KECE470 Pattern Recognition

Chapter 3. Linear Classifiers

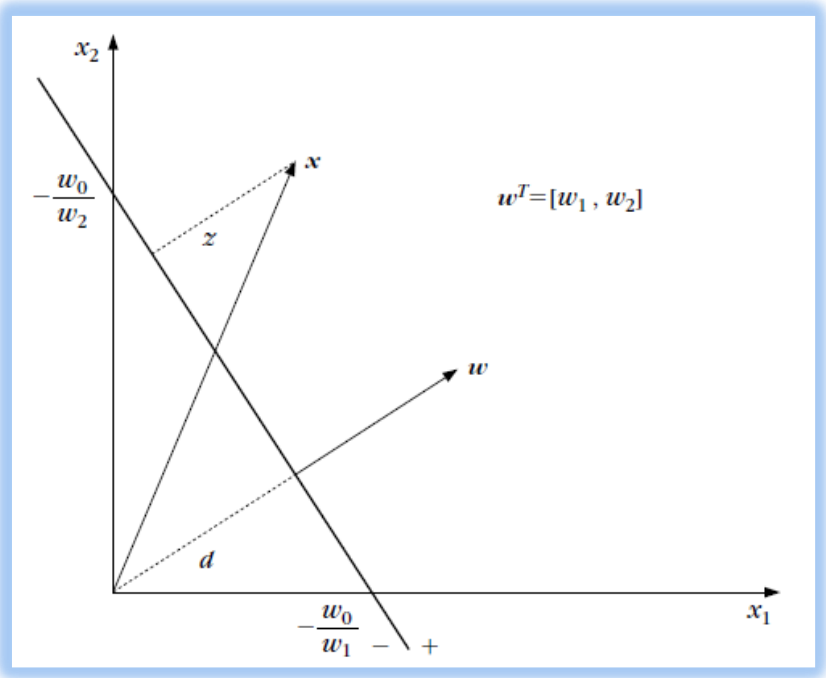
Chang-Su Kim

Many slides are modified from Serigos Theodoridis's own notes.

Linear Classifiers

- Linear classifiers are simple and computationally attractive
- Linear discriminant functions → linear decision surfaces (decision hyperplanes)

$$g(\mathbf{x}) = \mathbf{w}^T\mathbf{x} + w_0 = 0$$
$$\Leftrightarrow \quad \mathbf{w}^T(\mathbf{x} - \mathbf{x}_0) = 0$$



PERCEPTRON ALGORITHM

Linearly Separable Case

- There exists a hyperplane, $\mathbf{w}^{*T}\mathbf{x} = 0$, such that

$$\mathbf{w}^{*T}\mathbf{x} > 0 \;\; \forall \mathbf{x} \in \omega_1$$

$$\mathbf{w}^{*T}\mathbf{x} < 0 \;\; \forall \mathbf{x} \in \omega_2$$

  - This formulation also covers the case of a hyperplane not crossing the origin, *i.e.*, $\mathbf{w}^{*T}\mathbf{x} + w_0^*$=0, by defining the **extended** $(l+1)$-dimensional vectors

$$\mathbf{x}' \equiv [\mathbf{x}^T, 1]^T \text{ and } \mathbf{w}' \equiv [\mathbf{w}^{*T}, w_0^*]^T.$$

  - Then $\mathbf{w}^{*T}\mathbf{x} + w_0^* = \mathbf{w}'^T\mathbf{x}'$

Problem Formulation: Perceptron Cost

$$J(\mathbf{w}) = \sum_{\mathbf{x} \in Y} (\delta_{\mathbf{x}} \mathbf{w}^T \mathbf{x})$$

- Y is the set of vectors, misclassified by the hyperplane $\mathbf{w}$
- The variable $\delta_{\mathbf{x}}$

$$\delta_{\mathbf{x}} = \begin{cases} -1 & \text{if } x \in \omega_1 \\ +1 & \text{if } x \in \omega_2 \end{cases}$$

- For separating $\mathbf{w}$, $J(\mathbf{w}) = 0$ because $Y = \emptyset$
- $J(\mathbf{w})$ is continuous and piecewise linear

Optimization: Perceptron Algorithm

- Inspired by gradient descent

$$\mathbf{w}(t+1) = \mathbf{w}(t) - \rho_t \left.\frac{\partial J(\mathbf{w})}{\partial \mathbf{w}}\right|_{\mathbf{w}=\mathbf{w}(t)}$$

- Note that $\frac{\partial J(\mathbf{w})}{\partial \mathbf{w}} = \sum_{\mathbf{x}\in Y} \delta_{\mathbf{x}}\mathbf{x}$. Thus, we have

$$\mathbf{w}(t+1) = \mathbf{w}(t) - \rho_t \sum_{\mathbf{x}\in Y} \delta_{\mathbf{x}}\mathbf{x}$$

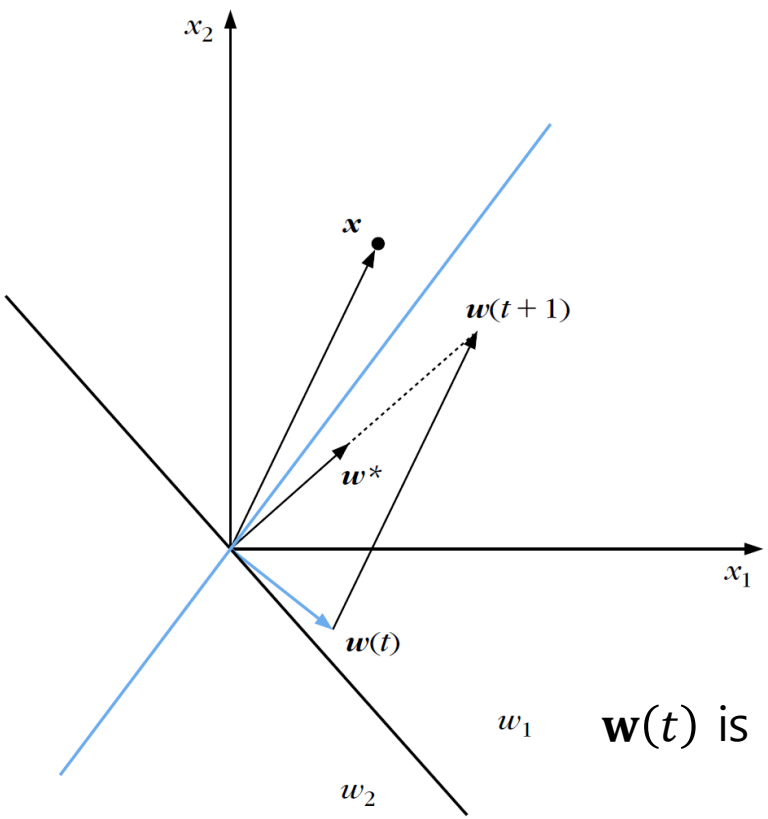
Perceptron Algorithm

- Choose $\mathbf{w}(0)$ randomly
- Choose ρ_0
- $t = 0$
- Repeat
  - $Y = \emptyset$
  - For $i = 1$ to N

    If $\delta_{\mathbf{x}_i}\mathbf{w}(t)^T\mathbf{x}_i \geq 0$ then $Y = Y \cup \{\mathbf{x}_i\}$
  - End For
  - $\mathbf{w}(t+1) = \mathbf{w}(t) - \rho_t \frac{\partial J(\mathbf{w})}{\partial \mathbf{w}}\Big|_{\mathbf{w}=\mathbf{w}(t)}$
  - Adjust ρ_t
  - $t = t + 1$
- Until $Y = \emptyset$

Perceptron Algorithm

- Remark
  - It converges to a solution in a finite number of steps, provided that $\rho_t \propto \frac{1}{t}$ (proof skipped)



$\mathbf{w}(t)$ is updated to $\mathbf{w}(t+1)$ to include $\mathbf{x}$

Perceptron Algorithm

Example 3.1

Figure 3.3 shows the dashed line

$$x_1 + x_2 - 0.5 = 0$$

corresponding to the weight vector $[1, 1, -0.5]^T$, which has been computed from the latest iteration step of the perceptron algorithm (3.9), with $\rho_t = \rho = 0.7$. The line classifies correctly

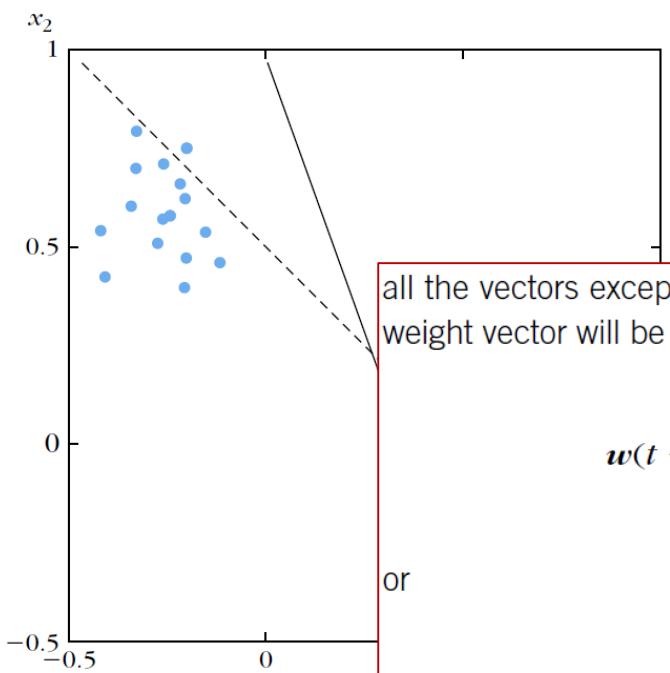


FIGURE 3.3

An example of the perceptron algorithm. After the
turned from its initial location (dotted line) to the r
classified.

all the vectors except $[0.4, 0.05]^T$ and $[-0.20, 0.75]^T$. According to the algorithm, the next weight vector will be

$$\boldsymbol{w}(t+1) = \begin{bmatrix} 1 \\ 1 \\ -0.5 \end{bmatrix} - 0.7(-1) \begin{bmatrix} 0.4 \\ 0.05 \\ 1 \end{bmatrix} - 0.7(+1) \begin{bmatrix} -0.2 \\ 0.75 \\ 1 \end{bmatrix}$$

or

$$\boldsymbol{w}(t+1) = \begin{bmatrix} 1.42 \\ 0.51 \\ -0.5 \end{bmatrix}$$

The resulting new (solid) line $1.42x_1 + 0.51x_2 - 0.5 = 0$ classifies all vectors correctly, and the algorithm is terminated.

Terminology

If $\mathbf{w}^T\mathbf{x} + w_0 > 0$ assign $\mathbf{x}$ to ω_1

If $\mathbf{w}^T\mathbf{x} + w_0 < 0$ assign $\mathbf{x}$ to ω_2

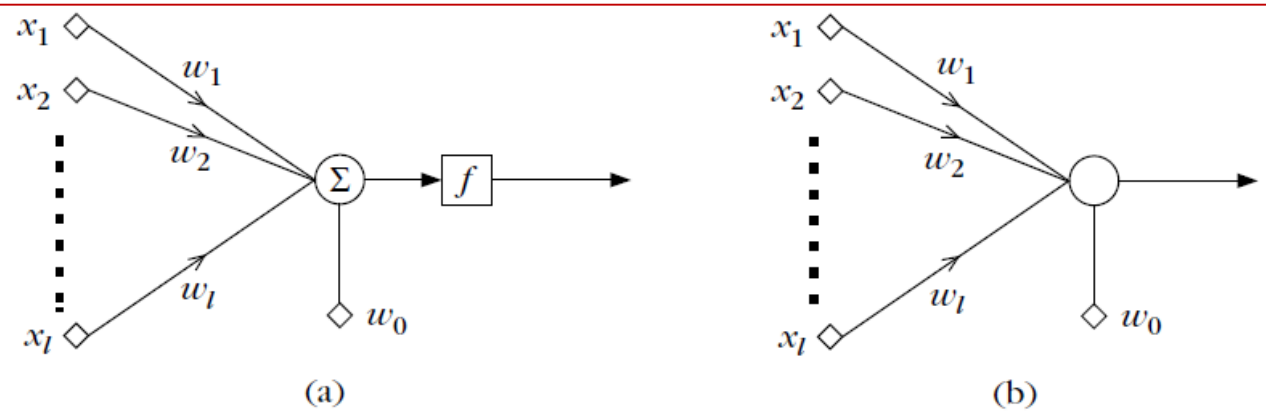


FIGURE 3.5

The basic perceptron model. (a) A linear combiner is followed by the activation function. (b) The combiner and the activation function are merged together.

- **Perceptron** or **neuron**
- **Synaptic weights** or **synapses**
- **Activation function**: *e.g.* $f(x) = 2\delta(x) - 1$

Variants

- Reward and punishment schemes
  - Training vectors enter the algorithm cyclically

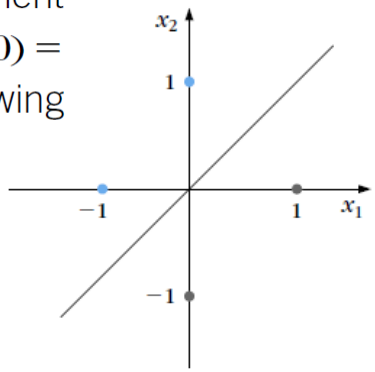
$$\mathbf{w}(t+1) = \mathbf{w}(t) + \rho\mathbf{x}_{(t)} \quad \text{if } \mathbf{x}_{(t)} \in \omega_1 \text{ and } \mathbf{w}^T(t)\mathbf{x}_{(t)} \leq 0$$

$$\mathbf{w}(t+1) = \mathbf{w}(t) - \rho\mathbf{x}_{(t)} \quad \text{if } \mathbf{x}_{(t)} \in \omega_2 \text{ and } \mathbf{w}^T(t)\mathbf{x}_{(t)} \geq 0$$

$$\mathbf{w}(t+1) = \mathbf{w}(t) \quad \text{otherwise}$$

Example 3.2

Figure 3.4 shows four points in the two-dimensional space. Points $(-1,0),(0,1)$ belong to class ω_1, and points $(0,-1),(1,0)$ belong to class ω_2. The goal of this example is to design a linear classifier using the perceptron algorithm in its reward and punishment form. The parameter ρ is set equal to one, and the initial weight vector is chosen as $\boldsymbol{w}(0) = [0,0,0]^T$ in the extended three-dimensional space. According to (3.21)–(3.23), the following computations are in order:



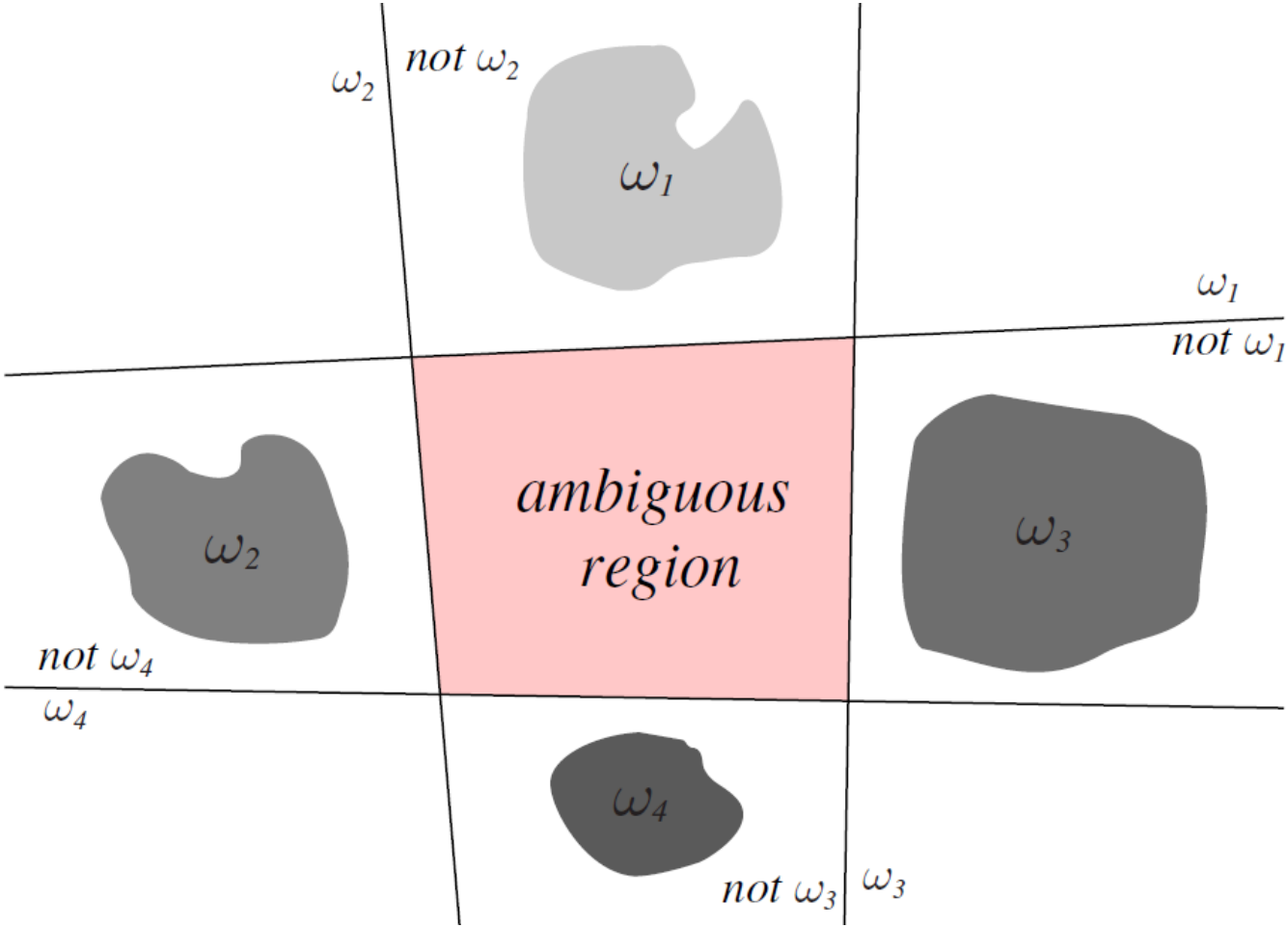
Variants

- Pocket Algorithm
  - Converges to an optimal solution, even if not linearly separable

- Initialize the weight vector $\mathbf{w}(0)$ randomly. Define a stored (in the pocket!) vector $\mathbf{w}_s$. Set a history counter h_s of $\mathbf{w}_s$ to zero.
- At the t-th iteration step, update $\mathbf{w}(t+1)$ according to the perceptron rule. Use $\mathbf{w}(t+1)$ to test the number h of training vectors correctly classified. If $h > h_s$ replace $\mathbf{w}_s$ with $\mathbf{w}(t+1)$ and h_s with h. Continue the iterations.

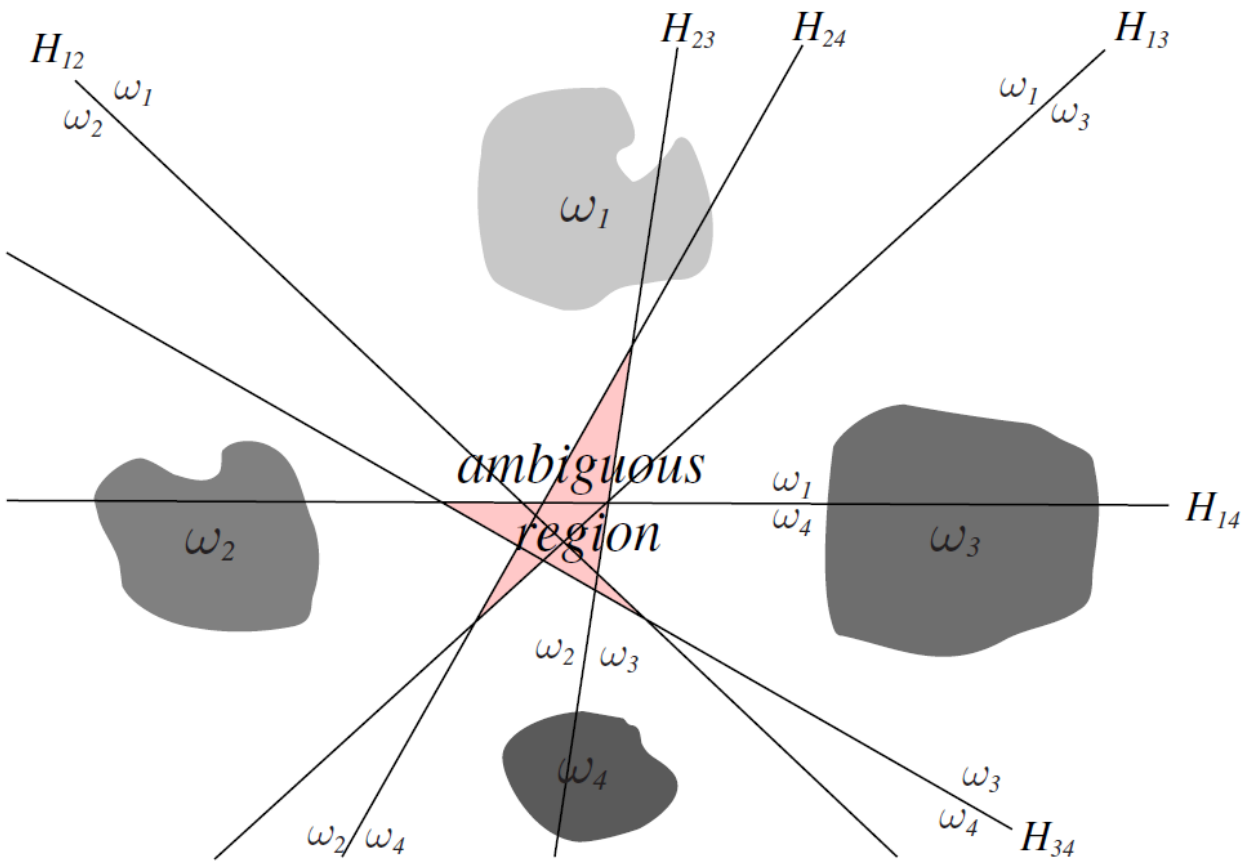
M-class Case

- Naïve approach I



M-class Case

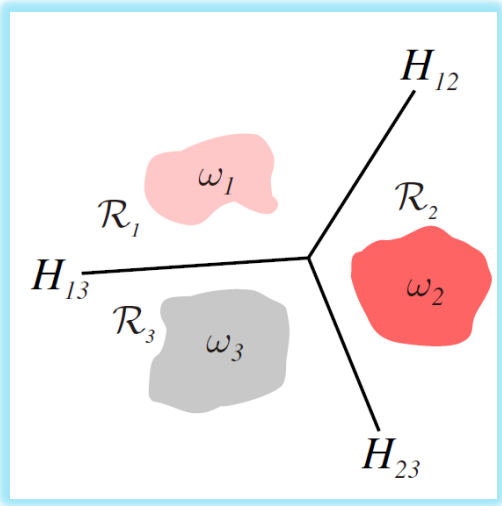
- Naïve approach II

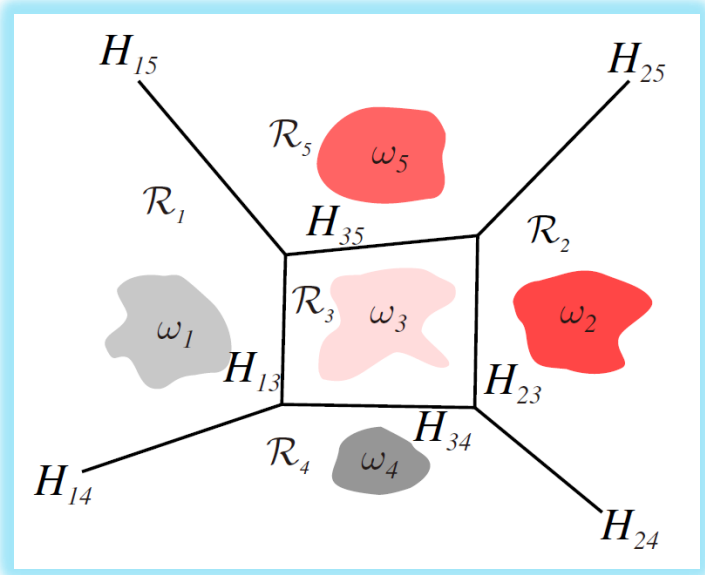


M-class Case

- Linear Machine
  - Define M linear discriminant functions $g_i(\mathbf{x})$
  - Assign $\mathbf{x}$ to ω_i if $g_i(\mathbf{x}) \geq g_j(\mathbf{x})$ for all j
- If R_i and R_j are contiguous, the boundary H_{ij} is given by the hyperplane

$$g_i(\mathbf{x}) = g_j(\mathbf{x})$$





Kesler's Construction

- Generalization to M-class task
  - Define a linear discriminant function $\mathbf{w}_i$, $i = 1,2, \dots, M$, for each class. Classify a feature vector $\mathbf{x}$ into class ω_i if

$$\mathbf{w}_i^T \mathbf{x} > \mathbf{w}_j^T \mathbf{x}, \forall j \neq i$$

  - For each training vector from class ω_i, construct $M - 1$ vectors $\mathbf{x}_{ij} = [0^T, 0^T, \dots, \mathbf{x}^T, \dots, -\mathbf{x}^T, \dots, 0^T]^T$. It is a block vector, having zeros everywhere except at the ith and jth block positions, where it has $\mathbf{x}$ and $-\mathbf{x}$, respectively.
  - Also construct the block vector $\mathbf{w} = [\mathbf{w}_1^T, \dots, \mathbf{w}_M^T]^T$.
  - If $\mathbf{x} \in \omega_i$, this imposes the requirement that $\mathbf{w}^T \mathbf{x}_{ij} > 0, \forall j \neq i$.
  - The task now is to design a linear classifier, in the extended space, so that each extended training vector lies in its positive side.

MEAN SQUARED ERROR (OR LEAST SQUARES) ALGORITHMS

MSE Estimation

- Given a vector $\mathbf{x}$, the classifier yields the output $\mathbf{w}^T\mathbf{x}$
  - A threshold can be accommodated by the vector extension
- The desired output is $y(\mathbf{x}) = \pm 1$

MSE Estimation

- Problem:

$$\text{Minimize } J(\mathbf{w}) = E[(y - \mathbf{w}^T\mathbf{x})^2]$$

- Solution

$$\widehat{\mathbf{w}} = R_{\mathbf{x}}^{-1} \times E[\mathbf{x}y]$$

Here, the autocorrelation matrix and the cross-correlation vector are

$$R_{\mathbf{x}} = E[\mathbf{x}\mathbf{x}^{\boldsymbol{T}}] = \begin{bmatrix} E[x_1x_1] & \cdots & E[x_1x_l] \\ E[x_2x_1] & \cdots & E[x_2x_l] \\ \vdots & \vdots & \vdots \\ E[x_lx_1] & \dots & E[x_lx_l] \end{bmatrix} \text{ and } E[\mathbf{x}y] = \begin{bmatrix} E[x_1y] \\ \vdots \\ E[x_ly] \end{bmatrix}$$

MSE Estimation

- Multiclass generalization
  - Design M linear discriminant functions $g_i(\mathbf{x}) = \mathbf{w}_i^T\mathbf{x}$ according to the MSE criterion
    - The output responses are chosen so that $y_i = 1$ if $\mathbf{x} \in \omega_i$ and $y_i = 0$ otherwise.
    - If $M = 2$, it provides the decision hyperplane $\mathbf{w}^T\mathbf{x} \equiv (\mathbf{w}_1 - \mathbf{w}_2)^T\mathbf{x}$, which tries to yield ± 1 according to the class of $\mathbf{x}$
  - Classify an input vector $\mathbf{x}$ into ω_i that yields the highest response $g_i(\mathbf{x})$

MSE Estimation

- Multiclass generalization
  - The MSE criterion
    - Let $\mathbf{y}^T = [y_1, \dots, y_M]$ and $W = [\mathbf{w}_1, \dots, \mathbf{w}_M]$
    - Then, we have

$$\widehat{W} = \arg\min_{W} E[\|\mathbf{y} - W^T\mathbf{x}\|^2]$$

$$= \arg\min_{W} E\left[\sum_{i=1}^{M}\left(y_i - \mathbf{w}_i^T\mathbf{x}\right)^2\right]$$

Method of Least Squares

- Given training samples $(\mathbf{x}_i, y_i)$, minimize the least squares criterion

$$J(\mathbf{w}) = \sum_{i=1}^{N} \left(y_i - \mathbf{x}_i^T \mathbf{w}\right)^2$$

- Solution

$$\hat{\mathbf{w}} = (X^T X)^{-1} X^T \mathbf{y}$$

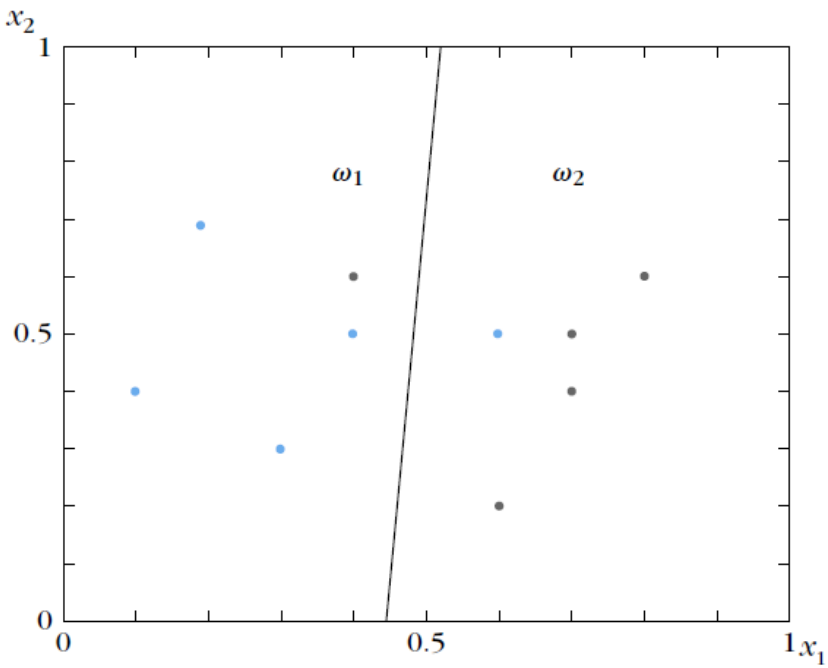
where

$$X = \begin{bmatrix} \mathbf{x}_1^T \\ \mathbf{x}_2^T \\ \vdots \\ \mathbf{x}_N^T \end{bmatrix} = \begin{bmatrix} x_{11} & x_{11} & \cdots & x_{1l} \\ x_{21} & x_{22} & \cdots & x_{2l} \\ \vdots & \vdots & \ddots & \vdots \\ x_{N1} & x_{N2} & \dots & x_{Nl} \end{bmatrix} \text{ and } \mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix}$$

Method of Least Squares

Example 3.4

Class ω_1 consists of the two-dimensional vectors $[0.2, 0.7]^T$, $[0.3, 0.3]^T$, $[0.4, 0.5]^T$, $[0.6, 0.5]^T$, $[0.1, 0.4]^T$ and class ω_2 of $[0.4, 0.6]^T$, $[0.6, 0.2]^T$, $[0.7, 0.4]^T$, $[0.8, 0.6]^T$, $[0.7, 0.5]^T$. Design the sum of error squares optimal linear classifier $w_1x_1 + w_2x_2 + w_0 = 0$.



MSE Regression (nonlinear)

- Regression task
  - A problem of designing a function $g(\mathbf{x})$, based on a set of training data points $(y_i, \mathbf{x}_i)$, so that the predicted value

$$\widehat{y_i} = g(\mathbf{x}_i)$$

    is as close to the true value y_i as possible
  - y_i need not be ± 1

- MSE regression

$$\hat{y} = \arg\min_{\tilde{y}} \overbrace{\boxed{E[\|y - \tilde{y}\|^2]}}^{E[\|y-\tilde{y}\|^2|\mathbf{x}]} = E[y|\mathbf{x}]$$

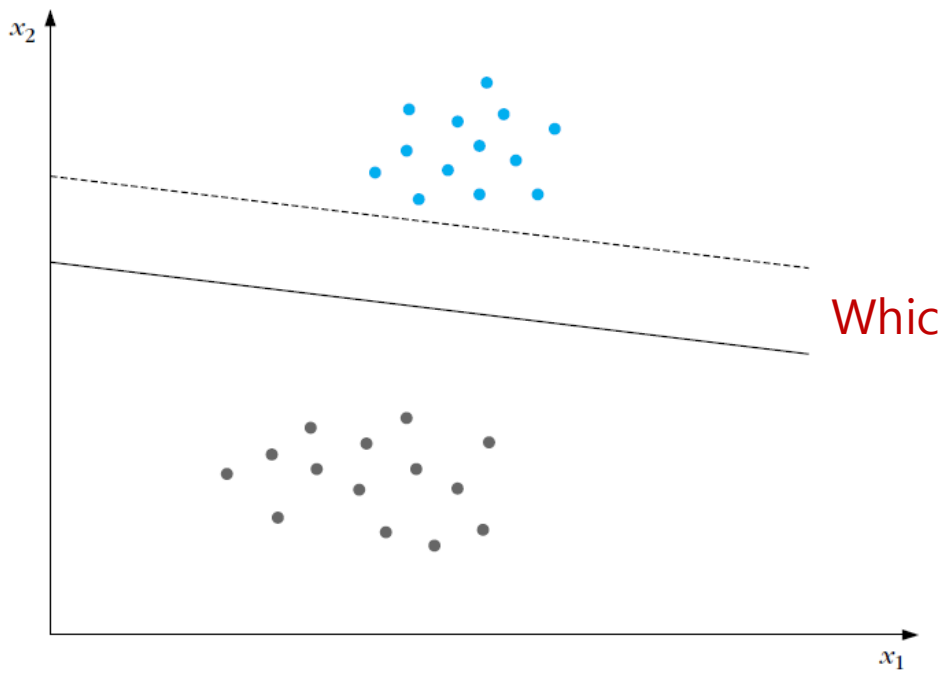
  - The MSE regression function is linear, when $(y, \mathbf{x})$ are jointly Gaussian

SUPPORT VECTOR MACHINES

Separable Classes

- The goal is again to design a separating hyperplane

$$g(\mathbf{x}) = \mathbf{w}^T\mathbf{x} + w_0 = 0$$



Which line is better?

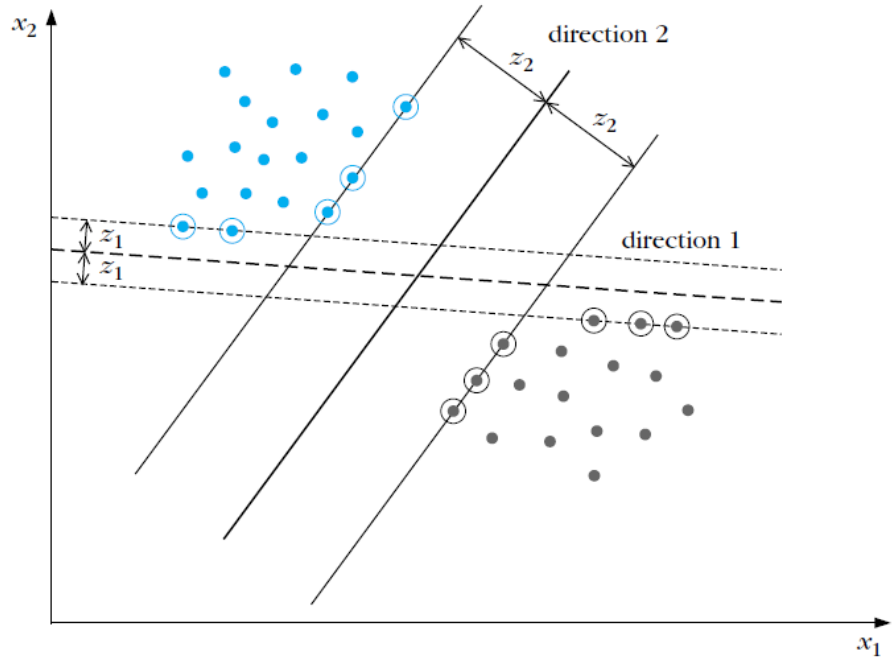
Separable Classes

- Generalization performance of a classifier: capability to operate accurately on non-training data

- In this figure
  - Margin: $2z_1$ or $2z_2$, which is given by

    $$\frac{1}{\|\mathbf{w}\|} + \frac{1}{\|\mathbf{w}\|} = \frac{2}{\|\mathbf{w}\|}$$
  - while

    $$\mathbf{w}^T\mathbf{x} + w_0 \geq 1, \quad \forall \mathbf{x} \in \omega_1$$
    $$\mathbf{w}^T\mathbf{x} + w_0 \leq -1, \forall \mathbf{x} \in \omega_2$$



Separable Classes

- The SVM problem can be formulated as computing the parameters $\mathbf{w}$, w_0 so that

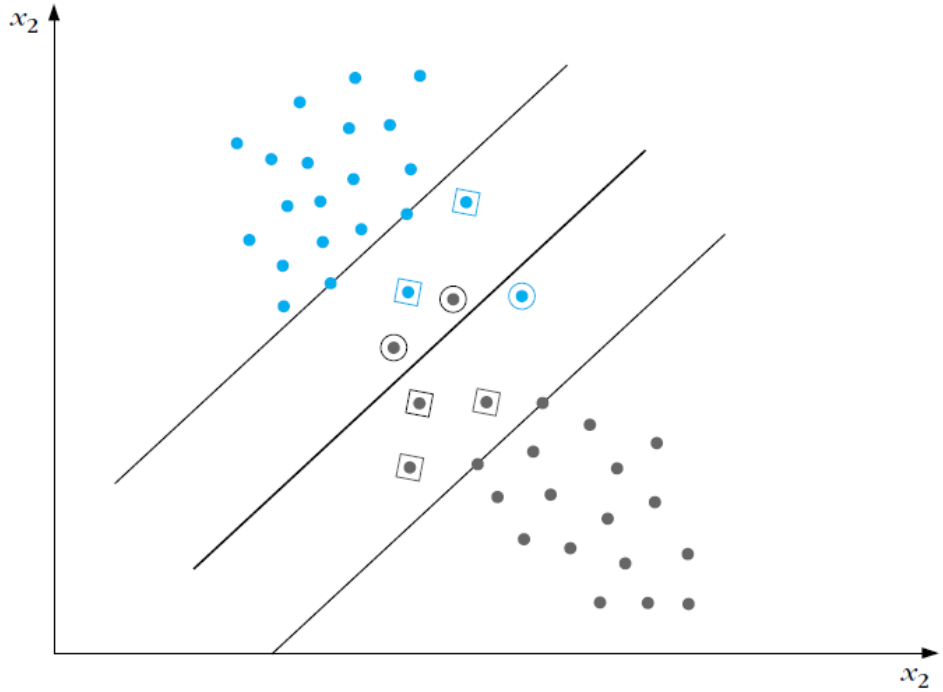
$$\text{minimize } J(\mathbf{w}, w_0) \equiv \frac{1}{2}\|\mathbf{w}\|^2$$
$$\text{subject to } y_i(\mathbf{w}^T\mathbf{x}_i + w_0) \geq 1, \quad i = 1,2,\ldots,N$$
$$\text{where } y_i = 1 \text{ if } \mathbf{x}_i \in \omega_1 \text{ and } -1 \text{ if } \mathbf{x}_i \in \omega_2$$

- This is a standard optimization problem and can be solved using, for example, Matlab
- The optimization techniques are beyond the scope of this lecture

Non-separable Classes

- The margin is defined as the distance between the parallel hyperplanes $\mathbf{w}^T\mathbf{x} + w_0 = \pm 1$

- The training vectors belong to one of the following three categories:
  1) Vectors that fall outside the band and are correctly classified
     $y_i(\mathbf{w}^T\mathbf{x} + w_0) > 1$
  2) Vectors falling inside the band and are correctly classified
     $0 \leq y_i(\mathbf{w}^T\mathbf{x} + w_0) < 1$
  3) Vectors that are misclassified
     $y_i(\mathbf{w}^T\mathbf{x} + w_0) < 0$



Non-separable Classes

- Constraints

$$y_i[\mathbf{w}^T\mathbf{x} + w_0] \geq 1 - \xi_i$$

with slack variable ξ_i

1) $\xi_i = 0$
2) $0 < \xi_i \leq 1$
3) $\xi_i > 1$

- Cost function

$$J(\mathbf{w}, w_0, \xi) \equiv \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{N} I(\xi_i)$$

where

$$I(\xi_i) = \begin{cases} 1 & \xi_i > 0 \\ 0 & \xi_i = 0 \end{cases}$$

- Because $I(\xi_i)$ is not differentiable, this is not easy to optimize

Non-separable Classes

- Problem formulation

Minimize $J(\mathbf{w}, w_0, \xi) \equiv \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{N}\xi_i$

Subject to $y_i[\mathbf{w}^T\mathbf{x}_i + w_0] \geq 1 - \xi_i,\ i = 1,2,\ldots,N$

$\xi_i \geq 0,\ i = 1,2,\ldots,N$

- This is a typical optimization problem, which can be easily solved

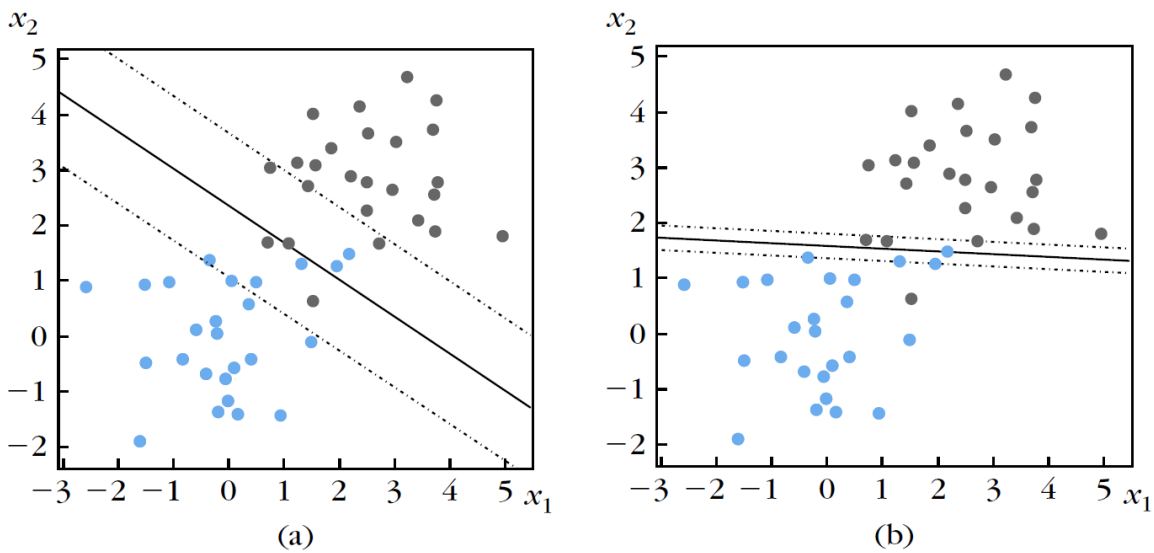


FIGURE 3.13

An example of two nonseparable classes and the resulting SVM linear classifier (full line) with the associated margin (dotted lines) for the values (a) $C = 0.2$ and (b) $C = 1000$. In the latter