第四章
判别阈方程

任课教师：柳欣老师

email：starxliu@163.com

4.1 判别域代数界面方程法

有监督分类

1 用判别域界面方程分类的概念
2 线性判别函数
3 判别函数值的鉴别意义、权空间及解空间
4 Fisher线性判别
5 一次准则函数及梯度下降法
6 二次准则函数及其解法
9 广义线性判别函数

4.1 判别域代数界面方程法

1 用判别域界面方程分类的概念

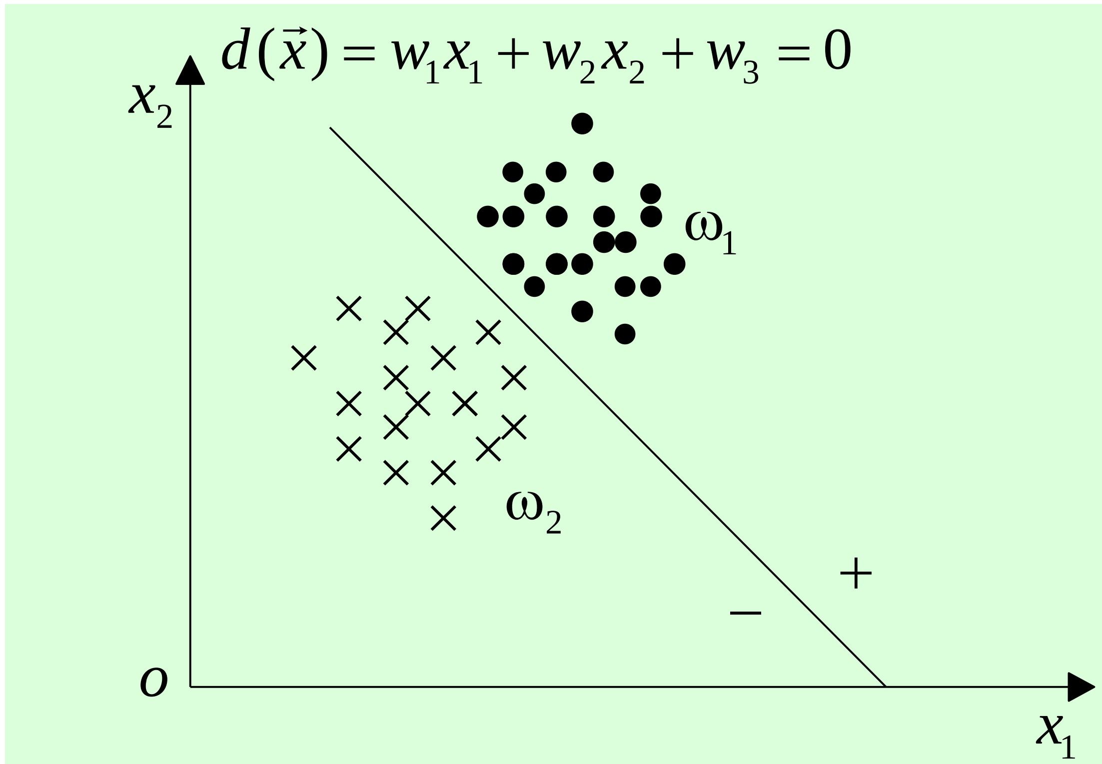
1. 分类的基本原理

不同模式对应特征点在不同的区域中散布。运用已知类别的训练样本进行学习，产生若干个代数界面 $d(\bar{x})=0$，将特征空间划分成一些互不重叠的子区域。

2. 判别函数

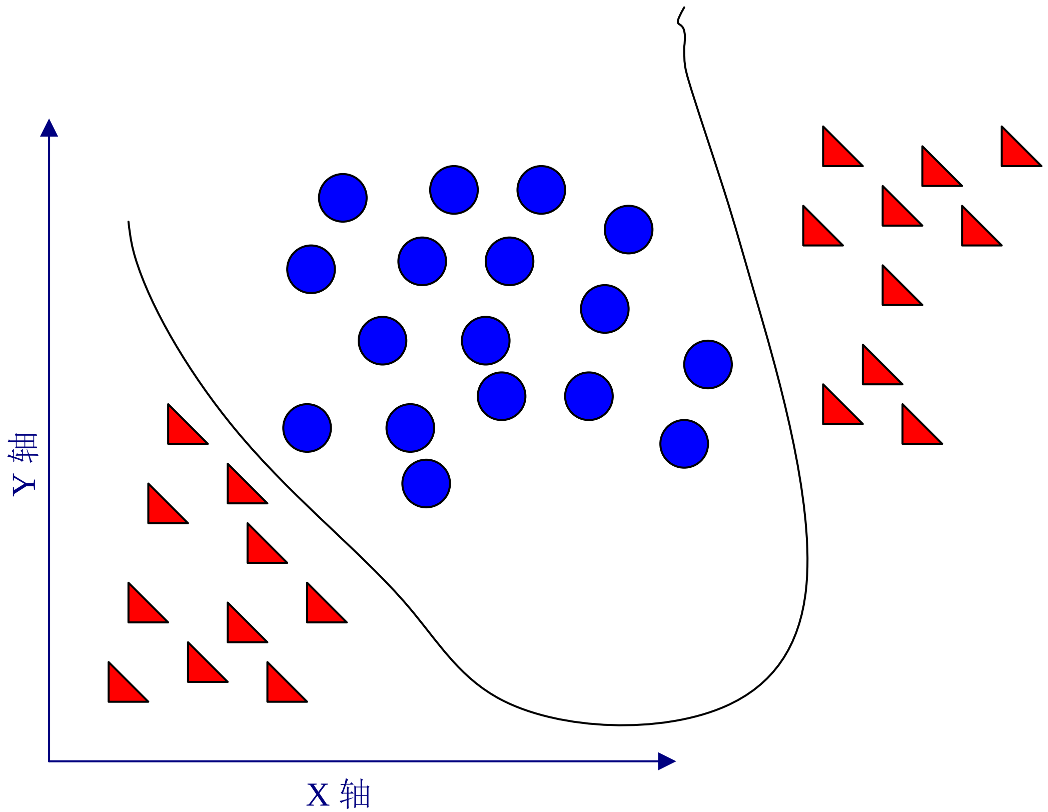
表示界面的函数 $d(\bar{x})$ 称为判别函数(Discriminant Function)。

4.1 判别域代数界面方程法



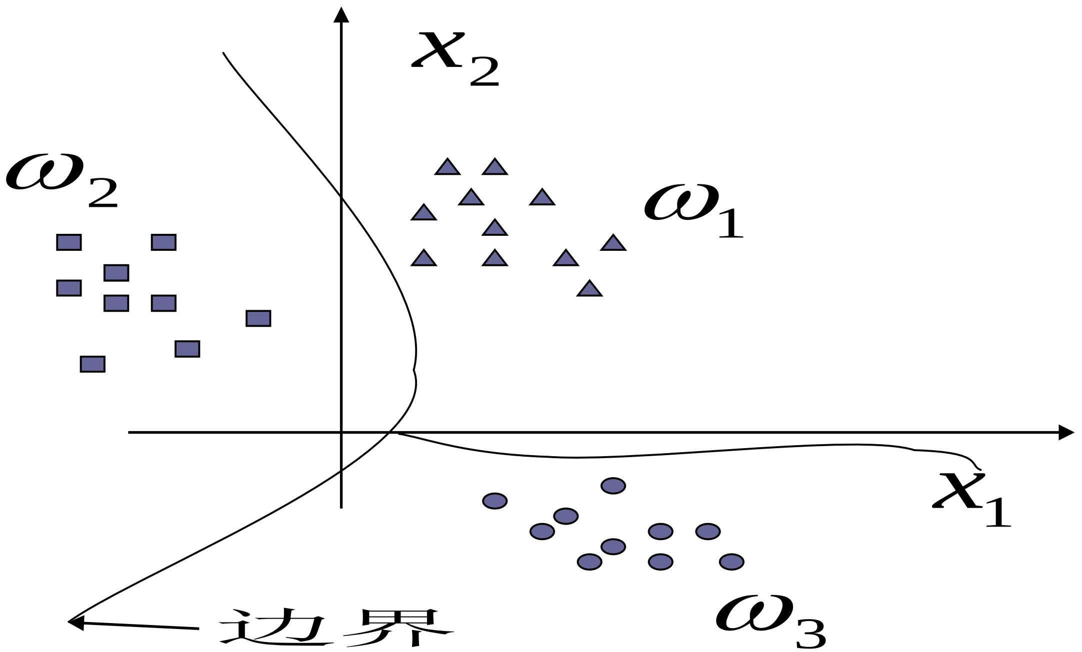
两类的分类问题，它们的边界线就是一个判别函数

4.1 判别域代数界面方程法



两类问题中线性不可分的实例

4.1 判别域代数界面方程法



三类的分类问题，它们的边界线也是一个判别函数

4.1 判别域代数界面方程法

3. 线性可分的定义

对于来自两类的一组模式 $\vec{x}_1, \vec{x}_2, \ldots. \vec{x}_N$, 如果能用一个线性判别函数正确分类, 则称他们是线性可分的。

4. 本章分类方法的基本技术思路

第一步：利用训练样本求出分类器/判别函数

第二步：利用判别函数对未知类别样本分类

4. 1 判别域代数界面方程法

将二维模式推广到n维，线性判别函数的一般形式为：

$$d(\boldsymbol{X}) = w_1x_1 + w_2x_2 + \ldots + w_\mathrm{n}x_\mathrm{n} + w_{\mathrm{n+1}} = \boldsymbol{W}_0^{\mathrm{T}}\boldsymbol{X} + w_{n+1} \qquad (3\text{-}2)$$

式中：$\boldsymbol{X} = [x_1, x_2, \ldots, x_n]^{\mathrm{T}}$

$\boldsymbol{W}_0 = [w_1, w_2, \ldots, w_n]^{\mathrm{T}}$：权向量，即参数向量。

$$d(\boldsymbol{X}) = w_1x_1 + w_2x_2 + \ldots + w_\mathrm{n}x_\mathrm{n} + w_{\mathrm{n+1}} \cdot 1$$

$$= \begin{bmatrix} w_1 & w_2 & \cdots\cdots w_n & w_{n+1} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \\ 1 \end{bmatrix} = \boldsymbol{W}^{\mathrm{T}}\boldsymbol{X}$$

式中：

$\boldsymbol{X} = [x_1, x_2, \ldots, x_n, 1]^{\mathrm{T}}$

为增广模式向量。

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

为增广权向量，

1. 两类情况

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

$d(\boldsymbol{X})=0$：不可判别情况，可以$\boldsymbol{X}\in\omega_1$或$\boldsymbol{X}\in\omega_2$或拒绝

2. 多类情况

对M个线性可分模式类，ω_1，ω_2，… ω_M，有三种划分方式：

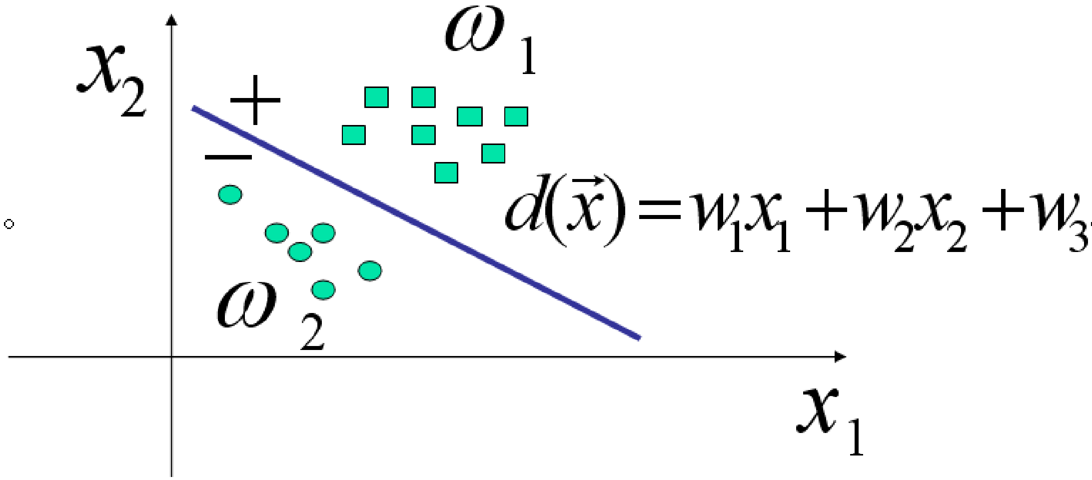
$\omega_i/\overline{\omega}_i$ 两分法　　ω_i/ω_j 两分法　　ω_i/ω_j 两分法特例

4. 1 判别域代数界面方程法

一、两类问题

对于两类问题，待识别模式增广特征矢量 $\vec{x}$ 可通过下面的判别规则进行分类识别：设 $d(\vec{x})$ 为判别函数，

$$d(\vec{x}) = \vec{w}'\vec{x} \begin{cases} > 0 \Rightarrow \vec{x} \in \omega_1 \\ < 0 \Rightarrow \vec{x} \in \omega_2 \\ = 0 \Rightarrow \vec{x} \in \omega_i \text{或拒判} \end{cases}$$



4.2 判别域代数界面方程多类法

二、多类问题

处理多类问题主要有以下几种方法:

1. $\omega_i/\overline{\omega}_i$ 两分法（第一种情况）

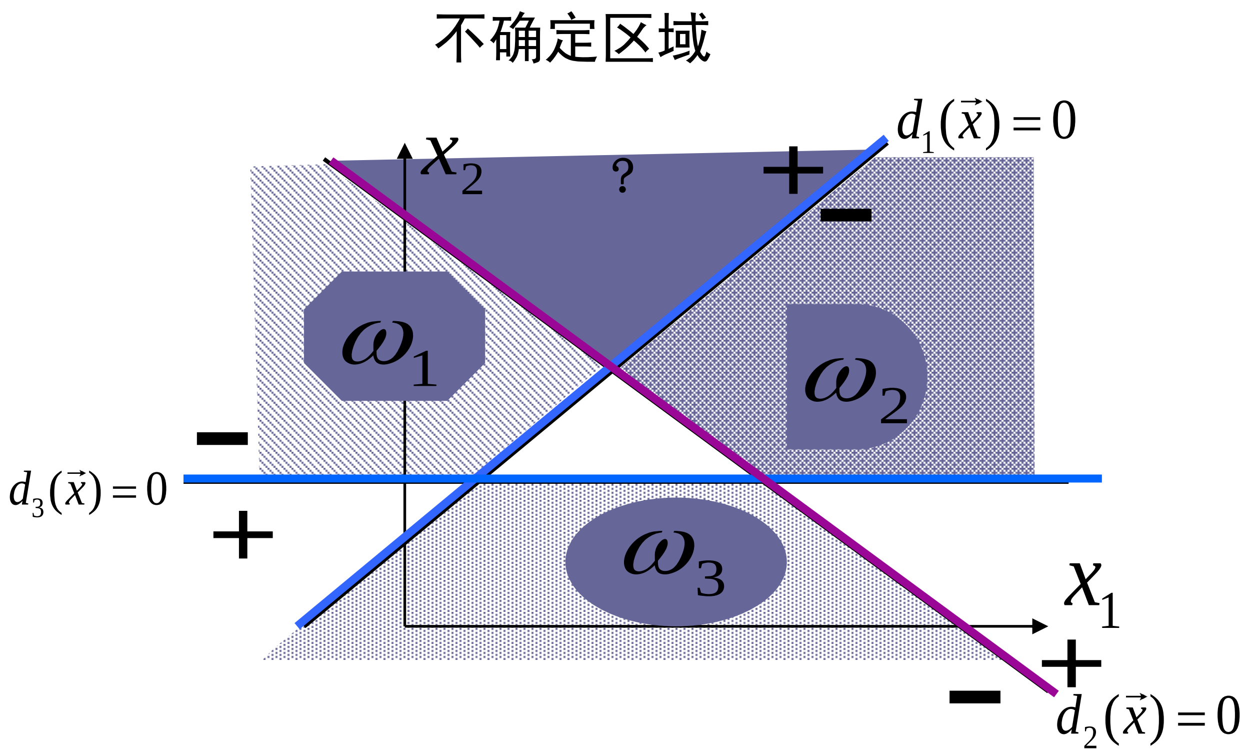
基本思想:将属于ω_i类和不属于ω_i类的模式分划开。c类问题转化为$c-1$个两类问题。可建立c个判别函数。

$$d_i(\vec{x}) = \vec{w}_i'\vec{x} \qquad (i = 1, 2, \cdots, c)$$

通过训练, 其中每个判别函数都具有下面的性质:

$$d_i(\vec{x}) \begin{cases} > 0, 若\vec{x} \in \omega_i, \\ < 0, 若\vec{x} \in \overline{\omega}_i, \end{cases} \qquad (i = 1, 2, \cdots, c)$$

4.2 判别域代数界面方程多类法



多类问题图例（第一种情况）

1、第一种情况（续）

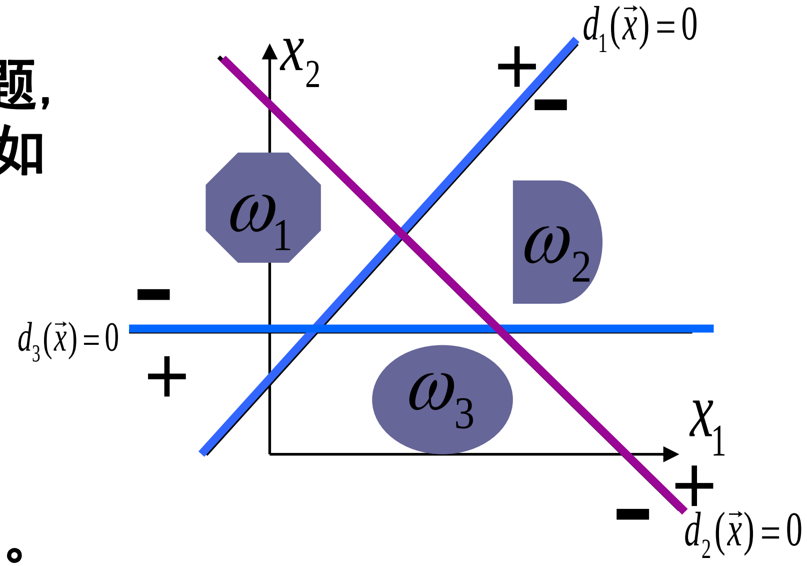
判别规则为：

如果 $\begin{cases} d_i(\vec{x}) > 0 \\ d_j(\vec{x}) \le 0 \end{cases} \quad \forall j \neq i$ 则判 $\vec{x} \in \omega_i$

比如对图的三类问题，如果对于任一模式 $\vec{x}$ 如果它的

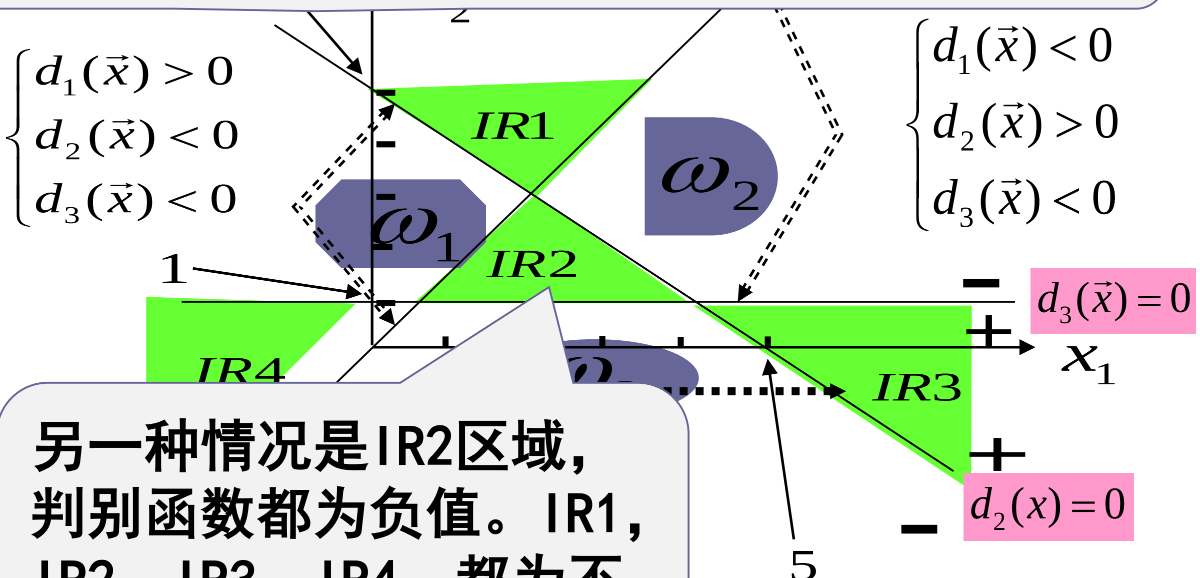
$$d_1(\vec{x}) > 0$$
$$d_2(\vec{x}) < 0$$
$$d_3(\vec{x}) < 0$$

则该模式属于 ω_1 类。



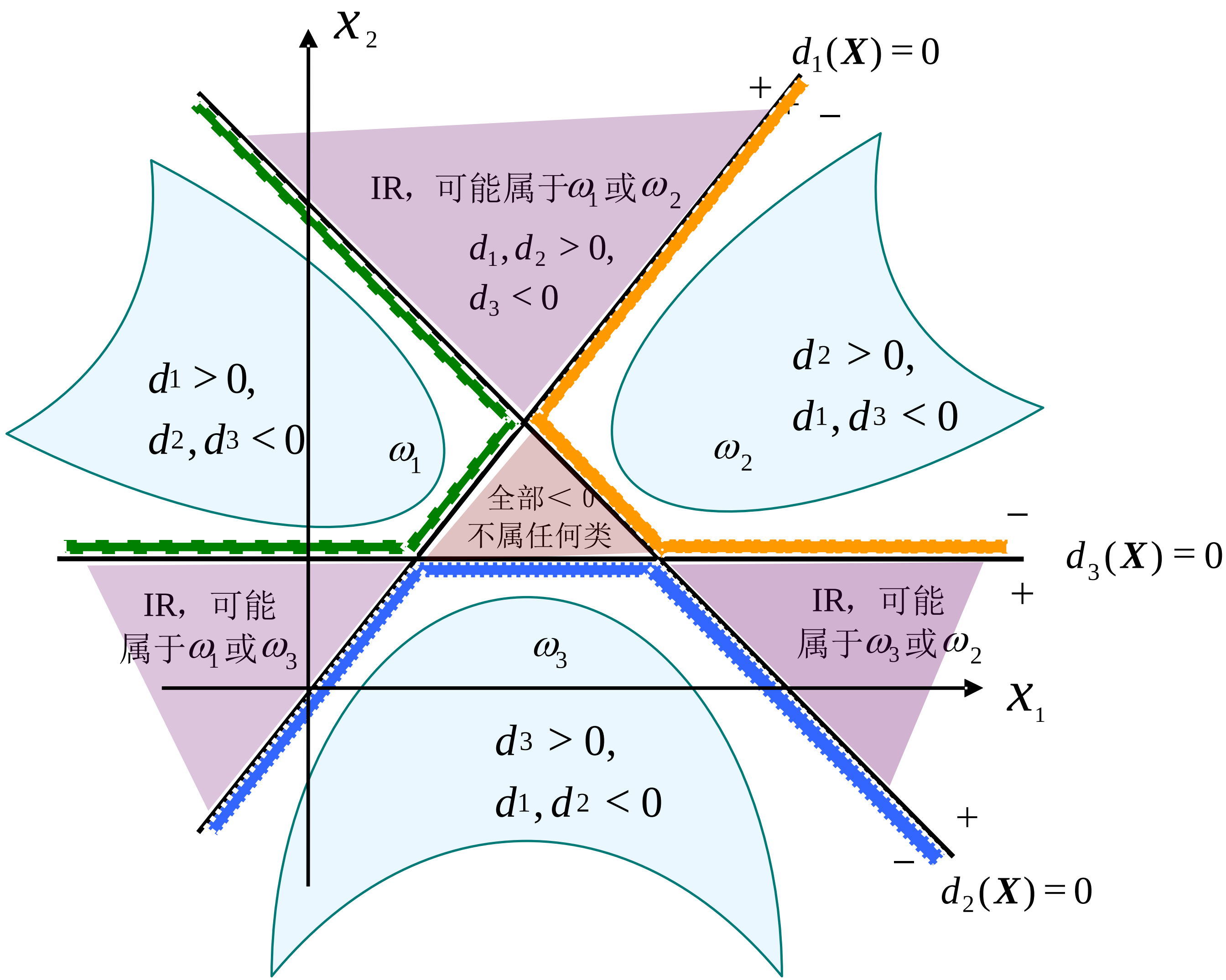
1、第一种情况（续）

如果某个X使二个以上的判别函数 $d_i>0$ 。则此模式X就无法作出确切的判决。如图



另一种情况是IR2区域，判别函数都为负值。IR1，IR2，IR3，IR4。都为不确定区域。

对某一模式区，$d_i(\boldsymbol{X})>0$的条件超过一个，或全部的$d_i(\boldsymbol{X})<0$，分类失效。相当于不确定区(indefinite region，IR)。



此法将 M 个多类问题分成M个两类问题，识别每一类均需M个判别函数。识别出所有的M类仍是这M个函数。

1、第一种情况（续）

例　　已知三类ω$_1$，ω$_2$，ω$_3$的判别函数分别为：

$$\begin{cases} d_1(\vec{x}) = -x_1 + x_2 \\ d_2(\vec{x}) = x_1 + x_2 - 5 \\ d_3(\vec{x}) = -x_2 + 1 \end{cases}$$

求：当 $\vec{x} = (x_1, x_2)' = (6, 5)'$ 时属于哪一类?

解：　三个判别边界分别为：

$$\begin{cases} d_1(\vec{x}) = -x_1 + x_2 = 0 \\ d_2(\vec{x}) = x_1 + x_2 - 5 = 0 \\ d_3(\vec{x}) = -x_2 + 1 = 0 \end{cases}$$

1、第一种情况（续）

将 $\vec{x}=(x_1,x_2)'=(6,5)'$ 代入方程组:

$$\begin{cases} d_1(\vec{x}) = -x_1 + x_2 \\ d_2(\vec{x}) = x_1 + x_2 - 5 \\ d_3(\vec{x}) = -x_2 + 1 \end{cases}$$

得：

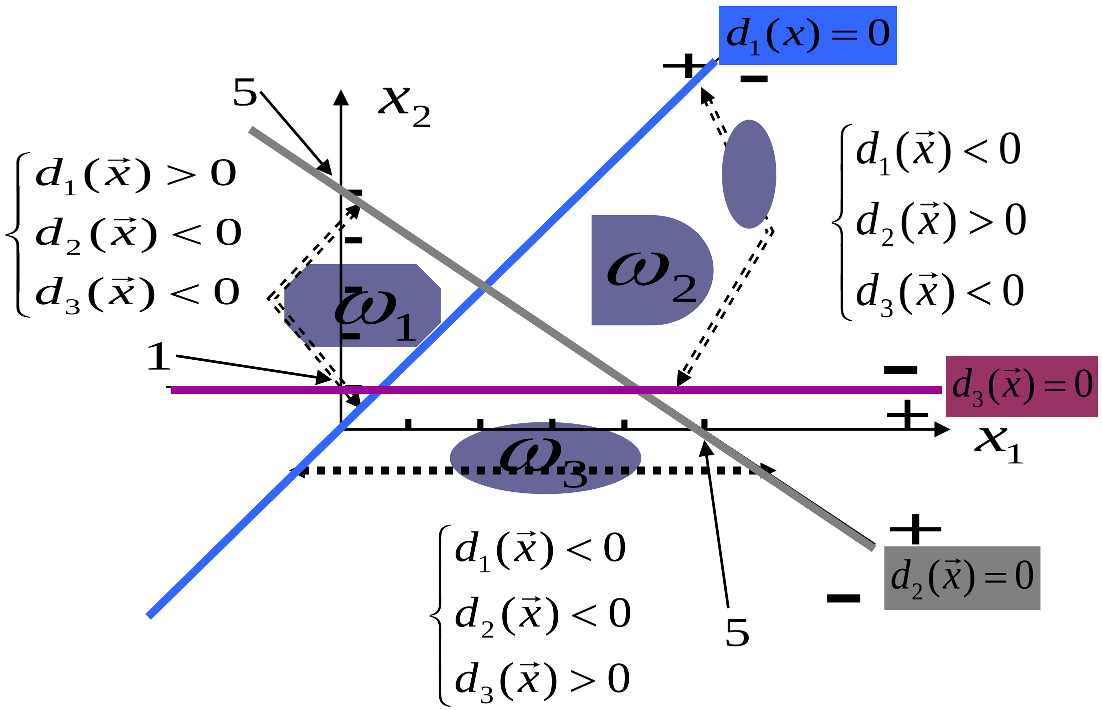
$$d_1(\vec{x}) = -1, d_2(\vec{x}) = 6, d_3(\vec{x}) = -4.$$

结论：　因为

$$d_1(x)<0, d_2(x)>0, d_3(x)<0$$

所以它属于ω_2类。

1、第一种情况（续）



$d_1(x)=0$
x_2
x_1
5
1
5
$\begin{cases} d_1(\vec{x})>0 \\ d_2(\vec{x})<0 \\ d_3(\vec{x})<0 \end{cases}$
$\begin{cases} d_1(\vec{x})<0 \\ d_2(\vec{x})>0 \\ d_3(\vec{x})<0 \end{cases}$
$\begin{cases} d_1(\vec{x})<0 \\ d_2(\vec{x})<0 \\ d_3(\vec{x})>0 \end{cases}$
ω_1
ω_2
ω_3
$d_3(\vec{x})=0$
$d_2(\vec{x})=0$
+
−

此类方法同样存在不确定区。

2. ω_i/ω_j 两分法（第二种情况）

对 c 类中的任意两类 ω_i 和 ω_j 都分别建立一个判别函数，这个判别函数将属于ω_i 的模式与属于ω_j 的模式区分开。此函数对其他模式分类不提供信息，因此总共需要$c(c-1)/2$个这样的判别函数。

通过训练得到区分两类 ω_i 和 ω_j 的判别函数为：

$$d_{ij}(\vec{x}) = \vec{w}_{ij}^{'}\vec{x} \qquad (i, j = 1, 2, \cdots, c; i \neq j)$$

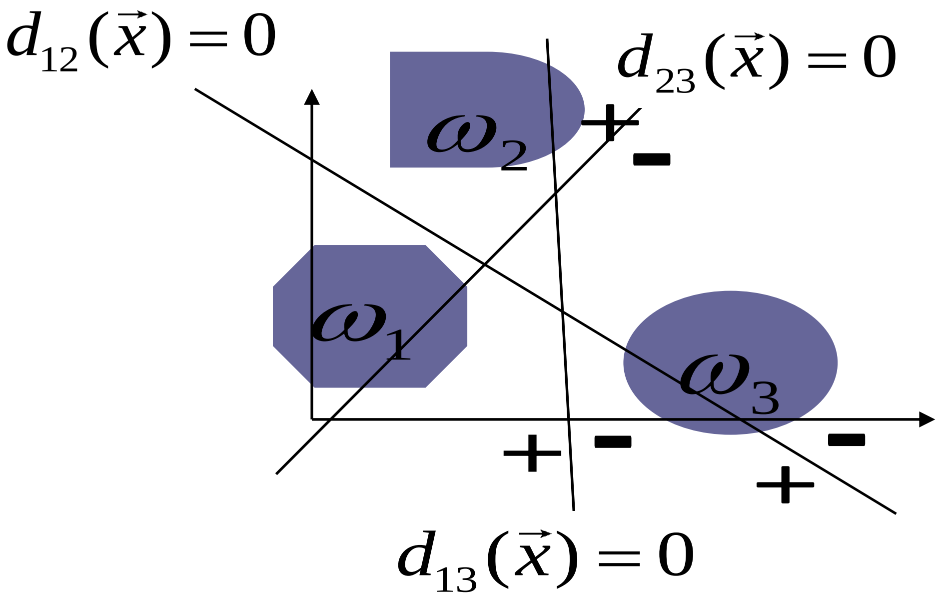
它具有性质：
$$d_{ij}(\vec{x}) = \vec{w}_{ij}^{'}\vec{x} \begin{cases} > 0, \text{若} \vec{x} \in \omega_i \\ < 0, \text{若} \vec{x} \in \omega_j \end{cases}$$

$$d_{ij}(\vec{x}) = -d_{ji}(\vec{x})$$

判别规则是：

如果： $d_{ij}(x) > 0, \forall j \neq i$ 则判 $x \in \omega_i$

2、第二种情况（续）



多类问题图例（第二种情况）

例：设有一个二维三类问题，三个判别函数为：

$d_{12}(\vec{x}) = -x_1 - x_2 + 5 \quad d_{13}(\vec{x}) = -x_1 + 3 \quad d_{23}(\vec{x}) = -x_1 + x_2$

求模式 $\vec{x} = (4,3)'$ 属于哪一类?

解：将模式 $\vec{x} = (4,3)'$ 代入 $d_{ij}(\vec{x})$ 有：

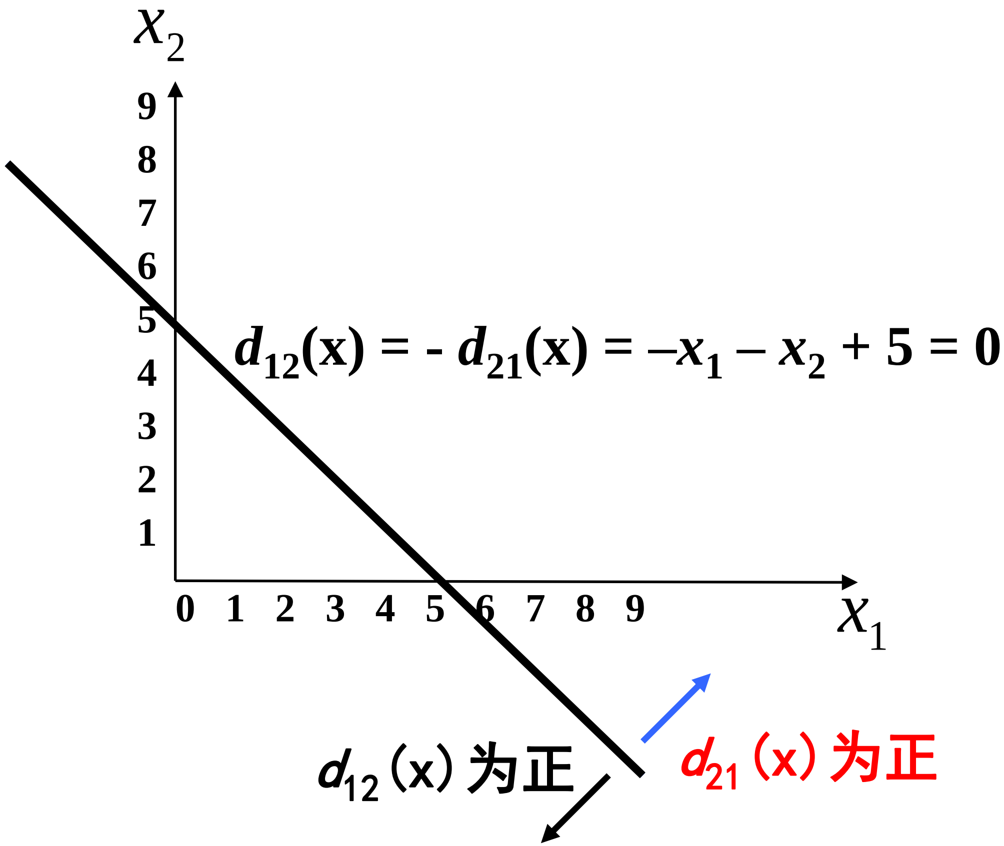
$$d_{12}(\vec{x}) = -2 \Rightarrow \vec{x} \notin \omega_1$$

$$d_{23}(\vec{x}) = -1 \Rightarrow \vec{x} \notin \omega_2$$

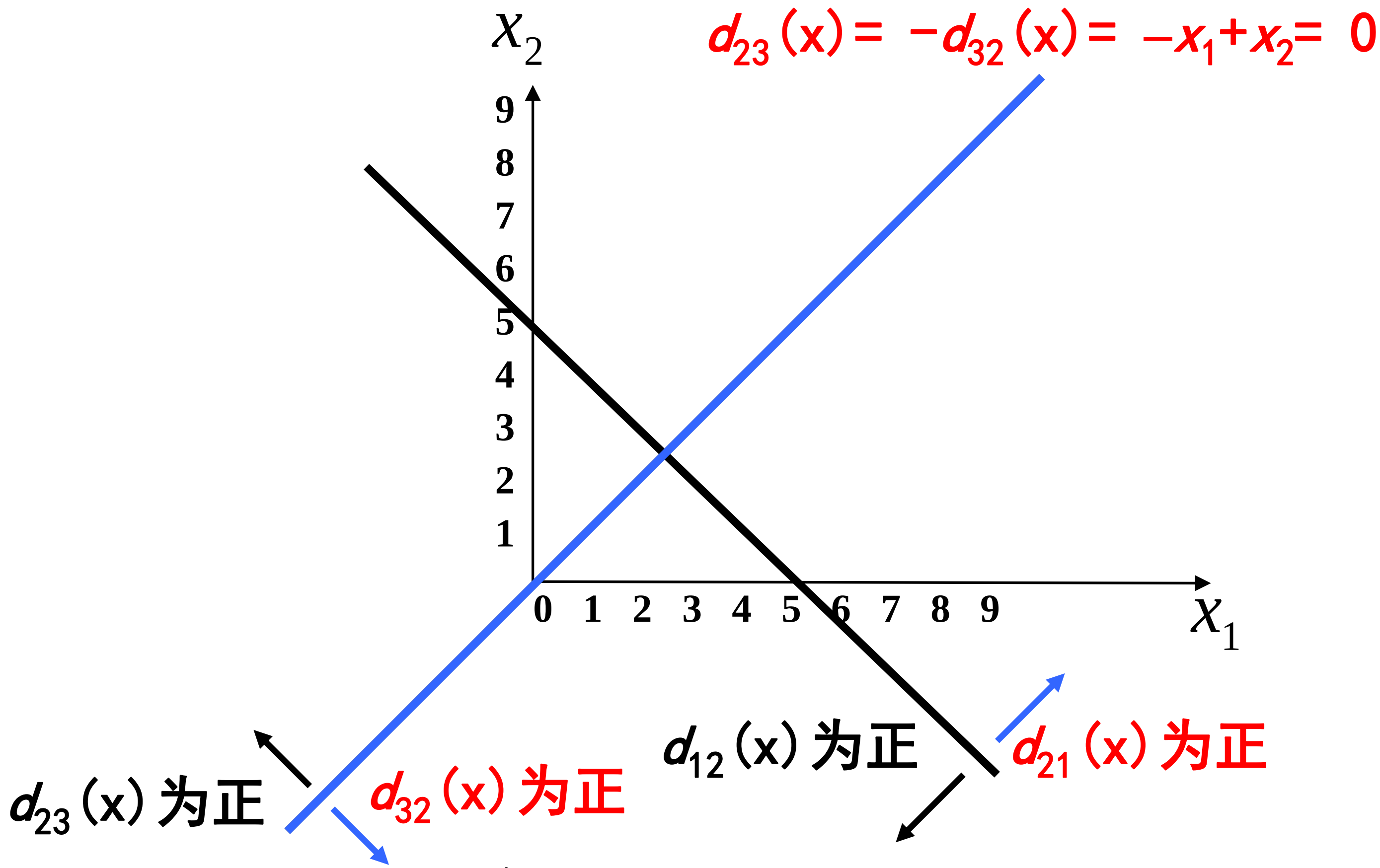
$$d_{13}(\vec{x}) = -1 \Rightarrow \vec{x} \notin \omega_1$$

上面三式等效为：$d_{21}(\vec{x}) = 2 \quad d_{31}(\vec{x}) = 1 \quad d_{32}(\vec{x}) = 1$

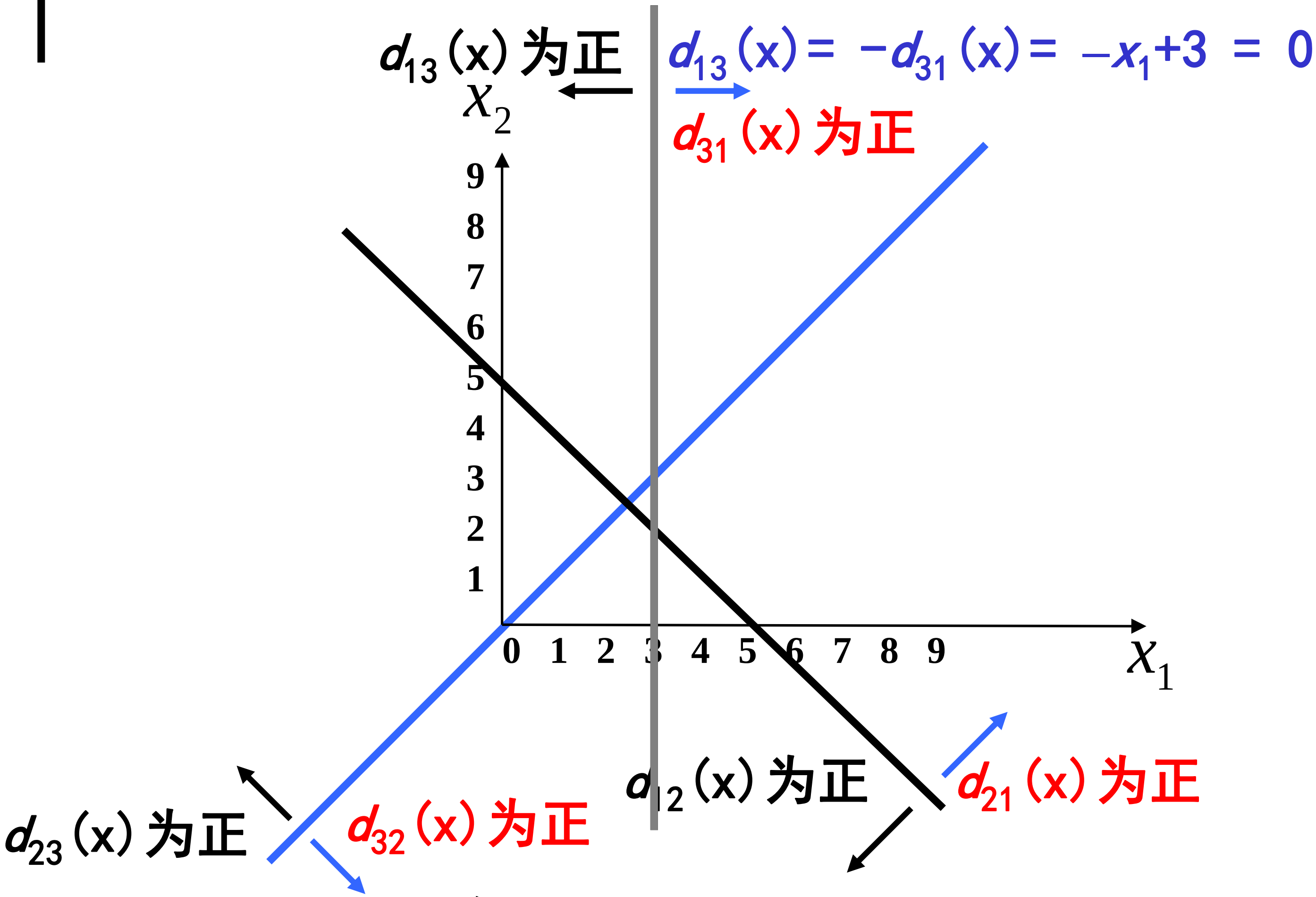
由于 $d_{3j}(\vec{x}) > 0 \ (j = 1,2)$，所以判 $\vec{x} \in \omega_3$



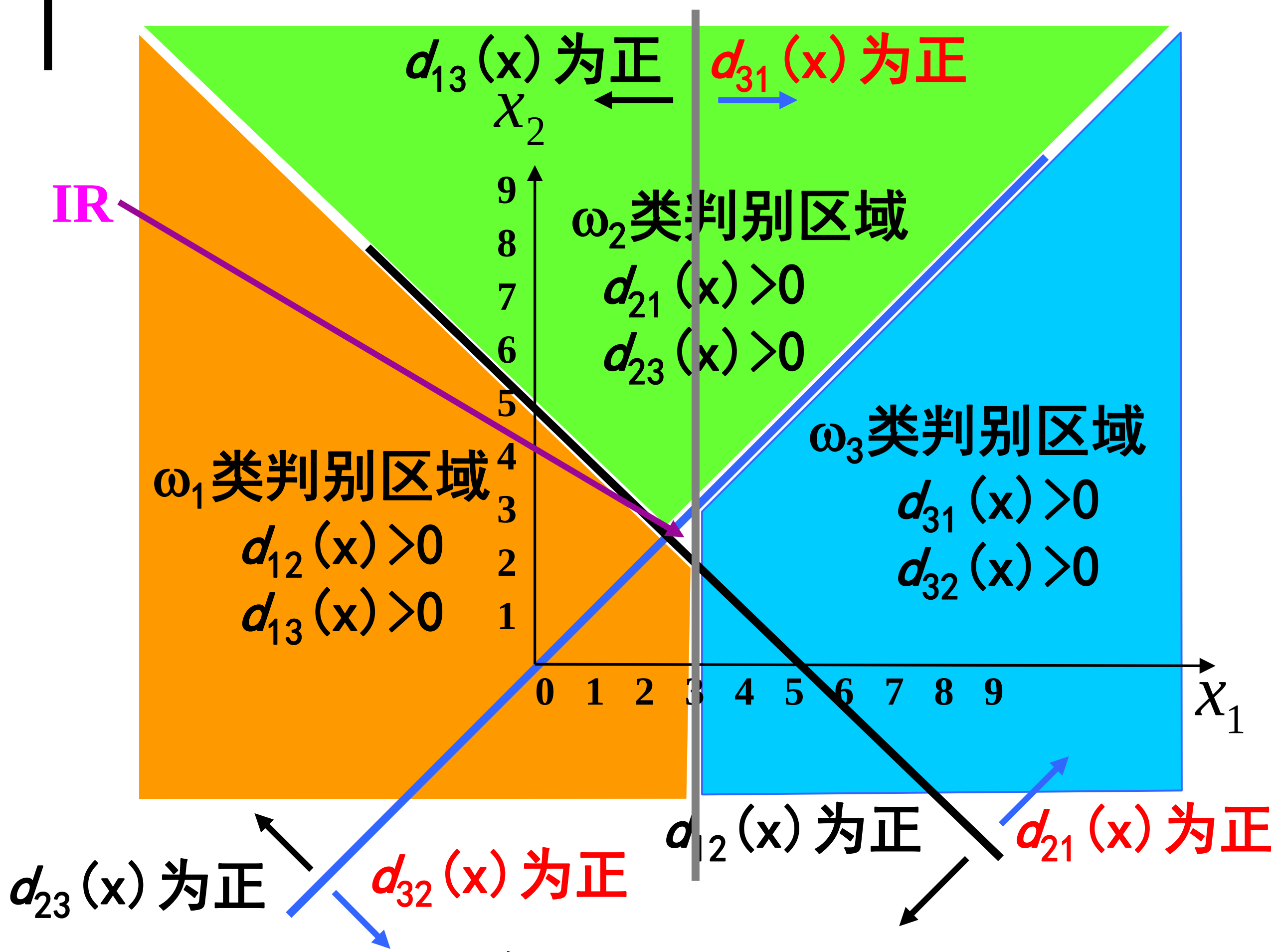
ω_i / ω_j 两分法例题图示



ω_i / ω_j两分法例题图示



ω_i/ω_j两分法例题图示



ω_i/ω_j两分法例题图示

3. 没有不确定区的 ω_i/ω_j 两分法（第三种情况）

令方法2中的判别函数为：

$$d_{ij}(\vec{x}) = d_i(\vec{x}) - d_j(\vec{x}) = (\vec{w}_i - \vec{w}_j)'\vec{x}$$

则 $d_{ij}(\vec{x}) > 0$ 等价于 $d_i(\vec{x}) > d_j(\vec{x})$，于是对每一类 ω_i 均建立一个判别函数 $d_i(\vec{x})$，c 类问题有 c 个判别函数

$$d_i(\vec{x}) = \vec{w}_i'\vec{x} \qquad i = 1,2,\cdots,c$$

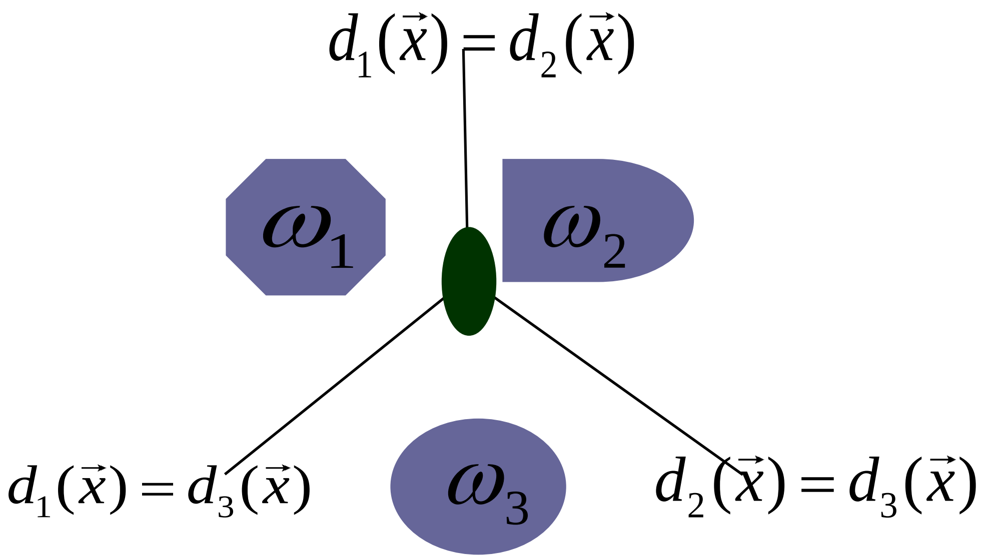
故判决规则成为：

如果 $d_i(\vec{x}) > d_j(\vec{x}) \quad \forall j \neq i$ 则判 $\vec{x} \in \omega_i$

判决规则的另一种表达形式

如果 $d_i(\vec{x}) = \max_j \left[d_j(\vec{x}) \right]$ 则判 $\vec{x} \in \omega_i$

3、第三种情况（续）



多类问题图例（第三种情况）

例：设有一个二维三类问题，三个判别函数为：

$$d_1(\vec{x}) = -x_1 + x_2 \quad d_2(\vec{x}) = x_1 + x_2 - 1 \quad d_3(\vec{x}) = -x_2$$

求模式 $\vec{x} = (1,1)'$ 属于哪一类?

解：将模式 $\vec{x} = (1,1)'$ 代入上面各式得：

$$d_1(\vec{x}) = 0 \qquad d_2(\vec{x}) = 1 \qquad d_3(\vec{x}) = -1$$

由于 $\begin{cases} d_2(\vec{x}) > d_3(\vec{x}) \\ d_2(\vec{x}) > d_1(\vec{x}) \end{cases}$ 所以 $\vec{x} \in \omega_2$

上述三种方法小结:

当 $c>3$ 时，ω_i/ω_j 法比 $\omega_i/\overline{\omega}_i$ 法需要更多的判别函数式，这是一个缺点。

但是 $\omega_i/\overline{\omega}_i$ 法是将 ω_i 类与其余的 $c-1$ 类区分开，而 ω_i/ω_j 法是将 ω_i 类和 ω_j 类分开，显然 ω_i/ω_j 法使模式更容易线性可分，这是它的优点。

方法(3)判别函数的数目和方法(1)相同，但没有不确定区，分析简单，是最常用的一种方法。

判别函数值的鉴别意义、权空间及解空间

3.3.1. 判别函数值的大小、正负的数学意义

n 维特征空间 X^n 中，两类问题的线性判别界面方程为：

$$d(\vec{x}) = \vec{w}_0' \vec{x} + w_{n+1}$$

4.3 线性判别函数的几何性质

模式空间与超平面

1. 概念

模式空间：以n维模式向量X的n个分量为坐标变量的欧氏空间。

模式向量的表示：点、有向线段。

线性分类：用$d(\boldsymbol{X})$进行分类，相当于用超平面$d(\boldsymbol{X})=0$把模式空间分成不同的决策区域。

2. 讨论

设判别函数：$d(\boldsymbol{X})=\boldsymbol{W}_0^{\mathrm{T}}\boldsymbol{X}+w_{n+1}$

式中，$\boldsymbol{W}_0=[w_1,w_2,\cdots,w_n]^{\mathrm{T}}$ $\boldsymbol{X}=[x_1,x_2,\cdots,x_n]^{\mathrm{T}}$

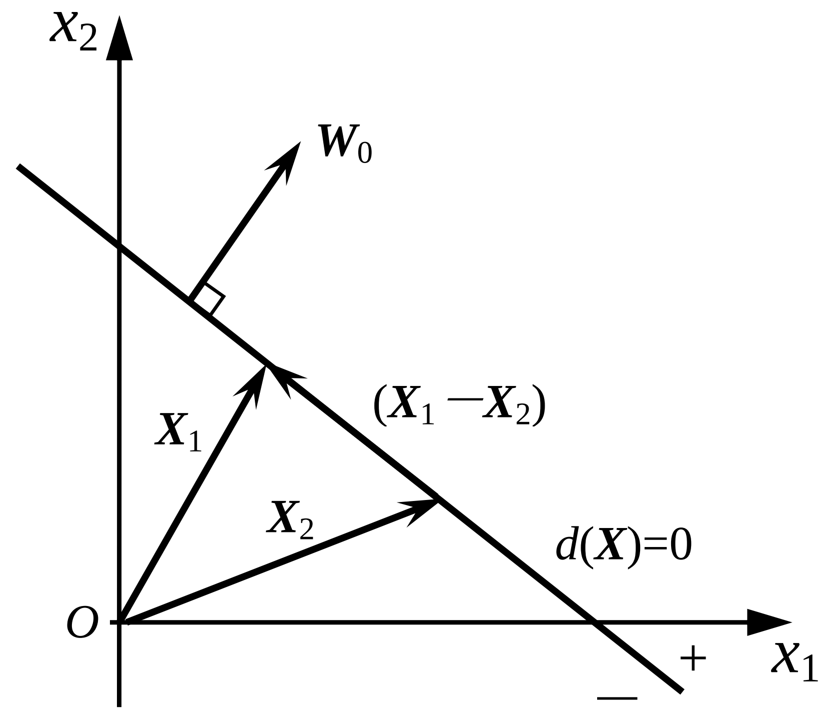
超平面：$d(\boldsymbol{X})=\boldsymbol{W}_0^{\mathrm{T}}\boldsymbol{X}+w_{n+1}=0$

(1) 模式向量X_1和X_2在超平面上

$$\boldsymbol{W}_0^{\mathrm{T}}\boldsymbol{X}_1 + w_{n+1} = \boldsymbol{W}_0^{\mathrm{T}}\boldsymbol{X}_2 + w_{n+1}$$

$$\boldsymbol{W}_0^{\mathrm{T}}(\boldsymbol{X}_1 - \boldsymbol{X}_2) = 0$$

—— $\boldsymbol{W}_0$是超平面的法向量，方向由超平面的负侧指向正侧。



设超平面的单位法线向量为$\boldsymbol{U}$：

$$\boldsymbol{U} = \frac{\boldsymbol{W}_0}{\|\boldsymbol{W}_0\|}$$

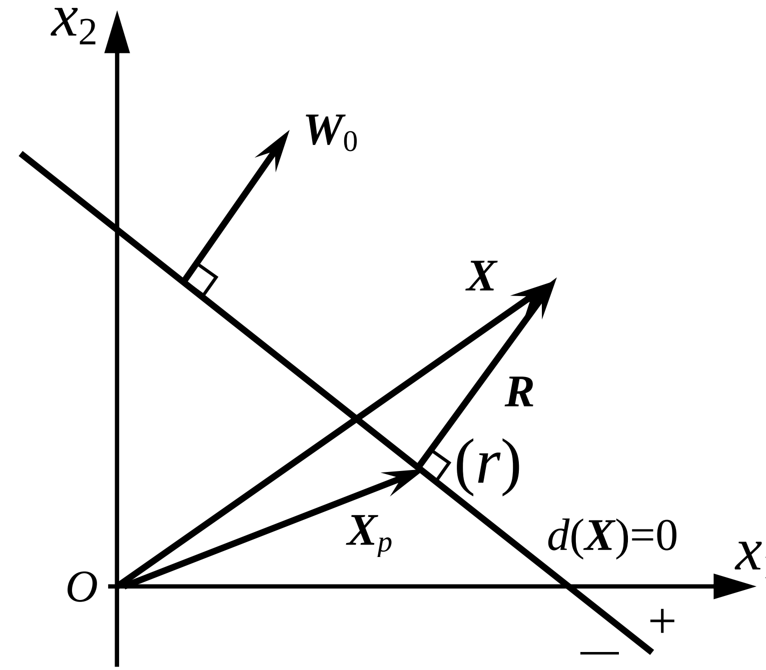
$$\|\boldsymbol{W}_0\| = \sqrt{w_1^2 + w_2^2 + \cdots + w_n^2}$$

(2) **X**不在超平面上

将X向超平面投影得向量X_p，

构造向量**R**：

$$\boldsymbol{R} = r \cdot \boldsymbol{U} = r \frac{\boldsymbol{W}_0}{\|\boldsymbol{W}_0\|}$$



r：**X**到超平面的代数距离。有

$$d(\boldsymbol{X}) = \boldsymbol{W}_0^{\mathrm{T}}\boldsymbol{X} + w_{n+1}$$

$$\boldsymbol{X} = \boldsymbol{X}_p + \boldsymbol{R} = \boldsymbol{X}_p + r\frac{\boldsymbol{W}_0}{\|\boldsymbol{W}_0\|}$$

$$d(\boldsymbol{X}) = \boldsymbol{W}_0^{\mathrm{T}}\left(\boldsymbol{X}_p + r\frac{\boldsymbol{W}_0}{\|\boldsymbol{W}_0\|}\right) + w_{n+1} = (\boldsymbol{W}_0^{\mathrm{T}}\boldsymbol{X}_p + w_{n+1}) + \boldsymbol{W}_0^{\mathrm{T}} \cdot r\frac{\boldsymbol{W}_0}{\|\boldsymbol{W}_0\|}$$

$$= r\|\boldsymbol{W}_0\|$$

—— 判别函数$d(X)$ 正比于点X到超平面的代数距离。

X到超平面的距离：$r = \dfrac{d(\boldsymbol{X})}{\|\boldsymbol{W}_0\|}$

—— 点X到超平面的代数距离（带正负号）正比于$d(X)$函数值。

(3) X在原点

$$d(\boldsymbol{X}) = \boldsymbol{W}_0^{\mathrm{T}}\boldsymbol{X} + w_{n+1} = w_{n+1}$$

得

$$r_0 = \frac{w_{n+1}}{\|\boldsymbol{W}_0\|}$$

—— 超平面的位置由阈值权w_{n+1}决定：

$w_{n+1} > 0$时，原点在超平面的正侧；

$w_{n+1} < 0$时，原点在超平面负侧；

$w_{n+1} = 0$ 时，超平面通过原点。

判别函数值的鉴别意义、权空间及解空间

此方程表示一超平面 π。它有以下三个性质：

- (1) 系数矢量 $\vec{w}_0=(w_1,w_2,....w_n)'$ ，是该平面的法矢量。
- (2) 判别函数 $d(\vec{x})$ 的绝对值正比于 $\vec{x}$ 到超平面 $d(\vec{x})=0$ 的距离。
- (3) 判别函数值的正负表示出特征点位于哪个半空间中。

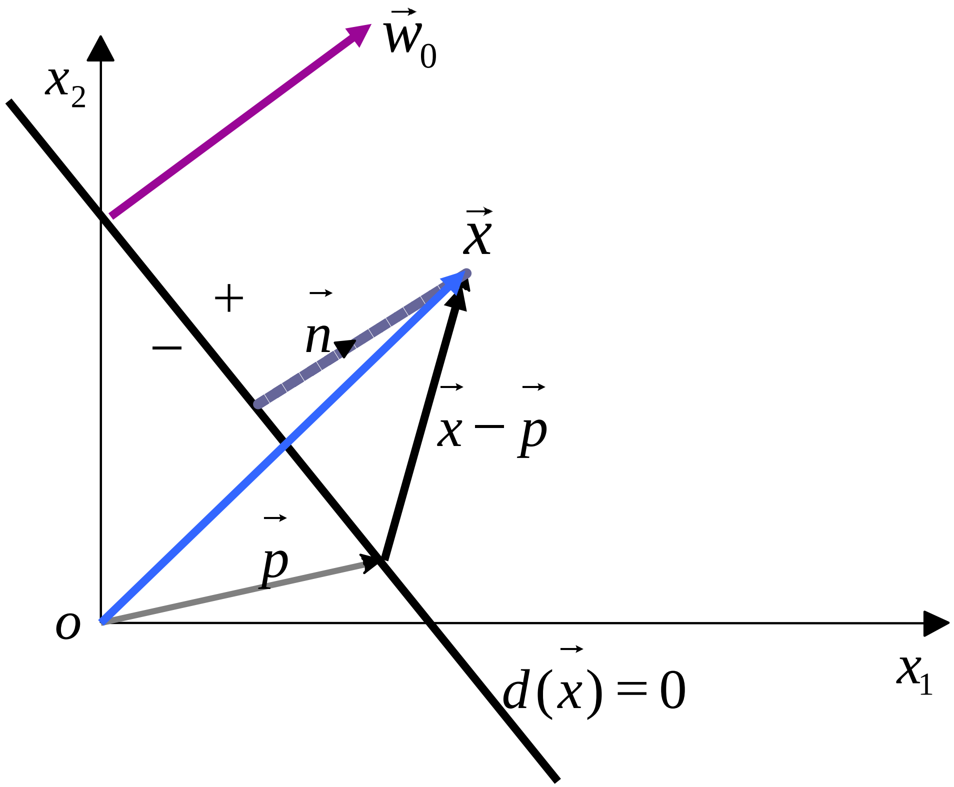


图 点面距离及界面的正负侧示意图

证明：系数矢量$\vec{w}_0=(w_1,w_2,\ldots w_n)'$，是该平面的法矢量，即 $\vec{w}_0\perp$ 平面 π 。

设点 $\vec{x}_1$ 、$\vec{x}_2$ 在判别界面中，故它们满足方程，于是有

$$\vec{w}_0'\vec{x}_1 + w_{n+1} = 0$$

$$\vec{w}_0'\vec{x}_2 + w_{n+1} = 0$$

上面二式相减，可得： $\vec{w}_0'(\vec{x}_1-\vec{x}_2)=0$

这就表明： $\vec{w}_0\perp(\vec{x}_1-\vec{x}_2)$

而差矢量 $(\vec{x}_1-\vec{x}_2)$ 在判别界面中，由于 $\vec{x}_1$ 、$\vec{x}_2$ 是 π 中的任意两点，故 $\vec{w}_0\perp$ 平面 π 。

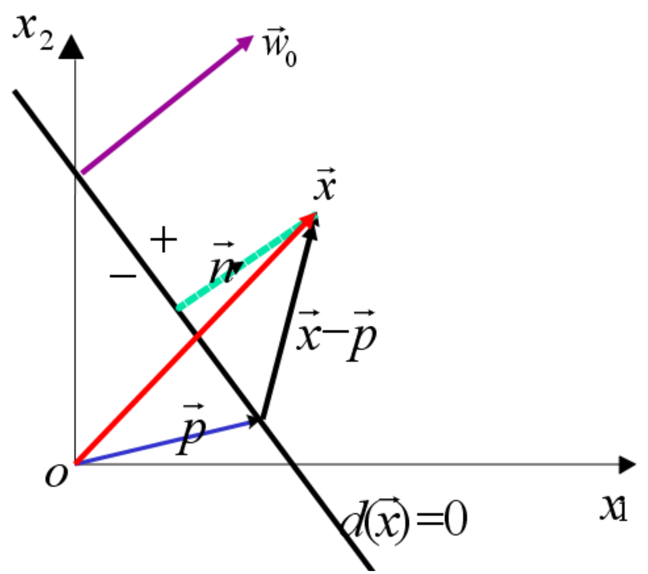
证明：判别函数 $d(\vec{x})$ 的绝对值正比于 $\vec{x}$ 到超平面 $d(\vec{x})=0$ 的距离。

平面 π 的方程可以写成：$\dfrac{\vec{w}_0'}{\|\vec{w}_0\|}\vec{x} = \dfrac{-w_{n+1}}{\|\vec{w}_0\|}$

设平面 π 的单位法矢量 $\vec{n} \triangleq \dfrac{\vec{w}_0'}{\|\vec{w}_0\|}$，上式可写成 $\vec{n}'\vec{x} = \dfrac{-w_{n+1}}{\|\vec{w}_0\|}$

设 $\vec{p}$ 是平面 π 中的任一点，$\vec{x}$ 是特征空间 X^n 中任一点，点 $\vec{x}$ 到平面 π 的距离为差矢量 $(\vec{x}-\vec{p})$ 在 $\vec{n}$ 上的投影的绝对值，即

$$d_x = |\vec{n}'(\vec{x}-\vec{p})| = |\vec{n}'\vec{x}-\vec{n}'\vec{p}|$$



$$d_x = \left|\vec{n}'(\vec{x}-\vec{p})\right| = \left|\vec{n}'\vec{x}-\vec{n}'\vec{p}\right|$$

$$= \left|\frac{\vec{w}_0'}{\left\|\vec{w}_0\right\|}\vec{x} - \frac{\vec{w}_0'}{\left\|\vec{w}_0\right\|}\vec{p}\right| = \left|\frac{\vec{w}_0'}{\left\|\vec{w}_0\right\|}\vec{x} + \frac{w_{n+1}}{\left\|\vec{w}_0\right\|}\right|$$

$$= \frac{\left|\vec{w}_0'\vec{x} + w_{n+1}\right|}{\left\|\vec{w}_0\right\|} = \frac{1}{\left\|\vec{w}_0\right\|}\left|d(\vec{x})\right|$$

上式表明，$d(\vec{x})$ 的值 $\left|d(\vec{x})\right|$ 正比于 $\vec{x}$ 到超平面 $d(\vec{x})=0$ 的距离 d_x 。

证明：判别函数值的正负表示出特征点位于哪个半空间中。

两矢量 $\vec{n}$ 和 $(\vec{x}-\vec{p})$ 的数积为：

$$\vec{n}'(\vec{x}-\vec{p}) = \|\vec{n}\|\|\vec{x}-\vec{p}\|\cos(\vec{n},(\vec{x}-\vec{p})) = \frac{\vec{w}_0'\vec{x}+w_{n+1}}{\|w_0\|}$$

当 $\vec{n}$ 和 $(\vec{x}-\vec{p})$ 夹角小于 90° 时，即 $\vec{x}$ 在 $\vec{n}$ 指向的那个半空间中， $\cos(\vec{n},(\vec{x}-\vec{p}))$ >0；

反之，当 $\vec{n}$ 和 $(\vec{x}-\vec{p})$ 夹角大于 90° 时，即 $\vec{x}$ 在 $\vec{n}$ 背向的半空间中，$\cos(\vec{n},(\vec{x}-\vec{p}))$<0。

由于 $\|\vec{w}_0\|>0$，故 $\vec{n}'(\vec{x}-\vec{p})$ 和 $\vec{w}_0'\vec{x}+w_{n+1}$ 同号。

由于 $\|\vec{w}_0\| > 0$，故 $\vec{n}'(\vec{x}-\vec{p})$ 和 $\vec{w}_0'\vec{x}+w_{n+1}$ 同号。

即 $\vec{x}$ 在 $\vec{n}$ 指向的半空间中时，$\vec{w}_0\vec{x}+w_{n+1} > 0$

当 $\vec{x}$ 在 $\vec{n}$ 背向的半空间中时，$\vec{w}_0\vec{x}+w_{n+1} < 0$

这说明判别函数值的正负表示出特征点位于哪个半空间中，或者换句话说，表示特征点位于界面的哪一侧。

4.4 广义线性判别函数

- 以下情况不能使用线性判别函数。

$$\begin{cases} x < b \text{或} x > a, & x \in \omega_1 \\ b < x < a, & x \in \omega_2 \end{cases}$$

- 要达到分类效果，需要设计这样的判别函数：

$$d(x) = (x-a)(x-b)$$

- 决策规则为：

$$\begin{cases} d(x) > 0, & x \in \omega_1 \\ d(x) < 0, & x \in \omega_2 \end{cases}$$

4.4 广义线性判别函数

目的：

对非线性边界：通过某映射，把模式空间X变成X^*，以便将X空间中非线性可分的模式集，变成在X^*空间中线性可分的模式集。

$$d(\boldsymbol{X})=w_1x_1+w_2x_2+\ldots+w_nx_n+w_{n+1}$$

1. 非线性多项式函数

非线性判别函数的形式之一是非线性多项式函数。

设一训练用模式集，$\{\boldsymbol{X}\}$在模式空间X中线性不可分，非线性判别函数形式如下：

$$d(\boldsymbol{X})=w_1f_1(\boldsymbol{X})+w_2f_2(\boldsymbol{X})+\ldots+w_kf_k(\boldsymbol{X})+w_{k+1}=\sum_{i=1}^{k+1}w_if_i(\boldsymbol{X})$$

式中$\{f_i(\boldsymbol{X}),\ \ i=1,2,\cdots,k\}$是模式$\boldsymbol{X}$的单值实函数，$f_{k+1}(\boldsymbol{X})=1$。

$f_i(\boldsymbol{X})$取什么形式及$d(\boldsymbol{X})$取多少项，取决于非线性边界的复杂程度。

广义形式的模式向量定义为：

$$\boldsymbol{X}^* = [x_1^*, x_2^*, \cdots, x_k^*, 1]^{\mathrm{T}} = [f_1(\boldsymbol{X}), f_2(\boldsymbol{X}), \cdots, f_k(\boldsymbol{X}), 1]^{\mathrm{T}}$$

这里X^*空间的维数k高于X空间的维数n，上式可写为

$$d(\boldsymbol{X}) = \boldsymbol{W}^{\mathrm{T}} \boldsymbol{X}^* = d(\boldsymbol{X}^*), \quad \boldsymbol{W} = [w_1, w_2, \cdots, w_k, w_{k+1}]^{\mathrm{T}}$$

上式是线性的。讨论线性判别函数并不会失去一般性的意义。

问题：

非线性变换可能非常复杂 。

维数大大增加： 维数灾难。

假设X为二维模式向量，$f_i(\boldsymbol{X})$选用二次多项式函数，原判别函数为 $d(\boldsymbol{X})=w_{11}x_1^2+w_{12}x_1x_2+w_{22}x_2^2+w_1x_1+w_2x_2+w_3$

广义线性判别函数：

定义： $x^*_1=f_1(\boldsymbol{X})=x_1^2$ $x^*_2=f_2(\boldsymbol{X})=x_1x_2$

$x^*_3=f_3(\boldsymbol{X})=x_2^2$ $x^*_4=f_4(\boldsymbol{X})=x_1$ $x^*_5=f_5(\boldsymbol{X})=x_2$

即：$\boldsymbol{X}^*=\left[x_1^2,x_1x_2,x_2^2,x_1,x_2,1\right]^{\mathrm{T}}$ $\boldsymbol{W}=\left[w_{11},w_{12},w_{22},w_1,w_2,w_3\right]^{\mathrm{T}}$

$d(\boldsymbol{X})$线性化为：$d(\boldsymbol{X}^*)=\boldsymbol{W}^{\mathrm{T}}\boldsymbol{X}^*$

4.5 广义线性判别函数举例

例：有一个三次判别函数：$z=g(x)=x^3+2x^2+3x+4$。试建立一映射$x\rightarrow\mathbf{y}$，使得z转化为$\mathbf{y}$的线性判别函数。

答：映射X→Y如下：

$$\mathbf{y}=\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix}=\begin{bmatrix} 1 \\ x \\ x^2 \\ x^3 \end{bmatrix},\quad \mathbf{a}=\begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \end{bmatrix}=\begin{bmatrix} 4 \\ 3 \\ 2 \\ 1 \end{bmatrix}$$

$$z=g(x)=h(\mathbf{y})=\mathbf{a}^T\mathbf{y}=\sum_{i=1}^{4}a_iy_i$$

4.5 广义线性判别函数举例

例：设在三维空间中一个类别分类问题拟采用二次曲面。如欲采用广义线性方程求解，试问其广义样本向量与广义权向量的表达式，其维数是多少？

答：设次二次曲面为:

二次曲面

$$ax_1^2 + bx_2^2 + cx_3^2 + dx_1x_2 + ex_1x_3 + fx_2x_3 + gx_1 + hx_2 + lx_3 + m = 0$$

广义权向量

$$\mathbf{a} = (a, b, c, d, e, f, g, h, l, m)^T$$

广义样本向量

$$\mathbf{y} = (x_1^{\ 2}, x_2^{\ 2}, x_3^{\ 2}, x_1x_2, x_1x_3, x_2x_3, x_1, x_2, x_3, 1,)^T$$

维数为10

$$z = g(\mathbf{x}) = h(\mathbf{y}) = \mathbf{a}^T\mathbf{y}$$

广义线性判别函数

4.6 感知器算法（perception approach）

对一种分类学习机模型的称呼，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。但“赏罚概念（reward-punishment concept）”得到广泛应用。

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

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

对样本进行规范化处理，即ω_2类样本全部乘以(－1)，则有：

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

感知器算法通过对已知类别的训练样本集的学习，寻找一个满足上式的权向量。

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，并进行规范化处理。任取权向量初始值$\boldsymbol{W}(1)$，开始迭代。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$\boldsymbol{W}^{\mathrm{T}}(k)\boldsymbol{X}_i \le 0$，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。

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：

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

（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。

梯度法

1. 梯度概念

函数在某点的梯度是这样一个向量，它的方向与取得最大方向导数的方向一致,而它的模为方向导数的最大值．记为：

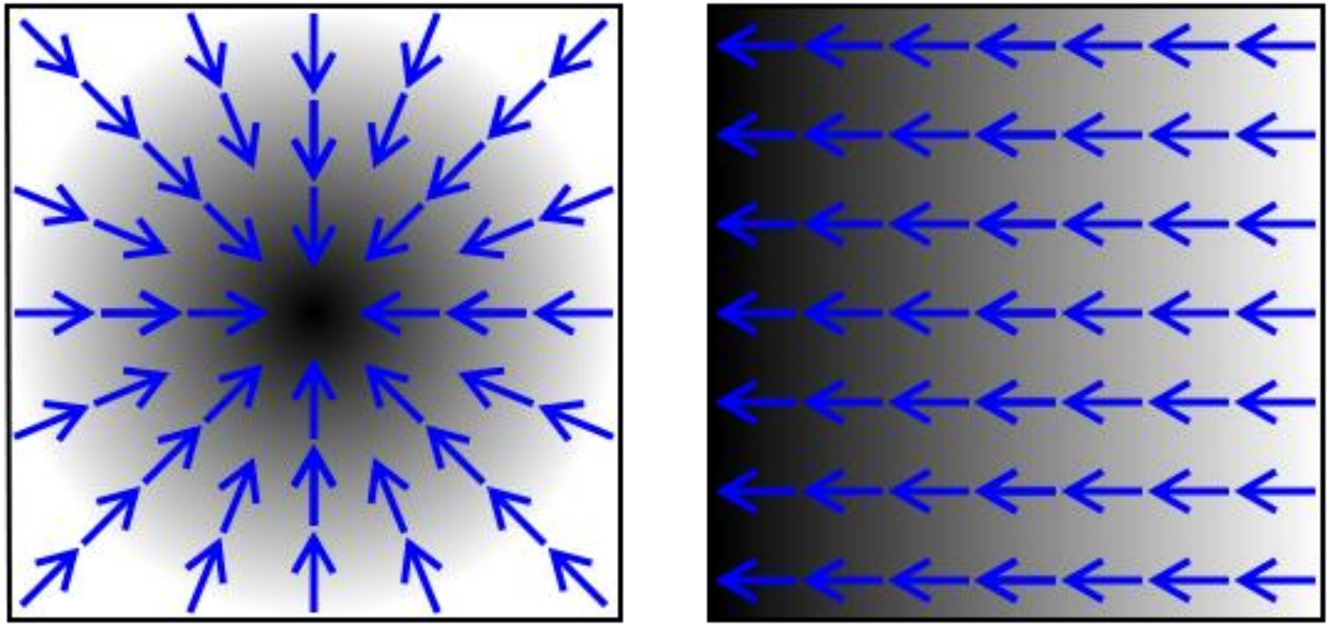
$$\nabla f(\boldsymbol{Y}) = \frac{\mathrm{d}}{\mathrm{d}\boldsymbol{Y}} f(\boldsymbol{Y}) = \left[\frac{\partial f}{\partial y_1}, \frac{\partial f}{\partial y_2}, \cdots, \frac{\partial f}{\partial y_n} \right]^{\mathrm{T}}$$

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。



梯度示例：

上面两个图中，标量场是黑白的，黑色表示大的数值，而其相应的梯度用蓝色箭头表示。

（http://zh.wikipedia.org/wiki/%E6%A2%AF%E5%BA%A6）

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。

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$\nabla J(k)$：

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

权向量修正为：

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

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。

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

随着权向量*W*向理想值接近，准则函数关于**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**W**不再改变，算法收敛。

——— 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。

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。

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*(***W***, ***X***)的选择，*J*(***W***, ***X***)的形式不同，得到的具体算法不同。

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：

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

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

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

或：矩阵论中有

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

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

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

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

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

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

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

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

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

2. 求$\boldsymbol{W}(k+1)$

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

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

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

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

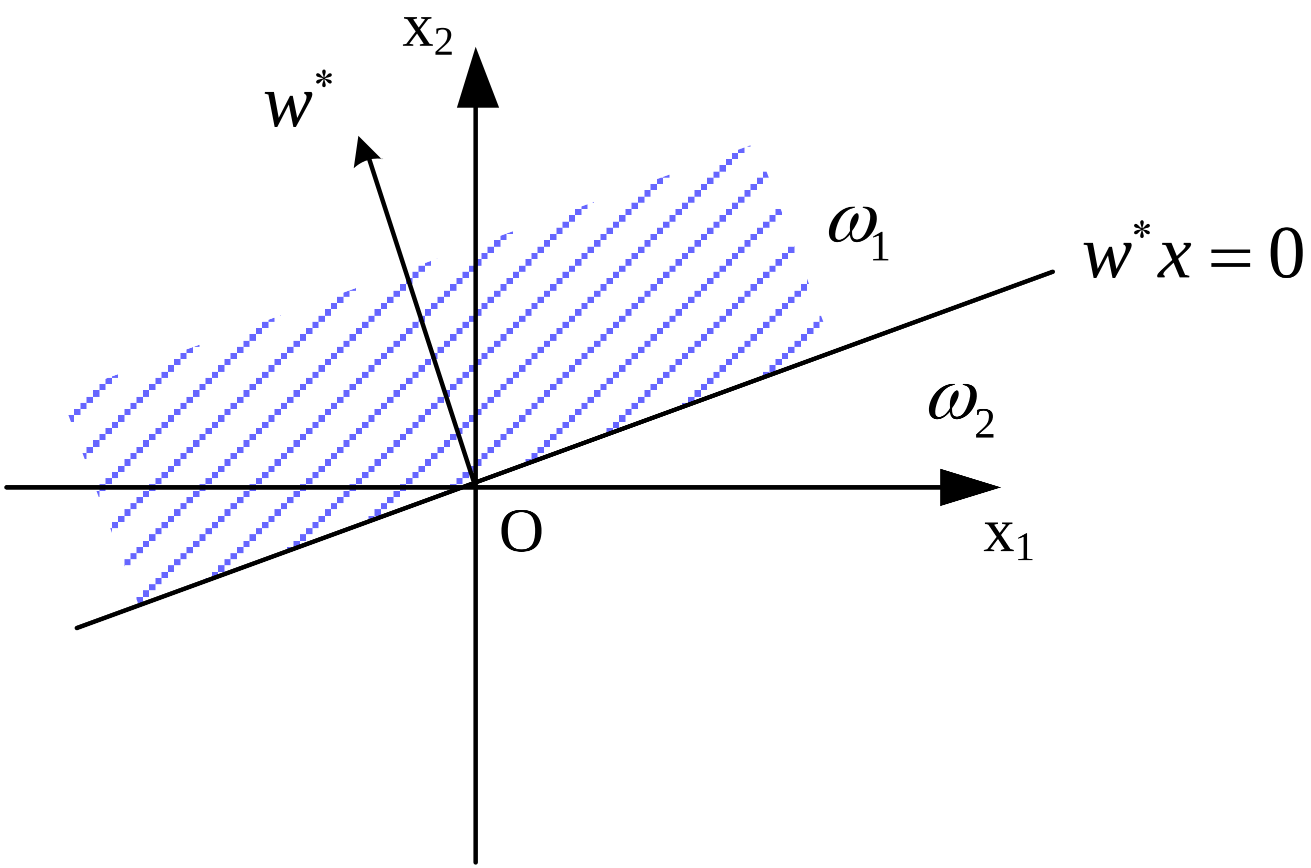
上式即为固定增量算法，与感知器算法形式完全相同 。

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

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。

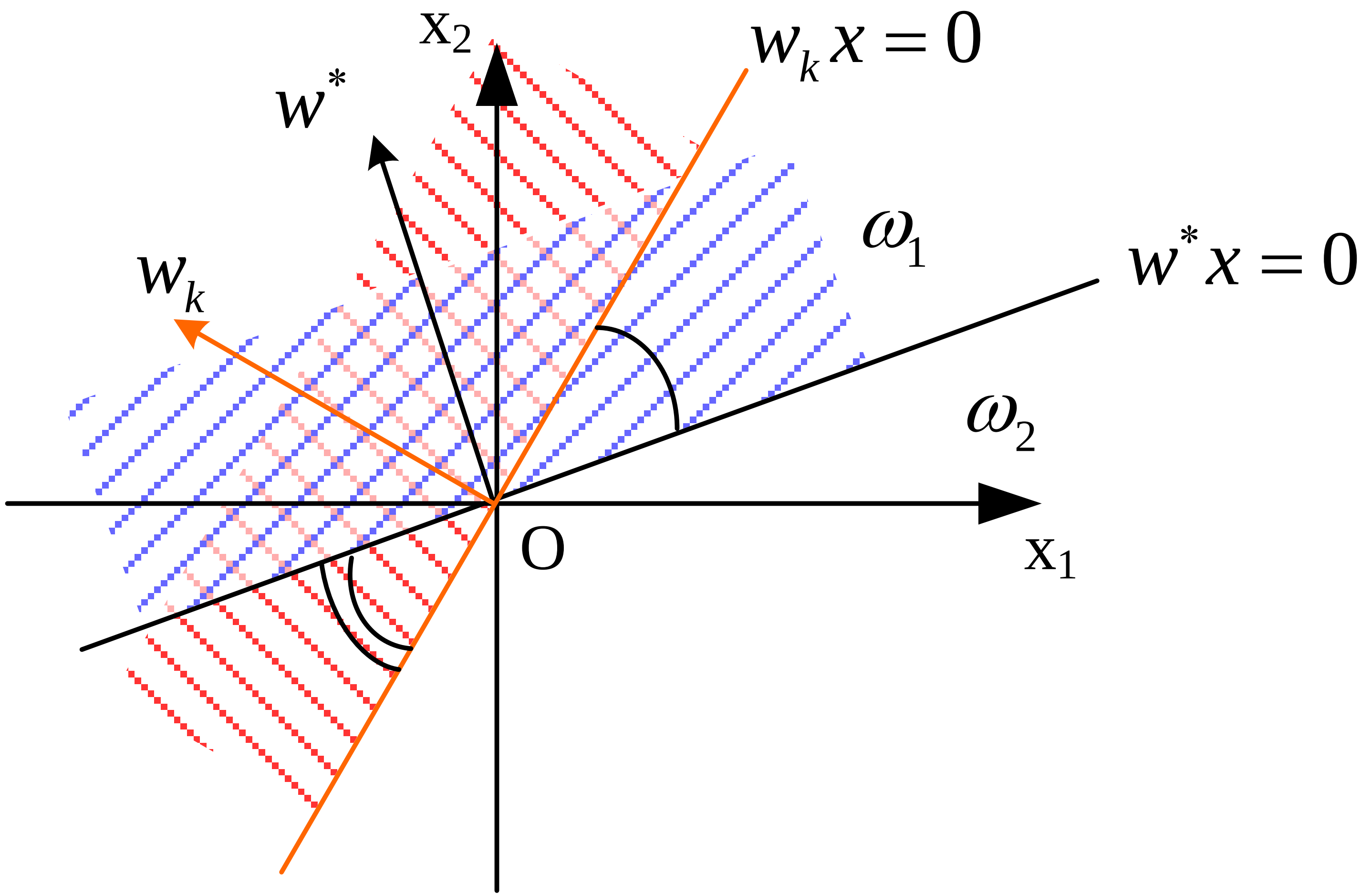
只要模式类是线性可分的，算法就会给出解。

感知器算法的几何解释



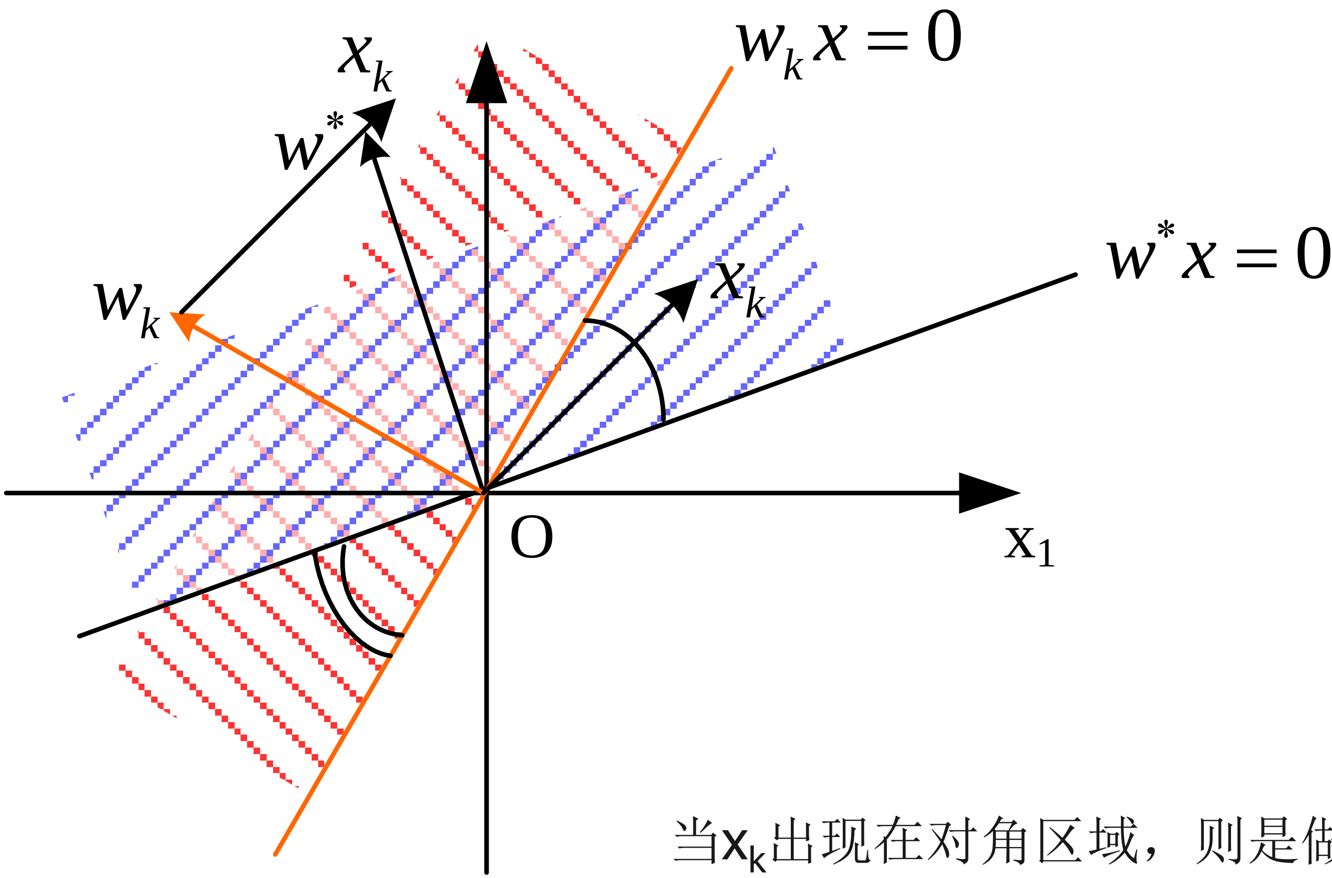
理想的分界面

感知器算法的几何解释



一次迭代过程中的分界面

感知器算法的几何解释



当x_k出现在对角区域，则是做减法。

对分界面法线向量的调整

最小平方误差算法

（least mean square error, LMSE；）

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，只有当模式类可分离时才收敛，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) 模式本身不可分

LMSE算法特点：

对可分模式收敛。

对于类别不可分的情况也能指出来。

LMSE算法

1) **原理** LMSE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。

说明：

∴ 两式等价。

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

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

补充：LMSE一般指$\boldsymbol{B}$固定，当$\boldsymbol{B}$可变时称为Ho-Kashyap算法。

分类器的不等式方程

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：

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

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

写成矩阵形式为：

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

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

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

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

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

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 LMSE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：

"最小二乘"：

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

—— 最小：使方程组两边误差最小，也即使J最小。

—— 二乘：次数为2

$\left.\begin{array}{l}\text{—— 最小：使方程组两边误差最小，也即使}J\text{最小。}\\ \text{—— 二乘：次数为2}\end{array}\right\}$ 最小平方（误差算法）

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

$\Big\downarrow$ 转化为

对$\boldsymbol{XW}=\boldsymbol{B}$求解

$\Big\downarrow$ 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：

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

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

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

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

准则函数：

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

$\boldsymbol{XW}=\boldsymbol{B}$ 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。

2) 推导LMSE算法递推公式

与问题相关的两个梯度：

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

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

求递推公式：

(1) 求$\boldsymbol{W}$的递推关系

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

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

式中：$\boldsymbol{X}^{\#} = \left(\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}\right)^{-1}\boldsymbol{X}^{\mathrm{T}}$ 称为$\boldsymbol{X}$的伪逆，也称为广义逆阵。

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

由上式可知：只要求出$\boldsymbol{B}$，就可求出$\boldsymbol{W}$。

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

(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：

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

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

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

(3) 求$\boldsymbol{W}(k+1)$的迭代式

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

$$=\boldsymbol{X}^{\#}\boldsymbol{B}(k)+\underbrace{\boldsymbol{X}^{\#}c\boldsymbol{e}(k)}_{=\mathbf{0}}+\boldsymbol{X}^{\#}c\left|\boldsymbol{e}(k)\right|=\boldsymbol{W}(k)+c\boldsymbol{X}^{\#}\left|\boldsymbol{e}(k)\right|$$

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

$$=\boldsymbol{W}(k)-X^{\#}\boldsymbol{B}(k)=0$$

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

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

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。

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

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

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

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

$\boldsymbol{W}(k+1)$、$\boldsymbol{B}(k+1)$计算的先后次序无关。

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

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

$$\cdots$$

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

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

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

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