线性分类器与非线性分类器

郑 伟 诗

http://www.isee-ai.cn/~zhwshi/

中山大学

机器智能与先进计算
教育部重点实验室

线性分类器与非线性分类器

- 线性判别函数
- 线性判别分析
- 支持向量机
- 非线性分类
- VC维理论初步

分类器、判别函数及判定面

- 在基于贝叶斯原则的分类方法中，我们使用训练样本来估计概率密度，进而通过计算给定测试样本的后验概率判定其类别。
- 在本章中，我们从决策边界而非条件概率密度入手。
- 线性分类面未必最优，但是处理简便，特别是在小样本下不至于因模型复杂而产生过拟合。
- 非线性分类面则提供更为复杂的表达能力，往往更符合实际的分类问题。

分类器、判别函数及判定面

- 在一个包含c类的分类问题中，我们可为每个类别定义一个判别函数：

$$g_i(\mathbf{x}), i = 1, \ldots, c$$

其中$\mathbf{x}$为样本的特征向量。$\mathbf{x}$所属类别由取值最大的判别函数决定：

若$g_i(\mathbf{x}) > g_j(\mathbf{x}),\ \forall j \neq i$, 则$\mathbf{x}$类标被判定为$\omega_i$。

- 在贝叶斯最小风险决策中：

$$g_i(\mathbf{x}) = -R(\alpha_i|\mathbf{x})$$

- 在贝叶斯最小错误率分类中：

$$g_i(\mathbf{x}) = P(\omega_i|\mathbf{x})$$

分类器、判别函数及判定面

- 一个包含c类的分类任务将特征空间分成c个判决区域$R_1,\ldots,R_c$：

  若$g_i(\mathbf{x}) > g_j(\mathbf{x}), \forall j \neq i$，则$\mathbf{x} \in R_i$

  R_i表示在特征空间中$\mathbf{x}$被赋以类标ω_i的区域。

  判决区域之间的边界称为判决边界或判定面。

- 二分类情况：二分类器对应两个判别函数，设为$g_1(\mathbf{x})$和$g_2(\mathbf{x})$。此时，也可定义一个单一的判别函数：

  $$g(\mathbf{x}) \equiv g_1(\mathbf{x}) - g_2(\mathbf{x})$$

  如果$g(\mathbf{x}) > 0$，则判为ω_1；否则判为ω_2。

第一部分：线性判别函数

线性判别函数

- 线性判别函数：由$\mathbf{x}$的各分量线性组合而成的判别函数

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

其中$\mathbf{w}$是权向量，而w_0为偏置。

- 二类线性分类器使用以下判定规则：

如果$g(\mathbf{x}) > 0$则判定为ω_1，如果$g(\mathbf{x}) < 0$则判定为ω_2。

$$\Leftrightarrow$$

如果$\mathbf{w}^{\mathrm{T}}\mathbf{x} > -w_0$则判定为$\omega_1$，如果$\mathbf{w}^{\mathrm{T}}\mathbf{x} < -w_0$则判定为$\omega_2$。

如果$g(\mathbf{x}) = 0$，那么$\mathbf{x}$可以被归到任意一类。此时，方程$g(\mathbf{x}) = 0$定义了一个判定面，它把归类于ω_1的点与归类于ω_2的点分开来。由于$g(\mathbf{x})$是线性的，该判定面是一个超平面。

线性判别函数

在判定面上：

$$\mathbf{w}^{\mathrm{T}}\mathbf{x}_1 + w_0 = \mathbf{w}^{\mathrm{T}}\mathbf{x}_2 + w_0 = 0$$

$$\Rightarrow \mathbf{w}^{\mathrm{T}}(\mathbf{x}_1 - \mathbf{x}_2) = 0$$

$\Rightarrow$ $\mathbf{w}$和判定面正交（指向R_1）

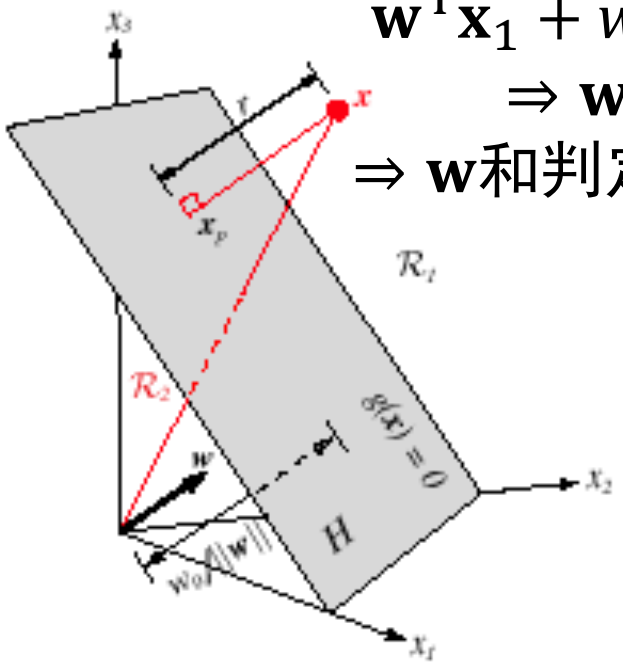


FIGURE 5.2. The linear decision boundary H, where $g(\mathbf{x}) = \mathbf{w}^t\mathbf{x} + w_0 = 0$, separates the feature space into two half-spaces $\mathcal{R}_1$ (where $g(\mathbf{x}) > 0$) and $\mathcal{R}_2$ (where $g(\mathbf{x}) < 0$). From: Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*. Copyright © 2001 by John Wiley & Sons, Inc.

线性判别函数

- 线性判别函数$g(\mathbf{x})$是特征空间中$\mathbf{x}$到判定面距离的一种代数度量：

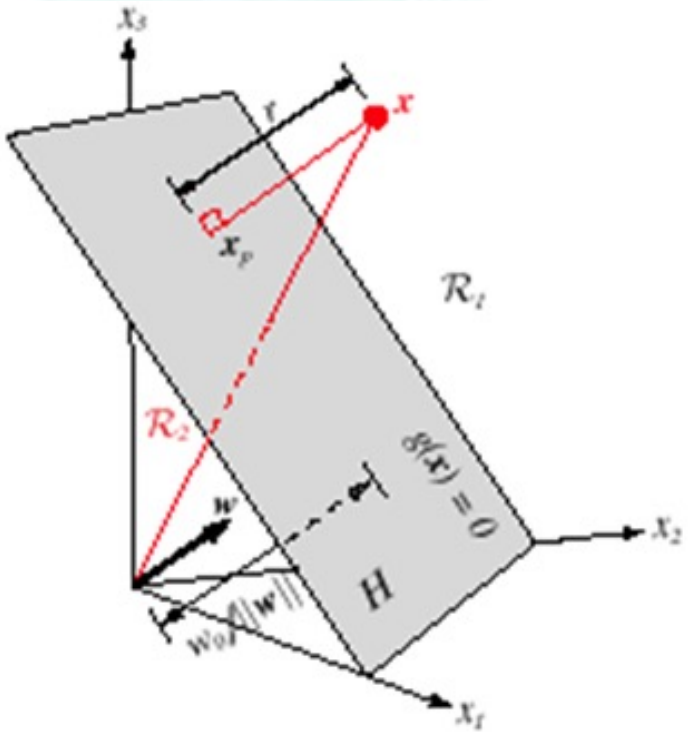
$$\mathbf{x} = \mathbf{x}_p + r \cdot \frac{\mathbf{w}}{\|\mathbf{w}\|} \quad (\text{因}\mathbf{w}\text{与}\mathbf{x} - \mathbf{x}_p\text{平行})$$

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

$$= g(\mathbf{x}_p) + r\|\mathbf{w}\| = r\|\mathbf{w}\|$$

因此$r = \dfrac{g(\mathbf{x})}{\|\mathbf{w}\|} = d(\mathbf{x}, H)$

特别地，$d(O, H) = \dfrac{w_0}{\|\mathbf{w}\|}$



线性判别函数

- 多类情况
  - 定义c个线性判别函数：

$$g_i(\mathbf{x}) = \mathbf{w}_i^{\mathrm{T}}\mathbf{x} + w_{i0} \qquad i = 1, \ldots, c$$

如果$g_i(\mathbf{x}) > g_j(\mathbf{x})$，$\forall j \neq i$，则把$\mathbf{x}$分为$\omega_i$类；
如果$g_i(\mathbf{x}) = g_j(\mathbf{x})$，则类别不定。

  - 该分类器将特征空间分为c个判决区域$R_1, \ldots, R_c$。
  - 对于两个相邻的判决区域R_i和R_j，它们的分界面是由下式定义的超平面H_{ij}的一部分：

$$g_i(\mathbf{x}) - g_j(\mathbf{x}) = (\mathbf{w}_i - \mathbf{w}_j)^{\mathrm{T}}\mathbf{x} + (w_{i0} - w_{j0}) = 0$$

$\mathbf{w}_i - \mathbf{w}_j$与$H_{ij}$正交，且：

$$d(\mathbf{x}, H_{ij}) = \frac{g_i(\mathbf{x}) - g_j(\mathbf{x})}{\|\mathbf{w}_i - \mathbf{w}_j\|}$$

线性判别函数

- 线性分类器的判决区域是凸的，这限制了分类器的适应性和精确性。

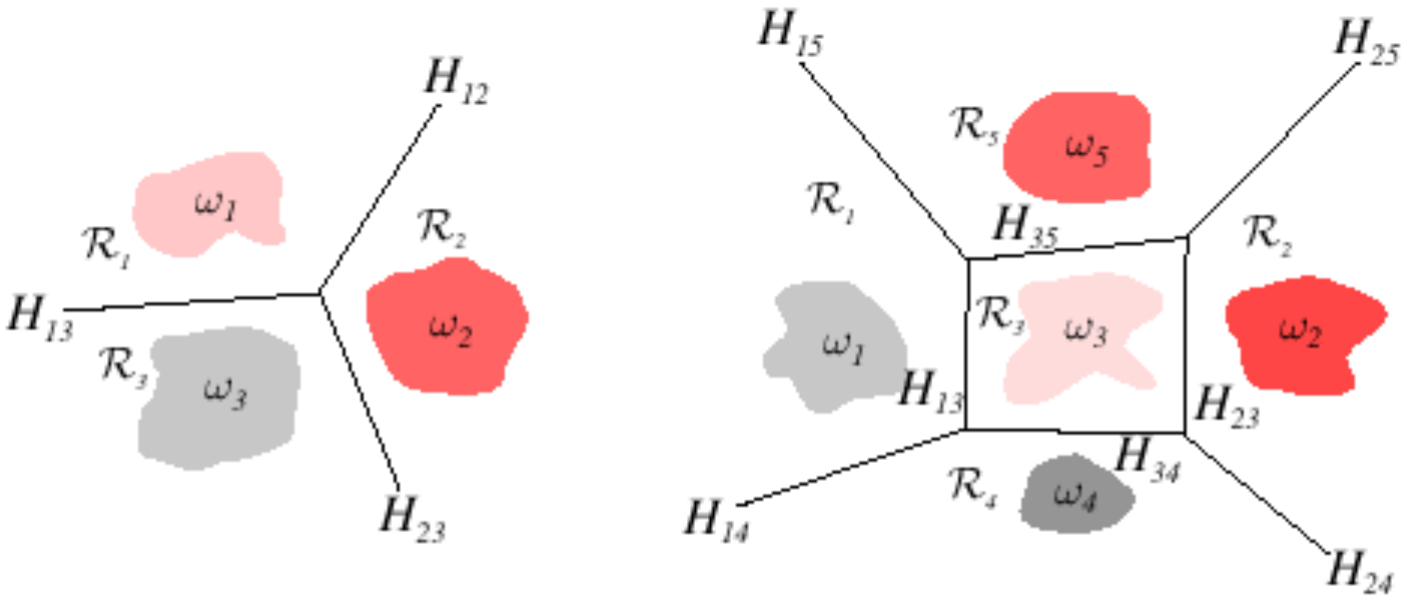


FIGURE 5.4. Decision boundaries produced by a linear machine for a three-class problem and a five-class problem. From: Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*. Copyright © 2001 by John Wiley & Sons, Inc.

线性判别函数

- 下面主要考虑二分类的情况。多分类可视为对二分类的推广。
- 为表达方便，引入下面的增广形式：

$$g(\mathbf{x}) = \mathbf{w}^{\mathrm{T}}\mathbf{x} + w_0 = \sum_{i=1}^{d} w_i\, x_i + w_0 \cdot 1 = \mathbf{a}^{\mathrm{T}}\mathbf{y}$$

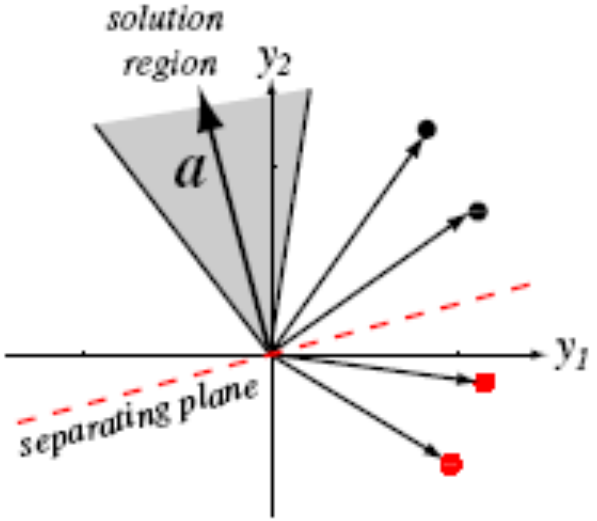
$$\mathbf{a} = \begin{bmatrix} w_0 \\ w_1 \\ \vdots \\ w_d \end{bmatrix} \qquad \mathbf{y} = \begin{bmatrix} 1 \\ x_1 \\ \vdots \\ x_d \end{bmatrix}$$

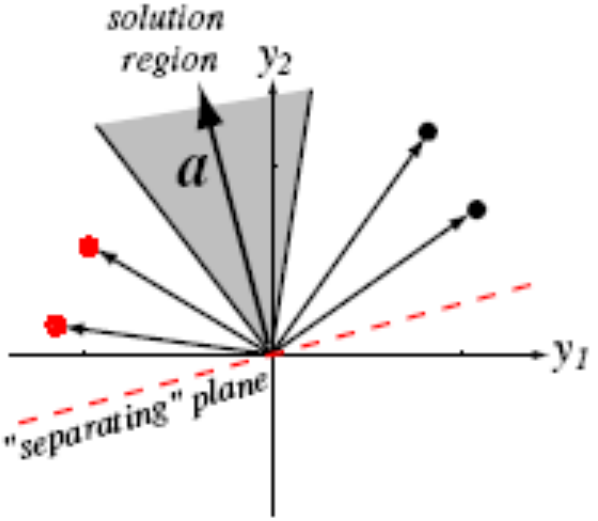
其中$\mathbf{y}$为增广特征向量，$\mathbf{a}$为增广权向量。

线性判别函数

- 基于该判别函数的分类准则为：
  - 对于一个样本$\mathbf{y}$，如果有$\mathbf{a}^{\mathrm{T}}\mathbf{y} > 0$就将其分类为$\omega_1$
  - 如果有$\mathbf{a}^{\mathrm{T}}\mathbf{y} < 0$就将其分类为$\omega_2$
- 给定线性判别函数的形式$g(\mathbf{x}) = \mathbf{a}^{\mathrm{T}}\mathbf{y}$，在一个分类器的学习任务中，我们的目标是用一组带有类标的训练样本确定判别函数中的权向量$\mathbf{a}$。
- 若有n个训练样本$\mathbf{y}_i, i = 1, \cdots, n$，则其中每个$\mathbf{y}_i$都将对权向量$\mathbf{a}$产生一个约束，使其位于某个分类超平面的一侧。
- 如果严格的解向量存在，则其必位于 n 个半空间的交叠区。

线性判别函数





- ω_1
- ω_2

为表达方便，可将增广特征向量进行归整化：
如果$\mathbf{y}_i$属于ω_2，则将$\mathbf{y}_i$替换为$-\mathbf{y}_i$，从而各训练样本对权向量$\mathbf{a}$的约束可规整化为：$\mathbf{a}^{\mathrm{T}}\mathbf{y}_i > 0$

线性判别函数

- 对于线性可分的训练集，由于在所有训练样本上满足前述约束的权向量存在，通常采用一种根据被错分的训练样本对权向量进行纠正的“错误-纠正”优化策略。
- 然而对于现实中的分类问题，样本往往并非线性可分，因此在所有训练样本上满足前述约束的权向量常常不存在。试图通过“错误-纠正”过程寻找权向量的过程易陷于不能收敛的状态。
- 我们进而考虑一种兼顾所有训练样本，而不仅仅是被错分样本的训练策略，以求权向量能稳定收敛。
- 其中的一种代表性方法是最小平方误差法/ 最小均方误差：寻找尽可能满足 $\mathbf{a}^{\mathrm{T}}\mathbf{y}_i = b_i$ 的权向量 $\mathbf{a}$，其中 b_i 是正的常数。
  - 因此我们就将线性不等式求解的问题变换为约束更强，但也更容易理解的问题，即线性方程组的求解。

最小平方误差法

- 最小平方误差与伪逆
  对于规整化的训练样本$\mathbf{y}_i, i = 1, \cdots, n$，我们希望找到一个权向量$\mathbf{a}$，使得对任意给定的一些正的常数$b_i$，有：
  $$\mathbf{a}^{\mathrm{T}}\mathbf{y}_i = b_i, i = 1, \cdots, n$$
  写成矩阵形式即为：
  $$\begin{bmatrix} y_{10} & y_{11} & \cdots & y_{1d} \\ y_{20} & y_{21} & \cdots & y_{2d} \\ \vdots & \vdots & & \vdots \\ y_{n0} & y_{n1} & \cdots & y_{nd} \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_d \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix} \Leftrightarrow \mathbf{Ya} = \mathbf{b}$$
- 当方程数多于未知数时，$\mathbf{a}$是超定的。
- 我们定义一个误差向量：
  $$\mathbf{e} = \mathbf{Ya} - \mathbf{b}$$

最小平方误差法

- 寻找满足最小化误差平方和准则的权向量$\mathbf{a}$：

$$J_s(\mathbf{a}) = \|\mathbf{Ya} - \mathbf{b}\|^2 = \|\mathbf{e}\|^2 = \sum_{i=1}^{n} (\mathbf{a}^{\mathrm{T}}\mathbf{y}_i - b_i)^2$$

- 该目标函数梯度：

$$\nabla J_s = \sum_{i=1}^{n} 2(\mathbf{a}^{\mathrm{T}}\mathbf{y}_i - b_i)\mathbf{y}_i = 2\mathbf{Y}^{\mathrm{T}}(\mathbf{Ya} - \mathbf{b})$$

- 令梯度为$\mathbf{0}$，可得：

$$\mathbf{Y}^{\mathrm{T}}\mathbf{Ya} = \mathbf{Y}^{\mathrm{T}}\mathbf{b}$$

- 如果$\mathbf{Y}^{\mathrm{T}}\mathbf{Y}$是非奇异的：

$$\mathbf{a} = (\mathbf{Y}^{\mathrm{T}}\mathbf{Y})^{-1}\mathbf{Y}^{\mathrm{T}}\mathbf{b} = \mathbf{Y}^{\dagger}\mathbf{b}$$

这里的矩阵$\mathbf{Y}^{\dagger}$是$\mathbf{Y}$的伪逆矩阵。

- 说明：对任意固定的$\mathbf{b}$，采用最小平方误差法获得的解不一定能将样本全部分开，即使样本是线性可分的。

最小平方误差法

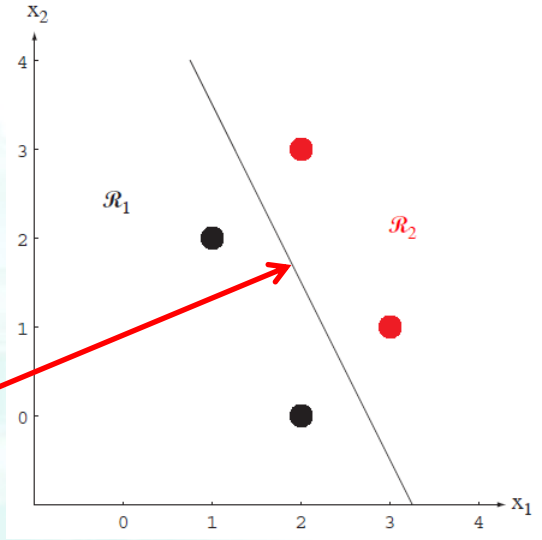
- 例：用伪逆矩阵构造线性分类器

ω_1: $(1,2)^T$ and $(2,0)^T$
ω_2: $(3,1)^T$ and $(2,3)^T$

Sample matrix

$$\mathbf{Y} = \begin{pmatrix} 1 & 1 & 2 \\ 1 & 2 & 0 \\ -1 & -3 & -1 \\ -1 & -2 & -3 \end{pmatrix}$$

$$\mathbf{a}^T \begin{bmatrix} 1 \\ x_1 \\ x_2 \end{bmatrix} = 0$$



Pseudoinverse

$$\mathbf{Y}^\dagger = \begin{pmatrix} 5/4 & 13/12 & 3/4 & 7/12 \\ -1/2 & -1/6 & -1/2 & -1/6 \\ 0 & -1/3 & 0 & -1/3 \end{pmatrix}$$

取相等的b_i，例如
$\mathbf{b} = [1,1,1,1]^T$

此时，解为：
$\mathbf{a} = \mathbf{Y}^\dagger \mathbf{b} =$
$(11/3,-4/3,-2/3)^T$

最小平方误差法

- 对最优判别函数的渐进逼近
  - 最小平方误差法的解存在一个性质，如果$\mathbf{b} = \mathbf{1}_n$，则当样本数趋向无穷多时，它以最小均方误差逼近贝叶斯判别函数：

$$g_0(\mathbf{x}) = P(\omega_1|\mathbf{x}) - P(\omega_2|\mathbf{x})$$

  - 具体而言，最小化目标函数$J_s(\mathbf{a})$等价于最小化以下函数

$$\varepsilon^2 = \int [\mathbf{a}^{\mathrm{T}}\mathbf{y} - g_0(\mathbf{x})]^2 p(\mathbf{x}) d\mathbf{x}$$

最小平方误差法

- Widrow-Hoff或最小均方（least-mean-squared, LMS）过程
  - 利用梯度下降法来最小化$J_s(\mathbf{a})$:

$$\nabla J_s = \sum_{i=1}^{n} 2(\mathbf{a}^{\mathrm{T}}\mathbf{y}_i - b_i)\mathbf{y}_i = 2\mathbf{Y}^{\mathrm{T}}(\mathbf{Ya} - \mathbf{b})$$

$$\mathbf{a}(k+1) = \mathbf{a}(k) - \eta(k)\nabla J_s$$

$$\eta(k) = \eta(1)/k$$

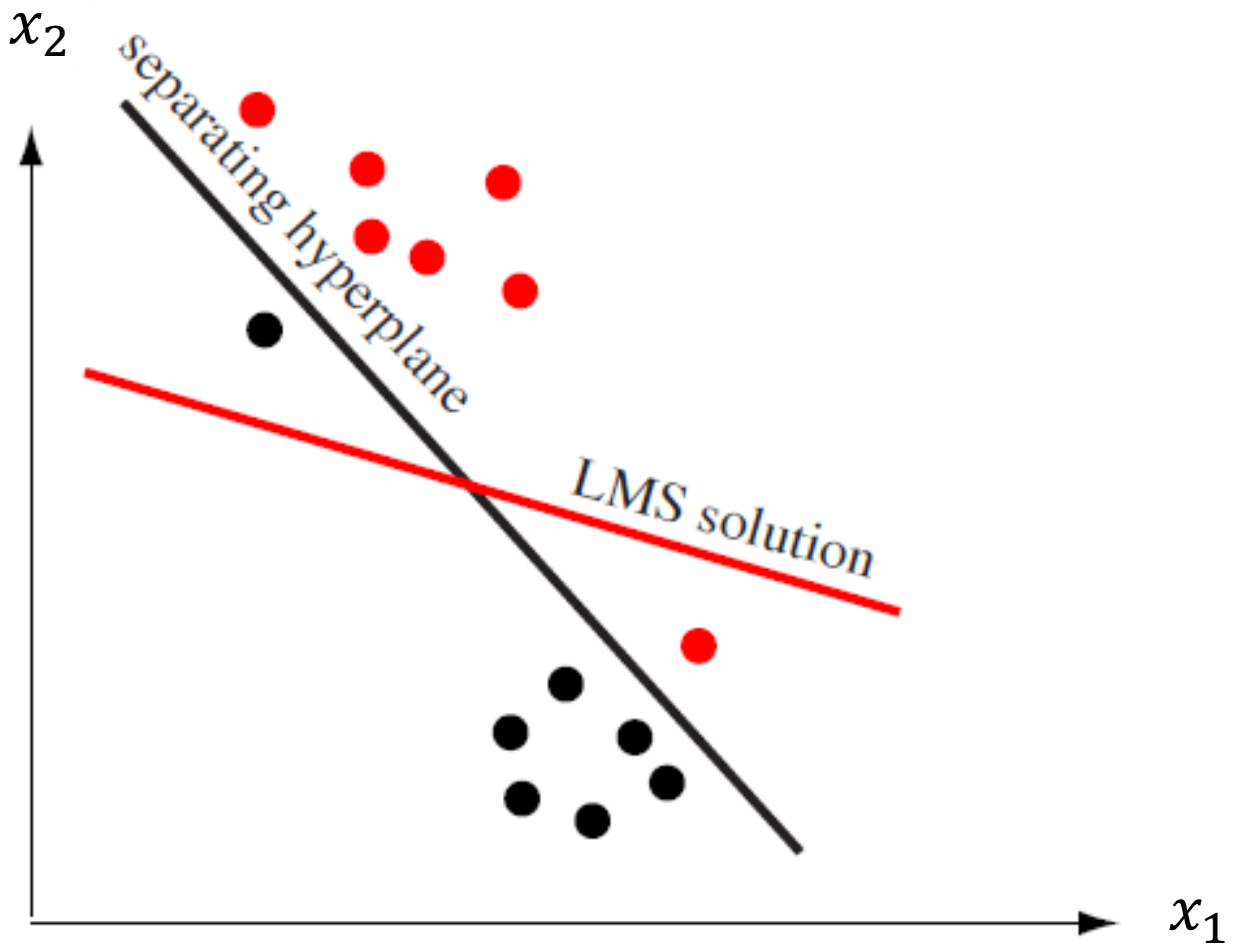
$$\mathbf{a}(1)\text{: 任意}$$

  - 优点：（1）避免了$\mathbf{Y}^{\mathrm{T}}\mathbf{Y}$是奇异矩阵所带来的问题；
    （2）避免了大矩阵的逆运算。
  - 通过将样本逐个输入并更新$\mathbf{a}$，可进一步提高效率，由此获得如下的Widrow-Hoff过程，也称为最小均方过程:

$$\mathbf{a}(k+1) = \mathbf{a}(k) - \eta(k)\left(\mathbf{a}^{\mathrm{T}}\mathbf{y}_k - b_k\right)\mathbf{y}_k$$

最小平方误差法

- LMS算法未必收敛于分类超平面，即使这个平面存在。

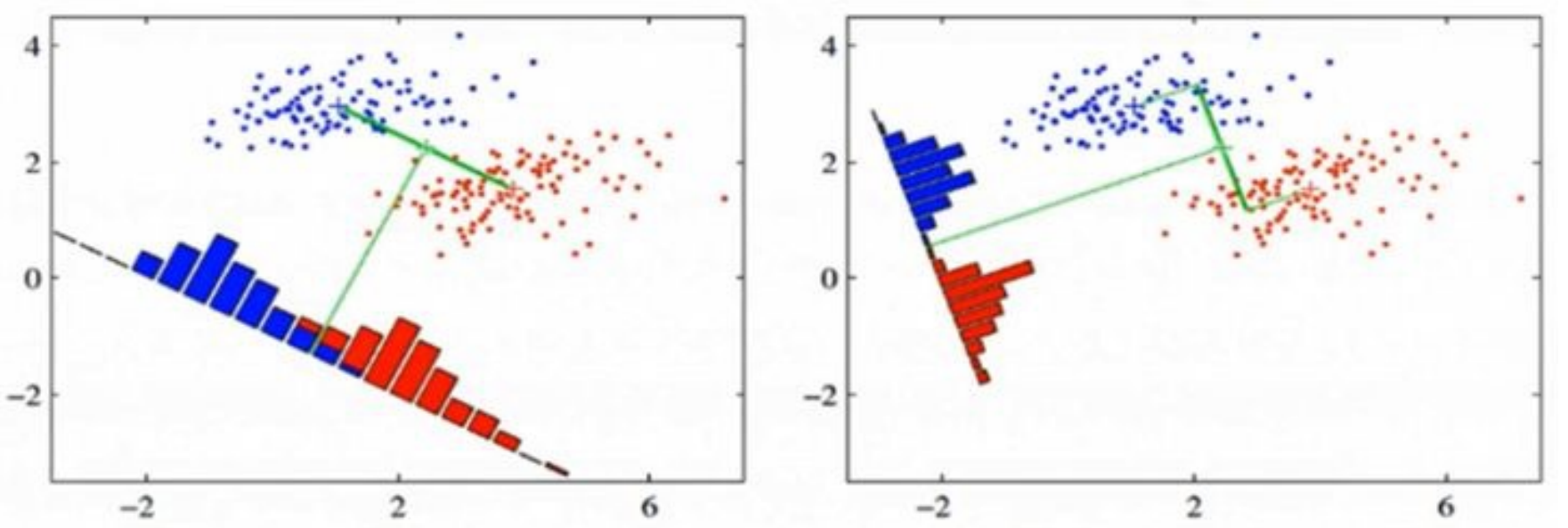


第二部分：线性判别分析

多重线性判别分析

- 线性判别分析也称Fisher线性判别分析。其目的是寻找有利于分类的判别性方向（判别性的特征向量投影方向）。
- 假设我们有一组n个d维训练样本$\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n$，它们分属于$C$个不同类别。其中容量$n_i$的样本子集$D_i$属于类别$\omega_i$。构造$\mathbf{x}_k$中各分量的线性组合：$y_k = \mathbf{w}^{\mathrm{T}}\mathbf{x}_k$。
- 对应的n个结果$\{y_1, y_2, \ldots, y_n\}$，如果$\|\mathbf{w}\| = 1$，那么$y_k$就是把$\mathbf{x}_k$向$\mathbf{w}$方向投影的结果，我们希望投影后的$y_k$有利于分类。

多重线性判别分析



右图中的投影方向使得投影后的点比左边更容易分开。

多重线性判别分析

- 对于c类问题，将Fisher线性判别将产生$c-1$个判别性投影方向。也就是说，需要从d维空间向$c-1$维空间投影。
- 此时，类内散布为：

$$\mathbf{S}_W = \sum_{i=1}^{c} \mathbf{S}_i$$

其中

$$\mathbf{S}_i = \sum_{\mathbf{x} \in D_i} (\mathbf{x} - \mathbf{m}_i)(\mathbf{x} - \mathbf{m}_i)^{\mathrm{T}}$$

$$\mathbf{m}_i = \frac{1}{n_i} \sum_{\mathbf{x} \in D_i} \mathbf{x}$$

多重线性判别分析

- 总的均值向量为：

$$\mathbf{m} = \frac{1}{n}\sum_{\mathbf{x}\in D}\mathbf{x}$$

- 总散布矩阵：

$$\mathbf{S}_T = \sum_{\mathbf{x}\in D}(\mathbf{x}-\mathbf{m})(\mathbf{x}-\mathbf{m})^{\mathrm{T}}$$

$$= \sum_{i=1}^{c}\sum_{\mathbf{x}\in D_i}(\mathbf{x}-\mathbf{m}_i+\mathbf{m}_i-\mathbf{m})(\mathbf{x}-\mathbf{m}_i+\mathbf{m}_i-\mathbf{m})^{\mathrm{T}}$$

$$= \sum_{i=1}^{c}\sum_{\mathbf{x}\in D_i}(\mathbf{x}-\mathbf{m}_i)(\mathbf{x}-\mathbf{m}_i)^{\mathrm{T}} + \sum_{i=1}^{c}\sum_{\mathbf{x}\in D_i}(\mathbf{m}_i-\mathbf{m})(\mathbf{m}_i-\mathbf{m})^{\mathrm{T}}$$

$$= \mathbf{S}_W + \sum_{i=1}^{c} n_i\,(\mathbf{m}_i-\mathbf{m})(\mathbf{m}_i-\mathbf{m})^{\mathrm{T}}$$

多重线性判别分析

- 定义

$$\mathbf{S}_B = \sum_{i=1}^{c} n_i \left(\mathbf{m}_i - \mathbf{m}\right)\left(\mathbf{m}_i - \mathbf{m}\right)^{\mathrm{T}}$$

则

$$\mathbf{S}_T = \mathbf{S}_W + \mathbf{S}_B$$

- 通过下列的$c-1$个判别函数将训练样本从d维空间向$c-1$维空间投影：

$$y_i = \mathbf{w}_i^{\mathrm{T}}\mathbf{x},\ i = 1, \ldots, c-1$$

其中y_i可以看作是一个$c-1$维向量$\mathbf{y}$的分量。

多重线性判别分析

- $\mathbf{w}_i$可以看作是一个$d\times(c-1)$矩阵$\mathbf{W}$的列向量。则

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

- 对原始样本$\mathbf{x}_1,\ldots,\mathbf{x}_n$进行投影后，得到一组新的$c-1$维样本$\mathbf{y}_1,\ldots,\mathbf{y}_n$。这些新样本本身又具有它们自己的均值向量：

$$\tilde{\mathbf{m}}_i=\frac{1}{n_i}\sum_{\mathbf{y}\in Y_i}\mathbf{y}=\mathbf{W}^{\mathrm{T}}\mathbf{m}_i,\qquad \tilde{\mathbf{m}}=\frac{1}{n}\sum_{\mathbf{y}\in Y}\mathbf{y}=\mathbf{W}^{\mathrm{T}}\mathbf{m}$$

和散布矩阵：

$$\tilde{\mathbf{S}}_W=\sum_{i=1}^{c}\sum_{y\in Y_i}(\mathbf{y}-\tilde{\mathbf{m}}_i)(\mathbf{y}-\tilde{\mathbf{m}}_i)^{\mathrm{T}}=\mathbf{W}^{\mathrm{T}}\mathbf{S}_W\mathbf{W}$$

$$\tilde{\mathbf{S}}_B=\sum_{i=1}^{c}n_i(\tilde{\mathbf{m}}_i-\tilde{\mathbf{m}})(\tilde{\mathbf{m}}_i-\tilde{\mathbf{m}})^{\mathrm{T}}=\mathbf{W}^{\mathrm{T}}\mathbf{S}_B\mathbf{W}$$

多重线性判别分析

- 散布矩阵的行列式是其散布程度的标量化度量。
- 我们希望找到一个变换矩阵$\mathbf{W}$，能够使得变换（投影）后类间散布和类内散布的行列式比值最大：

$$J(\mathbf{W}) = \frac{\left|\tilde{\mathbf{S}}_B\right|}{\left|\tilde{\mathbf{S}}_W\right|} = \frac{\left|\mathbf{W}^{\mathrm{T}}\mathbf{S}_B\mathbf{W}\right|}{\left|\mathbf{W}^{\mathrm{T}}\mathbf{S}_W\mathbf{W}\right|}$$

- 最优矩阵$\mathbf{W}$的各列向量是下列方程中最大的$c-1$个本征值对应的本征向量：

$$\mathbf{S}_B\mathbf{w}_i = \lambda_i\mathbf{S}_W\mathbf{w}_i$$

多重线性判别分析

- 如果$\mathbf{S}_W$是非奇异的，则：

$$\mathbf{S}_W^{-1}\mathbf{S}_B\mathbf{w}_i = \lambda_i\mathbf{w}_i$$

这是一个典型的本征值问题。不过其计算并不方便。

- 我们先通过求解多项式的根来求解本征值：

$$|\mathbf{S}_B - \lambda_i\mathbf{S}_W| = 0$$

并进一步通过方程

$$(\mathbf{S}_B - \lambda_i\mathbf{S}_W)\mathbf{w}_i = \mathbf{0}$$

求解$\mathbf{w}_i$。

第三部分：支持向量机

支持向量机

- 学习与经验风险最小化：基于机器学习的分类器需要从有限的训练样本中学习判别函数$g(\mathbf{x})$，这一目标一般通过最小化某类误差函数，比如训练集上的经验风险，来达到：

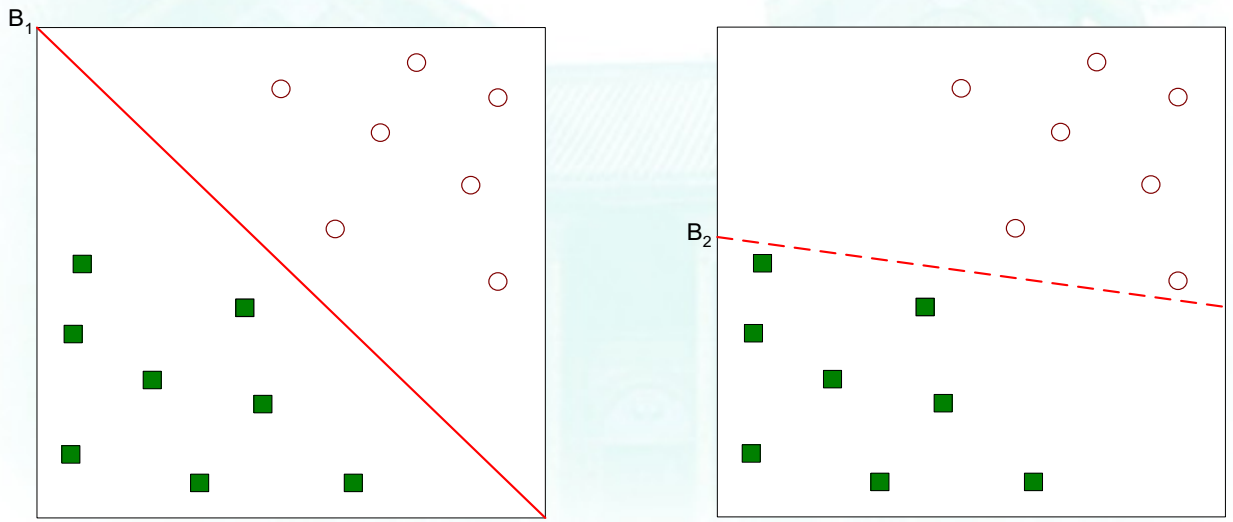
$$R_{emp}(\mathbf{w}, w_0) = \frac{1}{n}\sum_{k=1}^{n}[z_k - g(\mathbf{x}_k; \mathbf{w}, w_0)]^2$$

其中类标签

$$z_k = \begin{cases} +1, & \text{如果}\mathbf{x}_k \in \omega_1 \\ -1, & \text{如果}\mathbf{x}_k \in \omega_2 \end{cases}$$

支持向量机

- 但是传统的在训练数据上最小化经验风险的方法对新的测试数据泛化性能不佳：
  - 可能存在许多不同的函数，都能很好地逼近训练数据集
  - 难以确定哪个函数能最佳刻画数据分布的真实结构



哪个解更好？

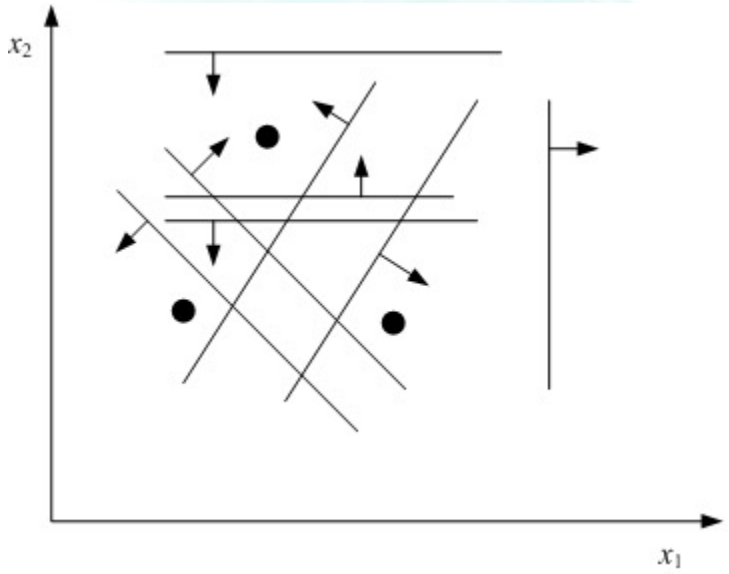
支持向量机

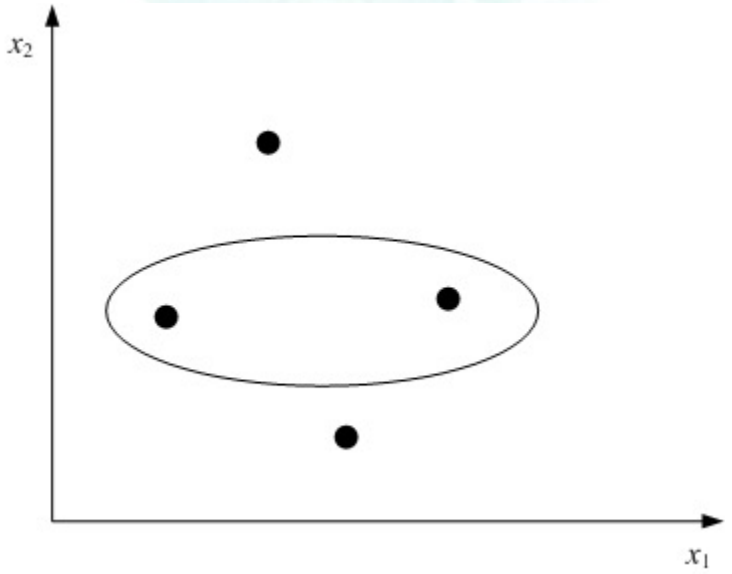
- 统计学习中的容量和VC维
  - 模型容量反映其对给定数据集的划分能力；
  - 为了确保泛化误差的上界，必须控制模型的容量。

- 在统计学习中，Vapnik-Chervonenkis (VC)维是最常用的模型容量度量。它反映了函数集的学习能力，VC维越大则学习机器容量越大。
  - VC维就是某模型对应的分类器集合能够粉碎的最多样本数，即最大的分类能力；
  - 粉碎某个样本集就是说能表达该样本集的所有可能对分。

支持向量机

- 简单函数集的VC维实例
- 线性指示函数($d = 2$)：存在可以被粉碎的3个点，但找不到这样的4个点。
  $\Rightarrow$ VC维 = 3。一般情况下，其VC维 = $d + 1$





支持向量机

- 结构化风险最小化：满足以下条件的函数具有好的泛化性
  1. 经验风险小
  2. VC维低
- 真实错误率与经验错误率关系：下面的不等式以概率$1-\delta$成立

$$err_{true} \leq err_{train} + \sqrt{\frac{\mathrm{VC}(\log(2n/\mathrm{VC})+1)-\log(\delta/4)}{n}}$$

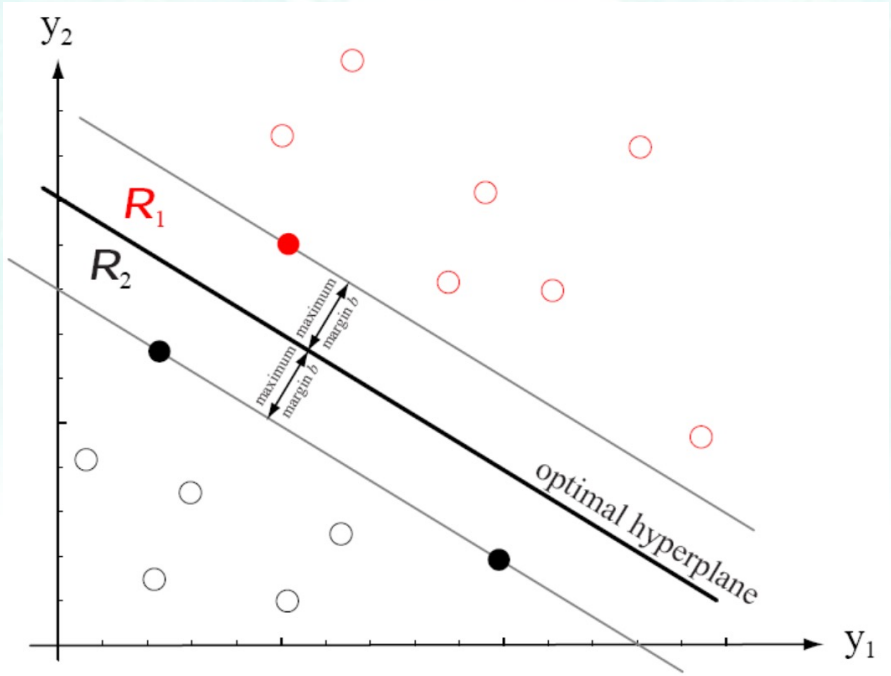
其中n为训练样本容量。

(Vapnik,"Structural Risk Minimization Principle", 1995.)

- 分离间隔与最优超平面：
  - Vapnik证明了最大化类间的分离间隔等价于最小化VC维
  - 最优超平面即为具有最大类间分离间隔的分类面

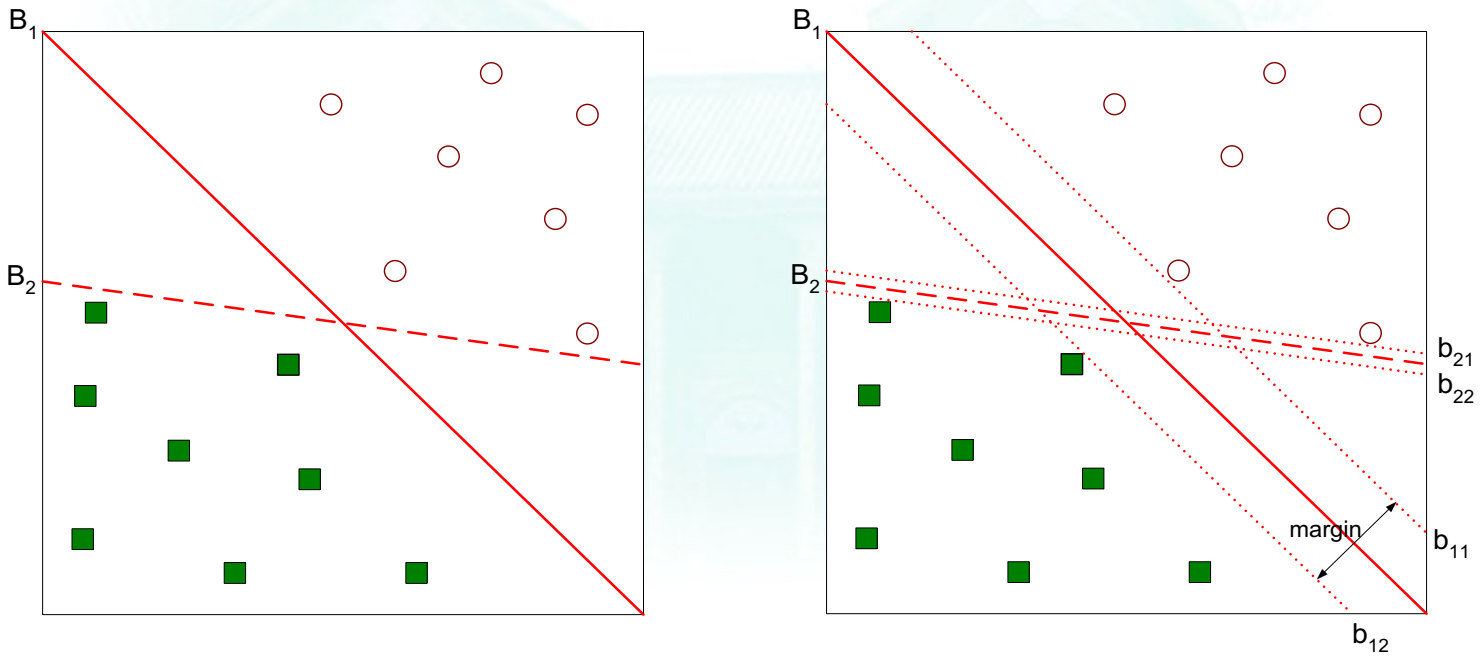
支持向量机

- 分离间隔与支持向量
  - 分离间隔：由决策边界到最近的训练样本（支持向量）的距离定义的空白区域
  - 支持向量：定义最优分类超平面的训练样本，对应于最难被分类的样本



支持向量机

- 训练支持向量机的目标是找到一个具有类间最大间隔的分类超平面。间隔越大，分类器的泛化性也越好。
- 其训练目标等价于求解具有线性约束的二次规划问题。
- 其基本形式是二分类，但可以扩展成多类。



支持向量机

- 先考虑可分情况下的线性支持向量机：采用如下的线性判别函数

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

判决准则：如果$g(\mathbf{x}) > 0$，则将$\mathbf{x}$判为ω_1类，如果$g(\mathbf{x}) < 0$，则将$\mathbf{x}$判为ω_2类。

- 给定训练样本$\mathbf{x}_k$的类标签：

$$z_k = \begin{cases} +1 & \text{如果} \mathbf{x}_k \in \omega_1 \\ -1 & \text{如果} \mathbf{x}_k \in \omega_2 \end{cases}$$

- 训练时，分类模型约束条件的规范化形式：

$$z_k g(\mathbf{x}_k) > 0$$

即

$$z_k\left(\mathbf{w}^{\mathrm{T}}\mathbf{x}_k + w_0\right) > 0$$

其中$k = 1,2,\ldots,n$。

支持向量机

- 点$\mathbf{x}_k$距离分类超平面的距离应该满足以下约束：

$$\frac{z_k g(\mathbf{x}_k)}{\|\mathbf{w}\|} \geq b, \qquad b > 0$$

- 为了确保唯一性：

$$b\|\mathbf{w}\| = 1$$

- 上述约束条件变为：

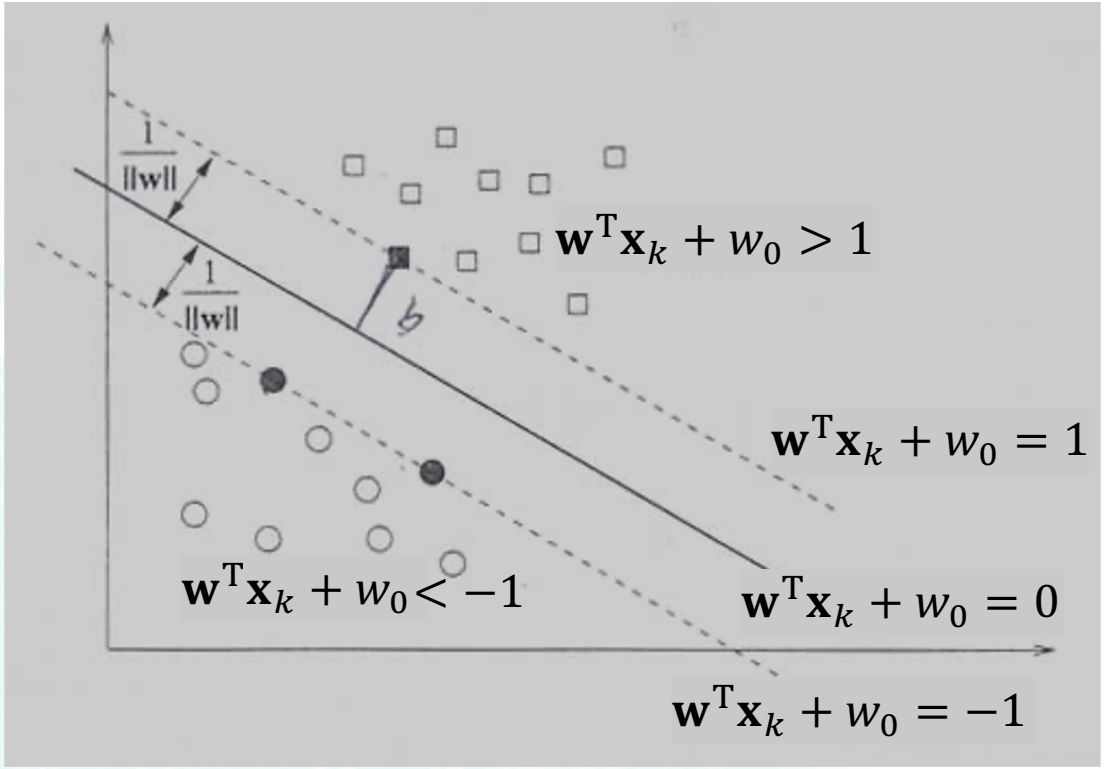
$$\frac{z_k g(\mathbf{x}_k)}{\|\mathbf{w}\|} \geq b = \frac{1}{\|\mathbf{w}\|}$$

$$\Rightarrow z_k g(\mathbf{x}_k) \geq 1$$

支持向量机

最大化间隔：

$$\frac{2}{||\mathbf{w}||}$$



问题1：$\min_{\mathbf{w}} \frac{1}{2}\|\mathbf{w}\|^2$

$s.t.\ z_k(\mathbf{w}^{\mathrm{T}}\mathbf{x}_k + w_0) \geq 1,\ k = 1,2,\ldots,n$

二次规划问题！

支持向量机

- 使用拉格朗日优化：

$$L(\mathbf{w}, w_0, \boldsymbol{\lambda}) = \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{k=1}^{n} \lambda_k \left[z_k \left(\mathbf{w}^{\mathrm{T}} \mathbf{x}_k + w_0 \right) - 1 \right],$$

$$\lambda_k \geq 0$$

- 为获得$\min_{\mathbf{w}, w_0} \max_{\boldsymbol{\lambda}} L$，$L$对$\mathbf{w}$和$w_0$的梯度应该是$\mathbf{0}$：

$$\mathbf{w} = \sum_{k=1}^{n} \lambda_k z_k \mathbf{x}_k \text{ 且 } \sum_{k=1}^{n} \lambda_k z_k = 0$$

- 把它们代入拉格朗日形式，我们可以得到一个更容易解决的对偶问题（问题2）：

$$\max_{\boldsymbol{\lambda}} \sum_{k=1}^{n} \lambda_k - \frac{1}{2} \sum_{k,j=1}^{n} \lambda_k \lambda_j \, z_k z_j \mathbf{x}_j^{\mathrm{T}} \mathbf{x}_k$$

$$\text{s.t.} \sum_{k=1}^{n} z_k \lambda_k = 0, \lambda_k \geq 0, k = 1,2,\ldots,n$$

支持向量机

- 根据KKT(Karush－Kuhn－Tucker)条件，我们知道：

$$\lambda_k\left(z_k(\mathbf{w}^{\mathrm{T}}\mathbf{x}_k + w_0) - 1\right) = 0$$

因此，如果$\mathbf{x}_k$不是支持向量，那么$\lambda_k = 0$，即问题的解完全由支持向量决定。

- 问题的解由以下式子给出：

$$\mathbf{w} = \sum_{k=1}^{n} \lambda_k z_k \mathbf{x}_k$$

$$w_0 = z_k - \mathbf{w}^{\mathrm{T}}\mathbf{x}_k \quad （对应\lambda_k \neq 0的\mathbf{x}_k）$$

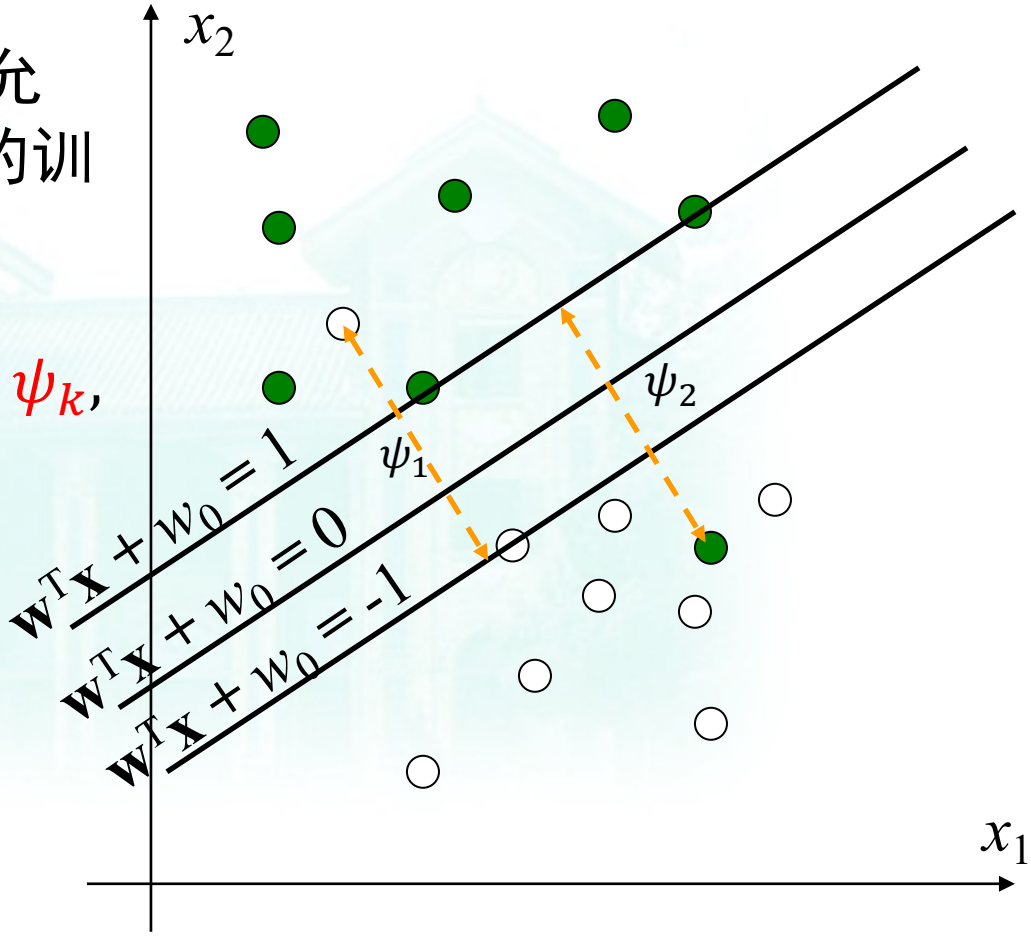
$$g(\mathbf{x}) = \sum_{k=1}^{n} \lambda_k z_k (\mathbf{x}_k^{\mathrm{T}}\mathbf{x}) + w_0$$

支持向量机

- 线性支持向量机：不可分情况

引入松弛变量ψ_k，以允许部分困难或带噪声的训练数据被错分。

$$z_k(\mathbf{w}^{\mathrm{T}}\mathbf{x}_k + w_0) \geq 1 - \psi_k,$$
$$k = 1,2,\ldots,n$$

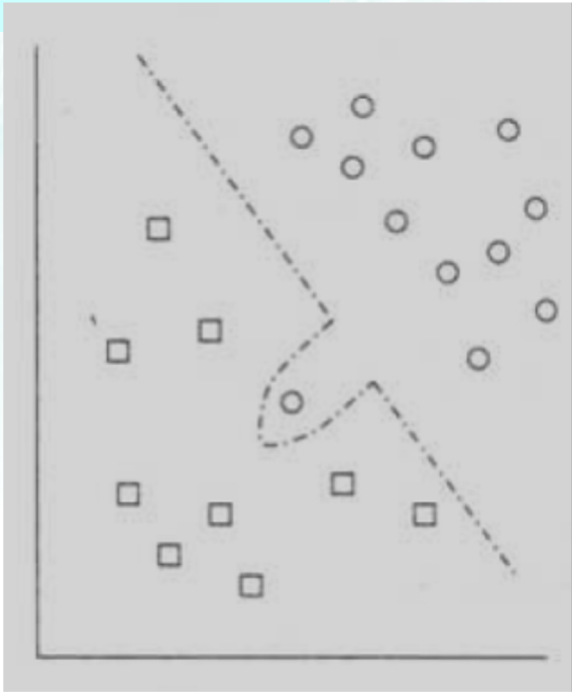


支持向量机

问题3：

$$\min_{\mathbf{w},\boldsymbol{\psi}} \frac{1}{2}\|\mathbf{w}\|^2 + c\sum_{k=1}^{n}\psi_k$$
$$s.t.\ z_k\left(\mathbf{w}^{\mathrm{T}}\mathbf{x}_k + w_0\right) \geq 1 - \psi_k,$$
$$\psi_k \geq 0,\ k = 1,2,\ldots,n$$

- 其目标是在最大化可分样本间隔的同时，最小化错误ψ_k之和。
- 参数c为正则化系数，控制模型复杂度与训练误差之间的折中。



支持向量机

- 我们可以将问题3转换为以下的最大化问题（对偶问题）：

$$\text{问题4：} \max_{\boldsymbol{\lambda}} \sum_{k=1}^{n} \lambda_k - \frac{1}{2} \sum_{k,j=1}^{n} \lambda_k \lambda_j z_k z_j \mathbf{x}_j^{\mathrm{T}} \mathbf{x}_k$$

$$s.t. \sum_{k=1}^{n} z_k \lambda_k = 0, 0 \le \lambda_k \le c, k = 1,2,\ldots,n$$

其中误差变量ψ_k的作用是将拉格朗日系数λ_k限制在0到c之间。

- KKT条件：

$$\lambda_k \left(z_k (\mathbf{w}^{\mathrm{T}} \mathbf{x}_k + w_0) - 1 + \psi_k \right) = 0$$
$$(c - \lambda_k)\psi_k = 0$$

- 与可分情形类似，对应非零λ_k的$\mathbf{x}_k$被称为支持向量。

支持向量机

$$\lambda_k\left(z_k(\mathbf{w}^{\mathrm{T}}\mathbf{x}_k + w_0) - 1 + \psi_k\right) = 0$$
$$(c - \lambda_k)\psi_k = 0$$

- 两种情况
  - $0 < \lambda_k < c$
    - $\psi_k = 0$： 支持向量到最优分类超平面距离为 $1/\|\mathbf{w}\|$，称为边界向量。
  - $\lambda_k = c$
    - $\psi_k > 1$：错分样本
    - $0 < \psi_k \leq 1$：正确分类，但是到最优分类超平面的距离小于$1/\|\mathbf{w}\|$
    - $\psi_k = 0$：边界向量（稀有情况）
- 忽略最后的稀有情况，我们将所有$\lambda_k = c$的支持向量视为存在错误。
- 所有不是支持向量的样本都被正确分类并位于间隔带之外。

第四部分：非线性分类

广义线性判别函数

- 现实问题中的分类边界常常是非线性的，简单的线性判别函数不能表达现实中的复杂情况。
- 可将线性判别函数推广为更一般的广义线性判别函数：

$$g(\mathbf{x}) = a_1 f_1(\mathbf{x}) + a_2 f_2(\mathbf{x}) + \cdots + a_N f_N(\mathbf{x}) + a_{N+1}$$

其中$f_1(\mathbf{x}), 1 \le i \le N$，是模式向量$\mathbf{x}$的标量函数。

- 引入$f_{N+1}(\mathbf{x}) = 1$，我们得到：

$$g(\mathbf{x}) = \sum_{i=1}^{N+1} a_i f_i(\mathbf{x}) = \mathbf{a}^{\mathrm{T}}\mathbf{y}$$

其中$\mathbf{a} = [a_1, a_2, \ldots, a_N, a_{N+1}]^{\mathrm{T}}$

$$\mathbf{y} = [f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_N(\mathbf{x}), f_{N+1}(\mathbf{x})]^{\mathrm{T}}$$

$g(\mathbf{x})$的后一种形式意味着任何广义线性判别函数可视为$N+1$维空间中的线性函数。

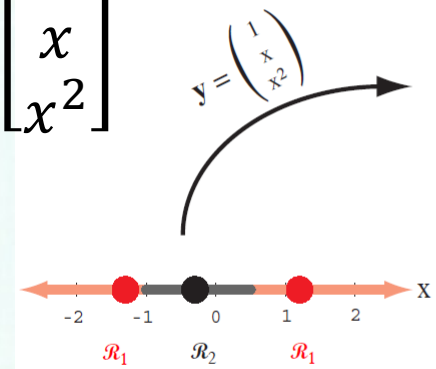
  - $g(\mathbf{x})$在R^d中保持其非线性特性。

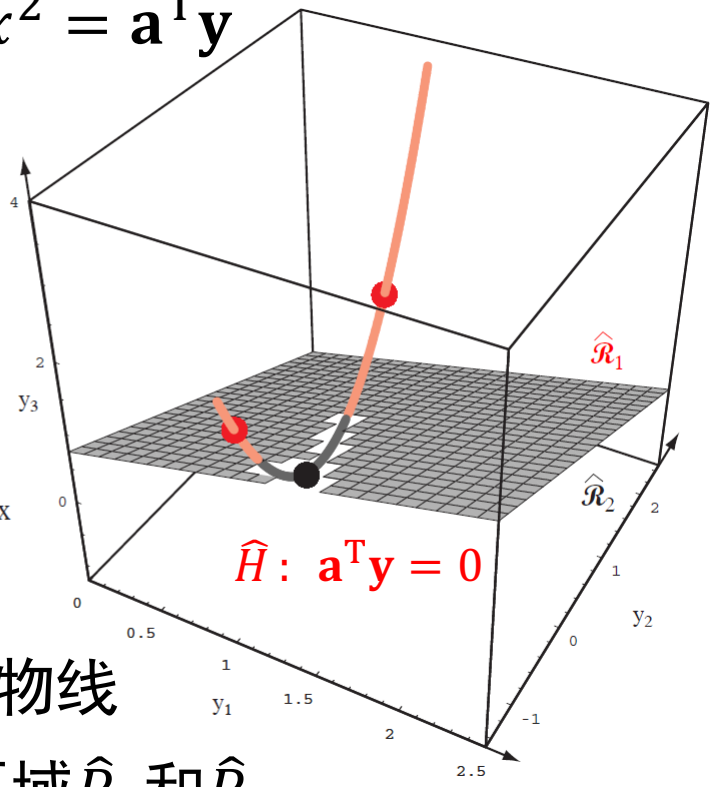
广义线性判别函数

- 广义线性判别函数在变换空间中利用经过原点的超平面来进行样本分类。
- 例子：

$$g(x) = -1 + x + 2x^2 = \mathbf{a}^{\mathrm{T}}\mathbf{y}$$

$$\mathbf{a} = \begin{bmatrix} -1 \\ 1 \\ 2 \end{bmatrix}, \mathbf{y} = \begin{bmatrix} 1 \\ x \\ x^2 \end{bmatrix}$$





- 将直线映射为三维空间中的抛物线
- 平面$\hat{H}$将$\mathbf{y}$空间分成两个判决区域$\hat{R}_1$和$\hat{R}_2$
- 而原始x空间中相应的判决域R_1和R_2非简单连通

广义线性判别函数

- 最常用的广义线性判别函数为多项式判别函数，即$f_i(\mathbf{x})(1 \leq i \leq N)$为关于$\mathbf{x}$中各分量的多项式。
  - 比如二维特征空间中的二次判别函数：

$$g(\mathbf{x}) = a_1 x_1^2 + a_2 x_1 x_2 + a_3 x_2^2 + a_4 x_1 + a_5 x_2 + a_6$$

其中：$\mathbf{a} = [a_1, a_2, \ldots, a_6]^{\mathrm{T}}$

$$\mathbf{y} = [x_1^2, x_1 x_2, x_2^2, x_1, x_2, 1]^{\mathrm{T}}$$

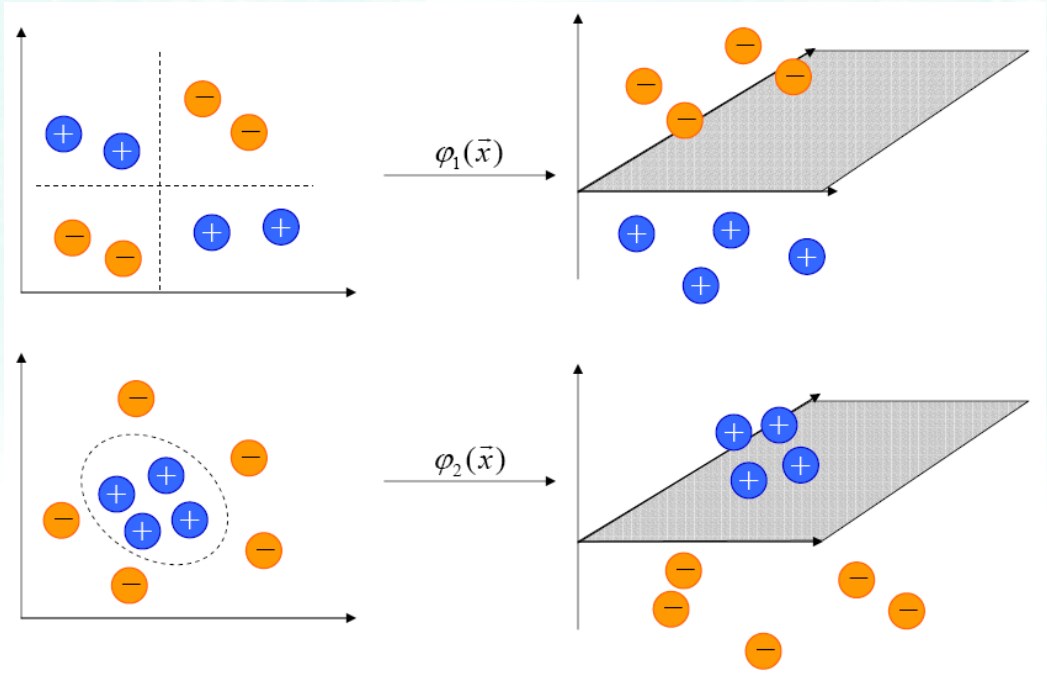
非线性支持向量机

- 我们也可以利用非线性映射的思想对线性支持向量机进行推广：

$$\mathbf{x}_k \rightarrow \varphi(\mathbf{x}_k)$$

其中$\varphi(\cdot)$为设定的非线性映射函数。

  - 通过适当的非线性映射$\varphi(\cdot)$将数据映射到足够高维的空间中，两类数据总是可以被一个超平面分开。



Linearly Separable in Higher Dimension

非线性支持向量机

- 此时，最优超平面的决策函数为：

$$g(\mathbf{x}) = \sum_{k \in \mathrm{SV}} \lambda_k z_k \left(\varphi^{\mathrm{T}}(\mathbf{x}_k) \varphi(\mathbf{x}) \right) + w_0$$

其中SV表示支持向量索引集。

- 决策规则与此前相同：
  - 如果$g(\mathbf{x}) > 0$则将$\mathbf{x}$归类为ω_1，否则归类为ω_2。

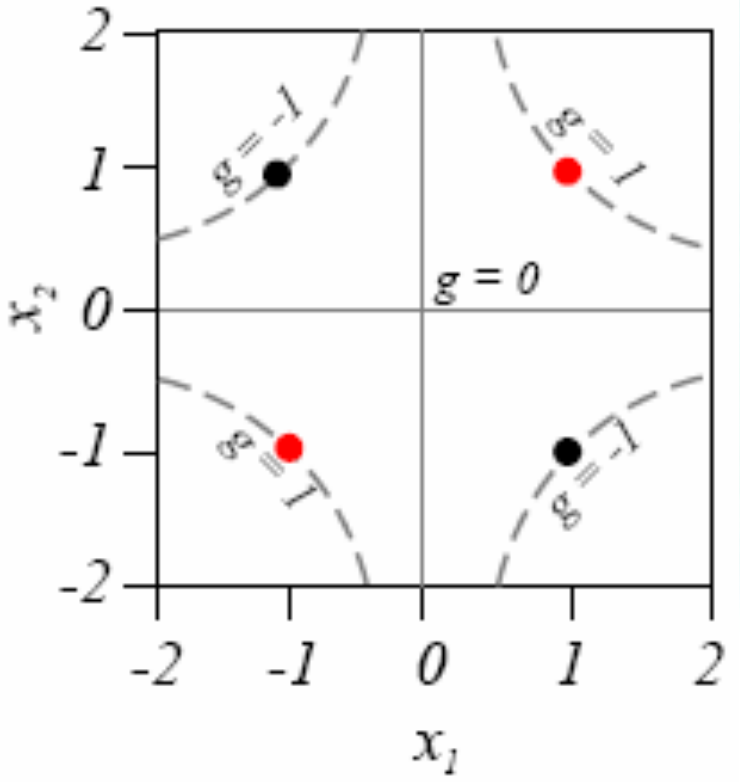
- 这种直接映射的缺点是可能大幅增加计算量。

非线性支持向量机

- 例子：异或问题(XOR)是最简单的非线性可分问题之一

$(1,1)$与$(-1,-1)$属于ω_1

$(1,-1)$与$(-1,1)$属于ω_2



非线性支持向量机

考虑以下映射（还存在许多其他可能映射）：

$$\mathbf{y} = \varphi(\mathbf{x}) = \begin{bmatrix} x_1^2 \\ \sqrt{2}\, x_1 \\ \sqrt{2}\, x_1 x_2 \\ \sqrt{2}\, x_2 \\ x_2^2 \\ 1 \end{bmatrix}$$

上述变换将$\mathbf{x}_k$映射到一个6维空间中：

$$\mathbf{y}_1 = \varphi(\mathbf{x}_1) = \begin{bmatrix} 1 \\ \sqrt{2} \\ \sqrt{2} \\ \sqrt{2} \\ 1 \\ 1 \end{bmatrix} \quad \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

$$\mathbf{y}_3 = \varphi(\mathbf{x}_3) = \begin{bmatrix} 1 \\ -\sqrt{2} \\ \sqrt{2} \\ -\sqrt{2} \\ 1 \\ 1 \end{bmatrix} \quad \begin{bmatrix} -1 \\ -1 \end{bmatrix}$$

$$\mathbf{y}_2 = \varphi(\mathbf{x}_2) = \begin{bmatrix} 1 \\ \sqrt{2} \\ -\sqrt{2} \\ -\sqrt{2} \\ 1 \\ 1 \end{bmatrix} \quad \begin{bmatrix} 1 \\ -1 \end{bmatrix}$$

$$\mathbf{y}_4 = \varphi(\mathbf{x}_4) = \begin{bmatrix} 1 \\ -\sqrt{2} \\ -\sqrt{2} \\ \sqrt{2} \\ 1 \\ 1 \end{bmatrix} \quad \begin{bmatrix} -1 \\ 1 \end{bmatrix}$$

非线性支持向量机

我们寻求最大化：

$$\sum_{k=1}^{4}\lambda_k-\frac{1}{2}\sum_{k,j=1}^{4}\lambda_k\lambda_j z_k z_j\varphi^{\mathrm{T}}(\mathbf{x}_j)\varphi(\mathbf{x}_k)$$

$$s.t.\sum_{k=1}^{4}z_k\lambda_k=0,\lambda_k\geq 0,k=1,2,..,4$$

对上式求解得：

$$\lambda_1=\lambda_2=\ \lambda_3=\lambda_4=\frac{1}{8}$$

由于所有$\lambda_k\neq 0$，因此所有的$\mathbf{x}_k$都是支持向量。

非线性支持向量机

下面计算$\mathbf{w}$：

$$\mathbf{w}=\sum_{k=1}^{4}\lambda_k z_k \varphi(\mathbf{x}_k)=\frac{1}{8}\begin{pmatrix}1\\ \sqrt{2}\\ \sqrt{2}\\ \sqrt{2}\\ 1\\ 1\end{pmatrix}-\frac{1}{8}\begin{pmatrix}1\\ \sqrt{2}\\ -\sqrt{2}\\ -\sqrt{2}\\ 1\\ 1\end{pmatrix}+\frac{1}{8}\begin{pmatrix}1\\ -\sqrt{2}\\ \sqrt{2}\\ -\sqrt{2}\\ 1\\ 1\end{pmatrix}-\frac{1}{8}\begin{pmatrix}1\\ -\sqrt{2}\\ -\sqrt{2}\\ \sqrt{2}\\ 1\\ 1\end{pmatrix}=\frac{1}{2}\begin{pmatrix}0\\ 0\\ \sqrt{2}\\ 0\\ 0\\ 0\end{pmatrix}$$

w_0的解可以利用任意一个支持向量求得，例如$\mathbf{x_1}$：

$$\mathbf{w}^{\mathrm{T}}\varphi(\mathbf{x}_1)+w_0=z_1 \quad \text{或者} \quad w_0=z_1-\mathbf{w}^{\mathrm{T}}\varphi(\mathbf{x}_1)=0$$

$$\varphi(\mathbf{x}_1)=\begin{bmatrix}1\\ \sqrt{2}\\ \sqrt{2}\\ \sqrt{2}\\ 1\\ 1\end{bmatrix}$$

非线性支持向量机

决策函数为：

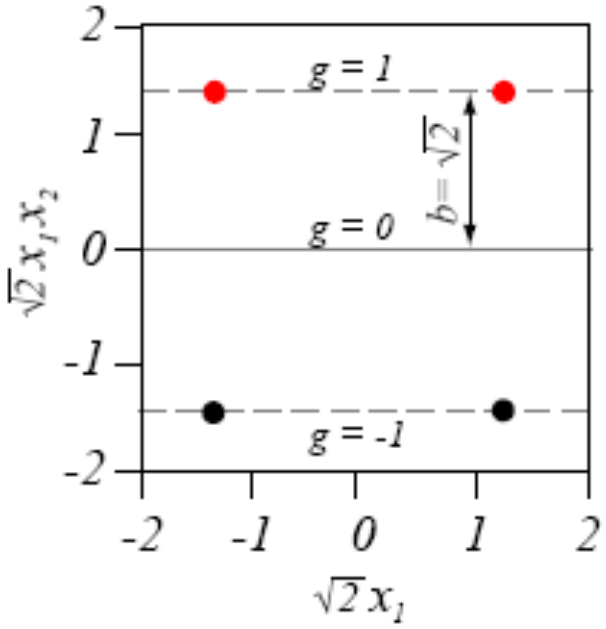
$$g(\mathbf{x}) = \mathbf{w}^{\mathrm{T}}\varphi(\mathbf{x}) + w_0 = x_1 x_2$$

如果$g(\mathbf{x}) > 0$则$\mathbf{x}$归类为ω_1，如果$g(\mathbf{x}) < 0$则$\mathbf{x}$归类为ω_2。

间隔裕量b为：

$$b = \frac{1}{\|\mathbf{w}\|} = \sqrt{2}$$

$$\varphi(\mathbf{x}) = \begin{bmatrix} x_1^2 \\ \sqrt{2}\,x_1 \\ \sqrt{2}\,x_1 x_2 \\ \sqrt{2}\,x_2 \\ x_2^2 \\ 1 \end{bmatrix}$$



支持向量机中的核技巧

- 前述非线性支持向量机的判别函数由投影空间中的特征向量内积决定：

$$g(\mathbf{x}) = \sum_{k \in \mathrm{SV}} \lambda_k z_k \left(\varphi^{\mathrm{T}}(\mathbf{x}_k) \varphi(\mathbf{x}) \right) + w_0$$

- 可引入核函数表达该内积：

$$K(\mathbf{x}, \mathbf{y}) = \varphi^{\mathrm{T}}(\mathbf{x}) \varphi(\mathbf{y})$$

- 核函数优势：
  - 不需知道$\varphi(\cdot)$，可避免直接高维运算
  - 判别式简化为：

$$g(\mathbf{x}) = \sum_{k \in \mathrm{SV}} \lambda_k z_k K(\mathbf{x}, \mathbf{x}_k) + w_0$$

$$\varphi(\mathbf{x}) = \begin{bmatrix} x_1^2 \\ \sqrt{2}\, x_1 \\ \sqrt{2}\, x_1 x_2 \\ \sqrt{2}\, x_2 \\ x_2^2 \\ 1 \end{bmatrix}$$

$$\Downarrow$$

$$K(\mathbf{x}, \mathbf{y}) = \left(\mathbf{x}^{\mathrm{T}} \mathbf{y} + 1 \right)^2$$

支持向量机中的核技巧

- 核函数的选择不是唯一的：例如考虑

$$\mathbf{x} \in R^2, \varphi(\mathbf{x}) = \begin{bmatrix} x_1^2 \\ \sqrt{2}x_1x_2 \\ x_2^2 \end{bmatrix} \in R^3, \text{ 则：} K(\mathbf{x},\mathbf{y}) = (\mathbf{x}^{\mathrm{T}}\mathbf{y})^2$$

因为：

$$\varphi^{\mathrm{T}}(\mathbf{x})\varphi(\mathbf{y}) = x_1^2y_1^2 + 2x_1x_2y_1y_2 + x_2^2y_2^2 = (x_1y_1 + x_2y_2)^2$$
$$= (\mathbf{x}^{\mathrm{T}}\mathbf{y})^2 = K(\mathbf{x},\mathbf{y})$$

当然，根据具体问题，也可以选择其它核函数。

- 对给定核函数，$\varphi(\cdot)$和高维空间也都不是唯一的：

$$\varphi(\mathbf{x}) = \frac{1}{\sqrt{2}}\begin{bmatrix} (x_1^2 - x_2^2) \\ 2x_1x_2 \\ (x_1^2 - x_2^2) \end{bmatrix} \in R^3 \text{ 或者} \varphi(\mathbf{x}) = \begin{bmatrix} x_1^2 \\ x_1x_2 \\ x_1x_2 \\ x_2^2 \end{bmatrix} \in R^4$$

上面两个映射函数也都对应核函数$K(\mathbf{x},\mathbf{y})$。

支持向量机中的核技巧

- 通过选择不同的核函数，支持向量机可实现不同的学习效果。常用的一些核函数比如：
  多项式：$K(\mathbf{x}, \mathbf{x}_k) = (\mathbf{x}^{\mathrm{T}}\mathbf{x}_k)^d$
  s型函数：$K(\mathbf{x}, \mathbf{x}_k) = \tanh(a(\mathbf{x}^{\mathrm{T}}\mathbf{x}_k) + b)$
  高斯：$K(\mathbf{x}, \mathbf{x}_k) = \exp\left(-\frac{\|\mathbf{x}-\mathbf{x}_k\|^2}{2\sigma^2}\right)$

- 算法过程：
  1. 选择一个核函数
  2. 选择目标函数中的经验误差权重c
  3. 求解二次规划问题（可利用现成软件包）
  4. 利用解出的支持向量构造判别函数

支持向量机中的核技巧

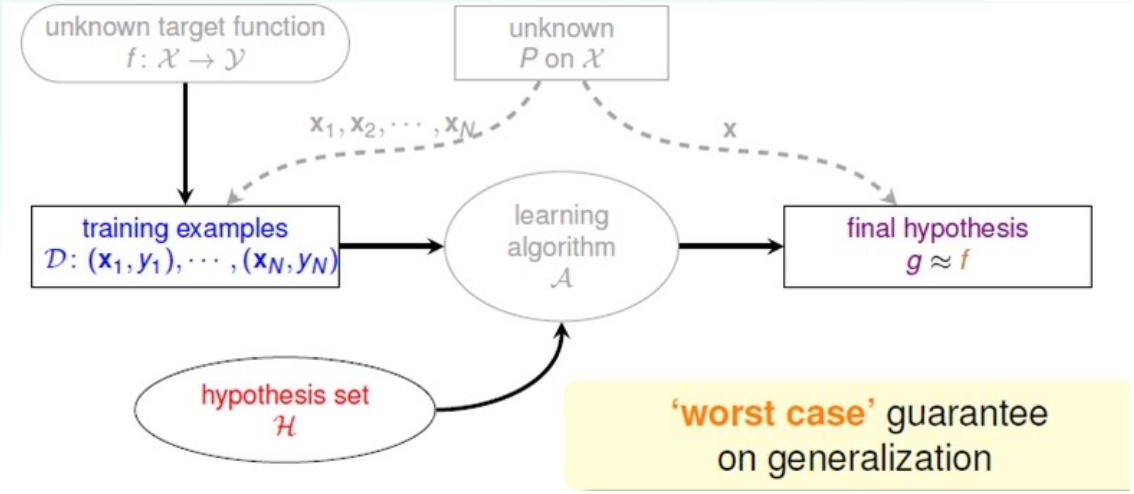
- 支持向量机的一些特点：
  - 可实现全局优化，没有局部最优；
  - 能有效避免高维空间中的过拟合现象，在小训练样本下有良好的泛化性能；
  - 其复杂度取决于支持向量的个数，而不是变换空间维度；
  - 性能取决于核函数及其参数的选择
    - 核函数不能自适应训练样本
    - 如何针对给定问题选择最佳核函数仍是一个开放问题

第五部分：VC维

(VAPNIK-CHERVONENKIS DIMENSION)

VC维定义

- 对一个指示函数集，如果存在H个样本能够被函数集中的函数按所有可能的2的H次方种形式分开，则称函数集能够把H个样本打散；
- 函数集的VC维就是它能打散的最大样本数目H。
- 若对任意数目的样本都有函数能将它们打散，则函数集的VC维是无穷大，有界实函数的VC维可以通过用一定的阈值将它转化成指示函数来定义。
- VC维反映了函数集的学习能力，VC维越大则学习机器越复杂（容量越大）



VC维的理解

- “打散”的概念同样也是针对假设集和样本。我们说“一个样本S可以被假设集H打散”当且仅当用这个假设集中的假设可以实现样本S的所有可能的“二分”，$|\Pi_H(S)|=2^m$

- “二分”指的是给定一个样本S，和一个假设，用对S进行分类的结果称为二分。因此对一个假设集合H，可以产生多种不同的“二分”，而这些二分构成了假设集H对样本S的二分集合 $\Pi_H(S)$

假设集H的VC维是能够被H打散的最大样本集的大小

$$VCdim(H)=\max\{m:\Pi_H(m)=2^m\}$$

对一个假设集合H的生长函数$\Pi_H:\mathbb{N}\to\mathbb{N}$定义为：

$$\forall m\in N,\Pi_H(m)=\max_{s\in\mathcal{X}^m}|\Pi_H(S)|$$

定义一个二分(dichotomy)的集合：$\Pi_H(S)=\{(h(x_1),\ldots,h(x_m)):h\in H\}$

每一个h对样本进行二分类，都会得到一个结果，所有的结果形成的集合即$\Pi_H(S)$，显然$|\Pi_H(S)|\leq 2^m$

Sauer引理

令H是一个假设集，且它的VC-维度$VCdim(H) = d$。那么对于所有的$m \in \mathbb{N}$，下面的不等式都成立：

$$\Pi_H(m) \leq \sum_{i=0}^{d} \begin{bmatrix} m \\ i \end{bmatrix}$$

令H是一个假设集，且它的VC-维$VCdim(H) = d$。那么对于所有的$m \geq d$，有：

$$\Pi_H(m) \leq (\frac{em}{d})^d = O(m^d)$$

VC维的用处

令H是取值为$\{-1,+1\}$的函数族，且它的VC-维$VCdim(H)=d$。则对于任意的$\delta>0$，至少以概率$1-\delta$，对于任意的$h\in H$，有：

$$R(h)\leq\widehat{R}(h)+\sqrt{\frac{2d\log\frac{em}{d}}{m}}+\sqrt{\frac{\log\frac{1}{\delta}}{2m}}$$

该泛化界也可以写作：

$$R(h)\leq\widehat{R}(h)+O(\sqrt{\frac{log(m/d)}{m/d}})$$

也就是说上界由m/d决定，m越大上界越小，d越大H越复杂，上界也越大。

泛化误差 给定一个假设$h\in H$，一个目标概念$c\in C$，以及一个潜在的分布D，则h的泛化误差或风险定义为

$$R(h)=Pr_{x\sim D}[h(x)\neq c(x)]=E_{x\sim D}[1_{h(x)\neq c(x)}]$$

经验误差 给定一个假设$h\in H$，一个目标概念$c\in C$，以及一个样本集$S=(x_1,\cdots,c_m)$，则h的经验误差或经验风险定义为

$$\hat{R}(h)=\frac{1}{m}\sum_{i=1}^{m}1_{h(x_i)\neq c(x_i)}$$

PAC学习

PAC（Probably Approximately Correct）框架借助样本复杂度（欲达到近似解所需要的样本点数目）和学习算法的时间空间复杂度（依赖于概念类计算表示的代价）来定义可学习的概念类。

如果存在一个算法$\mathcal{A}$以及一个多项式函数$poly(\cdot,\cdot,\cdot,\cdot)$使得对于任意$\epsilon > 0$以及$\delta > 0$，对于所有在$\mathcal{X}$上的分布D以及任意目标概念$c \in C$，对于满足

$m \geq poly(1/\epsilon, 1/\delta, n, size(c))$的任意样本规模$m$均有下式成立，

那么概念类C是PAC可学习 的（PAC-learnable）

$$Pr_{s\sim D^m}\left[R(h_s) \leq \epsilon\right] \geq 1-\delta$$