UNIT-VI

SUPPORT VECTOR MACHINES

- **Linear separability and optimal hyperplane**
- **Determination of optimal hyperplane**
- **Optimal hyperplane for non-separable patterns**
- **Design of an SVM.**
- **Examples of SVM.**

Regression

Linear Regression

- Dependent variable is continuous in nature

(simple LR, Multiple LR)

$y = \alpha_0 + \alpha_1 x_1$ (y= c+mx)

$y = \alpha_0 + \alpha_1 x_1 + \alpha_2 x_2 + \alpha_3 x_3 + \ldots\ldots\ldots\ldots\ldots\ldots\ldots + \alpha_m x_m$

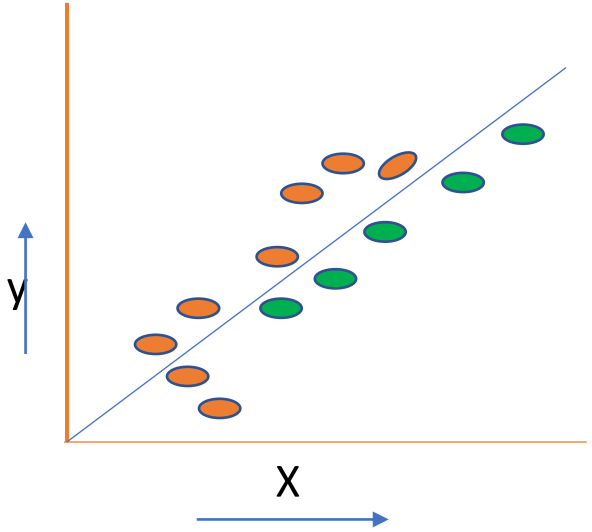
$y = \alpha_0 + \sum_{i=1}^{m} \alpha_i x_i$

$\alpha_i \rightarrow regression\ coef$

x_i -> independent variable

Y = 0.9 + 1.2 x_1 +2 x_2 +4 x_3 + 1 x_5

Linear model when applied on non-linear data errors are obtained, high loss, less accuracy



Non linear Regression

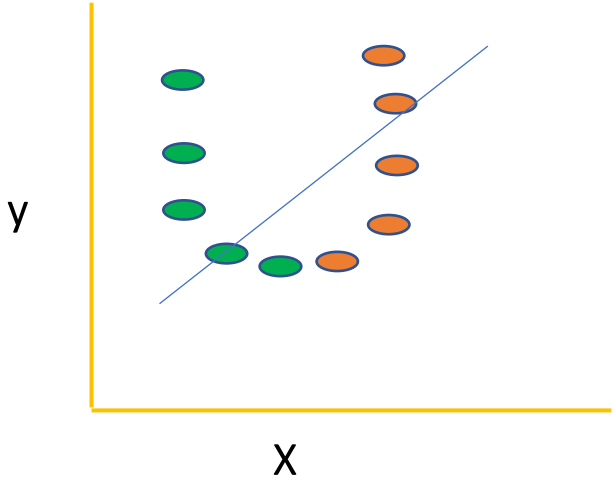
0 - > y = constant

1 -> y = mx +c

2 -> y = a x^2 + b x^1+c

$y = \alpha_0 + \alpha_1 x^1 + \alpha_2 x^2 + \alpha_3 x^2 + \ldots\ldots\ldots\ldots\ldots\ldots\ldots + \alpha_n x^n$

$y = \alpha_0 + \sum_{i=1}^{m} \alpha_i x_i$



Radial Basis Function (RBF)

1. Multiquadrics:

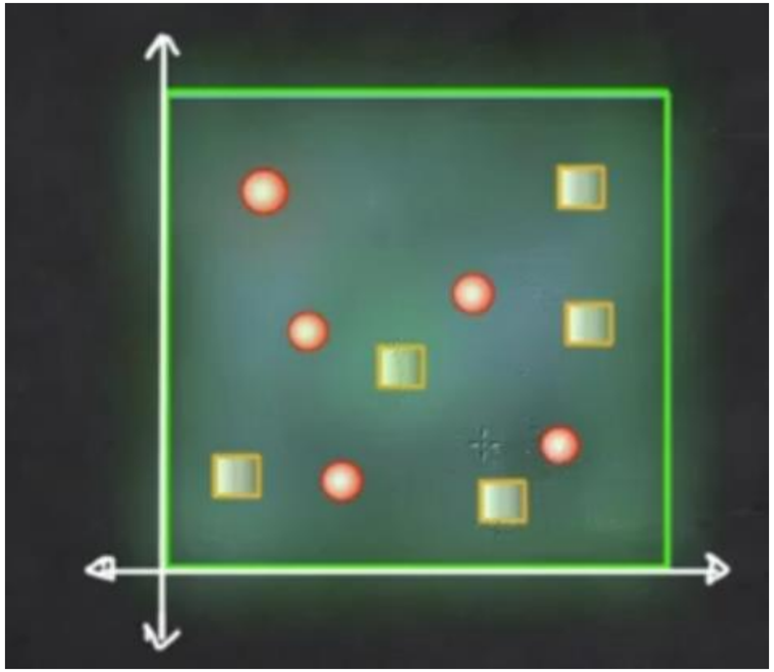
$$\varphi(r) = (r^2 + c^2)^{1/2} \quad \text{for some } c > 0 \text{ and } r \in \mathbb{R}$$

2. Inverse multiquadries:

$$\varphi(r) = \frac{1}{(r^2 + c^2)^{1/2}} \quad \text{for some } c > 0 \text{ and } r \in \mathbb{R}$$

3. Gaussian functions:

$$\varphi(r) = \exp\left(-\frac{r^2}{2\sigma^2}\right) \quad \text{for some } \sigma > 0 \text{ and } r \in \mathbb{R}$$



2D input space where circles ->c1, square -> c2

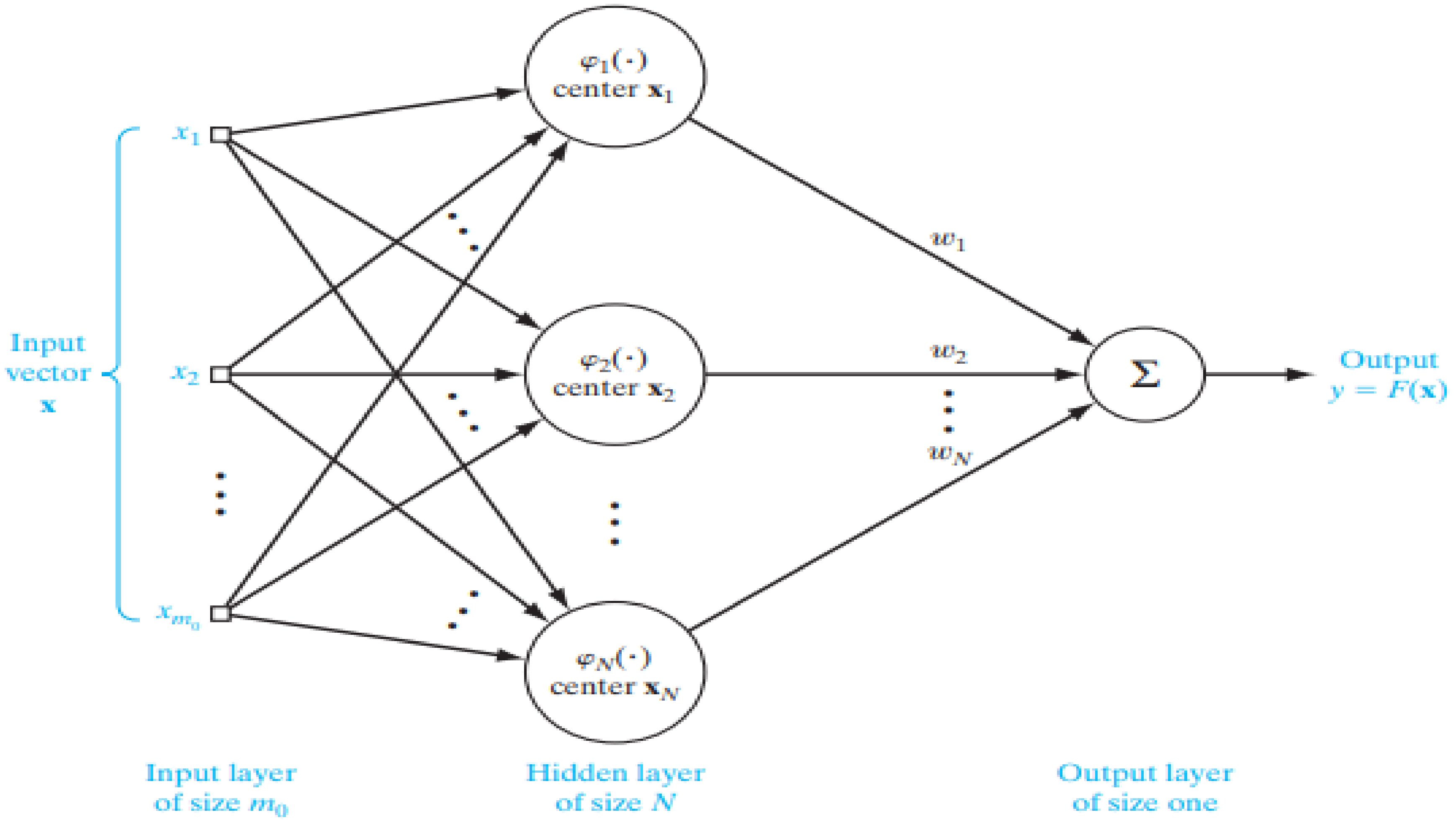
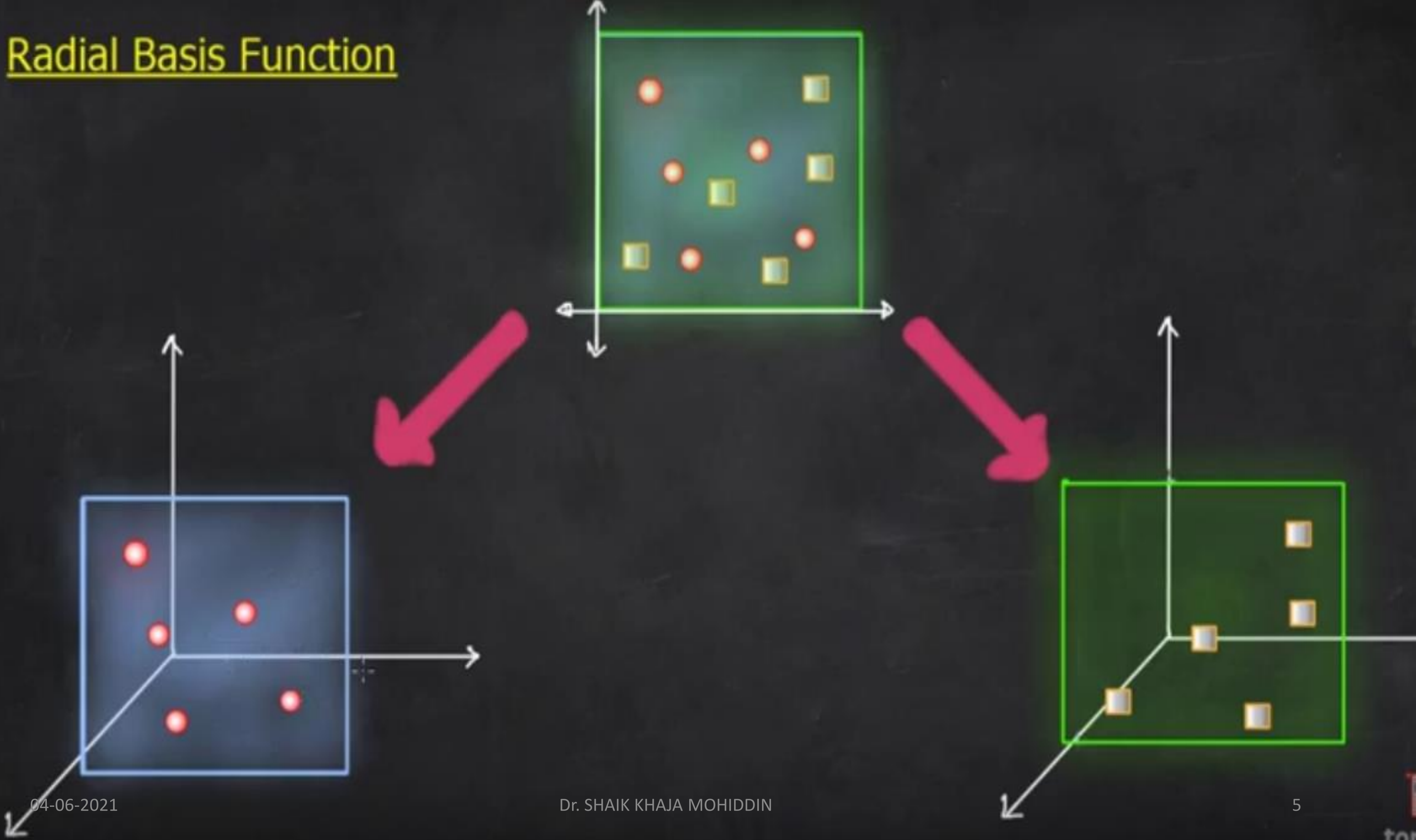
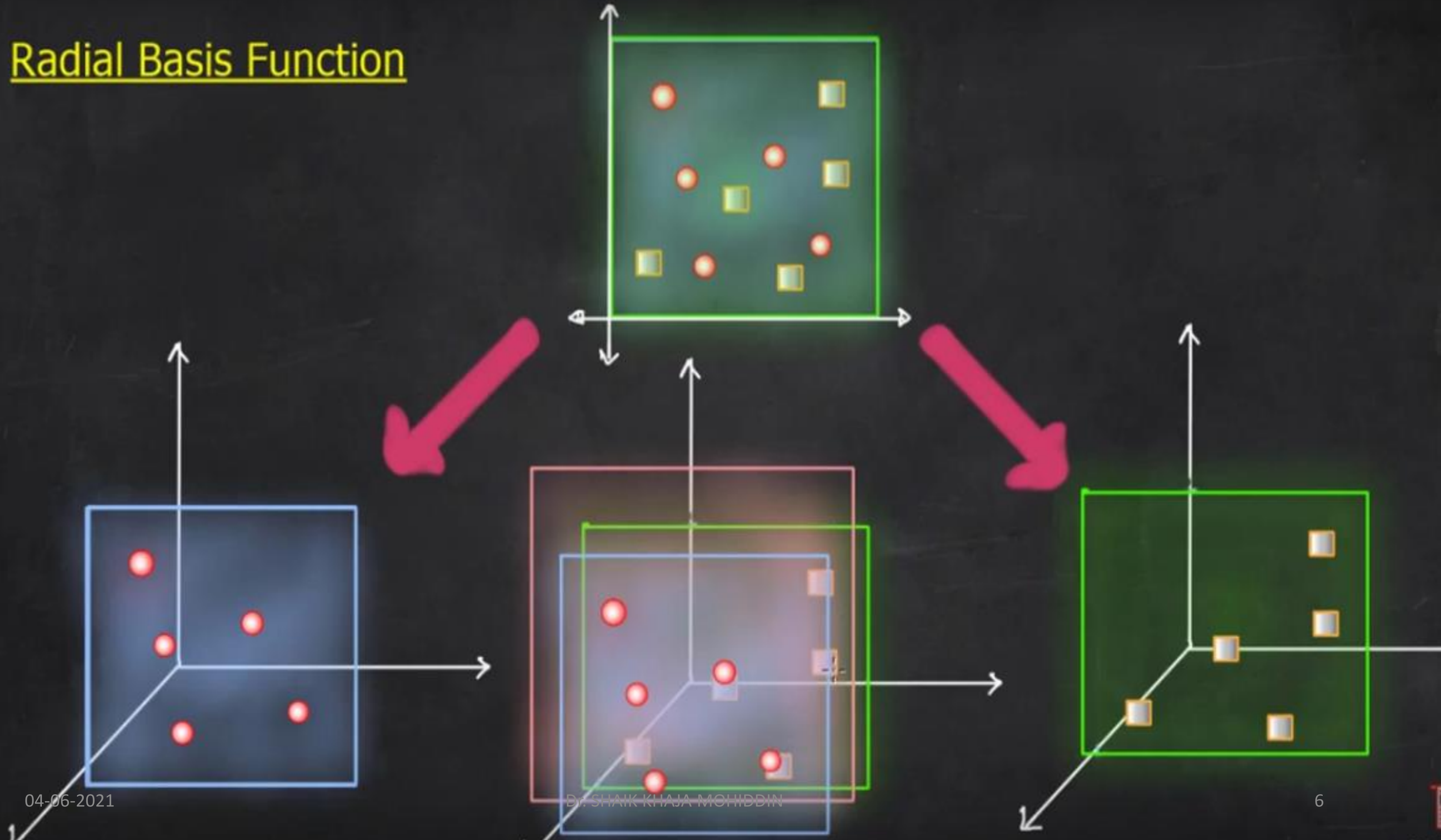

$\varphi_1(\cdot)$ center $\mathbf{x}_1$
$\varphi_2(\cdot)$ center $\mathbf{x}_2$
$\varphi_N(\cdot)$ center $\mathbf{x}_N$
Input vector $\mathbf{x}$
x_1
x_2
x_{m_0}
w_1
w_2
w_N
Σ
Output $y = F(\mathbf{x})$
Input layer of size m_0
Hidden layer of size N
Output layer of size one

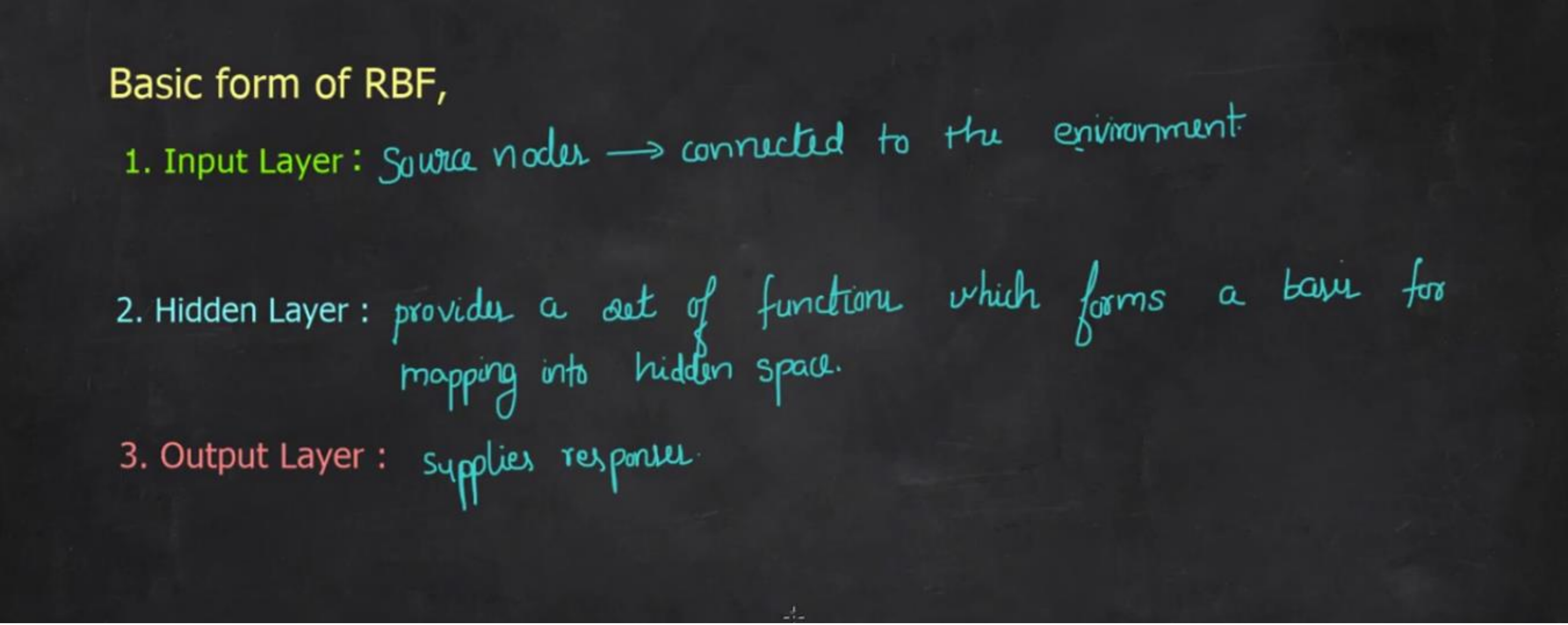
FIGURE 5.3 Structure of an RBF network, based on interpolation theory.

Radial Basis Function



Radial Basis Function





Basic form of RBF,

1. Input Layer : Source nodes → connected to the environment.

2. Hidden Layer : provides a set of functions which forms a basis for mapping into hidden space.

3. Output Layer : supplies responses.

Example. The XOR Problem

To illustrate the significance of the idea of φ-separability of patterns, consider the simple yet important XOR problem. In the XOR problem there are four points (patterns): (1, 1), (0, 1),(0, 0), and (1, 0), in a two-dimensional input space, as depicted in Fig. a. The requirement is to construct a pattern classifier that produces the binary output 0 in response to the input pattern (1, 1), or (0, 0), and the binary output 1 in response to the input pattern (0, 1) or (1. 0). Thus points that are closest in the input space, in terms of the Hamming distance, map to regions that are maximally apart in the output space.

Define a pair of Gaussian hidden functions as follows:

$$\varphi_1(\mathbf{x}) = e^{\|\mathbf{x}-\mathbf{t}_1\|^2}, \qquad \mathbf{t}_1 = [1, 1]^T$$

$$\varphi_2(\mathbf{x}) = e^{-\|\mathbf{x}-\mathbf{t}_2\|^2}, \qquad \mathbf{t}_2 = [0, 0]^T$$

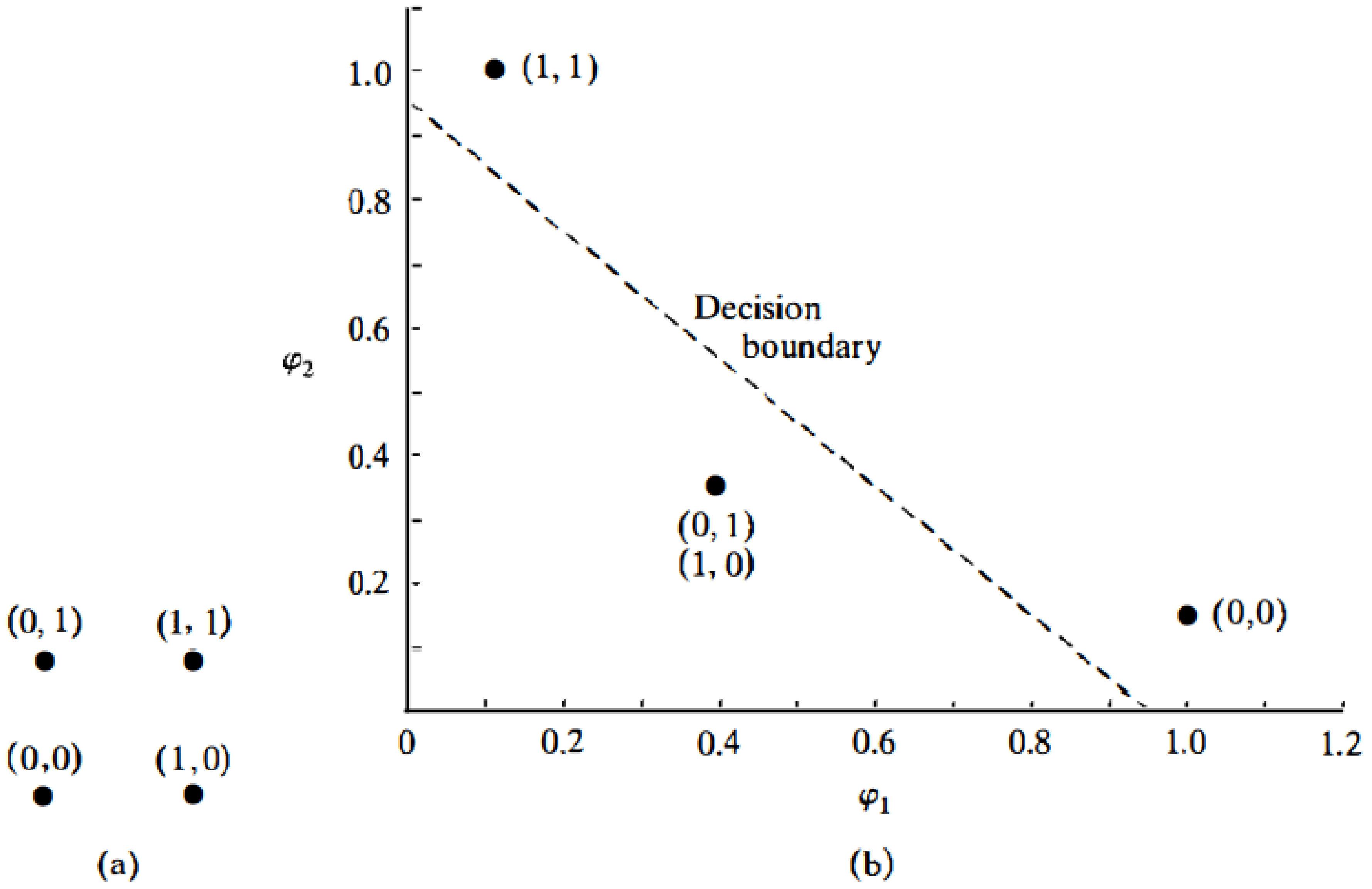


FIGURE 5.2 (a) The four patterns of the XOR problem; (b) Decision-making

Property / approach	RBF	MLP
Deciding the no. of hidden layer	Yes (one)	no
Decide the no. of nodes in hidden layer	yes	no
Training	Fast	slow
Interpretation of functionality of hidden neurons	Easy	difficult
Disadvantage		
Classification time	high	less

- Support vector machines (SVM), pioneered by Vapnik (Boser, Guyon, and Vapnik, 1992; Cortes and Vapnik, 1995; Vapnik, 1995, 1998).
- Like multilayer perceptrons and radial-basis function networks, support vector machines can be used for pattern classification and nonlinear regression.
- Main idea of a support vector machine is to construct a hyperplane as the decision surface in such a way that the margin of separation between positive and negative examples is maximized.
- The support vector learning algorithm to construct the following three types of learning machines (among others):
- • Polynomial learning machines
- • Radial-basis function networks
- • Two-layer perceptrons (i.e., with a single hidden layer)

LINEAR SEPARABILITY AND OPTIMAL HYPERPLANE

Consider the training sample $\{(\mathbf{x}_i, d_i)\}_{i=1}^{N}$

where

x_i is the input pattern for the i^{th} example and

d_i is the corresponding desired response (target output).

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

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

- For a given weight vector **w** and 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

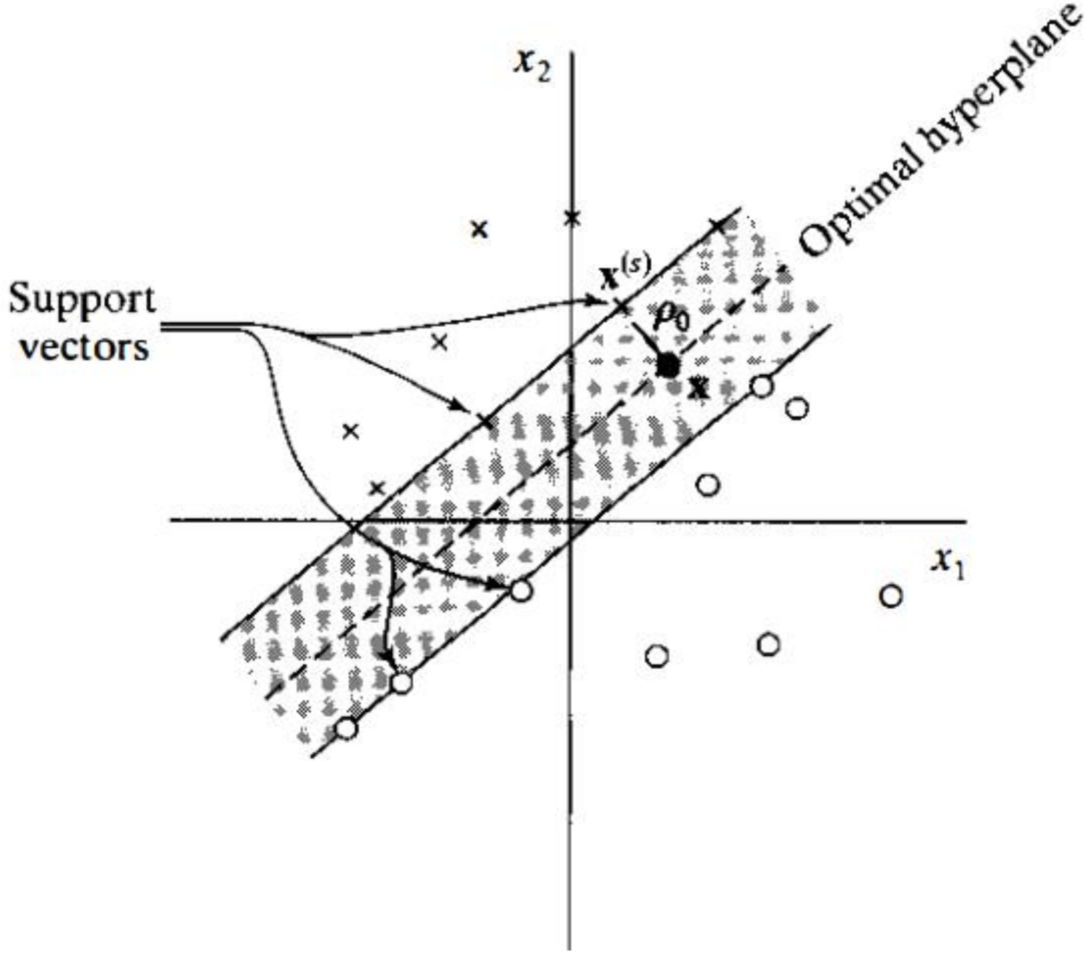


Figure illustrates the geometric construction of an optimal hyperplane for a two-dimensional input space.

- Let $\boldsymbol{W_0}$ and $\boldsymbol{b_0}$ denote the optimum values of the weight vector and bias, respectively.
- The optimal hyperplane representing a multidimensional linear 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 $\boldsymbol{x_p}$ 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

The discriminant function $, g(\mathbf{x}_p) = 0,$

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

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

In particular, the distance from the origin (i.e., x = 0) to the optimal hyperplane is given $b_o/\|\mathbf{w}_o\|.$

If $\boldsymbol{b_0} > 0$, the origin is on the positive side of the optimal hyperplane;

if $\boldsymbol{b_0} < 0$, it is on the negative side.

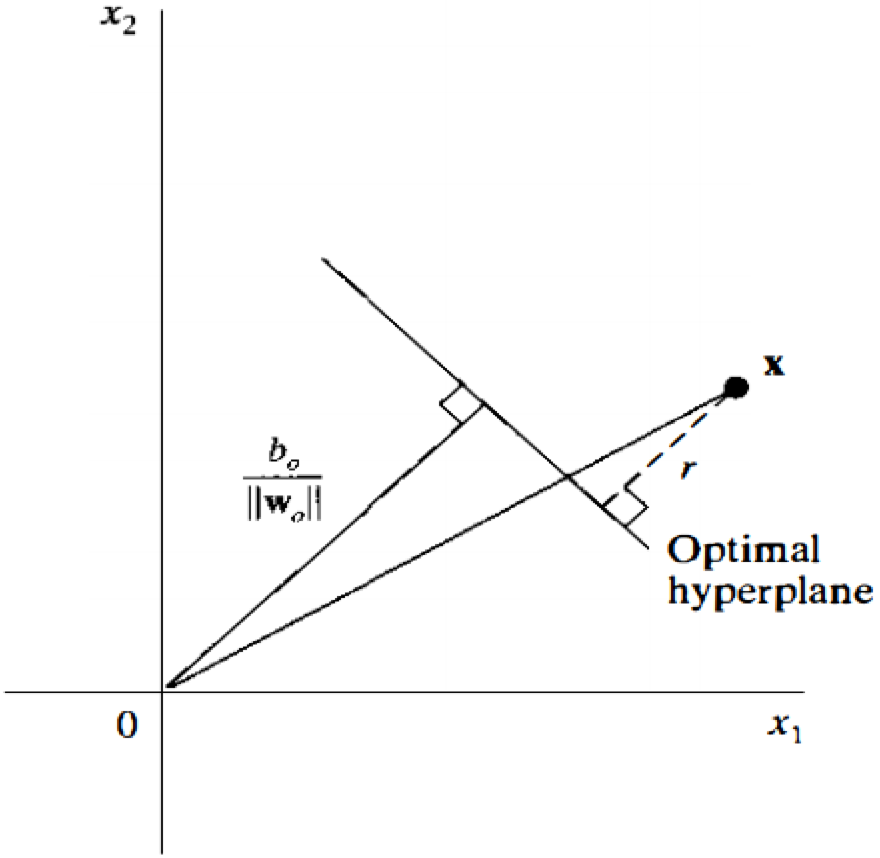
If $\boldsymbol{b_0} = 0$, the optimal hyperplane passes through the origin.

A geometric interpretation of these algebraic results is given in following figure.

The pair ($\boldsymbol{W_0}$, $\boldsymbol{b_0}$) must satisfy the constraint

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

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



Geometric interpretation of algebraic distances of points to the optimal hyperplane for a two-dimensional case

The particular data points $(\boldsymbol{x_i}, \boldsymbol{d_i})$ 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 $\boldsymbol{x}^{(s)}$ for which $\boldsymbol{d}^{(s)}$ = + 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 $\boldsymbol{x}^{(s)}$ 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 $\boldsymbol{x}^{(\boldsymbol{s})}$ lies on the positive side of the optimal hyperplane and the minus sign indicates that $\boldsymbol{x}^{(\boldsymbol{s})}$ 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 . Then, we follows that
- **ρ=2r**
- $= \frac{\mathbf{2}}{||\boldsymbol{W}_{\mathbf{0}}||}$
- 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 **w** and 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 Φ **(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 α_i are called Lagrange multipliers.
- The solution to the constrained optimization problem is determined by the saddle point of the Lagrangian function **J (w, b, α)** which has to be minimized with respect to **w** and **b.**
- **I**t also has to be maximized with respect to α.

- Thus, differentiating **J (w, b, α)** with respect to ***w*** and ***b*** and setting the results equal to zero, we get the following two conditions of optimality:

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

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

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

- 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 coefficient α_i .
- It is also important to note that at the ***saddle point***, for each Lagrange multiplier α_i ' 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 optimal value 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 $\boldsymbol{W_0}$ to be an optimal primal solution and to be an optimal dual solution, it is necessary and sufficient that $\boldsymbol{w_0}$ 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 expand 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 we may reformulate **J (w, b, α) =Q(α)**

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

- subject to the constraints
- $\sum_{i=1}^{N} \alpha_i \, d_i \quad = 0$
- α_i **≥0 for i=1,2,3. , N**
- The function **Q(α)** 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 $\boldsymbol{\alpha_i}$, denoted by we may compute the optimum weight vector so write $\boldsymbol{W_0}$

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

- To compute the optimum bias $\boldsymbol{b_0}$ we may use $\boldsymbol{W_0}$ the thus write

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

OPTIMAL HYPERPLANE FOR NONSEPARABLE PATTERNS

- we consider the more difficult case of non-separable patterns.
- For a given 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 falls inside the region of separation but on the right side of the decision surface, as illustrated in Fig. a .
- The data point 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

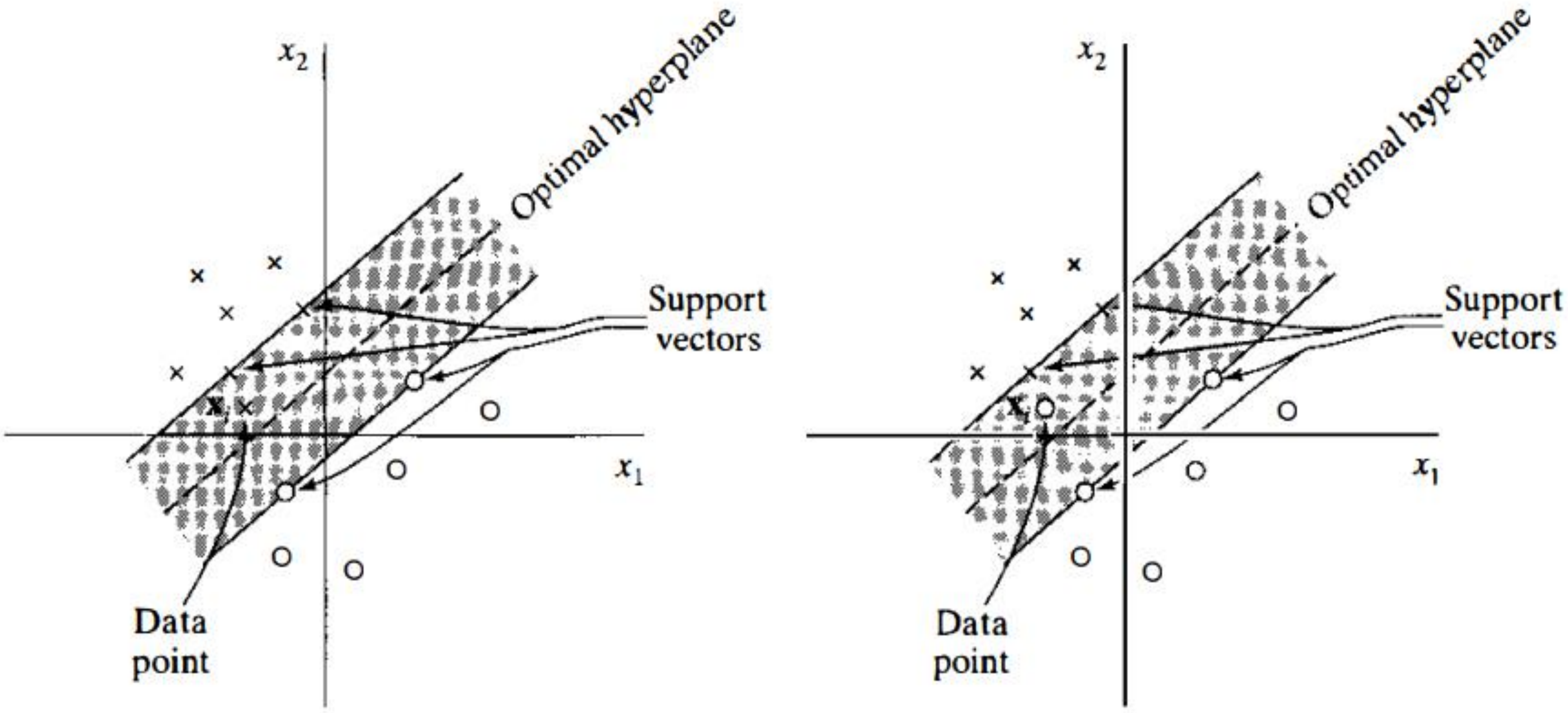


FIGURE 6.3 (a) Data point $\mathbf{x}_i$ (belonging to class $\mathscr{C}_1$) falls inside the region of separation, but on the right side of the decision surface. (b) Data point $\mathbf{x}_i$ (belonging to class $\mathscr{C}_2$) falls on the wrong side of the decision surface.

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

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

- 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.

- The support vectors are those particular data points that satisfy precisely $\xi_i > 0$ even if,
- Note that if an example with $\xi_i > 0$ is left out of the training set, the decision surface would change.
- The support vectors are thus defined in exactly the same way for both linearly separable and non-separable cases.
- Our goal is to find a separating hyperplane for which the misclassification error, averaged on the training set, is minimized. We may do this by minimizing the cost functional

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

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

- with respect to the weight vector **w**, subject to the constraint described in the above equation and the constraints $\|\breve{\mathbf{w}}\|^2$

The function $I(\xi)$ an indicator function, defined by

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

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 $\Phi(\xi)$

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

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

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

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

- The parameter C controls the tradeoff between complexity of the machine and the number of non-separable 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 $h \leq \min\left\{\left\lceil \frac{D^2}{\rho^2} \right\rceil, m_0\right\} + 1$

and then by using bounds on the generalization performance of the machine based on the VC dimension.

- 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

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

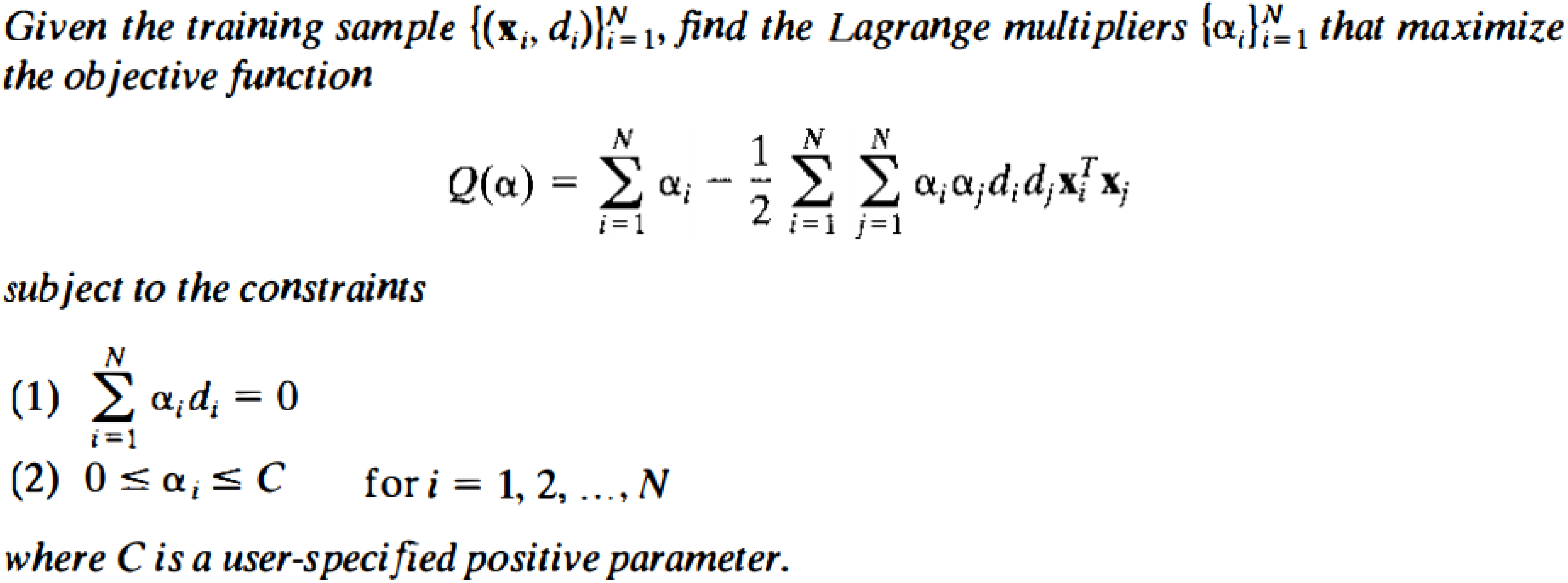
$$\xi_i \geq 0 \quad \text{for all } i$$

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.

Using the method of Lagrange multipliers and proceeding in a manner, we may formulate the dual problem for non separable 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) $0 \leq \alpha_i \leq C \qquad \text{for } i = 1, 2, \ldots, N$

where C is a user-specified positive parameter.

DESIGN OF SVM (SUPPORT VECTOR MACHINES)

- Basically, the idea of a support vector machine depends 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.

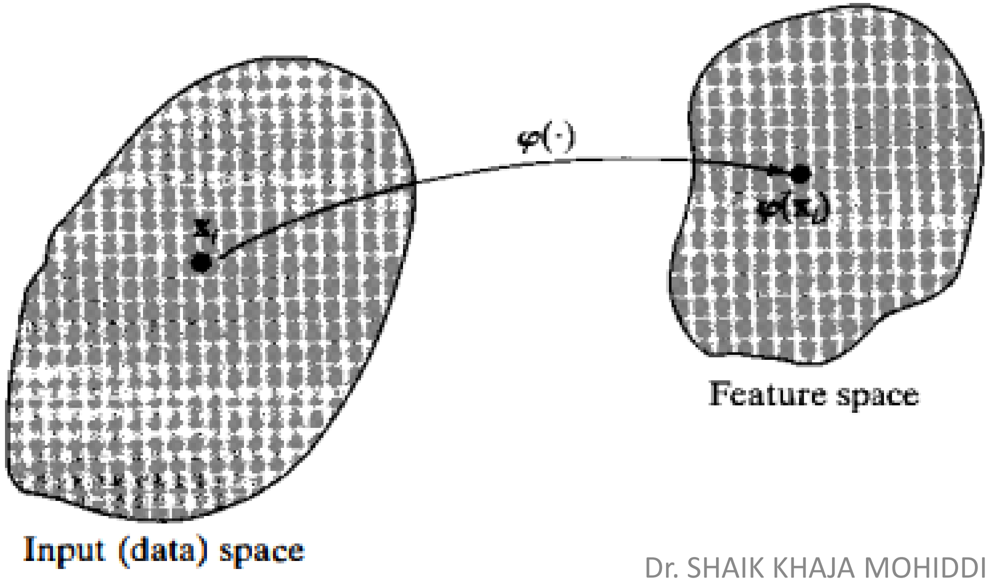


FIGURE 6.4 Nonlinear map $\varphi(\cdot)$ from the input space to the feature space.

Inner-Product Kernel

- Let $\boldsymbol{m_0}$ be the dimension of the input space.
- Let $\boldsymbol{m_1}$ be the dimension of the feature space.
- Let $\overline{\boldsymbol{x}}$ denote the input vector which is drawn from the input space
- let $\{\varphi_j(\overline{\boldsymbol{x}})\}_{j=1}^{m}$ denote a set of nonlinear transformations from the input space to the feature space.
- The equation of a hyperplane which is acting as the decision surface is given as:

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

the above equation is simplified as

$$\sum_{j=0}^{m_1} w_j\varphi_j(\mathbf{x}) = 0$$

$$\boldsymbol{w_0} + \sum_{j=1}^{m_1} w_j\varphi_j(\mathbf{x}) \cdot \quad (\text{ as } \varphi_0(\text{x})=1)$$

Where it is assumed that φ_0(x)= = 1 for all x, so that $\boldsymbol{w_0}$ denotes the bias b.

$$\sum_{j=0}^{m_1} w_j \varphi_j(\mathbf{x}) = 0$$

- defines the decision surface computed in the feature space in terms of the linear weights of the machine. The quantity φ_j represents the input supplied to the weight $\boldsymbol{w_j}$ via the feature space.
- We define the vector $\varphi\ \overline{(\boldsymbol{x}})$

$$\boldsymbol{\varphi}(\mathbf{x}) = [\varphi_0(\mathbf{x}), \varphi_1(\mathbf{x}), \ldots, \varphi_{m_1}(\mathbf{x})]^T$$

- as φ_0(x)=1

the decision surface is represented as $\mathbf{w}^T \boldsymbol{\varphi}(\mathbf{x}) = 0$

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

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

now substituting the values of W in the above equation we get

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

- The term $\boldsymbol{\varphi}^{\mathrm{T}}(\mathbf{x}_i)\varphi(\mathbf{x})$ denote the inner product of two vectors induced in the feature space by the input vector $\boldsymbol{x}$ and the input pattern $\boldsymbol{x_i}$.
- introduce the inner-product kernel denoted by K(x, Xi) and defined by

$$K(\mathbf{x}, \mathbf{x}_i) = \boldsymbol{\varphi}^T(\mathbf{x})\boldsymbol{\varphi}(\mathbf{x}_i)$$

$$= \sum_{j=0}^{m_1} \varphi_j(\mathbf{x})\varphi_j(\mathbf{x}_i) \qquad \text{for } i = 1, 2, \ldots, N$$

- From this definition we immediately see that the inner-product kernel is a symmetric function of its arguments, as shown by

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

- For an optimal hyperplane is now defined by

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

Mercer's Theorem

Inner product kernel $K(\mathbf{x}, \mathbf{x}_i)$ is an important special case of ***Mercer's theorem*** that arises in functional analysis

Let K(x, x') be a continuous symmetric kernel that is defined in the closed interval a ≤ x ≤ b and likewise for x ' . The kernel K(x, x') can be expanded in the series

$$K(\mathbf{x}, \mathbf{x}') = \sum_{i=1}^{\infty} \lambda_i \varphi_i(\mathbf{x}) \varphi_i(\mathbf{x}')$$

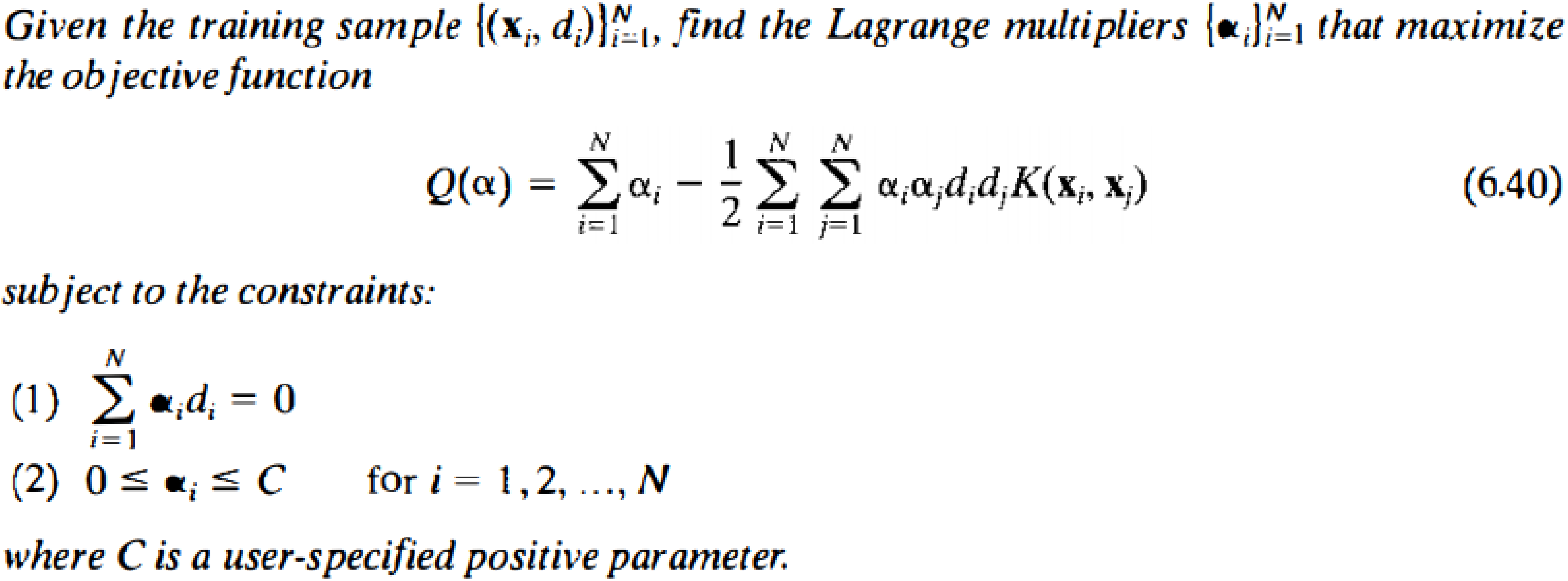
Where λ_i

In mercers theorem we make the following observations

1. Dimensionality of the feature space is infinitely large.
2. For $\lambda_i = 1,$ the i^{th} image $\sqrt{\lambda_i}\,\varphi_j$(x) induces in the feature space

OPTIMUM DESIGN OF SVM

- The expansion of the inner product kernel K($\boldsymbol{x}, \boldsymbol{x}_i$) permits us to construct a decision surface that is nonlinear in the input space, but its image in the feature space.



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*

$$Q(\alpha) = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j d_i d_j K(\mathbf{x}_i, \mathbf{x}_j) \qquad (6.40)$$

subject to the constraints:

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

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

where C is a user-specified positive parameter.

Examples of support vector machines

- The requirement on the kernel K($\boldsymbol{x}, \boldsymbol{x}_i$) is to satisfy Mercer's theorem.

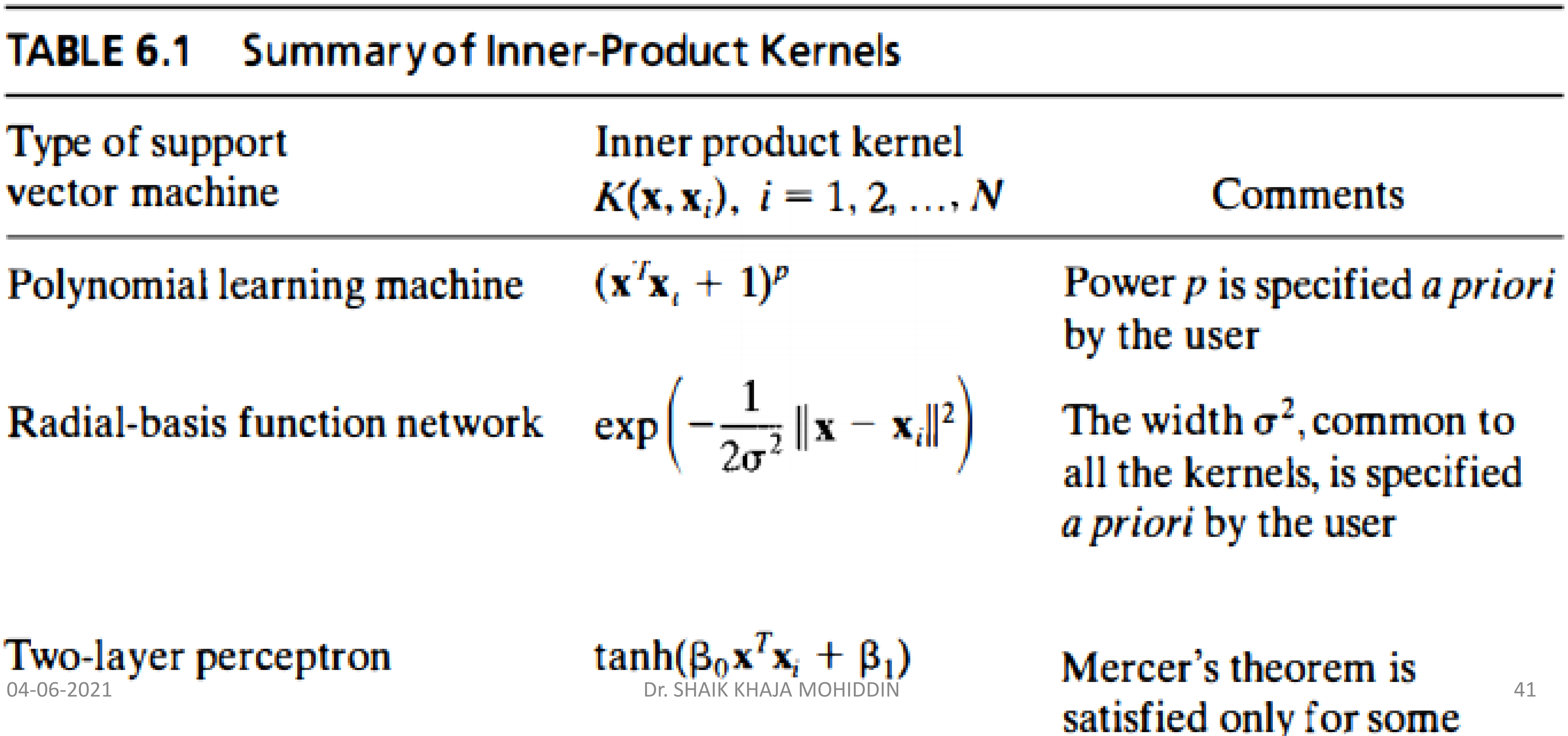


TABLE 6.1 Summary of Inner-Product Kernels

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

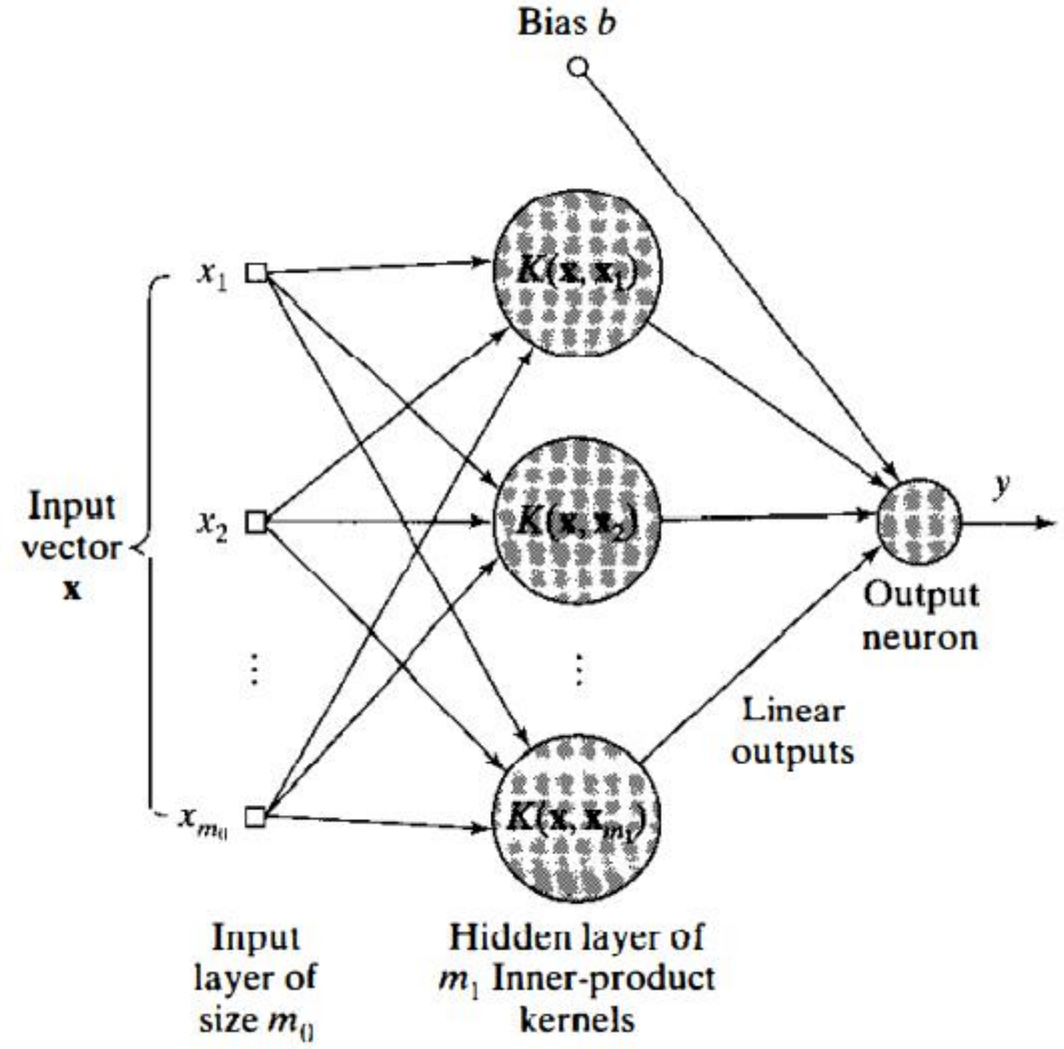
- the inner-product kernels for three common types of support vector machines: polynomial learning machine, radial-basis function network, and two-layer perceptron. The following points are noteworthy.
    - The inner-product kernels for polynomial and radial-basis function types of support vector machines always satisfy Mercer's theorem.
    - 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.
    - 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 perceptron's:

Architecture of SVM

- Irrespective of how a support vector machine is implemented , it differs from 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 (Vapnik, 1995, 1998)

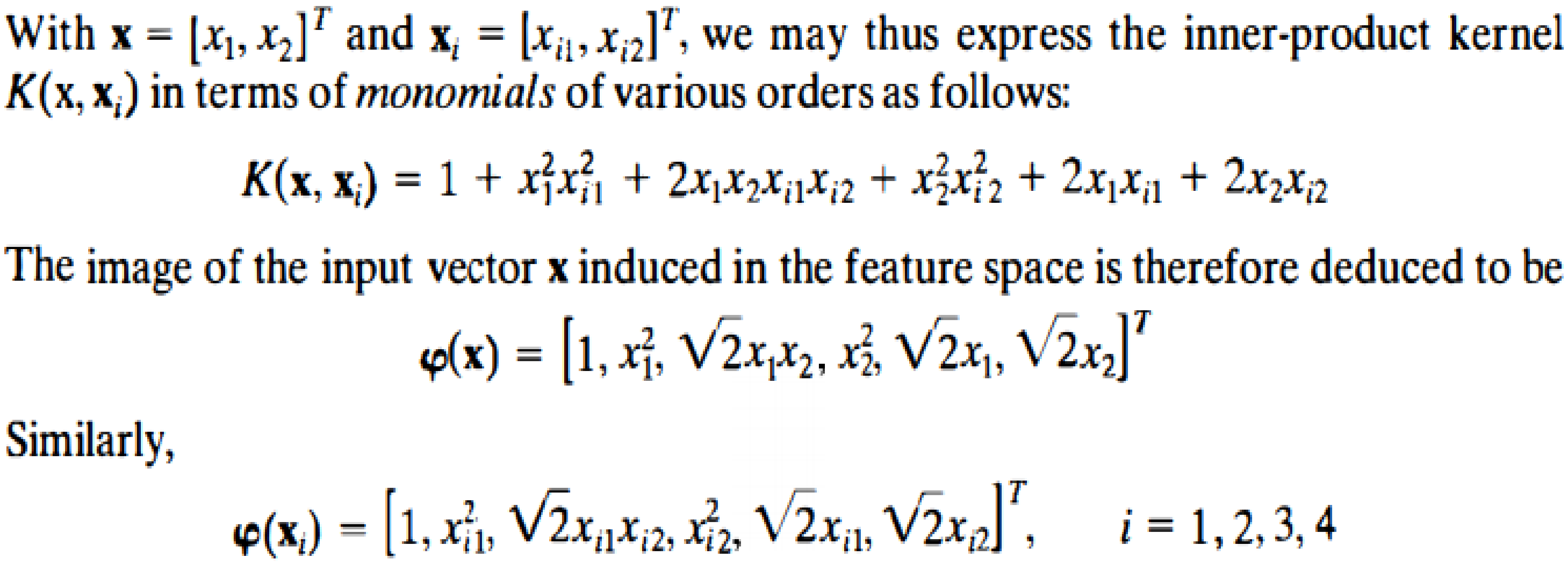

Bias b
x_1
x_2
x_{m_0}
Input vector $\mathbf{x}$
$K(\mathbf{x}, \mathbf{x}_1)$
$K(\mathbf{x}, \mathbf{x}_2)$
$K(\mathbf{x}, \mathbf{x}_{m_1})$
y
Output neuron
Linear outputs
Input layer of size m_0
Hidden layer of m_1 Inner-product kernels

- **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.

- **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 (defined in accordance with Mercer's theorem) and solving the dual form of the constrained optimization problem formulated in the input (data) space.

EXAMPLE: XOR PROBLEM (REVISITED)

Let consider an XOR problem using SVM (cherkassky and mulier ,1998) $K(\mathbf{x}, \mathbf{x}_i) = (1 + \mathbf{x}^T\mathbf{x}_i)^2$



With $\mathbf{x} = [x_1, x_2]^T$ and $\mathbf{x}_i = [x_{i1}, x_{i2}]^T$, we may thus express the inner-product kernel $K(\mathbf{x}, \mathbf{x}_i)$ in terms of *monomials* of various orders as follows:

$$K(\mathbf{x}, \mathbf{x}_i) = 1 + x_1^2 x_{i1}^2 + 2x_1x_2x_{i1}x_{i2} + x_2^2x_{i2}^2 + 2x_1x_{i1} + 2x_2x_{i2}$$

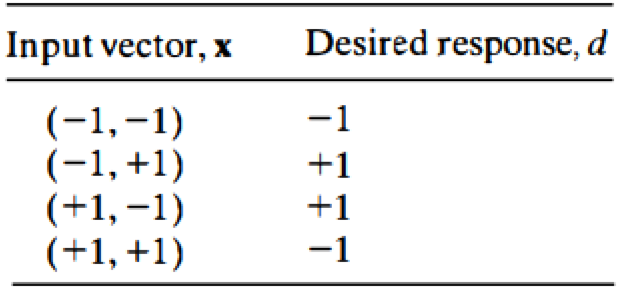
The image of the input vector $\mathbf{x}$ induced in the feature space is therefore deduced to be

$$\boldsymbol{\varphi}(\mathbf{x}) = [1, x_1^2, \sqrt{2}x_1x_2, x_2^2, \sqrt{2}x_1, \sqrt{2}x_2]^T$$

Similarly,

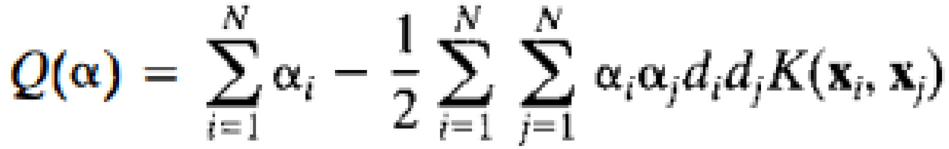
$$\boldsymbol{\varphi}(\mathbf{x}_i) = [1, x_{i1}^2, \sqrt{2}x_{i1}x_{i2}, x_{i2}^2, \sqrt{2}x_{i1}, \sqrt{2}x_{i2}]^T, \qquad i = 1, 2, 3, 4$$

XOR truth table:

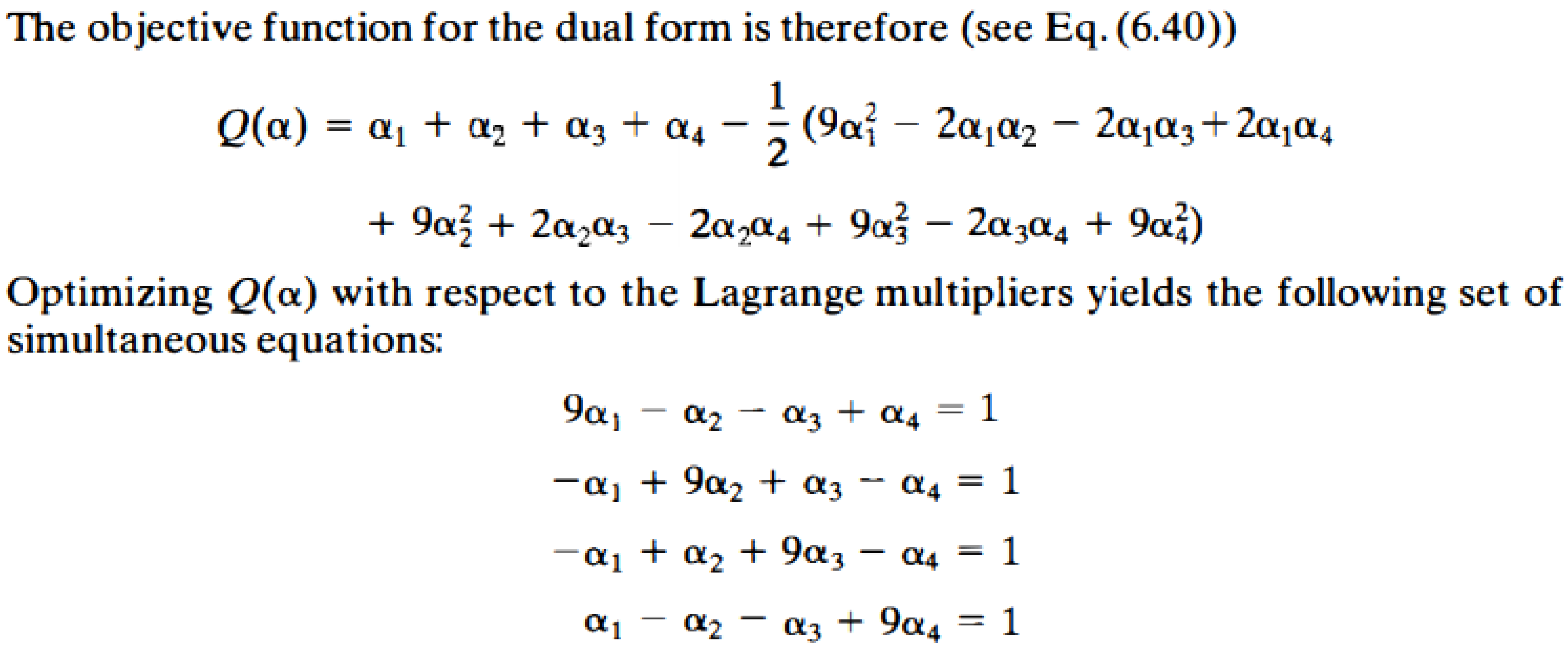


Input vector, **x**	Desired response, d
$(-1,-1)$	-1
$(-1,+1)$	$+1$
$(+1,-1)$	$+1$
$(+1,+1)$	-1

As we have from the objective function



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

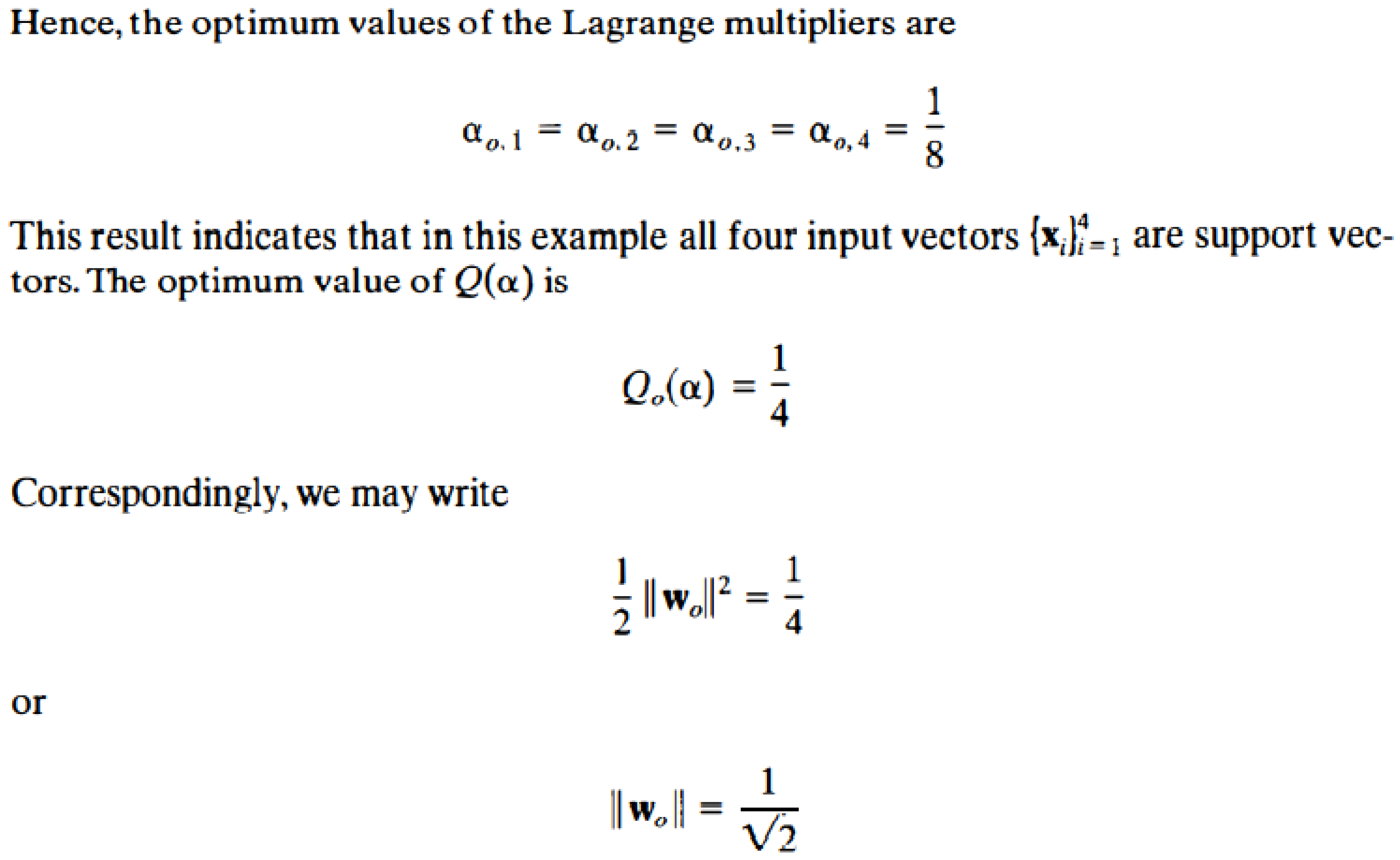


The objective function for the dual form is therefore (see Eq. (6.40))

$$Q(\alpha) = \alpha_1 + \alpha_2 + \alpha_3 + \alpha_4 - \frac{1}{2}(9\alpha_1^2 - 2\alpha_1\alpha_2 - 2\alpha_1\alpha_3 + 2\alpha_1\alpha_4$$
$$+ 9\alpha_2^2 + 2\alpha_2\alpha_3 - 2\alpha_2\alpha_4 + 9\alpha_3^2 - 2\alpha_3\alpha_4 + 9\alpha_4^2)$$

Optimizing $Q(\alpha)$ with respect to the Lagrange multipliers yields the following set of simultaneous equations:

$$9\alpha_1 - \alpha_2 - \alpha_3 + \alpha_4 = 1$$
$$-\alpha_1 + 9\alpha_2 + \alpha_3 - \alpha_4 = 1$$
$$-\alpha_1 + \alpha_2 + 9\alpha_3 - \alpha_4 = 1$$
$$\alpha_1 - \alpha_2 - \alpha_3 + 9\alpha_4 = 1$$



Hence, the optimum values of the Lagrange multipliers are

$$\alpha_{o,1} = \alpha_{o,2} = \alpha_{o,3} = \alpha_{o,4} = \frac{1}{8}$$

This result indicates that in this example all four input vectors $\{\mathbf{x}_i\}_{i=1}^{4}$ are support vectors. The optimum value of $Q(\alpha)$ is

$$Q_o(\alpha) = \frac{1}{4}$$

Correspondingly, we may write

$$\frac{1}{2}\|\mathbf{w}_o\|^2 = \frac{1}{4}$$

or

$$\|\mathbf{w}_o\| = \frac{1}{\sqrt{2}}$$

From the equation

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

$$\mathbf{w}_o = \frac{1}{8}[-\varphi(\mathbf{x}_1) + \varphi(\mathbf{x}_2) + \varphi(\mathbf{x}_3) - \varphi(\mathbf{x}_4)]$$

$$= \frac{1}{8}\left[-\begin{bmatrix} 1 \\ 1 \\ \sqrt{2} \\ 1 \\ -\sqrt{2} \\ -\sqrt{2} \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \\ -\sqrt{2} \\ 1 \\ -\sqrt{2} \\ \sqrt{2} \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \\ -\sqrt{2} \\ 1 \\ \sqrt{2} \\ -\sqrt{2} \end{bmatrix} - \begin{bmatrix} 1 \\ 1 \\ \sqrt{2} \\ 1 \\ \sqrt{2} \\ \sqrt{2} \end{bmatrix}\right]$$

$$= \begin{bmatrix} 0 \\ 0 \\ -1/\sqrt{2} \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

The first element of $\boldsymbol{W}_{\mathbf{0}}$ indicates that the bias b is zero.

The optimal hyperplane is defined by

$$\mathbf{w}^T \varphi(\mathbf{x}) = 0$$

That is,

$$\left[0, 0, \frac{-1}{\sqrt{2}}, 0, 0, 0\right]\begin{bmatrix} 1 \\ x_1^2 \\ \sqrt{2}x_1x_2 \\ x_2^2 \\ \sqrt{2}x_1 \\ \sqrt{2}x_2 \end{bmatrix} = 0$$

which reduces to

$$-x_1x_2 = 0$$

The polynomial form of support vector machine for the **XOR** problem is as shown in Fig. 6.6a. For both $x_1 = x_2 = -1$ and $x_1 = x_2 = +1$, the output $y = -1$; and for both $x_1 = -1, x_2 = +1$ and $x_1 = +1$ and $x_2 = -1$, we have $y = +1$. Thus the **XOR** problem is solved as indicated in Fig. 6.6b.

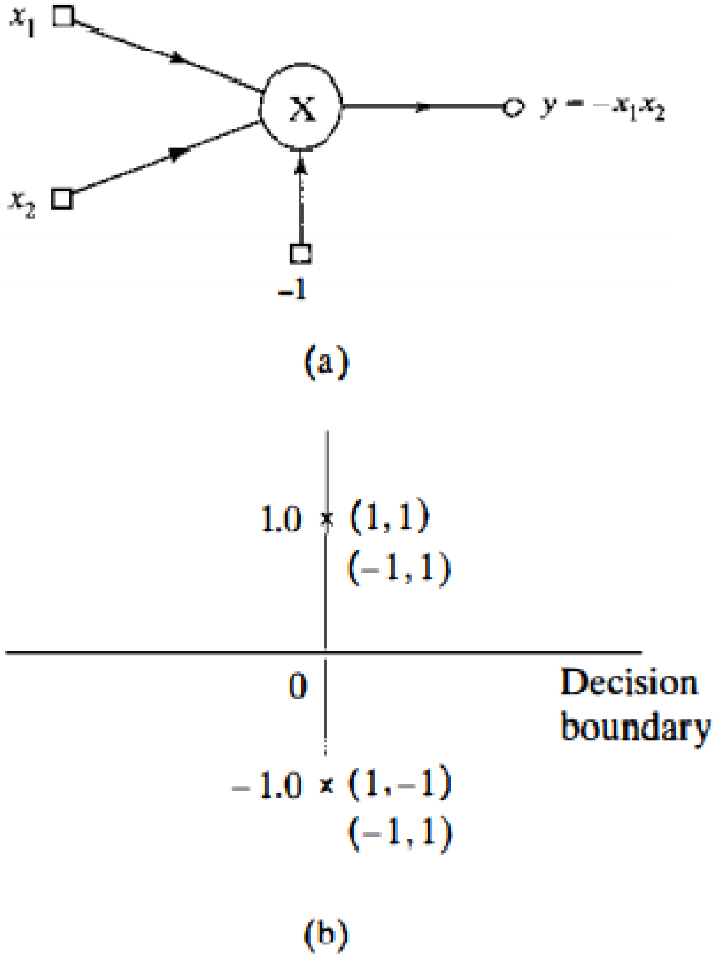


FIGURE 6.6 (a) Polynomial machine for solving the XOR problem. (b) Induced images in the feature space due to the four data points of the XOR problem.