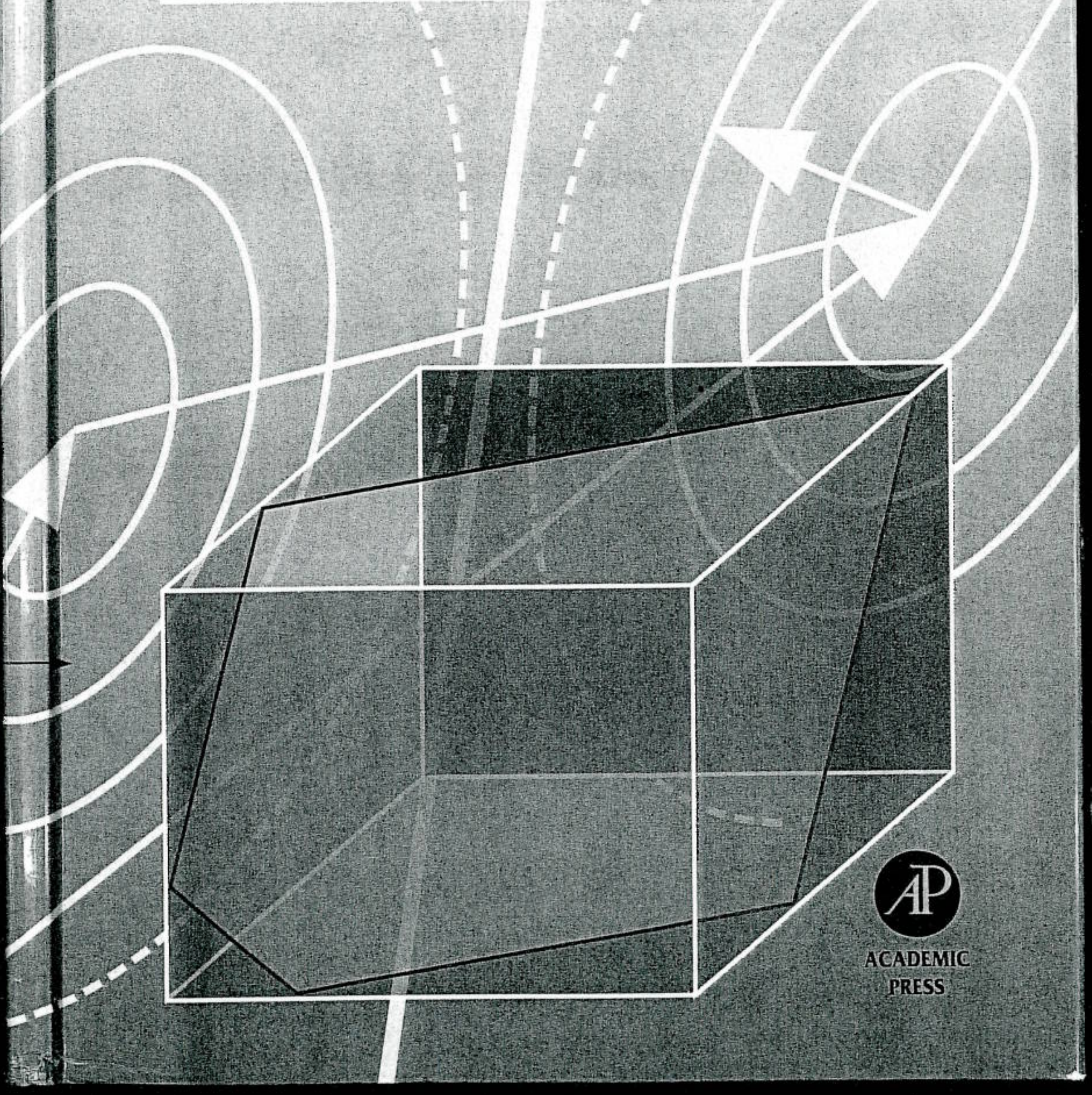
PATTERN RECOGNITION

SERGIOS THEODORIDIS
KONSTANTINOS KOUTROUMBAS



AP
ACADEMIC PRESS

ɔpagation, weight
nectionist Models
ds.), pp. 105–116,

nd analysis in the
, MA, 1974.
' radial basis func-
Information Pro-
ı, pp. 1133–1140,

ɔrks classification
:d.), pp. 850–859,

ɘ evaluation of a
' *Transactions on*

CHAPTER 5

FEATURE SELECTION

5.1 INTRODUCTION

In all previous chapters, we considered the features to be available prior to the design of the classifier. The goal of this chapter is to study methodologies related to the selection of these variables. As we pointed out very early in the book, a major problem associated with pattern recognition is the so-called curse of dimensionality (Section 2.5.6). The number of features at the disposal of the designer of a classification system is usually very large. As we will see in Chapter 7, this number can easily become of the order of a few dozens or even hundreds. There is more than one reason for the necessity to reduce the number of features to a sufficient minimum. Computational complexity is the obvious one. A related reason is that although two features may carry good classification information when treated separately, there is little gain if they are combined together in a feature vector, because of a high mutual correlation. Thus, complexity increases without much gain. Another major reason is that imposed by the required generalization properties of the classifier, as discussed in Section 4.9 of the previous chapter. According to the discussion there and as we will state more formally at the end of this chapter, the higher the ratio of the number of training patterns N to the number of free classifier parameters, the better the generalization properties of the resulting classifier. A large number of features is directly translated into a large number of classifier parameters (e.g., synaptic weights in a neural network, weights in a linear classifier). Thus, for a finite and usually limited number N of training patterns, keeping the number of features as small as possible is in line with our desire to design classifiers with good generalization capabilities. Furthermore, the ratio N/l enters the scene from another nearby corner. One important step in the design of a classification system is the performance evaluation stage, in which the classification error probability of the designed classifier is estimated. We not only need to design a classification system, we must also assess its performance. As is pointed out in Chapter 10, the classification error estimate improves as this ratio becomes higher. In [Fine 83] it is pointed out that in some cases ratios as high as 10 to 20 were considered necessary.

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

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

5.2 PREPROCESSING

5.2.1 Outlier Removal

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

5.2.2 Data Normalization

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

$$\begin{aligned} \bar{x}_k &= \frac{1}{N}\sum_{i=1}^{N} x_{ik}, \qquad k = 1, 2, \ldots, l \\ \sigma_k^2 &= \frac{1}{N-1}\sum_{i=1}^{N}(x_{ik} - \bar{x}_k)^2 \\ \hat{x}_{ik} &= \frac{x_{ik} - \bar{x}_k}{\sigma_k} \end{aligned}$$

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

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

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

5.2.3 Missing Data

In practice, it may happen that the number of available data is not the same for all features. If the number of training data is high enough, we can afford to discard some of them and keep a smaller number, the same for all features, in order to form the feature vectors. However, in many cases it is a luxury to drop available data. In these cases missing data have to be predicted heuristically. An obvious thought is to replace missing values with the corresponding mean, computed from the available values of the respective features. More elaborate techniques, exploiting the statistical information about the underlying distribution, have also been proposed and used [Ghah 94, Lowe 90]. We will return to the missing data problem in Chapter 11.

5.3 FEATURE SELECTION BASED ON STATISTICAL HYPOTHESIS TESTING

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

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

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

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

5.3.1 Hypothesis Testing Basics

Let x be a random variable with a probability density function, which is assumed to be known within an unknown parameter θ. As we have already seen in Chapter 2, in the case of a Gaussian this parameter may be the mean value or its variance. Our interest here lies in the following hypothesis test:

$$H_1:\quad \theta \neq \theta_0$$
$$H_0:\quad \theta = \theta_0$$

The decision on this test is reached in the following context. Let $x_i,\ i = 1, 2, \ldots, N$, be the experimental samples of the random variable x. A function $f(\cdot, \ldots, \cdot)$ is selected, depending on the specific problem, and let $q = f(x_1, x_2, \ldots, x_N)$. The

FIGUI
of the s

functio
ized in
it has a
that is,
of q tha
accept
and $\bar{D}$ a
questio
Then th
bility is
tice, we
sometir
us now
of x.

The Kn

Let x b
sample:

A popu

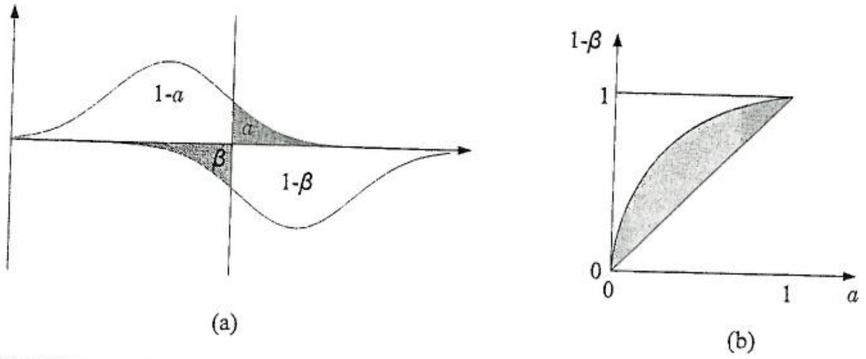


FIGURE 5.2: Example of (a) overlapping pdf's of the same feature in two classes and (b) the resulting ROC curve.

two completely separated class distributions, moving the threshold to sweep the whole range of values for a in [0, 1], $1-\beta$ remains equal to unity. Thus, the aforementioned area varies between zero, for complete overlap, and 1/2 (the area of the upper triangle), for complete separation, and *it is a measure of the class discrimination capability of the specific feature*. In practice, the ROC curve can easily be constructed by sweeping the threshold and computing percentages of wrong and correct classifications over the available training feature vectors. Other related criteria that test the overlap of the classes have also been suggested (see Problem 5.5).

5.5 CLASS SEPARABILITY MEASURES

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

5.5.1 Divergence

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

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

As pointed ou
difference betv
$\frac{P(\omega_1|x)}{P(\omega_2|x)}$ can con
associated with
Alternatively (
the ratio ln $\frac{p(x|}{p(x|}$
discriminating
overlapped cla
to consider the

Similar argume

The sum

is known as the
ω_1, ω_2, with res
divergence is cc

and the average

Divergence is ba
density function
shown propertie

As pointed out in Chapter 2, the classification error probability depends on the difference between $P(\omega_1|\boldsymbol{x})$ and $P(\omega_2|\boldsymbol{x})$, e.g., equation (2.12). Hence, the ratio $\frac{P(\omega_1|\boldsymbol{x})}{P(\omega_2|\boldsymbol{x})}$ can convey useful information concerning the discriminatory capabilities associated with an adopted feature vector $\boldsymbol{x}$, with respect to the two classes ω_1, ω_2. Alternatively (for given values of $P(\omega_1)$, $P(\omega_2)$), the same information resides in the ratio $\ln \frac{p(\boldsymbol{x}|\omega_1)}{p(\boldsymbol{x}|\omega_2)} \equiv D_{12}(\boldsymbol{x})$ and this can be used as a measure of the underlying discriminating information of class ω_1 with respect to ω_2. Clearly, for completely overlapped classes we get $D_{12}(\boldsymbol{x}) = 0$. Since $\boldsymbol{x}$ takes different values, it is natural to consider the mean value over class ω_1, that is,

$$D_{12} = \int_{-\infty}^{+\infty} p(\boldsymbol{x}|\omega_1) \ln \frac{p(\boldsymbol{x}|\omega_1)}{p(\boldsymbol{x}|\omega_2)} d\boldsymbol{x} \tag{5.11}$$

Similar arguments hold for class ω_2 and we define

$$D_{21} = \int_{-\infty}^{+\infty} p(\boldsymbol{x}|\omega_2) \ln \frac{p(\boldsymbol{x}|\omega_2)}{p(\boldsymbol{x}|\omega_1)} d\boldsymbol{x} \tag{5.12}$$

The sum

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

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

$$\begin{aligned} d_{ij} &= D_{ij} + D_{ji} \\ &= \int_{-\infty}^{+\infty} (p(\boldsymbol{x}|\omega_i) - p(\boldsymbol{x}|\omega_j)) \ln \frac{p(\boldsymbol{x}|\omega_i)}{p(\boldsymbol{x}|\omega_j)} d\boldsymbol{x} \end{aligned} \tag{5.13}$$

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

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

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

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

If the components of the feature vector are statistically independent, then it can be shown (Problem 5.8) that

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

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

$$d_{ij} = \frac{1}{2}\text{trace}\{\Sigma_i^{-1}\Sigma_j + \Sigma_j^{-1}\Sigma_i - 2I\} + \frac{1}{2}(\mu_i - \mu_j)^T(\Sigma_i^{-1} + \Sigma_j^{-1})(\mu_i - \mu_j) \tag{5.14}$$

For the one-dimensional case this becomes

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

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

Let us now investigate (5.14). If the covariance matrices of the two Gaussian distributions are equal, $\Sigma_i = \Sigma_j = \Sigma$, then the divergence is further simplified to

$$d_{ij} = (\mu_i - \mu_j)^T\Sigma^{-1}(\mu_i - \mu_j)$$

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

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

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

5.5.2 Chernoff Bound and Brattacharyya Distance

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

$$P_e = \int_{-\infty}^{\infty} \min\big[P(\omega_i)p(\boldsymbol{x}|\omega_i), P(\omega_j)p(\boldsymbol{x}|\omega_j)\big]d\boldsymbol{x} \tag{5.15}$$

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

$$\min[a, b] \leq a^s b^{1-s} \quad \text{for} \quad a, b \geq 0, \quad \text{and} \quad 0 \leq s \leq 1 \tag{5.16}$$

Combining (5.15) and (5.16), we get

$$P_e \leq P(\omega_i)^s P(\omega_j)^{1-s} \int_{-\infty}^{\infty} p(\boldsymbol{x}|\omega_i)^s p(\boldsymbol{x}|\omega_j)^{1-s} d\boldsymbol{x} \equiv \epsilon_{CB} \tag{5.17}$$

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

$$P_e \leq \epsilon_{CB} = \sqrt{P(\omega_i)P(\omega_j)} \int_{-\infty}^{\infty} \sqrt{p(\boldsymbol{x}|\omega_i)p(\boldsymbol{x}|\omega_j)}d\boldsymbol{x} \tag{5.18}$$

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

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

where

$$B = \frac{1}{8}(\boldsymbol{\mu}_i - \boldsymbol{\mu}_j)^T\left(\frac{\Sigma_i + \Sigma_j}{2}\right)^{-1}(\boldsymbol{\mu}_i - \boldsymbol{\mu}_j) + \frac{1}{2}\ln\frac{\left|\frac{\Sigma_i+\Sigma_j}{2}\right|}{\sqrt{|\Sigma_i||\Sigma_j|}} \tag{5.19}$$

and $|\cdot|$ denotes the determinant of the respective matrix. The term B is known as the *Brattacharyya distance* and it is used as a class separability measure. It can be shown (Problem 5.9) that it corresponds to the optimum Chernoff bound when

$\Sigma_i = \Sigma_j$. It is readily seen that in this case the Brattacharyya distance becomes proportional to the Mahalanobis distance between the means.

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

Example 5.4. Assume that $P(\omega_1) = P(\omega_2)$ and that the corresponding distributions are Gaussians $\mathcal{N}(\mu, \sigma_1^2 I)$ and $\mathcal{N}(\mu, \sigma_2^2 I)$. The Brattacharyya distance becomes

$$B = \frac{1}{2}\ln\frac{\left(\frac{\sigma_1^2+\sigma_2^2}{2}\right)^l}{\sqrt{\sigma_1^{2l}\sigma_2^{2l}}} = \frac{1}{2}\ln\left(\frac{\sigma_1^2+\sigma_2^2}{2\sigma_1\sigma_2}\right)^l \tag{5.20}$$

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

$$P_e \leq 0.2225$$

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

$$P_e \leq 0.0707$$

Thus, the greater the difference of the variances, the smaller the error bound. The decrease is bigger for higher dimensions due to the dependence on l. Figure 5.3 shows the pdf's for the same mean and $\sigma_1 = 1$, $\sigma_2 = 0.01$. The figure is self-explanatory as to how the Bayesian classifier discriminates between two classes of

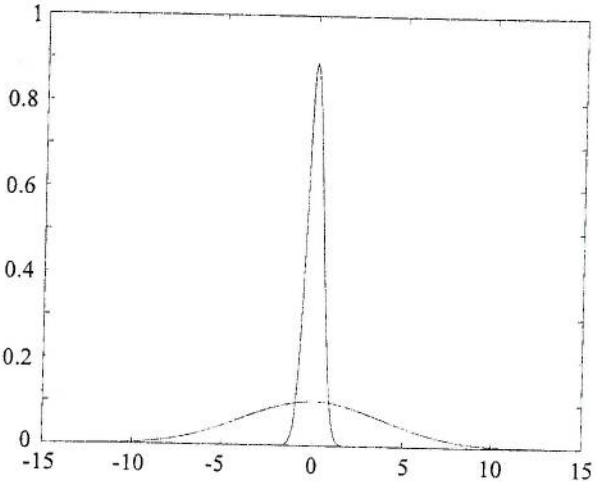


FIGURE 5.3: Gaussian pdf's with the same mean and different variances.

the same mea
the probabilit

5.5.3 Scat

A major disa
they are not ε
will now turr
related to the
To this end, tl

Within-class s

where S_i is th

and P_i the *a p*
number of sar
is a measure c

Between-class

where μ_0 is th

trace$\{S_b\}$ is a
each individua

Mixture scatte

That is, S_m is t

the same mean and significantly different variances. Furthermore, as $\sigma_2/\sigma_1 \longrightarrow 0$, the probability of error tends to zero (why?).

5.5.3 Scatter Matrices

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

Within-class scatter matrix

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

where S_i is the covariance matrix for class ω_i

$$S_i = E[(x - \mu_i)(x - \mu_i)^T]$$

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

Between-class scatter matrix

$$S_b = \sum_{i=1}^{M} P_i(\mu_i - \mu_0)(\mu_i - \mu_0)^T$$

where μ_0 is the global mean vector

$$\mu_0 = \sum_{i}^{M} P_i \mu_i$$

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

Mixture scatter matrix

$$S_m = E[(x - \mu_0)(x - \mu_0)^T]$$

That is, S_m is the covariance matrix of the feature vector with respect to the global mean. It is not difficult to show (Problem 5.10) that

$$S_m = S_w + S_b$$

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

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

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

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

A variant of J_2 commonly encountered in practice is

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

↳ b

As we will see later on, criteria J_2 and J_3 have the advantage of being invariant under linear transformations, and we will adopt them to derive features in an optimal way. In [Fuku 90] a number of different criteria are also defined by using various combinations of S_w, S_b, S_m in a "trace" or "determinant" formulation. However, whenever a determinant is used, one should be careful with S_b, since $|S_b| = 0$ for $M < l$. This is because S_b is the sum of M $l \times l$ matrices, of rank one each.

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

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

FDR is sometimes used to quantify the separability capabilities of individual features. FDR reminds us of the test statistic q appearing in the hypothesis statistical tests dealt with before. However, here the use of FDR is suggested in a more

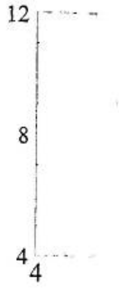


FIGURE
class dist
and (c) sr

"primitiv
multiclas

where the
feature ur

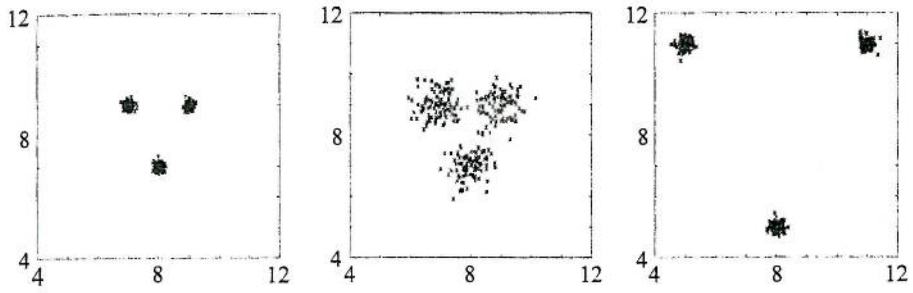
Example 5
class varia
are 164.7,
best is for c
with large

5.6 FEA

Having de
individual
that is, to
major dire

5.6.1 S

Features a
can be ad



(a) (b) (c)

FIGURE 5.4: Classes with (a) small within-class variance and small between-class distances, (b) large within-class variance and small between-class distances and (c) small within-class variance and large between-class distances.

"primitive" fashion, independent of the underlying statistical distributions. For the multiclass case, averaging forms of FDR can be used. One possibility is

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

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

Example 5.5. Figure 5.4 shows three cases of classes at different locations and within-class variances. The resulting values for the J_3 criterion involving the S_w and S_m matrices are 164.7, 12.5, and 620.9 for the cases in Figures 5.4a, b, and c, respectively. That is, the best is for distant well-clustered classes and the worst for the case of closely located classes with large within-class variance.

5.6 FEATURE SUBSET SELECTION

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

5.6.1 Scalar Feature Selection

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

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

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

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

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

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

$$\rho_{ij} = \frac{\sum_{n=1}^{N} x_{ni} x_{nj}}{\sqrt{\sum_{n=1}^{N} x_{ni}^2 \sum_{n=1}^{N} x_{nj}^2}} \tag{5.21}$$

It can be shown that $|\rho_{ij}| \leq 1$ (Problem 5.11). The selection procedure evolves along the following steps:

- Select a class separability criterion C and compute its values for all the available features x_k, $k = 1, 2, \ldots, m$. Rank them in descending order and choose the one with the best C value. Let us say that this is x_{i_1}.
- To select the second feature, compute the cross-correlation coefficient defined in Eq. (5.21) between the chosen x_{i_1} and each of the remaining $m - 1$ features, that is, $\rho_{i_1 j}$, $j \neq i_1$.

- Choose the feature x_{i_2} for which

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

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

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

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

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

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

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

5.6.2 Feature Vector Selection

Treating features individually, that is, as scalars, has the advantage of computational simplicity but may not be effective for complex problems and for features with high mutual correlation. We will now focus on techniques measuring classification capabilities of feature vectors. It does not require much thought to see that computational burden is the major limiting factor of such an approach. Indeed, if we want to act according to what "optimality" suggests, we should form *all* possible vector combinations of l features out of the m originally available. For each combination we should use one of the separability criteria introduced previously (e.g., Brattacharrya distance, J_2) and select the best feature vector combination. Recalling our combinatorics basics, we obtain the total number of vectors as

$$\binom{m}{l} = \frac{m!}{l!(m-l)!} \tag{5.22}$$

This is a large number even for small values of l, m. Indeed, for $m = 20$, $l = 5$, the number equals 15504. Furthermore, in many practical cases the number l is not even known a priori. Thus, one has to try feature combinations for different values of l and select the "best" value for it (beyond which no gain in performance is obtained) and the corresponding "best" l-dimensional feature vector. As we will see in Chapter 10, sometimes it is desirable to base our feature selection decision not on the values of an adopted class separability criterion but on the performance of the classifier itself. That is, for each feature vector combination the classification error probability of the classifier has to be estimated and the combination resulting in the minimum error probability selected. This approach may increase the complexity requirements even more, depending, of course, on the classifier type. In order to reduce complexity, a number of efficient searching techniques have been suggested. Some of them are suboptimal and some optimal (under certain assumptions or constraints).

Suboptimal Searching Techniques

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

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

Thus, starting from m, at each step we drop out one feature from the "best" combination until we obtain a vector of l features. Obviously, this is a *suboptimal* searching procedure, since nobody can guarantee that the optimal two-dimensional vector has to originate from the optimal three-dimensional one. The number of combinations searched via this method is $1 + 1/2((m + 1)m - l(l + 1))$ (Problem 5.13), which is substantially less than that of the full search procedure.

Sequential Forward Selection. Here, the reverse to the preceding procedure is followed:

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

'or different
erformance
:tor. As we
-e selection
but on the
ombination
ted and the
is approach
' course, on
it searching
me optimal

ia an exam-
We wish to
steps:

· the feature

mbinations,
ompute the
best value,

one feature
, $[x_2, x_3]^T$,
ie.

the "best"
suboptimal
limensional
number of
)) (Problem
:.

rocedure is

'eature with

ier from the
he criterion

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

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

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

- *Step III: Exclusion*
  1. $X'_k = X_{k+1} - \{x_r\}$; that is, remove x_r.
  2. $x_s = arg\max_{y \in X'_k} C(X'_k - \{y\})$; that is, find the least significant feature in the new set.
  3. If $C(X'_k - \{x_s\}) < C(X_{k-1})$ then $X_k = X'_k$ and go to step I; no further backward search is performed.
  4. Put $X'_{k-1} = X'_k - \{x_s\}$ and $k = k - 1$.
  5. If $k = 2$ put $X_k = X'_k$ and $C(X_k) = C(X'_k)$ and go to step I.
  6. Go to step III.

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

Optimal Searching Techniques

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

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

This property allows identifying the optimal combination but at a considerably reduced computational cost with respect to (5.22). Algorithms based on the *dynamic programming* concept (Chapter 8) offer one possibility to approaching the problem. A computationally more efficient way is to formulate the problem as a combinatorial optimization task and employ the so-called *branch and bound* methods to obtain the optimal solution [Lawe 66, Yu 93]. These methods compute the optimal value without involving exhaustive enumeration of all possible combinations. A more detailed description of the branch and bound methods is given in Chapter 15 and can also be found in [Fuku 90]. However, the complexity of these techniques is still higher than that of the previously mentioned suboptimal techniques.

A comparative study of various feature selection searching schemes can be found in [Kitt 78, Devi 82, Pudi 94, Jain 97]

5.7 OPTIMAL FEATURE GENERATION

So far, the class separability measuring criteria have been used in a rather "passive" way, that is, to measure the classification effectiveness of features generated in

some wa

way, that

point of

and the f

an m-dim

dimensio

We will c

where A

Obviousl

on the cho

criterion,

the same

class and

correspor

Thus, the

Our task

necessari

It can be

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

or

An experi

lem. It su

some way. In this section we will employ these measuring criteria in an "active" way, that is, as an integral part of the feature generation process itself. From this point of view, this section can be considered as a bridge between this chapter and the following one. Our major task can be summarized as follows: If $\boldsymbol{x}$ is an m-dimensional vector of measurement samples, transform it into another l-dimensional vector $\boldsymbol{y}$ so that an adopted class separability criterion is optimized. We will confine ourselves to linear transformations,

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

where A^T is an $l \times m$ matrix. Any of the criteria exposed so far can be used. Obviously, the degree of complexity of the optimization procedure depends heavily on the chosen criterion. We will demonstrate the method via the J_3 scattering matrix criterion, involving S_w and S_b matrices. Its optimization is straightforward and at the same time it has some interesting implications. Let S_{xw}, S_{xb} be the within-class and between-class scatter matrices of $\boldsymbol{x}$. From the respective definitions, the corresponding matrices of $\boldsymbol{y}$ become

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

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

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

b or m

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

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

It can be shown that (Problem 5.14)

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

or

$$(S_{xw}^{-1} S_{xb})A = A(S_{yw}^{-1} S_{yb}) \tag{5.23}$$

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

$$B^T S_{yw} B = I, \qquad B^T S_{yb} B = D \tag{5.24}$$

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

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

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

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

Combining (5.23) and (5.24), we finally obtain

$$(S_{xw}^{-1} S_{xb}) C = C D \tag{5.25}$$

where $C = AB$ is an $m \times l$ dimensional matrix. Equation (5.25) is a typical eigenvalue–eigenvector problem, with the diagonal matrix D having the eigenvalues of $S_{xw}^{-1} S_{xb}$ on its diagonal and C having the corresponding eigenvectors as its columns. However, $S_{xw}^{-1} S_{xb}$ is an $m \times m$ matrix, and the question is which l out of a total of m eigenvalues we must choose for the solution of (5.25). From its definition, matrix S_{xb} is of rank $M - 1$, where M is the number of classes (Problem 5.15). Thus, $S_{xw}^{-1} S_{xb}$ is also of rank $M - 1$ and there are $M - 1$ nonzero eigenvalues. Let us focus on the two possible alternatives separately.

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

  $$\hat{y} = C^T x \tag{5.26}$$

  This guarantees the maximum J_3 value. *As a matter of fact, in reducing the number of data from m to $M - 1$, there is no loss in class separability power, as this is measured by* J_3. Indeed,

  $$J_{3,x} = \text{trace}\{S_{xw}^{-1} S_{xb}\} = \lambda_1 + \cdots + \lambda_{M-1} + 0 \tag{5.27}$$

and

$$J_{3,\hat{y}} = \text{trace}\{(C^T S_{xw} C)^{-1}(C^T S_{xb} C)\} \tag{5.28}$$

Rearranging (5.25), we get

$$C^T S_{xb} C = C^T S_{xw} C D \tag{5.29}$$

Combining (5.28) and (5.29), we obtain

$$J_{3,\hat{y}} = \text{trace}\{D\} = \lambda_1 + \cdots + \lambda_{M-1} = J_{3,x} \tag{5.30}$$

It is most interesting to view this from a slightly different perspective. Let us recall the Bayesian classifier for an M class problem. Of the M conditional class probabilities, $P(\omega_i|\boldsymbol{x})$, $i = 1, 2, \ldots, M$, only $M - 1$ are independent, since they all add up to one. In general, $M - 1$ is the *minimum* number of discriminant functions needed for an M-class classification task (Problem 5.16). *Hence, the linear operation $C^T \boldsymbol{x}$, which computes the $M - 1$ components of $\hat{\boldsymbol{y}}$, can be seen as the optimal linear classifier, where optimality is with respect to J_3.* Therefore, this procedure can be viewed as a combination of the feature selection and classifier design stages, provided the classifier is a linear one. In Chapter 3 the optimal linear classifier was computed so as to minimize the mean (least) squares error. In this section it was designed to maximize J_3. From this point of view, this section can also be seen as a bridge with Chapter 3. This can be further strengthened by investigating the specific form that this classifier takes for the two-class problem. In this case, there is only one nonzero eigenvalue and it is not difficult to show (Problem 5.17) that

$$\hat{y} = (\mu_1 - \mu_2)^T S_{xw}^{-1} \boldsymbol{x}$$

The resulting linear classifier is also known as *Fisher's linear discriminant.* For Gaussian random vectors, with equal covariance matrices in both classes, this is nothing other than the optimal Bayesian classifier with the exception of a threshold value (Problem 2.11). Recall from Problem 3.14 that this is also directly related to the linear MSE classifier.

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

Remarks

- If J_3 is used with another combination of matrices, such as S_w and S_m, then, in general, the rank of the corresponding matrix product involved in the trace is m and there are m nonzero eigenvalues. In such cases the transformation matrix C is formed so that its columns are the eigenvectors corresponding to the l *largest eigenvalues*. According to (5.30), this guarantees the maximum value of J_3.
- A geometric interpretation of (5.26) reveals that $\hat{y}$ is the projection of the original vector $\boldsymbol{x}$ onto the subspace spanned by the eigenvectors $\boldsymbol{v}_i$ of $S_w^{-1}S_b$. It must be pointed out that these *are not* mutually orthogonal. Indeed, although matrices S_w, S_b (S_m) are symmetric, products of the form $S_w^{-1}S_b$ are not; thus, the eigenvectors are not mutually orthogonal (Problem 5.18). Furthermore, as we saw during the proof, once we decide on which subspace to project (by selecting the appropriate combination of eigenvectors) *the value of J_3 remains invariant under any linear transformation within this subspace*. That is, it is independent of the coordinate system and its value depends only on the particular subspace. In general, projection of the original feature vectors onto a lower dimensional subspace is associated with some information loss. An extreme example is shown in Figure 5.5, where the two classes coincide after projection on the v_1 axis. The choice of the subspace corresponding to the optimal J_3 value guarantees no loss of information for $l = M - 1$ (as this is measured by the J_3 criterion). Thus, this is a good choice, provided that J_3 is a good criterion for the problem of interest. Of course, this is not always the case; it depends on the specific classification task. A more extensive treatment of the topic, also involving other optimizing criteria, can be found in [Fuku 90].

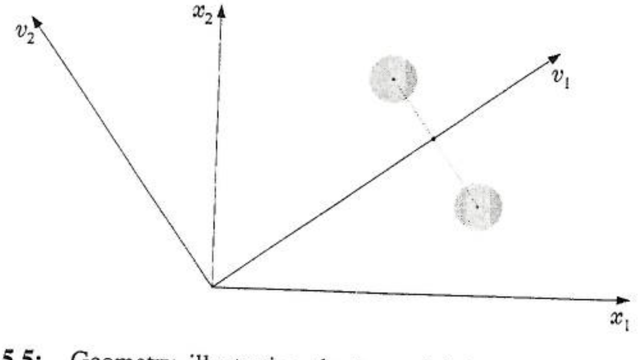


FIGURE 5.5: Geometry illustrating the loss of information associated with projections in lower dimensional subspaces.

- No doubt, scattering matrix criteria are not the only ones to compute the optimal transformation matrix. For example, [Wata 97] suggested using a different transformation matrix for each class and optimizing with respect to the classification error. This is within the spirit of the recent trend, to optimize directly with respect to the quantity of interest, which is the classification error probability. For the optimization, smooth versions of the error rate are used to guarantee differentiability. Other ways to compute the transformation matrix will be discussed in the next chapter.
- Besides linear ones, nonlinear transformations can also be employed for optimal feature selection. For example, in [Samm 69] a nonlinear technique is proposed that attempts to preserve maximally all the distances between vectors. Let $\boldsymbol{x}_i, \boldsymbol{y}_i, i = 1, 2, \ldots, N$, be the feature vectors in the original m-dimensional and the transformed l-dimensional space, respectively. The transformation into the lower dimensional space is performed so as to maximize

$$J = \frac{1}{\sum_{i=1}^{N-1} \sum_{j=i+1}^{N} d^o(i, j)} \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} \frac{(d^o(i, j) - d(i, j))^2}{d^o(i, j)} \qquad (5.31)$$

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

5.8 NEURAL NETWORKS AND FEATURE GENERATION/SELECTION

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

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

where the notation of the previous chapter has been adopted. Such a network has a unique minimum and the outputs of the hidden layer constitute the projection of the input m-dimensional space onto an l-dimensional subspace. In [Bour 88] it is shown that this is basically a projection onto the subspace spanned by the l principal eigenvectors of the input correlation matrix, a topic on which we will focus in the next chapter. An extension of this idea is to use three hidden layers [Kram 91]. Such a network performs a nonlinear principal component analysis. The major drawback of such an architecture is that nonlinear optimization techniques have to ence is given

comes

$\frac{\Sigma_i+(1-s)\Sigma_j}{2}$ |

$|^s| \Sigma_j |^{1-s}$

nce matrices
rattacharyya

etween-class

$[-1, 1]$.

ectors, with
probable, the

ward search

$^T S_1 A)^{-1}$

an M-class
n principle,

$^1 S_b$ is equal

5.18 Show that if matrices S_1, S_2 are two covariance matrices, then the eigenvectors of $S_1^{-1} S_2$ are orthogonal with respect to S_1, that is,

$$v_i^T S_1 v_j = \delta_{ij}$$

Hint: Use the fact that S_1, S_2 can be simultaneously diagonalized (Appendix B).

5.19 Show that in a multilayer perceptron with a linear output node, minimizing the squared error is equivalent with maximizing (5.32).
Hint: Assume the weights of the nonlinear nodes fixed and compute first the LS optimal weights driving the linear output nodes. Then substitute these values into the sum of error squares cost function.

References

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

[Riss 83] Ris
length,"

[Samm 69] S
actions o

[Seti 97] Seti
Neural N

[Su 94] Su K
projectio
Vol. 2(1),

[Swai 73] Sw
remote se
nition, pp

[Tou 74] Tou

[Yu 93] Yu B.,
Pattern R

[Walp 78] Wa
tists, Mac

[Wang 98] Wa
tissues fr
tions on I

[Wata 97] Wat
recognitio
1997.

[Vapn 82] Vap
Verlag, 19

[Vapn 95] Vap

tion, and small sam-
'achine Intelligence,

ɔgnition and Signal
ılphen aan den Rijn,

ing auto-associative

," Annals of Mathe-

ırvey," Operational

redictive minimum
networks," Neural

ı for nonparametric
, Vol. 23, pp. 433–

ır neural networks,"
7.

work optimization:
in Neural Systems,

Bayes decision in
ılysis and Machine

tion and multivari-
6(2), pp. 296–317,

ion for supervised
and Remote Sens-

s of generalization
Neural Computa-
San Mateo, C.A.,

rocesses, 3rd ed.,

feature selection,"

recognition: Rec-
'ysis and Machine

Springer-Verlag,

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

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

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

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

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

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

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

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

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

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

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

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