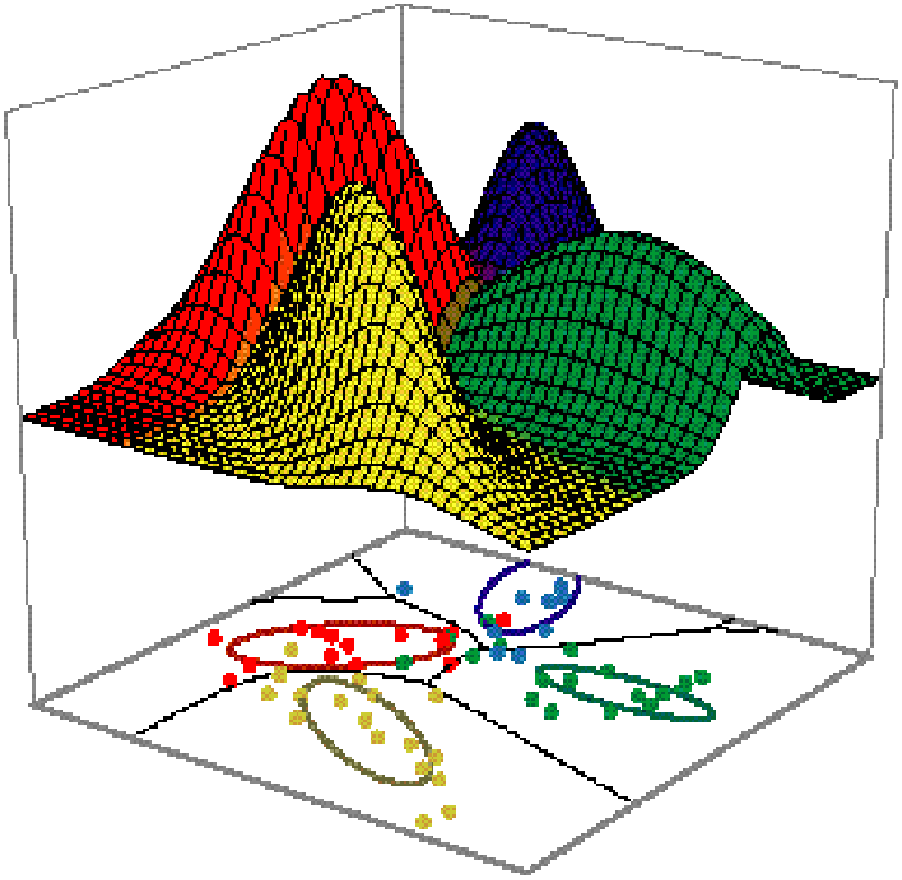
Linear Classification

Idea



Linear discriminant function

- Definition

  It is a function that is a linear combination of the components of x

  $$g(x) = w^t x + w_0 \qquad (1)$$

  where w is the weight vector and w_0 the bias

- A two-category classifier with a discriminant function of the form (1) uses the following rule:

  Decide ω_1 *if* $g(x) > 0$ and ω_2 if $g(x) < 0$

  $\Leftrightarrow$ Decide ω_1 *if* $w^t x > -w_0$ and ω_2 otherwise

  If $g(x) = 0 \Rightarrow x$ is assigned to either class

Decisions surface

- The equation $g(x) = 0$ defines the decision surface that separates points assigned to the category ω_1 from points assigned to the category $\omega_{2.}$
- When $g(x)$ is linear, the decision surface is a hyperplane.
- Algebraic measure of the distance from x to the hyperplane (interesting result!)

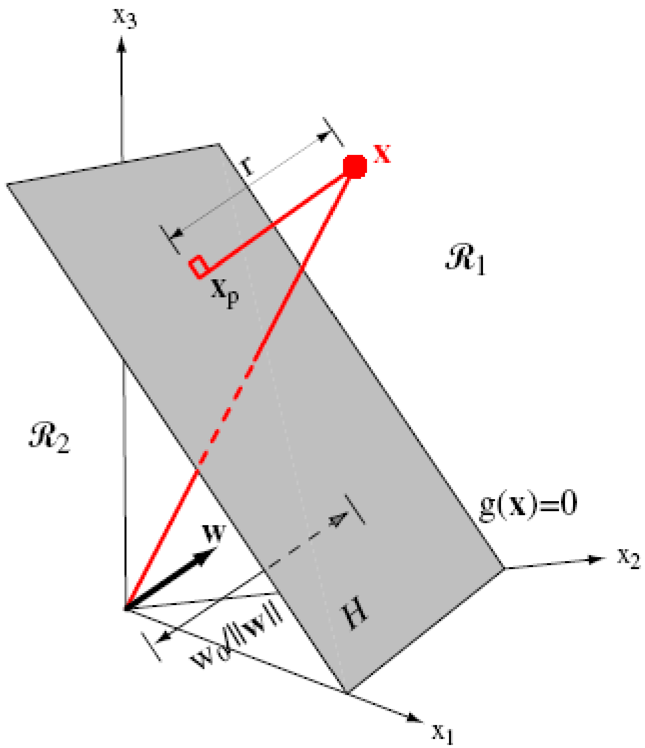


Figure 5.2: The linear decision boundary H, where $g(\mathbf{x}) = \mathbf{w}^t\mathbf{x} + w_0 = 0$, separates the feature space into two half-spaces $\mathcal{R}_1$ (where $g(\mathbf{x}) > 0$) and $\mathcal{R}_2$ (where $g(\mathbf{x}) < 0$).

Geometric analysis

$$x = x_p + r\frac{w}{\|w\|} \text{(since } w \text{ is colinear with } x - x_p \text{ and } \frac{w}{\|w\|} = 1)$$

$$\text{since } g(x_p) = 0, w^t w = \|w\|^2,$$

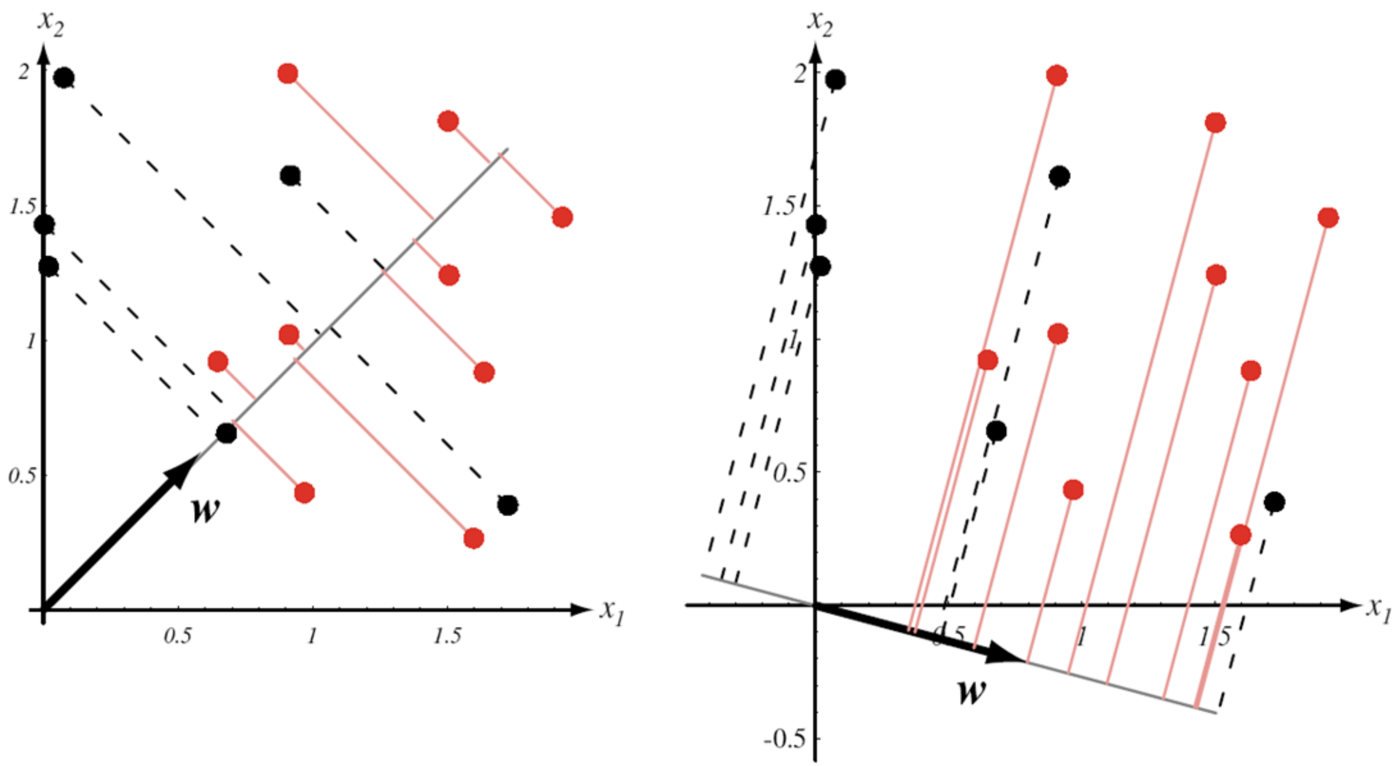
$$g(x) = w^t x + w_0 = r\|w\|$$

$$\text{therefore } r = \frac{g(x)}{\|w\|}$$

$$\text{in particular } d(0, H) = \frac{w_0}{\|w\|}$$

The orientation of the surface is determined by the normal vector w and the location of the surface is determined by the bias.

Fisher Linear Discriminant Analysis (FLDA)



- $X=\{x_1, x_2,\ldots, x_n\}=D_1 \cup D_2$
- $y=w^t x$: $X \rightarrow Y=\{y_1, y_2,\ldots,y_n\}=Y_1 \cup Y_2$
- Sample mean $m_i = \frac{1}{n_i}\sum_{x\in D_i} x$

after projection

$$\tilde{m}_i = \frac{1}{n_i}\sum_{y\in Y_i} y = \frac{1}{n_i}\sum_{x\in D_i} w^t x = w^t m_i$$

$$|\tilde{m}_1 - \tilde{m}_2|^2 = |w^t(m_1 - m_2)|^2 = w^t S_B w$$

where $S_B = (m_1 - m_2)(m_1 - m_2)^t$ is between-class scatter matrix.

$rank(S_B) \leq 1$

$$\tilde{s}_i^2 = \sum_{y \in Y_i} (y - \tilde{m}_i)^2 = w^t S_i w$$

where $S_i = \sum_{x \in D_i} (x - m_i)(x - m_i)^t$ is within-class scatter matrix of class ω_i

- The sum of these scatters can be written $S_W = S_1 + S_2$
- Fisher criterion function：

$$\boxed{J(w) = \frac{(\tilde{m}_1 - \tilde{m}_2)^2}{\tilde{s}_1^2 + \tilde{s}_2^2} = \frac{w^t S_B w}{w^t S_W w}}$$

- A vector w that maximizes J(w) must satisfy $S_B w=\lambda S_W w$, which is a generalized eigenvalue problem
- If S_W is nonsingular, $S_W^{-1}S_B w=\lambda w$
- In our particular case, it is unnecessary to solve for the eigenvalues and eigenvectors of $S_W^{-1}S_B$， due to the fact that $S_B w$ is always in the direction of m_1-m_2.
- Since the scale factor for w is immaterial, the solution for the w that optimizes J(w)：
$w=S_W^{-1}(m_1-m_2)$

How to find y_0 ?

- Bayesian optimal rule
- Experience

$$y_0^{(1)} = \frac{\tilde{m}_1 + \tilde{m}_2}{2}$$

$$y_0^{(2)} = \frac{n_1\tilde{m}_1 + n_2\tilde{m}_2}{n_1 + n_2} = \tilde{m}$$

$$y_0^{(3)} = \frac{\tilde{m}_1 + \tilde{m}_2}{2} + \frac{\ln(P(\omega_1)/P(\omega_2))}{n_1 + n_2 - 2}$$

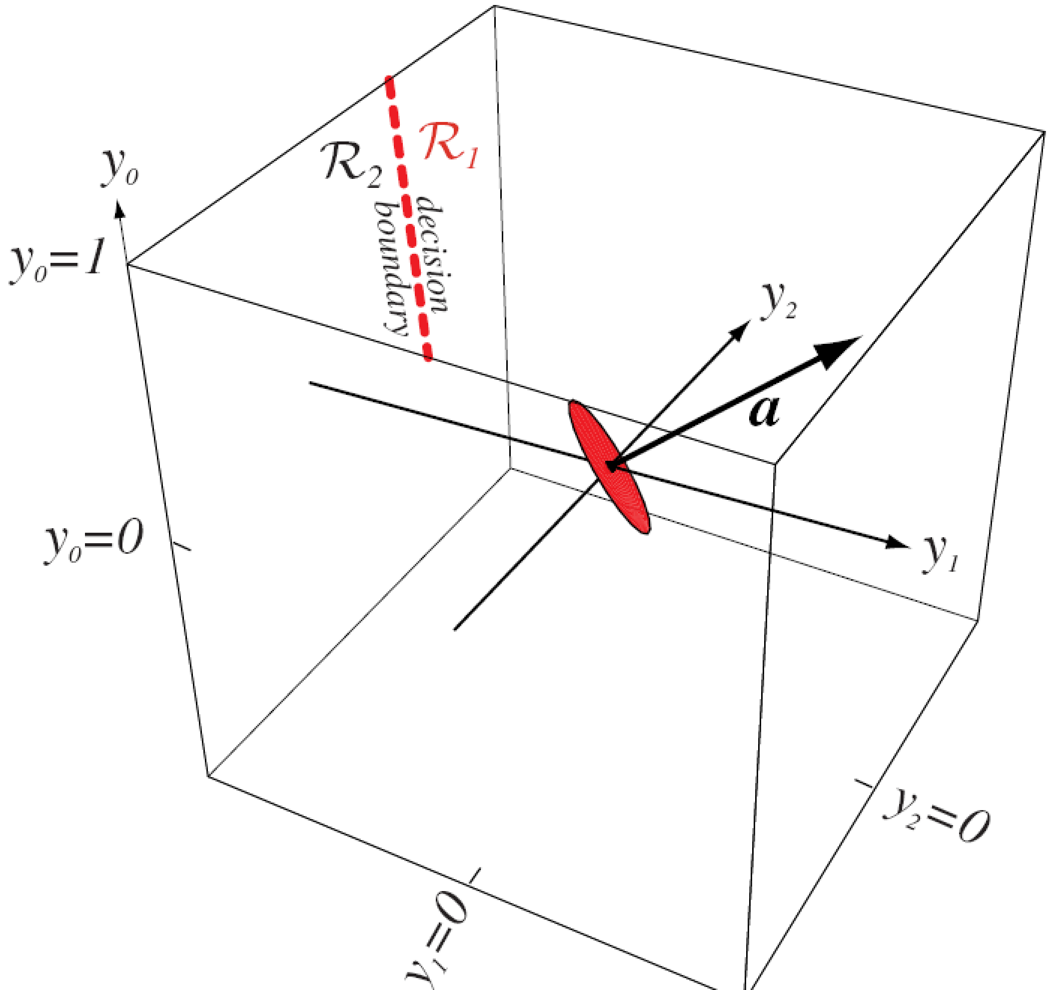
- If $y>y_0$ ➡ $\omega = \omega_1$;otherwise $\omega = \omega_2$

Augmented feature vector y and augmented weight vector a

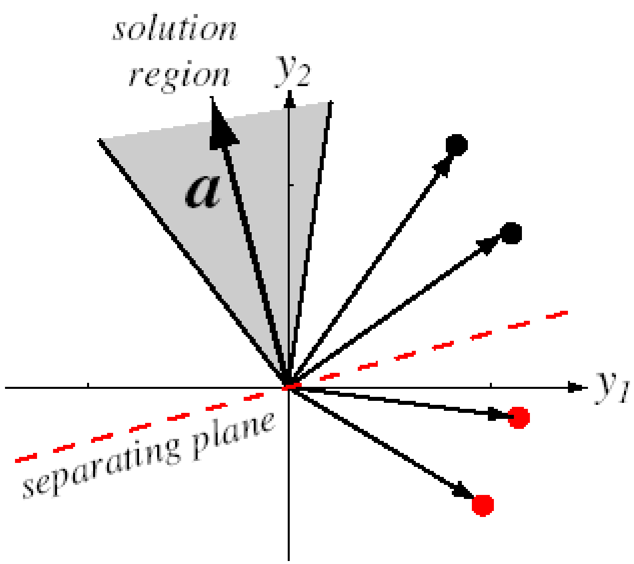
$$g(x) = w_0 + \sum_{i=1}^{d} w_i x_i = \sum_{i=0}^{d} w_i x_i = a^T y$$

where

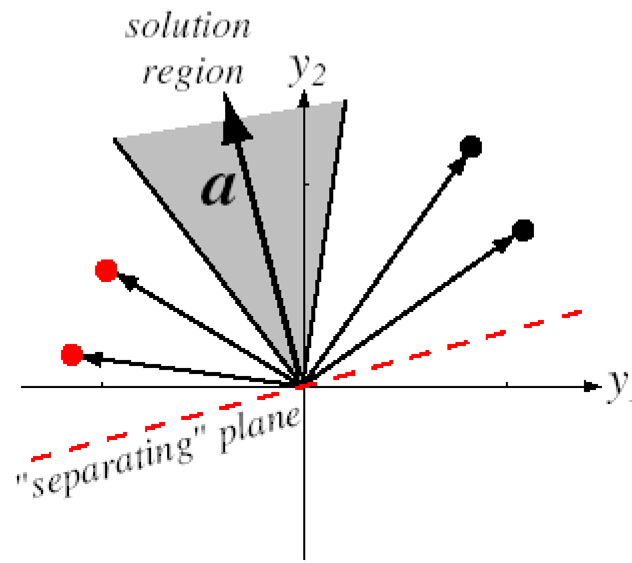
$$y = \begin{bmatrix} 1 \\ x_1 \\ x_2 \\ \vdots \\ x_d \end{bmatrix} = \begin{bmatrix} 1 \\ x \end{bmatrix}, a = \begin{bmatrix} w_0 \\ w_1 \\ w_2 \\ \vdots \\ w_d \end{bmatrix} = \begin{bmatrix} w_0 \\ w \end{bmatrix}$$


y_0
$y_0=1$
$y_0=0$
$\mathcal{R}_2$
$\mathcal{R}_1$
decision boundary
y_2
$\mathbf{a}$
y_1
$y_2=0$
$y_1=0$

Linearly separable and normalization



unnormalized

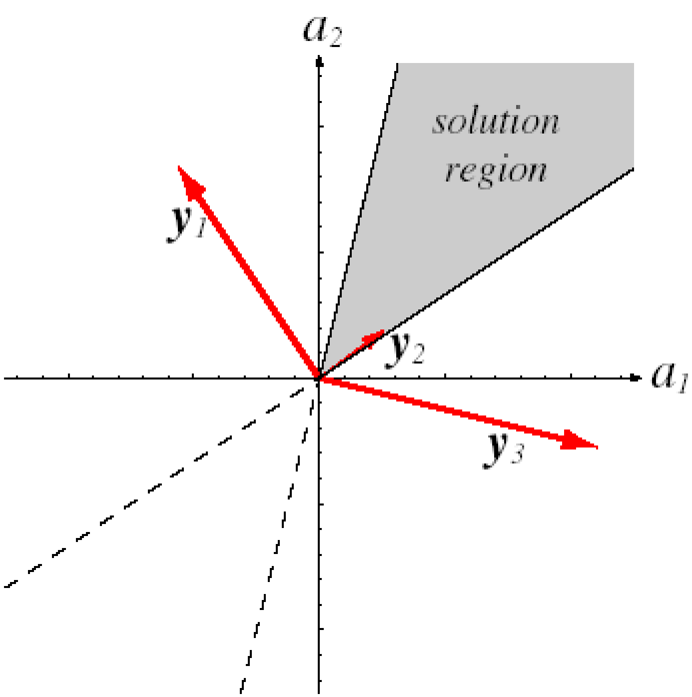


normalized

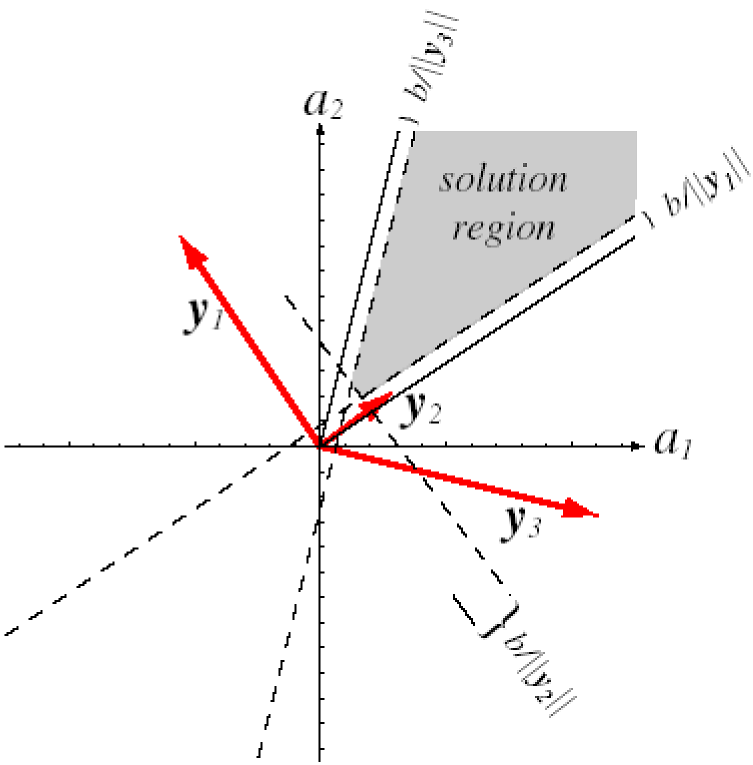
Solution vector

- If solution vector exists, it is not unique.
- Additional limiting condition
  - seek a unit-length solution vector that maximizes the minimum distance from the samples to the separating plane.
  - seek the minimum-length weight vector satisfying $a^t y_i > b$ for all i, where b is a positive constant called the *margin*.
- Define a criterion function J(a) that is minimized if is a solution vector.

Margin



without margin



with margin

Gradient descent procedures

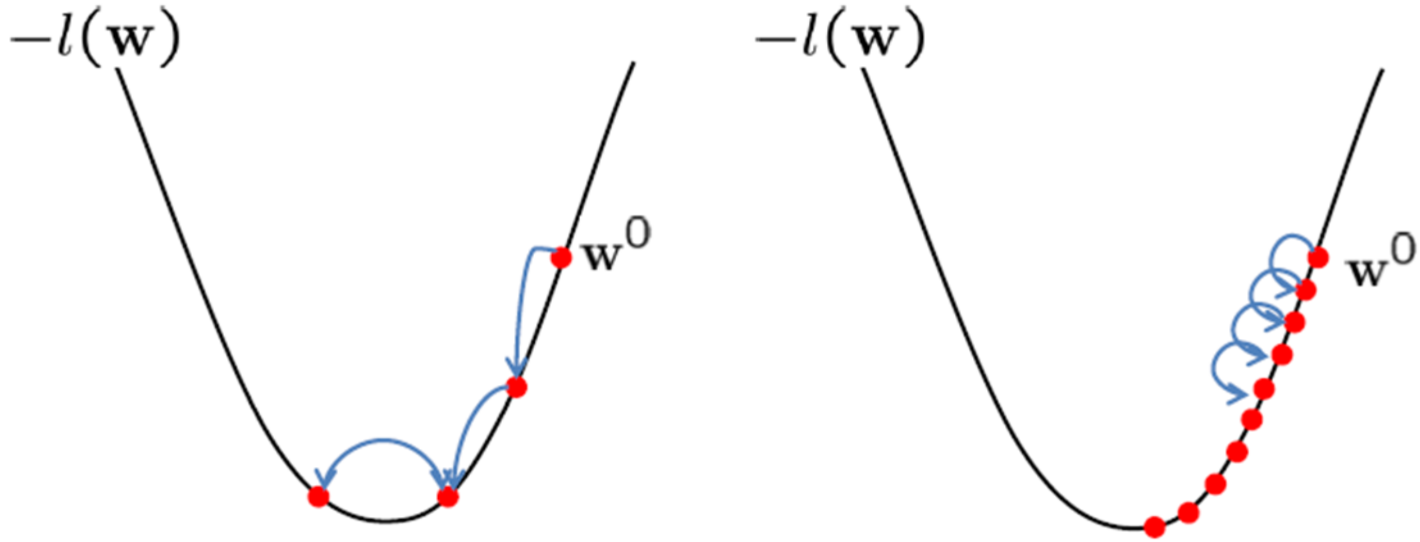
$$\mathbf{a}(k+1) = \mathbf{a}(k) - \eta(k)\nabla J(\mathbf{a}(k)),$$

η(k) is learning rate that sets the step size

Algorithm 1 (Basic gradient descent)

begin initialize $\mathbf{a}$, criterion $\theta, \eta(\cdot), k = 0$
    do $k \leftarrow k + 1$
        $\mathbf{a} \leftarrow \mathbf{a} - \eta(k)\nabla J(\mathbf{a})$
    until $\eta(k)\nabla J(\mathbf{a}) < \theta$
return $\mathbf{a}$
end

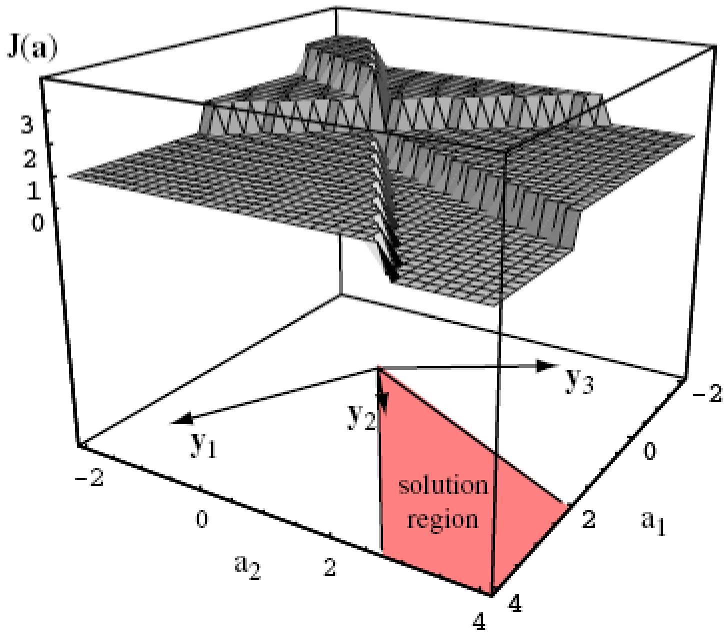
Effect of step-size η



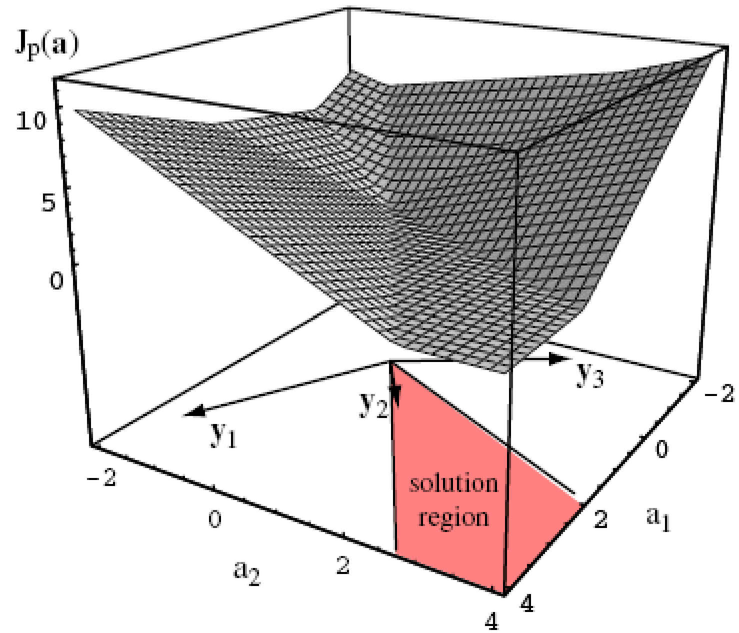
Large η => Fast convergence but larger residual error
Also possible oscillations

Small η => Slow convergence but small residual error

Perceptron



The number of misclassified samples



perceptron criterion

Criterion function

$$\boxed{J_p(a) = \sum_{y \in Y} (-a^t y)}$$

Y is the set of samples misclassified by a. (If no samples are misclassified, Y is empty and we define $J_p(a)$ to be 0.)

$$\nabla J_p = \sum_{y \in Y} (-y)$$

$$a(k+1) = a(k) + \eta(k) \sum_{y \in Y_k} y$$

Batch perceptron algorithm

Algorithm 3 (Batch Perceptron)

begin initialize $\mathbf{a}, \eta(\cdot)$, criterion $\theta, k = 0$

  do $k \leftarrow k + 1$

    $\mathbf{a} \leftarrow \mathbf{a} + \eta(k) \sum_{\mathbf{y} \in \mathcal{Y}_k} \mathbf{y}$

  until $\eta(k) \sum_{\mathbf{y} \in \mathcal{Y}_k} \mathbf{y} < \theta$

  return $\mathbf{a}$

end

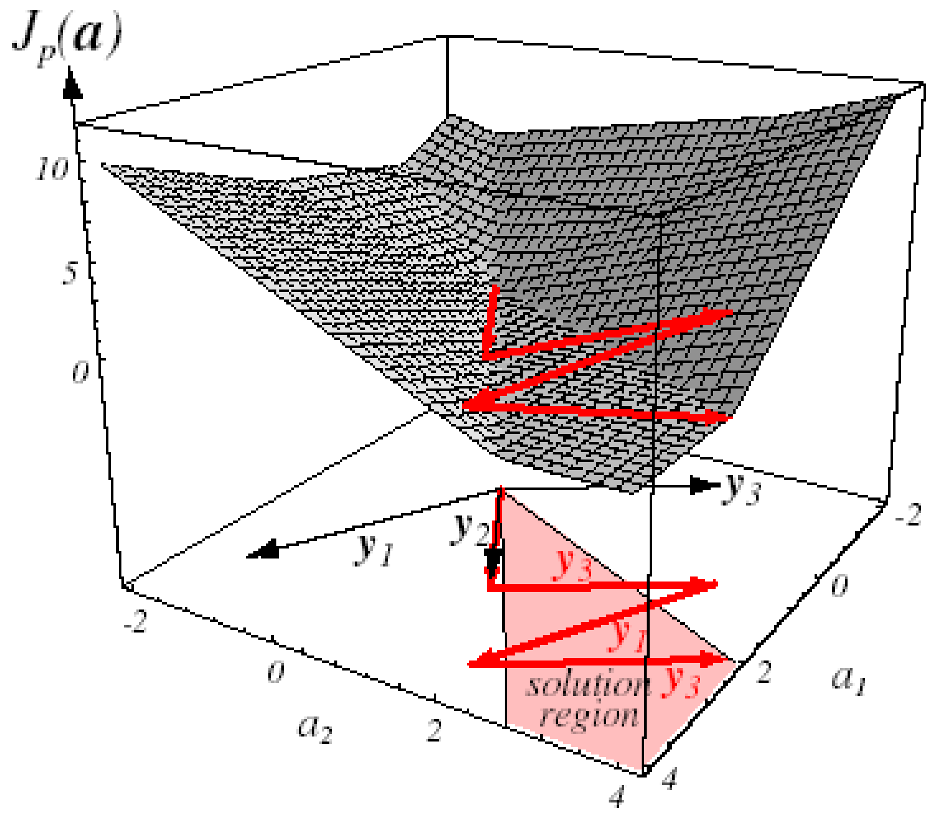
Fixed-increment single-sample perceptron

Algorithm 4 (Fixed-increment single-sample Perceptron)

```
begin initialize  a, k = 0
        do  k ← (k + 1) mod n
              if  y_k is misclassified by a  then  a ← a − y_k
        until all patterns properly classified
    return a
end
```

Theorem 5.1 (Perceptron Convergence) *If training samples are linearly separable then the sequence of weight vectors given by Algorithm 4 will terminate at a solution vector.*

Example



Variable increment perceptron with margin

$$\left.\begin{array}{ll} \mathbf{a}(1) & \text{arbitrary} \\ \mathbf{a}(k+1) = \mathbf{a}(k) + \eta(k)\mathbf{y}^k & k \geq 1, \end{array}\right\}$$

Algorithm 5 (Variable increment Perceptron with margin)

<u>**begin initialize**</u> $\mathbf{a}$, criterion θ, margin $b, \eta(\cdot), k = 0$

    <u>**do**</u> $k \leftarrow k + 1$

        <u>**if**</u> $\mathbf{a}^t\mathbf{y}_k + b < 0$ <u>**then**</u> $\mathbf{a} \leftarrow \mathbf{a} - \eta(k)\mathbf{y}_k$

    <u>**until**</u> $\mathbf{a}^t\mathbf{y}_k + b \leq 0$ for all k

  <u>**return**</u> $\mathbf{a}$

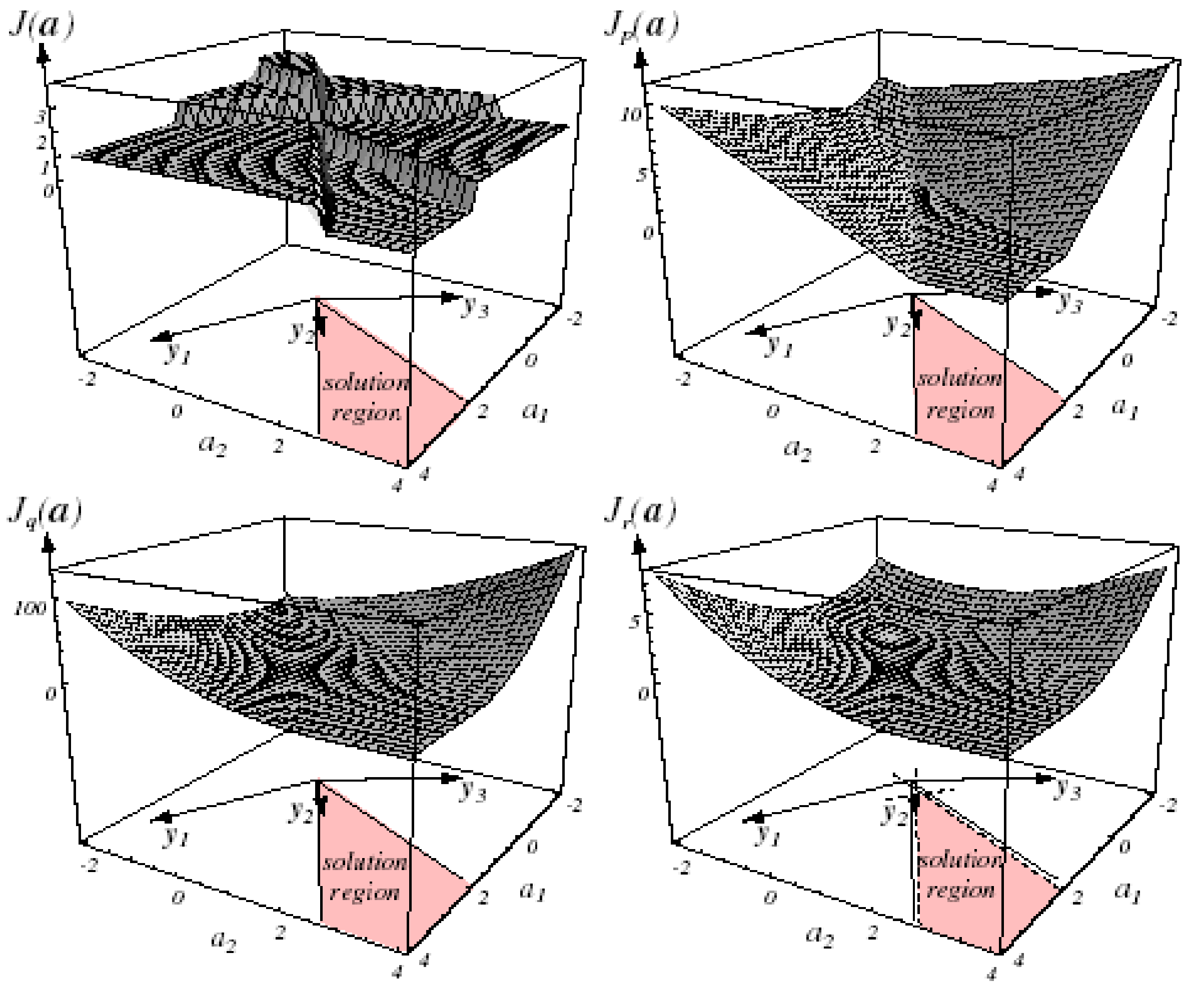
<u>**end**</u>

Balanced Winnow

Algorithm 7 (Balanced Winnow)

begin initialize $\mathbf{a}^+, \mathbf{a}^-, \eta(\cdot), k \leftarrow 0, \alpha > 1$
        if $\text{sign}[\mathbf{a}^{+t}\mathbf{y}_k - \mathbf{a}^{-t}\mathbf{y}_k] \neq z_k$ (pattern misclassified)
        then **if** $z_k = +1$ **then** $a_i^+ \leftarrow \alpha^{+y_i} a_i^+;\ a_i^- \leftarrow \alpha^{-y_i} a_i^-$ for all i
            if $z_k = -1$ **then** $a_i^+ \leftarrow \alpha^{-y_i} a_i^+;\ a_i^- \leftarrow \alpha^{+y_i} a_i^-$ for all i
    return $\mathbf{a}^+, \mathbf{a}^-$
end

Relaxation procedures



$$J_p(a) = \sum_{y \in Y} (-a^t y)$$

$$J_q(a) = \sum_{y \in Y} (a^t y)^2$$

$$\boxed{J_r(a) = \frac{1}{2} \sum_{y \in Y} \frac{(a^t y - b)^2}{\|y\|^2}}$$

Batch relaxation with margin

$$\left.\begin{array}{ll}\mathbf{a}(1) & \text{arbitrary} \\ \mathbf{a}(k+1) = \mathbf{a}(k) + \eta(k) \sum\limits_{\mathbf{y}\in\mathcal{Y}} \frac{b-\mathbf{a}^t\mathbf{y}}{\|\mathbf{y}\|^2}\mathbf{y}. & \end{array}\right\}$$

Algorithm 8 (Batch relaxation with margin)

begin initialize $\mathbf{a}, \eta(\cdot), k = 0$
    do $k \leftarrow k + 1$
        $\mathcal{Y}_k = \{\}$
        $j = 0$
        do $j \leftarrow j + 1$
            if $\mathbf{y}_j$ is misclassified **then** Append $\mathbf{y}_j$ to $\mathcal{Y}_k$
        until $j = n$
        $\mathbf{a} \leftarrow \mathbf{a} + \eta(k) \sum\limits_{\mathbf{y}\in\mathcal{Y}} \frac{b-\mathbf{a}^t\mathbf{y}}{\|\mathbf{y}\|^2}\mathbf{y}$
    until $\mathcal{Y}_k = \{\}$
return a
end

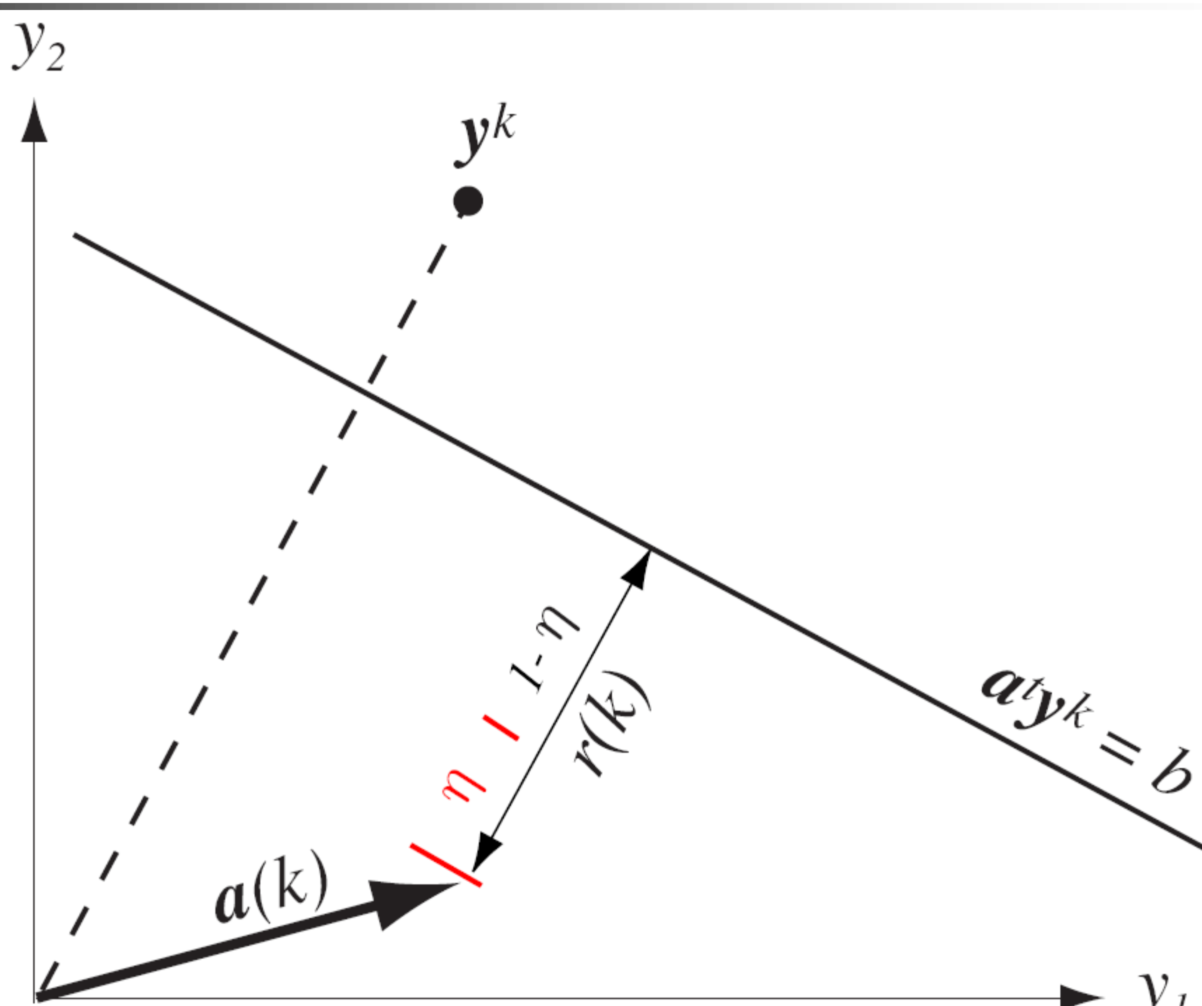
Single-sample relaxation with margin

$$\left.\begin{array}{ll} \mathbf{a}(1) & \text{arbitrary} \\ \mathbf{a}(k+1) = \mathbf{a}(k) + \eta \frac{b - \mathbf{a}^t(k)\mathbf{y}^k}{\|\mathbf{y}^k\|^2}\mathbf{y}^k, & \end{array}\right\}$$

Algorithm 9 (Single-sample relaxation with margin)

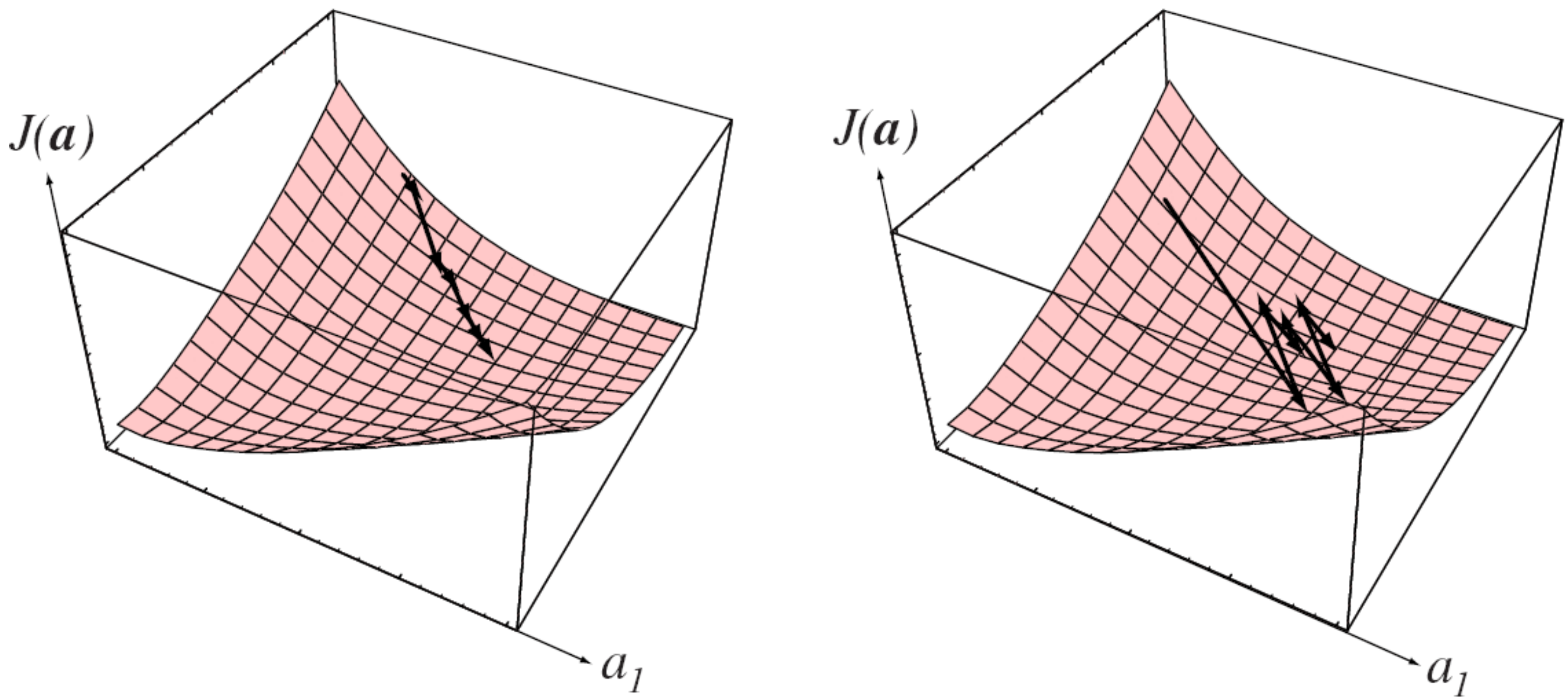
<u>**begin**</u> <u>**initialize**</u> $\mathbf{a}, \eta(\cdot), k = 0$
    <u>**do**</u> $k \leftarrow k + 1$
        <u>**if**</u> $\mathbf{y}_k$ is misclassified <u>**then**</u> $\mathbf{a} \leftarrow \mathbf{a} + \eta(k)\frac{b - \mathbf{a}^t\mathbf{y}}{\|\mathbf{y}_k\|^2}\mathbf{y}_k$
        <u>**until**</u> all patterns properly classified
    <u>**return**</u> $\mathbf{a}$
<u>**end**</u>

Geometric explanation



$a^t(k+1)\, y^k - b = (1-\eta)\,(a^t(k)\, y^k - b) \quad (0<\eta<2)$

- If $\eta<1$, then $a^t(k+1)\, y^k$ is still less than b "underrelaxation"
- If $\eta>1$, then $a^t(k+1)\, y^k$ greater than b "overrelaxation"



Nonseparable behavior

- Error-correcting procedures
- Since no weight vector can correctly classify every sample in a nonseparable set (by definition), it is clear that the corrections in an error-correction procedure can never cease.

Minimum Squared Error Procedures

- try to make $\boldsymbol{a}^t\mathbf{y}_i = b_i$, where b_i are some arbitrarily specified positive constants.

$$\begin{pmatrix} Y_{10} & Y_{11} & \cdots & Y_{1d} \\ Y_{20} & Y_{21} & \cdots & Y_{2d} \\ \vdots & \vdots & & \vdots \\ \vdots & \vdots & & \vdots \\ \vdots & \vdots & & \vdots \\ Y_{n0} & Y_{n1} & \cdots & Y_{nd} \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_d \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ \vdots \\ \vdots \\ b_n \end{pmatrix}$$

$$\text{or} \qquad \mathbf{Ya} = \mathbf{b}.$$

Minimum squared error criterion

- If ***Y*** is nonsingular， ***a*=Y^{-1}*b*.**
- If ***Y*** is rectangular (usually more equations than unknowns), ***a*** is overdetermined, and ordinarily no exact solution exists

$$J_s(a) = \|Ya - b\|^2 = \sum_{i=1}^{n} (a^t y_i - b_i)^2$$

- Two ways to minimize $J_s(a)$
  - Pseudoinverse
  - Widrow-Hoff procedure

Pseudoinverse

$$\nabla J_s = \sum_{i=1}^{n} 2(a^t y_i - b_i) y_i = 2Y^t (Ya - b)$$

$$\Downarrow$$

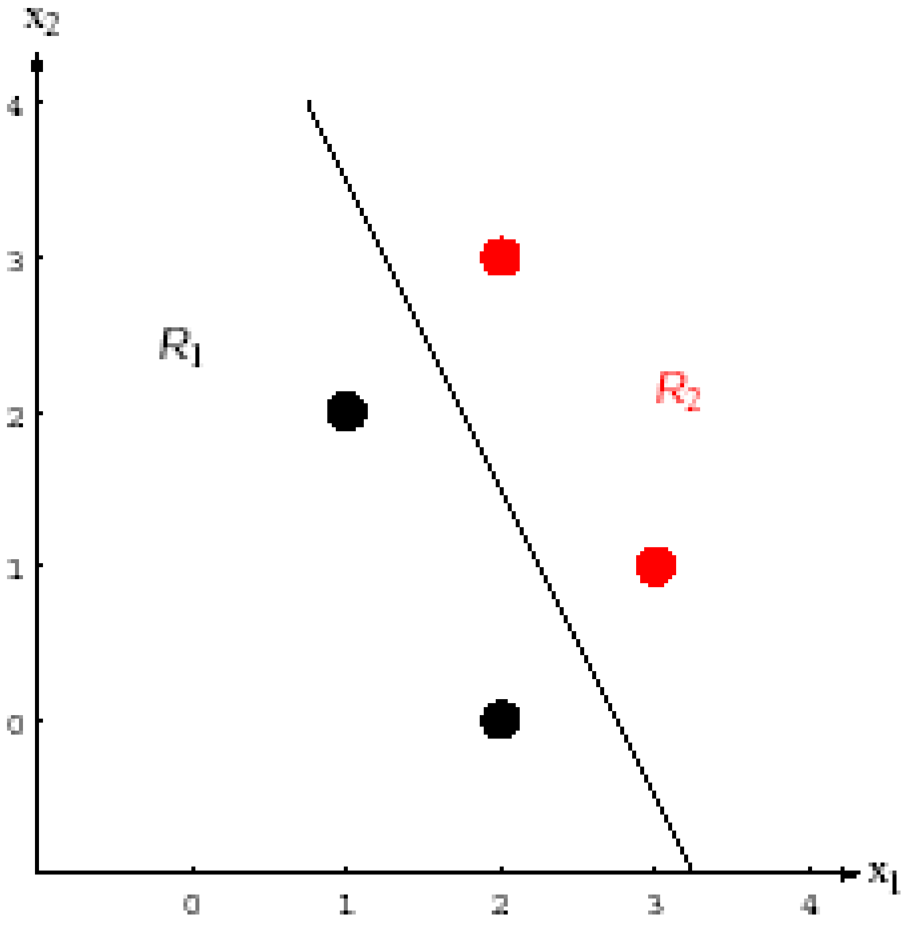
$$Y^t Y a = Y^t b \quad \Rightarrow \quad a = (Y^t Y)^{-1} Y^t b = Y^+ b$$

$$Y^+ \equiv (Y^t Y)^{-1} Y^t, \quad \text{pseudoinverse matrix}$$

$$\Downarrow \quad YY^+ \neq I$$

$$Y^+ \equiv \lim_{\varepsilon \to 0} (Y^t Y + \varepsilon I)^{-1} Y^t$$

Example



$$\mathbf{Y} = \begin{pmatrix} 1 & 1 & 2 \\ 1 & 2 & 0 \\ -1 & -3 & -1 \\ -1 & -2 & -3 \end{pmatrix}$$

$$\mathbf{Y}^{\dagger} \equiv \lim_{\epsilon \to 0} (\mathbf{Y}^t \mathbf{Y} + \epsilon \mathbf{I})^{-1} \mathbf{Y}^t = \begin{pmatrix} 5/4 & 13/12 & 3/4 & 7/12 \\ -1/2 & -1/6 & -1/2 & -1/6 \\ 0 & -1/3 & 0 & -1/3 \end{pmatrix}$$

$$\mathbf{b} = (1, 1, 1, 1)^t$$

$$\mathbf{a} = \mathbf{Y}^{\dagger}\mathbf{b} = (11/3, -4/3, -2/3)^t$$

Hyperplane: $4x_1+2x_2-11=0$

Widrow-Hoff procedure

$$\nabla J_s = \sum_{i=1}^{n} 2(a^t y_i - b_i) y_i = 2Y^t (Ya - b)$$

$$a(1) \qquad \text{arbitrary}$$

$$a(k+1) = a(k) - \eta(k) Y^t (Ya(k) - b)$$

If $\eta(k) = \eta(1)/k$, where $\eta(1)$ is any positive constant, then this rule generates a sequence of weight vectors that converges to a limiting vector **a** satisfying

Least Mean Squared (LMS)

$$\left.\begin{array}{ll} \mathbf{a}(1) & \text{arbitrary} \\ \mathbf{a}(k+1) = \mathbf{a}(k) + \eta(k)(b_k - \mathbf{a}(k)^t\mathbf{y}^k)\mathbf{y}^k, & \end{array}\right\}$$

Algorithm 10 (LMS)

begin initialize $\mathbf{a}, \mathbf{b}$, criterion $\theta, \eta(\cdot), k = 0$

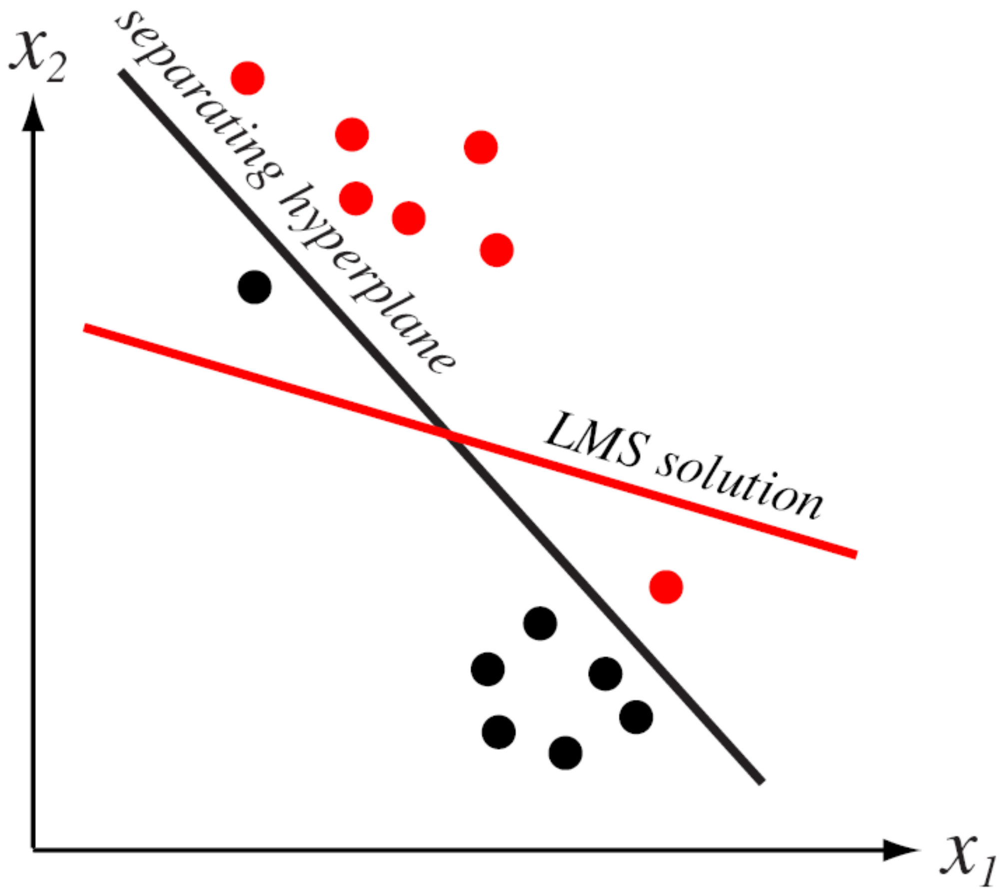
    do $k \leftarrow k + 1$

        $\mathbf{a} \leftarrow \mathbf{a} + \eta(k)(b_k - \mathbf{a}^t\mathbf{y}^k)\mathbf{y}^k$

    until $\eta(k)(b_k - \mathbf{a}^t\mathbf{y}^k)\mathbf{y}^k < \theta$

    return $\mathbf{a}$

end


x_2
separating hyperplane
LMS solution
x_1

Ho-Kashyap Procedures

- MSE criterion $J_s(a,b) = \|Ya - b\|^2$
- If linearly separable, then $Y\hat{a} = \hat{b} > 0$
- But we usually do not know $\hat{b}$ beforehand.
- If the samples are separable, and if both a and b in $J_s(a)$ are allowed to vary (b>0), then the minimum value of J_s is zero, and a that achieves that minimum is a separating vector.

$$\nabla_a J_s = 2Y^t(Ya - b)$$

$$\nabla_b J_s = -2(Ya - b) \xrightarrow{\text{set } 0} a = Y^+ b$$

We are not so free to modify b, since b>0, and we must avoid converging to b=0. One way is to start with b>0 and to refuse to reduce any of its components.

$$b(k+1) = b(k) - \eta \frac{1}{2}[\nabla_b J_s - |\nabla_b J_s|]$$

$$\left.\begin{array}{ll}\mathbf{b}(1) > \mathbf{0} & \text{but otherwise arbitrary} \\ \mathbf{b}(k+1) = \mathbf{a}(k) + 2\eta(k)\mathbf{e}^{+}(k), & \end{array}\right\}$$

where $\mathbf{a}(k) = \mathbf{Y}^{\dagger}\mathbf{b}(k), \qquad k = 1, 2, \ldots$

$$\mathbf{e}^{+}(k) = \frac{1}{2}(\mathbf{e}(k) + |\mathbf{e}(k)|),$$

$$\mathbf{e}(k) = \mathbf{Y}\mathbf{a}(k) - \mathbf{b}(k),$$

Algorithm 11 (Ho-Kashyap)

begin initialize $\mathbf{a}, \mathbf{b}, \eta(\cdot) < 1$, criteria b_{min}, k_{max}

  do $k \leftarrow k + 1$

    $\mathbf{e} \leftarrow \mathbf{Ya} - \mathbf{b}$

    $\mathbf{e}^{+} \leftarrow 1/2(\mathbf{e} + \mathrm{Abs}[\mathbf{e}])$

    $\mathbf{b} \leftarrow \mathbf{a} + 2\eta(k)\mathbf{e}^{+}$

    $\mathbf{a} \leftarrow \mathbf{Y}^{\dagger}\mathbf{b}$

    if $\mathrm{Abs}[\mathbf{e}] \leq b_{min}$ **then return** $\mathbf{a}, \mathbf{b}$ *and* **exit**

  until $k = k_{max}$

 Print NO SOLUTION FOUND

end

- e(k)=0 and we have a solution
- e(k)≤0 and we have proof that the samples are not linearly separable.

Modification (I)

- $\mathbf{Y}^t\mathrm{e}(k)=0$

$$\left.\begin{array}{rcl} \mathbf{b}(1) & > & 0 \qquad \text{but otherwise arbitrary} \\ \mathbf{a}(1) & = & \mathbf{Y}^{\dagger}\mathbf{b}(1) \\ \mathbf{b}(k+1) & = & \mathbf{b}(k) + \eta(\mathbf{e}(k) + |\mathbf{e}(k)|) \\ \mathbf{a}(k+1) & = & \mathbf{a}(k) + \eta\mathbf{Y}^{\dagger}|\mathbf{e}(k)|, \end{array}\right\}$$

- It varies both the weight vector a and the margin vector b
- It provides evidence of nonseparability
- But it requires the computation of Y^+.

Modification (II)

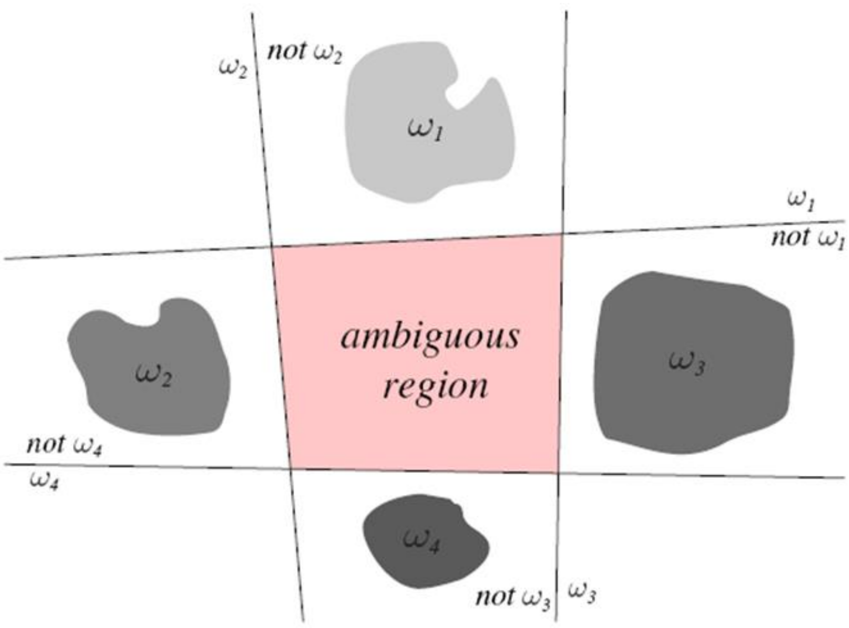
- Avoid the need for computing Y^+

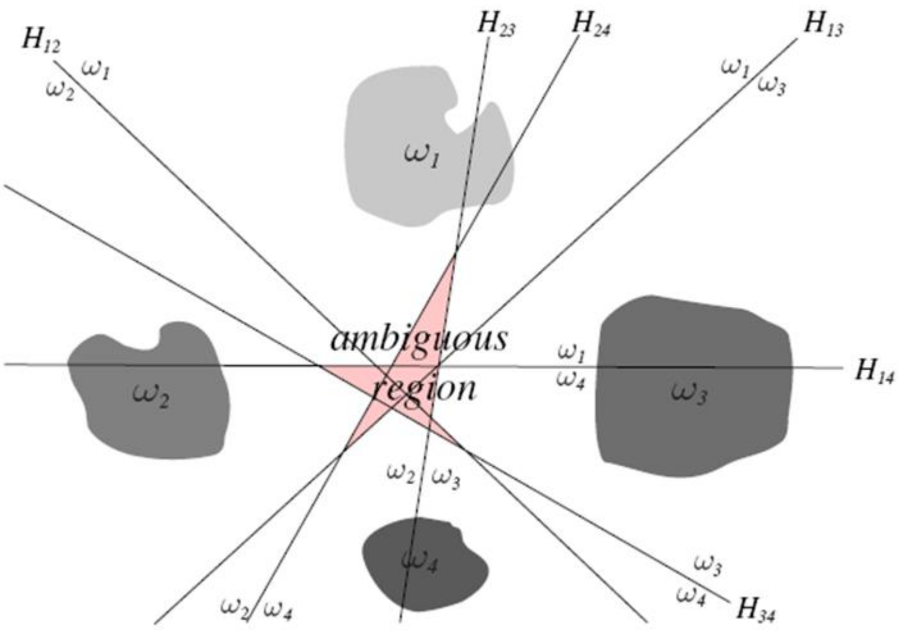
$$\left.\begin{array}{rcl} \mathbf{b}(1) & > & 0 \quad \text{but otherwise arbitrary} \\ \mathbf{a}(1) & = & \text{arbitrary} \\ \mathbf{b}(k+1) & = & \mathbf{b}(k) + (\mathbf{e}(k) + |\mathbf{e}(k)|) \\ \mathbf{a}(k+1) & = & \mathbf{a}(k) + \eta \mathbf{R}\mathbf{Y}^t |\mathbf{e}(k)| \end{array}\right\}$$

where R is an arbitrary, constant, positive-definite $\hat{d} \times \hat{d}$ matrix.

- Assuring convergence if $0 < \eta < 2/\lambda_{max}$, λ_{max} is the largest eigenvalue of $\mathbf{Y}^t\mathbf{Y}$.

The multi-category case





- We define c linear discriminant functions

$$g_i(x) = w_i^t x + w_{i0} \qquad i = 1,\ldots,c$$

and assign x to ω_i if $g_i(x) > g_j(x)\ \forall\, j \neq i$; in case of ties, the classification is undefined.

- In this case, the classifier is a "linear machine".

- A linear machine divides the feature space into c decision regions, with $g_i(x)$ being the largest discriminant if x is in the region $\mathcal{R}_i$
- For a two contiguous regions $\mathcal{R}_i$ and $\mathcal{R}_j$; the boundary that separates them is a portion of hyperplane H_{ij} defined by:

$$g_i(x) = g_j(x)$$

$$\Leftrightarrow (w_i - w_j)^t x + (w_{i0} - w_{j0}) = 0$$

- $w_i - w_j$ is normal to H_{ij} and

$$d(x, H_{ij}) = \frac{g_i - g_j}{\|w_i - w_j\|}$$

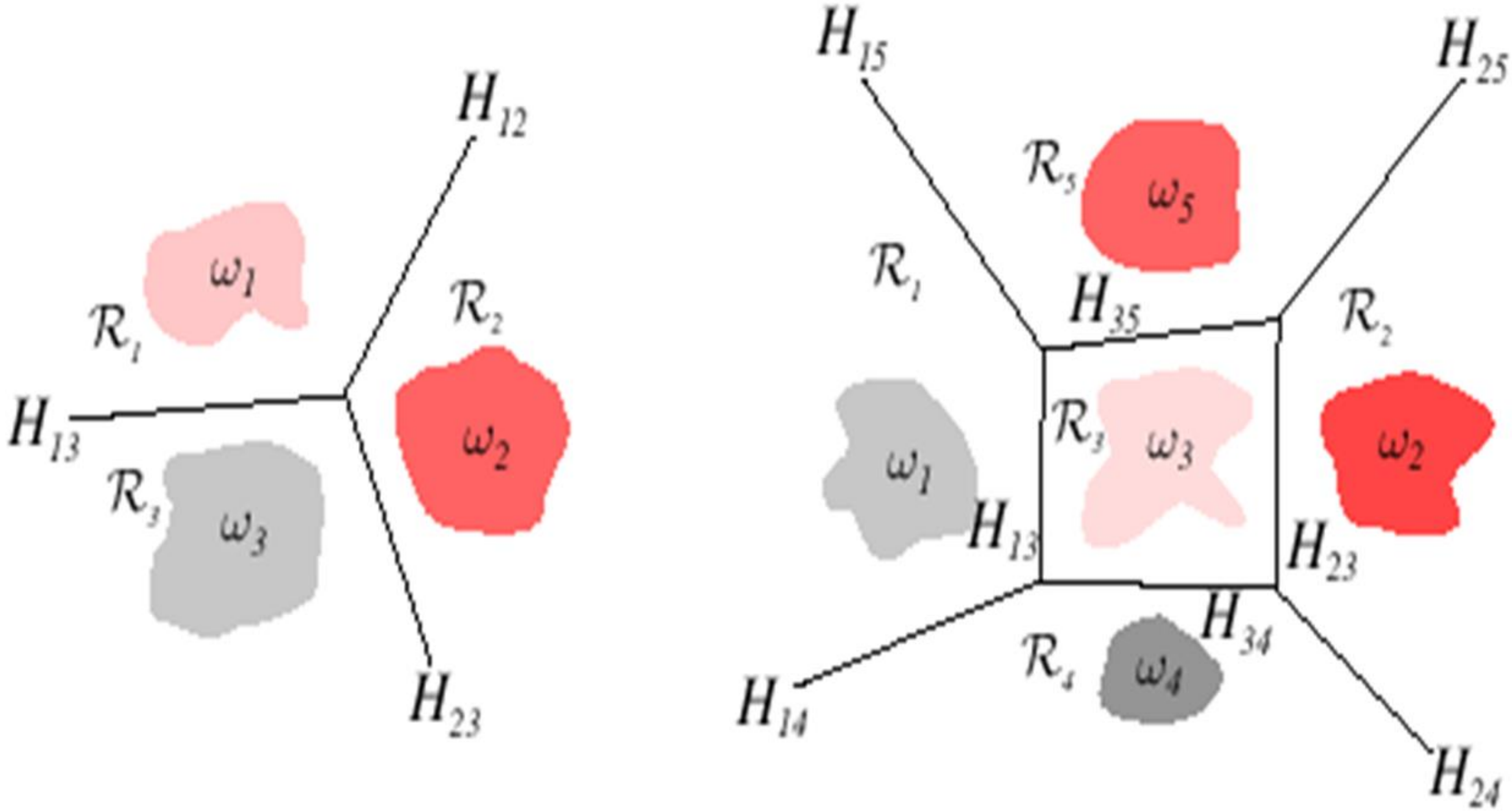
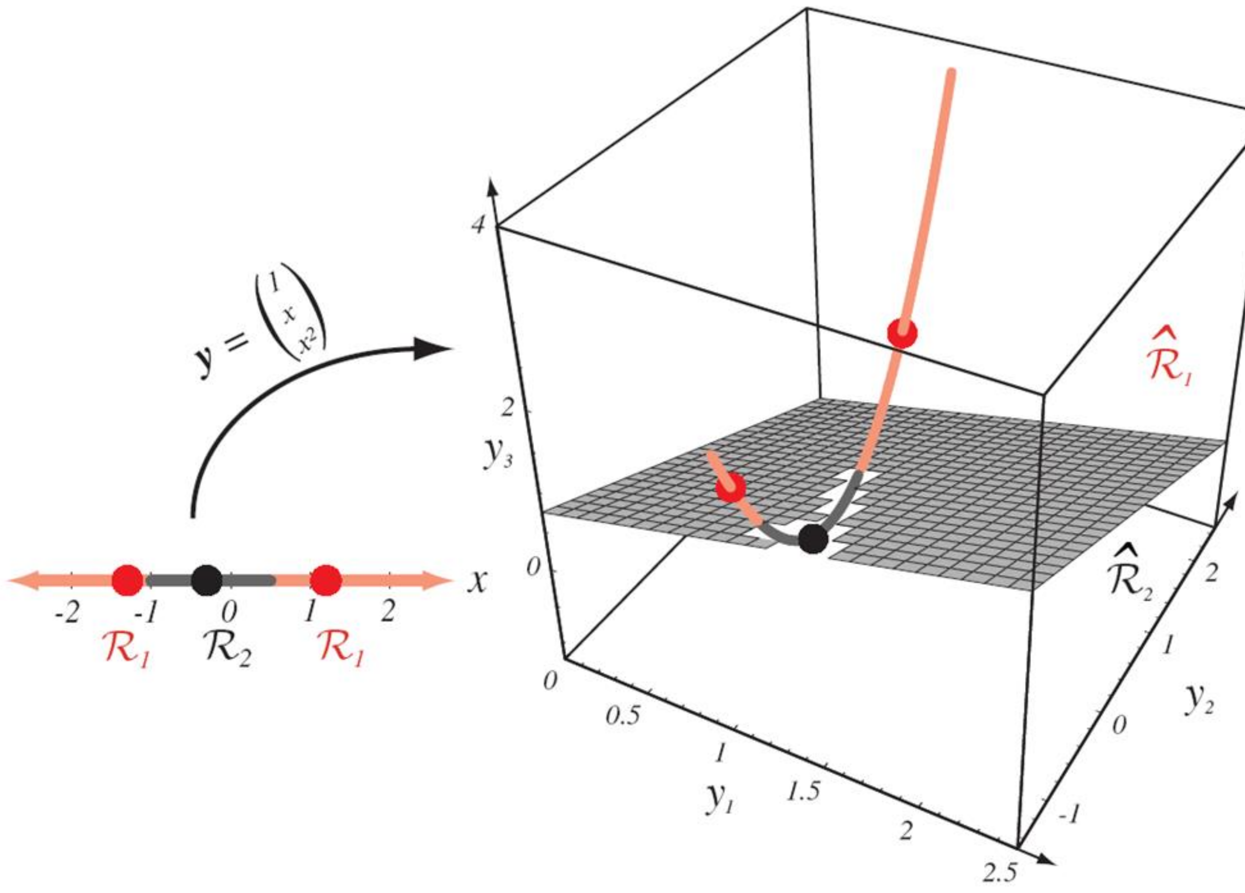


FIGURE 5.4. Decision boundaries produced by a linear machine for a three-class problem and a five-class problem. From: Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*. Copyright © 2001 by John Wiley & Sons, Inc.

It is easy to show that the decision regions for a linear machine are convex, this restriction limits the flexibility and accuracy of the classifier.

Generalized Linear Discriminant Functions



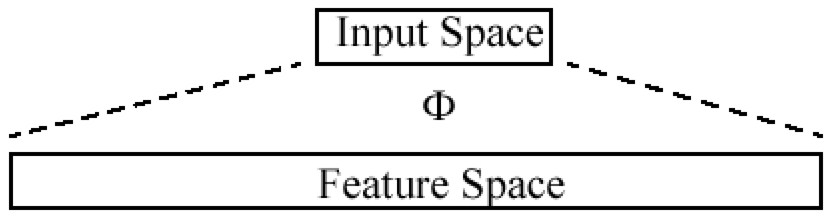
- Decision boundaries which separate between classes may not always be linear.
- The complexity of the boundaries may sometimes request the use of highly nonlinear surfaces.
- A popular approach to generalize the concept of linear decision functions is to consider a generalized decision function as:

$$g(x) = w_1 f_1(x) + w_2 f_2(x) + \dots + w_N f_N(x) + w_{N+1} \qquad (1)$$

where $f_i(x)$, $1 \le i \le N$ are scalar functions of the pattern x, $x \in R^n$ *(Euclidean Space)*

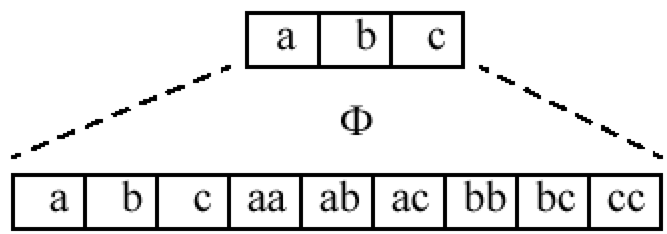
Extending the Hypothesis Space

Idea:



==> Find hyperplane in feature space!

Example:



==> The separating hyperplane in features space is a degree two polynomial in input space.

What you should know

- Directly design classifier from samples
  - Type of classifier or discriminant function
  - Criterion function L(a)
  - L(a*)=min L(a)
- Perceptron
- Minimum Squared Error Procedures