



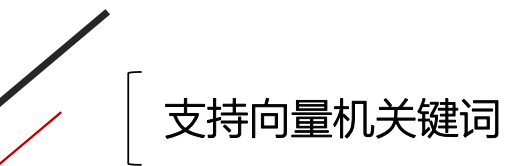




# 支持向量机

Support Vector Machines

- 
- 
- 
- 
- [ 1 ] 线性可分支持向量机与硬间隔最大化
  - [ 2 ] 线性支持向量机与软间隔最大化
  - [ 3 ] 非线性支持向量机与核技巧
  - [ 4 ] 序列最小最优化算法



二类分类模型

学习策略：间隔最大化

凸二次规划问题

间隔，对偶，核技巧

## 支持向量机分类

### 线性可分支持向量机

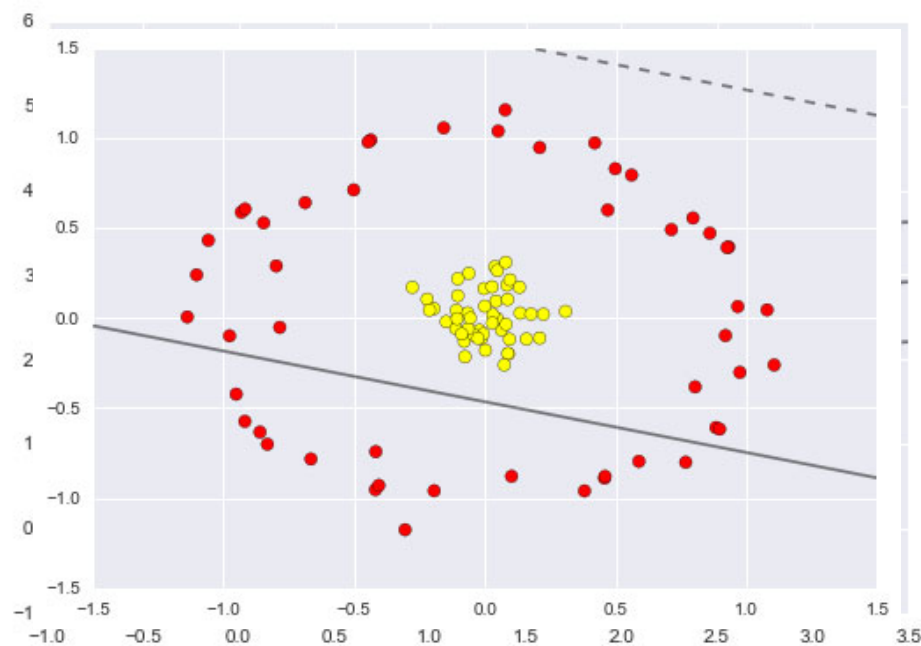
硬间隔最大化。

### 线性支持向量机

训练数据近似线性可分时，  
通过软间隔最大化

### 非线性支持向量机

当训练数据线性不可分时，  
通过使用核技巧(kernel  
trick)及软间隔最大化



# 1



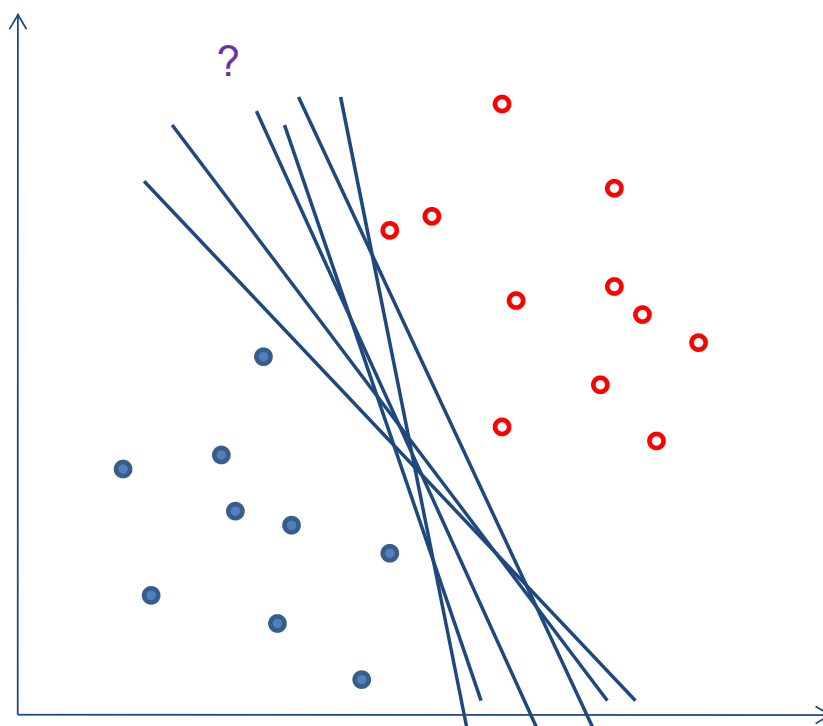
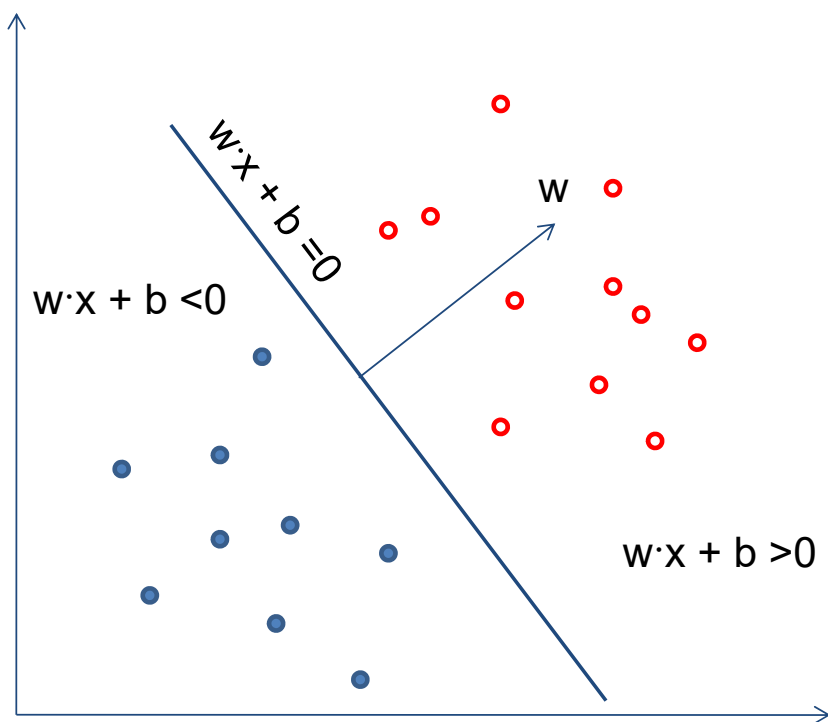
PART

线性可分支支持向量机  
与硬间隔最大化

## 7.1 线性可分支持向量机与硬间隔最大化

- 假设特征空间上的训练数据集:  $T = \{(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)\}$
- 正例和负例  $x_i \in \mathcal{X} = \mathbf{R}^n$ ,  $y_i \in \mathcal{Y} = \{+1, -1\}$ ,  $i = 1, 2, \dots, N$
- 学习的目标: 找到分类超平面,
- 线性可分支持向量机: 给定线性可分训练数据集, 通过间隔最大化或等价地求解相应的凸二次规划问题学习得到的分离超平面为  $w^* \cdot x + b^* = 0$
- 决策函数:  $f(x) = \text{sign}(w^* \cdot x + b^*)$

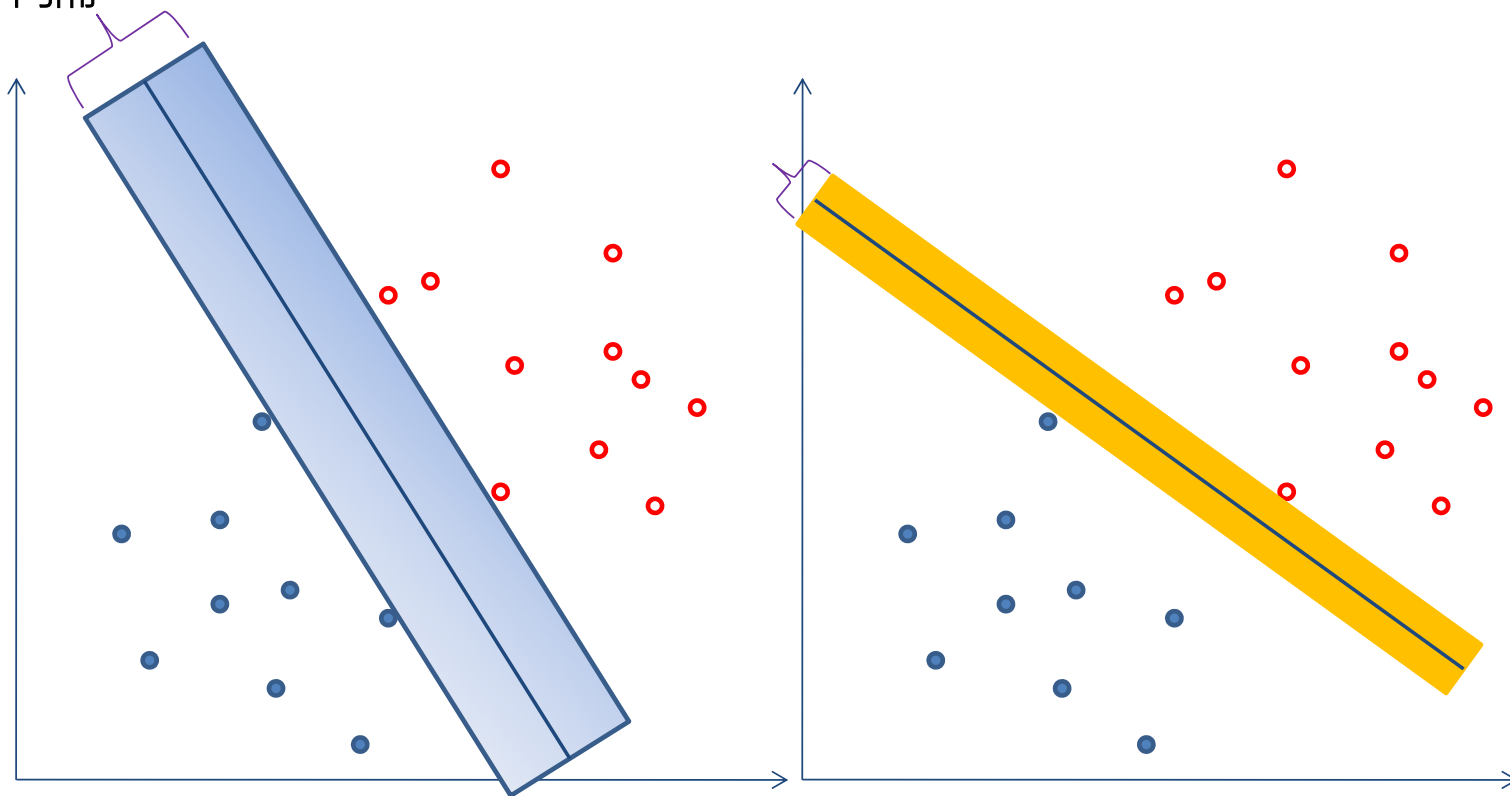
线性可分支持向量机与硬间隔最大化





线性可分支持向量机与硬间隔最大化

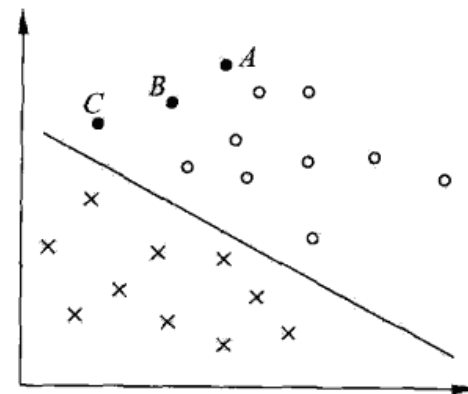
间隔





## 函数间隔和几何间隔

- 点到分离超平面的远近  $|w \cdot x + b|$
- $\rightarrow$  表示分类预测的确信程度
- $w \cdot x + b$  的符号与类标记  $y$  的符号是否一致
- $\rightarrow$  表示分类是否正确
- 所以:  $y(w \cdot x + b)$  表示分类的正确性和确信度



## 函数间隔和几何间隔

- 函数间隔

- 样本点的函数间隔

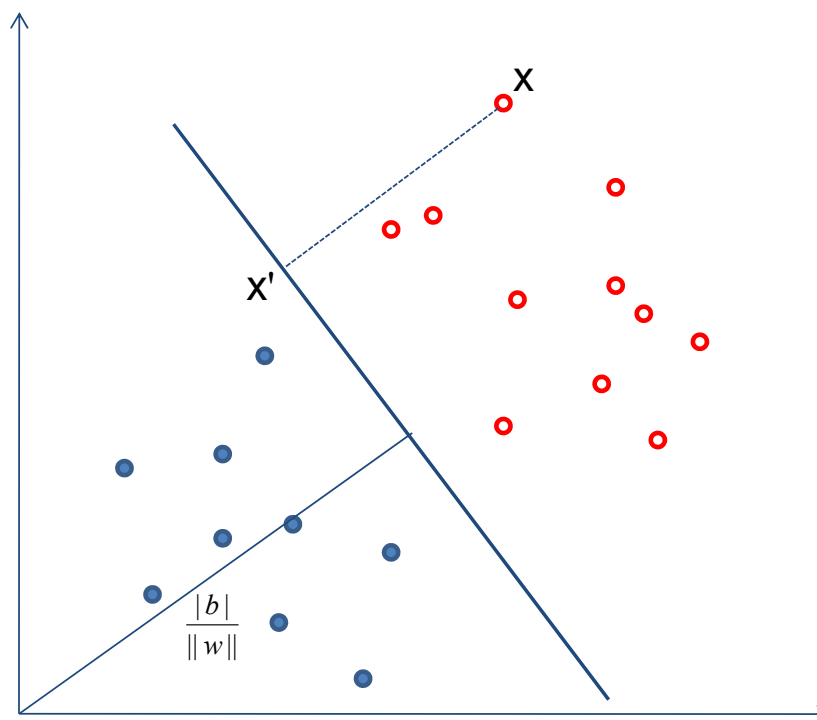
$$\hat{\gamma}_i = y_i(w \cdot x_i + b)$$

- 训练数据集的函数间隔

$$\hat{\gamma} = \min_{i=1, \dots, N} \hat{\gamma}_i$$

- 表示分类预测的正确性和确信度

- 当成比例改变 $w$ 和 $b$



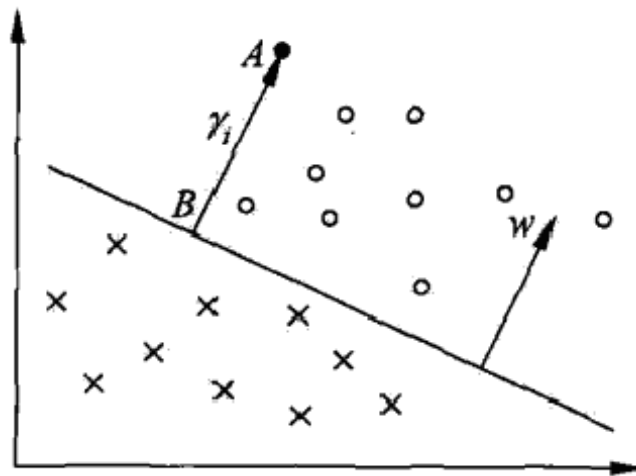
- 几何间隔

- 样本点的几何间隔：正例和负例

$$\gamma_i = + \left( \frac{w}{\|w\|} \cdot x_i + \frac{b}{\|w\|} \right) \quad \gamma_i = - \left( \frac{w}{\|w\|} \cdot x_i + \frac{b}{\|w\|} \right)$$



$$\gamma_i = y_i \left( \frac{w}{\|w\|} \cdot x_i + \frac{b}{\|w\|} \right)$$



- 几何间隔

- 对于给定的训练数据集 $T$ 和超平面 $(w, b)$ , 样本点的几何间隔:

$$\gamma_i = y_i \left( \frac{w}{\|w\|} \cdot x_i + \frac{b}{\|w\|} \right)$$

- 训练数据集的几何间隔:  $\gamma = \min_{i=1, \dots, N} \gamma_i$

- 函数间隔与几何间隔的关系:  $\gamma_i = \frac{\hat{\gamma}_i}{\|w\|}$

$$\gamma = \frac{\hat{\gamma}}{\|w\|}$$

间隔最大化

- 最大间隔分类超平面  $\max_{w,b} \gamma$

$$\text{s.t. } y_i \left( \frac{w}{\|w\|} \cdot x_i + \frac{b}{\|w\|} \right) \geq \gamma, \quad i=1,2,\dots,N$$

- 根据几何间隔和函数间隔的关系

$$\max_{w,b} \frac{\hat{\gamma}}{\|w\|}$$

$$\text{s.t. } y_i(w \cdot x_i + b) \geq \hat{\gamma}, \quad i=1,2,\dots,N$$

- 考虑

– 可以取  $\hat{\gamma}=1$

– 最大化  $\frac{1}{\|w\|}$  和最小化  $\frac{1}{2}\|w\|^2$  等价



- 线性可分支持向量机学习的最优化问题

$$\begin{aligned} \min_{w, b} \quad & \frac{1}{2} \|w\|^2 \\ \text{s.t.} \quad & y_i(w \cdot x_i + b) - 1 \geq 0, \quad i = 1, 2, \dots, N \end{aligned}$$

- 凸二次规划(convex quadratic programming)

## 线性可分支持向量机学习算法

- 输入：线性可分训练数据集  $T = \{(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)\}$   
 $x_i \in \mathcal{X} = \mathbf{R}^n$   $y_i \in \mathcal{Y} = \{-1, +1\}$ ,  $i = 1, 2, \dots, N$

- 输出：最大间隔分离超平面和分类决策函数
- 1、构造并求解约束最优化问题

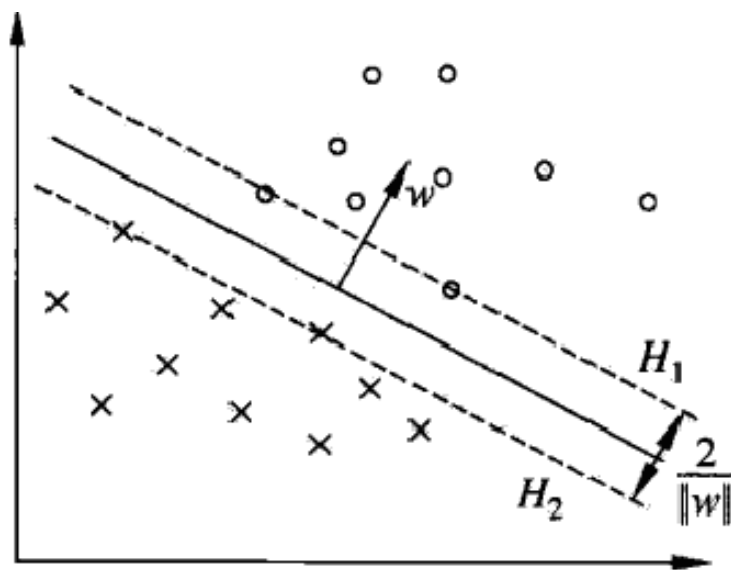
- $$\min_{w, b} \quad \frac{1}{2} \|w\|^2$$
$$\text{s.t.} \quad y_i(w \cdot x_i + b) - 1 \geq 0, \quad i = 1, 2, \dots, N$$

求得  $w^*$  和  $b^*$

- 2、得到分离超平面  $w^* \cdot x + b^* = 0$
- 分类决策函数  $f(x) = \text{sign}(w^* \cdot x + b^*)$

## 支持向量和间隔

- 在线性可分情况下，训练数据集的样本点中与分离超平面距离最近的样本点的实例称为支持向量(support vector);
- 支持向量是使约束条件式等号成立的点，即  $y_i(w \cdot x_i + b) - 1 = 0$
- 正例：  $H_1 : w \cdot x + b = 1$
- 负例：  $H_2 : w \cdot x + b = -1$





例题

$$\min_{w,b} \quad \frac{1}{2}(w_1^2 + w_2^2)$$

$$\text{s.t.} \quad 3w_1 + 3w_2 + b \geq 1$$

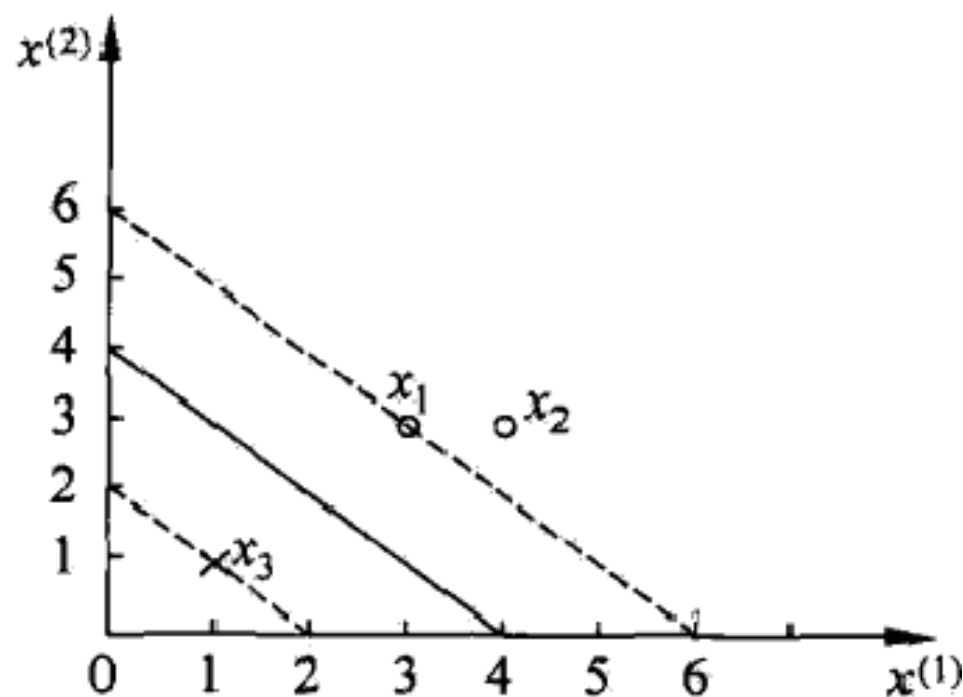
$$4w_1 + 3w_2 + b \geq 1$$

$$-w_1 - w_2 - b \geq 1$$

$$w_1 = w_2 = \frac{1}{2}, \quad b = -2$$

$$\frac{1}{2}x^{(1)} + \frac{1}{2}x^{(2)} - 2 = 0$$

$x_1 = (3, 3)^T$  与  $x_3 = (1, 1)^T$  为支持向量



## 学习的对偶算法

原始带约束最优化问题

$$\begin{aligned} \min_{w,b} \quad & \frac{1}{2} \|w\|^2 \\ \text{s.t.} \quad & y_i(w \cdot x_i + b) - 1 \geq 0, \quad i = 1, 2, \dots, N \end{aligned}$$

无约束的拉格朗日函数

$$\begin{aligned} \min_{w,b} \max_{\alpha} L(w, b, \alpha) &= \frac{1}{2} \|w\|^2 - \sum_{i=1}^N \alpha_i y_i (w \cdot x_i + b) + \sum_{i=1}^N \alpha_i \\ \text{s.t.} \quad & \alpha_i \geq 0 \end{aligned}$$

对偶问题

$$\begin{aligned} \max_{\alpha} \min_{w,b} L(w, b, \alpha) &= \frac{1}{2} \|w\|^2 - \sum_{i=1}^N \alpha_i y_i (w \cdot x_i + b) + \sum_{i=1}^N \alpha_i \\ \text{s.t.} \quad & \alpha_i \geq 0 \end{aligned}$$

对偶优化问题

$$\begin{aligned} \min_{\alpha} \quad & \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j y_i y_j (x_i \cdot x_j) - \sum_{i=1}^N \alpha_i \\ \text{s.t.} \quad & \sum_{i=1}^N \alpha_i y_i = 0 \\ & \alpha_i \geq 0, \quad i = 1, 2, \dots, N \end{aligned} \quad \Longrightarrow \quad \alpha^* \quad \Longrightarrow \quad \begin{aligned} w^* &= \sum_{i=1}^N \alpha_i^* y_i x_i \\ b^* &= y_j - \sum_{i=1}^N \alpha_i^* y_i (x_i \cdot x_j) \end{aligned}$$

原始带约束最优化问题→无约束的拉格朗日函数

$$\begin{aligned} \min_{\boldsymbol{x}} \quad & f(\boldsymbol{x}) \\ \text{s.t.} \quad & h(\boldsymbol{x}) = 0 \\ & g(\boldsymbol{x}) \leq 0 \end{aligned}$$

等式约束和不等式约束： $h(\boldsymbol{x}) = 0, g(\boldsymbol{x}) \leq 0$ 分别是由一个等式方程和一个不等式方程组成的方程组。

拉格朗日乘子： $\boldsymbol{\lambda} = (\lambda_1, \lambda_2, \dots, \lambda_m)$        $\boldsymbol{\mu} = (\mu_1, \mu_2, \dots, \mu_n)$

拉格朗日函数： $L(\boldsymbol{x}, \boldsymbol{\lambda}, \boldsymbol{\mu}) = f(\boldsymbol{x}) + \boldsymbol{\lambda}h(\boldsymbol{x}) + \boldsymbol{\mu}g(\boldsymbol{x})$

## 学习的对偶算法

原始带约束最优化问题  $\rightarrow$  无约束的拉格朗日函数

$$\begin{aligned} \min_{\mathbf{x}} \quad & f(\mathbf{x}) \\ \text{s.t.} \quad & h(\mathbf{x}) = 0 \\ & g(\mathbf{x}) \leq 0 \end{aligned}$$

$$\begin{aligned} L(\mathbf{x}, \boldsymbol{\lambda}, \boldsymbol{\mu}) &= f(\mathbf{x}) + \boldsymbol{\lambda}h(\mathbf{x}) + \boldsymbol{\mu}g(\mathbf{x}) \\ \lambda &\geq 0, \mu \geq 0 \end{aligned}$$

$$\begin{aligned} \min_{w, b} \quad & \frac{1}{2} \|w\|^2 \\ \text{s.t.} \quad & y_i(w \cdot x_i + b) - 1 \geq 0, \quad i = 1, 2, \dots, N \end{aligned}$$

$$\begin{aligned} \min_{w, b} \max_{\alpha} L(w, b, \alpha) &= \frac{1}{2} \|w\|^2 + \sum_{i=1}^N \alpha_i (1 - y_i(w x_i + b)) \\ \text{s.t.} \quad & \alpha_i \geq 0 \end{aligned}$$

## 学习的对偶算法

无约束的拉格朗日函数→对偶问题

$$\min_{w,b} \max_{\alpha} L(w,b,\alpha)$$

$$\text{s.t. } \alpha_i \geq 0$$

$$\max_{\alpha} \min_{w,b} L(w,b,\alpha)$$

$$\text{s.t. } \alpha_i \geq 0$$

原问题：极小极大，对偶问题：极大极小

弱对偶关系

$$\min \max L \geq \max \min L$$

强对偶关系

$$\min \max L = \max \min L$$

## 学习的对偶算法

$$\begin{aligned} (1) \text{ 求 } \min_{\mathbf{w}, b} L(\mathbf{w}, b, \boldsymbol{\alpha}) &= \frac{1}{2} \|\mathbf{w}\|^2 + \sum_{i=1}^m \alpha_i (1 - y_i (\mathbf{w}^T \mathbf{x}_i + b)) \\ &= \frac{1}{2} \|\mathbf{w}\|^2 + \sum_{i=1}^m (\alpha_i - \alpha_i y_i \mathbf{w}^T \mathbf{x}_i - \alpha_i y_i b) \\ &= \frac{1}{2} \mathbf{w}^T \mathbf{w} + \sum_{i=1}^m \alpha_i - \sum_{i=1}^m \alpha_i y_i \mathbf{w}^T \mathbf{x}_i - \sum_{i=1}^m \alpha_i y_i b \end{aligned}$$

对 $\mathbf{w}$ 和 $b$ 分别求偏导数并令其等于0:

$$\frac{\partial L}{\partial \mathbf{w}} = \frac{1}{2} \times 2 \times \mathbf{w} + 0 - \sum_{i=1}^m \alpha_i y_i \mathbf{x}_i - 0 = 0 \implies \mathbf{w} = \sum_{i=1}^m \alpha_i y_i \mathbf{x}_i$$

$$\frac{\partial L}{\partial b} = 0 + 0 - 0 - \sum_{i=1}^m \alpha_i y_i = 0 \implies \sum_{i=1}^m \alpha_i y_i = 0$$

## 学习的对偶算法

$$\max_{\alpha} \min_{w, b} L(w, b, \alpha)$$

$$\text{s.t. } \alpha_i \geq 0$$

对偶优化问题

$$\begin{aligned}
 (1) \text{ 求 } \min_{w, b} L(w, b, \alpha) &= \frac{1}{2} \|\mathbf{w}\|^2 + \sum_{i=1}^m \alpha_i (1 - y_i (\mathbf{w}^T \mathbf{x}_i + b)) \\
 &= \frac{1}{2} \|\mathbf{w}\|^2 + \sum_{i=1}^m (\alpha_i - \alpha_i y_i \mathbf{w}^T \mathbf{x}_i - \alpha_i y_i b) \\
 &= \frac{1}{2} \mathbf{w}^T \mathbf{w} + \sum_{i=1}^m \alpha_i - \sum_{i=1}^m \alpha_i y_i \mathbf{w}^T \mathbf{x}_i - \sum_{i=1}^m \alpha_i y_i b \\
 &= -\frac{1}{2} \mathbf{w}^T \sum_{i=1}^m \alpha_i y_i \mathbf{x}_i + \sum_{i=1}^m \alpha_i \\
 &= -\frac{1}{2} \left( \sum_{i=1}^m \alpha_i y_i \mathbf{x}_i \right)^T \left( \sum_{i=1}^m \alpha_i y_i \mathbf{x}_i \right) + \sum_{i=1}^m \alpha_i \quad \left[ \sum_{i=1}^m \alpha_i y_i \mathbf{x}_i \right] \\
 &= -\frac{1}{2} \sum_{i=1}^m \alpha_i y_i \mathbf{x}_i^T \sum_{i=1}^m \alpha_i y_i \mathbf{x}_i + \sum_{i=1}^m \alpha_i \\
 &= \sum_{i=1}^m \alpha_i - \frac{1}{2} \sum_{i=1}^m \sum_{j=1}^m \alpha_i \alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j
 \end{aligned}$$

学习的对偶算法

$$\min_{w,b} L(w, b, \alpha)$$

$$\begin{aligned} & \max_{\alpha} \min_{w,b} L(w, b, \alpha) \\ & \text{s.t. } \alpha_i \geq 0 \end{aligned}$$

(2) 求  $\min_{w,b} L(w, b, \alpha)$  对  $\alpha$  的极大, 即是对偶问题

$$\begin{aligned} & \max_{\alpha} \quad -\frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j y_i y_j (x_i \cdot x_j) + \sum_{i=1}^N \alpha_i \\ & \text{s.t.} \quad \sum_{i=1}^N \alpha_i y_i = 0 \end{aligned}$$

---

$$\begin{aligned} & \min_{\alpha} \quad \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j y_i y_j (x_i \cdot x_j) - \sum_{i=1}^N \alpha_i \\ & \text{s.t.} \quad \sum_{i=1}^N \alpha_i y_i = 0 \\ & \quad \alpha_i \geq 0, \quad i = 1, 2, \dots, N \end{aligned}$$



## 学习的对偶算法

原始带约束最优化问题

$$\min_{w,b} \frac{1}{2} \|w\|^2$$

$$\text{s.t. } y_i(w \cdot x_i + b) - 1 \geq 0, \quad i = 1, 2, \dots, N$$

无约束的拉格朗日函数

$$\min_{w,b} \max_{\alpha} L(w, b, \alpha) = \frac{1}{2} \|w\|^2 - \sum_{i=1}^N \alpha_i y_i (w \cdot x_i + b) + \sum_{i=1}^N \alpha_i$$

$$\text{s.t. } \alpha_i \geq 0$$

对偶问题

$$\max_{\alpha} \min_{w,b} L(w, b, \alpha) = \frac{1}{2} \|w\|^2 - \sum_{i=1}^N \alpha_i y_i (w \cdot x_i + b) + \sum_{i=1}^N \alpha_i$$

$$\text{s.t. } \alpha_i \geq 0$$

对偶优化问题

$$\min_{\alpha} \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j y_i y_j (x_i \cdot x_j) - \sum_{i=1}^N \alpha_i$$

$$\text{s.t. } \sum_{i=1}^N \alpha_i y_i = 0$$

$$\alpha_i \geq 0, \quad i = 1, 2, \dots, N$$

$$\Longrightarrow \alpha^* \Longrightarrow$$

$$w^* = \sum_{i=1}^N \alpha_i^* y_i x_i$$

$$b^* = y_j - \sum_{i=1}^N \alpha_i^* y_i (x_i \cdot x_j)$$

## KKT条件

$$\min_{\alpha} \quad \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j y_i y_j (x_i \cdot x_j) - \sum_{i=1}^N \alpha_i$$

$$\text{s.t.} \quad \sum_{i=1}^N \alpha_i y_i = 0$$

$$\alpha_i \geq 0, \quad i = 1, 2, \dots, N$$

$\alpha^* = (\alpha_1^*, \alpha_2^*, \dots, \alpha_l^*)^T$  是对偶最优化问题的解

KKT 条件成立

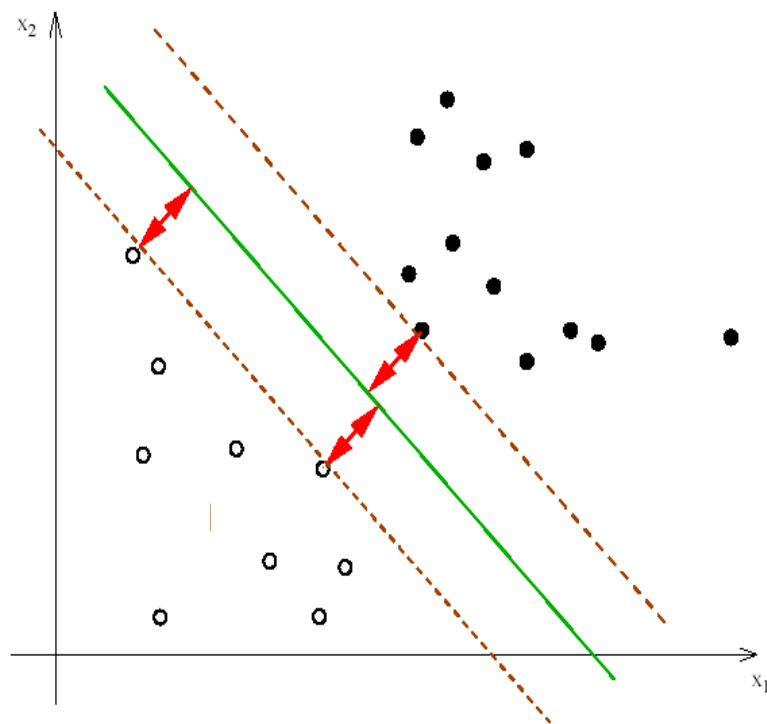
$$\nabla_w L(w^*, b^*, \alpha^*) = w^* - \sum_{i=1}^N \alpha_i^* y_i x_i = 0$$

$$\nabla_b L(w^*, b^*, \alpha^*) = - \sum_{i=1}^N \alpha_i^* y_i = 0$$

$$\alpha_i^* (y_i (w^* \cdot x_i + b^*) - 1) = 0, \quad i = 1, 2, \dots, N$$

$$y_i (w^* \cdot x_i + b^*) - 1 \geq 0, \quad i = 1, 2, \dots, N$$

$$\alpha_i^* \geq 0, \quad i = 1, 2, \dots, N$$



$$\text{得: } w^* = \sum_i \alpha_i^* y_i x_i \quad b^* = y_j - \sum_{i=1}^N \alpha_i^* y_i (x_i \cdot x_j)$$

$$\text{分离超平面: } \sum_{i=1}^N \alpha_i^* y_i (x \cdot x_i) + b^* = 0$$

$$\text{分类决策函数: } f(x) = \text{sign} \left( \sum_{i=1}^N \alpha_i^* y_i (x \cdot x_i) + b^* \right)$$

# 例题

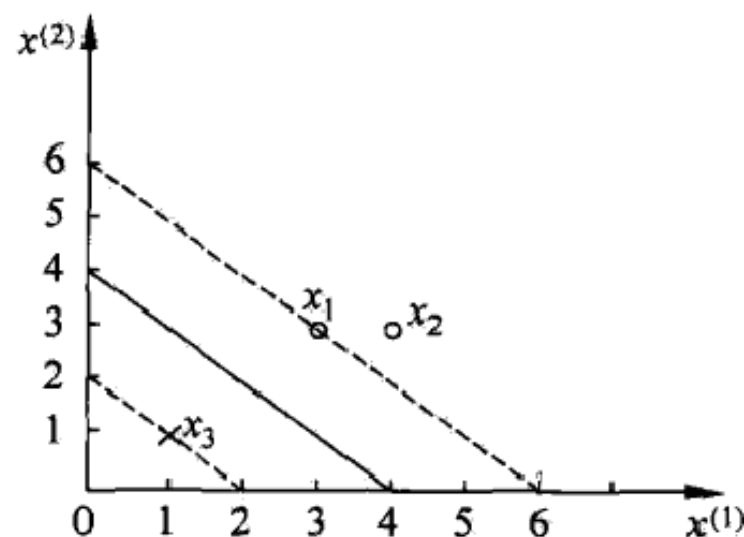
正例点  $x_1 = (3,3)^T$ ,  $x_2 = (4,3)^T$  负例点  $x_3 = (1,1)^T$

解：对偶形式

$$\begin{aligned} \min_{\alpha} \quad & \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j y_i y_j (x_i \cdot x_j) - \sum_{i=1}^N \alpha_i \\ &= \frac{1}{2} (18\alpha_1^2 + 25\alpha_2^2 + 2\alpha_3^2 + 42\alpha_1\alpha_2 - 12\alpha_1\alpha_3 - 14\alpha_2\alpha_3) - \alpha_1 - \alpha_2 - \alpha_3 \\ \text{s.t.} \quad & \alpha_1 + \alpha_2 - \alpha_3 = 0 \\ & \alpha_i \geq 0, \quad i=1,2,3 \end{aligned}$$

将  $\alpha_3 = \alpha_1 + \alpha_2$  带入目标函数并记为

$$s(\alpha_1, \alpha_2) = 4\alpha_1^2 + \frac{13}{2}\alpha_2^2 + 10\alpha_1\alpha_2 - 2\alpha_1 - 2\alpha_2$$



例题

- 对  $\alpha_1, \alpha_2$  求偏导数，并令其为0，易知  $s(\alpha_1, \alpha_2)$  在  $\left(\frac{3}{2}, -1\right)^T$
- 取极值，但该点不满足约束条件  $\alpha_2 \geq 0$ ，所以最小值应在边界上达到
- 当  $\alpha_1 = 0$  时，最小值  $s\left(0, \frac{2}{13}\right) = -\frac{2}{13}$
- 当  $\alpha_2 = 0$  时，最小值  $s\left(\frac{1}{4}, 0\right) = -\frac{1}{4}$
- 于是  $s(\alpha_1, \alpha_2)$  在  $\alpha_1 = \frac{1}{4}, \alpha_2 = 0$  获得极小， $\alpha_3 = \alpha_1 + \alpha_2 = \frac{1}{4}$
- 这样  $\alpha_1^* = \alpha_3^* = \frac{1}{4}$  对应的实例向量为支持向量

例题

计算得：

$$w_1^* = w_2^* = \frac{1}{2}$$

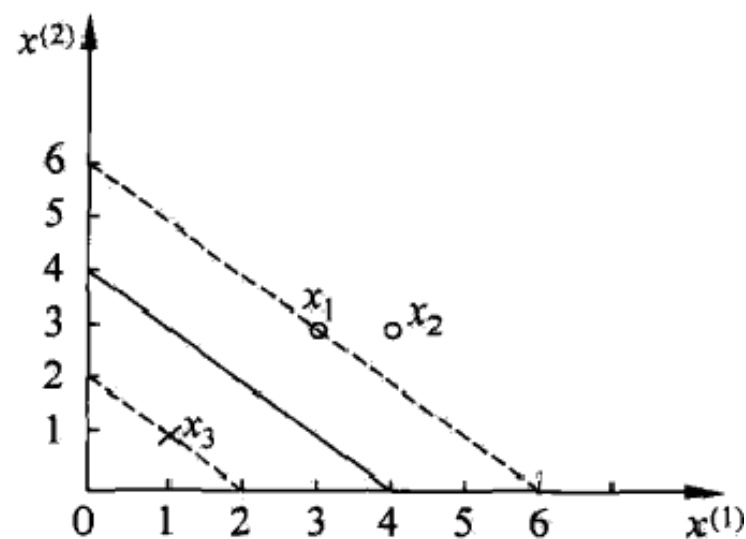
$$b^* = -2$$

分离超平面为：

$$\frac{1}{2}x^{(1)} + \frac{1}{2}x^{(2)} - 2 = 0$$

分类决策函数为：

$$f(x) = \text{sign}\left(\frac{1}{2}x^{(1)} + \frac{1}{2}x^{(2)} - 2\right)$$




# 2



## PART

线性支持向量机  
与软间隔最大化



## 线性支持向量机与软间隔最大化

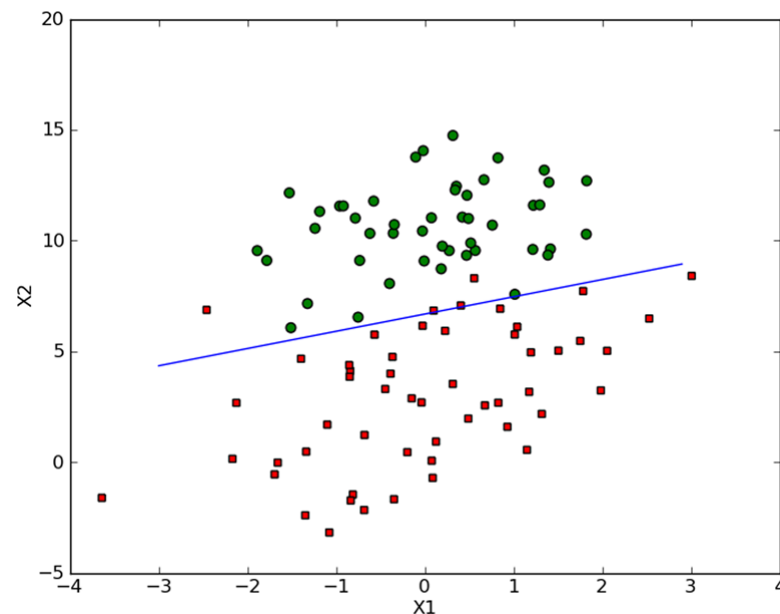
训练数据中有一些特异点 (outlier)，不能满足函数间隔大于等于1的约束条件。

解决方法：对每个样本点  $(x_i, y_i)$  引进一个松弛变量使得函数间隔加上松弛变量  $\xi_i \geq 0$  大于等于1，约束条件变为：

$$y_i(w \cdot x_i + b) \geq 1 - \xi_i$$

目标函数变为： $\frac{1}{2} \|w\|^2 + C \sum_{i=1}^N \xi_i$

$C > 0$  为惩罚参数



## 线性支持向量机与软间隔最大化

- 线性不可分的线性支持向量机的学习问题:

$$\begin{aligned} \min_{w, b, \xi} \quad & \frac{1}{2} \|w\|^2 + C \sum_{i=1}^N \xi_i \\ \text{s.t.} \quad & y_i(w \cdot x_i + b) \geq 1 - \xi_i, \quad i=1, 2, \dots, N \\ & \xi_i \geq 0, \quad i=1, 2, \dots, N \end{aligned}$$

- 设该问题的解是 $w^*, b^*$ , 可得到分离超平面和决策函数

$$\begin{aligned} w^* \cdot x + b^* &= 0 \\ f(x) &= \text{sign}(w^* \cdot x + b^*) \end{aligned}$$



## 线性支持向量机与软间隔最大化

- 原始问题的拉格朗日函数：

$$L(w, b, \xi, \alpha, \mu) = \frac{1}{2} \|w\|^2 + C \sum_{i=1}^N \xi_i - \sum_{i=1}^N \alpha_i (y_i (w \cdot x_i + b) - 1 + \xi_i) - \sum_{i=1}^N \mu_i \xi_i$$

- 其中： $\alpha_i \geq 0, \mu_i \geq 0$
- 对偶问题是拉格朗日函数的极大极小问题
- 首先求  $L(w, b, \xi, \alpha, \mu)$  对  $w, b, \xi$  的极小，由

$$\nabla_w L(w, b, \xi, \alpha, \mu) = w - \sum_{i=1}^N \alpha_i y_i x_i = 0$$

$$w = \sum_{i=1}^N \alpha_i y_i x_i$$

$$\nabla_b L(w, b, \xi, \alpha, \mu) = -\sum_{i=1}^N \alpha_i y_i = 0$$

$$\text{得：} \quad \sum_{i=1}^N \alpha_i y_i = 0$$

$$\nabla_{\xi_i} L(w, b, \xi, \alpha, \mu) = C - \alpha_i - \mu_i = 0$$

$$C - \alpha_i - \mu_i = 0$$

线性支持向量机与软间隔最大化

- 得:  $\min_{w,b,\xi} L(w,b,\xi,\alpha,\mu) = -\frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j y_i y_j (x_i \cdot x_j) + \sum_{i=1}^N \alpha_i$

- 再对  $\min_{w,b,\xi} L(w,b,\xi,\alpha,\mu)$  求 $\alpha$ 的极大, 得到对偶问题:


$$\max_{\alpha} \quad -\frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j y_i y_j (x_i \cdot x_j) + \sum_{i=1}^N \alpha_i$$

$$\text{s.t.} \quad \sum_{i=1}^N \alpha_i y_i = 0$$

$$C - \alpha_i - \mu_i = 0$$

$$\alpha_i \geq 0$$

$$\mu_i \geq 0, \quad i=1,2,\dots,N$$

  $0 \leq \alpha_i \leq C$

- 原始问题的对偶问题：

$$\begin{aligned} \min_{\alpha} \quad & \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j y_i y_j (x_i \cdot x_j) - \sum_{i=1}^N \alpha_i \\ \text{s.t.} \quad & \sum_{i=1}^N \alpha_i y_i = 0 \\ & 0 \leq \alpha_i \leq C, \quad i = 1, 2, \dots, N \end{aligned}$$

- 定理：设  $\alpha^* = (\alpha_1^*, \alpha_2^*, \dots, \alpha_N^*)^T$  是对偶问题的一个解，若存在  $\alpha^*$  的一个分量  $\alpha_j^*$ ， $0 < \alpha_j^* < C$ ，则原始问题的解  $w^*, b^*$  为：

$$w^* = \sum_{i=1}^N \alpha_i^* y_i x_i \qquad b^* = y_j - \sum_{i=1}^N y_i \alpha_i^* (x_i \cdot x_j)$$

## 线性支持向量机支持向量

在线性不可分的情况下，将对偶问题的解  $\alpha^* = (\alpha_1^*, \alpha_2^*, \dots, \alpha_N^*)^T$  中对应于  $\alpha_i^* > 0$  的样本点  $(x_i, y_i)$  的实例  $x_i$  称为支持向量（软间隔的支持向量）。

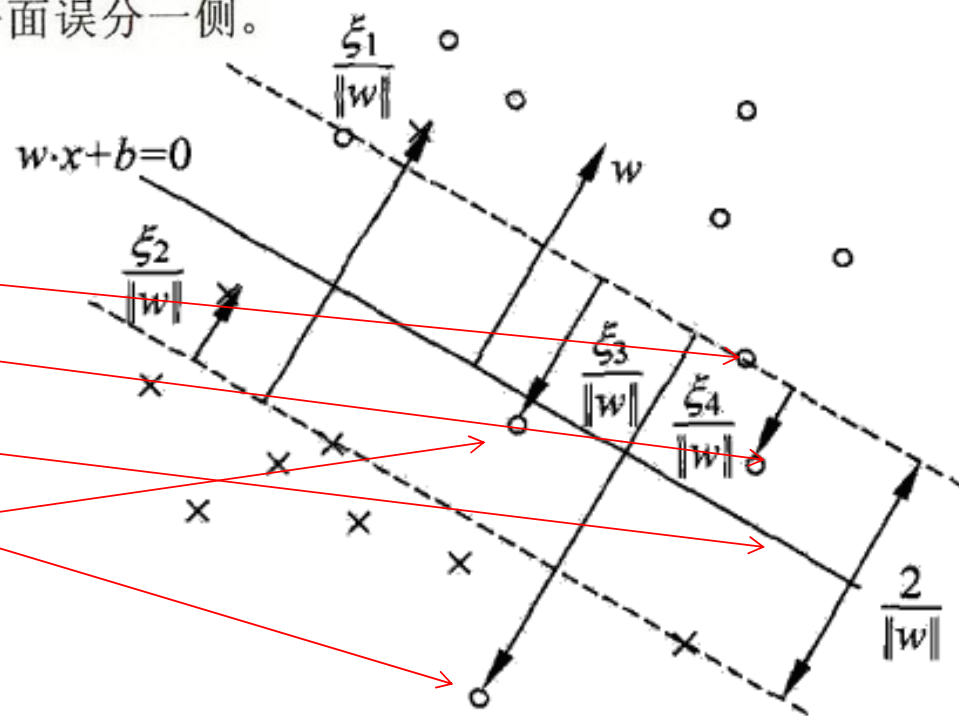
软间隔的支持向量  $x_i$  或者在间隔边界上，  
或者在间隔边界与分离超平面之间，或者在分离超平面误分一侧。

若  $\alpha_i^* < C$ ，则  $\xi_i = 0$

若  $\alpha_i^* = C$ ， $0 < \xi_i < 1$

若  $\alpha_i^* = C$ ， $\xi_i = 1$

若  $\alpha_i^* = C$ ， $\xi_i > 1$



线性损失函数

松弛变量  $\xi_i$ : 代价

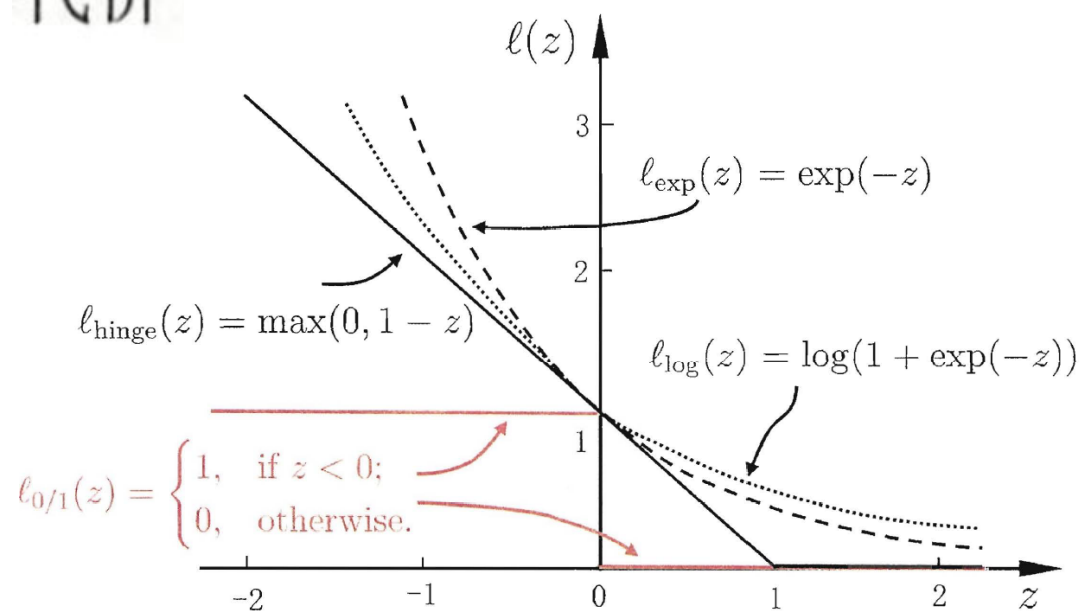


图 6.5 三种常见的替代损失函数: hinge损失、指数损失、对率损失

## 合页损失函数hinge loss function

- 线性支持向量机学习还有另外一种解释，就是最小化以下目标函数：

$$\sum_{i=1}^N [1 - y_i(w \cdot x_i + b)]_+ + \lambda \|w\|^2$$

- 第一项：  $L(y(w \cdot x + b)) = [1 - y(w \cdot x + b)]_+$
- 称为合页损失函数

$$[z]_+ = \begin{cases} z, & z > 0 \\ 0, & z \leq 0 \end{cases}$$

合页损失函数hinge loss function

- 线性支持向量机原始最优化问题:

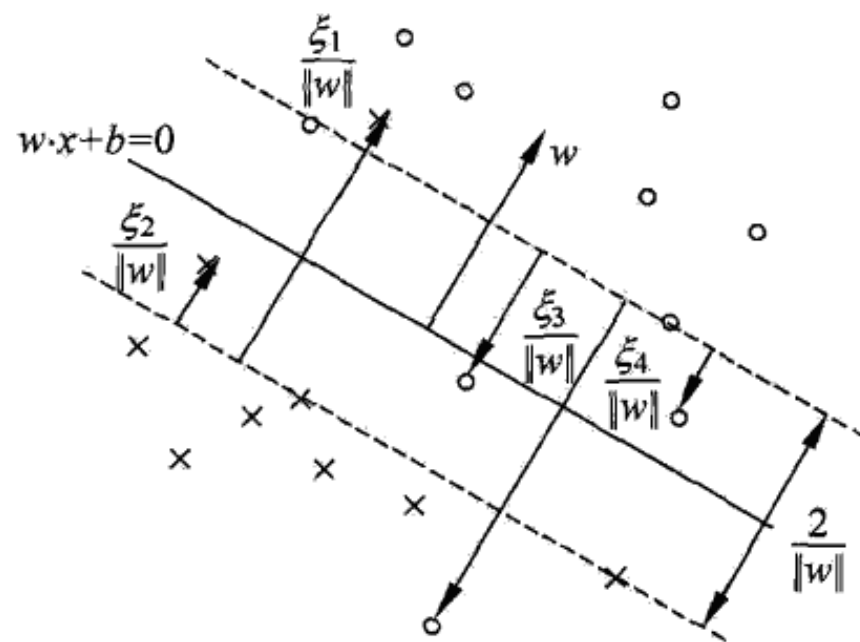
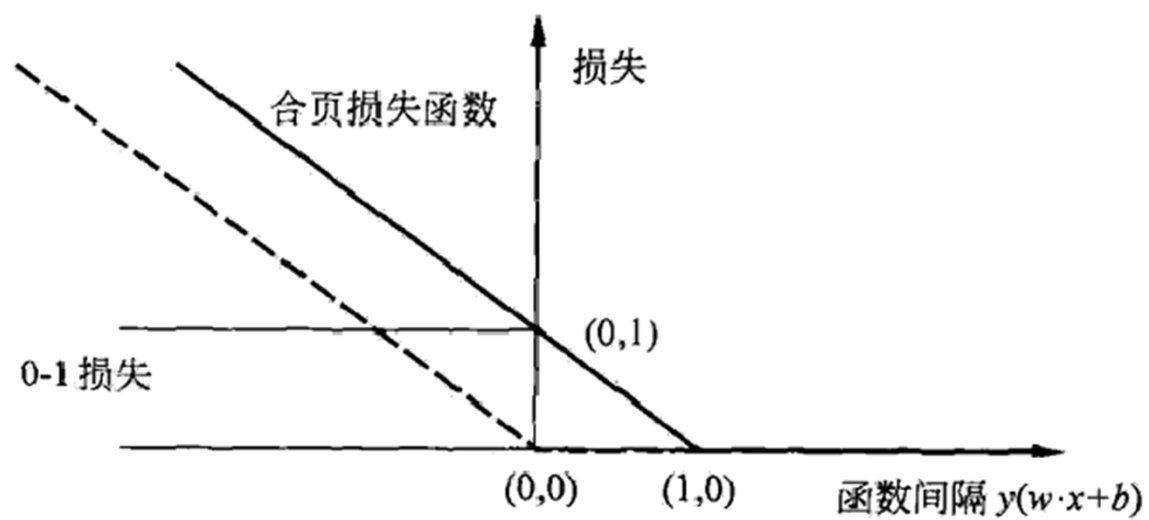
$$\min_{w, b, \xi} \quad \frac{1}{2} \|w\|^2 + C \sum_{i=1}^N \xi_i$$

$$\text{s.t.} \quad y_i(w \cdot x_i + b) \geq 1 - \xi_i, \quad i = 1, 2, \dots, N$$

$$\xi_i \geq 0, \quad i = 1, 2, \dots, N$$

- 等价于: 
$$\min_{w, b} \quad \sum_{i=1}^N [1 - y_i(w \cdot x_i + b)]_+ + \lambda \|w\|^2$$

## 合页损失函数hinge loss function





## 回顾：支持向量机分类

### 线性可分支持向量机

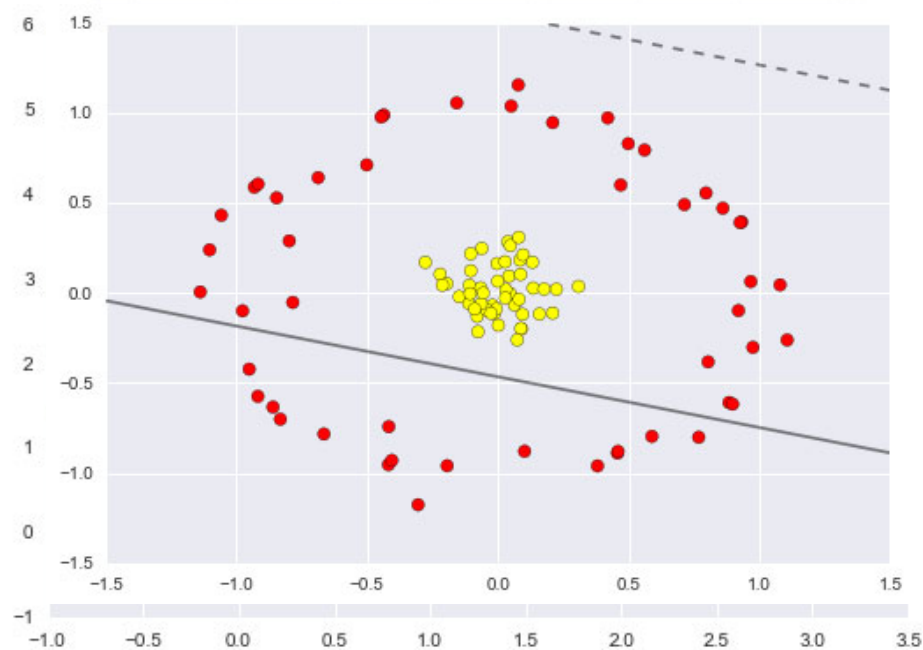
硬间隔最大化。


### 线性支持向量机

训练数据近似线性可分时，  
通过软间隔最大化

### 非线性支持向量机

当训练数据线性不可分时，  
通过使用核技巧(kernel  
trick)及软间隔最大化





回顾：支持向量机关键词

二类分类模型

学习策略：间隔最大化

凸二次规划问题

间隔，对偶，核技巧

上周存疑

<http://59.78.194.131:8889/notebooks/Support-Vector-Machines.ipynb>

回顾：学习的对偶算法

原始带约束最优化问题

$$\min_{w,b} \frac{1}{2} \|w\|^2$$

$$\text{s.t. } y_i(w \cdot x_i + b) - 1 \geq 0, \quad i = 1, 2, \dots, N$$

无约束的拉格朗日函数

$$\min_{w,b} \max_{\alpha} L(w, b, \alpha) = \frac{1}{2} \|w\|^2 - \sum_{i=1}^N \alpha_i y_i (w \cdot x_i + b) + \sum_{i=1}^N \alpha_i$$

$$\text{s.t. } \alpha_i \geq 0$$

对偶问题

$$\max_{\alpha} \min_{w,b} L(w, b, \alpha) = \frac{1}{2} \|w\|^2 - \sum_{i=1}^N \alpha_i y_i (w \cdot x_i + b) + \sum_{i=1}^N \alpha_i$$

$$\text{s.t. } \alpha_i \geq 0$$

对偶优化问题

$$\min_{\alpha} \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j y_i y_j (x_i \cdot x_j) - \sum_{i=1}^N \alpha_i$$

$$\text{s.t. } \sum_{i=1}^N \alpha_i y_i = 0$$

$$\alpha_i \geq 0, \quad i = 1, 2, \dots, N$$

$$\Longrightarrow \alpha^* \Longrightarrow$$

$$w^* = \sum_{i=1}^N \alpha_i^* y_i x_i$$

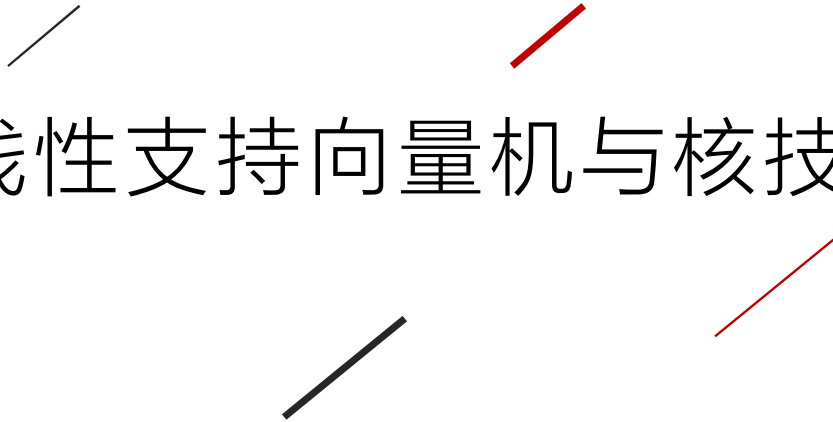
$$b^* = y_j - \sum_{i=1}^N \alpha_i^* y_i (x_i \cdot x_j)$$

# 3

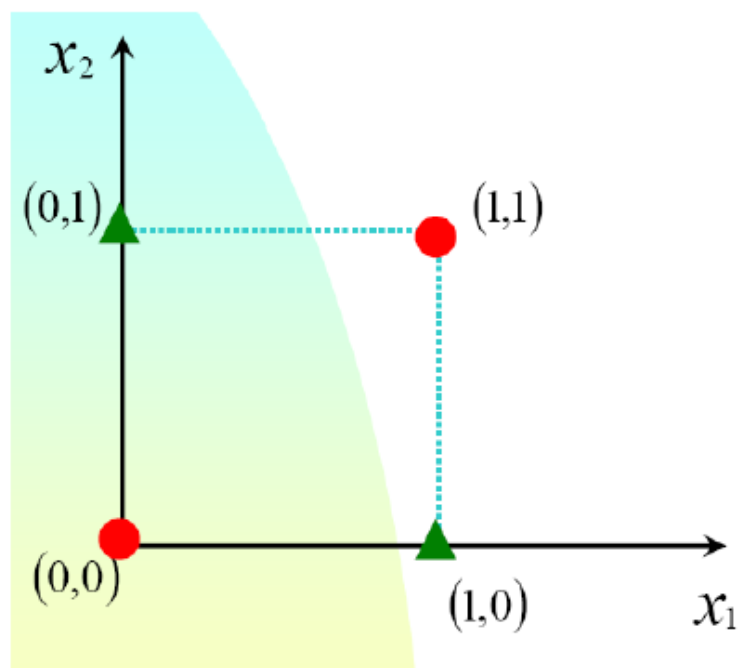


PART

## 非线性支持向量机与核技巧



## 例子



$$\omega_1 : \{(0,0), (1,1)\}$$

$$\omega_2 : \{(1,0), (0,1)\}$$

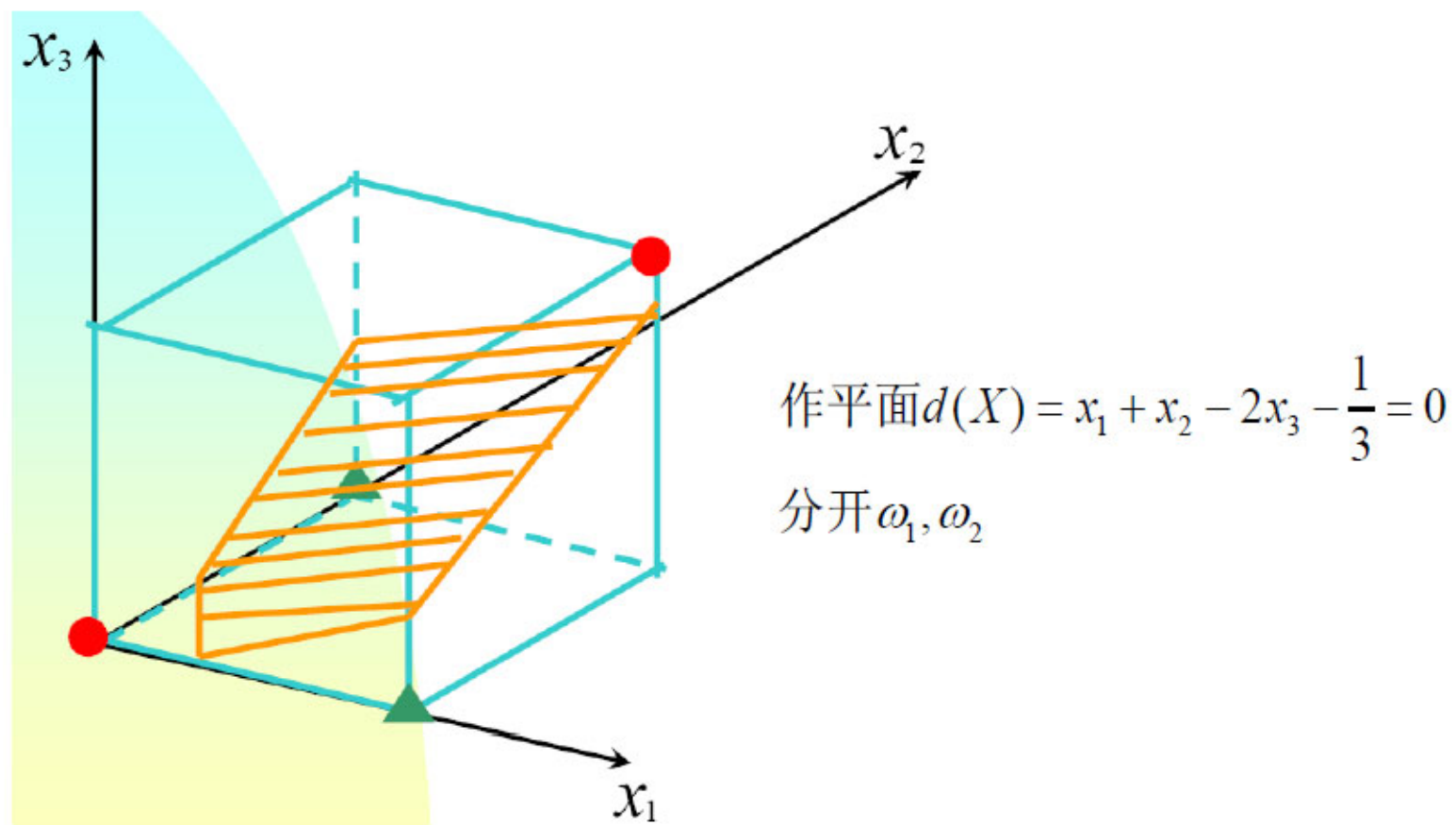
线性识别函数不存在  
把2维空间 $(x_1, x_2)$   
变为3维空间 $(x_1, x_2, x_3)$

$$x_3 = x_1 x_2$$

$(x_1, x_2)$	$(x_1, x_2, x_3)$
$(0,0)$	$(0,0,0)$
$(1,1)$	$(1,1,1)$
$(1,0)$	$(1,0,0)$
$(0,1)$	$(0,1,0)$

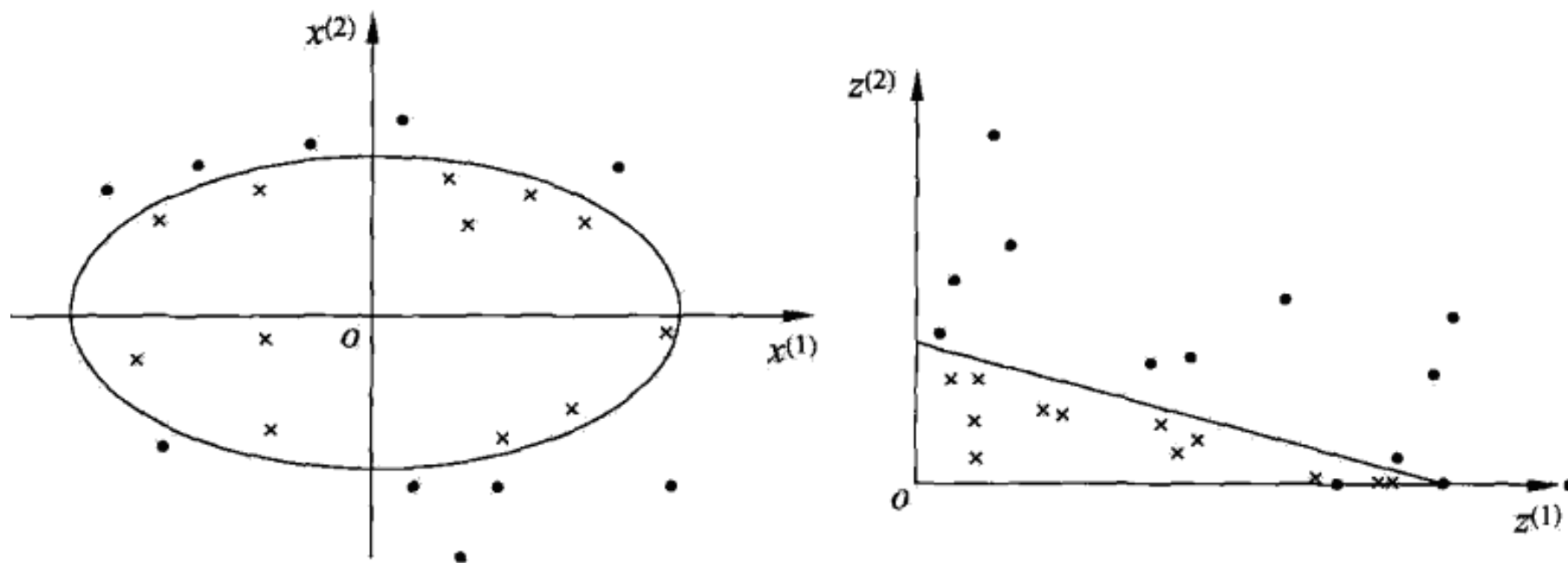
合页损失函数hinge loss function

例子



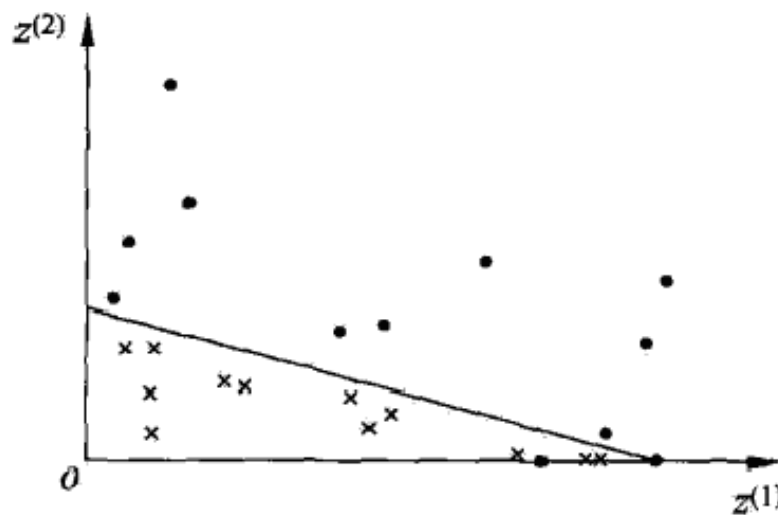
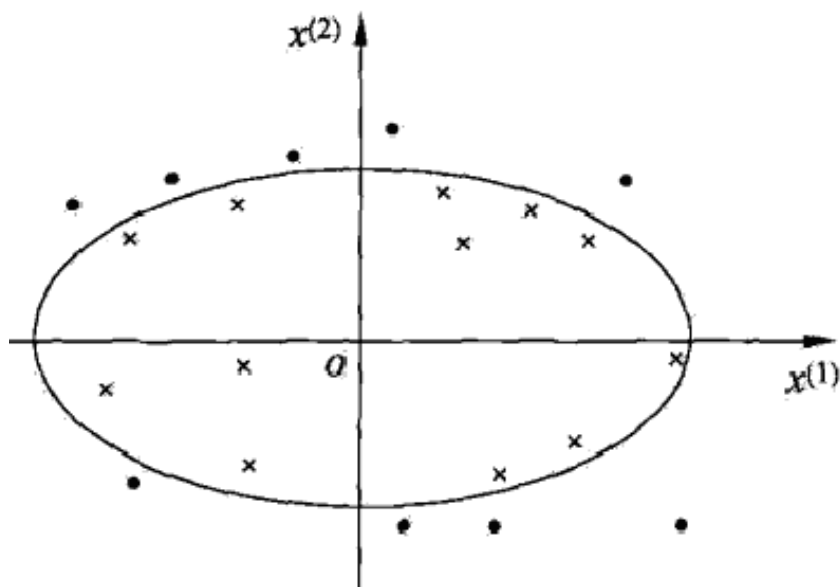
合页损失函数hinge loss function

例子




$$z = \phi(x) = ((x^{(1)})^2, (x^{(2)})^2)^T$$

- 非线性分类问题:
- 如果能用 $\mathbf{R}^n$ 中的一个超曲面将正负例正确分开, 则称这个问题为非线性可分问题.







[ 可视化演示

# *SVM with a polynomial Kernel visualization*

*Created by:  
Udi Aharoni*

## 非线性支持向量机与核函数

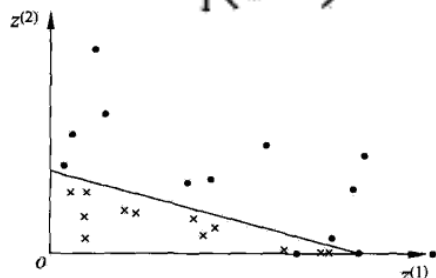
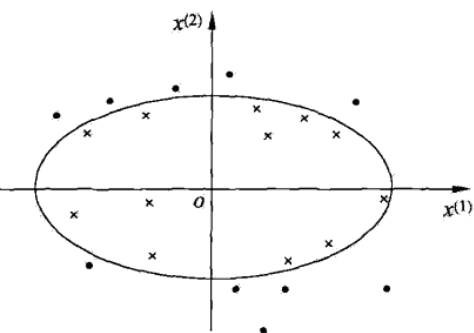
- 非线性问题往往不好求解，所以希望能用解线性分类问题的方法解决这个问题。
- 采取的方法是进行一个非线性变换，将非线性问题变换为线性问题，通过解变换后的线性问题的方法求解原来的非线性问题。
- 原空间： $\mathcal{X} \subset \mathbf{R}^2$ ,  $x = (x^{(1)}, x^{(2)})^T \in \mathcal{X}$

新空间：

$$\mathcal{Z} \subset \mathbf{R}^2, z = (z^{(1)}, z^{(2)})^T \in \mathcal{Z}$$
$$w_1(x^{(1)})^2 + w_2(x^{(2)})^2 + b = 0$$



$$z = \phi(x) = ((x^{(1)})^2, (x^{(2)})^2)^T$$
$$w_1 z^{(1)} + w_2 z^{(2)} + b = 0$$



- 用线性分类方法求解非线性分类问题分为两步：
  - 首先使用一个变换将原空间的数据映射到新空间;
  - 然后在新空间里用线性分类学习方法从训练数据中学习分类模型。
- 核技巧就属于这样的方法
  - 核技巧应用到支持向量机，其基本想法：
  - 通过一个非线性变换将输入空间(欧氏空间 $\mathbf{R}^n$ 或离散集合)对应于一个特征空间(希尔伯特空间)，使得在输入空间中的超曲面模型对应于特征空间中的超平面模型(支持向量机)。分类问题的学习任务通过在特征空间中求解线性支持向量机就可以完成。

## 非线性支持向量机与核函数

- 核函数定义：
- 设 $X$ 是输入空间(欧氏空间 $R^n$ 的子集或离散集合)，又设 $H$ 为特征空间(希尔伯特空间)，如果存在一个从 $X$ 到 $H$ 的映射
$$\phi(x): X \rightarrow H$$
- 使得对所有  $x, z \in X$
- 函数 $K(x, z)$ 满足条件  $K(x, z) = \phi(x) \cdot \phi(z)$
- 则称  $K(x, z)$  为核函数,  $\phi(x)$  为映射函数,
- 式中  $\phi(x) \cdot \phi(z)$  为  $\phi(x)$  和  $\phi(z)$  的内积

## 非线性支持向量机与核函数

- 例：
- 假设输入空间是 $\mathbf{R}^2$ ，核函数是  $K(x, z) = (x \cdot z)^2$ ，试找出其相关的特征空间 $\mathcal{H}$ 和映射  $\phi(x): \mathbf{R}^2 \rightarrow \mathcal{H}$
- 解：取特征空间  $\mathcal{H} = \mathbf{R}^3$ ，记  $x = (x^{(1)}, x^{(2)})^T$ ， $z = (z^{(1)}, z^{(2)})^T$

$$(x \cdot z)^2 = (x^{(1)}z^{(1)} + x^{(2)}z^{(2)})^2 = (x^{(1)}z^{(1)})^2 + 2x^{(1)}z^{(1)}x^{(2)}z^{(2)} + (x^{(2)}z^{(2)})^2$$

- 可以取：  $\phi(x) = ((x^{(1)})^2, \sqrt{2}x^{(1)}x^{(2)}, (x^{(2)})^2)^T$
- 容易验证：  $\phi(x) \cdot \phi(z) = (x \cdot z)^2 = K(x, z)$

## 非线性支持向量机与核函数

- 例：
- 假设输入空间是 $\mathbf{R}^2$ ，核函数是 $K(x, z) = (x \cdot z)^2$ ，试找出其相关的特征空间 $\mathcal{H}$ 和映射 $\phi(x) : \mathbf{R}^2 \rightarrow \mathcal{H}$
- 解：
- 同样：

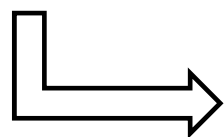
$$\text{仍取 } \mathcal{H} = \mathbf{R}^3 \quad \phi(x) = \frac{1}{\sqrt{2}} ((x^{(1)})^2 - (x^{(2)})^2, 2x^{(1)}x^{(2)}, (x^{(1)})^2 + (x^{(2)})^2)^T$$

$$\text{还可以取 } \mathcal{H} = \mathbf{R}^4 \quad \phi(x) = ((x^{(1)})^2, x^{(1)}x^{(2)}, x^{(1)}x^{(2)}, (x^{(2)})^2)^T$$

- 都满足条件

非线性支持向量机与核函数

$$\min_{\alpha} \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j y_i y_j (x_i \cdot x_j) - \sum_{i=1}^N \alpha_i$$



$$\min_{\alpha} \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j y_i y_j K(x_i, x_j) - \sum_{i=1}^N \alpha_i$$

- 核技巧的想法是：
- 在学习与预测中只定义核函数 $K(x, z)$ ，而不显式地定义映射函数，通常，直接计算 $K(x, z)$ 比较容易，而通过  $\phi(x)$  和  $\phi(z)$  计算 $K(x, z)$ 并不容易。
- 注意： $\phi$ 是输入空间 $\mathbf{R}^n$ 到特征空间 $\mathbf{H}$ 的映射，特征空间 $\mathbf{H}$ 一般是高维，映射可以不同。

## 核函数在支持向量机的应用

- 注意到：
- 线性支持向量机对偶问题中，无论是目标函数还是决策函数都只涉及输入实例和实例之间的内积。

- 目标函数中的内积  $\mathbf{x}_i \cdot \mathbf{x}_j$  用核函数  $K(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i) \cdot \phi(\mathbf{x}_j)$  代替，目标函数：

$$W(\alpha) = \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j) - \sum_{i=1}^N \alpha_i$$

- 决策函数：  $f(\mathbf{x}) = \text{sign} \left( \sum_{i=1}^{N_s} a_i^* y_i \phi(\mathbf{x}_i) \cdot \phi(\mathbf{x}) + b^* \right) = \text{sign} \left( \sum_{i=1}^{N_s} a_i^* y_i K(\mathbf{x}_i, \mathbf{x}) + b^* \right)$



## 正定核函数

**定义 7.6 (核函数)** 设  $\mathcal{X}$  是输入空间 (欧氏空间  $\mathbf{R}^n$  的子集或离散集合), 又设  $\mathcal{H}$  为特征空间 (希尔伯特空间), 如果存在一个从  $\mathcal{X}$  到  $\mathcal{H}$  的映射

$$\phi(x) : \mathcal{X} \rightarrow \mathcal{H} \quad (7.65)$$

使得对所有  $x, z \in \mathcal{X}$ , 函数  $K(x, z)$  满足条件

$$K(x, z) = \phi(x) \cdot \phi(z) \quad (7.66)$$

则称  $K(x, z)$  为核函数,  $\phi(x)$  为映射函数, 式中  $\phi(x) \cdot \phi(z)$  为  $\phi(x)$  和  $\phi(z)$  的内积。

**定义 7.7 (正定核的等价定义)** 设  $\mathcal{X} \subset \mathbf{R}^n$ ,  $K(x, z)$  是定义在  $\mathcal{X} \times \mathcal{X}$  上的对称函数, 如果对任意  $x_i \in \mathcal{X}$ ,  $i = 1, 2, \dots, m$ ,  $K(x, z)$  对应的 Gram 矩阵

$$K = [K(x_i, x_j)]_{m \times m} \quad (7.87)$$

是半正定矩阵, 则称  $K(x, z)$  是正定核。

正定核

- 假设 $K(x,z)$ 是定义在 $\mathcal{X} \times \mathcal{X}$ 上的对称函数，并且对任意的 $x_1, x_2, \dots, x_m \in \mathcal{X}$
- $K(x,z)$ 关于 $x_1, x_2, \dots, x_m$ 的Gram矩阵是半正定的，可以依据函数 $K(x,z)$ ，构成一个希尔伯特空间(Hilbert space);
- 其步骤是首先定义映射 $\phi$ ，并构成向量空间 $S$ ，然后在 $S$ 上定义内积构成内积空间; 最后将 $S$ 完备化构成希尔伯特空间.

构成希尔伯特空间

- 1、定义映射，构成向量空间S
- 映射： $\phi: x \rightarrow K(\cdot, x)$
- 对任意  $x_i \in \mathcal{X}$ ,  $\alpha_i \in \mathbf{R}$ ,  $i=1, 2, \dots, m$
- 定义线性组合： $f(\cdot) = \sum_{i=1}^m \alpha_i K(\cdot, x_i)$
- 考虑由线性组合为元素的集合S, 由于集合S对加法和数乘运算是封闭的，S构成一个向量空间。

构成希尔伯特空间

- 2、在S上定义内积，构成内积空间
- 在S上定义一个运算“\*”: 对任意f, g属于S

$$f(\cdot) = \sum_{i=1}^m \alpha_i K(\cdot, x_i) \quad g(\cdot) = \sum_{j=1}^l \beta_j K(\cdot, z_j)$$

- 定义运算\*: 
$$f * g = \sum_{i=1}^m \sum_{j=1}^l \alpha_i \beta_j K(x_i, z_j)$$
- 证明内积空间:
  - (1)  $(cf) * g = c(f * g)$ ,  $c \in \mathbf{R}$
  - (2)  $(f + g) * h = f * h + g * h$ ,  $h \in S$
  - (3)  $f * g = g * f$
  - (4)  $f * f \geq 0$ ,  $f * f = 0 \Leftrightarrow f = 0$

构成希尔伯特空间

- 3、将内积空间 $S$ 完备化为希尔伯特空间
- 由：
$$f \cdot g = \sum_{i=1}^m \sum_{j=1}^l \alpha_i \beta_j K(x_i, z_j)$$
- 内积得到范数： $\|f\| = \sqrt{f \cdot f}$
- 因此， $S$ 是一个赋范向量空间；根据泛函分析理论，对于不完备的赋范向量空间 $S$ ，一定可以使之完备化，得到完备的赋范向量空间 $H$ ；一个内积空间，当作为一个赋范向量空间是完备的时候，就是希尔伯特空间，这样，就得到了希尔伯特空间 $H$ 。
- 再生性： $K(\cdot, x) \cdot f = f(x) \quad K(\cdot, x) \cdot K(\cdot, z) = K(x, z)$

正定核的充要条件

- 设  $K: \mathcal{X} \times \mathcal{X} \rightarrow \mathbf{R}$  是对称函数, 则  $K(x, z)$  为正定核函数的充要条件是对任意  $x_i \in \mathcal{X}, i=1, 2, \dots, m$ ,  $K(x, z)$  对应的 Gram 矩阵  $K = [K(x_i, x_j)]_{m \times m}$  是半正定的。

$$K(x, z) = \phi(x) \cdot \phi(z) \iff \begin{cases} K(x, z) \text{ 是对称函数} \\ [K(x_i, x_j)]_{m \times n} \text{ 是半正定矩阵} \end{cases}$$

正定核的必要性证明

$$K(x, z) = \phi(x) \cdot \phi(z) \Rightarrow \begin{cases} K(x, z) \text{ 是对称函数} \\ [K(x_i, x_j)]_{m \times n} \text{ 是半正定矩阵} \end{cases}$$

$$K(x, z) = \phi(x) \cdot \phi(z)$$

$$K(z, x) = \phi(z) \cdot \phi(x)$$

$$\phi(x) \cdot \phi(z) = \phi(z) \cdot \phi(x)$$

$$K(x, z) = K(z, x)$$

$K(x, z)$  是对称函数

设  $A$  是实对称矩阵。如果对任意的实非零列向量  $x$  有  $x^T A x \geq 0$ , 就称  $A$  为半正定矩阵。

$$(c_1, c_2, \dots, c_m) K(x_i, x_j) (c_1, c_2, \dots, c_m)^T$$

$$= \sum_{i,j=1}^m c_i c_j K(x_i, x_j) = \sum_{i,j=1}^m c_i c_j (\phi(x_i) \cdot \phi(x_j))$$

$$= \left( \sum_i c_i \phi(x_i) \right) \cdot \left( \sum_j c_j \phi(x_j) \right)$$

$$= \left\| \sum_i c_i \phi(x_i) \right\|^2 \geq 0$$

正定核的充分性证明

$$K(x, z) = \phi(x) \cdot \phi(z) \leftarrow \begin{cases} K(x, z) \text{ 是对称函数} \\ [K(x_i, x_j)]_{m \times n} \text{ 是半正定矩阵} \end{cases}$$

充分性。已知对称函数  $K(x, z)$  对任意  $x_1, x_2, \dots, x_m \in \mathcal{X}$ ,  $K(x, z)$  关于  $x_1, x_2, \dots, x_m$  的 Gram 矩阵是半正定的。根据前面的结果, 对给定的  $K(x, z)$ , 可以构造从  $\mathcal{X}$  到某个希尔伯特空间  $\mathcal{H}$  的映射:

$$\phi: x \rightarrow K(\cdot, x) \quad (7.86)$$

由式 (7.83) 可知,

$$K(\cdot, x) \cdot f = f(x)$$

并且

$$K(\cdot, x) \cdot K(\cdot, z) = K(x, z)$$

由式 (7.86) 即得

$$K(x, z) = \phi(x) \cdot \phi(z)$$

表明  $K(x, z)$  是  $\mathcal{X} \times \mathcal{X}$  上的核函数。



## 常用核函数

### 1、多项式核函数 (Polynomial kernel function)

$$K(x, z) = (x \cdot z + 1)^p$$

对应的支持向量机为P次多项式分类器，分类决策函数：

$$f(x) = \text{sign} \left( \sum_{i=1}^{N_s} a_i^* y_i (x_i \cdot x + 1)^p + b^* \right)$$

### 2、高斯核函数 (Gaussian Kernel Function)

$$K(x, z) = \exp \left( -\frac{\|x - z\|^2}{2\sigma^2} \right)$$

决策函数：

$$f(x) = \text{sign} \left( \sum_{i=1}^{N_s} a_i^* y_i \exp \left( -\frac{\|x - z\|^2}{2\sigma^2} \right) + b^* \right)$$

### 3、字符串核函数

$$k_n(s, t) = \sum_{u \in \Sigma^n} [\phi_n(s)]_u [\phi_n(t)]_u = \sum_{u \in \Sigma^n} \sum_{(i,j): s(i)=t(j)=u} \lambda^{l(i)} \lambda^{l(j)}$$

## 非线性支持向量机学习算法

- 输入：线性不可分训练数据集  $T = \{(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)\}$   
 $x_i \in \mathcal{X} = \mathbf{R}^n$       $y_i \in \mathcal{Y} = \{-1, +1\}$ ,  $i = 1, 2, \dots, N$
- 输出：分类决策函数
- 1、选取适当的核函数和参数C，构造最优化问题：

$$\begin{aligned} \min_{\alpha} \quad & \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j y_i y_j K(x_i, x_j) - \sum_{i=1}^N \alpha_i \\ \text{s.t.} \quad & \sum_{i=1}^N \alpha_i y_i = 0 \\ & 0 \leq \alpha_i \leq C, \quad i = 1, 2, \dots, N \end{aligned}$$

求得最优解： $\alpha^* = (\alpha_1^*, \alpha_2^*, \dots, \alpha_N^*)^T$

- 2、选择  $\alpha^*$  的一个正分量  $0 < \alpha_j^* < C$ , 计算

$$b^* = y_j - \sum_{i=1}^N \alpha_i^* y_i K(x_i \cdot x_j)$$

- 3、构造决策函数

$$f(x) = \text{sign} \left( \sum_{i=1}^N \alpha_i^* y_i K(x \cdot x_i) + b^* \right)$$

当 $K(x,z)$ 是正定核函数时, 构造的最优化问题是凸二次规划问题, 解是存在的。

# 4



PART

## 序列最小最优化算法





## 序列最小最优化算法

- 序列最小最优化(sequential minimal optimization SMO)算法：  
1998年由Platt提出。
- 动机：
- 支持向量机的学习问题可以形式化为求解凸二次规划问题. 这样的凸二次规划问题具有全局最优解，并且有许多最优化算法可以用于这一问题的求解；
- 但是当训练样本容量很大时，这些算法往往变得非常低效，以致无法使用. 所以，如何高效地实现支持向量机学习就成为一个重要的问题。

- SMO (Sequential minimal optimization)
- 解如下凸二次规划的对偶问题

$$\min_{\alpha} \quad \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j y_i y_j K(x_i, x_j) - \sum_{i=1}^N \alpha_i$$

$$\text{s.t.} \quad \sum_{i=1}^N \alpha_i y_i = 0$$

$$0 \leq \alpha_i \leq C, \quad i=1, 2, \dots, N$$

- 注意：变量是拉格朗日乘子 $\alpha_i$ ，一个对应一个样本

## SMO算法

$$\min_{\alpha} \quad \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j y_i y_j K(x_i, x_j) - \sum_{i=1}^N \alpha_i$$

$$\text{s.t.} \quad \sum_{i=1}^N \alpha_i y_i = 0$$

$$0 \leq \alpha_i \leq C, \quad i=1,2,\dots,N$$

- 启发式算法，基本思路：
- 如果所有变量的解都满足此最优化问题的KKT条件，那么得到解；
- 否则，选择两个变量，固定其它变量，针对这两个变量构建一个二次规划问题，称为子问题，可通过解析方法求解，提高了计算速度。
- 子问题的两个变量：一个是违反KKT条件最严重的那个，另一个由约束条件自动确定。

$$\alpha_1 = -y_1 \sum_{i=2}^N \alpha_i y_i$$

- SMO算法包括两个部分：
  - 求解两个变量二次规划的解析方法
  - 选择变量的启发式方法

两个变量二次规划的求解过程

$$\min_{\alpha} \quad \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j y_i y_j K(x_i, x_j) - \sum_{i=1}^N \alpha_i$$

$$\text{s.t.} \quad \sum_{i=1}^N \alpha_i y_i = 0$$

$$0 \leq \alpha_i \leq C, \quad i=1, 2, \dots, N$$

- 选择两个变量，其它固定，SMO的最优化问题的子问题：

$$\begin{aligned} \min_{\alpha_1, \alpha_2} \quad W(\alpha_1, \alpha_2) = & \frac{1}{2} K_{11} \alpha_1^2 + \frac{1}{2} K_{22} \alpha_2^2 + y_1 y_2 K_{12} \alpha_1 \alpha_2 \\ & - (\alpha_1 + \alpha_2) + y_1 \alpha_1 \sum_{i=3}^N y_i \alpha_i K_{i1} + y_2 \alpha_2 \sum_{i=3}^N y_i \alpha_i K_{i2} \end{aligned}$$

$$\text{s.t.} \quad \alpha_1 y_1 + \alpha_2 y_2 = - \sum_{i=3}^N y_i \alpha_i = \zeta$$

$$0 \leq \alpha_i \leq C, \quad i=1, 2$$



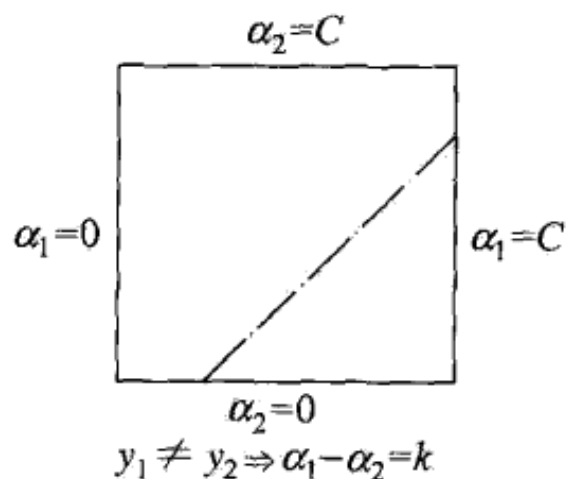
两个变量二次规划的求解过程

$$\min_{\alpha} \quad \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j y_i y_j K(x_i, x_j) - \sum_{i=1}^N \alpha_i$$

$$\text{s.t.} \quad \sum_{i=1}^N \alpha_i y_i = 0$$

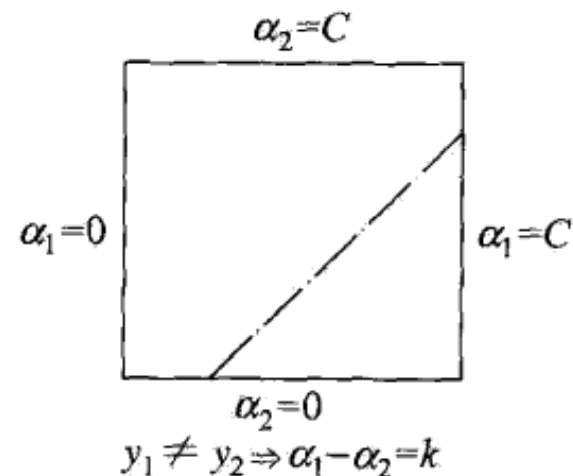
$$0 \leq \alpha_i \leq C, \quad i=1, 2, \dots, N$$

- 两个变量，约束条件用二维空间中的图形表示



- 假设二次规划问题的初始可行解为  $\alpha_1^{\text{old}}, \alpha_2^{\text{old}}$ ，最优解为  $\alpha_1^{\text{new}}, \alpha_2^{\text{new}}$
- 设  $\alpha_2$  未经剪辑时的最优解为  $\alpha_2^{\text{new,unc}}$

两个变量二次规划的求解过程



- 根据不等式条件  $\alpha_2^{\text{new}}$  的取值范围:

$$L \leq \alpha_2^{\text{new}} \leq H$$

- 左图:  $L = \max(0, \alpha_2^{\text{old}} - \alpha_1^{\text{old}})$   $H = \min(C, C + \alpha_2^{\text{old}} - \alpha_1^{\text{old}})$
- 右图:  $L = \max(0, \alpha_2^{\text{old}} + \alpha_1^{\text{old}} - C)$   $H = \min(C, \alpha_2^{\text{old}} + \alpha_1^{\text{old}})$

两个变量二次规划的求解过程

$$\begin{aligned} \min_{\alpha} \quad & \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j y_i y_j K(x_i, x_j) - \sum_{i=1}^N \alpha_i \\ \text{s.t.} \quad & \sum_{i=1}^N \alpha_i y_i = 0 \\ & 0 \leq \alpha_i \leq C, \quad i=1, 2, \dots, N \end{aligned}$$

- 求解过程：
- 先求沿着约束方向未经剪辑时的  $\alpha_2^{\text{new,unc}}$
- 再求剪辑后的  $\alpha_2^{\text{new}}$
- 记：  $g(x) = \sum_{i=1}^N \alpha_i y_i K(x_i, x) + b$
- 令：  $E_i = g(x_i) - y_i = \left( \sum_{j=1}^N \alpha_j y_j K(x_j, x_i) + b \right) - y_i, \quad i=1, 2$
- $E$  为输入  $x$  的预测值和真实输出  $y$  的差,  $i=1, 2$

## 两个变量二次规划的求解过程

- 定理:
- 最优化问题 沿约束方向未经剪辑的解:

$$\alpha_2^{\text{new,unc}} = \alpha_2^{\text{old}} + \frac{y_2(E_1 - E_2)}{\eta}$$
$$\eta = K_{11} + K_{22} - 2K_{12} = \|\Phi(x_1) - \Phi(x_2)\|^2$$

- 剪辑后的解

$$\alpha_2^{\text{new}} = \begin{cases} H, & \alpha_2^{\text{new,unc}} > H \\ \alpha_2^{\text{new,unc}}, & L \leq \alpha_2^{\text{new,unc}} \leq H \\ L, & \alpha_2^{\text{new,unc}} < L \end{cases}$$

- 得到 $\alpha_1$ 的解  $\alpha_1^{\text{new}} = \alpha_1^{\text{old}} + y_1 y_2 (\alpha_2^{\text{old}} - \alpha_2^{\text{new}})$

两个变量二次规划的求解过程

- 证明： 引进记号

$$v_i = \sum_{j=3}^N \alpha_j y_j K(x_i, x_j) = g(x_i) - \sum_{j=1}^2 \alpha_j y_j K(x_i, x_j) - b, \quad i=1,2$$

- 目标函数写成：

$$W(\alpha_1, \alpha_2) = \frac{1}{2} K_{11} \alpha_1^2 + \frac{1}{2} K_{22} \alpha_2^2 + y_1 y_2 K_{12} \alpha_1 \alpha_2 \\ - (\alpha_1 + \alpha_2) + y_1 v_1 \alpha_1 + y_2 v_2 \alpha_2$$

- 由  $\alpha_1 y_1 = \zeta - \alpha_2 y_2$  及  $y_i^2 = 1$

$$\alpha_1 = (\zeta - y_2 \alpha_2) y_1$$


两个变量二次规划的求解过程

- 得到只是 $\alpha_2$  的函数的目标函数

$$W(\alpha_2) = \frac{1}{2}K_{11}(\zeta - \alpha_2 y_2)^2 + \frac{1}{2}K_{22}\alpha_2^2 + y_2 K_{12}(\zeta - \alpha_2 y_2)\alpha_2 \\ - (\zeta - \alpha_2 y_2)y_1 - \alpha_2 + v_1(\zeta - \alpha_2 y_2) + y_2 v_2 \alpha_2$$

- 对 $\alpha_2$ 求导  $\frac{\partial W}{\partial \alpha_2} = K_{11}\alpha_2 + K_{22}\alpha_2 - 2K_{12}\alpha_2 - K_{11}\zeta y_2 + K_{12}\zeta y_2 + y_1 y_2 - 1 - v_1 y_2 + y_2 v_2$
- 令其为0:

$$(K_{11} + K_{22} - 2K_{12})\alpha_2 = y_2(y_2 - y_1 + \zeta K_{11} - \zeta K_{12} + v_1 - v_2) \\ = y_2 \left[ y_2 - y_1 + \zeta K_{11} - \zeta K_{12} + \left( g(x_1) - \sum_{j=1}^2 y_j \alpha_j K_{1j} - b \right) - \left( g(x_2) - \sum_{j=1}^2 y_j \alpha_j K_{2j} - b \right) \right]$$

两个变量二次规划的求解过程

- 将  $\zeta = \alpha_1^{\text{old}} y_1 + \alpha_2^{\text{old}} y_2$  代入:

$$\begin{aligned}(K_{11} + K_{22} - 2K_{12})\alpha_2^{\text{new,unc}} &= y_2 ((K_{11} + K_{22} - 2K_{12})\alpha_2^{\text{old}} y_2 + y_2 - y_1 + g(x_1) - g(x_2)) \\ &= (K_{11} + K_{22} - 2K_{12})\alpha_2^{\text{old}} + y_2(E_1 - E_2)\end{aligned}$$

- 将  $\eta = K_{11} + K_{22} - 2K_{12}$  代入:

$$\alpha_2^{\text{new,unc}} = \alpha_2^{\text{old}} + \frac{y_2(E_1 - E_2)}{\eta}$$

## 两个变量二次规划的求解过程

- 得到定理:
- 最优化问题沿约束方向未经剪辑的解:

$$\alpha_2^{\text{new,unc}} = \alpha_2^{\text{old}} + \frac{y_2(E_1 - E_2)}{\eta}$$
$$\eta = K_{11} + K_{22} - 2K_{12} = \|\Phi(x_1) - \Phi(x_2)\|^2$$

- 剪辑后的解

$$\alpha_2^{\text{new}} = \begin{cases} H, & \alpha_2^{\text{new,unc}} > H \\ \alpha_2^{\text{new,unc}}, & L \leq \alpha_2^{\text{new,unc}} \leq H \\ L, & \alpha_2^{\text{new,unc}} < L \end{cases}$$

- 得到 $\alpha_1$ 的解  $\alpha_1^{\text{new}} = \alpha_1^{\text{old}} + y_1 y_2 (\alpha_2^{\text{old}} - \alpha_2^{\text{new}})$



- SMO算法在每个子问题中选择两个变量优化，其中至少一个变量是违反KKT条件的

### 1、第一个变量的选择：外循环

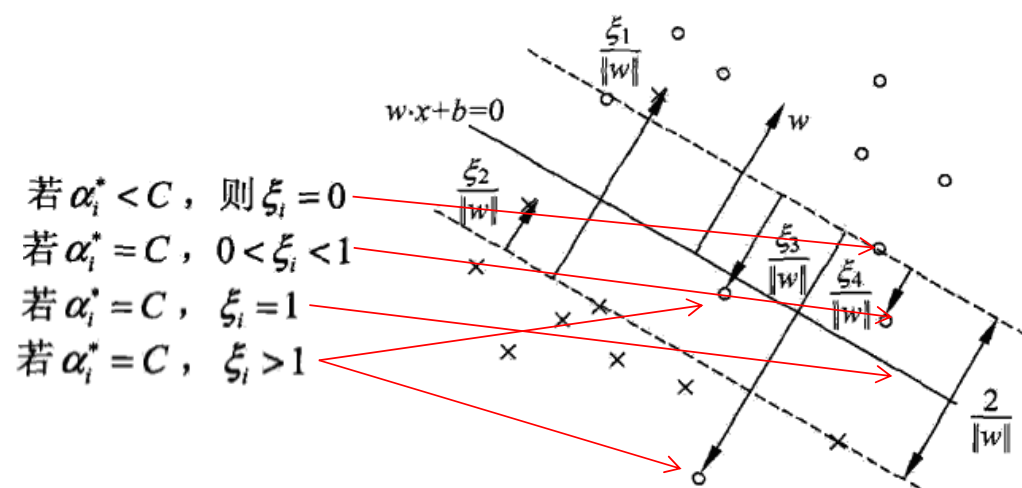
违反KKT最严重的样本点， 检验样本点是否满足KKT条件：

$$\alpha_i = 0 \Leftrightarrow y_i g(x_i) \geq 1$$

$$0 < \alpha_i < C \Leftrightarrow y_i g(x_i) = 1 \quad \leftarrow \text{先检查}$$

$$\alpha_i = C \Leftrightarrow y_i g(x_i) \leq 1$$

$$g(x_i) = \sum_{j=1}^N \alpha_j y_j K(x_i, x_j) + b$$



## 2、第二个变量的检查：内循环，

- 选择的标准是希望能使目标函数有足够大的变化
- 即对应  $|E_1 - E_2|$  最大，即  $E_1$ ,  $E_2$  的符号相反，差异最大
- 如果内循环通过上述方法找到的点不能使目标函数有足够的下降
- 则：遍历间隔边界上的样本点，测试目标函数下降
- 如果下降不大，则遍历所有样本点
- 如果依然下降不大，则丢弃外循环点，重新选择

计算阈值**b**和**E<sub>i</sub>**

3、每次完成两个变量的优化后，重新计算**b**，**E<sub>i</sub>**

• 由KKT条件：

$$\sum_{i=1}^N \alpha_i y_i K_{i1} + b = y_1$$

$$b_1^{\text{new}} = y_1 - \sum_{i=3}^N \alpha_i y_i K_{i1} - \alpha_1^{\text{new}} y_1 K_{11} - \alpha_2^{\text{new}} y_2 K_{21}$$

$$E_i = g(x_i) - y_i = \left( \sum_{j=1}^N \alpha_j y_j K(x_j, x_i) + b \right) - y_i, \quad i=1,2$$

$$E_1 = \sum_{i=3}^N \alpha_i y_i K_{i1} + \alpha_1^{\text{old}} y_1 K_{11} + \alpha_2^{\text{old}} y_2 K_{21} + b^{\text{old}} - y_1$$

计算阈值**b**和**E<sub>i</sub>**

3、每次完成两个变量的优化后，重新计算**b**， **E<sub>i</sub>**

• 由KKT条件：

$$\sum_{i=1}^N \alpha_i y_i K_{i1} + b = y_1$$

$$b_1^{\text{new}} = y_1 - \sum_{i=3}^N \alpha_i y_i K_{i1} - \alpha_1^{\text{new}} y_1 K_{11} - \alpha_2^{\text{new}} y_2 K_{21}$$

$$E_1 = \sum_{i=3}^N \alpha_i y_i K_{i1} + \alpha_1^{\text{old}} y_1 K_{11} + \alpha_2^{\text{old}} y_2 K_{21} + b^{\text{old}} - y_1$$

$$y_1 - \sum_{i=3}^N \alpha_i y_i K_{i1} = -E_1 + \alpha_1^{\text{old}} y_1 K_{11} + \alpha_2^{\text{old}} y_2 K_{21} + b^{\text{old}}$$

$$b_1^{\text{new}} = -E_1 - y_1 K_{11} (\alpha_1^{\text{new}} - \alpha_1^{\text{old}}) - y_2 K_{21} (\alpha_2^{\text{new}} - \alpha_2^{\text{old}}) + b^{\text{old}}$$

计算阈值**b**和**E<sub>i</sub>**

- 如果:  $0 < \alpha_2^{\text{new}} < C$

$$b_2^{\text{new}} = -E_2 - y_1 K_{12}(\alpha_1^{\text{new}} - \alpha_1^{\text{old}}) - y_2 K_{22}(\alpha_2^{\text{new}} - \alpha_2^{\text{old}}) + b^{\text{old}}$$

$$E_i^{\text{new}} = \sum_S y_j \alpha_j K(x_i, x_j) + b^{\text{new}} - y_i$$

$S$  是所有支持向量  $x_j$  的集合

## SMO算法

- 输入：训练数据集  $T = \{(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)\}$ ，精度  $\varepsilon$   
 $x_i \in \mathcal{X} = \mathbf{R}^n$   $y_i \in \mathcal{Y} = \{-1, +1\}$ ， $i = 1, 2, \dots, N$

- 输出：近似解  $\alpha$

(1) 取初值  $\alpha^{(0)} = 0$ ，令  $k = 0$

(2) 选取优化变量  $\alpha_i^{(k)}, \alpha_j^{(k)}$ ，解析求解两个变量的最优化问题  
求得最优解  $\alpha_i^{(k+1)}, \alpha_j^{(k+1)}$ ，更新  $\alpha$  为  $\alpha^{(k+1)}$ ；

(3) 若在精度  $\varepsilon$  范围内满足停机条件

$$\sum_{i=1}^N \alpha_i y_i = 0 \quad 0 \leq \alpha_i \leq C, \quad i = 1, 2, \dots, N$$

$$y_i \cdot g(x_i) = \begin{cases} \geq 1, & \{x_i \mid \alpha_i = 0\} \\ = 1, & \{x_i \mid 0 < \alpha_i < C\} \\ \leq 1, & \{x_i \mid \alpha_i = C\} \end{cases} \quad g(x_i) = \sum_{j=1}^N \alpha_j y_j K(x_j, x_i) + b$$

则转 (4)；否则令  $k = k + 1$ ，转 (2)；

(4) 取  $\hat{\alpha} = \alpha^{(k+1)}$

在精度 $\varepsilon$ 内检查KKT条件的方法

放宽精度的要求，加快收敛

$$\alpha_i = 0 \Leftrightarrow y_i g(x_i) \geq 1$$

$$0 < \alpha_i < C \Leftrightarrow y_i g(x_i) = 1$$

$$\alpha_i = C \Leftrightarrow y_i g(x_i) \leq 1$$

$$\alpha_i = 0 \Rightarrow$$


$$0 < \alpha_i < C \Rightarrow$$

$$\alpha_i = C \Rightarrow$$

$$y_i d_i \geq 1 - \varepsilon$$

$$1 - \varepsilon \leq y_i d_i \leq 1 + \varepsilon$$

$$y_i d_i \leq 1 + \varepsilon$$



## 代码&资料推荐

代码演示

[./Support-Vector-Machines.ipynb](#)

机器学习-白板推导系列(六)-支持向量机SVM (Support Vector Machine)

<https://www.bilibili.com/video/av28186618>

机器学习-白板推导系列(七)-核方法 (Kernel Method)

<https://www.bilibili.com/video/av34731384>

南瓜书PumpkinBook

<https://datawhalechina.github.io/pumpkin-book/#/>

支持向量机原理详解(六): 序列最小最优化(SMO)算法(Part I & II)

<https://zhuanlan.zhihu.com/p/64580199>

<https://zhuanlan.zhihu.com/p/62367247>



