模式识别

第4讲 判别函数分类法（II）
（线性判别算法、非线性判别函数）

2018~2019学年

内容安排

一、绪论、数学基础（第1讲）

二、聚类分析（第2讲）

三、判别函数分类法（几何分类法）（第3、4讲）

四、统计决策分类法（概率分类法）（第5、6讲）

五、特征提取与选择（第7讲）

六、模糊模式识别（第8讲）

七、神经网络模式识别（第9讲）

期末考试（平时作业：40%，期末考试：60%）

三、判别函数分类法

3.1 判别函数

3.2 线性判别函数

3.3 广义线性判别函数

3.4 线性判别函数的几何性质

3.5 Fisher线性判别

3.6 感知器算法

3.7 梯度算法

3.8 最小平方误差算法

3.9 非线性判别函数

3.6 感知器算法

对线性判别函数，当模式维数已知时，判别函数的形式实际上已经确定，如：三维时

$$d(\boldsymbol{X})=w_1x_1+w_2x_2+w_3x_3+w_4=\boldsymbol{W}^{\mathrm{T}}\boldsymbol{X}$$

$$\boldsymbol{X}=[x_1,x_2,x_3,1]^{\mathrm{T}}\quad \boldsymbol{W}=[w_1,w_2,w_3,w_4]^{\mathrm{T}}$$

只要求出权向量，分类器的设计即告完成。本节开始介绍如何通过各种算法，利用已知类别的模式样本**训练权向量*W***。

1. 概念理解

1）训练与学习

训练：用已知类别的模式样本指导机器对分类规则进行反复修改，最终使分类结果与已知类别信息完全相同的过程。

学习：从分类器的角度讲 $\left\{\begin{array}{l}\text{非监督学习}\\ \text{有监督学习} \longleftrightarrow \text{训练}\end{array}\right.$

2）确定性分类器

处理确定可分情况的分类器。通过几何方法将特征空间分解为对应不同类的子空间，又称为**几何分类器**。

3）**感知器（Perceptron）**

一种早期神经网络分类学习模型，属于有关机器学习的仿生学领域中的问题，由于**无法实现非线性分类**而下马（Minsky and Papert）。但“赏罚概念（reward-punishment）”得到广泛应用。

2. 感知器算法

两类线性可分的模式类：ω_1, ω_2，设 $d(\boldsymbol{X}) = \boldsymbol{W}^{\mathrm{T}}\boldsymbol{X}$

其中，$\boldsymbol{W} = [w_1, w_2, w_3, w_4]^{\mathrm{T}}$，$\boldsymbol{X} = [x_1, x_2, \cdots, x_n, 1]^{\mathrm{T}}$

应具有性质 $$d(\boldsymbol{X}) = \boldsymbol{W}^{\mathrm{T}}\boldsymbol{X} \begin{cases} > 0, & 若\ \boldsymbol{X} \in \omega_1 \\ < 0, & 若\ \boldsymbol{X} \in \omega_2 \end{cases}$$

对样本进行规范化处理，即ω_2类样本全部乘以(－1)，则有：

$$d(\boldsymbol{X}) = \boldsymbol{W}^{\mathrm{T}}\boldsymbol{X} > 0$$

感知器算法的基本思想：用训练模式验证当前权向量的合理性，如果不合理，就根据误差进行反向纠正，直到全部训练样本都被合理分类。本质上是梯度下降方法类。

感知器算法步骤：

（1）选择N个分属于ω_1和 ω_2类的模式样本构成训练样本集

$$\{\boldsymbol{X}_1, \ \ldots, \boldsymbol{X}_N\}$$

构成增广向量形式，并进行符号规范化处理。

任取权向量初始值$\boldsymbol{W}(1)$，开始迭代。初始迭代次数k=1 。

（2）用全部训练样本进行一轮迭代，计算$\boldsymbol{W}^{\mathrm{T}}(k)\boldsymbol{X}_i$的值，并修正权向量。

分两种情况，更新权向量的值：

① 若$\boldsymbol{W}^{\mathrm{T}}(k)\boldsymbol{X}_i \le 0$，说明分类器对第$i$个模式做了错误分类，

权向量校正为：$\boldsymbol{W}(k+1)=\boldsymbol{W}(k)+c\boldsymbol{X}_i$

c： 正的校正增量（步长）。

② 若$\boldsymbol{W}^{\mathrm{T}}(k)\boldsymbol{X}_i > 0$，分类正确，权向量不变：

$$\boldsymbol{W}(k+1)=\boldsymbol{W}(k)$$

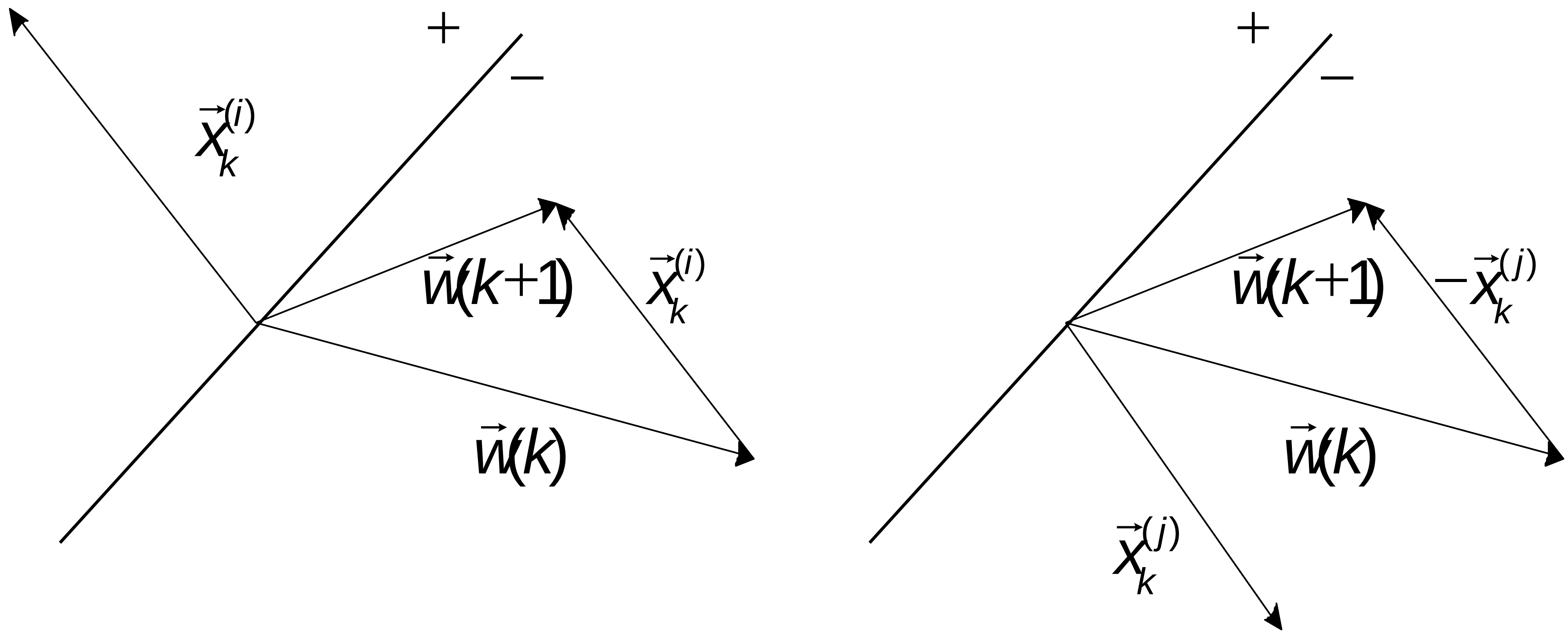
统一写为：

$$\boldsymbol{W}(k+1)=\begin{cases}\boldsymbol{W}(k) & \text{若}\boldsymbol{W}^{\mathrm{T}}(k)\boldsymbol{X}_i > 0\\ \boldsymbol{W}(k)+c\boldsymbol{X}_i & \text{若}\boldsymbol{W}^{\mathrm{T}}(k)\boldsymbol{X}_i \le 0\end{cases}$$

（3）分析分类结果：只要有一个错误分类，回到（2），直至对所有样本正确分类。

感知器算法是一种赏罚过程：

- 分类正确时对权向量“赏”——不罚，即权向量不变；
- 分类错误时对权向量“罚”——修改，向正确的方向转换。



权空间中感知器算法权矢量校正过程示意图

注意：右边的$x_k^{(j)}$是未进行符号规范化的属于j–类矢量，符号规范化后，与权矢量的夹角大于90°，内积为负，也需要调整权向量！左边的$x_k^{(i)}$ 是i–类矢量，也需要校正权矢量。

3. 感知器算法的收敛性

收敛性：经过算法的有限次迭代运算后，求出了一个使所有样本都能正确分类的W，则称算法是收敛的。

可以证明：**模式类线性可分时，感知器算法是收敛的。**

感知器算法收敛定理

如果训练模式是线性可分的，感知器训练算法在有限次迭代后便可以收敛到正确的解矢量 $\vec{w}^*$。

证明思路：

如果第k+1次迭代生成的权矢量比第k次迭代生成的权矢量更接近解矢量，则收敛，即：

$$\left\|\vec{w}^* - \vec{w}(k)\right\|^2 - \left\|\vec{w}^* - \vec{w}(k+1)\right\|^2 > 0$$

定理 3.5.1(收敛定理) 如果训练模式是线性可分的,感知器训练算法在有限次迭代后便可以收敛到正确分类的解矢量 w^*。

证明 为证明方便又不影响结论的正确性,因改变 w 的尺度并不影响判别结果,不妨设 $\rho=1$, $x\in\omega_2$已乘以 -1,并设在感知器训练过程中,错判的模式为 $x_{i_1}, x_{i_2}, \cdots, x_{i_k}, \cdots$,其中 $x_{i_j}\in\{x_i\}$,第 k 步权矢量 $w(k)$等于初始权矢量 $w(1)$用 $x_{i_j}(j=1,2,\cdots,k)$校正,按感知器算法,有

$$w(k+1)=w(k)+x_{i_k} \tag{3-5-1}$$

由于训练模式是线性可分的,所以存在使式(3-3-9)成立的解矢量,设为 w^*。令

$$a=\max_i \|x_i\|^2$$

$$b=\min_i[x_i^{\mathrm{T}}w^*]>0$$

$$c>0$$

由式(3-5-1)可得

$$w(k+1)-cw^*=[w(k)-cw^*]+x_{i_k} \tag{3-5-2}$$

上式两边取模方,有

$$\|w(k+1)-cw^*\|^2=\|w(k)-cw^*\|^2+\|x_{i_k}\|^2+2[w(k)-cw^*]^{\mathrm{T}}x_{i_k}$$

由于 x_{i_k}被错分,故有 $w^{\mathrm{T}}(k)x_{i_k}\leqslant 0$,从而得

$$\|w(k+1)-cw^*\|^2\leqslant\|w(k)-cw^*\|^2+\|x_{i_k}\|^2-2cx_{i_k}^{\mathrm{T}}w^*$$

因 $x_{i_k}^{\mathrm{T}}w^*>0$,并由 a、b 的设定,可有

$$\|w(k+1)-cw^*\|^2\leqslant\|w(k)-cw^*\|^2-2cb+a$$

取

$$c = a / b$$

可得

$$\| w(k+1) - cw^* \|^2 \leqslant \| w(k) - cw^* \|^2 - a \tag{3-5-3}$$

由上式可知，每次校正后，从 $w(k)$ 到 cw^* 的平方距离至少减小 a，经过 k 次校正后，可知

$$\| w(k+1) - cw^* \|^2 \leqslant \| w(1) - cw^* \|^2 - ka$$

由于平方距离不可能是负的，所以经过不超过 k_0 次校正后，算法将停止，其中

$$k_0 = \frac{\| w(1) - cw^* \|^2}{a} \tag{3-5-4}$$

证毕。

例3.8 已知两类训练样本

$$\omega_1\text{：}\ \boldsymbol{X}_1=[0,0]^{\mathrm{T}}\qquad \boldsymbol{X}_2=[0,1]^{\mathrm{T}}$$

$$\omega_2\text{：}\ \boldsymbol{X}_3=[1,0]^{\mathrm{T}}\qquad \boldsymbol{X}_4=[1,1]^{\mathrm{T}}$$

用感知器算法求出将模式分为两类的权向量解和判别函数。

解：所有样本写成增广向量形式；

进行规范化处理，属于ω_2的样本乘以(－1)。

$$\boldsymbol{X}_1=[0,0,1]^{\mathrm{T}}\qquad \boldsymbol{X}_2=[0,1,1]^{\mathrm{T}}\qquad \boldsymbol{X}_3=[-1,0,-1]^{\mathrm{T}}\qquad \boldsymbol{X}_4=[-1,-1,-1]^{\mathrm{T}}$$

任取$\boldsymbol{W}(1)=\boldsymbol{0}$，取$c$=1，迭代过程为：

第一轮：

$$\boldsymbol{W}^{\mathrm{T}}(1)\boldsymbol{X}_1=[0,0,0]\begin{bmatrix}0\\0\\1\end{bmatrix}=0,\ \leq 0,\qquad 故\boldsymbol{W}(2)=\boldsymbol{W}(1)+\boldsymbol{X}_1=[0,0,1]^{\mathrm{T}}$$

$$\boldsymbol{W}^{\mathrm{T}}(2)\boldsymbol{X}_2=[0,0,1]\begin{bmatrix}0\\1\\1\end{bmatrix}=1,\ >0,\qquad 故\boldsymbol{W}(3)=\boldsymbol{W}(2)=[0,0,1]^{\mathrm{T}}$$

$$\boldsymbol{W}^{\mathrm{T}}(3)\boldsymbol{X}_3=[0,0,1]\begin{bmatrix}-1\\0\\-1\end{bmatrix}=-1,\leq 0,\qquad 故\boldsymbol{W}(4)=\boldsymbol{W}(3)+\boldsymbol{X}_3=[-1,0,0]^{\mathrm{T}}$$

$$\boldsymbol{W}^{\mathrm{T}}(4)\boldsymbol{X}_4=[-1,0,0]\begin{bmatrix}-1\\-1\\-1\end{bmatrix}=1,>0,\qquad 故\boldsymbol{W}(5)=\boldsymbol{W}(4)=[-1,0,0]^{\mathrm{T}}$$

有两个$\boldsymbol{W}^{\mathrm{T}}(k)\boldsymbol{X}_i\leq 0$的情况（错判），进行第二轮迭代。

第二轮：

$$\boldsymbol{W}^{\mathrm{T}}(5)\boldsymbol{X}_1 = 0 \le 0,\quad 故\boldsymbol{W}(6) = \boldsymbol{W}(5) + \boldsymbol{X}_1 = [-1,0,1]^{\mathrm{T}}$$

$$\boldsymbol{W}^{\mathrm{T}}(6)\boldsymbol{X}_2 = 1 > 0,\quad 故\boldsymbol{W}(7) = \boldsymbol{W}(6) = [-1,0,1]^{\mathrm{T}}$$

$$\boldsymbol{W}^{\mathrm{T}}(7)\boldsymbol{X}_3 = 0 \le 0,\quad 故\boldsymbol{W}(8) = \boldsymbol{W}(7) + \boldsymbol{X}_3 = [-2,0,0]^{\mathrm{T}}$$

$$\boldsymbol{W}^{\mathrm{T}}(8)\boldsymbol{X}_4 = 2 > 0,\quad 故\boldsymbol{W}(9) = \boldsymbol{W}(8) = [-2,0,0]^{\mathrm{T}}$$

第三轮：

$$\boldsymbol{W}^{\mathrm{T}}(9)\boldsymbol{X}_1 = 0 \le 0,\quad 故\boldsymbol{W}(10) = \boldsymbol{W}(9) + \boldsymbol{X}_1 = [-2,0,1]^{\mathrm{T}}$$

$$\boldsymbol{W}^{\mathrm{T}}(10)\boldsymbol{X}_2 = 1 > 0,\quad 故\boldsymbol{W}(11) = \boldsymbol{W}(10)$$

$$\boldsymbol{W}^{\mathrm{T}}(11)\boldsymbol{X}_3 = 1 > 0,\quad 故\boldsymbol{W}(12) = \boldsymbol{W}(11)$$

$$\boldsymbol{W}^{\mathrm{T}}(12)\boldsymbol{X}_4 = 1 > 0,\quad 故\boldsymbol{W}(13) = \boldsymbol{W}(12)$$

第四轮：

$$\boldsymbol{W}^{\mathrm{T}}(13)\boldsymbol{X}_1 = 1 > 0,\quad 故\boldsymbol{W}(14) = \boldsymbol{W}(13)$$

$$\boldsymbol{W}^{\mathrm{T}}(14)\boldsymbol{X}_2 = 1 > 0,\quad 故\boldsymbol{W}(15) = \boldsymbol{W}(14)$$

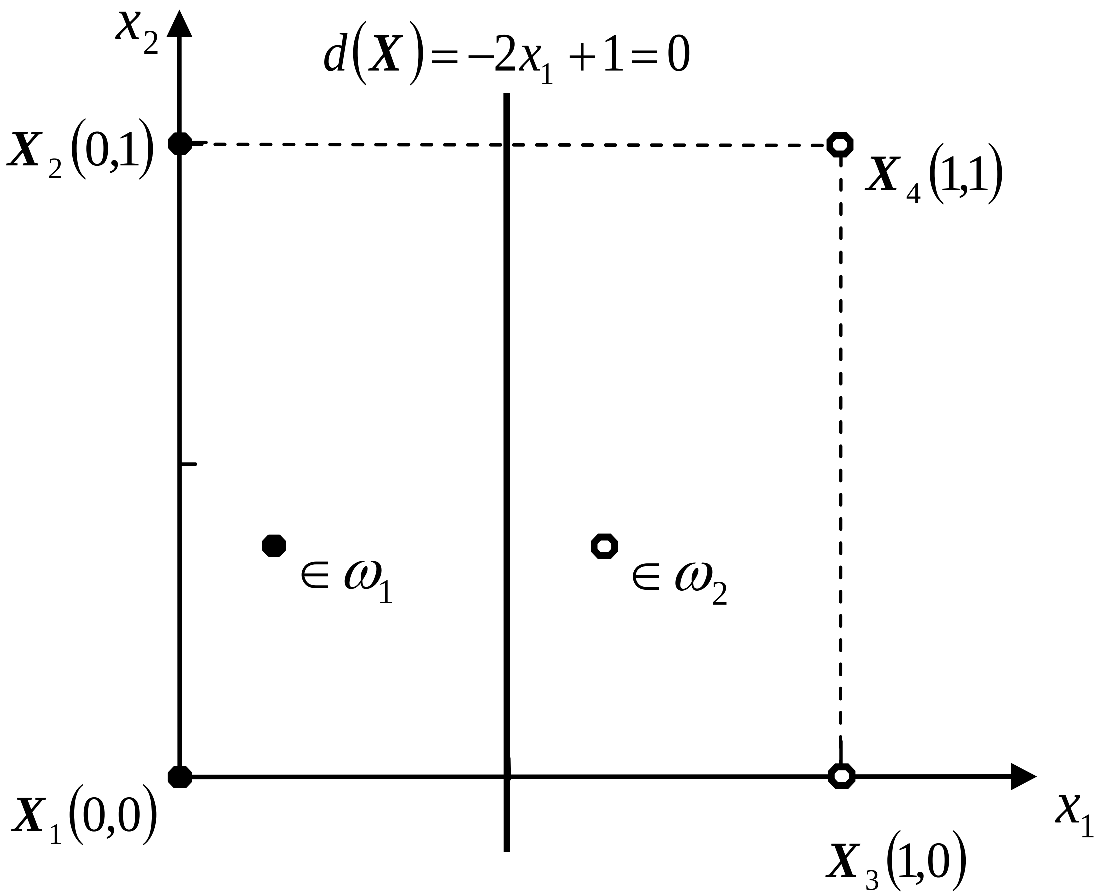
$$\boldsymbol{W}^{\mathrm{T}}(15)\boldsymbol{X}_3 = 1 > 0,\quad 故\boldsymbol{W}(16) = \boldsymbol{W}(15)$$

$$\boldsymbol{W}^{\mathrm{T}}(16)\boldsymbol{X}_4 = 1 > 0,\quad 故\boldsymbol{W}(17) = \boldsymbol{W}(16)$$

该轮迭代的分类结果全部正确，故解向量$\boldsymbol{W}=[-2,0,1]^{\mathrm{T}}$

相应的判别函数为：$d(\boldsymbol{X})=-2x_1+1$

判别界面$d(\boldsymbol{X})=0$如图示。法向量为（-2,0），指向-x_1方向。



当c、$\boldsymbol{W}(1)$取其他值时，结果可能不一样，所以**感知器算法的解不唯一**。

4. 感知器算法用于多类情况

采用多类情况3的方法时，应有：

若 $\boldsymbol{X} \in \omega_i$，则 $d_i(\boldsymbol{X}) > d_j(\boldsymbol{X}), \quad \forall j \neq i, \quad j = 1, \cdots, M$

对于M类模式应存在M个判决函数：$\{d_i, \quad i = 1, \cdots, M\}$

算法主要内容：

设有 M 种模式类别：$\omega_1, \omega_2, \cdots, \omega_M$

设其权向量初值为：$\boldsymbol{W}_j(1), \quad j = 1, \cdots, M$

训练样本为增广向量形式，但**不需要符号规范化处理**。

第k次迭代时，一个属于ω_i类的模式样本 $\boldsymbol{X}$ 被送入分类器，计算所有判别函数

$$d_j(k) = \boldsymbol{W}_j^{\mathrm{T}}(k)\boldsymbol{X}, \quad j = 1, \cdots, M$$

分二种情况修改权向量：

① 若 $d_i(k) > d_j(k), \quad \forall j \neq i; \; j = 1,2,\cdots,M$ 则权向量不变；

$$\boldsymbol{W}_j(k+1) = \boldsymbol{W}_j(k), \quad j = 1,2,\cdots,M$$

② 若第l个权使 $d_i(k) \leq d_l(k)$，则相应权作调整，其余不调整：

$$\begin{cases} \boldsymbol{W}_i(k+1) = \boldsymbol{W}_i(k) + c\boldsymbol{X} \\ \boldsymbol{W}_l(k+1) = \boldsymbol{W}_l(k) - c\boldsymbol{X}, \text{ 对满足} d_i(k) \leq d_l(k) \text{ 的} l \\ \boldsymbol{W}_j(k+1) = \boldsymbol{W}_j(k), \; j \neq i, l \end{cases}$$

可以证明：只要模式类在情况3判别函数时是可分的，则经过有限次迭代后算法收敛。c为正的校正增量。

例3.9 设有三个线性可分的模式类，三类的训练样本分别为

$$\omega_1: \; \boldsymbol{X}_1 = [0,0]^{\mathrm{T}}; \quad \omega_2: \; \boldsymbol{X}_2 = [1,1]^{\mathrm{T}}; \quad \omega_3: \; \boldsymbol{X}_3 = [-1,1]^{\mathrm{T}}$$

现采用多类情况3的方式分类，试用感知器算法求出判别函数。

解：增广向量形式：

$$\boldsymbol{X}_1=[0,0,1]^{\mathrm{T}},\quad \boldsymbol{X}_2=[1,1,1]^{\mathrm{T}},\quad \boldsymbol{X}_3=[-1,1,1]^{\mathrm{T}}$$

注意，这里任一类的样本都不能乘以（－1）。

任取初始权向量 $\boldsymbol{W}_1(1)=\boldsymbol{W}_2(1)=\boldsymbol{W}_3(1)=[0,0,0]^{\mathrm{T}}$ ； c=1

第一次迭代：

$$d_1(1)=\boldsymbol{W}_1^{\mathrm{T}}(1)\boldsymbol{X}_1=0$$

$$d_2(1)=\boldsymbol{W}_2^{\mathrm{T}}(1)\boldsymbol{X}_1=0$$

$$d_3(1)=\boldsymbol{W}_3^{\mathrm{T}}(1)\boldsymbol{X}_1=0$$

$\boldsymbol{X}_1\in\omega_1$ ，但 $d_1(1)>d_2(1)$ 且 $d_1(1)>d_3(1)$ 不成立，

三个权向量都需要修改：

$$\boldsymbol{W}_1(2)=\boldsymbol{W}_1(1)+\boldsymbol{X}_1=[0,0,1]^{\mathrm{T}}$$

$$\boldsymbol{W}_2(2)=\boldsymbol{W}_2(1)-\boldsymbol{X}_1=[0,0,-1]^{\mathrm{T}}$$

$$\boldsymbol{W}_3(2)=\boldsymbol{W}_3(1)-\boldsymbol{X}_1=[0,0,-1]^{\mathrm{T}}$$

第二次迭代：
$$d_1(2)=\boldsymbol{W}_1^{\mathrm{T}}(2)\boldsymbol{X}_2=1$$
$$d_2(2)=\boldsymbol{W}_2^{\mathrm{T}}(2)\boldsymbol{X}_2=-1$$
$$d_3(2)=\boldsymbol{W}_3^{\mathrm{T}}(2)\boldsymbol{X}_2=-1$$

$\boldsymbol{X}_2\in\omega_2$，但 $d_2(2)>d_1(2)$ 且 $d_2(2)>d_3(2)$ 不成立，修改权向量：

$$\boldsymbol{W}_1(3)=\boldsymbol{W}_1(2)-\boldsymbol{X}_2=[-1,-1,0]^{\mathrm{T}}$$
$$\boldsymbol{W}_2(3)=\boldsymbol{W}_2(2)+\boldsymbol{X}_2=[1,1,0]^{\mathrm{T}}$$
$$\boldsymbol{W}_3(3)=\boldsymbol{W}_3(2)-\boldsymbol{X}_2=[-1,-1,-2]^{\mathrm{T}}$$

第三次迭代：$\boldsymbol{X}_3\in\omega_3$，但 $d_3(3)>d_1(3)$ 且 $d_3(3)>d_2(3)$ 不成立，继续修改权向量。

以上进行的一轮迭代运算中，三个样本都未正确分类，进行下一轮迭代。

第四次迭代：　……

在第五、六、七迭代中，对所有三个样本都已正确分类。

权向量的解：

$$\boldsymbol{W}_1=\boldsymbol{W}_1(7)=\boldsymbol{W}_1(6)=\boldsymbol{W}_1(5)=[0,-2,0]^{\mathrm{T}}$$

$$\boldsymbol{W}_2=\boldsymbol{W}_2(7)=\boldsymbol{W}_2(6)=\boldsymbol{W}_2(5)=[2,0,-2]^{\mathrm{T}}$$

$$\boldsymbol{W}_3=\boldsymbol{W}_3(7)=\boldsymbol{W}_3(6)=\boldsymbol{W}_3(5)=[-2,0,-2]^{\mathrm{T}}$$

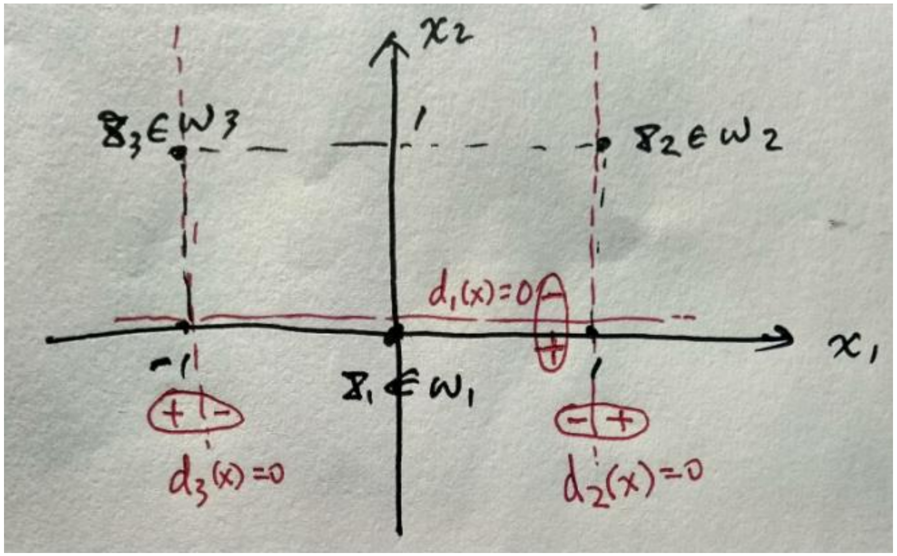
判别函数：

$$d_1(\boldsymbol{X})=-2x_2$$

$$d_2(\boldsymbol{X})=2x_1-2$$

$$d_3(\boldsymbol{X})=-2x_1-2$$

（画出样本位置与判别函数）



3.7 梯度算法

3.7.1 梯度算法基本原理

1. 梯度概念

设函数$f(\boldsymbol{Y})$是向量$\boldsymbol{Y}=\left[y_1,y_2,\cdots,y_n\right]^{\mathrm{T}}$的函数，则$f(\boldsymbol{Y})$的梯度定义为：

$$\nabla f(\boldsymbol{Y})=\frac{\mathrm{d}}{\mathrm{d}\boldsymbol{Y}}f(\boldsymbol{Y})=\left[\frac{\partial f}{\partial y_1},\frac{\partial f}{\partial y_2},\cdots,\frac{\partial f}{\partial y_n}\right]^{\mathrm{T}}$$

梯度向量的最重要性质之一：指出函数 f 在其自变量增加时，增长最快的方向。

即：

梯度的**方向**是函数$f(\boldsymbol{Y})$ 在Y点增长最快的方向，

梯度的**模**是$f(\boldsymbol{Y})$在增长最快的方向上的增长率 (增长率最大值)。

显然：负梯度指出了最陡下降方向。——梯度算法的依据。

2. 梯度算法

设两个线性可分的模式类ω_1和ω_2的样本共N个，ω_2类样本乘(-1)。将两类样本分开的判决函数$d(\boldsymbol{X})$应满足：

$$d(\boldsymbol{X}_i)=\boldsymbol{W}^{\mathrm{T}}\boldsymbol{X}_i>0 \qquad i=1,2,\cdots,N \qquad \text{——}N\text{个不等式}$$

梯度算法的目的仍然是求一个满足上述条件的权向量，主导思想是将联立不等式求解$\boldsymbol{W}$的问题，转换成求准则函数极小值的问题。

用负梯度向量的值对权向量$\boldsymbol{W}$进行修正，实现使准则函数达到极小值的目的。

准则函数的选取原则：

具有唯一的最小值，并且这个最小值发生在$\boldsymbol{W}^{\mathrm{T}}\boldsymbol{X}_i>0$时。

基本思路：

定义一个对错误分类敏感的准则函数$J(\boldsymbol{W}, \boldsymbol{X})$，在$J$的梯度方向上对权向量进行修改。一般关系表示成从$\boldsymbol{W}(k)$导出$\boldsymbol{W}(k+1)$：

$$\boldsymbol{W}(k+1)=\boldsymbol{W}(k)+c(-\nabla J)=\boldsymbol{W}(k)-c\nabla J$$

$$\boldsymbol{W}(k+1)=\boldsymbol{W}(k)-c\left[\frac{\partial J(\boldsymbol{W},\boldsymbol{X})}{\partial \boldsymbol{W}}\right]_{\boldsymbol{W}=\boldsymbol{W}(k)}$$

其中c是正的比例因子。

梯度法求解步骤：

（1）将样本写成规范化增广向量形式，选择准则函数，设置初始权向量$\boldsymbol{W}(1)$，括号内为迭代次数k=1。

（2）依次输入训练样本$\boldsymbol{X}$。设第k次迭代时输入样本为$\boldsymbol{X}_i$，此时已有权向量$\boldsymbol{W}(k)$，求$\nabla J(k)$：

$$\nabla J(k)=\left.\frac{\partial J\left(\boldsymbol{W},\boldsymbol{X}_i\right)}{\partial \boldsymbol{W}}\right|_{\boldsymbol{W}=\boldsymbol{W}(k)}$$

权向量修正为：

$$\boldsymbol{W}\left(k+1\right)=\boldsymbol{W}\left(k\right)-c\nabla J(k)$$

迭代次数k加1，输入下一个训练样本，计算新的权向量，直至对全部训练样本完成一轮迭代。

（3）在一轮迭代中，如果有一个样本使$\nabla J\neq 0$，回到(2)进行下一轮迭代。否则，$\boldsymbol{W}$不再变化，算法收敛。

例3.10 选择准则函数，$J(\boldsymbol{W},\boldsymbol{X})=\left|\boldsymbol{W}^{\mathrm{T}}\boldsymbol{X}\right|-\boldsymbol{W}^{\mathrm{T}}\boldsymbol{X}$，简单地考虑$\boldsymbol{X}$为一维增广模式的情况$\boldsymbol{X}$=[0,1]，此时$\boldsymbol{W}=[w_1\ w_2]$，因为X的第1项为0，简记为标量：$J(\boldsymbol{W},\boldsymbol{X})=|w|-w$

错误分类时：

思考：$|w|-w$的导数？

$$\boldsymbol{W}^{\mathrm{T}}\boldsymbol{X}<0 \Rightarrow w\cdot 1<0 \Rightarrow w<0$$

$$\nabla J=\frac{\partial J}{\partial \boldsymbol{W}}(\boldsymbol{W},\boldsymbol{X})\bigg|_{X=1}=\frac{\partial}{\partial w}\left(|w|-w\right)=\frac{\partial}{\partial w}(-2w)=-2$$

$$\therefore\quad \boldsymbol{W}(k+1)=\boldsymbol{W}(k)-c\cdot(-2)=\boldsymbol{W}(k)+2c,$$ 对权向量校正。

$$\boldsymbol{W}(k+1)=\boldsymbol{W}(k)-c\nabla J$$

正确分类时：

$$\boldsymbol{W}^{\mathrm{T}}\boldsymbol{X}>0 \Rightarrow w\cdot 1>0 \Rightarrow w>0 \quad \nabla J=\frac{\partial}{\partial w}\left(|w|-w\right)=0$$

$$\therefore\quad \boldsymbol{W}(k+1)=\boldsymbol{W}(k)-c\cdot 0=\boldsymbol{W}(k),$$ 对权向量不做修正。

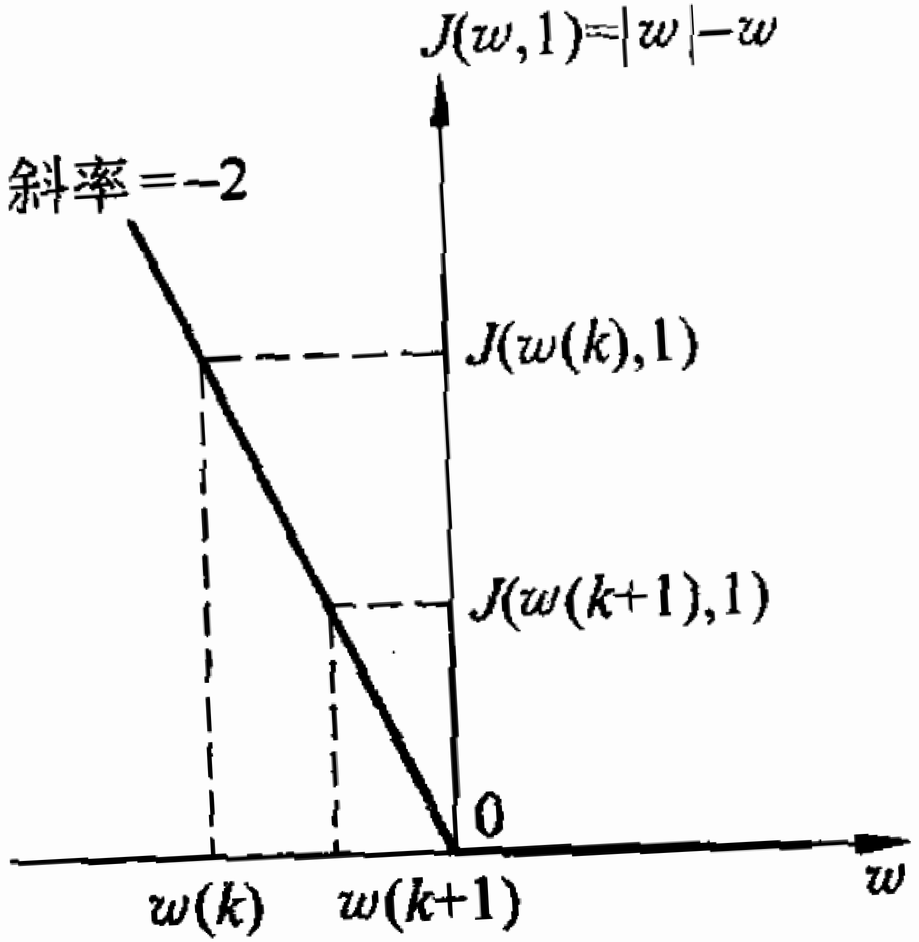


图 3.16　梯度法的几何说明

这是错误分类时，权向量调整示意图。此时梯度为-2，负梯度为2，因此$W(k+1)=W(k)+2$，可见：$J(w(k+1),1)>J(w(k),1)$

说明：

a) $$\boldsymbol{W}(k+1)=\boldsymbol{W}(k)-c\nabla J=\boldsymbol{W}(k)-c\left[\frac{\partial J(\boldsymbol{W},\boldsymbol{X})}{\partial \boldsymbol{W}}\right]_{\boldsymbol{W}=\boldsymbol{W}(k)}$$

随着权向量W向理想值接近，准则函数关于$\boldsymbol{W}$的导数（∇J）越来越趋近于零，这意味着准则函数J越来越接近最小值。当最终$\nabla J=0$时，J达到最小值，此时$\boldsymbol{W}$不再改变，算法收敛。

——将感知器算法中联立不等式求解$\boldsymbol{W}$的问题，转换为求函数J极小值的问题。

b) c值的选择很重要，如c值太小，收敛太慢；但若太大，搜索又可能过头，甚至引起发散。一般，开始阶段c取大一些，收尾阶段取小——变步长。

c) 梯度算法是求解权向量的一般解法，算法的具体计算形式取决于准则函数$J(\boldsymbol{W}, \boldsymbol{X})$的选择，$J(\boldsymbol{W}, \boldsymbol{X})$的形式不同，得到的具体算法不同。

3.7.2 固定增量法

准则函数：$J(\boldsymbol{W},\boldsymbol{X})=\frac{1}{2}\left(\left|\boldsymbol{W}^{\mathrm{T}}\boldsymbol{X}\right|-\boldsymbol{W}^{\mathrm{T}}\boldsymbol{X}\right)$

该准则函数有唯一最小值“0”，且发生在$\boldsymbol{W}^{\mathrm{T}}\boldsymbol{X}>0$ 的时候。

求$\boldsymbol{W}(k)$的递推公式：

设 $\boldsymbol{X}=\left[x_1,x_2,\cdots,x_n,1\right]^{\mathrm{T}}$ ， $\boldsymbol{W}=\left[w_1,w_2,\cdots,w_n,w_{n+1}\right]^{\mathrm{T}}$

1. 求J的梯度 $\nabla J=\dfrac{\partial J(\boldsymbol{W},\boldsymbol{X})}{\partial \boldsymbol{W}}=?$

方法：函数对向量求导=函数对向量的分量求导，即

$$\frac{\partial f}{\partial \boldsymbol{X}}=\left[\frac{\partial f}{\partial x_1},\cdots,\frac{\partial f}{\partial x_n}\right]^{\mathrm{T}}$$

$$J(\boldsymbol{W},\boldsymbol{X})=\frac{1}{2}\left(\left|\boldsymbol{W}^{\mathrm{T}}\boldsymbol{X}\right|-\boldsymbol{W}^{\mathrm{T}}\boldsymbol{X}\right)$$

①首先求$\boldsymbol{W}^{\mathrm{T}}\boldsymbol{X}$部分：

$$\frac{\partial\left(\boldsymbol{W}^{\mathrm{T}}\boldsymbol{X}\right)}{\partial\boldsymbol{W}}=\frac{\partial}{\partial\boldsymbol{W}}\left(\sum_{i=1}^{n}w_i x_i+w_{n+1}\right)$$

$$=\left[\frac{\partial}{\partial w_1}\left(\sum_{i=1}^{n}w_i x_i+w_{n+1}\right),\cdots,\frac{\partial}{\partial w_k}\left(\sum_{i}w_i x_i+w_{n+1}\right),\cdots,\frac{\partial}{\partial w_{n+1}}\left(\sum_{i}w_i x_i+w_{n+1}\right)\right]^{\mathrm{T}}$$

$$=\left[x_1,\cdots,x_k,\cdots,x_n,1\right]^{\mathrm{T}}=\boldsymbol{X}$$

另：矩阵论中有（也可以直接按梯度定义演算）

$$\frac{d\boldsymbol{X}}{d\boldsymbol{X}^{\mathrm{T}}}=\frac{d\boldsymbol{X}^{\mathrm{T}}}{d\boldsymbol{X}}=\boldsymbol{I}_{n\times n}$$

$$\therefore\quad\frac{\partial\left(\boldsymbol{W}^{\mathrm{T}}\boldsymbol{X}\right)}{\partial\boldsymbol{W}}=\boldsymbol{I}_{(n+1)\times(n+1)}\boldsymbol{X}_{(n+1)\times 1}=\boldsymbol{X}_{(n+1)\times 1}$$

$$J(\boldsymbol{W},\boldsymbol{X})=\frac{1}{2}\left(\left|\boldsymbol{W}^{\mathrm{T}}\boldsymbol{X}\right|-\boldsymbol{W}^{\mathrm{T}}\boldsymbol{X}\right)$$

② 由①的结论 $\frac{\partial\left(\boldsymbol{W}^{\mathrm{T}}\boldsymbol{X}\right)}{\partial\boldsymbol{W}}=\boldsymbol{X}$ 有：

$$\boldsymbol{W}^{\mathrm{T}}\boldsymbol{X}>0\text{时},\frac{\partial\left(\left|\boldsymbol{W}^{\mathrm{T}}\boldsymbol{X}\right|\right)}{\partial\boldsymbol{W}}=\frac{\partial\left(\boldsymbol{W}^{\mathrm{T}}\boldsymbol{X}\right)}{\partial\boldsymbol{W}}=\boldsymbol{X}$$

$$\boldsymbol{W}^{\mathrm{T}}\boldsymbol{X}\le 0\text{时},\frac{\partial\left(\left|\boldsymbol{W}^{\mathrm{T}}\boldsymbol{X}\right|\right)}{\partial\boldsymbol{W}}=\frac{\partial\left(-\boldsymbol{W}^{\mathrm{T}}\boldsymbol{X}\right)}{\partial\boldsymbol{W}}=-\boldsymbol{X}$$

$$\therefore\quad\frac{\partial\left(\left|\boldsymbol{W}^{\mathrm{T}}\boldsymbol{X}\right|\right)}{\partial\boldsymbol{W}}=\left[\operatorname{sgn}\left(\boldsymbol{W}^{\mathrm{T}}\boldsymbol{X}\right)\right]\cdot\boldsymbol{X}$$

其中 $\operatorname{sgn}\left(\boldsymbol{W}^{\mathrm{T}}\boldsymbol{X}\right)=\begin{cases}+1, & \text{若}\boldsymbol{W}^{\mathrm{T}}\boldsymbol{X}>0\\ -1, & \text{若}\boldsymbol{W}^{\mathrm{T}}\boldsymbol{X}\le 0\end{cases}$

$$\therefore\quad\nabla J=\frac{\partial J(\boldsymbol{W},\boldsymbol{X})}{\partial\boldsymbol{W}}=\frac{1}{2}\left[\boldsymbol{X}\operatorname{sgn}\left(\boldsymbol{W}^{\mathrm{T}}\boldsymbol{X}\right)-\boldsymbol{X}\right]$$

2. 求$\boldsymbol{W}(k+1)$

将 $\nabla J=\dfrac{\partial J(\boldsymbol{W},\boldsymbol{X})}{\partial \boldsymbol{W}}=\dfrac{1}{2}\left[\boldsymbol{X}\operatorname{sgn}\left(\boldsymbol{W}^{\mathrm{T}}\boldsymbol{X}\right)-\boldsymbol{X}\right]$代入

$$\boldsymbol{W}(k+1)=\boldsymbol{W}(k)-c\nabla J=\boldsymbol{W}(k)-c\left[\frac{\partial J(\boldsymbol{W},\boldsymbol{X})}{\partial \boldsymbol{W}}\right]_{\boldsymbol{W}=\boldsymbol{W}(k)}$$

得：$\boldsymbol{W}(k+1)=\boldsymbol{W}(k)-c\dfrac{1}{2}\left[\boldsymbol{X}\operatorname{sgn}\left(\boldsymbol{W}^{\mathrm{T}}(k)\boldsymbol{X}\right)-\boldsymbol{X}\right]$

$$=\boldsymbol{W}(k)+\frac{c}{2}\left[\boldsymbol{X}-\boldsymbol{X}\operatorname{sgn}\left(\boldsymbol{W}^{\mathrm{T}}(k)\boldsymbol{X}\right)\right]$$

$$=\boldsymbol{W}(k)+\begin{cases}0, & \text{若}\boldsymbol{W}^{\mathrm{T}}(k)\boldsymbol{X}>0\\ c\boldsymbol{X}, & \text{若}\boldsymbol{W}^{\mathrm{T}}(k)\boldsymbol{X}\le 0\end{cases}$$

即：$$\boldsymbol{W}(k+1)=\boldsymbol{W}(k)+\begin{cases}0, & 若\boldsymbol{W}^{\mathrm{T}}(k)\boldsymbol{X}>0\\ c\boldsymbol{X}, & 若\boldsymbol{W}^{\mathrm{T}}(k)\boldsymbol{X}\le 0\end{cases}$$

上式即为**固定增量算法**，与感知器算法形式完全相同 。

$$\boldsymbol{W}(k+1)=\begin{cases}\boldsymbol{W}(k), & 若\boldsymbol{W}^{\mathrm{T}}(k)\boldsymbol{X}_i>0\\ \boldsymbol{W}(k)+c\boldsymbol{X}_i, & 若\boldsymbol{W}^{\mathrm{T}}(k)\boldsymbol{X}_i\le 0\end{cases}$$

由此可以看出，**感知器算法是梯度法的特例**。即：梯度法是将感知器算法中联立不等式求解W的问题，转换为求函数J极小值的问题，将原来有多个解的情况，变成求最优解的情况。

只要模式类是线性可分的，算法就会给出解。

3.8 最小平方误差算法（最小均方误差算法）

（Least Mean Square Error, LMS, MSE；亦称Ho-Kashyap算法）

上述的感知器算法、梯度算法、固定增量算法或其他类似方法，只有当模式是线性可分时才收敛，在不可分的情况下，算法会来回摆动，始终不收敛。当一次次迭代而又不见收敛时，造成不收敛现象的原因分不清，有两种可能：

a) 迭代过程本身收敛缓慢

b) 模式本身线性不可分

LMS算法特点：

- 对线性可分模式收敛。
- 对线性不可分的情况也能指出来。

1. 分类器的不等式方程组

两类分类问题的解相当于求一组线性不等式的解。如果给出分属于ω_1，ω_2两个模式类的训练样本集$\{\boldsymbol{X}_i,\ i=1,2,\cdots,N\}$，应满足：

$$\boldsymbol{W}^{\mathrm{T}}\boldsymbol{X}_i > 0$$

其中，$\boldsymbol{X}_i$是规范化增广样本向量，$\boldsymbol{X}_i=[x_{i1},x_{i2},\cdots,x_{in},1]^{\mathrm{T}}$。

上式分开写为：

$$\begin{array}{c} \omega_1\text{类} \\ \\ \\ \omega_2\text{类} \end{array} \begin{cases} w_1x_{11}+w_2x_{12}+\cdots\cdots+w_nx_{1n}+w_{n+1}>0 & \text{对}\boldsymbol{X}_1 \\ w_1x_{21}+w_2x_{22}+\cdots\cdots+w_nx_{2n}+w_{n+1}>0 & \text{对}\boldsymbol{X}_2 \\ \vdots & \\ -w_1x_{N1}-w_2x_{N2}-\cdots-w_nx_{Nn}-w_{n+1}>0 & \text{对}\boldsymbol{X}_N \end{cases}$$

$$
\begin{array}{ll}
\omega_1\text{类} & \left\{\begin{array}{ll}
w_1x_{11}+w_2x_{12}+\cdots\cdots+w_nx_{1n}+w_{n+1}>0 & \text{对}\boldsymbol{X}_1 \\
w_1x_{21}+w_2x_{22}+\cdots\cdots+w_nx_{2n}+w_{n+1}>0 & \text{对}\boldsymbol{X}_2 \\
\vdots & \\
\omega_2\text{类}\quad -w_1x_{N1}-w_2x_{N2}-\cdots-w_nx_{Nn}-w_{n+1}>0 & \text{对}\boldsymbol{X}_N
\end{array}\right.
\end{array}
$$

写成矩阵形式为：

$$
\begin{bmatrix}
x_{11} & x_{12} & \cdots & x_{1n} & 1 \\
x_{21} & x_{22} & \cdots & x_{2n} & 1 \\
\vdots & \vdots & \vdots & \vdots & \vdots \\
-x_{N1} & -x_{N2} & \cdots & -x_{Nn} & -1
\end{bmatrix}_{N\times(n+1)}
\begin{bmatrix}
w_1 \\ w_2 \\ \vdots \\ w_n \\ w_{n+1}
\end{bmatrix}_{(n+1)\times 1}
>
\begin{bmatrix}
0 \\ 0 \\ \vdots \\ 0 \\ 0
\end{bmatrix}_{N\times 1}
=\boldsymbol{0}
$$

令$N\times(n+1)$的长方矩阵为$\boldsymbol{X}$，则 $\boldsymbol{W}^{\mathrm{T}}\boldsymbol{X}_i>0$ 变为：

$$
\boldsymbol{XW}>\boldsymbol{0}
$$

$$\begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1n} & 1 \\ x_{21} & x_{22} & \cdots & x_{2n} & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ -x_{N1} & -x_{N2} & \cdots & -x_{Nn} & -1 \end{bmatrix}_{N\times(n+1)} \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \\ w_{n+1} \end{bmatrix}_{(n+1)\times 1} > \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix}_{N\times 1} = \mathbf{0}$$

$$\boldsymbol{XW} > \mathbf{0}$$

式中：

$$\boldsymbol{X} = \begin{bmatrix} \boldsymbol{X}_1^{\mathrm{T}} \\ \boldsymbol{X}_2^{\mathrm{T}} \\ \vdots \\ \boldsymbol{X}_i^{\mathrm{T}} \\ \vdots \\ -\boldsymbol{X}_{N-1}^{\mathrm{T}} \\ -\boldsymbol{X}_N^{\mathrm{T}} \end{bmatrix}_{N\times(n+1)} \begin{matrix} \left.\begin{matrix} \\ \\ \\ \\ \end{matrix}\right\} \in \omega_1 \\ \left.\begin{matrix} \\ \\ \\ \end{matrix}\right\} \in \omega_2 \end{matrix}$$

$$\boldsymbol{W} = [w_1, w_2, \cdots, w_n, w_{n+1}]^{\mathrm{T}}$$

$\mathbf{0}$为零向量

感知器算法是通过解不等式组 $\boldsymbol{XW} > \mathbf{0}$，求出$\boldsymbol{W}$。

2. LMS算法

1） 原理

LMS算法把对满足 $\boldsymbol{XW} > \mathbf{0}$ 的求解，改为满足

$$\boldsymbol{XW} = \boldsymbol{B}$$

的求解。式中：

$\boldsymbol{B} = [b_1, b_2, \cdots, b_i, \cdots, b_N]^{\mathrm{T}}$ 为各分量均为正值的矢量。

$\therefore$ 两式等价。

说明：

① 在方程组中当行数>>列数时，通常无解，称为矛盾方程组，一般求近似解。在模式识别中，通常训练样本数N总是大于模式的维数n，因此方程的个数(行数)>>模式向量的维数(列数)，是矛盾方程组，只能求近似解$\boldsymbol{W}^*$，即

$$\|\boldsymbol{XW}^* - \boldsymbol{B}\| = 极小$$

② LMS算法的出发点：选择一个准则函数，使得当J达到最小值时，$\boldsymbol{XW}=\boldsymbol{B}$ 可得到近似解（最小二乘近似解）。

准则函数定义为：

$$J(\boldsymbol{W},\boldsymbol{X},\boldsymbol{B})=\frac{1}{2}\left\|\boldsymbol{XW}-\boldsymbol{B}\right\|^2$$

“最小二乘”：

$$\left.\begin{array}{l}\text{—— 最小：使方程组两边误差最小，}\\ \qquad\qquad\text{也即使}J\text{最小。}\\ \text{—— 二乘：二次方。}\end{array}\right\}\ \text{最小平方（误差算法）}$$

③ LMS算法的思路：对$\boldsymbol{XW}>\boldsymbol{0}$求解

$$\Big\downarrow \text{转化为}$$

对$\boldsymbol{XW}=\boldsymbol{B}$求解

$$\Big\downarrow \text{转化为}$$

通过求准则函数极小找 $\boldsymbol{W}$、$\boldsymbol{B}$

考察向量($\boldsymbol{XW}$－$\boldsymbol{B}$) 有：

$$\boldsymbol{XW}-\boldsymbol{B}=\begin{bmatrix} x_{11} & \cdots & x_{1n} & 1 \\ x_{21} & \cdots & x_{2n} & 1 \\ \vdots & \cdots & \vdots & 1 \\ x_{i1} & \cdots & x_{in} & 1 \\ \vdots & \cdots & \vdots & 1 \\ -x_{N1} & \cdots & -x_{Nn} & 1 \end{bmatrix}_{N\times(n+1)} \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \\ w_{n+1} \end{bmatrix}_{(n+1)\times 1} - \begin{bmatrix} b_1 \\ \vdots \\ b_i \\ \vdots \\ b_N \end{bmatrix}_{N\times 1}$$

$$=\begin{bmatrix} x_{11}w_1+\cdots+x_{1n}w_n+w_{n+1}-b_1 \\ \vdots \\ x_{i1}w_1+\cdots+x_{in}w_n+w_{n+1}-b_i \\ \vdots \\ -x_{N1}w_1-\cdots-x_{Nn}w_n-w_{n+1}-b_N \end{bmatrix}_{N\times 1} = \begin{bmatrix} \boldsymbol{W}^{\mathrm{T}}\boldsymbol{X}_1-b_1 \\ \vdots \\ \boldsymbol{W}^{\mathrm{T}}\boldsymbol{X}_i-b_i \\ \vdots \\ \boldsymbol{W}^{\mathrm{T}}\boldsymbol{X}_N-b_N \end{bmatrix}$$

$$\|\boldsymbol{XW}-\boldsymbol{B}\|^2=\left(\sqrt{\text{向量各分量的平方和}}\right)^2=\text{向量各分量的平方和}$$

$$\|\boldsymbol{XW}-\boldsymbol{B}\|^2=\left(\boldsymbol{W}^{\mathrm{T}}\boldsymbol{X}_1-b_1\right)^2+\cdots+\left(\boldsymbol{W}^{\mathrm{T}}\boldsymbol{X}_N-b_N\right)^2=\sum_{i=1}^{N}\left(\boldsymbol{W}^{\mathrm{T}}\boldsymbol{X}_i-b_i\right)^2$$

准则函数：

$$J(\boldsymbol{W},\boldsymbol{X},\boldsymbol{B})=\frac{1}{2}\|\boldsymbol{XW}-\boldsymbol{B}\|^2=\frac{1}{2}\sum_{i=1}^{N}\left(\boldsymbol{W}^{\mathrm{T}}\boldsymbol{X}_i-b_i\right)^2$$

使准则函数最小的、$\boldsymbol{XW}=\boldsymbol{B}$ 的近似解也称“最优近似解”：

—— 使方程组两边所有误差之和最小（即最优）的解。

可以看出：

① 当函数J达到最小值，等式$\boldsymbol{XW}=\boldsymbol{B}$有最优解。即又将问题转化为求准则函数极小值（线性优化）的问题。

② 因为J有两个变量$\boldsymbol{W}$和$\boldsymbol{B}$，有更多的自由度供选择求解，故可望改善算法的收敛速率。

$$\boldsymbol{X}^{\#}=\left(\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}\right)^{-1}\boldsymbol{X}^{\mathrm{T}}$$

$\boldsymbol{X}$列满秩时：$\boldsymbol{X}^{\#}\boldsymbol{X}=I_n$

$\boldsymbol{X}$行满秩时：$\boldsymbol{X}\boldsymbol{X}^{\#}=I_m$

2）推导LMS算法递推公式

与问题相关的两个梯度：

$$\frac{\partial J}{\partial \boldsymbol{W}}=\boldsymbol{X}^{\mathrm{T}}\left(\boldsymbol{X}\boldsymbol{W}-\boldsymbol{B}\right) \tag{3-45}$$

$$\frac{\partial J}{\partial \boldsymbol{B}}=-\frac{1}{2}\left[\left(\boldsymbol{X}\boldsymbol{W}-\boldsymbol{B}\right)+\left|\boldsymbol{X}\boldsymbol{W}-\boldsymbol{B}\right|\right] \tag{3-46}$$

求递推公式：

(1) 求$\boldsymbol{W}$的递推关系

使J对$\boldsymbol{W}$求最小，令$\dfrac{\partial J}{\partial \boldsymbol{W}}=0$，得：

$$\boldsymbol{X}^{\mathrm{T}}\left(\boldsymbol{X}\boldsymbol{W}-\boldsymbol{B}\right)=0\Rightarrow \boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}\boldsymbol{W}=\boldsymbol{X}^{\mathrm{T}}\boldsymbol{B}\Rightarrow \boldsymbol{W}=\left(\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}\right)^{-1}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{B}=\boldsymbol{X}^{\#}\boldsymbol{B} \tag{3-47}$$

式中：$\boldsymbol{X}^{\#}=\left(\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}\right)^{-1}\boldsymbol{X}^{\mathrm{T}}$称为$\boldsymbol{X}$的左伪逆（广义逆），

$\boldsymbol{X}$为$N\times(n+1)$长方阵，$\boldsymbol{X}^{\#}$为$(n+1)\times N$长方阵。

由(3-47)式可知：只要求出$\boldsymbol{B}$，就可求出$\boldsymbol{W}$。

补充证明：

$$\frac{\partial J}{\partial \boldsymbol{W}} = \boldsymbol{X}^{\mathrm{T}}(\boldsymbol{XW} - \boldsymbol{B}) \tag{3-45}$$

$$J = \frac{1}{2}\left\|\boldsymbol{XW} - \boldsymbol{B}\right\|^2 = \frac{1}{2}(\boldsymbol{XW} - \boldsymbol{B})^{\mathrm{T}}(\boldsymbol{XW} - \boldsymbol{B})$$

$$= \frac{1}{2}(\boldsymbol{W}^{\mathrm{T}}\boldsymbol{X}^{\mathrm{T}} - \boldsymbol{B}^{\mathrm{T}})(\boldsymbol{XW} - \boldsymbol{B}) = \frac{1}{2}(\boldsymbol{W}^{\mathrm{T}}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{XW} - 2\boldsymbol{W}^{\mathrm{T}}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{B} + \boldsymbol{B}^{\mathrm{T}}\boldsymbol{B})$$

于是有： $\frac{\partial J}{\partial \boldsymbol{W}} = \frac{1}{2}(2\boldsymbol{X}^{\mathrm{T}}\boldsymbol{XW} - 2\boldsymbol{X}^{\mathrm{T}}\boldsymbol{B}) = \boldsymbol{X}^{\mathrm{T}}(\boldsymbol{XW} - \boldsymbol{B})$

补充证明：

$$\frac{\partial J}{\partial \boldsymbol{B}} = -\frac{1}{2}\left[(\boldsymbol{XW} - \boldsymbol{B}) + \left|\boldsymbol{XW} - \boldsymbol{B}\right|\right] \tag{3-46}$$

$$\frac{\partial J}{\partial \boldsymbol{B}} = \frac{1}{2}(-2\boldsymbol{XW} + 2\boldsymbol{B}) = -(\boldsymbol{XW} - \boldsymbol{B})$$

为了后续迭代需要，须确保 $\frac{\partial J}{\partial \boldsymbol{B}} \le 0$, 于是令 $\frac{\partial J}{\partial \boldsymbol{B}} = -\frac{1}{2}\left[(\boldsymbol{XW} - \boldsymbol{B}) + |\boldsymbol{XW} - \boldsymbol{B}|\right]$

$$\frac{\partial J}{\partial \boldsymbol{B}}=-\frac{1}{2}\left[(\boldsymbol{XW}-\boldsymbol{B})+|\boldsymbol{XW}-\boldsymbol{B}|\right] \quad (3\text{-}46)$$

(2) 求$\boldsymbol{B}(k+1)$的迭代式

利用梯度算法公式 $\boldsymbol{W}(k+1)=\boldsymbol{W}(k)-c\left[\frac{\partial J(\boldsymbol{W},\boldsymbol{X})}{\partial \boldsymbol{W}}\right]_{\boldsymbol{W}=\boldsymbol{W}(k)}$ 有：

$$\boldsymbol{B}(k+1)=\boldsymbol{B}(k)-c'\left[\frac{\partial J}{\partial \boldsymbol{B}}\right]_{\boldsymbol{B}=\boldsymbol{B}(k)}$$

(3-46)代入，得

$$\boldsymbol{B}(k+1)=\boldsymbol{B}(k)+\frac{c'}{2}\left[(\boldsymbol{XW}(k)-\boldsymbol{B}(k))+|\boldsymbol{XW}(k)-\boldsymbol{B}(k)|\right]$$

令 $c'/2=c$ ，定义$\boldsymbol{XW}(k)-\boldsymbol{B}(k)=\boldsymbol{e}(k)$ (3-49)

$$\boldsymbol{B}(k+1)=\boldsymbol{B}(k)+c\left[\boldsymbol{e}(k)+|\boldsymbol{e}(k)|\right] \quad (3\text{-}50)$$

$$X^{\#}e(k)=X^{\#}[XW(k)-B(k)]$$
$$=X^{\#}XW(k)-X^{\#}B(k)$$
$$=W(k)-W(k)$$
$$=\mathbf{0}$$

(3) 求$W(k+1)$的迭代式

将(3-50)代入(3-47)式$W=X^{\#}B$ 有：

$$W(k+1)=X^{\#}B(k+1)=X^{\#}\{B(k)+c[e(k)+|e(k)|]\}$$

$$=X^{\#}B(k)+\underbrace{X^{\#}ce(k)}_{=0}+X^{\#}c|e(k)|=W(k)+cX^{\#}|e(k)|$$

$$X^{\#}e(k)=\left(X^{T}X\right)^{-1}X^{T}[XW(k)-B(k)]$$

$$=\left(X^{T}X\right)^{-1}X^{T}\left[XX^{\#}B(k)-B(k)\right]=\mathbf{0}$$

推导是错的

$$\boxed{XW(k)-B(k)=e(k)} \qquad (3\text{-}49)$$

$$\boxed{B(k+1)=B(k)+c\left[e(k)+|e(k)|\right] \qquad (3\text{-}50)}$$

总结： 设初值$B(1)$，各分量均为正值，括号中数字代表迭代次数

$$\boldsymbol{W}(1)=\boldsymbol{X}^{\#}\boldsymbol{B}(1)$$

……

$$\boldsymbol{e}(k)=\boldsymbol{X}\boldsymbol{W}(k)-\boldsymbol{B}(k)$$

$$\boldsymbol{W}(k+1)=\boldsymbol{W}(k)+c\boldsymbol{X}^{\#}\left|\boldsymbol{e}(k)\right|$$

$$\boldsymbol{B}(k+1)=\boldsymbol{B}(k)+c\left[\boldsymbol{e}(k)+\left|\boldsymbol{e}(k)\right|\right]$$

$\boldsymbol{W}(k+1)$、$\boldsymbol{B}(k+1)$互相独立，先后次序无关。

或：另一算法：先算$\boldsymbol{B}(k+1)$，再算$\boldsymbol{W}(k+1)$。

$$\boldsymbol{W}(1)=\boldsymbol{X}^{\#}\boldsymbol{B}(1)$$

……

$$\boldsymbol{e}(k)=\boldsymbol{X}\boldsymbol{W}(k)-\boldsymbol{B}(k)$$

$$\boldsymbol{B}(k+1)=\boldsymbol{B}(k)+c\left[\boldsymbol{e}(k)+\left|\boldsymbol{e}(k)\right|\right]$$

$$\boldsymbol{W}(k+1)=\boldsymbol{X}^{\#}\boldsymbol{B}(k+1)$$

求出$\boldsymbol{B}$，$\boldsymbol{W}$后，再迭代出下一个$\boldsymbol{e}$，从而计算出新的$\boldsymbol{B}$，$\boldsymbol{W}$。

3）模式类别可分性判别

可以证明：当模式集线性可分，且校正系数c满足$0 < c \le 1$时，该算法收敛，可求得解W。

理论上不能证明该算法到底需要迭代多少步才能达到收敛，通常在每次迭代计算后检查一下$\boldsymbol{XW}(k)$ 和误差向量$\boldsymbol{e}(k)$ ，从而可以判断是否收敛。

分以下几种情况：

① 如果$\boldsymbol{e}(k)=\boldsymbol{0}$ ，表明$\boldsymbol{XW}(k)=\boldsymbol{B}(k) >\boldsymbol{0}$，有解。

② 如果$\boldsymbol{e}(k)>\boldsymbol{0}$ ，表明$\boldsymbol{XW}(k)>\boldsymbol{B}(k) >\boldsymbol{0}$， 隐含有解。继续迭代，可使 $\boldsymbol{e}(k) \to \boldsymbol{0}$ 。

③ 如果$\boldsymbol{e}(k)<\boldsymbol{0}$（所有分量为负数或零，但不全为零），停止迭代，无解。此时若继续迭代，数据不再发生变化。

情况③分析：

$$\boldsymbol{e}(k)<\mathbf{0}$$

$$\boldsymbol{B}(k+1)=\boldsymbol{B}(k)+c\left[\boldsymbol{e}(k)+\left|\boldsymbol{e}(k)\right|\right]$$

$$\boldsymbol{B}(k+1)=\boldsymbol{B}(k)$$

$$\boldsymbol{W}(k+1)=\boldsymbol{X}^{\#}\boldsymbol{B}(k+1)$$

$$\boldsymbol{W}(k+1)=\boldsymbol{W}(k)$$

$$\boldsymbol{e}(k+1)=\boldsymbol{X}\boldsymbol{W}(k+1)-\boldsymbol{B}(k+1)$$

$$\boldsymbol{e}(k+1)=\boldsymbol{e}(k)$$

综上所述：**只有当$\boldsymbol{e}(k)$中有大于零的分量（说明：含有纠正信息）时，才需要继续迭代**，一旦$\boldsymbol{e}(k)$的全部分量只有0和负数，则立即停止。事实上，往往早在$\boldsymbol{e}(k)$全部分量都达到非正值以前，就能看出其中有些分量向正值变化得极慢，可及早采取对策。

通过反证法可以证明：在线性可分情况下，算法进行过程中不会出现 $\boldsymbol{e}(k)$的分量全为负的情况；若出现$\boldsymbol{e}(k)$的分量全为负，则说明模式类线性不可分。

4）LMS算法描述

(1) 根据N个分属于两类的样本，写出规范化增广样本矩阵$\boldsymbol{X}$。

(2) 求$\boldsymbol{X}$的伪逆矩阵 $\boldsymbol{X}^{\#}=\left(\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}\right)^{-1}\boldsymbol{X}^{\mathrm{T}}$。

(3) 设置初值c和$\boldsymbol{B}(1)$，c为正的校正增量，$\boldsymbol{B}(1)$的各分量大于零，迭代次数k=1 。计算 $\boldsymbol{W}(1)=\boldsymbol{X}^{\#}\boldsymbol{B}(1)$

开始迭代：

(4) 计算 $\boldsymbol{e}(k)=\boldsymbol{X}\boldsymbol{W}(k)-\boldsymbol{B}(k)$，进行可分性判别。

如果$\boldsymbol{e}(k)=\mathbf{0}$，线性可分，解为$\boldsymbol{W}(k)$，算法结束。

如果$\boldsymbol{e}(k)>\mathbf{0}$，线性可分，若进入(5)可使$\boldsymbol{e}(k)\rightarrow\mathbf{0}$，得最优解。

如果$\boldsymbol{e}(k)<\mathbf{0}$，线性不可分，停止迭代，无解，算法结束。

否则，说明$\boldsymbol{e}(k)$的各分量值有正有负，进入(5)。

(5) 计算$\boldsymbol{W}(k+1)$和$\boldsymbol{B}(k+1)$。

方法1：分别计算 $\boldsymbol{W}(k+1)=\boldsymbol{W}(k)+c\boldsymbol{X}^{\#}|\boldsymbol{e}(k)|$

$$\boldsymbol{B}(k+1)=\boldsymbol{B}(k)+c[\boldsymbol{e}(k)+|\boldsymbol{e}(k)|]$$

方法2：先计算 $\boldsymbol{B}(k+1)=\boldsymbol{B}(k)+c[\boldsymbol{e}(k)+|\boldsymbol{e}(k)|]$

再计算 $\boldsymbol{W}(k+1)=\boldsymbol{X}^{\#}\boldsymbol{B}(k+1)$

迭代次数k加1，并进行以下的赋值后返回(4)。

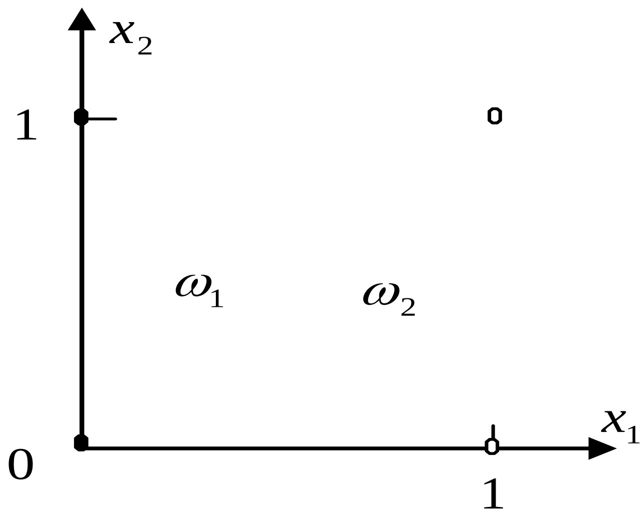
$$W(k)=W(k+1) \quad B(k)=B(k+1)$$

3. 算法特点

(1) 算法尽管略为复杂一些，但提供了线性可分的测试特征。

(2) 同时利用N个训练样本，同时修改$\boldsymbol{W}$和$\boldsymbol{B}$，故收敛速度快。

(3) 计算矩阵$(\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X})^{-1}$复杂，但可用迭代算法计算。



例3.11 已知两类模式训练样本：

$$\omega_1：[0,0]^T,[0,1]^T$$

$$\omega_2：[1,0]^T,[1,1]^T$$

试用LMS算法求解权向量。

解：(1) 写出规范化增广样本矩阵：

$$\boldsymbol{X}=\begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 1 \\ -1 & 0 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

(2) 求伪逆矩阵$\boldsymbol{X}^{\#}=\left(\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}\right)^{-1}\boldsymbol{X}^{\mathrm{T}}$

$$\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}=\begin{bmatrix}0 & 0 & -1 & -1\\ 0 & 1 & 0 & -1\\ 1 & 1 & -1 & -1\end{bmatrix}\begin{bmatrix}0 & 0 & 1\\ 0 & 1 & 1\\ -1 & 0 & -1\\ -1 & -1 & -1\end{bmatrix}=\begin{bmatrix}2 & 1 & 2\\ 1 & 2 & 2\\ 2 & 2 & 4\end{bmatrix}$$

求逆矩阵：$\boldsymbol{A}^{-1}=\dfrac{1}{|\boldsymbol{A}|}\boldsymbol{A}^{*}$

$|\boldsymbol{A}|$——$\boldsymbol{A}$的行列式

$\boldsymbol{A}^{*}$——$\boldsymbol{A}$的伴随矩阵

若 $\boldsymbol{A}=\begin{bmatrix}a_{11} & a_{12} & a_{13}\\ a_{21} & a_{22} & a_{23}\\ a_{31} & a_{32} & a_{33}\end{bmatrix}$，则 $\boldsymbol{A}^{*}=\begin{bmatrix}A_{11} & A_{21} & A_{31}\\ A_{12} & A_{22} & A_{32}\\ A_{13} & A_{23} & A_{33}\end{bmatrix}$

A_{ij}是a_{ij}的代数余子式，注意两者的行和列的标号互换。

代数余子式定义：

划去a_{ij}所在的行和列的元素，余下元素构成的行列式做a_{ij}的余子式，记作M_{ij}，将$(-1)^{i+j}M_{ij}=A_{ij}$叫做元素a_{ij}的代数余子式。例：

$$A_{23}=(-1)^{2+3}\begin{vmatrix} a_{11} & a_{12} \\ a_{31} & a_{32} \end{vmatrix}=-\begin{vmatrix} a_{11} & a_{12} \\ a_{31} & a_{32} \end{vmatrix}$$

$$\boldsymbol{A}=\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}$$

行列式:

$$\left|\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}\right|=\begin{vmatrix} 2 & 1 & 2 \\ 1 & 2 & 2 \\ 2 & 2 & 4 \end{vmatrix}=16+4+4-8-8-4=4$$

$$\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}=\begin{bmatrix} 2 & 1 & 2 \\ 1 & 2 & 2 \\ 2 & 2 & 4 \end{bmatrix}$$

$$\left(\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}\right)^{-1}=\frac{1}{\left|\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}\right|}\begin{bmatrix} 4 & 0 & -2 \\ 0 & 4 & -2 \\ -2 & -2 & 3 \end{bmatrix}=\frac{1}{4}\begin{bmatrix} 4 & 0 & -2 \\ 0 & 4 & -2 \\ -2 & -2 & 3 \end{bmatrix}$$

$$\boldsymbol{A}^{-1}=\frac{1}{|\boldsymbol{A}|}\boldsymbol{A}^{*}$$

$\boldsymbol{X}^{\#}=\left(\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}\right)^{-1}\boldsymbol{X}^{\mathrm{T}}$

$$\boldsymbol{X}^{\#}=\frac{1}{4}\begin{bmatrix}4 & 0 & -2\\ 0 & 4 & -2\\ -2 & -2 & 3\end{bmatrix}\begin{bmatrix}0 & 0 & -1 & -1\\ 0 & 1 & 0 & -1\\ 1 & 1 & -1 & -1\end{bmatrix}=\frac{1}{4}\begin{bmatrix}-2 & -2 & -2 & -2\\ -2 & 2 & 2 & -2\\ 3 & 1 & -1 & 1\end{bmatrix}$$

(3) 取 $\boldsymbol{B}(1)=[1,1,1,1]^{\mathrm{T}}$ 和 c=1 开始迭代：

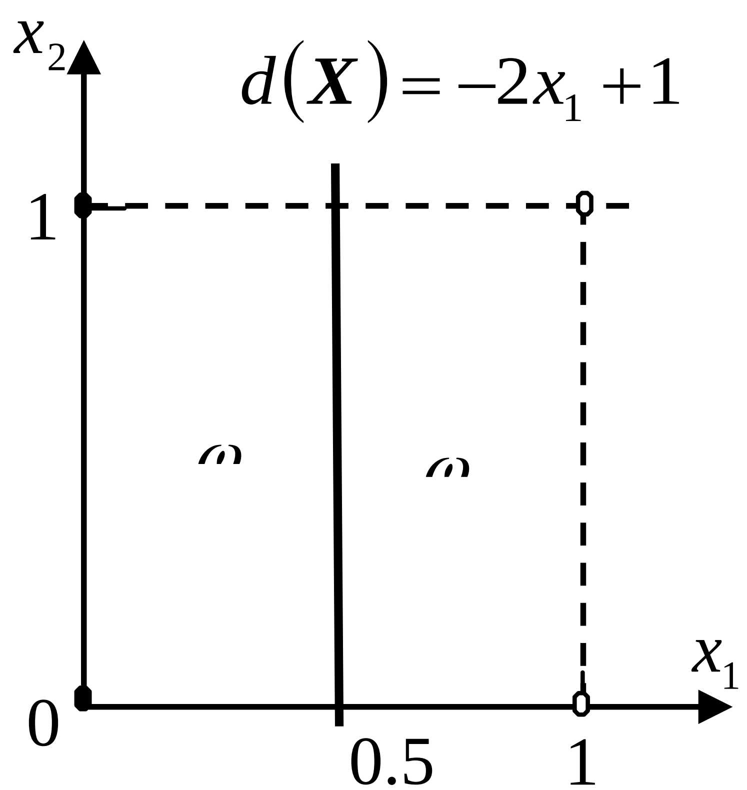
$$\boldsymbol{W}(1)=\boldsymbol{X}^{\#}\boldsymbol{B}(1)=\frac{1}{4}\begin{bmatrix}-2 & -2 & -2 & -2\\ -2 & 2 & 2 & -2\\ 3 & 1 & -1 & 1\end{bmatrix}\begin{bmatrix}1\\1\\1\\1\end{bmatrix}=\frac{1}{4}\begin{bmatrix}-8\\0\\4\end{bmatrix}=\begin{bmatrix}-2\\0\\1\end{bmatrix}$$

.

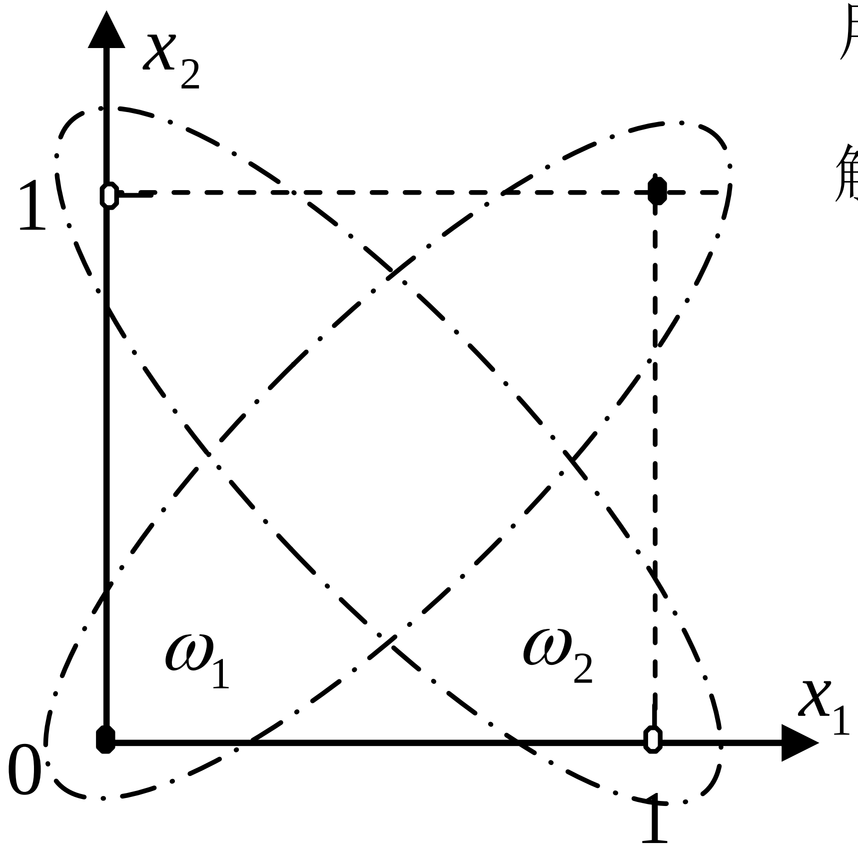
$$\boldsymbol{e}(1)=\boldsymbol{X}\boldsymbol{W}(1)-\boldsymbol{B}(1)=\begin{bmatrix}0 & 0 & 1\\ 0 & 1 & 1\\ -1 & 0 & -1\\ -1 & -1 & -1\end{bmatrix}\begin{bmatrix}-2\\0\\1\end{bmatrix}-\begin{bmatrix}1\\1\\1\\1\end{bmatrix}=\begin{bmatrix}1\\1\\1\\1\end{bmatrix}-\begin{bmatrix}1\\1\\1\\1\end{bmatrix}=\begin{bmatrix}0\\0\\0\\0\end{bmatrix}$$

解为 $\boldsymbol{W}(1)$，判断函数为：$d\left(\boldsymbol{X}\right)=-2x_1+1$

图示如下：



例3.12 已知模式训练样本：ω_1：$[0,0]^T, [1,1]^T$，ω_2：$[0,1]^T$，$[1,0]^T$



用LMS算法求解权向量。

解：(1) 规范化增广样本矩阵：

$$X = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 1 & 1 \\ 0 & -1 & -1 \\ -1 & 0 & -1 \end{bmatrix}$$

(2) 求 $X^{\#} = \left(X^T X\right)^{-1} X^T$：

$$X^{\#} = \frac{1}{4}\begin{bmatrix} -2 & 2 & 2 & -2 \\ -2 & 2 & -2 & 2 \\ 3 & -1 & -1 & -1 \end{bmatrix}$$

(3) 取 $B(1) = [1,1,1,1]^T$ 和 c=1，迭代：

$$\boldsymbol{W}(1)=\boldsymbol{X}^{\#}\boldsymbol{B}(1)=\begin{bmatrix}0 & 0 & 0\end{bmatrix}^{\mathrm{T}}$$

$$\boldsymbol{e}(1)=\boldsymbol{X}\boldsymbol{W}(1)-\boldsymbol{B}(1)=\begin{bmatrix}0 & 0 & 0 & 0\end{bmatrix}^{\mathrm{T}}-\begin{bmatrix}1 & 1 & 1 & 1\end{bmatrix}^{\mathrm{T}}=\begin{bmatrix}-1 & -1 & -1 & -1\end{bmatrix}^{\mathrm{T}}$$

$\boldsymbol{e}(1)$全部分量为负，无解，停止迭代。为线性不可分模式。

小结：

(1) 感知器法、梯度法、最小均方误差算法讨论的分类算法都是通过模式样本来确定判别函数的系数，所以要使一个分类器设计完善，必须采用有代表性的数据，训练判别函数的权系数。它们能合理反映模式数据的总体。

(2) 要获得一个有较好判别性能的线性分类器，所需要的训练样本的数目的确定。

用线性二分法容量N_0来确定训练样本的数目：

$$N_0 = 2(n+1) \qquad n\text{为模式维数}$$

通常训练样本的数目不能低于N_0，选为N_0的5~10倍左右。

如
- 二维：不能低于6个样本，最好选在30~60个样本之间。
- 三维：不能低于8个样本，最好选在40~80个样本之间。

3.9 非线性判别函数（不作要求）

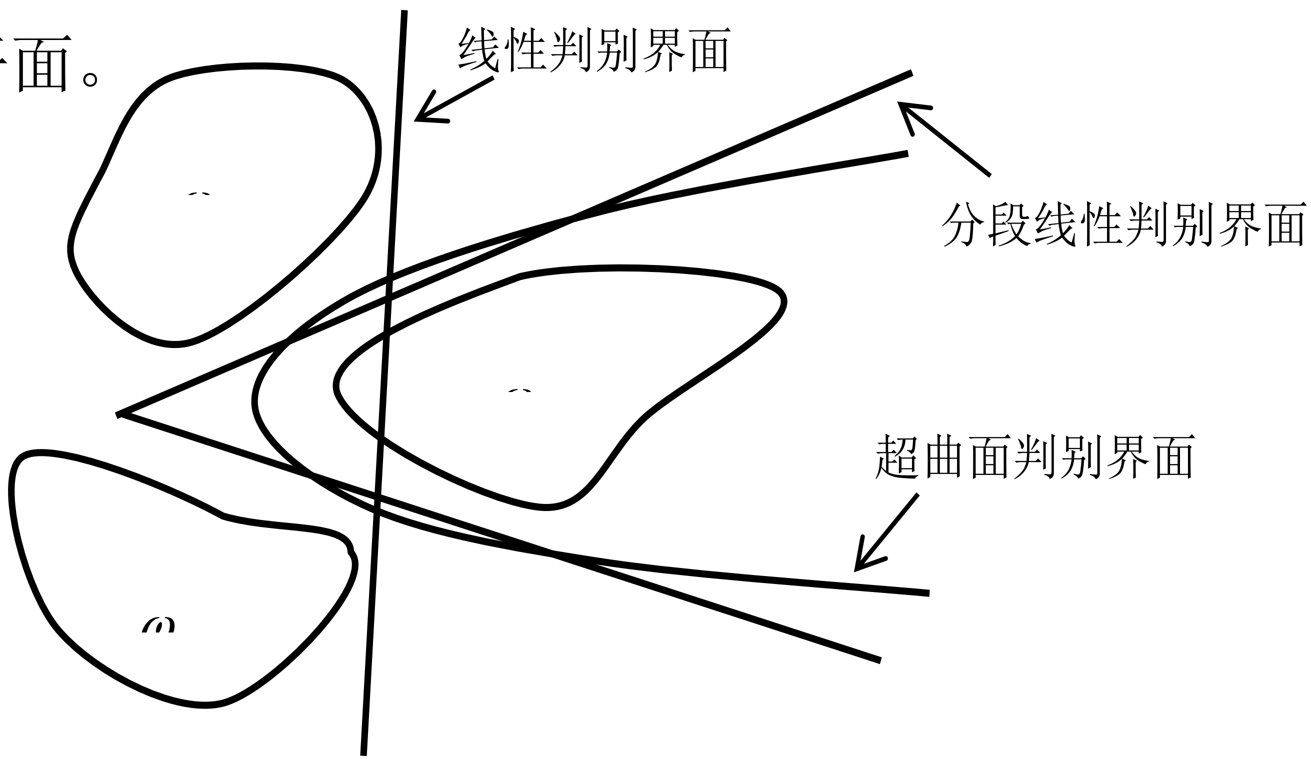
线性判别函数的特点：形式简单，容易学习；

用于线性可分的模式类。

非线性判别函数：用于线性不可分情况。分段线性、超曲面。

3.9.1 分段线性判别函数

基本组成为超平面。



特点

* 相对简单；
* 能逼近各种形状的超曲面。

1．一般分段线性判别函数

设有M类模式，将ω_i类（i=1,2, …,M）划分为l_i个子类：

$$\omega_i:\ \{\omega_i^1, \omega_i^2, \cdots, \omega_i^{l_i}\} \qquad i = 1,2,\cdots,M$$

其中第n个子类的判别函数：

$$d_i^n(\boldsymbol{X}) = (\boldsymbol{W}_i^n)^{\mathrm{T}}\boldsymbol{X} \qquad n = 1,2,\cdots,l_i;\quad i = 1,2,\cdots,M$$

ω_i类的判别函数定义为：

$$d_i(\boldsymbol{X}) = \max\{d_i^n(\boldsymbol{X}), \quad n = 1,2,\cdots,l_i\}$$

M类的判决规则：

若 $d_j(\boldsymbol{X}) = \max\{d_i(\boldsymbol{X}), \quad i = 1,2,\cdots,M\}$，则 $X \in \omega_j$

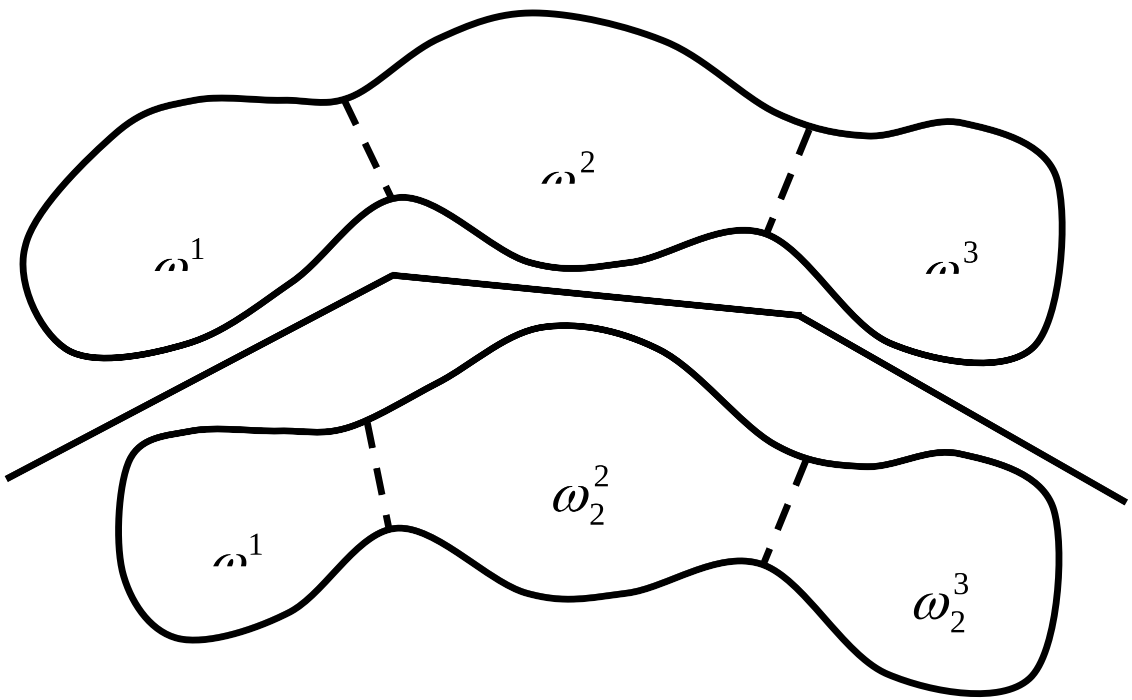
用各类判别函数进行分类判决实际是用各类选出的子类判别函数进行判决 → 判别面由各子类的判别函数决定

若ω_i类的第n个子类和ω_j类的第m个子类相邻，判别界面方程为：

$$d_i^n(\boldsymbol{X}) = d_j^m(\boldsymbol{X})$$

子类之间的判别界面组成各类之间的判别界面

类间判别界面分段线性



2．基于距离的分段线性判别函数

1）最小距离分类器

设 ω_1类均值向量：$\boldsymbol{M}_1 = \dfrac{1}{N_1}\sum_{i=1}^{N_1}\boldsymbol{X}_i$

ω_2类均值向量：$\boldsymbol{M}_2 = \dfrac{1}{N_2}\sum_{i=1}^{N_2}\boldsymbol{X}_i$

N_1，N_2：两类样本数。

任一模式$\boldsymbol{X}$到$\boldsymbol{M}_1$和$\boldsymbol{M}_2$的欧氏距离平方：

$$d_1(\boldsymbol{X}) = \|\boldsymbol{X} - \boldsymbol{M}_1\|^2 \qquad d_2(\boldsymbol{X}) = \|\boldsymbol{X} - \boldsymbol{M}_2\|^2$$

判决规则： 若$d_1(\boldsymbol{X}) < d_2(\boldsymbol{X})$， 则 $X \in \omega_1$

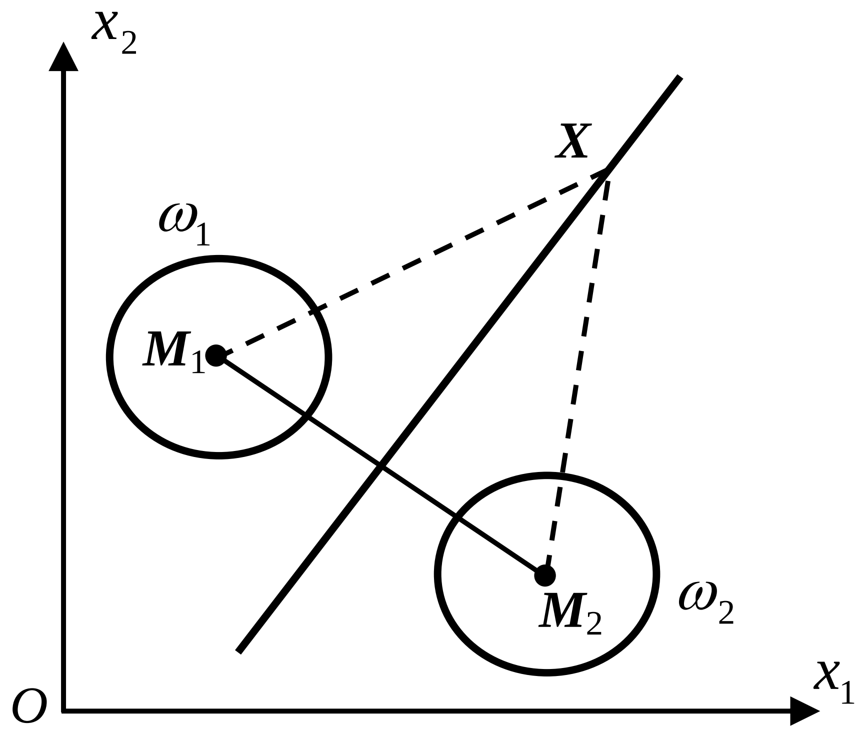
若$d_2(\boldsymbol{X}) < d_1(\boldsymbol{X})$， 则 $X \in \omega_2$

判别界面方程：

$$\|\boldsymbol{X} - \boldsymbol{M}_1\|^2 = \|\boldsymbol{X} - \boldsymbol{M}_2\|^2$$

化简得：$2(\boldsymbol{M}_1-\boldsymbol{M}_2)^{\mathrm{T}}\boldsymbol{X}+(\boldsymbol{M}_2^{\mathrm{T}}\boldsymbol{M}_2-\boldsymbol{M}_1^{\mathrm{T}}\boldsymbol{M}_1)=0$

——X的线性方程，确定一个超平面。



最小距离分类器

2）分段线性距离分类器

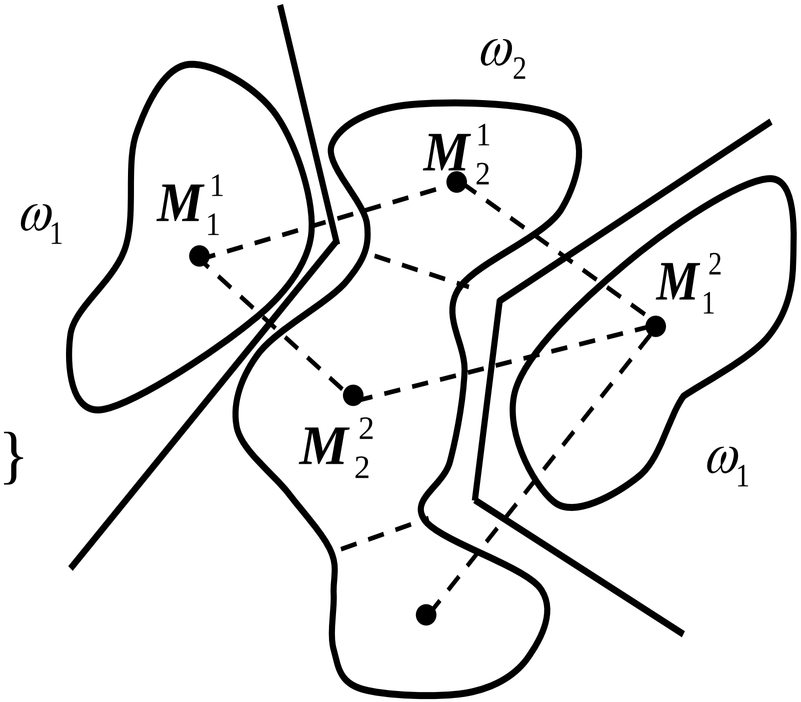
设：M类模式，其中ω_i类划分为l_i个子类，第n个子类的均值向量为 $\boldsymbol{M}_i^n$。每个子类的判别函数：

$$d_i^n(\boldsymbol{X}) = \|\boldsymbol{X} - \boldsymbol{M}_i^n\|^2, \quad n = 1,2,\cdots,l_i; \ i = 1,2,\cdots,M$$

每个类的判别函数（为什么选极小值？）：

$$d_i(\boldsymbol{X}) = \min\{d_i^n(\boldsymbol{X}), \quad n = 1,2,\cdots,l_i\},$$

$$i = 1,2,\cdots,M$$



判决规则：

若 $d_j(\boldsymbol{X}) = \min\{d_i(\boldsymbol{X}), \quad i = 1,2,\cdots,M\}$

则 $\boldsymbol{X} \in \omega_j$

3.9.2 分段线性判别函数的学习方法

1. 已知子类划分时的学习方法

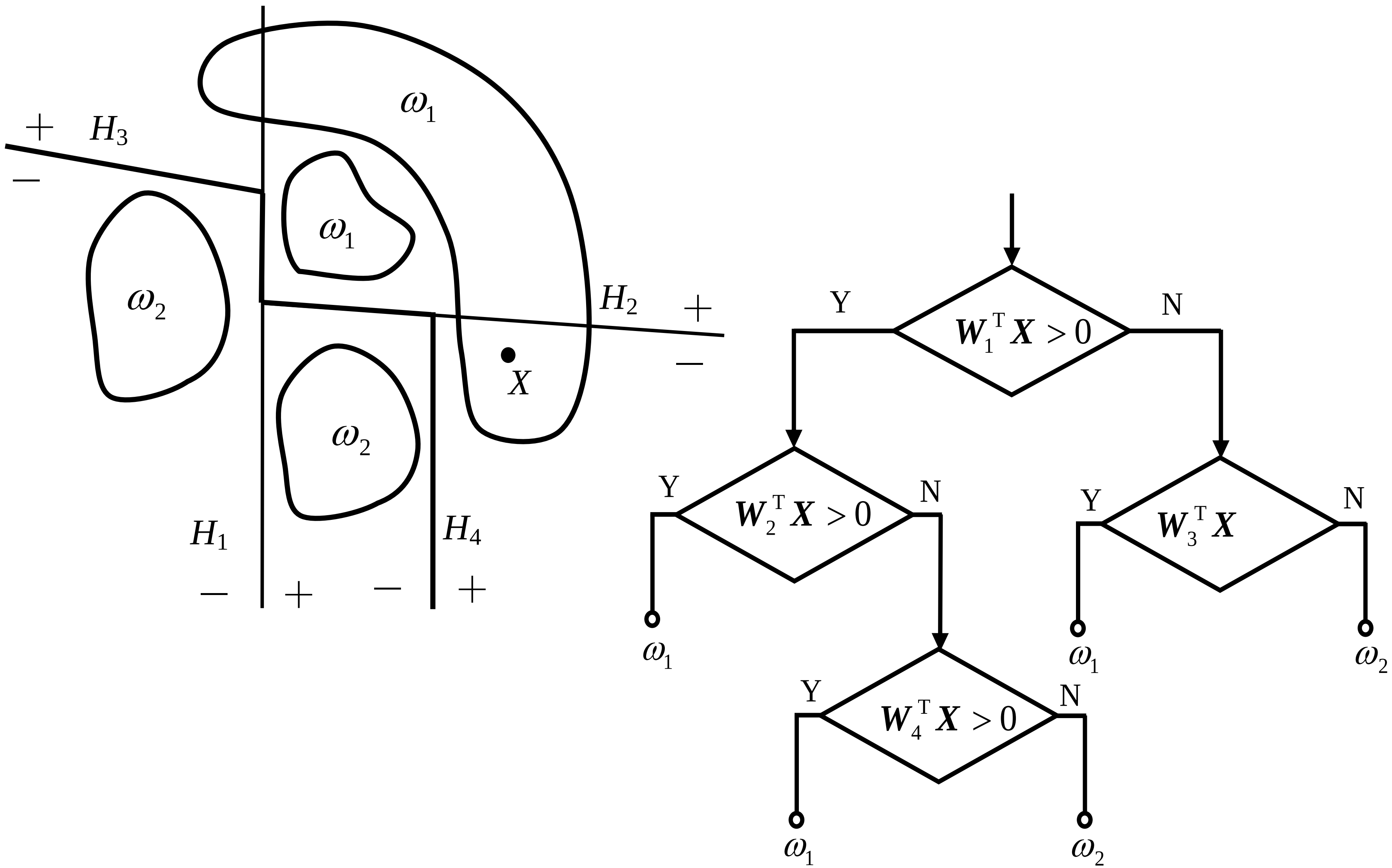
* 每个子类看成独立的类；
* 在一类范围内根据多类情况3，学习各子类判别函数；
* 继而得到各类判别函数。

2. 已知子类数目时的学习方法

用类似于固定增量算法的错误修正算法学习分段线性判别函数

3. 未知子类数目时的学习方法

树状分段线性分类器



树状分段线性分类器判别函数的学习及分类过程

3.9.3 势函数法

1. 势函数概念

借用点能源的势能概念解决模式分类问题。

划分属于ω_1和ω_2类模式样本：

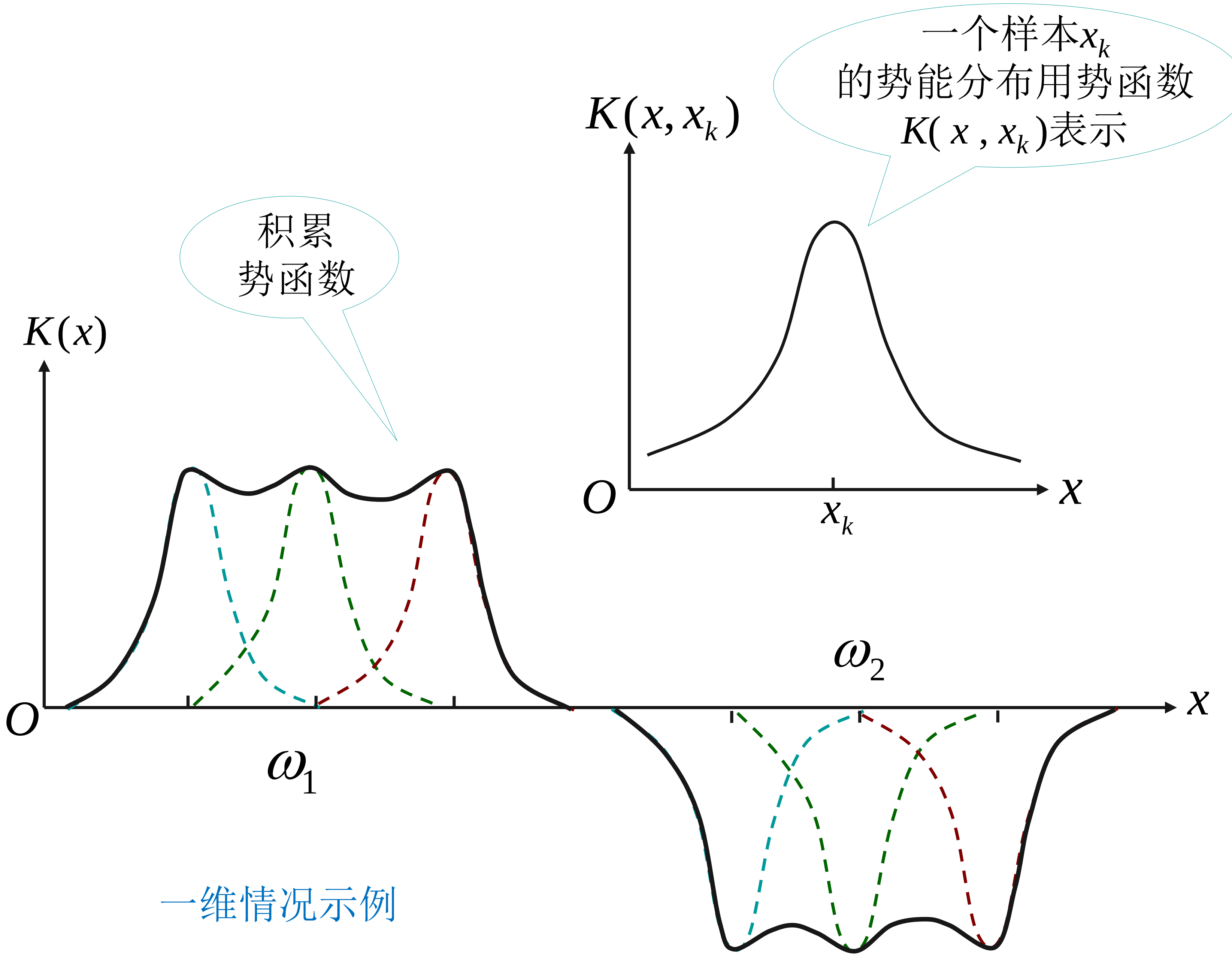
样本是模式空间中的点，

将每个点比拟为点能源，在点上势能达到峰值，随着与该点距离的增大，势能分布迅速减小。

ω_1类样本势能为正——势能积累形成“高地”；

ω_2类样本势能×(-1)——势能积累形成“凹地”；

在两类电势分布之间，选择合适的等势面（如零等势面），即可认为是判别界面了。



一个样本x_k
的势能分布用势函数
$K(x, x_k)$表示
$K(x, x_k)$
O
x
x_k
积累
势函数
$K(x)$
O
ω_1
ω_2
x

一维情况示例

2. 势函数法判别函数的产生

依次输入样本，利用势函数逐步积累势能的过程。

判别函数由模式空间中样本向量

$$\left\{\boldsymbol{X}_k, k=1,2,\cdots \text{且} \boldsymbol{X}_k \in \left(\omega_1 \cup \omega_2\right)\right\}$$

的势函数$K(\boldsymbol{X}, \boldsymbol{X}_k)$累加产生，分类器计算积累势$K(\boldsymbol{X})$，最后取$d(\boldsymbol{X})=K(\boldsymbol{X})$。

势函数法：

设初始积累势函数函数 $K_0(\boldsymbol{X})=0$ ，下标为迭代次数。

第一步：加入训练样本 $\boldsymbol{X}_1$，

$$K_1(\boldsymbol{X})=\begin{cases} K_0(\boldsymbol{X})+K(\boldsymbol{X},\boldsymbol{X}_1), & \text{若 } \boldsymbol{X}_1 \in \omega_1 \\ K_0(\boldsymbol{X})-K(\boldsymbol{X},\boldsymbol{X}_1), & \text{若 } \boldsymbol{X}_1 \in \omega_2 \end{cases}$$

$K_1(\boldsymbol{X})$ 描述了加入第一个样本后的边界划分。

第二步：加第二个训练样本$\boldsymbol{X}_2$，分三种情况：

① 若 $\boldsymbol{X}_2 \in \omega_1$ 且 $K_1(\boldsymbol{X}_2) > 0$

或 $\boldsymbol{X}_2 \in \omega_2$ 且 $K_1(\boldsymbol{X}_2) < 0$

分类正确，势函数不变：$K_2(\boldsymbol{X}) = K_1(\boldsymbol{X})$

② 若 $\boldsymbol{X}_2 \in \omega_1$ 但 $K_1(\boldsymbol{X}_2) \le 0$ ，错误分类，修改势函数：

$$K_2(\boldsymbol{X}) = K_1(\boldsymbol{X}) + K(\boldsymbol{X}, \boldsymbol{X}_2) = \pm K\left(\boldsymbol{X}, \boldsymbol{X}_1\right) + K\left(\boldsymbol{X}, \boldsymbol{X}_2\right)$$

③ 若 $\boldsymbol{X}_2 \in \omega_2$ 但 $K_1(\boldsymbol{X}_2) \ge 0$ ，错误分类，修改势函数：

$$K_2(\boldsymbol{X}) = K_1(\boldsymbol{X}) - K(\boldsymbol{X}, \boldsymbol{X}_2) = \pm K\left(\boldsymbol{X}, \boldsymbol{X}_1\right) - K\left(\boldsymbol{X}, \boldsymbol{X}_2\right)$$

……

第k步：设$K_k(\boldsymbol{X})$ 为加入训练样本$\boldsymbol{X}_1$，$\boldsymbol{X}_2$，…，$\boldsymbol{X}_k$后的积累势函数，则加入第k+1个样本，有：

① 若 $\boldsymbol{X}_{k+1}\in\omega_1$ 且 $K_k(\boldsymbol{X}_{k+1})>0$ 或 $\boldsymbol{X}_{k+1}\in\omega_2$ 且 $K_k(\boldsymbol{X}_{k+1})<0$ 正确分类，$K_{k+1}(\boldsymbol{X})=K_k(\boldsymbol{X})$

② 若 $\boldsymbol{X}_{k+1}\in\omega_1$ 但 $K_k(\boldsymbol{X}_{k+1})\le 0$，错误分类：$K_{k+1}(\boldsymbol{X})=K_k(\boldsymbol{X})+K(\boldsymbol{X},\boldsymbol{X}_{k+1})$

③ 若 $\boldsymbol{X}_{k+1}\in\omega_2$ 但 $K_k(\boldsymbol{X}_{k+1})\ge 0$，错误分类：$K_{k+1}(\boldsymbol{X})=K_k(\boldsymbol{X})-K(\boldsymbol{X},\boldsymbol{X}_{k+1})$

以上决定积累位势的迭代算法可写为：

$$K_{k+1}(\boldsymbol{X})=K_k(\boldsymbol{X})+r_{k+1}K(\boldsymbol{X},\boldsymbol{X}_{k+1}) \tag{3-57}$$

其中r_{k+1} 为校正项系数，定义为：

$$r_{k+1}=\begin{cases}0, & \text{对于}\boldsymbol{X}_{k+1}\in\omega_1\text{且}K_k\left(\boldsymbol{X}_{k+1}\right)>0\\ 0, & \text{对于}\boldsymbol{X}_{k+1}\in\omega_2\text{且}K_k\left(\boldsymbol{X}_{k+1}\right)<0\\ 1, & \text{对于}\boldsymbol{X}_{k+1}\in\omega_1\text{且}K_k\left(\boldsymbol{X}_{k+1}\right)\le 0\\ -1, & \text{对于}\boldsymbol{X}_{k+1}\in\omega_2\text{且}K_k\left(\boldsymbol{X}_{k+1}\right)\ge 0\end{cases} \tag{3-58}$$

$$K_{k+1}(\boldsymbol{X}) = K_k(\boldsymbol{X}) + r_{k+1} K(\boldsymbol{X}, \boldsymbol{X}_{k+1})$$

$$r_{k+1} = \begin{cases} 0, & 对于\boldsymbol{X}_{k+1} \in \omega_1 且 K_k(\boldsymbol{X}_{k+1}) > 0 \\ 0, & 对于\boldsymbol{X}_{k+1} \in \omega_2 且 K_k(\boldsymbol{X}_{k+1}) < 0 \\ 1, & 对于\boldsymbol{X}_{k+1} \in \omega_1 且 K_k(\boldsymbol{X}_{k+1}) \le 0 \\ -1, & 对于\boldsymbol{X}_{k+1} \in \omega_2 且 K_k(\boldsymbol{X}_{k+1}) \ge 0 \end{cases}$$

从所给的训练样本集$\{\boldsymbol{X}_1, \boldsymbol{X}_2, \cdots, \boldsymbol{X}_k, \cdots\}$中略去不使积累势发生变化的那些样本，可得一简化样本序列$\{\hat{\boldsymbol{X}}_1, \hat{\boldsymbol{X}}_2, \cdots, \hat{\boldsymbol{X}}_j, \cdots\}$（校正错误的样本），算法可规纳为：

$$K_{k+1}(\boldsymbol{X}) = \sum_{\hat{X}_j} \alpha_j K(\boldsymbol{X}, \hat{\boldsymbol{X}}_j) \qquad (3\text{-}59)$$

式中，

$$\alpha_j = \begin{cases} 1, & 对于\hat{\boldsymbol{X}}_j \in \omega_1 \\ -1, & 对于\hat{\boldsymbol{X}}_j \in \omega_2 \end{cases} \qquad (3\text{-}60)$$

即：由(k+1)个训练样本产生的积累势，等于两类中校正错误的样本的总势能之差。

$$K_{k+1}(\boldsymbol{X})=K_k(\boldsymbol{X})+r_{k+1}K(\boldsymbol{X},\boldsymbol{X}_{k+1}) \quad (3\text{-}57)$$

从势函数算法可看出，积累势函数起着判别函数的作用，因此可直接用作判别函数，故取 $d(\boldsymbol{X})=K(\boldsymbol{X})$ 。

由(3-57)式得：

$$d_{k+1}(\boldsymbol{X})=d_k(\boldsymbol{X})+r_{k+1}K(\boldsymbol{X},\boldsymbol{X}_{k+1}) \qquad (3\text{-}61)$$

式中r_{k+1}按(3-58)式取值：

$$r_{k+1}=\begin{cases} 0, & \text{对于}\boldsymbol{X}_{k+1}\in\omega_1\text{且}d_k(\boldsymbol{X}_{k+1})>0 \\ 0, & \text{对于}\boldsymbol{X}_{k+1}\in\omega_2\text{且}d_k(\boldsymbol{X}_{k+1})<0 \\ 1, & \text{对于}\boldsymbol{X}_{k+1}\in\omega_1\text{且}d_k(\boldsymbol{X}_{k+1})\le 0 \\ -1, & \text{对于}\boldsymbol{X}_{k+1}\in\omega_2\text{且}d_k(\boldsymbol{X}_{k+1})\ge 0 \end{cases}$$

也可简写成：$r_{k+1}=\dfrac{1}{2}\alpha_{k+1}\{1-\alpha_{k+1}\operatorname{sgn}[d_k(\boldsymbol{X}_{k+1})]\}$

式中 α_{k+1} 取值同(3-60)：$\alpha_{k+1}=\begin{cases} 1, & \text{对于}\hat{\boldsymbol{X}}_j\in\omega_1 \\ -1, & \text{对于}\hat{\boldsymbol{X}}_j\in\omega_2 \end{cases}$

3. 势函数的选择

1）势函数应具备的条件

两个n维向量 $\boldsymbol{X}$ 和 $\boldsymbol{X}_k$ 的函数$K(\boldsymbol{X}, \boldsymbol{X}_k)$，如同时满足下列三个条件，都可做为势函数：

① $K(\boldsymbol{X}, \boldsymbol{X}_k) = K(\boldsymbol{X}_k, \boldsymbol{X})$，当且仅当 $\boldsymbol{X} = \boldsymbol{X}_k$ 时达到最大值。

② 当向量 $\boldsymbol{X}$ 与 $\boldsymbol{X}_k$ 的距离趋于无穷时，$K(\boldsymbol{X}, \boldsymbol{X}_k)$ 趋于零。

③ $K(\boldsymbol{X}, \boldsymbol{X}_k)$ 是光滑函数，且是$\boldsymbol{X}$与$\boldsymbol{X}_k$之间距离的单调下降函数。

2）构成势函数的两种方法

① Ⅰ型势函数：用对称的有限项多项式展开，即：

$$K(\boldsymbol{X}, \boldsymbol{X}_k) = \sum_{i=1}^{m} \varphi_i(\boldsymbol{X}_k)\varphi_i(\boldsymbol{X})$$

式中$\{\varphi_i(\boldsymbol{X}), i = 1,2,\cdots\}$，在模式定义域内应为正交函数集。

“正交函数”概念：已知函数$y(x)$和$z(x)$，

a) 内积：定义为$(y,z)=\int_a^b y(x)z(x)d(x)$，是一个实数。

b) 正交：满足$(y, z)=0$。

例：$\int_{-\pi}^{\pi}\cos x\sin x dx=0$

$d_{k+1}(\boldsymbol{X})=d_k(\boldsymbol{X})+r_{k+1}K(\boldsymbol{X},\boldsymbol{X}_{k+1})$

将这类势函数代入(3-61)式，有判别函数：

$$d_{k+1}(\boldsymbol{X})=d_k(\boldsymbol{X})+r_{k+1}\sum_{i=1}^{m}\varphi_i(\boldsymbol{X}_{k+1})\varphi_i(\boldsymbol{X})$$
$$=d_k(\boldsymbol{X})+\sum_{i=1}^{m}r_{k+1}\varphi_i(\boldsymbol{X}_{k+1})\varphi_i(\boldsymbol{X})$$

② II 型势函数：直接选择双变量 $\boldsymbol{X}$ 和 $\boldsymbol{X}_k$ 的对称函数作为势函数，即$K(\boldsymbol{X},\boldsymbol{X}_k)=K(\boldsymbol{X}_k,\boldsymbol{X})$，如：

$$K(\boldsymbol{X}, \boldsymbol{X}_k) = \exp\left\{-\alpha \left\| \boldsymbol{X} - \boldsymbol{X}_k \right\|^2\right\} \qquad (3\text{-}66)$$

$$K(\boldsymbol{X}, \boldsymbol{X}_k) = \frac{1}{1 + \alpha \left\| \boldsymbol{X} - \boldsymbol{X}_k \right\|^2} \qquad (3\text{-}67)$$

$$K(\boldsymbol{X}, \boldsymbol{X}_k) = \left| \frac{\sin \alpha \left\| \boldsymbol{X} - \boldsymbol{X}_k \right\|^2}{\alpha \left\| \boldsymbol{X} - \boldsymbol{X}_k \right\|^2} \right| \qquad (3\text{-}68)$$

式中α为正常数。

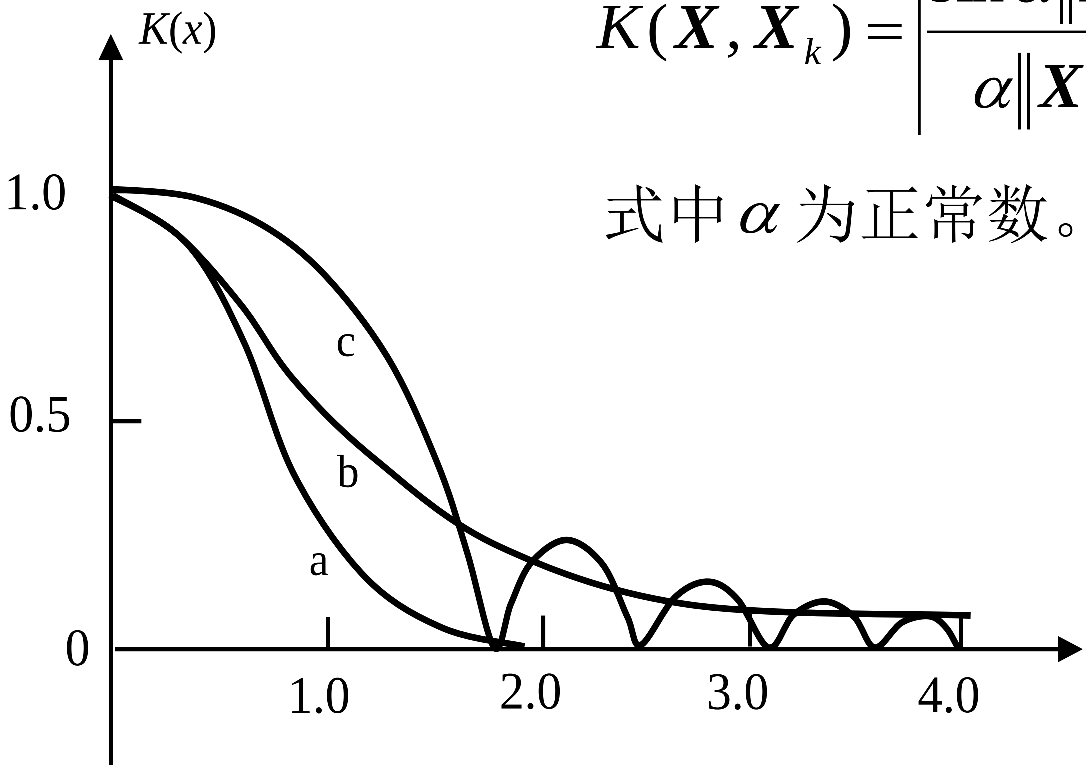


图3.25 一维Ⅱ型势函数举例

曲线a —— (3-66)

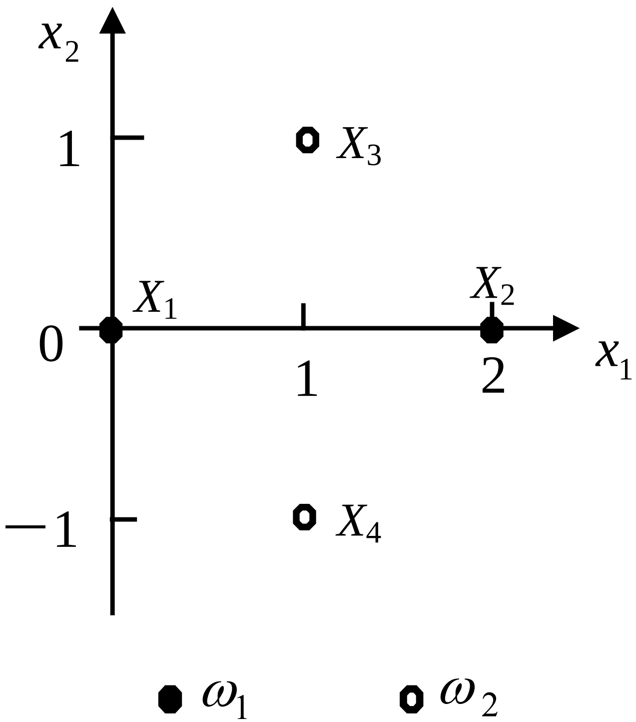
曲线b —— (3-67)

曲线c —— (3-68)

曲线 c 含有正弦函数，具有振荡特点，只有第一个振荡周期可用。

例3.14 设两类训练样本集 ω_1：$\boldsymbol{X}_1=[0,0]^{\mathrm{T}}$， $\boldsymbol{X}_2=[2,0]^{\mathrm{T}}$

ω_2：$\boldsymbol{X}_3=[1,1]^{\mathrm{T}}$， $\boldsymbol{X}_4=[1,-1]^{\mathrm{T}}$



样本分布如图所示。用 II 型势函数进行分类，求判别函数。

解：两类模式不是线性可分的，这里选择指数型的势函数，$\alpha=1$。二维情况下势函数为：

$$K(X,X_k)=\exp\left\{-\|X-X_k\|^2\right\}=\exp\{-[(x_1-x_{k1})^2+(x_2-x_{k2})^2]\}$$

式中，$\boldsymbol{X}=[x_1,x_2]^{\mathrm{T}}$，$\boldsymbol{X}_k=[x_{k1},x_{k2}]^{\mathrm{T}}$

开始迭代：

$$K(\boldsymbol{X},\boldsymbol{X}_k)=\exp\left\{-\left\|\boldsymbol{X}-\boldsymbol{X}_k\right\|^2\right\}=\exp\{-[(x_1-x_{k1})^2+(x_2-x_{k2})^2]\}$$

第一步： $\because \boldsymbol{X}_1=[0,0]^{\mathrm{T}}\in\omega_1$

$$\therefore K_1(\boldsymbol{X})=K(\boldsymbol{X},\boldsymbol{X}_1)=\exp\left\{-[(x_1-0)^2+(x_2-0)^2]\right\}=\exp\left\{-\left[x_1^2+x_2^2\right]\right\}$$

第二步： $\because \boldsymbol{X}_2=[2,0]^{\mathrm{T}}\in\omega_1 \quad K_1(\boldsymbol{X}_2)=\exp\{-(4+0)\}=e^{-4}>0$

分类正确,不修正。

$$\therefore K_2(\boldsymbol{X})=K_1(\boldsymbol{X})=\exp\left\{-[x_1^2+x_2^2]\right\}$$

第三步： $\because \boldsymbol{X}_3=[1,1]^{\mathrm{T}}\in\omega_2 \quad K_2(\boldsymbol{X}_3)=e^{-2}>0$，分类错误，修正。

$$\therefore K_3(\boldsymbol{X})=K_2(\boldsymbol{X})-K(\boldsymbol{X},\boldsymbol{X}_3)=\exp\left\{-[x_1^2+x_2^2]\right\}-\exp\{-[(x_1-1)^2+(x_2-1)^2]\}$$

第四步： $\because \boldsymbol{X}_4=[1,-1]^{\mathrm{T}}\in\omega_2 \quad K_3(\boldsymbol{X}_4)=e^{-(1+1)}-e^{-(0+4)}=e^{-2}-e^{-4}>0$

分类错误，修正。

$$\therefore K_4(\boldsymbol{X})=K_3(\boldsymbol{X})-K(\boldsymbol{X},\boldsymbol{X}_4)$$

$$=\exp\left\{-[x_1^2+x_2^2]\right\}-\exp\{-[(x_1-1)^2+(x_2-1)^2]\}-\exp\{-[(x_1-1)^2+(x_2+1)^2]\}$$

第五步：$\because \boldsymbol{X}_5 = \boldsymbol{X}_1 = [0,0]^{\mathrm{T}} \in \omega_1 \quad K_4(\boldsymbol{X}_5) = e^0 - e^{-2} - e^{-2} > 0$，不修正。

$$\therefore K_5(\boldsymbol{X}) = K_4(\boldsymbol{X})$$

第六步：$\because \boldsymbol{X}_6 = \boldsymbol{X}_2 = [2,0]^{\mathrm{T}} \in \omega_1 \quad K_5(\boldsymbol{X}_6) = e^{-4} - e^{-2} - e^{-2} < 0$， 修正。

$$\therefore K_6(\boldsymbol{X}) = K_5(\boldsymbol{X}) + K(\boldsymbol{X}, \boldsymbol{X}_6)$$

$$= \exp\left\{-[x_1^2 + x_2^2]\right\} - \exp\{-[(x_1-1)^2 + (x_2-1)^2]\}$$

$$- \exp\{-[(x_1-1)^2 + (x_2+1)^2]\} + \exp\{-[(x_1-2)^2 + x_2^2]\}$$

第七步：

$\because \boldsymbol{X}_7 = \boldsymbol{X}_3 = [1,1]^{\mathrm{T}} \in \omega_2 \quad K_6(\boldsymbol{X}_7) = e^{-2} - e^0 - e^{-4} + e^{-2} = 2e^{-2} - 1 - e^{-4} < 0$

不修正。 $\therefore K_7(\boldsymbol{X}) = K_6(\boldsymbol{X})$

第八步：

$\because \boldsymbol{X}_8 = \boldsymbol{X}_4 = [1,-1]^{\mathrm{T}} \in \omega_2 \quad K_7(\boldsymbol{X}_8) = e^{-2} - e^{-2} - e^0 + e^{-2} = e^{-2} - 1 < 0$

不修正。 $\therefore K_8(\boldsymbol{X}) = K_7(\boldsymbol{X})$

第七步：$K_7(\boldsymbol{X})=K_6(\boldsymbol{X})$　　第八步：$K_8(\boldsymbol{X})=K_7(\boldsymbol{X})$

第九步：$\because \boldsymbol{X}_9=\boldsymbol{X}_1=[0,0]^{\mathrm{T}}\in\omega_1$　$K_8(\boldsymbol{X}_9)=e^0-e^{-2}-e^{-2}+e^{-4}>0$

不修正。$\therefore K_9(\boldsymbol{X})=K_8(\boldsymbol{X})$

第十步：$\because \boldsymbol{X}_{10}=\boldsymbol{X}_2=[2,0]^{\mathrm{T}}\in\omega_1$　$K_9(\boldsymbol{X}_8)=e^{-4}-e^{-2}-e^{-2}+e^0>0$

不修正。$\therefore K_{10}(\boldsymbol{X})=K_9(\boldsymbol{X})$

从$\boldsymbol{X}_7$至$\boldsymbol{X}_{10}$的四次迭代中，所有训练样本皆被正确分类，故算法已收敛于判别函数，分类器设计完毕。

判别函数：

$$d(\boldsymbol{X})=K_{10}(\boldsymbol{X})$$

$$=\exp\left\{-[x_1^2+x_2^2]\right\}-\exp\{-[(x_1-1)^2+(x_2-1)^2]\}$$

$$-\exp\{-[(x_1-1)^2+(x_2+1)^2]\}+\exp\{-[(x_1-2)^2+x_2^2]\}$$

判别界面：$d(\boldsymbol{X})=0$

用Ⅱ型势函数构成判别函数的特点：

当训练样本的维数和数目较高时，需要计算和存储更多的指数项，但正因为判别函数由许多新项组成，故有很强的分类能力。

课后作业

- 见另文。
- 下次上课前提交。
- 最好使用电子档。

End of This Part