支持向量机模型

戚 琦

网络与交换技术国家重点实验室 网络智能研究中心 科研楼511

qiqi8266@bupt.edu.cn 13466759972

大纲

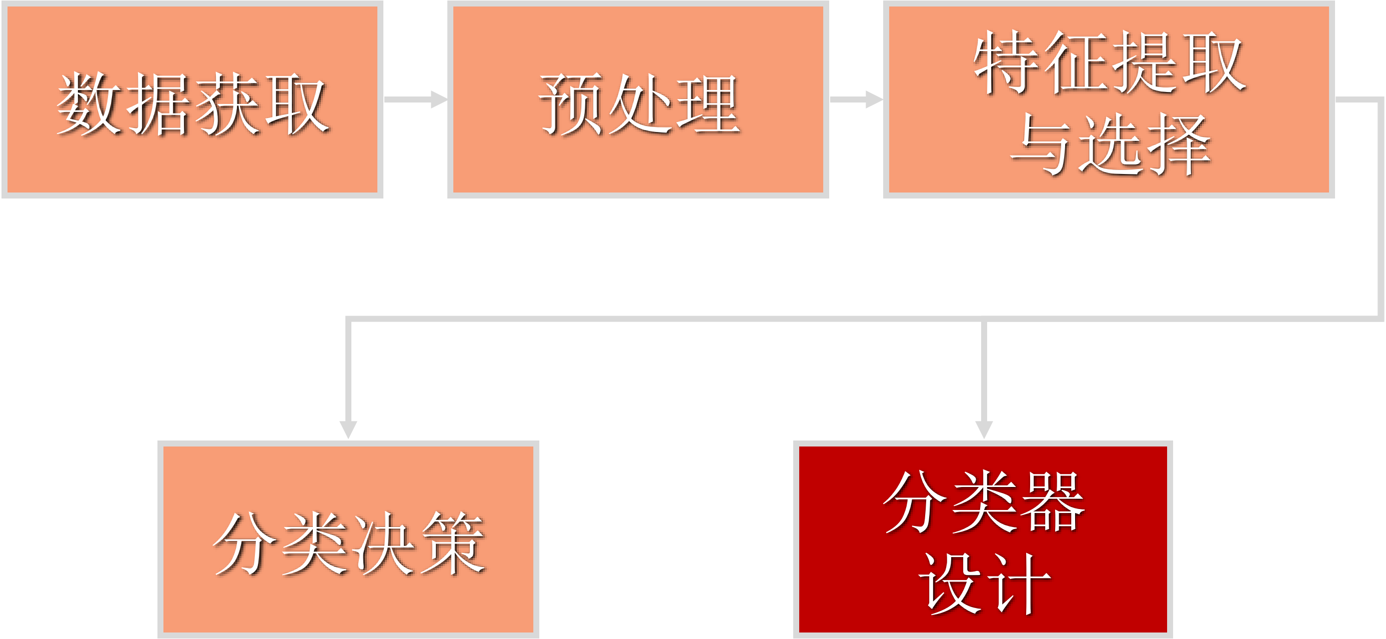
- 支持向量机基本概念---线性可分类问题
- 非线性问题
- 对偶问题

推荐书籍《统计学习方法》

视频课程：浙江大学胡老师《机器学习》

台湾大学Hsuan-Tien Lin《Machine Learning Techniques》

回顾：监督学习



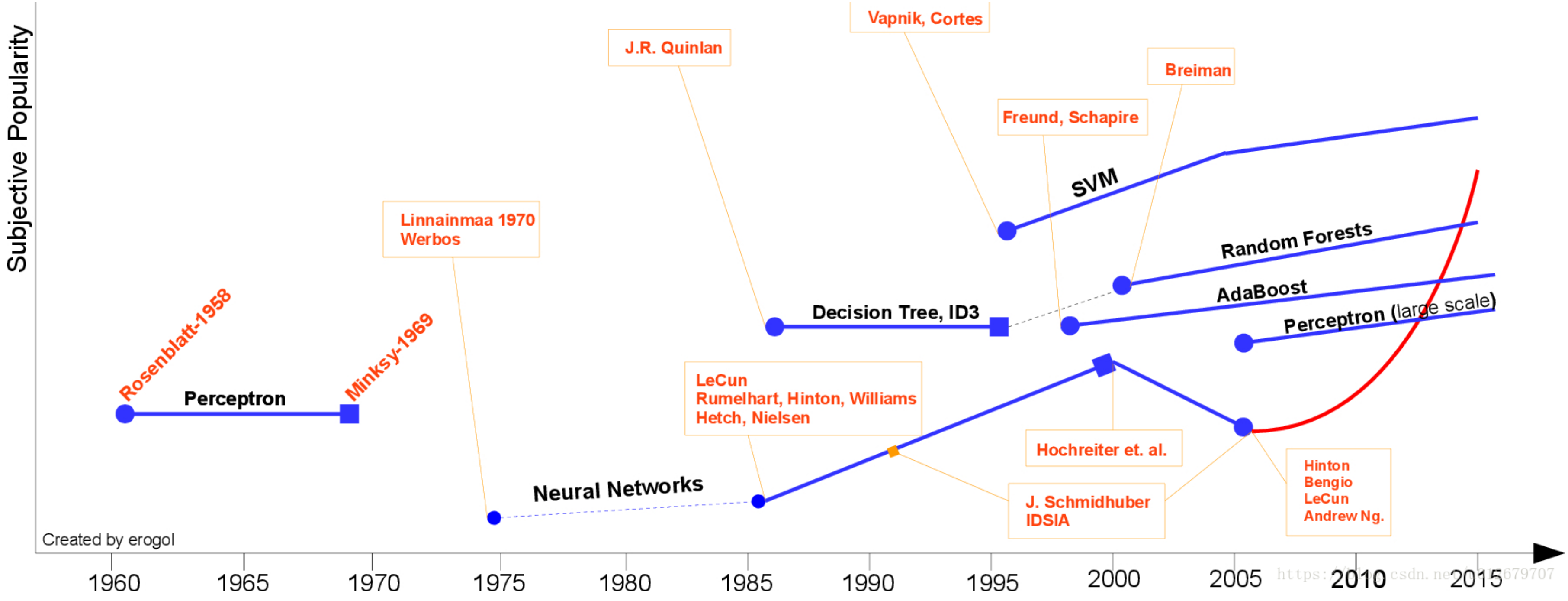
支持向量机

- Vapnik 从1960年开始关于统计学习理论研究
- 1995年Vapnik发展了支持向量机理论
- 支持向量机是基于统计学习理论的一种实用的机器学习方法
- 支持向量机具有优美的数学表达
- SVM在解决小样本、非线性及高维模式识别问题中表现出许多特有的优势，并能够推广应用到函数拟合等其他机器学习问题中

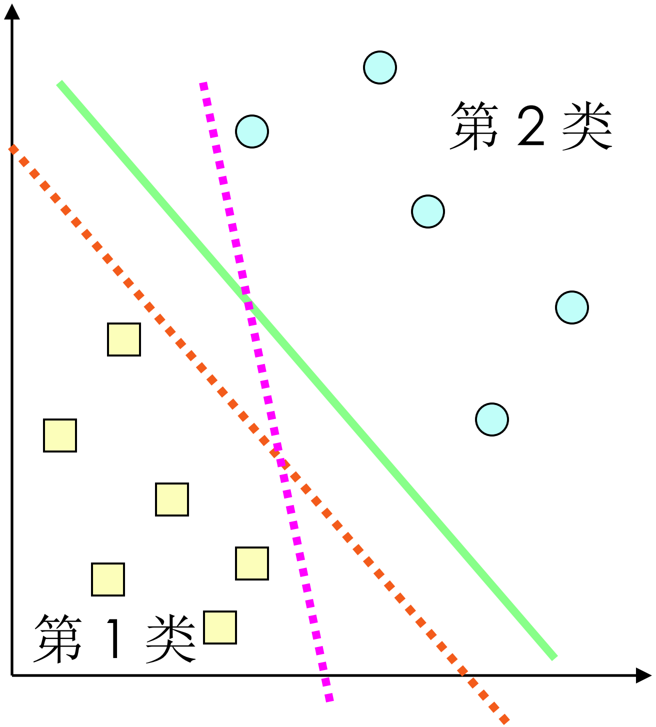
Vladimir Naumovich Vapnik，俄罗斯统计学家、数学家

支持向量机

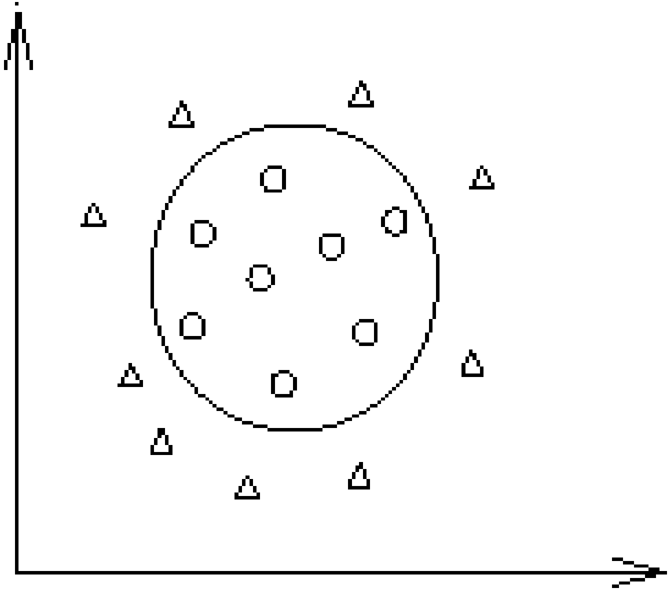
机器学习算法的时间线来自于Eren Golge



线性可分样本集



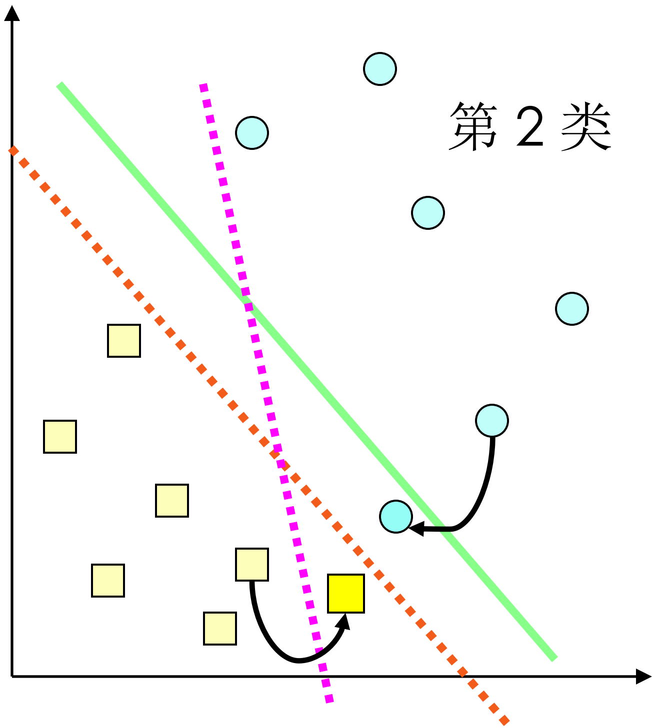
线性可分样本



非线性可分样本

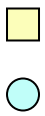
线性可分

- 问题：无数条线，哪个直线是最好的



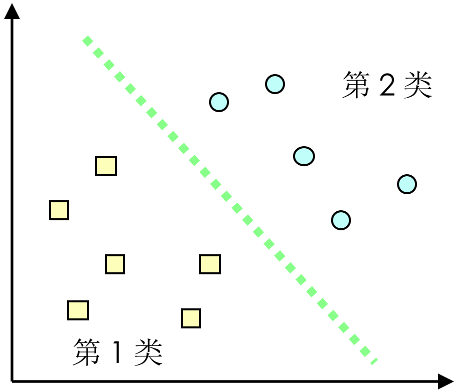
对于误差的容忍程度

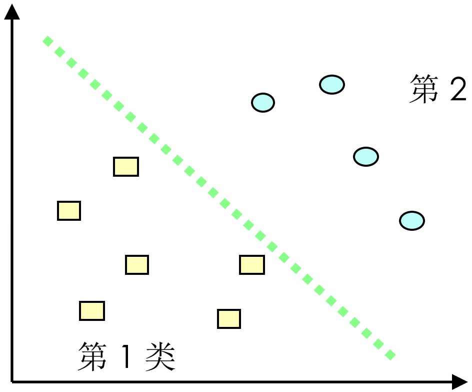
线性可分

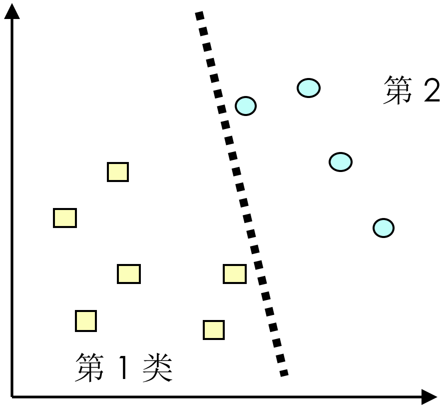


如何定义

- 如何定义这条线？
- 定义一个衡量每一条线的标准，每条线都能算出来一个性能指标
- 哪条线能让这个性能指标最大

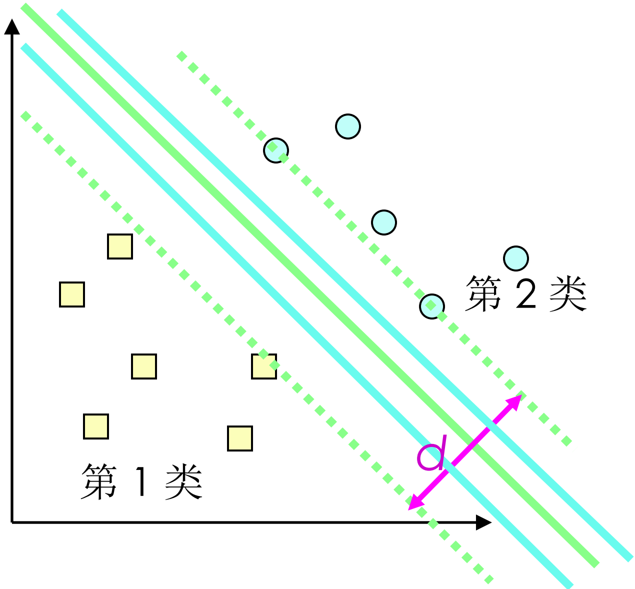






好的决策边界：间隔大

- 决策边界离两类数据应尽可能远

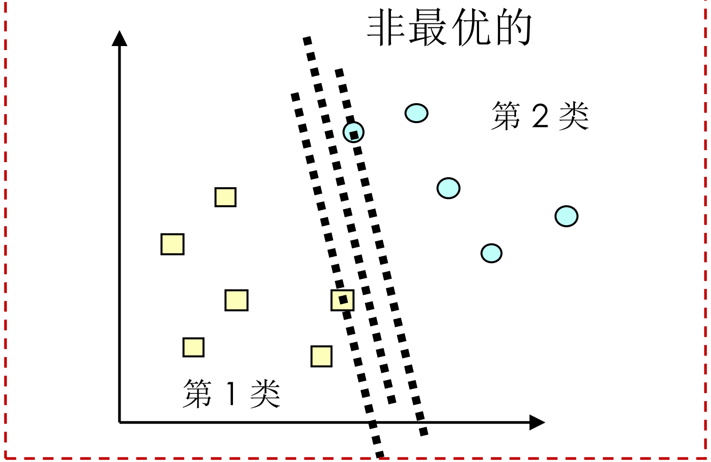


d：最大化间隔（Margin）

不唯一，选择处于中间位置的线

让这条线平行移动，直到能够插到某一个或几个圆圈为止

这个距离作为性能指标，绿色线是这个距离最大的一条线



将平行线插到的向量称为支持向量

定义

训练数据和标签：$(x_1, y_1)\ (x_2, y_2) \dots (x_n, y_n)$

向量　标签

$$x_1 = \begin{bmatrix} x_{11} \\ x_{12} \\ \dots \\ x_{1m} \end{bmatrix}$$

$y_i = 1 \quad y_i = -1$

线性模型：(w, b)　$w^{\mathrm{T}}x + b$=0（超平面Hyperplane）

w：向量，维度与x一样

常数

$$w = \begin{bmatrix} w_1 \\ w_2 \\ \dots \\ w_m \end{bmatrix}$$

要找到描述这个超平面的线性方程

通过所有x和y的取值，来算出来w和b

线性可分的训练集

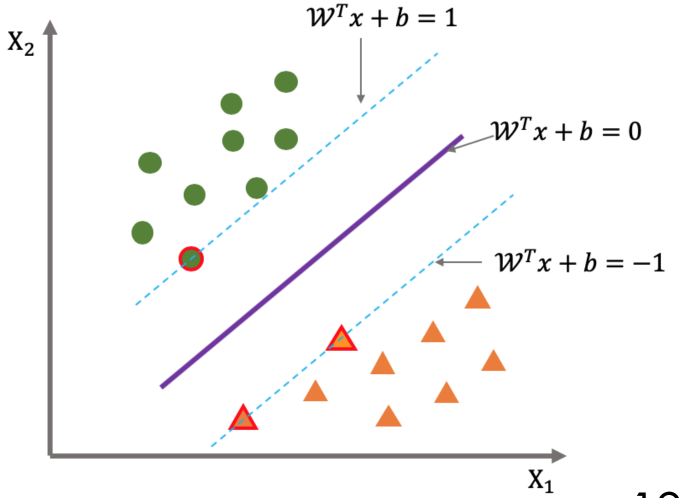
- 一个训练集线性可分是指：

$\{(x_i, y_i)\}, i = 1\sim N$

$\exists(w, b)$，使：

对$\forall\ i = 1\sim N$，有：

1. 若y_i=+1,则$w^{\mathrm{T}}x + b$ ≥0
2. 若y_i=-1,则 $w^{\mathrm{T}}x + b$ ≤0



1963年

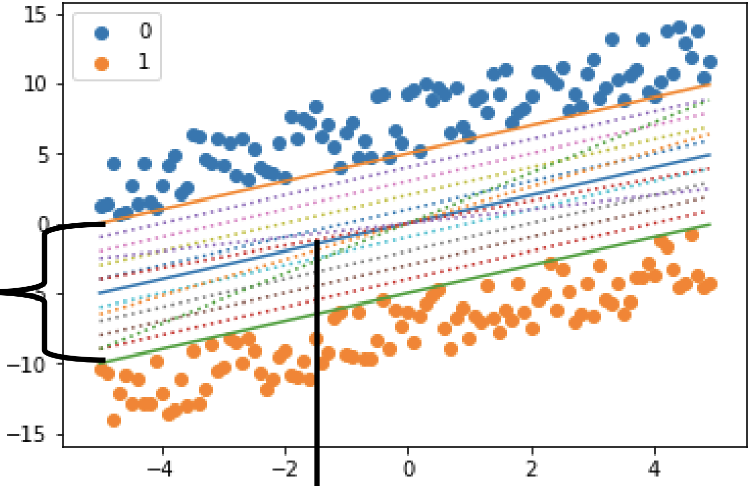
统一表示

$y_i[w^{\mathrm{T}}x + b] \geq 0$

如果两类数据之间存在分离超平面，则称数据为**线性可分**（linearly Separable）。

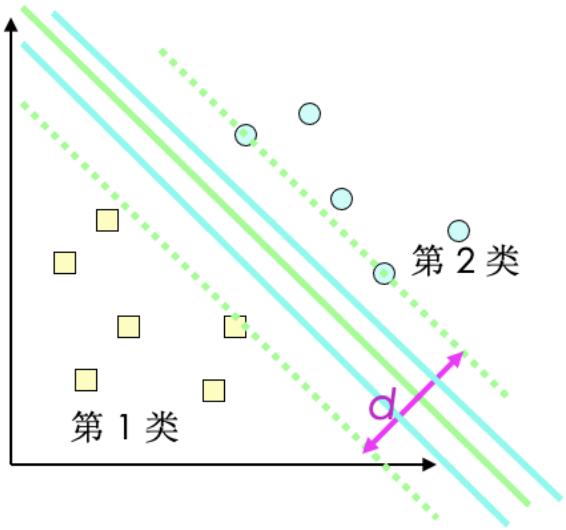
*线性可分数据的超平面

- 当训练数据集线性可分时，存在无穷多个分离超平面可将两类数据正确分开。
- 感知机利用误分类最小的策略，求得分离超平面，但有无穷多个解
- 支持向量机利用**间隔最大化**求最优分离超平面，解唯一



最大间隔分类器

- 针对分离超平面不唯一的问题，一种解决方法是使**分离超平面离两类数据尽量远**。
- 希望在两类数据之间有一条**“隔离带”**，而且这条隔离带越宽越好。
- 这就是所谓**最大间隔分类器**(maximal margin classifier);俗称“最宽街道法”(widest street approach)，即在两类数据之间建一条最宽的街道。



求最大间隔

- 如何得到间隔最大的超平面

$w^{\mathrm{T}}x+b$=0与 $aw^{\mathrm{T}}x+ab$=0是同一个平面，$a \in R^+$

点到平面的距离：

平面：$w_1x+w_2y+b=0$

则（x_0, y_0）到此平面的距离：

$$d=\frac{|w_1x_0+\ w_2y_0+b|}{\sqrt{w_1^2+w_2^2}}$$

向量x_0到平面的距离：

$w^{\mathrm{T}}x+b=0$的距离

$$d=\frac{w^Tx_0+b}{\|w\|^2}$$

$$\sqrt{w_1^2+w_2^2+\cdots+w_m^2}$$

总结：SVM是最大化间隔的分类算法

优化问题

$$d = \frac{w^T x_0 + b}{\|w\|^2}$$

用a缩放 $(w, b) \longrightarrow (aw, ab)$

最终使其在支持向量x_0上有： $|w^T x_0 + b| = 1$

此时支持向量与平面的距离： $d = \frac{1}{\|w\|^2}$ 最小化$\|w\|$，其他点大于d

y_i取正负1，可以协调两个类，也可以改成任意整数，差距就是a

支持向量机做了一个优化问题

最小化（Minimize） $\|w\|^2$

约束条件（Subject to ） $y_i[w^{\mathrm{T}}x + b] \geq 1$（i=1~N）

优化问题

数据集如果线性可分，就能求得w和b，满足条件：

$$\min \ \frac{1}{2}\|w\|^2 = \frac{1}{2}(w_1^2 + w_2^2 + \cdots + w_m^2)$$

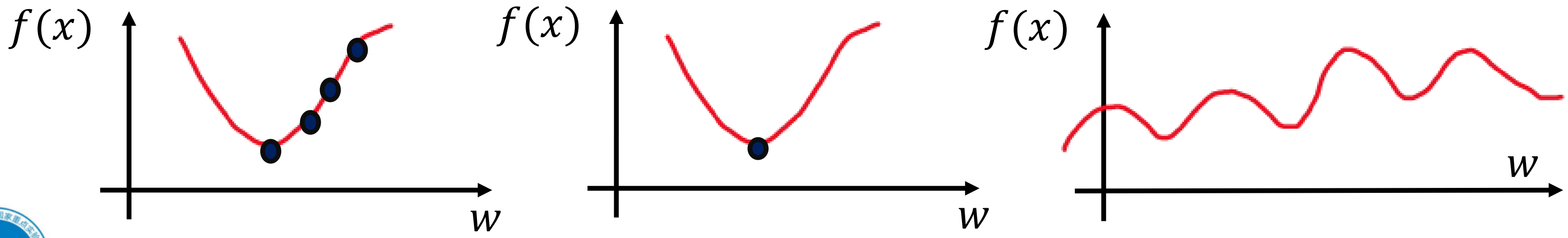
$\frac{1}{2}$是为了求导方便

$$s.t. \quad y_i[w^{\mathrm{T}}x_i + b] \geq 1 \text{（i=1~N）}$$

目标优化问题是一个凸优化问题中的一种二次规划问题

*二次规划问题

- 二次规划（quantity programming）
  - 目标函数是二次项；约束条件是一次项
  - 无解或者一个极小值
  - 二次规划是计算机已解决的问题，其局部极值就是全局极值
- 用试探方法，求极值，例如梯度下降
- 多维情况下很困难，有时人眼无法看到所有情况



小结

- **SVM是最大化间隔（Margin）的分类算法**
- 优化问题

训练样本$\{(x_i, y_i)\}_{i=1\sim N}$

- 优化目标

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

$$s.t. \quad y_i[w^T x_i + b] \geq 1$$

$$(i = 1\sim N)$$

凸优化问题中的二次规划问题

$\Rightarrow$

求解得到分离超平面

$$w^* \cdot x + b^* = 0$$

决策函数

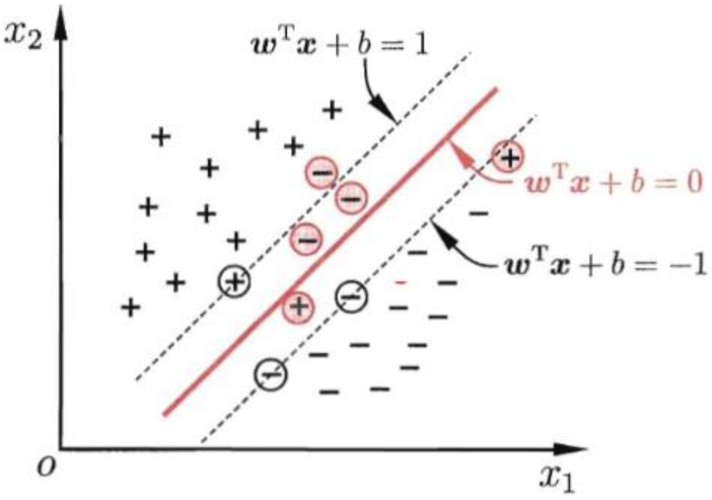
$$f(x) = sign(w^* \cdot x + b^*)$$

大纲

- 支持向量机基本概念
- 非线性问题
  - 软间隔
  - 高维映射
  - 核函数
- 对偶问题

非线性---软间隔

■ 训练数据中的特异点，去掉后，剩下大部分样本点组成的集合满足线性可分



■ 引入“软间隔”的概念：**允许支持向量机在一些样本上不满足约束，**即允许一些实例位于街道之上或者位于错误的一侧，这样就称为软间隔分类

软间隔最大化

给每个样本，加一个**松弛变量**，让w和b满足条件

将线性不可分的学习问题转换为**凸二次规划问题**，即**软件间隔最大化**

正则项

最小化：$\frac{1}{2}\|w\|^2 + \boxed{C\sum_{i=1}^{N}\xi_i}$ 松弛变量

约束条件：（1）$y_i[w^{\mathrm{T}}x + b] \geq 1 - \xi_i$

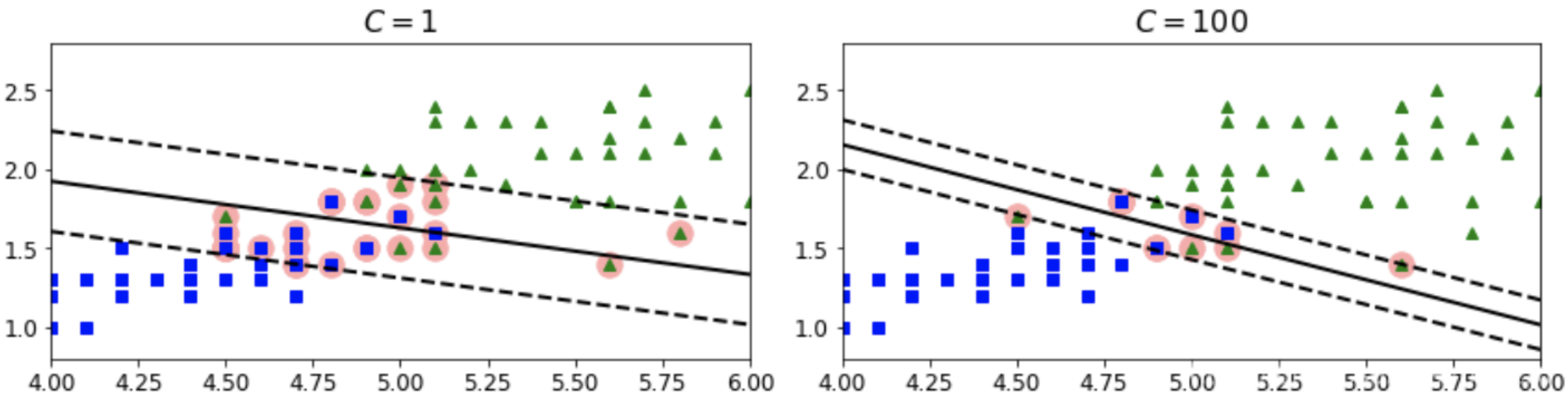
（2）$\xi_i \geq 0$, $i=1 \sim N$

线性支持向量机

求解得到分离超平面$w^* \cdot x + b^* = 0$，以及决策函数

$$f(x) = sign(w^* \cdot x + b^*)$$

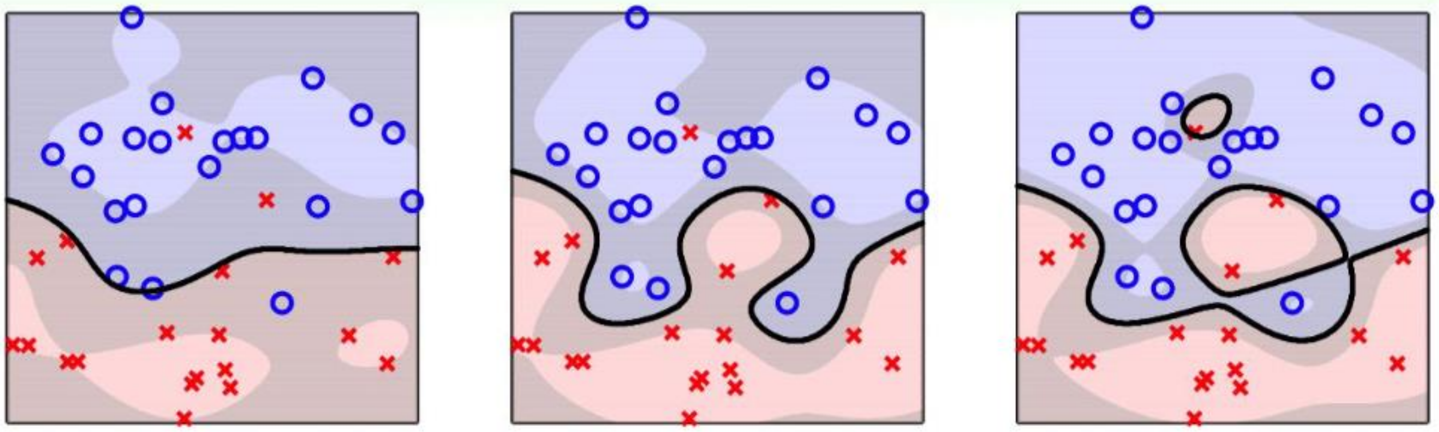
软间隔参数



尽可能在保持街道宽阔或限制间隔违例之间找到一个平衡

间隔或限制间隔违例之间的关系由超参数C控制，C越大，间隔越窄，但是违例也更少，反之则间隔越宽，但是违例也越多。

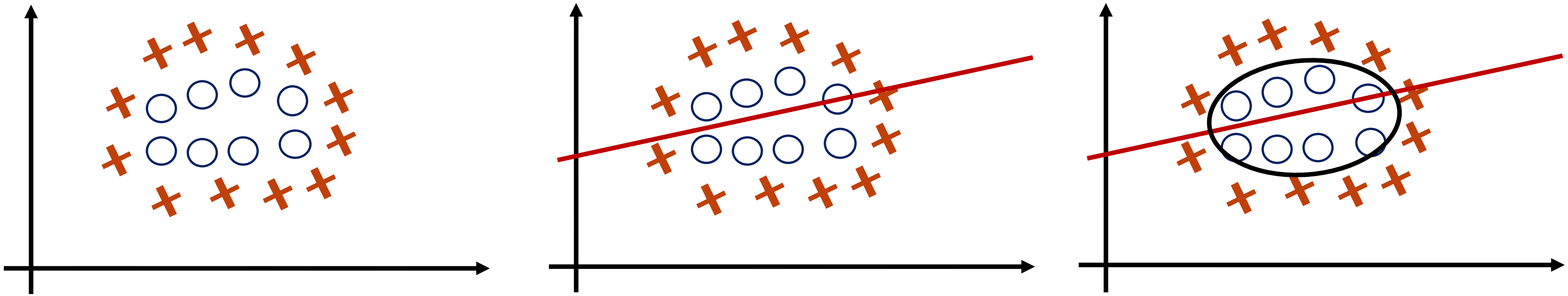
软间隔参数



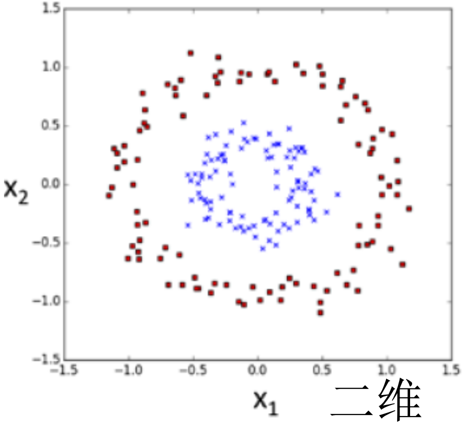
$C = 1$ $C = 10$ $C = 100$

- C 越大越容易造成过拟合

非线性可分

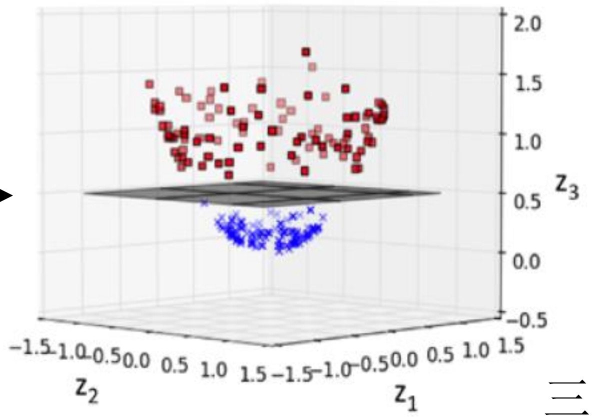


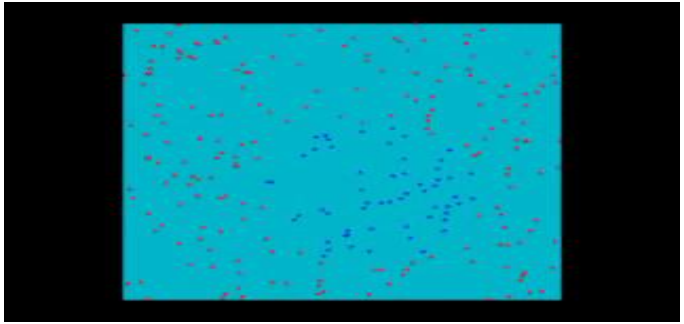
仍然找直线，但是到一个高维空间中找直线



二维

ϕ





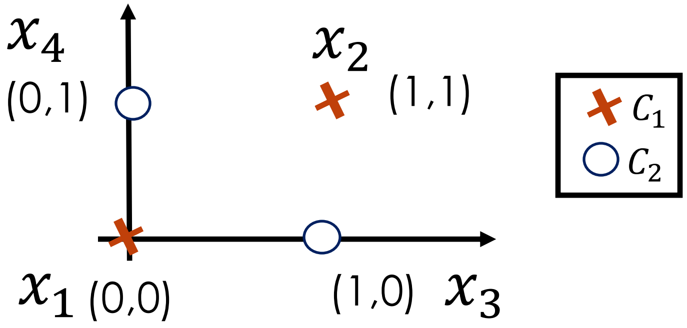
三维

高维映射

定义一个高维映射 $\boldsymbol{\phi(x)}$

低维 $x \longrightarrow \phi(x)$ 高维

例如，最简单的非线性可分问题：异或问题



$$x_1 = \begin{bmatrix}0\\0\end{bmatrix} \in C_1 \quad x_2 = \begin{bmatrix}1\\1\end{bmatrix} \in C_1$$

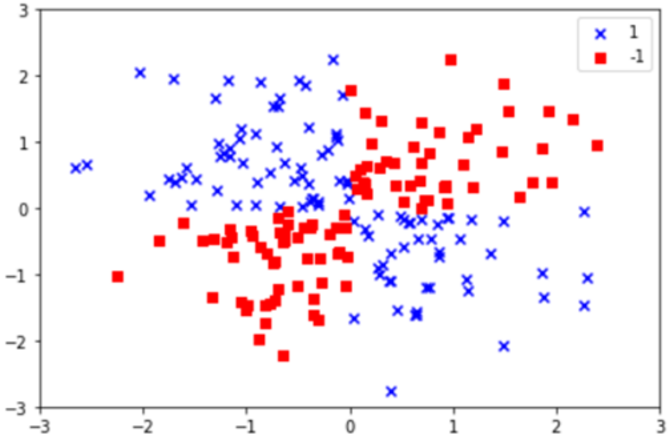
$$x_3 = \begin{bmatrix}1\\0\end{bmatrix} \in C_2 \quad x_4 = \begin{bmatrix}0\\1\end{bmatrix} \in C_2$$

$$x = \begin{bmatrix}a\\b\end{bmatrix} \xrightarrow{\text{二维变成五维}} \varphi(x) = \begin{bmatrix}a^2\\b^2\\a\\b\\ab\end{bmatrix}$$

$$\phi(x_1) = \begin{bmatrix}0\\0\\0\\0\\0\end{bmatrix} \in C_1 \quad \phi(x_2) = \begin{bmatrix}1\\1\\1\\1\\1\end{bmatrix} \in C_1$$

$$\phi(x_3) = \begin{bmatrix}1\\0\\1\\0\\0\end{bmatrix} \in C_2 \quad \phi(x_4) = \begin{bmatrix}0\\1\\0\\1\\0\end{bmatrix} \in C_2$$

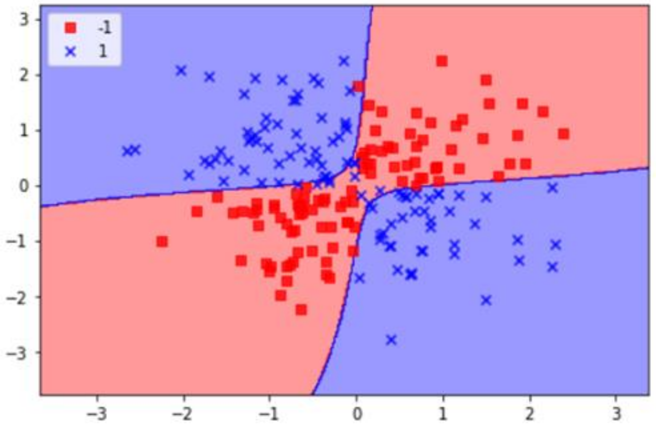
高维映射



二维异或数据集

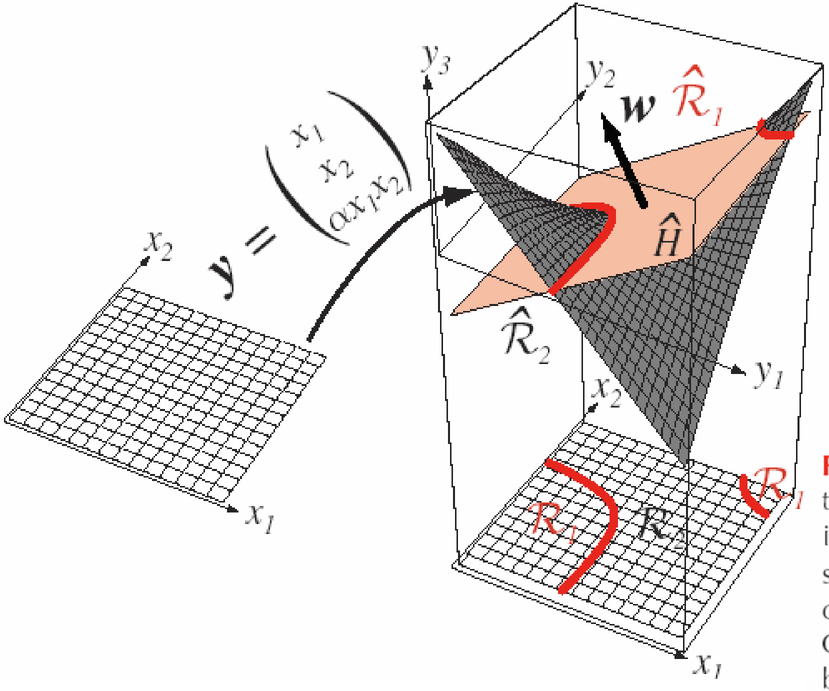
$\phi(x)$

高斯核或径向基函数核



高维空间中的异或数据

高维映射



线性分类器变为一个超平面，
把空间分为2个部分

FIGURE 5.6. The two-dimensional input space **x** is mapped through a polynomial function f to **y**. Here the mapping is $y_1 = x_1$, $y_2 = x_2$ and $y_3 \propto x_1 x_2$. A linear discriminant in this transformed space is a hyperplane, which cuts the surface. Points to the positive side of the hyperplane $\hat{H}$ correspond to category ω_1, and those beneath it correspond to category ω_2. Here, in terms of the **x** space, $\mathcal{R}_1$ is a not simply connected. From: Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*. Copyright © 2001 by John Wiley & Sons, Inc.

高维映射

关键思想

为了解决非线性分割问题，将 x_i变换到一个高维空间

- 输入空间： x_i 所在的空间
- 特征空间：变换后 $\phi(x_i)$ 的空间

如何变换？

- 利用一个适当的变换ϕ，使分类变得容易
- 特征空间中的线性算子等价于输入空间中的非线性算子

高维映射（核技巧）

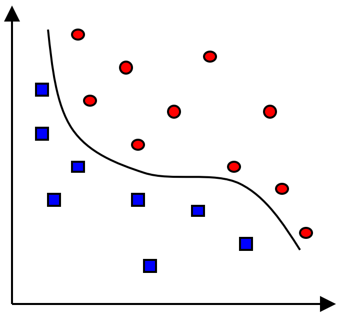
变换可能出现的问题

- 难以得到一个好的分类且计算开销大

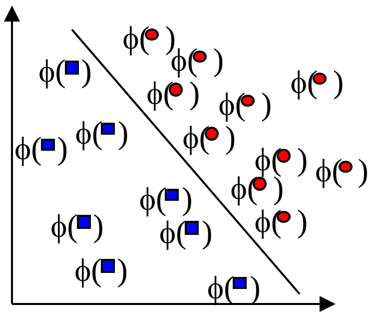
需要同时解决两个问题

- 最小化 $\|w\|^2$能得到好的分类
- 进行高效的计算（利用**核函数**技巧）

输入空间



$f(\cdot)$



特征空间

核函数

高维映射 $x \longrightarrow \phi(x)$

$$\begin{cases} y_i[w^{\mathrm{T}}\phi(x_i) + b] \geq 1 - \xi_i \\ \xi_i \geq 0 \end{cases}$$

$\phi(x)$是无限维，无法求整体优化

定义$\phi(x_1)$与$\phi(x_2)$两个无限维向量**内积**

⇨ **核函数（Kernel Function）**

$$K(x_1, x_2) = \phi(x_1)^T \phi(x_2)$$

只要知道一个核函数，不知道无限维映射 $\phi(x_i)$的显示表达，仍然可以求解最优

为什么可以通过求内积代替显示ϕ ----对偶问题！

*核函数

- 假设知道了K，如何求解$\phi(x_i)$，让优化问题可求解？
- 核函数也是有限制的，K要满足某种特定条件，才能拆成内积，$K(x_1,x_2)$能写成$\phi(x_1)^T\phi(x_2)$的充要条件：

（1） $K(x_1,x_2)=K(x_2,x_1)$ 交换性

（2） $\forall C_i,x_i,i=1\sim N$，有 半正定性

$$\sum_{i=1}^{N}\sum_{j=1}^{N}C_iC_jK(x_i,x_j)\geq 0$$

核函数举例

$$K(x_1, x_2) = \phi(x_1)^T \phi(x_2) \xrightarrow{\text{换符号}} K(x, y) = \phi(x)^T \phi(y)$$

- 定义核函数

$$K(\mathrm{x}, \mathrm{y}) = (1 + x_1 y_1 + x_2 y_2)^2$$

- 代入变换

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

$$\phi(\begin{bmatrix} y_1 \\ y_2 \end{bmatrix}) = (1, \sqrt{2}y_1, \sqrt{2}y_2, y_1^2, y_2^2, \sqrt{2}y_1 y_2)$$

$$\langle \phi(\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}), \phi(\begin{bmatrix} y_1 \\ y_2 \end{bmatrix}) \rangle = (1 + x_1 y_1 + x_2 y_2)^2 = K(\mathrm{x}, \mathrm{y}) \quad \text{内积}$$

- 内积可由 K计算，不必通过映射 $\phi(\bullet)$计算

常用核函数

常用核函数

名称	表达式	参数
线性核	$\kappa(\boldsymbol{x}_i, \boldsymbol{x}_j) = \boldsymbol{x}_i^\top \boldsymbol{x}_j$	
多项式核	$\kappa(\boldsymbol{x}_i, \boldsymbol{x}_j) = (\boldsymbol{x}_i^\top \boldsymbol{x}_j)^d$	$d \geq 1$为多项式的次数
高斯核	$\kappa(\boldsymbol{x}_i, \boldsymbol{x}_j) = \exp\left(-\frac{\lVert\boldsymbol{x}_i-\boldsymbol{x}_j\rVert^2}{2\delta^2}\right)$	$\delta > 0$为高斯核的带宽(width)
拉普拉斯核	$\kappa(\boldsymbol{x}_i, \boldsymbol{x}_j) = \exp\left(-\frac{\lVert\boldsymbol{x}_i-\boldsymbol{x}_j\rVert}{\delta}\right)$	$\delta > 0$
Sigmoid核	$\kappa(\boldsymbol{x}_i, \boldsymbol{x}_j) = \tanh(\beta\boldsymbol{x}_i^\top \boldsymbol{x}_j + \theta)$	tanh为双曲正切函数, $\beta > 0$, $\theta < 0$

- **基本经验：文本数据常用线性核，情况不明先尝试高斯核**

  程序实现时，可以调用相关的软件包

二阶多项式特征变换

二阶多项式函数：$\phi_2(x)$

$$\phi_2(x) = (1, x_1, x_2, \ldots, x_d, x_1^2, x_1x_2, \ldots, x_1x_d, x_2x_1, x_2^2, \ldots, x_2x_d, \ldots, x_d^2)$$

$$\begin{aligned}
\Phi_2(\mathbf{x})^T\Phi_2(\mathbf{x}') &= 1 + \sum_{i=1}^{d} x_i x_i' + \sum_{i=1}^{d}\sum_{j=1}^{d} x_i x_j x_i' x_j' \\
&= 1 + \sum_{i=1}^{d} x_i x_i' + \sum_{i=1}^{d} x_i x_i' \sum_{j=1}^{d} x_j x_j' \\
&= 1 + \mathbf{x}^T\mathbf{x}' + (\mathbf{x}^T\mathbf{x}')(\mathbf{x}^T\mathbf{x}')
\end{aligned}$$

- 计算复杂度为 $O(d)$，而非 $O(d^2)$

image: Hsuan-Tien Lin

广义二阶多项式核

$$\Phi_2(\mathbf{x}) = (1, x_1, \ldots, x_d, x_1^2, \ldots, x_d^2) \quad \Leftrightarrow \quad K_{\Phi_2}(\mathbf{x}, \mathbf{x}') = 1 + \mathbf{x}^T\mathbf{x}' + (\mathbf{x}^T\mathbf{x}')^2$$

$$\Phi_2(\mathbf{x}) = (1, \sqrt{2}x_1, \ldots, \sqrt{2}x_d, x_1^2, \ldots, x_d^2) \quad \Leftrightarrow \quad K_2(\mathbf{x}, \mathbf{x}') = 1 + 2\mathbf{x}^T\mathbf{x}' + (\mathbf{x}^T\mathbf{x}')^2$$

$$\Phi_2(\mathbf{x}) = (1, \sqrt{2\gamma}x_1, \ldots, \sqrt{2\gamma}x_d, \gamma x_1^2, \ldots, \gamma x_d^2)$$

$$\Leftrightarrow K_2(\mathbf{x}, \mathbf{x}') = 1 + 2\gamma\mathbf{x}^T\mathbf{x}' + \gamma^2(\mathbf{x}^T\mathbf{x}')^2$$

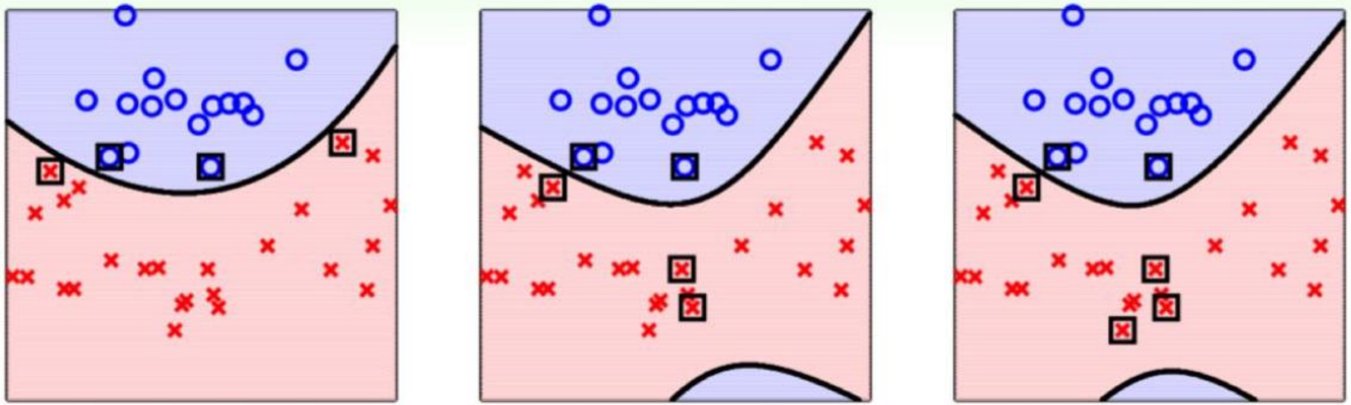
- 最常用的二阶核函数:

$$K_2(\mathbf{x}, \mathbf{x}') = (1 + \boxed{\gamma}\mathbf{x}^T\mathbf{x}')^2, \ \gamma > 0$$

- 不同的 γ 取值的对应相同维度的空间，但是不同的内积导致距离和边界都不相同

image: Hsuan-Tien Lin

广义二阶多项式核



$(1+0.001\mathbf{x}^T\mathbf{x}')^2$ $1+\mathbf{x}^T\mathbf{x}'+(\mathbf{x}^T\mathbf{x}')^2$ $(1+1000\mathbf{x}^T\mathbf{x}')^2$

- 不同的 γ 取值对应的支持向量和间隔不同
- γ 是需要选择的超参数

image: Hsuan-Tien Lin

广义多项式核

$$\begin{aligned} K_2(\mathbf{x}, \mathbf{x}') &= (\zeta + \gamma \mathbf{x}^T \mathbf{x}')^2 \text{ with } \gamma > 0, \zeta \geq 0 \\ K_3(\mathbf{x}, \mathbf{x}') &= (\zeta + \gamma \mathbf{x}^T \mathbf{x}')^3 \text{ with } \gamma > 0, \zeta \geq 0 \\ &\vdots \\ K_Q(\mathbf{x}, \mathbf{x}') &= (\zeta + \gamma \mathbf{x}^T \mathbf{x}')^Q \text{ with } \gamma > 0, \zeta \geq 0 \end{aligned}$$

- 多项式核支持向量机
- (γ, ζ, Q) 是需要选择的超参数

image: Hsuan-Tien Lin

高斯核

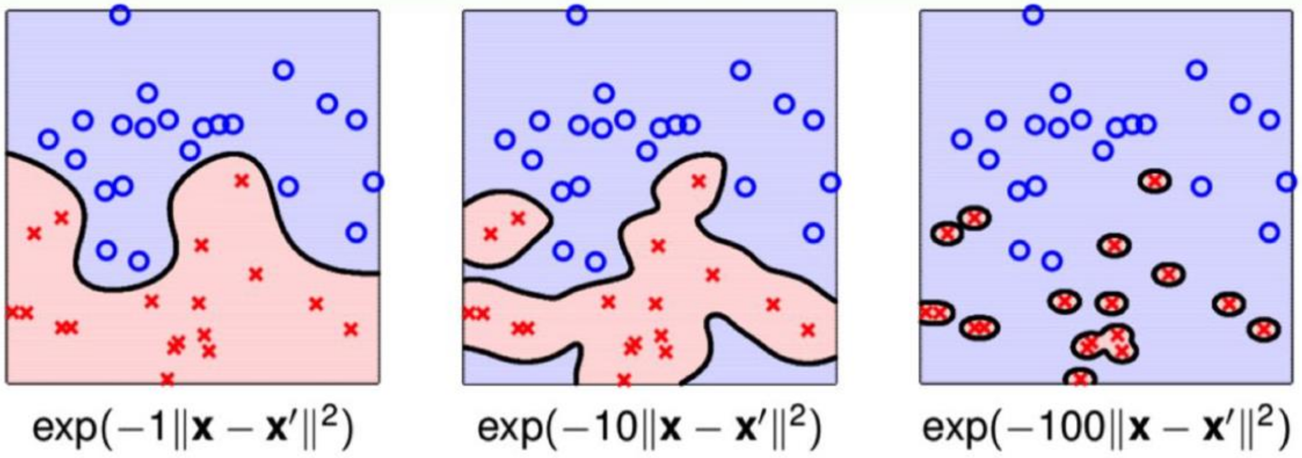
- 无限维的特征变换 $\Phi(\mathbf{x})$
- $K(x, x') = \exp(-(\mathrm{x}-\mathrm{x}')^2)$

$$
\begin{aligned}
\text{when } \mathbf{x} = (x),\ K(x, x') &= \exp(-(x-x')^2) \\
&= \exp(-(x)^2)\exp(-(x')^2)\exp(2xx') \\
&\overset{\text{Taylor}}{=} \exp(-(x)^2)\exp(-(x')^2)\left(\sum_{i=0}^{\infty}\frac{(2xx')^i}{i!}\right) \\
&= \sum_{i=0}^{\infty}\left(\exp(-(x)^2)\exp(-(x')^2)\sqrt{\frac{2^i}{i!}}\sqrt{\frac{2^i}{i!}}(x)^i(x')^i\right) \\
&= \boldsymbol{\Phi}(x)^T\boldsymbol{\Phi}(x')
\end{aligned}
$$

with infinite dimensional $\boldsymbol{\Phi}(x) = \exp(-x^2)\cdot\left(1, \sqrt{\frac{2}{1!}}x, \sqrt{\frac{2^2}{2!}}x^2, \ldots\right)$

- 一般形式： $K(x, x') = \exp(-\lambda(\mathrm{x}-\mathrm{x}')^2),\ \lambda > 0$

高斯核



- λ 越大，越容易过拟合
- 高斯核支持向量机
- λ 是需要选择的参数

image: Hsuan-Tien Lin

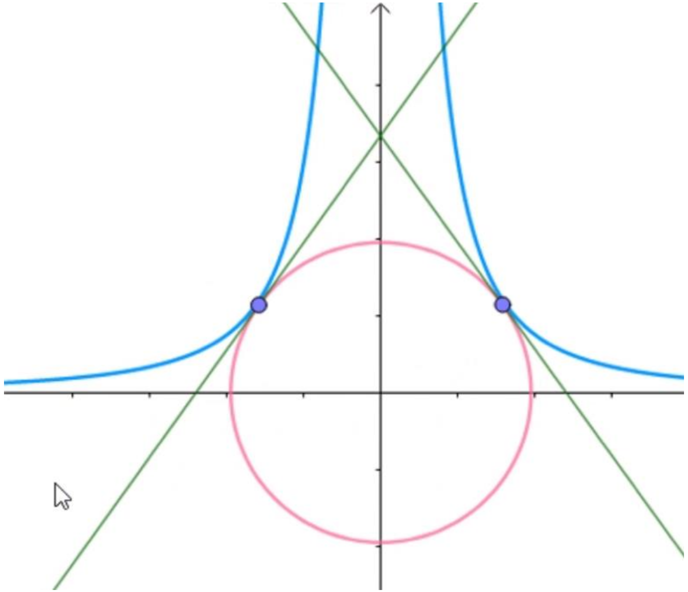
大纲

- 支持向量机基本概念
- 非线性问题
- 对偶问题
  - 定义
  - 求解
  - 三类对偶问题
  - 算法

求解带约束的最小值

简单的二维空间求解

- Problem 1. $\min f(x)$
- Problem 2. $\min f(x) \quad s.t.\, g(x) = 0$
- Problem 3. $\min f(x) \quad s.t.\, g(x) \leq 0$



$$\begin{cases} \min f(x,y) \\ g(x,y) = 0 \end{cases} \Leftrightarrow \begin{cases} \min(x^2 + y^2) \\ x^2 y = 3 \end{cases}$$

$$\Downarrow$$

$$\begin{cases} \nabla f = \lambda \nabla g \\ g(x,y) = 0 \end{cases} \Leftrightarrow \begin{cases} \begin{pmatrix} 2x \\ 2y \end{pmatrix} = \lambda \begin{pmatrix} 2xy \\ x^2 \end{pmatrix} \\ x^2 y = 3 \end{cases}$$

对偶问题—普适定义

原问题（prime problem）

$$\min f(w)$$

$$s.t.\ \ g_i(w) \le 0(i \sim K)$$
$$h_i(w) = 0(i \sim M)$$

引入**拉格朗日乘子** ⇒

定义：$L(w, \alpha, \beta)$

$$= f(w) + \sum_{i=1}^{K} \alpha_i g_i(w) + \sum_{i=1}^{M} \beta_i h_i(w)$$

$$= f(w) + \alpha^T g(w) + \beta^T h(w)$$

$$\begin{bmatrix} g_1(w) \\ g_2(w) \\ \dots \\ g_K(w) \end{bmatrix} \quad \begin{bmatrix} h_1(w) \\ h_2(w) \\ \dots \\ h_M(w) \end{bmatrix}$$

凸优化问题

目标函数$f(w)$与约束函数$g_i(w)$为连续可微凸函数
$h_i(w)$为仿射函数（即ax+b的形式）

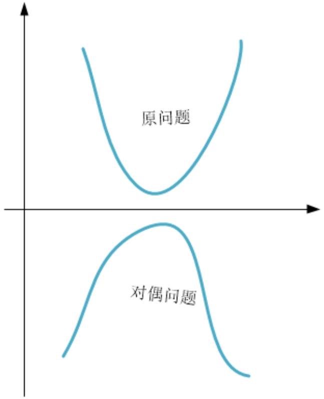
对偶问题—普适定义

- **对偶问题定义**　（Dual Problem）

最大化$\theta(\alpha,\beta)=\underline{inf}\{L(w,\alpha,\beta)\}$

求最小值：所有w，求最小值

限制条件：　$\alpha \succcurlyeq 0$　(即　$\alpha_i \geq 0$,　$i=1\sim K$)



限定α和β的情况下，遍历所有w求L的最小值，**每确定一个α和β都能算出L的最小值，在外面再针对所有的α和β求L的最大值**

θ：所有下界里的最大值；比较大的更接近与原问题的解

*原问题与对偶问题

定理：如果w^* 是原问题的解，而α^*和β^*是对偶问题解，则有 $f(w^*) \geq \theta(\alpha^*, \beta^*)$

证明：$\theta(\alpha^*, \beta^*) = \inf\{L(w, \alpha^*, \beta^*)\} \leq L(w^*, \alpha^*, \beta^*)$

固定α^*, β^*所有w,求最小

w^*是原问题的解，带入L后，
L(w, α^*, β^*)一定小于某个特定的w^*

$$= f(w^*) + \sum_{i=1}^{K} \alpha_i^* g_i(w^*) + \sum_{i=1}^{M} \beta_i^* h_i(w^*)$$

$\boxed{\leq 0}$

$$\leq f(w^*)$$

*强对偶定理（KKT条件）

定义： $G = f(w^*) - \theta(\alpha^*, \beta^*) \geq 0$为原问题与对偶问题的间距 （Duality Gap）

强对偶定理：若$f(w)$为凸函数，且$g(w) = Aw + b$, $h(w) = Cw + d$，则原问题与对偶问题间距为0，即

$$\boldsymbol{f(w^*) = \theta(\alpha^*, \beta^*)}$$

对$\forall i = 1 \sim K$，$\alpha_i^* = 0$，或$g_i^*(w^*) = 0$

（KKT条件，KKT是三个人的名字）

对于某些特定优化问题，可以证明G=0，即原问题与对偶问题相等

原始问题比较难以求解，通过构建其对偶问题，解决这个对偶问题得到其**原问题的下界**（在**弱对偶**情况下，对于最小化问题来说），或者得到原问题的解（**强对偶**情况下）。

《凸优化》

对偶问题的一些理解

- 无论原问题难度如何，对偶问题都是凸问题，凸问题是一类比较容易求解问题，当原问题是一个特别难的问题时，化简为对偶问题，相对容易求解；
- 转化为对偶问题后，引入的拉格朗日乘子数量是所有约束个数的总和，当原问题中变量个数，远小于约束个数时，就不要转化为对偶问题求解了。
- 对偶问题是凸问题，相对容易求解。我们要知道对偶问题的解只是原问题的一个下界，只有当对偶间隙为0(强对偶)时，对偶问题的最优值才和原问题的最优值相等。

支持向量机的对偶问题

■ 优化问题

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

$$s.t. \quad y_i(w^T \underline{x_i} + b) \geq 1$$

或$\phi(x_i), i = 1\sim N$

引入非线性变换时，特征维度 d 会很大，计算复杂度高
方案：把 d + 1 个变量、n 个约束的优化问题转化为维度为 n 个变量、n + 1 个约束的优化问题

■ 支持向量机中引入对偶问题的优点

➢ **对偶问题容易求解**

➢ **自然引入核函数**

原问题与对偶问题

原问题

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

$$s.t.\ y_i(w^T x_i + b) \geq 1$$

$$i = 1, \dots, N$$

拉格朗日函数

$$L(w,b,\alpha)$$

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

为每个约束条件引入拉格朗日乘子

$$\alpha = (\alpha_1, \alpha_2, \dots, \alpha_N)^T$$

KKT条件： α_i=0或者$y_i(w^T x_i + b) - 1 = 0$

根据拉格朗日对偶性，**原问题的对偶问题是极大极小问题：**

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

求$L(w,b,\alpha)$对w和b的极小，再对α求极大

求解

（1）求 $\min_{w,b} L(w,b,\alpha)$

将拉格朗日函数$L(w,b,\alpha)$分别对w,b求偏导，令其等于0

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

(x_i, y_i)是样本

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

得

$$w = \sum_{i=1}^{N} \alpha_i y_i x_i \qquad \sum_{i=1}^{N} \alpha_i y_i = 0$$

推导过程

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

代入拉格朗日函数，可得

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

求解

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

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

$$s.t.\ \sum_{i=1}^{N}\alpha_i y_i=0 \qquad \alpha_i\geq 0,\quad i=1,2,\dots,N$$

将求极大转换为求极小，得到与之等价的对偶最优化问题

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

可引入内积由核函数替代
$K(x_1,x_2)=\phi^T(x_1)\phi(x_2)$

$$s.t.\ \sum_{i=1}^{N}\alpha_i y_i=0 \qquad \alpha_i\geq 0,\quad i=1,2,\dots,N$$

求解α_i：二次规划问题

求解说明

- α_i是拉格朗日乘子，对应训练样本 (x_i, y_i).
- 根据KKT条件：对任意训练样本(x_i, y_i)， 总有α_i=0或 $y_i(w^T x_i + b) = 1$
- 若α_i=0，则该样本将不会在目前函数与属于条件的求和中出现，即不会对判别有任何影响；
- 若$\alpha_i > 0$, 则必有$y_i(w^T x_i + b) = 1$，所对应的样本点位于最大间隔边界上，是一个**支持向量**。
- **支持向量机的一个重要性质：训练完成后，大部分的训练样本都不需保留，最终模型仅与支持向量有关。**

软间隔支持向量机的对偶问题

原问题：

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

$$s.t. \quad y_i(\|w\|^2 + b) \geq 1 - \xi_i, \quad \xi_i \geq 0,\ i = 1, \ldots, n$$

对偶问题：

$$\max_{\text{all } \alpha_i \geq 0, \beta_i \geq 0} \left(\min_{b,\mathbf{w},\xi} \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n}\xi_i + \sum_{i=1}^{n}\alpha_i(1 - \xi_i - y_i(\mathbf{w}^T\mathbf{x}_i + b)) + \sum_{i=1}^{n}\beta_i(-\xi_i) \right)$$

把和 ξ_i 有关的项整理在一起：

$$\max_{\text{all } \alpha_i \geq 0, \beta_i \geq 0} \left(\min_{b,\mathbf{w},\xi} \frac{1}{2}\|w\|^2 + \sum_{i=1}^{n}\alpha_i(1 - y_i(\mathbf{w}^T\mathbf{x}_i + b)) + \sum_{i=1}^{n}(C - \alpha_i - \beta_i)\xi_i \right)$$

令 $\mathcal{L}$ 对 ξ_i 的偏导为 0，可得 $\frac{\partial \mathcal{L}}{\partial \xi_i} = C - \alpha_i - \beta_i = 0$，代入上式：

$$\max_{\text{all } 0 \leq \alpha_i \leq C} \left(\min_{b,\mathbf{w}} \frac{1}{2}\|w\|^2 + \sum_{i=1}^{n}\alpha_i(1 - y_i(\mathbf{w}^T\mathbf{x}_i + b)) \right)$$

软间隔支持向量机的对偶问题

优化目标：

$$\max_{\text{all } 0\leq\alpha_i\leq C}\left(\min_{b,\mathbf{w}} \frac{1}{2}\|w\|^2+\sum_{i=1}^{n}\alpha_i(1-y_i(\mathbf{w}^T\mathbf{x}_i+b))\right)$$

令 $\mathcal{L}(b,\mathbf{w},\boldsymbol{\alpha})$ 对 $\mathbf{w}$ 和 b 的偏导为 0，可得

$$\mathbf{w}=\sum_{i=1}^{n}\alpha_i y_i \mathbf{x}_i,\quad \sum_{i=1}^{n}\alpha_i y_i=0$$

代入优化目标，可得

$$\min_{\boldsymbol{\alpha}}\quad \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j \mathbf{x}_i^T\mathbf{x}_j-\sum_{i=1}^{n}\alpha_i$$

$$s.t.\quad \sum_{i=1}^{n}\alpha_i y_i=0,\qquad 0\leq\alpha_{\mathrm{i}}\leq C,\qquad i=1,\dots,n$$

核函数支持向量机的对偶问题

- 用核函数$K\left(x_i, x_j\right) = \phi^T(x_i)\phi(x_j)$代替对偶问题目标函数中的内积$x_i \cdot x_j$:

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

- 分类函数中的内积也用核函数代替，决策函数为:

$$f(x) = sign(\sum_{i=1}^{N}\alpha_i^* y_i \phi(x_i)\cdot\phi(x) + b_i)$$

$$=sign(\sum_{i=1}^{N}\alpha_i^* y_i K(x, x_i) + b_i)$$

将原来输入空间变换到一个新的特征空间，在新的特征空间里训练样本中学习线性支持向量机。

线性可分支持向量机学习算法

（1）给定线性可分的数据集，首先求解对偶问题的解α^*

（2）计算

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

选一个符合约束条件的α_j^*计算 $b^* = y_i - \sum_{i=1}^{N} \alpha_i^* y_i (x_i \cdot x_j)$

得到原问题的解w^*和b^*

（3）得到分离超平面和分类决策函数

线性支持向量机学习算法

（1）选择惩罚参数C>0，求解对偶问题的解α^*

（2）计算 $$w^* = \sum_{i=1}^{N} \alpha_i^* y_i x_i$$

选一个符合约束条件的α_j^*计算 $$b^* = y_i - \sum_{i=1}^{N} \alpha_i^* y_i (x_i \cdot x_j)$$

得到原问题的解w^*和b^*

（3）得到分离超平面和分类决策函数

■ 非线性支持向量机学习算法

（1）选取适当的核函数K(x1,x2)和适当的C，求解对偶问题的解α^*

（2）选一个符合约束条件的α_j^*计算

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

（3）得到分类决策函数： $f(x) = sign(\sum_{i=1}^{N} \alpha_i^* y_i K(x, x_i) + b_i)$

例子

- 正例点：$x_1 = (3,3)^{\mathrm{T}}, y_1 = 1$； $x_2 = (4,3)^{\mathrm{T}}$, $y_2 = 1$,
- 负例点：$x_3 = (1,1)^{\mathrm{T}}, y_3 = -1$
- 解：对偶形式

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

$$= \frac{1}{2}(\underline{18\alpha_1^2} + 25\alpha_2^2 + 2\alpha_3^2 + 42\alpha_1\alpha_2 - 12\alpha_1\alpha_3 - 14\alpha_2\alpha_3) - \alpha_1 - \alpha_2 - \alpha_3$$

$$\boxed{1 * 1 * (3 \quad 3)\begin{pmatrix}3\\3\end{pmatrix}}$$

$$\text{s.t.} \quad \alpha_1 + \alpha_2 - \alpha_3 = 0$$

$$\alpha_i \geqslant 0, \quad i = 1,2,3$$

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

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

- 对 α_1,α_2 求偏导数，并令其为0，易知 $s(\alpha_1,\alpha_2)$ 在 $\left(\frac{3}{2},-1\right)^{\mathrm{T}}$ 取极值，但该点不满足约束条件 $\alpha_2 \geqslant 0$ ，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 $s(\alpha_1,\alpha_2)$ $\alpha_1=\frac{1}{4},\alpha_2=0$ 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

- 计算：$\alpha_1, \alpha_2, \alpha_3$带入 $w^* = \sum_{i=1}^{N} \alpha_i^* y_i x_i$ 得到 $w_1^* = w_2^* = \frac{1}{2}$

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

$$b^* = -2$$

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

求解α_i的方法

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

可引入内积由核函数替代
$K(x_1, x_2) = \phi^T(x_1)\phi(x_2)$

$$s.t. \sum_{i=1}^{N}\alpha_i y_i = 0 \qquad \alpha_i \geq 0, \quad i = 1,2,\dots,N$$

求解α_i：二次规划问题

- 凸二次优化问题具有全局最优解
- 样本容量较大时，非常低效
- 如何高效实现SVM的学习？

*SMO算法

- 1998年Platt提出**启发式SMO算法**，求解对偶问题
  - 变量为拉格朗日乘子，一个变量α_i对应一个样本（x_i, y_i）
  - 在每次迭代中，选择两个α值（违反KKT条件最多的一对）进行优化，其余参数都视为常数，从而问题就变成了类似于二次方程求最大值的问题

对已有的模型来说，及其对应样本的 KKT 条件为：

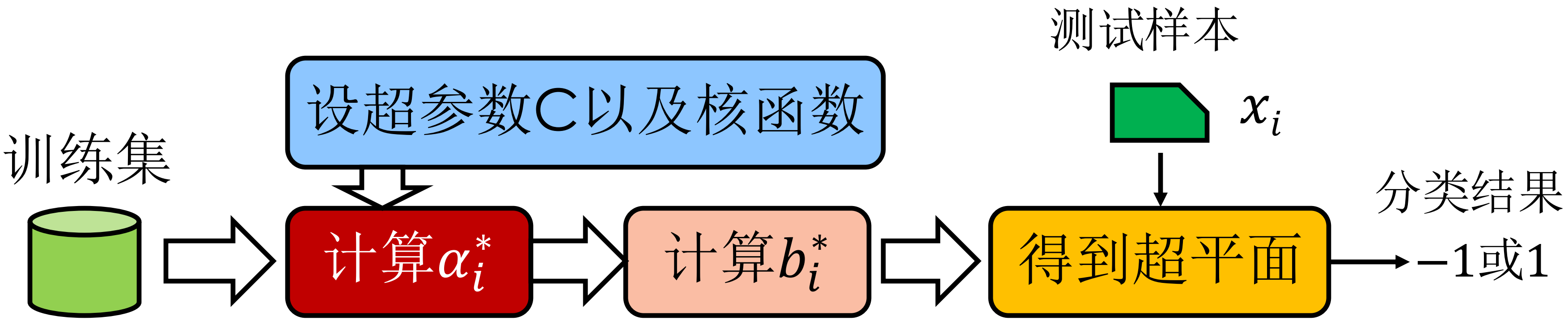
$$\alpha_i = 0 \Leftrightarrow y_i f(x_i) > 1$$
$$0 < \alpha_i < C \Leftrightarrow y_i f(x_i) = 1$$
$$\alpha_i = C \Leftrightarrow y_i f(x_i) < 1$$

- $\alpha_i = 0 \Leftrightarrow$**样本离间隔超平面**比较远
- $0 < \alpha_i < C \Leftrightarrow$**样本落在间隔超平面**上
- $\alpha_i = C \Leftrightarrow$**样本在间隔超平面**以内

SMO算法通过将优化问题分解为一系列较小的优化问题来解决SVM的优化问题，每个子问题只涉及两个α值。由于SMO算法的高效性和易于实现性，它已成为SVM中最受欢迎的训练算法之一。

总结1

- 支持向量机由简至繁的模型：
  - ➢线性可分支持向量机----间隔最大
  - ➢线性支持向量机----软间隔
  - ➢非线性支持向量机----核函数



总结2

- SVM 的基本思想是，通过寻找最优的“**分离超平面**”(separating hyperplane)，将两类数据分离开。
- SVM 特别适用于变量很多的数据，因为在高维空间，数据被“打散”，故更容易用超平面进行分离。
- SVM 在变量较多的数据中有很多成功的应用，比如文本分析与图像识别。例如，SVM 曾在手写数字识别的 MNIST 数据集取得巨大成功。

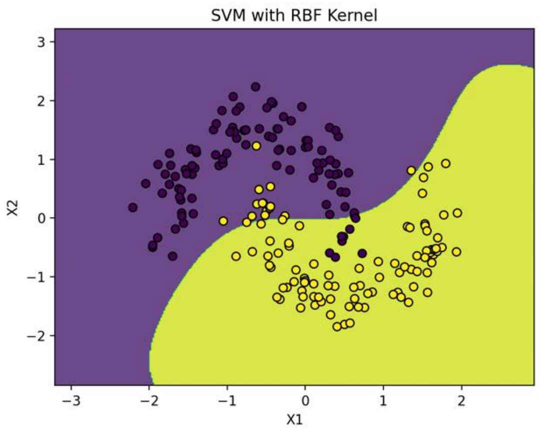
来自AI的例子

请建立一个非线性数据集，并且用SVM进行分类

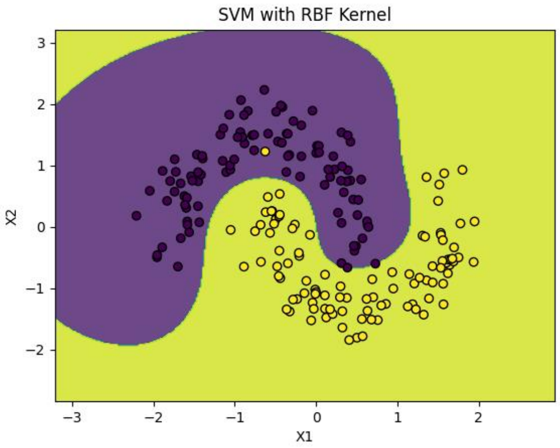
这里是一个使用Python和Scikit-learn库生成非线性数据集并用SVM进行分类的例子。我们将生成一个包含两个类别的非线性数据集，并使用RBF核函数的SVM对数据进行分类。

python　　Copy code

```python
import numpy as np
import matplotlib.pyplot as plt
from sklearn import datasets
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.svm import SVC

# 生成非线性数据集
X, y = datasets.make_moons(n_samples=200, noise=0.15, random_state=42)

# 将数据集分为训练集和测试集
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, ran

# 数据预处理：标准化
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
```



C=0.1



C=10

讨论

- SVM与逻辑回归的区别
- SVM与Fisher判别法的区别