\bar{N}!}{t!(\bar{N}-t)!}(P_j^{(2)})^t(1-P_j^{(2)})^{\bar{N}-t}P_j^{(1)} +$$

$$\sum_{t=r-1}^{\bar{N}-1} \sum_{r=1}^{\bar{N}} \frac{\bar{N}!}{r!(\bar{N}-r)!}(P_j^{(1)})^r(1-P_j^{(1)})^{\bar{N}-r}\,\frac{\bar{N}!}{t!(\bar{N}-t)!}(P_j^{(2)})^t(1-P_j^{(2)})^{\bar{N}-t}P_j^{(2)} +$$

$$\sum_{r=0}^{\bar{N}} \frac{\bar{N}!}{r!(\bar{N}-r)!}(P_j^{(1)})^r(1-P_j^{(1)})^{\bar{N}-r}\,\frac{\bar{N}!}{r!(\bar{N}-r)!}(P_j^{(2)})^r(1-P_j^{(2)})^{\bar{N}-r}(P_j^{(1)}+P_j^{(2)})/2\,\Big]. \tag{3.46}$$

Equation (3.46) is a function of all $2m$ probabilities $P_j^{(i)}$. To estimate these $2m$ probabilities is a difficult task. There are three approaches one can use to analyse the small sample characteristics of the multinomial classifier theoretically.

In the first approach, one assumes a prior distribution for $2m$ probabilities $P_1^{(1)},\ P_2^{(1)}, \ldots , P_m^{(2)}$) and averages the error rate (3.46) over this prior distribution. The result, the *mean expected probability of misclassification,*

$$\bar{\bar{\varepsilon}}_N^{\mathrm{M}} = \int \bar{\varepsilon}_N^{\mathrm{M}} f^{prior}(P_1^{(1)},\ P_2^{(1)}, \ldots , P_m^{(2)})\ d P_1^{(1)}\ d P_2^{(1)} \ldots d P_m^{(2)} ,$$

characterises all possible classification problems defined by the prior distribution

$$f^{prior}(P_1^{(1)},\ P_2^{(1)}, \ldots , P_m^{(2)}).$$

In this case, one does not need to estimate values of unknown parameters. Unfortunately, the mean expected probability of misclassification is averaged over a variety of theoretically possible problems and yields little information about a particular problem. For the uniform prior distribution this approach results in very pessimistic estimates of the classifier performance.

In the second approach, one analyses the classifier for the best and the worst cases. The results yield lower and upper error bounds. First, let

$$P_3^{(1)} = P_4^{(1)} = \ldots = P_m^{(1)} = P_3^{(2)} = P_4^{(2)} = \ldots = P_m^{(2)} = \varepsilon,$$

where ε is an extremely small positive constant such that

$$P_3^{(1)} + P_4^{(1)} + .. + P_m^{(1)} = (m\text{-}2)\varepsilon ,$$

which is practically zero.

Then only two states, s_1 and s_2, actually influence the increase in the classification error. The contributions of other m-2 cells is zero. In this case, we virtually have a classification problem with two states, i.e. m = 2. We refer to this data model as a *most favourable* case. Note that one can suggest a more favourable but unrealistic model with three non-empty cells having cell probabilities $P_1^{(1)} = P_3^{(2)} =1- \varepsilon_B$, $P_2^{(1)} = P_2^{(2)} = \varepsilon_B$, $P_3^{(1)} = P_1^{(2)} = 0$. Here only a few observation vectors are sufficient to train the classifier in order to get an ideal classification rule where $\bar{\varepsilon}_N^{\mathrm{M}} = \varepsilon_B$. In the extreme case, when $\varepsilon_B = 0$, only one vector per class is sufficient to obtain perfect generalisation.

Now, let $m = 2r$ (r is an integer) and let us consider a data model in which

$$P_1^{(1)} = P_2^{(1)} =\ldots = P_r^{(1)} = 2\,\varepsilon_B\,/m, \qquad P_{r+1}^{(1)} = P_{r+2}^{(1)} =\ldots = P_m^{(1)} = 2(1-\varepsilon_B)/m,$$

$$P_1^{(2)} = P_2^{(2)} =\ldots = P_r^{(2)} = 2(1-\varepsilon_B)/m, \quad P_{r+1}^{(2)} = P_{r+2}^{(2)} =\ldots = P_m^{(2)} = 2\varepsilon_B\,/m. \qquad (3.47)$$

In this *very unfavourable* situation, one needs to estimate the conditional probabilities $P_j^{(i)}$ of *all* cells. Therefore, the increase in the expected probability of misclassification due to the finite number of training examples is large. In Table 3.7 we have the expected probabilities of misclassification for the very unfavourable model of the data computed by using formula (3.46).

Table 3.7. The expected probabilities of misclassification of the multinomial classifier for the least favourable case of distributions of the pattern classes.

$\bar{N}$	$\varepsilon_\infty^{(\mathrm{M})}\backslash m$	10	20	30	40	50	100	200	500	1000
10	0.10	.191	.286	.339	.371	.392	.441	.469	.487	0.493
	0.01	.068	.186	.260	.307	.339	.412	.454	.481	0.490
20	0.10	.128	.198	.251	.290	.318	.393	.441	.475	0.487
	0.01	.017	.075	.140	.191	.231	.340	.412	.463	0.481
30	0.10	.110	.155	.200	.237	.267	.353	.416	.463	0.481
	0.01	.011	.034	.077	.120	.160	.281	.374	.445	0.472
40	0.10	.103	.132	.168	.201	.229	.319	.393	.452	0.475
	0.01	.010	.019	.045	.078	.111	.233	.340	.428	0.463
50	0.10	.101	.119	.147	.175	.201	.291	.372	.441	0.469
	0.01	.010	.013	.029	.052	.079	.193	.310	.412	0.454
100	0.10	.100	.102	.103	.121	.134	.202	.292	.393	0.441
	0.01	.010	.010	.011	.014	.020	.080	.194	.341	0.412
200	0.10	.100	.100	.100	.101	.105	.134	.203	.321	0.393
	0.01	.010	.010	.010	.010	.010	.020	.081	.235	0.341
500	0.10	.100	.100	.100	.100	.100	.102	.121	.203	0.292
	0.01	.010	.010	.010	.010	.010	.010	.014	.081	0.195
1000	0.10	.100	0.10	.100	.100	.100	.100	.102	.135	0.203
	0.01	.010	0.01	.010	.010	.010	.010	.010	.021	0.081

We can use the generalisation error values calculated for the very unfavourable case as upper bounds for the expected PMC. Also, we can use the generalisation errors calculated for the most favourable case (m=2) as lower bounds of the expected PMC. Thus, the data in Table 3.7 can help us to evaluate a possible increase in the generalisation error due to the inaccurate estimation of the probabilities $P_j^{(i)}$.

In the third approach, we need to know all 2m probabilities and calculate the expected error according formula (3.46). In many practical situations, a significant portion of the probabilities $P_z^{(1)} + P_z^{(2)} = P_z$ are very close to zero. Then the states s_z with very small $P_z = (P_z^{(1)} + P_z^{(2)})/2$ play a very insignificant role in determining the small sample properties of the multinomial classification rule. This is an intermediate approach between the favourable and unfavourable cases.

Example 12. Consider a multi-modular ANN composed of six independent “expert” neural networks. Each expert performs the classification into three pattern classes. For this particular pattern classification problem the multinomial classifier is used as a “boss” classifier to make a final decision based on six decisions of individual experts. After designing six expert networks, we used an independent validation set (1000 vectors from each class) to estimate $2m = 2 \times 729$ probabilities $P_z^{(1)}$, $P_z^{(2)}$ (z = 1, 2, … , 729). In Figure 3.10 we have a scatter diagram of distribution of $m = 3^6 = 729$ estimates of the bi-variate vector ($P_z^{(1)}$, $P_z^{(2)}$). The distribution of ($P_z^{(1)}$, $P_z^{(2)}$) is depictured in two scales. In order to better display the number of cells with identical values of vector ($P_z^{(1)}$, $P_z^{(2)}$) a small amount of noise (uniform in an interval (-0.00035, 0.00035)) was added to each $P_z^{(i)}$ value.

From the total number of 729 cells, 297 cells have zero conditional probabilities, i.e. 40% of cells are empty. Thus, instead of 729 cells, the formal complexity of the problem is $m < 430$. Taking into account that in 146 cells, the conditional probability of one of the classes is zero and in another one, only 0.001, we may conclude that the actual complexity of the problem is still lower. The asymptotic and Bayes errors are $\varepsilon_B = 0.059$. The expected generalisation error rate calculated for $m = 430$ cells, according formula (3.46), is $\bar{\varepsilon}_N^{\mathrm{M}} = 0.34$ for $\bar{N} = 25$, $\bar{\varepsilon}_N^{\mathrm{M}} = 0.29$ for $\bar{N} = 50$, and $\bar{\varepsilon}_N^{\mathrm{M}} = 0.24$ for $\bar{N} = 100$.

An example with a high percentage of almost empty cells is *typical* for real-world problems with categorical variables. Usually, in practical problems, there is a high number of empty cells and the increase in the classification error is significantly smaller than that predicted by the least favourable case. One recommendation for the practical evaluation of the generalisation error is to estimate the number of non zero cells from the training data set and then use this number (say $m_{effective}$) in calculations according to Equations (3.46) and (3.47). In this approach, we analyse the least favourable model for $m_{effective}$ non empty cells. Hence, one must remember that this error estimate remains pessimistically biased.

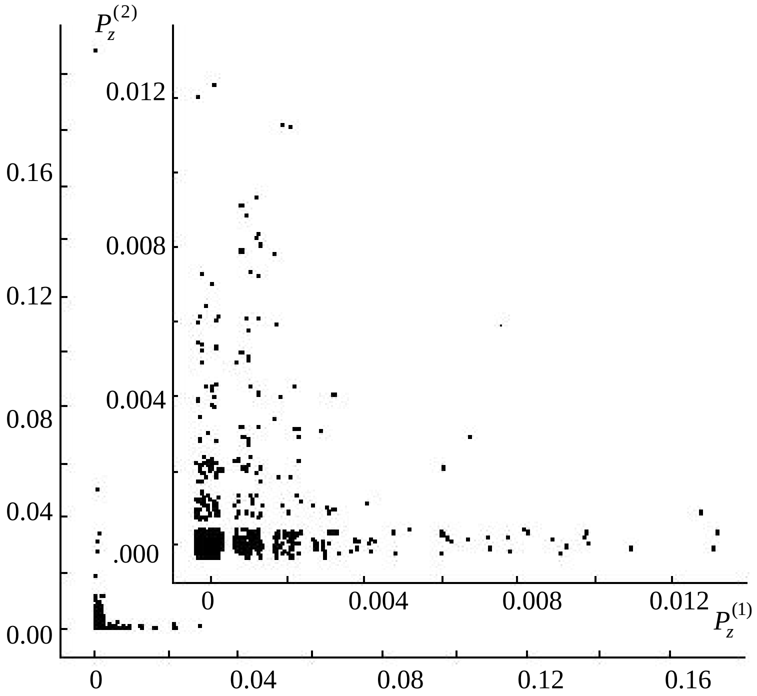

Fig. 3.10. The distribution of 729 values of probabilities $P_z^{(1)}$, $P_z^{(2)}$, which were obtained in decision making on the basis of six expert networks in the multi-modular ANN.

The high percentage of almost empty cells also explains the advantage of the application of the decision-tree classifiers. Here, in order to make classifications, many empty cells are merged with the non-empty ones. Thus, for most practical pattern recognition problems, the number of cells for which the designer needs to estimate probabilities $P_z^{(1)}$, $P_z^{(2)}$ is dramatically reduced. Then one can obtain a very simple classification rule.

3.9 Bibliographical and Historical Remarks

The performance of the adaptive linear classifier known as adaline was first studied by Widrow and Hoff (1960). They concluded that *N*, the training-set size required to achieve a particular learning quality, should be proportional to *n*, the number of variables. Cover (1965) introduced the concept of the capacity, a measure of the complexity, and showed that the generalisation error decreases in proportion to *n*/*N*, the ratio of the dimensionality to the training-set size.

A number of different approaches have been proposed to study the generalisation error in the finite training-set size situations. In Sections 3.1.5, 3.6.2 and 3.6.7 we reviewed the Vapnik and Chervonenkis, the information–theoretic, and the statistical–mechanic approaches. Unfortunately, the results are valid only when the training-set sizes are very large and the expected classification errors very close to the asymptotic error. However, for practitioners, the most interesting cases are where the number of training vectors is small and we have a significant difference between the asymptotic and generalisation errors.

Among other approaches not considered in this book, the most popular is the *probable almost correct* (PAC) framework (Valiant, 1984). The *combining statistical physics with VC-bounds* methodology allows one to incorporate some problem-specific information (see Kowalczyk, 1996, and references therein). In a stream of papers concerning this approach, researchers demonstrated that the introduction of limited information on the distribution of error patterns to the classical VC formalism permits much tighter bounds on learning curves. The phase transitions, as well as significant drops in learning errors, can be modelled for low training-set sizes, for which the classical VC-bounds are void. The *probably almost Bayes* (PAB) *decisions* framework (Anoulova *et al*, 1992) allows one to obtain upper bounds for the training-set size for classifying feature vectors with Boolean or real entries. Gaussian distributions with unknown means and unknown, diagonal and known covariance matrices were considered, however, the conclusions obtained are qualitatively different from those reported in this book.

A great deal of research has been done on the analysis of the small sample behaviour of statistical classifiers. As the statistical–mechanic approach, classical statistical analysis also requires knowledge of the input signal distribution $p_i(\boldsymbol{X})$. This is the weak point of both approaches. However, assumptions on the probabilistic structure of the pattern classes and on the parameters allows one to obtain narrower error bounds. In some cases, exact results can be obtained and only one question remains – how to use these results in practice, where the true pattern-class distributions are unknown.

Rao (1949) was the first to emphasise the problems that arise in cases where the number of training examples is approximately the number of dimensions. The first estimate of the difference between the generalisation and the asymptotic errors was obtained by numerical simulation at the Institute for Numerical Analysis of the University of California in Los Angeles (see references in Solomon, 1956). Sitgreaves (1961) derived the first exact formula for the expected classification error of the standard Fisher linear discriminant function (DF) in a form of a five times infinite sum of products of certain hypergeometric functions. Estes (1965) succeeded in calculating this sum and Pikelis improved the calculation accuracy and presented a table (Pikelis, 1974, see also Raudys and Pikelis, 1980, and references therein). The first asymptotic expansion for the expected classification error of the Fisher linear DF belongs to Okamoto (1963). It is obtained asymptotically, where $N \rightarrow \infty$, and yields inaccurate values when the dimensionality n is large and sample size N is small. John (1961) represented the linear discriminant function with known covariance matrix as a difference of two independent chi-square variables and expressed the expected error in a form of infinite sum. Raudys (1967) used this result and derived the first simple asymptotic formula for the expected probability of misclassification (PMC) of the Euclidean distance classifier (3.7). Faithful and unfaithful cases were first analysed here, as well as the double asymptotics approach (the thermodynamic limit) where both $N \to \infty$ and $n \to \infty$.

Deev, working in A. N. Kolmogorov's Laboratory of Statistical Methods at Moscow State University, formalised the double asymptotics approach in a strictly mathematical way. He formally required $N_1 \to \infty$, $N_2 \to \infty$, $n \to \infty$, $n/N \to const.$ and Mahalanobis distance $\delta = const.$ Under this approach, several subsequent asymptotic expansions were obtained for Gaussian and non-Gaussian models. Two simple formulae for the expected error of the standard Fisher linear DF were obtained by Deev (1970, 1972) and Raudys (1972). The representation of the increase in the conditional PMC as a sum of two chi-square random variables and subsequent expressions for its mean (3.11) and the variance (3.12) is from Efron (1975). Further analysis (Pikelis, 1976; Wyman *et al.*, 1990; Takeshita and Toriwaki, 1995) has shown that the double asymptotic expansions give very accurate estimates and better accuracy than the asymptotic expansions where only $N_1 \to \infty$, $N_2 \to \infty$, such as asymptotic expansions by Okamoto, Efron, Kharin, Fukunaga and several others. The double asymptotic approach was used to obtain the generalisation error for the quadratic DF, linear and non linear classifiers for independent Gaussian variables (Raudys, 1972), classifiers assuming block and tree type dependencies between the Gaussian variables (Deev, 1974; Zarudskij, 1979), and a classifier for independent categorical variables (Meshalkin, 1976). The bias term (3.9) to make the discriminant function unbiased was independently proposed by Deev (1970, 1972) and Chandrasekaran and Jain (1979). Barsov (1982) and Serdobolskij (1983*a*) derived asymptotic expected errors for two modifications of regularised DA. Formula (3.18) is from Raudys and Skurichina (1994), Raudys *et al.* (1995) and expression (3.17) is from Raudys and Duin (1998). Meshalkin and Serdobolskij (1978) and Serdobolskij (1979) proved the fundamental limit theorem for the generalisation error for arbitrary non-Gaussian classes. Table 3.2 is from Raudys and Pikelis, 1980, however, in Raudys, Pikelis and Juskevicius (1975) the reader can find more extensive tables.

The *curse of dimensionality* was first described by Allais (1966), Hughes (1965) and Lbov (1966), and the *scissors effect* was noticed in Raudys (1970) and Kanal and Chandrasekaran (1971). In small training-set cases, simple structured classification rules are more efficient than complex classification rules. In large training-set cases, complex classifiers are more efficient. Use of a random search optimisation procedure to analyse the generalisation error of the ZEE and maximal margin classifiers in the multivariate Gaussian case was suggested in Raudys (1993), along with the concept of intrinsic dimensionality. Table 3.3 is from Raudys (1997) and Table 3.6 for the Parzen window classifier is from Raudys (1991). The analytical expression for the generalisation error of the multinomial classifier was presented in Griskevicius and Raudys (1979).