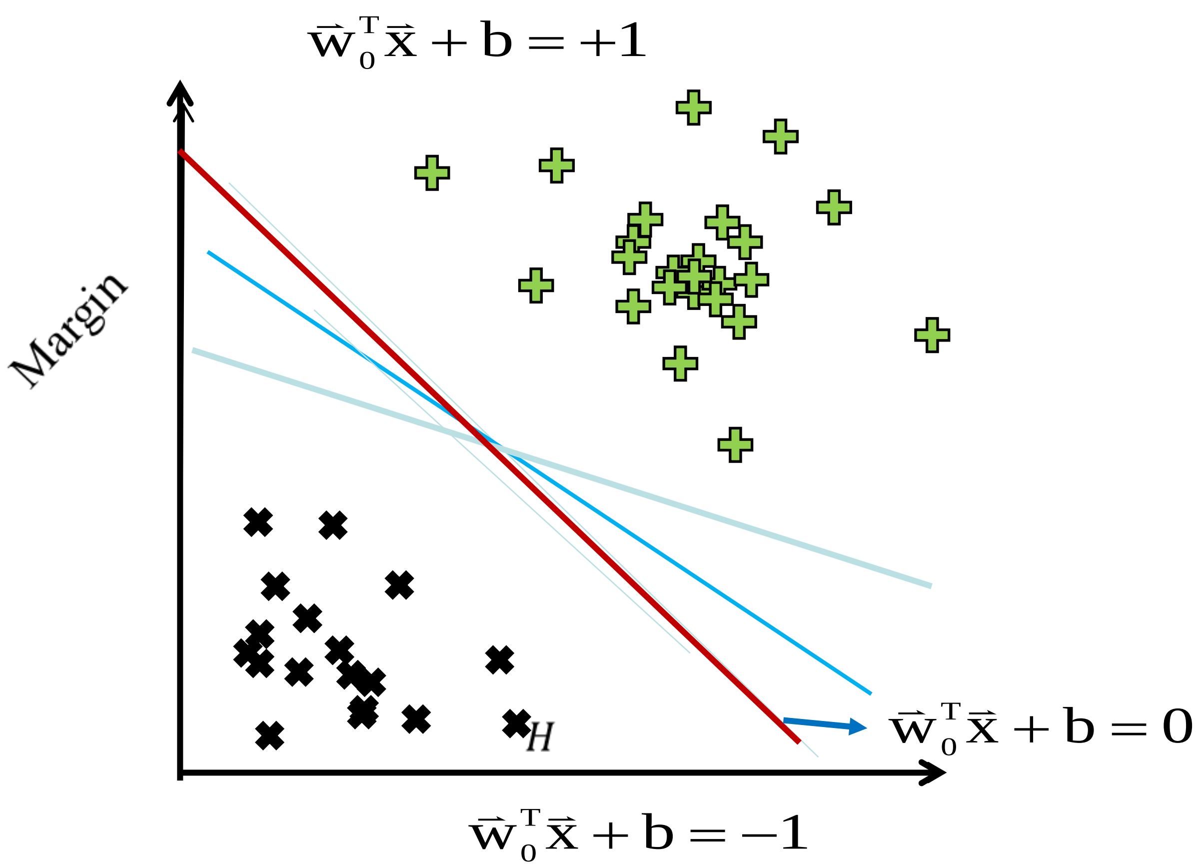
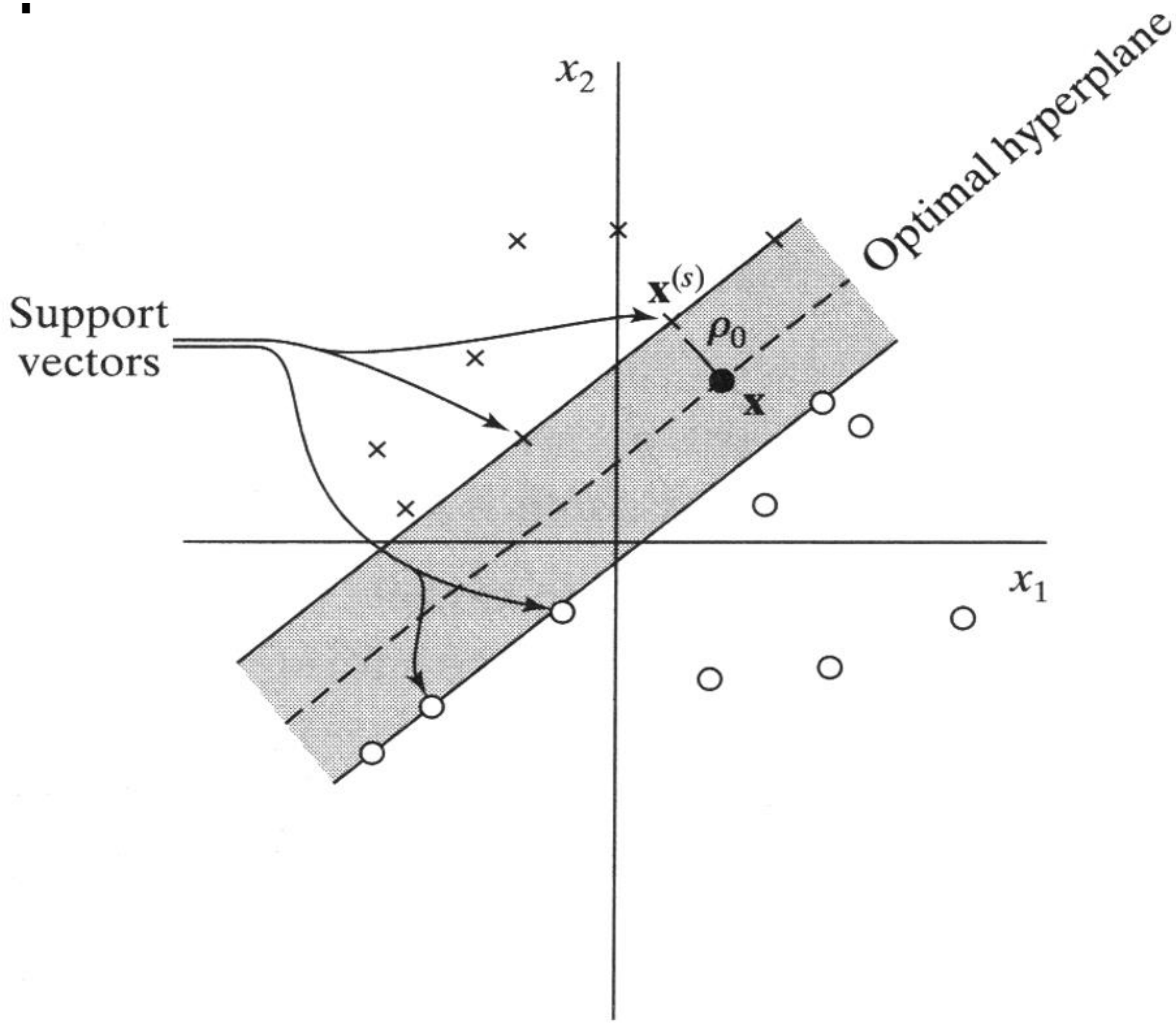
Support Vector Machines

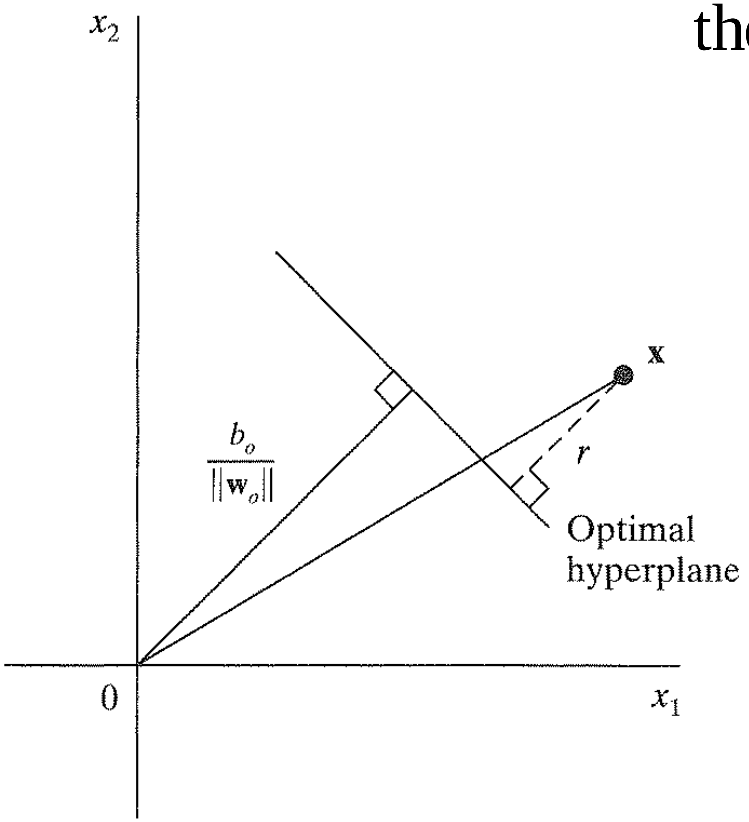
之前討論只要分界線合理讓所有資料正確即可，但我們不可能已經有所有資料



Support Vector Machines



The optimal hyperplane $\vec{w}_0^T \vec{x} + b = 0$



the distance from $\vec{x}$ to hyperplane is

$$g(x) = \vec{w}_0^T \vec{x} + b$$

Let x_p be the normal projection of x onto the optimal hyperplane

r: be the algebra distance
(r>0 on positive side, r<0 on negative side)

Then $$x = x_p + r\frac{\boldsymbol{w}_0}{\|\boldsymbol{w}_0\|}$$

$$g(\vec{x}) = \vec{w}_0^T \vec{x} + b = r\|\boldsymbol{w}_0\|$$

$$\Rightarrow r = \frac{g(\vec{x})}{\|\boldsymbol{w}_0\|}$$

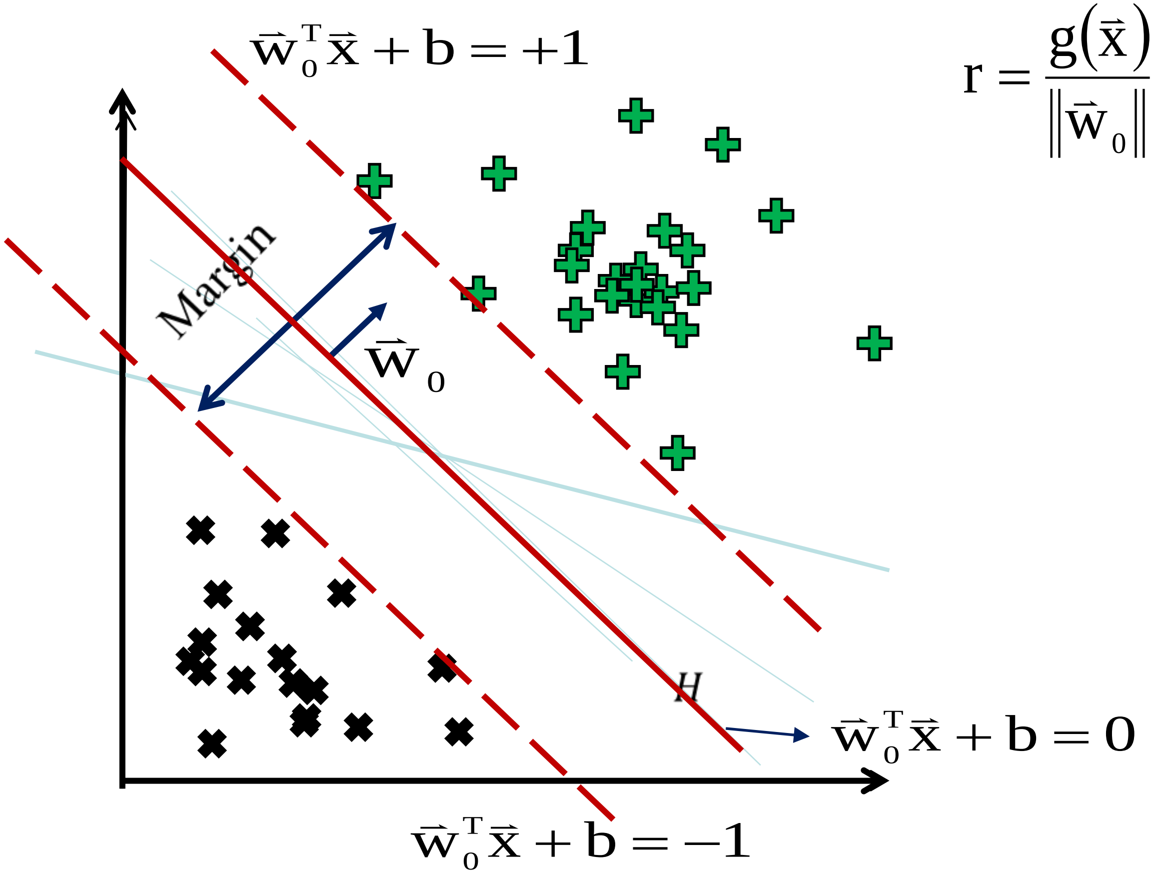
The algebraic distance from support vector to optimal hyperlane is

$$r = \frac{g(\vec{x})}{\|\vec{w}_0\|} = \begin{cases} \dfrac{1}{\|\vec{w}_0\|} & \text{if } d = +1 \\ \dfrac{-1}{\|\vec{w}_0\|} & \text{if } d = -1 \end{cases}$$

Margin of separation between two classes:

$$\rho = 2r = \frac{2}{\|\vec{w}_0\|}$$

➔ Maximum separation implies minimizes $\|\vec{w}_0\|$



$$r = \frac{g(\vec{x})}{\|\vec{w}_0\|} = \begin{cases} \dfrac{1}{\|\vec{w}_0\|} & \text{if } d = +1 \\ \dfrac{-1}{\|\vec{w}_0\|} & \text{if } d = -1 \end{cases}$$

Lagrangian function

$$\mathbf{w}_o^T\mathbf{x}_i + b_o \geq 1 \quad \text{for } d_i = +1$$
$$\mathbf{w}_o^T\mathbf{x}_i + b_o \leq -1 \quad \text{for } d_i = -1 \tag{6.6}$$

$$\rightarrow \quad d_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1 \quad \text{for } i = 1, 2, \ldots, N \tag{6.10}$$

The particular data points (x_i,d_i) for which the first or second line if Eq. (6.6) is satisfied with the quality sign are called *support vectors*.

Given the training sample $\{(\mathbf{x}_i, d_i)\}_{i=1}^{N}$, *find the optimum values of the weight vector* $\mathbf{w}$ *and bias b such that they satisfy the constraints*

Kuhn-Tucker conditions: $\underline{d_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1}$ for $i = 1, 2, \ldots, N$

and the weight vector $\mathbf{w}$ *minimizes the cost function:*

$$\Phi(\mathbf{w}) = \frac{1}{2}\mathbf{w}^T\mathbf{w}$$

Lagrangian function

➔ Minimize $$J(\mathbf{w}, b, \alpha) = \frac{1}{2}\mathbf{w}^T\mathbf{w} - \sum_{i=1}^{N} \alpha_i\left[d_i(\mathbf{w}^T\mathbf{x}_i + b) - 1\right] \qquad (6.11)$$

α_j are called *Lagrange multipliers*.

代入

Taking directive:

Condition 1: $$\frac{\partial J(\mathbf{w}, b, \alpha)}{\partial \mathbf{w}} = \mathbf{0} \rightarrow \boldsymbol{w} = \sum_{i=1}^{N} \boldsymbol{\alpha}_i \boldsymbol{d}_i \vec{\boldsymbol{x}}_i$$

Condition 2: $$\frac{\partial J(\mathbf{w}, b, \alpha)}{\partial b} = 0 \rightarrow \sum_{i=1}^{N} \alpha_i d_i = \mathbf{0}$$

Plug w into J(w,b,α), we will have

$$\boldsymbol{J}(\boldsymbol{w}, \boldsymbol{b}, \boldsymbol{\alpha}) = \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w} - \sum_{i=1}^{N} \boldsymbol{\alpha}_i[\boldsymbol{d}_i(\boldsymbol{w}^T\boldsymbol{x}_i + \boldsymbol{b}) - 1]$$

$$\boldsymbol{b}\sum \boldsymbol{\alpha}_i \boldsymbol{d}_i = 0$$

$$\boldsymbol{w}^{\mathrm{T}}\boldsymbol{w}$$

$$= \sum_{i=1}^{N} \boldsymbol{\alpha}_i - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{M} \boldsymbol{d}_i \boldsymbol{d}_j \boldsymbol{\alpha}_i \boldsymbol{\alpha}_j \boldsymbol{x}_i^T \boldsymbol{x}_j$$

= J(α)

$$\max \quad Q(\alpha) = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j d_i d_j \mathbf{x}_i^T \mathbf{x}_j \qquad (6.16)$$

$$\Rightarrow \min \quad \frac{1}{2}\boldsymbol{\alpha}^T \begin{bmatrix} d_1 d_1 \mathbf{x}_1^T \mathbf{x}_1 & d_1 d_2 \mathbf{x}_1^T \mathbf{x}_2 & \cdots\cdot & d_1 d_N \mathbf{x}_1^T \mathbf{x}_N \\ d_2 d_1 \mathbf{x}_2^T \mathbf{x}_1 & d_2 d_2 \mathbf{x}_2^T \mathbf{x}_2 & \cdots\cdot & d_2 d_N \mathbf{x}_2^T \mathbf{x}_N \\ \cdots\cdot & \cdots\cdot & \cdots\cdot & \cdots\cdot \\ d_N d_1 \mathbf{x}_N^T \mathbf{x}_1 & d_N d_2 \mathbf{x}_N^T \mathbf{x}_2 & \cdots\cdot & d_N d_N \mathbf{x}_N^T \mathbf{x}_N \end{bmatrix} \boldsymbol{\alpha} + -1^T \boldsymbol{\alpha}$$

Such that $\mathbf{d}^T \boldsymbol{\alpha} = 0 \qquad 0 \le \boldsymbol{\alpha} \le \infty$

This can be rewritten as

$$\min \frac{1}{2}\boldsymbol{\alpha}^T \mathbf{Q} \boldsymbol{\alpha} - 1^T \boldsymbol{\alpha} \quad \text{subject to} \quad \mathbf{d}^T \boldsymbol{\alpha} = 0$$

此為標準二次程式 **convex function 之optimization**

類神經網路暨醫學影像處理實驗室

Let the optimization solutions be α_1 , α_2 , α_3 ,…, α_N

At saddle point, for each Lagrange multiplier, Kuhn-Tucker Conditions of Optimization theory

$$\alpha_i (d_i (\vec{w}_0^T \vec{x}_i + b\ ) - 1) = 0$$

Then we will have

$$\alpha_i = 0 \quad \text{or} \quad d_i (\vec{w}_0^T \vec{x}_i + b\ ) - 1 = 0$$

➔1: All interior points have $\alpha_i = 0$

2: if $\alpha_I > 0$, then x_i is a support vector

Then $\boldsymbol{w} = \sum_{i=1}^{N} \alpha_i d_i \vec{x}_i$ is determined only by the support vectors

Once w is obtained, $\boldsymbol{w} = \sum_{i=1}^{N} \boldsymbol{\alpha}_i \boldsymbol{d}_i \vec{\boldsymbol{x}}_i$ by the support vectors

we can plug the w into $d_i(\vec{w}_0^T \vec{x}_i + b) - 1 = 0$

to obtain the b

From $\boldsymbol{J}(\boldsymbol{\alpha}) = \sum_{i=1}^{N} \boldsymbol{\alpha}_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{M} \boldsymbol{d}_i \boldsymbol{d}_j \boldsymbol{\alpha}_i \boldsymbol{\alpha}_j \boldsymbol{x}_i^T \boldsymbol{x}_j$

What if the data are not linearly separable?

The minimization is on $x_i^T x_i$ if we can transform to minimize on $z_i^T z_{i,}$ it would still work

The Dual Problem

Given the training sample $\{(\mathbf{x}_i, d_i)\}_{i=1}^{N}$, *find the Lagrange multipliers* $\{\alpha_i\}_{i=1}^{N}$ *that maximize the objective function*

$$Q(\alpha) = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j d_i d_j \mathbf{x}_i^T \mathbf{x}_j$$

只含α_i

subject to the constraints

(1) $\sum_{i=1}^{N} \alpha_i d_i = 0$

(2) $\alpha_i \geq 0 \quad \text{for } i = 1, 2, \ldots, N$

由dual problem決定optimal α_i (叫做$\alpha_{o,i}$)

代入 (6.17)得到optimal weight $\mathbf{w}_o$ $\quad \mathbf{w}_o = \sum_{i=1}^{N} \alpha_{o,i} d_i \mathbf{x}_i$

代入 (6.18)得到optimal weight b_o

$$b_o = 1 - \mathbf{w}_o^T \mathbf{x}^{(s)} \qquad \text{for } d^{(s)} = 1 \qquad (6.18)$$

Optimal hyper-plane for non-separable patterns

之前討論for linear separable，現在為non-separable pattern (允許部分pattern落入partition margin內)：

$$d_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1 - \xi_i, \qquad i = 1, 2, \ldots, N \tag{6.22}$$

The ξ_i are called slack variables

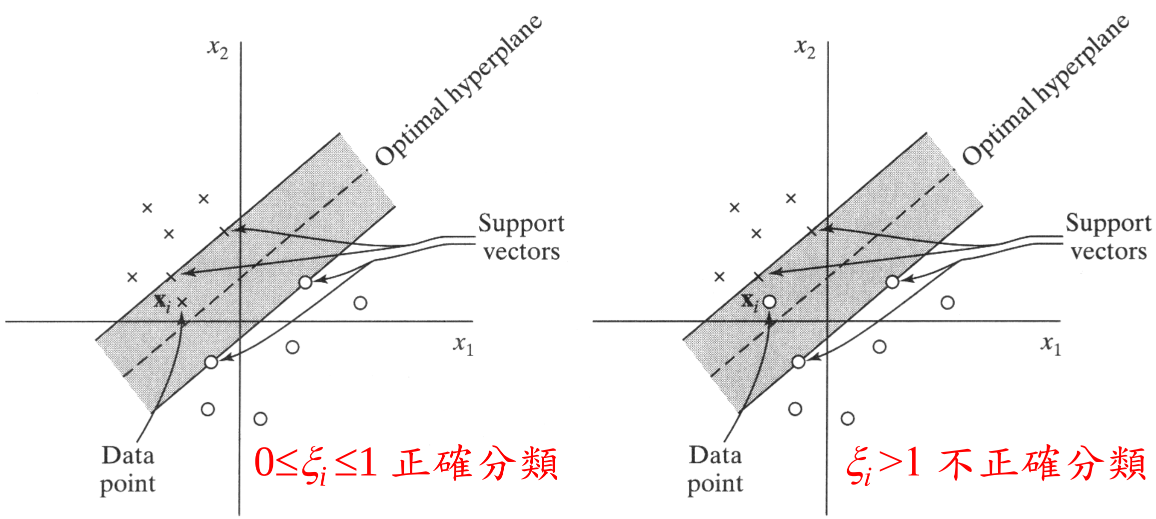


FIGURE 6.3 (a) Data point $\mathbf{x}_i$ (belonging to class $\mathscr{C}_1$) falls inside the region of separation, but on the right side of the decision surface. (b) Data point $\mathbf{x}_i$ (belonging to class $\mathscr{C}_2$) falls on the wrong side of the decision surface.

The misclassification error, averaged on the training set, is minimized

$$\Phi(\xi) = \sum_{i=1}^{N} I(\xi_i - 1)$$

$$I(\xi) = \begin{cases} 0 & \text{if } \xi \le 0 \\ 1 & \text{if } \xi > 0 \end{cases}$$

Correct but maybe inside the margin

Incorrect

為了計算方便更改為 $\Phi(\xi) = \sum_{i=1}^{N} \xi_i$

Then $\Phi(\mathbf{w}, \xi) = \frac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum_{i=1}^{N} \xi_i \qquad (6.23)$

The first term in Eq. (6.23) is related to minimizing the VC dimension of the support vector machine.
The second term is an upper bound on the number of the test errors.
The parameter C is user determined (1)*experimentally* (2)*analytically*.

Then for the soft classification:

Given the training sample $\{(\mathbf{x}_i, d_i)\}_{i=1}^{N}$, find the optimum values of the weight vector $\mathbf{w}$ *and bias b such that they satisfy the constraint*

$$d_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1 - \xi_i \qquad \text{for } i = 1, 2, \ldots, N$$

$$\xi_i \geq 0 \qquad \text{for all } i$$

and such that the weight vector $\mathbf{w}$ *and the slack variables* ξ_i *minimize the cost functional*

$$\Phi(\mathbf{w}, \xi) = \frac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum_{i=1}^{N}\xi_i$$

where C is a user-specified positive parameter.

Duality

Given the training sample $\{(\mathbf{x}_i, d_i)\}_{i=1}^{N}$, *find the Lagrange multipliers* $\{\alpha_i\}_{i=1}^{N}$ *that maximize the objective function*

$$Q(\alpha) = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j d_i d_j \mathbf{x}_i^T \mathbf{x}_j$$

subject to the constraints

(1) $\sum_{i=1}^{N} \alpha_i d_i = 0$

(2) $0 \le \alpha_i \le C \qquad \text{for } i = 1, 2, \ldots, N$

where C is a user-specified positive parameter.

Then the optimum solution for weight vector is

$$w_0 = \sum \alpha_{0i} d_i x_i$$

and the Kuhn-Tucker conditions are:

$$\alpha_i \left[d_i \left(\vec{w}^T x_i + b \right) - 1 + \xi_i \right] = 0 \qquad i = 1,2,\ldots,N$$

$$u_i \xi_i = 0 \qquad i = 1,2,\ldots,N$$

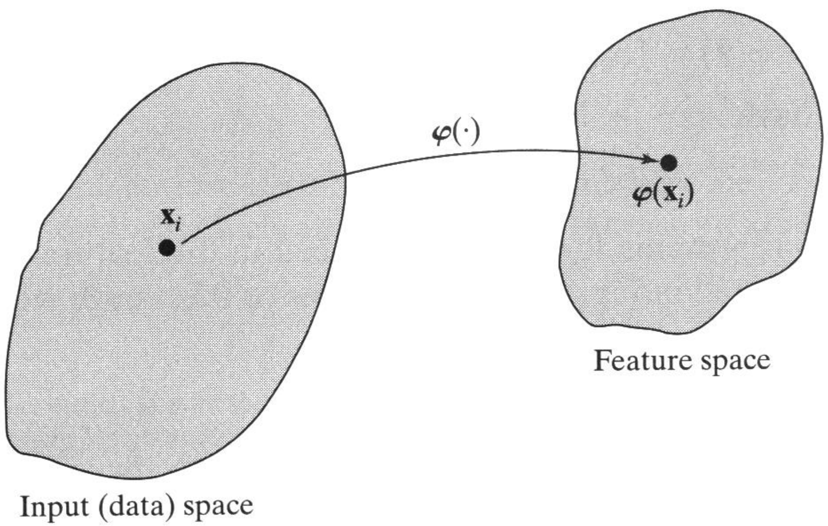


FIGURE 6.4 Nonlinear map $\varphi(\cdot)$ from the input space to the feature space.

We may define a hyperplane acting as the decision surface as follows

$$\sum_{j=1}^{m_1} w_j \varphi_j(\mathbf{x}) + b = 0 \qquad (6.29)$$

$$\mathbf{w}^T \boldsymbol{\varphi}(\mathbf{x}) = 0 \qquad (6.33)$$ Hyperplane

由6.12得知

$$\mathbf{w} = \sum_{i=1}^{N} \alpha_i d_i \boldsymbol{\varphi}(\mathbf{x}_i) \qquad (6.34)$$

6.34代入6.33

$$\sum_{i=1}^{N} \alpha_i d_i \boldsymbol{\varphi}^T(\mathbf{x}_i)\boldsymbol{\varphi}(\mathbf{x}) = 0 \qquad (6.35)$$

The term $\boldsymbol{\varphi}^T(x_i)\boldsymbol{\varphi}(x)$ represents the inner product of two vectors induced in the feature space by the input vector $\boldsymbol{x}$ and the input pattern $\boldsymbol{x}_i$

$$\begin{aligned} K(\mathbf{x}, \mathbf{x}_i) &= \boldsymbol{\varphi}^T(\mathbf{x})\boldsymbol{\varphi}(\mathbf{x}_i) \\ &= \sum_{j=1}^{m_1} \varphi_j(\mathbf{x})\varphi_j(\mathbf{x}_i) \qquad \text{for } i = 1, 2, \ldots, N \end{aligned} \qquad (6.36)$$

$$K(\mathbf{x}, \mathbf{x}_i) = K(\mathbf{x}_i, \mathbf{x}) \qquad \text{for all } i \qquad (6.37)$$

$$\sum_{i=1}^{N} \alpha_i d_i K(\mathbf{x}, \mathbf{x}_i) = 0 \qquad (6.38)$$

K is a symmetric function

TABLE 6.1 Summary of Inner-Product Kernels

Type of support vector machine	Inner product kernel $K(\mathbf{x}, \mathbf{x}_i)$, $i = 1, 2, \ldots, N$	Comments
Polynomial learning machine	$(\mathbf{x}^T\mathbf{x}_i + 1)^p$	Power p is specified *a priori* by the user
Radial-basis function network	$\exp\left(-\frac{1}{2\sigma^2}\lVert\mathbf{x} - \mathbf{x}_i\rVert^2\right)$	The width σ^2, common to all the kernels, is specified *a priori* by the user
Two-layer perceptron	$\tanh(\beta_0\mathbf{x}^T\mathbf{x}_i + \beta_1)$	Mercer's theorem is satisfied only for some values of β_0 and β_1

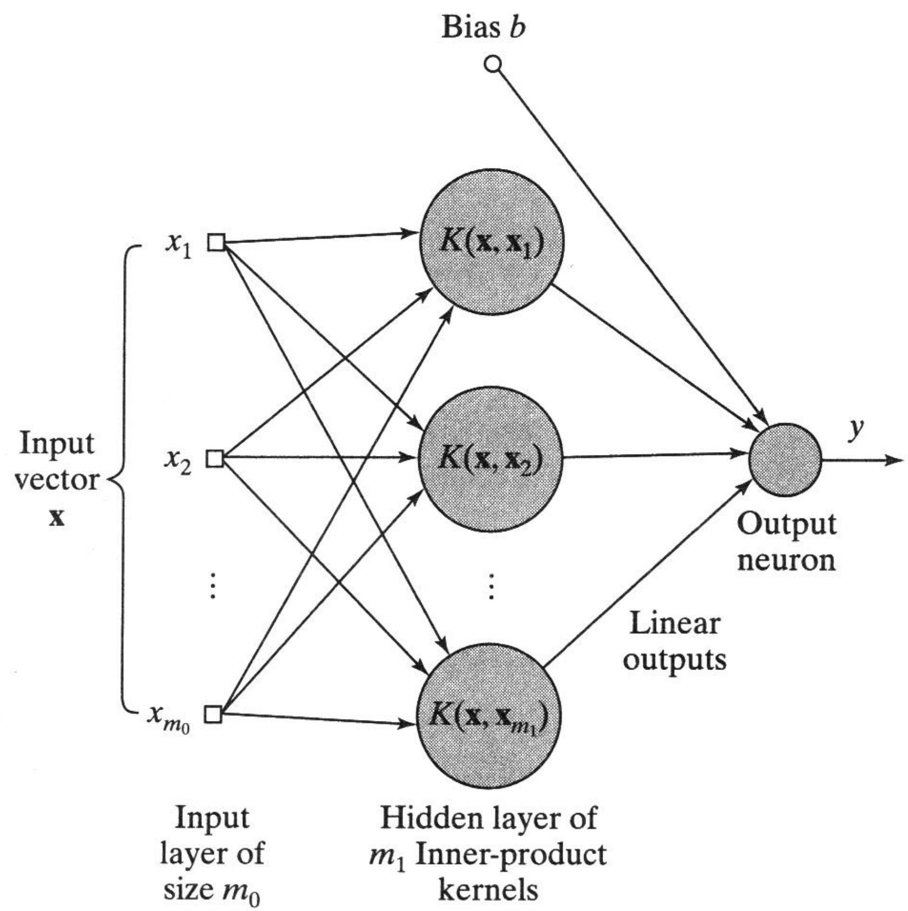


FIGURE 6.5 Architecture of support vector machine.