Support Vector Machines

LINEAR SEPARABILITY AND OPTIMAL HYPERPLANE

Consider the training sample $\{(\mathbf{x}_i, d_i)\}_{i=1}^{N}$ where $\mathbf{x}_i$ is the input pattern for the i_{th} example and d_i is the corresponding desired response. We assume that the pattern (class) represented by the subset $d_i = +1$ and the pattern represented by the subset $d_i = -1$ are "linearly separable."

The equation of a decision surface in the form of a hyperplane that does the separation is

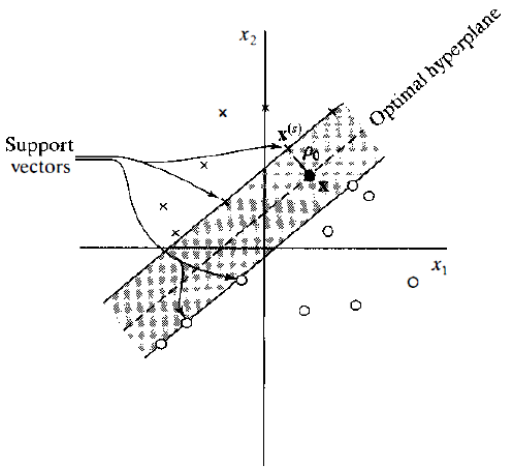
$$\mathbf{w}^T\mathbf{x} + b = 0$$

where x is an input vector, w is an adjustable weight vector, and b is a bias, We may write

$$\mathbf{w}^T\mathbf{x}_i + b \geq 0 \qquad \text{for } d_i = +1$$
$$\mathbf{w}^T\mathbf{x}_i + b < 0 \qquad \text{for } d_i = -1$$

For a given weight vector wand bias b, the separation between the hyperplane and the closest data point is called the margin of separation, denoted by ρ. The goal of a support vector machine is to find the particular hyperplane for which the margin of separation ρ is maximized. Under this condition, the decision surface is referred to as the optimal hyperplane.

Following Figureillustrates the geometric construction of an optimal hyperplane for a two-dimensional input space.



Let $\mathbf{w}_o$ and b_o denote the optimum values of the weight vector and bias, respectively.

The optimal hyperplane representing a multidimensional lin-ear decision surface in the input space, is defined by $\mathbf{w}_o^T\mathbf{x} + b_o = 0$

The discriminant function $g(\mathbf{x}) = \mathbf{w}_o^T\mathbf{x} + b_o$ gives an algebraic measure of the distance from x to the optimal hyperplane. Perhaps the easiest way to see this is to express x as $\mathbf{x} = \mathbf{x}_p + r\frac{\mathbf{w}_o}{\|\mathbf{w}_o\|}$

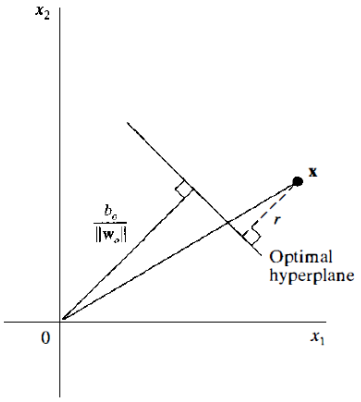
where $\mathbf{x}_p$ is the normal projection of x onto the optimal hyperplane, and r is the desired algebraic distance; r is positive if x is on the positive side of the optimal hyperplane and negative if x is on the negative side. Since $g(\mathbf{x}_p) = 0$ it follows that

$$g(\mathbf{x}) = \mathbf{w}_o^T\mathbf{x} + b_o = r\|\mathbf{w}_o\|$$

(or)

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

In particular, the distance from the origin (i.e., x = 0) to the optimal hyperplane is given $b_o/\|\mathbf{w}_o\|$.
If $b_o > 0$, the origin is on the positive side of the optimal hyperplane; if $b_o < 0$, it is on the negative side.
If $b_o = 0$, the optimal hyperplane passes through the origin.
A geometric interpretation of these algebraic results is given in following figure.



The pair ($\mathbf{w}_o$,b_o) must satisfy the constraint

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

The particular data points$(\mathbf{x}_i, d_i)$ for which the first or second line is satisfied with the equality sign are called support vector machine. These vectors play a prominent role in the operation of this class of learning machines.

Consider a support vector $\mathbf{x}^{(s)}$ for which $d^{(s)} = +1$. Then by definition, we have

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

the algebraic distance from the support vector $\mathbf{x}^{(s)}$ to the optimal hyperplane is

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

where the plus sign indicates that $\mathbf{x}^{(s)}$ lies on the positive side of the optimal hyperplane and the minus sign indicates that $\mathbf{x}^{(s)}$ lies on the negative side of the optimal hyper-plane.
Let ρ denote the optimum value of the margin of separation between the two classes that constitute the training set $\mathcal{T}$. Then, we follows that

$$\rho = 2r$$
$$= \frac{2}{\|\mathbf{w}_o\|}$$

It states that maximizing the margin of separation between classes is equivalent to minimizing the Euclidean norm of the weight vector w.

DETERMINATION OF OPTIMAL HYPERPLANE

Our goal is to develop a computationally efficient procedure for using the training sample $\mathcal{T} = \{(\mathbf{x}_i, d_i)\}_{i=1}^{N}$ to find the optimal hyperplane, subject to the constraint

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

The constrained optimization problem that we have to solve may now be stated as

Given the training sample $\{(\mathbf{x}_i, d_i)\}_{i=1}^{N}$ find the optimum values of the weight vector wand bias b such that they satisfy the constraints $d_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1 \qquad \text{for } i = 1, 2, \ldots, N$

and the weight vector w minimizes the cost function: $\Phi(\mathbf{w}) = \frac{1}{2}\mathbf{w}^T\mathbf{w}$

The scaling factor 1/2 is included here for convenience of presentation. This constrained optimization problem is called the primal problem. It is characterized as follows:

• The cost function $\Phi(\mathbf{w})$ is a convex function of w.

• The constraints are linear in w.

Accordingly, we may solve the constrained optimization problem using the method of Lagrange multipliers.

First, we construct the Lagrangian function:

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

where the auxiliary nonnegative variables ai are called Lagrange multipliers. The solution to the constrained optimization problem is determined by the saddle point of the Lagrangian function $J(\mathbf{w}, b, \alpha)$ which has to be minimized with respect to wand b; it also has to be maximized with respect to 0..

Thus, differentiating $J(\mathbf{w}, b, \alpha)$ with respect to wand b and setting the results equal to zero, we get the following two conditions of optimality:

$$\text{Condition 1:} \qquad \frac{\partial J(\mathbf{w}, b, \alpha)}{\partial \mathbf{w}} = \mathbf{0}$$

$$\text{Condition 2:} \qquad \frac{\partial J(\mathbf{w}, b, \alpha)}{\partial b} = 0$$

Application of optimality condition 1 to the Lagrangian function yields

$$\mathbf{w} = \sum_{i=1}^{N}\alpha_i d_i \mathbf{x}_i$$

Application of optimality condition 2 to the Lagrangian function yields

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

The solution vector w is defined in terms of an expansion that involves the N training examples. However this solution is unique by virtue of the convexity of the Lagrangian, the same cannot be said about the Lagrange coefficients α_i.

It is also important to note that at the saddle point, for each Lagrange multiplier ai' the product of that multiplier with its corresponding constraint vanishes, as shown by

$$\alpha_i\left[d_i(\mathbf{w}^T\mathbf{x}_i + b) - 1\right] = 0 \qquad \text{for } i = 1, 2, \ldots, N$$

This property follows from the Kuhn -Tucker conditions of optimization theory.

The primal problem deals with a convex cost function and linear constraints. Given such a constrained optimization problem, it is possible to construct another problem called the dual problem. This second

problem has the same optimalvalue as the primal problem, but with the Lagrange multipliers providing the optimal solution, In particular, we may state the following duality theorem

(a) If the primal problem has an optimal solution, the dual problem also has an optimal solution, and the corresponding optimal values are equal.

(b) In order for $\mathbf{w}_o$ to be an optimal primal solution and α_o to be an optimal dual solution, it is necessary and sufficient that $\mathbf{w}_o$ is feasible for the primal problem,

$$\Phi(\mathbf{w}_o) = J(\mathbf{w}_o, b_o, \alpha_o) = \min_{\mathbf{w}} J(\mathbf{w}, b_o, \alpha_o)$$

To postulate the dual problem for our primal problem, we expnd term by term, as follows:

$$J(\mathbf{w}, b, \alpha) = \frac{1}{2}\mathbf{w}^T\mathbf{w} - \sum_{i=1}^{N} \alpha_i d_i \mathbf{w}^T\mathbf{x}_i - b\sum_{i=1}^{N} \alpha_i d_i + \sum_{i=1}^{N} \alpha_i$$

The third term on the right-hand side of above equation is zero by virtue of the optimality condition. Then we have

$$\mathbf{w}^T\mathbf{w} = \sum_{i=1}^{N} \alpha_i d_i \mathbf{w}^T\mathbf{x}_i = \sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_i\alpha_j d_i d_j \mathbf{x}_i^T\mathbf{x}_j$$

Accordingly, setting the objective function $J(\mathbf{w}, b, \alpha) = Q(\alpha)$, we may reformulate

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

where the α_i are nonnegative.

We may now state 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

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

subject to the constraints

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

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

The function $Q(\alpha)$ to be maximized depends only on the input patterns in the form of a set of dot products, $\{\mathbf{x}_i^T\mathbf{x}_j\}_{(i,j)=1}^{N}$

Having determined the optimum Lagrange multipliers, denoted by $\alpha_{o,i}$ we may compute the optimum weight vector $\mathbf{w}_o$ so write

$$\mathbf{w}_o = \sum_{i=1}^{N} \alpha_{o,i} d_i \mathbf{x}_i$$

To compute the optimum bias b_o we may use the $\mathbf{w}_o$ thus write

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

OPTIMAL HYPERPLANE FOR NONSEPARABLE PATTERNS

In this section we consider the more difficult case of nonseparable patterns. Given such a set of training data, it is not possible to construct a separating hyperplane without encountering classification errors. So we would like to find an optimal hyperplane that minimizes the probability of classification error, averaged over the training set.

The margin of separation between classes is said to be soft if a data point $(\mathbf{x}_i, d_i)$ violates the following condition $d_i(\mathbf{w}^T\mathbf{x}_i + b) \geq +1, \quad i = 1, 2, \ldots, N$

This violation can arise in one of two ways:

• The data point $(\mathbf{x}_i, d_i)$ falls inside the region of separation but on the right side of the decision surface, as illustrated in Fig. a .

• The data point $(\mathbf{x}_i, d_i)$ falls on the wrong side of the decision surface, as illustrated in Fig. b.

Note that we have correct classification in case 1, but misclassification in case 2

To set the stage for a formal treatment of nonseparable data points, we introduce a new set of nonnegative scalar variables, $\{\xi_i\}_{i=1}^N$ into the definition of the separating hyperplane (i.e., decision surface) as shown here:

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

The ξ_i are called slack variables; they measure the deviation of a data point from the ideal condition of pattern separability. For $0 \leq \xi_i \leq 1$, the data point falls inside the region of separation but on the right side of the decision surface, as illustrated in Fig. a. For $\xi_i > 1$, it falls on the wrong side of the separating hyperplane, as illustrated in Fig. b.

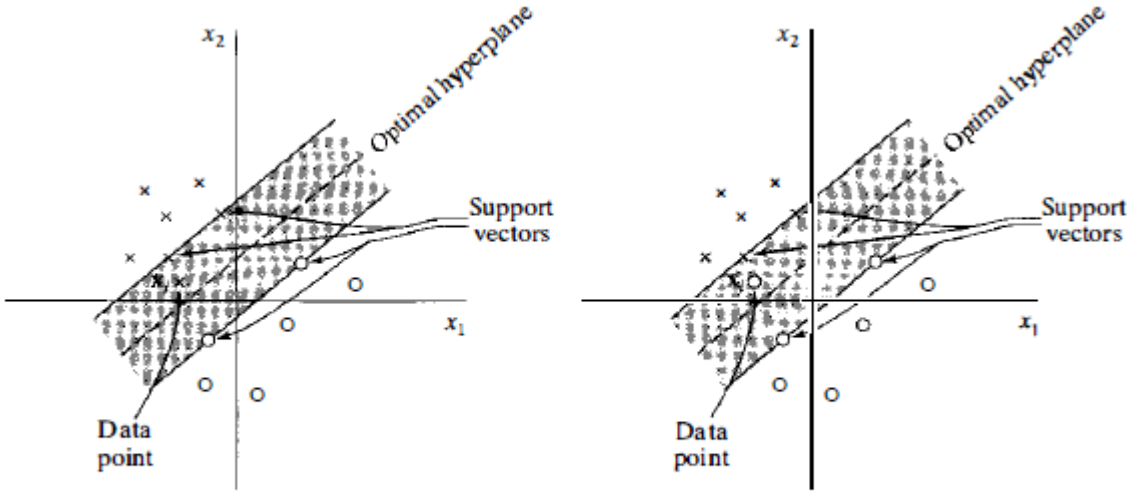


FIGURE (a) Data point $\mathbf{x}_i$ (belonging to class $\mathscr{C}_1$) falls inside the region of separation, but on the right side of the decision surface. (b) Data point $\mathbf{x}_i$ (belonging to class $\mathscr{C}_2$) falls on the wrong side of the decision surface.

The support vectors are those particular data points that satisfy precisely even if $\xi_i > 0$. Note that if an example with $\xi_i > 0$ is left out of the traiuiug set, the decision surface would change. The support vectors are thus defined in exactly the same way for both linearly separable and nonseparable cases.

Our goal is to find a separatiug hyperplane for which the misciassification error, averaged on the training set, is minimized.

We may do this by minimizing the functional $\Phi(\xi) = \sum_{i=1}^{N} I(\xi_i - 1)$ with respect to the weight vector w, subject to the constraint on $\|\mathbf{w}\|^2$. The function $I(\xi)$is an indicator function, defined by

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

Unfortunately, minimization of $\Phi(\xi)$ with respect to w is a nonconvex optimization problem that is NP-complete.

To make the optimization problem mathematically tractable, we approximate the functional $\Phi(\xi)$ *by writing*

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

we simplify the computation by formulating the fuuctional to be minimized with respect to the weight vector w as follows

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

The parameter C controls the tradeoff between complexity of the machine and the number of nonseparable points; it may therefore be viewed as a form of a "regularization" parameter. The parameter C has to be selected by the user. This can be done in one of two ways:

• The parameter C is determined experimentally via the standard use of a training! (validation) test set, which is a crude form of resampling .

• It is determined analytically by estimating the VC dimension via Eq. (6.19) and then by using bounds on the generalization performance of the machine based on the VC dimension.

In any event, the functional $\Phi(\mathbf{w},\xi)$ is optimized with respect to w and $\{\xi_i\}_{i=1}^{N}$ and $\xi_i \geq 0$. In so doing, the squared norm of w is treated as a quantity to be jointly minimized with respect to the nonseparable points rather than as a constraint imposed on the minimization of the number of nonseparable points.

The optimization problem for nonseparable patterns just stated, includes the optimization problem for linearly separable patterns as a special case.

→We may now formally state the primal problem for the nonseparable case as:

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

$$d_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1 - \xi_i \quad \text{for } i = 1, 2, \ldots, N$$
$$\xi_i \geq 0 \quad \text{for all } i$$

and such that the weight vector 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.

→we may formulate the dual problem for nonseparable patterns as:

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

Note that neither the slack variables ξ_i ,nor their Lagrange multipliers appear in the dual problem. The dual problem for the case of nonseparable patterns is thus similar to that for the simple case of linearly separable patterns except for a minor but important difference. The objective function $Q(\alpha)$ to be maximized is the same in both cases, The nonseparable case differs from the separable case in that the constraint $\alpha_i \ge 0$ is replaced with the more stringent constraint $0 \le \alpha_i \le C$.

The optimum solution for the weight vector W is given by

$$\mathbf{w}_o = \sum_{i=1}^{N_S} \alpha_{o,i} d_i \mathbf{x}_i$$

where N_S is the number of support vectors. The determination of the optimum values of the bias also follows a procedure similar to that described before. Specifically, the Kuhn-Tucker conditions are now defined by

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

and

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

The derivative of the Lagrangian function for the primal problem with respect to the slack variable ξ_i is zero, the evaluation of which yields

$$\alpha_i + \mu_i = C$$

By combining the above two equations we see that

$$\xi_i = 0 \quad \text{if} \quad \alpha_i < C$$

We may determine the optimum bias bo by taking any data point $(\mathbf{x}_i, d_i)$ in the training set for which we have $0 < \alpha_{o.i} < C$ and therefore $\xi_i = 0$.

DESIGN OF SVM (SUPPORT VECTOR MACHINES)

Basically, the idea of a support vector machine3 hinges on two mathematical operations summarized here and illustrated in following figure.

1. Nonlinear mapping of an input vector into a high-dimensional feature space that is hidden from both the input and output.

2. Construction of an optimal hyperplane for separating the features discovered in step 1.

The rationale for each of these two operations is explained in what follows.

→Operation 1 is performed in accordance with Cover's theorem on the separability of patterns. Consider an input space made up of nonlinearly separable patterns. Cover's theorem states that such a multidimensional space may be transformed into a new feature space where the patterns are linearly separable with high probability, provided two conditions are satisfied. First, the transformation is

nonlinear. Second, the dimensionality of the feature space is high enough. These two conditions are embodied in operation 1.

→Operation 2 exploits the idea of building an optimal separating hyperplane in accordance with the theory but with a fundamental difference:

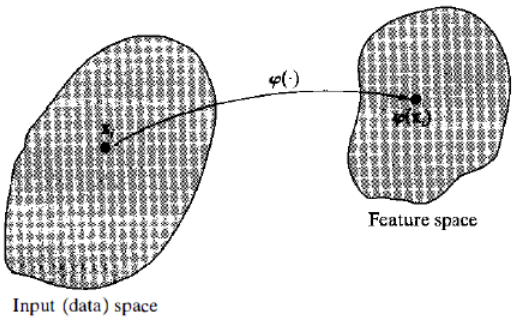


FIGURE 6.4 Nonlinear map φ(·) from the input space to the feature space.

The expansion of the inner-product kernel K(x, x_i,) permits us to construct a decision surface that is nonlinear in the input space, but its image in the feature space is linear. With this expansion at hand, we may now state the dual form for the constrained optimization of a support vector machine as follows:

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

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

Subject to constraints

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

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

where C is a user-specified positive parameter.

Note that constraint (1) arises from optimization of the Lagrangian $Q(\alpha)$ with respect to the bias b = $\mathbf{w}_o$ for $\varphi_0(\mathbf{x}) = 1$. The dual problem just stated is of the same form as that for the case of nonseparable patterns, except for the fact that the inner product $\mathbf{x}_i^T \mathbf{x}_j$ used therein has been replaced by the inner-product kernel $K(\mathbf{x}_i, \mathbf{x}_j)$. We may view $K(\mathbf{x}_i, \mathbf{x}_j)$ as the ij-th element of a symmetric N-by-N matrix K, as shown by

$$\mathbf{K} = \{K(\mathbf{x}_i, \mathbf{x}_j)\}_{(i,j)=1}^{N}$$

Having found the optimum values of the Lagrange multipliers, denoted by $\alpha_{o,i}$, we may determine the corresponding optimum value of the linear weight vector, $\mathbf{w}_o$ connecting the feature space to the output space. Specifically, recognizing that the image $\varphi(\mathbf{x}_i)$ plays the role of input to the weight vector w, we may define $\mathbf{w}_o$ as

$$\mathbf{w}_o = \sum_{i=1}^{N} \alpha_{o,i} d_i \varphi(\mathbf{x}_i)$$

where $\varphi(\mathbf{x}_i)$ is the image induced in the feature space due to x_i. Note the first component of $\mathbf{w}_o$ represents the optimum bias b_o.

EXAMPLE OF SVM

The requirement on the kernel $K(\mathbf{x}_i, \mathbf{x}_j)$ is to satisfy Mercer's theorem. Within this requirement there is some freedom in how it is chosen. In the following table we summarize the inner-product kernels for three common types of support vector machines: polynomial learning machine, radial-basis function network, and two-layer perceptron. The following points are noteworthy:

1. The inner-product kernels for polynomial and radial-basis function types of support vector machines always satisfy Mercer's theorem. In contrast. the innerproduct kernel for a two-layer perceptron type of support vector machine is somewhat restricted.
2. For all three machine types, the dimensionality of the feature space is determined by the number of support vectors extracted from the training data by the solution to the constrained optimization problem.
3. The underlying theory of a support vector machine avoids the need for heuristics often used in the design of conventional radial-basis function networks and multilayer perceptrons:

• In the radial-basis function type of a support vector machine, the number of radial-basis functions and their centers are determined automatically by the number of support vectors and their values, respectively,

• In the two-layer perceptron type of a support vector machine, the number of hidden neurons and their weight vectors are determined automatically by the number of support vectors and their values, respectively.

TABLE 6.1 Summary of Inner-Product Kernels

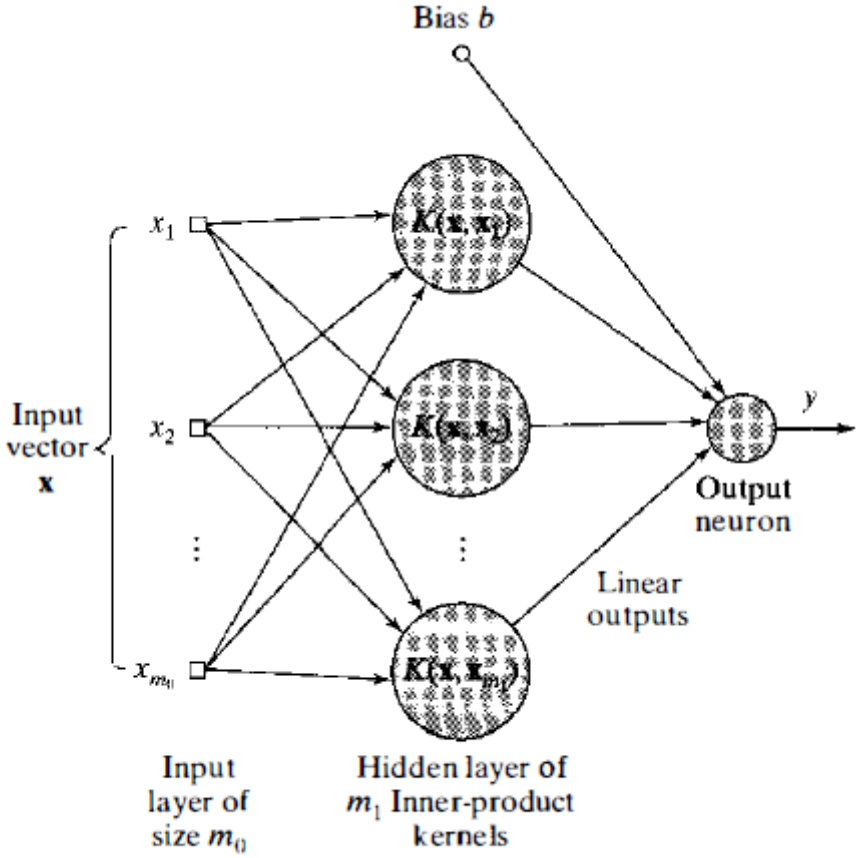
Type of support vector machine	Inner product kernel $K(\mathbf{x}, \mathbf{x}_i)$, $i = 1, 2, \ldots, N$	Comments
Polynomial learning machine	$(\mathbf{x}^T\mathbf{x}_i + 1)^p$	Power p is specified *a priori* by the user
Radial-basis function network	$\exp\left(-\frac{1}{2\sigma^2}\|\mathbf{x} - \mathbf{x}_i\|^2\right)$	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

Following figure displays the architecture of a support vector machine. Irrespective of how a support vector machine is implemented, it differs from the conventional approach to the design of a multilayer perceptron in a fundamental way. In the conventional approach, model complexity is controlled by keeping the number of features (i.e., hidden neurons) small. On the other hand, the support vector machine offers a solution to the design of a learning machine by controlling model complexity independently of dimensionality, as summarized here:

• **Conceptual problem:** Dimensionality of the feature (hidden) space is purposely made very large to enable the construction of a decision surface in the form of a hyperplane in that space. For good

generalization performance, the model complexity is controlled by imposing certain constraints on the construction of the separating hyperplane, which results in the extraction of a fraction of the training data as support vectors.

• **Computational problem:** Numerical optimization in a high-dimensional space suffers from the curse of dimensionality. This computational problem is avoided by using the notion of an inner-product kernel and solving the dual form of the constrained optimization problem formulated in the input space.



Architecture of support vector machine.