WDS

DataScience@Web

机器学习讨论班

2018年暑期

6. 支持向量机

丁金如

介绍内容

- 支持向量机SVM的原理和目标
- 线性可分支持向量机
- 线性支持向量机
- 非线性支持向量机
- 实验：手写体识别

基本概念

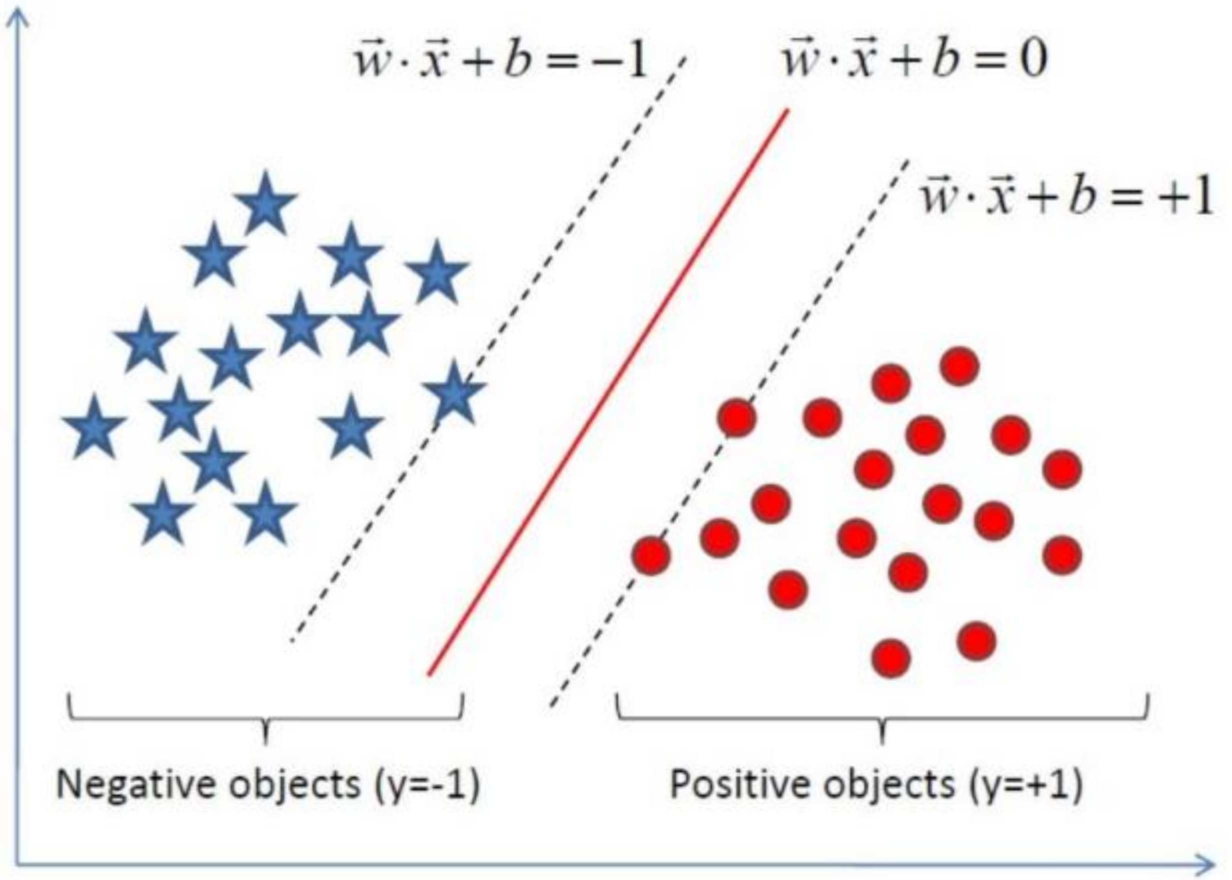
二次规划:一类典型的优化问题。在此类问题中，目标函数是变量的二次函数，而约束条件是变量的线性不等式。

SVM：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，其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。

输入：训练样本集$D=\{(x_1,y_1),\ldots,(x_m,y_m),\}\}$,$y_i\in\{-1,+1\}$

输出：超平面 $\boldsymbol{w}^T\boldsymbol{x}+b=0$

分类决策函数：$f(x)=sign(w^Tx+b)$


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

SVM模型

- 线性可分支持向量机
  - 硬间隔最大化
  - 硬间隔支持向量机
- 线性支持向量机
  - 软间隔最大化
  - 软间隔支持向量机
- 非线性支持向量机
  - 核函数

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

$$y(x) = w^T\varphi(x) + b$$

相应的分类决策函数 $f(x) = sign(w^T\varphi(x) + b)$

该决策函数称为线性可分支持向量机。

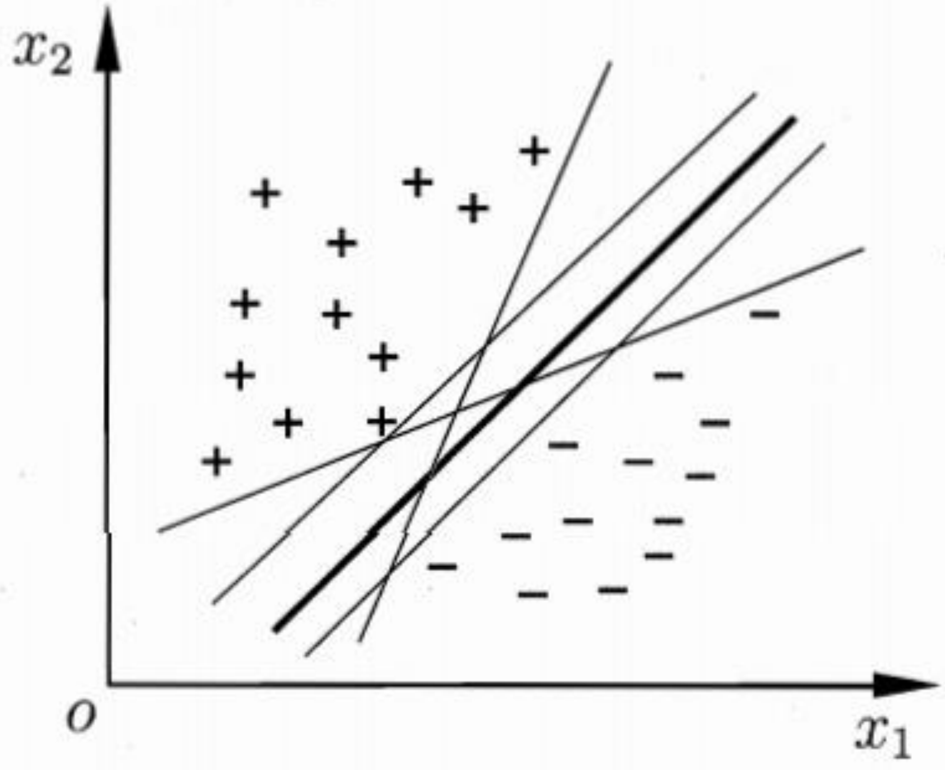
- $\varphi(x)$是某个确定的特征空间转换函数，它的作用是将x映射到(更高的)维度。
  - 最简单的映射： $\varphi(x) = x$

如何寻找最大间隔超平面？

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

目标函数：

$$\arg\max_{w,b}\left\{\frac{1}{\|w\|}\min_i\left[y_i \cdot \left(w^T \cdot \varphi(x_i) + b\right)\right]\right\}$$



建立目标函数

- 总可以通过等比例缩放的方法，使得两类点的函数值都满足 $|y| \geq 1$
- 约束条件：$y_i \cdot (w^T \cdot \varphi(x_i) + b) \geq 1$
- 原目标函数：

$$\arg\max_{w,b}\{\frac{1}{\|w\|}\min_{i}[y_i \cdot (w^T \cdot \varphi(x_i) + b)]\}$$

- 新目标函数：

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

建立目标函数

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

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

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

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

如何求解目标函数？ 拉格朗日乘子法

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

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

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

- 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$L(\boldsymbol{w}, b, \boldsymbol{\alpha})$分别对求偏导，并令其为0：

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

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

计算拉格朗日函数的对偶函数

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

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

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

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

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

$$w=\sum_{i=1}^{n}\alpha_iy_i\varphi(x_i)$$

$$0=\sum_{i=1}^{n}\alpha_iy_i$$

$$\alpha^*=arg\max_{\alpha}(\sum_{i=1}^{n}\alpha_i-\frac{1}{2}\sum_{i,j=1}^{n}\alpha_i\alpha_jy_iy_j\varphi^T(x_i)\varphi(x_j))$$

SMO：序列最小最优化算法

线性可分支持向量机学习算法

- 构造并求解约束最优化问题

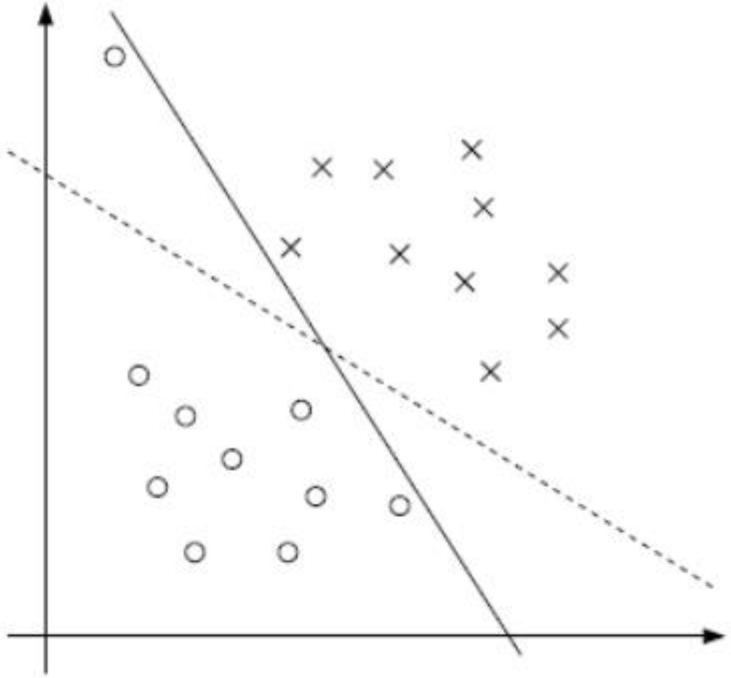
$$\alpha^* = arg\max_{\alpha}(\sum_{i=1}^{n}\alpha_i - \frac{1}{2}\sum_{i,j=1}^{n}\alpha_i\alpha_j y_i y_j \varphi^T(x_i)\varphi(x_j))$$

$$s.t. \sum_{i=1}^{n}\alpha_i y_i = 0 \quad \alpha_i \geq 0, i = 1,2,\cdots,n$$

- 求得最优解α^*
- 计算 $w^* = \sum_{i=1}^{n}\alpha_i^* y_i\varphi(x_i)$ $b^* = y_i - \sum_{i=1}^{n}\alpha_i^* y_i(\varphi(x_i)\cdot\varphi(x_j))$
- 求得分离超平面 $w^*\varphi(x) + b^* = 0$
- 分类决策函数 $f(x) = sign(w^*\varphi(x) + b^*)$

线性支持向量机

- 不一定分类完全正确的超平面就是最好的
- 样本数据本身线性不可分



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：

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

$$s.t. \quad y_i(w \cdot x_i + b) \geq 1 - \xi_i, i = 1,2,\cdots,n$$
$$\xi_i \geq 0, \ i = 1,2,\cdots,n$$

求解最终的目标函数

- 拉格朗日函数

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

- 对w, b, ξ求偏导

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

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

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

带入目标函数

- 将三式带入L中，得到：

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

- 对上式求关于α的极大，得到：

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

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

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

最终的目标函数

- 整理，得到对偶问题：

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

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

$$0 \leq \alpha_i \leq C, i = 1,2,\cdots,n$$

线性支持向量机学习算法

- 构造并求解约束最优化题：

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

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

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

- 求得最优解$\alpha^* = (\alpha_1{}^*, \alpha_2{}^*, \cdots, \alpha_n{}^*)^T$

线性支持向量机学习算法

- 计算

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

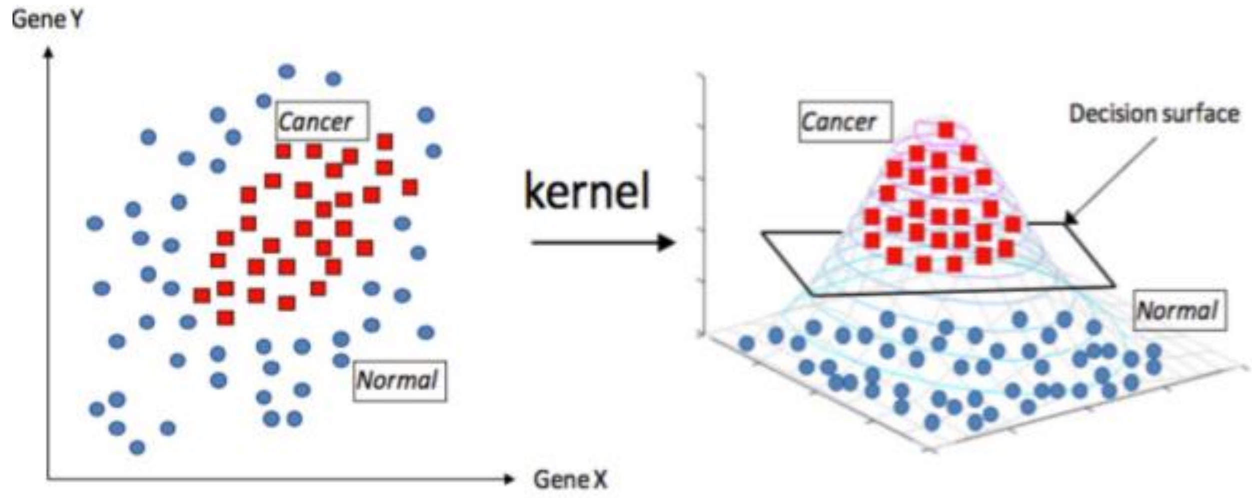
选择α^*的一个分量$\alpha_j{}^*$适合条件$0 \le \alpha_j{}^* \le C$，计算

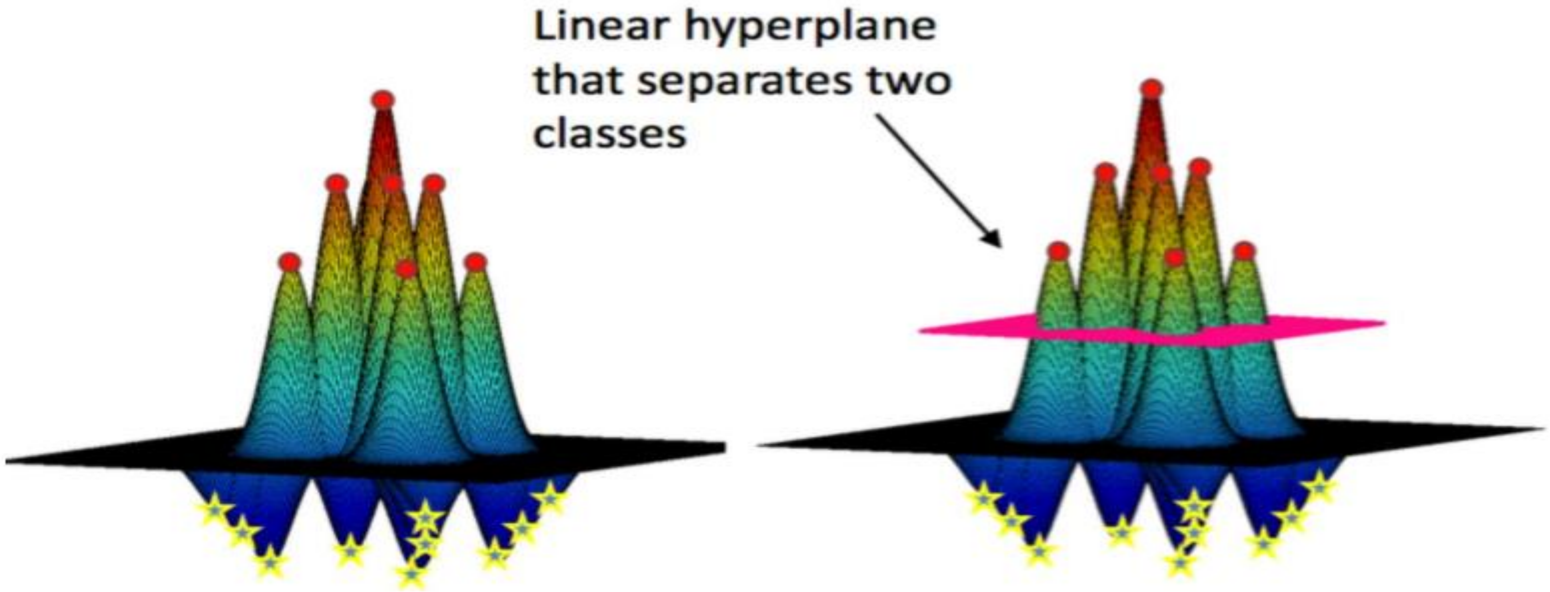
$$b^* = y_i - \sum_{i=1}^{n} \alpha_i{}^* y_i (x_i \cdot x_j)$$

- 求得分离超平面 $w^* \varphi(x) + b^* = 0$

- 分类决策函数 $f(x) = sign(w^* x + b^*)$

非线性支持向量机


Gene Y
Cancer
Normal
Gene X
kernel
Cancer
Decision surface
Normal


Linear hyperplane
that separates two
classes

什么是核函数?

核函数：设χ是输入空间，又设H为特征空间，如果存在一个从χ到H的映射

$$\varphi(x):\ \chi\ \rightarrow H$$

使得对所有$x, z\epsilon\chi$，函数$K(x,z)$满足条件

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

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

常见的核函数

- 多项式核函数：

$$K(x_i, x_j) = (x_i \cdot x_j + c)^d$$

- 高斯核函数:

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

- 在实际应用中，往往依赖先验领域的知识、交叉验证等方法才能选择有效的核函数。
  - 如果没有更多的先验信息，则使用高斯核函数

多项式核函数

$$K(\vec{x},\vec{y}\,) = (\vec{x}\cdot\vec{y}+c)^2$$

$$\Rightarrow (\vec{x}\cdot\vec{y})^2+2c\vec{x}\cdot\vec{y}+c^2$$

$$=\sum_{i=1}^{n}\sum_{j=1}^{n}(x_i x_j)(y_i y_j)+\sum_{i=1}^{n}(\sqrt{2c}x_i\cdot\sqrt{2c}x_j)+c^2$$

例子：若n=3, 即：$\varphi(\vec{x}) = \begin{pmatrix} x_1x_1 \\ x_1x_2 \\ x_1x_3 \\ x_2x_1 \\ x_2x_2 \\ x_2x_3 \\ x_3x_1 \\ x_3x_2 \\ x_3x_3 \\ \sqrt{2c}x_1 \\ \sqrt{2c}x_2 \\ \sqrt{2c}x_3 \\ c \end{pmatrix}$

高斯核函数

$$e^x = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \cdots + \frac{x^n}{n!} + R_n$$

$$\kappa(x_1, x_2) = e^{-\frac{\|x_1 - x_2\|^2}{2\sigma^2}} = e^{-\frac{(x_1 - x_2)^2}{2\sigma^2}} = e^{-\frac{x_1^2 + x_2^2 - 2x_1x_2}{2\sigma^2}} = e^{-\frac{x_1^2 + x_2^2}{2\sigma^2}} \cdot e^{\frac{x_1x_2}{\sigma^2}}$$

$$= e^{-\frac{x_1^2 + x_2^2}{2\sigma^2}} \cdot \left(1 + \frac{1}{\sigma^2} \cdot \frac{x_1x_2}{1!} + \left(\frac{1}{\sigma^2}\right)^2 \cdot \frac{(x_1x_2)^2}{2!} + \left(\frac{1}{\sigma^2}\right)^3 \cdot \frac{(x_1x_2)^3}{3!} + \cdots + \left(\frac{1}{\sigma^2}\right)^n \cdot \frac{(x_1x_2)^n}{n!} + \cdots\right)$$

$$= e^{-\frac{x_1^2 + x_2^2}{2\sigma^2}} \cdot \left(1 \cdot 1 + \frac{1}{1!} \frac{x_1}{\sigma} \cdot \frac{x_2}{\sigma} + \frac{1}{2!} \cdot \frac{x_1^2}{\sigma^2} \cdot \frac{x_2^2}{\sigma^2} + \frac{1}{3!} \cdot \frac{x_1^3}{\sigma^3} \cdot \frac{x_2^3}{\sigma^3} + \cdots + \frac{1}{n!} \cdot \frac{x_1^n}{\sigma^n} \cdot \frac{x_2^n}{\sigma^n} + \cdots\right)$$

$$= \Phi(x_1)^T \cdot \Phi(x_2)$$

- 其中 $\Phi(x) = e^{-\frac{x^2}{2\sigma^2}} \left(1, \sqrt{\frac{1}{1!}} \frac{x}{\sigma}, \sqrt{\frac{1}{2!}} \frac{x^2}{\sigma^2}, \sqrt{\frac{1}{3!}} \frac{x^3}{\sigma^3}, \cdots, \sqrt{\frac{1}{n!}} \frac{x^n}{\sigma^n}, \cdots\right)$

核函数在支持向量的应用

在线性支持向量机的对偶问题中，无论是目标函数还是决策函数，都只涉及输入实例与实例之间的内积。在对偶问题的目标函数中的内积$x_i \cdot x_j$可以用核函数$K(x_i, x_j) = \varphi(x_i) \cdot \varphi(x_j)$来代替。此时对偶问题的目标函数为：

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

非线性支持向量机学习算法

- 选取适当的核函数$K(x,z)$和合适的参数C，构造并求解约束最优化题：

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

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

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

- 求得最优解$\alpha^* = (\alpha_1{}^*, \alpha_2{}^*, \cdots, \alpha_n{}^*)^T$

非线性支持向量机学习算法

- 计算

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

选择α^*的一个正分量适合条件$0 < \alpha_j{}^* < C$，计算

$$b^* = y_i - \sum_{i=1}^{n} \alpha_i{}^* y_i K(x_i, x_j)$$

- 分类决策函数 $f(x) = sign(\sum_{i=1}^{n} \alpha_i{}^* y_i K(x, x_i) + b^*)$

实验

- Python sklearn
- 手写字体识别

谢 谢