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

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

5.6 CLASS SEPARABILITY MEASURES

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

5.6.1 Divergence

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

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

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

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

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

Similar arguments hold for class ω_2, and we define

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

The sum

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

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

$$d_{ij} = D_{ij} + D_{ji}$$

$$= \int_{-\infty}^{+\infty} (p(\boldsymbol{x}|\omega_i) - p(\boldsymbol{x}|\omega_j)) \ln \frac{p(\boldsymbol{x}|\omega_i)}{p(\boldsymbol{x}|\omega_j)} d\boldsymbol{x} \tag{5.21}$$

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

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

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

$$d_{ij} \geq 0$$

$$d_{ij} = 0 \qquad \text{if } i = j$$

$$d_{ij} = d_{ji}$$

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

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

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

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

For the one-dimensional case this becomes

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

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

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

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

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

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

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

5.6.2 Chernoff Bound and Bhattacharyya Distance

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

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

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

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

Combining (5.23) and (5.24), we get

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

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

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

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

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

where

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

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

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

Example 5.4

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

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

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

$$P_e \leq 0.2225$$

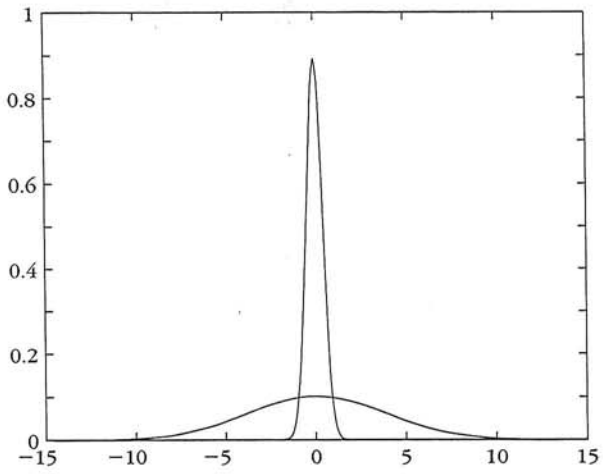

FIGURE 5.4
Gaussian pdfs with the same mean and different variances.

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

$$P_e \leq 0.0707$$

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

5.6.3 Scatter Matrices

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

Within-class scatter matrix

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

where Σ_i is the covariance matrix for class ω_i

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

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

Between-class scatter matrix

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

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

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

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

Mixture scatter matrix

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

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

$$S_m = S_w + S_b$$

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

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

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

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

A variant of J_2 commonly encountered in practice is

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

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. In practice, all three matrices are approximated by appropriate averaging using the available data samples.

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

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

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

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

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

Example 5.5

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

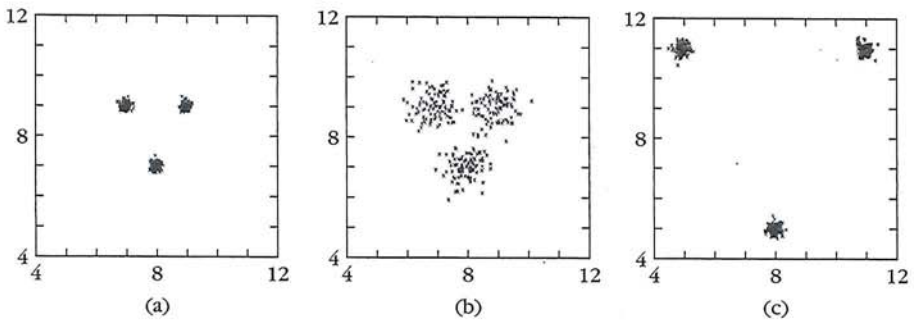

FIGURE 5.5

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

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

5.7 FEATURE SUBSET SELECTION

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

5.7.1 Scalar Feature Selection

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

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

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

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

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

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

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

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

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

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

  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

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

5.7.2 Feature Vector Selection

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

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

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

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

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

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

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

Suboptimal Searching Techniques

Sequential Backward Selection

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

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

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

Sequential Forward Selection

Here, the reverse to the preceding procedure is followed:

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

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

Floating Search Methods

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

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

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

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

Optimal Searching Techniques

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

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

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

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

Remark

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

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

5.8 OPTIMAL FEATURE GENERATION

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

The Two-class Case

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

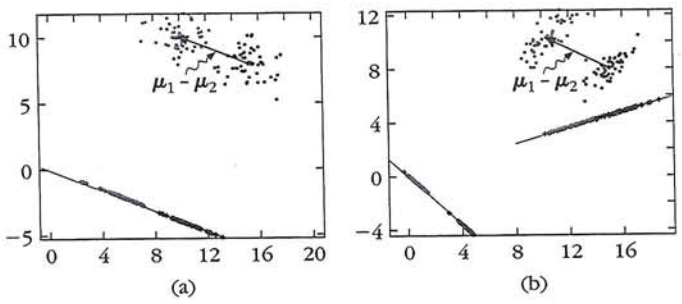

FIGURE 5.6

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

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

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

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

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

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

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

Multiclass Case

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

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

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

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

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

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

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

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

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

It can be shown that (Problem 5.17)

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

or

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Also

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

Rearranging (5.44), we get

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

Combining (5.47) and (5.48), we obtain

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

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

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

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

which is our familiar Fisher's linear discriminant.

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

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

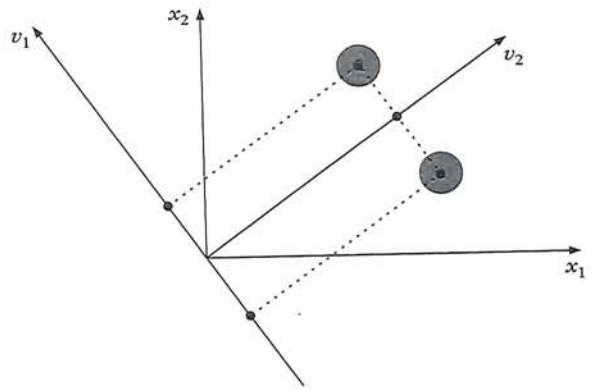

FIGURE 5.7

Geometry illustrating the loss of information associated with projections in lower dimensional subspaces. Projecting onto the direction of the principle eigenvector, v_1, there is no loss of information. Projection on the orthogonal direction results in a complete class overlap.

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

Remarks

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

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

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

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

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

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

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

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

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

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

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

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

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

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