Linear Models for Classification

4.1Discriminant Function

4.1.1 Two classes

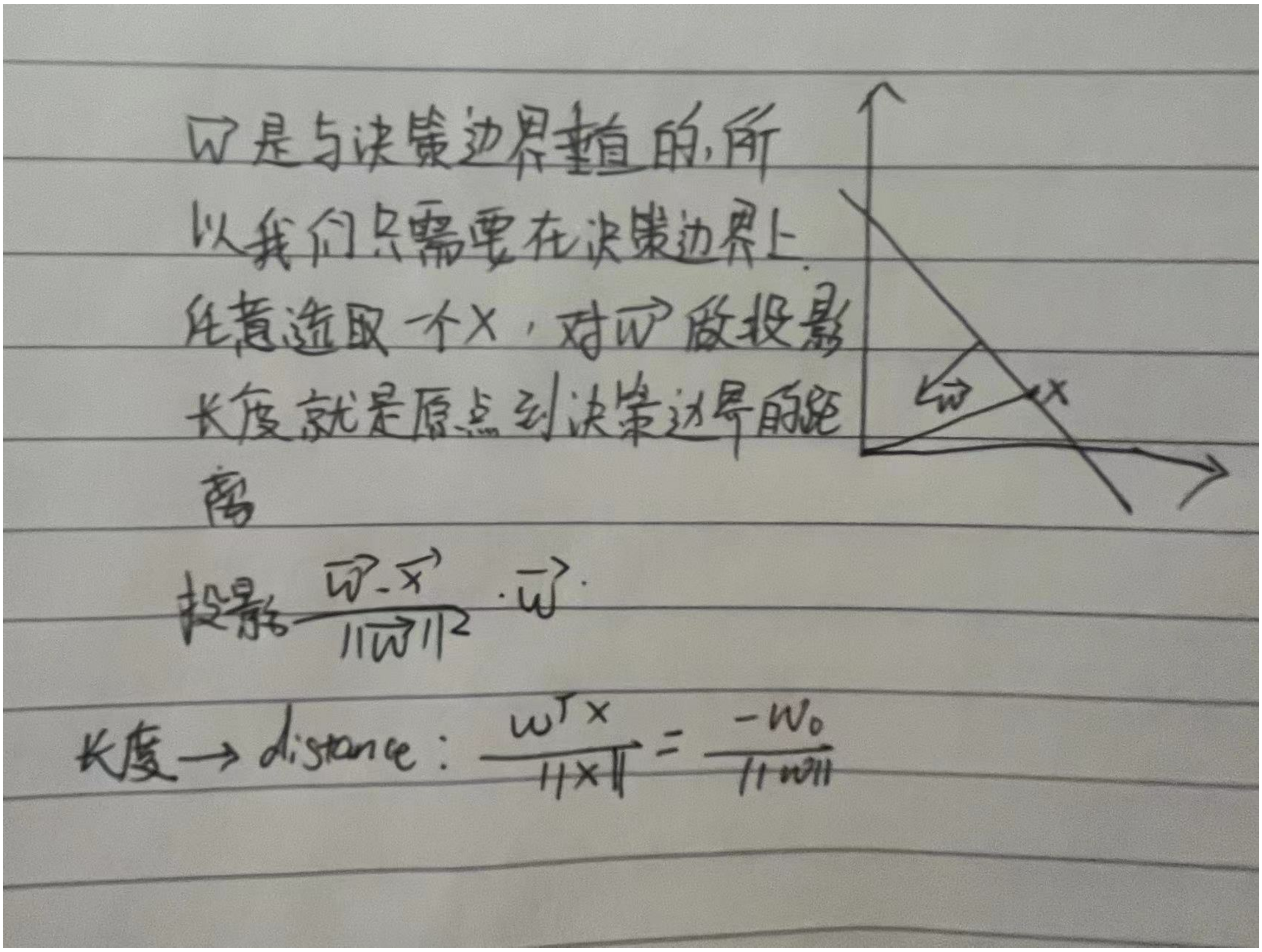
The simplest representation of a linear discriminant function is obtained by tak- ing a linear function of the input vector so that

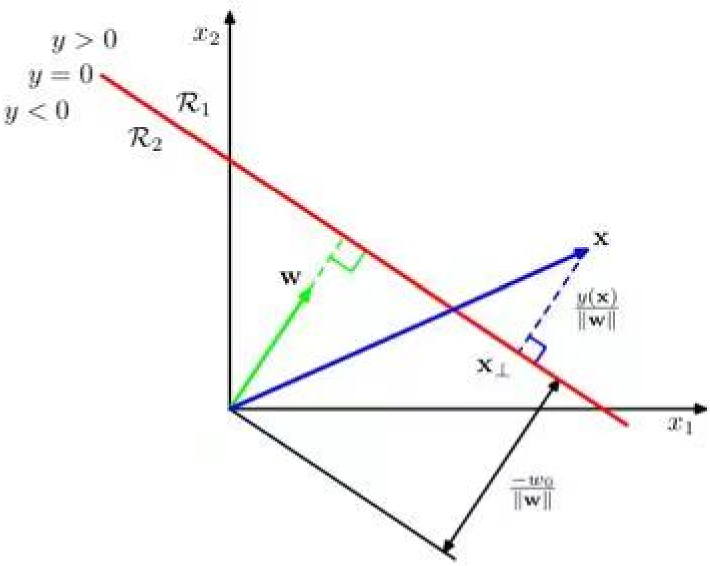
$$y(\mathbf{x}) = \mathbf{w}^{\mathrm{T}}\mathbf{x} + w_0$$

Consider two points xA and xB both of which lie on the decision surface. Because y(xA) = y(xB) = 0, we have wT(xA −xB) = 0 and hence the vector w is orthogonal to every vector lying within the decision surface, and so w determines the orientation of the decision surface.

$$\frac{\mathbf{w}^{\mathrm{T}}\mathbf{x}}{\|\mathbf{w}\|} = -\frac{w_0}{\|\mathbf{w}\|}$$

此公式适用于线性分类问题，及决策面可以用线性方程表示的情况





an arbitrary point x and let x⊥ be its orthogonal projection onto the decision surface, so that

$$\mathbf{x} = \mathbf{x}_\perp + r\frac{\mathbf{w}}{\|\mathbf{w}\|}$$

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

两边同时左乘 W^T，再加上 w_0.

利用 $y(x) = W^T x + w_0 = 0$

$y(x_\perp) = W^T x_\perp + w_0 = 0$

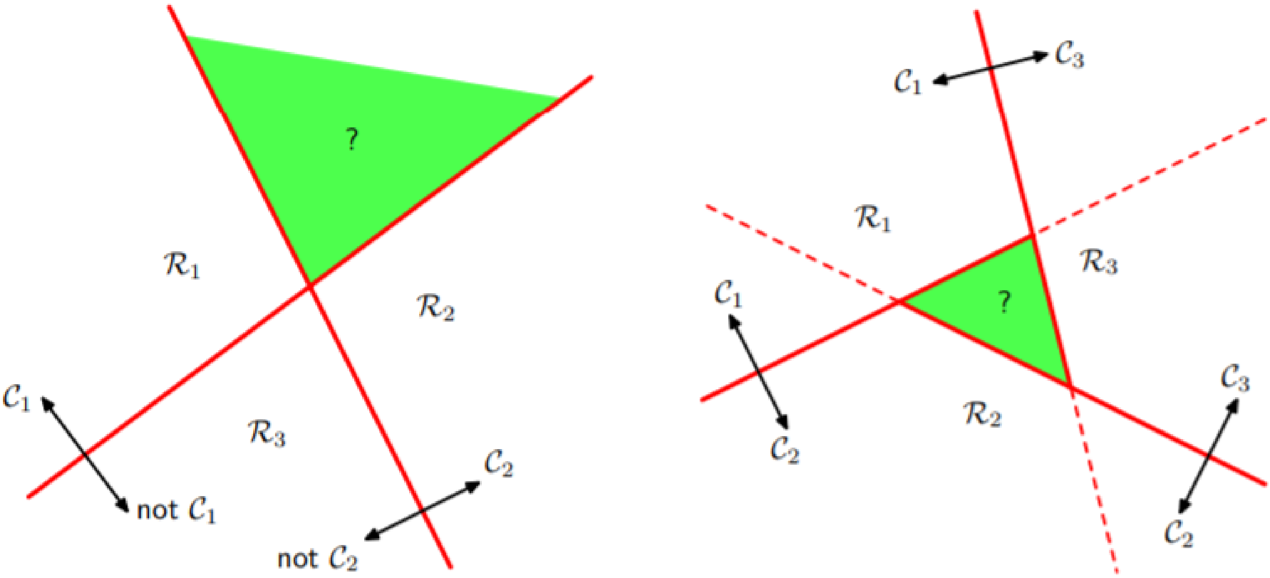
$$\underbrace{W^T x + w_0}_{y(x)} = \underbrace{\underline{W}^T x_\perp + r \cdot \frac{W^T W}{||W||} + w_0}_{y(x_\perp)}$$

$$r = \frac{y(x)}{||W||}$$

4.1.2 Multiple classes

One-versus-the-rest Consider the use of K−1 classifiers each of which solves a two-class problem of separating points in a particular class Ck from points not in that class.

one-versus-one An alternative is to introduce K(K − 1)/2 binary discriminant functions, one for every possible pair of classes.

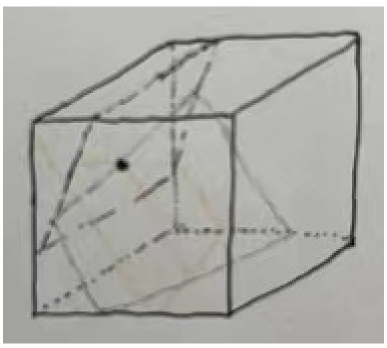


We can avoid these difficulties by considering a single K-class discriminant comprising K linear functions of the form

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

比如此时有三类数据，则判别函数为：

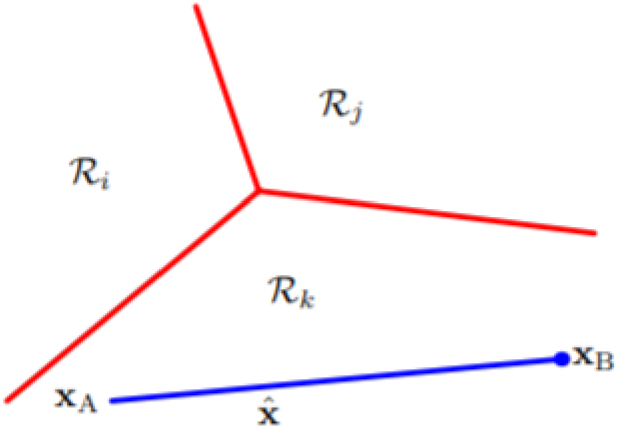
$$\begin{cases} y_1(x) = w_1^T x + w_{10} \\ y_2(x) = w_2^T x + w_{20} \\ y_3(x) = w_3^T x + w_{30} \end{cases}$$



and then assigning a point x to class Ck if yk(x) > yj (x) for all j ≠ k. The decision boundary between class Ck and class Cj is therefore given by yk(x) = yj (x) and hence corresponds to a (D − 1)-dimensional hyperplane defined by

$$\left(\mathbf{w}_k - \mathbf{w}_j\right)^{\mathrm{T}} \mathbf{x} + \left(w_{k0} - w_{j0}\right) = 0$$

The decision regions of such a discriminant are always singly connected and convex.



$$\widehat{\mathbf{x}} = \lambda \mathbf{x}_{\mathrm{A}} + (1-\lambda)\mathbf{x}_{\mathrm{B}}$$

where 0≤λ≤1. From the linearity of the discriminant functions, it follows that

$$y_k(\widehat{\mathbf{x}}) = \lambda y_k\left(\mathbf{x}_{\mathrm{A}}\right) + (1-\lambda) y_k\left(\mathbf{x}_{\mathrm{B}}\right)$$

Because both x_{A} and x_{B} lie inside$\mathcal{R}_k$, it follows that $y_k\left(\mathbf{x}_{\mathrm{A}}\right) > y_j\left(\mathbf{x}_{\mathrm{A}}\right)$, and $y_k\left(\mathbf{x}_{\mathrm{B}}\right) > y_j\left(\mathbf{x}_{\mathrm{B}}\right)$, for all $j \neq k$, and hence $y_k(\widehat{\mathbf{x}}) > y_j(\widehat{\mathbf{x}})$, and so $\widehat{\mathbf{x}}$ also lies inside $\mathcal{R}_k$. Thus $\mathcal{R}_k$ is singly connected and convex.

4.1.3 Least squares for classification

Each class Ck is described by its own linear model so that

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

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

$$\mathbf{y}(\mathbf{x}) = \widetilde{\mathbf{W}}^{\mathrm{T}} \widetilde{\mathbf{x}}$$

where $\widetilde{\mathbf{W}}$ is a matrix whose k^{th} column comprises the D+1 -dimensional vector $\widetilde{\mathbf{w}}_k = \left(w_{k0}, \mathbf{w}_k^{\mathrm{T}}\right)^{\mathrm{T}}$ and $\widetilde{\mathbf{x}}$ is the corresponding augmented input vector $\left(1, \mathbf{x}^{\mathrm{T}}\right)^{\mathrm{T}}$ with a dummy input$x_0 = 1$. This representation was discussed in detail in Section 3.1. A new input $\mathbf{x}$ is then assigned to the class for which the output $y_k = \widetilde{\mathbf{w}}_k^{\mathrm{T}} \widetilde{\mathbf{x}}$ is largest.

We now determine the parameter matrix $\widetilde{\mathbf{W}}$ by minimizing a sum-of-squares error function, as we did for regression in Chapter 3. Consider a training data set$\{\mathbf{x}_n, \mathbf{t}_n\}$ where $n = 1, \ldots, N$, and define a matrix $\mathbf{T} whose n^{\mathrm{th}}$ row is the vector $\mathbf{t}_n^{\mathrm{T}}$, together with a matrix$\widetilde{\mathbf{X}}$whosen^{th} row is $\widetilde{\mathbf{x}}_n^{\mathrm{T}}$. The sum-of-squares error function can then be written as

$$E_D(\widetilde{\mathbf{W}}) = \frac{1}{2}\mathrm{Tr}\left\{(\widetilde{\mathbf{X}}\widetilde{\mathbf{W}} - \mathbf{T})^{\mathrm{T}}(\widetilde{\mathbf{X}}\widetilde{\mathbf{W}} - \mathbf{T})\right\}$$

.

$$\tilde{W} = \begin{bmatrix} w_{10} & w_{20} & w_{k0} \\ \vdots & \vdots & \vdots \\ w_n & w_n & w_n \end{bmatrix}$$ 每一列都是 $\begin{pmatrix} w_{k0} \\ w_k^T \end{pmatrix} = \tilde{w}_k$

$$\tilde{X} = \begin{bmatrix} x_1^T \\ \vdots \\ x_k^T \end{bmatrix} \qquad T = \begin{bmatrix} t_1^T \\ \vdots \\ t_n^T \end{bmatrix}$$

$$\tilde{X} \cdot \tilde{W} = \begin{bmatrix} x_1^T \\ \vdots \\ x_k^T \end{bmatrix} \begin{bmatrix} \tilde{w}_1 & \cdots & \tilde{w}_k \end{bmatrix}$$

$$= \begin{bmatrix} P(C_1|x_1) & \cdots & P(C_k|x_1) \\ \vdots & & \vdots \\ P(C_1|x_k) & \cdots & P(C_k|x_k) \end{bmatrix}$$ x_1分到C_1中的估计/预测值

$$\tilde{X} \tilde{W} - T = \begin{bmatrix} P(C_1|x_1) - t_{11} & \cdots & & P(C_k|x_1) - t_{1k} \\ & P(C_2|x_2) - t_{22} & & \\ & & \ddots & \\ & & & P(C_k|x_k) - t_{kk} \end{bmatrix}$$

若 $\tilde{X}\tilde{W} - T = a$, $a_{ij} = P(C_j|x_i) - t_{ij}$

$$(\tilde{X}\tilde{W} - T)^T (\tilde{X}\tilde{W} - T) = \begin{bmatrix} a_{11} & \cdots & a_{k1} \\ \vdots & a_{22} & \\ a_{1k} & \cdots & a_{kk} \end{bmatrix} \begin{bmatrix} a_{11} & \cdots & a_{1k} \\ \vdots & & \vdots \\ a_{k1} & \cdots & a_{kk} \end{bmatrix}$$

$$= \begin{bmatrix} a_{11}^2 + a_{21}^2 + \cdots + a_{k1}^2 & & \\ & a_{12}^2 + a_{22}^2 + \cdots + a_{k2}^2 & \\ & & \ddots \\ & & & a_{1k}^2 + \cdots + a_{kk}^2 \end{bmatrix}_{k \times k}$$

$$\mathrm{Tr}\left\{(\tilde{X}\tilde{W} - T)^T (\tilde{X}\tilde{W} - T)\right\} = \underbrace{(a_{11}^2 + a_{21}^2 + \cdots + a_{k1}^2)}_{C_1\text{分类误差平方和}} + \cdots + \underbrace{(a_{1k}^2 + a_{2k}^2 + \cdots + a_{kk}^2)}_{C_k\text{分类误差平方和}}$$

Setting the derivative with respect to $\widetilde{\mathbf{W}}$ to zero, and rearranging, we then obtain the solution for $\widetilde{\mathbf{W}}$ in the form

$$\widetilde{\mathbf{W}} = \left(\widetilde{\mathbf{X}}^{\mathrm{T}} \widetilde{\mathbf{X}}\right)^{-1} \widetilde{\mathbf{X}}^{\mathrm{T}} \mathbf{T} = \widetilde{\mathbf{X}}^{\dagger} \mathbf{T}$$

相关推导

因此，令 $\frac{\partial E_D(\hat{W})}{\partial \hat{W}} = 0$，得:

$\hat{W} = \left(\hat{X}^T \hat{X}\right)^{-1} \hat{X}^T T = \hat{X}^{\dagger} T$

$\hat{X}^{\dagger} = \left(\hat{X}^T \hat{X}\right)^{-1} \hat{X}^T$ 为矩阵 $\hat{X}$ 得伪逆矩阵。因此，判别函数组为:

$y(x) = \hat{W}^T \hat{x} = T^T \left(\hat{X}^{\dagger}\right)^T \hat{x}$

Setting the derivative with respect to $\widetilde{\mathbf{W}}$ to zero, and rearranging, we then obtain the solution for $\widetilde{\mathbf{W}}$ in the form

$$\widetilde{\mathbf{W}} = \left(\widetilde{\mathbf{X}}^{\mathrm{T}} \tilde{\mathbf{X}}\right)^{-1} \widetilde{\mathbf{X}}^{\mathrm{T}} \mathbf{T} = \widetilde{\mathbf{X}}^{\dagger} \mathbf{T}$$

where $\widetilde{\mathbf{X}}^{\dagger}$ is the pseudo-inverse of the matrix $\widetilde{\mathbf{X}}$, as discussed in Section 3.1.1. We then obtain the discriminant function in the form

$$\mathbf{y}(\mathbf{x}) = \widetilde{\mathbf{W}}^{\mathrm{T}} \tilde{\mathbf{x}} = \mathbf{T}^{\mathrm{T}} \left(\widetilde{\mathbf{X}}^{\dagger}\right)^{\mathrm{T}} \tilde{\mathbf{x}}$$

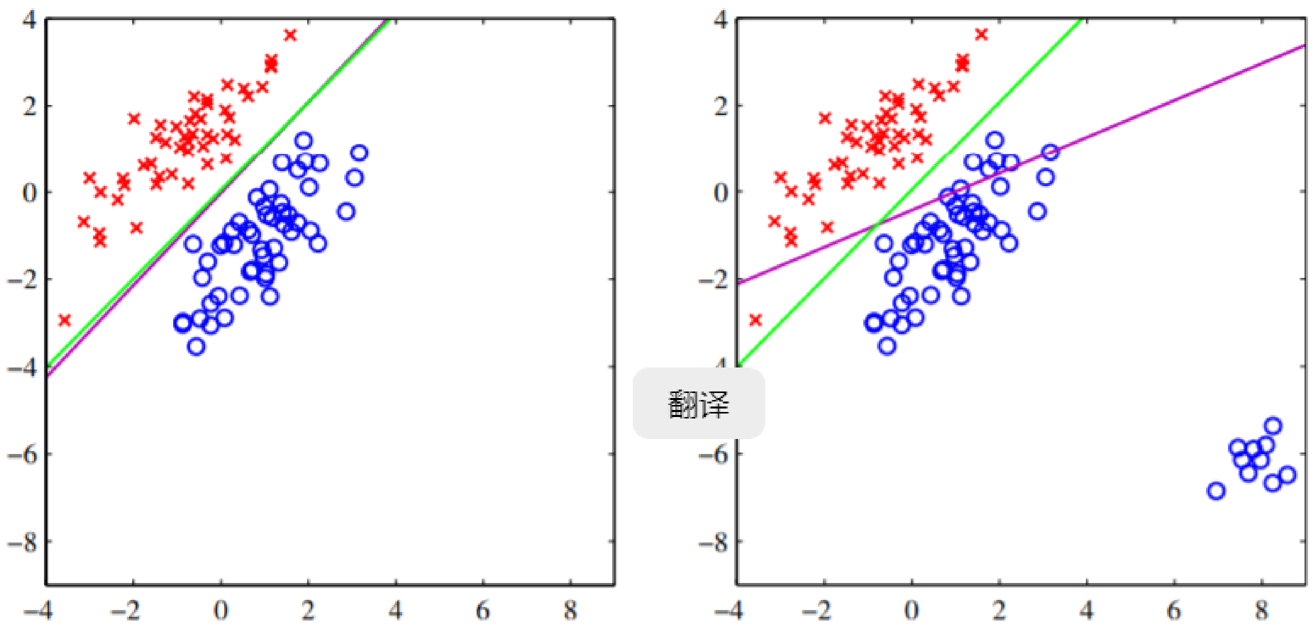
.

An interesting property of least-squares solutions with multiple target variables is that if every target vector in the training set satisfies some linear constraint

$$\mathbf{a}^{\mathrm{T}} \mathbf{t}_n + b = 0$$

for some constants a and b, then the model prediction for any value of x will satisfy the same constraint so that

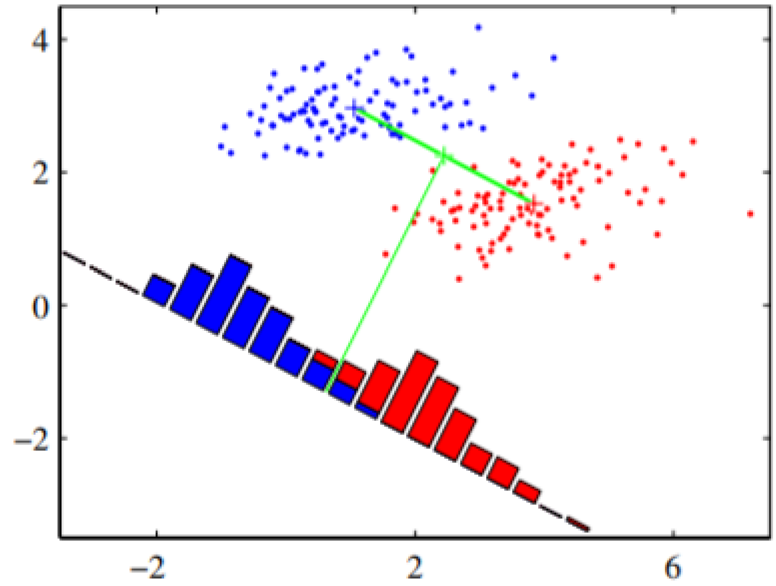
$$\mathbf{a}^{\mathrm{T}} \mathbf{y}(\mathbf{x}) + b = 0$$



4.1.4--4.1.6 Fisher's linear discriminant

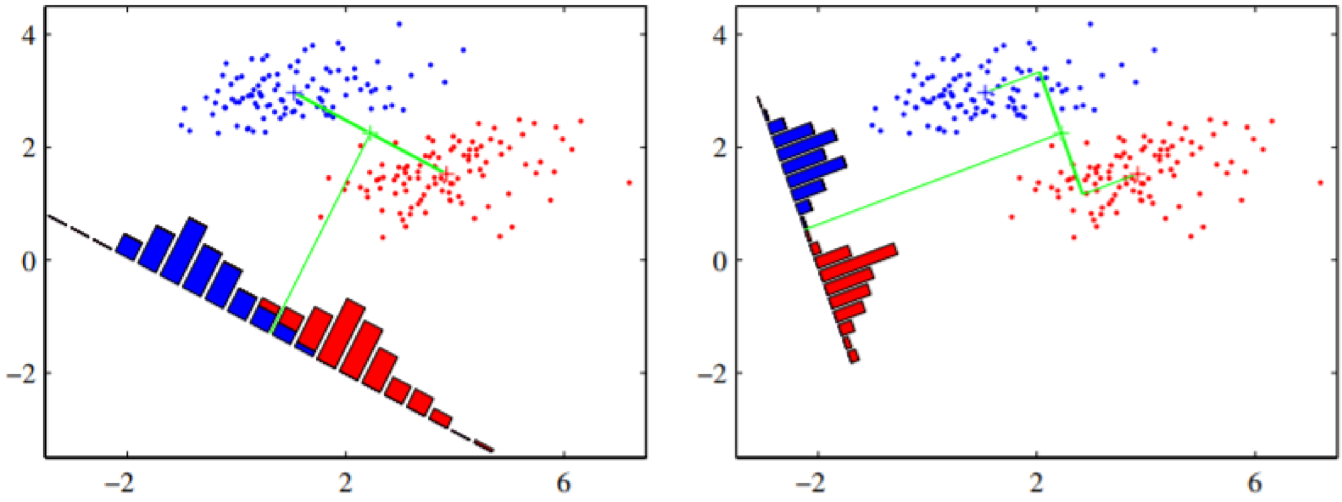
One way to view a linear classification model is in terms of dimensionality reduction. Consider first the case of two classes, and suppose we take the D-dimensional input vector x and project it down to one dimension using

$$y = \mathbf{w}^{\mathrm{T}}\mathbf{x}$$



If we place a threshold on y and classify $y \geqslant -w_0$ as class $\mathcal{C}_1$, and otherwise class $\mathcal{C}_2$, then we obtain our standard linear classifier discussed in the previous section.

D-dimensional	1-dimensional
mean vectors	
$\mathbf{m}_1 = \frac{1}{N_1}\sum_{n \in \mathcal{C}_1} \mathbf{x}_n, \quad \mathbf{m}_2 = \frac{1}{N_2}\sum_{n \in \mathcal{C}_2} \mathbf{x}_n$	$m_k = \mathbf{w}^{\mathrm{T}}\mathbf{m}_k$
within-class discrete degree	
$S_K = \sum_{n \in C_k} (x_n - m_k)(x_n - m_k)^T$ $S_W = S_1 + S_2$	$s_k^2 = \sum_{n \in \mathcal{C}_k} (y_n - m_k)^2$ $s_1^2 + s_2^2$
between-class discrete degree	
$\mathbf{S}_{\mathrm{B}} = (\mathbf{m}_2 - \mathbf{m}_1)(\mathbf{m}_2 - \mathbf{m}_1)^{\mathrm{T}}$	$(m_2 - m_1)^2$



where $y_n = \mathbf{w}^{\mathrm{T}}\mathbf{x}_n$. We can define the total within-class variance for the whole data set to be simply $s_1^2 + s_2^2$. The Fisher criterion is defined to be the ratio of the between-class variance to the within-class variance and is given by

$$J(\mathbf{w}) = \frac{(m_2 - m_1)^2}{s_1^2 + s_2^2}$$

We can make the dependence on w explicit by using (4.20), (4.23), and (4.24) to rewrite the Fisher criterion in the form

$$J(\mathbf{w}) = \frac{\mathbf{w}^{\mathrm{T}}\mathbf{S}_{\mathrm{B}}\mathbf{w}}{\mathbf{w}^{\mathrm{T}}\mathbf{S}_{\mathrm{W}}\mathbf{w}}$$

$$(m_2 - m_1)^2 = (w^T m_2 - m^T m_1)^2$$

$$= (w^T m_2 - w^T m_1)(w^T m_2 - w^T m_1)^T$$

$$= (w^T m_2 - w^T m_1)[w^T(m_2 - m_1)]^T$$

$$= (w^T m_2 - w^T m_1)[(w^T)^T (m_2 - m_1)^T]$$

$$= (w^T m_2 - w^T m_1)(m_2 - m_1)^T w$$

$$= w^T S_B w$$

$(AB)^T = A^T \cdot B^T$

$$s_k^2 = \sum_{n \in C_k} (y_n - m_k)^2 = \sum_{n \in C_k} (w^T x_n - w^T m_k)^2$$

$$= w^T \left[\sum_{n \in C_k} (x_n - m_k)(x_n - m_k)^T \right] w$$

$$= w^T S_w w$$

where $\mathbf{S}_\mathrm{B}$ is the between-class covariance matrix and is given by

$$\mathbf{S}_\mathrm{B} = (\mathbf{m}_2 - \mathbf{m}_1)(\mathbf{m}_2 - \mathbf{m}_1)^\mathrm{T}$$

and $\mathbf{S}_\mathrm{W}$ is the total within-class covariance matrix, given by

$$\mathbf{S}_\mathrm{W} = \sum_{n \in \mathcal{C}_1} (\mathbf{x}_n - \mathbf{m}_1)(\mathbf{x}_n - \mathbf{m}_1)^\mathrm{T} + \sum_{n \in \mathcal{C}_2} (\mathbf{x}_n - \mathbf{m}_2)(\mathbf{x}_n - \mathbf{m}_2)^\mathrm{T}$$

.

Differentiating (4.26) with respect to $\mathbf{w}$, we find that $J(\mathbf{w})$ is maximized when

$$\left(\mathbf{w}^\mathrm{T} \mathbf{S}_\mathrm{B} \mathbf{w}\right) \mathbf{S}_\mathrm{W} \mathbf{w} = \left(\mathbf{w}^\mathrm{T} \mathbf{S}_\mathrm{W} \mathbf{w}\right) \mathbf{S}_\mathrm{B} \mathbf{w}.$$

From (4.27), we see that $\mathbf{S}_\mathrm{B} \mathbf{w}$ is always in the direction of $(\mathbf{m}_2 - \mathbf{m}_1)$. Furthermore, we do not care about the magnitude of $\mathbf{w}$, only its direction, and so we can drop the scalar factors $\left(\mathbf{w}^\mathrm{T} \mathbf{S}_\mathrm{B} \mathbf{w}\right)$ and $\left(\mathbf{w}^\mathrm{T} \mathbf{S}_\mathrm{W} \mathbf{w}\right)$. Multiplying both sides of (4.29) by $\mathbf{S}_\mathrm{W}^{-1}$ we then obtain

$$\mathbf{w} \propto \mathbf{S}_\mathrm{W}^{-1} (\mathbf{m}_2 - \mathbf{m}_1)$$

.

Note that if the within-class covariance is isotropic, so that$\mathbf{S}_\mathrm{W}$ is proportional to the unit matrix, we find that $\mathbf{w}$ is proportional to the difference of the class means, as discussed above.

如果类内协方差矩阵是各向同性（各向同性即指随机向量的协方差矩阵为标量乘以单位矩阵，即每 个方向的方差相同，对角阵是因为相关性在考察两个不同类之间不重要），则 $S_W \propto I$ ，即类内 协方差矩阵正比于单位矩

阵，因此其逆矩阵也正比于单位矩阵，则: $w \propto m_2 - m_1$ 此时， w 正比于类内均值之差， w 与类内均值之差的方向相同。

设 $W^TS_WW=a$, $W^TS_BW=b$, 两项均为标量

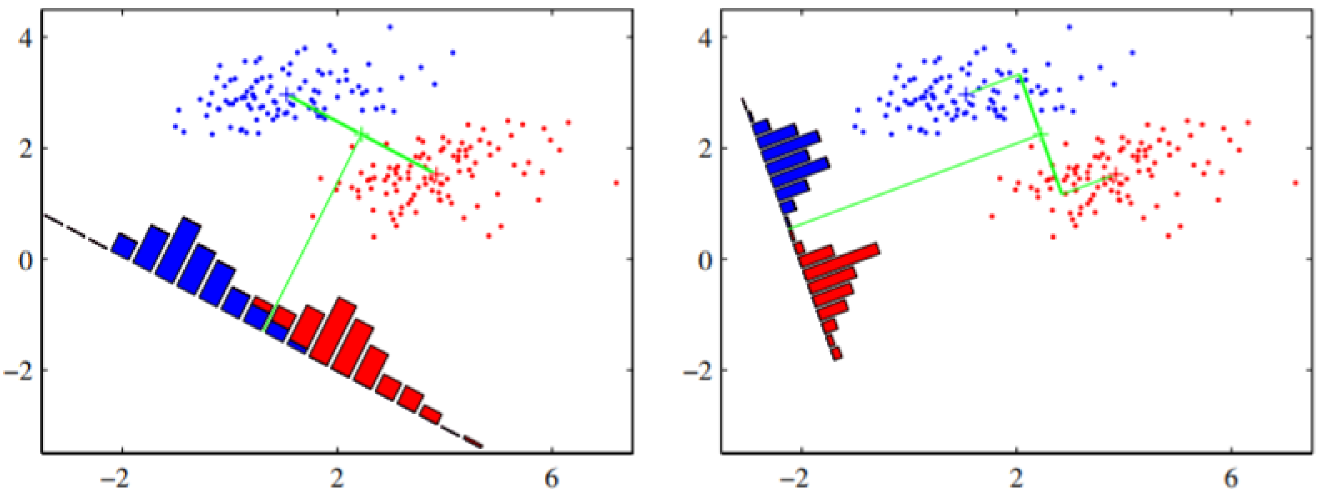
上式两边左乘 $S^{-1}W$，得

$$W=\frac{a}{b}S_W^{-1}S_BW=\frac{a}{b}S_W^{-1}(m_2-m_1)(m_2-m_1)^TW$$

设 $C=(m_2-m_1)^TW$ 标量

$$W=\frac{ac}{b}S_W^{-1}(m_2-m_1)$$

因此 $W \propto S^{-1}W(m_2-m_1)$



Fisher’s discriminant for multiple classes

We now consider the generalization of the Fisher discriminant to $K > 2$ classes, and we shall assume that the dimensionality D of the input space is greater than the number K of classes. Next, we introduce $D' > 1$ linear 'features' $y_k = \mathbf{w}_k^{\mathrm{T}}\mathbf{x}$, where k=1,… , D' . These feature values can conveniently be grouped together to form a vector $\mathbf{y}$. Similarly, the weight vectors $\{\mathbf{w}_k\}$ can be considered to be the columns of a matrix $\mathbf{W}$, so that

$$\mathbf{y} = \mathbf{W}^{\mathrm{T}}\mathbf{x}$$

Note that again we are not including any bias parameters in the definition of $\mathbf{y}$. The generalization of the within-class covariance matrix to the case of K classes follows from (4.28) to give

$$\mathbf{S}_{\mathrm{W}} = \sum_{k=1}^{K} \mathbf{S}_k$$

where

$$\mathbf{S}_k = \sum_{n \in \mathcal{C}_k} (\mathbf{x}_n - \mathbf{m}_k)(\mathbf{x}_n - \mathbf{m}_k)^{\mathrm{T}}$$
$$\mathbf{m}_k = \frac{1}{N_k} \sum_{n \in \mathcal{C}_k} \mathbf{x}_n$$

and N_k is the number of patterns in class $\mathcal{C}_k$. In order to find a generalization of the between-class covariance matrix, we follow Duda and Hart (1973) and consider first the total covariance matrix

$$\mathbf{S}_{\mathrm{T}} = \sum_{n=1}^{N} (\mathbf{x}_n - \mathbf{m})(\mathbf{x}_n - \mathbf{m})^{\mathrm{T}}$$

where$\mathbf{m}$ is the mean of the total data set

$$\mathbf{m} = \frac{1}{N} \sum_{n=1}^{N} \mathbf{x}_n = \frac{1}{N} \sum_{k=1}^{K} N_k \mathbf{m}_k$$

and $N = \sum_k N_k$ is the total number of data points. The total covariance matrix can be decomposed into the sum of the within-class covariance matrix, given by (4.40) and (4.41), plus an additional matrix $\mathbf{S}_{\mathrm{B}}$, which we identify as a measure of the between-class covariance

$$\mathbf{S}_{\mathrm{T}} = \mathbf{S}_{\mathrm{W}} + \mathbf{S}_{\mathrm{B}}$$

where

$$\mathbf{S}_{\mathrm{B}} = \sum_{k=1}^{K} N_k (\mathbf{m}_k - \mathbf{m})(\mathbf{m}_k - \mathbf{m})^{\mathrm{T}}$$

.

These covariance matrices have been defined in the original x -space. We can now define similar matrices in the projected $D' - dimensional y - space$

$$\mathbf{s}_{\mathrm{W}} = \sum_{k=1}^{K} \sum_{n \in \mathcal{C}_k} \left(\mathbf{y}_n - \boldsymbol{\mu}_k\right) \left(\mathbf{y}_n - \boldsymbol{\mu}_k\right)^{\mathrm{T}}$$

and

$$\mathbf{s}_{\mathrm{B}} = \sum_{k=1}^{K} N_k (\boldsymbol{\mu}_k - \boldsymbol{\mu})(\boldsymbol{\mu}_k - \boldsymbol{\mu})^{\mathrm{T}}$$

where

$$\boldsymbol{\mu}_k = \frac{1}{N_k} \sum_{n \in \mathcal{C}_k} \mathbf{y}_n, \quad \boldsymbol{\mu} = \frac{1}{N} \sum_{k=1}^{K} N_k \boldsymbol{\mu}_k$$

.

Again we wish to construct a scalar that is large when the between-class covariance is large and when the within-class covariance is small. There are now many possible choices of criterion (Fukunaga, 1990). One example is given by

$$J(\mathbf{W}) = \mathrm{Tr}\left\{\mathbf{s}_{\mathrm{W}}^{-1}\mathbf{s}_{\mathrm{B}}\right\}$$

This criterion can then be rewritten as an explicit function of the projection matrix $\mathbf{W}$ in the form

$$J(\mathbf{w}) = \mathrm{Tr}\left\{\left(\mathbf{W}\mathbf{S}_{\mathrm{W}}\mathbf{W}^{\mathrm{T}}\right)^{-1}\left(\mathbf{W}\mathbf{S}_{\mathrm{B}}\mathbf{W}^{\mathrm{T}}\right)\right\}$$

4.1.7 The perceptron algorithm

Another example of a linear discriminant model is the perceptron of Rosenblatt (1962), which occupies an important place in the history of pattern recognition algorithms. It corresponds to a two-class model in which the input vector \mathbf{x} is first transformed using a fixed nonlinear transformation to give a feature vector \phi(\mathbf{x}) , and this is then used to construct a generalized linear model of the form

$$y(\mathbf{x}) = f\left(\mathbf{w}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x})\right)$$

where the nonlinear activation function $f(\cdot)$ is given by a step function of the form

$$f(a) = \begin{cases} +1, & a \geqslant 0 \\ -1, & a < 0. \end{cases} \tag{4.53}$$

We therefore consider an alternative error function known as the perceptron criterion. To derive this, we note that we are seeking a weight vector $\mathbf{w}$ such that patterns$\mathbf{x}_n$ in class$\mathcal{C}_1$ will have$\mathbf{w}^{\mathrm{T}}\boldsymbol{\phi}\left(\mathbf{x}_n\right) > 0$, whereas patterns $\mathbf{x}_n$ in class $\mathcal{C}_2$ have$\mathbf{w}^{\mathrm{T}}\boldsymbol{\phi}\left(\mathbf{x}_n\right) < 0. Usingthet \in \{-1,+1\}$ target coding scheme it follows that we would like all patterns to satisfy $\mathbf{w}^{\mathrm{T}}\boldsymbol{\phi}\left(\mathbf{x}_n\right) t_n > 0$. The perceptron criterion associates zero error with any pattern that is correctly classified, whereas for a misclassified pattern$\mathbf{x}_n$ it tries to minimize the quantity $-\mathbf{w}^{\mathrm{T}}\boldsymbol{\phi}\left(\mathbf{x}_n\right) t_n$. The perceptron criterion is therefore given by

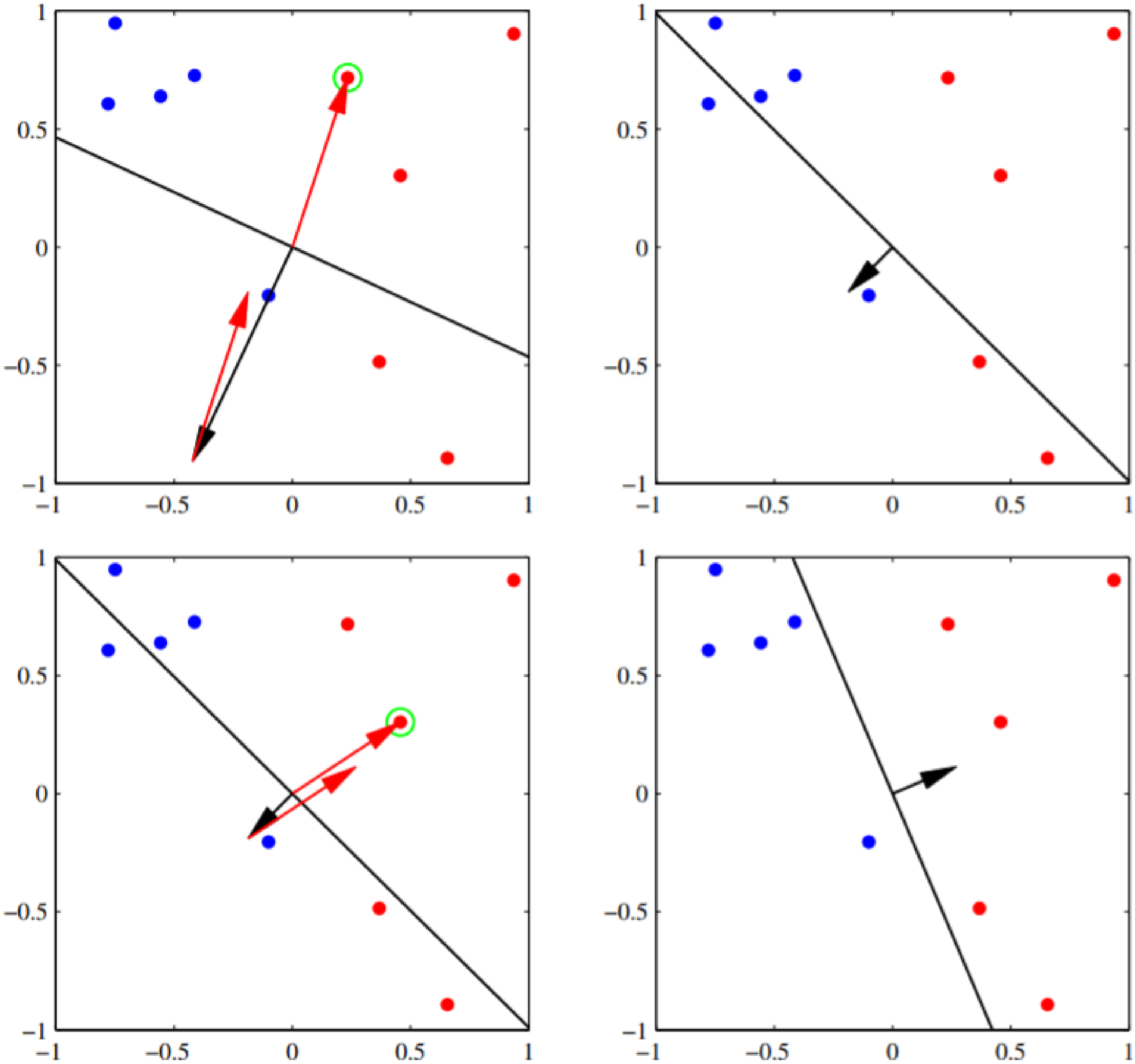
$$E_{\mathrm{P}}(\mathbf{w}) = -\sum_{n\in\mathcal{M}} \mathbf{w}^{\mathrm{T}}\boldsymbol{\phi}_n t_n$$

We now apply the stochastic gradient descent algorithm to this error function. The change in the weight vector $\mathbf{w}$ is then given by

$$\mathbf{w}^{(\tau+1)} = \mathbf{w}^{(\tau)} - \eta\nabla E_{\mathrm{P}}(\mathbf{w}) = \mathbf{w}^{(\tau)} + \eta\boldsymbol{\phi}_n t_n$$

If we consider the effect of a single update in the perceptron learning algorithm, we see that the contribution to the error from a misclassified pattern will be reduced because from (4.55) we have

$$-\mathbf{w}^{(\tau+1)\mathrm{T}}\boldsymbol{\phi}_n t_n = -\mathbf{w}^{(\tau)\mathrm{T}}\boldsymbol{\phi}_n t_n - (\boldsymbol{\phi}_n t_n)^{\mathrm{T}}\boldsymbol{\phi}_n t_n < -\mathbf{w}^{(\tau)\mathrm{T}}\boldsymbol{\phi}_n t_n$$


1
0.5
0
−0.5
−1
−1
−0.5
0
0.5
1