# Chapter 5

# Linear Discriminant Functions

## 5.1 Introduction

In Chap. **??** we assumed that the forms for the underlying *probability densities* were known, and used the training samples to estimate the values of their parameters. In this chapter we shall instead assume we know the proper forms for the *discriminant functions*, and use the samples to estimate the values of parameters of the classifier. We shall examine various procedures for determining discriminant functions, some of which are statistical and some of which are not. None of them, however, requires knowledge of the forms of underlying probability distributions, and in this limited sense they can be said to be nonparametric.

Throughout this chapter we shall be concerned with discriminant functions that are either linear in the components of $\mathbf{x}$, or linear in some given set of functions of $\mathbf{x}$. Linear discriminant functions have a variety of pleasant analytical properties. As we have seen in Chap. **??**, they can be optimal if the underlying distributions are cooperative, such as Gaussians having equal covariance, as might be obtained through an intelligent choice of feature detectors. Even when they are not optimal, we might be willing to sacrifice some performance in order to gain the advantage of their simplicity. Linear discriminant functions are relatively easy to compute and in the absense of information suggesting otherwise, linear classifiers are an attractive candidates for initial, trial classifiers. They also illustrate a number of very important principles which will be used more fully in neural networks (Chap. **??**).

The problem of finding a linear discriminant function will be formulated as a problem of minimizing a criterion function. The obvious criterion function for classification purposes is the *sample risk*, or *training error* — the average loss incurred in classifying the set of training samples. We must emphasize right away, however, that despite the attractiveness of this criterion, it is fraught with problems. While our goal will be to classify novel test patterns, a small training error does not guarantee a small test error — a fascinating and subtle problem that will command our attention in Chap. **??**. As we shall see here, it is difficult to derive the minimum-risk linear discriminant anyway, and for that reason we investigate several related criterion functions that are analytically more tractable.

TRAINING ERROR

Much of our attention will be devoted to studying the convergence properties

and computational complexities of various gradient descent procedures for minimizing criterion functions. The similarities between many of the procedures sometimes makes it difficult to keep the differences between them clear and for this reason we have included a summary of the principal results in Table 5.1 at the end of Sect. 5.10.

## 5.2 Linear Discriminant Functions and Decision Surfaces

### 5.2.1 The Two-Category Case

A discriminant function that is a linear combination of the components of $\mathbf{x}$ can be written as

$$g(\mathbf{x}) = \mathbf{w}^t\mathbf{x} + w_0, \tag{1}$$

THRESHOLD WEIGHT

where $\mathbf{w}$ is the *weight vector* and $w_0$ the *bias* or *threshold weight*. A two-category linear classifier implements the following decision rule: Decide $\omega_1$ if $g(\mathbf{x}) > 0$ and $\omega_2$ if $g(\mathbf{x}) < 0$. Thus, $\mathbf{x}$ is assigned to $\omega_1$ if the inner product $\mathbf{w}^t\mathbf{x}$ exceeds the threshold $-w_0$ and $\omega_2$ otherwise. If $g(\mathbf{x}) = 0$, $\mathbf{x}$ can ordinarily be assigned to either class, but in this chapter we shall leave the assignment undefined. Figure 5.1 shows a typical implementation, a clear example of the general structure of a pattern recognition system we saw in Chap. **??**.

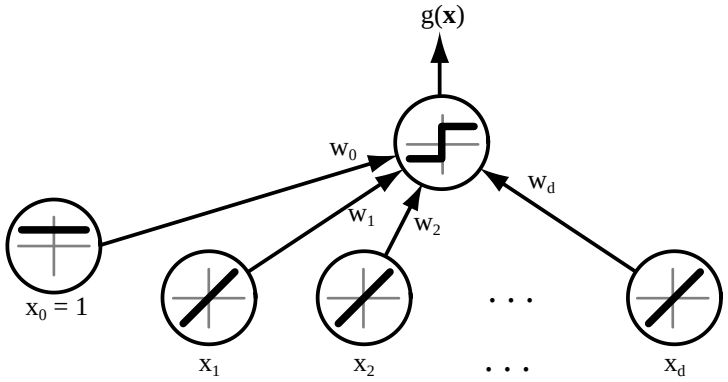


Figure 5.1: A simple linear classifier having $d$ input units, each corresponding to the values of the components of an input vector. Each input feature value $x_i$ is multiplied by its corresponding weight $w_i$; the output unit sums all these products and emits a $+1$ if $\mathbf{w}^t\mathbf{x} + w_0 > 0$ or a $-1$ otherwise.

The equation $g(\mathbf{x}) = 0$ defines the decision surface that separates points assigned to $\omega_1$ from points assigned to $\omega_2$. When g($\mathbf{x}$) is linear, this decision surface is a *hyperplane*. If $\mathbf{x}_1$ and $\mathbf{x}_2$ are both on the decision surface, then

$$\mathbf{w}^t\mathbf{x}_1 + w_0 = \mathbf{w}^t\mathbf{x}_2 + w_0$$

or

$$\mathbf{w}^t(\mathbf{x}_1 - \mathbf{x}_2) = 0,$$

and this shows that $\mathbf{w}$ is normal to any vector lying in the hyperplane. In general, the hyperplane $H$ divides the feature space into two halfspaces, decision region $\mathcal{R}_1$ for $\omega_1$ and region $\mathcal{R}_2$ for $\omega_2$. Since $g(\mathbf{x}) > 0$ if $\mathbf{x}$ is in $\mathcal{R}_1$, it follows that the normal vector $\mathbf{w}$ points into $\mathcal{R}_1$. It is sometimes said that any $\mathbf{x}$ in $\mathcal{R}_1$ is on the *positive* side of $H$, and any $\mathbf{x}$ in $\mathcal{R}_2$ is on the *negative* side.

The discriminant function g($\mathbf{x}$) gives an algebraic measure of the distance from $\mathbf{x}$ to the hyperplane. Perhaps the easiest way to see this is to express $\mathbf{x}$ as

$$\mathbf{x} = \mathbf{x}_p + r\frac{\mathbf{w}}{\|\mathbf{w}\|},$$

where $\mathbf{x}_p$ is the normal projection of $\mathbf{x}$ onto $H$, and $r$ is the desired algebraic distance — positive if $\mathbf{x}$ is on the positive side and negative if $\mathbf{x}$ is on the negative side. Then, since $g(\mathbf{x}_p) = 0$,

$$g(\mathbf{x}) = \mathbf{w}^t\mathbf{x} + w_0 = r\|\mathbf{w}\|,$$

or

$$r = \frac{g(\mathbf{x})}{\|\mathbf{w}\|}.$$

In particular, the distance from the origin to $H$ is given by $w_0/\|\mathbf{w}\|$. If $w_0 > 0$ the origin is on the positive side of $H$, and if $w_0 < 0$ it is on the negative side. If $w_0 = 0$, then $g(\mathbf{x})$ has the homogeneous form $\mathbf{w}^t\mathbf{x}$, and the hyperplane passes through the origin. A geometric illustration of these algebraic results is given in Fig. 5.2.

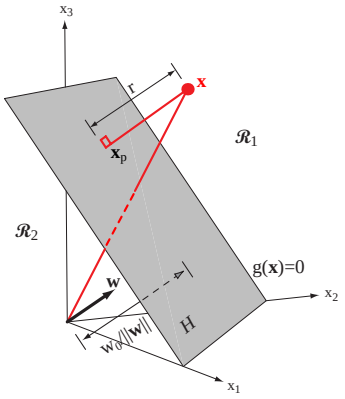


Figure 5.2: The linear decision boundary $H$, where $g(\mathbf{x}) = \mathbf{w}^t\mathbf{x} + w_0 = 0$, separates the feature space into two half-spaces $\mathcal{R}_1$ (where $g(\mathbf{x}) > 0$) and $\mathcal{R}_2$ (where $g(\mathbf{x}) < 0$).

To summarize, a linear discriminant function divides the feature space by a hyperplane decision surface. The orientation of the surface is determined by the normal vector $\mathbf{w}$, and the location of the surface is determined by the bias $w_0$. The discriminant function $g(\mathbf{x})$ is proportional to the signed distance from $\mathbf{x}$ to the hyperplane, with $g(\mathbf{x}) > 0$ when $\mathbf{x}$ is on the positive side, and $g(\mathbf{x}) < 0$ when $\mathbf{x}$ is on the negative side.

### 5.2.2 The Multicategory Case

There is more than one way to devise multicategory classifiers employing linear discriminant functions. For example, we might reduce the problem to $c-1$ two-class problems, where the $i$th problem is solved by a linear discriminant function that separates points assigned to $\omega_i$ from those not assigned to $\omega_i$. A more extravagant approach would be to use $c(c-1)/2$ linear discriminants, one for every pair of classes. As illustrated in Fig. 5.3, both of these approaches can lead to regions in which the classification is undefined. We shall avoid this problem by adopting the approach taken in Chap. **??**, defining $c$ linear discriminant functions

$$g_i(\mathbf{x}) = \mathbf{w}^t\mathbf{x}_i + w_{i0} \qquad i = 1, ..., c, \tag{2}$$

and assigning $\mathbf{x}$ to $\omega_i$ if $g_i(\mathbf{x}) > g_j(\mathbf{x})$ for all $j \neq i$; in case of ties, the classification is left undefined. The resulting classifier is called a *linear machine*. A linear machine divides the feature space into $c$ decision regions, with $g_i(\mathbf{x})$ being the largest discriminant if $\mathbf{x}$ is in region $\mathcal{R}_i$. If $\mathcal{R}_i$ and $\mathcal{R}_j$ are contiguous, the boundary between them is a portion of the hyperplane $H_{ij}$ defined by

LINEAR MACHINE

$$g_i(\mathbf{x}) = g_j(\mathbf{x})$$

or

$$(\mathbf{w}_i - \mathbf{w}_j)^t\mathbf{x} + (w_{i0} - w_{j0}) = 0.$$

It follows at once that $\mathbf{w}_i - \mathbf{w}_j$ is normal to $H_{ij}$, and the signed distance from $\mathbf{x}$ to $H_{ij}$ is given by $(g_i - g_j)/\|\mathbf{w}_i - \mathbf{w}_j\|$. Thus, with the linear machine it is not the weight vectors themselves but their *differences* that are important. While there are $c(c-1)/2$ pairs of regions, they need not all be contiguous, and the total number of hyperplane segments appearing in the decision surfaces is often fewer than $c(c-1)/2$, as shown in Fig. 5.4.

It is easy to show that the decision regions for a linear machine are convex and this restriction surely limits the flexibility and accuracy of the classifier (Problems 1 & 2). In particular, for good performance every decision region should be singly connected, and this tends to make the linear machine most suitable for problems for which the conditional densities $p(\mathbf{x}|\omega_i)$ are unimodal.

## 5.3 Generalized Linear Discriminant Functions

The linear discriminant function $g(\mathbf{x})$ can be written as

$$g(\mathbf{x}) = w_0 + \sum_{i=1}^{d} w_i x_i, \tag{3}$$

where the coefficients $w_i$ are the components of the weight vector $\mathbf{w}$. By adding additional terms involving the products of pairs of components of $\mathbf{x}$, we obtain the *quadratic discriminant function*

QUADRATIC DISCRIMINANT

$$g(\mathbf{x}) = w_0 + \sum_{i=1}^{d} w_i x_i + \sum_{i=1}^{d}\sum_{j=1}^{d} w_{ij} x_i x_j. \tag{4}$$

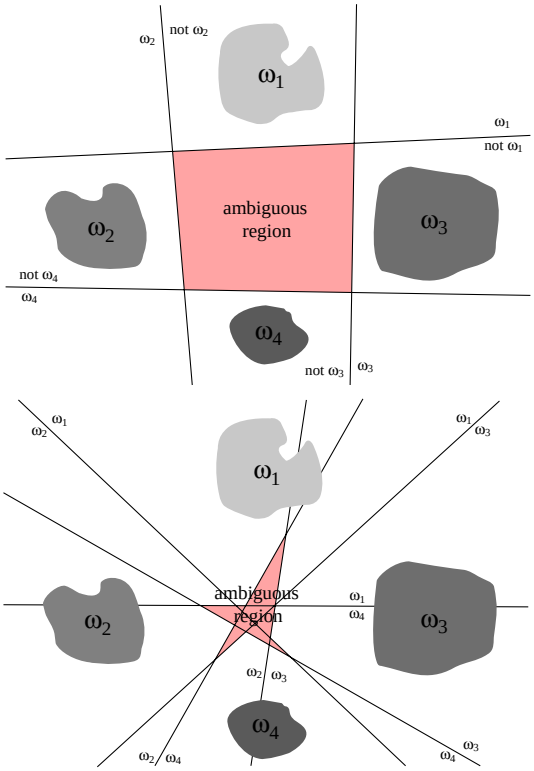


Figure 5.3: Linear decision boundaries for a four-class problem. The top figure shows $\omega_i$/not $\omega_i$ dichotomies while the bottom figure shows $\omega_i/\omega_j$ dichotomies. The pink regions have ambiguous category assigments.

Since $x_i x_j = x_j x_i$, we can assume that $w_{ij} = w_{ji}$ with no loss in generality. Thus, the quadratic discriminant function has an additional $d(d+1)/2$ coefficients at its disposal with which to produce more complicated separating surfaces. The separating surface defined by $g(\mathbf{x}) = 0$ is a second-degree or *hyperquadric* surface. The linear terms in g($\mathbf{x}$) can be eliminated by translating the axes. We can define $\mathbf{W} = [w_{ij}]$, a symmetric, nonsingular matrix and then the basic character of the separating surface can be described in terms of the scaled matrix $\bar{\mathbf{W}} = \mathbf{W}/(\mathbf{w}^t\mathbf{W}^{-1}\mathbf{w} - 4w_0)$. If $\bar{\mathbf{W}}$ is a positive multiple of the identity matrix, the separating surface is a *hypersphere*. If $\bar{\mathbf{W}}$ is positive definite, the separating surfaces is a *hyperellipsoid*. If some of the eigenvalues of $\bar{\mathbf{W}}$ are positive and others are negative, the surface is one of the variety of types of *hyperhyperboloids* (Problem 11). As we observed in Chap. **??**, these are the kinds of separating surfaces that arise in the general multivariate Gaussian case.

By continuing to add terms such as $w_{ijk}x_i x_j x_k$ we can obtain the class of *polynomial discriminant functions*. These can be thought of as truncated series expansions of some arbitrary g($\mathbf{x}$), and this in turn suggest the *generalized linear discriminant function* POLYNOMIAL DISCRIMINANT

$$g(\mathbf{x}) = \sum_{i=1}^{\hat{d}} a_i y_i(\mathbf{x}) \tag{5}$$

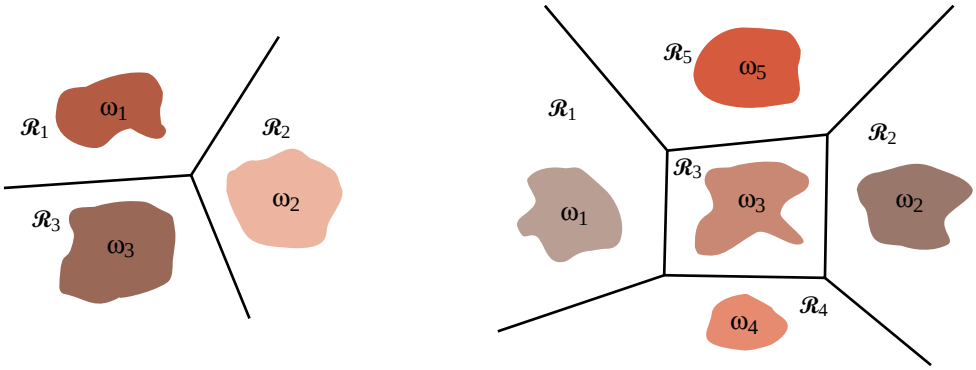


Figure 5.4: Decision boundaries produced by a linear machine for a three-class problem and a five-class problem.

or

$$g(\mathbf{x}) = \mathbf{a}^t\mathbf{y}, \tag{6}$$

PHI FUNCTION

where $\mathbf{a}$ is now a $\hat{d}$-dimensional weight vector, and where the $\hat{d}$ functions $y_i(\mathbf{x})$ — sometimes called $\varphi$ functions — can be arbitrary functions of $\mathbf{x}$. Such functions might be computed by a feature detecting subsystem. By selecting these functions judiciously and letting $\hat{d}$ be sufficiently large, one can approximate any desired discriminant function by such an expansion. The resulting discriminant function is not linear in $\mathbf{x}$, but it is linear in $\mathbf{y}$. The $\hat{d}$ functions $y_i(\mathbf{x})$ merely map points in $d$-dimenional $\mathbf{x}$-space to points in $\hat{d}$-dimensional $\mathbf{y}$-space. The homogeneous discriminant $\mathbf{a}^t\mathbf{y}$ separates points in this transformed space by a hyperplane passing through the origin. Thus, the mapping from $\mathbf{x}$ to $\mathbf{y}$ reduces the problem to one of finding a homogeneous linear discriminant function.

Some of the advantages and disadvantages of this approach can be clarified by considering a simple example. Let the quadratic discriminant function be

$$g(x) = a_1 + a_2x + a_3x^2, \tag{7}$$

so that the three-dimensional vector $\mathbf{y}$ is given by

$$\mathbf{y} = \begin{pmatrix} 1 \\ x \\ x^2 \end{pmatrix}. \tag{8}$$

The mapping from $x$ to $\mathbf{y}$ is illustrated in Fig. 5.5. The data remain inherently one-dimensional, since varying $x$ causes $\mathbf{y}$ to trace out a curve in three dimensions. Thus, one thing to notice immediately is that if $x$ is governed by a probability law $p(x)$, the induced density $\hat{p}(\mathbf{y})$ will be degenerate, being zero everywhere except on the curve, where it is infinite. This is a common problem whenever $\hat{d} > d$, and the mapping takes points from a lower-dimensional space to a higher-dimensional space.

The plane $\hat{H}$ defined by $\mathbf{a}^t\mathbf{y} = 0$ divides the $\mathbf{y}$-space into two decision regions $\hat{\mathcal{R}}_1$ and $\hat{\mathcal{R}}_2$. Figure **??** shows the separating plane corresponding to $\mathbf{a} = (-1, 1, 2)^t$, the decision regions $\hat{\mathcal{R}}_1$ and $\hat{\mathcal{R}}_2$, and their corresponding decision regions $\mathcal{R}_1$ and $\mathcal{R}_2$ in the original $x$-space. The quadratic discriminant function $g(x) = -1 + x + 2x^2$ is

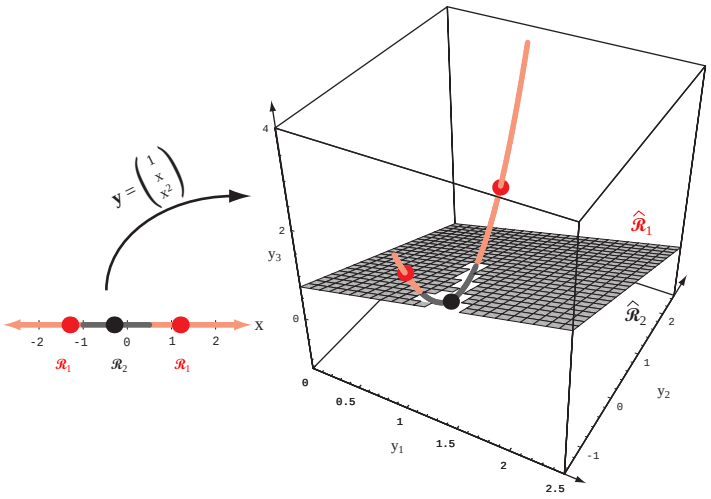


Figure 5.5: The mapping $\mathbf{y} = (1, x, x^2)^t$ takes a line and transforms it to a parabola in three dimensions. A plane splits the resulting $\mathbf{y}$ space into regions corresponding to two categories, and this in turn gives a non-simply connected decision region in the one-dimensional $x$ space.

positive if $x < -1$ or if $x > 0.5$, and thus $\mathcal{R}_1$ is multiply connected. Thus although the decision regions in $\mathbf{y}$-space are convex, this is by no means the case in $x$-space. More generally speaking, even with relatively simple functions $y_i(\mathbf{x})$, decision surfaces induced in an $\mathbf{x}$-space can be fairly complex (Fig. 5.6).

Unfortunately, the curse of dimensionality often makes it hard to capitalize on this flexibility in practice. A complete quadratic discriminant function involves $\hat{d} = (d\ +\ 1)(d\ +\ 2)/2$ terms. If $d$ is modestly large, say $d = 50$, this requires the computation of a great many terms; inclusion of cubic and higher orders leads to $O(\hat{d}^3)$ terms. Furthermore, the $\hat{d}$ components of the weight vector $\mathbf{a}$ must be determined from training samples. If we think of $\hat{d}$ as specifying the number of degrees of freedom for the discriminant function, it is natural to require that the number of samples be not less than the number of degrees of freedom (cf., Chap. **??**). Clearly, a general series expansion of $g(\mathbf{x})$ can easily lead to completely unrealistic requirements for computation and data. We shall see in Sect. **??** that this drawback can be accommodated by imposing a constraint of large margins, or bands between the training patterns, however. In this case, we are not technically speaking fitting all the free parameters; instead, we are relying on the assumption that the mapping to a high-dimensional space does not impose any spurious structure or relationships among the training points. Alternatively, multilayer neural networks approach this problem by employing multiple copies of a single nonlinear function of the input features, as we shall see in Chap. **??**.

While it may be hard to realize the potential benefits of a generalized linear discriminant function, we can at least exploit the convenience of being able to write $g(\mathbf{x})$ in the homogeneous form $\mathbf{a}^t\mathbf{y}$. In the particular case of the linear discriminant function

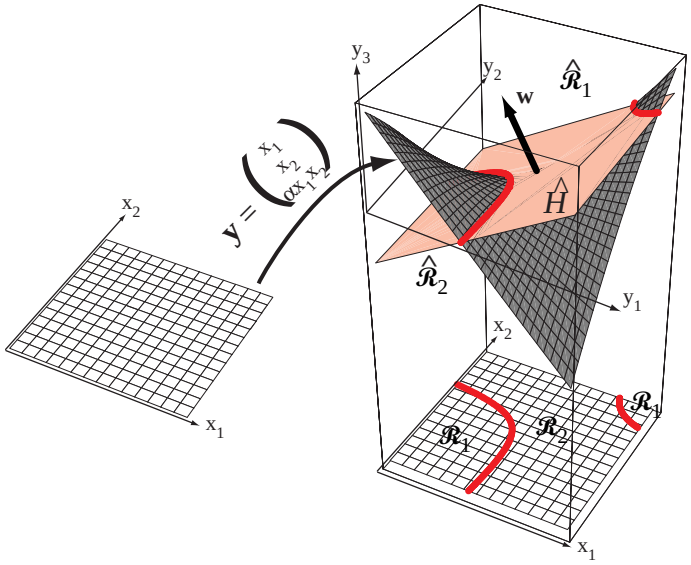


Figure 5.6: The two-dimensional input space $\mathbf{x}$ is mapped through a polynomial function $f$ to $\mathbf{y}$. Here the mapping is $y_1 = x_1$, $y_2 = x_2$ and $y_3 \propto x_1 x_2$. A linear discriminant in this transformed space is a hyperplane, which cuts the surface. Points to the positive side of the hyperplane $\hat{H}$ correspond to category $\omega_1$, and those beneath it $\omega_2$. Here, in terms of the $\mathbf{x}$ space, $\mathcal{R}_1$ is a not simply connected.

$$g(\mathbf{x}) = w_0 + \sum_{i=1}^{d} w_i x_i = \sum_{i=0}^{d} w_i x_i \tag{9}$$

where we set $x_0 = 1$. Thus we can write

$$\mathbf{y} = \begin{bmatrix} 1 \\ x_1 \\ \vdots \\ x_d \end{bmatrix} = \begin{bmatrix} 1 \\ \\ \mathbf{x} \\ \\ \end{bmatrix}, \tag{10}$$

AUGMENTED VECTOR

and $\mathbf{y}$ is sometimes called an *augmented feature vector*. Likewise, an *augmented weight vector* can be written as:

$$\mathbf{a} = \begin{bmatrix} w_0 \\ w_1 \\ \vdots \\ w_d \end{bmatrix} = \begin{bmatrix} w_0 \\ \\ \mathbf{w} \\ \\ \end{bmatrix}. \tag{11}$$

This mapping from $d$-dimensional $\mathbf{x}$-space to $(d+1)$-dimensional $\mathbf{y}$-space is mathematically trivial but nonetheless quite convenient. The addition of a constant component to $\mathbf{x}$ preserves all distance relationships among samples. The resulting $\mathbf{y}$ vectors all lie in a $d$-dimensional subspace, which is the $\mathbf{x}$-space itself. The hyperplane decision surface $\hat{H}$ defined by $\mathbf{a}^t\mathbf{y} = 0$ passes through the origin in $\mathbf{y}$-space, even though the corresponding hyperplane $H$ can be in any position in $\mathbf{x}$-space. The distance from $\mathbf{y}$ to $\hat{H}$ is given by $|\mathbf{a}^t\mathbf{y}|/\|\mathbf{a}\|$, or $|g(\mathbf{x})|/\|\mathbf{a}\|$. Since $\|\mathbf{a}\| > \|\mathbf{w}\|$, this distance is less

than, or at most equal to the distance from $\mathbf{x}$ to $H$. By using this mapping we reduce the problem of finding a weight vector $\mathbf{w}$ and a threshold weight $w_0$ to the problem of finding a single weight vector $\mathbf{a}$ (Fig. 5.7).

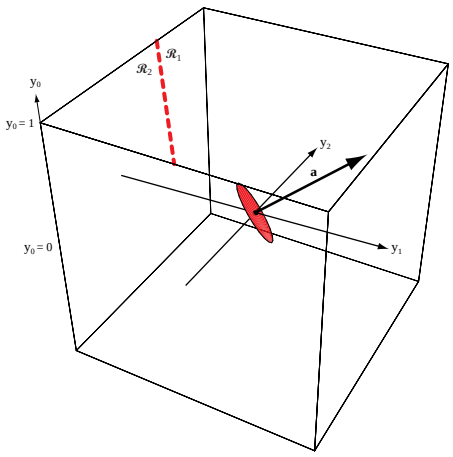


Figure 5.7: A three-dimensional augmented feature space $\mathbf{y}$ and augmented weight vector $\mathbf{a}$ (at the origin). The set of points for which $\mathbf{a}^t\mathbf{y} = 0$ is a plane (or more generally, a hyperplane) perpendicular to $\mathbf{a}$ and passing through the origin of $\mathbf{y}$-space, as indicated by the red disk. Such a plane need not pass through the origin of the two-dimensional $\mathbf{x}$-space at the top, of course, as shown by the dashed line. Thus there exists an augmented weight vector $\mathbf{a}$ that will lead to any straight decision line in $\mathbf{x}$-space.

## 5.4 The Two-Category Linearly-Separable Case

### 5.4.1 Geometry and Terminology

Suppose now that we have a set of $n$ samples $\mathbf{y}_1, ..., \mathbf{y}_n$, some labelled $\omega_1$ and some labelled $\omega_2$. We want to use these samples to determine the weights $\mathbf{a}$ in a linear discriminant function $g(\mathbf{x}) = \mathbf{a}^t\mathbf{y}$. Suppose we have reason to believe that there exists a solution for which the probability of error is very low. Then a reasonable approach is to look for a weight vector that classifies all of the samples correctly. If such a weight vector exists, the samples are said to be *linearly separable*.

LINEARLY SEPARABLE

A sample $\mathbf{y}_i$ is classified correctly if $\mathbf{a}^t\mathbf{y}_i > 0$ and $\mathbf{y}_i$ is labelled $\omega_1$ or if $\mathbf{a}^t\mathbf{y}_i < 0$ and $\mathbf{y}_i$ is labelled $\omega_2$. This suggests a "normalization" that simplifies the treatment of the two-category case, viz., the replacement of all samples labelled $\omega_2$ by their negatives. With this "normalization" we can forget the labels and look for a weight vector $\mathbf{a}$ such that $\mathbf{a}^t\mathbf{y}_i > 0$ for *all* of the samples. Such a weight vector is called a *separating vector* or more generally a *solution vector*.

SEPARATING VECTOR

The weight vector $\mathbf{a}$ can be thought of as specifying a point in *weight space*. Each sample $\mathbf{y}_i$ places a constraint on the possible location of a solution vector. The equation $\mathbf{a}^t\mathbf{y}_i = 0$ defines a hyperplane through the origin of weight space having $\mathbf{y}_i$ as a normal vector. The solution vector — if it exists — must be on the positive side

of every hyperplane. Thus, a solution vector must lie in the intersection of $n$ halfspaces; indeed any vector in this region is a solution vector. The corresponding region is called the *solution region*, and should not be confused with the decision region in feature space corresponding to any particular category. A two-dimensional example illustrating the solution region for both the normalized and the unnormalized case is shown in Fig. 5.8.

SOLUTION REGION

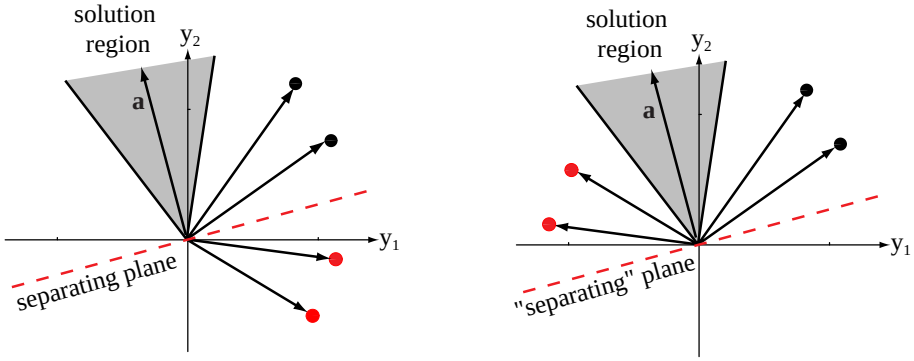


Figure 5.8: Four training samples (black for $\omega_1$, red for $\omega_2$) and the solution region in feature space. The figure on the left shows the raw data; the solution vectors leads to a plane that separates the patterns from the two categories. In the figure on the right, the red points have been "normalized" — i.e., changed in sign. Now the solution vector leads to a plane that places all "normalized" points on the same side.

From this discussion, it should be clear that the solution vector — again, if it exists — is not unique. There are several ways to impose additional requirements to constrain the solution vector. One possibility is to seek a unit-length weight vector that maximizes the minimum distance from the samples to the separating plane. Another possibility is to seek the minimum-length weight vector satisfying $\mathbf{a}^t\mathbf{y}_i \geq b$ for all $i$, where $b$ is a positive constant called the *margin*. As shown in Fig. 5.9, the solution region resulting form the intersections of the halfspaces for which $\mathbf{a}^t\mathbf{y}_i \geq b > 0$ lies within the previous solution region, being insultated from the old boundaries by the distance $b/\|\mathbf{y}_i\|$.

MARGIN

The motivation behind these attempts to find a solution vector closer to the "middle" of the solution region is the natural belief that the resulting solution is more likely to classify new test samples correctly. In most of the cases we shall treat, however, we shall be satisfied with any solution strictly within the solution region. Our chief concern will be to see that any iterative procedure used does not converge to a limit point on the boundary. This problem can always be avoided by the introduction of a margin, i.e., by requiring that $\mathbf{a}^t\mathbf{y}_i \geq b > 0$ for all $i$.

### 5.4.2 Gradient Descent Procedures

The approach we shall take to finding a solution to the set of linear inequalities $\mathbf{a}^t\mathbf{y}_i > 0$ will be to define a criterion function $J(\mathbf{a})$ that is minimized if $\mathbf{a}$ is a solution vector. This reduces our problem to one of minimizing a scalar function — a problem that can often be solved by a gradient descent procedure. Basic gradient descent is very simple. We start with some arbitrarily chosen weight vector $\mathbf{a}(1)$ and compute the gradient vector $\nabla J(\mathbf{a}(1))$. The next value $\mathbf{a}(2)$ is obtained by moving some

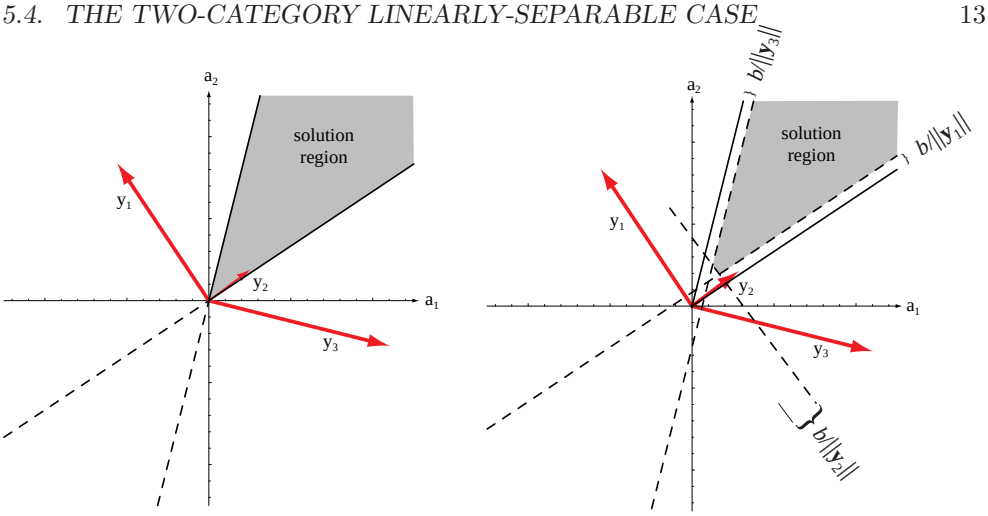


Figure 5.9: The effect of the margin on the solution region. At the left, the case of no margin ($b = 0$) equivalent to a case such as shown at the left in Fig. 5.8. At the right is the case $b > 0$, shrinking the solution region by margins $b/\|\mathbf{y}_i\|$.

distance from $\mathbf{a}(1)$ in the direction of steepest descent, i.e., along the negative of the gradient. In general, $\mathbf{a}(k+1)$ is obtained from $\mathbf{a}(k)$ by the equation

$$\mathbf{a}(k+1) = \mathbf{a}(k) - \eta(k)\boldsymbol{\nabla} J(\mathbf{a}(k)), \tag{12}$$

where $\eta$ is a positive scale factor or *learning rate* that sets the step size. We hope that such a sequence of weight vectors will converge to a solution minimizing $J(\mathbf{a})$. In algorithmic form we have: LEARNING RATE

**Algorithm 1 (Basic gradient descent)**

```
1 begin initialize  a, criterion θ, η(·), k = 0
2   do k ← k + 1
3       a ← a − η(k)∇J(a)
4   until η(k)∇J(a) < θ
5 return a
6 end
```

The many problems associated with gradient descent procedures are well known. Fortunately, we shall be constructing the functions we want to minimize, and shall be able to avoid the most serious of these problems. One that will confront us repeatedly, however, is the choice of the learning rate $\eta(k)$. If $\eta(k)$ is too small, convergence is needlessly slow, whereas if $\eta(k)$ is too large, the correction process will overshoot and can even diverge (Sect. 5.6.1).

We now consider a principled method for setting the learning rate. Suppose that the criterion function can be well approximated by the second-order expansion around a value $\mathbf{a}(k)$ as

$$J(\mathbf{a}) \simeq J(\mathbf{a}(k)) + \boldsymbol{\nabla} J^t(\mathbf{a} - \mathbf{a}(k)) + \frac{1}{2}(\mathbf{a} - \mathbf{a}(k))^t \mathbf{H}\,(\mathbf{a} - \mathbf{a}(k)), \tag{13}$$

where $\mathbf{H}$ is the *Hessian matrix* of second partial derivatives $\partial^2 J/\partial a_i \partial a_j$ evaluated at $\mathbf{a}(k)$. Then, substituting $\mathbf{a}(k+1)$ from Eq. 12 into Eq. 13 we find: HESSIAN MATRIX

$$J(\mathbf{a}(k+1)) \simeq J(\mathbf{a}(k)) - \eta(k)\|\boldsymbol{\nabla} J\|^2 + \frac{1}{2}\eta^2(k)\boldsymbol{\nabla} J^t \mathbf{H} \boldsymbol{\nabla} J.$$

From this it follows (Problem 12) that $J(\mathbf{a}(k+1))$ can be minimized by the choice

$$\eta(k) = \frac{\|\boldsymbol{\nabla} J\|^2}{\boldsymbol{\nabla} J^t \mathbf{H} \boldsymbol{\nabla} J}, \tag{14}$$

where $\mathbf{H}$ depends on $\mathbf{a}$, and thus indirectly on $k$. This then is the optimal choice of $\eta(k)$ given the assumptions mentioned. Note that if the criterion function $J(\mathbf{a})$ is quadratic throughout the region of interest, then $\mathbf{H}$ is constant and $\eta$ is a constant independent of $k$.

NEWTON'S ALGORITHM

An alternative approach, obtained by ignoring Eq. 12 and by choosing $\mathbf{a}(k+1)$ to minimize the second-order expansion, is *Newton's algorithm* where line 3 in Algorithm 1 is replaced by

$$\mathbf{a}(k+1) = \mathbf{a}(k) - \mathbf{H}^{-1}\boldsymbol{\nabla} J, \tag{15}$$

leading to the following algorithm:

**Algorithm 2 (Newton descent)**

```
1 begin initialize  a, criterion θ
2         do
3             a ← a − H^{-1}∇J(a)
4         until H^{-1}∇J(a) < θ
5   return a
6 end
```

Simple gradient descent and Newton's algorithm are compared in Fig. 5.10.

Generally speaking, Newton's algorithm will usually give a greater improvement *per step* than the simple gradient descent algorithm, even with the optimal value of $\eta(k)$. However, Newton's algorithm is not applicable if the Hessian matrix $\mathbf{H}$ is singular. Furthermore, even when $\mathbf{H}$ is nonsingular, the $O(d^3)$ time required for matrix inversion on each iteration can easily offset the descent advantage. In fact, it often takes less time to set $\eta(k)$ to a constant $\eta$ that is smaller than necessary and make a few more corrections than it is to compute the optimal $\eta(k)$ at each step (Computer exercise 1).

## 5.5 Minimizing the Perceptron Criterion Function

### 5.5.1 The Perceptron Criterion Function

Consider now the problem of constructing a criterion function for solving the linear inequalities $\mathbf{a}^t\mathbf{y}_i > 0$. The most obvious choice is to let $J(\mathbf{a};\ \mathbf{y}_1, ..., \mathbf{y}_n)$ be the number of samples misclassified by $\mathbf{a}$. However, because this function is piecewise constant, it is obviously a poor candidate for a gradient search. A better choice is the *Perceptron criterion function*

$$J_p(\mathbf{a}) = \sum_{\mathbf{y}\in\mathcal{Y}} (-\mathbf{a}^t\mathbf{y}), \tag{16}$$