第4.2节 SVM

内 容

- **SVM概述**
- 结构风险最小化
- 线性SVM
- SVM求解
- 处理线性不可分问题
- SVM训练算法

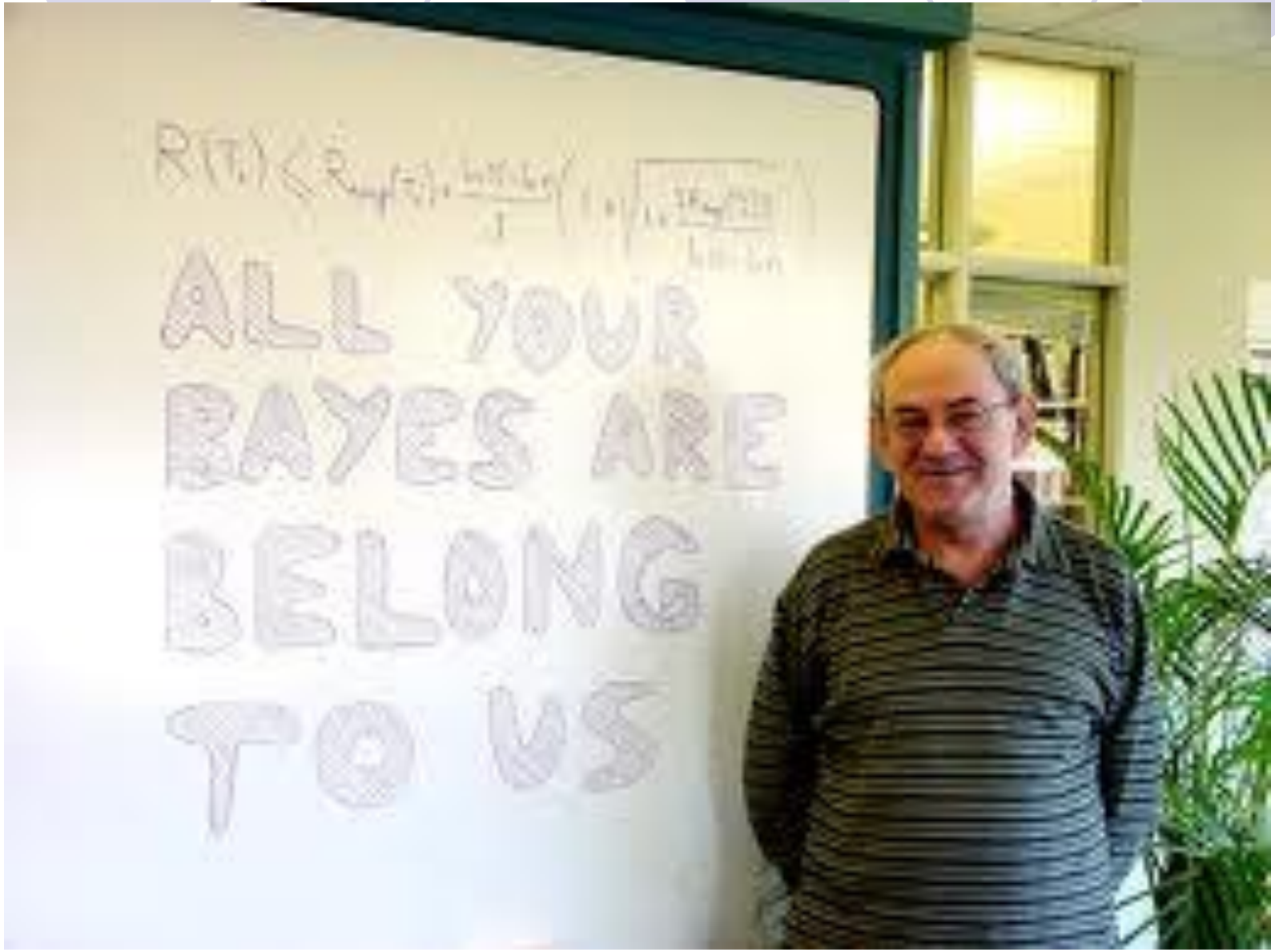
概述：支持向量机发展历史

- 1963年，Vapnik在解决模式识别问题时提出了**支持向量**方法。起决定性作用的样本为**支持向量**
- 1971年，Kimeldorf构造基于**支持向量**构建核空间的方法
- 1992年，Vapnik等人开始对**支持向量机**进行研究。
- 1995年，Vapnik等人正式提出统计学习理论。

概述

- 通俗来讲，它是一种二类分类模型，其基本模型定义为特征空间上的间隔最大的线性分类器，即支持向量机的学习策略便是间隔最大化，最终可转化为一个凸二次规划问题的求解。
- 上个世纪90年代，支持向量机获得全面发展，在实际应用中，获得比较满意的效果，成为**机器学习领域的标准工具**

概述：支持向量机发展历史



常用的机器学习方法比较

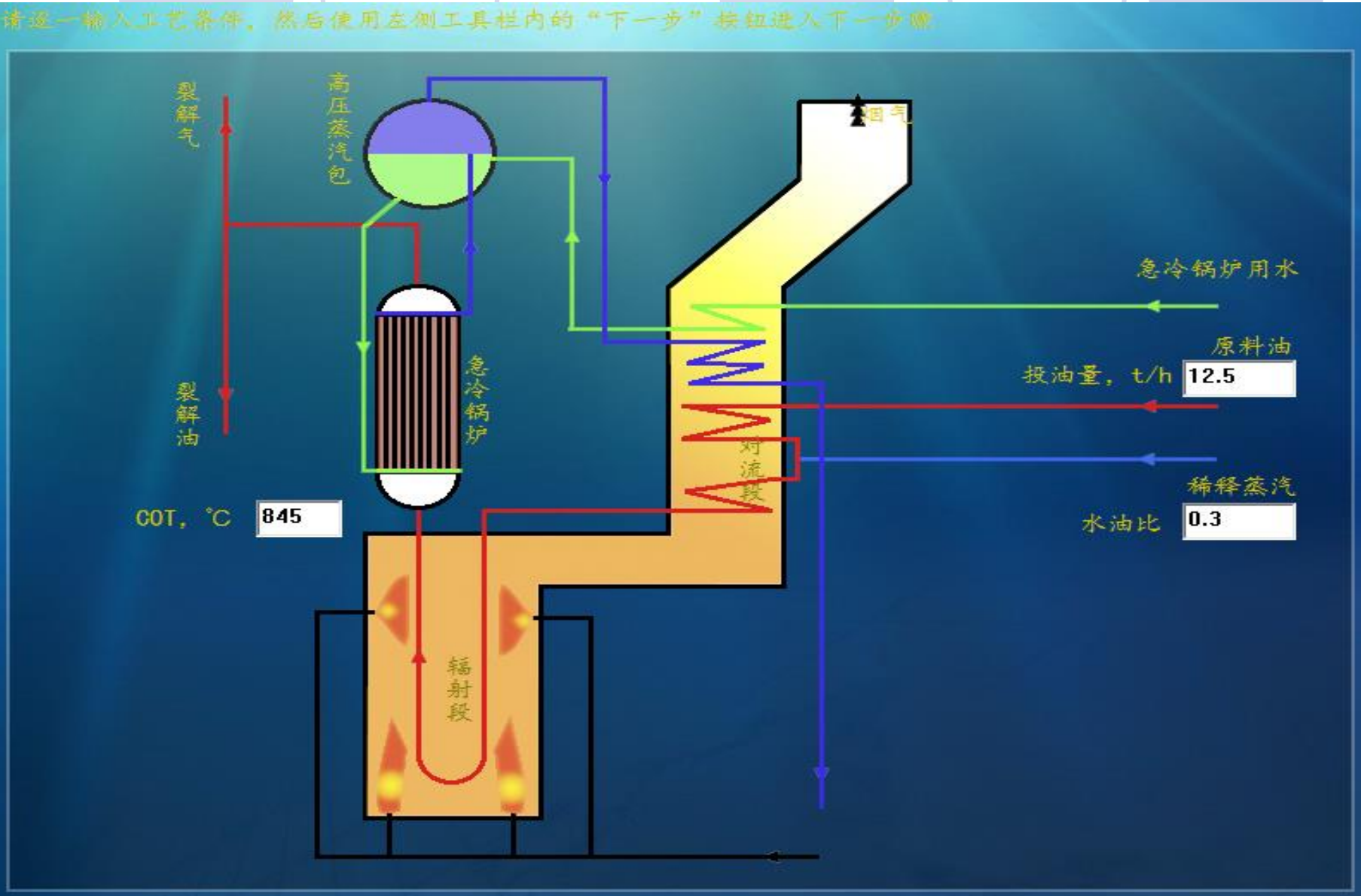
- 概率分布的方法（经典的方法）
  - Bayes方法, GMMs 用于复杂分布建模
- 决策树的方法（C4.5）
  - 属性具有明确含义时使用，一种经典的方法
- 近邻分类
  - 简单的方法，如果样本有代表性，维数不高时好用
- 支撑向量机
  - 高维空间的小样本学习、结构风险最小化
- Boosting算法
  - 大量训练样本下可以取得好的效果，速度很快
- 人工神经网络ANN
  - 非线性方法，大量训练样本下可以取得好的效果，速度较慢

SVM案例：手写体数字识别例子

- 贝尔实验室对美国邮政手写数字库进行的实验
- 该数据共包含7291个训练样本，2007个测试数据，输入数据的维数为16x16维

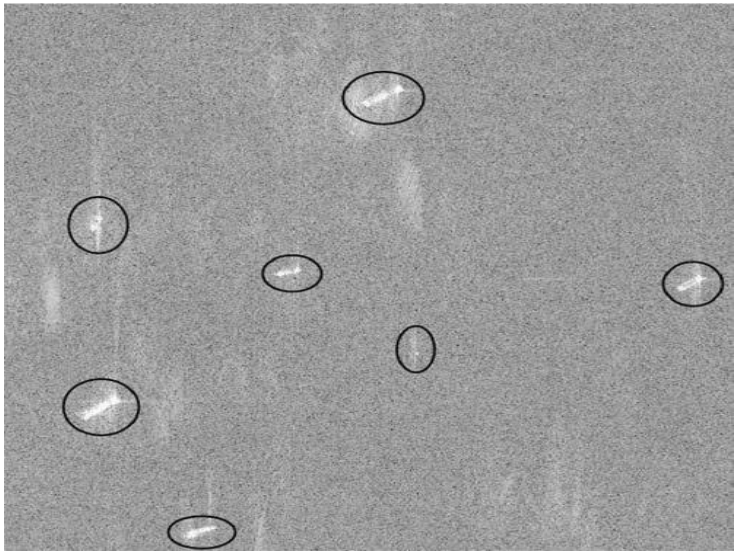
分类器/学习方法	错误率
人工表现	2.5%
决策树C4.5	16.2%
三层神经网络	5.9%
SVM	4.0%
DeepLearning	<1.0%

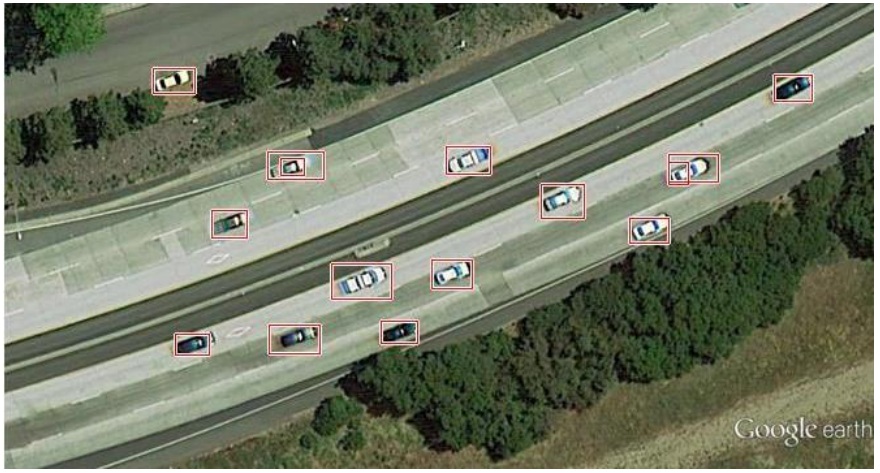
SVM案例：石脑油预测


请逐一输入工艺条件，然后使用左侧工具栏内的“下一步”按钮进入下一步骤
裂解气
高压蒸汽包
烟气
急冷锅炉用水
原料油
投油量，t/h
12.5
裂解油
急冷锅炉
对流段
稀释蒸汽
水油比
0.3
COT，℃
845
辐射段

SVM案例：目标检测

- 高清航拍图像目标检测识别



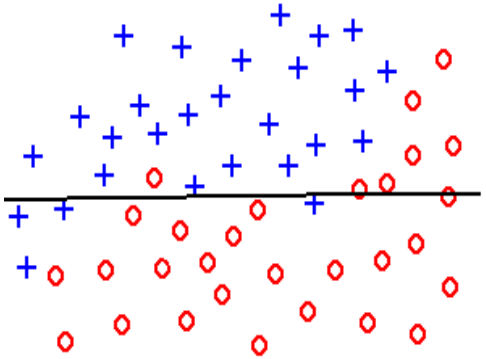


内 容

- SVM概述
- 结构风险最小化
- 线性SVM
- SVM求解
- 处理线性不可分问题
- SVM训练算法

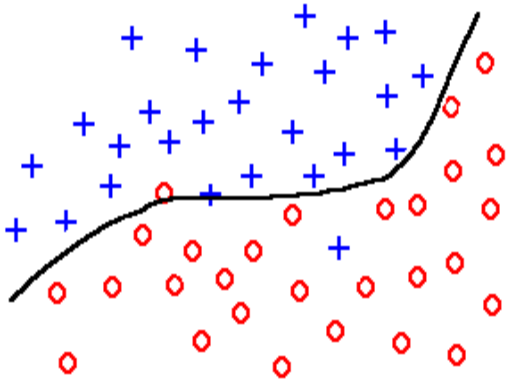
VC维与经验风险

分类问题图示：　过拟合与欠拟合



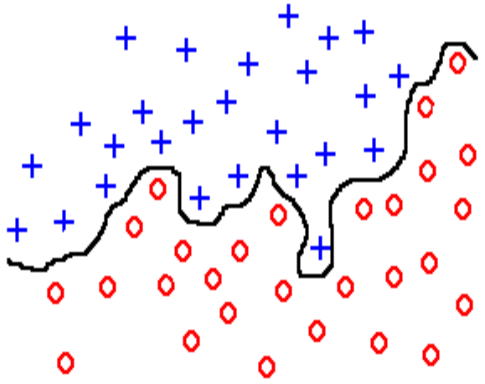
underfitting

欠拟合



good fit

较好的拟合



Overfitting

过拟合

问题: 经验风险小，并不意味着期望风险R小.

结构风险最小化

- 实际风险（测试误差）: Risk

$$Error_{true} = \int \frac{1}{2} | y - f(x,\alpha) | dP(x,y)$$

- 经验风险（训练误差）： Empirical risk

$$Error_{train}(\alpha) = \frac{1}{2n} \sum_{i=1}^{n} | y_i - f(x_i,\alpha) |$$

- 结构风险的界： 以概率 $1-\eta$

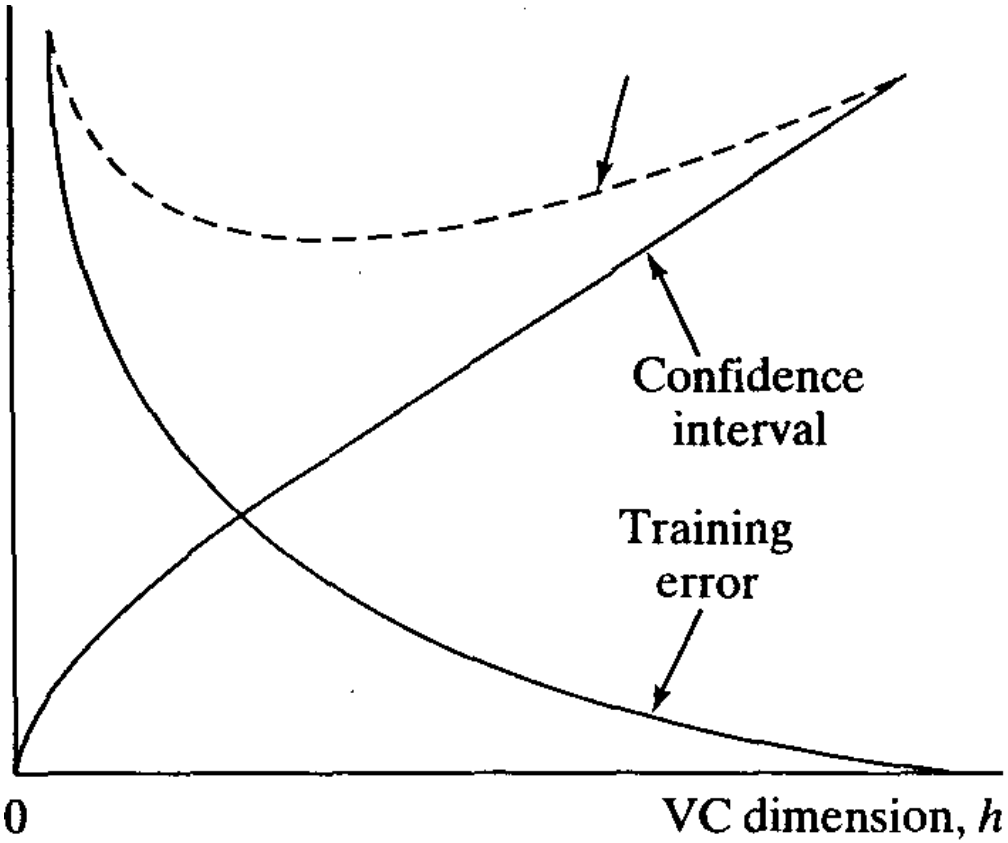
VC dimension

VC confidence

$$Error_{true} \le Error_{train} + \sqrt{\left(\frac{h(\log(2n/h)+1) - \log(\eta/4)}{n} \right)}$$

证明：在PAC理论部分

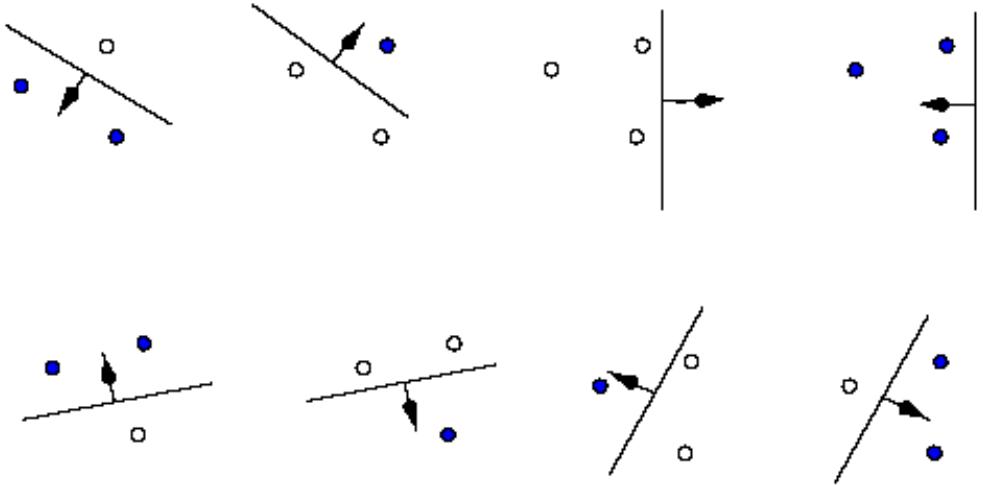
结构风险最小化原则



$$\cdots \subset S_{n-1} \subset S_n \subset S_{n+1} \cdots$$

结构风险最小化

- Vapnik-Chervonenkis (VC) dimension
  - VC 维定义为一组函数，如平面、直线等在空间打散（任意分类）样本的能力
  - 例如，直线的VC 维为3，当4个样本点时，无法任意分类
- （直线右侧分类-1，左侧为1）



内 容

- SVM概述
- 结构风险最小化
- 线性SVM
- SVM求解
- 处理线性不可分问题
- SVM训练算法

线性SVM

线性分类器解决的问题：

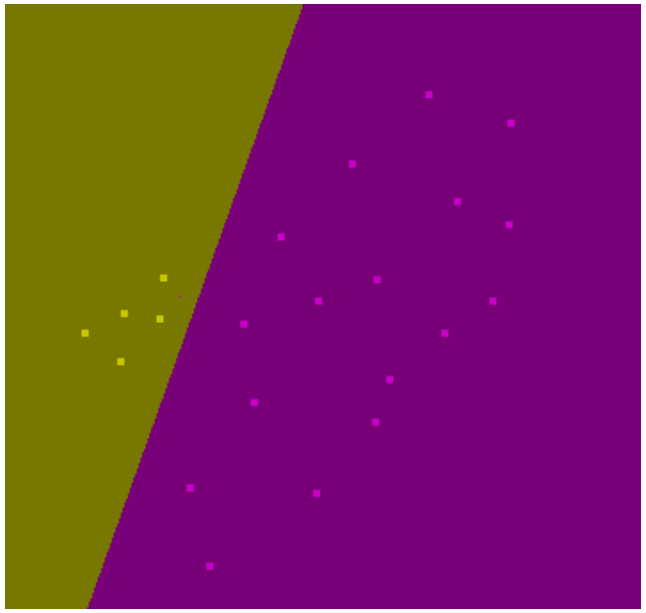
- 根据一个带有类别标号的训练集合，通过学习一个线性分类面，使得训练集合按照类别进行划分
- 通常转化成一个优化问题
- 以两类监督分类问题问题为例来解释

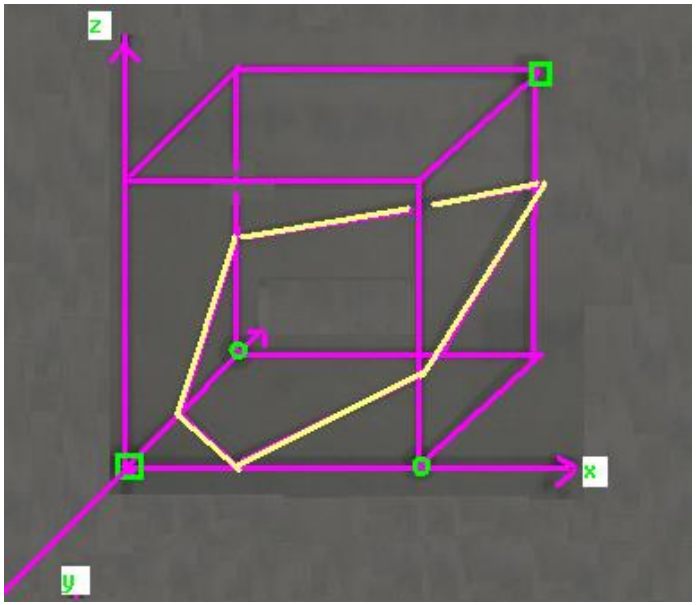
线性SVM

分类面：把一个空间按照类别切分两部分的平面，在二维空间中，分类面相当于一条直线，三维空间中相当于一个平面，高维空间为超平面

线性分类面函数形式为：　　$f(\mathbf{x}) = w^T\mathbf{x} + b$

w^T,b是分类面函数参数,x是输入的样本, w^T权向量,b是偏移量



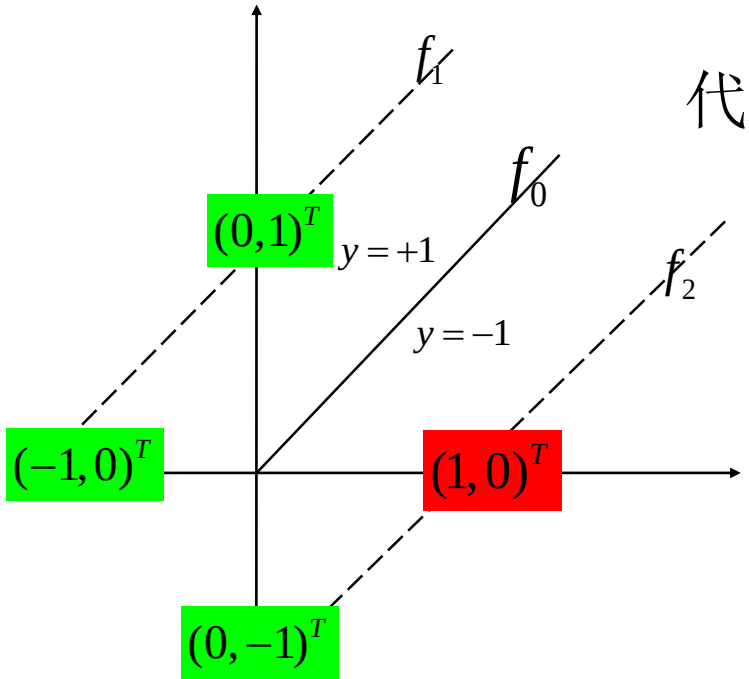


线性SVM

线性分类面函数： $f(x) = w^T \mathbf{x} + b$

$$y = \mathrm{sgn}\left(f\left(x\right)\right) = \begin{array}{l} f(x) = w^T \mathbf{x}_i + b > 0 \quad \text{for } y = +1 \\ f(x) = w^T \mathbf{x}_i + b < 0 \quad \text{for } y = -1 \end{array}$$

如果 $f(x) = w^T \mathbf{x}_i + b = 0$ 则为$\mathrm{x_i}$分类面上的点，反之也成立。



代入(1,0),(0,1)验证f_0

$w^T = (-1,1); b = 0$

$f_0(x) = (-1,1)\mathbf{x} = 0$

$f_1(x) = (-1,1)\mathbf{x} - 1 = 0$

$f_2(x) = (-1,1)\mathbf{x} + 1 = 0$

如果w相同，则分类面是平行的，b是一个偏移量

线性SVM

线性分类器学习：从给定的训练样本确定w^T和b这两个参数。得到参数以后，就确定了分类面，从而可以对输入样本进行分类。

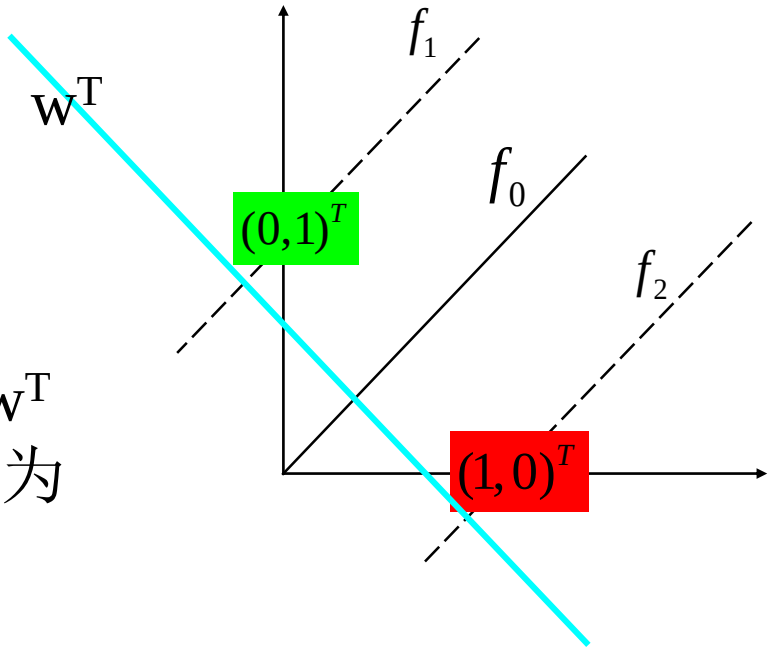
阐述一下各个参数的性质

$$w^T \cdot \mathbf{x} + b = 0;$$

$$w^T \cdot \mathbf{s_1} + b = w^T \cdot \mathbf{s_2} + b$$

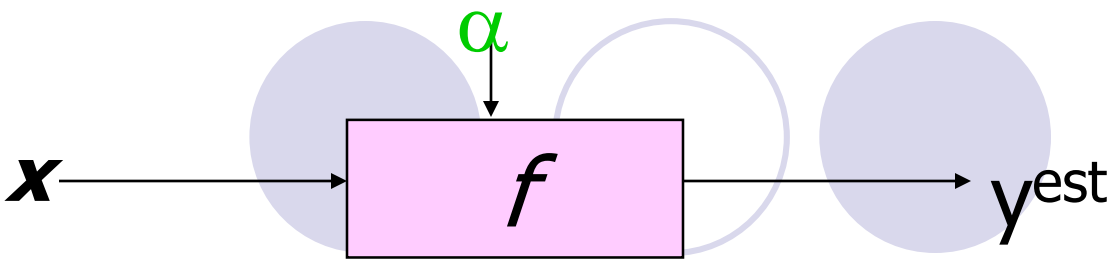
$$w^T \cdot (\mathbf{s_1} - \mathbf{s_2}) = 0$$

当s1和s2都在分类面上时，这表明w^T和分类面上任意向量正交，并称w^T为分类面的法向量。



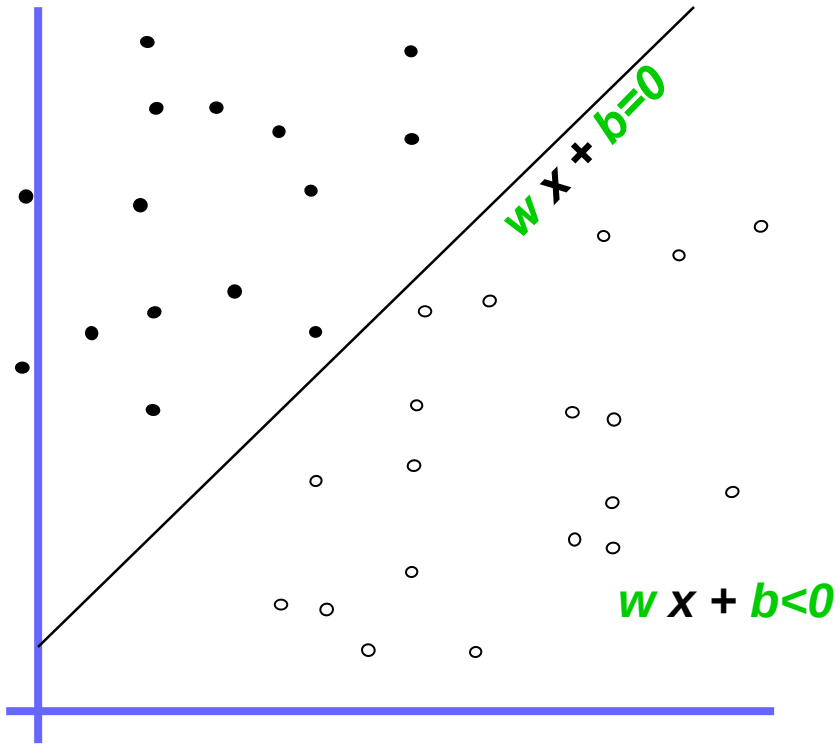
几何解释：线性分类器的作用就是把输入样本在法向量上投影变成一维变量，然后给一个阈值来分类

线性SVM



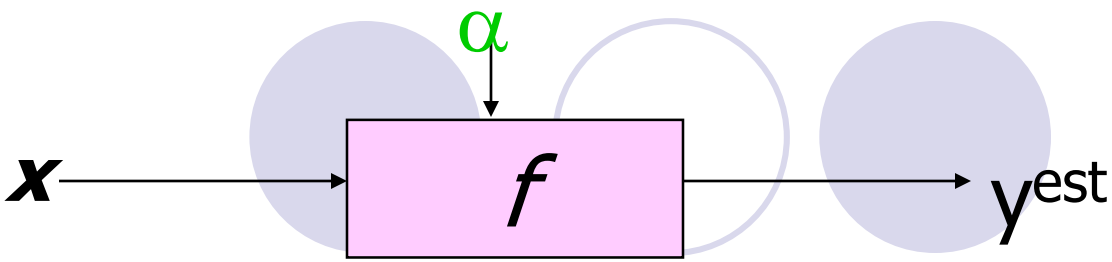
- 表示 +1
- 表示 -1

$f(\mathbf{x},\mathbf{w},b) = sign(\mathbf{w}\,\mathbf{x} + b)$



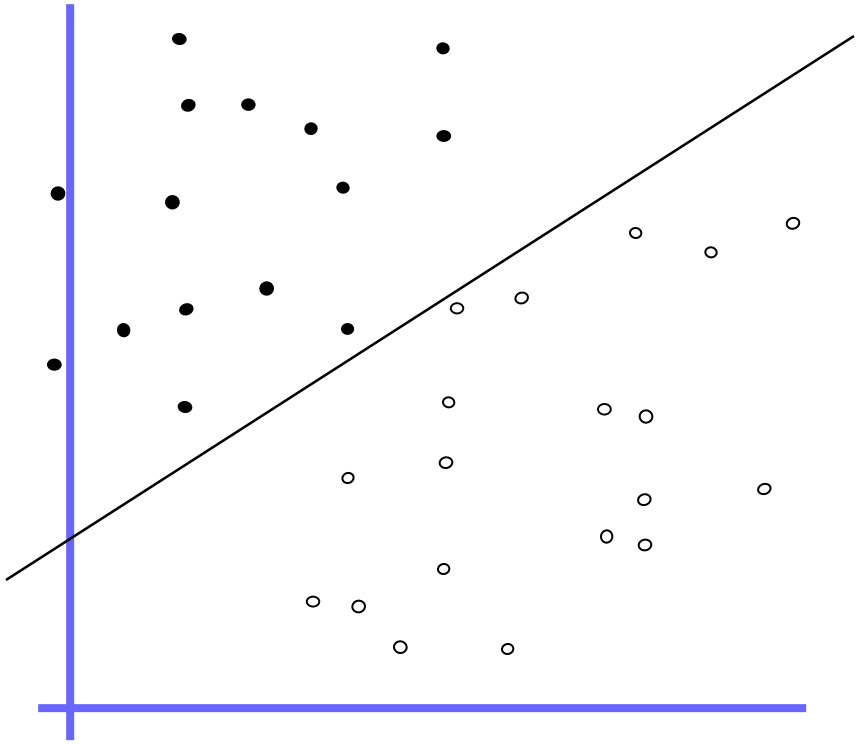
如何分类这些数据？

线性SVM



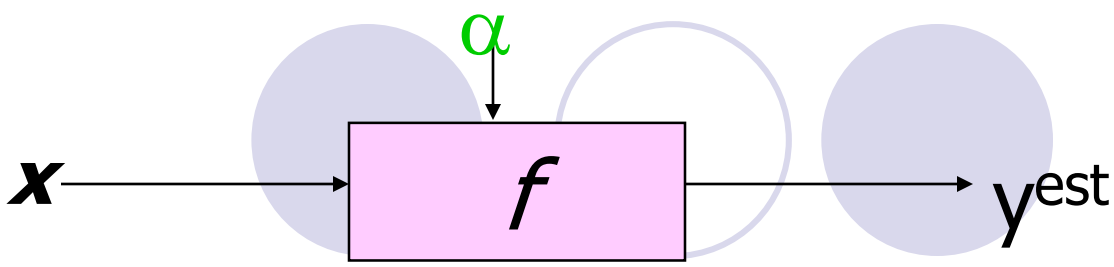
$f(\mathbf{x},\mathbf{w},b) = sign(\mathbf{w}\,\mathbf{x} + b)$

- 表示 +1
- 表示 -1



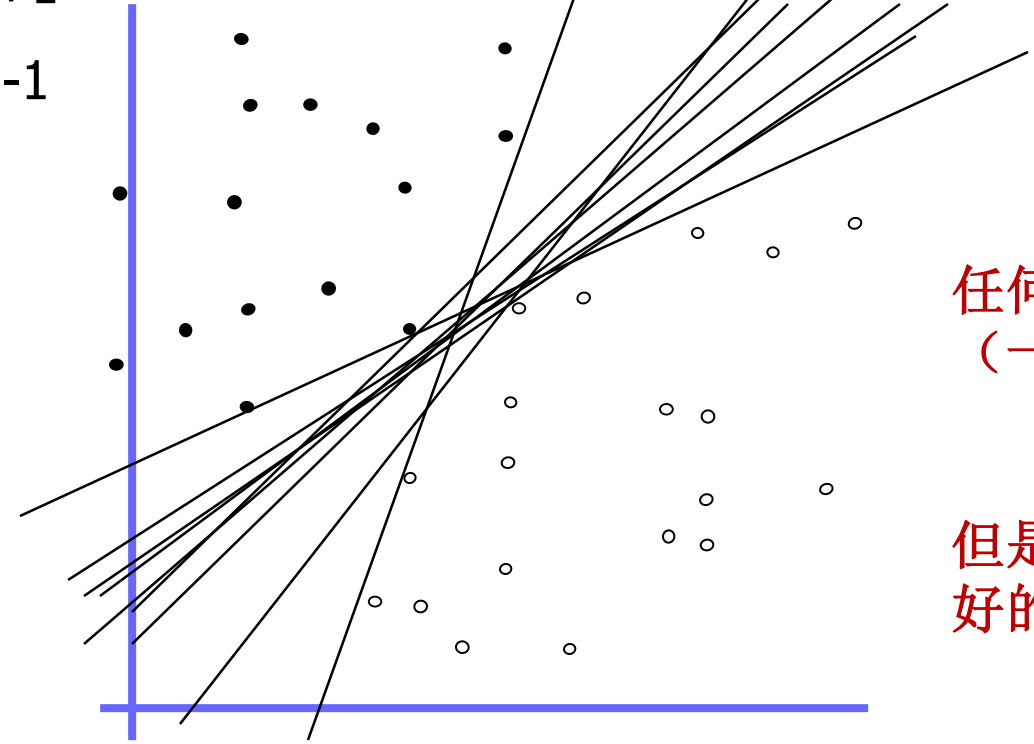
如何分类这些数据？

线性SVM



- 表示 +1
- 表示 -1

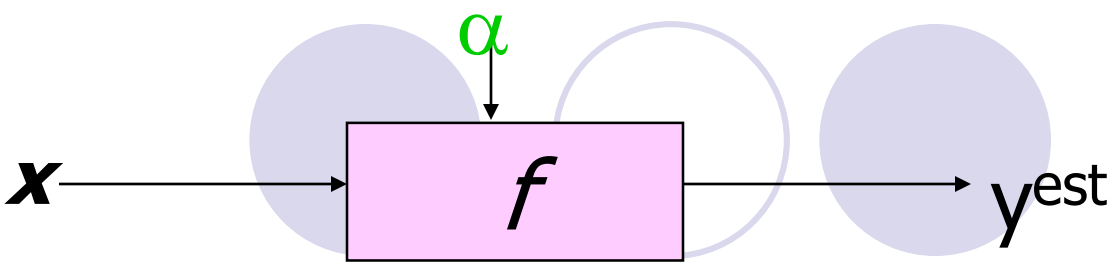
$f(\mathbf{x},\mathbf{w},b) = sign(\mathbf{w}\,\mathbf{x} + b)$



任何一个分类器（一条线）都有效

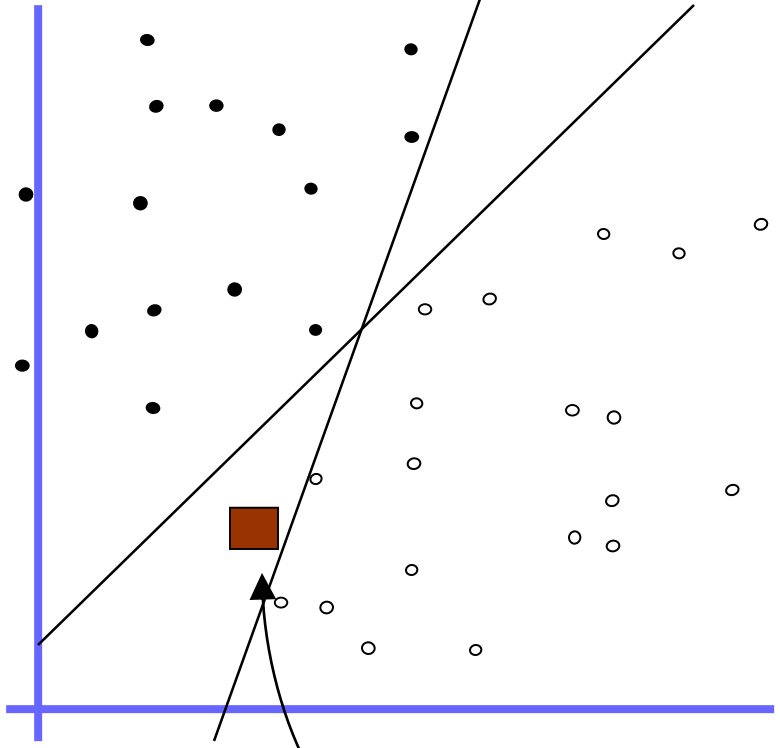
但是…哪一个是最好的?

线性SVM



- 表示 +1
- 表示 -1

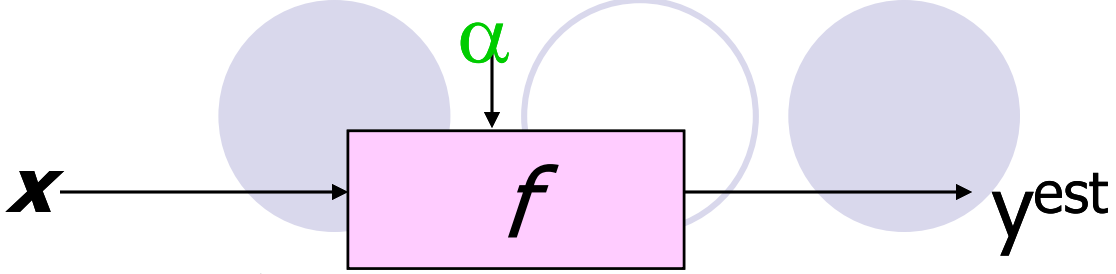
$$f(\mathbf{x},\mathbf{w},b) = sign(\mathbf{w}\,\mathbf{x} + b)$$



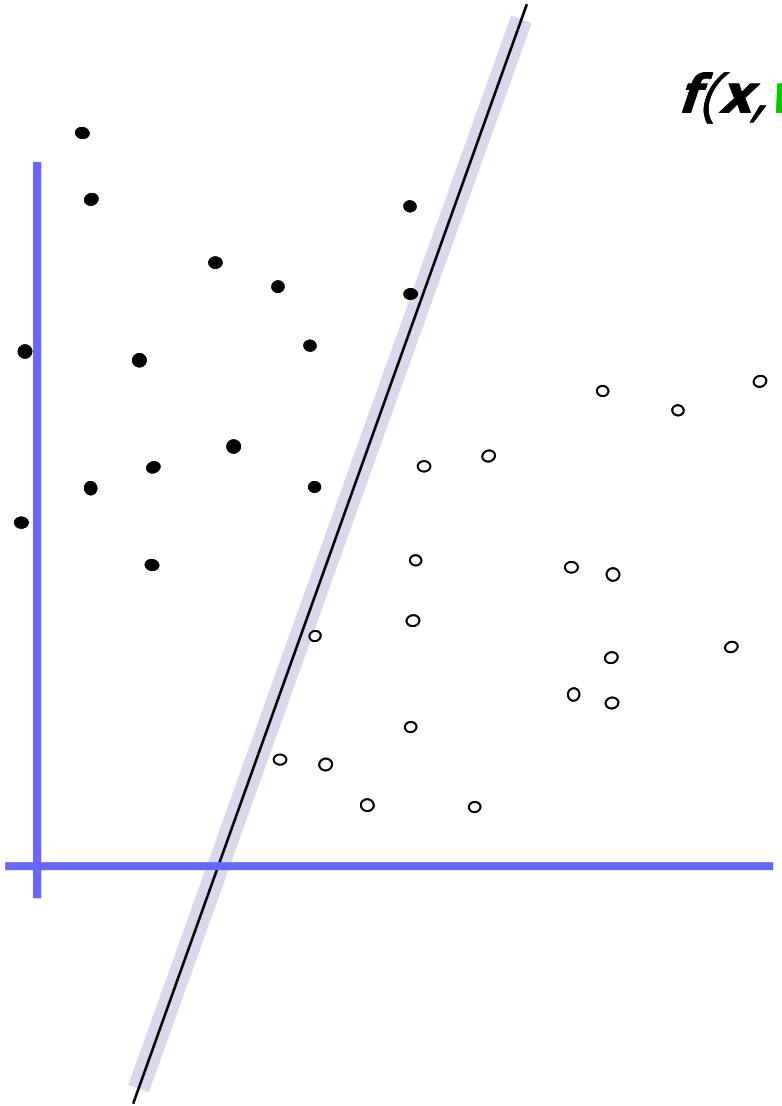
假设测试数据出现在这里

线性SVM

Max-margin



- 表示 +1
- 表示 -1



$f(x,w,b) = sign(w\,x + b)$

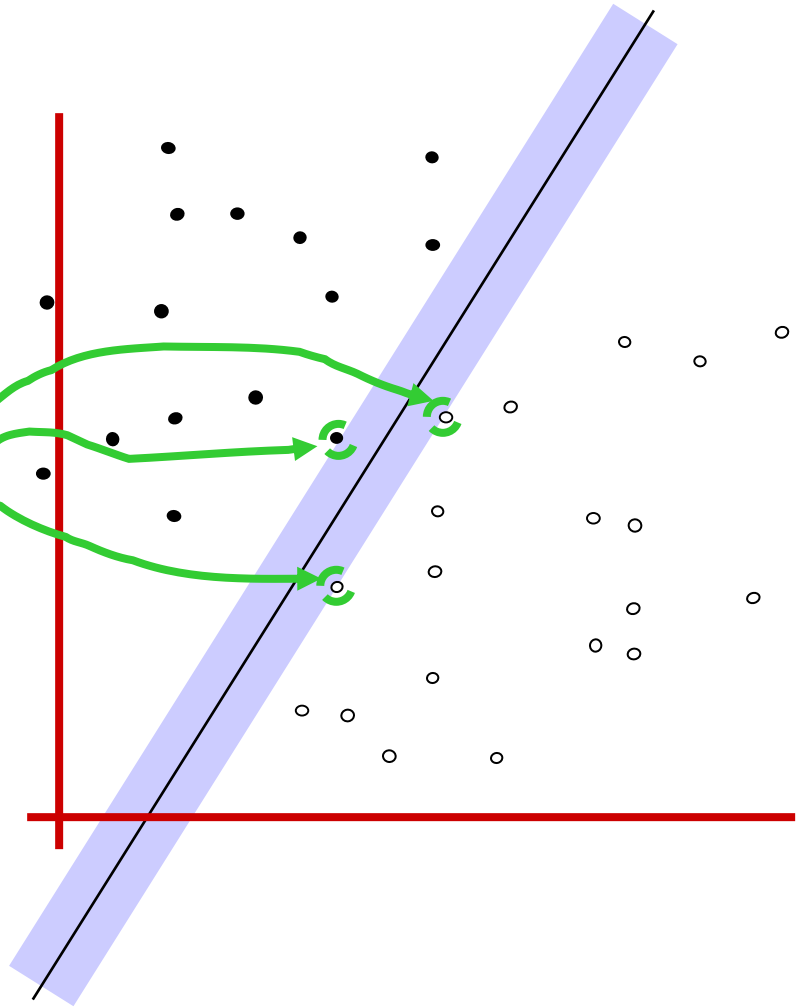
定义分类器的边界

以改善分类性能.

线性SVM

Max-margin

- 表示 +1
- 表示 -1



Support Vectors
是边界上的一些样本点

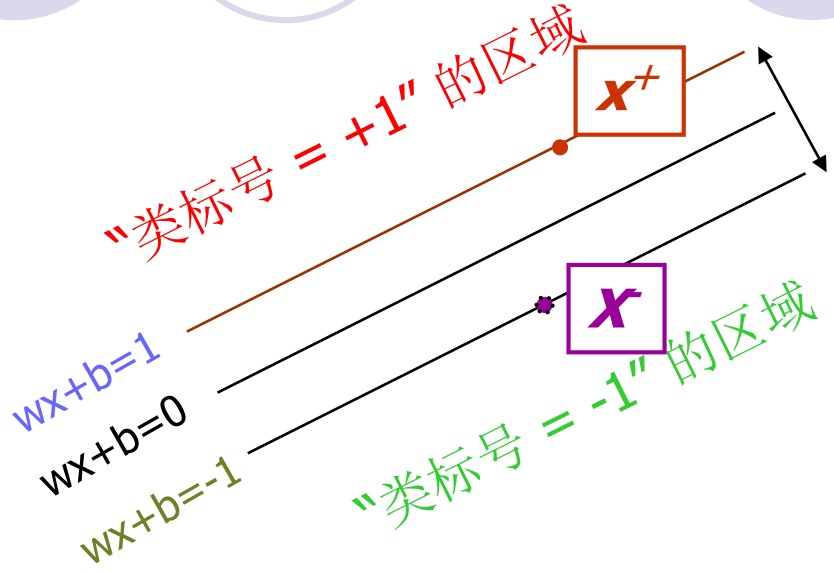
1. Maximum margin linear classifier就是最大化边界地带的线性分类器
2. 只有Margin上的样本点是重要的，其他样本都不重要
3. 实践证明这种假设效果非常好，理论证明它最符合PAC理论.

线性SVM

- SVM从线性可分情况下的分类面发展而来
- **Max-Margin不仅仅要求经验风险尽可能的小，而且要求分类间隔最大**
- SVM考虑寻找一个满足分类要求的分类面
- 两类样本中离分类面最近的点且平行于分类面的训练样本就叫做**支持向量**（Support Vectors）

线性SVM

Max-margin



M=Margin Width

线性关系:

- ***w*** . ***x***$^{+}$ + *b* = *+1*
- ***w*** . ***x***$^{-}$ + *b* = *-1*
- ***w*** . (***x***$^{+}$-***x***$^{-}$) = *2*

$$\text{Margin} = \frac{w \cdot (x^{+} - x^{-})}{|w|} = \frac{2}{|w|}$$

线性SVM

- 假定训练数据

$$(\mathbf{x_1}, y_1), \ldots, (\mathbf{x_l}, y_n), \mathbf{x} \in R^d, y \in \{+1, -1\}$$

- 线性分类面函数

$$(w^T . \mathbf{x}) + b = 0, w \in R^d, b \in R$$

- Max-margin转化成优化问题

$$\max\left(\frac{2}{\|w\|}\right) \Leftrightarrow \min\left(\|w\|^2\right)$$

线性SVM

- 最优分类面求解问题表示成**约束优化问题**
  - 最小化目标函数 $Q(w)=\frac{1}{2}\|w\|^2=\frac{1}{2}(w^T\cdot w)$
  - 约束条件 $y_i((w^T\cdot \mathbf{x_i})+b)\geq 1, i=1,...,n$

$$\min \frac{1}{2}\|w\|^2$$

$$y_i((w^T\cdot \mathbf{x_i})+b)\geq 1, i=1,...,n$$

内 容

- SVM概述
- 结构风险最小化
- 线性SVM
- SVM求解
- 处理线性不可分问题
- SVM训练算法

线性SVM求解

- 最优分类面问题表示成约束优化问题
  - 最小化目标函数 $Q(w)=\frac{1}{2}\|w\|^2=\frac{1}{2}(w^T\cdot w)$
  - 约束条件 $y_i((w^T\cdot \mathbf{x_i})+b)\geq 1, i=1,...,n$

- 定义Lagrange函数

$$L(w,b,\alpha)=\tfrac{1}{2}\|w\|^2-\sum_{i=1}^{n}\alpha_i(y_i\cdot((w^T\cdot \mathbf{x}_i)+b)-1)$$

线性SVM求解

Lagrange函数优化问题的回顾

- ✓ 1629年，Lagrange最初是为了解决等式约束的最优化解
- ✓ 1951年，Kuhn和Tucker进一步把这个方法扩展到具有不等式约束的情况下，而他们理论实际基于Karush的工作。
- ✓ 通过对偶理论简化约束条件即Karush-Kuhn-Tucker互补条件解决了支持向量机的优化问题

线性SVM求解

- Lagrange函数 $L(w,b,\alpha)=\frac{1}{2}\|w\|^2-\sum_{i=1}^{n}\alpha_i(y_i\cdot((w^T\cdot\mathbf{x}_i)+b)-1)$

$$\frac{\partial}{\partial b}L(w,b,\alpha)=0;\ \frac{\partial}{\partial w}L(w,b,\alpha)=0$$

$$\sum_{i=1}^{n}a_i y_i=0\ ;\quad w=\sum_{i=1}^{n}\alpha_i y_i x_i$$

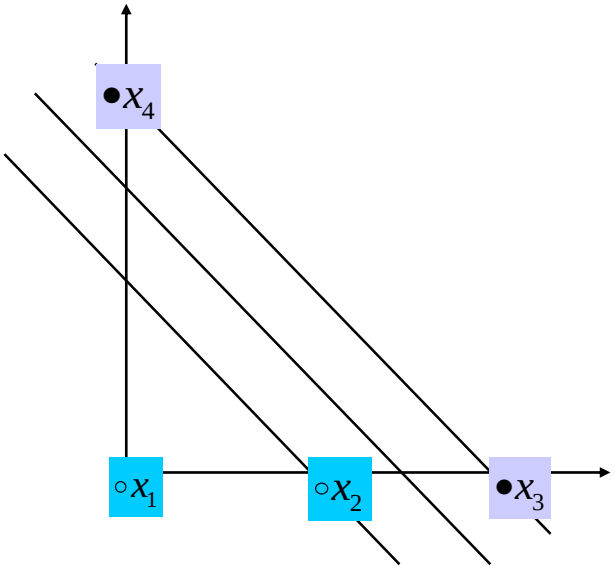
$$W(\alpha)=\sum_{i=1}^{n}\alpha_i-\frac{1}{2}\sum_{i,j=1}^{n}\alpha_i\alpha_j y_i y_j(\mathbf{x}_i\cdot\mathbf{x}_j)$$

KKT条件

$$\alpha_i\geq 0,\ \ i=1,\ldots,n,\ \ \ and\ \ \sum_{i=1}^{n}\alpha_i y_i=0$$

$$\alpha_i(y_i\cdot((w^T\cdot\mathbf{x}_i)+b)-1)=0$$

线性SVM求解:例子



$\mathbf{x}_1 = (0, 0)^T,\ y_1 = +1$

$\mathbf{x}_2 = (1, 0)^T,\ y_2 = +1$

$\mathbf{x}_3 = (2, 0)^T,\ y_3 = -1$

$\mathbf{x}_4 = (0, 2)^T,\ y_4 = -1$

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

代入x,y值

$$W(\alpha) = (\alpha_1 + \alpha_2 + \alpha_3 + \alpha_4) - \frac{1}{2}(\alpha_2^2 - 4\alpha_2\alpha_3 + 4\alpha_3^2 + 4\alpha_4^2)$$

可调用Matlab中的二次规划程序，求得α_1，α_2，α_3，α_4的值，进而求得w和b的值。

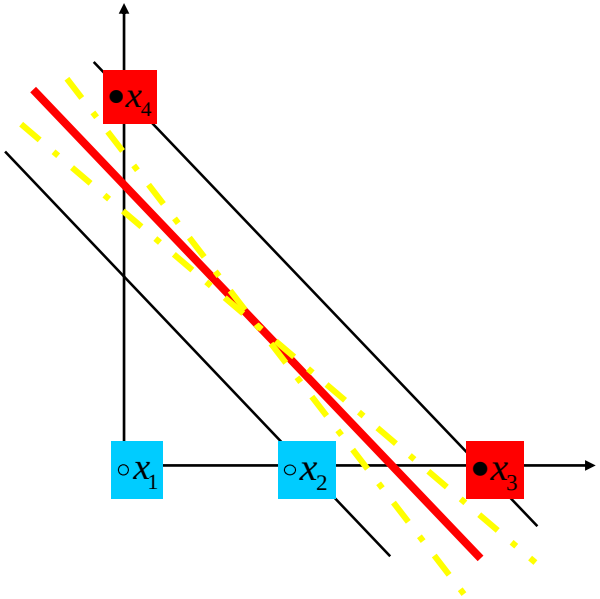
代入(3/2,0),(0,3/2)点可以知道

$$\begin{cases}\alpha_1 = 0\\ \alpha_2 = 1\\ \alpha_3 = 3/4\\ \alpha_4 = 1/4\end{cases}$$

$$w = \begin{bmatrix}1\\0\end{bmatrix} - 3/4\begin{bmatrix}2\\0\end{bmatrix} - 1/4\begin{bmatrix}0\\2\end{bmatrix} = \begin{bmatrix}-1/2\\-1/2\end{bmatrix}$$

$$b = -\frac{1}{2}[-1/2, -1/2]\begin{bmatrix}3\\0\end{bmatrix} = 3/4$$

$$f(x) = W^T x + b = \begin{bmatrix}-1/2\\-1/2\end{bmatrix}^T x + 3/4$$



思考：当数据量很大的时候怎么办？

内 容

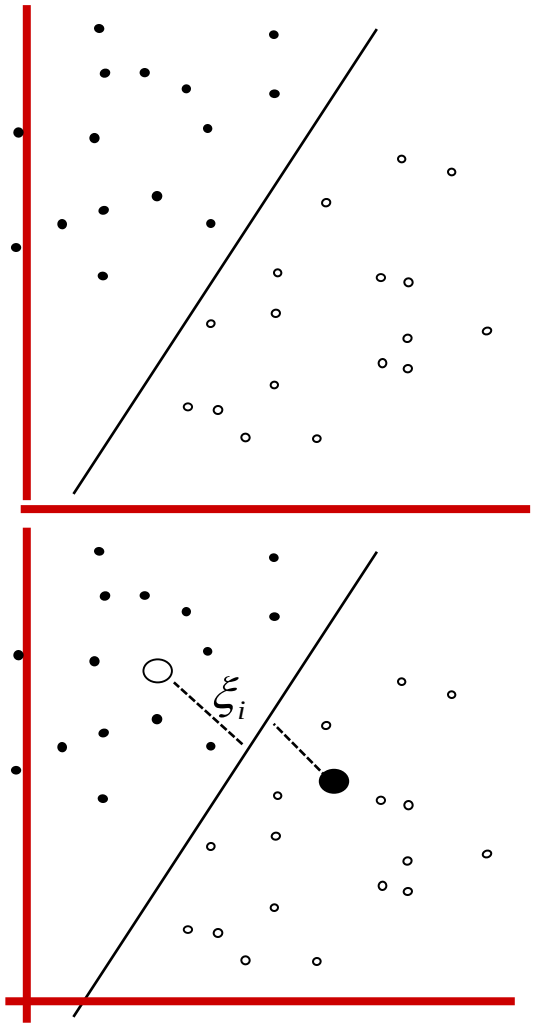
- SVM概述
- 结构风险最小化
- 线性SVM
- SVM求解
- 处理线性不可分问题
- SVM训练算法

处理线性不可分的情况

- 软间隔
- Kernel SVM

线性SVM求解:软间隔

- 最优化问题



$$\min \frac{1}{2}\|w\|^2$$

$$\text{s.t.} \quad y_i((w^T \cdot \mathbf{x_i}) + b) \geq 1, i = 1, ..., n$$

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

$$\text{s.t.} \qquad y_i\left(\mathbf{w} \cdot \mathbf{x}_i + b\right) - 1 + \xi_i \geq 0, \ i = 1, 2, ..., n$$

$$\xi_i \geq 0, \ i = 1, 2, ..., n$$

线性SVM:软间隔

- 将上述问题表示成拉格朗日乘子式

$$L=\frac{1}{2}\|w\|^2+C\sum_i\xi_i+\sum_i\alpha_i[1-\xi_i-y_i(wx_i+b)]-\sum_i\pi_i\xi_i$$

Kuhn-Tucker条件

$$\frac{\partial L}{\partial w}=w-\sum_i\alpha_i y_i x_i=0$$

$$\frac{\partial L}{\partial b}=\sum_i\alpha_i y_i=0$$

$$\frac{\partial L}{\partial \xi}=C-\alpha_i-\pi_i=0$$

$$\alpha_i\geq 0,\ \ \alpha_i[1-\xi_i-y_i(wx_i+b)]=0$$

$$\pi_i\geq 0$$

线性SVM:软间隔

- 得到

$$w = \sum_i \alpha_i y_i x_i$$

$$\sum_i \alpha_i y_i = 0;$$

$$0 \le \alpha_i \le C$$

$$\alpha_i [1 - \xi_i - y_i (w x_i - b)] = 0$$

$$\pi_i \ge 0$$

$$C - \alpha_i - \pi_i = 0$$

- 只要确定α，便可解出w与b

线性SVM:软间隔

- 将上述条件代入L中

$$L=\frac{1}{2}\|w\|^2+\sum_i\alpha_i-\sum_i\alpha_i y_i x_i w-b\sum_i\alpha_i y_i+\sum_i(C-\pi_i-\alpha_i)\xi_i$$

$$L=\frac{1}{2}\sum_{i,j}\alpha_i\alpha_j y_i y_j x_i\cdot x_j+\sum_i\alpha_i-\sum_{i,j}\alpha_i\alpha_j y_i y_j x_i\cdot x_j-b\sum_i\alpha_i y_i+\sum_i(C-\pi_i-\alpha_i)\xi_i$$

- 新的优化问题

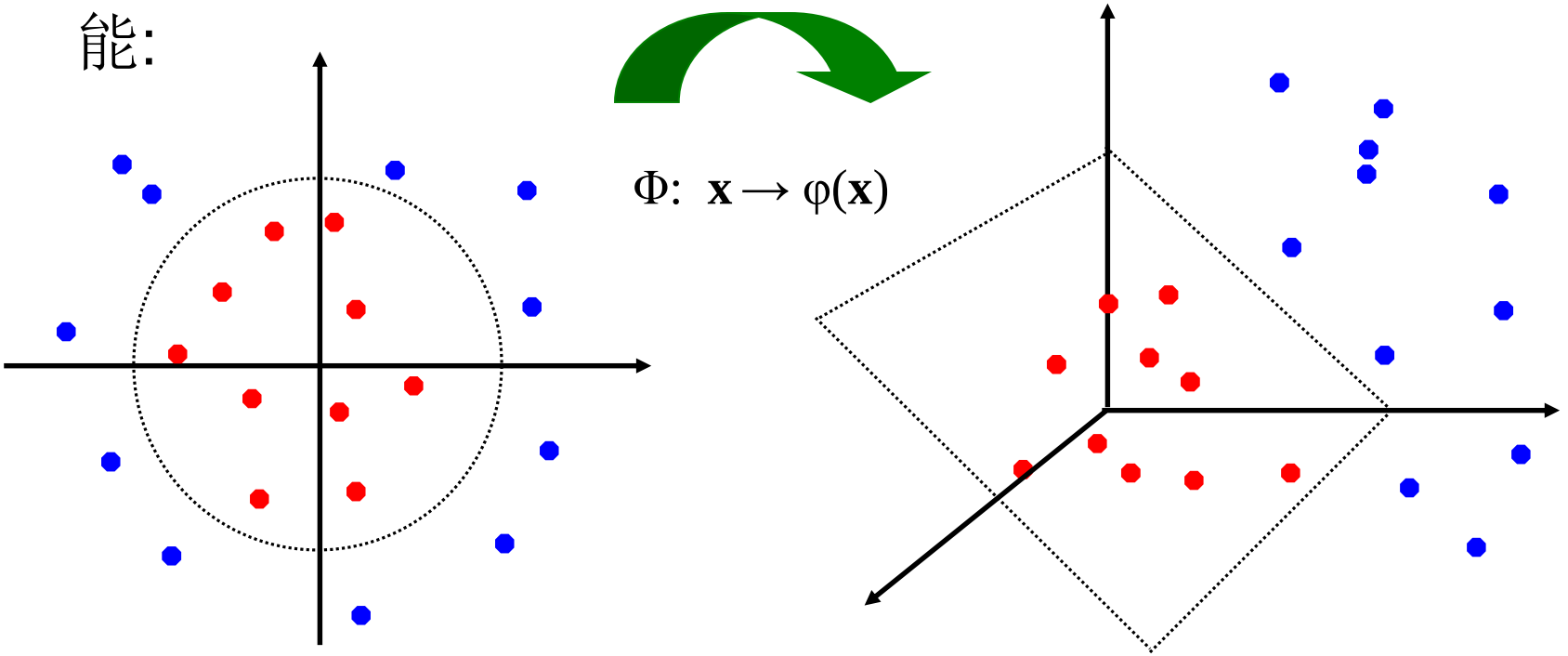
(Quadratic Programing)

$$L=\sum_i\alpha_i-\frac{1}{2}\sum_{i,j}\alpha_i\alpha_j y_i y_j x_i\cdot x_j$$

$$0\le\alpha_i\le C\ \sum_i\alpha_i y_i=0$$

Kernel SVMs

直观的想法: 原始数据可以映射到一个高维空间，这些原始数据在低维空间基于线性分类面可能是不可分的，或者分得不好，而在高维空间却可以获得好的分类性能:



什么样的函数可以做核函数？

- 如果 $K(x_i,x_j)$ 总可以写成：

  $K(x_i,x_j)= \varphi(x_i)^T\varphi(x_j)$ 的形式，则K可以做核函数.

- Mercer's 定律:

  每一个半正定的对称函数都可以是一个核函数

- 半正定的对称函数对应半正定的矩阵:

K=

$K(x_1,x_1)$	$K(x_1,x_2)$	$K(x_1,x_3)$	…	$K(x_1,x_N)$
$K(x_2,x_1)$	$K(x_2,x_2)$	$K(x_2,x_3)$		$K(x_2,x_N)$
…	…	…	…	…
$K(x_N,x_1)$	$K(x_N,x_2)$	$K(x_N,x_3)$	…	$K(x_N,x_N)$

核函数举例：

- 线性分类器的运算是内积运算
  - $K(x_i,x_j)=x_i^T x_j$
- 如果 $\Phi: x \rightarrow \varphi(x)$, 则内积变为:

$$K(x_i,x_j)= \varphi(x_i)^T \varphi(x_j)$$

- “核函数” 就变为了高维空间的内积函数.
- 例如:

2-维向量 $x=[x_1 \ x_2]$; 如果 $K(x_i,x_j)=(1 + x_i^T x_j)^2$,

且 $K(x_i,x_j)= \varphi(x_i)^T \varphi(x_j)$:

$$K(x_i,x_j)=(1 + x_i^T x_j)^2,$$

$$= 1+ x_{i1}^2 x_{j1}^2 + 2 x_{i1}x_{j1} x_{i2}x_{j2}+ x_{i2}^2 x_{j2}^2 + 2x_{i1}x_{j1} + 2x_{i2}x_{j2}$$

$$= [1 \ x_{i1}^2 \ \sqrt{2} x_{i1}x_{i2} \ x_{i2}^2 \ \sqrt{2}x_{i1} \ \sqrt{2}x_{i2}] [1 \ x_{j1}^2 \ \sqrt{2} x_{j1}x_{j2} \ x_{j2}^2 \ \sqrt{2}x_{j1} \ \sqrt{2}x_{j2}]^T$$

$$= \varphi(x_i)^T \varphi(x_j), \quad \text{其中 } \varphi(x) = [1 \ x_1^2 \ \sqrt{2} x_1x_2 \ x_2^2 \ \sqrt{2}x_1 \ \sqrt{2}x_2]$$

核函数举例：

- 线性: $K(x_i,x_j)= x_i^T x_j$
- 多项式函数 p: $K(x_i,x_j)= (1+ x_i^T x_j)^p$
- RBF核函数:

$$K(\mathbf{x_i},\mathbf{x_j}) = \exp(-\frac{\left\|\mathbf{x_i}-\mathbf{x_j}\right\|^2}{2\sigma^2})$$

Kernel SVMs

- 目标函数形式:

$$Q(\alpha) = \sum \alpha_i - \frac{1}{2} \sum \alpha_i \cdot \alpha_j \cdot y_i \cdot y_j \cdot K(x_i, x_j)$$

$$\alpha_i \geq 0, i = 1, ..., N$$

- 求解获得系数和支撑向量以后，分类器构造:

线性SVM:

$$f(x) = w^T x + b = \sum \alpha_i \cdot y_i \cdot x_i^T \cdot x + b = \sum \alpha_i \cdot y_i \cdot (x_i^T \cdot x) + b$$

Kernal SVM:

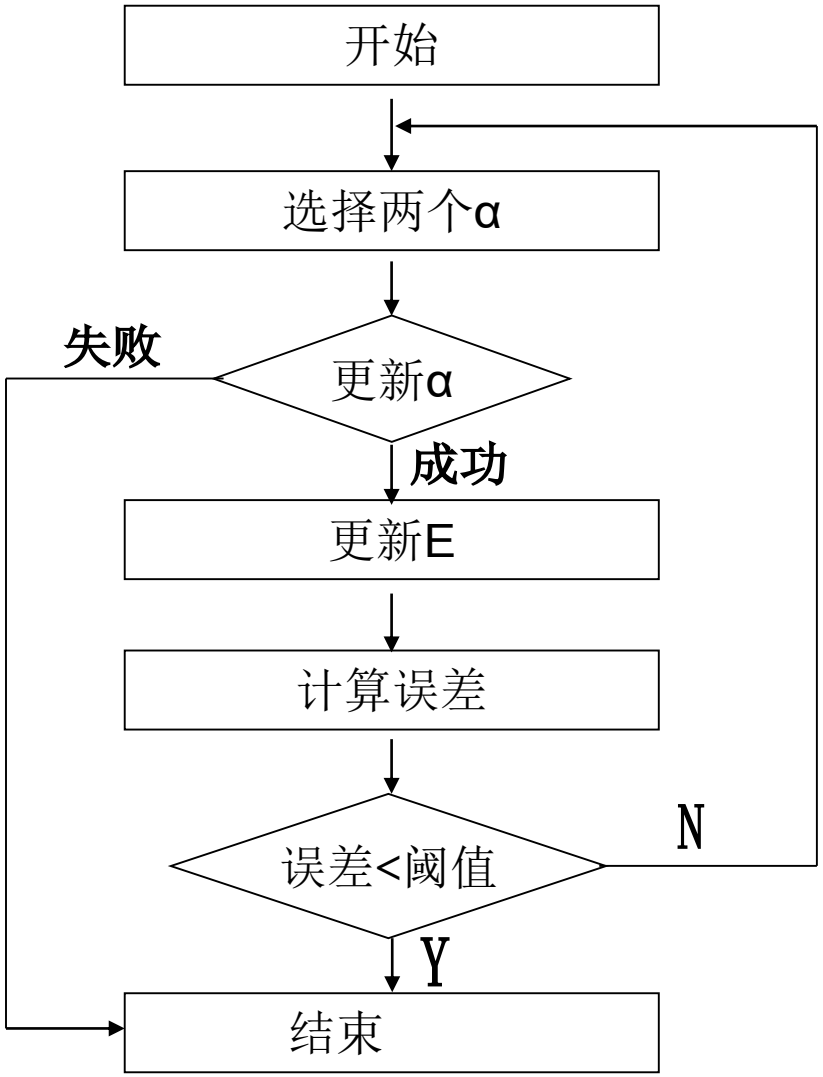
$$f(x) = \sum \alpha_i \cdot y_i \cdot K(x_i^T, x) + b = \sum \alpha_i \cdot y_i \cdot \varphi(x_i)^T \cdot \varphi(x) + b$$

内 容

- SVM概述
- 结构风险最小化
- 线性SVM
- SVM求解
- 处理线性不可分问题
- **SVM训练算法**

SVM训练算法-SMO

算法流程：每次选取两个α进行更新



最后得到迭代公式：

$$\alpha_2^{new} = \alpha_2^{old} + (E_1 - E_2) y_2 / k$$

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

通过逐步增加支持向量，分类函数逐渐变得复杂，所以VC维逐渐的增加，经验风险减小，可以看到这就是结构风险最小化的求解过程