

—武大本本科生课程



# 第3讲 线性判别分析

( Lecture 3 Linear discriminant analysis )

武汉大学计算机学院机器学习课程组

2024.03

# 第3章 线性判别分析

(Chapter 3: Linear Discriminant Analysis)

本章内容目录 (红色为本讲讲授内容)

3.1 判别函数

3.2 线性判别函数

3.3 线性判别函数的性质

3.4 线性分类器设计

3.4.1 梯度下降法

3.4.2 Fisher法(\*: 选学)

3.4.3 感知器法(\*: 神经网络中介绍)

3.4.4 逻辑回归(\*: 极大似然估计应用中介绍)

小结

# 机器学习中的建模

---

- 经验模型的应用性质

- **描述性建模**：以方便的形式给出数据的主要特征，实质上是对数据的概括，以便在大量的或有噪声的数据中仍能观察到重要特征。重在认识数据的主要概貌，理解数据的重要特征。
- **预测性建模**：以函数的形式给出感兴趣量（预测量）与可观测量之间的数量关系，实质上是根据观测到的对象特征来预测对象的其他特征。重在把握协变关系，据此进行预测。

- 描述性建模任务

- 聚类分析，数据降维，流形学习，密度估计，异常分析，可视化

- 预测性建模任务

- 分类（类别预测），回归（数值预测），评分（排名预测）

# 机器学习中的建模

---

- **预测性建模方法**

- **概率方法：生成式建模方法**，借助训练数据对同类数据的生成机制（概率分布）进行估计，基于概率关系对变量取值进行概率预测。把模式视为随机变量的抽样，利用统计决策理论（贝叶斯统计）成熟的判决准则与方法，对模式样本进行分类

如：贝叶斯分类器、贝叶斯网络（概率图模型）、高斯混合模型、隐马尔可夫模型、受限玻尔兹曼机、生成对抗网络，变分自动编码器

- **代数方法：判别式建模方法**，借助训练数据对观测量和预测量的函数关系进行直接建模，基于函数关系对变量取值进行数值预测。利用向量空间的直观概念，使用代数方程方法，对模式进行分类

如：KNN，感知机，判别分析，决策树，随机森林，支持向量机、逻辑回归，神经网络

## 3.1 判别函数

### 线性可分概念与线性分类算法

一个分类问题是否属于**线性可分**，取决于是否有可能找到一个点、直线、平面或**超平面**来分离两个相邻的类别。

如果每个类别样本的分布范围本身是全连通的单一凸集，且互不重叠，则这两个类别一定是线性可分的，如图所示。

**线性分类算法**主要有**线性判别函数**、**Fisher判别分析**、**单层感知器**、**逻辑回归**等。

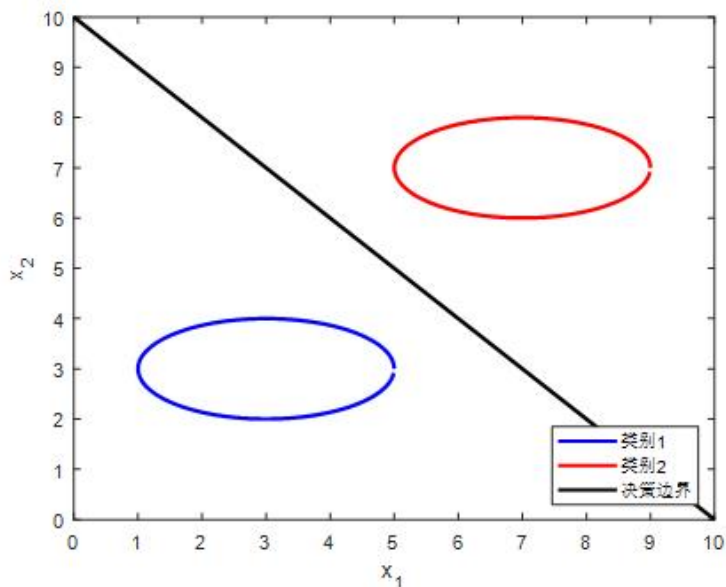


图3.1 线性可分情况

# 1. 判别函数的定义

直接用来对模式进行分类的决策函数。

若分属于 $\omega_1$ ,  $\omega_2$ 两类的 $n$ 维模式在空间中的分布区域, 可以用一代数方程 $d(X)=0$ 决定的超平面作为分隔面, 两类样本分布在分隔面的两侧, 那么就称 $d(X)$ 为判别函数(discriminant function)或称决策函数(decision function)。代数方程 $d(X)=0$ 表示的是 $n$ 维空间的 $(n-1)$ 维判决面 {或超平面(hyperplane)或超曲面(hypersurface), 视 $d(x)$ 形式而定}。

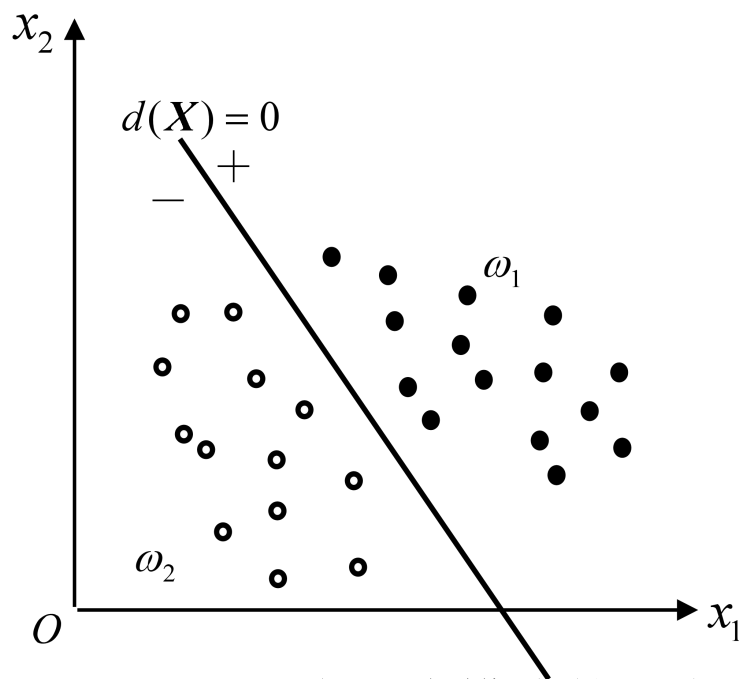


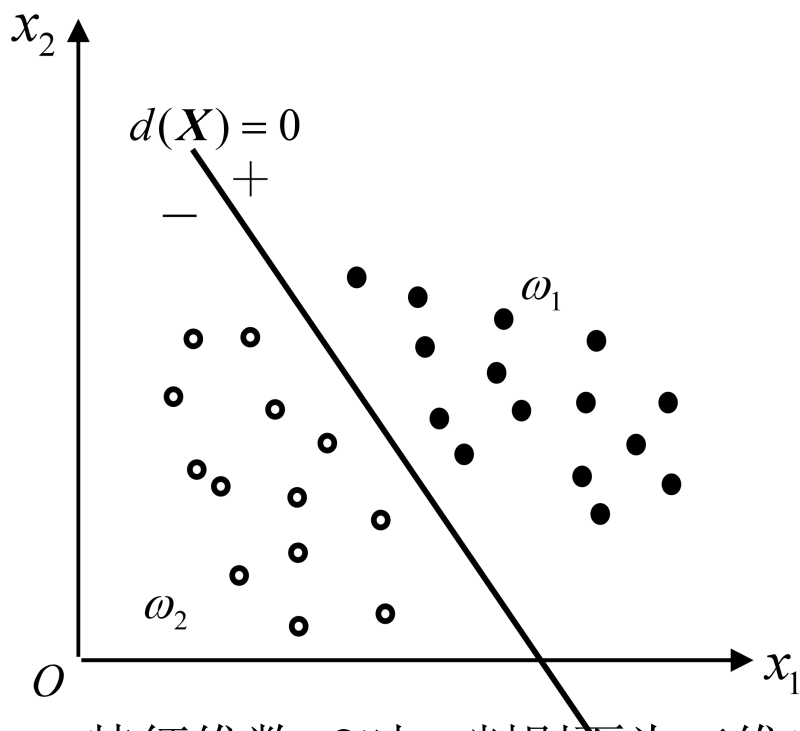
图3.2 两类二维模式的分布

例：一个二维的两类判别问题，模式分布如图示，这些分属于 $\omega_1$ 、 $\omega_2$ 两类的模式可用平面中的直线  $d(X)=0$  来划分。

$$d(X) = w_1x_1 + w_2x_2 + w_3 = 0$$

式中： $x_1, x_2$  为坐标变量，

$w_1, w_2, w_3$  为方程参数。



示例：线性判别函数（维数为2）。

将某一未知模式  $\mathbf{X}$  代入：

$$d(\mathbf{X}) = w_1 x_1 + w_2 x_2 + w_3$$

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

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

若  $d(\mathbf{X}) = 0$ ，则  $\mathbf{X} \in \omega_1$  或  $\mathbf{X} \in \omega_2$   
或：拒绝分类

特征维数=3时：判别面为三维空间中的平面。

特征维数>3时：判别面为高维空间中的超平面。

注：为了清晰地了解  $d(\mathbf{x})$  的含义，应该画出**判别函数值  $d(\mathbf{x})$  这一轴**，在没有画出的时候，就在自变量（模式）空间中画出  $d(\mathbf{x})$  取正负值的区域——这就是所谓**判别面的正侧、负侧**。

## 2. 判别函数正负值的设定

判别面的正负侧，是在训练判别函数的权值时**人为设定的**。但是一旦设定后，训练好的模型(分隔面)的正负侧就是确定的，而且由判别函数的梯度向(矢)量确定。

一般，令第1类样本的函数值大于零，第2类样本的函数值小于零。也就是说，类别标号值为 $\{1, -1\}$ 。也有用 $\{1, 0\}$ 表示正负类的。

具体值的大小不重要，主要便于算法公式的简洁。

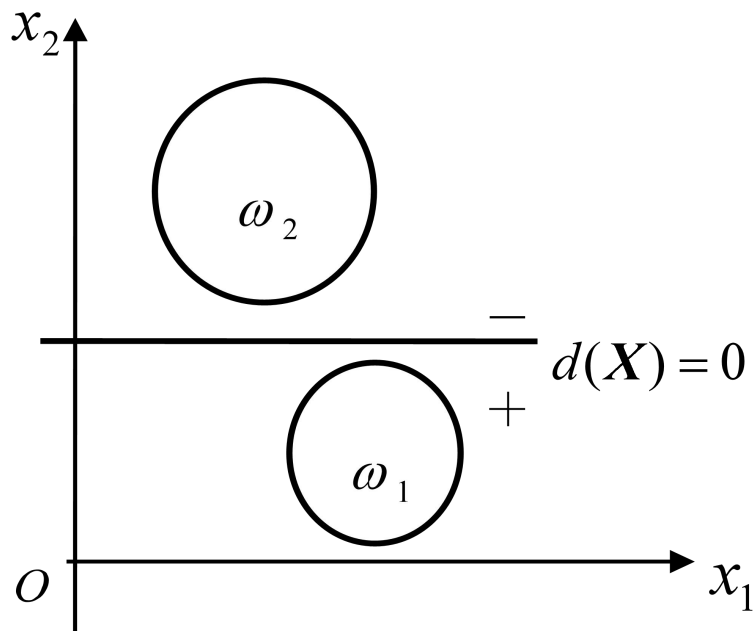


图3.3 判别函数正负的指定

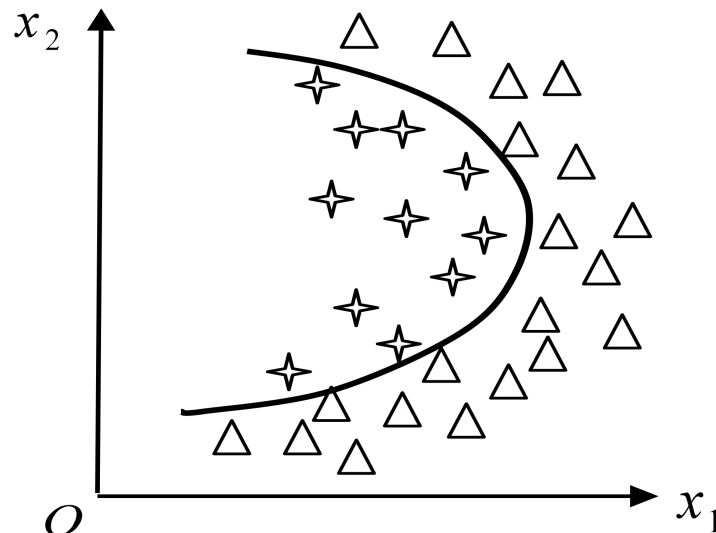
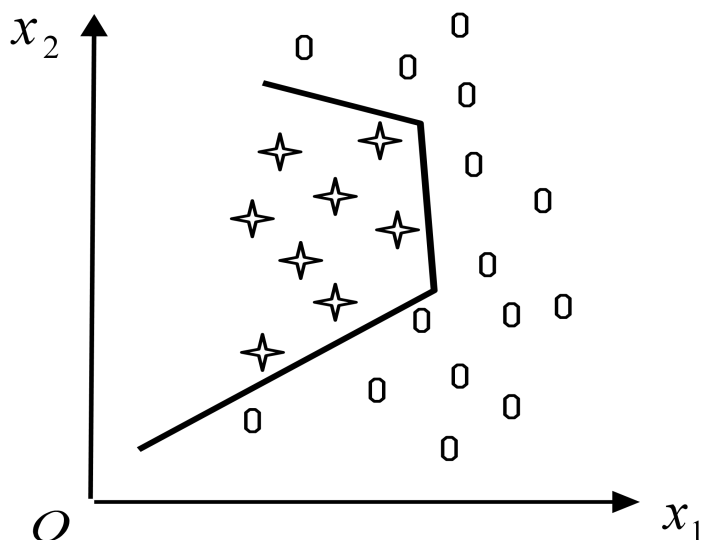


### 3. 确定判别函数的两个因素

1) 判决函数 $d(\mathbf{X})$ 的函数形式：它可以是特征的线性或非线性的函数。

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： $d(\mathbf{X}) = w_1x_1 + w_2x_2 + w_3x_3 + w_4$

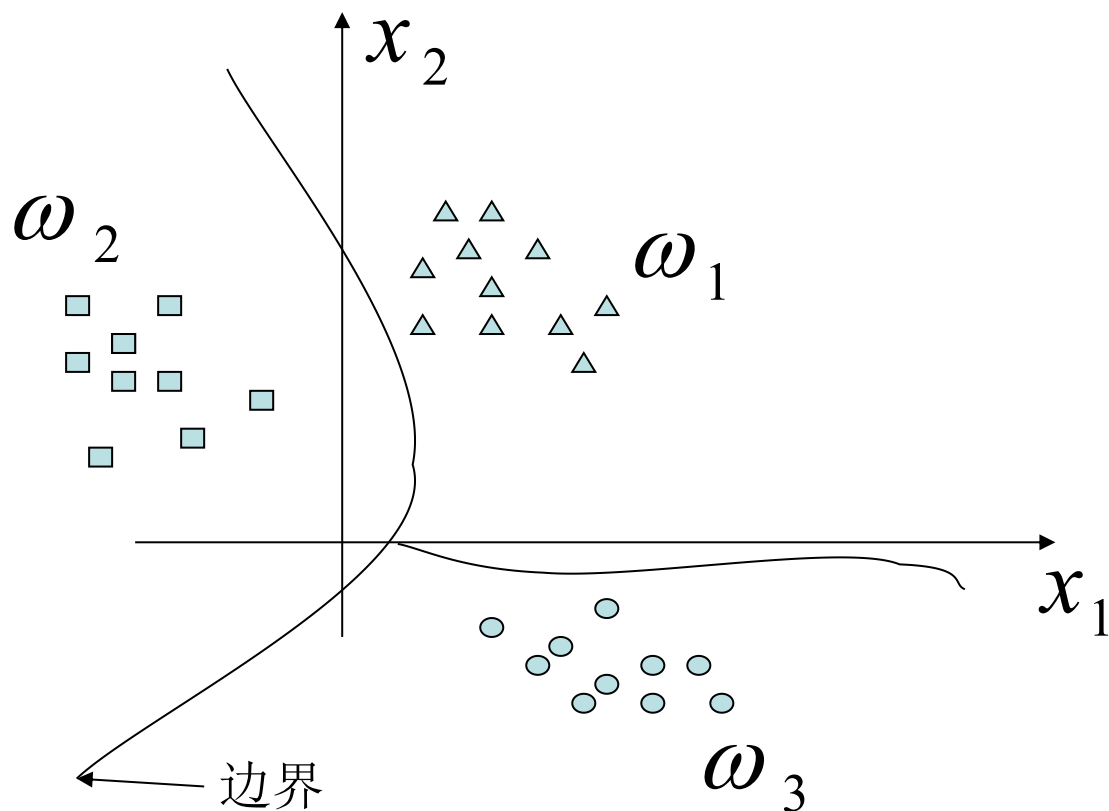
例：非线性判决函数（包括：分段线性）



2) 判决函数 $d(\mathbf{X})$ 的系数：用所给的模式样本，通过优化准则确定。主要介绍线性判别函数。一个一般的 $n$ 元线性函数应该具有什么样的性质才适合做两分类和多分类的判别函数

# 3.1 判别函数

如下图：三类的分类问题，它们的边界线就是一个判别函数。



# 3.1 判别函数

假设对一模式 **$X$** 已提取 **$n$** 个特征，表示为：

$$X = (x_1, x_2, x_3, \dots, x_n)^T$$

$X$ 是 **$n$** 维空间的一个向量

模式识别问题就是根据模式 **$X$** 的 **$n$** 个特征(指标)来判别模式属于 $\omega_1, \omega_2, \dots, \omega_m$ 类中的哪一类。

# 3.1 判别函数

- 判别函数包含两类：

一类是线性判别函数：

- 线性判别函数
- 广义线性判别函数

– （所谓广义线性判别函数就是把非线性判别函数映射到另外一个空间变成线性判别函数）

- 分段线性判别函数

另一类是非线性判别函数

## 3.2 线性判别函数

分别对两类问题和多类问题进行讨论。

➤(一) 两类问题 即： $\omega_i = (\omega_1, \omega_2)^T, M = 2$

1. 二维情况：特征向量取两个特征

$$X = (x_1, x_2)^T, n = 2$$

这种情况下线性判别函数为：

$$g(x) = w_1 x_1 + w_2 x_2 + w_3$$

$w_i$ 为参数； $x_1, x_2$ 为坐标向量

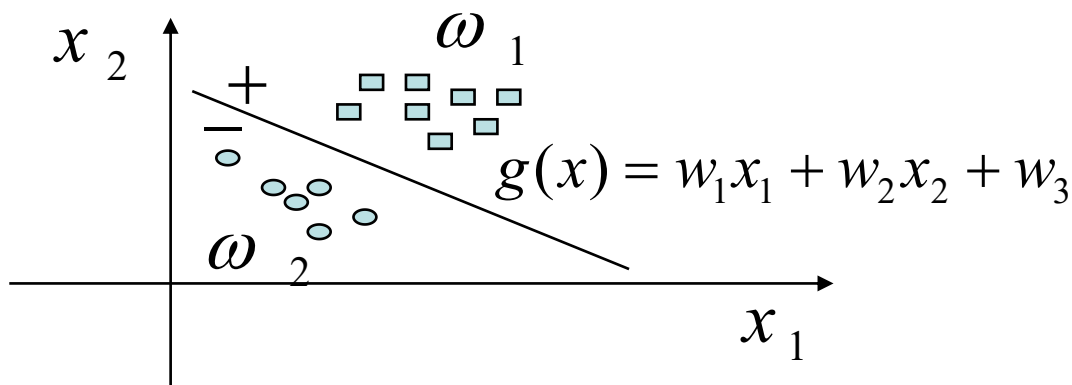
## 3.2 线性判别函数

### ➤ 1. 二维情况

在两类别情况下，判别函数  $g(x)$  具有以下性质：

$$g(x) = \begin{cases} > 0, X \in \omega_1 \\ < 0, X \in \omega_2 \end{cases} \quad g(x) = 0, X \text{不定}$$

这是二维情况下判别由判别边界分类。情况如图：



## 3.2 线性判别函数

### ➤ 2. n维情况

现提取n个特征为:  $X = (x_1, x_2, x_3, \dots, x_n)^T$

判别函数:  $g(x) = w_1 x_1 + w_2 x_2 + \dots + w_n x_n + w_{n+1}$

或  $g(x) = W_0^T X + w_{n+1}$

$W_0 = (w_1, w_2, \dots, w_n)^T$  为权向量,

$X = (x_1, x_2, \dots, x_n)^T$  为模式向量。

另外一种表示方法:  $g(x) = W^T X$

$W = (w_1, w_2, \dots, w_n, w_{n+1})^T$  为增广权向量,

$X = (x_1, x_2, \dots, x_n, 1)^T$  为增广模式向量。

## 3.2 线性判别函数

\*回顾：一元/多元线性回归(Multivariate linear regression)

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

其中：

$$\mathbf{x}_i = (x_{i1}, x_{i2}, \dots, x_{in}, 1)^T \quad (i = 1, 2, \dots, m),$$

设给定 $m$ 个模式, 各模式样本有 $n$ 个特征,  $y_i \in R$ .

$$\mathbf{w} = (w_1, w_2, \dots, w_n, w_{n+1})^T$$

我们试图从数据集 $D = \{(\mathbf{x}_i, y_i)\}_{i=1}^m$ 学得

$$g(\mathbf{x}_i) = \mathbf{w}^T \mathbf{x}_i, \text{使得 } g(\mathbf{x}_i) = y_i \text{ 的 } \mathbf{w}.$$

$\mathbf{w}$ 参数学习/训练出来之后, 所确定的 $g(\mathbf{x})$ 模型也叫预测模型。

若用 $g(\mathbf{x})$ 预测的是离散值, 此类学习任务称为“分类(Classification)”; 若用 $g(\mathbf{x})$ 预测的是连续值, 此类学习任务称为“回归(regression)”。由于这里 $g(\mathbf{x})$ 是线性的, 故称为多元线性回归(在大二《概率与统计》课程中, 大家学过最小二乘法求一元/多元线性回归)。



## 3.2 线性判别函数

### ➤ 2. n维情况

❖ 模式分类:

$$g(x) = W^T X \begin{cases} > 0, X \in \omega_1 \\ < 0, X \in \omega_2 \end{cases}$$

❖  $g(x) = W^T X = 0$  为判别边界/决策边界(decision boundary)。  
当  $n=2$  时，二维情况的判别边界为一直线；当  $n=3$  时，判别边界为平面；当  $n>3$  时，则判别边界称为超平面(Hyperplane)。

## 3.2 线性判别函数

### ➤(二) 多类问题

对于多类问题，模式有  $\omega_1, \omega_2, \dots, \omega_m$  个类别。可分三种情况讨论：

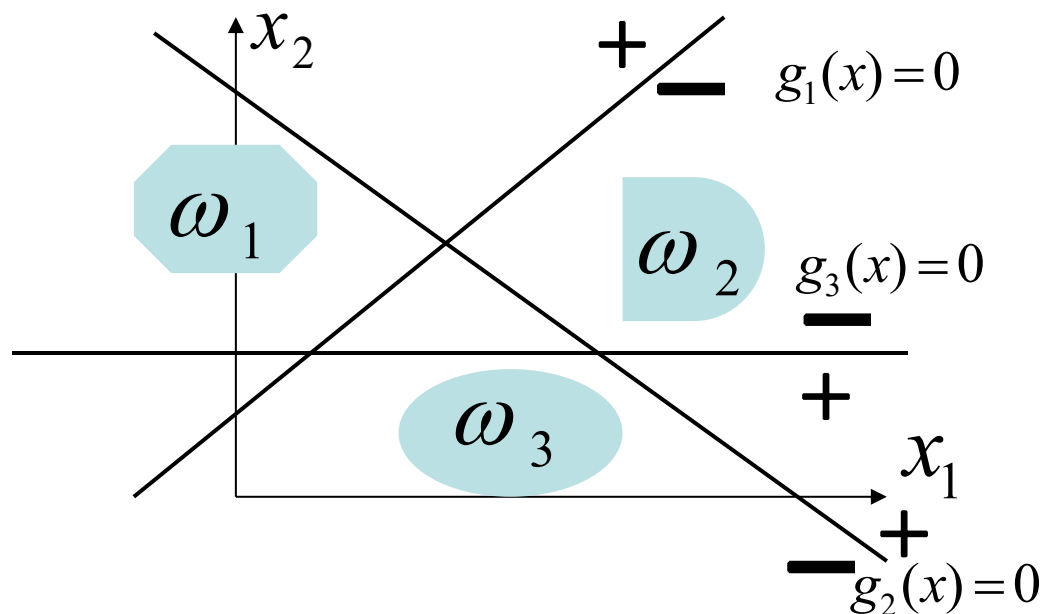
**多类情况1** ( $\omega_i/\bar{\omega}_i$ 是非两分法)：每一模式类与其他模式类间可用单个判别平面把一个类分开。这种情况，M类可有M个判别函数，且具有以下性质：

$$g_i(x) = W_i^T X \begin{cases} > 0, X \in \omega_i \\ < 0, \text{其他}, i = 1, 2, \dots, M. \end{cases}$$

式中  $W_i = (w_{i1}, w_{i2}, \dots, w_{in}, w_{in+1})^T$  为第*i*个判别函数的权向量。

## ➤ 多类情况1

- ❖ 下图所示，每一类别可用单个判别边界与其他类别分开。  
若一模式 $X$ 属于 $\omega_1$ ，则由图可清楚看出：这时 $g_1(x) > 0$ 而 $g_2(x) < 0$ ， $g_3(x) < 0$ 。 $\omega_1$ 类与其他类之间的边界由 $g_1(x)=0$ 确定。



## ➤ 多类情况1 (Cont.)

❖ 例：已知三类 $\omega_1, \omega_2, \omega_3$ 的判别函数分别为

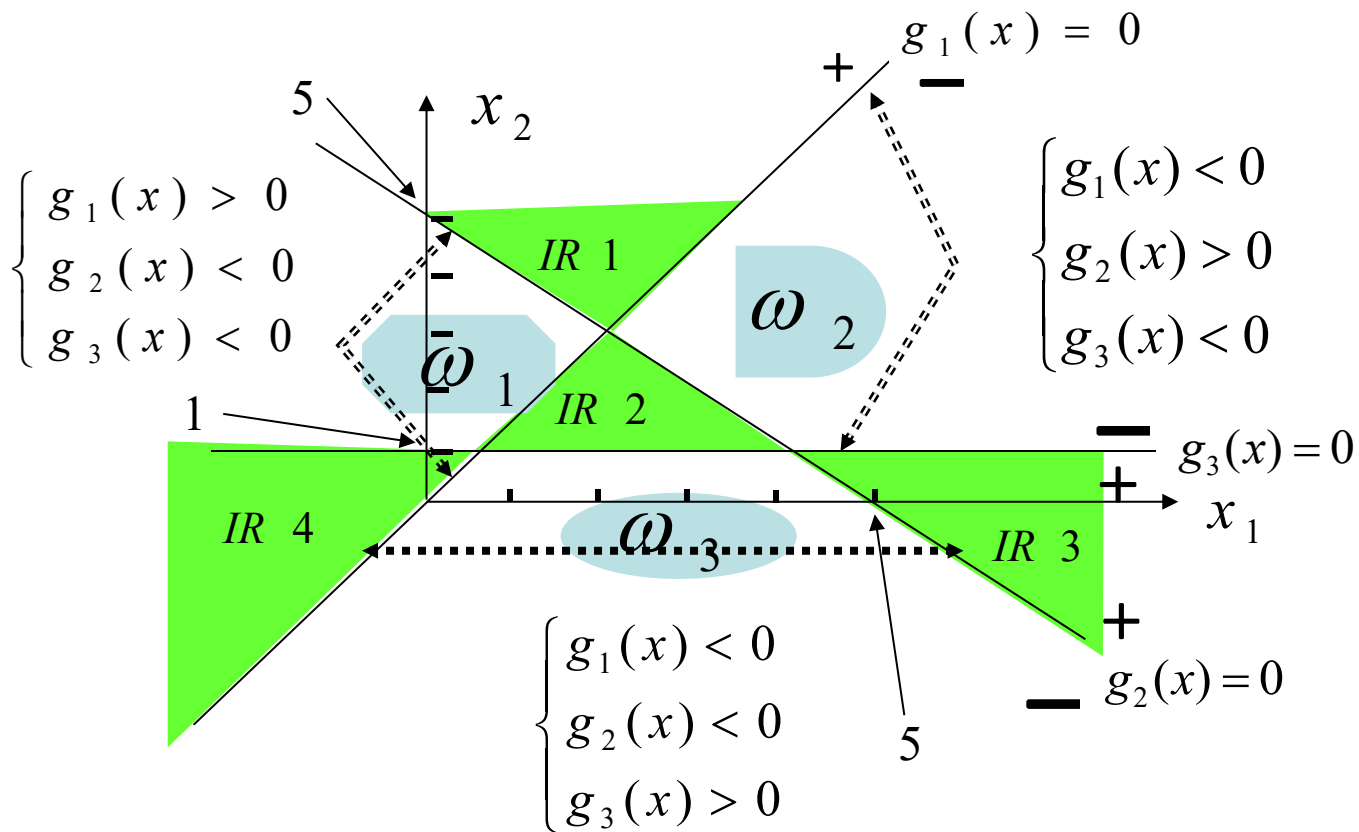
$$\begin{cases} g_1(x) = -x_1 + x_2 \\ g_2(x) = x_1 + x_2 - 5 \\ g_3(x) = -x_2 + 1 \end{cases}$$

则，三个判别边界为

$$\begin{cases} g_1(x) = -x_1 + x_2 = 0 \\ g_2(x) = x_1 + x_2 - 5 = 0 \\ g_3(x) = -x_2 + 1 = 0 \end{cases}$$

# ➤ 多类情况1 (Cont.)

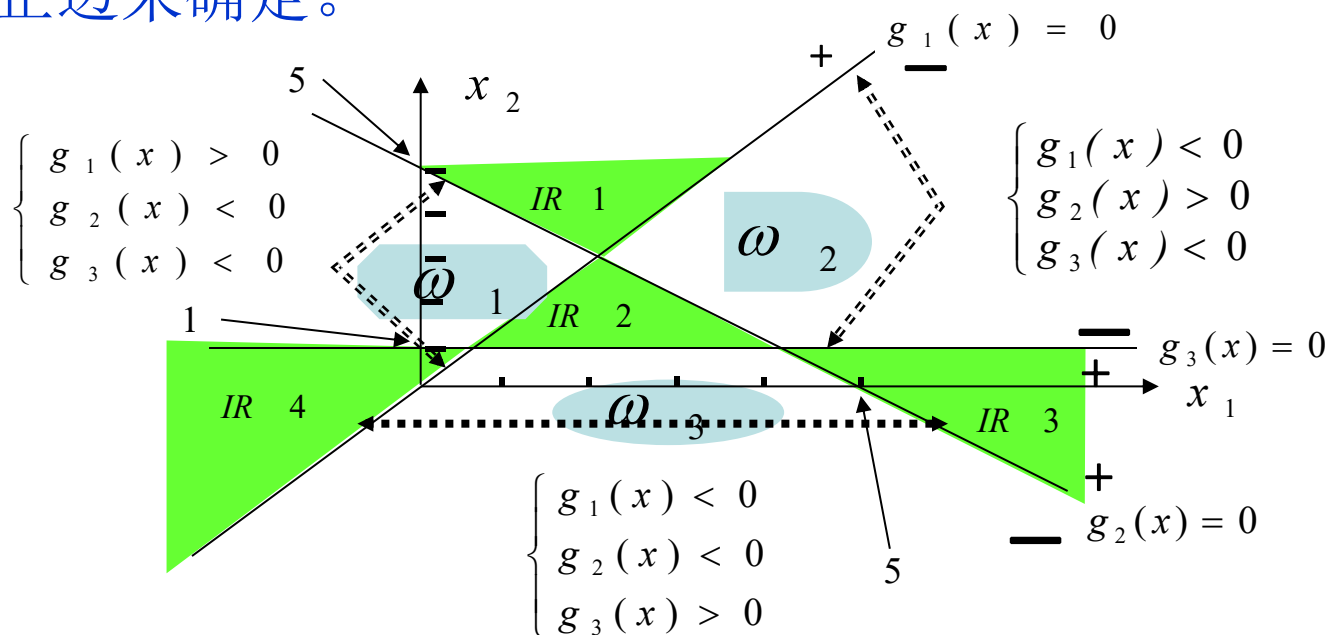
作图如下:



IR: Indeterminate Region(不确定区域)

## ➤ 多类情况1 (Cont.)

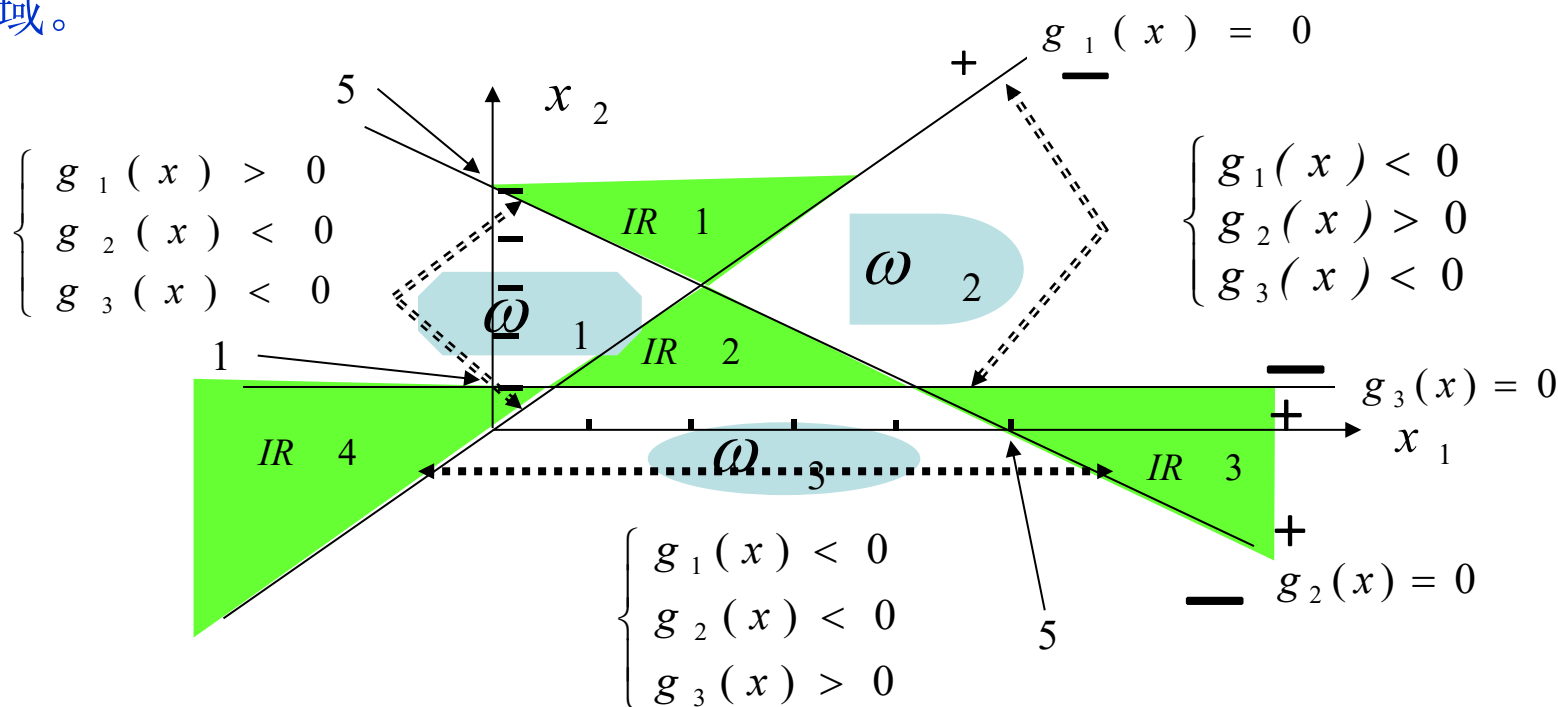
- ❖ 对于任一模式 $X$ ，若它的  $g_1(x) > 0$  ,  $g_2(x) < 0$  ,  $g_3(x) < 0$
- ❖ 则该模式属于 $\omega_1$ 类。相应 $\omega_1$ 类的区域由直线 $g_3(x) = -x_2 + 1 = 0$  的负边、直线 $g_2(x) = x_1 + x_2 - 5 = 0$  的负边和直线 $g_1(x) = -x_1 + x_2 = 0$  的正边来确定。



## ➤ 多类情况1 (Cont.)

❖ 必须指出，如果某个 $x$ 使二个以上的判别函数  $g_i(x) > 0$ ，则此模式 $x$ 就无法作出准确的判决。如图中 IR1, IR3, IR4区域。

❖ 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。



## ➤ 多类情况1 (Cont. )

❖ 问当 $x=(x_1, x_2)^T=(6, 5)^T$ 时属于那一类

代入判别函数方程组：

$$\begin{cases} g_1(x) = -x_1 + x_2 \\ g_2(x) = x_1 + x_2 - 5 \\ g_3(x) = -x_2 + 1 \end{cases}$$

得：

$$g_1(x) = -1, g_2(x) = 6, g_3(x) = -4.$$

❖ 结论：  $g_1(x) < 0$  ,  $g_2(x) > 0$  ,  $g_3(x) < 0$ ； 所以它属于 $\omega_2$ 类



例：设有一个二维三类问题，其三个判别函数为：

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

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

解：将 $\mathbf{X}=[7,5]^T$ 代入上三式，有：

$$d_1(X) = -7 + 5 + 1 = -1 < 0$$

$$d_2(X) = 7 + 5 - 4 = 8 > 0$$

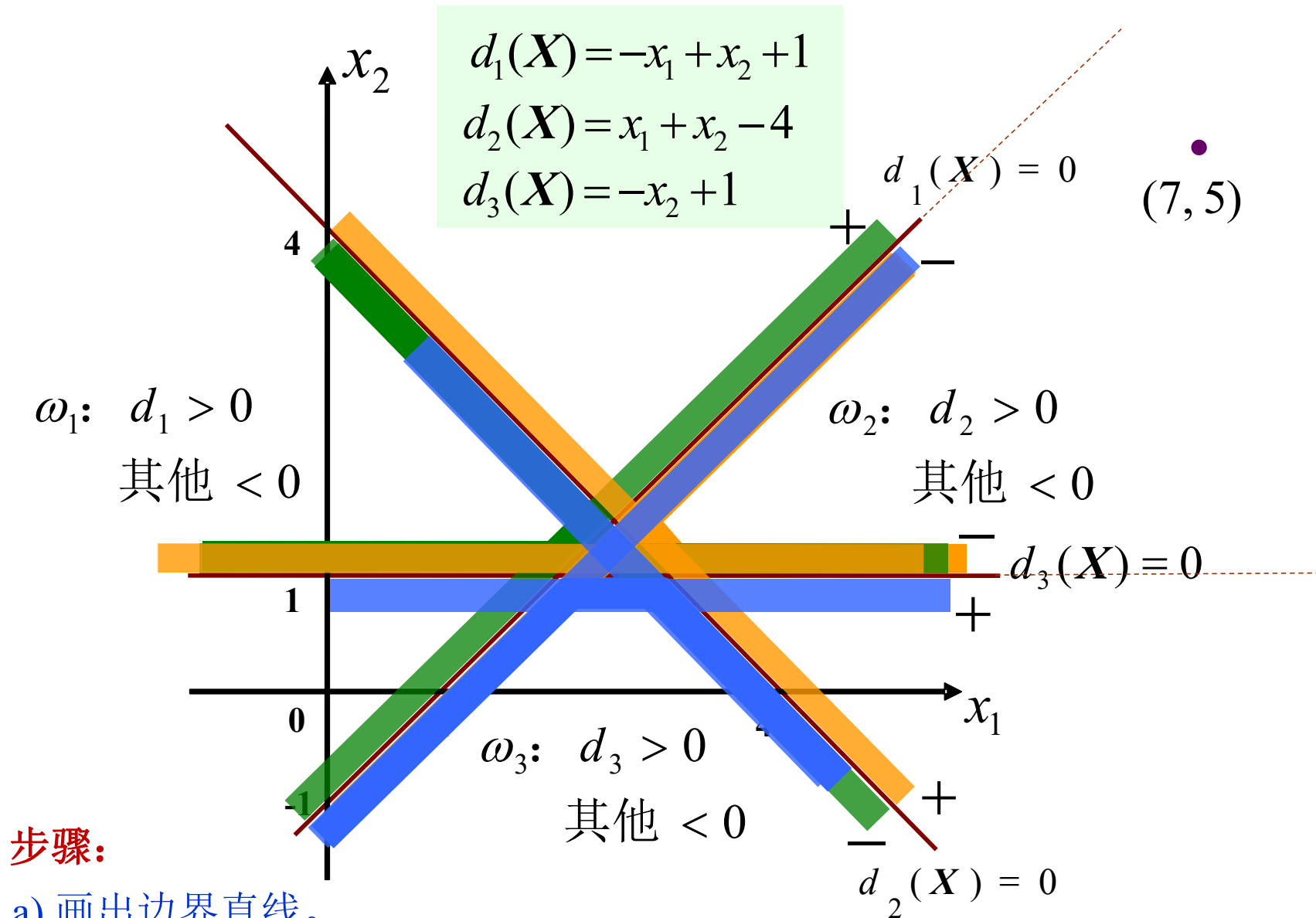
$$d_3(X) = -5 + 1 = -4 < 0$$

$$\because d_2(X) > 0, \quad d_1(X), d_3(X) < 0 \quad \therefore \mathbf{X} \in \omega_2$$

三个判别边界(界面)分别为：

$$-x_1 + x_2 + 1 = 0 \qquad x_1 + x_2 - 4 = 0 \qquad -x_2 + 1 = 0$$

图示如下：



步骤:

- 画出边界直线。
- 判别边界正负侧：找特殊点代入；或者：画向量 $W_0$  (以原点为起点)，箭头指向的方向就是 $d(X)$ 的正侧
- 找交集 (一正侧、多则负侧)。

## ➤ 多类情况2 ( $\omega_i/\omega_j$ 成对两分法)

➤ 每两个模式类间可用判别平面分开(即模式类成对可分)。

❖ 这样有  $C_M^2 = M(M-1)/2$  个判别平面。

❖ 对于两类问题,  $M=2$ , 则有一个判别平面。

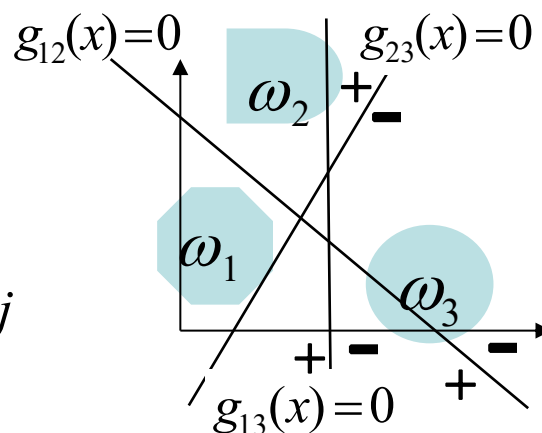
❖ 同理, 三类问题则有三个判别平面。

❖ 判别函数:  $g_{ij}(x) = W_{ij}^T X$

❖ 判别边界:  $g_{ij}(x) = 0$

❖ 判别条件:  $g_{ij}(x) \begin{cases} > 0 \rightarrow \text{当 } X \in \omega_i \\ < 0 \rightarrow \text{当 } X \in \omega_j \end{cases} i \neq j$

❖ 判别函数性质:  $g_{ij}(x) = -g_{ji}(x)$



## ➤ 多类情况2 (Cont.)

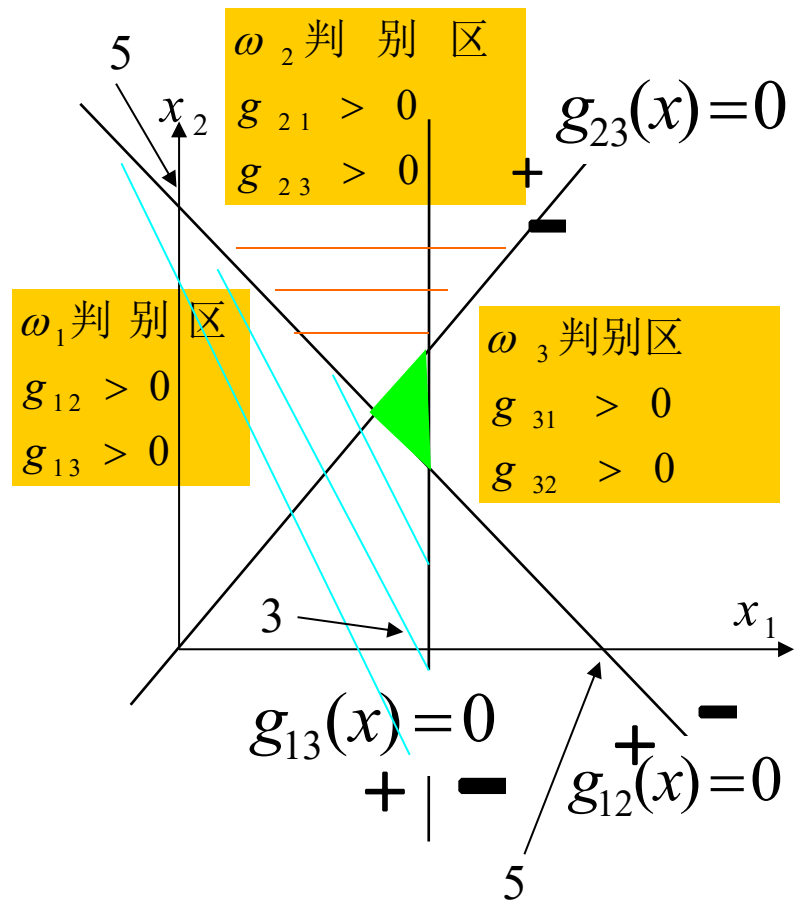
❖ 假设判别函数为:

$$\begin{cases} g_{12}(x) = -x_1 - x_2 + 5 \\ g_{13}(x) = -x_1 + 3 \\ g_{23}(x) = -x_1 + x_2 \end{cases}$$

❖ 判别边界为:

$$\begin{cases} g_{12}(x) = -x_1 - x_2 + 5 = 0 \\ g_{13}(x) = -x_1 + 3 = 0 \\ g_{23}(x) = -x_1 + x_2 = 0 \end{cases}$$

❖ 用上面方程式作出右图:



## ➤ 多类情况2 (Cont.)

❖ 结论：判别区间增大，不确定区间减小，比第一种情况小得多。

❖ 问：未知模式  $X = (x_1, x_2)^T = (4, 3)^T$  属于哪一类？

代入判别函数可得：

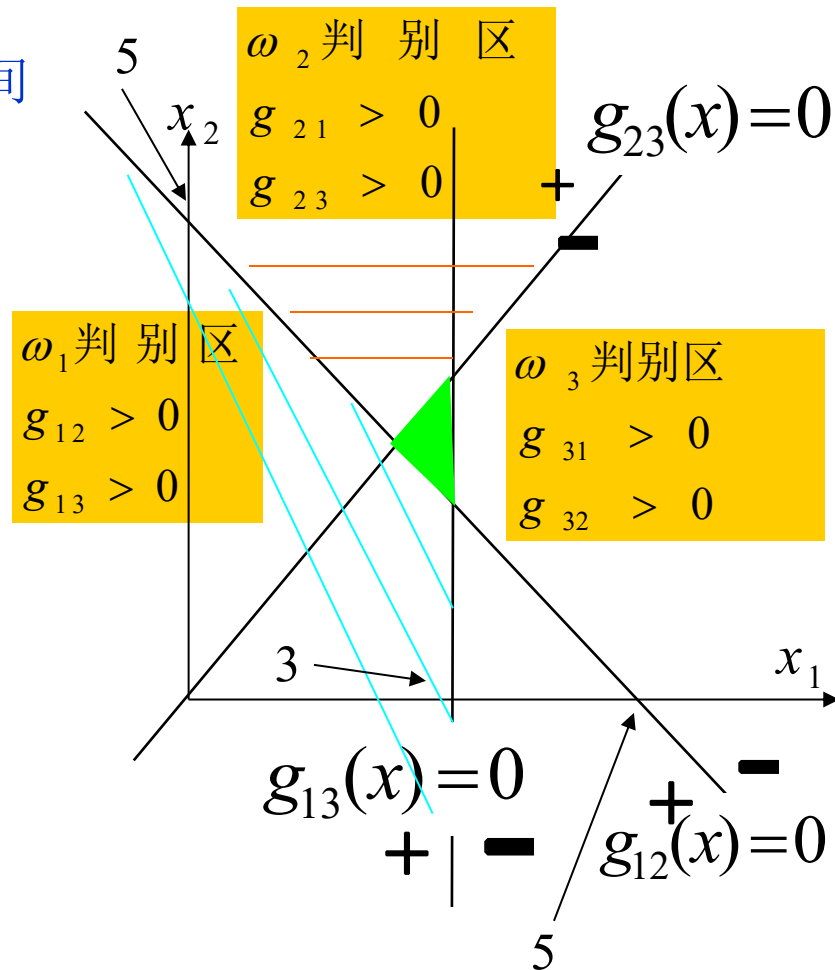
$$g_{12}(x) = -2, g_{13}(x) = -1, g_{23}(x) = -1$$

把下标对换取负，可得：

$$g_{21}(x) = 2, g_{31}(x) = 1, g_{32}(x) = 1$$

因为  $g_{3j}(x) > 0$

所以  $X$  属于  $\omega_3$  类



## ➤ 多类情况3 [ $\omega_i/\omega_j$ 成对两分法 (无IR区)]

❖ 每类都有一个判别函数, 存在  $M$  个判别函数

❖ 判别函数:  $g_i(x) = W_i^T X \quad i=1, 2, \dots, M$



❖ 判别规则:  $g_i(x) = W_i^T X \begin{cases} \text{最大, 当 } X \in \omega_i \\ \text{小, 其他} \end{cases}$

❖ 判别边界:  $g_i(x) = g_j(x)$  或  $g_i(x) - g_j(x) = 0$

❖ 就是说, 要判别模式  $X$  属于哪一类, 先把  $X$  代入  $M$  个判别函数中, 判别函数最大的那个类别就是  $X$  所属类别。类与类之间的边界可由  $g_i(x) = g_j(x)$  或  $g_i(x) - g_j(x) = 0$  来确定。

❖ 与多类情况1的区别: 多类情况3可有多于一个判别函数值大于0 (而多类情况1只有一个判别函数的值大于0)

## ➤ 多类情况3 (Cont.)

❖ 下图所示是 $M=3$  的例子。

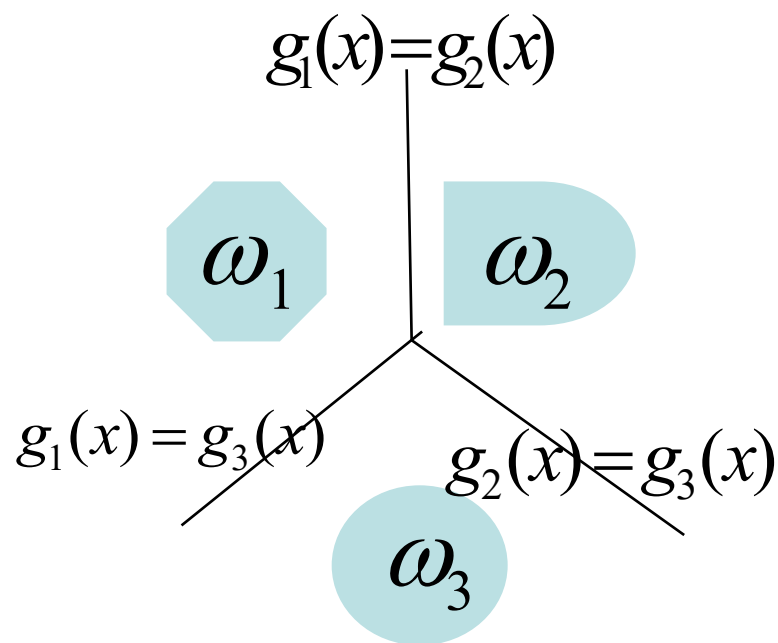
对于 $\omega_1$ 类模式，必然满足 $g_1(x) > g_2(x)$  和  $g_1(x) > g_3(x)$  。

假设判别函数为：

$$\begin{cases} g_1(x) = -x_1 + x_2 \\ g_2(x) = x_1 + x_2 - 1 \\ g_3(x) = -x_2 \end{cases}$$

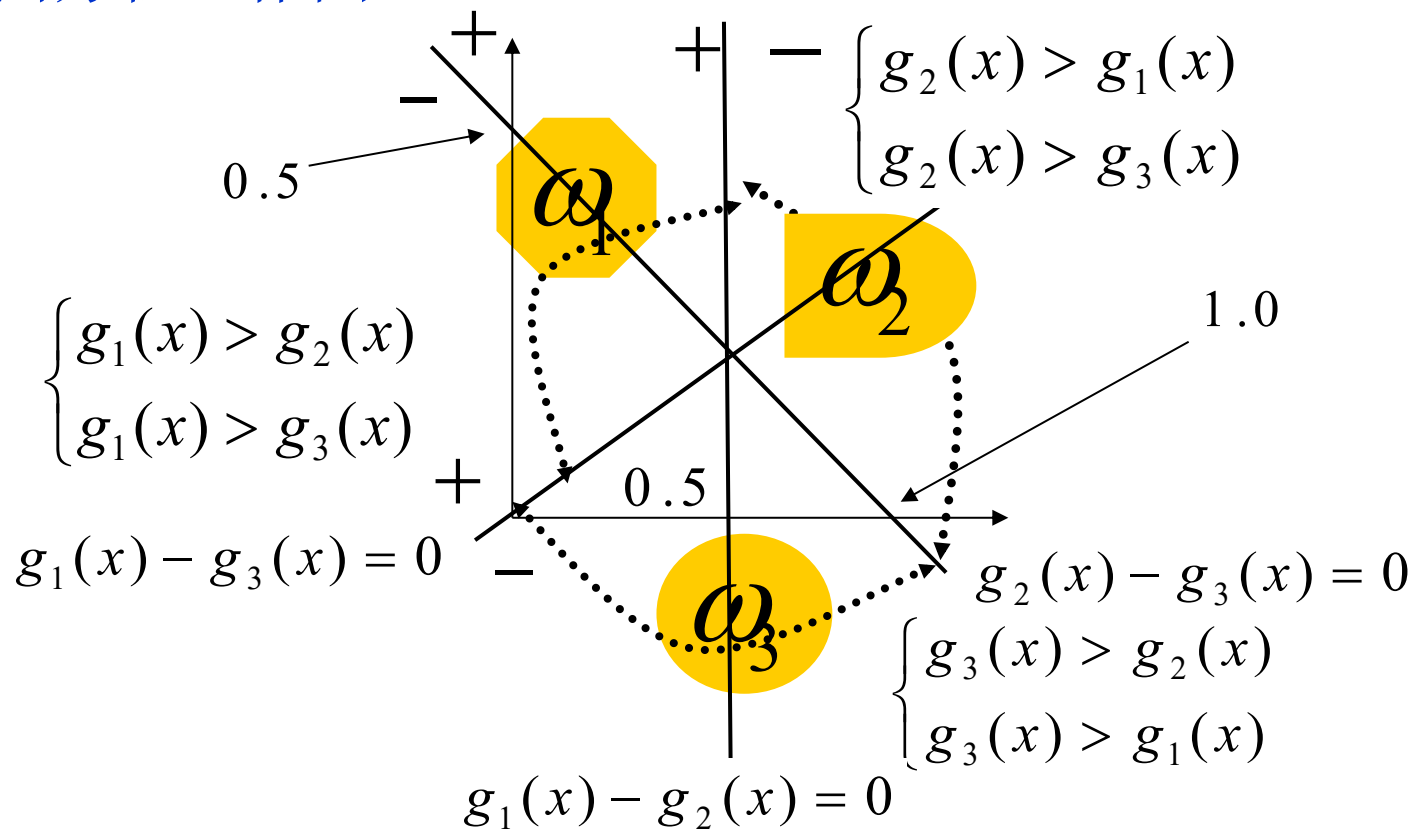
则判别边界为：

$$\begin{cases} g_1(x) - g_2(x) = -2x_1 + 1 = 0 \\ g_1(x) - g_3(x) = -x_1 + 2x_2 = 0 \\ g_2(x) - g_3(x) = x_1 + 2x_2 - 1 = 0 \end{cases}$$



## ➤ 多类情况3 (Cont.)

❖ 用上面方程组作图：



❖ 结论：  $IR$  不确定区没有了，所以这种是最好的情况。



## ➤ 多类情况3 (Cont.)

- ❖ 问：假设未知模式  $\mathbf{x} = (x_1, x_2)^T = (1, 1)^T$ ，则  $\mathbf{x}$  属于哪一类。  
把  $\mathbf{x}$  代入判别函数：  $g_1(x)$ ,  $g_2(x)$ ,  $g_3(x)$ .  
得判别函数为：  $g_1(x) = 0, g_2(x) = 1, g_3(x) = -1$   
因为  $g_2(x) > g_3(x), g_2(x) > g_1(x)$   
所以模式  $\mathbf{x} = (1, 1)^T$  属于  $\omega_2$  类。

## 3.3 线性判别函数的性质

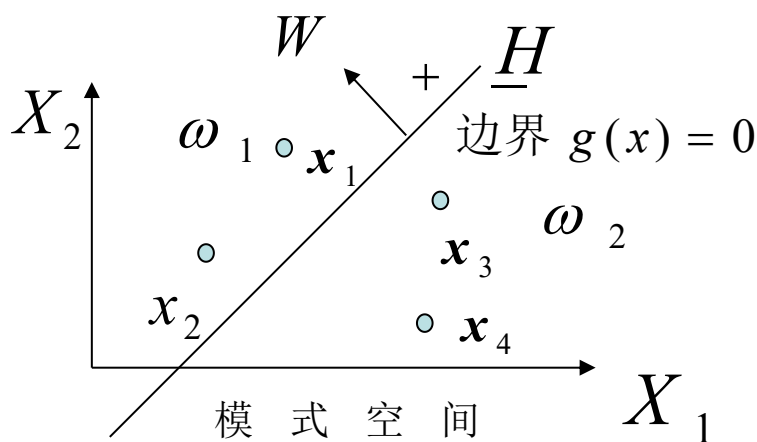
➤ 1. 模式空间与加权空间  $g(x) = W_0^T X + w_{n+1} = W^T X$

. 模式空间：由  $X = (x_1, x_2, x_3, \dots, x_n)^T$  构成的  $n$  维欧氏空间。

.  $W$  是此空间的加权向量，它决定模式的分界面  $H$ ， $W$  与  $H$  正交。

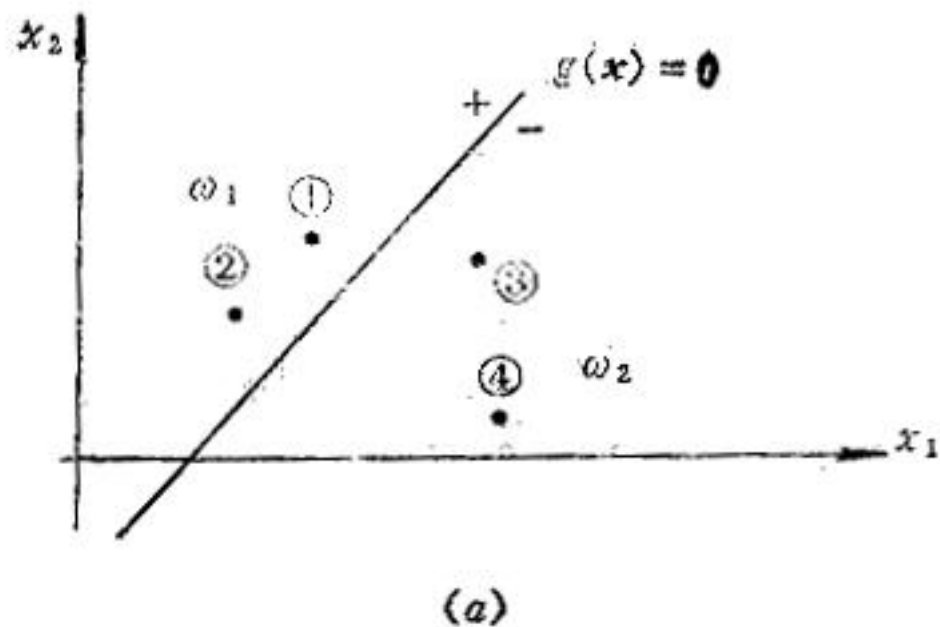
. 加权空间：以  $w_1, w_2, \dots, w_{n+1}$  为变量构成的  $n+1$  维欧氏空间。

. 模式空间与加权空间的几何表示如下图所示：



由于假设权向量  $W$  与模式向量  $X$  的内积为零 ( $g(x)=0$ )，故  $W$  与分界面  $H$  正交

# 模式空间



## ➤ 1. 模式空间与加权空间(Cont.)

❖ 加权空间的构造:  $g(x) = w_1x_1 + w_2x_2 + w_3$

❖ 设  $\mathbf{x}_1 = (x_{11}, x_{12})^T$  是加权空间分界面上的一点, 代入上式得:

$g(\mathbf{x}_1) = w_1x_{11} + w_2x_{12} + w_3 = 0$ , 这是加权空间的边界。

❖ 该式表示一个通过加权空间原点的平面, 此平面就是加权空间图中的平面①, 同样令  $g(\mathbf{x}_2) = g(\mathbf{x}_3) = g(\mathbf{x}_4) = 0$ , 分别作出通过加权空间原点的平面②③④; 图中用阴影表示的部分是各平面的正侧。

设:  $\mathbf{x}_1, \mathbf{x}_2 \in \omega_1$

$\mathbf{x}_3, \mathbf{x}_4 \in \omega_2$

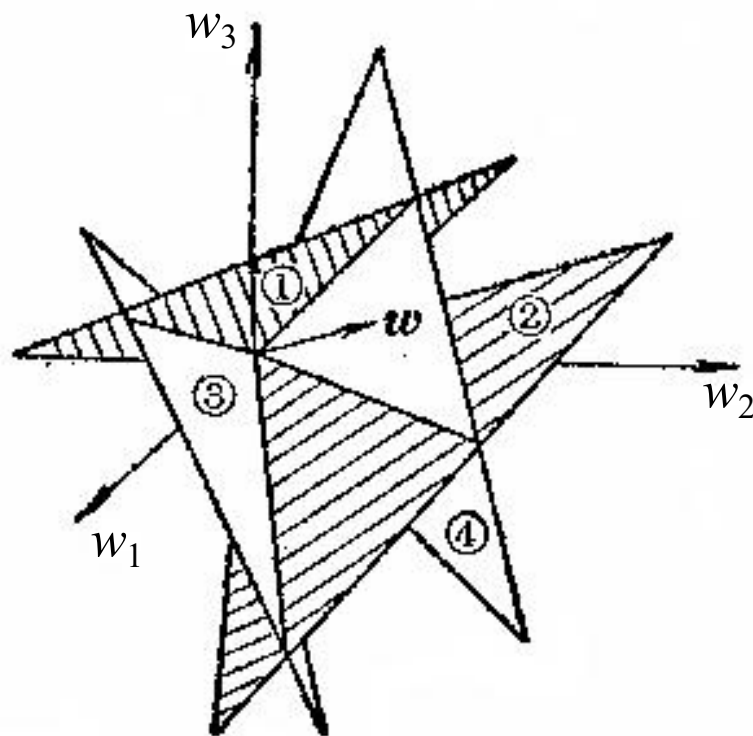
$$Q \quad g(\mathbf{x}) \begin{cases} > 0 & \mathbf{x} \in \omega_1 \\ < 0 & \mathbf{x} \in \omega_2 \end{cases}$$

最终形成凸多面锥

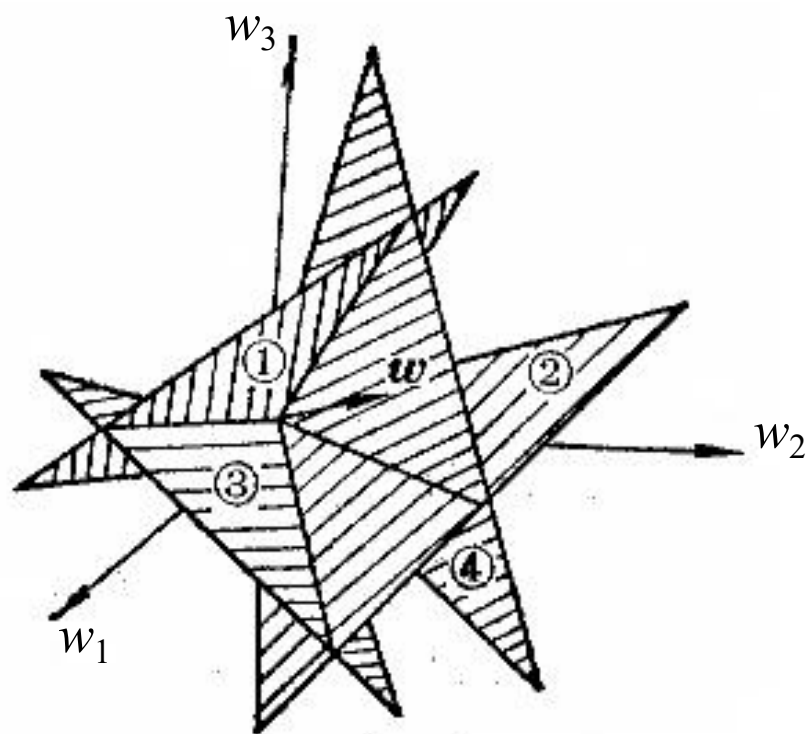
$$\left. \begin{aligned} w_1x_{11} + w_2x_{12} + w_3 &> 0 \\ w_1x_{21} + w_2x_{22} + w_3 &> 0 \end{aligned} \right\} \omega_1$$

$$\left. \begin{aligned} w_1x_{31} + w_2x_{32} + w_3 &< 0 \\ w_1x_{41} + w_2x_{42} + w_3 &< 0 \end{aligned} \right\} \omega_2$$

# 加权空间判别界面



## 正规化后的加权空间判别界面

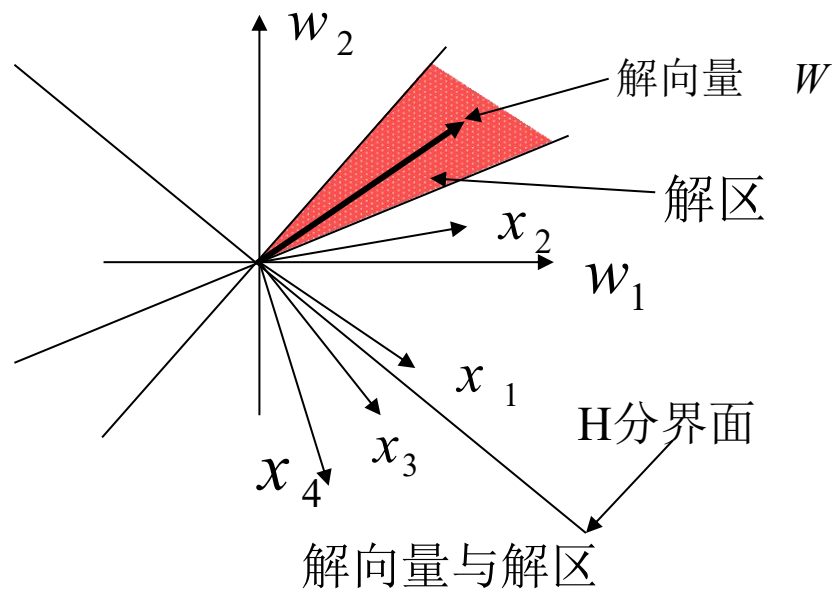


## ➤ 1. 模式空间与加权空间(Cont.)

- ❖ 这是一个不等式方程组，它的解  $W = (w_1, w_2, w_3)^T$  处于由  $\omega_1$  类所有模式决定的平面的正边和由  $\omega_2$  类所有模式决定的平面的负边，它的解区即为凸多面锥(convex polyhedra)。
- ❖ 如上图所示：(b)为加权空间，(c)为正规化后的加权空间。
- ❖ 由上可以得出结论：加权空间的所有分界面都通过坐标原点。这是加权空间的性质。
- ❖ 为了更清楚些，下面用二维权空间来表示解向量和解区。

## ➤ 2. 解向量和解区

- ❖ 在三维空间里，令 $w_3 = 0$  则为二维权空间。如右图：
- ❖ 给定一个模式 $X$ ，就决定一条直线： $g(x) = W^T X = 0$
- ❖ 即分界面 $H$ ， $W$ 与 $H$ 正交， $W$ 称为解向量。
- ❖ 解向量的变动范围称为解区。
- ❖ 因 $x_1, x_2 \in \omega_1$ ,  $x_3, x_4 \in \omega_2$ ，由图可见 $x_1, x_3$ 离的最近，所以分界面 $H$ 可以是 $x_1, x_3$ 之间的任一直线，由垂直于这些直线的 $W$ 就构成解区，解区为一扇形平面，即红色阴影区域。如右图所示。





## ➤ 2.解向量和解区(Cont.)

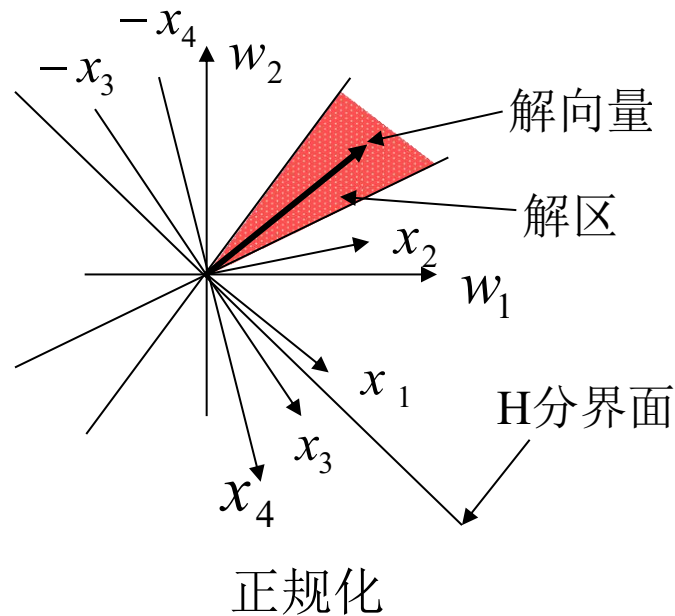
❖ 将不等式方程正规化:

$$\begin{cases} w_1 x_{11} + w_2 x_{12} + w_3 > 0 \\ w_1 x_{21} + w_2 x_{22} + w_3 > 0 \\ -w_1 x_{31} - w_2 x_{32} - w_3 > 0 \\ -w_1 x_{41} - w_2 x_{42} - w_3 > 0 \end{cases}$$

❖ 正规化(regularization):

$$g(x) = W^T X > 0$$

$$W = (w_1, w_2, \dots, w_n, w_{n+1})$$



### ➤ 3. 超平面的几何性质

❖  $g(x) = W_0^T X + w_{n+1} = 0$  决定一个决策界面，当 $g(x)$ 为线性时，这个决策界面便是一个超平面  $H$  (Hyperplane)，并有如下性质：

❖ 性质①：  $W_0$  与  $H$  正交（见右图）

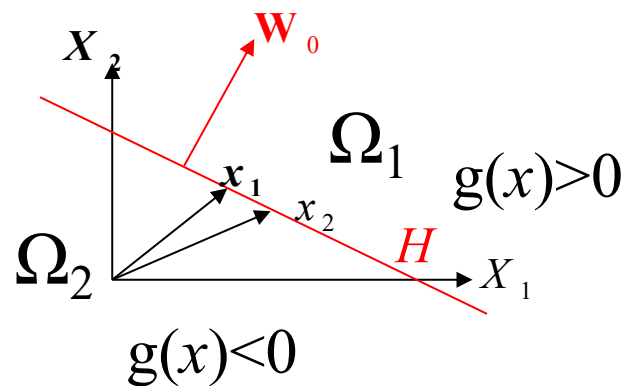
· 假设  $x_1, x_2$  是  $H$  上的两个向量

$$W_0^T \mathbf{x}_1 + w_{n+1} = W_0^T \mathbf{x}_2 + w_{n+1} = 0$$

所以  $W_0^T (\mathbf{x}_1 - \mathbf{x}_2) = 0$ ,  $(\mathbf{x}_1 - \mathbf{x}_2)$  向量一定在  $H$  上

·  $W_0$  与  $(x_1 - x_2)$  垂直，即  $W_0$  与  $H$  正交。

· 一般说，超平面  $H$  把特征空间分成两个半空间。即  $\Omega_1, \Omega_2$  空间，当  $x$  在  $\Omega_1$  空间时  $g(x) > 0$ ,  $W_0$  指向  $\Omega_1$ ，为  $H$  的正侧，反之为  $H$  的负侧。



### ➤ 3.超平面的几何性质(Cont.)

❖ 性质 ②:  $\|\mathbf{r}\| = \frac{g(x)}{\|\mathbf{W}_0\|}$

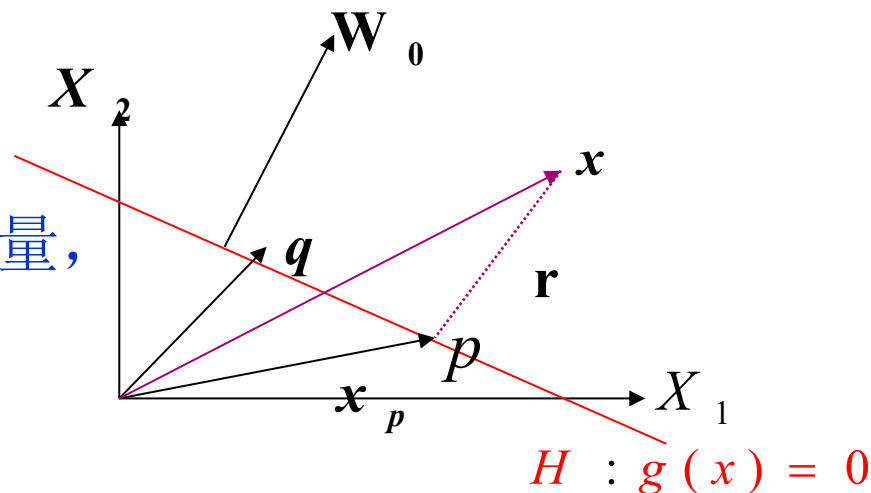
❖ 向量 $x$ 到超平面 $H$ 的正交投影  $\|\mathbf{r}\|$  正比于  $g(x)$  的函数值

$$x = x_p + \mathbf{r} = x_p + \|\mathbf{r}\| \frac{\mathbf{W}_0}{\|\mathbf{W}_0\|}$$

. 其中:  $x_p$  为  $x$  在  $H$  的投影向量,

.  $r = \|\mathbf{r}\|$  是  $x$  到  $H$  的垂直距离。

.  $\frac{\mathbf{W}_0}{\|\mathbf{W}_0\|}$  是  $\mathbf{W}_0$  方向的单位向量。



$\|\mathbf{W}_0\|$  在这里理解为权向量  $\mathbf{W}_0$  的模, 由下式计算:

$$\|\mathbf{W}_0\| = \sqrt{w_1^2 + w_2^2 + \dots w_n^2}$$

### ➤ 3.超平面的几何性质(Cont.)

❖ 证明:

$$g(x) = W_0^T x + w_{n+1} = W_0^T (x_p + \mathbf{r}) + w_{n+1}$$

$$\text{Q } p \text{ 在 } H \text{ 上, } \therefore W_0^T x_p + w_{n+1} = 0$$

$$\therefore g(x) = W_0^T \mathbf{r} = W_0^T \left( \|\mathbf{r}\| \frac{W_0}{\|W_0\|} \right) = \|\mathbf{r}\| \frac{W_0^T W_0}{\|W_0\|} = \|\mathbf{r}\| \|W_0\|$$

$$\therefore \|\mathbf{r}\| = \frac{g(x)}{\|W_0\|}, \|\mathbf{r}\| \text{ 是投影的绝对值 } (W_0^T W_0 = \|W_0\|^2)$$

❖ 这是超平面的第二个性质，向量 $\mathbf{x}$ 到超平面 $H$ 的正交投影正比于 $g(x)$ 的函数值。

### ➤ 3.超平面的几何性质(Cont.)

❖ 性质③:

$\|q\| = \frac{w_{n+1}}{\|W_0\|}$ , 原点 到超平面  $H$  的距离与  $w_{n+1}$  成正比

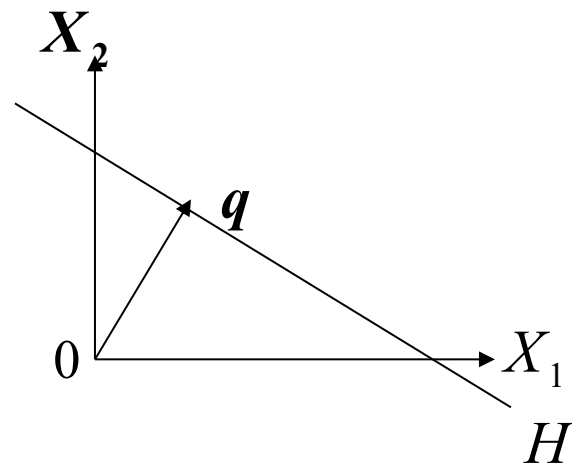
❖ 证明:

$$Q \quad g(x) = W_0^T x + w_{n+1} = w_{n+1} \text{ (因原点 } x = 0 \text{)}$$

$$\therefore \|r\| = \frac{g(x)}{\|W_0\|} = \frac{w_{n+1}}{\|W_0\|} = \|q\|$$

$$Q \quad x = 0 \text{ 时 } x \text{ 到 } H \text{ 的投影为 } \|r\| = \|q\|$$

$$\therefore \|q\| = \frac{w_{n+1}}{\|W_0\|}$$



## ➤ 3.超平面的几何性质(Cont.)

### ❖ 性质④:

若 $w_{n+1} > 0$ ,则 $H$ 在 origin 正侧, 若 $w_{n+1} < 0$ ,则 $H$ 在 origin 负侧。

若 $w_{n+1} = 0$ ,则 $g(x) = W_0^T x$ , 说明超平面 $H$ 通过 origin。

由以上4个性质, 在模式空间中超平面具有如下结论:

(a)超平面 $H$ 的平面与 $W_0$ 正交, 方向由 $W_0$ 决定。

(b)超平面 $H$ 的位置由阈值权 $w_{n+1}$ 决定。

(c)判别函数 $g(x)$ 正比于点 $x$ 到超平面 $H$ 的代数距离。

(d) $x$ 在超平面 $H$ 的正侧时,  $g(x) > 0$ ;  $x$ 在超平面 $H$ 的负侧时,  $g(x) < 0$ 。

## 2.4 广义线性判别函数(\*: 了解)

❖ 判别函数的一般形式:

$$g(x) = w_1 f_1(x) + w_2 f_2(x) + \dots + w_k f_k(x) + w_{k+1}$$
$$= \sum_{i=1}^{k+1} w_i f_i(x), i = 1, 2, \dots, k \quad \text{式中 } f_i(x) \text{ 是单值函数, } f_{k+1}(x) = 1$$

❖ 这样一个非线性判别函数通过映射，变换成线性判别函数。

$$g(x) = \sum_{i=1}^{k+1} w_i f_i(x) \xrightarrow{x \text{ 空间} \rightarrow \text{变换} \rightarrow y \text{ 空间}}$$

$$W^T Y = g(Y) \begin{cases} > 0, x \in \omega_1 \\ < 0, x \in \omega_2 \end{cases}$$

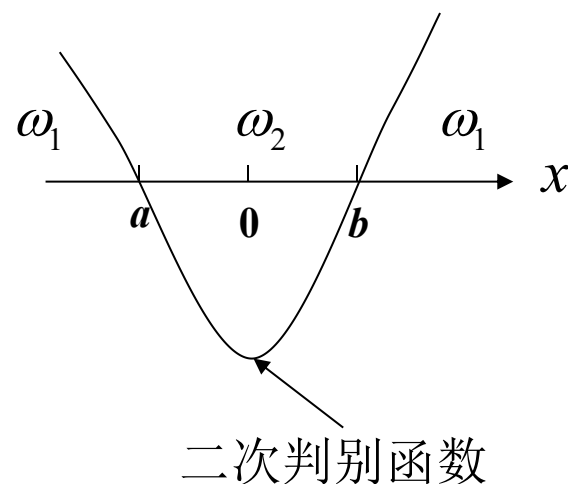
$$g(x) = \sum_{i=1}^{k+1} w_i f_i(x) \xrightarrow{x\text{空间} \rightarrow \text{变换} \rightarrow y\text{空间}} W^T Y = g(Y) \begin{cases} > 0, x \in \omega_1 \\ < 0, x \in \omega_2 \end{cases}$$

其中:  $W = \begin{bmatrix} w_1 \\ w_2 \\ \dots \\ w_k \\ w_{k+1} \end{bmatrix}$  (增广权向量)。  $Y = \begin{bmatrix} f_1(x) \\ f_2(x) \\ \dots \\ f_k(x) \\ 1 \end{bmatrix}$  (增广模式向量)

❖ 例: 见右图。

$a < x < b$ , 则  $x \in \omega_2$

$x > b$  or  $x < a$ , 则  $x \in \omega_1$





❖ 要用二次判别函数(Quadratic Discriminant Function)才能将二类分开:

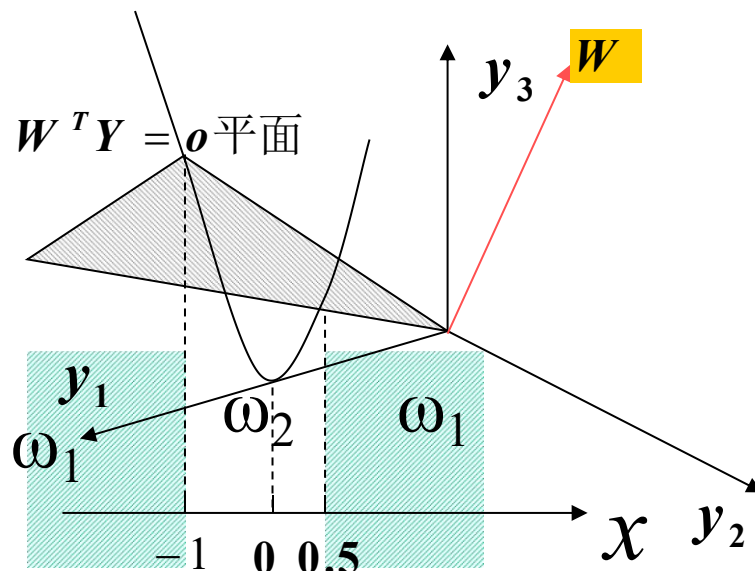
$$g(x) = (x-a)(x-b)$$

$$= a_1 + a_2 x + a_3 x^2 \begin{cases} > 0, x \in \omega_1 \\ < 0, x \in \omega_2 \end{cases}$$

将 $x$ 变换到另外一个特征空间 $Y$ ,  
即选择 $x$ 到 $Y$ 的一个有效映射, 总能将  
上式变成关于 $Y$ 的线性判别函数 (这种  
变换很重要)

$$\text{映射: } g(x) = W^T Y = g(Y) \begin{cases} > 0, x \in \omega_1 \\ < 0, x \in \omega_2 \end{cases}$$

$$W = \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix}, Y = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} 1 \\ x \\ x^2 \end{bmatrix}$$



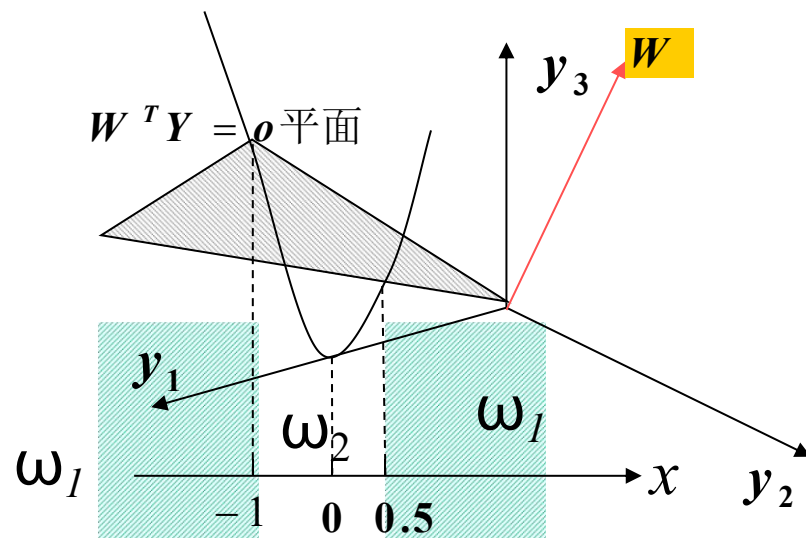
$$y_1 = (1, 0, 0)^T, \quad y_2 = (1, 0.5, 0.25)^T, \\ y_3 = (1, -1, 1)^T$$

讨论在 $x$ 空间它的判别边界：设 $a_1 = -1, a_2 = 1, a_3 = 2$

$$\text{即： } W = \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} -1 \\ 1 \\ 2 \end{bmatrix}, Y = \begin{bmatrix} 1 \\ x \\ x^2 \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix}$$

$$g(x) = 2x^2 + x - 1 = 0 \xrightarrow{\text{推出}} \begin{cases} x_1 = 0.5 \\ x_2 = -1 \end{cases}$$

$Y$ 空间判别平面： $W^T Y = 0$



$$y_1 = (1, 0, 0)^T, \quad y_2 = (1, 0.5, 0.25)^T, \\ y_3 = (1, -1, 1)^T$$

❖ 从图可以看出：在阴影上面是 $\omega_1$ 类，在阴影下面是 $\omega_2$ 类。

- 下面解释这种变化：对于任意高次的判别函数 $g(x)$ ，当然它也能看成是对任意判别函数的泰勒级数展开，然后取其截尾后的逼近，对于任意这样的 $g(x)$ ，总能通过合适、有效的变换，化作上述类似的线性判别函数来处理（需要注意的是，变换后已不再是 $x$ 的线性函数，而变成了 $y$ 的线性函数），因此变换后的线性判别函数也被叫做**广义线性判别函数**。如此变换后，我们依然可以用线性判别函数来解决复杂问题，从而达到简化的目的。

然而，天下没有免费的午餐，这种简单是有代价的，那就是导致变换后的特征空间维数大大增加，从而陷入维数灾难(the curse of dimensionality)，而且判别函数的参数数目也将很大。

## 2.4 线性分类器设计

前面我们讨论了线性判别函数形式： $g(\mathbf{x})=\mathbf{w}^T\mathbf{x}$

其中 $\mathbf{x}=(x_1, x_2, \dots, x_n, 1)^T$   $n$ 维特征向量(增广特征向量)

$\mathbf{w}=(w_1, w_2, \dots, w_n, w_{n+1})^T$   $n+1$ 维权向量(增广权向量)

$$\text{分类准则} \begin{cases} g(\mathbf{x}) > 0, \mathbf{x} \in \omega_1 \\ g(\mathbf{x}) < 0, \mathbf{x} \in \omega_2 \end{cases}$$

通常通过特征提取可以获得模式的 $n$ 维特征向量。对于线性判别函数，当模式的维数已知时，判别函数的形式实际上就已确定，剩下的问题就是确定权向量 $\mathbf{w}$ ，只要求出权向量，分类器(Classifier)设计即告完成。求解权向量的过程就是分类器的训练过程，使用已知类别的有限学习样本来获得分类器的权向量 $\mathbf{w}$ 被称为有监督分类/学习(Supervised classification/learning)。

利用方程组求解权向量:

对二类判别函数  $g(x) = W_1 X_1 + W_2 X_2 + W_3$

已知训练集:  $X_a, X_b, X_c, X_d$  且

当  $(X_a, X_b) \in \omega_1$  时  $g(x) > 0$

当  $(X_c, X_d) \in \omega_2$  时  $g(x) < 0$

设  $X_a = (X_{1a}, X_{2a})^T$   $X_b = (X_{1b}, X_{2b})^T$

$X_c = (X_{1c}, X_{2c})^T$   $X_d = (X_{1d}, X_{2d})^T$

判别函数可联立成:

$$X_{1a} W_1 + X_{2a} W_2 + W_3 > 0 \quad (1)$$

$$X_{1b} W_1 + X_{2b} W_2 + W_3 > 0 \quad (2)$$

$$X_{1c} W_1 + X_{2c} W_2 + W_3 < 0 \quad (3)$$

$$X_{1d} W_1 + X_{2d} W_2 + W_3 < 0 \quad (4)$$

求出  $W_1, W_2, W_3$

将③ ④式正规化(normalization), 得:

$$-X_{1c}W_1 - X_{2c}W_2 - W_3 > 0$$

$$-X_{1d}W_1 - X_{2d}W_2 - W_3 > 0$$

因此 $g(x) = XW > 0$ , 其中 $W = (W_1, W_2, W_3)^T$

$$X = \begin{bmatrix} X_{1a} & X_{2a} & 1 \\ X_{1b} & X_{2b} & 1 \\ -X_{1c} & -X_{2c} & -1 \\ -X_{1d} & -X_{2d} & -1 \end{bmatrix} \text{ 为各模式增广矩阵}$$

为 $N \times (n+1)$ 矩阵

$N$ 为样本数,  $n$ 为特征数(此处 $N=4, n=2$ )

训练过程就是对已知类别的样本集求解权向量 $\mathbf{W}$ ，这是一个线性联立不等式方程组求解的过程。

求解时：

- ①只有对线性可分的问题， $g(\mathbf{x}) = \mathbf{W}^T \mathbf{x}$ 才有解；
- ②联立方程的解是非单值，在不同条件下，有不同的解，所以就产生了求最优解问题；
- ③求解 $\mathbf{W}$ 的过程就是训练的过程。训练/学习方法共同点是，先给出准则函数(Criterion function)，再寻找使准则函数趋于极值的优化算法(Optimization algorithm)，不同的算法有不同的准则函数。算法可以分为迭代法和非迭代法(Non iterative method)。

## ➤ 2.4.1 梯度(下降)法—迭代法

要对不等式方程  $\mathbf{W}^T \mathbf{x} > 0$  求解，首先定义准则函数(目标函数)  $J(\mathbf{W})$ ，再求  $J(\mathbf{W})$  的极值使  $\mathbf{W}$  优化。因此求解权向量的问题就转化为对一目标函数求极值的问题。解决此类问题的方法是梯度下降法GD(Gradient Descent method, 1847年由法国数学家Cauchy首次提出)。

方法就是从初始值  $\mathbf{W}_1$  开始，算出  $\mathbf{W}_1$  处目标函数的梯度向量  $\nabla J(\mathbf{W}_1)$ ，则下一步的  $\mathbf{W}$  值为：

$$\mathbf{W}_2 = \mathbf{W}_1 - \rho_1 \nabla J(\mathbf{W}_1)$$

$\mathbf{W}_1$  为初始权向量                       $\rho_1$  为迭代步长

$J(\mathbf{W}_1)$  为目标函数

$\nabla J(\mathbf{W}_1)$  为  $\mathbf{W}_1$  处的目标函数的梯度向量



## 记号：多元函数的偏导——梯度(Gradient vector)

$$J(W, X) = w_1 x_1 + w_2 x_2 + \dots + w_n x_n + w_{n+1} = W^T X$$

$J$ 对 $W$ 的梯度：

$$\nabla J(W) = \frac{\partial J(W, X)}{\partial W} = \left[ \frac{\partial J}{\partial w_i} \right]_{(n+1) \times 1} = \begin{bmatrix} \frac{\partial J}{\partial w_1} \\ \vdots \\ \frac{\partial J}{\partial w_n} \\ \frac{\partial J}{\partial w_{n+1}} \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \\ 1 \end{bmatrix} = X$$

要特别说明的是，这里的 $J(W, X)$ 用线性函数表达式只是用来说明梯度的概念，它并非梯度下降法所选则的准则函数。

在梯度法对 $W$ 权向量的优化中， $\nabla J$ 的方向是 $W$ 增加时 $J$ 增长最快的方向，因此 $(-\nabla J)$ 的方向是 $W$ 增加时 $J$ 减小最快的方向。梯度法就是用这个负梯度向量的值对权向量 $W$ 进行修正，实现准则函数达到极小值的目的。

在第 $k$ 步的时候

$$W_{k+1} = W_k - \rho_k \nabla J(W_k) \quad \rho_k \text{ 为正比例因子/学习速率 (learning rate)}$$

这就是梯度下降法的迭代公式。这样一步步迭代就可以收敛于解向量。

$\rho_k$ 取值很重要：

$\rho_k$ 太大，迭代太快，引起振荡，甚至发散。

$\rho_k$ 太小，迭代太慢。

应该选最佳 $\rho_k$ 。

(\*：了解)

选最佳 $\rho/\rho_k$

目标函数 $J(W)$ 在 $W_k$ 附近二阶Taylor级数展开式为：

$$J(W) \approx J(W_k) + \nabla J^T(W - W_k) + 1/2(W - W_k)^T H^T (W - W_k) \quad (1)$$

其中 $H$ 为当 $W = W_k$ 时 $J(W)$ 的Hessian矩阵(二阶偏导数矩阵)

将 $W = W_{k+1} = W_k - \rho_k \nabla J(W_k)$ 代入(1)式得：

$$J(W_{k+1}) \approx J(W_k) - \rho_k \|\nabla J\|^2 + 1/2 \rho_k^2 \nabla J^T H^T \nabla J \quad (2)$$

其中 $\nabla J = \nabla J(W_k)$

对 $\rho_k$ 求导数，并令导数为零，有最佳步长为：

$$\rho_k = \|\nabla J\|^2 / (\nabla J^T H^T \nabla J)$$

这就是最佳 $\rho_k$ 的计算公式，但因二阶偏导数矩阵 $H$ 的计算量太大，因此此公式较少使用。

(\*: 了解)

若令  $W=W_{k+1}$ , (1)式为

$$J(W_{k+1})=J(W_k)+\nabla J^T(W_{k+1}-W_k)+1/2(W_{k+1}-W_k)^TH^T(W_{k+1}-W_k)$$

对  $W_{k+1}$  求导, 并令导数为零, 可得

最佳迭代公式:  $W_{k+1}=W_k-H^{-1}\nabla J$  —\*牛顿法的迭代公式

$H^{-1}$ 是 $H$ 的逆阵

讨论: 牛顿法是基于二阶Taylor级数且比梯度下降法收敛更快, 但是 $H$ 的计算量大并且要计算 $H^{-1}$ 。当 $H$ 为奇异矩阵时, 无法用牛顿法。

梯度下降算法在机器学习及神经网络中有着广泛应用，它主要用来求解最优参数。

下面通过一个简单示例来说明梯度下降法迭代计算过程。

**举例：**  $y = f(x) = x^2$ ，用梯度下降法求  $y$  取最小值(极小值)时的最优解  $x$ 。

**迭代方程：**  $x_{k+1} = x_k - \rho * f'(x_k)$  ( $k=0, 1, 2, \dots, n$ )

其中  $x_{k+1}$  为要求的解， $f'(x_k)$  为梯度(一元函数为导数或多元函数为偏导，这里用最简单的一元函数展示，故直接写成导数)

$\rho$  为学习速率 (或称步长，是一个重要的参数， $\rho$  的选择直接影响算法的效率)

**迭代求解过程：**

(1) 首先任取一初始点，如取  $x_0 = 3$ ，计算导数  $f'(x_0) = 6$

(2) 设定学习速率为  $\rho = 0.43$

(3) 开始算法迭代：

1)  $x_0=3$ ,  $f'(x_0)=6$ ,  $x_1=x_0-\rho*f'(x_0)=0.6$

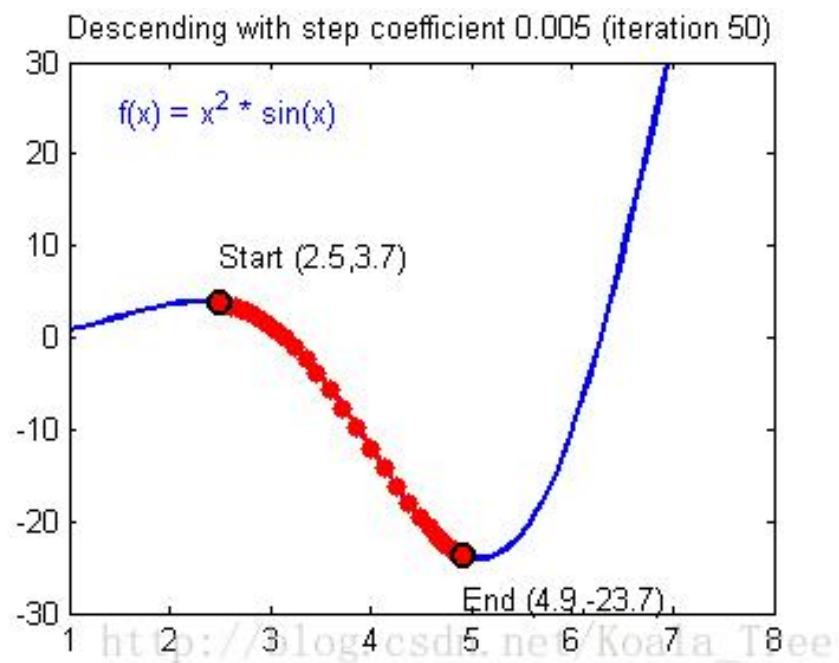
2)  $x_1=0.6$ ,  $f'(x_1)=1.2$ ,  $x_2=x_1-\rho*f'(x_1)=0.12$

3)  $x_2=0.12$ ,  $f'(x_2)=0.24$ ,  $x_3=x_2-\rho*f'(x_2)=0.024$

4)  $x_3=0.024$ ,  $f'(x_3)=0.048$ ,  $x_4=x_3-\rho*f'(x_3)=0.0048$

.....

(4) 当梯度  $f'(x_k)$  (本例为导数) 下降到很小或为0时，则求得的解  $x_{k+1}$  趋近于最优解，本示例迭代到第4)步时  $f'(x_3) = 0.048$  已经很小， $x_4=0.0048$  基本趋向于本例的真正(解析)解  $x=0$ 。



梯度下降法演示动画

# ●线性判别函数的梯度法求解

## ➤梯度求解算法的基本原理

### (1) 梯度概念

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

$$\nabla f(\mathbf{X}) = \frac{d}{d\mathbf{X}} f(\mathbf{X}) = \left[ \frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, \dots, \frac{\partial f}{\partial x_n} \right]^T$$

**梯度向量 最重要的性质之一：**

**当函数 $f$ 的自变量沿梯度变化时，函数的增长最快**  
即：

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

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

**显然：**负梯度指出了最陡下降方向——梯度算法的依据。

## (2) 线性判别函数的梯度算法

设两个线性可分的模式类 $\omega_1$ 和 $\omega_2$ 的样本共 $N$ 个， $\omega_2$ 类样本乘(-1)。将两类样本分开的判决函数 $g(\mathbf{X})$ 应满足：

$$g(\mathbf{X}_i) = \mathbf{W}^T \mathbf{X}_i > 0 \quad i = 1, 2, \dots, N \quad \text{——} N \text{个不等式}$$

梯度算法的目的仍然是求一个满足上述条件的权向量 $\mathbf{W}$ ，主导思想是将联立不等式求解 $\mathbf{W}$ 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 $\mathbf{W}$ 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 $\mathbf{W}^T \mathbf{X}_i > 0$ 时。



## 基本思路:

定义一个对错误分类敏感的准则函数 $J(\mathbf{W}, \mathbf{X})$ ，在 $J$ 的梯度方向上对权向量进行更新，从 $\mathbf{W}(k)$ 计算 $\mathbf{W}(k+1)$ 的递推关系为：

$$\mathbf{W}(k+1) = \mathbf{W}(k) + \rho(-\nabla J) = \mathbf{W}(k) - \rho \nabla J$$

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

其中 $\rho$ 是固定的、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 $\mathbf{W}(1)$ ，括号内为迭代次数 $k=1$ 。

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

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

权向量修正为：

$$\mathbf{W}(k+1) = \mathbf{W}(k) - \rho \nabla J(k)$$

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) 在每一轮迭代中，如果有一个样本使  $\nabla J \neq 0$ ，回到 (2) 进行下一轮迭代；否则， $\mathbf{W}$ 不再变化，算法收敛。

## ●固定增量算法(Fixed increment algorithm)-梯度算法的应用 (\*: 选学)

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

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

求  $W(k)$  的递推公式:

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

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

方法: 函数对向量求导=函数对向量的分量求导, 即

$$\frac{\partial f}{\partial W} = \left[ \frac{\partial f}{\partial w_1}, \dots, \frac{\partial f}{\partial w_n}, \frac{\partial f}{\partial w_{n+1}} \right]^T$$

①首先求  $\mathbf{W}^T \mathbf{X}$  部分:

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

$$\begin{aligned} \frac{\partial(\mathbf{W}^T \mathbf{X})}{\partial \mathbf{W}} &= \frac{\partial}{\partial \mathbf{W}} \left( \sum_{i=1}^n w_i x_i + w_{n+1} \right) \\ &= \left[ \frac{\partial}{\partial w_1} \left( \sum_{i=1}^n w_i x_i + w_{n+1} \right), \dots, \frac{\partial}{\partial w_k} \left( \sum_{i=1}^n w_i x_i + w_{n+1} \right), \dots, \frac{\partial}{\partial w_{n+1}} \left( \sum_{i=1}^n w_i x_i + w_{n+1} \right) \right]^T \\ &= [x_1, \dots, x_k, \dots, x_n, 1]^T = \mathbf{X} \end{aligned}$$

另：矩阵论中有（也可以直接按梯度定义演算）

$$\begin{aligned} \frac{d\mathbf{X}}{d\mathbf{X}^T} &= \frac{d\mathbf{X}^T}{d\mathbf{X}} = \mathbf{I}_{n \times n} \\ \therefore \frac{\partial(\mathbf{W}^T \mathbf{X})}{\partial \mathbf{W}} &= \mathbf{I}_{(n+1) \times (n+1)} \mathbf{X}_{(n+1) \times 1} = \mathbf{X}_{(n+1) \times 1} \end{aligned}$$

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

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

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

$$W^T X \leq 0 \text{ 时, } \frac{\partial(|W^T X|)}{\partial W} = \frac{\partial(-W^T X)}{\partial W} = -X$$

$$\therefore \frac{\partial(|W^T X|)}{\partial W} = [\text{sign}(W^T X)] \cdot X$$

$$\text{其中 } \text{sign}(W^T X) = \begin{cases} +1, & \text{若 } W^T X > 0 \\ -1, & \text{若 } W^T X \leq 0 \end{cases}$$

$$\therefore \nabla J = \frac{\partial J(W, X)}{\partial W} = \frac{1}{2} [X \text{sign}(W^T X) - X]$$

## 2. 求 $W(k+1)$

将  $\nabla J = \frac{\partial J(W, X)}{\partial W} = \frac{1}{2} [X \operatorname{sign}(W^T X) - X]$  代入

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

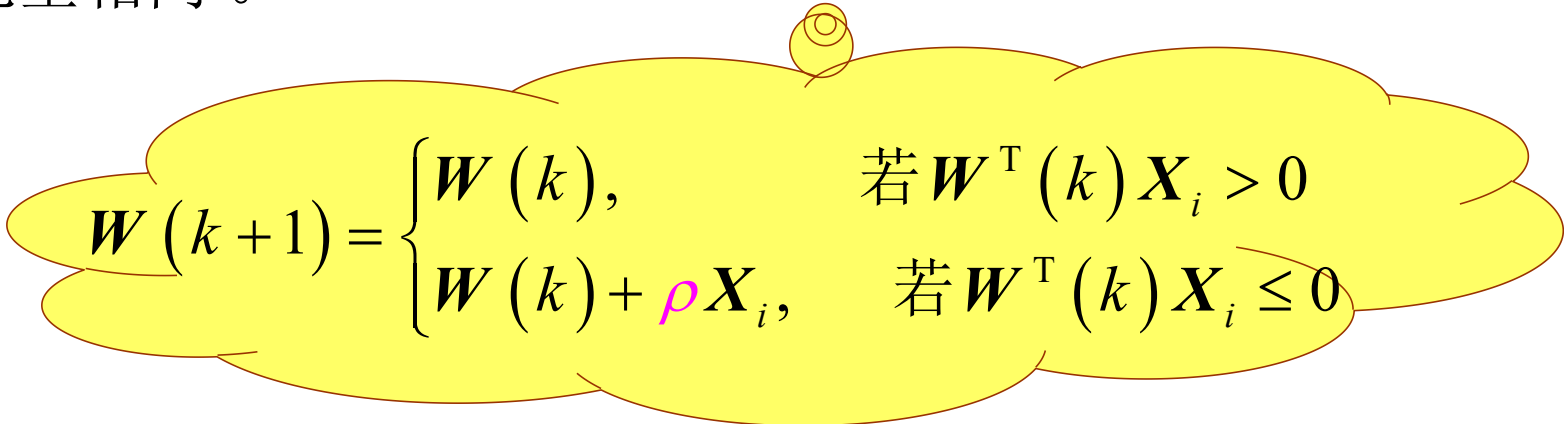
$$\text{得: } W(k+1) = W(k) - \rho \frac{1}{2} [X \operatorname{sign}(W^T(k)X) - X]$$

$$= W(k) + \frac{\rho}{2} [X - X \operatorname{sign}(W^T(k)X)]$$

$$= W(k) + \begin{cases} 0, & \text{若 } W^T(k)X > 0 \\ \rho X, & \text{若 } W^T(k)X \leq 0 \end{cases}$$

$$\text{即: } \mathbf{W}(k+1) = \mathbf{W}(k) + \begin{cases} 0, & \text{若 } \mathbf{W}^T(k)\mathbf{X} > 0 \\ \rho \mathbf{X}, & \text{若 } \mathbf{W}^T(k)\mathbf{X} \leq 0 \end{cases}$$

即为**固定增量算法**，与感知器算法（神经网络中的感知器）结论完全相同。


$$\mathbf{W}(k+1) = \begin{cases} \mathbf{W}(k), & \text{若 } \mathbf{W}^T(k)\mathbf{X}_i > 0 \\ \mathbf{W}(k) + \rho \mathbf{X}_i, & \text{若 } \mathbf{W}^T(k)\mathbf{X}_i \leq 0 \end{cases}$$

可以认为，神经网络中的**感知器算法是梯度法的特例**。梯度法是将感知器算法中联立不等式求解 $\mathbf{W}$ 的问题，转换为求目标函数极小值的问题，将原来有多个解的情况，变成求最优解的情况。

分类算法是通过模式样本来确定判别函数的系数，因此必须采用有代表性的数据，这样训练出来的判别函数，能合理反映模式数据的区隔情况。

**收敛性**(Convergence): 如果经过算法的有限次迭代运算后, 求出了一个使训练集中所有样本都能正确分类的 $W$ , 则称算法是收敛的。只要模型类别线性可分, 固定增量算法(感知器算法)就可以在有限的迭代步数里求出权向量 $W$ 的解。

**举例:** Two class training examples are as follows:

$$\omega_1 : X_1 = [0, 0]^T \quad X_2 = [0, 1]^T$$

$$\omega_2 : X_3 = [1, 0]^T \quad X_4 = [1, 1]^T$$

Use the **Fixed Increment (Perceptron) Algorithm** to classify them and find out the weight vector and the **discriminant function**.

Ans: Extend: all example to extended vectors;

Normalize: samples in  $\omega_2$  is multiplied by (-1).

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



Take  $W(1)=\mathbf{0}=(0,0,0)^T$ ,  $\rho=1$ . Iterative process is:

1<sup>st</sup> run:

$$W^T(1)X_1 = [0,0,0] \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = 0 \leq 0, \quad \text{So } W(2) = W(1) + X_1 = [0,0,1]^T$$

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

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

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

There are two cases of  $W^T(k)X_i \leq 0$  (misclassify), go to second run.

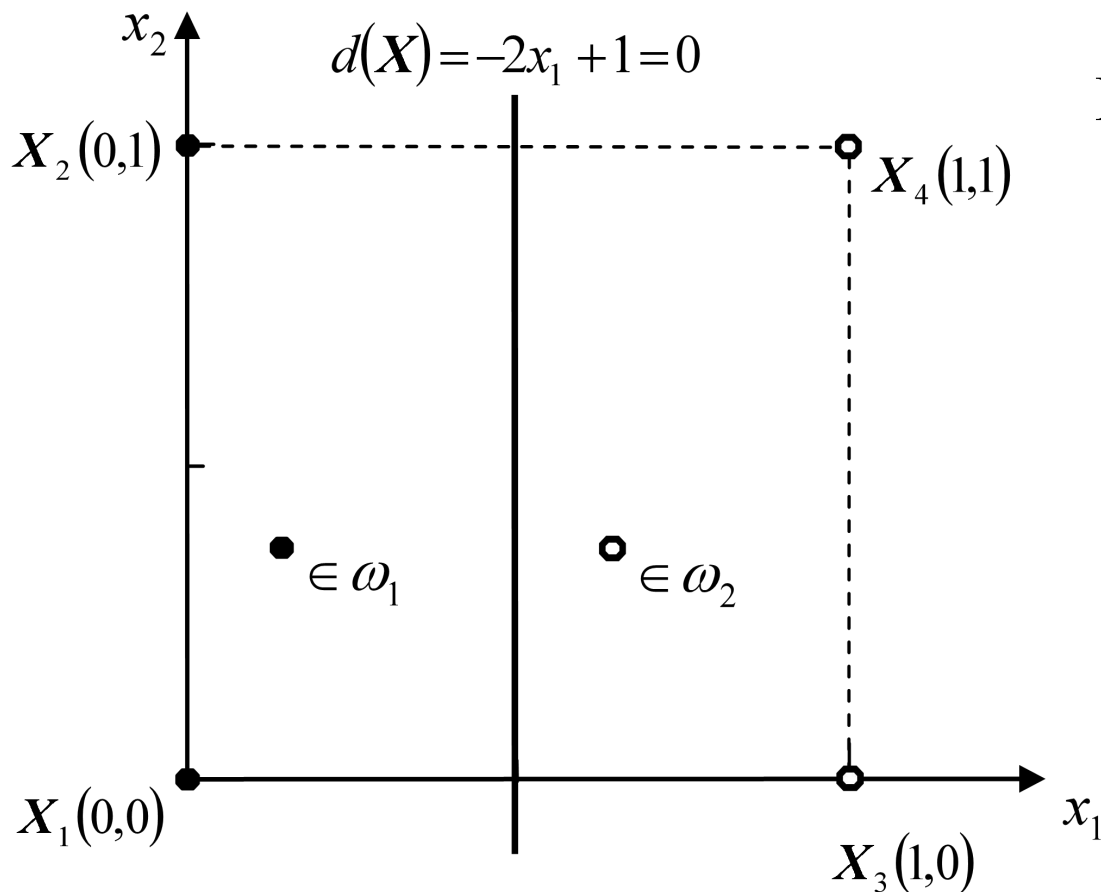
2<sup>nd</sup> run:  $W^T(5)X_1 = 0 \leq 0$ , So  $W(6) = W(5) + X_1 = [-1, 0, 1]^T$   
 $W^T(6)X_2 = 1 > 0$ , So  $W(7) = W(6) = [-1, 0, 1]^T$   
 $W^T(7)X_3 = 0 \leq 0$ , So  $W(8) = W(7) + X_3 = [-2, 0, 0]^T$   
 $W^T(8)X_4 = 2 > 0$ , So  $W(9) = W(8) = [-2, 0, 0]^T$

3<sup>rd</sup> run:  $W^T(9)X_1 = 0 \leq 0$ , So  $W(10) = W(9) + X_1 = [-2, 0, 1]^T$   
 $W^T(10)X_2 = 1 > 0$ , So  $W(11) = W(10)$   
 $W^T(11)X_3 = 1 > 0$ , So  $W(12) = W(11)$   
 $W^T(12)X_4 = 1 > 0$ , So  $W(13) = W(12)$

4<sup>th</sup> run:  $W^T(13)X_1 = 1 > 0$ , So  $W(14) = W(13)$   
 $W^T(14)X_2 = 1 > 0$ , So  $W(15) = W(14)$   
 $W^T(15)X_3 = 1 > 0$ , So  $W(16) = W(15)$   
 $W^T(16)X_4 = 1 > 0$ , So  $W(17) = W(16)$

All are classified correctly, So  $W = [-2, 0, 1]^T$

Discriminant function:  $d(X) = 2x_1 + 1$



Discri.face  $d(X)=0$

if take  $\rho$  、  $W(1)$  into  
other values, result will not  
be the same.

Solution is **not unique**.

## ➤ 2.4.2 Fisher判别分析(\*: 选学)

Fisher: 1890-1962, 英国数学家, 生物学家, 现代统计学奠基人之一。证明了孟德尔的遗传律符合达尔文的进化论。

Fisher判别是一种应用极为广泛的线性分类方法(1936年由Fisher提出)。

- Fisher判别的基本思想:

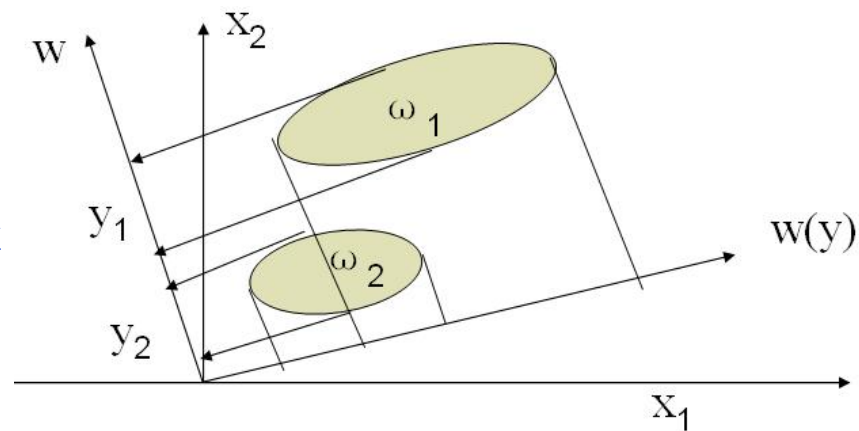
- 希望投影后的一维数据满足:

- ◆ 两类之间的距离尽可能远;
  - ◆ 每一类自身尽可能紧凑。

- 准则的描述:

- ◆ 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。

下面讨论通过映射投影来降低维数的方法。



X空间  $X = W^T x > 0$   $x \in \omega_1$

$X = W^T x < 0$   $x \in \omega_2$

映射Y空间  $Y \leftarrow W^T x > 0$   $x \in \omega_1$

$Y \leftarrow W^T x < 0$   $x \in \omega_2$

把X空间各点投影到Y空间的一条直线上，维数由2维降为1维。若适当选择W的方向，可以使二类分开。下面我们从数学上寻找最好的投影方向，即寻找最好的变换向量W\*的问题(以两类问题为例说明)。

在 $X$ 空间的均值向量:  $\bar{X}_i = \frac{1}{N_i} \sum_{x \in \omega_i} X$

$N_i$ 是 $\omega_i$ 的样本数  
( $i=1,2$ )

在 $Y$ 空间的投影均值:  $\bar{Y}_i = \frac{1}{N_i} \sum_{Y \in \omega_i} Y = \frac{1}{N_i} \sum_{X \in \omega_i} W^T X = W^T \bar{X}_i$

$$\therefore \bar{Y}_1 = W^T \bar{X}_1 \quad \bar{Y}_2 = W^T \bar{X}_2$$

投影样本之间的分离性用投影样本均值之差表示

$$|\bar{Y}_1 - \bar{Y}_2| = |W^T (\bar{X}_1 - \bar{X}_2)| \text{ 类间分离性越大越好}$$

投影样本类内离散度:

$$\sigma_i^2 = \sum_{Y \in \omega_i} (Y - \bar{Y}_i)^2 = \sum_{X \in \omega_i} (W^T X - W^T \bar{X}_i)^2 = W^T S_i W$$

其中  $S_i = \sum_{X \in \omega_i} (X - \bar{X}_i)(X - \bar{X}_i)^T$  - 样本类内离散度矩阵

$$\sigma_1^2 = W^T S_1 W \quad \sigma_2^2 = W^T S_2 W$$

注:  $s_i = \text{cov}(X_i) * (N_i - 1)$

$$S_1 = \sum_{X \in \omega_1} (X - \bar{X}_1)(X - \bar{X}_1)^T \quad S_2 = \sum_{X \in \omega_2} (X - \bar{X}_2)(X - \bar{X}_2)^T$$

投影样本总的离散度可用  $(\sigma_1^2 + \sigma_2^2)$  来表示，要求投影样本的总离散度越小越好。

*Fisher* 准则函数的构造  $J(W) = \frac{|\bar{Y}_1 - \bar{Y}_2|^2}{(\sigma_1^2 + \sigma_2^2)}$

类间距

总类内离散度

$$J(W) \text{ 的分子 } (\bar{Y}_1 - \bar{Y}_2)^2 = (W^T \bar{X}_1 - W^T \bar{X}_2)^2 = \mathbf{W}^T \mathbf{S}_b \mathbf{W}$$

其中  $S_b = (\bar{X}_1 - \bar{X}_2)(\bar{X}_1 - \bar{X}_2)^T$  - 样本类间离散度矩阵

$$J(W) \text{ 的分母 } \sigma_1^2 + \sigma_2^2 = W^T S_1 W + W^T S_2 W = W^T S_w W$$

$$S_w = S_1 + S_2 \text{ -- } S_w \text{ 为总的类内离散度矩阵}$$

所以Fisher准则函数为 $J(W) = \frac{W^T S_b W}{W^T S_w W}$

其中  $S_w$  为总的类内离散度矩阵,  $S_b$  为类间离散度矩阵

对 $J(W)$ 求极大值, 得  $W^* = S_w^{-1}(\bar{X}_1 - \bar{X}_2)$  推导过程见MLPR-第3讲数学知识补充.pdf

上式就是n维X空间向一维Y空间的最好投影方向, 它实际上是多维空间向一维空间的一种映射( $W^*$ 就是使模式样本的投影在类间最分散、类内最集中的最优解)。

$$S_t = S_w + S_b \quad S_t \text{ 为总体离散度矩阵}$$



这样就吧一个n维的问题转化为一维的问题。  
现在一维空间中设计 Fisher分类器：

$$Y = W^{*T} X > W_0 \Rightarrow X \in \omega_1$$

$$Y = W^{*T} X < W_0 \Rightarrow X \in \omega_2$$

$W_0$ 阈值的选择

- $1. W_0 = \frac{\bar{Y}_1 + \bar{Y}_2}{2}$
- $2. W_0 = \frac{N_1 \bar{Y}_1 + N_2 \bar{Y}_2}{N_1 + N_2} = \frac{N_1 W^T \bar{X}_1 + N_2 W^T \bar{X}_2}{N_1 + N_2}$

$$3. W_0 = \bar{Y}_1 + (\bar{Y}_2 - \bar{Y}_1) \frac{\sum_{k=1}^{N_1} (Y_{k1} - \bar{Y}_1)^2}{\sum_{k=1}^{N_1} (Y_{k1} - \bar{Y}_1)^2 + \sum_{k=1}^{N_2} (Y_{k2} - \bar{Y}_2)^2}$$

$Y_{ki}$  表示第  $i$  类中第  $k$  个样本的投影值

$N_1$  为  $\omega_1$  样本数

$N_2$  为  $\omega_2$  样本数

当  $W_0$  选定后，对任一样本  $X$ ，只要判断  $Y = W^{*T} X > W_0$ ，则

$X \in \omega_1$ ；  $Y = W^{*T} X < W_0$ ，则  $X \in \omega_2$ ；分类问题就解决了。

## 两类问题Fisher算法实现步骤:

(1)求两类样本均值向量  $\bar{\mathbf{x}}_1$ 和 $\bar{\mathbf{x}}_2$  ；

(2)求两类样本类内离散度矩阵 $S_i$  ( $i=1,2$ )；

(3)求总的类内散度矩阵 $S_w=S_1+S_2$ ；

(4)求最优投影向量 $W^*$ ，  $W^* = S_w^{-1}(\bar{\mathbf{x}}_1 - \bar{\mathbf{x}}_2)$  ；

(5)对两类已知样本，求出它们在 $W^*$ 上的投影点 $y_i$ ：

$$y_1 = W^{*T} X_1, y_2 = W^{*T} X_2$$

(6)求各类样本的均值  $\bar{y}_i, \bar{Y}_i = \frac{1}{N_i} \sum_{Y \in \omega_i} Y$  ；

(7)选取分类阈值 $W_0$ ，如取  $W_0 = \frac{\bar{Y}_1 + \bar{Y}_2}{2}$  ；

(8)对未知样本 $X$ ，计算它在 $W^*$ 上的投影点 $y$ :  $y = W^{*T} X$ ；

(9)根据判别规则对未知样本 $X$ 进行分类：  $Y = W^{*T} X > W_0$ ， 则  $X \in \omega_1$ ；  $Y = W^{*T} X < W_0$ ， 则  $X \in \omega_2$  。

## \*广义Fisher准则(\*: 了解)

基于两类问题的Fisher分类准则，能很容易地扩展为多类问题的Fisher准则，又称广义Fisher准则。在 $d$ 维空间中，对于 $M$ 类问题，一定存在 $(M-1)$ 个线性判别函数。因此广义Fisher准则所涉及的投影问题是：在 $d$ 维空间中，将 $M$ 类样本集合投影在 $(M-1)$ 维空间上(在此假设， $M \leq d$ ，即类别数小于特征空间的维数)。

多类问题的 $(M-1)$ 个判别函数为

$$y_i = \mathbf{w}_i^T \mathbf{x}, i = 1, 2, \dots, M-1$$

或者写为矩阵形式

$$\mathbf{y} = \mathbf{W}^T \mathbf{x}$$

$$\text{其中, } \mathbf{W} = [\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_{M-1}], \mathbf{y} = [y_1, y_2, \dots, y_{M-1}]^T$$

因此，广义Fisher准则所要解决的问题是：寻找 $M$ 类样本集合 $\mathbf{x}$ 在 $\mathbf{W}$ 张成的 $(M-1)$ 维空间的最佳投影向量集合 $\mathbf{W}^*$ 。

# Fisher判别分析举例：

## 1. 蠓的分类问题

两种蠓Af和Apf已由生物学家根据它们的触角和翼长加以区分 (Af是能传播花粉的益虫，Apf是会传播疾病的害虫)，两个矩阵中分别给出了6只Apf和9只Af蠓的触角长(对应于矩阵的第1列)和翼长(对应于矩阵的第2列)的数据 (See next slide)。根据触角长和翼长这两个特征来识别一个样本是Af还是Apf是重要的。

(1) 试给出该问题的Fisher分类器；

(2) 有三个待识别的模式样本，它们分别是

$(1.24, 1.80)^T$ ,  $(1.28, 1.84)^T$ ,  $(1.40, 2.04)^T$ ，试问这三个样本属于哪一种蠓。

## 2. 假设

(1)两种群Apf 和Af的两个特征的期望值、标准差、相关系数与由矩阵数据给出的样本统计量一致；

(2)两种群Apf 和Af的两个特征服从二元正态分布；

(3)所给样本数据无误差。

蠓Apf		蠓Af	
触角长	翼长	触角长	翼长
1.14	1.78	1.24	1.72
1.18	1.96	1.36	1.74
1.20	1.86	1.38	1.64
1.26	2.00	1.38	1.82
1.30	2.00	1.38	1.90
1.28	1.96	1.40	1.70
		1.48	1.82
		1.54	2.08
		1.56	1.78

### 3. 应用Fisher进行判别分析

#### (1) 求样本均值向量

$$\bar{x}_1^{(1)} = \frac{1}{6}(1.14 + 1.18 + \dots + 1.28) = 1.227$$

$$\bar{x}_1^{(2)} = \frac{1}{6}(1.78 + 1.96 + \dots + 1.96) = 1.927$$

$$\bar{x}_2^{(1)} = \frac{1}{9}(1.24 + 1.36 + \dots + 1.56) = 1.413$$

$$\bar{x}_2^{(2)} = \frac{1}{9}(1.72 + 1.74 + \dots + 1.78) = 1.800$$

$$\bar{\mathbf{x}}_1 = \begin{bmatrix} \bar{x}_1^{(1)} \\ \bar{x}_1^{(2)} \end{bmatrix} = \begin{bmatrix} 1.227 \\ 1.927 \end{bmatrix}$$

$$\bar{\mathbf{x}}_2 = \begin{bmatrix} \bar{x}_2^{(1)} \\ \bar{x}_2^{(2)} \end{bmatrix} = \begin{bmatrix} 1.413 \\ 1.800 \end{bmatrix}$$

(2)求两类样本类内离散度矩阵 $S_i$

$$\begin{aligned} S_1 &= \begin{pmatrix} 1.14-1.227 \\ 1.78-1.927 \end{pmatrix} (1.14-1.227 \quad 1.78-1.927) + \dots \\ &\quad + \begin{pmatrix} 1.28-1.227 \\ 1.96-1.927 \end{pmatrix} (1.28-1.227 \quad 1.96-1.927) \\ &= \begin{pmatrix} 0.0197 & 0.0225 \\ 0.0225 & 0.0389 \end{pmatrix} \end{aligned}$$

$$\begin{aligned} S_2 &= \begin{pmatrix} 1.24-1.413 \\ 1.72-1.804 \end{pmatrix} (1.24-1.413 \quad 1.72-1.804) + \dots \\ &\quad + \begin{pmatrix} 1.56-1.413 \\ 1.78-1.804 \end{pmatrix} (1.56-1.413 \quad 1.78-1.804) \\ &= \begin{pmatrix} 0.0784 & 0.0536 \\ 0.0536 & 0.1352 \end{pmatrix} \end{aligned}$$



(3)求总的类内离散度矩阵 $S_w$

$$S_w = S_1 + S_2 = \begin{pmatrix} 0.0981 & 0.0761 \\ 0.0761 & 0.1741 \end{pmatrix}$$

(4)求权向量 $W^*$

$$S_w^{-1} = \begin{pmatrix} 15.4209 & -6.7422 \\ -6.7422 & 8.6905 \end{pmatrix}$$

$$W^* = S_w^{-1}(\bar{\mathbf{x}}_1 - \bar{\mathbf{x}}_2) = (-3.7326 \quad 2.3593)^T$$

(5) 求两类已知样本 $X_1$ 和 $X_2$ 在 $W^*$ 上的投影点 $y_i$

$$y_1 = W^{*T} X_1, y_2 = W^{*T} X_2$$

$$y_1 = [-0.0555 \quad 0.2199 \quad -0.0907 \quad 0.0156 \quad -0.1337 \quad -0.1534]^T$$

$$y_2 = [-0.5703 \quad -0.9711 \quad -1.2816 \quad -0.8570 \quad -0.6682 \quad -1.2147 \quad -1.2302 \quad -0.8407 \quad -1.6232]^T$$

(6)求各类样本的均值  $\bar{Y}_i$

$$\bar{Y}_i = \frac{1}{N_i} \sum_{Y \in \omega_i} Y$$

$$\bar{y}_1 = -0.0330$$

$$\bar{y}_2 = -1.0286$$

## (7)选取分类阈值 $W_0$

可以有三种取法，这里取：

$$W_0 = \frac{\bar{Y}_1 + \bar{Y}_2}{2} = -0.5308$$

(8)对待测的样本 $x$ ，计算它在 $W^*$ 上的投影点 $y$

公式： $y=W^{*T} x$

$$X = \begin{pmatrix} 1.24 & 1.80 \\ 1.28 & 1.84 \\ 1.40 & 2.04 \end{pmatrix}$$

$\Rightarrow$

$$Y = (-0.3816, -0.4365, -0.4126)^T$$

(9)根据判别规则对样本 $X$ 进行分类

$Y=W^*T X > W_0$ ， 则 $X \in \omega_1$ ；

$Y=W^*T X < W_0$ ， 则 $X \in \omega_2$ 。

根据(7)， 分类阈值 $W_0 = -0.5308$

而：

$$Y = (-0.3816, -0.4365, -0.4126)^T$$

可以看出，  $y_i > W_0$

因此， 这三个样本都属于Apf蠓( $\omega_1$ 类)

(回顾本例二分类问题在上一讲的 $K$ 近邻法的Python程序实现)

## •Fisher线性判别分析Python编程:

其他部分套用sklearn机器学习库中实现Fisher分类的方法是采用discriminant\_analysis类的LinearDiscriminantAnalysis。

Fisher分类器Python关键语句:

```
from sklearn import discriminant_analysis
```

```
Fisher_clf = discriminant_analysis.LinearDiscriminantAnalysis()
```

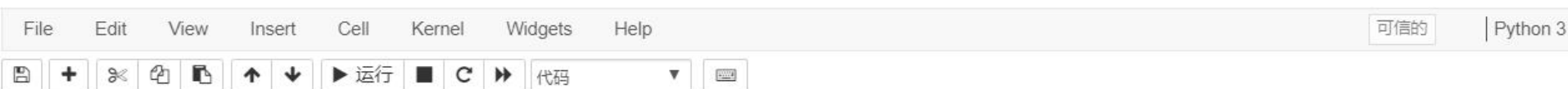
第2讲的Python程序模板中的相关语句，如Fisher\_clf.fit()、Fisher\_clf.predict()等。

Fisher LDA Python实现两例：1. 蠓的二分类；2. 鸢尾花三分类。  
可参见博文：

<https://yuanyx.blog.csdn.net/article/details/114813129>



## 本例蠓虫二分类问题，采用Fisher LDA的Python程序运行界面截图：



```
In [1]: ▶ #Fisher线性判别分析-Fisher LDA
#蠓虫的Fisher二分类程序
#Filename: Fisher_LDA_Midge.ipynb
#Import Library
import numpy as np
from sklearn import discriminant_analysis
#Assumed you have X (predictor) and Y (target) for training data set and x_test(predictor) of test_dataset
X=np.array([[1.14, 1.78], [1.18, 1.96], [1.20, 1.86], [1.26, 2.00], [1.30, 2.00], [1.28, 1.96],
[1.24, 1.72], [1.36, 1.74], [1.38, 1.64], [1.38, 1.82], [1.38, 1.90], [1.40, 1.70], [1.48, 1.82], [1.54, 2.08], [1.56, 1.78]])
y=np.array([0,0,0,0,0,0,1,1,1,1,1,1,1,1,1])
#(X,y)作为训练集, 前6个为Apf类 (类标签:0), 后9个样本为Af类 (类标签:1)
#定义Fisher分类器对象fisher_clf
fisher_clf = discriminant_analysis.LinearDiscriminantAnalysis()
#调用该对象的训练方法
fisher_clf.fit(X,y)
x_test=np.array([[1.24, 1.8], [1.28, 1.84], [1.4, 2.04]]) #待测试的三个样本
y_test=([0,0,0]) #待测试的三个样本的类标签
#(x_test,y_test)三个样本作为测试集
#调用该对象的测试方法
y_pred=fisher_clf.predict(x_test)
print(' 测试数据集的正确标签为:',y_test)
print(' 测试数据集的预测标签为:',y_pred)
from sklearn.metrics import accuracy_score
testing_acc=accuracy_score(y_test, y_pred)*100
print(' Fisher线性分类器测试准确率: {:.2f}%'.format(testing_acc))
```

测试数据集的正确标签为: [0, 0, 0]  
测试数据集的预测标签为: [0 0 0]  
Fisher线性分类器测试准确率: 100.00%

## 蠓虫二分类问题Fisher LDA的Python程序清单:

```
#Fisher线性判别分析-Fisher LDA
#蠓虫的Fisher二分类程序
#Filename: Fisher_LDA_Midge.ipynb
#Import Library
import numpy as np
from sklearn import discriminant_analysis
#Assumed you have X (predictor) and Y (target) for training data set and x_test(predictor) of test_dataset
X=np.array([[1.14,1.78],[1.18,1.96],[1.20,1.86],[1.26,2.00],[1.30,2.00],[1.28,1.96],
[1.24,1.72],[1.36,1.74],[1.38,1.64],[1.38,1.82],[1.38,1.90],[1.40,1.70],[1.48,1.82],[1.54,2.08],[1.56,1.78]])
y=np.array([0,0,0,0,0,0,1,1,1,1,1,1,1,1,1])
#(X,y)作为训练集,前6个为Apf类 (类标签:0), 后9个样本为Af类 (类标签:1)
#定义Fisher分类器对象fisher_clf
fisher_clf = discriminant_analysis.LinearDiscriminantAnalysis()
#调用该对象的训练方法
fisher_clf.fit(X,y)
x_test=np.array([[1.24,1.8],[1.28,1.84],[1.4,2.04]]) #待测试的三个样本
y_test=([0,0,0]) #待测试的三个样本的类标签
#(x_test,y_test)三个样本作为测试集
#调用该对象的测试方法
y_pred=fisher_clf.predict(x_test)
print('测试数据集的正确标签为:',y_test)
print('测试数据集的预测标签为:',y_pred)
from sklearn.metrics import accuracy_score
testing_acc=accuracy_score(y_test, y_pred)*100
print('Fisher线性分类器测试准确率: {:.2f}%'.format(testing_acc))
```

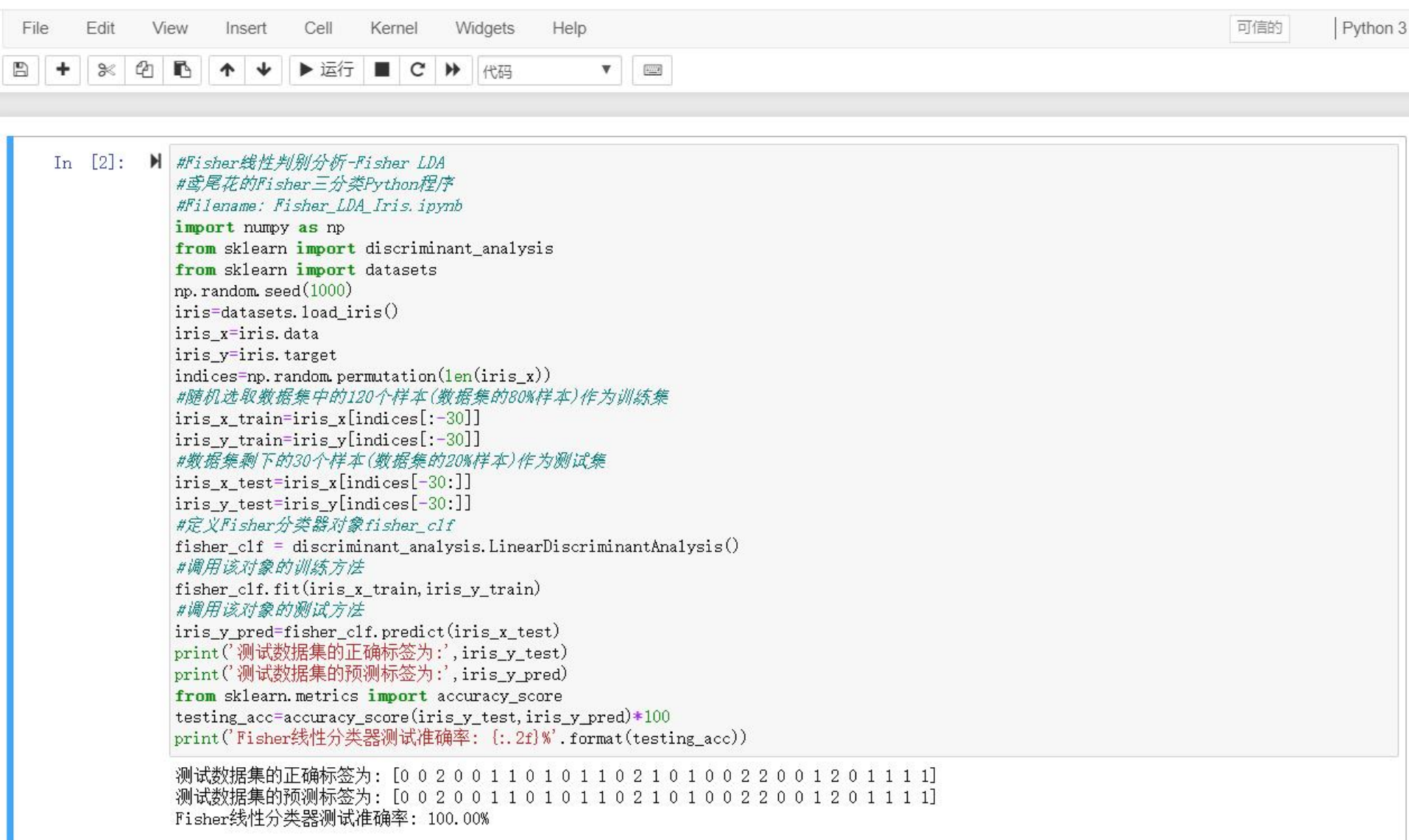
## 输出结果:

测试数据集的正确标签为: [0, 0, 0]

测试数据集的预测标签为: [0 0 0]

Fisher线性分类器测试准确率: 100.00%

以下是鸢尾花三分类问题，采用Fisher LDA的Python程序运行界面截图：



```
In [2]: #Fisher线性判别分析-Fisher LDA
#鸢尾花的Fisher三分类Python程序
#Filename: Fisher_LDA_Iris.ipynb
import numpy as np
from sklearn import discriminant_analysis
from sklearn import datasets
np.random.seed(1000)
iris=datasets.load_iris()
iris_x=iris.data
iris_y=iris.target
indices=np.random.permutation(len(iris_x))
#随机选取数据集集中的120个样本(数据集的80%样本)作为训练集
iris_x_train=iris_x[indices[:-30]]
iris_y_train=iris_y[indices[:-30]]
#数据集剩下的30个样本(数据集的20%样本)作为测试集
iris_x_test=iris_x[indices[-30:]]
iris_y_test=iris_y[indices[-30:]]
#定义Fisher分类器对象fisher_clf
fisher_clf = discriminant_analysis.LinearDiscriminantAnalysis()
#调用该对象的训练方法
fisher_clf.fit(iris_x_train, iris_y_train)
#调用该对象的测试方法
iris_y_pred=fisher_clf.predict(iris_x_test)
print('测试数据集的正确标签为:', iris_y_test)
print('测试数据集的预测标签为:', iris_y_pred)
from sklearn.metrics import accuracy_score
testing_acc=accuracy_score(iris_y_test, iris_y_pred)*100
print('Fisher线性分类器测试准确率: {:.2f}%'.format(testing_acc))

测试数据集的正确标签为: [0 0 2 0 0 1 1 0 1 0 1 1 0 2 1 0 1 0 0 2 2 0 0 1 2 0 1 1 1 1]
测试数据集的预测标签为: [0 0 2 0 0 1 1 0 1 0 1 1 0 2 1 0 1 0 0 2 2 0 0 1 2 0 1 1 1 1]
Fisher线性分类器测试准确率: 100.00%
```

## 以下是鸢尾花三分类问题Fisher LDA的Python程序清单：

```
#Fisher线性判别分析-Fisher LDA
#鸢尾花的Fisher三分类Python程序
#Filename: Fisher_LDA_Iris.ipynb
import numpy as np
from sklearn import discriminant_analysis
from sklearn import datasets
np.random.seed(1000)
iris=datasets.load_iris()
iris_x=iris.data
iris_y=iris.target
indices=np.random.permutation(len(iris_x))
#随机选取数据集中的120个样本(数据集的80%样本)作为训练集
iris_x_train=iris_x[indices[:-30]]
iris_y_train=iris_y[indices[:-30]]
#数据集剩下的30个样本(数据集的20%样本)作为测试集
iris_x_test=iris_x[indices[-30:]]
iris_y_test=iris_y[indices[-30:]]
#定义Fisher分类器对象fisher_clf
fisher_clf = discriminant_analysis.LinearDiscriminantAnalysis()
#调用该对象的训练方法
fisher_clf.fit(iris_x_train,iris_y_train)
#调用该对象的测试方法
iris_y_pred=fisher_clf.predict(iris_x_test)
print('测试数据集的正确标签为:',iris_y_test)
print('测试数据集的预测标签为:',iris_y_pred)
from sklearn.metrics import accuracy_score
testing_acc=accuracy_score(iris_y_test,iris_y_pred)*100
print('Fisher线性分类器测试准确率: {:.2f}%'.format(testing_acc))
```

## 输出结果：

测试数据集的正确标签为: [0 0 2 0 0 1 1 0 1 0 1 1 0 2 1 0 1 0 0 2 2 0 0 1 2 0 1 1 1 1]

测试数据集的预测标签为: [0 0 2 0 0 1 1 0 1 0 1 1 0 2 1 0 1 0 0 2 2 0 0 1 2 0 1 1 1 1]

Fisher线性分类器测试准确率: 100.00%

## 本章小结:

判断分析是利用原有的模式分类信息，得到判别函数(判别函数是这种分类的函数关系式，可以是与分类信息相关的若干个特征或指标的线性关系式)，然后利用该函数去判断未知的模式或样本属于哪一类。因此，这是一个学习和预测的过程。

分类器设计步骤：1、抽取类别标志明确的样本集合作为训练样本。2、确定准则函数 $J(\mathbf{w}, \mathbf{x})$ ，准则函数应满足：(1)  $J$ 为 $\mathbf{w}$ 、 $\mathbf{x}$ 的函数；(2)  $J$ 应能充分表征分类器的性能，其优化结果满足分类要求；3、使用**最优化方法**求出准则函数的极值解 $\mathbf{w}^*$ 。

## 思考题/课外作业2:

1. 若  $X=[2 \ 3]$ ,  $Y=[3 \ 1]$ , 求  $\text{COV}(X,Y)$ .

2. 已知  $X = \begin{bmatrix} 4 & 5 & 1 \\ 3 & 1 & 0 \\ 2 & 3 & 2 \end{bmatrix}$ , 求  $X^{-1}$ ,  $\text{COV}(X)$

3. 有一个二类问题, 其判别函数为  $g(x)=3x_1+5x_2-6x_3-2$ 。试将下面三个模式分别进行分类:

$x_1=[4 \ 7 \ 1]^T$ ,  $x_2=[1 \ -5 \ 2]^T$ ,  $x_3=[4 \ 4 \ 5]^T$ 。

4. 有一个三类问题，其判别函数为：

$$g_1(\mathbf{X}) = x_1 + 2x_2 - 4, \quad g_2(\mathbf{X}) = x_1 - 4x_2 + 4$$

$$g_3(\mathbf{X}) = -x_1 + 3$$

(1) 设这些函数是在**多类情况1**条件下确定的，绘出判别界面(边界)及每一模式类别的区域。

(2) 设为**多类情况2**，并使  $g_{12}(\mathbf{X}) = g_1(\mathbf{X})$ ,  $g_{13}(\mathbf{X}) = g_2(\mathbf{X})$ ,  $g_{23}(\mathbf{X}) = g_3(\mathbf{X})$ ，绘出判别界面及每一模式类别的区域。

(3) 设  $g_1(\mathbf{X})$ ,  $g_2(\mathbf{X})$ ,  $g_3(\mathbf{X})$  是在**多类情况3**条件下确定的，绘出其判别界面及每一模式类别的区域。

End of this lecture.  
Thanks!



## 思考题1~3参考解答:

### 1.分析: 对于对于二维随机向量(X,Y)

*Definition :*

$$\text{cov}(X, Y) = E[(X - E(X))(Y - E(Y))] = E(XY) - E(X)E(Y)$$

$$\text{cov}(X, Y) = \begin{bmatrix} s_{11}^2 & s_{12}^2 \\ s_{21}^2 & s_{22}^2 \end{bmatrix}$$

$$s_{11}^2 = \sum (x_i - \bar{x})^2 / (n-1)$$

$$s_{12}^2 = s_{21}^2 = \sum (x_i - \bar{x})(y_i - \bar{y}) / (n-1)$$

$$= \sum x_i y_i - \sum x_i \sum y_i / n$$

$$s_{22}^2 = \sum (y_i - \bar{y})^2 / (n-1)$$

$$r_{ij} = \frac{s_{ij}^2}{\sqrt{s_{ii}^2 s_{jj}^2}}$$

解：

$$n = 2$$

$$\therefore \text{cov}(X, Y) = \begin{bmatrix} s_{11}^2 & s_{12}^2 \\ s_{21}^2 & s_{22}^2 \end{bmatrix}$$

$$s_{11}^2 = \{(2 - 2.5)^2 + (3 - 2.5)^2\} / (n - 1) = 0.5$$

$$s_{12}^2 = s_{21}^2 = \{(2 - 2.5)(3 - 2) + (3 - 2.5)(1 - 2)\} / (n - 1) = -1$$

$$s_{22}^2 = \{(3 - 2)^2 + (1 - 2)^2\} / (n - 1) = 2$$

$$\therefore \text{cov}(X, Y) = \begin{bmatrix} 0.5 & -1 \\ -1 & 2 \end{bmatrix}$$

$$r_{12} = s_{12}^2 / \sqrt{s_{11}^2 \cdot s_{22}^2} = -1$$

若 $X=[1,3]$ ,  $Y=[2,5]$ , 试计算 $\text{COV}(X,Y)$ 。

解:  $X=[1,3]$ ,  $Y=[2,5]$

$$n = 2$$

$$\hat{\sigma} \text{ cov}(X, Y) = \begin{bmatrix} s_{11}^2 & s_{12}^2 \\ s_{21}^2 & s_{22}^2 \end{bmatrix}$$

$$s_{11}^2 = \{(1-2)^2 + (3-2)^2\}/(n-1) = 2$$

$$s_{12}^2 = s_{21}^2 = \{(1-2)(2-3.5) + (3-2)(5-3.5)\}/(n-1) = 3$$

$$s_{22}^2 = \{(2-3.5)^2 + (5-3.5)^2\}/(n-1) = 2$$

$$\therefore \text{cov}(X, Y) = \begin{bmatrix} 2 & 3 \\ 3 & 4.5 \end{bmatrix}$$

## 2.分析:

对于n维随机变量  $(X_1, X_2, \dots, X_n)$ ,  $X_i$  和  $X_j$  的协方差定义为  $\sigma_{ij} = COV(X_i, X_j) = E(X_i - EX_i)(X_j - EX_j)$ , 则称

$$\Sigma = \begin{pmatrix} \sigma_{11}^2 & \sigma_{12}^2 & \lambda & \sigma_{1n}^2 \\ \sigma_{21}^2 & \sigma_{22}^2 & \lambda & \sigma_{2n}^2 \\ \acute{\theta} & \acute{\theta} & \acute{\theta} & \acute{\theta} \\ \sigma_{n1}^2 & \sigma_{n2}^2 & \lambda & \sigma_{nn}^2 \end{pmatrix}$$

为  $(X_1, X_2, \dots, X_n)$  的协方差矩阵

$$\text{对于模式样本矩阵 } X = \begin{pmatrix} x_{11} & x_{12} & \bar{\lambda} & x_{1n} \\ x_{21} & x_{22} & \bar{\lambda} & x_{2n} \\ \bar{\theta} & \bar{\theta} & \bar{\theta} & \bar{\theta} \\ x_{m1} & x_{m2} & \bar{\lambda} & x_{mn} \end{pmatrix}$$

$$\text{有模式样本协方差矩阵 } \Sigma = \begin{pmatrix} s_{11}^2 & s_{12}^2 & \bar{\lambda} & s_{1n}^2 \\ s_{21}^2 & s_{22}^2 & \bar{\lambda} & s_{2n}^2 \\ \bar{\theta} & \bar{\theta} & \bar{\theta} & \bar{\theta} \\ s_{n1}^2 & s_{n2}^2 & \bar{\lambda} & s_{nn}^2 \end{pmatrix}$$

$$s_{11}^2 = \sum (x_{i1} - \bar{x}_1)^2 / (m-1)$$

$$s_{12}^2 = \sum (x_{i1} - \bar{x}_1)(x_{i2} - \bar{x}_2) / (m-1)$$

$$= \sum x_{i1}x_{i2} - \sum x_{i1} \sum x_{i2} / m$$

.....

$$s_{ij}^2 = \sum_{k=1}^m (x_{ki} - \bar{x}_i)(x_{kj} - \bar{x}_j) / (m-1)$$

$$= \sum_{k=1}^m x_{ki}x_{kj} - \sum_{k=1}^m x_{ki} \sum_{k=1}^m x_{kj} / m$$

$$\text{其中, } \bar{x}_j = \sum_{k=1}^m x_{kj} / m$$

.....

$$\text{相关系数 } r_{ij} = \frac{s_{ij}^2}{\sqrt{s_{ii}^2 s_{jj}^2}}$$

答案:  $COV(X) = \begin{bmatrix} 1 & 1 & -\frac{1}{2} \\ 1 & 4 & 1 \\ -\frac{1}{2} & 1 & 1 \end{bmatrix}$

[Python code:](#)

```
import numpy as np
X=np.array([[4,5,1],[3,1,0],[2,3,2]])
np.linalg.inv(X)
np.cov(X.T)
```

```
In [1]: ► import numpy as np
X=np.array([[4,5,1],[3,1,0],[2,3,2]])
np.linalg.inv(X)
np.cov(X.T)
```

```
Out[1]: array([[ 1. ,  1. , -0.5],
               [ 1. ,  4. ,  1. ],
               [-0.5,  1. ,  1.]])
```

补例：若  $X = \begin{bmatrix} 1 & 3 \\ 2 & 5 \end{bmatrix}$ ，试计算  $\text{COV}(X)$ 。



解:  $X = \begin{bmatrix} 1 & 3 \\ 2 & 5 \end{bmatrix}$

$$n = 2$$

$$\hat{\sigma}^2 \text{cov}(X) = \begin{bmatrix} s_{11}^2 & s_{12}^2 \\ s_{21}^2 & s_{22}^2 \end{bmatrix}$$

$$s_{11}^2 = \{(1-1.5)^2 + (2-1.5)^2\} / (n-1) = 0.5$$

$$s_{12}^2 = s_{12}^2 = \{(1-1.5)(3-4) + (2-1.5)(5-4)\} / (n-1) = 1$$

$$s_{22}^2 = \{(3-4)^2 + (5-4)^2\} / (n-1) = 2$$

$$\therefore \text{cov}(X) = \begin{bmatrix} 0.5 & 1 \\ 1 & 2 \end{bmatrix}$$

### \*MATLAB Code:

```
X=[1 3;2 5]
```

```
cov(X)
```

```
X =
```

```
1 3
```

```
2 5
```

```
ans=
```

```
0.5 1
```

```
1 2
```

---

### \*\*Python Code:

File Edit View Insert Cell Kernel Widgets Help

```
In [2]: 1 #Filename: python_statistics_2.ipynb
        2 import numpy as np
        3 X=np.array([[1,3],[2,5]]).T
        4 print(X)
        5 np.cov(X) #Compute the covariance matrix
```

```
[[1 2]
 [3 5]]
```

```
Out[2]: array([[ 0.5,  1. ],
               [ 1. ,  2. ]])
```

3.答案：X1属于 $\omega_1$ 类；X2属于第 $\omega_2$ 类；无法判断X3属于二类中的哪一类。