支持向量机(SVM)

主要内容

- 理解支持向量机SVM的原理和目标
- 掌握支持向量机的计算过程和算法步骤
- 理解软间隔最大化的含义
- 线性不可分时，加入核函数的方法
- 了解SMO算法的过程

支持向量机概述

支持向量机概述| 基本原理

预测真实风险=经验风险+置信风险

经验风险最小化原则只在占很小比例的样本上做到误差很小，不能保证在更大比例的真实数据上也没有误差，泛化能力难以保证。

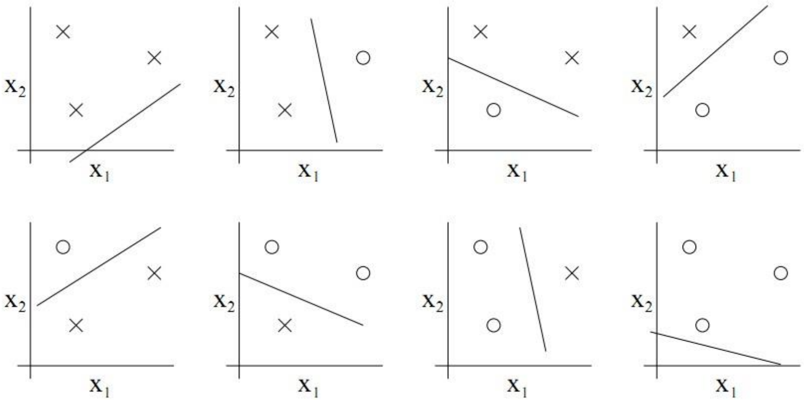
置信风险，代表了在多大程度上可以信任分类器在未知数据上分类的结果。置信风险无法精确度量，只能计算一个上界。
置信风险与两个量有关：一是样本量，给定的样本数量越大，学习结果的置信风险越小；二是分类函数的VC维，VC维越大，推广能力越差，置信风险会变大。

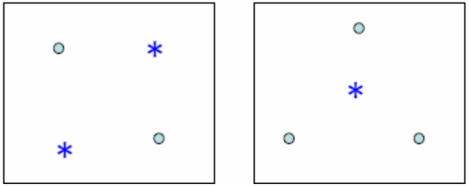
真实风险：$R(w) \leq R_{emp}(w)+\Phi(n/h)$
公式中$R(w)$就是真实风险，$R_{emp}(w)$就是经验风险，$\Phi(n/h)$就是置信风险。

支持向量机概述|基本原理-VC维

VC维的直观定义是：对一个指标函数集，如果存在k个样本能够被函数集中的函数按所有可能的2^k种形式分开，则称函数集能够把k个样本打散；函数集的VC维就是它能打散的最大样本数目k。

VC维越高，其复杂程度越高，因此其作为函数复杂度的度量。比如直线的VC维为3。





平面上任何一条直线
都不能正确划分

支持向量机概述|统计学习理论-基本思想

支持向量机通过使用结构风险最小化原理在属性空间构建最优分类超平面，使得分类器得到全局最优，并在整个样本空间的期望风险以某个概率满足一定上界。

支持向量机的基本思想：

1、在线性可分情况下，在原空间寻找两类样本的最优分类超平面；

2、在线性不可分的情况下，通过使用核函数将低维输入空间的样本映射到高维属性空间，从而使得在高维属性空间采用线性算法对样本的非线性进行分析成为可能，并在该特征空间中寻找最优分类超平面。

3、考虑到随机误差与边界上的线性不可分，加入了松弛变量进行分析，使得分割平面变得光滑，提高泛化能力。

优点：

1、基于统计学习理论中结构风险最小化原则和VC维理论，具有良好的泛化能力，即由有限的训练样本得到的小的误差能够保证使独立的测试集仍保持小的误差。

2、支持向量机的求解问题对应的是一个凸优化问题，因此局部最优解一定是全局最优解。

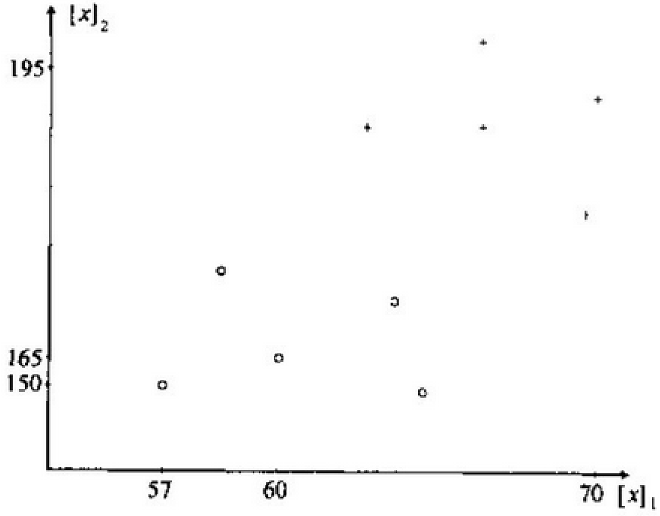
3、核函数的成功应用，将非线性问题转化为线性问题求解。

4、分类间隔的最大化，使得支持向量机算法具有较好的鲁棒性。

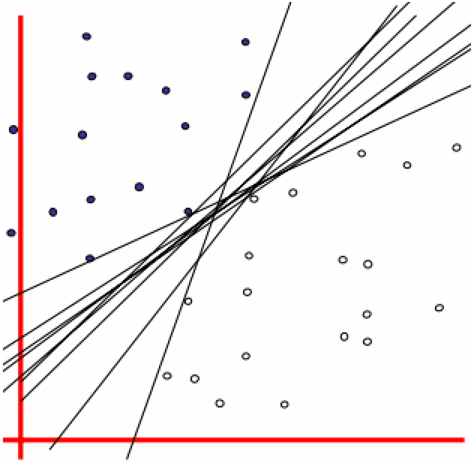
支持向量机详述

支持向量机详述| 统计学习理论-结构风险最小化原则

病人编号	年龄 $[x]_1$	胆固醇水平 $[x]_2$	有否心脏病 y
1	$[x]_1 = 60$	$[x]_2 = 165$	$y = -1$
2	$[x]_1 = 57$	$[x]_2 = 150$	$y = -1$
$\vdots$	$\vdots$	$\vdots$	$\vdots$
10	$[x]_1 = 70$	$[x]_2 = 190$	$y = 1$



线性可分支持向量机



- 给定线性可分训练数据集，通过间隔最大化得到的分离超平面为

$$w^{*T}\cdot\Phi(x)+b^{*}=0$$

相应的分类决策函数 $f(x)=sign\left(w^{*T}\cdot\Phi(x)+b^{*}\right)$

该决策函数称为线性可分支持向量机。

- φ(x)是某个确定的特征空间转换函数，它的作用是将x映射到(更高的)维度。

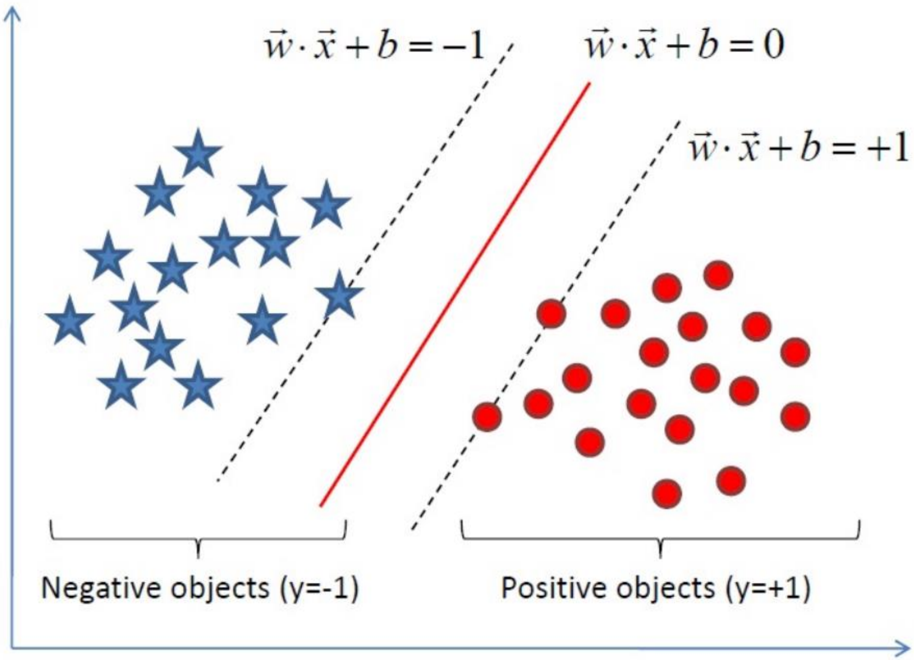
最简单直接的：$\Phi(x)=x$

- 稍后会看到，求解分离超平面问题可以等价为求解相应的凸二次规划问题。

整理符号

- 分割平面： $w^{*T}\cdot\Phi\left(x\right)+b^{*}=0$
- 训练集： $x_1,x_2,\cdots,x_n$
- 目标值： $y_1,y_2,\cdots,y_n,\quad y_i\in\left\{-1,1\right\}$
- 新数据的分类：$f(x)=sign(w^{*T}\cdot\Phi(x)+b^{*})$

线性可分支持向量机



$\vec{w} \cdot \vec{x} + b = -1$
$\vec{w} \cdot \vec{x} + b = 0$
$\vec{w} \cdot \vec{x} + b = +1$
Negative objects (y=-1)
Positive objects (y=+1)

推导目标函数

令$\Phi(x)=x$

- 函数距离：每个点到超平面的距离为$|w^T\cdot x+b|$，等价于$y_i(w^T\cdot x+b)$
- 几何距离：由于只要成比例变化w和b就会造成函数距离的增大，而超平面并不改变，因此使用法向量w进行规范化

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

- **定义超平面（w,b）关于训练数据集T的几何间超平面关于T中所有样本点（x_i,y_i）的几何间隔的最小值：**

$$\min\left[y_i\left(\frac{w^T}{\|w\|}\cdot x_i+\frac{b}{\|w\|}\right)\right]$$

最大间隔分离超平面

- 目标函数： $$\max_{w,b}\left\{\frac{1}{\|w\|}\min_i\left[y_i\cdot\left(w^T\cdot x_i+b\right)\right]\right\}$$
- 约束条件：任意点的函数距离应该大于等于支持向量上的点。
$$s.t.\quad y_i\left(w^T\cdot x_i+b\right)\geq\min_i\left[y_i\cdot\left(w^T\cdot x_i+b\right)\right],\quad i=1,2,\cdots,n$$
- 此处可令 $$\min_i\left[y_i\cdot\left(w^T\cdot x_i+b\right)\right]=1$$
- 则目标函数为： $$\max_{w,b}\frac{1}{\|w\|}$$

$$s.t.\quad y_i\left(w\cdot x_i+b\right)\geq 1,\quad i=1,2,\cdots,n$$

建立目标函数

$$\max_{w,b} \frac{1}{\|w\|}$$

$$s.t. \quad y_i(w \cdot x_i + b) \ge 1, \quad i = 1, 2, \cdots, n$$

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

$$s.t. \quad y_i(w \cdot x_i + b) \ge 1, \quad i = 1, 2, \cdots, n$$

拉格朗日乘子法

$$\min_{w,b} \frac{1}{2}\|w\|^2, \; s.t. \; y_i\left(w^T \cdot \Phi(x_i)+b\right) \geq 1, \; i=1,2\cdots N$$

$$L(w,b,\alpha)=\frac{1}{2}\|w\|^2-\sum_{i=1}^{n}\alpha_i\left(y_i\left(w^T \cdot \Phi(x_i)+b\right)-1\right)$$

- 原问题是极小极大问题

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

- 原始问题的对偶问题，是极大极小问题

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

拉格朗日函数

- 将拉格朗日函数L(**w**,b,**a**)分别对**w**，b求偏导并令其为0：

$$\frac{\partial L}{\partial w} = 0 \Rightarrow w = \sum_{i=1}^{n} \alpha_i y_i \Phi(x_n)$$

$$\frac{\partial L}{\partial b} = 0 \Rightarrow 0 = \sum_{i=1}^{n} \alpha_i y_i$$

计算拉格朗日函数的对偶函数

$$w=\sum_{i=1}^{n}\alpha_i y_i \Phi(x_n)$$

$$0=\sum_{i=1}^{n}\alpha_i y_i$$

$$L(w,b,\alpha)=\frac{1}{2}\|w\|^2-\sum_{i=1}^{n}\alpha_i\left(y_i\left(w^T\cdot\Phi(x_i)+b\right)-1\right)$$

$$=\frac{1}{2}w^Tw-w^T\sum_{i=1}^{n}\alpha_i y_i\Phi(x_i)-b\sum_{i=1}^{n}\alpha_i y_i+\sum_{i=1}^{n}\alpha_i$$

$$=\frac{1}{2}w^T\sum_{i=1}^{n}\alpha_i y_i\Phi(x_i)-w^T\sum_{i=1}^{n}\alpha_i y_i\Phi(x_i)-b\cdot 0+\sum_{i=1}^{n}\alpha_i$$

$$=\sum_{i=1}^{n}\alpha_i-\frac{1}{2}\left(\sum_{i=1}^{n}\alpha_i y_i\Phi(x_i)\right)^T\sum_{i=1}^{n}\alpha_i y_i\Phi(x_i)$$

$$=\sum_{i=1}^{n}\alpha_i-\frac{1}{2}\sum_{i,j=1}^{n}\alpha_i\alpha_j y_i y_j\Phi^T(x_i)\Phi(x_j)$$

$$a^*=\arg\max_{\alpha}\left(\sum_{i=1}^{n}\alpha_i-\frac{1}{2}\sum_{i,j=1}^{n}\alpha_i\alpha_j y_i y_j\Phi^T(x_i)\Phi(x_j)\right)$$

继续求$\min_{w,b} L(w,b,\alpha)$对α的极大

$$\max_{\alpha} \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j \left(\Phi(x_i) \cdot \Phi(x_j) \right)$$

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

$$\alpha_i \geq 0, \quad i = 1, 2, \ldots, n$$

整理目标函数：添加负号

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

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

$$\alpha_i \geq 0, \quad i = 1,2,\ldots,n$$

线性可分支持向量机学习算法——思路整理

- 构造并求解约束最优化问题

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

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

$$\alpha_i \geq 0, \quad i = 1,2,\ldots,n$$

- 求得最优解α*

线性可分支持向量机学习算法

- 计算

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

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

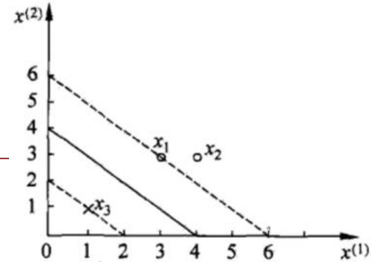
- 求得分离超平面

$$w^* \Phi(x) + b^* = 0$$

- 分类决策函数

$$f(x) = sign\left(w^* \Phi(x) + b^*\right)$$

举例



- 给定3个数据点：正例点$x_1=(3,3)^T$，$x_2=(4,3)^T$，负例点$x_3=(1,1)^T$，求线性可分支持向量机。
- 目标函数：

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

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

$$s.t.\ \ \alpha_1 + \alpha_2 - \alpha_3 = 0$$

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

将约束带入目标函数，化简计算

- 将 $\alpha_1+\alpha_2=\alpha_3$
- 带入目标函数，得到关于α_1，α_2的函数：

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

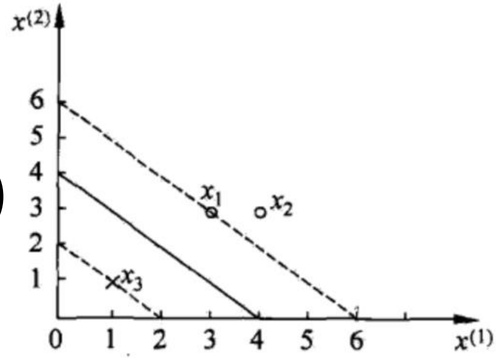
- 对α_1，α_2求偏导并令其为0，易知$s(\alpha_1,\alpha_2)$在点(1.5,-1)处取极值。而该点不满足条件$\alpha_2\geq 0$，所以，最小值在边界上达到。
- 当α1=0时，最小值s(0,2/13)=-2/13=-0.1538
- 当α2=0时，最小值s(1/4,0)=-1/4=-0.25
- 于是，$s(\alpha_1,\alpha_2)$在α_1=1/4， α_2=0时达到最小，此时，$\alpha_3=\alpha_1+\alpha_2$=1/4

分离超平面

- $\alpha_1=\alpha_3=1/4$对应的点x_1,x_3是支持向量。
- 带入公式：

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

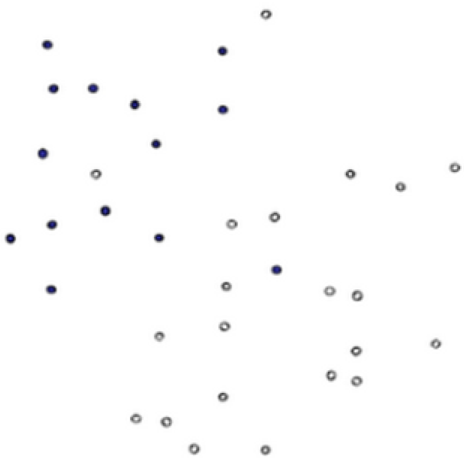
$$b^* = y_i - \sum_{i=1}^{N} \alpha_i^* y_i \left(\Phi(x_i)\cdot\Phi(x_j)\right)$$

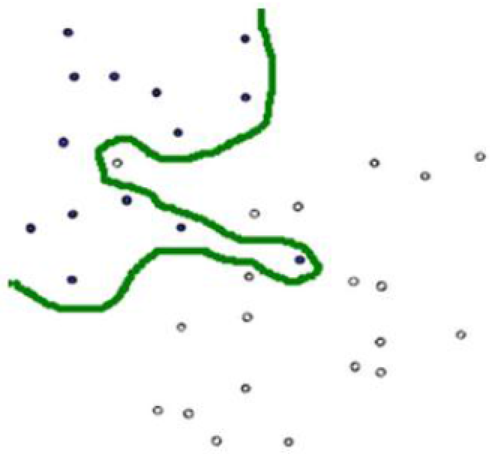


- 得到$w_1=w_2=0.5$，b=-2
- 因此，分离超平面为 $\frac{1}{2}x_1+\frac{1}{2}x_2-2=0$
- 分离决策函数为 $f(x)=sign\left(\frac{1}{2}x_1+\frac{1}{2}x_2-2\right)$

线性不可分的情形—线性支持向量机

- 大部分情况都不是线性可分的
- 线性不可分时无法使用前述数学技巧
- 也可以使用加惩罚函数的方法解决：
  http://www.cnblogs.com/LeftNotEasy/archive/2011/05/18/2034566.html





线性支持向量机

- 若数据线性不可分，则增加松弛因子$\xi_i \geq 0$，使函数间隔加上松弛变量大于等于1。这样，约束条件变成

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

- 目标函数：

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

线性SVM的目标函数

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

$$s.t. \quad y_i\left(w\cdot x_i + b\right) \geq 1-\xi_i, \quad i=1,2,\cdots,n$$

$$\xi_i \geq 0, \quad i=1,2,\cdots,n$$

带松弛因子的SVM拉格朗日函数

- 拉格朗日函数

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

- 对w,b, ξ求偏导

$$\frac{\partial L}{\partial w}=0\Rightarrow w=\sum_{i=1}^{n}\alpha_i y_i\phi(x_n)$$

$$\frac{\partial L}{\partial b}=0\Rightarrow 0=\sum_{i=1}^{n}\alpha_i y_i$$

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

带入目标函数

- 将三式带入L中，得到

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

- 对上式求关于α的极大，得到：

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

$$\begin{aligned} s.t. \; & \sum_{i=1}^{n}\alpha_i y_i = 0 \\ & C - \alpha_i - \mu_i = 0 \\ & \alpha_i \geq 0 \\ & \mu_i \geq 0, \quad i = 1,2,\ldots,n \end{aligned}$$

$$0 \leq \alpha_i \leq C$$

最终的目标函数

- 整理，得到对偶问题：

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

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

$$0\le\alpha_i\le C,\qquad i=1,2,\ldots,n$$

线性支持向量机学习算法

- 构造并求解约束最优化问题

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

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

$$0 \le \alpha_i \le C, \quad i = 1,2,\ldots,n$$

- 求得最优解α*

线性支持向量机学习算法

- 计算

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

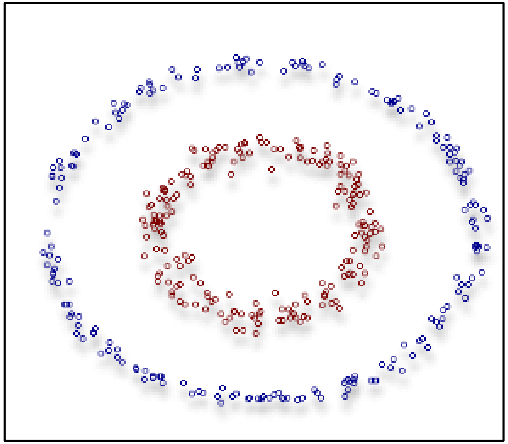
$$b^* = \frac{\max\limits_{i:\ y_i=-1} w^* \cdot x_i + \min\limits_{i:\ y_i=1} w^* \cdot x_i}{2}$$

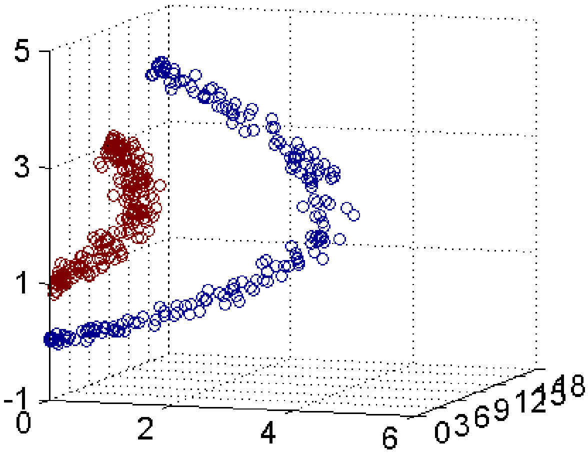
  - 注意：计算b*时，需要使用满足条件$0<\alpha_j<C$的向量
  - 实践中往往取支持向量的所有值取平均，作为b*
- 求得分离超平面 $$w^{*T} x + b^* = 0$$
- 分类决策函数

$$f(x) = sign\left(w^{*T} x + b^*\right)$$

非线性可分——映射至高维空间

转换方式： $Z_1=X_1^2$ $Z_2=X_2^2$ $Z_3=X_3^2$





图片摘自：http://www.chinakdd.com/article-W82k0g2822JE712.html

引入核函数

- 参考：http://www.cnblogs.com/jerrylead/archive/2011/03/18/1988406.html
- 假设n=3，见下算例
- 这时发现我们可以只计算原始样本x和z内积的平方（时间复杂度是O(n)），就等价于计算映射后高维样本的内积。

$$K(x,z) = (x^T z)^2$$

展开后，得

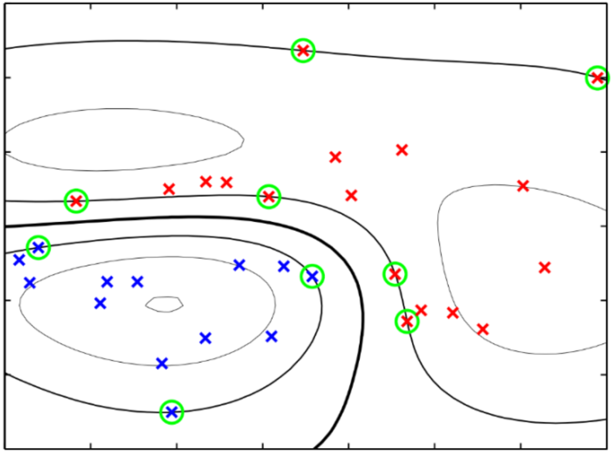
$$K(x,z) = (x^T z)^2 = \left(\sum_{i=1}^{n} x_i z_i\right)\left(\sum_{j=1}^{n} x_j z_j\right) = \sum_{i=1}^{n}\sum_{j=1}^{n} x_i x_j z_i z_j$$

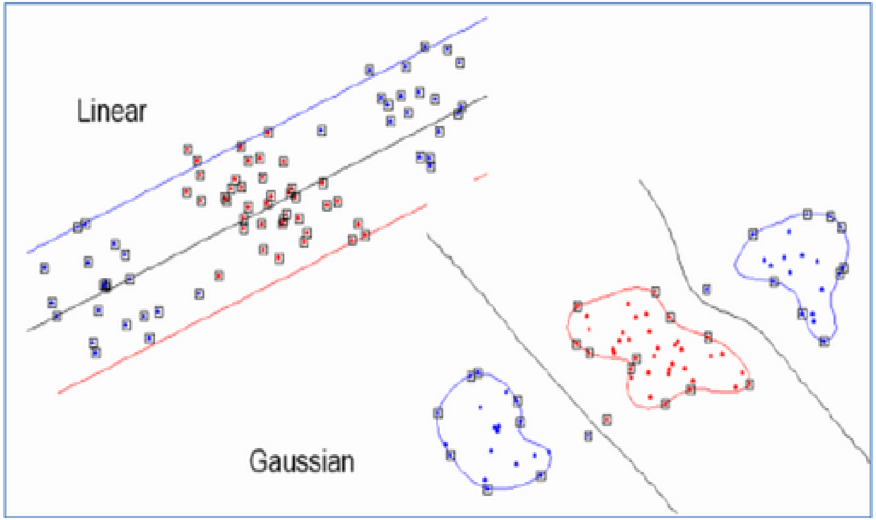
$$= \sum_{i=1}^{n}\sum_{j=1}^{n} (x_i x_j)(z_i z_j) = \phi(x)^T \phi(z)$$

$$\phi(x) = \begin{bmatrix} x_1x_1 \\ x_1x_2 \\ x_1x_3 \\ x_2x_1 \\ x_2x_2 \\ x_2x_3 \\ x_3x_1 \\ x_3x_2 \\ x_3x_3 \end{bmatrix}.$$

高斯核

- 把原始数据映射到无穷维，能够比较每两点之间的相似性。
- 粗线是分割超“平面”
- 其他线是y(x)的等高线
- 绿色圈点是支持向量点





几种常用核函数

h 次多项式核函数：$K(X_i, X_j) = (X_i . X_j + 1)^h$

高斯径向基函数核函数：$K(X_i, X_j) = e^{-\| X_i - X_j \|^2 / 2\sigma^2}$

S 型核函数：$K(X_i, X_j) = \tanh(\kappa X_i \cdot X_j - \delta)$

$$\kappa(x_i, x_j) = (x_i . . x_j + 1)^d$$

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

SVM的求解方法

SVM中系数的求解：SMO

- 序列最小最优化
  - Sequential Minimal Optimization
- 有多个拉格朗日乘子
- 每次只选择其中两个乘子做优化，其他乘子设为是常数。
  - 将N个解问题，转换成两个变量的求解问题：并且目标函数是凸的。

SMO：序列最小最优化

- 考察目标函数，假设α1和α2是变量，其他是定值：

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

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

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

$$\min_{\alpha_1,\alpha_2} W\left(\alpha_1,\alpha_2\right)$$

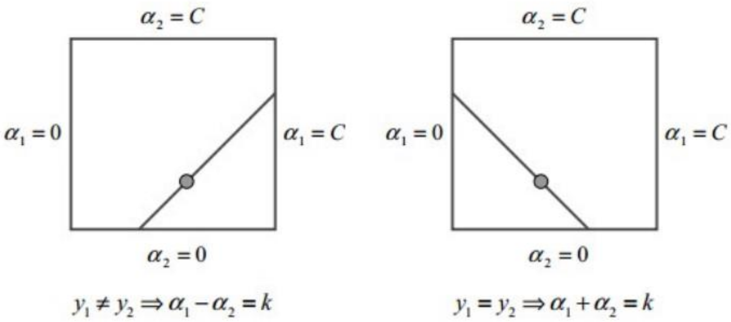
$$= \frac{1}{2}\kappa_{11}\alpha_1^2 + \frac{1}{2}\kappa_{22}\alpha_2^2 + y_1 y_2 \alpha_1 \alpha_2 \kappa_{12} - \left(\alpha_1 + \alpha_2\right)$$

$$+ y_1\alpha_1\sum_{i=3}^{N} y_i\alpha_i\kappa_{i1} + y_2\alpha_2\sum_{i=3}^{N} y_i\alpha_i\kappa_{i2}$$

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

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

二变量优化问题



$$\begin{cases} L = \max\{0, \alpha_j - \alpha_i\} \\ H = \max\{C, C + \alpha_j - \alpha_i\} \end{cases}, \quad y_i \neq y_j$$

$$\begin{cases} L = \max\{0, \alpha_j + \alpha_i - C\} \\ H = \max\{C, \alpha_j - \alpha_i\} \end{cases}, \quad y_i = y_j$$

SMO的迭代公式

- 迭代公式：

$$g(x)=\sum_{i=1}^{N} y_i\alpha_i\kappa(x_i, x)+b$$

$$\eta=\kappa(x_1, x_1)+\kappa(x_2, x_2)-2\kappa(x_1, x_2)=\left\|\Phi(x_1)-\Phi(x_2)\right\|^2$$

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

$$\alpha_j^{new}=\alpha_j^{old}+\frac{y_i(E_i-E_j)}{\eta}$$

SMO算法

- 1. 取初值$\alpha^{(0)}=0$，令k=0
- 2. 选择优化变量$\alpha_1^{(k)}$，$\alpha_2^{(k)}$，解析求解两个变量的优化问题，求得最优解$\alpha_1^{(k+1)}$，$\alpha_2^{(k+1)}$，更新α为$\alpha^{(k+1)}$
- 3. 若在精度ε范围内满足退出条件(下一页)，则转4；否则，k++，转2
- 4. 取$\alpha=\alpha^{(k+1)}$

退出条件

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

$$0 \le \alpha_i \le C, \quad i = 1,2,\ldots n$$

$$y_i \cdot g(x_i) = \begin{cases} \ge 1, & \{x_i | \alpha_i = 0\} \text{// 落在边界外} \\ = 1, & \{x_i | 0 < \alpha_i < C\} \text{// 落在边界上} \\ \le 1, & \{x_i | \alpha_i = C\} \text{// 落在边界内} \end{cases}$$

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

总结与思考

- SVM可以用来划分多类别吗？
  - 直接多分类
  - 1 vs rest / 1 vs 1
- SVM和Logistic回归的比较
  - 经典的SVM，直接输出类别，不给出后验概率；
  - Logistic回归，会给出属于哪个类别的后验概率。
  - 比较二者的损失函数
- SVM框架下引入Logistic函数：输出条件后验概率
- SVM用于回归问题：SVR；
- 体会SVM的目标函数的建立过程
  - 原始目标函数和Lagrange函数有什么联系？

参考文献

- Christopher M. Bishop, *Pattern Recognition and Machine Learning*, Springer Press, 2006
- 李航，统计学习方法，清华大学出版社，2012
- Stephen Boyd,Lieven Vandenberghe. *Convex Optimization*, Cambridge University Press. 2004
  - 中译本：王书宁，许鋆，黄晓霖，凸优化，清华大学出版社，2013
- Charlie Frogner. *Support Vector Machines*. 2011
- John C. Platt. *Sequential Minimal Optimization: A Fast Algorithm for Training Support Vector Machines*. 1998
- Andrew W. Moore. *Support Vector Machines*, 2001