第5章
线性判别函数
Linear Discriminant Functions

向 世 明

smxiang@nlpr.ia.ac.cn

http://www.escience.cn/people/smxiang/index.html

时空数据分析与学习课题组(STDAL)

中科院自动化研究所　模式识别国家重点实验室

助教：张明亮(zhangmingliang2018@ia.ac.cn)

程真(chengzhen2019@nlpr.ia.ac.cn)

张姣(zhangjiao2019@ia.ac.cn)

University of Chinese Academy of Sciences

5.1 引言

统计模式识别方法

生成模型
(Density-based, Bayes decision)

Parametric
- ✓ Gaussian
- ✓ Dirichlet
- ✓ Bayesian network
- ✓ Hidden Markov model

Non-Parametric
- ✓ Histogram density
- ✓ Parzen window
- ✓ K-nearest neighbor

Semi-Parametric
- ✓ Gaussian mixture

Discriminative模型
(discriminant/decision function)

- ✓ **Linear Methods**
- ✓ Neural network
- ✓ Logistic regression
- ✓ Decision tree
- ✓ Kernel (SVM)
- ✓ Boosting

a.k.a. Non-parametric

5.1 引言

- **前面几章主要内容回顾：**
  - 贝叶斯决策理论
  - 概率密度估计——函数已知情形时，参数估计
  - 非参数估计——密度函数未知
- **本章主要任务：**
  - 假定用于分类的**判别函数的参数形式已知，直接从样本来估计判别函数的参数**。
  - 优点：
    - **不需要有关概率密度函数确切的参数形式。**

5.1 引言

- **模式分类的途径**
  - **估计类条件概率密度函数**
    - 利用贝叶斯公式求出后验概率，然后决策
    - 核心步骤：概率密度估计（参数估计和非参数估计）
  - **直接估计后验概率**
    - 不需要估计类条件概率密度函数
    - K－近邻分类器
  - **直接计算判别函数**
    - 不需要估计类条件概率密度函数
    - **直接找到可用于分类的判别函数**

5.1 引言

- **回顾Bayes分类器**
  - 已知：类先验概率 $p(\omega_i)$ 和类条件密度函数 $p(\mathbf{x}|\omega_i)$
  - 任务：估计一个决策函数，借此进行分类
  - 主要方法：参数估计与非参数估计
  - 特点：需要大量的样本，需要知道某些概率及其形式
- **可否利用样本直接设计分类器？**
  - **方法分类**
    - **线性判别函数、支持向量机、Fisher线性判别函数**
    - **广义线性判别函数、非线性判别函数、核学习机**

5.1 引言

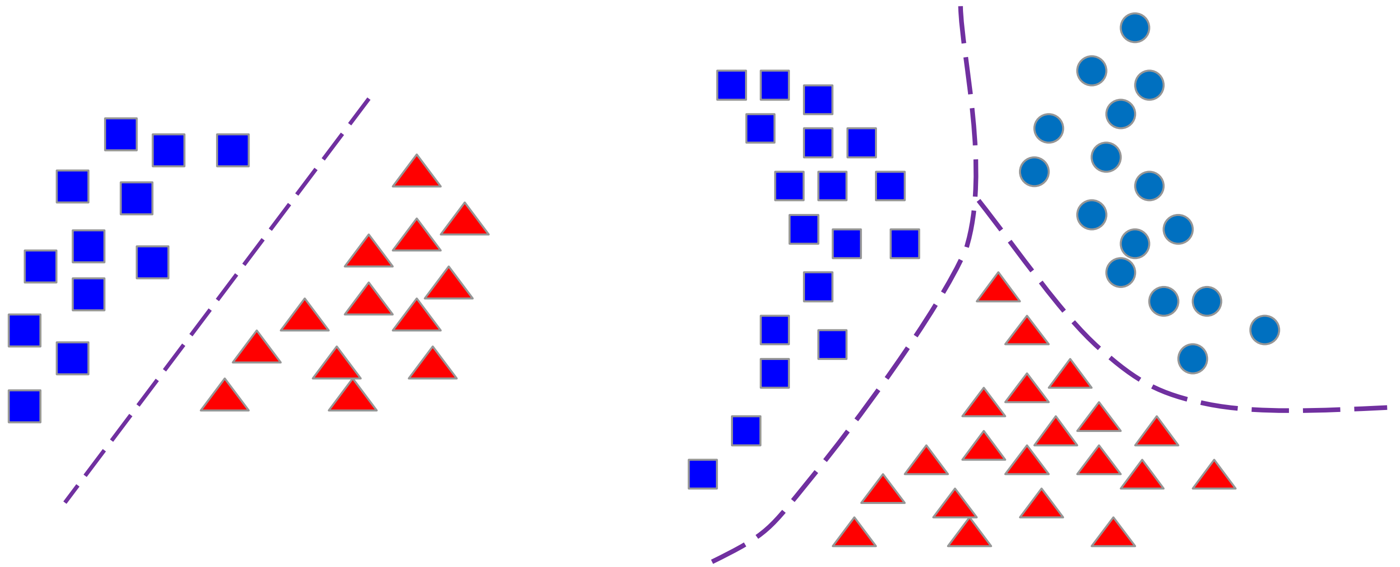
- **本章学习判别函数的基本技术路线：**
  - 假定有 n 个 d 维空间中的样本，每个样本的类别标签已知，且一共有 c 个不同的类别。
  - 假定判别函数的形式已知，采用样本来训练判别函数的参数，即寻找一个判别函数。
  - 对于给定的新样本 $\mathbf{x} \in \mathrm{R}^d$，判定它属于 $\omega_1, \omega_2, \ldots, \omega_c$ 中的哪个类别。

5.1 引言

- **基于判别函数的分类器**
  - 采用已知类别标签的训练样本进行学习，获得**若干个代数界面**，这些界面将样本所在的空间分成**若干个相互不重叠的区域**。每个区域包含属于同一类的样本。
  - **表示界面的函数称为判别函数。**
  - 判别函数是分类器最常用的表述形式。

5.1 引言

- **判别函数：判定样本属于哪一个类别区域的函数**



图中，边界线可构造为一个判别函数

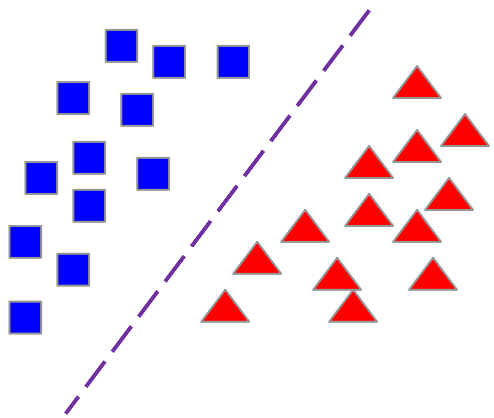
5.1 引言

- **基于判别函数的判别准则**
  - **对于c 类分类问题（线性机器）：**
    - 设 $g_i(\mathbf{x})$, $i = 1,2,\ldots, c$，表示每个类别对应的判别函数
    - 决策规则：
      - **如果 $g_i(\mathbf{x}) > g_j(\mathbf{x}), \forall j \neq i$，则 x 被分为第 ω_i 类。**
  - **对于两类分类问题**
    - **可以只用一个判别函数：** $g(\mathbf{x}) = g_1(\mathbf{x}) - g_2(\mathbf{x})$
    - 判别准则：$g(\mathbf{x}) > 0$，分为第一类；否则为第二类。

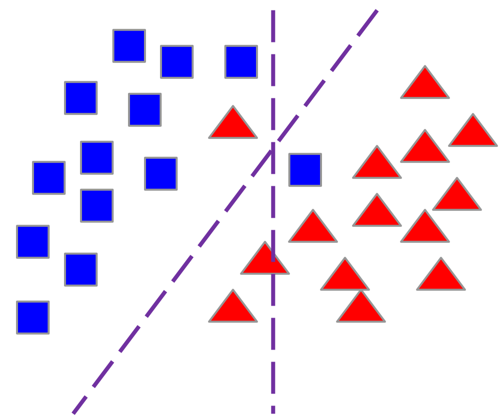
比如：$g(\mathbf{x}) = p(\omega_1 | \mathbf{x}) - p(\omega_2 | \mathbf{x})$, $g(\mathbf{x}) = \log \dfrac{p(\mathbf{x} | \omega_1)}{p(\mathbf{x} | \omega_2)} + \log \dfrac{p(\omega_1)}{p(\omega_2)}$

5.1 引言

- **线性可分**
  - 对于 n 个 d 维空间中的样本$\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n$，假定这些样本来自于两个类别 ω_1 或 ω_2。**如果存在一个线性判别函数能对这些样本正确地分类，则称这些样本是线性可分的；**否则是线性不可分的。



线性可分



线性不可分

5.2 线性判别函数与决策面

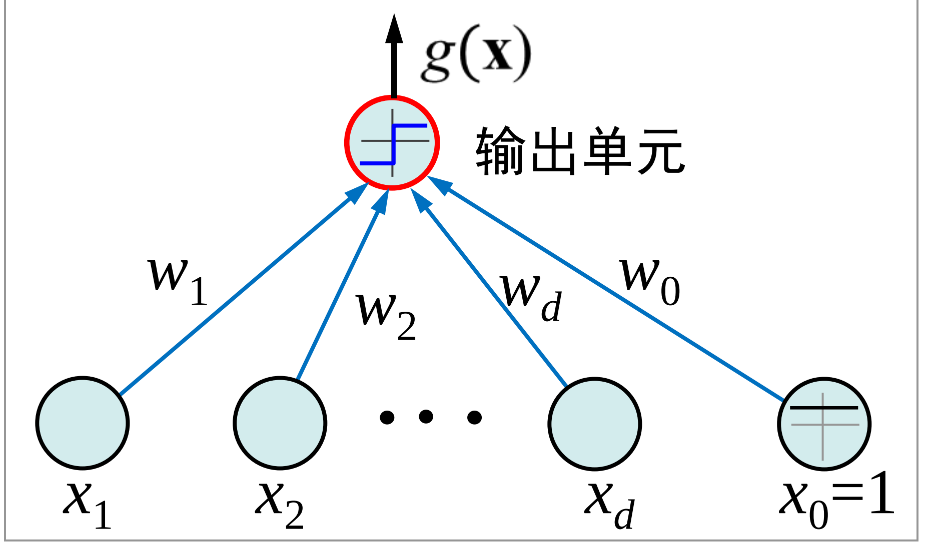
- 线性判别函数（基本形式）：

$$g(\mathbf{x}) = \mathbf{w}^T \mathbf{x} + w_0 \qquad (1)$$

权重向量　偏移(阈值)

- 两类情形的决策规则：

$$\begin{cases} \mathbf{x} \in \omega_1, & \text{if } g(\mathbf{x}) > 0 \\ \mathbf{x} \in \omega_2, & \text{if } g(\mathbf{x}) < 0 \\ \text{uncertain}, & \text{if } g(\mathbf{x}) = 0 \end{cases}$$

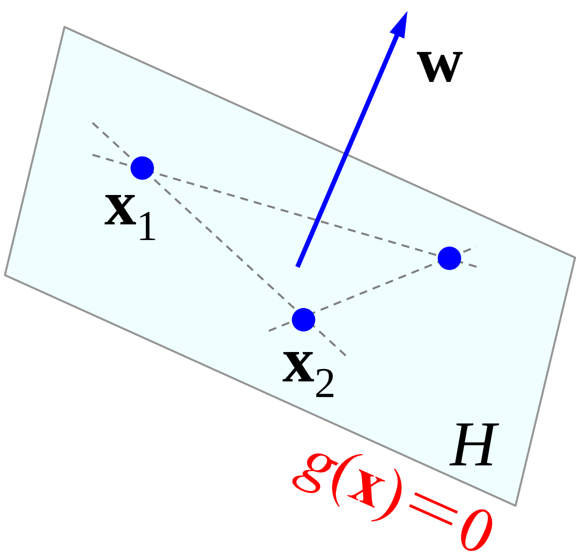


线性分类器（神经网络描述）

5.2 线性判别函数与决策面

- 两类情形的决策面
  - $g(\mathbf{x})=0$ 定义了一个决策面，它是类 ω_1 和 ω_2 的分界面。
  - $g(\mathbf{x})=0$ 是一个超平面，记为 H。位于该平面的任意向量与 $\mathbf{w}$ 垂直：
    - 如果 $\mathbf{x}_1$ 和 $\mathbf{x}_2$ 位于该超平面内，于是有：

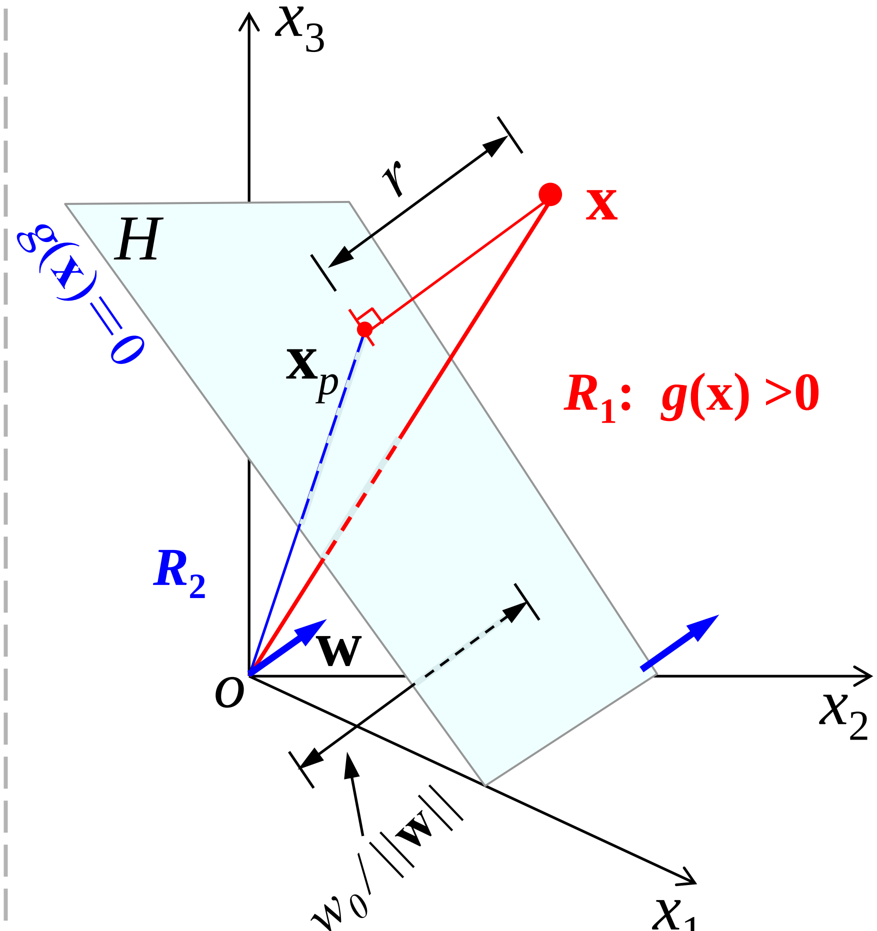
$$g(\mathbf{x_1}) - g(\mathbf{x_2}) = \mathbf{w}^T(\mathbf{x}_1\text{-}\mathbf{x}_2) = 0$$



5.2 线性判别函数与决策面

- 两类情形的决策面
  - 对于任意样本 $\mathbf{x}$，将其向决策面内投影，并写成两个向量之和：

$$\mathbf{x} = \mathbf{x}_p + r\frac{\mathbf{w}}{\|\mathbf{w}\|}$$



其中，$\mathbf{x}_p$为 $\mathbf{x}$ 在超平面 H上的投影，r 为点 $\mathbf{x}$ 到超平面 H 的代数距离。如果 $\mathbf{x}$ 在超平面正侧，则 $r>0$; 反之 $r<0$ 。

5.2 线性判别函数与决策面

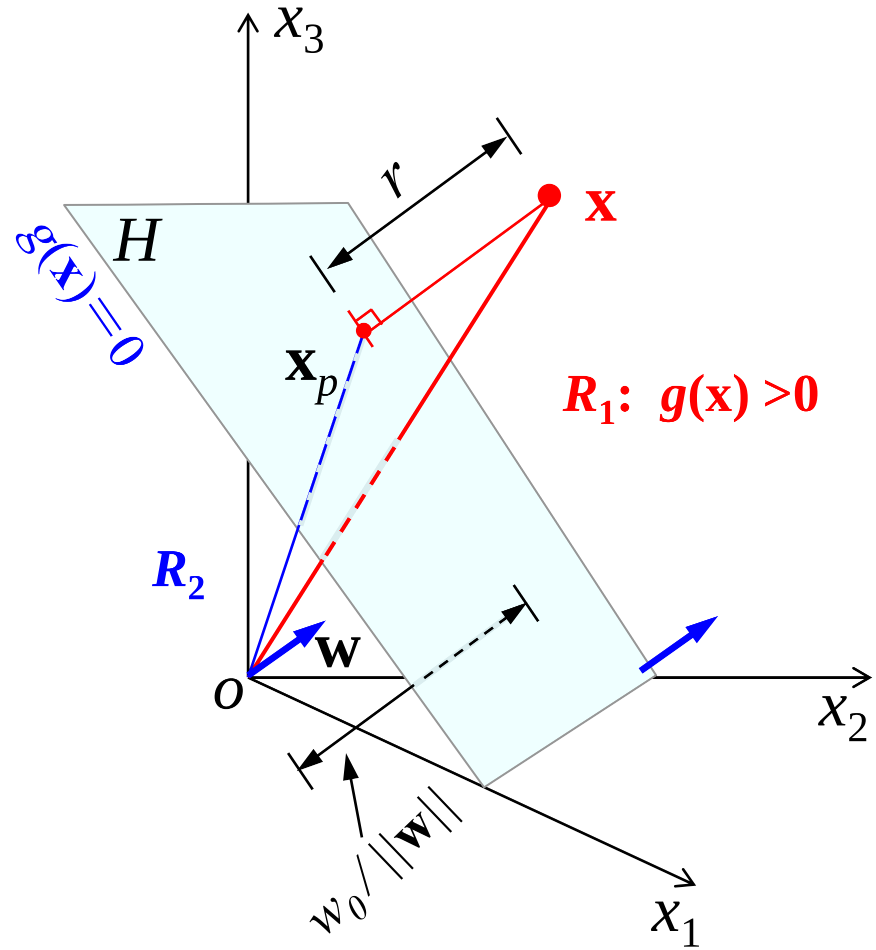
- 两类情形的决策面
  - 注意 $g(\mathbf{x}_p)=0$，于是有：

$$g(\mathbf{x})=\mathbf{w}^T\left(\mathbf{x}_p+r\frac{\mathbf{w}}{\|\mathbf{w}\|}\right)+w_0$$

$$=r\|\mathbf{w}\|$$

$$\Rightarrow \quad r=\frac{g(\mathbf{x})}{\|\mathbf{w}\|}$$

此外，可得坐标原点到超平面的距离为：$w_0/\|\mathbf{w}\|$

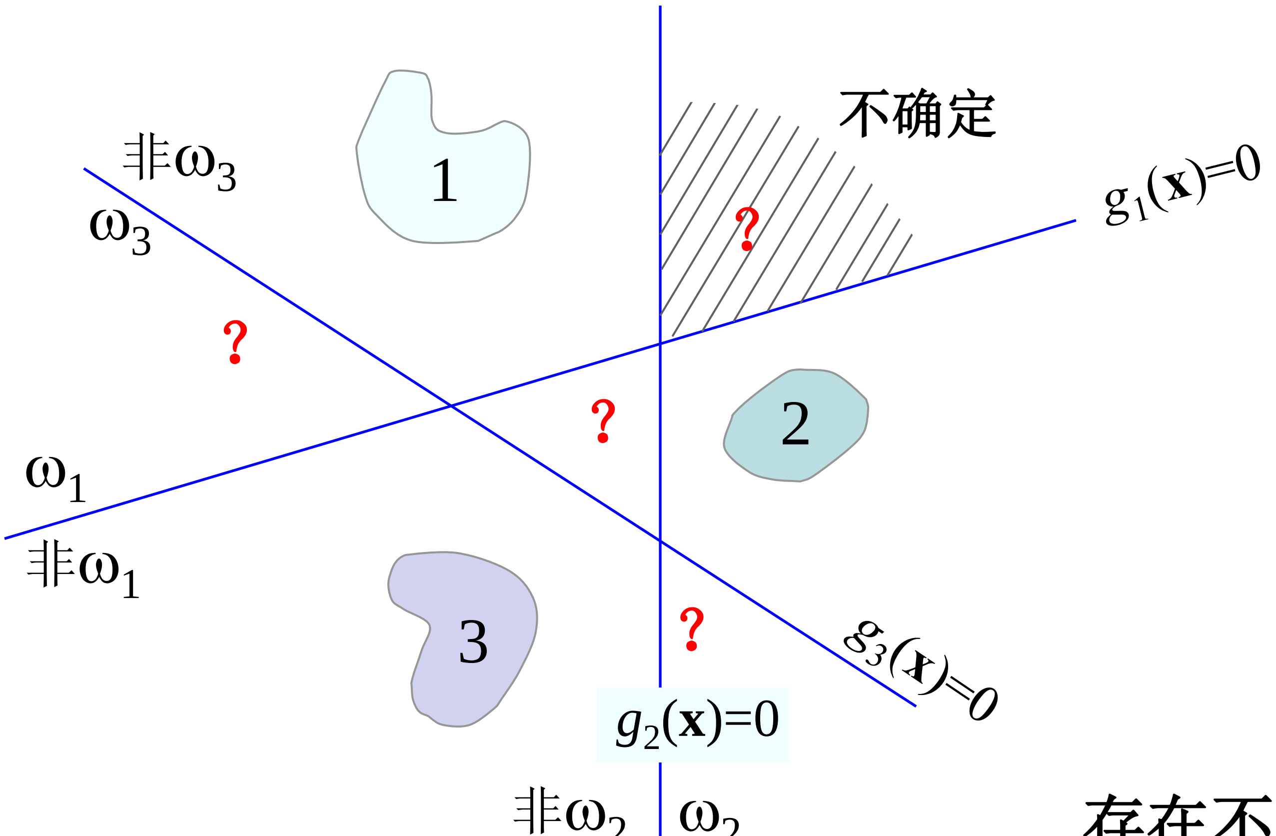


5.2 线性判别函数与决策面

- **多类情形** $(c>2)$一采用多个两类分类器
  - One-vs-all：逐一与所有的其它类进行配对，可以构造c 个两类分类器。
  - One-vs-one：两两（类-类）配对，可以构造$c(c-1)/2$个两类分类器。
  - **逐步一对多：**将 c 类问题逐步转化个两类分类问题。第一个分类器将其中一个类样本与其余各类样本分开，接着在**其余各类中**设计第二个分类器，直至仅剩下两个分类器为止。
- **多个两类分器：** $g_i(\mathbf{x})=\mathbf{w}_i^T\mathbf{x}+w_{i0},\quad i=1,2,\ldots,k$

5.2 线性判别函数与决策面

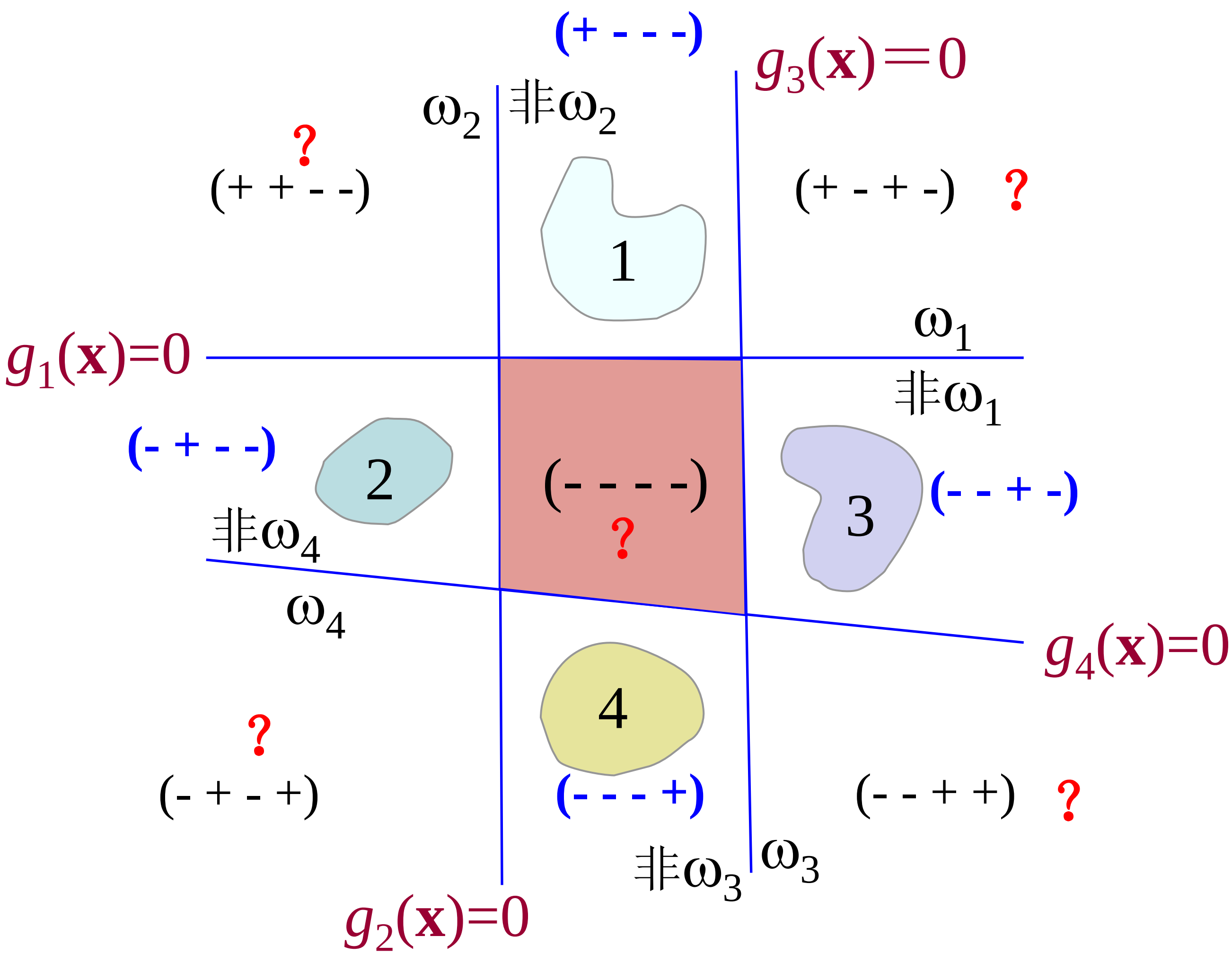
- 多类情形：One-vs-all



存在不确定区域

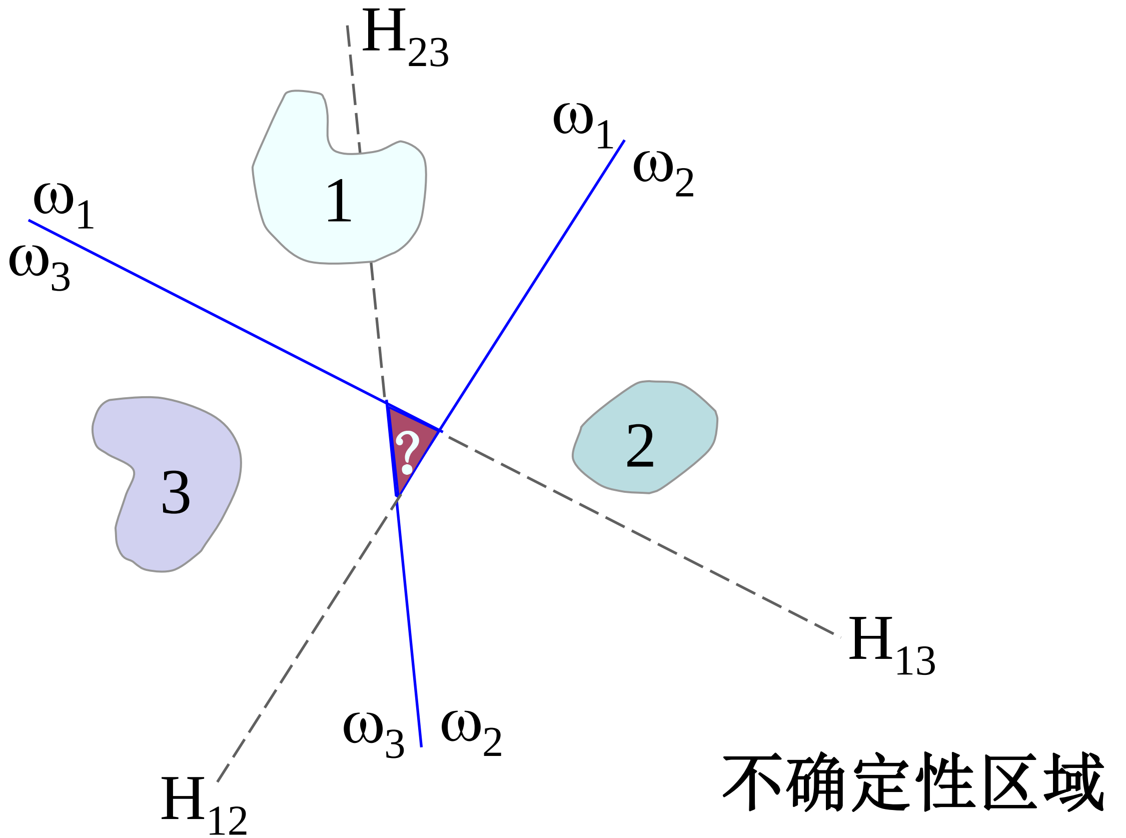
5.2 线性判别函数与决策面

- One vs all (4类):



5.2 线性判别函数与决策面

- 多类情形：one-vs-one



5.2 线性判别函数与决策面

- 多类情形—线性机器
  - 考虑one-vs-all情形，构建 c 个两类线性分类器：

$$g_i(\mathbf{x}) = \mathbf{w}_i^T \mathbf{x} + w_{i0}, \quad i = 1, 2, ..., c$$

  - 对样本点 $\mathbf{x}$，可以采用如下**更简洁的**决策规则：

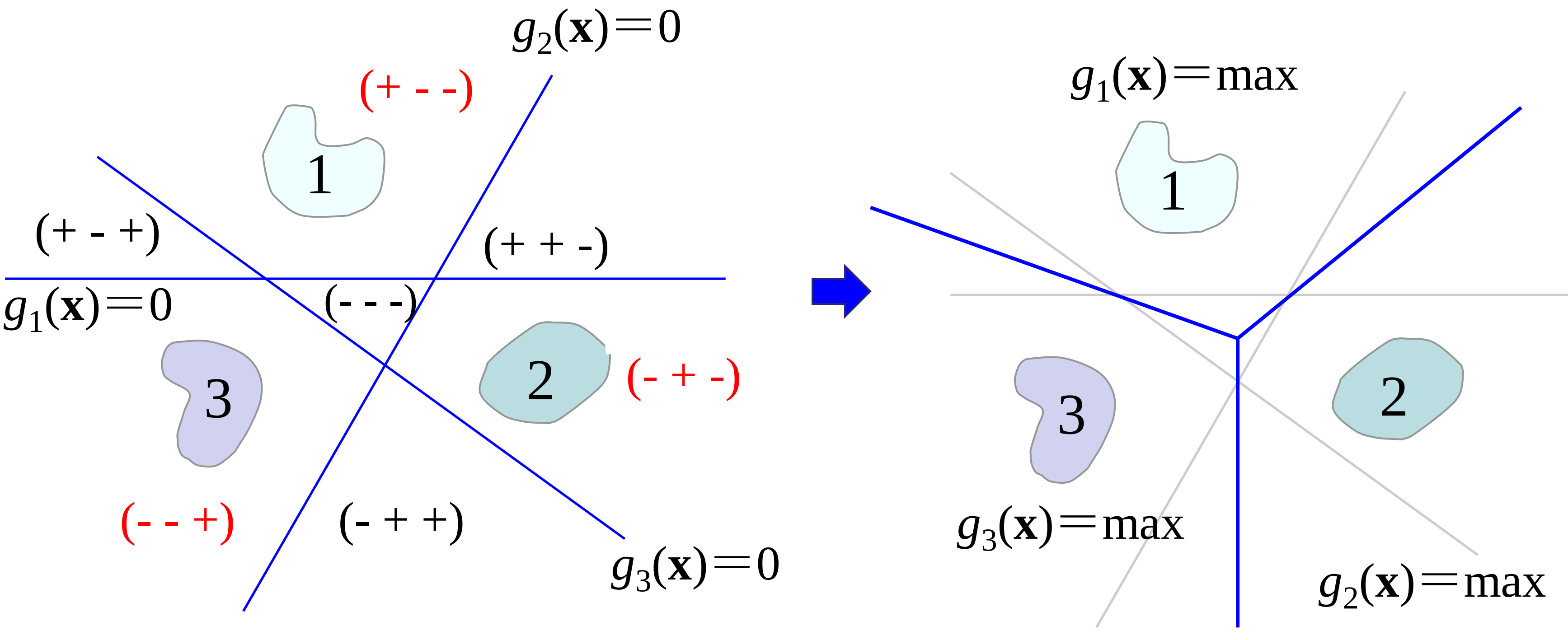
对 $j \neq i$, 如果 $g_i(\mathbf{x}) > g_j(\mathbf{x})$, $\mathbf{x}$ 则被分类 ω_i 类；否则不决策

$$\mathbf{x} \in \omega_i, \quad g_i(\mathbf{x}) = \max_{j=1,2,..c} g_j(\mathbf{x})$$

线性机器将样本空间分为 c 个**可以决策的区域$R_1,\ldots, R_c$**

5.2 线性判别函数与决策面

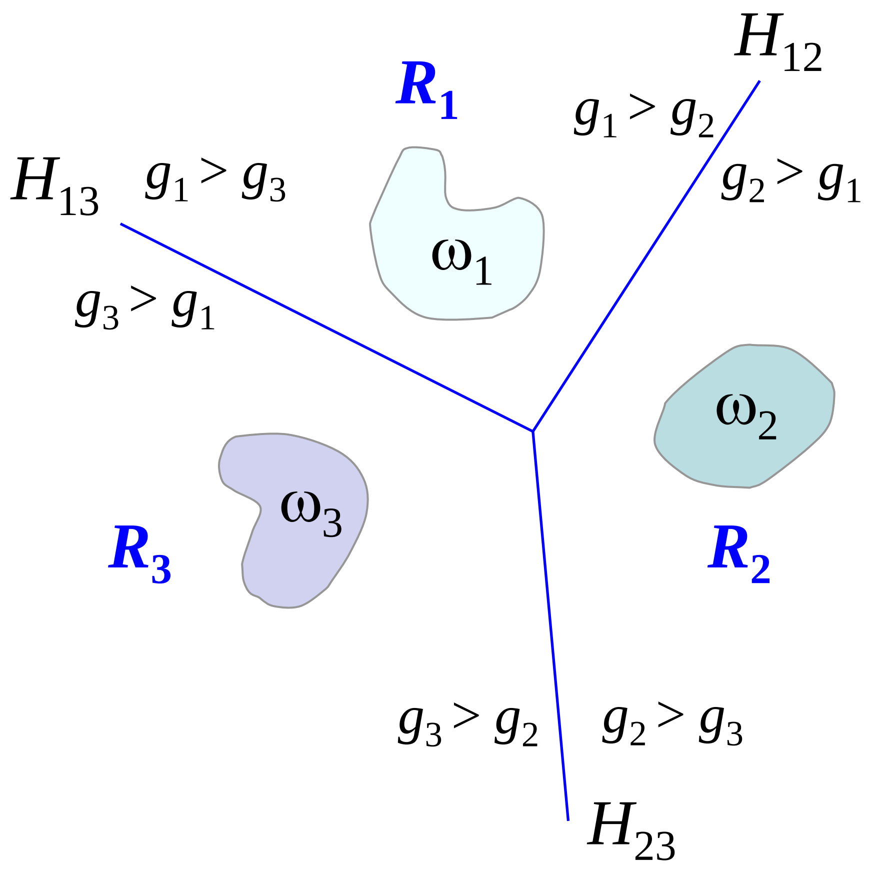
- 多类情形—线性机器：（变成“最大”决策）



线性机器将样本空间分为 c 个可以决策的区域$\boldsymbol{R}_1,\ldots,\boldsymbol{R}_c$

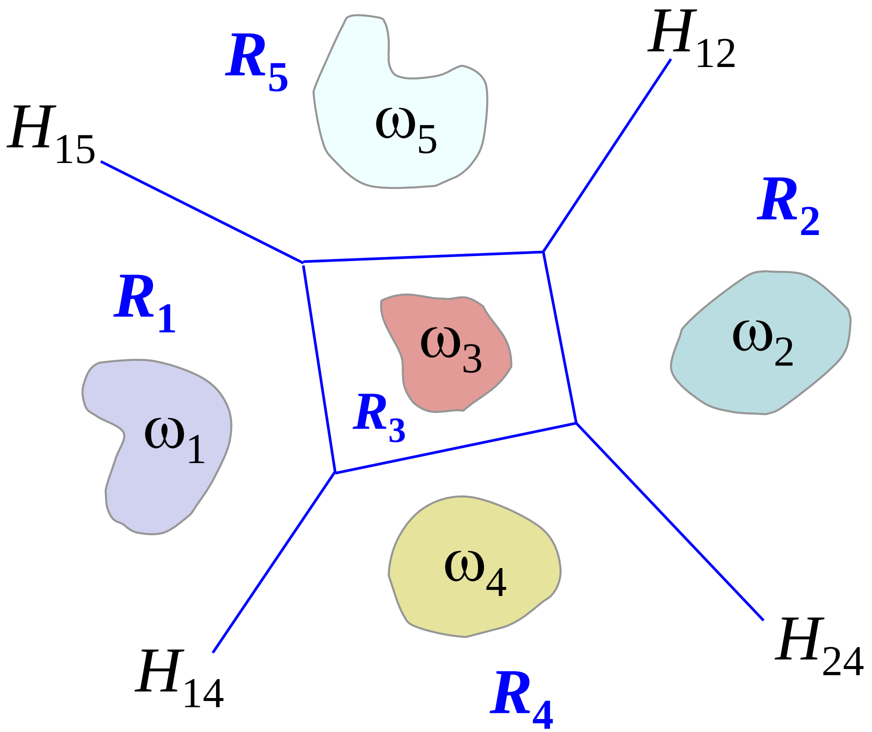
5.2 线性判别函数与决策面

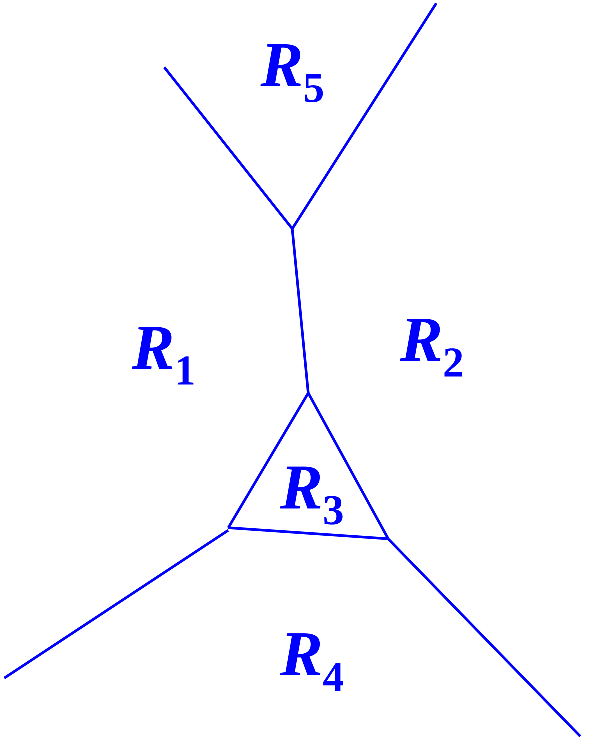
- 线性机器的决策面：一个例子 （One-vs-one）



5.2 线性判别函数与决策面

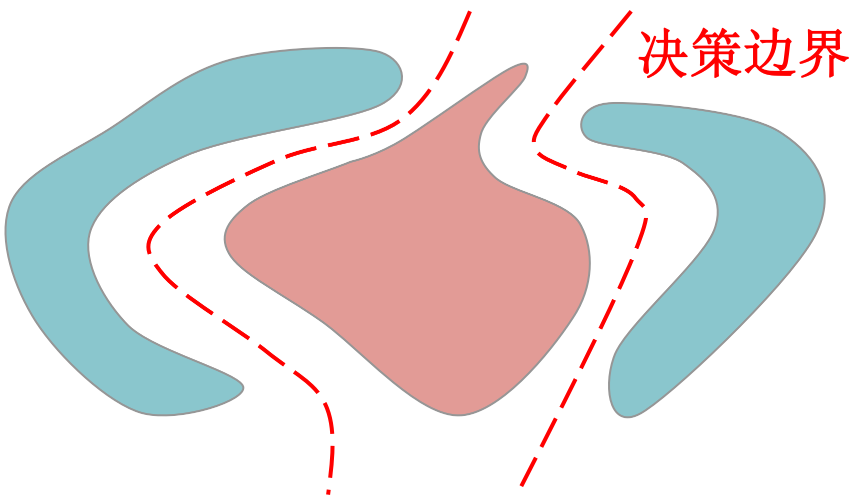
- 线性机器决策面：例子
  - 可以多达 $c(c\text{-}1)/2$ 个决策边界（有些可能可以删除）！





5.2 线性判别函数与决策面

- 线性决策面优缺点：
  - 所有的决策区域都是凸的　　　——便于分析
  - 所有的决策区域都是单通连的　——便于分析
  - 凸决策区域：限制分类器的灵活性和精度
  - 单通连区域：不利于**复杂分布数据**的分类（比如：分离的多模式分布）



5.3 广义线性判别函数

- **线性判别函数形式简单，计算方便，且已被充分研究。**
  人们期望将其推广至非线性判别函数。

一种有效的途径是将原来的数据点 $\mathbf{x}$ 通过一种适当的**非线性映射**将其映射为新的数据点 $\mathbf{y}$，从而**在新的数据空间内可以应用线性判别函数方法**。

- 线性情形

$$g(\mathbf{x}) = w_0 + \sum_{i=1}^{d} w_i x_i, \qquad \text{其中，} \mathbf{x} = [x_1, x_2, \ldots, x_d]^T$$

- 推广

  – 多项式判别函数，比如二次推广：

$$g(\mathbf{x}) = w_0 + \sum_{i=1}^{d} w_i x_i + \sum_{i=1}^{d}\sum_{j=1}^{d} w_{ij} x_i x_j$$

$$= \sum_{i=1}^{\hat{d}} a_i \, y_i(\mathbf{x}) \quad \text{（记为）}$$

$$\begin{cases} y_1(\mathbf{x}) = 1; \\ y_2(\mathbf{x}) = x_1; \\ y_3(\mathbf{x}) = x_2; \\ \quad \cdots \\ y_{d+1}(\mathbf{x}) = x_d; \\ y_{d+2}(\mathbf{x}) = x_1^2; \\ y_{d+3}(\mathbf{x}) = x_1 x_2; \\ \quad \cdots \\ y_{\frac{(d+1)(d+2)}{2}}(\mathbf{x}) = x_d^2; \end{cases}$$

1. 共有 $(d+1)(d+2)/2$ 个系数待估计 ($w_{ij} = w_{ji}$)
2. $g(\mathbf{x})=0$为决策面，它是一个二次超曲面

5.3 广义线性判别函数

- 一般情形

$$g(\mathbf{x}) = \sum_{i=1}^{\hat{d}} a_i \boxed{y_i(\mathbf{x})}$$

$y_i(\mathbf{x})$: 变换函数

令 $\mathbf{a} = [a_1, a_2, \ldots, a_{\hat{d}}]^T$， $\mathbf{y} = [y_1, y_2, \ldots, y_{\hat{d}}]^T$ 可以简写为：

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

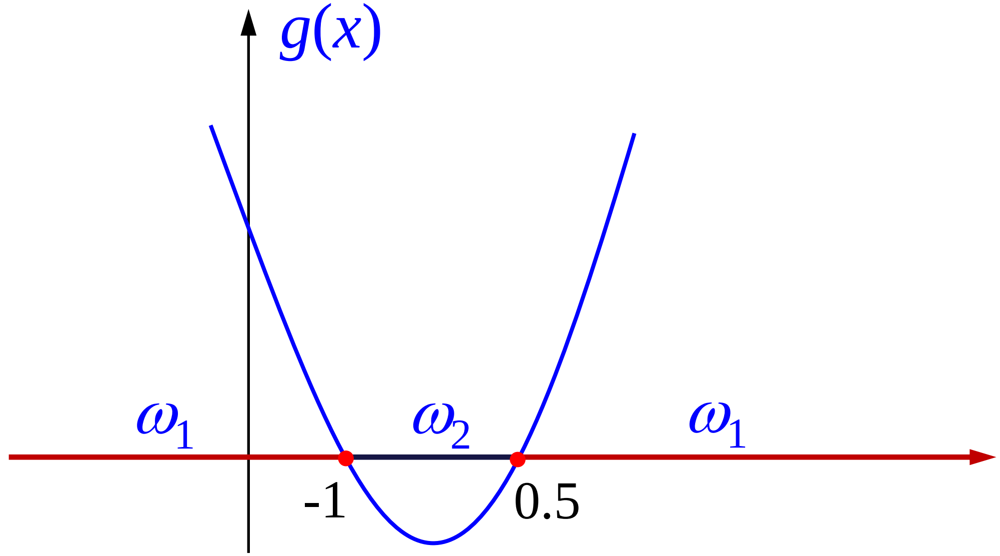
---

1. **a**为**广义权重向量**， **y 是经由 x 所变成的新数据点。**
2. 广义判别函数 $g(\mathbf{x})$ 对 **x** 而言是非线性的，对 **y** 是线性的。
3. $g(\mathbf{x})$ 对 **y** 是齐次的，意味着决策面通过新空间的坐标原点。且任意点 **y** 到决策面的代数距离为 $\mathbf{a}^T\mathbf{y} / \|\mathbf{a}\|$ 。
4. 当新空间的维数足够高时，$g(\mathbf{x})$ 可以逼近任意判别函数。
5. 但是，新空间的维数远远高于原始空间的维数 d 时，会造成维数灾难问题。

中国科学院大学 University of Chinese Academy of Sciences

5.3 广义线性判别函数

- 例子1
  - 设有一维样本空间X，我们期望如果 $x< -1$ 或者 $x>0.5$，则 x 属于第一类ω_1；如果 $-1<x<0.5$，则属于第二类ω_2，请设计一个判别函数 $g(x)$。



5.3 广义线性判别函数

- 例子1
  - 设有一维样本空间X，我们期望如果 $x< -1$ 或者 $x>0.5$ ，则 x 属于第一类ω_1；如果 $-1<x<0.5$，则属于第二类ω_2，请设计一个判别函数 $g(x)$。
  - 决策函数：$g(x) = (x-0.5)(x+1)$
  - 决策规则：$g(x)>0, x$ 属于ω_1；$g(x)<0, x$ 属于ω_2

$$\begin{aligned} g(x) &= (x-0.5)(x+1) \\ &= -0.5 + 0.5x + x^2 \\ &= a_1 + a_2 x + a_3 x^2 \end{aligned}$$

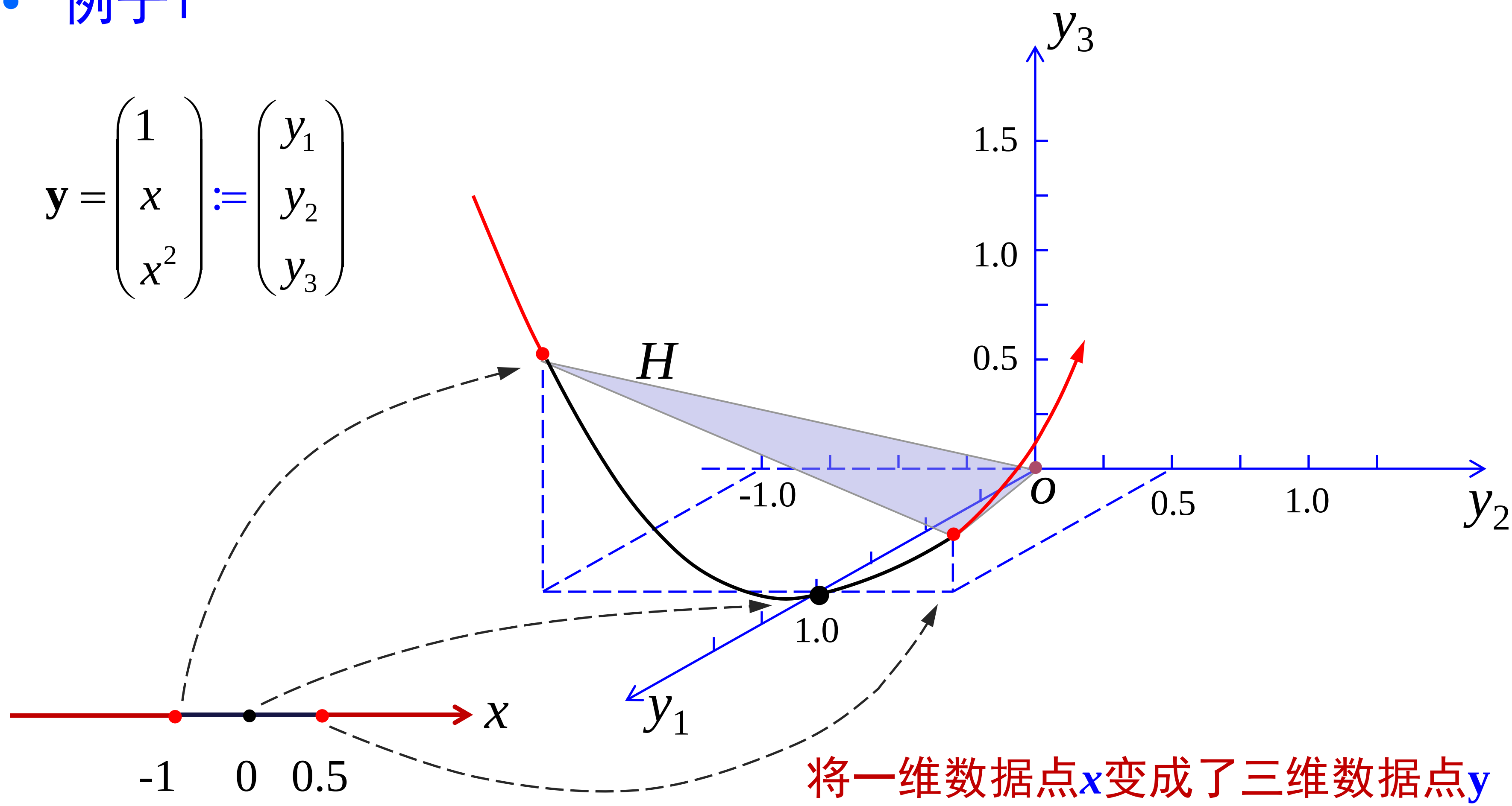
$\Longrightarrow$ 映射关系

$$\mathbf{y} = \begin{pmatrix} 1 \\ x \\ x^2 \end{pmatrix} := \begin{pmatrix} y_1 \\ y_2 \\ y_3 \end{pmatrix}$$

5.3 广义线性判别函数

- 例子1

$$\mathbf{y} = \begin{pmatrix} 1 \\ x \\ x^2 \end{pmatrix} := \begin{pmatrix} y_1 \\ y_2 \\ y_3 \end{pmatrix}$$



将一维数据点**x**变成了三维数据点**y**

- 例子2
  - 对线性判别函数采用齐次增广表示
    - 此时，增广样本向量 $\mathbf{y}$ 与增广权重向量 $\mathbf{a}$如下：

$$\mathbf{y}=\begin{pmatrix}1\\ \mathbf{x}\end{pmatrix}=\begin{bmatrix}1 & x_1 & \cdots & x_d\end{bmatrix}^T,\quad \mathbf{a}=\begin{pmatrix}w_0\\ \mathbf{w}\end{pmatrix}=\begin{bmatrix}w_0 & w_1 & \cdots & w_d\end{bmatrix}^T$$

  - 线性判别函数的齐次简化： $g(\mathbf{x})=\mathbf{w}^T\mathbf{x}+w_0=\mathbf{a}^T\mathbf{y}$
  - Y空间中任意一点 $\mathbf{y}$ 到 H 的距离为： $r=\dfrac{g(\mathbf{x})}{\|\mathbf{a}\|}=\dfrac{\mathbf{a}^T\mathbf{y}}{\|\mathbf{a}\|}$

线性齐次空间增加了一个维度，仍可保持欧氏距离不变，分类效果与原来的决策面相同。但分类面将过坐标原点，对于某些分析，将具有优势。

上述增广样本向量将在后面的讨论中经常使用。

5.4 感知准则函数

- **线性可分性**
  - 现有 n 个 样本: $\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_n$, 这些样本来自于两个类别 $\boldsymbol{\omega}_1$ 或 $\boldsymbol{\omega}_2$。
  - 任务：寻找一个线性判别函数 $g(\mathbf{x}) = \mathbf{a}^T\mathbf{y}$，使对这 n 个样本的错分概率最小。
  - 如果存在一个权向量 $\mathbf{a}$，对所有 $\mathbf{y}\in \boldsymbol{\omega}_1$，均有 $\mathbf{a}^T\mathbf{y} > 0$，且对所有 $\mathbf{y}\in \boldsymbol{\omega}_2$，均有 $\mathbf{a}^T\mathbf{y}<0$，则这组样本集为线性可分的；否则为线性不可分的。**（广义判别函数意义下）**
- 本节考虑**“两类线性可分”**情形

5.4 感知准则函数

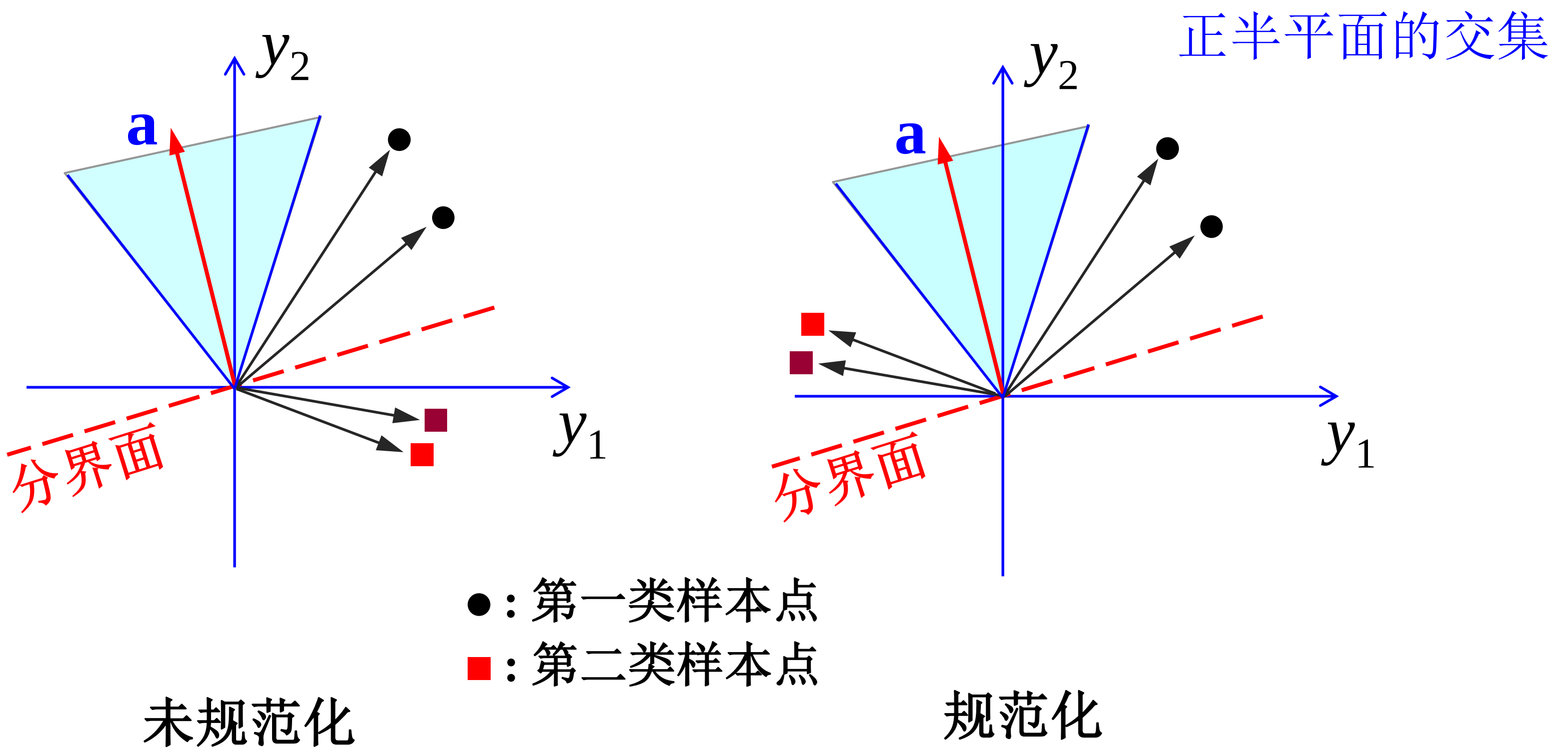
- **样本规范化**
  - 如果样本集是线性可分的，将属于 ω_2 的所有样本由 **y变成 –y，对所有 n 样本，将得到 $\mathbf{a}^T\mathbf{y}>0$** 。
  - 经过上述处理之后，在训练的过程中就不必考虑原来的样本类别。这一操作过程称为对样本的**规范化**(normalization)处理。
  - 规范化增广样本：首先将所有样本写成齐次坐标形式，然后将属于 ω_2 的所有样本由 **y变成 –y。**
  - **后续讲解**主要将集中于"**规范化增广样本**"。"增广"是指"齐次坐标表示"的含义，即 $\mathbf{y} = (\mathbf{x}^T,1)^T \in R^{d+1}$。

5.4 感知准则函数－两类可分情形

- **解区与解向量**
  - 在线性可分的情形下，满足 $\mathbf{a}^T\mathbf{y}_i > 0,\ i = 1, 2, \ldots, n$ 的权向量 $\mathbf{a}$ 称为**解向量**。
  - 权向量 $\mathbf{a}$ 可以理解为权空间中的一点，每个样本 $\mathbf{y}_i$ 对 $\mathbf{a}$ 的位置均可能起到限制作用，即要求 $\mathbf{a}^T\mathbf{y}_i > 0$。
  - 任何一个样本点 $\mathbf{y}_i$ 均可以确定一个超平面 H_i ：$\mathbf{a}^T\mathbf{y}_i = 0$，其法向量为 $\mathbf{y}_i$。如果解向量 $\mathbf{a}^*$存在，它必定在 H_i 的正侧，因为只有在正侧才能满足 $(\mathbf{a}^*)^T\mathbf{y}_i > 0$。
  - 按上述方法，n 个样本将产生 n 个超平面。每个超平面将空间分成两个半空间。**如果解向量存在，它必定在所有这些正半空间的交集区域内。**这个区域内的所有向量均是一个可行的解向量 $\mathbf{a}^*$。

5.3 感知准则函数—两类可分情形

- **解区与解向量**



给定一个可行的 **a**, 即可得到一个分界面

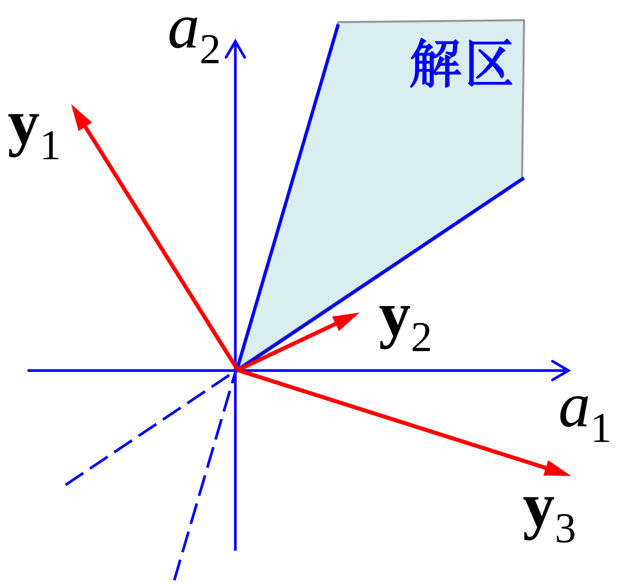
5.4 感知准则函数

- **限制解区**
  - 可行的解向量不是唯一的，有无穷多个。
  - 经验：**越靠近区域中间的解向量**，越能对新的样本正确分类
  - 可以引入一些条件来限制解空间
    - 比如：寻找一个单位长度的解向量 $\mathbf{a}$，能最大化样本到分界面的最小距离
    - 比如：寻找一个最小长度的解向量 $\mathbf{a}$，使$\mathbf{a}^T\mathbf{y}_i \geq b > 0$。此时可以将 b 称为 间隔 (margin)。
      - 解更加可靠，推广性更强
      - 防止算法收敛到解区的边界

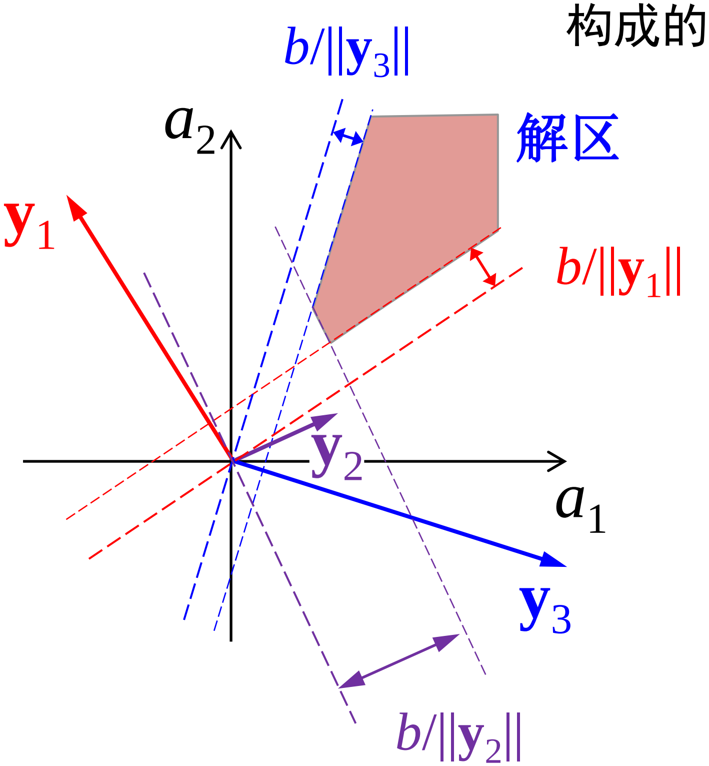
5.4 感知准则函数

- **限制解区：移动一个间隔**

将正半空间向外推一定的距离后构成的交集



b=0,不考虑margin



b>0,考虑margin

$$\because \ r = \frac{g(\mathbf{x})}{\|\mathbf{w}\|}$$

5.4 感知准则函数

- **感知准则函数**
  - 任务：设有一组样本$\mathbf{y}_1$，$\mathbf{y}_2$，…，$\mathbf{y}_n$，各样本均规范化表示。我们的目的是要寻找一个解向量 **a**，使

$$\mathbf{a}^T\mathbf{y}_i > 0, \quad i=1,2,\ldots,n$$

  - 在线性可分情形下，满足上述不等式的 **a** 是无穷多的，因此需要引入一个准则。

5.4 感知准则函数

- **感知准则函数—基本思想**
  - 考虑如下准则函数：

$$J_p(\mathbf{a}) = \sum_{\mathbf{y}\in Y} (-\mathbf{a}^T\mathbf{y}),\ \text{其中，}Y\text{为错分样本集合}$$

  - 当 y 被错分时， $\mathbf{a}^T\mathbf{y} \le 0$，则 $-\mathbf{a}^T\mathbf{y} \ge 0$。因此 $J_p(\mathbf{a})$ 总是大于等于0。在可分情形下，当且仅当 Y 为空集时 $J_p(\mathbf{a})$ 将等于零，这时将不存在错分样本。
  - 因此，目标是最小化 $J_p(\mathbf{a})$: $\min_{\mathbf{a}} J_p(\mathbf{a})$
  - 这即是Frank Rosenblatt 于50年代提出的感知学习机思想。

5.4 感知准则函数

- **感知准则函数**
  - 考察 $J_p(\mathbf{a})$ 对 $\mathbf{a}$ 的导数： $\dfrac{\partial J_p(\mathbf{a})}{\partial \mathbf{a}} = -\sum_{\mathbf{y}\in Y} \mathbf{y}$
  - 因此，根据梯度下降法，有如下更新准则：

$$\mathbf{a}_{k+1} = \mathbf{a}_k + \eta_k \sum_{\mathbf{y}\in Y_k} \mathbf{y}$$

这里，$\mathbf{a}_{k+1}$是当前迭代的结果， $\mathbf{a}_k$ 是前一次迭代的结果，Y_k 是被 $\mathbf{a}_k$ 错分的样本集合， η_k 为步长因子（更新动力因子，学习率）。

5.4 感知准则函数

- **感知准则函数—算法**

Batch Perceptron—基本算法

begin initialize: $\mathbf{a}$, η, certain θ (small value), $k=0$

  do $k \leftarrow k+1$

    $\mathbf{a} = \mathbf{a} + \eta_k \sum_{\mathbf{y} \in Y(k)} \mathbf{y}$ // $Y(k) = Y_k$

  until $|\eta_k \sum \mathbf{y}| < \theta$, $\mathbf{y} \in Y_k$ // 一个较松的停止条件

  return $\mathbf{a}$

end

1. 之所以称为“batch perception”是因为在迭代过程中同时考虑多个样本。
2. 计算复杂度低，能以较快的速度收敛到极小值点

可变增量批处理修正方法（另一种表述形式）

Batch Variable-Increment Perceptron

begin initialize: $\mathbf{a}$, η_0, k=0

  do $k \leftarrow k$+1 (mod n)

    Y_k={}, j = 0

    do $j \leftarrow j + 1$

      if $\mathbf{y}_j$ is misclassified, then append $\mathbf{y}_j$ to Y_k

    until $j = n$

    $\mathbf{a} = \mathbf{a} + \eta_k \sum_{\mathbf{y} \in Y(k)} \mathbf{y}$ //发现所有错分，然后再修正

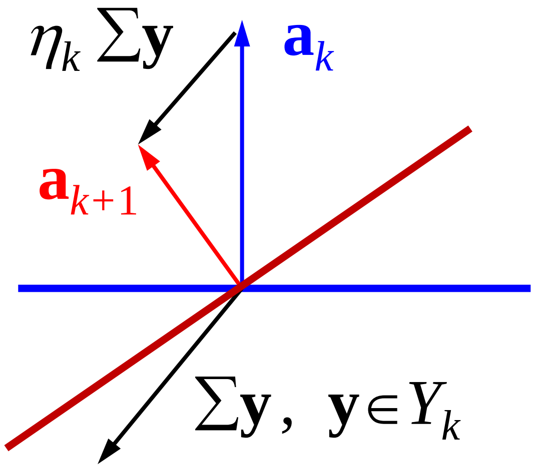
  until Y_k={} //直到所有样本均正确分类

  return $\mathbf{a}$

end

5.4 感知准则函数

- **感知准则函数—算法解释**
  - 由于所有被 $\mathbf{a}_k$ 错分的样本必然位于以为法向量的超平面的负侧，所以这些样本的和也必然在该侧。
  - $\mathbf{a}_{k+1}$在更新的过程中，会向错分类样本之和靠近，因而朝着有利的方向移动。**一旦这些错分样本点穿过超平面，就正确分类了。**
  - 对于线性可分的样本集，算法可以在有限步内找到最优解。
  - 收敛速度取决于初始权向量和步长



5.4 感知准则函数

- **固定增量单样本修正方法**
  - 每次迭代只考虑一个错分样本 $\mathbf{y}^k$，梯度下降法可以写成：$\mathbf{a}_{k+1} = \mathbf{a}_k + \eta_k \mathbf{y}^k$。考虑固定增量，即令 $\eta_k = 1$：

Fixed-Increment Single-Sample Perceptron

```
begin initialize: a, k=0
    do k←k+1 (mod n)
        if y^k is misclassified by a, then a = a + y^k
    until  all patterns properly classified
    return a
end
```

“固定增量”并不改变分类决策，相当于将样本作了一个 $1/\eta_k$ 的缩放。

5.4 感知准则函数

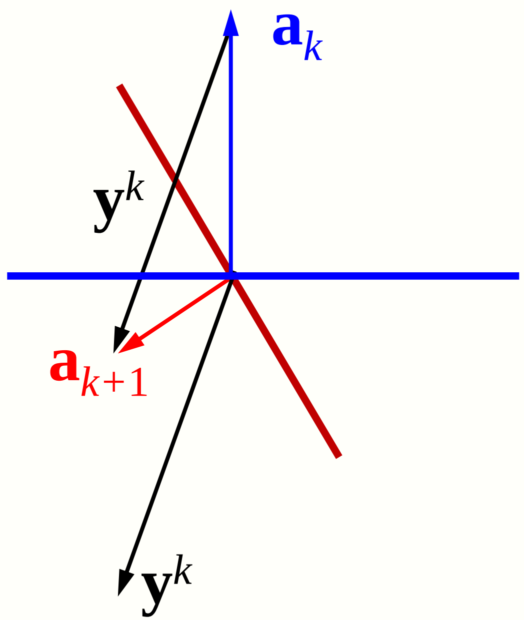
- **可变增量单样本修正方法**
  - 梯度下降法可以写成：$\mathbf{a}_{k+1} = \mathbf{a}_k + \eta_k \mathbf{y}^k$

Variable-Increment Perceptron with Margin

```
begin initialize: a, margin b, η₀ , k=0
    do k ← k+1 (mod n)
        if aᵀyᵏ ≤ b, then a = a + η_k yᵏ  //发现一个错分，马上修正
    until aᵀyᵏ > b for all k
    return a
end
```

5.4 感知准则函数

- **固定增量单样本修正－算法解释**
  - 梯度下降：$\mathbf{a}_{k+1} = \mathbf{a}_k + \mathbf{y}^k$。
  - 权向量总能得到修正：
    - 由于a_k将 y^k 分错，所以 $(a_k)^T y^k \le 0$。此时，a_k不在 y^k 确定的超平面$(a_k)^T y^k = 0$的正侧。
    - 若将 y^k 加到a_k上，则 a_k 将向该超平面的正侧移动，可能会穿过这个超平面（即正确分类）。



$(\mathbf{a}_{k+1})^T \mathbf{y}^k = (\mathbf{a}_k)^T \mathbf{y}^k + ||\mathbf{y}^k||^2$

$(\mathbf{a}_{k+1})^T \mathbf{y}^k$在原来的基础上增加了一个正数：$||\mathbf{y}^k||^2$

5.4 感知准则函数

- **感知准则函数－算法收敛性**
  - 不失一般性，以**固定增量单样本修正方法为例来说明算法的收敛性。**
    - 对于权向量 $\mathbf{a}_k$，如果错分某样本，则将得到一次修正。由于在分错样本时 $\mathbf{a}_k$ 才得到修正，不妨假定只考虑由错分样本组成的序列。即是说，每次都只需利用一个分错样本来更正权向量。
    - 记错分样本序列为 $\mathbf{y}^1, \mathbf{y}^2, \ldots, \mathbf{y}^k \ldots$。考虑此情形的算法收敛性问题。

5.4 感知准则函数

- **感知准则函数—收敛性定理**
  - 在样本线性可分的情形下，固定增量单样本权向量修正方法收敛，并可得到一个可行解。
- **证明思路**
  - 设 $\mathbf{a}$ 是一个解向量，只要证明 $\|\mathbf{a}_{k+1}-\mathbf{a}\|<\|\mathbf{a}_k-\mathbf{a}\|$即可。

5.4 感知准则函数

- **证明**
  - 设 $\mathbf{a}$ 是一个解向量，对于任意一个正的标量α， $\alpha\mathbf{a}$也为一个可行解，于是有:

$$\mathbf{a}_{k+1}-\alpha\mathbf{a}=(\mathbf{a}_k-\alpha\mathbf{a})+\mathbf{y}^k$$

$$\|\mathbf{a}_{k+1}-\alpha\mathbf{a}\|^2=\|\mathbf{a}_k-\alpha\mathbf{a}\|^2+2(\mathbf{a}_k-\alpha\mathbf{a})^T\mathbf{y}^k+\|\mathbf{y}^k\|^2$$

由于 $\mathbf{y}^k$ 被错分，有 $(\mathbf{a}_k)^T\mathbf{y}^k \le 0$。但 $\mathbf{a}^T\mathbf{y}^k > 0$, 于是:

$$\|\mathbf{a}_{k+1}-\alpha\mathbf{a}\|^2 \le \|\mathbf{a}_k-\alpha\mathbf{a}\|^2-2\alpha\mathbf{a}^T\mathbf{y}^k+\|\mathbf{y}^k\|^2$$

因此，寻找一个合适的α，满足$\|\mathbf{a}_{k+1}-\alpha\mathbf{a}\|^2 \le \|\mathbf{a}_k-\alpha\mathbf{a}\|^2$即可。

- 证明 (续)
  - 令 $\beta^2 = \max_{i=1,..,n} \|\mathbf{y}_i\|^2, \quad \gamma = \min_i \mathbf{a}^T\mathbf{y}_i$

$$\|\mathbf{a}_{k+1} - \alpha\mathbf{a}\|^2 \le \|\mathbf{a}_k - \alpha\mathbf{a}\|^2 - 2\alpha\gamma + \beta^2$$

令 $\alpha = \beta^2 / \gamma$

$$\|\mathbf{a}_{k+1} - \alpha\mathbf{a}\|^2 \le \|\mathbf{a}_k - \alpha\mathbf{a}\|^2 - \beta^2$$

$$\|\mathbf{a}_{k+1} - \alpha\mathbf{a}\|^2 < \|\mathbf{a}_k - \alpha\mathbf{a}\|^2$$

因此，每次迭代，当前解离可行解越来越近。经过 k+1次迭代后:

$$\|\mathbf{a}_{k+1} - \alpha\mathbf{a}\|^2 \le \|\mathbf{a}_1 - \alpha\mathbf{a}\|^2 - k\beta^2$$

由于 $\|\mathbf{a}_{k+1} - \alpha\mathbf{a}\|$ 总是非负的，所以至多经过如下次更正即可：

$$k_0 = \|\mathbf{a}_1 - \alpha\mathbf{a}\|^2 / \beta^2$$

5.5 松驰方法

- **学习准则**
  - 在感知函数准则中，目标函数中采用了$-\mathbf{a}^T\mathbf{y}$ 的形式。实际上有很多其它准则也可以用于感知函数的学习。
  - **线性准则：** $J_p(\mathbf{a}) = \sum_{\mathbf{y}\in Y}(-\mathbf{a}^T\mathbf{y}),$
  - **平方准则：** $J_q(\mathbf{a}) = \sum_{\mathbf{y}\in Y}(\mathbf{a}^T\mathbf{y})^2,$

    Y为错分样本集合
  - **松驰准则：** $J_r(\mathbf{a}) = \frac{1}{2}\sum_{\mathbf{y}\in Y}\frac{(\mathbf{a}^T\mathbf{y}-b)^2}{\|\mathbf{y}\|^2},$

    Y为 $\mathbf{a}^T\mathbf{y}\le b$ 的样本集合

5.5 松驰方法

- **学习准则**
  - 线性准则的目标函数是分段线性的，因此其梯度是不连续的。
  - 平方准则的梯度是连续的，但目标函数过于平滑，收敛速度很慢（达到目标函数为零的区域的路径很平缓）。同时，目标函数过于受到最长样本的影响。
  - $J_r(\mathrm{a})$ 则避免了这些缺点。
  - $J_r(\mathrm{a})$ 最终达到零。此时对所有的 $\mathbf{y}$， $\mathbf{a}^T\mathbf{y} > b$，则意味着集合 Y 是空集。

5.5 松驰方法

- **学习：训练**
  - **梯度：**
$$\frac{\partial J_r(\mathbf{a})}{\partial \mathbf{a}} = \sum_{\mathbf{y}\in Y} \frac{\mathbf{a}^T\mathbf{y}-b}{\|\mathbf{y}\|^2}\mathbf{y}$$
  - **梯度下降准则：** $\mathbf{a}_{k+1} = \mathbf{a}_k - \eta_k \sum_{\mathbf{y}\in Y} \frac{\mathbf{a}^T\mathbf{y}-b}{\|\mathbf{y}\|^2}\mathbf{y}$

5.5 松弛方法

Batch Relaxation with Margin

begin initialize: $\mathbf{a}$, b η_0, k=0
    do $k \leftarrow k+1 \pmod{n}$
        $Y_k = \{\}$, $j = 0$
        do $j \leftarrow j + 1$
            if $\mathbf{a}^T\mathbf{y}_j \le b$, then append $\mathbf{y}_j$ to Y_k //如果错分
        until $j = n$
        $\mathbf{a}_{k+1} = \mathbf{a}_k - \eta_k \sum_{\mathbf{y} \in Y} \left((\mathbf{a}^T\mathbf{y} - b) / \|\mathbf{y}\|^2 \right) \cdot \mathbf{y}$,
    until $Y_k = \{\}$
    return $\mathbf{a}$
end

5.5 松驰方法

- **单样本松驰算法**
  - 可假定序列中每个样本都为错分样本（此时为 $\mathbf{a}^T\mathbf{y} \le b$ ），记为： $\mathbf{y}^1, \mathbf{y}^2, \ldots, \mathbf{y}^k \ldots$。
  - 仍可假定在梯度下降过程中，更新步长 η_k 为常数，且 $\eta_k = \eta$ 。
  - 更新准则：

$$\mathbf{a}_{k+1} = \mathbf{a}_k - \eta_k \frac{\mathbf{a}_k^T \mathbf{y}^k - b}{\|\mathbf{y}^k\|^2} \mathbf{y}^k = \mathbf{a}_k - \eta_k \frac{\mathbf{a}_k^T \mathbf{y}^k - b}{\|\mathbf{y}^k\|} \frac{\mathbf{y}^k}{\|\mathbf{y}^k\|}$$

5.5 松驰方法

- **单样本松驰算法**

Single Sample Relaxation with Margin

begin initialize: $\mathbf{a}$, margin b, η, k=0
 do $k \leftarrow k+1 \pmod{n}$
  if $\mathbf{a}^T\mathbf{y}^k \le b$, then $\mathbf{a} = \mathbf{a} - \left(\eta\left(\mathbf{a}^T\mathbf{y}^k - b\right)/\|\mathbf{y}^k\|^2\right)\cdot\mathbf{y}^k$
 until $\mathbf{a}^T\mathbf{y}^k > b$ for all $\mathbf{y}^k$
 return $\mathbf{a}$
end

5.5 松驰方法

- **几何解释**

点 $\mathbf{a}_k$ 到超平面 $\mathbf{a}^T\mathbf{y}^k = b$ 的距离：

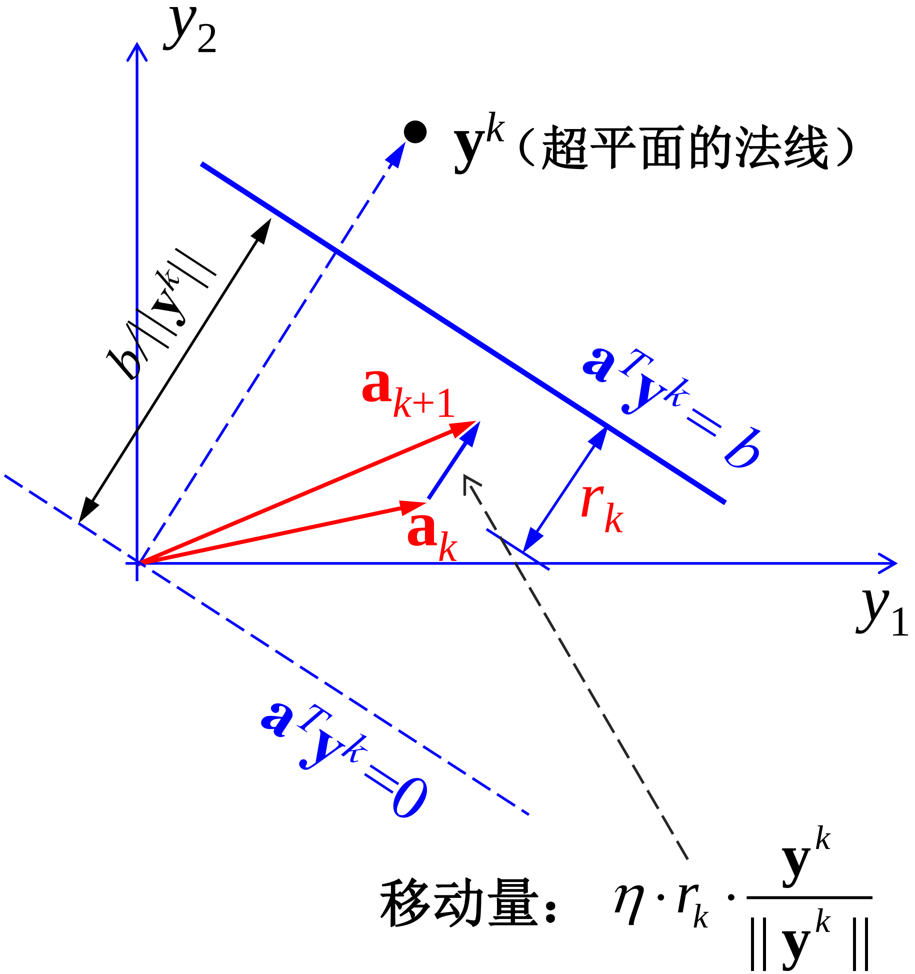
$$r_k = \frac{b - \mathbf{a}_k^T \mathbf{y}^k}{\|\mathbf{y}^k\|}$$

点 $\mathbf{a}_k$ 沿着单位向量方向 $\mathbf{y}^k/\|\mathbf{y}^k\|$ 移动其 ηr_k 倍的距离，得到新的 $\mathbf{a}_{k+1}$。

根据更新准则，有：

$$(\mathbf{a}_{k+1})^T\mathbf{y}^k - b = (1-\eta)\,((\mathbf{a}_k)^T\mathbf{y}^k - b)$$

(离超平面更近了)



5.5 松弛方法

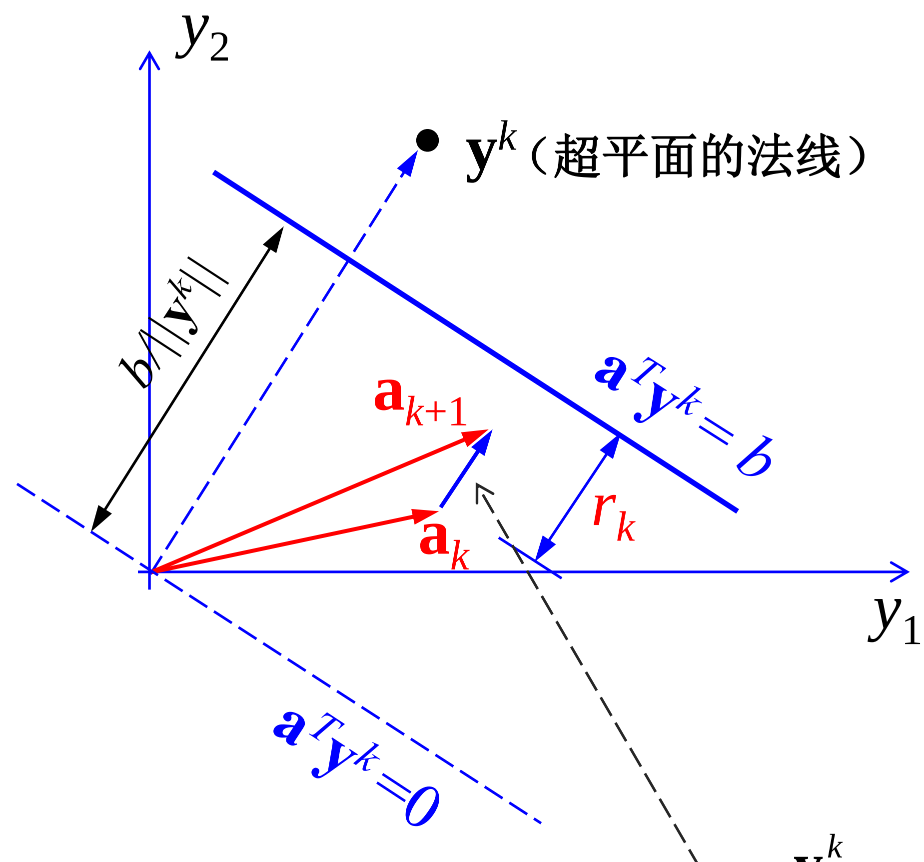
- **几何解释**

如果 $\eta=1$，$\mathbf{a}_k$ 将直接移动到该超平面。由于 $\mathbf{y}^k$ 被错分引起的压力 “$\mathbf{a}^T\mathbf{y}^k < b$” 被释放，所以称为松弛方法。

如果 $\eta < 1$，仍有 $(\mathbf{a}_{k+1})^T\mathbf{y}^k < b$。此时尽管 $\mathbf{y}^k$ 仍被错分，但 $\mathbf{a}_{k+1}$ 比 $\mathbf{a}_k$ 更好。因为 $\mathbf{a}_{k+1}$ 离跨过超平面更近。此时，称为软松弛。

如果 $\eta > 1$，$\mathbf{a}_{k+1}$ 将跨过超平面，$\mathbf{y}^k$ 将被正确分类。称为超松弛。

实际中取 $0 < \eta < 2$。



移动量：$\eta \cdot r_k \cdot \dfrac{\mathbf{y}^k}{\|\mathbf{y}^k\|}$

5.5 松驰方法

- **收敛性—考虑单样本松驰法**
  - 设 $\mathbf{a}$ 是一个解向量，因此对任意 $\mathbf{y}_i$，有 $\mathbf{a}^T\mathbf{y}_i > b$，于是：

$$\|\mathbf{a}_{k+1}-\mathbf{a}\|^2 = \|\mathbf{a}_k-\mathbf{a}\|^2 - 2\eta\frac{b-\mathbf{a}_k^T\mathbf{y}^k}{\|\mathbf{y}^k\|^2}(\mathbf{a}-\mathbf{a}_k)^T\mathbf{y}^k + \eta^2\frac{(b-\mathbf{a}_k^T\mathbf{y}^k)^2}{\|\mathbf{y}^k\|^2}$$

$$(\mathbf{a}-\mathbf{a}_{k+1})^T\mathbf{y}^k = \mathbf{a}^T\mathbf{y}^k - \mathbf{a}_{k+1}^T\mathbf{y}^k > b-\mathbf{a}_{k+1}^T\mathbf{y}^k$$

$$\Rightarrow \quad \|\mathbf{a}_{k+1}-\mathbf{a}\|^2 < \|\mathbf{a}_k-\mathbf{a}\|^2 - \eta(2-\eta)\frac{(b-\mathbf{a}_k^T\mathbf{y}^k)^2}{\|\mathbf{y}^k\|^2}$$

$$\Rightarrow \quad \|\mathbf{a}_{k+1}-\mathbf{a}\|^2 < \|\mathbf{a}_k-\mathbf{a}\|^2$$

（由于$0<\eta<2$)

5.6 最小平方误差（MSE）准则函数

- **动机**
  - 对两类分问题，感知准则函数是寻找一个解向量 $\mathbf{a}$，对所有样本 $\mathbf{y}_i$，满足 $\mathbf{a}^T\mathbf{y}_i > 0$, i=1,2,…n。或者说，求解一个不等式组，使满足 $\mathbf{a}^T\mathbf{y}_i > 0$的数目最大，从而错分样本最少。
  - 现在将不等式改写为等式形式：

$$\mathbf{a}^T\mathbf{y}_i = b_i > 0$$

其中，b_i 是任意给定的正常数，通常取$b_i = 1$，或者 $b_i = n_j / n$。其中，n_j, j =1 or 2，为属于第 j 类样本的总数，且 $n_1 + n_2 = n$。

5.6 MSE 准则函数

- **方法**
  - 可得一个线性方程组（例如，齐次表示）：

$$\begin{pmatrix} y_{10} & y_{11} & \cdots & y_{1d} \\ y_{20} & y_{21} & \cdots & y_{2d} \\ \vdots & \vdots & & \vdots \\ \vdots & \vdots & & \vdots \\ y_{n0} & y_{n1} & \cdots & y_{nd} \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ \vdots \\ a_d \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ \vdots \\ b_d \end{pmatrix} \quad \text{or} \quad \mathbf{Ya} = \mathbf{b}$$

  - 如果 **Y** 可逆，则 $\mathbf{a} = \mathbf{Y}^{-1}\mathbf{b}$
  - 但通常情形下，$n >> d + 1$，因此，考虑定义一个误差向量：$\mathbf{e} = \mathbf{Ya} - \mathbf{b}$，并使误差向量最小。

5.6 MSE 准则函数

- **平方误差准则函数：** $J_s(\mathbf{a}) = \|\mathbf{e}\|^2 = \|\mathbf{Ya} - \mathbf{b}\|^2 = \sum_{i=1}^{n} (\mathbf{a}^T \mathbf{y}_i - b_i)^2$
  - 偏导数：$$\frac{\partial J_s(\mathbf{a})}{\partial \mathbf{a}} = \sum_{i=1}^{n} 2(\mathbf{a}^T \mathbf{y}_i - b_i)\mathbf{y}_i = 2\mathbf{Y}^T(\mathbf{Ya} - \mathbf{b})$$
  - 令偏导数为零，得：
$$\mathbf{Y}^T\mathbf{Ya} = \mathbf{Y}^T\mathbf{b}, \quad \Rightarrow \mathbf{a} = (\mathbf{Y}^T\mathbf{Y})^{-1}\mathbf{Y}^T\mathbf{b} = \mathbf{Y}^{+}\mathbf{b}$$

其中，$\mathbf{Y}^{+}$ 即为**Y**的伪逆。

  - 实际计算（正则化技术）：$\mathbf{Y}^{+} \approx (\mathbf{Y}^T\mathbf{Y} + \varepsilon\mathbf{I})^{-1}\mathbf{Y}^T \Big|_{\varepsilon \to 0}$

（即回归分析方法）

5.6 MSE 准则函数

- **梯度下降法**
  - 计算伪逆需要求矩阵的逆，计算复杂度高。如果原始样本的维数很高，比如 $d > 5000$，将十分耗时。
  - 采用梯度下降法： $\mathbf{a}_{k+1} = \mathbf{a}_k + \eta_k \mathbf{Y}^T(\mathbf{b} - \mathbf{Y}\mathbf{a}_k)$
    - 梯度下降法得到的 $\mathbf{a}_{k+1}$ 将收敛于一个解，该解满足方程： $\mathbf{Y}^T(\mathbf{b} - \mathbf{Y}\mathbf{a}) = \mathbf{0}$
  - 也可以采用序列更新方法，此方法需要的计算存储量会更小：

$$\mathbf{a}_{k+1} = \mathbf{a}_k + \eta_k (b_k - (\mathbf{a}_k)^T \mathbf{y}^k) \mathbf{y}^k$$

（此时考虑单个样本对误差的贡献）

5.6 MSE 准则函数

- **Widrow-Hoff方法（序列最小平方更新方法）**

Widrow-Hoff (Least mean squared) Approach

begin initialize: $\mathbf{a}, \mathbf{b}, \eta$, threshold θ, k=0
  do $k \leftarrow k+1 \pmod{n}$
    $\mathbf{a} = \mathbf{a} + \eta_k \boxed{(b_k - (\mathbf{a}_k)^T \mathbf{y}^k)\, \mathbf{y}^k}$
  until $\| (b_k - (\mathbf{a}_k)^T \mathbf{y}^k)\, \mathbf{y}^k \| < \theta$
  return $\mathbf{a}$
end

注：$\mathbf{y}^k$ 为使 $(\mathbf{a}_k)^T \mathbf{y}^k \neq b_k$ 的样本，因为相等时对更新无贡献

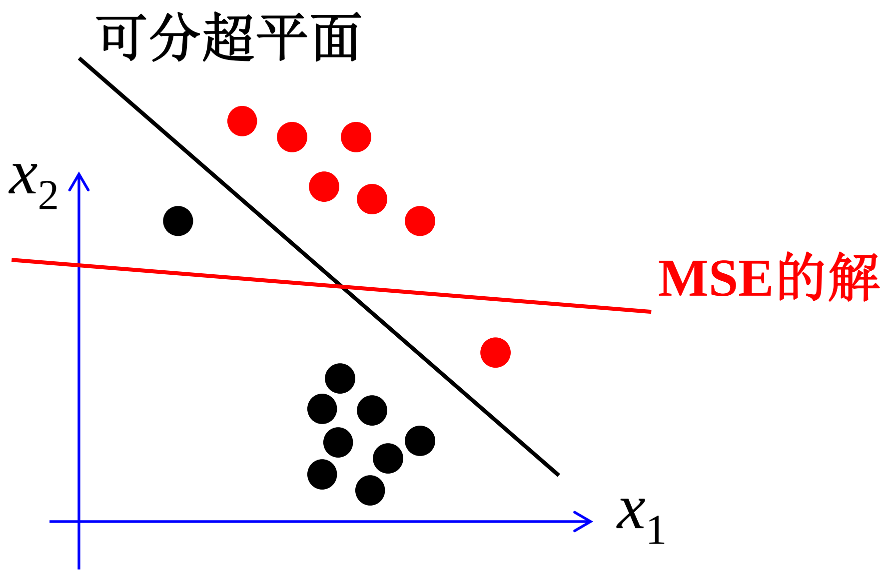
5.6 MSE 准则函数

- **Widrow-Hoff方法 vs 松驰算法**
  - 松驰算法是一种错误更正方法，要求 $\mathbf{a}^T\mathbf{y}^k > b$ for all $\mathbf{y}^k$。
  - Widrow-Hoff 要求更正不相等情形：$(\mathbf{a}_k)^T\ \mathbf{y}^k \neq b_k$。但是，**实际上，满足$(\mathbf{a}_k)^T\ \mathbf{y}^k = b_k$ 几乎是不可能的**。因此，迭代将会无穷次进行下去。所以要求 η_k 需要随着 k 的增加而逐渐减小，以保证算法的收敛性。一般来讲，实际计算中取：$\eta_k = \eta_1 / k$。

5.6 MSE 准则函数

- **Widrow-Hoff方法 vs 感知器准则**
  - 相对于感知器准则，最小平方准则方法可能并不收敛于可分超平面，即使该平面是存在的。
    - MSE方法的本质是最小化样本至超平面的距离的平方和。

将超平面旋转一个角度，不会改变距离平方和



5.7 Ho-Kashyap 方法

- **Ho-Kashyap 方法**
  - MSE算法上最小化 $\|\mathbf{Ya}-\mathbf{b}\|^2$，所得到的最优解并不需要位于可分超平面上。
  - 如果训练样本**是线性可分的**，则可找到一个权向量 **a**，对所有样本，均有 $\mathbf{a}^T\mathbf{y}_i > 0$。换句话说，一定存在一个 **a** 和 **b**, 使

$$\mathbf{Ya} = \mathbf{b} > \mathbf{0}$$

但是，事先并不知道 **b**。因此，MSE准则函数可以更新为：

$$J_s(\mathbf{a}, \mathbf{b}) = \|\mathbf{Ya}-\mathbf{b}\|^2$$

注：直接优化 $J_s(\mathbf{a}, \mathbf{b})$ 将导致平凡解，所以需要给 **b** 加一个 **b>0** 的约束条件。此时，**b** 可以解释为margin。

5.7 Ho-Kashyap 方法

- **梯度下降法**
  - 梯度

$$\frac{\partial J_s(\mathbf{a},\mathbf{b})}{\partial \mathbf{a}} = 2\mathbf{Y}^T(\mathbf{Y}\mathbf{a}-\mathbf{b}), \quad \frac{\partial J_s(\mathbf{a},\mathbf{b})}{\partial \mathbf{b}} = -2(\mathbf{Y}\mathbf{a}-\mathbf{b})$$

对 **a** 而言，总有 **a = Y⁺b**，其中 **Y⁺** 为 **Y** 的伪逆。

对 **b**，需要同时满足约束条件 **b>0**。梯度更新：

$$\mathbf{b}_{k+1} = \mathbf{b}_k - \eta_k \frac{\partial J_s(\mathbf{a},\mathbf{b})}{\partial \mathbf{b}}$$

由于 $\mathbf{b}_k$ 总是大于零，要使$\mathbf{b}_{k+1}$也大于零，可以要求$\partial J_s(\mathbf{a}, \mathbf{b}) / \partial \mathbf{b}$为负。

- **梯度下降法**
  - b 的梯度下降：

元素取绝对值

$$\mathbf{b}_1 > \mathbf{0},\quad \mathbf{b}_{k+1} = \mathbf{b}_k - \eta_k \boxed{\frac{1}{2}\left(\frac{\partial J_s(\mathbf{a},\mathbf{b})}{\partial \mathbf{b}} - \left|\frac{\partial J_s(\mathbf{a},\mathbf{b})}{\partial \mathbf{b}}\right|\right)} \quad (\leq 0)$$

  - 更新 a 和 b：

$$\mathbf{a}_k = \mathbf{Y}^+\mathbf{b}_k$$

$$\mathbf{b}_1 > \mathbf{0},\quad \mathbf{b}_{k+1} = \mathbf{b}_k + 2\eta_k \mathbf{e}_k^+$$

其中，$\mathbf{e}_k^+ = \frac{1}{2}\left((\mathbf{Y}\mathbf{a}_k - \mathbf{b}_k) + |\mathbf{Y}\mathbf{a}_k - \mathbf{b}_k|\right)$，　$\because \frac{\partial J_s(\mathbf{a},\mathbf{b})}{\partial \mathbf{b}} = -2(\mathbf{Y}\mathbf{a} - \mathbf{b})$

为了防止 **b** 收敛于 **0**，可以让 **b** 从一个非负向量（ $\mathbf{b}_1 > \mathbf{0}$ ）开始进行更新。

由于要求 $\partial J_s(\mathbf{a}, \mathbf{b}) / \partial \mathbf{b}$ 等于**0**，在开始迭代时可令 $\partial J_s(\mathbf{a}, \mathbf{b}) / \partial \mathbf{b}$的元素为正的分量等于零，从而加快收敛速度。

5.7 Ho-Kashyap 方法

Ho-Kashyap Algorithm

begin initialize: $\mathbf{a}$, $\mathbf{b}$, $\eta_0 < 1$, $k=0$, threshold b_{min}, k_{max}

    do $k \leftarrow k+1 \pmod{n}$

        $\mathbf{e} \leftarrow \mathbf{Ya} - \mathbf{b}$

        $\mathbf{e}^+ \leftarrow 1/2(\mathbf{e} + \mathrm{abs}(\mathbf{e}))$

        $\mathbf{b} \leftarrow \mathbf{b} + 2\,\eta_k\,\mathbf{e}^+$

        $\mathbf{a} = \mathbf{Y}^+\mathbf{b}$

        if $\mathrm{abs}(\mathbf{e}) \leq b_{min}$, then return $\mathbf{a}$,$\mathbf{b}$ and exit

    until $k = k_{max}$

    print “No solution found!”

end

5.7 Ho-Kashyap 方法

- **Ho-Kashyap算法**
  - 由于权向量序列 $\{\mathbf{a}_k\}$ 完全取决定于 $\{\mathbf{b}_k\}$，因此本质上讲 **Ho-Kashyap** 算法是一个生成**margin** 序列$\{\mathbf{b}_k\}$ 的方法。
  - 由于初始 $\mathbf{b}_1>\mathbf{0}$，且更新因子$\eta > 0$，因此 $\mathbf{b}_k$ 总是大于**0**。
  - 对于更新因子$0<\eta \le 1$，如果问题线性可分，则总能找到元素全为正的**b** 。
  - 如果 $\mathbf{e}_k = \mathbf{Y}\mathbf{a}_k - \mathbf{b}_k$全为 0，此时， $\mathbf{b}_k$ 将不再更新，因此获得一个解。**如果 $\mathbf{e}_k$ 有一部分元素小于0，则可以证明该问题不是线性可分的** *。

*证明见：Richard O. Duda, Peter E. Hart, David G. Stork, Pattern Classification, 2nd Edition, John Wiley, 2001. page: 251-253

5.8 多类线性判别函数

- **对 c 类分类问题**
  - 对于 c 类分类问题:
    - 设 $g_i(\mathbf{x})$, i=1,2,…, c, 表示每个类别对应的判别函数
    - **线性机器的决策规则**：
      - 如果 $g_i(\mathbf{x}) \geq g_j(\mathbf{x}),\ \forall\ j \neq i$，则 $\mathbf{x}$ 被分为第 ω_i 类。
    - 考虑多类线性可分的情形，对**规范化增广样本表示方法**，此时决策规则为：

$$\mathbf{a}_i^T\mathbf{y} \geq \mathbf{a}_j^T\mathbf{y},\quad \forall\ j \neq i,\ \text{则 } \mathbf{y} \text{ 被分为第 } i \text{ 类}$$

- **方法一：MSE多类扩展**
  - 可以直接采用 c 个两类分类器的组合，且这种组合具有与两类分类问题类似的代数描述形式
  - 线性变换(注，此处不必采用规范化增广坐标表示)：

$$\mathbf{y}=\mathbf{W}^T\mathbf{x}+\mathbf{b},\quad \mathbf{W}\in R^{d\times c},\mathbf{b}\in R^c$$

  - 决策准则： $\text{if}\quad j=\arg\max\left(\mathbf{W}^T\mathbf{x}+\mathbf{b}\right),\quad \text{then}\quad \mathbf{x}\in\omega_j$
  - 回归值的构造：

比如:设第1,2个样本属于第一类,则 $\mathbf{y}_1=\mathbf{y}_2=[1,0,0\ldots,0]^T$

$$\mathbf{Y}=\begin{pmatrix} 1 & 1 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & 1 \end{pmatrix}=[\mathbf{y}_1,\mathbf{y}_2,\cdots,\mathbf{y}_n]\in R^{c\times n}$$

- **方法一：MSE多类扩展**

  – 目标函数：

$$\min_{\mathbf{W},\mathbf{b}} \sum_{i=1}^{n} \left\| \mathbf{W}^T \mathbf{x} + \mathbf{b} - \mathbf{y}_i \right\|_2^2$$

令： $\hat{\mathbf{W}} = \begin{pmatrix} \mathbf{W} \\ \mathbf{b}^T \end{pmatrix} \in R^{(d+1)\times c}$， $\hat{\mathbf{x}} = \begin{pmatrix} \mathbf{x} \\ 1 \end{pmatrix} \in R^{d+1}$， $\hat{\mathbf{X}} = \left(\hat{\mathbf{x}}_1, \hat{\mathbf{x}}_2, \cdots, \hat{\mathbf{x}}_n \right) \in R^{(d+1)\times n}$

$$\Downarrow$$

$$\sum_{i=1}^{n} \left\| \mathbf{W}^T \mathbf{x} + \mathbf{b} - \mathbf{y}_i \right\|_2^2 = \left\| \hat{\mathbf{W}}^T \hat{\mathbf{X}} - \mathbf{Y} \right\|_F^2$$

($\|.\|_F$为Frobenius范数)

$$\Downarrow$$

$$\min_{\hat{\mathbf{W}}} \left\| \hat{\mathbf{W}}^T \hat{\mathbf{X}} - \mathbf{Y} \right\|_F^2$$

$$\Downarrow$$

$$\hat{\mathbf{W}} = \left(\hat{\mathbf{X}} \hat{\mathbf{X}}^T \right)^{-1} \hat{\mathbf{X}} \mathbf{Y}^T \in R^{(d+1)\times c} \qquad \hat{\mathbf{W}} = \left(\hat{\mathbf{X}} \hat{\mathbf{X}}^T + \lambda \mathbf{I} \right)^{-1} \hat{\mathbf{X}} \mathbf{Y}^T \in R^{(d+1)\times c}$$

λ：一个小正数

- **方法二：感知器准则扩展方法—逐步修正(不讲-自学)**
  - 目标：一次性学习出c个权重向量（以下采用规范化增广样本表示）
  - S_1: 设置任意的初始权重向量 $\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_c$
  - S_2: 考察某个属于第i类的样本 $\mathbf{y}^k$:
    - 如果 $(\mathbf{a}_i)^T\mathbf{y}^k > (\mathbf{a}_j)^T\mathbf{y}^k$，对任意 $j \neq i$ 均成立，则所有权向量不变；
    - 如果存在 j 使 $(\mathbf{a}_i)^T\mathbf{y}^k \leq (\mathbf{a}_j)^T\mathbf{y}^k$，则可以选择一个 j（比如使 $(\mathbf{a}_j)^T\mathbf{y}^k$ 最大者），对权值分量进行修正：

$$\begin{cases} \left.\begin{aligned} \mathbf{a}_i(k+1) &= \mathbf{a}_i(k) + \eta_k \mathbf{y}^k \\ \mathbf{a}_j(k+1) &= \mathbf{a}_j(k) - \eta_k \mathbf{y}^k \end{aligned}\right\} \text{相当于将样本}[(\mathbf{y}^k)^T,(-\mathbf{y}^k)]^T\text{错分} \\ \mathbf{a}_m(k+1) = \mathbf{a}_m(k), \quad m \neq i, j \end{cases}$$

  - S_3: 如果对所有样本均正确分类，则停止；否则考察另一个样本。

- **方法三：Kelser 构造**（注：此处采用规范化增广样本表示） **(不讲-自学)**
  - 设计一个两类分类线性分类器。
  - 解决这一问题的思路是：**将$\mathbf{a}_1$, $\mathbf{a}_2$, …, $\mathbf{a}_c$ 组合成一个长向量，同时构造新的训练样本**。
    - 比如，如果样本 $\mathbf{y}$ 属于第一类，则有

$$\mathbf{a}_1^T\mathbf{y} > \mathbf{a}_j^T\mathbf{y},\ j = 2,3,\dots,c$$

此时，可构造如下 c–1 个新样本：

$$\boldsymbol{\eta}_{12} = \begin{pmatrix} \mathbf{y} \\ -\mathbf{y} \\ \mathbf{0} \\ \vdots \\ \mathbf{0} \end{pmatrix},\quad \boldsymbol{\eta}_{13} = \begin{pmatrix} \mathbf{y} \\ \mathbf{0} \\ -\mathbf{y} \\ \vdots \\ \mathbf{0} \end{pmatrix},\quad \dots,\quad \boldsymbol{\eta}_{1c} = \begin{pmatrix} \mathbf{y} \\ \mathbf{0} \\ \mathbf{0} \\ \vdots \\ -\mathbf{y} \end{pmatrix}$$

$$\mathbf{a}_1^T\mathbf{y} > \mathbf{a}_2^T\mathbf{y} \qquad \mathbf{a}_1^T\mathbf{y} > \mathbf{a}_3^T\mathbf{y} \qquad \mathbf{a}_1^T\mathbf{y} > \mathbf{a}_c^T\mathbf{y}$$

- **方法三：Kelser 构造**
  - 因此，对 n 个样本来讲，由于有 c 个类别，一共可构造成出 $(c\text{-}1)n$ 个新样本，它们的维数为 $c(d+1)$ (考虑线性齐次坐标），这样就生成了一个新的两类分类问题。
  - 如果线性可分，则需要找一个权向量 **a** 使

$$\mathbf{a}^T \boldsymbol{\eta}_{ij} > 0,\ j \neq i$$

其中，$\mathbf{a} = \begin{pmatrix} \mathbf{a}_1 \\ \mathbf{a}_2 \\ \vdots \\ \mathbf{a}_c \end{pmatrix}$.

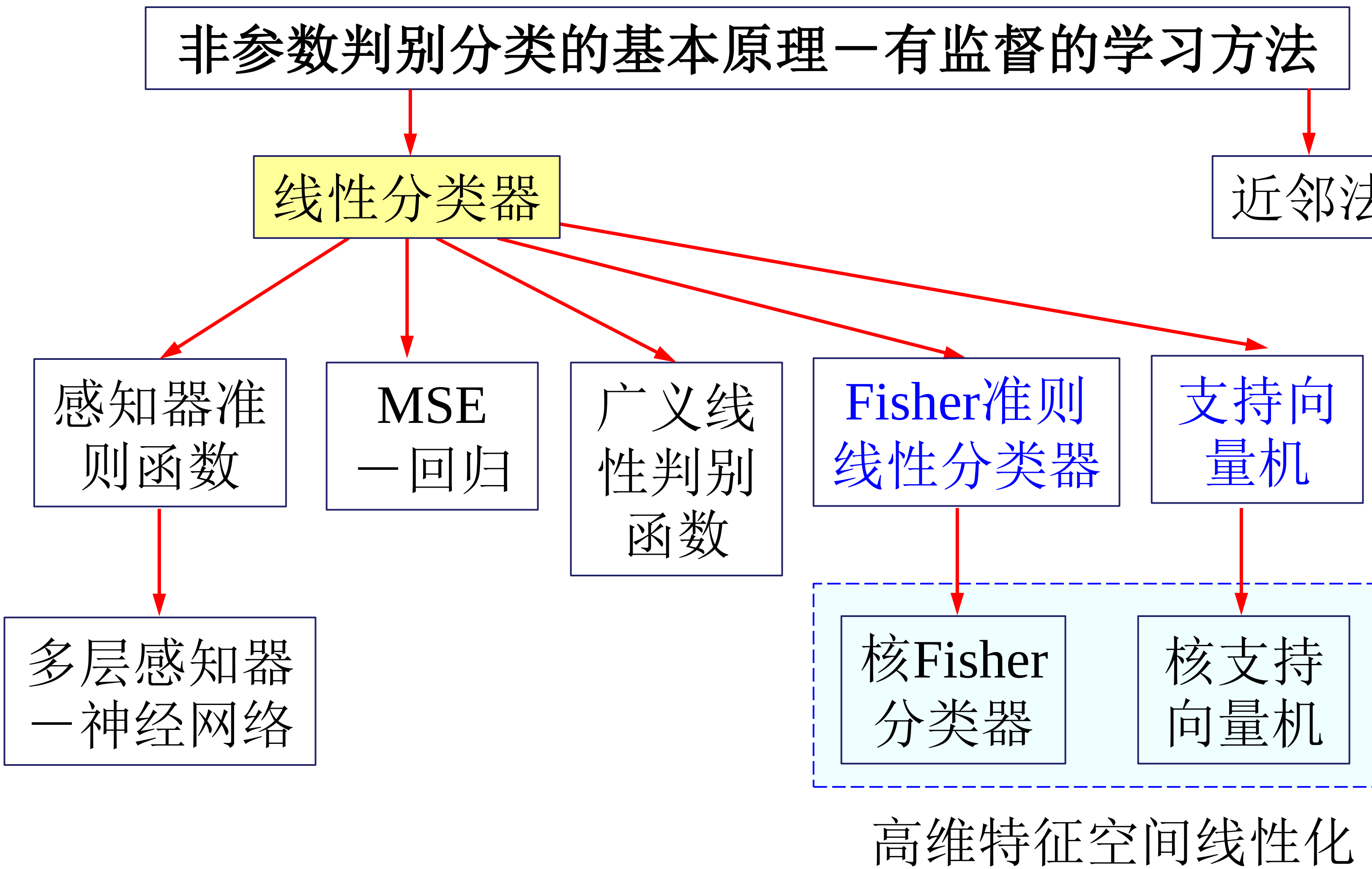
比如：$\mathbf{a}_1^T \mathbf{y} > \mathbf{a}_j^T \mathbf{y} \Leftrightarrow \mathbf{a}^T \boldsymbol{\eta}_{1j} > 0$

- **方法三：Kelser 构造**
  - 优点：
    - 可以将一个**多类问题转化为一个两类分类问题**，便于采用现有的两类分类器构造方法
    - 由此获得的一个多类线性分类器可以保证不会出现歧义区域。
  - 缺点：
    - 缺点十分明确：增加了样本的规模，增大了样本空的维数，对大数据处理极度不利。

5.9 本章小结

- **概念**
  - 判别函数、线性判别函数、广义线性判别函数、可分性、分界面、决策规则、点到超平面的距离、规范化增广样本表示、多类分类
- **线性判别函数**
  - 感知准则函数、松弛感知准则函数、平方误差准则函数
- **算法**
  - 感知准则函数的批量更新方法、感知准则函数的单样本更新方法、松弛感知准则函数单样本更新方法、MSE梯度下降法、Ho-Kashyap方法、多类扩展方法

5.9 本章小结


非参数判别分类的基本原理－有监督的学习方法
线性分类器
近邻法
感知器准则函数
MSE－回归
广义线性判别函数
Fisher准则线性分类器
支持向量机
多层感知器－神经网络
核Fisher分类器
核支持向量机
高维特征空间线性化

下次课内容

- **神经网络基础**
  - 发展历史
  - 网络结构
- **基本模型**
  - 单层感知器、多层感知器、反向传播算法等

Thank All of You!

(Questions?)

向世明

smxiang@nlpr.ia.ac.cn

people.ucas.ac.cn/~xiangshiming

时空数据分析与学习课题组(STDAL)

中科院自动化研究所· 模式识别国家重点实验室