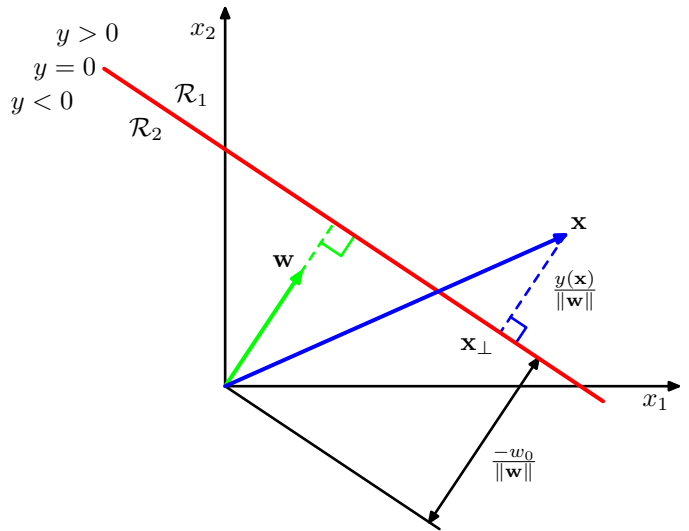


Figure 4.1 Illustration of the geometry of a linear discriminant function in two dimensions. The decision surface, shown in red, is perpendicular to \mathbf{w} , and its displacement from the origin is controlled by the bias parameter w_0 . Also, the signed orthogonal distance of a general point \mathbf{x} from the decision surface is given by $y(\mathbf{x})/\|\mathbf{w}\|$.



an arbitrary point \mathbf{x} and let \mathbf{x}_\perp be its orthogonal projection onto the decision surface, so that

$$\mathbf{x} = \mathbf{x}_\perp + r \frac{\mathbf{w}}{\|\mathbf{w}\|}. \quad (4.6)$$

Multiplying both sides of this result by \mathbf{w}^T and adding w_0 , and making use of $y(\mathbf{x}) = \mathbf{w}^T \mathbf{x} + w_0$ and $y(\mathbf{x}_\perp) = \mathbf{w}^T \mathbf{x}_\perp + w_0 = 0$, we have

$$r = \frac{y(\mathbf{x})}{\|\mathbf{w}\|}. \quad (4.7)$$

This result is illustrated in Figure 4.1.

As with the linear regression models in Chapter 3, it is sometimes convenient to use a more compact notation in which we introduce an additional dummy ‘input’ value $x_0 = 1$ and then define $\tilde{\mathbf{w}} = (w_0, \mathbf{w})$ and $\tilde{\mathbf{x}} = (x_0, \mathbf{x})$ so that

$$y(\mathbf{x}) = \tilde{\mathbf{w}}^T \tilde{\mathbf{x}}. \quad (4.8)$$

In this case, the decision surfaces are D -dimensional hyperplanes passing through the origin of the $D + 1$ -dimensional expanded input space.

4.1.2 Multiple classes

Now consider the extension of linear discriminants to $K > 2$ classes. We might be tempted to build a K -class discriminant by combining a number of two-class discriminant functions. However, this leads to some serious difficulties (Duda and Hart, 1973) as we now show.

Consider the use of $K - 1$ classifiers each of which solves a two-class problem of separating points in a particular class \mathcal{C}_k from points not in that class. This is known as a *one-versus-the-rest* classifier. The left-hand example in Figure 4.2 shows an

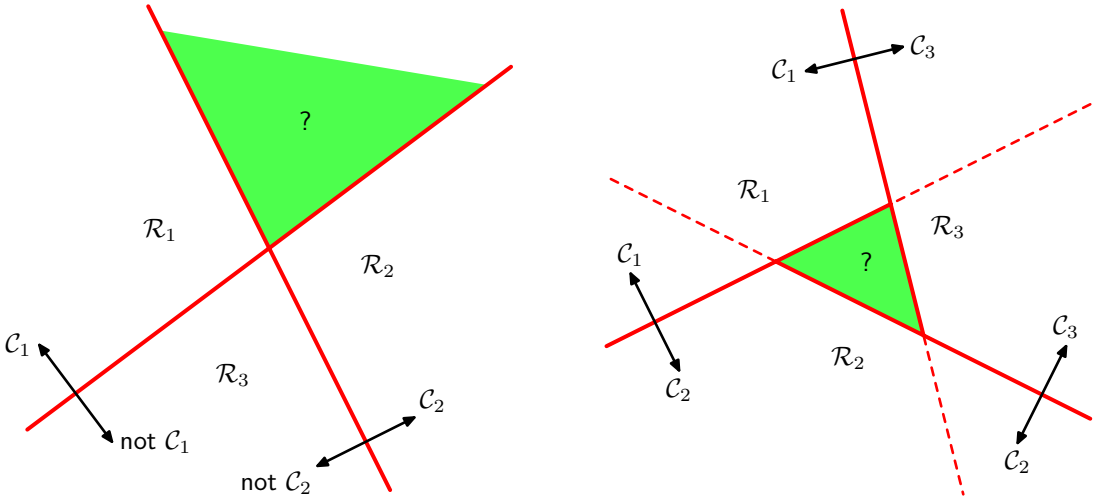


Figure 4.2 Attempting to construct a K class discriminant from a set of two class discriminants leads to ambiguous regions, shown in green. On the left is an example involving the use of two discriminants designed to distinguish points in class \mathcal{C}_k from points not in class \mathcal{C}_k . On the right is an example involving three discriminant functions each of which is used to separate a pair of classes \mathcal{C}_k and \mathcal{C}_j .

example involving three classes where this approach leads to regions of input space that are ambiguously classified.

An alternative is to introduce $K(K-1)/2$ binary discriminant functions, one for every possible pair of classes. This is known as a *one-versus-one* classifier. Each point is then classified according to a majority vote amongst the discriminant functions. However, this too runs into the problem of ambiguous regions, as illustrated in the right-hand diagram of Figure 4.2.

We can avoid these difficulties by considering a single K -class discriminant comprising K linear functions of the form

$$y_k(\mathbf{x}) = \mathbf{w}_k^T \mathbf{x} + w_{k0} \quad (4.9)$$

and then assigning a point \mathbf{x} to class \mathcal{C}_k if $y_k(\mathbf{x}) > y_j(\mathbf{x})$ for all $j \neq k$. The decision boundary between class \mathcal{C}_k and class \mathcal{C}_j is therefore given by $y_k(\mathbf{x}) = y_j(\mathbf{x})$ and hence corresponds to a $(D-1)$ -dimensional hyperplane defined by

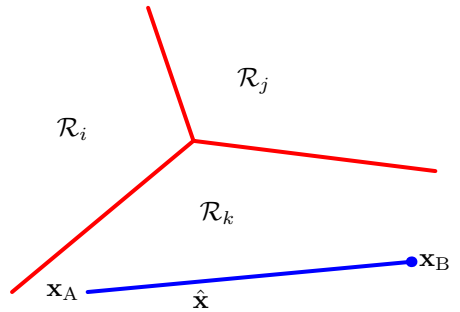
$$(\mathbf{w}_k - \mathbf{w}_j)^T \mathbf{x} + (w_{k0} - w_{j0}) = 0. \quad (4.10)$$

This has the same form as the decision boundary for the two-class case discussed in Section 4.1.1, and so analogous geometrical properties apply.

The decision regions of such a discriminant are always singly connected and convex. To see this, consider two points \mathbf{x}_A and \mathbf{x}_B both of which lie inside decision region \mathcal{R}_k , as illustrated in Figure 4.3. Any point $\hat{\mathbf{x}}$ that lies on the line connecting \mathbf{x}_A and \mathbf{x}_B can be expressed in the form

$$\hat{\mathbf{x}} = \lambda \mathbf{x}_A + (1 - \lambda) \mathbf{x}_B \quad (4.11)$$

Figure 4.3 Illustration of the decision regions for a multiclass linear discriminant, with the decision boundaries shown in red. If two points \mathbf{x}_A and \mathbf{x}_B both lie inside the same decision region \mathcal{R}_k , then any point $\hat{\mathbf{x}}$ that lies on the line connecting these two points must also lie in \mathcal{R}_k , and hence the decision region must be singly connected and convex.



where $0 \leq \lambda \leq 1$. From the linearity of the discriminant functions, it follows that

$$y_k(\hat{\mathbf{x}}) = \lambda y_k(\mathbf{x}_A) + (1 - \lambda) y_k(\mathbf{x}_B). \quad (4.12)$$

Because both \mathbf{x}_A and \mathbf{x}_B lie inside \mathcal{R}_k , it follows that $y_k(\mathbf{x}_A) > y_j(\mathbf{x}_A)$, and $y_k(\mathbf{x}_B) > y_j(\mathbf{x}_B)$, for all $j \neq k$, and hence $y_k(\hat{\mathbf{x}}) > y_j(\hat{\mathbf{x}})$, and so $\hat{\mathbf{x}}$ also lies inside \mathcal{R}_k . Thus \mathcal{R}_k is singly connected and convex.

Note that for two classes, we can either employ the formalism discussed here, based on two discriminant functions $y_1(\mathbf{x})$ and $y_2(\mathbf{x})$, or else use the simpler but equivalent formulation described in Section 4.1.1 based on a single discriminant function $y(\mathbf{x})$.

We now explore three approaches to learning the parameters of linear discriminant functions, based on least squares, Fisher's linear discriminant, and the perceptron algorithm.

4.1.3 Least squares for classification

In Chapter 3, we considered models that were linear functions of the parameters, and we saw that the minimization of a sum-of-squares error function led to a simple closed-form solution for the parameter values. It is therefore tempting to see if we can apply the same formalism to classification problems. Consider a general classification problem with K classes, with a 1-of- K binary coding scheme for the target vector \mathbf{t} . One justification for using least squares in such a context is that it approximates the conditional expectation $\mathbb{E}[\mathbf{t}|\mathbf{x}]$ of the target values given the input vector. For the binary coding scheme, this conditional expectation is given by the vector of posterior class probabilities. Unfortunately, however, these probabilities are typically approximated rather poorly, indeed the approximations can have values outside the range $(0, 1)$, due to the limited flexibility of a linear model as we shall see shortly.

Each class \mathcal{C}_k is described by its own linear model so that

$$y_k(\mathbf{x}) = \mathbf{w}_k^T \mathbf{x} + w_{k0} \quad (4.13)$$

where $k = 1, \dots, K$. We can conveniently group these together using vector notation so that

$$\mathbf{y}(\mathbf{x}) = \widetilde{\mathbf{W}}^T \tilde{\mathbf{x}} \quad (4.14)$$