

# 机器学习

## Machine learning

## 第三章 线性分类

### Linear Classifier

授课人：周晓飞  
zhouxiaofei@iie.ac.cn  
2022-9-23

# 第三章 线性分类

3.1 概述

3.2 基础知识

**3.3 感知机**

3.4 线性鉴别分析

3.5 logistic 模型

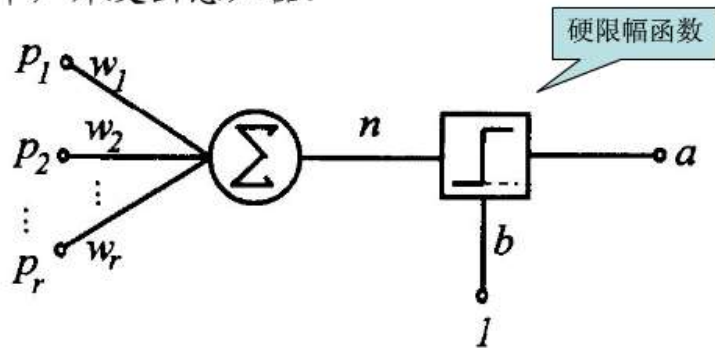
## 基本知识

### 1. 神经网络形成阶段（1943-1958），开拓性的贡献：

- McCulloch & Pitts（1943）引入神经网络的概念作为计算工具；

McCulloch和Pitts 1943年，发表第一个系统的ANN研究——阈值加权和 (M-P) 数学模型。

1947年，开发出感知器。



- Hebb（1949）提出自组织学习的第一个规则；
- Rosenblatt（1957）提出感知器作为有教师学习的一个模型。

## 基本知识

### 2. 线性分类

- 决策函数

$$g(\mathbf{x}) = \sum_{i=1}^m \mathbf{w}_i \mathbf{x}_i + w_0 = \mathbf{w}^T \mathbf{x} + w_0$$

- 增广表示

$$g(\mathbf{x}) = \sum_{i=0}^m \mathbf{w}_i \mathbf{x}_i = \tilde{\mathbf{w}}^T \tilde{\mathbf{x}}$$

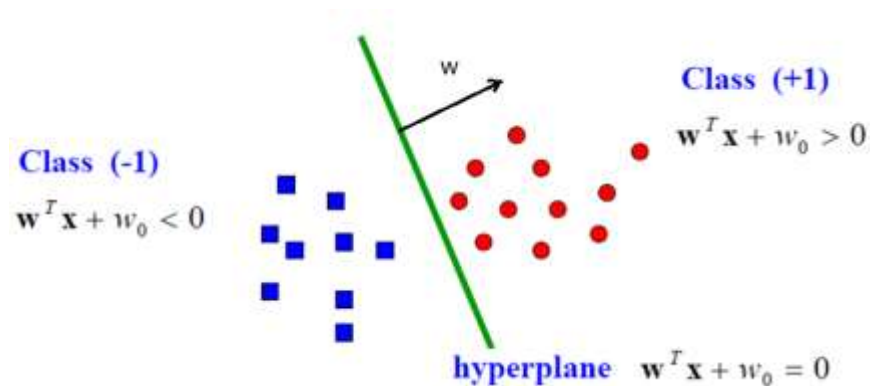
$$\text{其中, } \tilde{\mathbf{w}} = \begin{pmatrix} \mathbf{w} \\ w_0 \end{pmatrix}, \quad \tilde{\mathbf{x}} = \begin{pmatrix} \mathbf{x} \\ 1 \end{pmatrix}$$

## 基本知识

- 决策超平面  $g(x) = \mathbf{w}^T \mathbf{x} + w_0 = 0$
- 分类判别

*If  $\mathbf{w}^T \mathbf{x} + w_0 > 0$  assign  $\mathbf{x}$  to  $\omega_1$*

*If  $\mathbf{w}^T \mathbf{x} + w_0 < 0$  assign  $\mathbf{x}$  to  $\omega_2$*



## 基本知识

- 决策函数几何含义

刻画了样本到超平面的距离  $g(\mathbf{x}) = \|\mathbf{w}\| \cdot z$

- 验证函数:  $y_i(\mathbf{w}^T \mathbf{x}_i + w_0)$

$$\mathbf{w}^T \mathbf{x}_i + w_0 \geq 0 \quad \text{For all } i, \text{ such that } y_i = +1$$

$$\mathbf{w}^T \mathbf{x}_i + w_0 \leq 0 \quad \text{For all } i, \text{ such that } y_i = -1$$

$$\text{Together: } y_i(\mathbf{w}^T \mathbf{x}_i + w_0) \geq 0$$

## 基本知识

### 3. 优化方法 — 梯度下降

$$\min_w J(w) = \sum_i J_i(w)$$

- 梯度下降 (GD)

$$w = w - \eta \frac{\partial J(w)}{\partial w} = w - \eta \nabla J(w)$$

## 基本知识

### 3. 优化方法 — 梯度下降

$$\min_w J(w) = \sum_i J_i(w)$$

- 梯度下降 (GD)

$$w = w - \eta \frac{\partial J(w)}{\partial w} = w - \eta \nabla J(w) = w - \eta \sum_i \frac{\partial J_i(w)}{\partial w} = w - \eta \sum_i \nabla J_i(w)$$



## 基本知识

### 3. 优化方法 — 梯度下降

$$\min_w J(w) = \sum_i J_i(w)$$

- 梯度下降 (GD)

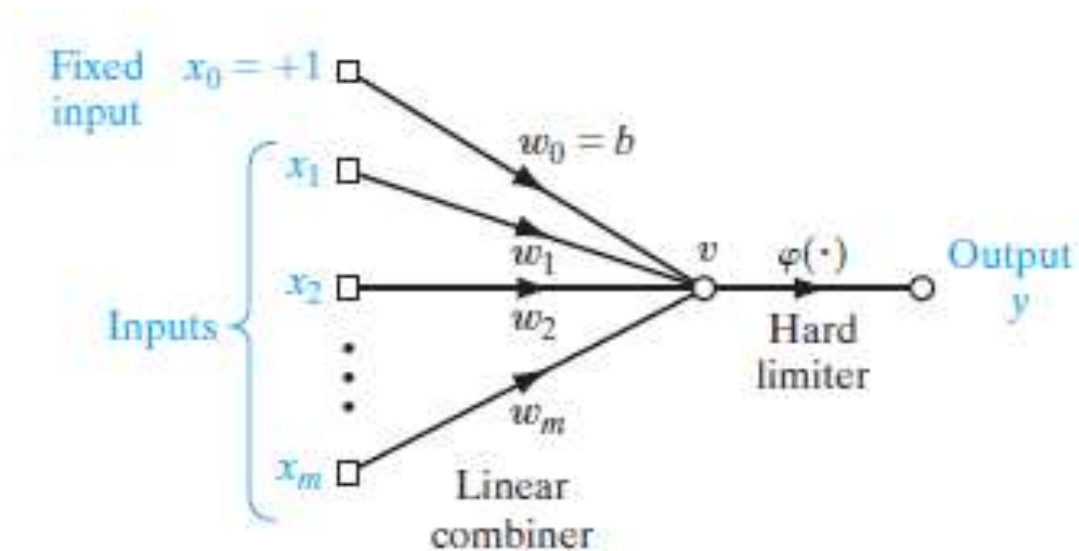
$$w = w - \eta \frac{\partial J(w)}{\partial w} = w - \eta \nabla J(w) = w - \eta \sum_i \frac{\partial J_i(w)}{\partial w} = w - \eta \sum_i \nabla J_i(w)$$

- 随机梯度下降 (SGD)

$$w = w - \eta \frac{\partial J_i(w)}{\partial w}$$

# 感知机

## 感知机结构



## 信号流

- 输入

$$\mathbf{x}(n) = [+1, x_1(n), x_2(n), \dots, x_m(n)]^T$$

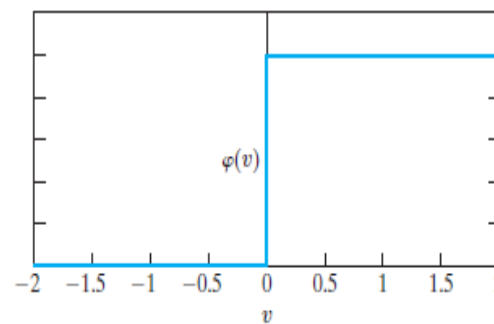
- 神经元连接权

$$\mathbf{w}(n) = [b, w_1(n), w_2(n), \dots, w_m(n)]^T$$

- 神经元局部感受域

$$\begin{aligned} v(n) &= \sum_{i=0}^m w_i(n) x_i(n) \\ &= \mathbf{w}^T(n) \mathbf{x}(n) \end{aligned}$$

- 硬激活函数



## 感知机学习准则

### 1. 目标：最小化 **错分样本的** 误差代价。

- 代价函数（错分样本的误差函数）：

$$J(\mathbf{w}) = \sum_{\mathbf{x}(n) \in E} -\mathbf{w}^T \mathbf{x}(n) d(n) \quad (1.1)$$

或者

$$J(\mathbf{w}) = \sum_{\mathbf{x}(n)} -\mathbf{w}^T \mathbf{x}(n) (d(n) - y(n)) \quad (1.2)$$

其中， $E$  为错误分类样本集； $d(n) \in \{-1, +1\}$  为  $\mathbf{x}(n)$  的已知类别标签； $y(n) \in \{-1, +1\}$  为感知器的输出类别

## 感知机学习准则

**问题：** $(d(n)-y(n))$ 能否替代“错误分类样本集筛选”、 $(d(n)-y(n))$ 能否替代  $d(n)$  ？

**答 1：**当样本被正确分类时 $(d(n)-y(n))=0$ ，正确分类样本被忽略，

$(d(n)-y(n))$ 可替代 “错误分类样本集筛选”；

**答 2：**当样本被错误分类时， $(d(n)-y(n))\neq 0$ ，两种情况

$d(n)=+1, y(n)=-1$ 时， $(d(n)-y(n))=+2$ ,

$(d(n)-y(n))$ 与 $d(n)$ 符号相同

$d(n)=-1, y(n)=+1$ 时， $(d(n)-y(n))=-2$ ,

$(d(n)-y(n))$ 与 $d(n)$ 符号相同

$(d(n)-y(n))$ 能替代 $d(n)$ ；

## 感知机学习准则

### 2. $J(w)$ 的含义：错分样本到分类超平面误差距离的总和

$$|z| = \frac{|\mathbf{w}^T \mathbf{x}|}{\|\mathbf{w}\|}$$

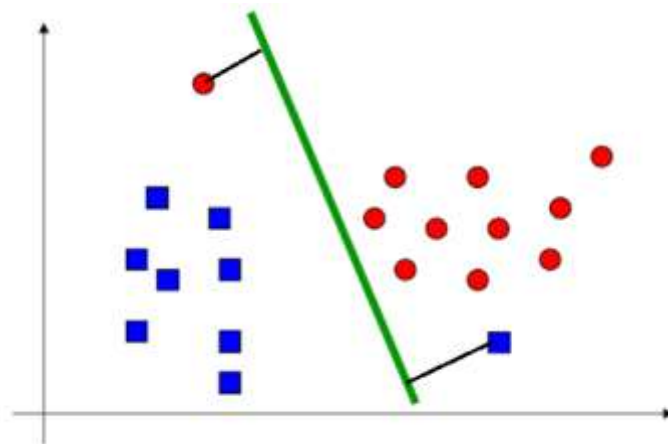
样本到超平面的距离：

正确分类样本：

$$|z| = \frac{|\mathbf{w}^T \mathbf{x}|}{\|\mathbf{w}\|} = \frac{d\mathbf{w}^T \mathbf{x}}{\|\mathbf{w}\|}$$

错误分类样本：

$$|z| = \frac{|\mathbf{w}^T \mathbf{x}|}{\|\mathbf{w}\|} = \frac{-d\mathbf{w}^T \mathbf{x}}{\|\mathbf{w}\|}$$



## 感知机优化

### Batch Perception

$$\nabla J(\mathbf{w}) = \sum_x -(d(n) - y(n)) \mathbf{x}(n)$$

$$\mathbf{w}(n+1) = \mathbf{w}(n) - \eta(n) \sum_x -(d(n) - y(n)) \mathbf{x}(n)$$

$$\nabla J(\mathbf{w}) = \sum_{\mathbf{x}(n) \in E} -\mathbf{x}(n) d(n)$$

$$\mathbf{w}(n+1) = \mathbf{w}(n) - \eta(n) \sum_{\mathbf{x}(n) \in E} -\mathbf{x}(n) d(n)$$

## 感知机优化

### Online Perception

$$\nabla J(\mathbf{w}) = -(d(n) - y(n))\mathbf{x}(n)$$

$$\mathbf{w}(n+1) = \mathbf{w}(n) - \eta(n)[-(d(n) - y(n))]\mathbf{x}(n)$$

$$\nabla J(\mathbf{w}) = -\mathbf{x}(n)d(n)_{|\mathbf{x}(n) \in E}$$

$$\mathbf{w}(n+1) = \mathbf{w}(n) - \eta(n)(-\mathbf{x}(n)d(n))_{|\mathbf{x}(n) \in E}$$

## 感知器算法流程

*Variables and Parameters:*

$\mathbf{x}(n)$  =  $(m + 1)$ -by-1 input vector  
=  $[+1, x_1(n), x_2(n), \dots, x_m(n)]^T$

$\mathbf{w}(n)$  =  $(m + 1)$ -by-1 weight vector  
=  $[b, w_1(n), w_2(n), \dots, w_m(n)]^T$

$b$  = bias

$y(n)$  = actual response (quantized)

$d(n)$  = desired response

$\eta$  = learning-rate parameter, a positive constant less than unity

1. *Initialization.* Set  $\mathbf{w}(0) = \mathbf{0}$ . Then perform the following computations for time-step  $n = 1, 2, \dots$
2. *Activation.* At time-step  $n$ , activate the perceptron by applying continuous-valued input vector  $\mathbf{x}(n)$  and desired response  $d(n)$ .
3. *Computation of Actual Response.* Compute the actual response of the perceptron as

$$y(n) = \text{sgn}[\mathbf{w}^T(n)\mathbf{x}(n)]$$

where  $\text{sgn}(\cdot)$  is the signum function.

4. *Adaptation of Weight Vector.* Update the weight vector of the perceptron to obtain

$$\mathbf{w}(n + 1) = \mathbf{w}(n) + \eta[d(n) - y(n)]\mathbf{x}(n)$$

where

$$d(n) = \begin{cases} +1 & \text{if } \mathbf{x}(n) \text{ belongs to class } \mathcal{C}_1 \\ -1 & \text{if } \mathbf{x}(n) \text{ belongs to class } \mathcal{C}_2 \end{cases}$$

5. *Continuation.* Increment time step  $n$  by one and go back to step 2.



## 误差修正基本规则

### 1. 固定增量的感知机修正

- 固定增量感知器收敛定理 (Rosenblatt, 1962)

若训练样本是线性可分，则感知器训练算法在有限次迭代后可以收敛到正确的解向量  $w$ 。

## 误差修正基本规则

### 2. 误差修正自适应规则

- 增量自适应调整

设  $\eta(n)$  满足下式:  $\eta(n)\mathbf{x}^T(n)\mathbf{x}(n) \geq |\mathbf{w}^T(n)\mathbf{x}(n)|$

对于错误分类样本来说, 上式等价于:

$$\eta(n)\mathbf{x}^T(n)\mathbf{x}(n) \geq -d(n)\mathbf{w}^T(n)\mathbf{x}(n)$$

if  $d(n)=+1$ ,  $\eta(n)\mathbf{x}^T(n)\mathbf{x}(n) \geq -\mathbf{w}^T(n)\mathbf{x}(n)$ ,  $0 \geq -\mathbf{w}^T(n)\mathbf{x}(n) - \eta(n)\mathbf{x}^T(n)\mathbf{x}(n)$

if  $d(n)=-1$ ,  $\eta(n)\mathbf{x}^T(n)\mathbf{x}(n) \geq \mathbf{w}^T(n)\mathbf{x}(n)$ ,  $0 \geq \mathbf{w}^T(n)\mathbf{x}(n) - \eta(n)\mathbf{x}^T(n)\mathbf{x}(n)$

## 误差修正基本规则

- 增量自适应调整的证明:

修正准则:  $\mathbf{w}(n+1) = \mathbf{w}(n) + \eta(n)\mathbf{x}(n)d(n)_{|\mathbf{x}(n) \in E}$

两边同乘  $-\mathbf{x}^T(n)d(n)$ , 计算损失函数 (错分代价):  $-\mathbf{x}^T(n)\mathbf{w}(n)d(n)$

$$-d(n)\mathbf{x}^T(n)\mathbf{w}(n+1) = -d(n)\mathbf{x}^T(n)\mathbf{w}(n) - \eta(n)\mathbf{x}^T(n)\mathbf{x}(n)_{|\mathbf{x} \in E}$$

当错分样本的正确标签为  $d=+1$ , 损失函数 (错分代价):

$$-\mathbf{x}^T(n)\mathbf{w}(n+1) = -\mathbf{x}^T(n)\mathbf{w}(n) \underbrace{-\eta(n)\mathbf{x}^T(n)\mathbf{x}(n)}_{<0}_{|\mathbf{x} \in E} \leq 0$$

$>0$                        $<0$

当错分样本的正确标签为  $d=-1$ , 损失函数 (错分代价):

$$\mathbf{x}^T(n)\mathbf{w}(n+1) = \mathbf{x}^T(n)\mathbf{w}(n) \underbrace{-\eta(n)\mathbf{x}^T(n)\mathbf{x}(n)}_{<0}_{|\mathbf{x} \in E} \leq 0$$

$>0$                        $<0$

- 基本规则可以保证误差变小,
- 自适应规则保证误差为 0。

## 误差修正基本规则

- 自适应修正的几何过程:

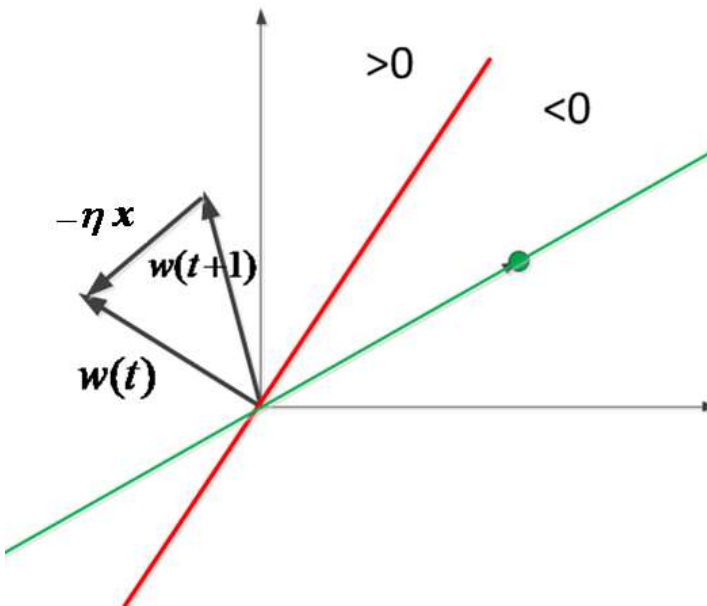
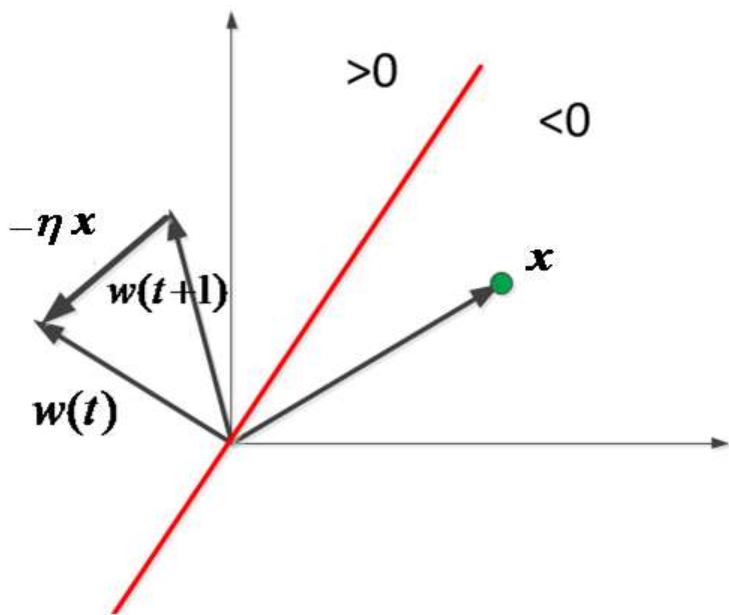
Online Perception 为例

$$\alpha \neq +1, \quad \mathbf{w}(n+1) = \mathbf{w}(n) + \eta(n) \mathbf{x}(n)_{|x \in E}$$

$$\alpha \neq -1, \quad \mathbf{w}(n+1) = \mathbf{w}(n) - \eta(n) \mathbf{x}(n)_{|x \in E}$$

## 误差修正基本规则

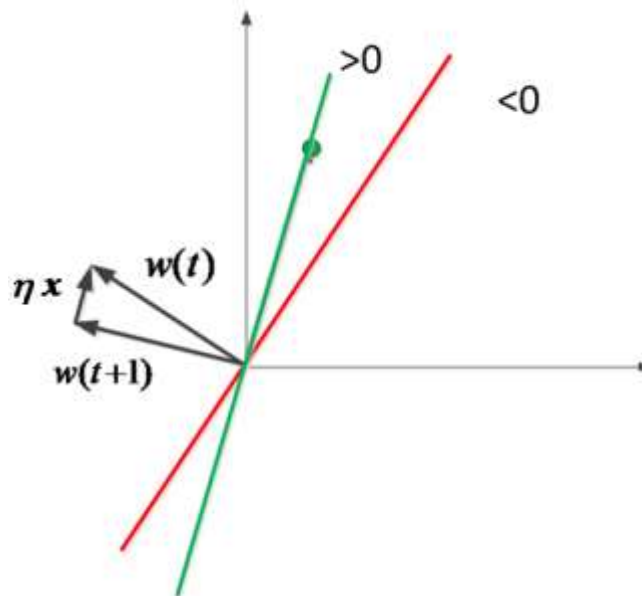
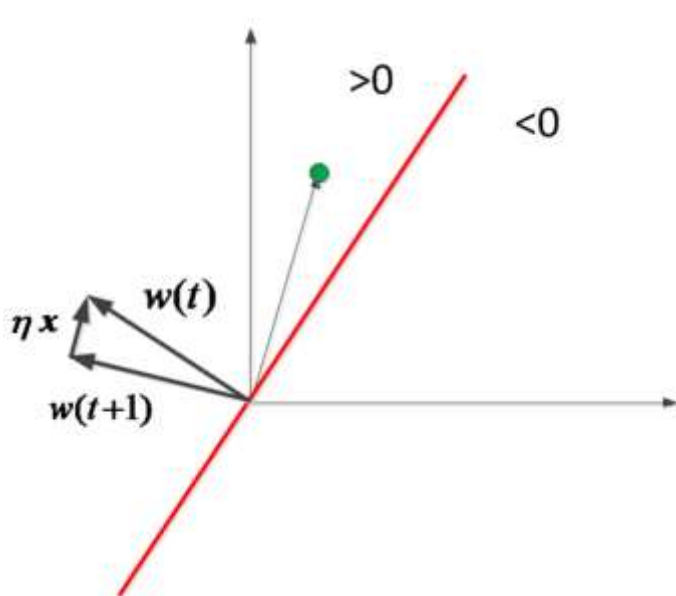
当  $d=+1$ ,  $w(n+1)=w(n)-(-\eta(n)x(n))|_{x \in E}$



修正后的分类面（绿线）

## 误差修正基本规则

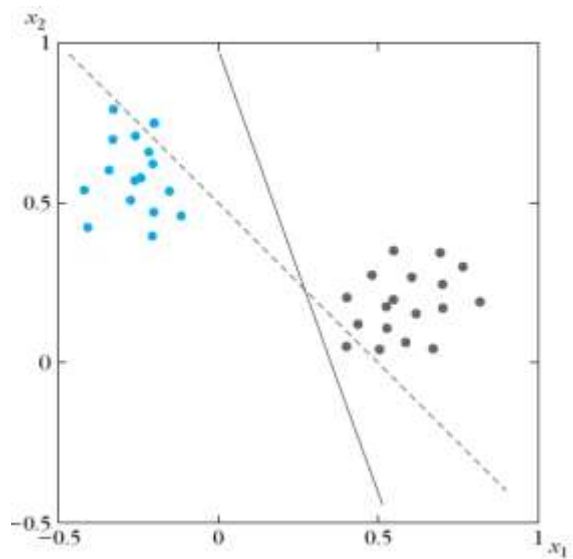
当  $d = -1$ ,  $w(n+1) = w(n) - \eta(n)x(n)_{|x \in E}$



修正后的分类面（绿线）

## 例子--1

Initial: the dashed line  $x_1 + x_2 - 0.5 = 0$



corresponding to the weight vector  $[1, 1, -0.5]^T$ ,  $\rho_t = \rho = 0.7$

## 例子--1

Optimization (GD):  $w(n+1) = w(n) - \eta(n) \sum_{x \in E} -d(n)x(n)$

all the vectors except  $[0.4, 0.05]^T$  and  $[-0.20, 0.75]^T$ .

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

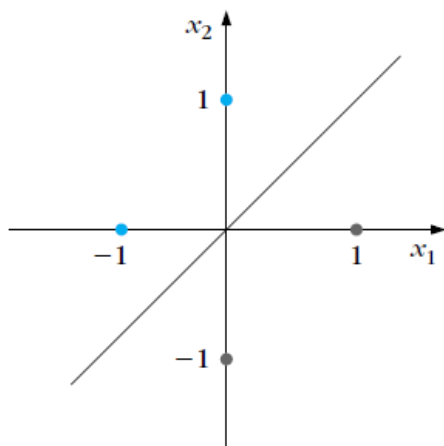
or

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

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



## 例子--2



$(-1, 0), (0, 1)$  belong to C1

$(0, -1), (1, 0)$  belong to C2

Initial:  $\mathbf{w}(0) = (0, 0, 0)^T$

The parameter  $\eta$  is set equal to one.

Data:

$(-1, 0, 1), (0, 1, 1) \in C1, d = +1, \mathbf{w}^T \mathbf{x} > 0$

$(0, -1, 1), (1, 0, 1) \in C2, d = -1, \mathbf{w}^T \mathbf{x} \leq 0$

## 例子--2

Optimization (SGD):

$$\mathbf{w}(n+1) = \mathbf{w}(n) - \eta(n)(-d(n)\mathbf{x}(n))_{|x \in E} = \mathbf{w}(n) + \eta(n)(d(n)\mathbf{x}(n))_{|x \in E}$$

Step 1.

$$\mathbf{w}^T(0) \begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix} = 0, \quad \mathbf{w}(1) = \mathbf{w}(0) + \begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix}$$

Step 2.

$$\mathbf{w}^T(1) \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix} = 1 > 0, \quad \mathbf{w}(2) = \mathbf{w}(1)$$

Step 3.

$$\mathbf{w}^T(2) \begin{bmatrix} 0 \\ -1 \\ 1 \end{bmatrix} = 1 > 0, \quad \mathbf{w}(3) = \mathbf{w}(2) - \begin{bmatrix} 0 \\ -1 \\ 1 \end{bmatrix} = \begin{bmatrix} -1 \\ 1 \\ 0 \end{bmatrix}$$

## 例子--2

Step 4.

$$w^T(3) \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix} = -1 < 0, \quad w(4) = w(3)$$

Step 5.

$$w^T(4) \begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix} = 1 > 0, \quad w(5) = w(4)$$

Step 6.

$$w^T(5) \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix} = 1 > 0, \quad w(6) = w(5)$$

Step 7.

$$w^T(6) \begin{bmatrix} 0 \\ -1 \\ 1 \end{bmatrix} = -1 < 0, \quad w(7) = w(6)$$

*Have a beak !*

# 第三章 线性分类

3.1 概述

3.2 基础知识

3.3 感知机

**3.4 线性鉴别分析**

3.5 logistic 模型

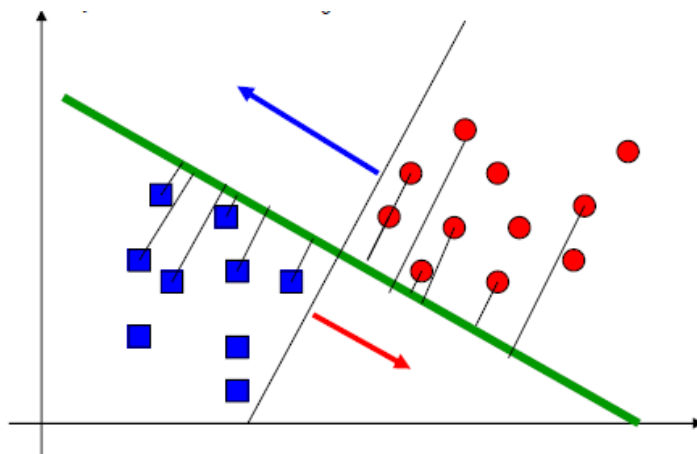
# 线性鉴别分析

## 基本思想

求线性变换

$$y = w^T x$$

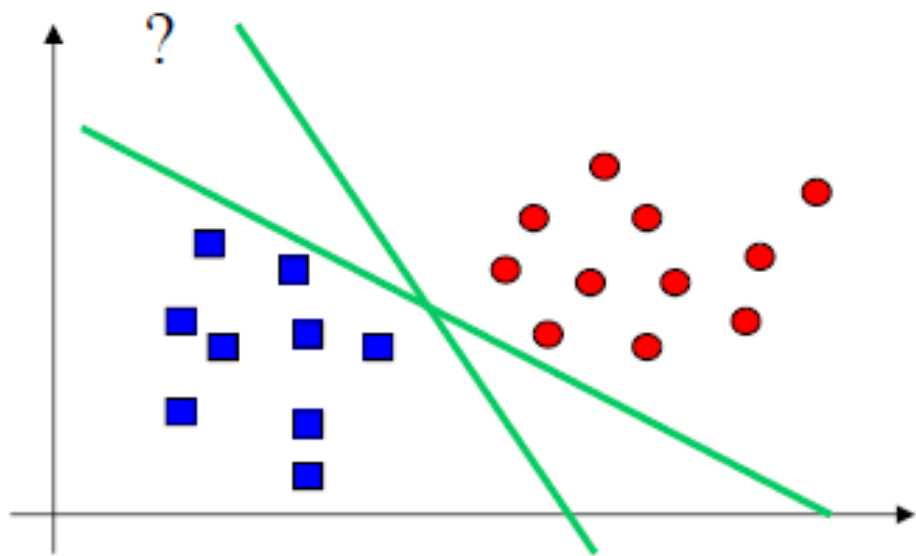
使得样本集 $\{x_i\}$ 线性变换成一维变量 $\{y_i\}$ 后，类别间距大，类内间距小，



# 线性鉴别分析

## 基本思想

怎么找到这个方向?

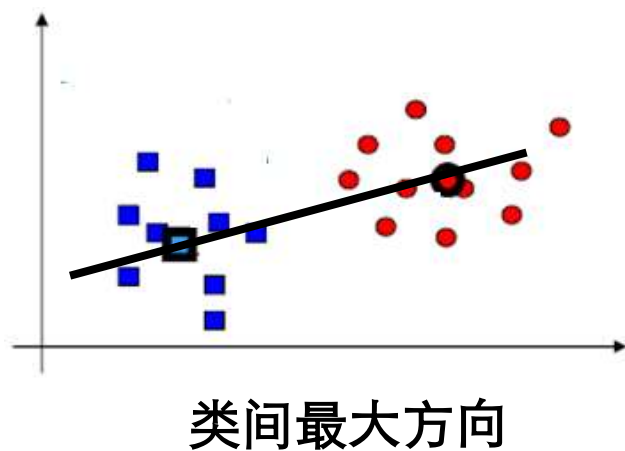


# 线性鉴别分析

## 基本思想

假设： 如果用各类的均值代表类别， **类别间最大的方向**

$$\mathbf{u}_1 = \frac{1}{N_1} \sum_{i \in C_1} \mathbf{x}_i, \quad \mathbf{u}_2 = \frac{1}{N_2} \sum_{i \in C_2} \mathbf{x}_i$$

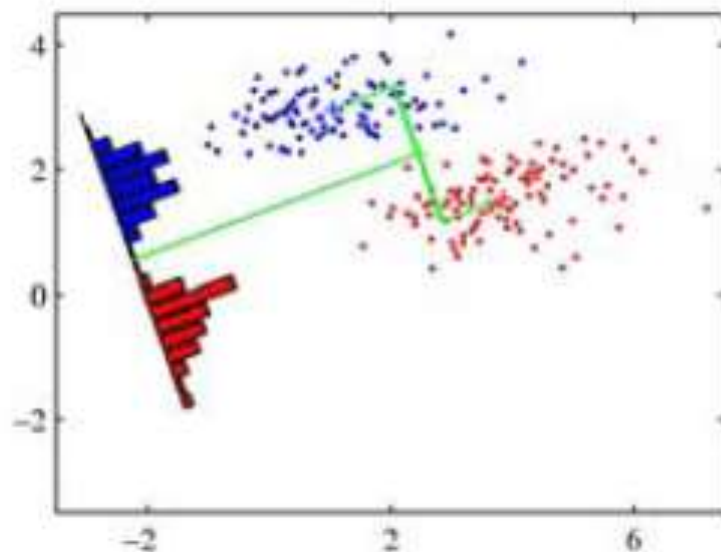
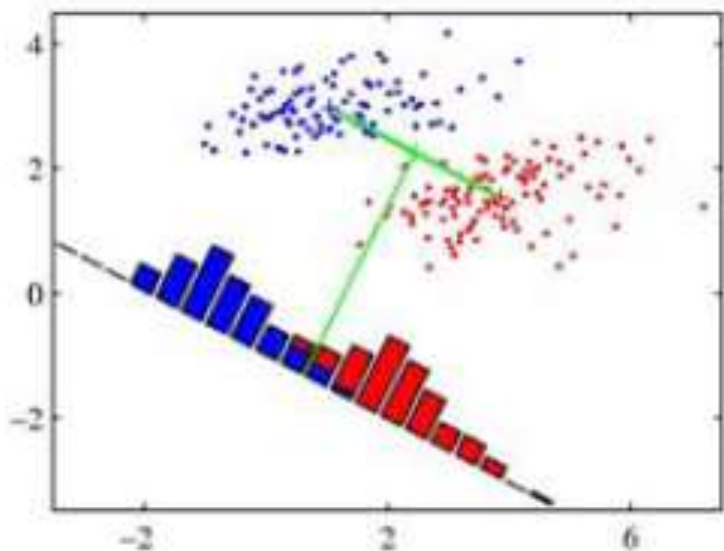




# 线性鉴别分析

## 基本思想

问题：只考虑类间，有可能线性不可分



# 线性鉴别分析

## 目标函数 ( Fisher Criterion )

$$\max J(\mathbf{w}) = \frac{(\mathbf{m}_1 - \mathbf{m}_2)^2}{S_1^2 + S_2^2}$$

$$J = \frac{\text{类别间距}}{\text{类内的平均距离}}$$

# 线性鉴别分析

## 类别间距离

样本投影后的类别间距离： $(m_1 - m_2)^2$ ；其中， $m_i$ 表示第  $i$  类样本投影后的均值。

第  $k$  类样本平均值（类心）：

$$u_k = \frac{1}{|C_k|} \sum_{x_i \in C_k} x_i$$

# 线性鉴别分析

## 类别间距离

样本投影后的类别间距离： $(m_1 - m_2)^2$ ；其中， $m_i$ 表示第  $i$  类样本投影后的均值。

第  $k$  类样本平均值（类心）：

$$u_k = \frac{1}{|C_k|} \sum_{x_i \in C_k} x_i$$

两个类别的类心：

$$u_1 = \frac{1}{|C_1|} \sum_{x_i \in C_1} x_i, \quad u_2 = \frac{1}{|C_2|} \sum_{x_i \in C_2} x_i$$

# 线性鉴别分析

## 类别间距离

样本投影后的类别间距离： $(m_1 - m_2)^2$ ；其中， $m_i$ 表示第 $i$ 类样本投影后的均值。

样本  $x_i$  投影到  $w$  方向后，为  $y_i$ ： $y_i = w^T x_i$

投影后的类心：

$$\begin{aligned} m_k &= \frac{1}{|C_k|} \sum_{x_i \in C_k} y_i \\ &= \frac{1}{|C_k|} \sum_{x_i \in C_k} w^T x_i \\ &= w^T \left( \frac{1}{|C_k|} \sum_{x_i \in C_k} x_i \right) \\ &= w^T u_k \end{aligned}$$

# 线性鉴别分析

## 类别间距离

样本投影后的类别间距离： $(m_1 - m_2)^2$ ；其中， $m_i$ 表示第 $i$ 类样本投影后的均值。

投影后两类的类心：

$$m_1 = \mathbf{w}^T \mathbf{u}_1, \quad m_2 = \mathbf{w}^T \mathbf{u}_2$$

$\mathbf{w}$  方向投影后，类间距： $m_1 - m_2 = \mathbf{w}^T (\mathbf{u}_1 - \mathbf{u}_2)$

$$\begin{aligned} (m_1 - m_2)^2 &= (m_1 - m_2)(m_1 - m_2)^T \\ &= \mathbf{w}^T (\mathbf{u}_1 - \mathbf{u}_2)(\mathbf{u}_1 - \mathbf{u}_2)^T \mathbf{w} \\ &= \mathbf{w}^T S_b \mathbf{w} \end{aligned}$$

其中，类间散度矩阵： $S_b = (\mathbf{u}_1 - \mathbf{u}_2)(\mathbf{u}_1 - \mathbf{u}_2)^T$

若考虑先验可以定义： $S_b = p(\omega_1)p(\omega_2)(\mathbf{u}_1 - \mathbf{u}_2)(\mathbf{u}_1 - \mathbf{u}_2)^T$

# 线性鉴别分析

## 类别内的距离

样本投影后的类别内距离：投影后的各类样本方差  $S_1^2 + S_2^2$

样本均方差（类别内松散程度）

$$\sigma_k^2 = \sum_{x_i \in C_k} (x_i - u_k)^2 = \sum_{x_i \in C_k} \tilde{x}_i^2$$

两类的均方差：

$$\sigma_1^2 = \sum_{x_i \in C_1} (x_i - u_1)^2 = \sum_{x_i \in C_1} \tilde{x}_i^2,$$

$$\sigma_2^2 = \sum_{x_i \in C_2} (x_i - u_2)^2 = \sum_{x_i \in C_2} \tilde{x}_i^2$$

# 线性鉴别分析

## 类别内的距离

样本投影后的类别内距离：投影后的各类样本方差  $S_1^2 + S_2^2$

样本  $x_i$  投影到  $w$  方向后为  $y_i$ ：  $y_i = w^T x_i$

在投影方向  $w$  上，第  $k$  类别内，样本距离

$$\begin{aligned} S_k^2 &= \sum_{x_i \in C_k} (y_i - m_k)^2 = \sum_{x_i \in C_k} (w^T (x_i - u_k))^2 = \sum_{x_i \in C_k} (w^T \tilde{x}_i)^2 \\ &= \sum_{x_i \in C_k} (w^T \tilde{x}_i) (w^T \tilde{x}_i)^T = \sum_{x_i \in C_k} w^T \tilde{x}_i \tilde{x}_i^T w = w^T \left( \sum_{x_i \in C_k} \tilde{x}_i \tilde{x}_i^T \right) w \\ &= w^T (X_k X_k^T) w \end{aligned}$$

其中，  $X_k = [\tilde{x}_1, \tilde{x}_2, \dots, \tilde{x}_i, \dots]_{C_k}$  第  $k$  类样本矩阵（ $\tilde{x}_i$  是列向量）



# 线性鉴别分析

## 类别内的距离

**样本投影后的类别内距离：投影后的各类样本方差**  $S_1^2 + S_2^2$

在投影方向  $w$  上，类别内距离

$$\begin{aligned} S_1^2 + S_2^2 &= w^T (X_1 X_1^T) w + w^T (X_2 X_2^T) w \\ &= w^T (X_1 X_1^T + X_2 X_2^T) w \\ &= w^T S_w w \end{aligned}$$

其中，类内散度矩阵：

$$S_w = X_1 X_1^T + X_2 X_2^T$$

**若考虑先验可以定义：**  $S_w = p(\omega_1) X_1 X_1^T + p(\omega_2) X_2 X_2^T$

# 线性鉴别分析

## 求解过程

$$\max J(\mathbf{w}) = \frac{(\mathbf{m}_1 - \mathbf{m}_2)^2}{S_1^2 + S_2^2} = \frac{\mathbf{w}^T \mathbf{S}_b \mathbf{w}}{\mathbf{w}^T \mathbf{S}_w \mathbf{w}}$$

- 广义的 Rayleigh 商，可用 Lagrange 乘子求解， 假设：  $\mathbf{w}^T \mathbf{S}_w \mathbf{w} = c$

$$L(\mathbf{w}, \lambda) = \mathbf{w}^T \mathbf{S}_b \mathbf{w} - \lambda(\mathbf{w}^T \mathbf{S}_w \mathbf{w} - c)$$

$$\frac{\partial L(\mathbf{w}, \lambda)}{\partial \mathbf{w}} = 2\mathbf{S}_b \mathbf{w} - 2\lambda \mathbf{S}_w \mathbf{w} = 0$$

$$\mathbf{S}_w^{-1} \mathbf{S}_b \mathbf{w} = \lambda \mathbf{w}$$

- 最优解  $\mathbf{w}$  是  $\mathbf{S}_w^{-1} \mathbf{S}_b$  的特征向量

# 线性鉴别分析

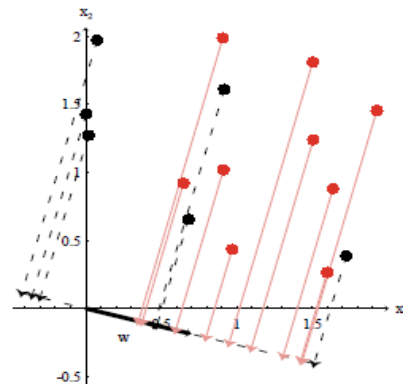
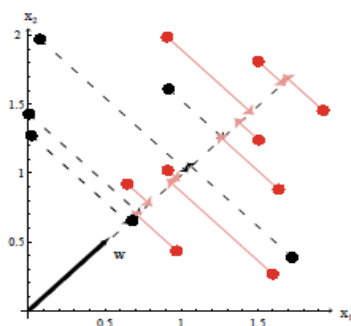
## 求解过程

实际并没有求特征值，因为  $S_b \mathbf{w}$  在  $u_1 - u_2$  方向上

$$S_b \mathbf{w} = (u_1 - u_2)(u_1 - u_2)^T \mathbf{w} = \beta(u_1 - u_2)$$

$$S_w^{-1} S_b \mathbf{w} = \lambda \mathbf{w} \quad \Rightarrow \quad S_w^{-1} \beta(u_1 - u_2) = \lambda \mathbf{w}$$

$$\mathbf{w} = S_w^{-1} (u_1 - u_2)$$



*Have a break!*

# 第三章 线性分类

3.1 概述

3.2 基础知识

3.3 感知机

3.4 线性鉴别分析

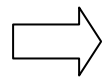
**3.5 logistic 模型**

# logistic 模型

## 基本思想

假设 likelihood ratio 的对数为线性判别函数

$$\log\left(\frac{p(\mathbf{x}|\omega_i)}{p(\mathbf{x}|\omega_M)}\right) = \beta_{i,0} + \boldsymbol{\beta}_i^T \mathbf{x}, \quad i=1,2,\dots,M-1$$



$$\log\left(\frac{p(\omega_i|\mathbf{x})}{p(\omega_M|\mathbf{x})}\right) = w_{i,0} + \mathbf{w}_i^T \mathbf{x}, \quad i=1,2,\dots,M-1$$

# logistic 模型

## 基本思想

### 多类问题

$$\ln\left(\frac{p(\omega_i | \mathbf{x})}{p(\omega_M | \mathbf{x})}\right) = w_{i,0} + \mathbf{w}_i^T \mathbf{x}, \quad i=1, \dots, M-1$$

$$\sum_{i=1}^M p(\omega_i | \mathbf{x}) = 1$$

$$\Rightarrow \begin{cases} p(\omega_M | \mathbf{x}) = \frac{1}{1 + \sum_{i=1}^{M-1} \exp(w_{i,0} + \mathbf{w}_i^T \mathbf{x})} & (1) \\ p(\omega_i | \mathbf{x}) = \frac{\exp(w_{i,0} + \mathbf{w}_i^T \mathbf{x})}{1 + \sum_{i=1}^{M-1} \exp(w_{i,0} + \mathbf{w}_i^T \mathbf{x})}, i=1, \dots, M-1 & (2) \end{cases}$$

# logistic 模型

## 基本思想

两类问题:

$$\begin{cases} p(\omega_2 | \mathbf{x}) = \frac{1}{1 + \exp(w_0 + \mathbf{w}^T \mathbf{x})} \\ p(\omega_1 | \mathbf{x}) = \frac{\exp(w_0 + \mathbf{w}^T \mathbf{x})}{1 + \exp(w_0 + \mathbf{w}^T \mathbf{x})} = \frac{1}{1 + \exp(-(w_0 + \mathbf{w}^T \mathbf{x}))} \end{cases}$$

令  $v = \mathbf{w}^T \mathbf{x} + w_0$ , 则

$$\begin{cases} p(\omega_2 | \mathbf{x}) = \frac{1}{1 + \exp(v)} \\ p(\omega_1 | \mathbf{x}) = \frac{1}{1 + \exp(-v)} \end{cases}$$



# logistic 模型

## 基本思想

两类问题:

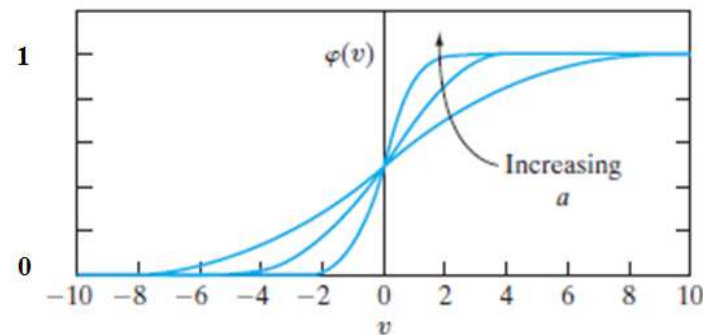
$$\begin{cases} p(\omega_2 | \mathbf{x}) = \frac{1}{1 + \exp(w_0 + \mathbf{w}^T \mathbf{x})} \\ p(\omega_1 | \mathbf{x}) = \frac{\exp(w_0 + \mathbf{w}^T \mathbf{x})}{1 + \exp(w_0 + \mathbf{w}^T \mathbf{x})} = \frac{1}{1 + \exp(-(w_0 + \mathbf{w}^T \mathbf{x}))} \end{cases}$$

令  $v = \mathbf{w}^T \mathbf{x} + w_0$ , 则

$$\begin{cases} p(\omega_2 | \mathbf{x}) = \frac{1}{1 + \exp(v)} \\ p(\omega_1 | \mathbf{x}) = \frac{1}{1 + \exp(-v)} \end{cases}$$

Logistic 函数 (Sigmoid 函数)

$$\varphi(v) = \frac{1}{1 + \exp(-av)}$$



# logistic 模型

## 基本思想

两类问题:

$$\begin{cases} p(\omega_2 | \mathbf{x}) = \frac{1}{1 + \exp(w_0 + \mathbf{w}^T \mathbf{x})} \\ p(\omega_1 | \mathbf{x}) = \frac{\exp(w_0 + \mathbf{w}^T \mathbf{x})}{1 + \exp(w_0 + \mathbf{w}^T \mathbf{x})} = \frac{1}{1 + \exp(-(w_0 + \mathbf{w}^T \mathbf{x}))} \end{cases}$$

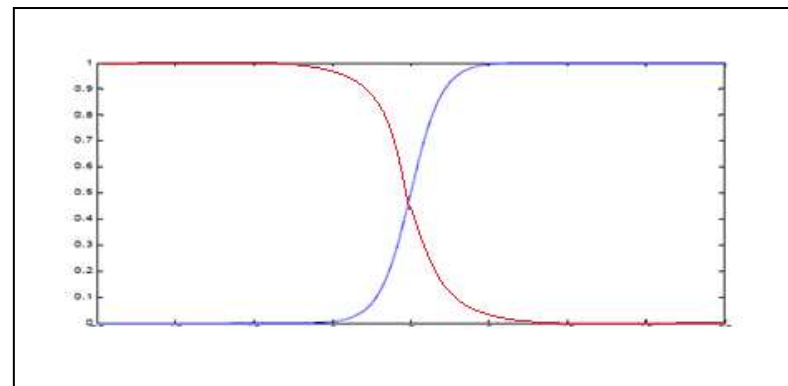
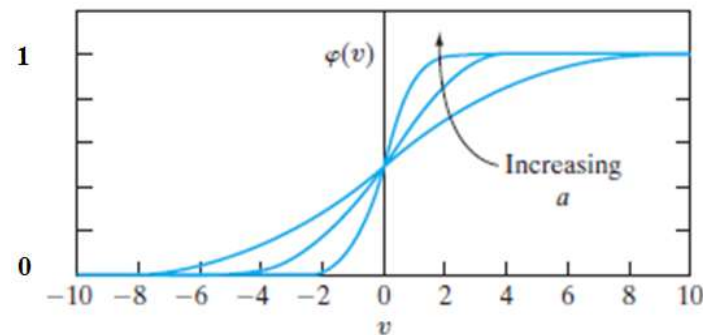
令  $v = \mathbf{w}^T \mathbf{x} + w_0$ , 则

$$\begin{cases} p(\omega_2 | \mathbf{x}) = \frac{1}{1 + \exp(v)} \\ p(\omega_1 | \mathbf{x}) = \frac{1}{1 + \exp(-v)} \end{cases}$$

是两个对称函数

Logistic 函数 (Sigmoid 函数)

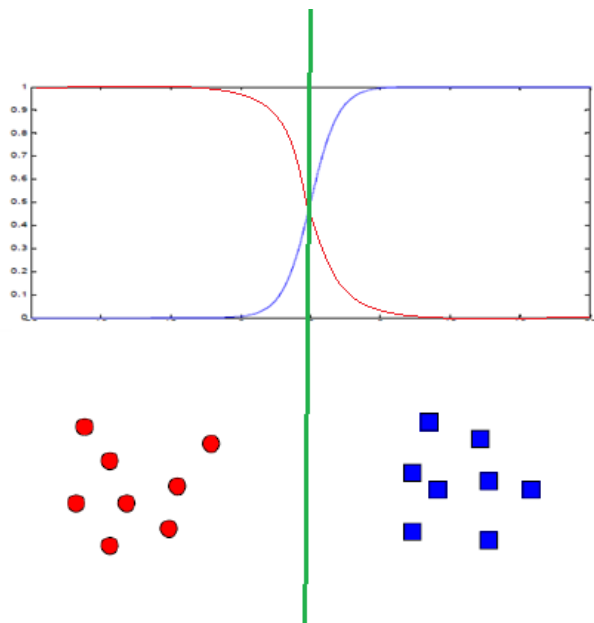
$$\varphi(v) = \frac{1}{1 + \exp(-av)}$$



# logistic 模型

## 基本思想

求参数  $w$  和  $w_0$ ，相当于确定了一个线性判别函数  $g(x) = w^T x + w_0$



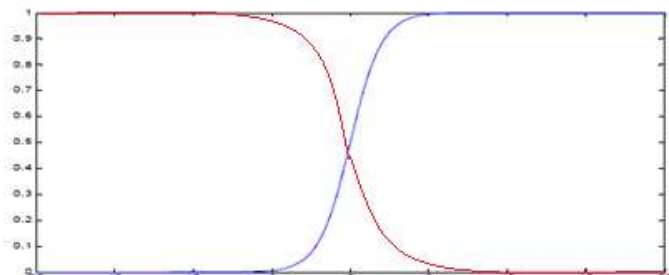
$$\begin{cases} p(\omega_2 | \mathbf{x}) = \frac{1}{1 + \exp(w_0 + \mathbf{w}^T \mathbf{x})} \\ p(\omega_1 | \mathbf{x}) = \frac{\exp(w_0 + \mathbf{w}^T \mathbf{x})}{1 + \exp(w_0 + \mathbf{w}^T \mathbf{x})} = \frac{1}{1 + \exp(-(w_0 + \mathbf{w}^T \mathbf{x}))} \end{cases}$$

# logistic 模型

## 学习过程

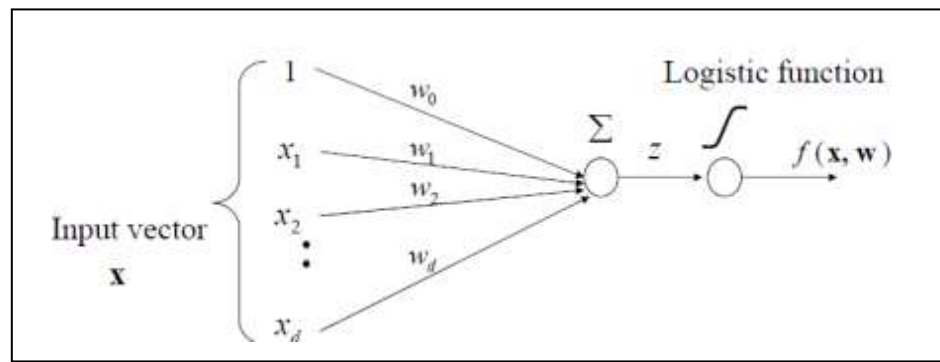
### 学习目标：

标签  $\omega_1$  类,  $p(\omega_1|\mathbf{x})$  越大,  $p(\omega_2|\mathbf{x})$  越小,  
标签  $\omega_2$  类,  $p(\omega_2|\mathbf{x})$  越大,  $p(\omega_1|\mathbf{x})$  越小,



等价于

标签  $\omega_1$  类,  $p(\omega_1|\mathbf{x})$  越大,  $1-p(\omega_1|\mathbf{x})$  越小,  
标签  $\omega_2$  类,  $1-p(\omega_1|\mathbf{x})$  越大,  $p(\omega_1|\mathbf{x})$  越小,



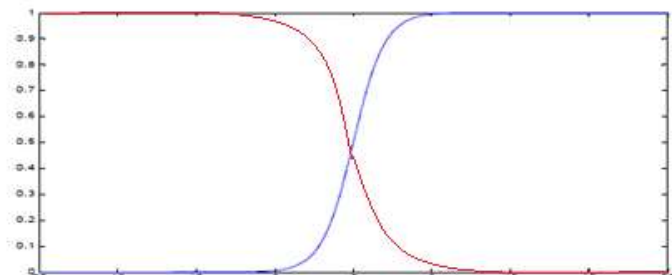
# logistic 模型

## 学习过程

优化准则：

标签  $\omega_1$  类， $p(\omega_1|\mathbf{x})$  越大

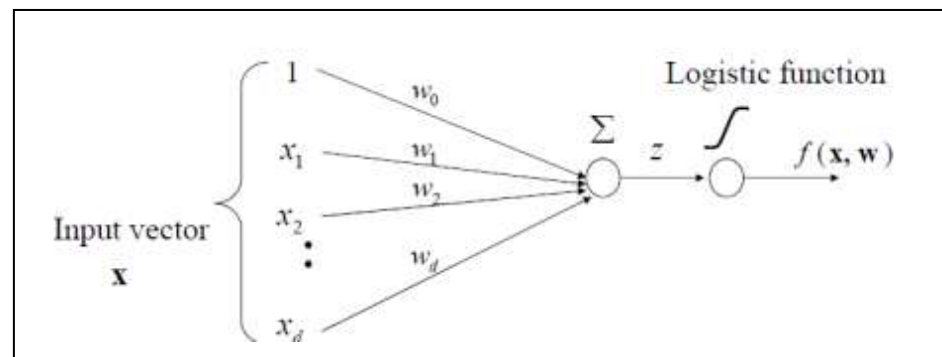
标签  $\omega_2$  类， $p(\omega_2|\mathbf{x})$  越大



等价于

标签  $\omega_1$  类， $p(\omega_1|\mathbf{x})$  越大

标签  $\omega_2$  类， $1-p(\omega_1|\mathbf{x})$  越大



# logistic 模型

## 学习过程

最大似然求取参数  $\theta = \{w_i, w_{i,0}\}_{i=1,\dots,M-1}$

$$L(\theta) = \ln \left\{ \prod_{k=1}^{N_1} p(\mathbf{x}_k^{(1)} | \omega_1; \theta) \prod_{k=1}^{N_2} p(\mathbf{x}_k^{(2)} | \omega_2; \theta) \dots \prod_{k=1}^{N_M} p(\mathbf{x}_k^{(M)} | \omega_M; \theta) \right\}$$

$$p(\mathbf{x}_k^{(m)} | \omega_m; \theta) = \frac{p(\mathbf{x}_k^{(m)})P(\omega_m | \mathbf{x}_k^{(m)}; \theta)}{P(\omega_m)}$$

将(1)(2)带入后

$$L(\theta) = \sum_{k=1}^{N_1} \ln P(\omega_1 | \mathbf{x}_k^{(1)}) + \sum_{k=1}^{N_2} \ln P(\omega_2 | \mathbf{x}_k^{(2)}) + \dots + \sum_{k=1}^{N_M} \ln P(\omega_M | \mathbf{x}_k^{(M)}) + C$$

$$C = \ln \frac{\prod_{k=1}^N p(\mathbf{x}_k)}{\prod_{m=1}^M P(\omega_m)^{N_m}}$$

忽略先验的最大后验估计，就是最大似然估计

# logistic 模型

## 学习过程

最大  $L(\theta)$  问题转化为最小  $-L(\theta)$

求得  $\nabla L(\theta) = \frac{-\partial L(\theta)}{\partial \theta}$ ，采用梯度下降方法，

求解  $\theta = \{w_i, w_{i,0}\}_{i=1, \dots, M-1}$ ；  $m$  类与其他  $m-1$  类别的线性决策函数。

*Have a break!*



# logistic 模型

## 模型理解

两类问题,

**Discriminant functions:**

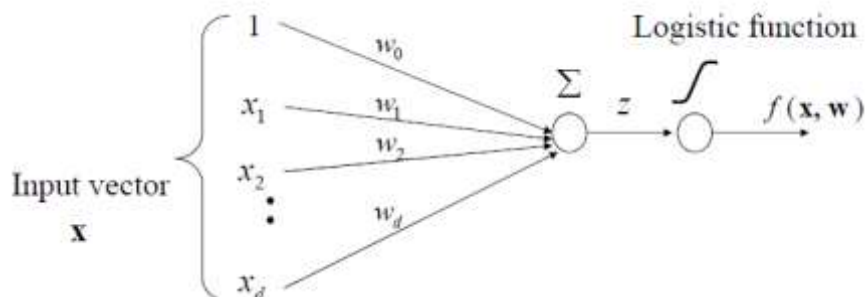
$$g_1(\mathbf{x}) = g(\mathbf{w}^T \mathbf{x})$$

$$g_0(\mathbf{x}) = 1 - g(\mathbf{w}^T \mathbf{x})$$

**Sigmoid function:**

$$g(z) = 1 / (1 + e^{-z})$$

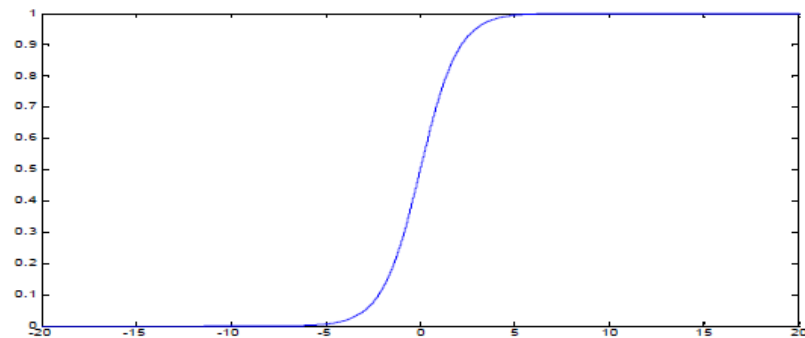
单层神经元、Logistic 激活函数



**function**

$$g(z) = \frac{1}{(1 + e^{-z})}$$

- Is also referred to as a **sigmoid function**
- Replaces the threshold function with smooth switching
- takes a real number and outputs the number in the interval  $[0,1]$



# logistic 模型

## 模型理解

### – Probabilistic interpretation

$$f(\mathbf{x}, \mathbf{w}) = p(y = 1 \mid \mathbf{w}, \mathbf{x}) = g_1(\mathbf{x}) = g(\mathbf{w}^T \mathbf{x})$$

$$p(y = 0 \mid \mathbf{x}, \mathbf{w}) = 1 - p(y = 1 \mid \mathbf{x}, \mathbf{w})$$

**Decision boundary:**  $g_1(\mathbf{x}) = g_0(\mathbf{x})$

the boundary it must hold:

$$\log \frac{g_0(\mathbf{x})}{g_1(\mathbf{x})} = \log \frac{1 - g(\mathbf{w}^T \mathbf{x})}{g(\mathbf{w}^T \mathbf{x})} = 0$$

# logistic 模型

## 模型理解

线性决策界:

$$\log \frac{g_o(\mathbf{x})}{g_1(\mathbf{x})} = \log \frac{\frac{\exp-(\mathbf{w}^T \mathbf{x})}{1 + \exp-(\mathbf{w}^T \mathbf{x})}}{1} = \log \exp-(\mathbf{w}^T \mathbf{x}) = \mathbf{w}^T \mathbf{x} = 0$$

# logistic 模型

## 模型优化

$$p(y = 1 \mid \mathbf{x}) = \frac{e^{\mathbf{w}^T \mathbf{x} + b}}{1 + e^{\mathbf{w}^T \mathbf{x} + b}} = \frac{1}{1 + e^{-(\mathbf{w}^T \mathbf{x} + b)}}$$

$$p(y = 0 \mid \mathbf{x}) = \frac{1}{1 + e^{\mathbf{w}^T \mathbf{x} + b}} = 1 - p(y = 1 \mid \mathbf{x})$$

$$p(y_i \mid \mathbf{x}_i; \mathbf{w}, b) = y_i p_1(\hat{\mathbf{x}}_i; \boldsymbol{\beta}) + (1 - y_i) p_0(\hat{\mathbf{x}}_i; \boldsymbol{\beta})$$

或者 
$$p(y_i \mid \mathbf{x}_i; \mathbf{w}, b) = y_i p_1(\hat{\mathbf{x}}_i; \boldsymbol{\beta}) + (1 - y_i) (1 - p_1(\hat{\mathbf{x}}_i; \boldsymbol{\beta}))$$

**最大化的似然估计：**

$$\ell(\mathbf{w}, b) = \sum_{i=1}^m \ln p(y_i \mid \mathbf{x}_i; \mathbf{w}, b)$$

# 小结

## 1. 掌握基础知识:

线性模型的基本表达、向量相似计算、常用的统计量;

## 2. 重点掌握线性分类模型: 感知器、线性鉴别;

了解logistic鉴别;

## 3. 掌握随机梯度下降优化方法.

## 参考文献

1. Pattern Recognition 2nd. 《模式识别》(第二版), 边肇祺, 张学工等, 清华大学出版社, 2000.1。
2. Pattern Classification, 2nd. 模式分类, 第二版。
3. 周志华, 机器学习, 清华大学出版社, 2016.