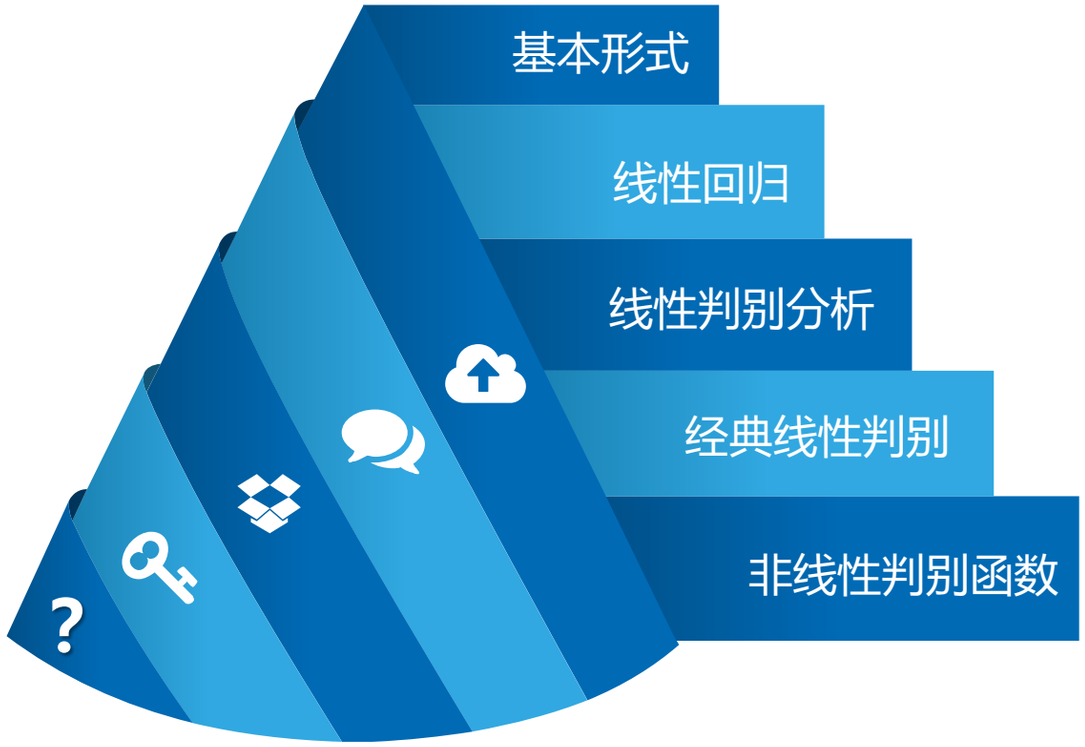
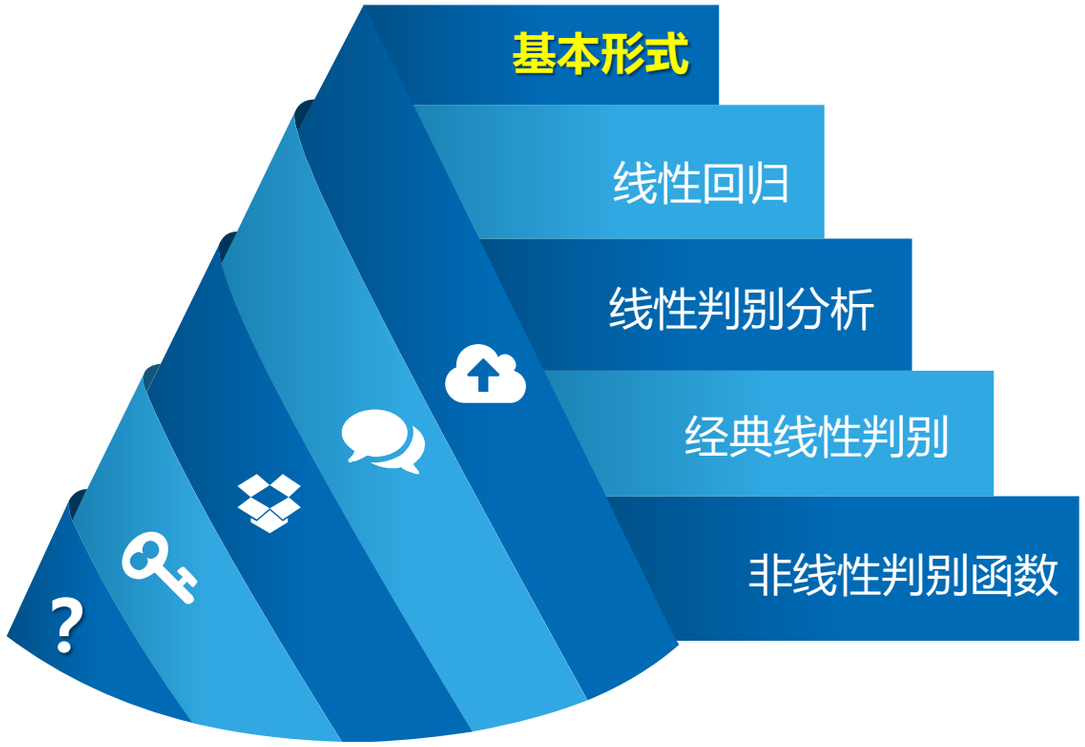
第4讲 线性模型

周文晖

杭州电子科技大学



- **线性模型基本形式**
  线性方程，基本形式，…
- **线性回归**
  参数估计，最小二乘，对数几率回归，…
- **线性判别分析**
  基本概念，广义线性判别，几何性质 …
- **经典线性判别准则**
  Fisher判别，感知器准则、梯度法 …
- **非线性判别函数**
  分段线性，…



线性模型基本形式

线性方程，基本形式，…

线性回归

参数估计，最小二乘，对数几率回归，…

线性判别分析

基本概念，广义线性判别，几何性质 …

经典线性判别准则

Fisher判别，感知器准则、梯度法 …

非线性判别函数

分段线性，…

统计决策方法的局限

$$P(\omega_i \mid X) = \frac{P(\omega_i)P(X \mid \omega_i)}{P(X)}$$

统计决策方法思路： 通过贝叶斯公式将后验估计问题转为类先验概率和类条件概率计算；
将诊断推理转为因果推理。

局限：

- 必须精确知道类先验概率和类条件概率
- 类条件概率估计
  - 估计类条件概率密度函数：参数估计和非参数估计
  - 直接估计类条件概率，利用条件独立性，采用朴素贝叶斯方法
- 挑战：类条件概率密度函数难以估计；类条件概率计算（尤其是联合概率计算）困难。

线性模型基本形式——数值表示

线性模型的一般形式

$$f(\boldsymbol{x}) = w_1 x_1 + w_2 x_2 + \ldots + w_d x_d + b$$

$\boldsymbol{x} = (x_1; x_2; \ldots; x_d)$ 是由d 个属性（特征）描述的一个示例（样本）。

一个样本表示为一个 d 维特征空间上的点，其中 x_i 是 $\boldsymbol{x}$ 在第 i 个属性（第 i 维）上的取值。

线性模型的向量形式

$$f(\boldsymbol{x}) = \boldsymbol{w}^{\mathrm{T}}\boldsymbol{x} + b \qquad \text{其中} \quad \boldsymbol{w} = (w_1; w_2; \ldots; w_d)$$

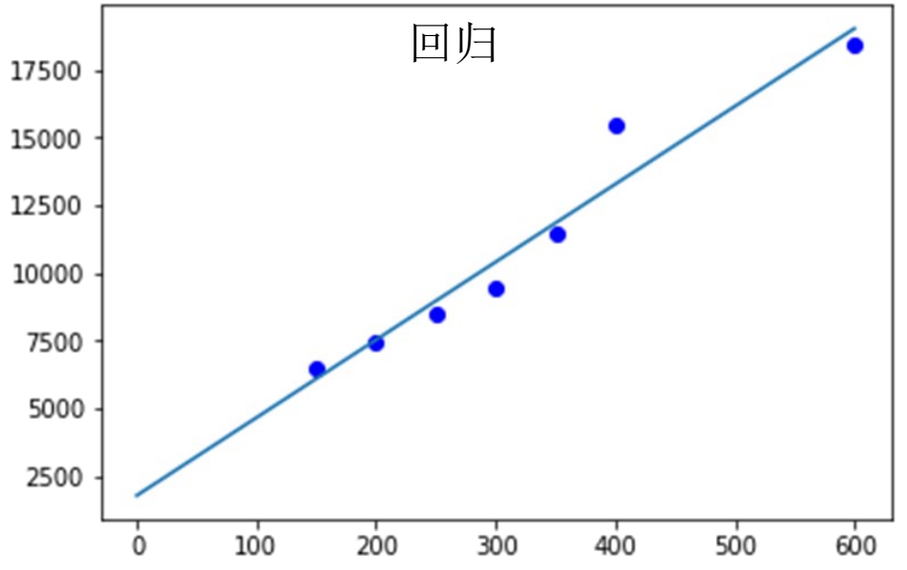
模型的参数由权重向量 $\boldsymbol{w}$ 与偏置项 b 构成，通过训练得到。

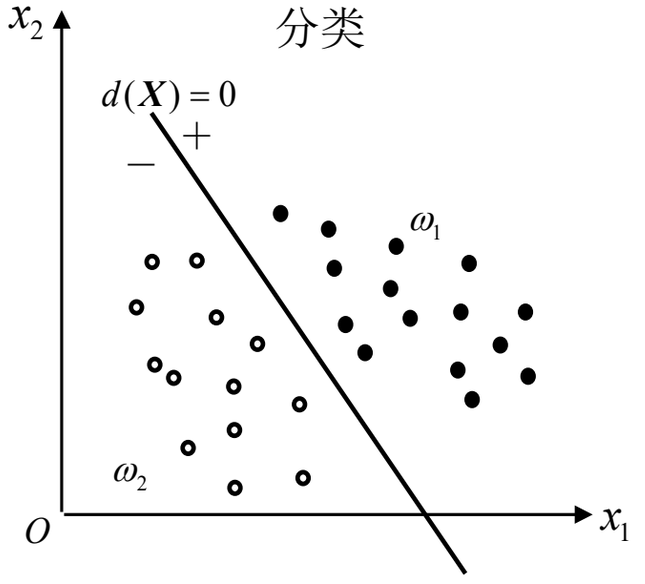
线性模型基本形式——几何表示

线性模型的几何意义：d 维特征空间中的一个超平面 $f(\boldsymbol{x}) = w_1x_1 + w_2x_2 + \ldots + w_dx_d + b$

一维特征空间：线性模型表示为一条直线；　　二维特征空间：线性模型表示为一个平面

线性模型可以用来解决模式识别的两个基本问题：





线性模型基本形式——增广表示

线性模型的增广向量形式

将权重向量和特征向量进行增广合并，以简化表达，即 $[\boldsymbol{w},b]\to\boldsymbol{w}$ ， $[\boldsymbol{x},1]\to\boldsymbol{x}$

$$f(\boldsymbol{x})=w_1x_1+w_2x_2+\ldots+w_nx_n+b\cdot 1=[w_1,w_2,...,w_n,b]\begin{bmatrix}x_1\\x_2\\\vdots\\x_n\\1\end{bmatrix}=\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}$$

式中： $\boldsymbol{x}=[x_1,x_2,....,x_n,1]^{\mathrm{T}}$ 为增广模式向量。

$\boldsymbol{w}=[w_1,w_2,....,w_n,w_{n+1}]^{\mathrm{T}}$ 为增广权向量。

线性模型优点

优点：◇ 形式简单、易于建模

◇ 可解释性

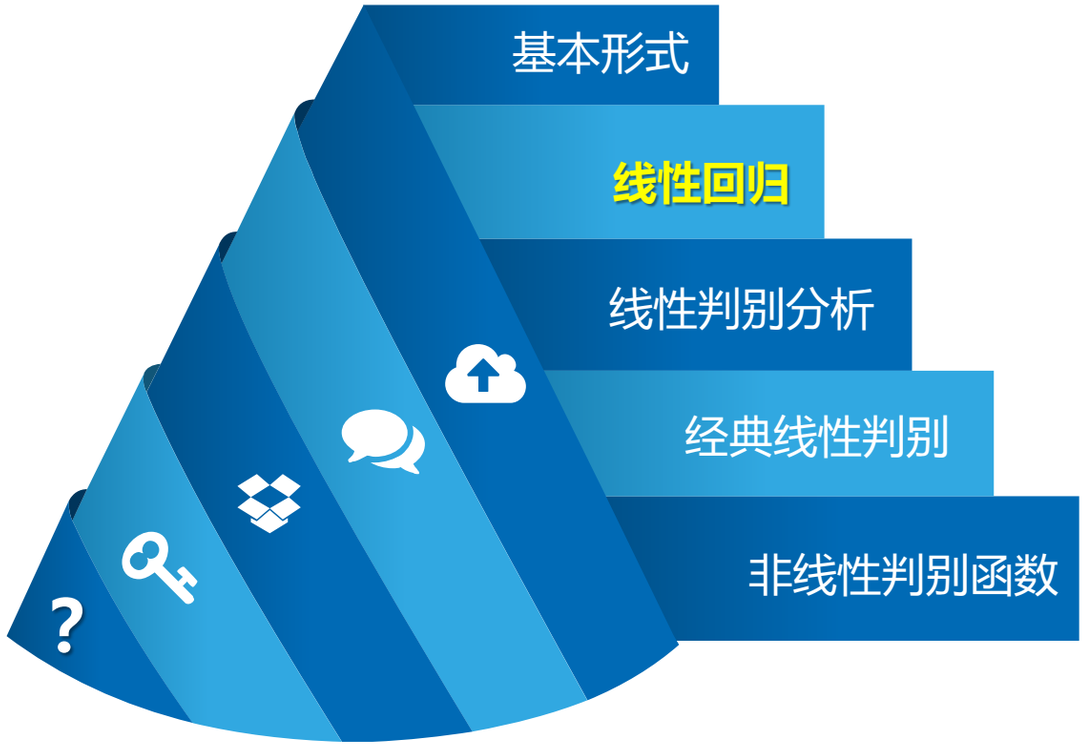
◇ 非线性模型的基础

- 许多非线性模型可在线性模型基础上引入层级结构或高维映射而得到。

一个例子：

- 综合考虑色泽、根蒂和敲声来判断西瓜好不好
- 其中根蒂的系数最大，表明根蒂最要紧；而敲声的系数比色泽大，说明敲声比色泽更重要

$$f_{\text{好瓜}}(\boldsymbol{x}) = 0.2 \cdot x_{\text{色泽}} + 0.5 \cdot x_{\text{根蒂}} + 0.3 \cdot x_{\text{敲声}} + 1$$



- **线性模型基本形式**
  线性方程，基本形式，…
- **线性回归**
  参数估计，最小二乘，对数几率回归，…
- **线性判别分析**
  基本概念，广义线性判别，几何性质 …
- **经典线性判别准则**
  Fisher判别，感知器准则、梯度法 …
- **非线性判别函数**
  分段线性，…

线性回归

给定数据集 $D=\{(\boldsymbol{x}_1,y_1),(\boldsymbol{x}_2,y_2),\ldots,(\boldsymbol{x}_m,y_m)\}$

其中 $\boldsymbol{x}_i=(x_{i1};x_{i2};\ldots;x_{id})\quad y_i\in\mathbb{R}$

线性回归（linear regression）问题：

◇ 学得一个线性模型以尽可能准确地预测实值输出标记。

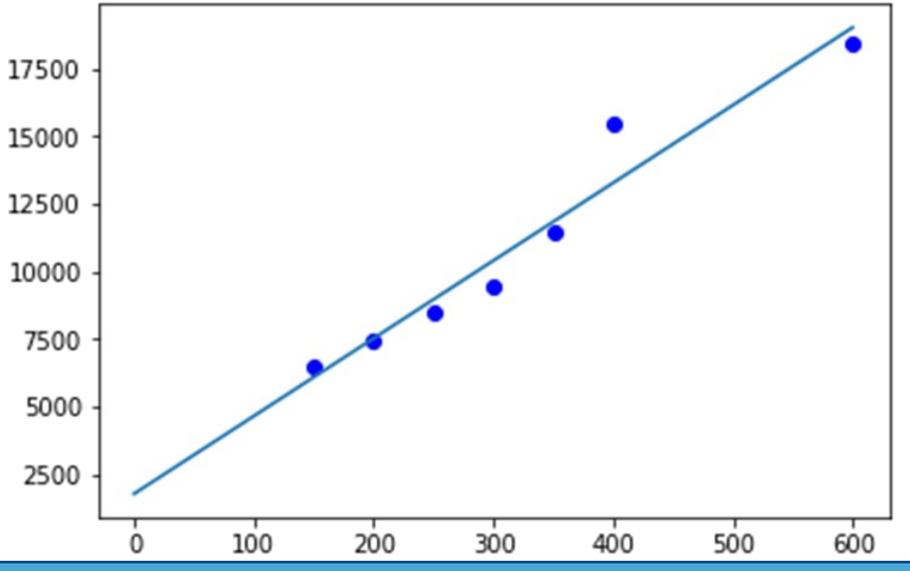
例，设定一元线性回归的线性预测函数为

$$f(\boldsymbol{x})=w\cdot\boldsymbol{x}+b$$

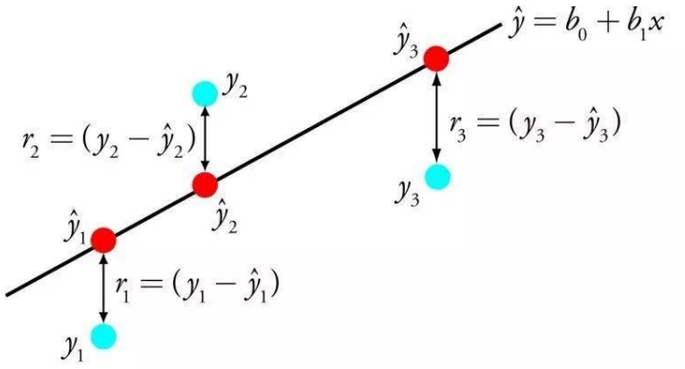
通过学习到 w 和 b，使得 $f(x_i)\simeq y_i$

寻找 w 和 b 的最佳值为线性回归模型的参数估计过程。

面积	价格
150	6450
200	7450
250	8450
300	9450
350	11450
400	15450
600	18450



线性回归——最小二乘法

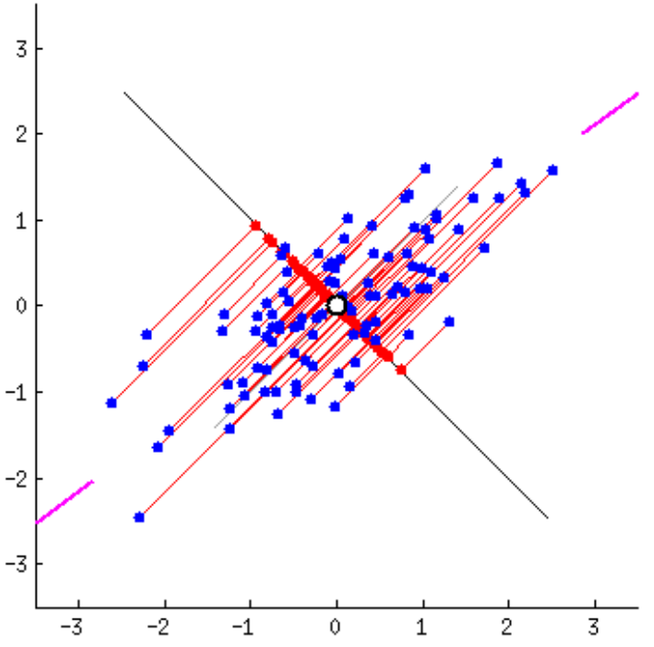


为确定 w 和 b，关键是使得 $f(\boldsymbol{x})$ 与 $\boldsymbol{y}$ 之间的差距最小。

均方误差是回归任务中最常用的性能度量，线性回归问题转化为寻找使得均方误差最小的 w 和 b：

$$(w^*, b^*) = \arg\min_{(w,b)} \sum_{i=1}^{m} (f(x_i) - y_i)^2$$

$$= \arg\min_{(w,b)} \sum_{i=1}^{m} (y_i - wx_i - b)^2$$



采用均方误差（MSE）的意义：

(1) 均方误差的几何意义：欧几里得距离，或简称欧氏距离（Euclidean distance）

(2) 基于均方误差最小化的模型参数求解过程称为“最小二乘法”（Least square method ）

线性回归——最小二乘法参数估计

求解 w 和 b 的最佳值，使得均方误差 $E_{(w,b)}=\sum_{i=1}^{m}(y_i-wx_i-b)^2$ 最小的过程，称为线性回归模型的最小二乘“参数估计”。

对于一元线性回归问题，其求解思路：(1) 将 $E_{(w,b)}$ 分别对 w 和 b 求偏导；

(2) 令偏导为零，得到最优解的闭式（closed-form）解。

对 w 求偏导：

$$\frac{\partial E_{(w,b)}}{\partial w}=\frac{\partial}{\partial w}\left[\sum_{i=1}^{m}(y_i-wx_i-b)^2\right]=\sum_{i=1}^{m}\frac{\partial}{\partial w}\left[(y_i-wx_i-b)^2\right]=\sum_{i=1}^{m}\left[2\cdot(y_i-wx_i-b)\cdot(-x_i)\right]$$

$$=\sum_{i=1}^{m}\left[2\cdot\left(wx_i^2-y_ix_i+bx_i\right)\right]=2\cdot\left(w\sum_{i=1}^{m}x_i^2-\sum_{i=1}^{m}y_ix_i+b\sum_{i=1}^{m}x_i\right)=2\left(w\sum_{i=1}^{m}x_i^2-\sum_{i=1}^{m}(y_i-b)\,x_i\right)$$

线性回归——最小二乘法参数估计

对 b 求偏导：

$$\frac{\partial E_{(w,b)}}{\partial b}=\frac{\partial}{\partial b}\left[\sum_{i=1}^{m}(y_i-wx_i-b)^2\right]=\sum_{i=1}^{m}\frac{\partial}{\partial b}\left[(y_i-wx_i-b)^2\right]=\sum_{i=1}^{m}[2\cdot(y_i-wx_i-b)\cdot(-1)]$$

$$=\sum_{i=1}^{m}[2\cdot(b-y_i+wx_i)]=2\cdot\left[\sum_{i=1}^{m}b-\sum_{i=1}^{m}y_i+\sum_{i=1}^{m}wx_i\right]=2\left(mb-\sum_{i=1}^{m}(y_i-wx_i)\right)$$

令 $\frac{\partial E_{(w,b)}}{\partial b}$ 为零，求解 b 的最优解

$$\frac{\partial E_{(w,b)}}{\partial b}=2\left(mb-\sum_{i=1}^{m}(y_i-wx_i)\right)=0 \quad \Rightarrow \quad b=\frac{1}{m}\sum_{i=1}^{m}(y_i-wx_i)$$

令 $\frac{1}{m}\sum_{i=1}^{m}y_i=\bar{y}$, $\frac{1}{m}\sum_{i=1}^{m}x_i=\bar{x}$，则 $b=\bar{y}-w\bar{x}$

线性回归——最小二乘法参数估计

令 $\frac{\partial E_{(w,b)}}{\partial w}$ 为零，求解 w 的最优解

$$\frac{\partial E_{(w,b)}}{\partial w} = 2\left(w\sum_{i=1}^{m}x_i^2 - \sum_{i=1}^{m}(y_i - b)\,x_i\right) = 0 \quad \Rightarrow \quad w\sum_{i=1}^{m}x_i^2 = \sum_{i=1}^{m}y_i x_i - \sum_{i=1}^{m}bx_i$$

将 $b = \bar{y} - w\bar{x}$ 代入左式，

$$\Rightarrow \quad w\sum_{i=1}^{m}x_i^2 = \sum_{i=1}^{m}y_i x_i - \sum_{i=1}^{m}(\bar{y} - w\bar{x})x_i \quad \Rightarrow \quad w\sum_{i=1}^{m}x_i^2 = \sum_{i=1}^{m}y_i x_i - \bar{y}\sum_{i=1}^{m}x_i + w\bar{x}\sum_{i=1}^{m}x_i$$

$$\Rightarrow \quad w(\sum_{i=1}^{m}x_i^2 - \bar{x}\sum_{i=1}^{m}x_i) = \sum_{i=1}^{m}y_i x_i - \bar{y}\sum_{i=1}^{m}x_i \quad \Rightarrow \quad w = \frac{\sum_{i=1}^{m}y_i x_i - \bar{y}\sum_{i=1}^{m}x_i}{\sum_{i=1}^{m}x_i^2 - \bar{x}\sum_{i=1}^{m}x_i}$$

$$\bar{y}\sum_{i=1}^{m}x_i = \frac{1}{m}\sum_{i=1}^{m}y_i\sum_{i=1}^{m}x_i = \bar{x}\sum_{i=1}^{m}y_i$$

$$\bar{x}\sum_{i=1}^{m}x_i = \frac{1}{m}\sum_{i=1}^{m}x_i\sum_{i=1}^{m}x_i = \frac{1}{m}(\sum_{i=1}^{m}x_i)^2$$

$$\Rightarrow \quad w = \frac{\sum_{i=1}^{m}y_i(x_i - \bar{x})}{\sum_{i=1}^{m}x_i^2 - \frac{1}{m}(\sum_{i=1}^{m}x_i)^2}$$

多元线性回归

对于更一般情况，$f(\boldsymbol{x}_i)=\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i+b$ ，使得 $f(\boldsymbol{x}_i)\simeq y_i$ 。

类似于一元线性回归中的最小二乘法，多元线性回归问题转为

$$(\boldsymbol{w}^*,b^*)=\arg\min_{(\boldsymbol{w},b)}\sum_{i=1}^{m}\left(f(\boldsymbol{x}_i)-y_i\right)^2=\arg\min_{(\boldsymbol{w},b)}\sum_{i=1}^{m}\left(y_i-\left(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i+b\right)\right)^2$$

为便于讨论，采用增广向量形式 $\hat{\boldsymbol{w}}=(\boldsymbol{w};b)=(w_1;\ldots;w_d;b)\in\mathbb{R}^{(d+1)\times 1}$ ， $\hat{\boldsymbol{x}}_i=(x_1;\ldots;x_d;1)\in\mathbb{R}^{(d+1)\times 1}$ ，有

$$\hat{\boldsymbol{w}}^*=\arg\min_{\hat{\boldsymbol{w}}}\sum_{i=1}^{m}\left(y_i-\hat{\boldsymbol{x}}_i^{\mathrm{T}}\hat{\boldsymbol{w}}\right)^2$$

根据向量内积的定义可知，上式可以写成如下向量内积的形式

$$\hat{\boldsymbol{w}}^*=\arg\min_{\hat{\boldsymbol{w}}}\begin{bmatrix}y_1-\hat{\boldsymbol{x}}_1^{\mathrm{T}}\hat{\boldsymbol{w}} & \cdots & y_m-\hat{\boldsymbol{x}}_m^{\mathrm{T}}\hat{\boldsymbol{w}}\end{bmatrix}\begin{bmatrix}y_1-\hat{\boldsymbol{x}}_1^{\mathrm{T}}\hat{\boldsymbol{w}}\\ \vdots\\ y_m-\hat{\boldsymbol{x}}_m^{\mathrm{T}}\hat{\boldsymbol{w}}\end{bmatrix}=\arg\min_{\hat{\boldsymbol{w}}}(\boldsymbol{y}-\mathbf{X}\hat{\boldsymbol{w}})^{\mathrm{T}}(\boldsymbol{y}-\mathbf{X}\hat{\boldsymbol{w}})$$

$$\begin{bmatrix}y_1-\hat{\boldsymbol{x}}_1^{\mathrm{T}}\hat{\boldsymbol{w}}\\ \vdots\\ y_m-\hat{\boldsymbol{x}}_m^{\mathrm{T}}\hat{\boldsymbol{w}}\end{bmatrix}=\begin{bmatrix}y_1\\ \vdots\\ y_m\end{bmatrix}-\begin{bmatrix}\hat{\boldsymbol{x}}_1^{\mathrm{T}}\hat{\boldsymbol{w}}\\ \vdots\\ \hat{\boldsymbol{x}}_m^{\mathrm{T}}\hat{\boldsymbol{w}}\end{bmatrix}=\boldsymbol{y}-\begin{bmatrix}\hat{\boldsymbol{x}}_1^{\mathrm{T}}\\ \vdots\\ \hat{\boldsymbol{x}}_m^{\mathrm{T}}\end{bmatrix}\cdot\hat{\boldsymbol{w}}=\boldsymbol{y}-\mathbf{X}\hat{\boldsymbol{w}}$$

多元线性回归——最小二乘法

令 $E_{\hat{\boldsymbol{w}}}=(\boldsymbol{y}-\mathbf{X}\hat{\boldsymbol{w}})^{\mathrm{T}}(\boldsymbol{y}-\mathbf{X}\hat{\boldsymbol{w}})$，对 $\hat{\boldsymbol{w}}$ 求导，并令导数为零，得到 $\hat{\boldsymbol{w}}$ 最优解的闭式解。

将 $E_{\hat{\boldsymbol{w}}}$ 展开，有 $E_{\hat{\boldsymbol{w}}}=\boldsymbol{y}^{\mathrm{T}}\boldsymbol{y}-\boldsymbol{y}^{\mathrm{T}}\mathbf{X}\hat{\boldsymbol{w}}-\hat{\boldsymbol{w}}^{\mathrm{T}}\mathbf{X}^{\mathrm{T}}\boldsymbol{y}+\hat{\boldsymbol{w}}^{\mathrm{T}}\mathbf{X}^{\mathrm{T}}\mathbf{X}\hat{\boldsymbol{w}}$

对 $\hat{\boldsymbol{w}}$ 求导 $\dfrac{\partial E_{\hat{\boldsymbol{w}}}}{\partial\hat{\boldsymbol{w}}}=\dfrac{\partial\boldsymbol{y}^{\mathrm{T}}\boldsymbol{y}}{\partial\hat{\boldsymbol{w}}}-\dfrac{\partial\boldsymbol{y}^{\mathrm{T}}\mathbf{X}\hat{\boldsymbol{w}}}{\partial\hat{\boldsymbol{w}}}-\dfrac{\partial\hat{\boldsymbol{w}}^{\mathrm{T}}\mathbf{X}^{\mathrm{T}}\boldsymbol{y}}{\partial\hat{\boldsymbol{w}}}+\dfrac{\partial\hat{\boldsymbol{w}}^{\mathrm{T}}\mathbf{X}^{\mathrm{T}}\mathbf{X}\hat{\boldsymbol{w}}}{\partial\hat{\boldsymbol{w}}}$

由矩阵微分公式 $\dfrac{\partial\boldsymbol{a}^{\mathrm{T}}\boldsymbol{x}}{\partial\boldsymbol{x}}=\dfrac{\partial\boldsymbol{x}^{\mathrm{T}}\boldsymbol{a}}{\partial\boldsymbol{x}}=\boldsymbol{a}$ ；$\dfrac{\partial\boldsymbol{x}^{\mathrm{T}}\mathbf{A}\boldsymbol{x}}{\partial\boldsymbol{x}}=(\mathbf{A}+\mathbf{A}^{\mathrm{T}})\boldsymbol{x}$ ，可得

$$\frac{\partial E_{\hat{\boldsymbol{w}}}}{\partial\hat{\boldsymbol{w}}}=0-\mathbf{X}^{\mathrm{T}}\boldsymbol{y}-\mathbf{X}^{\mathrm{T}}\boldsymbol{y}+(\mathbf{X}^{\mathrm{T}}\mathbf{X}+\mathbf{X}^{\mathrm{T}}\mathbf{X})\hat{\boldsymbol{w}}=2\mathbf{X}^{\mathrm{T}}(\mathbf{X}\hat{\boldsymbol{w}}-\boldsymbol{y})$$

令上式为零可得 $\hat{\boldsymbol{w}}$ 最优解的闭式解

多元线性回归——最小二乘法

求解 $\hat{\boldsymbol{w}}$ 最优解的闭式解：$\dfrac{\partial E_{\hat{\boldsymbol{w}}}}{\partial \hat{\boldsymbol{w}}} = 2\mathbf{X}^{\mathrm{T}}\left(\mathbf{X}\hat{\boldsymbol{w}} - \boldsymbol{y}\right) = 0$

(1) 当 $\mathbf{X}^{\mathrm{T}}\mathbf{X}$ 是满秩矩阵或正定矩阵（即可逆），则有

$$\hat{\boldsymbol{w}}^{*} = \left(\mathbf{X}^{\mathrm{T}}\mathbf{X}\right)^{-1}\mathbf{X}^{\mathrm{T}}\boldsymbol{y}$$

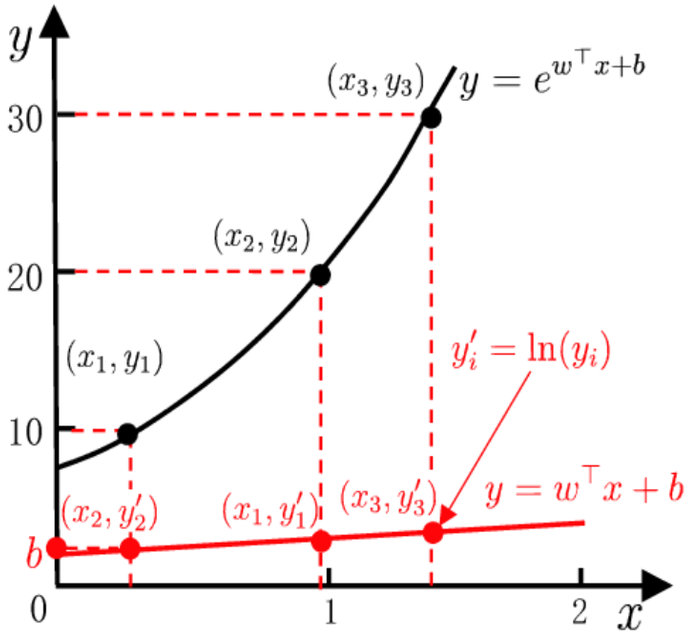
其中 $\left(\mathbf{X}^{\mathrm{T}}\mathbf{X}\right)^{-1}$ 是 $\mathbf{X}^{\mathrm{T}}\mathbf{X}$ 的逆矩阵，线性回归模型为 $f\left(\hat{\boldsymbol{x}}_i\right) = \hat{\boldsymbol{x}}_i^{\mathrm{T}}\left(\mathbf{X}^{\mathrm{T}}\mathbf{X}\right)^{-1}\mathbf{X}^{\mathrm{T}}\boldsymbol{y}$

(2) 若 $\mathbf{X}^{\mathrm{T}}\mathbf{X}$ 不是满秩矩阵，则可能存在多个解，则根据归纳偏好选择合适的解。

线性回归→广义线性回归

从线性回归推广到广义线性回归。

如**对数线性回归模型**（ log-Linear Regression ）：将输出标记的对数作为线性模型逼近的目标。



$$y = e^{w^\top x+b}$$

$$\ln y = \boldsymbol{w}^\top \boldsymbol{x} + b$$

更一般形式，**广义线性回归模型**（GLM，Generalized Linear Model）

$$y = g^{-1}\left(\boldsymbol{w}^\top \boldsymbol{x} + b\right)$$

$g(\cdot)$称为联系函数（link function），为单调可微函数

广义线性回归意义：

1）非线性模型→线性模型

2）输入空间 $\boldsymbol{X}$ 到输出空间 $\boldsymbol{y}$ 的**非线性映射**。

对数几率回归（逻辑回归、罗杰斯特回归）

几率（odds）：即胜算的意思，来自于统计学领域。

以中国 vs. 巴西足球赛为例，中国队赢面是1，输面是99，则中国队赢的几率是1/99，巴西队赢的几率是99。两队输赢程度是一样的，但 1/99=0.01010… 和 99 这两个数的尺度不同，难以做出直观的判断，而**对数几率log(odds)** 可解决这个问题。

	中国队赢	巴西队赢
几率	1/99	99
对数几率	-4.60	4.60

对 odds 加了 log 后，中国队赢和巴西队赢这两种情况的 log(odds) 的绝对值都是 4.6，一眼可看出两者的输赢程度相同，并通过正负号可判断赢面多还是赢面少。

当 log(odds) 为 0 时，赢面和输面一样多。

对数几率回归（log odds，logit）

在线性回归模型中，假定样本 $\boldsymbol{x}$ 和真实 $\boldsymbol{y}$ 呈线性关系 $\boldsymbol{y}=\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}+b$；

在二分类问题中，考虑 $\boldsymbol{y}$ 的几率形式，并假定样本 $\boldsymbol{x}$ 和 y 的对数几率呈线性关系，即**对数几率回归**。

在对数几率回归中，模型输出 y 定义为样本为正例的概率，则对数几率为：

$$\ln\frac{y}{1-y}$$

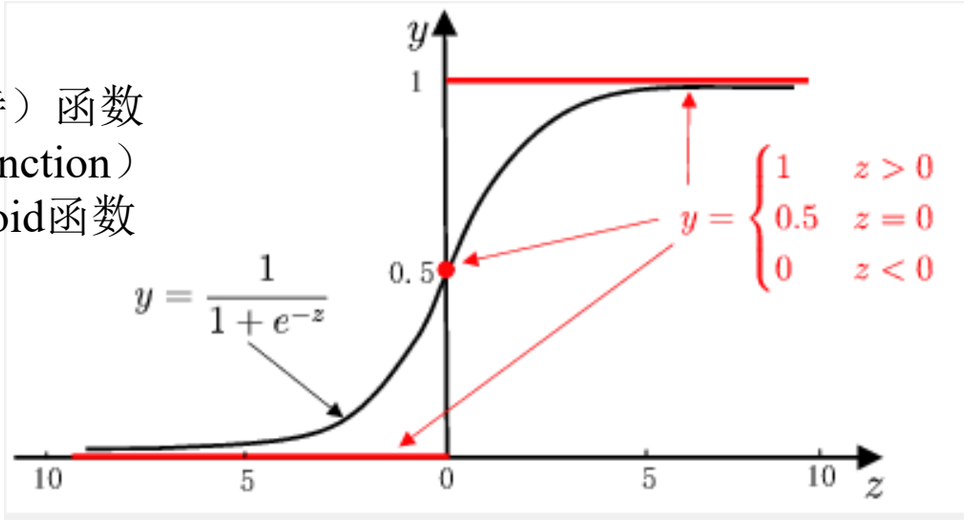
1-y为样本为负例的概率

因此，有 $\ln\frac{y}{1-y}=\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}+b$

$$\frac{y}{1-y}=e^{\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}+b}$$

$$y=\frac{e^{\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}+b}}{1+e^{\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}+b}}\xrightarrow{z=\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}+b}\frac{e^{z}}{1+e^{z}}=\frac{1}{1+e^{-z}}$$

逻辑（斯特）函数
（logistic function）
或称 Sigmoid函数



对数几率回归——对数几率函数

对数几率函数（Sigmoid函数）优点：

(1) 可将 $(-\infty,+\infty)$结果，映射到(0,1)间，作为概率；

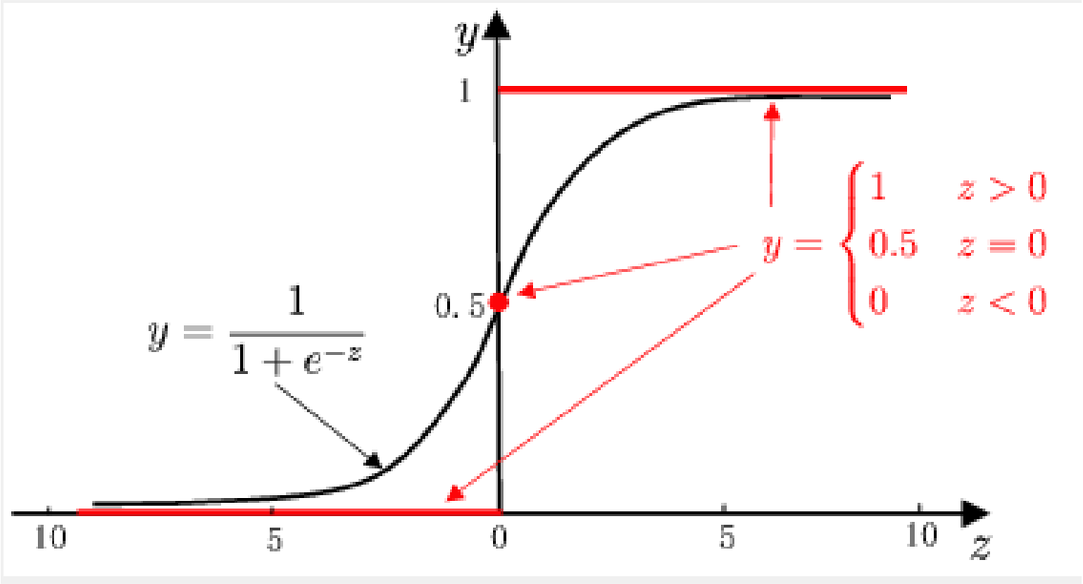
(2) 当 $z<0$时，Sigmoid(z) <0.5

当 $z>0$时，Sigmoid(z) >0.5

因此可将0.5作为分类的决策边界

(3) 单调可微、任意阶可导

$$y'=\left(\frac{1}{1+e^{-z}}\right)'=\frac{-e^{-z}}{\left(1+e^{-z}\right)^2}=\frac{1}{1+e^{-z}}\left(1-\frac{1}{1+e^{-z}}\right)=y(1-y)$$



对数几率回归——特点

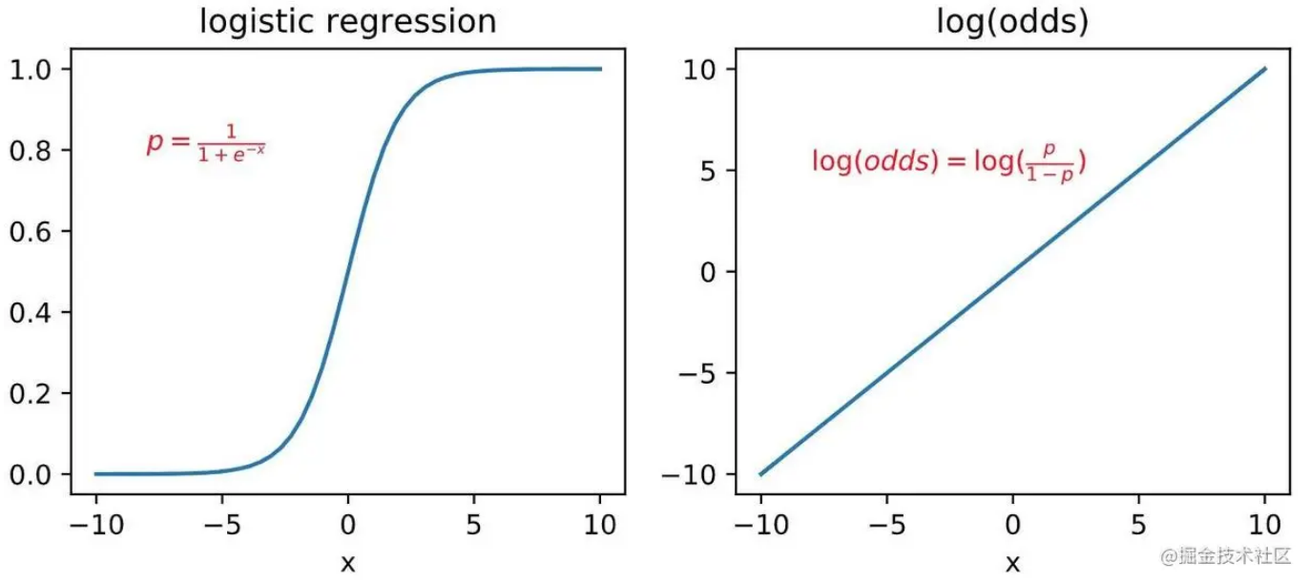
对数几率回归的特点：

1、虽然名字里有“回归”，但实际上是最为经典的分类算法之一；

2、直接对分类的可能性建模，无需事先假设数据分布，避免了假设分布不准确带来的问题；

3、分类的软判决，可得到的是“类别”的近似概率预测；

4、对数几率函数是任意阶可导的凸函数，可直接应用现有数值优化算法求取 w 和 b 的最优解。



Sigmoid函数与logit 函数转换关系；

逻辑（斯特）函数（logistic function）又称为Sigmoid函数；

log(odds) 又称为 logit 函数；

对数几率回归——最大似然法

求解 $y=\dfrac{1}{1+e^{-(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}+b)}}$ 中 w 和 b 的最优解，即求解 $\ln\dfrac{y}{1-y}=\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}+b$ 中w 和 b 的最优解

将 y 视为类后验概率 $p(y=1\mid\boldsymbol{x})$，则对数几率为：$\ln\dfrac{p(y=1\mid\boldsymbol{x})}{p(y=0\mid\boldsymbol{x})}=\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}+b$

显然有：$p(y=1\mid\boldsymbol{x})=\dfrac{e^{\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}+b}}{1+e^{\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}+b}}=p_1(\boldsymbol{x})$　　$p(y=0\mid\boldsymbol{x})=\dfrac{1}{1+e^{\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}+b}}=p_0(\boldsymbol{x})$

为便于讨论，令 $\boldsymbol{\beta}=(\boldsymbol{w};b)$，$\hat{\boldsymbol{x}}=(\boldsymbol{x};1)$，则 $\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}+b$ 可简写为 $\boldsymbol{\beta}^{\mathrm{T}}\hat{\boldsymbol{x}}$

采用对数形式：$\ln p(y_i=1\mid\boldsymbol{x})=\boldsymbol{\beta}^{\mathrm{T}}\hat{\boldsymbol{x}}-\ln\left(1+e^{\boldsymbol{\beta}^{\mathrm{T}}\hat{x}}\right)$　　$\ln p(y_i=0\mid\boldsymbol{x})=-\ln\left(1+e^{\boldsymbol{\beta}^{\mathrm{T}}\hat{x}}\right)$

考虑到 y 取值只是 0 或 1，则统一的对数形式：$\ln p(y_i\mid\boldsymbol{x})=y_i\boldsymbol{\beta}^{\mathrm{T}}\hat{\boldsymbol{x}}-\ln\left(1+e^{\boldsymbol{\beta}^{\mathrm{T}}\hat{x}}\right)$

对数几率回归——最大似然法

似然函数（likelihood function）

$$l(\boldsymbol{\beta}) = p(\boldsymbol{y} \mid \boldsymbol{x}; \boldsymbol{\beta}) = p(y_1 \mid x_1, y_2 \mid x_2, \ldots, y_m \mid x_m; \boldsymbol{\beta}) = \prod_{i=1}^{m} p(y_i \mid x_i; \boldsymbol{\beta})$$ **当满足i.i.d**

对数似然函数 $\ell(\boldsymbol{\beta}) = \ln \prod_{i=1}^{m} p(y_i \mid x_i; \boldsymbol{\beta}) = \sum_{i=1}^{m} \ln p(y_i \mid x_i; \boldsymbol{\beta}) = \sum_{i=1}^{m} y_i \boldsymbol{\beta}^{\mathrm{T}} \hat{\boldsymbol{x}}_i - \ln\left(1 + e^{\boldsymbol{\beta}^{\mathrm{T}} \hat{\boldsymbol{x}}_i}\right)$

$\boldsymbol{\beta}$ 的最优解应满足似然函数值最大（对数似然函数值也为最大），即有最大似然估计:

最大似然估计：

如果在参数 $\boldsymbol{\beta} = \hat{\boldsymbol{\beta}}$ 下，$\ell(\boldsymbol{\beta})$ 最大，则 $\hat{\boldsymbol{\beta}}$ 应是最佳的参数值，称作最大似然估计量。

为了便于求解，将最大化似然函数转化为最小化问题 $J(\boldsymbol{\beta}) = \frac{1}{m} \sum_{i=1}^{m} -y_i \boldsymbol{\beta}^{\mathrm{T}} \hat{\boldsymbol{x}} + \ln\left(1 + e^{\boldsymbol{\beta}^{\mathrm{T}} \hat{x}}\right)$

加负号将最大化转为最小化问题。

对数几率回归——最大似然法求解

求解 $\boldsymbol{\beta}$ 的最优解，即 $\hat{\boldsymbol{\beta}} = \arg\min_{\boldsymbol{\beta}} J(\boldsymbol{\beta})$

可以采用经典数值优化算法，如梯度下降法、牛顿法等求解 $\boldsymbol{\beta}$ 的最优解，

牛顿法求解

牛顿法第 t+1 轮迭代解的更新公式 $\boldsymbol{\beta}^{t+1} = \boldsymbol{\beta}^{t} - \left(\frac{\partial^2 J(\boldsymbol{\beta})}{\partial \boldsymbol{\beta} \cdot \partial \boldsymbol{\beta}^{\mathrm{T}}}\right)^{-1} \frac{\partial J(\boldsymbol{\beta})}{\partial \boldsymbol{\beta}}$

其中关于 $\boldsymbol{\beta}$ 的一阶、二阶导数分别为：

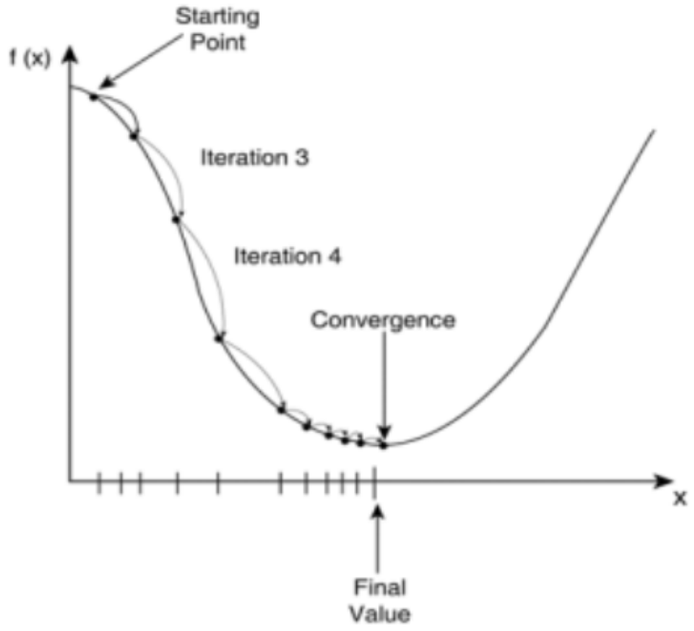
$$\frac{\partial J(\boldsymbol{\beta})}{\partial \boldsymbol{\beta}} = -\frac{1}{m}\sum_{i=1}^{m} \hat{\boldsymbol{x}}_i \left(y_i - p_1(\hat{\boldsymbol{x}}_i;\boldsymbol{\beta})\right)$$

$$\frac{\partial^2 J(\boldsymbol{\beta})}{\partial \boldsymbol{\beta} \cdot \partial \boldsymbol{\beta}^{\mathrm{T}}} = \frac{1}{m}\sum_{i=1}^{m} \hat{\boldsymbol{x}}_i \hat{\boldsymbol{x}}_i^{T} p_1(\hat{\boldsymbol{x}}_i;\boldsymbol{\beta})\left(1 - p_1(\hat{\boldsymbol{x}}_i;\boldsymbol{\beta})\right)$$

其中 $p_1(\hat{\boldsymbol{x}}_i;\boldsymbol{\beta}) = \frac{e^{\boldsymbol{\beta}^{\mathrm{T}}\hat{\boldsymbol{x}}}}{1+e^{\boldsymbol{\beta}^{\mathrm{T}}\hat{\boldsymbol{x}}}}$

对数几率回归——最大似然法求解

$$\hat{\boldsymbol{\beta}} = \arg\min_{\boldsymbol{\beta}} J(\boldsymbol{\beta})$$



梯度下降法求解

$$\frac{\partial J(\boldsymbol{\beta})}{\partial \boldsymbol{\beta}} = -\frac{1}{m}\sum_{i=1}^{m} \hat{\boldsymbol{x}}_i \left(y_i - p_1(\hat{\boldsymbol{x}}_i;\boldsymbol{\beta})\right)$$

将 $\frac{\partial J(\beta)}{\partial \beta}$ 带入参数 $\boldsymbol{\beta}$ 的更新公式：$\boldsymbol{\beta}^* = \boldsymbol{\beta} - \eta\frac{\partial J}{\partial \boldsymbol{\beta}}$ ，得到最终的参数 $\boldsymbol{\beta}$ 更新公式

$$\boldsymbol{\beta}^* = \boldsymbol{\beta} + \frac{\eta}{m}\sum_{i=1}^{m} \hat{\boldsymbol{x}}_i \left(y_i - p_1(\hat{\boldsymbol{x}}_i;\boldsymbol{\beta})\right)$$

η 为学习率

- **线性模型基本形式**
  线性方程，基本形式，…
- **线性回归**
  参数估计，最小二乘，对数几率回归，…
- **线性判别分析**
  基本概念，广义线性判别，几何性质 …
- **经典线性判别准则**
  Fisher判别，感知器准则、梯度法 …
- **非线性判别函数**
  分段线性，…

线性判别分析——基本概念

判别函数 (discriminant function)

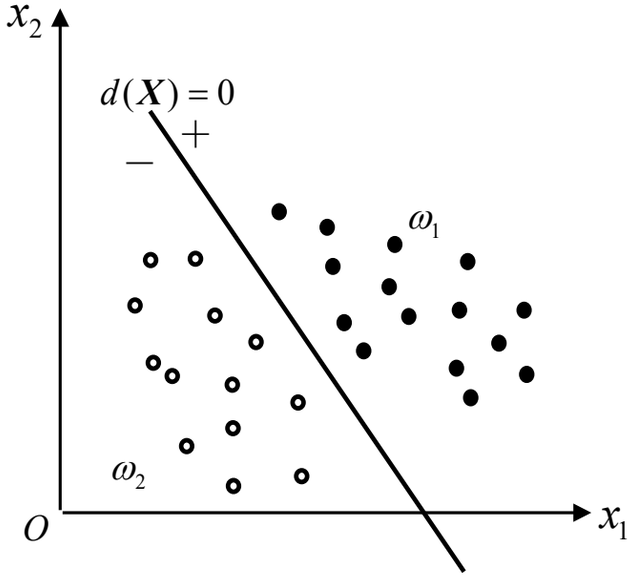
定义：直接用来对模式进行分类的准则函数。

若分属于ω_1，ω_2的两类模式可用一方程 $d(\boldsymbol{X})=0$来划分，那么称$d(\boldsymbol{X})$ 为判别函数，或称判决函数、决策函数。

例：一个二维的两类判别问题，模式分布如图示，这些分属于ω_1，ω_2两类的模式可用一直线方程 $d(\boldsymbol{X})=0$来划分。

$$d(\boldsymbol{X})=w_1x_1+w_2x_2+w_3=0$$

式中：　x_1,x_2 为坐标变量，w_1,w_2,w_3为方程参数。



两类二维模式的分布

判别函数 (discriminant function)

将某一未知模式 $\boldsymbol{X}$ 代入：　$d(\boldsymbol{X}) = w_1 x_1 + w_2 x_2 + w_3$

若 $d(\boldsymbol{X}) > 0$，则 $\boldsymbol{X} \in \omega_1$ 类；

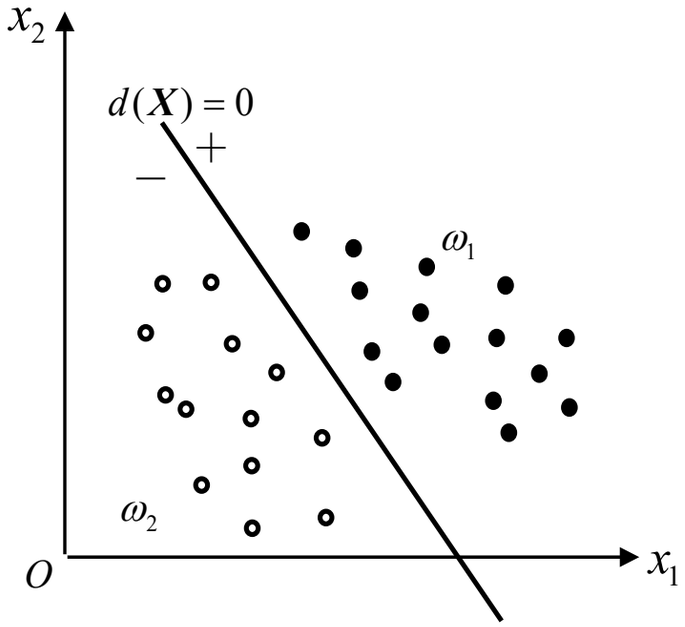
若 $d(\boldsymbol{X}) < 0$，则 $\boldsymbol{X} \in \omega_2$ 类；

若 $d(\boldsymbol{X}) = 0$，则 $\boldsymbol{X} \in \omega_1$ 或 $\boldsymbol{X} \in \omega_2$ 或拒绝

方程 $d(\boldsymbol{X})$ 定义了一个决策面，

维数=3时：判别面为一个平面。

维数>3时：判别面为一个超平面。

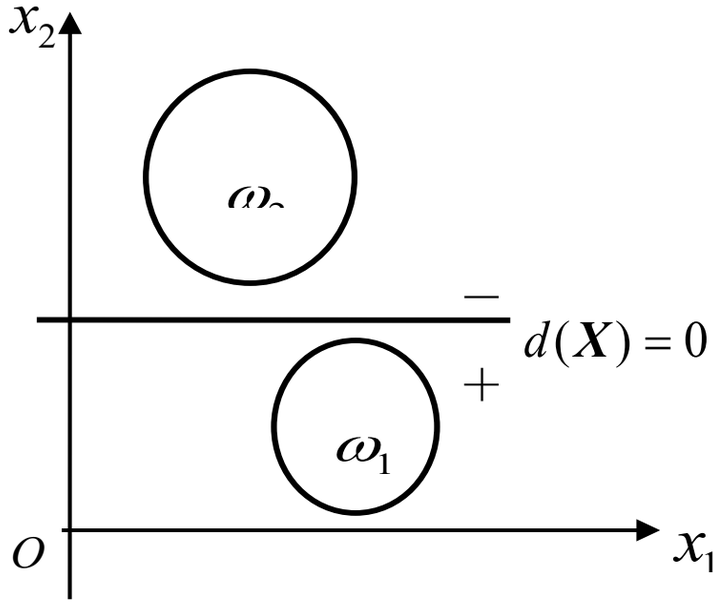


判别函数 (discriminant function)

判别函数正负值的确定

判别界面的正负侧，是在训练判别函数的权值时确定的。

$d(\boldsymbol{X})$ 表示的是一种分类的标准，它可以是1、2、3维的，也可以是更高维的。



判别函数正负的确定

判别函数 (discriminant function)

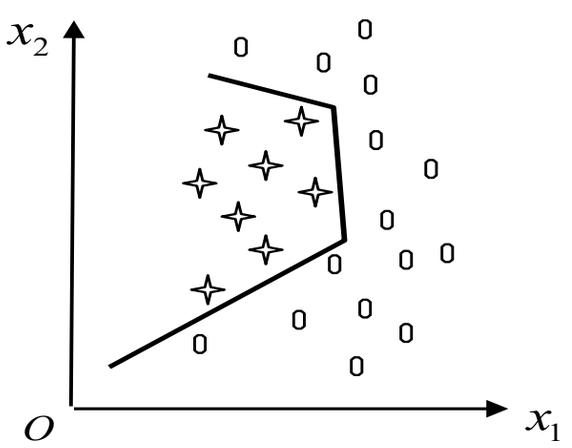
确定判别函数的两个因素

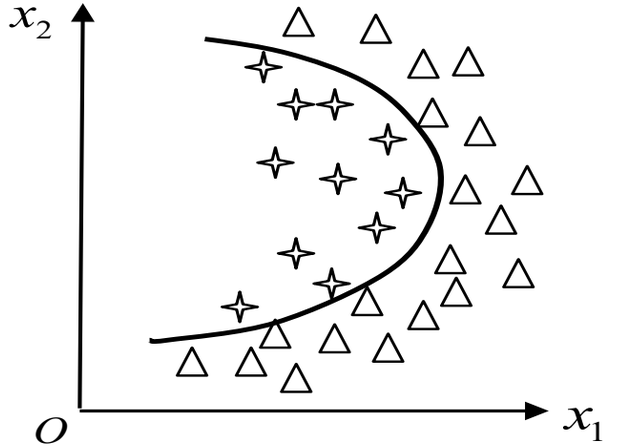
1）判决函数 $d(\boldsymbol{X})$ 的几何性质。它可以是线性的或非线性的函数，维数在特征提取时已经确定。

2）判决函数 $d(\boldsymbol{X})$ 的系数，由所给的模式样本确定。

如：已知三维线性分类 —— 判决函数的性质就确定了判决函数的形式： $d(\boldsymbol{X}) = w_1 x_1 + w_2 x_2 + w_3 x_3 + w_4$

例：非线性判决函数





线性判别函数

线性判别函数的一般形式

将二维模式推广到n维，线性判别函数的一般形式为：$d(\boldsymbol{X}) = w_1 x_1 + w_2 x_2 + \ldots + w_n x_n + w_{n+1} = \boldsymbol{W}_0^{\mathrm{T}} \boldsymbol{X} + w_{n+1}$

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。

增广向量的形式：

$$d(\boldsymbol{X}) = w_1 x_1 + w_2 x_2 + \ldots + w_n x_n + w_{n+1} \cdot 1 = [w_1, w_2, \ldots, w_n, w_{n+1}] \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \\ 1 \end{bmatrix} = \boldsymbol{W}^{\mathrm{T}} \boldsymbol{X}$$

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，

线性判别函数

线性判别函数的性质

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，$\ldots\omega_M$，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

线性判别函数

(1)多类情况1：$\omega_i/\overline{\omega}_i$ 两分法

用线性判别函数将属于ω_i类的模式与其余不属于ω_i类的模式分开。

$$d_i(\boldsymbol{X})=\boldsymbol{W}_i^{\mathrm{T}}\boldsymbol{X}\begin{cases}>0, & \text{若}\boldsymbol{X}\in\omega_i\\ <0, & \text{若}\boldsymbol{X}\in\overline{\omega}_i\end{cases}\qquad i=1,\cdots,\ M$$

每个类别用一个判别面分割，因此M类需要M个线性判别函数。

识别分类时：

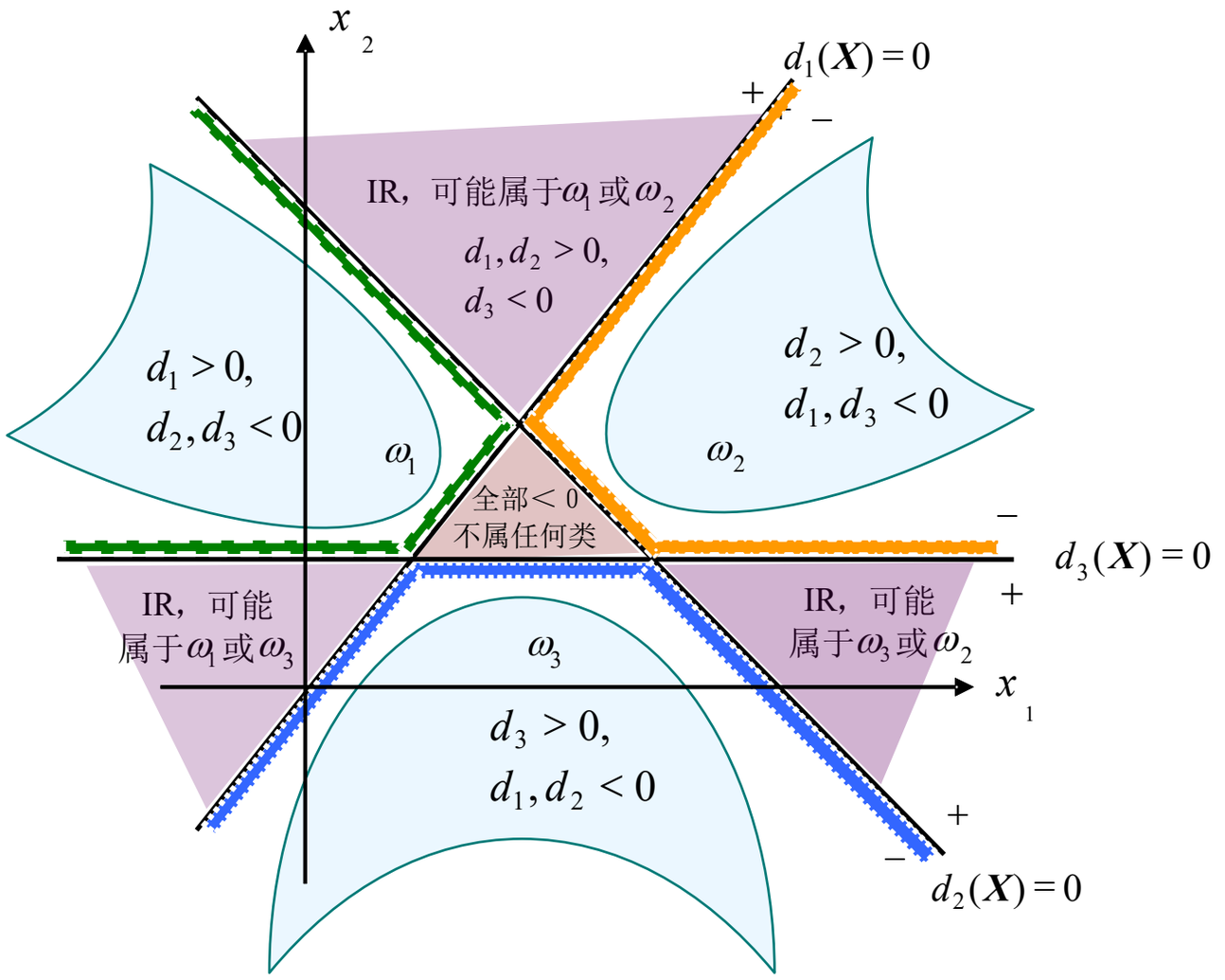
将某个待分类模式 $\boldsymbol{X}$ 分别代入 M 个类的$d(\boldsymbol{X})$中，若只有$d_i(\boldsymbol{X})>0$，其他$d(\boldsymbol{X})$均<0，则判为ω_i类。

线性判别函数

(1)多类情况1：$\omega_i/\overline{\omega}_i$ 两分法

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。

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)。



线性判别函数

(1)多类情况1：$\omega_i/\overline{\omega}_i$ 两分法

例 设有一个三类问题，其判别式为：

$$d_1(X)=-x_1+x_2+1$$
$$d_2(X)=x_1+x_2-4$$
$$d_3(X)=-x_2+1$$

现有一模式，$X=[-7,5]^T$，试判定应属于哪类？
并画出三类模式的分布区域。

$X(-7,5)$

解：将 $X=[-7,5]^T$ 代入上三式，有：

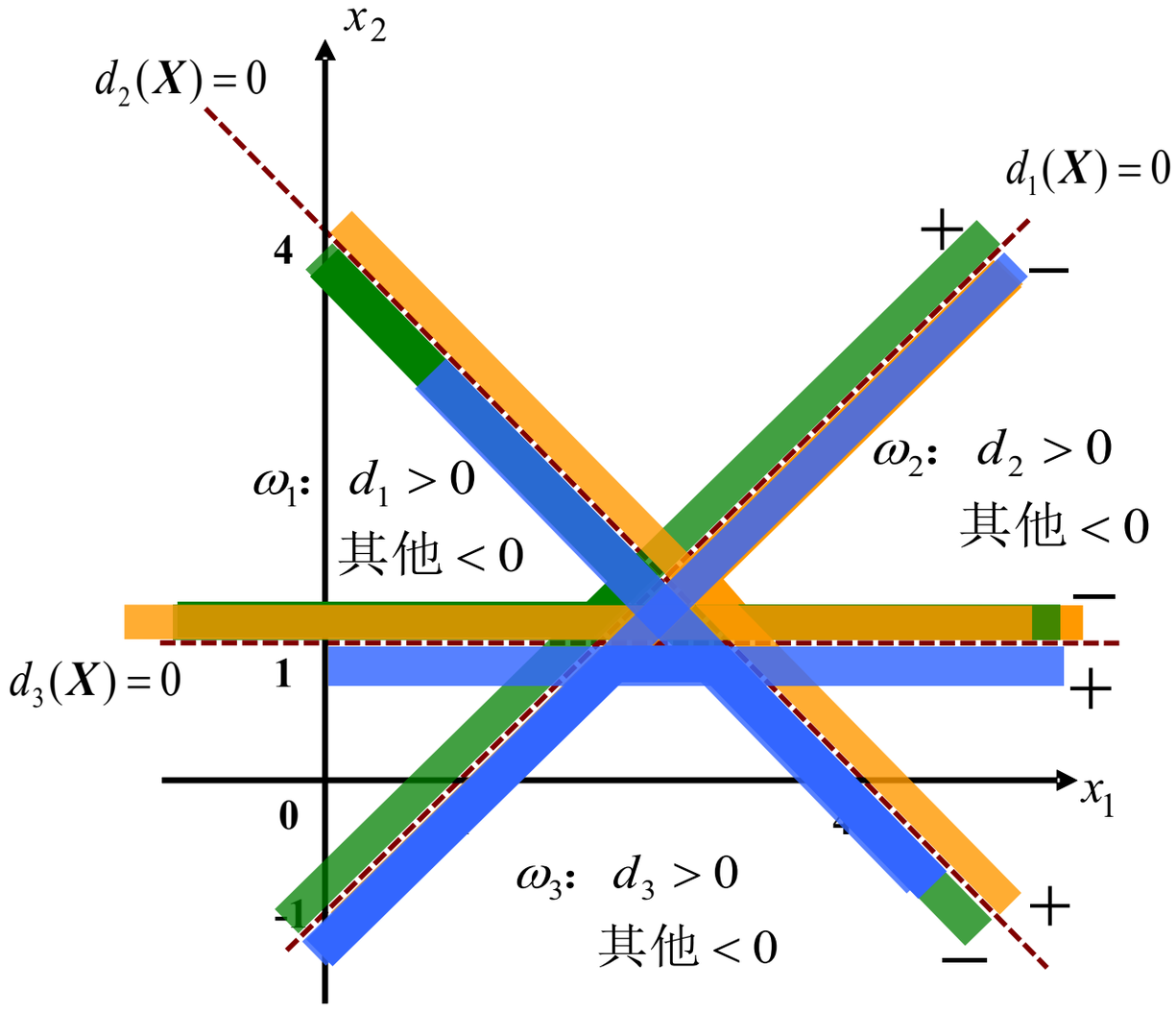
$$d_1(X)=7+5+1=13>0$$

$$d_2(X)=-7+5-4=-6<0$$

$$d_3(X)=-5+1=-4<0$$

$$\because d_1(X)>0,\quad d_2(X),d_3(X)<0\quad \therefore X\in\omega_1$$

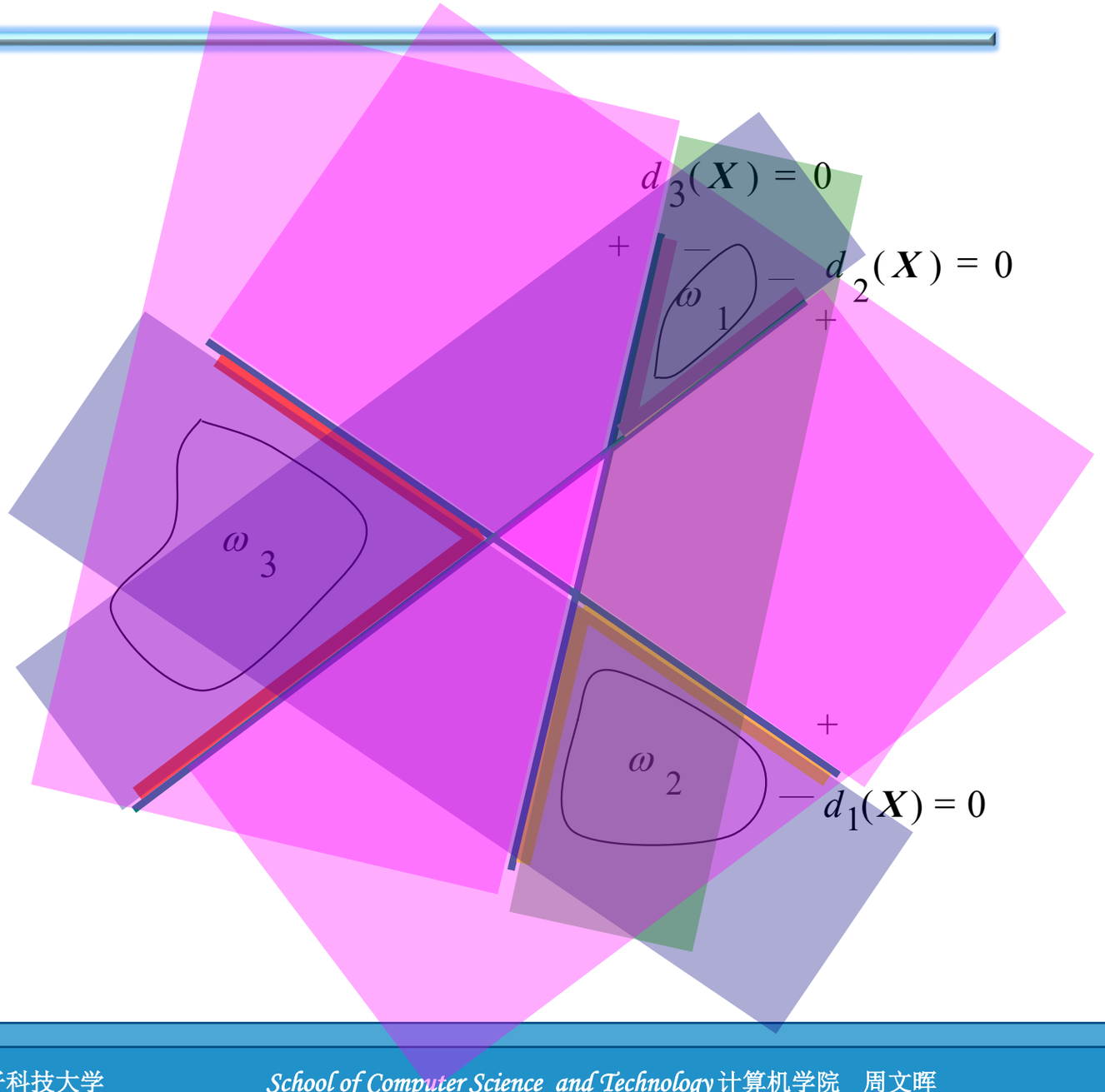
三个判别界面分别为：$d_1(X)=0$　$d_2(X)=0$　$d_3(X)=0$



线性判别函数

(1)多类情况1：$\omega_i/\bar{\omega}_i$ 两分法

例 已知 $d_i(\boldsymbol{X})$ 的位置和正负侧，分析三类模式的分布区域



线性判别函数

(2) 多类情况2：ω_i/ω_j两分法

一个判别界面只能分开两个类别，不能把其余所有的类别都分开。判决函数为：　　；

$$d_{ij}(\boldsymbol{X})=\boldsymbol{W}_{ij}^{\mathrm{T}}\boldsymbol{X} \quad 且 \quad d_{ji}=-d_{ij}$$

判别函数性质：$d_{ij}(\boldsymbol{X})>0,\ \forall j\neq i;\ j=1,2,...,M,$　则$\boldsymbol{X}\in\omega_i$

识别分类时：

M类需要$M(M\text{-}1)/2$个线性判别函数。

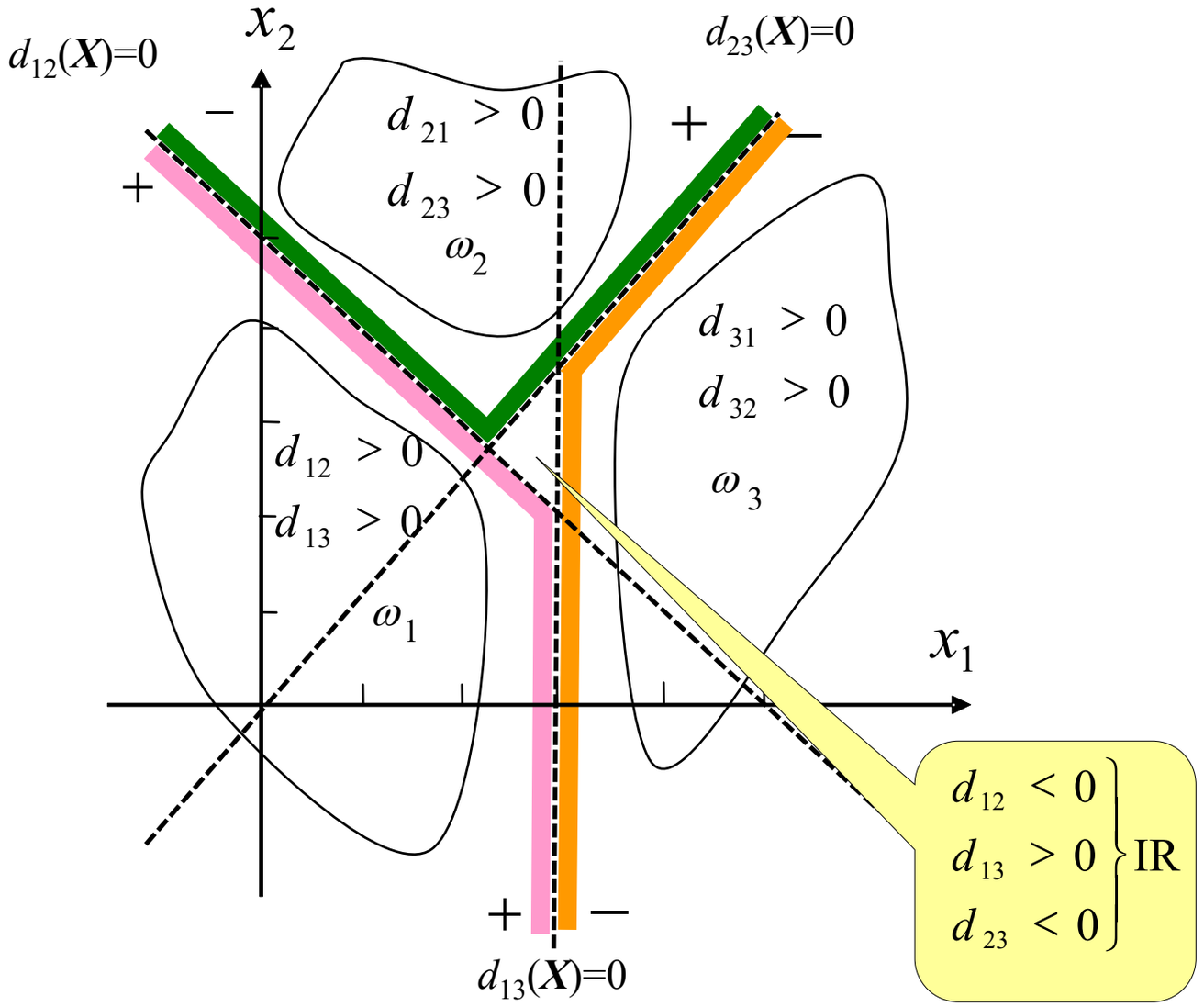
在 M 类模式中，与i 有关的M-1个判决函数全为正时，$\boldsymbol{X}\in\omega_i$。其中若有一个为负，则为IR区。

如：三类问题，如果 $d_{12}(\boldsymbol{X})>0, d_{13}(\boldsymbol{X})>0$；则 $\boldsymbol{X}\in\omega_1$ 类，而 $d_{23}(\boldsymbol{X})$ 在判别 ω_1 类模式时不起作用。

线性判别函数

(2) 多类情况2：ω_i/ω_j两分法

如：三类问题



线性判别函数

(2) 多类情况2：ω_i/ω_j两分法

例 一个三类问题，三个判决函数为：

$$d_{12}(\boldsymbol{X})=-x_1-x_2+5 \qquad d_{13}(\boldsymbol{X})=-x_1+3 \qquad d_{23}(\boldsymbol{X})=-x_1+x_2$$

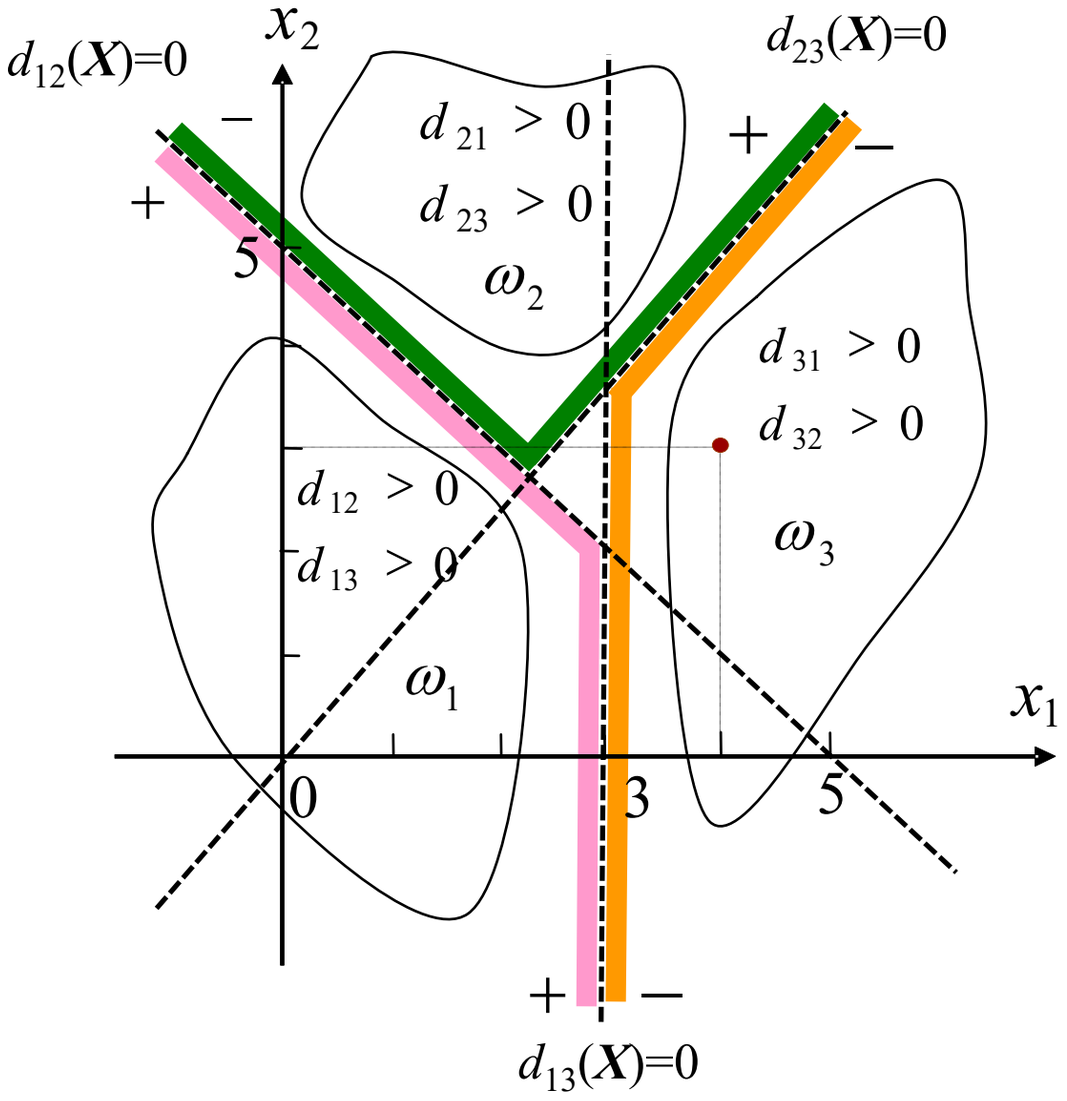
问模式 $\boldsymbol{X}=[4,3]^{\mathrm{T}}$ 属于哪类？

解：计算得 $d_{12}(\boldsymbol{X})=-2,\ d_{13}(\boldsymbol{X})=-1,\ d_{23}(\boldsymbol{X})=-1$

可写成：$d_{21}(\boldsymbol{X})=2,\ d_{31}(\boldsymbol{X})=1,\ d_{32}(\boldsymbol{X})=1$

$$\left.\begin{array}{l} d_{31}(\boldsymbol{X})>0 \\ d_{32}(\boldsymbol{X})>0 \end{array}\right\} \Rightarrow \boldsymbol{X}=[4,3]^{\mathrm{T}}\in\omega_3$$

与 $d_{12}(\boldsymbol{X})$ 值无关。



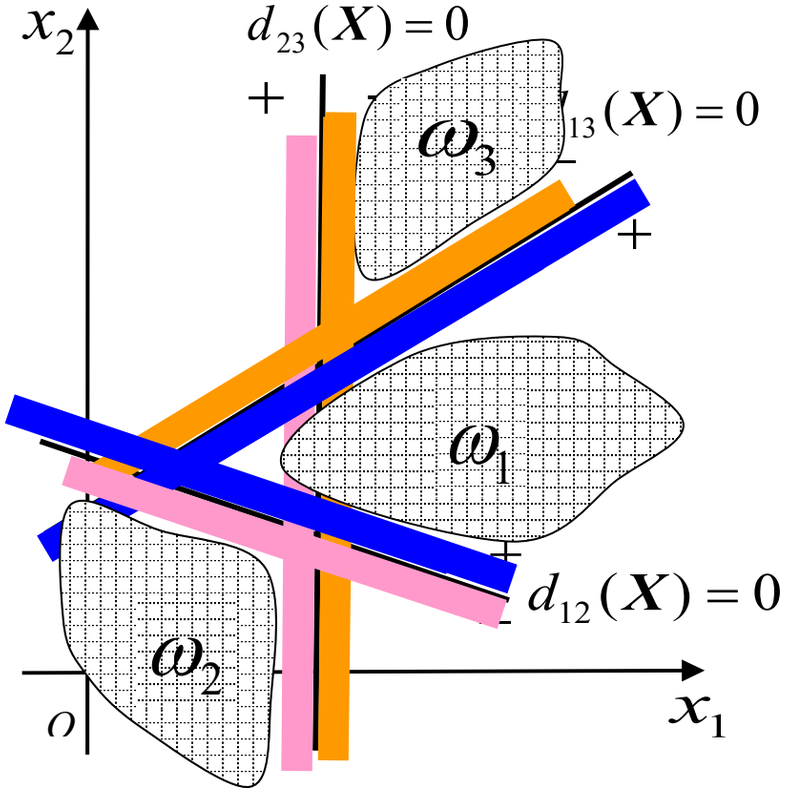
线性判别函数

(2) 多类情况2：ω_i/ω_j两分法

分类时：每分离出一类，需要与 i 有关的M-1个判决函数；要分开M类模式，共需$M(M\text{-}1)/2$个判决函数。对三类问题需要3(3-1)/2=3个判决函数。

即：每次从M类中取出两类的组合：

$$C_M^2 = \frac{M(M-1)}{2!}$$



例 已知$d_{ij}(\boldsymbol{X})$的位置和正负侧，分析三类模式的分布区域 。

线性判别函数

(3) 多类情况3：ω_i/ω_j两分法特例

当ω_i/ω_j两分法中的判别函数 $d_{ij}(\boldsymbol{X})$，可以分解为 $d_{ij}(\boldsymbol{X})=d_i(\boldsymbol{X})-d_j(\boldsymbol{X})$ 时，

那么 $d_i(\boldsymbol{X})>d_j(\boldsymbol{X})$ 就相当于多类情况2中的 $d_{ij}(\boldsymbol{X})>0$。

因此对判别函数 $d_i(\boldsymbol{X})=\boldsymbol{W}_i^{\mathrm{T}}\boldsymbol{X}$，　$i=1,\cdots,M$ 的M类情况，判别函数性质为：

$$d_i(\boldsymbol{X})>d_j(\boldsymbol{X}),\ \forall j\neq i;\ j=1,2,\cdots,M,\quad \text{则}\boldsymbol{X}\in\omega_i$$

或：

$$d_i(\boldsymbol{X})=\max\left\{d_k(\boldsymbol{X}),\quad k=1,\cdots,M\right\},\quad \text{则}\ \boldsymbol{X}\in\omega_i$$

识别分类时：

判别界面需要做差值。对ω_i类，应满足$d_i>$其他所有d

线性判别函数

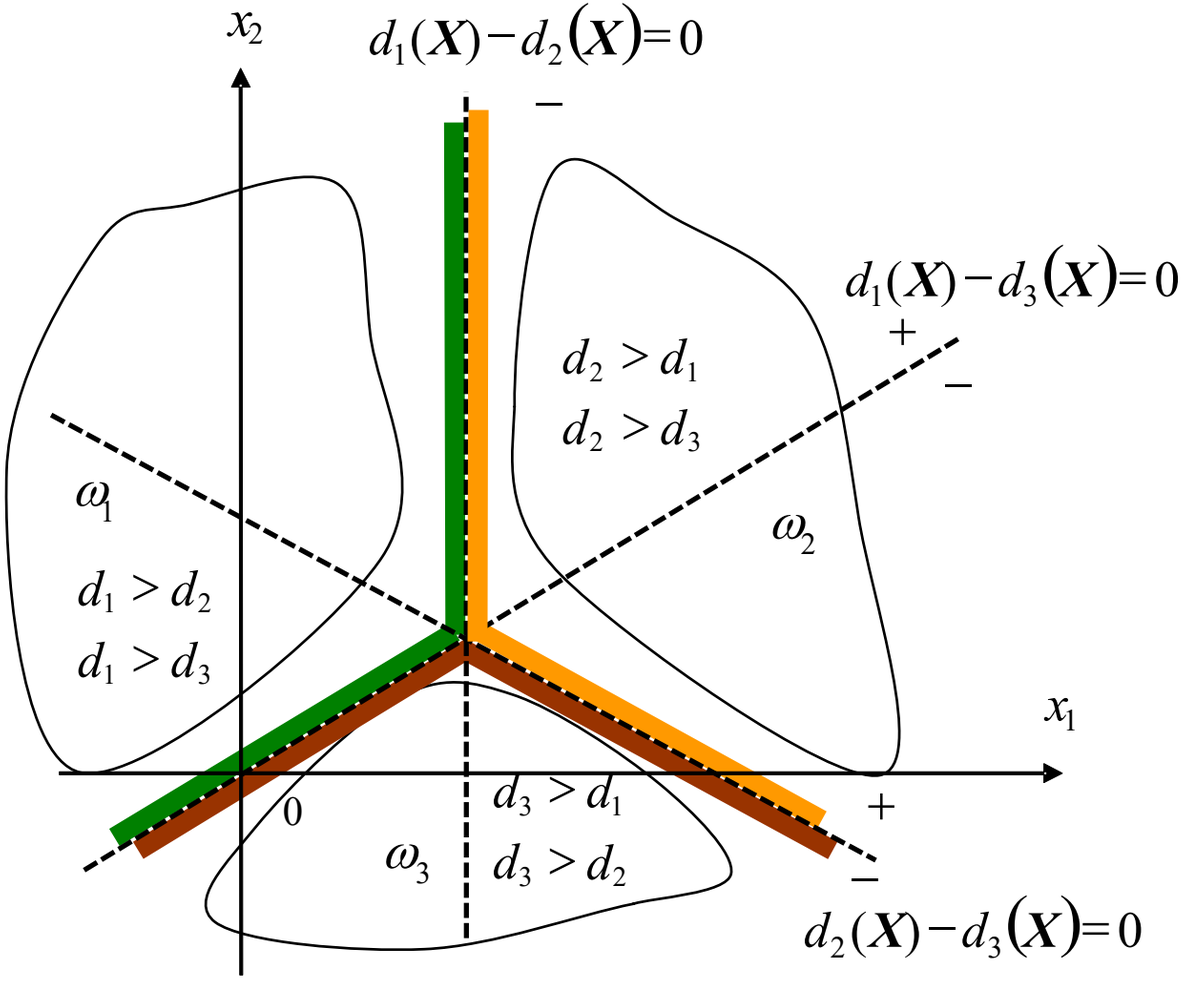
(3) 多类情况3：ω_i/ω_j两分法特例

特点:

a) 是第二种情况的特例。由于$d_{ij}(\boldsymbol{X})= d_i(\boldsymbol{X}) - d_j(\boldsymbol{X})$，若在第三种情况下可分，则在第二种情况下也可分，但反过来不一定。

b) 除边界区外，没有不确定区域。如右图三条判别线相交于一点，该点满足 $d_1=d_2=d_3$。

c) 把 M 类情况分成了$(M-1)$个两类问题。并且 ω_i 类的判别界面全部与$\omega_j\ (\forall j \neq i)$类的判别界面相邻。



线性判别函数

(3) 多类情况3：ω_i/ω_j两分法特例

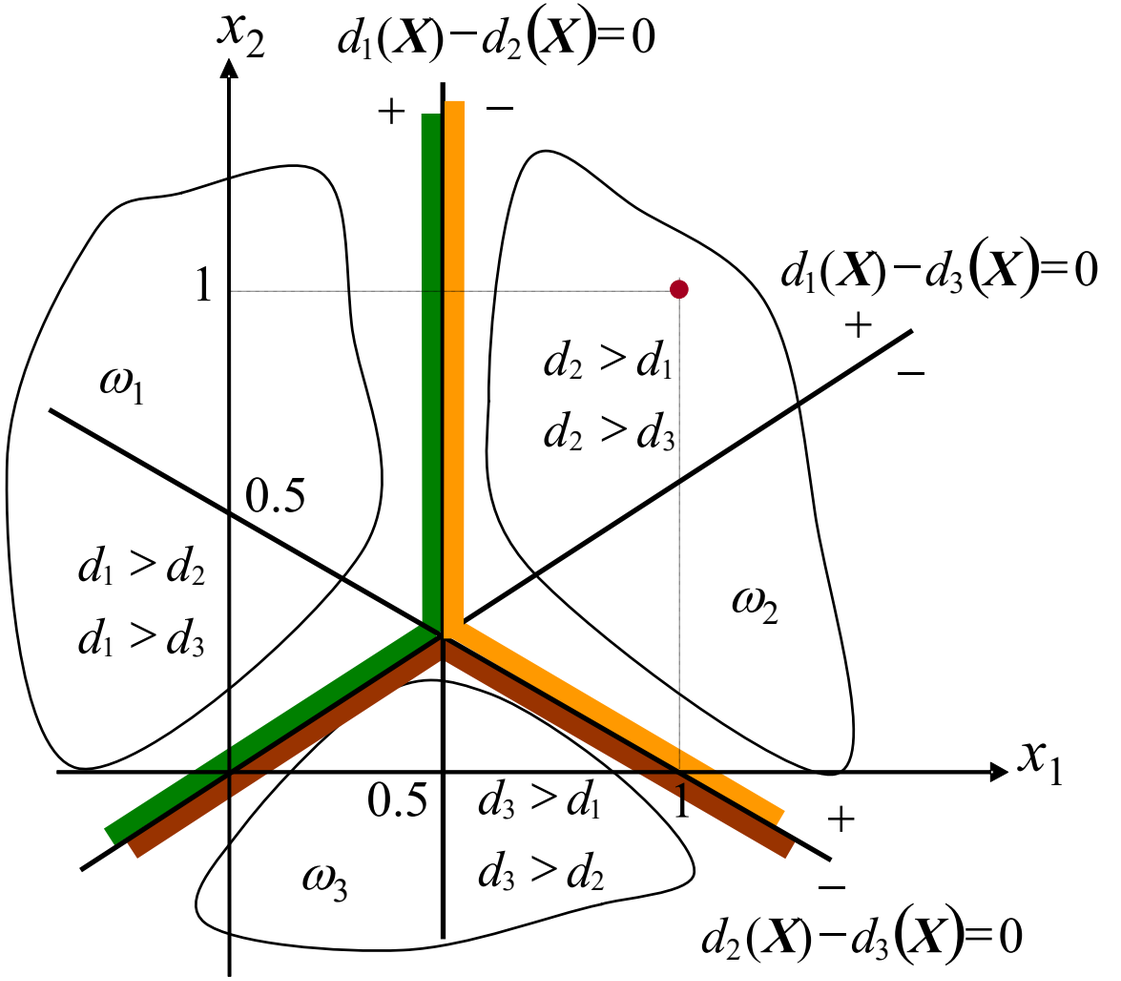
例 一个三类模式（$M=3$）分类器，其判决函数为：

$$d_1(\boldsymbol{X})=-x_1+x_2$$
$$d_2(\boldsymbol{X})=x_1+x_2-1$$
$$d_3(\boldsymbol{X})=-x_2$$

试判断 $\boldsymbol{X}_0$=[1,1]$^{\mathrm{T}}$ 属于哪一类，且分别给出三类的判决界面。

解：

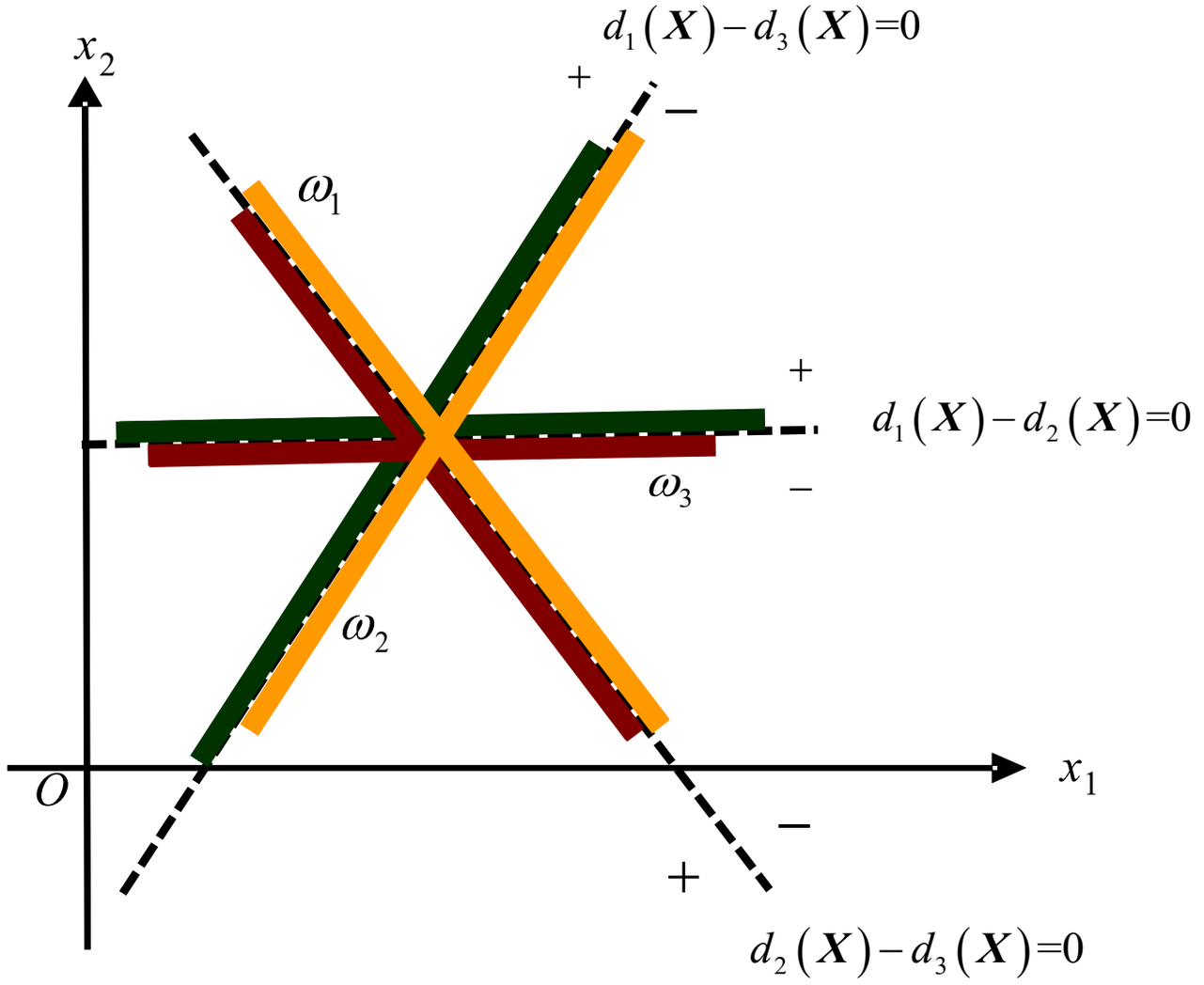
$$\left.\begin{aligned}d_1(\boldsymbol{X}_0)&=-1+1=0\\d_2(\boldsymbol{X}_0)&=1+1-1=1\\d_3(\boldsymbol{X}_0)&=-1\end{aligned}\right\}\Rightarrow\left.\begin{aligned}d_2(\boldsymbol{X}_0)&>d_1(\boldsymbol{X}_0)\\d_2(\boldsymbol{X}_0)&>d_3(\boldsymbol{X}_0)\end{aligned}\right\}\Rightarrow\boldsymbol{X}_0\in\omega_2$$



线性判别函数

(3) 多类情况3：ω_i/ω_j两分法特例

例 已知判决界面的位置和正负侧，分析三类模式的分布区域 。



线性判别函数

小结

(1) 明确概念：线性可分。

一旦线性判别函数的系数 $\boldsymbol{W}_k$ 被确定以后，这些函数就可以作为模式分类的基础。

(2) $\omega_i/\overline{\omega}_i$ 与 ω_i/ω_j 分法的比较：

对于M类模式的分类，$\omega_i/\overline{\omega}_i$ 两分法共需要M个判别函数，但 ω_i/ω_j 两分法需要$M(M\text{-}1)/2$个。

当时M>3时，后者需要更多个判别式（缺点），但对模式的线性可分的可能性要更大一些（优点）。

原因：

一种类别模式的分布要比M-1类模式的分布更为聚集， ω_i/ω_j 分法受到的限制条件少，故线性可分的可能性大。

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。

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：

$$d(\boldsymbol{X})=w_1 f_1(\boldsymbol{X})+w_2 f_2(\boldsymbol{X})+\ldots+w_k f_k(\boldsymbol{X})+w_{k+1}=\sum_{i=1}^{k+1} w_i f_i(\boldsymbol{X})$$

式中$\{f_i(\boldsymbol{X})\ ,\ i=1,2,...,k\}$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。

广义线性判别函数

广义形式的模式向量定义为：$\boldsymbol{X}^{*}=[x_1^{*},x_2^{*},\cdots,x_k^{*},1]^{\mathrm{T}}=[f_1(\boldsymbol{X}),f_2(\boldsymbol{X}),\cdots,f_k(\boldsymbol{X}),1]^{\mathrm{T}}$

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

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。

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$x\rightarrow\boldsymbol{y}$，使得$z$化为$\boldsymbol{y}$的线性判别函数。

解：映射X→Y如下：

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

$$z=g(x)=h(\mathbf{y})=\mathbf{w}^{T}\mathbf{y}=\sum_{i=1}^{4}w_i y_i$$

广义线性判别函数

例 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：$\boldsymbol{X}^* = [x_1^2, x_1x_2, x_2^2, x_1, x_2, 1]^{\mathrm{T}}$　　$\mathbf{w} = [w_{11}, w_{12}, w_{22}, w_1, w_2, w_3]^{\mathrm{T}}$

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

广义线性判别函数

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

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

广义线性判别函数 $z = g(\mathbf{x}) = h(\mathbf{y}) = \mathbf{w}^T\mathbf{y}$

广义线性判别函数

存在的问题：

1) 非线性变换可能非常复杂 。

2) 维数大大增加： 维数灾难。

随着小样本学习理论和支持向量机的迅速发展，广义线性判别函数的 “维数灾难” 问题在一定程度上找到了解决的办法。

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

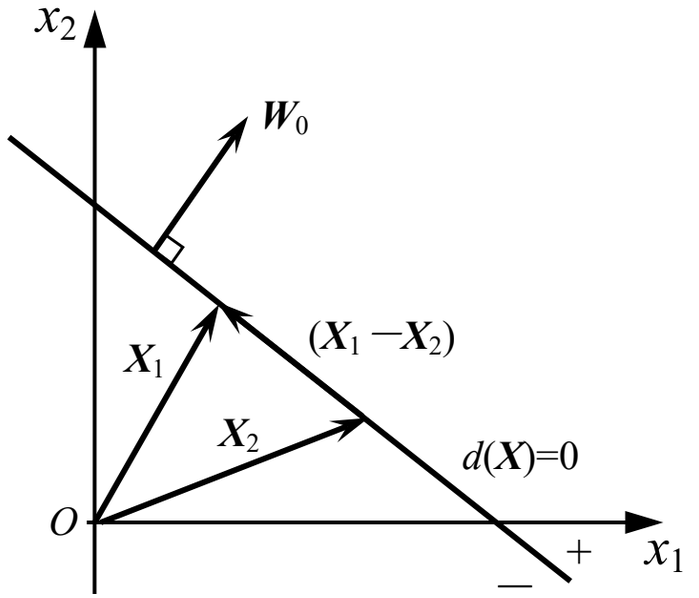
线性判别函数的几何性质

(1) 模式向量$\boldsymbol{X}_1$和$\boldsymbol{X}_2$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。

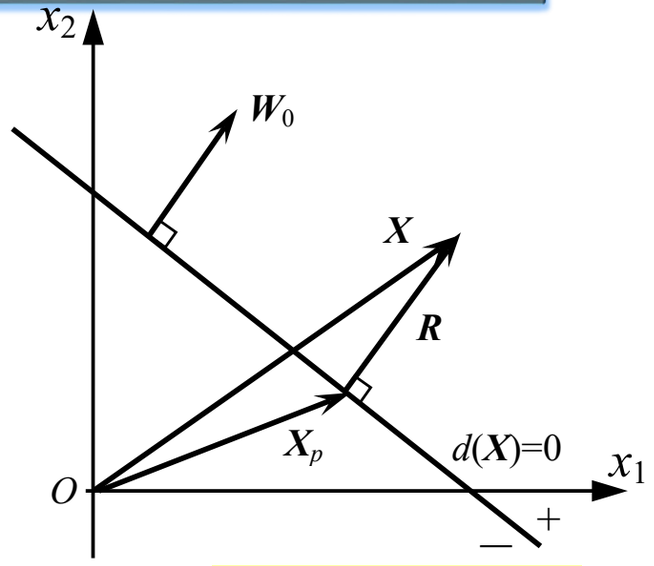


设超平面的单位法线向量为$\boldsymbol{U}$：

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

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

线性判别函数的几何性质



$$d(X)=W_0^{\mathrm{T}}X+w_{n+1}$$

(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：

$$R=r\cdot U=r\frac{W_0}{\|W_0\|}$$

r：X到超平面的垂直距离。有： $X=X_p+R=X_p+r\frac{W_0}{\|W_0\|}$

$$d(X)=W_0^{\mathrm{T}}\left(X_p+r\frac{W_0}{\|W_0\|}\right)+w_{n+1}=W_0^{\mathrm{T}}X_p+w_{n+1}+r\frac{W_0^{\mathrm{T}}W_0}{\|W_0\|}$$

由于X_p在超平面上，即$d(X_p)=0$;

$$d(X)=r\frac{W_0^{\mathrm{T}}W_0}{\|W_0\|}=r\|W_0\|$$

—— 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： $r=\frac{d(X)}{\|W_0\|}$ —— 点X到超平面的代数距离（带正负号）正比于$d(X)$函数值。

线性判别函数的几何性质

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

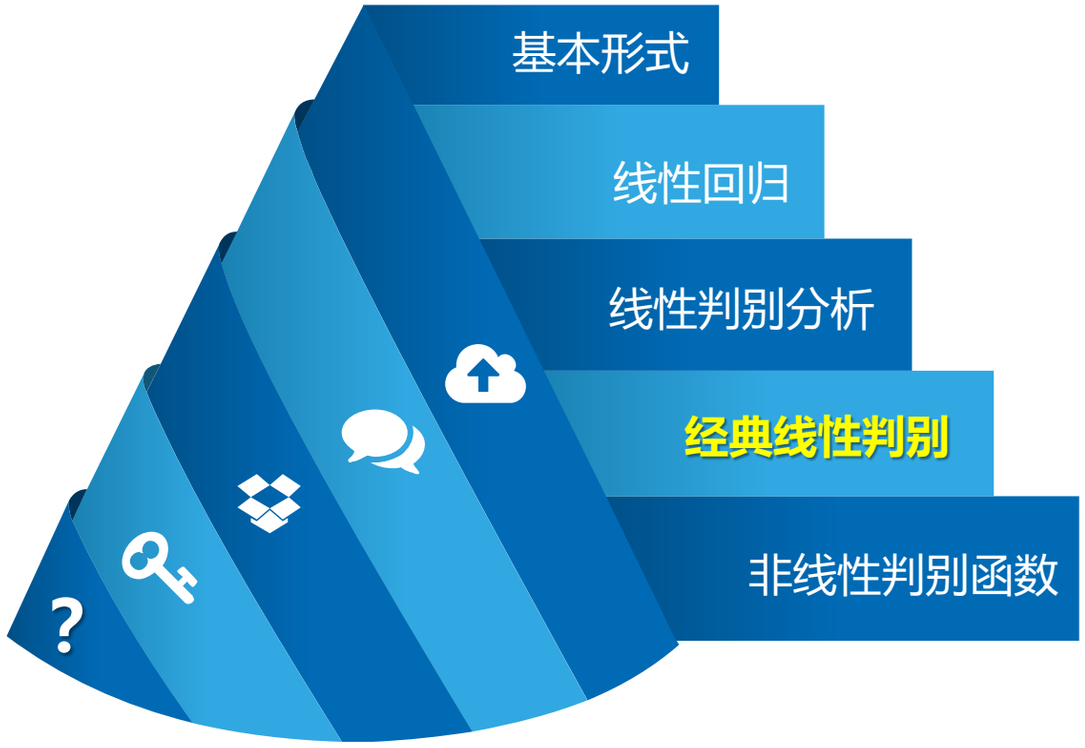
得　$r_0=\dfrac{d(\boldsymbol{X})}{\|\boldsymbol{W}_0\|}=\dfrac{w_{n+1}}{\|\boldsymbol{W}_0\|}$

—— 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。



- **线性模型基本形式**
  线性方程，基本形式，…
- **线性回归**
  参数估计，最小二乘，对数几率回归，…
- **线性判别分析**
  基本概念，广义线性判别，几何性质 …
- **经典线性判别准则**
  Fisher判别，感知器准则、梯度法 …
- **非线性判别函数**
  分段线性，…

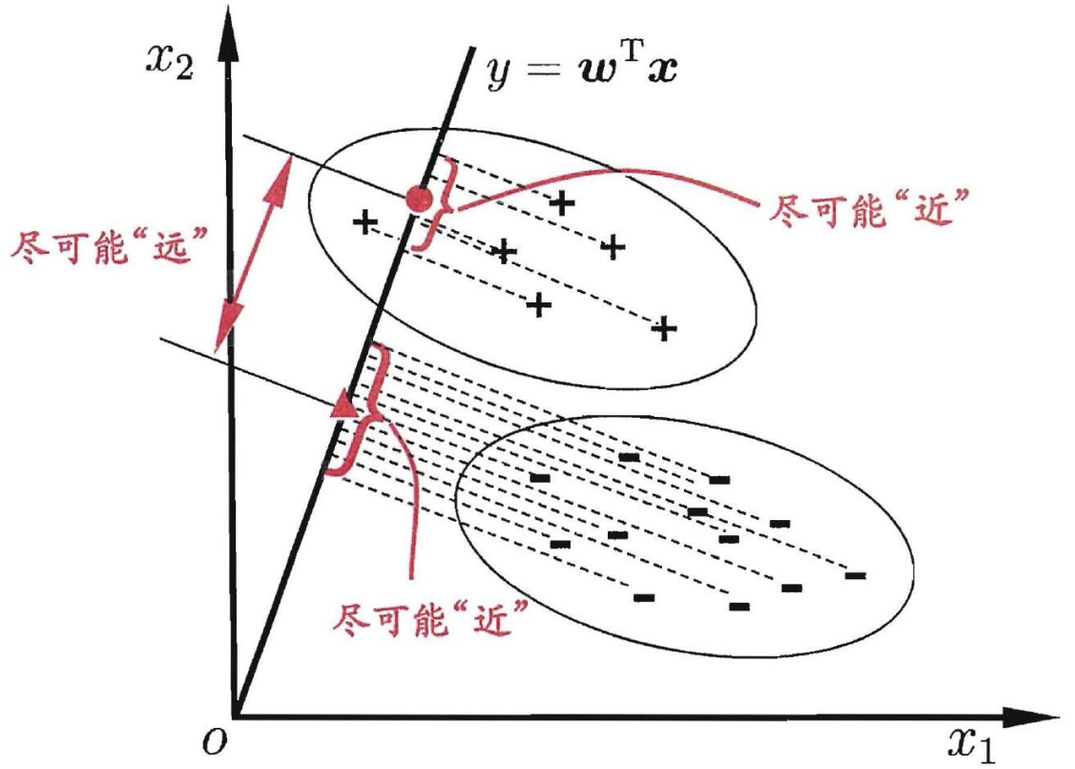
Fisher线性判别（Fisher Linear Discrimination，FLD）

Fisher准则是英国统计学家R. A. Fisher在1936年首先提出的一种线性判别方法。

Fisher准则的基本原理：投影（降维）

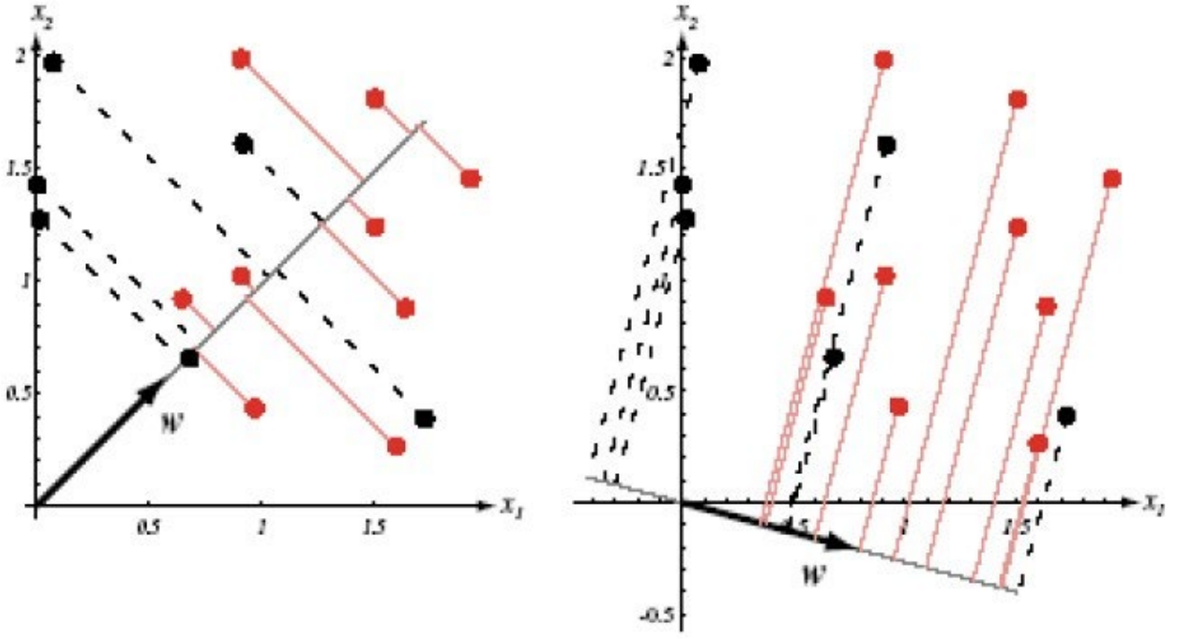
找到一个最合适的投影轴，将多维数据投影到该投影轴上，使两类样本在该投影轴上投影之间的距离尽可能远，而每一类样本的投影尽可能紧凑，从而使分类效果为最佳。

用投影后数据的统计性质 (均值和离散度的函数) 作为判别优劣的标准。



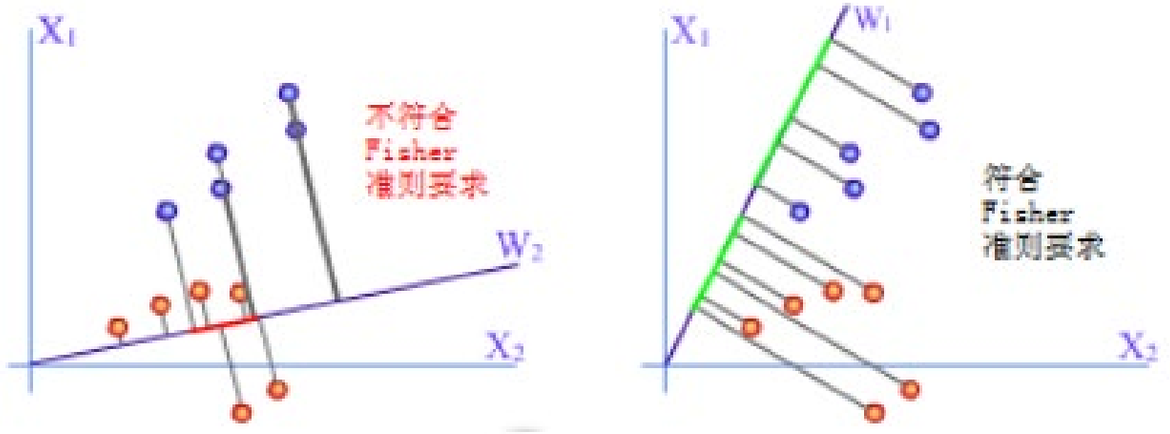
Fisher线性判别

把同一组样本点向两个不同的方向作投影，显然右图更易分开。



问题：

如何根据实际情况找到一条最好的、最易于分类的投影线？



Fisher线性判别（Fisher Linear Discrimination，FLD）

d 维到一维的数学变换

样本集：N个d 维样本 $\mathcal{X}=\{x_1,...,x_N\}$

ω_1 类包含样本 $\mathcal{X}_1=\{x_1^1,...,x_{N_1}^1\}$ ，ω_2 类包含样本 $\mathcal{X}_2=\{x_1^2,...,x_{N_2}^2\}$

在 Y 空间的一维投影 $y_i=w^T x_i$ ， $i=1,...,N$

在$\boldsymbol{X}$空间，类均值向量 $m_i=\frac{1}{N_i}\sum_{x_j\in\mathcal{X}_i}x_j$， $i=1,2$

类内离散度矩阵 within-class scatter $S_i=\sum_{x_j\in\mathcal{X}_i}(x_j-m_i)(x_j-m_i)^T$， $i=1,2$

总类内离散度矩阵 $S_w=S_1+S_2$

样本类间离散度矩阵 $\boldsymbol{S_b=(m_1-m_2)(m_1-m_2)^T}$

离散度矩阵在形式上与协方差矩阵很相似，但协方差矩阵是一种总体期望值，而离散矩阵只是表示有限个样本在空间分布的离散程度。

Fisher线性判别（Fisher Linear Discrimination，FLD）

d 维到一维的数学变换

一维投影空间Y：

各类样本均值 $\tilde{m}_i = \frac{1}{N_i}\sum_{y\in\psi_i} y, \quad i=1,2$

样本类内离散度 $\tilde{S}_i = \sum_{y\in\psi_i}(y-\tilde{m}_i)^2, \quad i=1,2$

总类内离散度 $\tilde{S}_w = \tilde{S}_1 + \tilde{S}_2$

样本类间离散度 $\tilde{S}_b = (\tilde{m}_1 - \tilde{m}_2)^2$

以上定义描述 *d* 维空间样本点到一向量投影后的**分散情况**。样本离散度的定义与随机变量方差相类似。

Fisher线性判别（Fisher Linear Discrimination，FLD）

d 维到一维投影后，样本x与其投影y的统计量之间的关系：

各类样本均值 $\tilde{m}_i = \frac{1}{N_i}\sum_{\mathbf{y}\in\psi_i} y = \frac{1}{N_i}\sum_{\mathbf{y}\in K_i} \mathbf{w}^T\mathbf{x} = \mathbf{w}^T\mathbf{m}_i, \quad i = 1,2$

样本类内离散度 $\tilde{S}_i = \sum_{y\in\psi_i}(y-\tilde{m}_i)^2 = \sum_{x\in K_i}(\mathbf{w}^T\mathbf{x}-\mathbf{w}^T\mathbf{m}_i)^2 = \mathbf{w}^T\left[\sum_{x\in K_i}(\mathbf{x}-\mathbf{m}_i)(\mathbf{x}-\mathbf{m}_i)^T\right]\mathbf{w} = \mathbf{w}^T S_i\mathbf{w}$

总类内离散度 $\tilde{S}_w = \tilde{S}_1 + \tilde{S}_2 = \mathbf{w}^T(S_1+S_2)\mathbf{w} = \mathbf{w}^T S_w\mathbf{w}$

样本类间离散度 $\tilde{S}_b = (\tilde{m}_1-\tilde{m}_2)^2 = (\mathbf{w}^T\mathbf{m}_1-\mathbf{w}^T\mathbf{m}_2)^2 = \mathbf{w}^T(\mathbf{m}_1-\mathbf{m}_2)(\mathbf{m}_1-\mathbf{m}_2)^T\mathbf{w} = \mathbf{w}^T S_b\mathbf{w}$

Fisher线性判别（Fisher Linear Discrimination，FLD）

Fisher判别准则：*d* 维到一维投影后，在一维Y空间里各个样本尽可能做到：

1、分得开

两类均值差 $(\tilde{m}_1-\tilde{m}_2)^2$ 越大越好

2、各类样本内部尽量密集

类内离散度 $\tilde{S}_1^2$ 和 $\tilde{S}_2^2$ 越小越好

Fisher准则函数(Fisher's Criterion)： $J_F(w)=\dfrac{(\tilde{m}_1-\tilde{m}_2)^2}{\tilde{S}_1+\tilde{S}_2}=\dfrac{\tilde{S}_b}{\tilde{S}_w}=\dfrac{\mathbf{w}^T S_b \mathbf{w}}{\mathbf{w}^T S_w \mathbf{w}}$

Fisher最佳投影方向的求解 $\mathbf{w}^*=\underset{\mathbf{w}}{\operatorname{argmax}} J_F(\mathbf{w})$

采用拉格朗日乘子算法求解，得： $\mathbf{w}^*=S_w^{-1}(\mathbf{m}_1-\mathbf{m}_2)$

Fisher线性判别（Fisher Linear Discrimination，FLD）

实现步骤：　1）计算各类样品均值向量$\boldsymbol{m}_i$，$\boldsymbol{m}_i$是各个类的均值，N_i是类ω_i的样品个数

$$\boldsymbol{m}_i = \frac{1}{N_i}\sum_{x_j \in \boldsymbol{X}_i} x_j, \quad i = 1,2$$

2）计算样品类内离散度矩阵 $\boldsymbol{S}_i$ 和总类内离散度矩阵 $\boldsymbol{S}_w$

$$\boldsymbol{S}_i = \sum_{x_j \in \boldsymbol{X}_i} (x_j - \boldsymbol{m}_i)(x_j - \boldsymbol{m}_i)^T, \quad i = 1,2$$

$$\boldsymbol{S}_w = \boldsymbol{S}_1 + \boldsymbol{S}_2$$

3）计算样品类间离散度矩阵 $\boldsymbol{S}_b$

$$\boldsymbol{S}_b = (\boldsymbol{m}_1 - \boldsymbol{m}_2)(\boldsymbol{m}_1 - \boldsymbol{m}_2)^T$$

4）求最佳投影向量w*

$$\mathbf{w}^* = \boldsymbol{S}_w^{-1}(\mathbf{m}_1 - \mathbf{m}_2)$$

Fisher线性判别（Fisher Linear Discrimination，FLD）

实现步骤：　5）对训练集内所有样品进行投影　$y=\left(\mathbf{w}^*\right)^T \boldsymbol{x}$

6）计算在一维投影空间上的分割阈值 y_0

在一维投影空间Y上的各样本均值 $\tilde{m}_i=\frac{1}{N_i}\sum_{\mathbf{y}\in\psi_i} y \quad i=1,2$

类内离散度$\tilde{S}_i^2$和总类内离散度 $\tilde{S}_w$ 为：$\tilde{S}_i=\sum_{y\in\psi_i}\left(y-\tilde{m}_i\right)^2$，且 $\tilde{S}_w=\tilde{S}_1+\tilde{S}_2$

分割阈值 y_0 的选取可以有不同方案：

$$y_0=\frac{N_1\tilde{\boldsymbol{m}}_1+N_2\tilde{\boldsymbol{m}}_2}{N_1+N_2} \quad 或 \quad y_0=\frac{\tilde{\boldsymbol{m}}_1+\tilde{\boldsymbol{m}}_2}{2}+\frac{\ln\left(p\left(\omega_1\right)/p\left(\omega_2\right)\right)}{N_1+N_2-2}$$

7）对于给定的$\boldsymbol{X}$，计算它在投影向量$\mathbf{w}^*$上的投影点Y

根据决策规则分类
$$\begin{cases} Y>y_0 \Rightarrow \boldsymbol{X}\in\omega_1 \\ Y<y_0 \Rightarrow \boldsymbol{X}\in\omega_2 \end{cases}$$

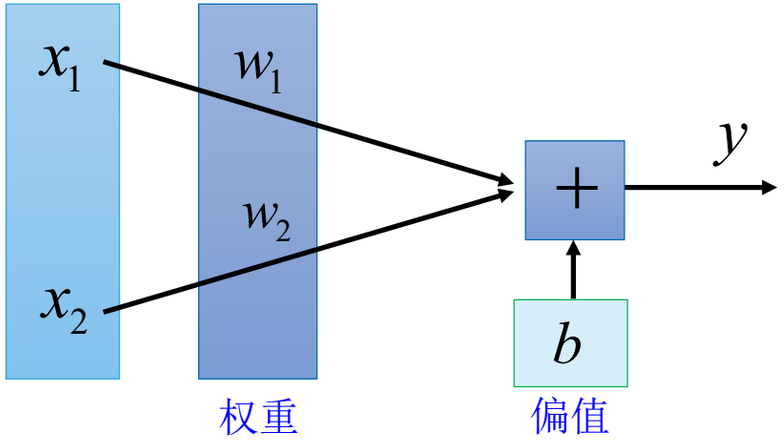
感知器准则（Perceptron）

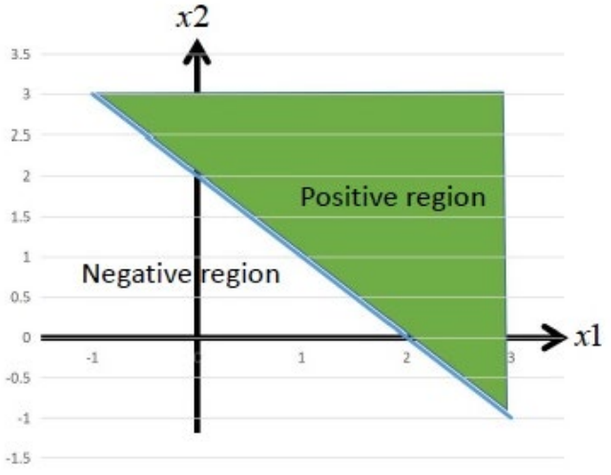
感知器准则是五十年代由Rosenblatt提出的一种自学习判别函数生成方法；

Rosenblatt将其用于脑模型感知器(Perceptron)，因此被称为感知准则函数；

特点：随意确定的判别函数初始值，在对样本分类训练过程中逐步修正直至最终确定。

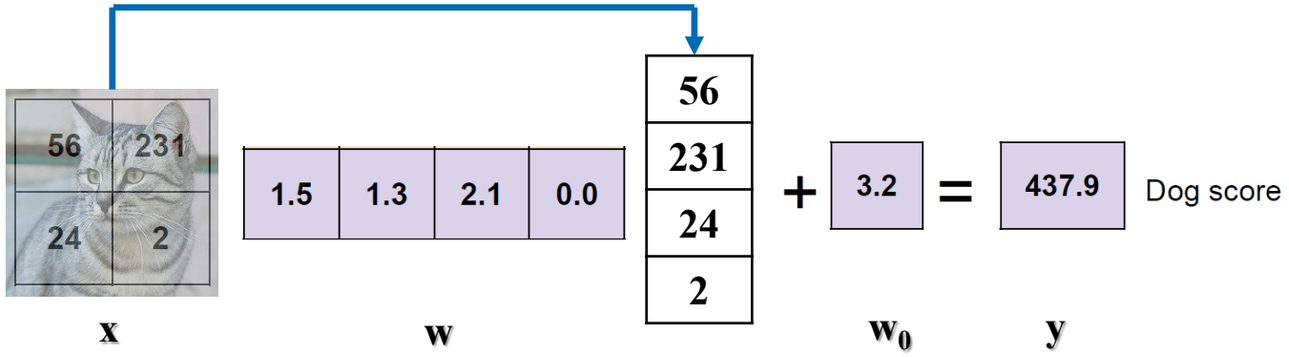
$$y = x_1 w_1 + x_2 w_2 + b$$





$$w_1 = 1, w_2 = 1, b = -2$$

感知器准则（Perceptron）



基本概念

1、线性判别函数的齐次简化　　$g(x) = w^T x + w_0 = \boldsymbol{\alpha}^T y$

$$y = \begin{bmatrix} 1 \\ x \end{bmatrix}$$

增广样本向量

$$\boldsymbol{\alpha} = \begin{bmatrix} w_0 \\ w \end{bmatrix}$$

增广权向量

只要求出权向量，分类器即设计完成。

问题为：如何通过各种算法，利用已知类别的模式样本训练出权向量*w*。

感知器准则（Perceptron）

基本概念

2、线性可分性：训练样本集中的两类样本在特征空间可以用一个线性分界面正确无误地分开。在线性可分条件下，对合适的（广义）权向量 a 应有：

$$\text{如果 } \mathbf{y}_i \in \omega_1, \text{则} \boldsymbol{a}^T \mathbf{y}_i > 0$$
$$\text{如果 } \mathbf{y}_i \in \omega_2, \text{则} \boldsymbol{a}^T \mathbf{y}_i < 0$$

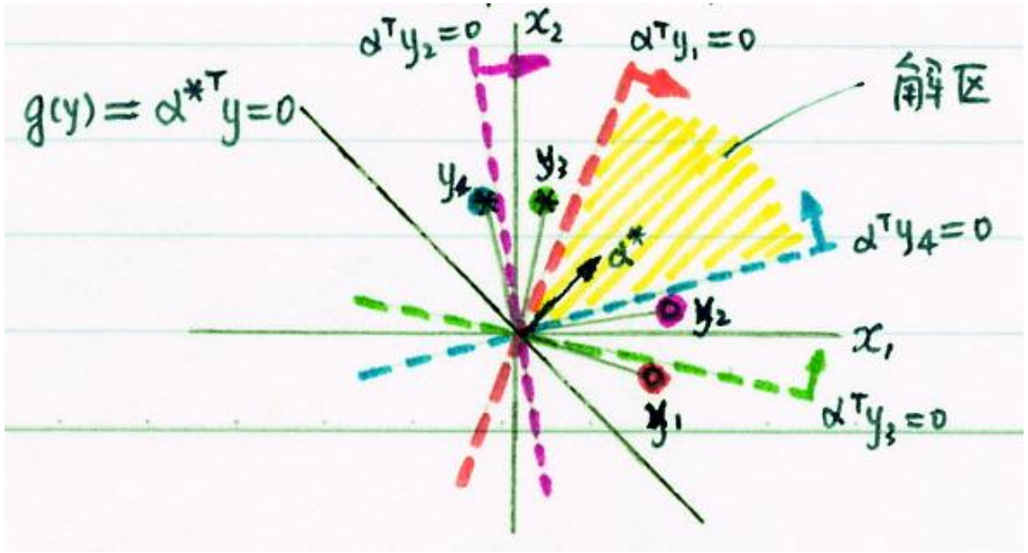
3、样本的规范化

$$\mathbf{y}'_n = \begin{cases} \mathbf{y}_i & \text{如果 } \mathbf{y}_i \in \omega_1 \\ -\mathbf{y}_j & \text{如果 } \mathbf{y}_j \in \omega_2 \end{cases} \Rightarrow \boldsymbol{a}^T \mathbf{y}'_n > 0 \quad i = 1, \ldots, N$$

无需考虑样本原来的类别标志，只要找一个对全部样本 $\mathbf{y}'_n$ 都满足 $\boldsymbol{a}^T \mathbf{y}'_n > 0$ 的权向量 $\boldsymbol{a}$ 就可以。上述过程称为样本的规范化。$\mathbf{y}'_n$ 称为规范化增广样本向量。

感知器准则（Perceptron）

基本概念



4、解向量和解区：

解向量：在线性可分情况下，满足 $a^T \mathbf{y}_n > 0$ 的权向量，记作 a^* 。

解区：权值空间中所有解向量组成的区域。

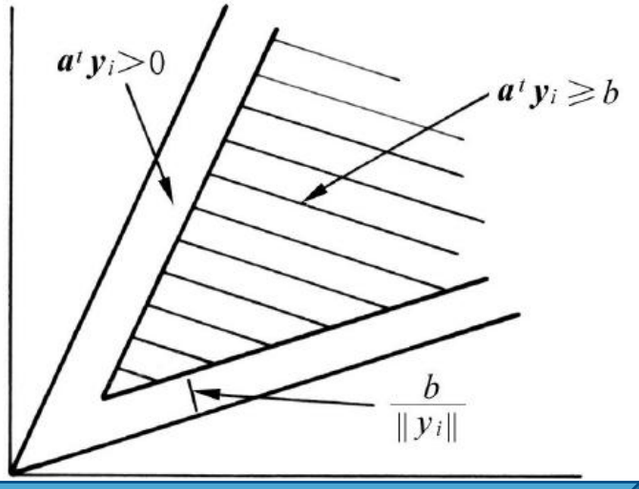
对每个样本 $\mathbf{y}_i$，$\alpha^T y_i = 0$ 为权空间中的一个超平面 $\hat{H}_i$，解区只可能在 $\hat{H}_i$ 的正侧。所有样本对应的超平面 $\hat{H}_i$ 的正侧的交集就是解区。

余量：解区中所有权向量都是解，但越靠近解区中间的解向量，所得的分类面离各类样本越远（最近距离越远），对将来的样本错分的可能性越小。

引入余量b>0，要求解向量满足 $\alpha^T y_i \geq b$ ，i=1,…, N

若要求 b 最大则为“最优分类面” 目的：

1. 解更可靠，（推广性更强）
2. 防止算法收敛到解区边界



感知器准则（Perceptron）

基本概念

5、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相同的过程。

学习：从分类器的角度讲 { 非监督学习；监督学习 ⟷ 训练

6、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。

7、感知器

感知器作为人工神经网络的基石，由于无法实现非线性分类而被冷落。但其“赏罚概念（reward-punishment concept）”得到广泛应用。

感知器准则（Perceptron）

感知器算法（perception approach）

问题提出：

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

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

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。

感知器准则（Perceptron）

感知器算法（perception approach）

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：$\boldsymbol{W}(k+1)=\boldsymbol{W}(k)+\eta\boldsymbol{X}_i$

② 若$\boldsymbol{W}^{\mathrm{T}}(k)\boldsymbol{X}_i > 0$，分类正确，权向量不变：$\boldsymbol{W}(k+1)=\boldsymbol{W}(k)$　　η：正的校正增量，学习率。

统一写为：

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

（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。

感知器准则（Perceptron）

感知器算法（perception approach）

感知器算法是一种赏罚过程

分类正确时，对权向量“赏”——这里用“不罚”，即权向量不变；

分类错误时，对权向量“罚”——对其修改，向正确的方向转换。

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$\boldsymbol{W}$，则称算法是收敛的。

可以证明感知器算法是收敛的。收敛条件：模式类别线性可分。

理论证明，只要模式类是线性可分的，无论 a 的初值是什么，经过有限次叠代，都可收敛。

感知器准则（Perceptron）

例 已知两类训练样本 ω_1：$X_1=[0,0]^T$，$X_2=[0,1]^T$；ω_2：$X_3=[1,0]^T$，$X_4=[1,1]^T$

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)。

$$X_1=[0,0,1]^T \quad X_2=[0,1,1]^T \quad X_3=[-1,0,-1]^T \quad X_4=[-1,-1,-1]^T$$

任取$W(1)$=**0**，取 $\eta=1$，迭代过程为：

第一轮：

$$W^T(1)X_1=[0,0,0]\cdot\begin{bmatrix}0\\0\\1\end{bmatrix}=0,\ \le 0,\text{故}W(2)=W(1)+X_1=[0,0,1]^T$$

$$W^T(2)X_2=[0,0,1]\cdot\begin{bmatrix}0\\1\\1\end{bmatrix}=1,\ >0,\text{故}W(3)=W(2)=[0,0,1]^T$$

$$W^T(3)X_3=[0,0,1]\cdot\begin{bmatrix}-1\\0\\-1\end{bmatrix}=-1,\ \le 0,\text{故}W(4)=W(3)+X_3=[-1,0,0]^T$$

$$W^T(4)X_4=[-1,0,0]\cdot\begin{bmatrix}-1\\-1\\-1\end{bmatrix}=1,\ >0,\text{故}W(5)=W(4)=[-1,0,0]^T$$

有两个$W^T(k)X_i \le 0$的情况（错判），进行第二轮迭代。

感知器准则（Perceptron）

$\boldsymbol{X}_1=[0,0,1]^{\mathrm{T}}$　$\boldsymbol{X}_2=[0,1,1]^{\mathrm{T}}$　$\boldsymbol{X}_3=[-1,0,-1]^{\mathrm{T}}$　$\boldsymbol{X}_4=[-1,-1,-1]^{\mathrm{T}}$　$\boldsymbol{W}(5)=[-1,0,0]^{\mathrm{T}}$

第二轮：

$\boldsymbol{W}^{\mathrm{T}}(5)\boldsymbol{X}_1=0\le 0$，故$\boldsymbol{W}(6)=\boldsymbol{W}(5)+\boldsymbol{X}_1=[-1,0,1]^{\mathrm{T}}$　$\boldsymbol{W}^{\mathrm{T}}(6)\boldsymbol{X}_2=1>0$，故$\boldsymbol{W}(7)=\boldsymbol{W}(6)=[-1,0,1]^{\mathrm{T}}$

$\boldsymbol{W}^{\mathrm{T}}(7)\boldsymbol{X}_3=0\le 0$，故$\boldsymbol{W}(8)=\boldsymbol{W}(7)+\boldsymbol{X}_3=[-2,0,0]^{\mathrm{T}}$　$\boldsymbol{W}^{\mathrm{T}}(8)\boldsymbol{X}_4=2>0$，故$\boldsymbol{W}(9)=\boldsymbol{W}(8)=[-2,0,0]^{\mathrm{T}}$

第三轮：

$\boldsymbol{W}^{\mathrm{T}}(9)\boldsymbol{X}_1=0\le 0$，故$\boldsymbol{W}(10)=\boldsymbol{W}(9)+\boldsymbol{X}_1=[-2,0,1]^{\mathrm{T}}$　$\boldsymbol{W}^{\mathrm{T}}(10)\boldsymbol{X}_2=1>0$，故$\boldsymbol{W}(11)=\boldsymbol{W}(10)$

$\boldsymbol{W}^{\mathrm{T}}(11)\boldsymbol{X}_3=1>0$，故$\boldsymbol{W}(12)=\boldsymbol{W}(11)$　$\boldsymbol{W}^{\mathrm{T}}(12)\boldsymbol{X}_4=1>0$，故$\boldsymbol{W}(13)=\boldsymbol{W}(12)$

第四轮：

$\boldsymbol{W}^{\mathrm{T}}(13)\boldsymbol{X}_1=1>0$，故$\boldsymbol{W}(14)=\boldsymbol{W}(13)$　$\boldsymbol{W}^{\mathrm{T}}(14)\boldsymbol{X}_2=1>0$，故$\boldsymbol{W}(15)=\boldsymbol{W}(14)$

$\boldsymbol{W}^{\mathrm{T}}(15)\boldsymbol{X}_3=1>0$，故$\boldsymbol{W}(16)=\boldsymbol{W}(15)$　$\boldsymbol{W}^{\mathrm{T}}(16)\boldsymbol{X}_4=1>0$，故$\boldsymbol{W}(17)=\boldsymbol{W}(16)$

该轮迭代的分类结果全部正确，故解向量相应的判别函数为：

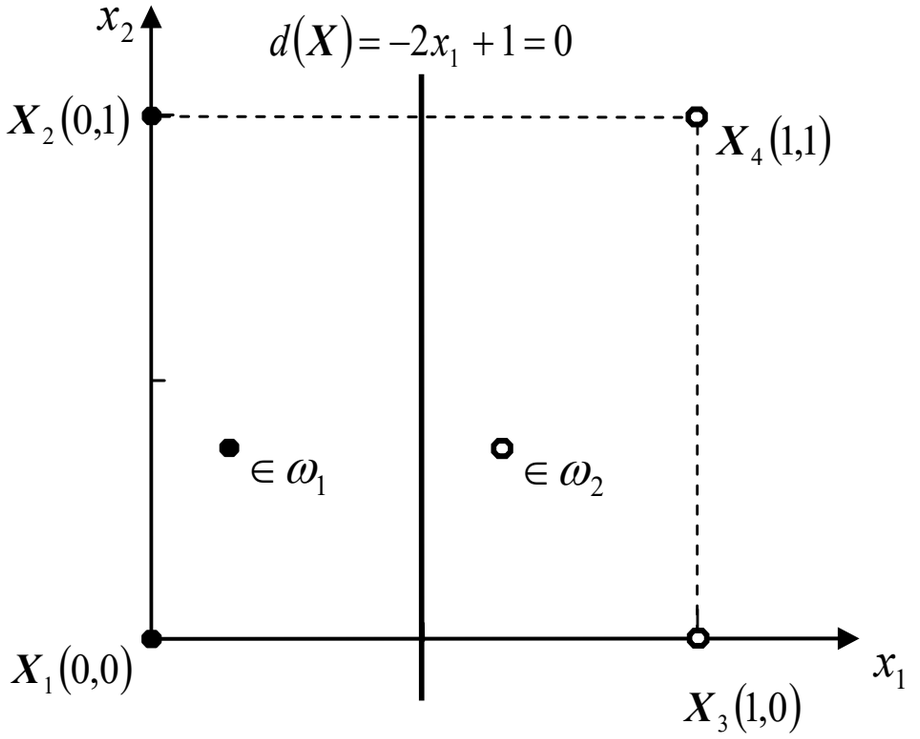
$$d(\boldsymbol{X})=-2x_1+1$$

感知器准则（Perceptron）

例 已知两类训练样本 ω_1：$X_1=[0,0]^T$，$X_2=[0,1]^T$；ω_2：$X_3=[1,0]^T$，$X_4=[1,1]^T$

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。

判别界面$d(X)$=0如图示。

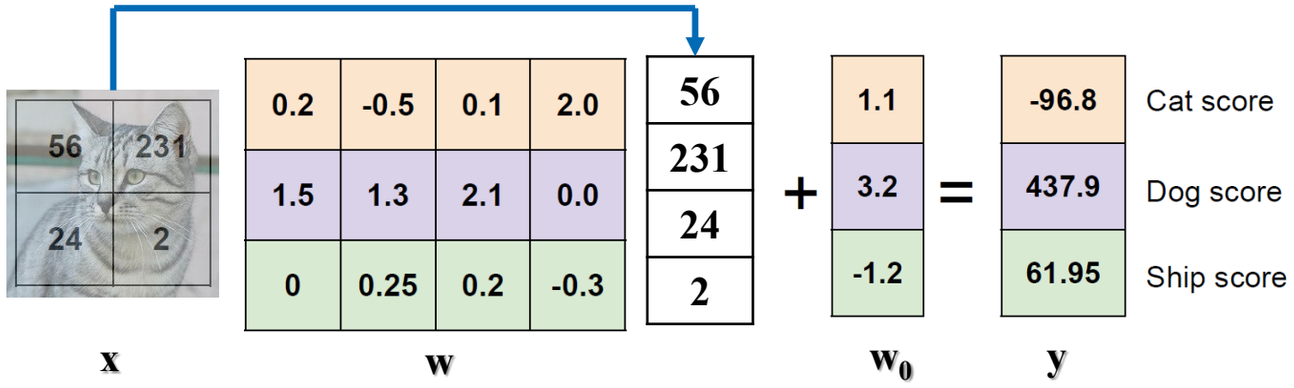


当 η 、$W(1)$取其他值时，结果可能不一样，所以感知器算法的解不是单值的。

感知器准则（Perceptron）

感知器算法用于多类情况（采用ω_i/ω_j两分法特例的多类分类方式）

对于 M 类模式应存在 M 个判决函数：$\{d_i,\quad i=1,...,M\}$。若 $\boldsymbol{X}\in\omega_i$，则 $d_i(\boldsymbol{X})>d_j(\boldsymbol{X}),\quad \forall j\neq i,\ j=1,...,M$



算法主要内容：

设有 M 种模式类别：$\omega_1,\omega_2,...,\omega_M$，其权向量初值为：$\boldsymbol{W}_j(1),\quad j=1,...,M$

训练样本为增广向量形式，但**不需要规范化处理**。

感知器准则（Perceptron）

感知器算法用于多类情况

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

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

分二种情况修改权向量：

① 若 $d_i(k)>d_j(k),\quad \forall j\neq i;\ j=1,2,...,M$ 则权向量不变；$\boldsymbol{W}_j(k+1)=\boldsymbol{W}_j(k),\quad j=1,2,...,M$

② 若第 l 个权向量使 $d_i(k)\le d_l(k)$ ，则相应的权向量作调整，即：

$$\begin{cases}\boldsymbol{W}_i(k+1)=\boldsymbol{W}_i(k)+\eta\boldsymbol{X}\\ \boldsymbol{W}_l(k+1)=\boldsymbol{W}_l(k)-\eta\boldsymbol{X}\\ \boldsymbol{W}_j(k+1)=\boldsymbol{W}_j(k),\ j\neq i,l\end{cases}$$

，η为正的校正增量，学习率

可以证明：对于采用ω_i/ω_j两分法特例的多类分类方式，经过有限次迭代后算法收敛。

感知器准则（Perceptron）

感知器算法用于多类情况

例 设有三个线性可分的模式类，三类的训练样本分别为 ω_1：$\boldsymbol{X}_1=[0,0]^{\mathrm{T}}$； ω_2：$\boldsymbol{X}_2=[1,1]^{\mathrm{T}}$； ω_3：$\boldsymbol{X}_3=[-1,1]^{\mathrm{T}}$

现采用ω_i/ω_j两分法特例的多类分类方式，试用感知器算法求出判别函数。

解：增广向量形式： $\boldsymbol{X}_1=[0,0,1]^{\mathrm{T}}$， $\boldsymbol{X}_2=[1,1,1]^{\mathrm{T}}$， $\boldsymbol{X}_3=[-1,1,1]^{\mathrm{T}}$ 注意，这里任一类的样本都不能乘以（－1）。

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

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， 因此三个权向量都需要修改：

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

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

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

感知器准则（Perceptron）

感知器算法用于多类情况

例 设有三个线性可分的模式类，三类的训练样本分别为 ω_1：$\boldsymbol{X}_1=[0,0]^{\mathrm{T}}$； ω_2：$\boldsymbol{X}_2=[1,1]^{\mathrm{T}}$； ω_3：$\boldsymbol{X}_3=[-1,1]^{\mathrm{T}}$

现采用ω_i/ω_j两分法特例的多类分类方式，试用感知器算法求出判别函数。

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，修改为权向量。

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。

感知器准则（Perceptron）

感知器算法用于多类情况

例 设有三个线性可分的模式类，三类的训练样本分别为 ω_1：$\boldsymbol{X}_1=[0,0]^{\mathrm{T}}$； ω_2：$\boldsymbol{X}_2=[1,1]^{\mathrm{T}}$； ω_3：$\boldsymbol{X}_3=[-1,1]^{\mathrm{T}}$

现采用ω_i/ω_j两分法特例的多类分类方式，试用感知器算法求出判别函数。

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：$d_1(\boldsymbol{X})=-2x_2$　　$d_2(\boldsymbol{X})=2x_1-2$　　$d_3(\boldsymbol{X})=-2x_1-2$

梯度法

梯度下降——数学基础知识

导数 Derivation

设函数$y=f(x)$在$U(x_0,\ \delta)$ 内有定义，

如果极限 $\lim\limits_{\Delta x\to 0}\dfrac{\Delta y}{\Delta x}=\lim\limits_{\Delta x\to 0}\dfrac{f(x_0+\Delta x)-f(x_0)}{\Delta x}$ 存在，

则称这个极限值为 $y=f(x)$ 在点x_0 处的导数。

记为 $y'|_{x=x_0}$

导数的意义：

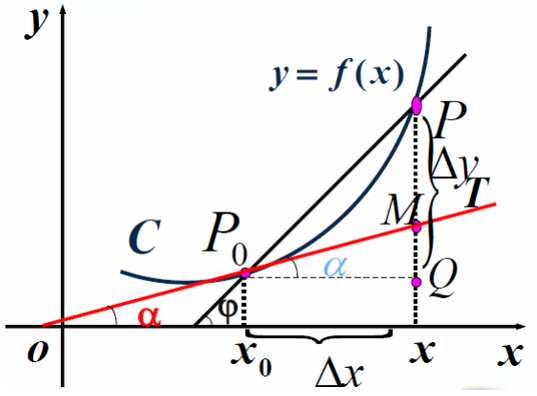
1、变化率、变化速度

反映变量 y 在点 x_0 处的变化率。

或是 y 随自变量 x 变化而变化的快慢程度。

如：速度$v=\Delta d/\Delta t$，加速度 $a=\Delta v/\Delta t$

2、导数的几何意义——曲线上某一点处的切线斜率



切线斜率 $\tan\alpha=\lim\limits_{\varphi\to\alpha}\tan\varphi=\lim\limits_{\Delta x\to 0}\dfrac{\Delta y}{\Delta x}=f'(x_0)$

梯度法

梯度下降——数学基础知识

偏导数 Partial Derivation

二元偏导数定义：设二元函数 $z=f(x,y)$ 在 $P_0(x_0,y_0)$ 的某邻域内有定义，

如果极限 $\lim\limits_{\Delta x\to 0}\dfrac{\Delta y}{\Delta x}=\lim\limits_{\Delta x\to 0}\dfrac{f(x_0+\Delta x,y_0)-f(x_0,y_0)}{\Delta x}$ 存在，

则称此极限值为函数 f 在 P_0 处对于 x 的偏导数，记为 $\left.\dfrac{\partial f}{\partial x}\right|_{P_0}$.

同理有对于 y 的偏导数

$$\left.\frac{\partial f}{\partial y}\right|_{P_0}=\lim_{\Delta x\to 0}\frac{f(x_0,y_0+\Delta y)-f(x_0,y_0)}{\Delta y}$$

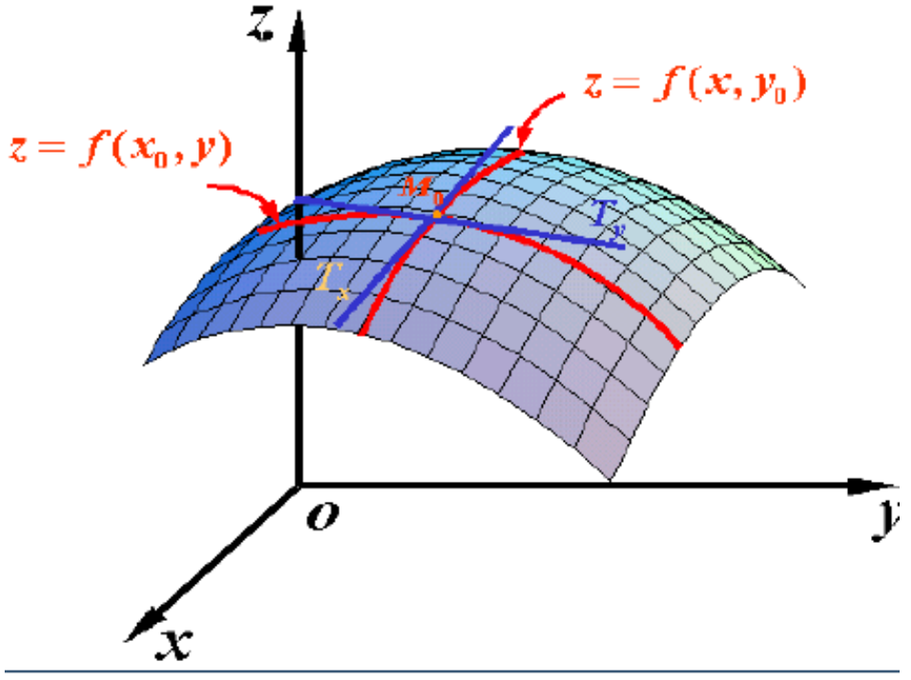
梯度法

梯度下降——数学基础知识

偏导数的几何意义

偏导数 $f'_x(x_0,y_0)$ 就是曲线 $z=f(x,y_0)$ 在 x_0 处的 x 方向切线斜率

偏导数 $f'_y(x_0,y_0)$ 就是曲线 $z=f(x_0,y)$ 在 y_0 处的 y 方向切线斜率



梯度法

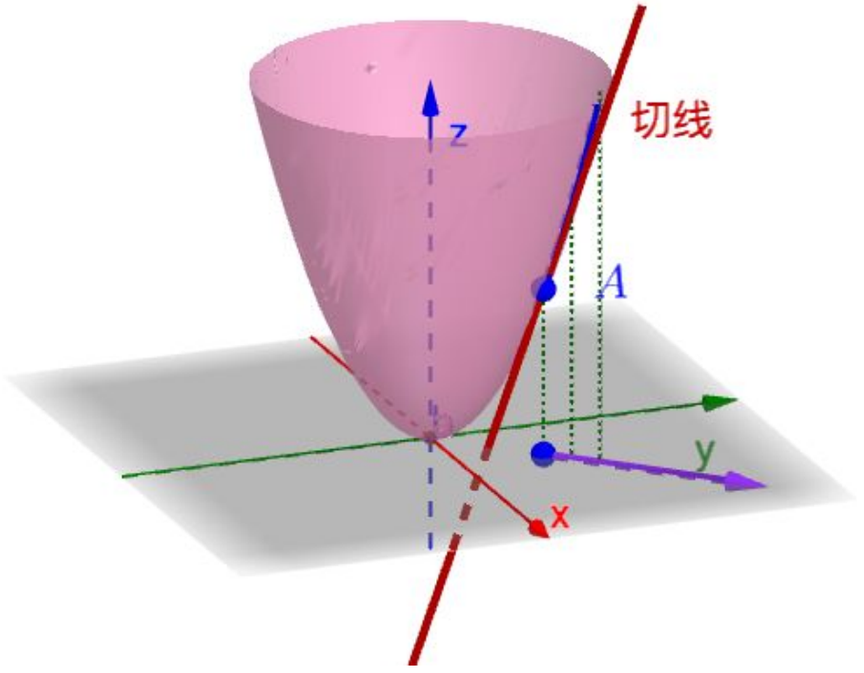
梯度下降——数学基础知识

方向导数 Directional Derivative

设 f 是定义于 R^n 中某区域 D 上的函数，点 $P_0 \in D$ ，l 为一给定的非零向量，P 为一动点，向量 P_0P 与 l 的方向始终一致，如果极限

$$\lim_{\|P_0P\|\to 0} \frac{f(P)-f(P_0)}{\|P_0P\|}$$

存在，则称此极限为函数 f 在 P_0 处沿 l 方向的方向导数，记为 $\left.\frac{\partial f}{\partial l}\right|_{P_0}$.



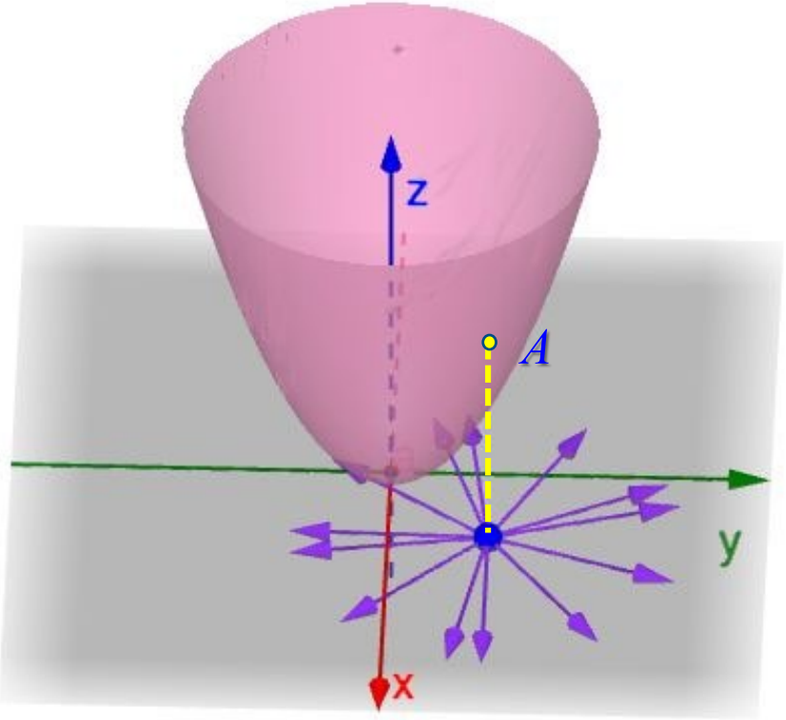
梯度法

梯度下降——数学基础知识

梯度 Gradient

显然A点不止一个方向导数，而是360度都有方向导数。

梯度的几何意义：是一个矢量，其方向上的方向导数最大，其大小正好是此最大方向导数。



二元函数梯度定义：设二元函数 $z=f(x,y)$ 在 $P_0(x_0,y_0)$ 点

存在偏导数，该点处梯度为：$\nabla G=\left(\partial f/\partial x\big|_{P_0},\partial f/\partial y\big|_{P_0}\right)$

梯度方向为：$\theta=\tan^{-1}\left(\frac{\partial f}{\partial y}/\frac{\partial f}{\partial x}\right)$

梯度幅值为：$\|\nabla f\|=\sqrt{(\frac{\partial f}{\partial x})^2+(\frac{\partial f}{\partial y})^2}$

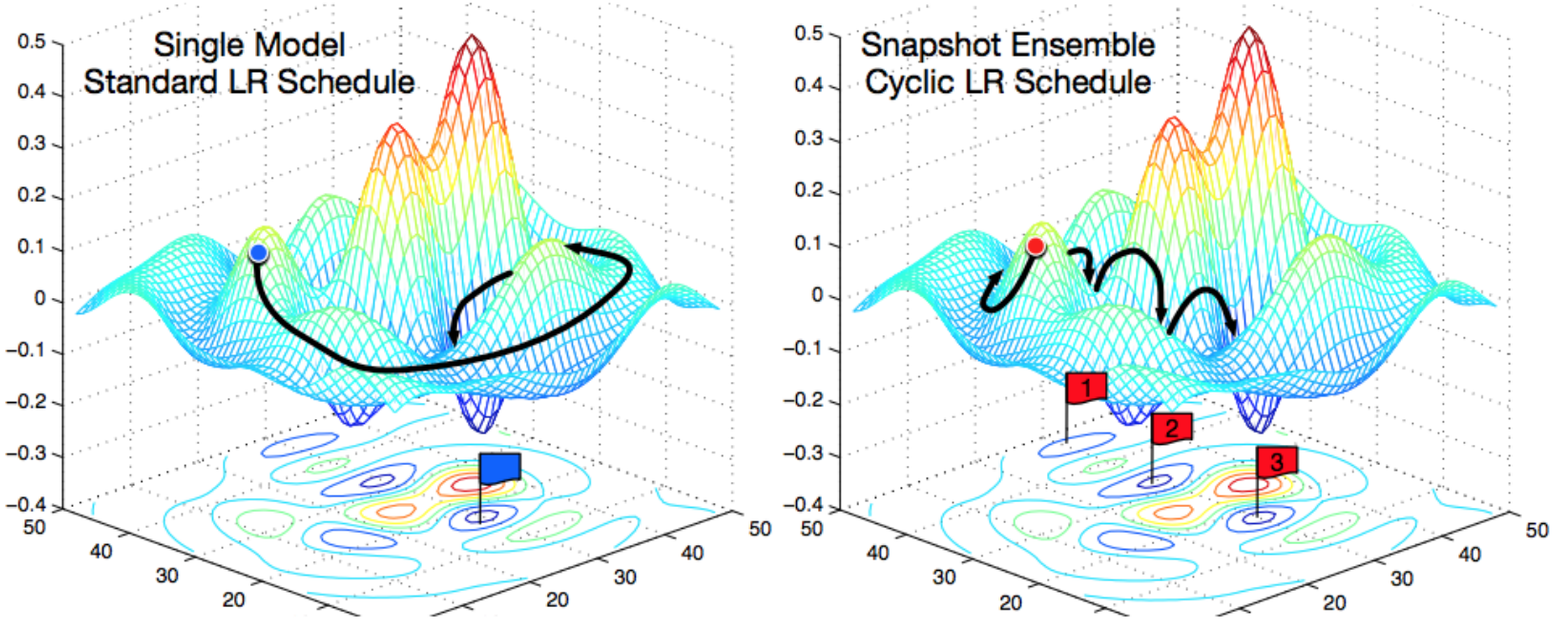
梯度法

梯度下降——数学基础知识

梯度 Gradient

设 f 是 R_n 中区域 D 上的数量场，如果 f 在 $P_0 \in D$ 处可微，称向量 $\left(\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, \ldots, \frac{\partial f}{\partial x_n}\right)\Big|_{P_0}$ 为 f 在 P_0 处的梯度。

梯度下降



梯度（下降）法

1、梯度概念小结

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

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

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)。

梯度（下降）法

梯度（下降）法基本原理

2、梯度算法 设两个线性可分的模式类ω_1和ω_2，共 N 个样本，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： $d(\boldsymbol{X}_i)=\boldsymbol{W}^{\mathrm{T}}\boldsymbol{X}_i>0 \quad i=1,2,...,N$ —— N个不等式

梯度算法的目的是求一个满足上述条件的权向量，即联立不等式求解$\boldsymbol{W}$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（损失函数）达到极小值的目的。

准则函数（损失函数）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$\boldsymbol{W}(k+1)$： $\boldsymbol{W}(k+1)=\boldsymbol{W}(k)+\eta(-\nabla J)=\boldsymbol{W}(k)-\eta\nabla J$

具体为：

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

梯度法

梯度法步骤：

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

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）在下一轮迭代中，如果有样本使 $\nabla J\neq 0$ ，则回到(2)再次迭代。直至 $\boldsymbol{W}$ 不再变化，算法收敛。

随着权向量$\boldsymbol{W}$向理想值接近，准则函数关于$\boldsymbol{W}$的导数 $\nabla J(k)$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。

梯度法

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。

感知器算法是梯度法的特例。

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

步骤1：求 J 的梯度 $\nabla J=\frac{\partial J(\boldsymbol{W},\boldsymbol{X})}{\partial \boldsymbol{W}}=?$

步骤2：求$\boldsymbol{W}(k+1)$

梯度法

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

固定增量法

步骤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 $\dfrac{\partial f}{\partial \boldsymbol{X}}=\left[\dfrac{\partial f}{\partial x_1},\ldots,\dfrac{\partial f}{\partial x_n}\right]^{\mathrm{T}}$

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

$$\frac{\partial\left(\boldsymbol{W}^{\mathrm{T}}\boldsymbol{X}\right)}{\partial \boldsymbol{W}}=\frac{\partial}{\partial \boldsymbol{W}}\left(\sum_{i=1}^{n}w_i x_i+w_{n+1}\right)=\left[\frac{\partial}{\partial w_1}\left(\sum_{i=1}^{n}w_i x_i+w_{n+1}\right),\ldots,\frac{\partial}{\partial w_k}\left(\sum_{i=1}^{n}w_i x_i+w_{n+1}\right),\ldots,\frac{\partial}{\partial w_{n+1}}\left(\sum_{i=1}^{n}w_i x_i+w_{n+1}\right)\right]^{\mathrm{T}}=\left[x_1,\ldots,x_k,\ldots,x_n,1\right]^{\mathrm{T}}=\boldsymbol{X}$$

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

$$\begin{cases}\boldsymbol{W}^{\mathrm{T}}\boldsymbol{X}>0\text{时},\dfrac{\partial\left(\left|\boldsymbol{W}^{\mathrm{T}}\boldsymbol{X}\right|\right)}{\partial \boldsymbol{W}}=\dfrac{\partial\left(\boldsymbol{W}^{\mathrm{T}}\boldsymbol{X}\right)}{\partial \boldsymbol{W}}=\boldsymbol{X}\\ \boldsymbol{W}^{\mathrm{T}}\boldsymbol{X}\le 0\text{时},\dfrac{\partial\left(\left|\boldsymbol{W}^{\mathrm{T}}\boldsymbol{X}\right|\right)}{\partial \boldsymbol{W}}=\dfrac{\partial\left(-\boldsymbol{W}^{\mathrm{T}}\boldsymbol{X}\right)}{\partial \boldsymbol{W}}=-\boldsymbol{X}\end{cases}$$

③ $\nabla J=\dfrac{\partial J(\boldsymbol{W},\boldsymbol{X})}{\partial \boldsymbol{W}}$ 有：

$$\therefore\quad \nabla J=\frac{\partial J(\boldsymbol{W},\boldsymbol{X})}{\partial \boldsymbol{W}}=\begin{cases}0 & \text{当}\boldsymbol{W}^{\mathrm{T}}\boldsymbol{X}>0\text{时}\\ -\boldsymbol{X} & \text{当}\boldsymbol{W}^{\mathrm{T}}\boldsymbol{X}\le 0\text{时}\end{cases}$$

梯度法

固定增量法

步骤2: 求$\boldsymbol{W}(k+1)$

$$\boldsymbol{W}(k+1)=\boldsymbol{W}(k)-\eta\nabla J=\boldsymbol{W}(k)-\eta\left[\frac{\partial J(\boldsymbol{W},\boldsymbol{X})}{\partial \boldsymbol{W}}\right]_{\boldsymbol{W}=\boldsymbol{W}(k)}=\boldsymbol{W}(k)+\begin{cases}0, & \text{若}\boldsymbol{W}^{\mathrm{T}}(k)\boldsymbol{X}>0\\ \eta\boldsymbol{X}, & \text{若}\boldsymbol{W}^{\mathrm{T}}(k)\boldsymbol{X}\le 0\end{cases}$$

上式即为固定增量算法，与感知器算法形式相同 。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$\boldsymbol{W}$的问题，转换为求函数$\boldsymbol{J}$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。

最小平方误差算法（Least Mean Square Error, LMSE；亦称Ho-Kashyap算法）

感知器算法、梯度算法等类似方法，**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。

最小平方误差算法

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和ω_2，两个模式类的训练样本集（规范化增广样本向量）$\{\boldsymbol{X}_i,\ i=1,2,...,N\}$，应满足：$\boldsymbol{W}^{\mathrm{T}}\boldsymbol{X}_i>0$

上式展开写为：

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

写成矩阵形式为：

$$\begin{bmatrix} x_{11} & x_{12} & ... & x_{1n} & 1 \\ x_{21} & x_{22} & ... & x_{2n} & 1 \\ ... & & & & \\ -x_{N1} & -x_{N2} & ... & -x_{Nn} & -1 \end{bmatrix}_{N\times(n+1)} \begin{bmatrix} w_1 \\ w_2 \\ ... \\ w_n \\ w_{n+1} \end{bmatrix}_{(n+1)\times1} > \begin{bmatrix} 0 \\ 0 \\ ... \\ 0 \\ 0 \end{bmatrix}_{N\times1} = \boldsymbol{0}$$

改写为 $\Rightarrow \boldsymbol{XW}>\boldsymbol{0}$

通过解不等式组$\boldsymbol{XW}>\boldsymbol{0}$，求出$\boldsymbol{W}$

最小平方误差算法

LMSE算法原理

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：$\boldsymbol{B}=[b_1,b_2,...,b_i,...,b_N]^{\mathrm{T}}$ 为各分量均为正值的矢量。 ∴ 两式等价。

说明：

① 当方程组中当行数>>列数时，通常无精确解，称为矛盾方程组（或超定方程组 ），一般求近似解。

在模式识别中，通常训练样本数 N *通常*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

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

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）。

最小平方误差算法

LMSE算法原理

准则函数定义为：$J(\boldsymbol{W},\boldsymbol{X},\boldsymbol{B})=\frac{1}{2}\|\boldsymbol{XW}-\boldsymbol{B}\|^2$

“最小二乘”：

—— 最小：使方程组两边误差最小，也即使J最小。

—— 二乘：次数为2，乘了两次

} 最小平方（误差算法）

③ LMSE算法的思路：

对$\boldsymbol{XW}>\boldsymbol{0}$求解

↓ 转化为

对$\boldsymbol{XW}=\boldsymbol{B}$求解

↓ 转化为

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

最小平方误差算法

LMSE算法原理

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

$$\boldsymbol{XW}-\boldsymbol{B}=\begin{bmatrix} x_{11} & \dots & x_{1n} & 1 \\ x_{21} & \dots & x_{2n} & 1 \\ \dots & \dots & \dots & 1 \\ x_{i1} & \dots & x_{in} & 1 \\ \dots & \dots & \dots & 1 \\ -x_{N1} & \dots & -x_{Nn} & 1 \end{bmatrix}_{N\times(n+1)} \begin{bmatrix} w_1 \\ w_2 \\ \dots \\ w_n \\ w_{n+1} \end{bmatrix}_{(n+1)\times 1} - \begin{bmatrix} b_1 \\ \dots \\ b_i \\ \dots \\ b_N \end{bmatrix}_{N\times 1} = \begin{bmatrix} x_{11}w_1+\dots+x_{1n}w_n+w_{n+1}-b_1 \\ \dots \\ x_{i1}w_1+\dots+x_{in}w_n+w_{n+1}-b_i \\ \dots \\ -x_{N1}w_1-\dots-x_{Nn}w_n-w_{n+1}-b_N \end{bmatrix}_{N\times 1} = \begin{bmatrix} \boldsymbol{W}^{\mathrm{T}}\boldsymbol{X}_1-b_1 \\ \dots \\ \boldsymbol{W}^{\mathrm{T}}\boldsymbol{X}_i-b_i \\ \dots \\ \boldsymbol{W}^{\mathrm{T}}\boldsymbol{X}_N-b_N \end{bmatrix}$$

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

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

最小平方误差算法

LMSE算法原理

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。

最小平方误差算法

LMSE递推公式

与问题相关的两个梯度：$\dfrac{\partial J}{\partial \boldsymbol{W}}=\boldsymbol{X}^{\mathrm{T}}\left(\boldsymbol{X}\boldsymbol{W}-\boldsymbol{B}\right)$　　$\dfrac{\partial J}{\partial \boldsymbol{B}}=-\dfrac{1}{2}\left[\left(\boldsymbol{X}\boldsymbol{W}-\boldsymbol{B}\right)+\left|\boldsymbol{X}\boldsymbol{W}-\boldsymbol{B}\right|\right]$

求递推公式：

(1) 求$\boldsymbol{W}$的递推关系

使J对$\boldsymbol{W}$求最小，令 $\dfrac{\partial J}{\partial \boldsymbol{W}}=0$，得：　$\boldsymbol{X}^{\mathrm{T}}\left(\boldsymbol{X}\boldsymbol{W}-\boldsymbol{B}\right)=0\Rightarrow \boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}\boldsymbol{W}=\boldsymbol{X}^{\mathrm{T}}\boldsymbol{B}\Rightarrow \boldsymbol{W}=\left(\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}\right)^{-1}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{B}=\boldsymbol{X}^{\#}\boldsymbol{B}$

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

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

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

最小平方误差算法

LMSE递推公式

(2) 求$\boldsymbol{B}(k+1)$的迭代式

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

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

代入 $\frac{\partial J}{\partial \boldsymbol{B}}=-\frac{1}{2}\left[(\boldsymbol{XW}-\boldsymbol{B})+|\boldsymbol{XW}-\boldsymbol{B}|\right]$，有 $\boldsymbol{B}(k+1)=\boldsymbol{B}(k)+\frac{\eta'}{2}\left[\left(\boldsymbol{XW}(k)-\boldsymbol{B}(k)\right)+\left|\boldsymbol{XW}(k)-\boldsymbol{B}(k)\right|\right]$

令 $\eta=\eta'/2$，定义$\boldsymbol{e}(k)=\boldsymbol{XW}(k)-\boldsymbol{B}(k)$ ，有：

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

最小平方误差算法

LMSE递推公式

(3) 求$\boldsymbol{W}(k+1)$的迭代式

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

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

$$=\boldsymbol{X}^{\#}\boldsymbol{B}(k)+\boldsymbol{X}^{\#}\eta\boldsymbol{e}(k)+\boldsymbol{X}^{\#}\eta\left|\boldsymbol{e}(k)\right|$$

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

$\boldsymbol{X}^{\#}\eta\boldsymbol{e}(k)=\mathbf{0}$

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

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

$$=\mathbf{0}$$

最小平方误差算法

LMSE递推公式总结　设初值 $\boldsymbol{B}(1)$，各分量均为正值，括号中数字代表迭代次数 。

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

.....

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

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

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

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

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

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

.....

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

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

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

求出$\boldsymbol{B}$，$\boldsymbol{W}$后，再迭代出下一个$\boldsymbol{e}$，

从而计算出新的$\boldsymbol{B}$，　$\boldsymbol{W}$。

最小平方误差算法

3）模式类别可分性判别

可以证明：当模式类线性可分，且学习率 η 满足 $0<\eta\le 1$ 时，该算法收敛，可求得解W。

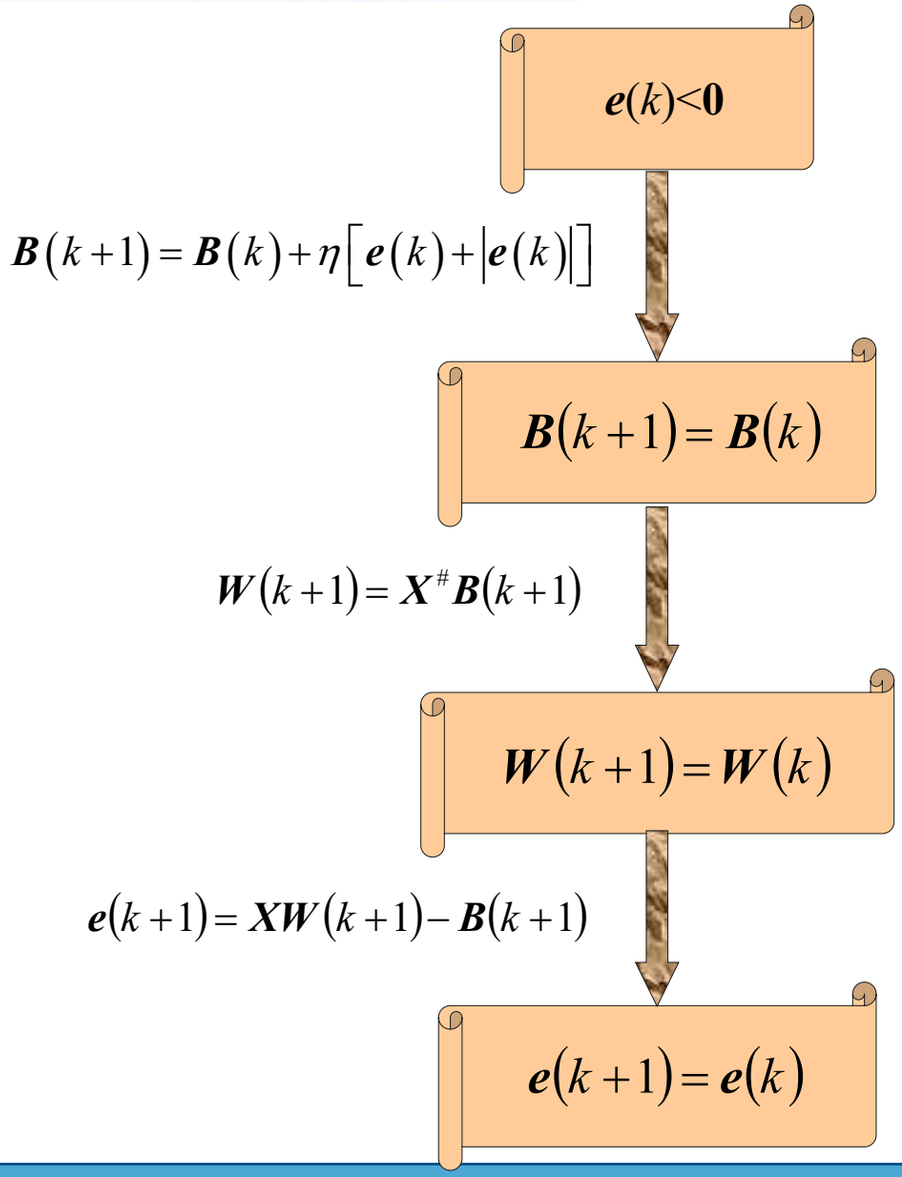
理论上不能证明该算法到底需要迭代多少步才能达到收敛，通常在每次迭代计算后检查一下$\boldsymbol{XW}(k)$和误差向量$\boldsymbol{e}(k)$，从而可以判断是否收敛。

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。



最小平方误差算法

综上所述：

1、只有当$\boldsymbol{e}(k)$中有大于零的分量时，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。

2、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。

最小平方误差算法

4) LMSE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 η 和$\boldsymbol{B}(1)$，η为正的校正增量，$\boldsymbol{B}(1)$的各分量大于零，迭代次数 k=1 。开始迭代：

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

……

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

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

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

如果$\boldsymbol{e}(k)<\boldsymbol{0}$，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)。

最小平方误差算法

4) LMSE算法描述

(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，返回(4)。

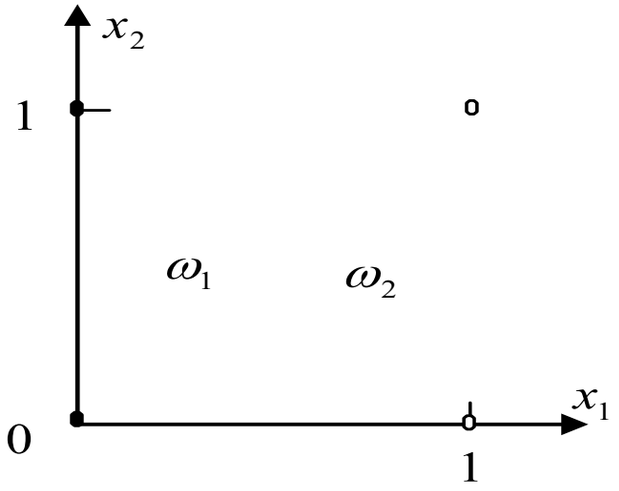
3. 算法特点

(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。

最小平方误差算法



例1 已知两类模式训练样本：ω_1：$[0,0]^T$, $[0,1]^T$，ω_2：$[1,0]^T$, $[1,1]^T$；

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

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

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

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

最小平方误差算法

例1 已知两类模式训练样本：ω_1：$[0,0]^T$, $[0,1]^T$，ω_2：$[1,0]^T$, $[1,1]^T$；

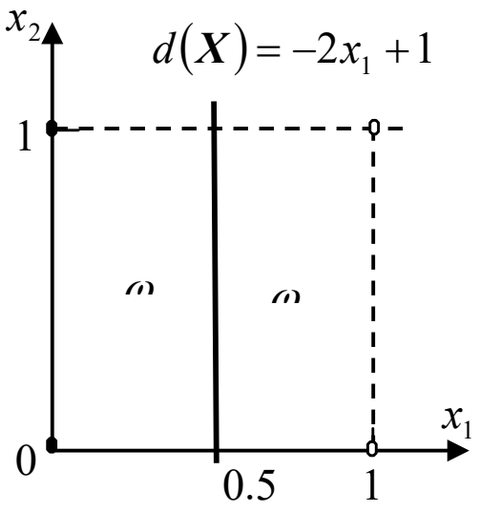
试用LMSE算法求解权向量。

解：(3) 取 $B(1)=[1,1,1,1]^T$ 和 η=1 开始迭代：

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

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

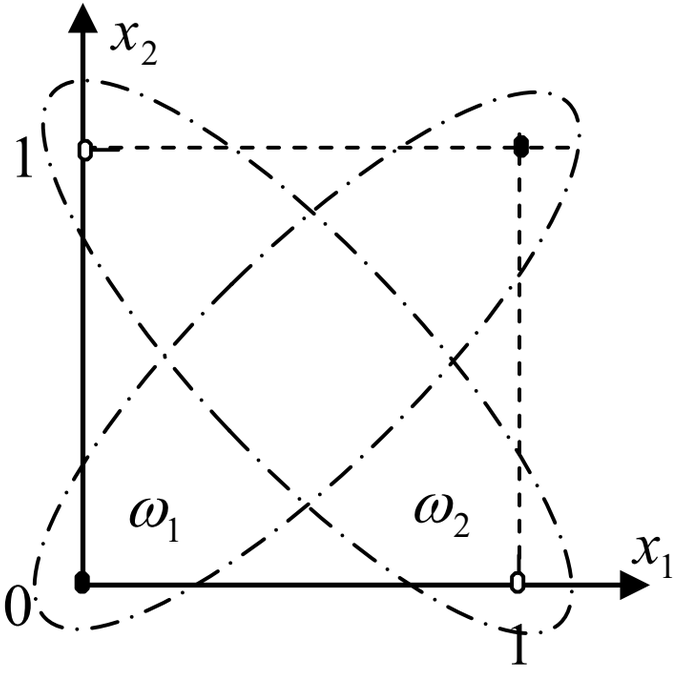
解为 $\boldsymbol{W}(1)$，判断函数为：$d(\boldsymbol{X})=-2x_1+1$



最小平方误差算法

例 已知模式训练样本：ω_1：$[0,0]^{\mathrm{T}}, [1,1]^{\mathrm{T}}$，$\omega_2$：$[0,1]^{\mathrm{T}}$，$[1,0]^{\mathrm{T}}$；

用LMSE算法求解权向量。



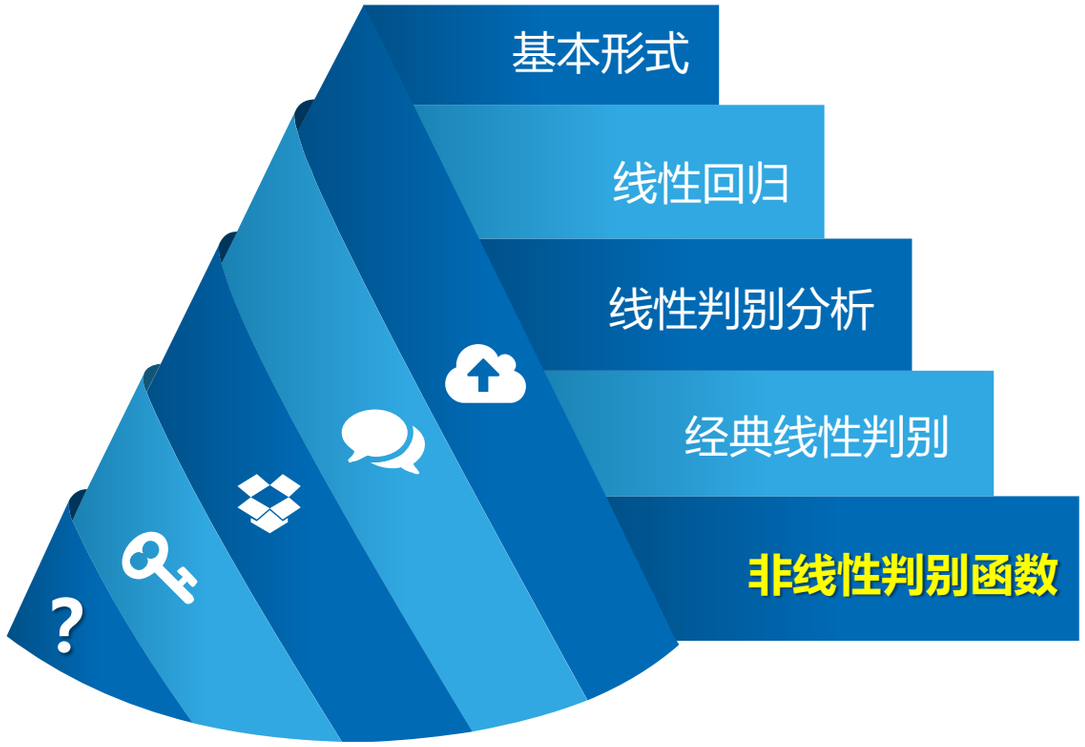
解：(1) 规范化增广样本矩阵：$\boldsymbol{X}=\begin{bmatrix}0 & 0 & 1\\ 1 & 1 & 1\\ 0 & -1 & -1\\ -1 & 0 & -1\end{bmatrix}$

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

(3) 取 $\boldsymbol{B}(1)=[1,1,1,1]^{\mathrm{T}}$ 和 $\eta=1$，迭代：$\boldsymbol{W}(1)=\boldsymbol{X}^{\#}\boldsymbol{B}(1)=[0 \quad 0 \quad 0]^{\mathrm{T}}$

$$\boldsymbol{e}(1)=\boldsymbol{X}\boldsymbol{W}(1)-\boldsymbol{B}(1)=[0 \quad 0 \quad 0 \quad 0]^{\mathrm{T}}-[1 \quad 1 \quad 1 \quad 1]^{\mathrm{T}}=[-1 \quad -1 \quad -1 \quad -1]^{\mathrm{T}}$$

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



- **线性模型基本形式**
  线性方程，基本形式，…
- **线性回归**
  参数估计，最小二乘，对数几率回归，…
- **线性判别分析**
  基本概念，广义线性判别，几何性质 …
- **经典线性判别准则**
  Fisher判别，感知器准则、梯度法 …
- **非线性判别函数**
  分段线性，…

非线性判别函数

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。

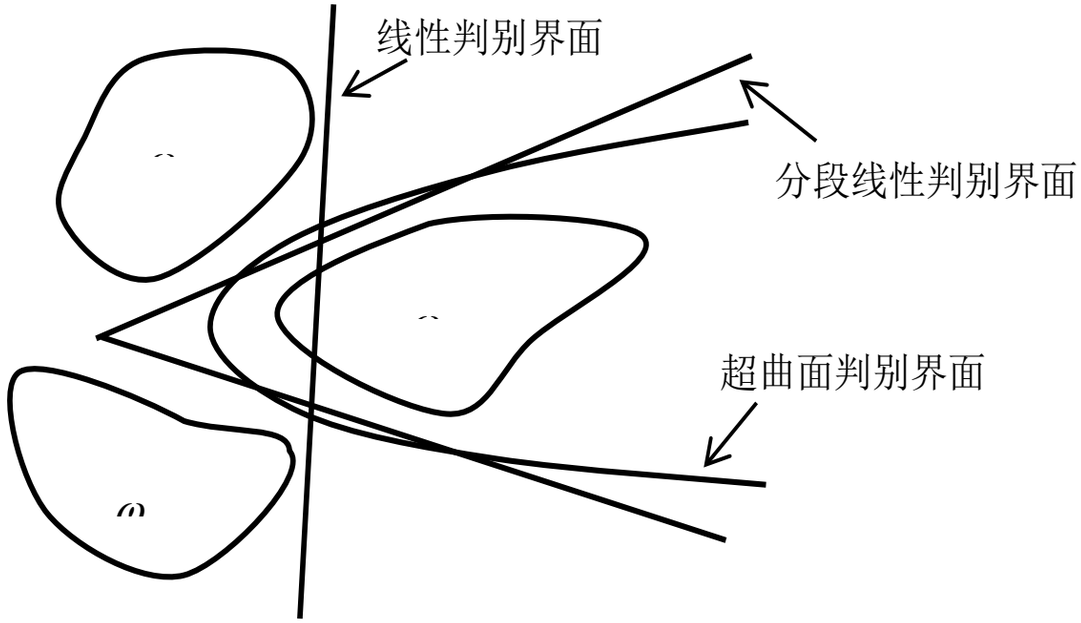
非线性判别函数：用于线性不可分情况。分段线性、超曲面。

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。



非线性判别函数

1．一般分段线性判别函数

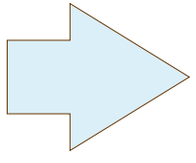
设有M类模式，将ω_i类（i=1,2, …,M）划分为l_i个子类：　$\omega_i:\{\omega_i^1,\omega_i^2,...,\omega_i^{l_i}\}$　　$i=1,2,...,M$

其中第n个子类的判别函数：　$d_i^n(\boldsymbol{X})=(\boldsymbol{W}_i^n)^{\mathrm{T}}\boldsymbol{X}$　　$n=1,2,...,l_i;\quad i=1,2,...,M$

ω_i类的判别函数定义为：　　$d_i(\boldsymbol{X})=\max\{d_i^n(\boldsymbol{X}),\quad n=1,2,...,l_i\}$

M类的判决规则：若 $d_j(\boldsymbol{X})=\max\{d_i(\boldsymbol{X}),\quad i=1,2,...,M\}$ ，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



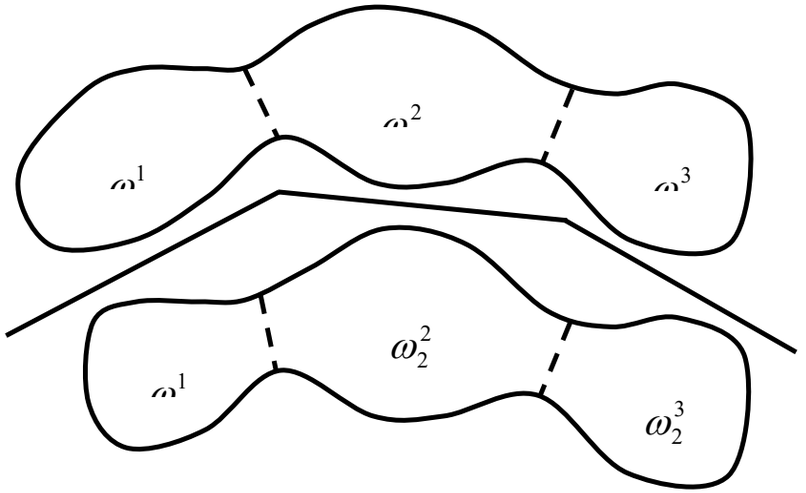
判别面由各子类
的判别函数决定

非线性判别函数

1．一般分段线性判别函数

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

非线性判别函数

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：$\boldsymbol{M}_1 = \frac{1}{N_1}\sum_{i=1}^{N_1}\boldsymbol{X}_i$　　ω_2类均值向量：$\boldsymbol{M}_2 = \frac{1}{N_2}\sum_{i=1}^{N_2}\boldsymbol{X}_i$　　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：

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

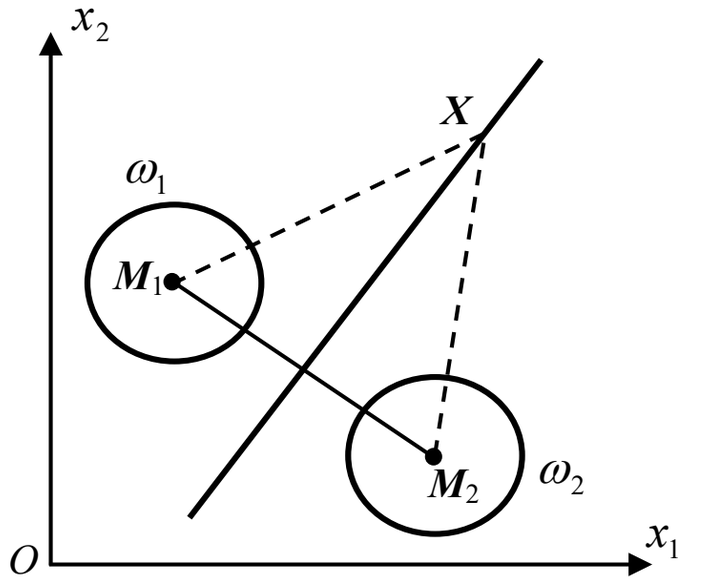
判决规则：

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：

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



—— 最小距离分类器

非线性判别函数

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：

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

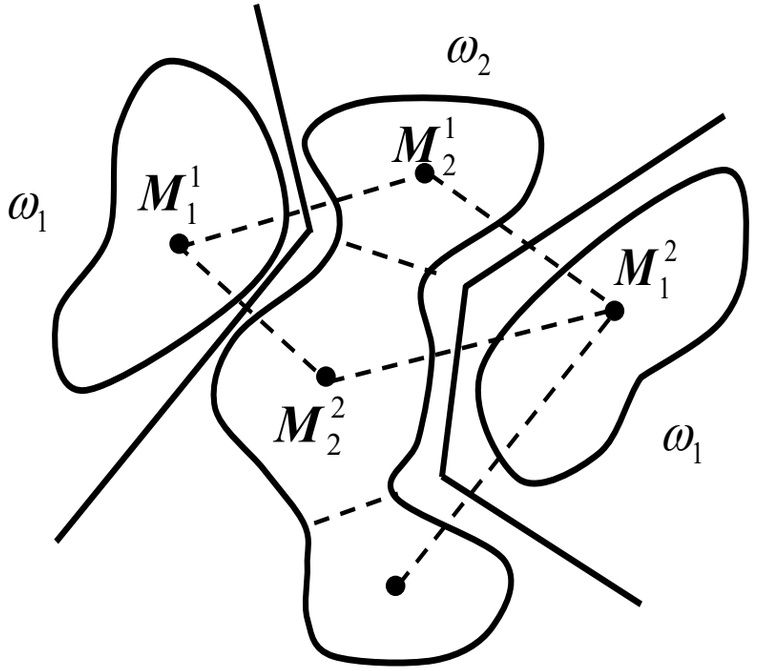
每类的判别函数：

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

判决规则：

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

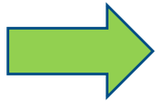
则 $\boldsymbol{X} \in \omega_j$



类别不平衡问题

类别不平衡（class imbalance）

- 不同类别训练样例数相差很大情况（正类为小类）

类别平衡正例预测 $\frac{y}{1-y} > 1$  $\frac{y}{1-y} > \frac{m^+}{m^-}$ 正负类比例

类别不平衡学习的一个基本策略：再缩放（rescaling）

- 欠采样（undersampling）
  - 去除一些反例使正反例数目接近（EasyEnsemble [Liu et al.,2009]）
- 过采样（oversampling）
  - 增加一些正例使正反例数目接近（SMOTE [Chawla et al.2002]）
- 阈值移动（threshold-moving）

Any Questions?