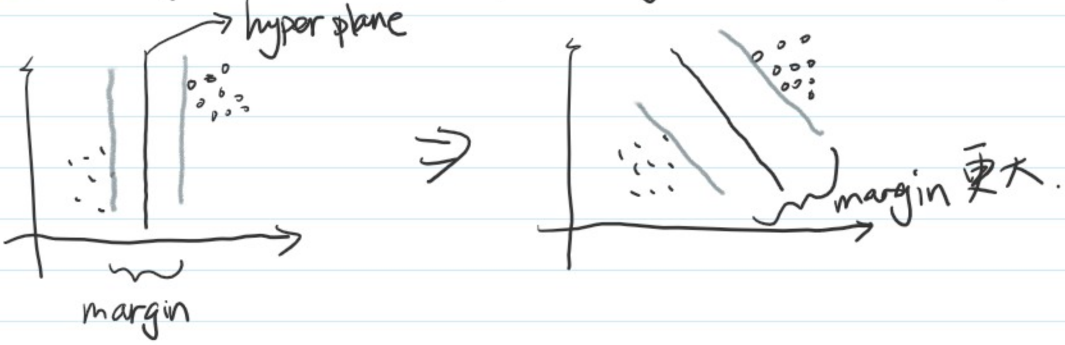
Support Vector Machine

2019年6月28日 16:35

SVM 支持向量机.

目标：找到区分两类的超平面（hyper plane）使边际(margin)最大化.

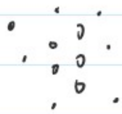


区分两类可以有无数个超平面，但SVM要使边际最大.

Max Margin Hyperplane (MMH).

类别：线性可区分　　　　线性不可区分.

linear separable　　　　linear inseparable.



→ 无法用直线来区分.

linear separable

定义超平面：$WX+b=0$

$W=(w_1, w_2, \cdots w_n)$　　$X=(x_1, x_2 \cdots x_n)^T$

先从简单的2-D情况讨论.

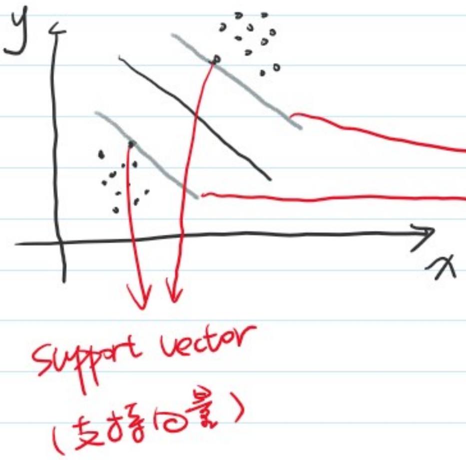
$w_1x_1+w_2x_2+b=0$　把b看成w_0.

即　$w_0+w_1x_1+w_2x_2=0 \Rightarrow$ hyper plane.

即 $w_0 + w_1x_1 + w_2x_2 = 0 \Rightarrow$ hyper plane.

在 hyper plane 上方的点：$w_0 + w_1x + w_2x > 0$.

在 hyper plane 下方的点：$w_0 + w_1x + w_2x < 0$.



调整 weight，使超平面两边边际

$H_1: w_0 + w_1x_1 + w_2x_2 \geq 1 \quad (y=1)$

$H_2: w_0 + w_1x_1 + w_2x_2 \leq 1 \quad (y=-1)$

$y_i(w_0 + w_1x_1 + w_2x_2) \geq 1 \quad (*)$

hyper plane 到 H_1 或 H_2 上的距离为 $\frac{1}{\|W\|}$ → W 的范数.

$W = (w_1, w_2 \cdots w_n)$

$\|W\| = \sqrt{(w_1^2 + w_2^2 + \dots + w_n^2)}$

∴ 最大边际距离为 $\frac{2}{\|W\|}$

把此式变为有限制的凸优化问题 (convex quadratic optimization)
利用 Karush-Kuhn-Tucker (KKT) 条件和拉格朗日公式，可以推出 MMH 可被表示为决策边界 (decision boundary)

$$d(X^T) = \sum_{i=1}^{l} y_i \alpha_i X_i X^T + b_0.$$

其中，X_i 为支持向量点，y_i 是 X_i 的类别标记.
X^T 是被测试的实例，α_i 是拉格朗日方程中的参数.
l 是支持向量的个数.

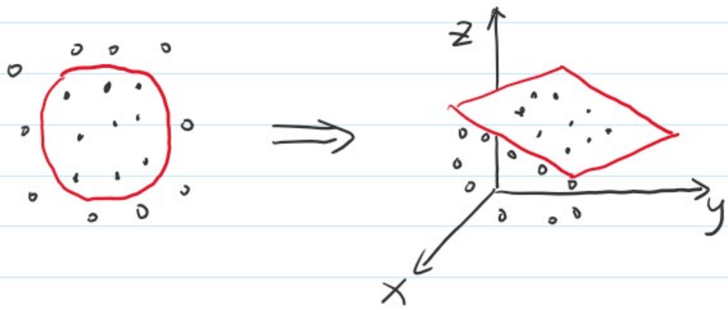
Linear inseparable

数据集在空间中不可能被对应的超平面分开

数据集在空间中不可能被对应的超平面分开

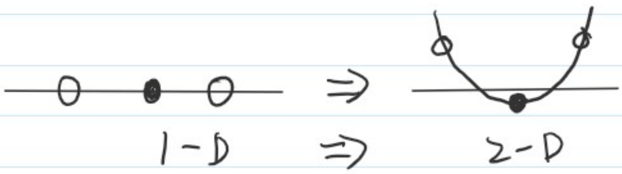
如何处理?

用**非线性映射**把原数据集中的向量点转化到更高维度的空间中.



在更高维的空间中找到一个线性超平面来区分数据集.

例:



如何使用非线性映射转化?

如: 3维向量 $X=(x_1, x_2, x_3)$

$\Downarrow$

6维向量 $\Phi=(\varphi_1, \varphi_2, \varphi_3, \varphi_4, \varphi_5, \varphi_6)$

其中: $\varphi_1=x_1$ $\varphi_2=x_2$, $\varphi_3=x_3$ $\varphi_4=x_1\cdot x_1$ $\varphi_5=x_1\cdot x_2$ $\varphi_6=x_1\cdot x_3$

new hyper plane

$d(\Phi)=w\cdot\Phi+b$ 线性的超平面

通过 KKT 和拉格朗日公式求得 w 和 b.

$$\therefore d(\Phi)=w_1x_1+w_2x_2+w_3x_3+w_4(x_1)^2+w_5x_1x_2+w_6x_1x_3+b$$

$$=w_1\varphi_1+w_2\varphi_2+w_3\varphi_3+w_4\varphi_4+w_5\varphi_5+w_6\varphi_6+b.$$

核方法. (kernel trick).

用核方法来进行非线性映射, 并降低内积时的复杂度.

$$K(x_i, x_j)=\Phi(x_i)\cdot\Phi(x_j)$$

$$K(x_i, x_j) = \Phi(x_i) \cdot \Phi(x_j)$$

常用的核函数

h度多项式核函数 (polynomial kernel of degree h)

$$k(x_i, x_j) = (x_i \cdot x_j + 1)^h$$

高斯径向基核函数 (Gaussian radial basis function kernel)

$$K(x_i, x_j) = e^{-\|x_i - x_j\|^2 / 2\sigma^2}$$ (σ为方差值).

S型核函数 (sigmoid function kernel)

$$K(x_i, x_j) = \tanh(k x_i \cdot x_j - \delta)$$

大多数情况下，图像分类用高斯径向基核函数
而文字分类不用高斯径向基核函数.

例: Example of kernel function

$$X = (x_1, x_2, x_3) \quad Y = (y_1, y_2, y_3)$$

$$f(x) = (x_1x_1, x_1x_2, x_1x_3, x_2x_1, x_2x_2, x_2x_3, x_3x_1, x_3x_2, x_3x_3)$$

$$K(X, Y) = (\langle X, Y \rangle)^2$$

假设 $X = (1, 2, 3) \quad Y = (4, 5, 6)$

$$f(x) = (1, 2, 3, 2, 4, 6, 3, 6, 9)$$

$$f(y) = (16, 20, 24, 20, 25, 30, 24, 30, 36)$$

$$\langle f(x), f(y) \rangle = 16 + 40 + 72 + 48 + 100 + 180 + 72 + 180 + 324 = 1024.$$

而使用 $K(X, Y)$

$$\Downarrow$$

$$K(X, Y) = (4 + 10 + 18)^2 = 1024$$

从而大大简化了计算量.

从而大大简化了计算量.

SVM的扩展应用：多类分类

第一类 | 其他类 ⇒ 第二类 | 其他类 ⋯ 第n类 | 其他类.

one | rest.