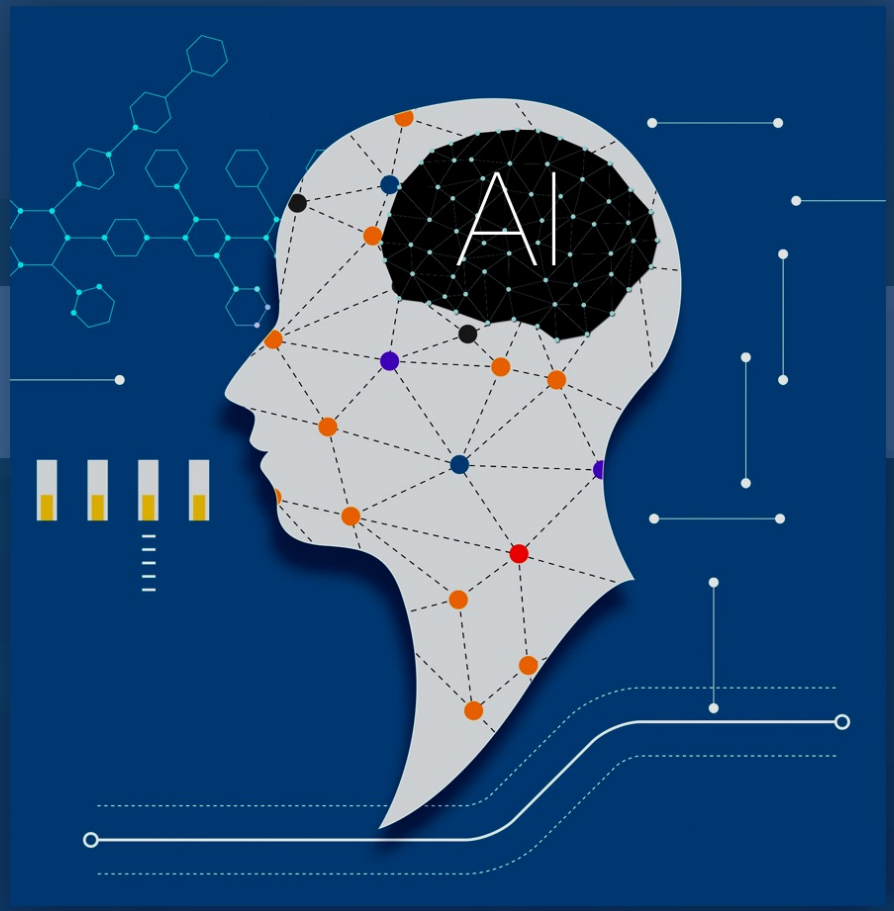
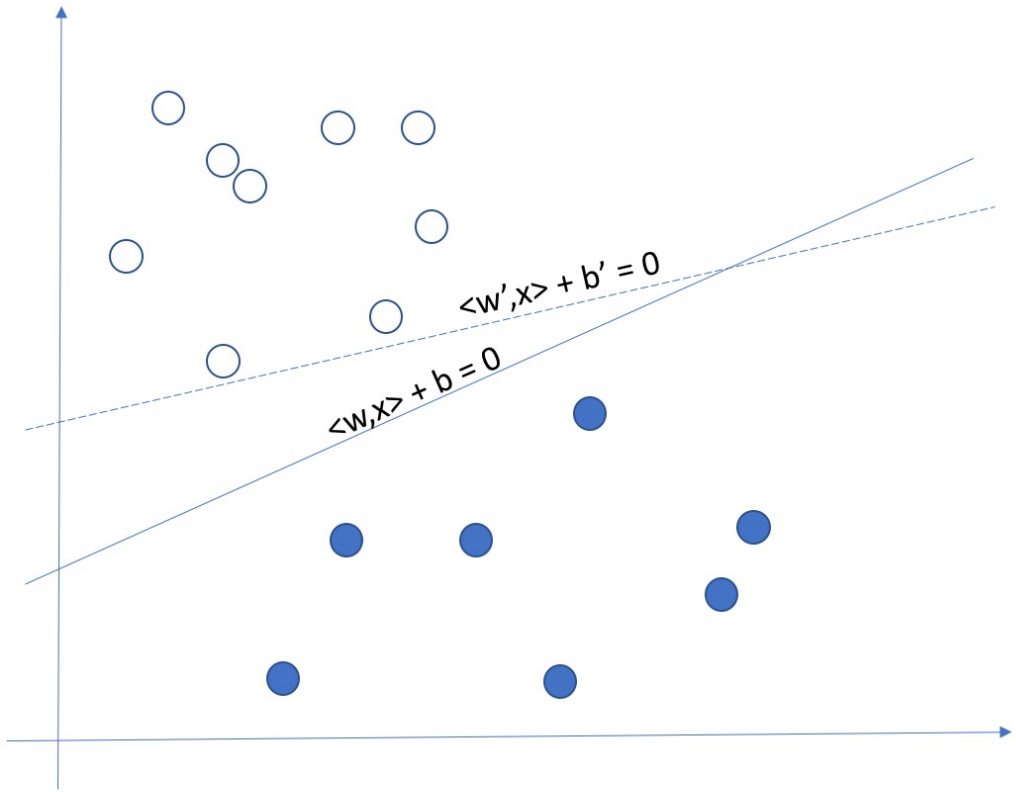


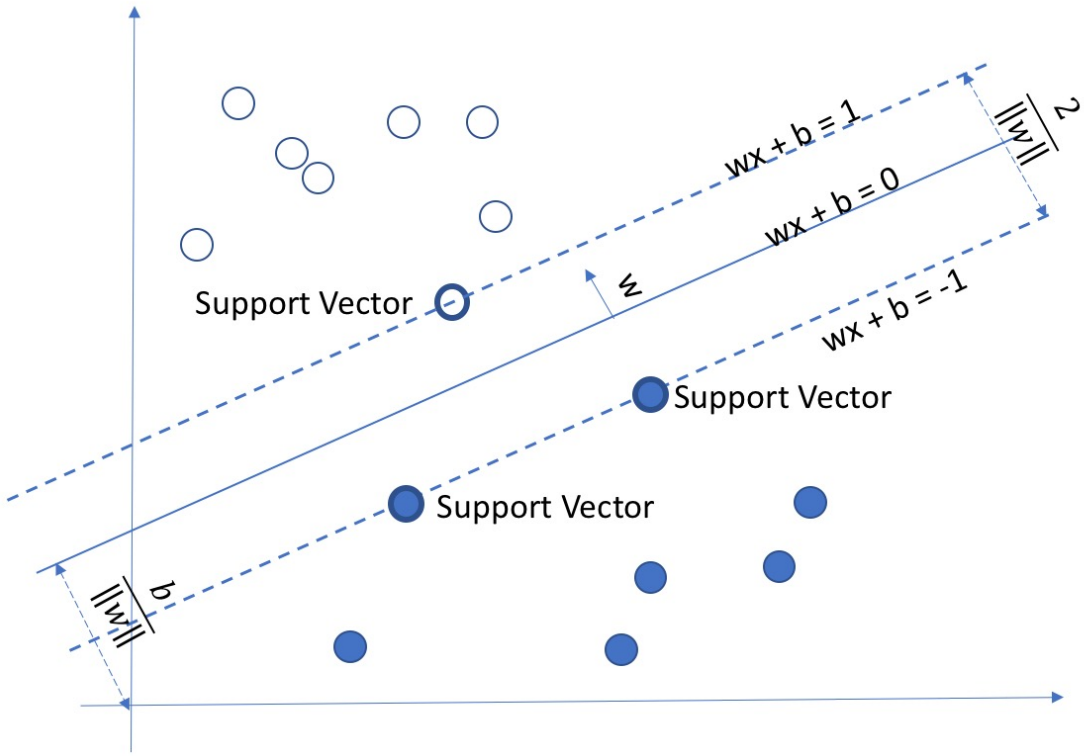
贪心科技
让每个人享受个性化教育服务

支持向量机SVM

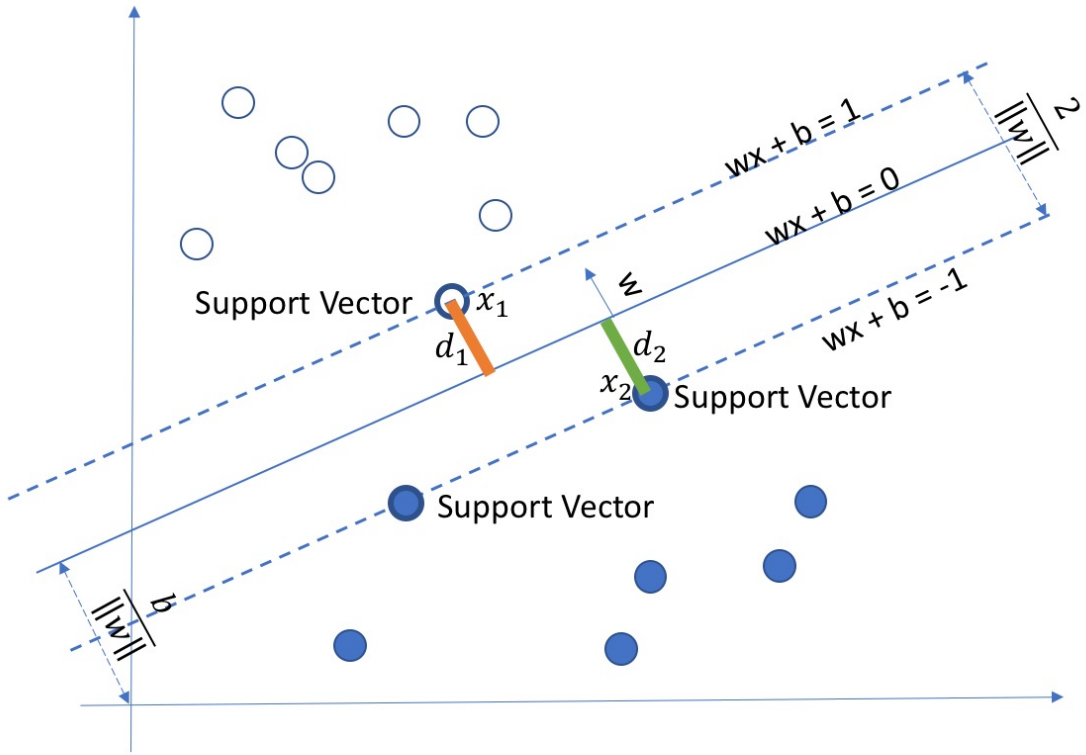
线性分类器



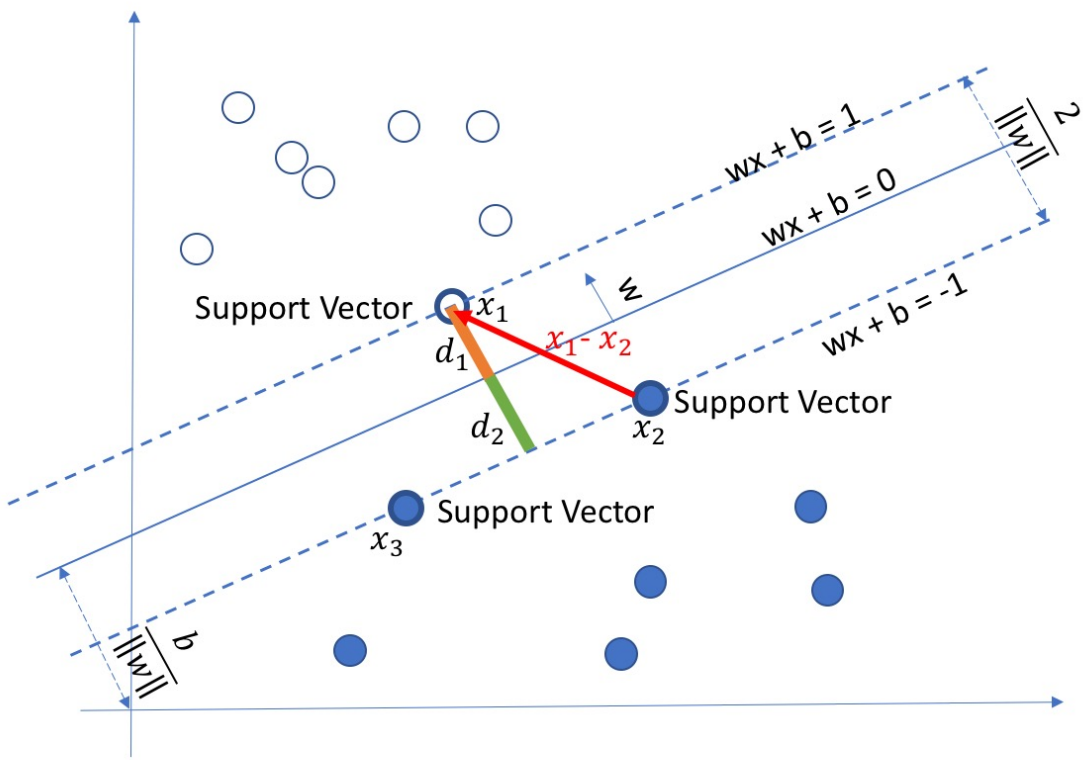
SVM（支持向量机）


wx + b = 1
wx + b = 0
wx + b = -1
2/||w||
w
Support Vector
Support Vector
Support Vector
b/||w||

SVM（支持向量机）


wx + b = 1
wx + b = 0
wx + b = -1
$\frac{2}{\|w\|}$
$\frac{b}{\|w\|}$
w
Support Vector
x_1
d_1
d_2
x_2
Support Vector
Support Vector

SVM（支持向量机）



$$w^T x_1 + b = 1$$

$$w^T x_2 + b = -1$$

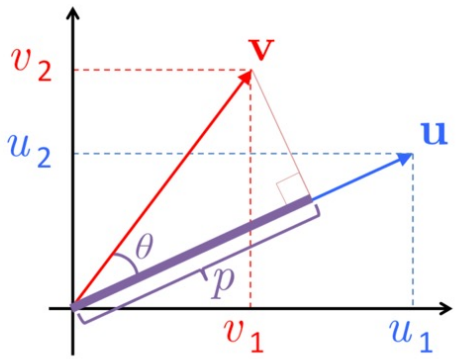
$$(w^T x_1 + b) - (w^T x_2 + b) = 2$$

$$w^T (x_1 - x_2) = 2$$

$$d_1 = d_2 = \frac{w^T(x_1 - x_2)}{2\|w\|_2} = \frac{2}{2\|w\|_2} = \frac{1}{\|w\|_2} = \frac{\frac{w^T}{\|w\|_2}(x_1 - x_2)}{2}$$

$$d_1 + d_2 = \frac{2}{\|w\|_2}$$

向量内积



$$u = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \qquad v = \begin{bmatrix} v_1 \\ v_2 \end{bmatrix}$$

$$\begin{aligned} \|\mathbf{u}\|_2 &= \text{length}(\mathbf{u}) \in \mathbb{R} \\ &= \sqrt{u_1^2 + u_2^2} \end{aligned}$$

$$\begin{aligned} \mathbf{u}^\intercal \mathbf{v} &= \mathbf{v}^\intercal \mathbf{u} \\ &= u_1 v_1 + u_2 v_2 \\ &= \|\mathbf{u}\|_2 \ \|\mathbf{v}\|_2 \cos\theta \\ &= p\|\mathbf{u}\|_2 \quad \text{where } p = \|\mathbf{v}\|_2 \cos\theta \end{aligned}$$

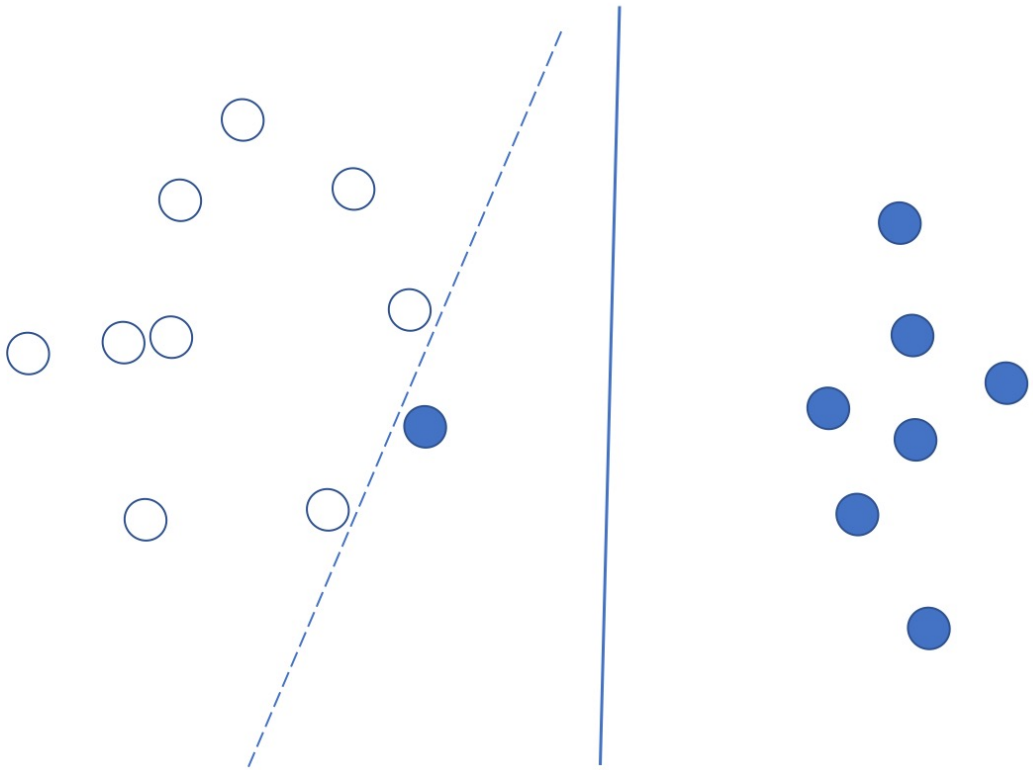
SVM 数学模型

$$\min_{\mathbf{w},b} \quad \frac{1}{2}\mathbf{w}^\top\mathbf{w}$$

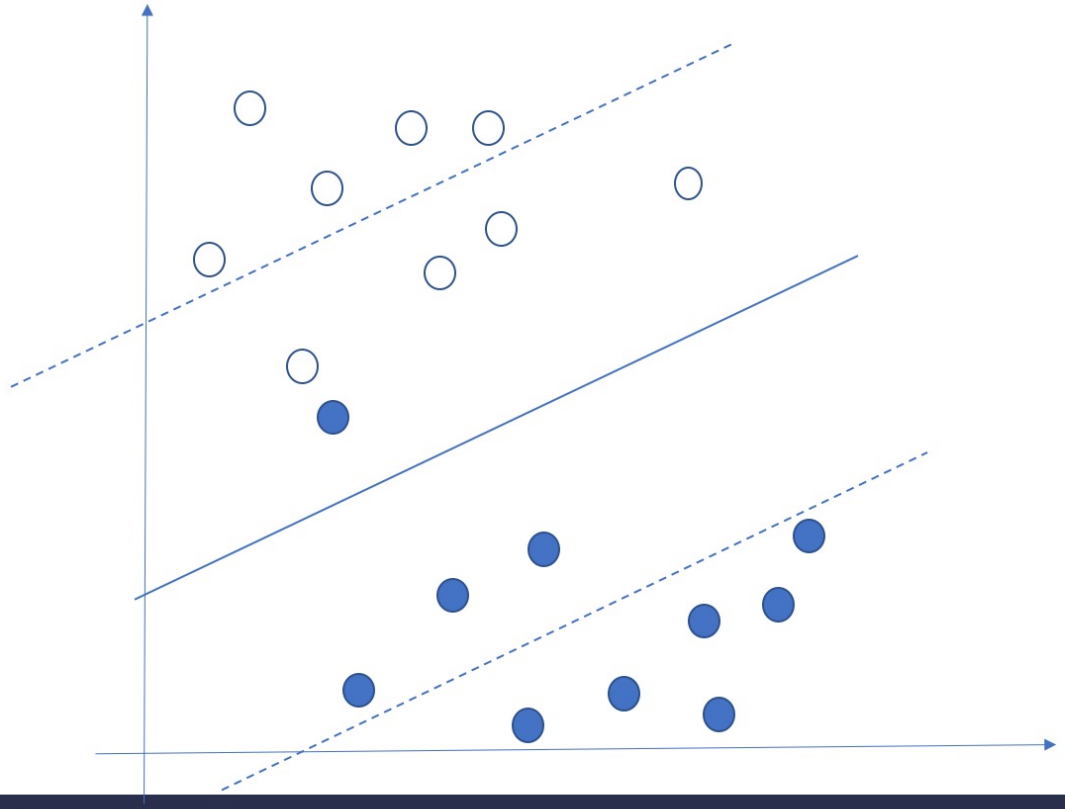
$$\text{s.t.} \quad y_i(\mathbf{w}^\top\mathbf{x}_i + b) \geq 1, \quad i = 1, \ldots, n$$

$$h(\mathbf{x}) = sign(\mathbf{w}^\top\mathbf{x} + b)$$

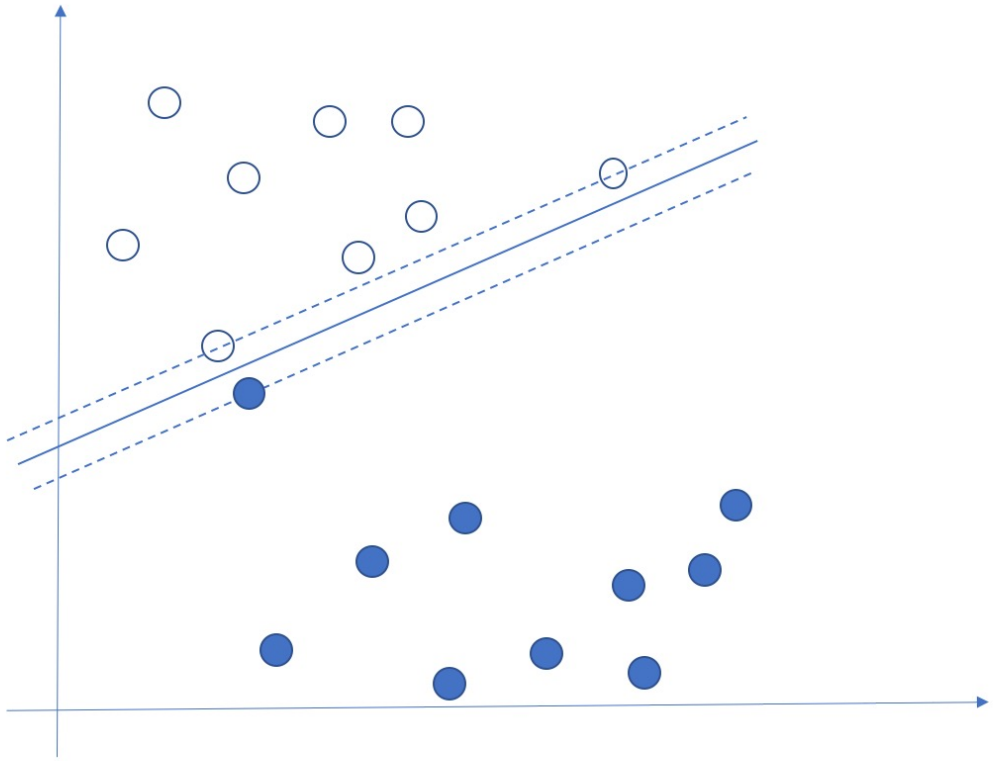
SVM 异常值（outlier）



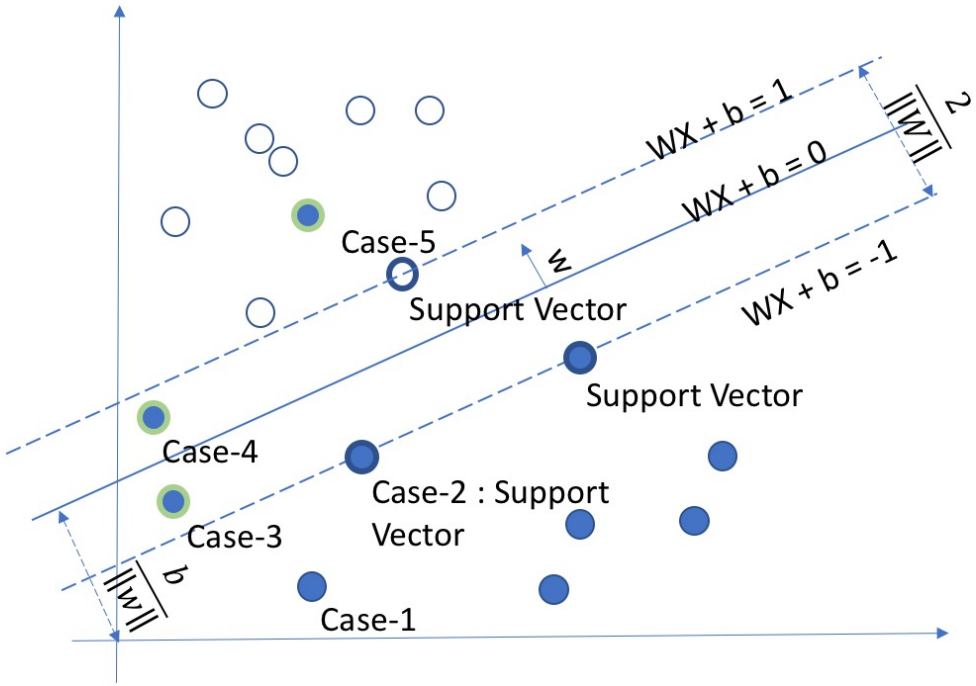
SVM 异常值（outlier）处理一：放松限制



SVM 异常值（outlier）处理二：不放松限制



SVM 异常值（outlier）处理三 ： 必须放松限制（处理线性不可分的情形）



带松弛变量的 SVM 数学模型

$$\min_{\mathbf{w},b,\xi\geq 0} \quad \frac{1}{2}\mathbf{w}^\top\mathbf{w} + C\sum_i \xi_i$$

$$\text{s.t.} \quad y_i(\mathbf{w}^\top\mathbf{x}_i + b) \geq 1 - \xi_i, \quad i = 1, \ldots, n$$

$$\xi_i \geq 0$$

$$h(\mathbf{x}) = sign(\mathbf{w}^\top\mathbf{x} + b)$$

Hinge Loss（合页损失函数）

$$y_i(\mathbf{w}^\top \mathbf{x}_i + b) \geq 1 - \xi_i \Rightarrow \xi_i \geq 1 - y_i(\mathbf{w}^\top \mathbf{x}_i + b)$$

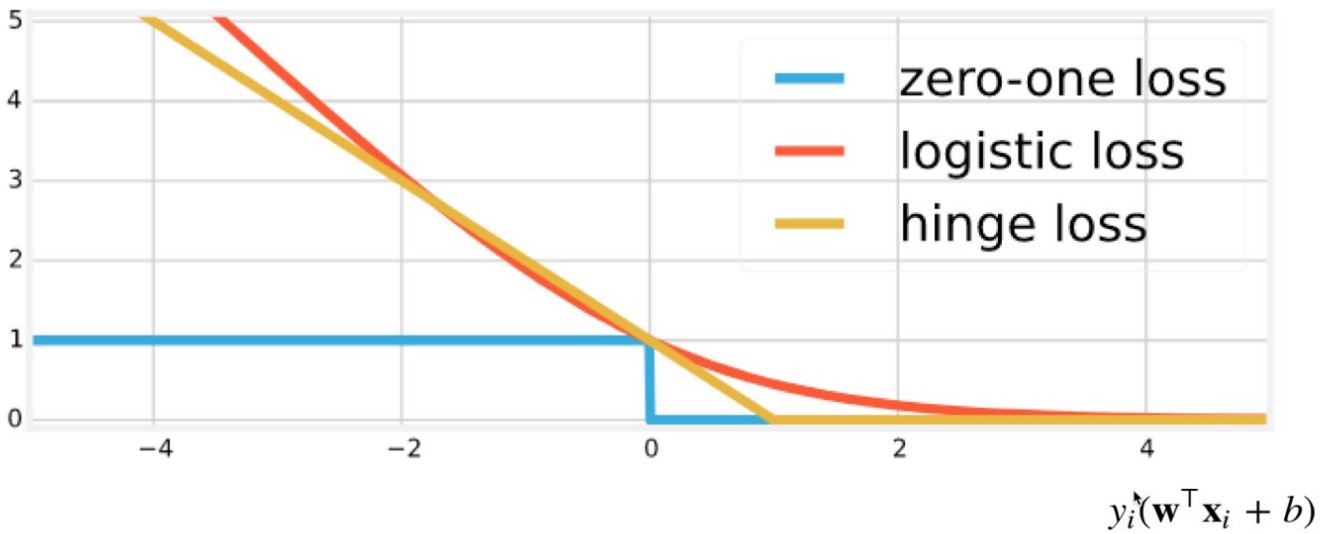
$$\xi_i \geq 0$$

$$\Downarrow$$

$$\xi_i = \max(0, 1 - y_i(\mathbf{w}^\top \mathbf{x}_i + b))$$

Hinge Loss（合页损失函数）

$$\xi_i = \max(0, 1 - y_i(\mathbf{w}^\top \mathbf{x}_i + b))$$



Hinge Loss（合页损失函数）

- Convex 凸函数, 容易优化
- 在自变量小于0的部分梯度比较小, 对错误分类的惩罚比较轻
- 在自变量大于等于1的部分, 值为0: 只要对某个数据分类是正确的, 并且正确的可能性足够高, 那么就用不着针对这个数据进一步优化了
- 在自变量等于0处不可导, 需要分段求导
- 使得在求解最优化时, 只有支持向量(support vector)会参与确定分界线, 而且支持向量的个数远小于训练数据的个数

求解 SVM

- 求解 w, b
  - 方法一: 二次规划 (Quadratic Programming), 经典运筹学的最优化问题, 可以在多项式时间内求得最优解
  - 方法二: 转换为对偶问题

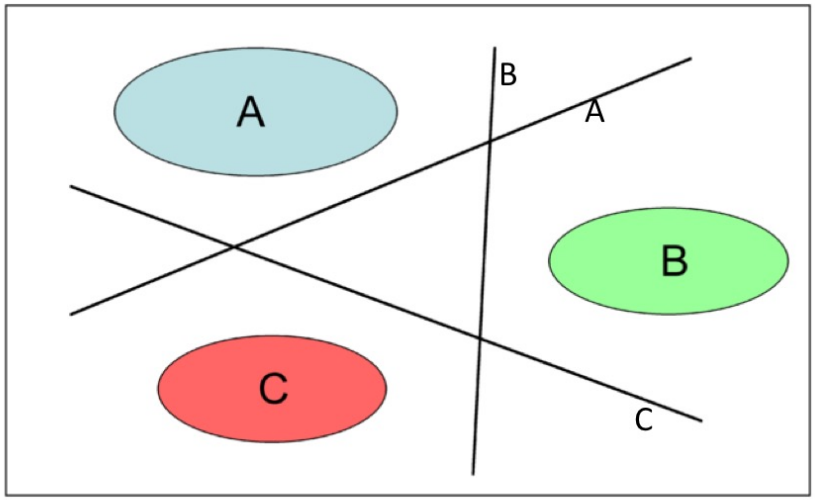
扩展SVM到支持多个类别

- 两种方法:
- 1. OVR (one versus rest): 对于K个类别的情况, 训练K个SVM, 第 j 个SVM用于判断任意条数据是属于类别 j 还是属于类别非 j. 预测的时候, 具有最大值的 $\mathrm{w}_{\mathrm{i}}^{T}\boldsymbol{x}$ 表示给定的数据 $\boldsymbol{x}$ 属于类别 i.

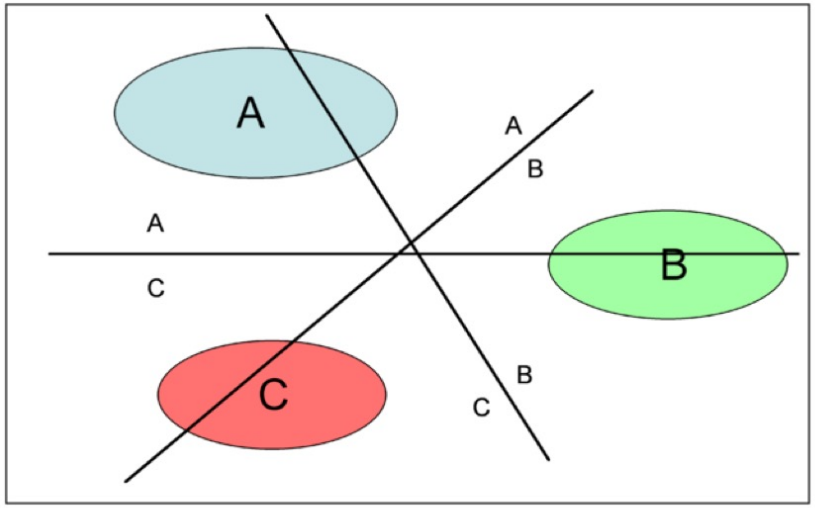
- 2. OVO (one versus one), 对于K个类别的情况, 训练K * (K-1) /2 个SVM, 每一个SVM只用于判读任意条数据是属于K中的特定两个类别. 预测的时候, 使用K * (K-1) /2 个SVM做 K * (K-1) /2 次预测, 使用计票的方式决定数据被分类为哪个类别的次数最多, 就认为数据 $\boldsymbol{x}$ 属于此类别.

扩展SVM到支持多个类别

One versus All



One versus One



实例演示

扩展内容

带松弛变量的 SVM 数学模型

$$\min_{\mathbf{w},b,\xi\geq 0} \quad \frac{1}{2}\mathbf{w}^\top\mathbf{w} + C\sum_i \xi_i$$

$$\text{s.t.} \quad y_i(\mathbf{w}^\top\mathbf{x}_i + b) \geq 1 - \xi_i, \quad i = 1, \ldots, n$$

$$\xi_i \geq 0$$

$$h(\mathbf{x}) = sign(\mathbf{w}^\top\mathbf{x} + b)$$

将原 SVM 最优化问题，添加拉格朗日算子，转化为一个新的最优化问题

$$L(\mathbf{w}, b, \xi, \alpha, \lambda) = \frac{1}{2}\mathbf{w}^\top \mathbf{w} + C\sum_i \xi_i + \sum_i \alpha_i(1 - \xi_i - y_i(\mathbf{w}^\top \mathbf{x}_i + b)) - \sum_i \lambda_i \xi_i$$

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

$$\frac{\partial L}{\partial \mathbf{w}} = 0 \quad \rightarrow \quad \mathbf{w} = \sum_i \alpha_i y_i \mathbf{x}_i$$

$$\frac{\partial L}{\partial \xi_i} = 0 \quad \rightarrow \quad C - \alpha_i - \lambda_i = 0$$

将原 SVM 最优化问题，添加拉格朗日算子，转化为一个新的最优化问题（续）

代入 $\mathbf{w} = \sum_i \alpha_i y_i \mathbf{x}_i$ $\sum_i \alpha_i y_i = 0$

$$L(\xi, \alpha, \lambda) = \sum_i \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j \mathbf{x}_i^\top \mathbf{x}_j + \sum_i \xi_i (C - \alpha_i - \lambda_i)$$

代入 $C - \alpha_i - \lambda_i = 0$

$$\max_{\alpha \geq 0, \lambda \geq 0} \sum_i \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j \mathbf{x}_i^\top \mathbf{x}_j$$

$$\text{s.t.} \quad \sum_i \alpha_i y_i = 0, \quad C - \alpha_i - \lambda_i = 0$$

将原 SVM 最优化问题，添加拉格朗日算子，转化为一个新的最优化问题（续）

$$\max_{\alpha\geq 0,\lambda\geq 0} \quad \sum_i \alpha_i - \frac{1}{2}\sum_{i,j} \alpha_i\alpha_j y_i y_j \mathbf{x}_i^\top \mathbf{x}_j$$

$$\text{s.t.} \quad \sum_i \alpha_i y_i = 0, \quad C - \alpha_i - \lambda_i = 0$$

由于λ_i唯一需要满足的条件是大于等于 0,

约束条件 $C - \alpha_i - \lambda_i = 0$ 也可以改为:

$\alpha_i \leq C$

SVM 对偶形式

两个数据点属于同一类别使值增加，否则减小

衡量两个数据之间的相似性

$$\max_{\alpha\geq 0,\lambda\geq 0} \quad \sum_i \alpha_i - \frac{1}{2}\sum_{i,j} \alpha_i \alpha_j y_i y_j \mathbf{x}_i^\top \mathbf{x}_j$$

$$\text{s.t.} \quad \sum_i \alpha_i y_i = 0, \quad C - \alpha_i - \lambda_i = 0$$

不同数据点的权重不同，不同的类别的权重一致

使用核函数

$$\max_{\alpha \geq 0} \quad \sum_i \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j k(\mathbf{x}_i, \mathbf{x}_j)$$

$$\text{s.t.} \quad \sum_i \alpha_i y_i = 0, \quad \alpha_i \leq C, \quad i = 1, \ldots, n$$

$$K(\mathbf{x}_i, \mathbf{x}_j) = \langle \Phi(\mathbf{x}_i), \Phi(\mathbf{x}_j) \rangle$$

使用核函数 $k(\mathbf{x}_i, \mathbf{x}_j)$ 替代 $\mathbf{x}_i^\top \mathbf{x}_j$

The Kernel Trick（核技巧）

- 如果一个算法可以表达为关于一个正定核 K1 的函数, 那么可以将它转化为关于另外一个正定核 K2 的函数

- SVM 可以使用 The Kernel Trick

使用核函数

两个数据点属于同一类别使值增加，否则减小

衡量两个数据之间的相似性

$$\max_{\alpha \geq 0} \quad \sum_i \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j k(\mathbf{x}_i, \mathbf{x}_j)$$

$$\text{s.t.} \quad \sum_i \alpha_i y_i = 0, \quad \alpha_i \leq C, \quad i = 1, \ldots, n$$

不同数据点的权重不同，不同的类别的权重一致

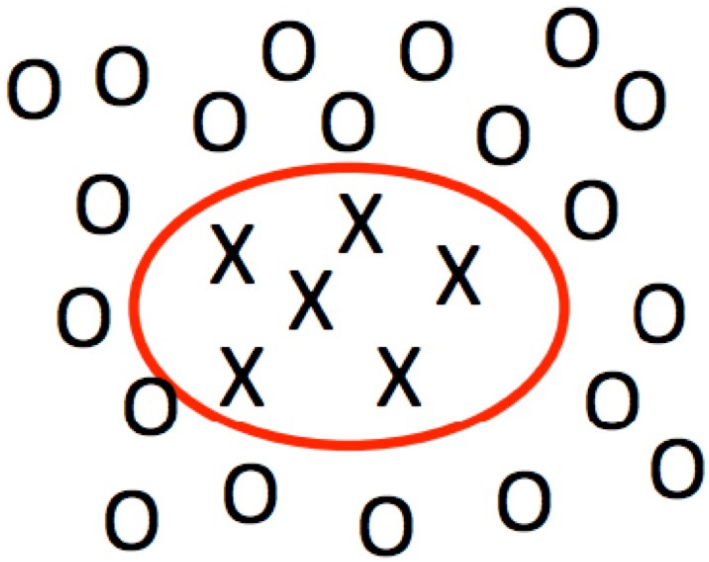
使用核函数, 预测公式

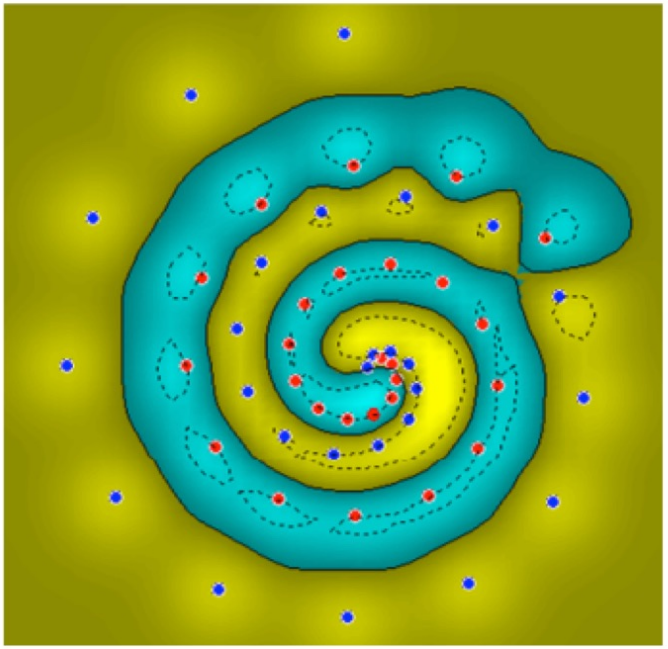
$$b = y_i - \sum_j \alpha_j y_j k(\mathbf{x}_j, \mathbf{x}_i) \quad \forall i \quad C > \alpha_i > 0$$

$$\mathbf{w}^\top \phi(\mathbf{x}) + b = \sum_i \alpha_i y_i k(\mathbf{x}_i, \mathbf{x}) + b$$

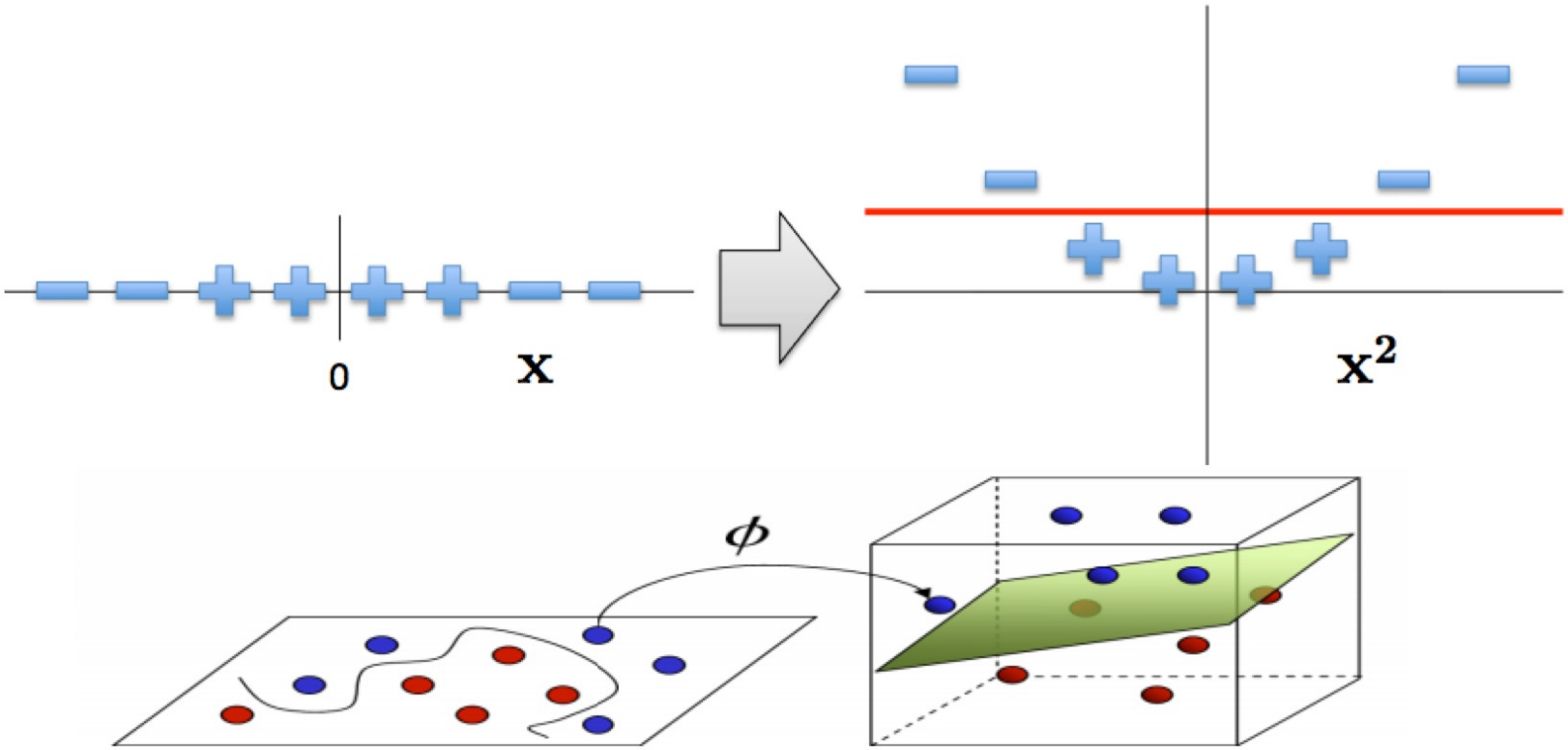
只有当 $\mathbf{x}_i$ 为支持向量的时候, $\alpha_i > 0$

为什么要使用核函数（处理线性不可分的情况）





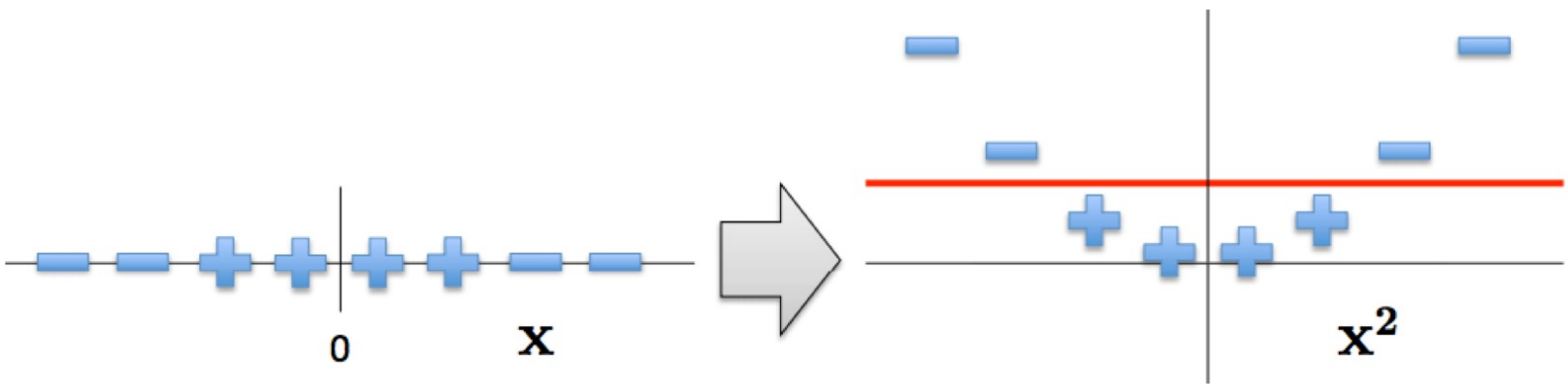
将特征映射到更高的维度



将原理的特征映射到更高的维度（续）

$$\Phi : \mathcal{X} \mapsto \hat{\mathcal{X}} = \Phi(\mathbf{x})$$

$$\Phi([x_{i1}, x_{i2}]) = [x_{i1}, x_{i2}, x_{i1}x_{i2}, x_{i1}^2, x_{i2}^2]$$



直接扩展到高纬的问题

- 一. 增大了计算量
  - 计算量与数据量和每一条数据的维度正相关
- 二. 没有办法增加到无限维

成为 Kernel 的条件

$$K(\mathbf{x}_i, \mathbf{x}_j) = \langle \Phi(\mathbf{x}_i), \Phi(\mathbf{x}_j) \rangle$$

Gram 矩阵:

$$G_{ij} = K(\mathbf{x}_i, \mathbf{x}_j)$$

一. 为对称矩阵

二. 为半正定矩阵 $\mathbf{z}^{\mathsf{T}} \mathrm{G} \mathbf{z} \geq 0$

$$\mathbf{z} \in \mathbb{R}^n$$

常用的 Kernel

多项式核（Polynomial Kernel）

$$K(\mathbf{x}_i, \mathbf{x}_j) = (\langle \mathbf{x}_i, \mathbf{x}_j \rangle + c)^d$$

- C ≥ 0 控制低阶项的强度

- 特殊情况, 当 C = 0, d =1 成为线性核(Linear Kernel), 就于无核函数的SVM一样

$$K(\mathbf{x}_i, \mathbf{x}_j) = \langle \mathbf{x}_i, \mathbf{x}_j \rangle$$

多项式核（Polynomial Kernel）举例

$$\mathbf{x}_i = [x_{i1}, x_{i2}] \qquad \mathbf{x}_j = [x_{j1}, x_{j2}]$$

$$\begin{aligned} K(\mathbf{x}_i, \mathbf{x}_j) &= \langle \mathbf{x}_i, \mathbf{x}_j \rangle^2 \\ &= (x_{i1}x_{j1} + x_{i2}x_{j2})^2 \\ &= (x_{i1}^2 x_{j1}^2 + x_{i2}^2 x_{j2}^2 + 2x_{i1}x_{i2}x_{j1}x_{j2}) \\ &= \langle \Phi(\mathbf{x}_i), \Phi(\mathbf{x}_j) \rangle \end{aligned}$$

$$\Phi(\mathbf{x}_i) = [x_{i1}^2, x_{i2}^2, \sqrt{2}x_{i1}x_{i2}]$$

$$\Phi(\mathbf{x}_j) = [x_{j1}^2, x_{j2}^2, \sqrt{2}x_{j1}x_{j2}]$$

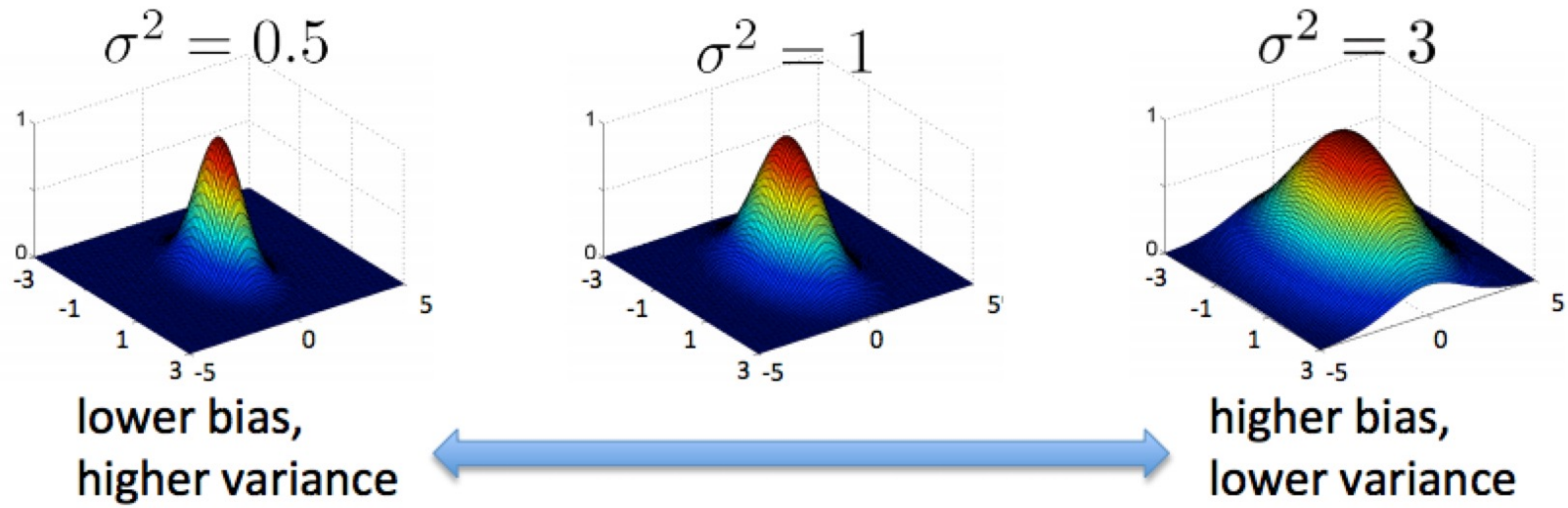
高斯核（Gaussian Kernel），也称为 Radial Basis Function（RBF）Kernel

$$K(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|_2^2}{2\sigma^2}\right)$$

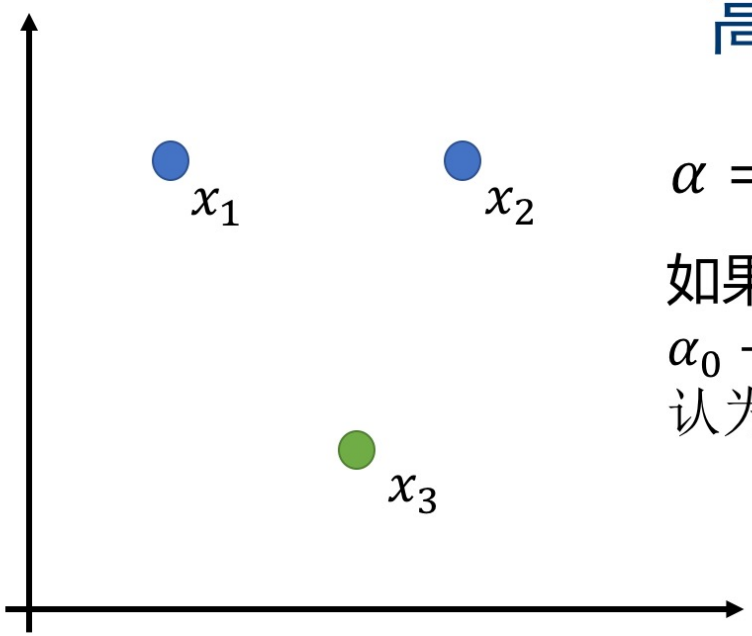
当$\mathbf{x}_i = \mathbf{x}_j$，值为1，当 x_i 与 x_j 距离增加，值倾向于0

使用高斯核之前需要将特征正规化

高斯核（Gaussian Kernel）参数的意义



高斯核举例



$\alpha = [-0.5, 1.0, 1.0, 0.0]$

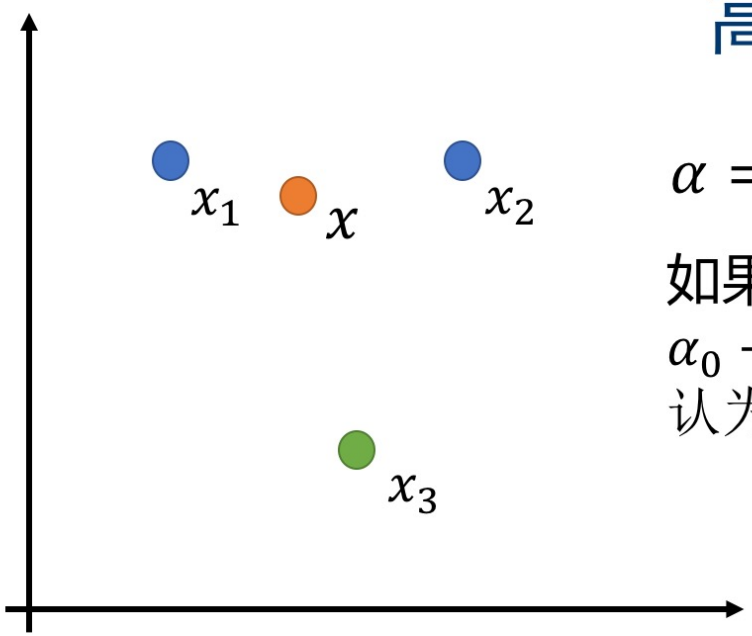
如果

$\alpha_0 + \alpha_1 K(x, x_1) + \alpha_2 K(x, x_2) + \alpha_3 K(x, x_3) \geq 0,$

认为输出1

$$K(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|_2^2}{2\sigma^2}\right)$$

高斯核举例



$\alpha = [-0.5, 1.0, 1.0, 0.0]$

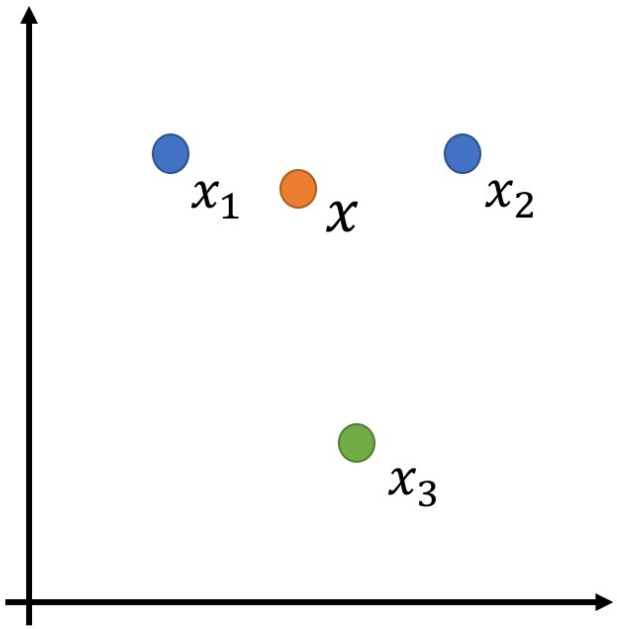
如果

$\alpha_0 + \alpha_1 K(x, x_1) + \alpha_2 K(x, x_2) + \alpha_3 K(x, x_3) \geq 0,$

认为输出1

$$K(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|_2^2}{2\sigma^2}\right)$$

高斯核举例



$\alpha = [-0.5, 1.0, 1.0, 0.0]$

如果

$\alpha_0 + \alpha_1 K(x, x_1) + \alpha_2 K(x, x_2) + \alpha_3 K(x, x_3) \geq 0,$

认为输出1

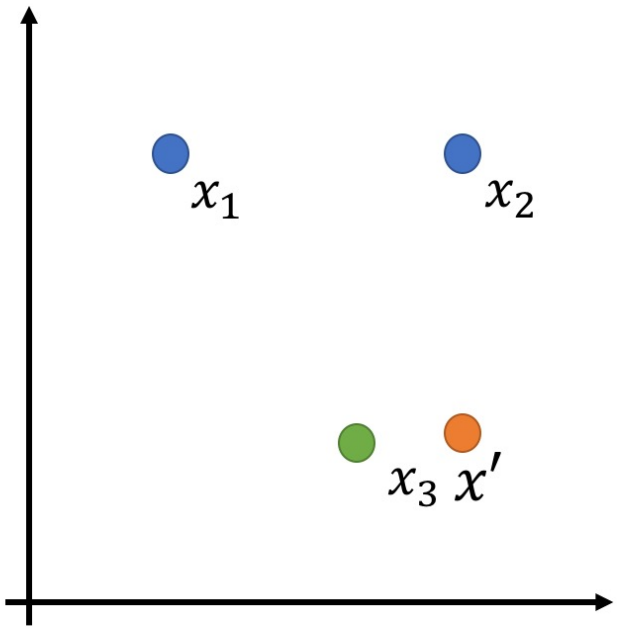
因为 x 接近 x_1, 所以 $K(x, x_1) \approx 1$, 其他情况≈ 0

$\alpha_0 + \alpha_1 K(x, x_1) + \alpha_2 K(x, x_2) + \alpha_3 K(x, x_3)$

$= -0.5 + 1 * 1 + 1 * 0 + 0 * 0 = 0.5 \geq 0$

$$K(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|_2^2}{2\sigma^2}\right)$$

高斯核举例



$\alpha = [-0.5, 1.0, 1.0, 0.0]$

如果

$\alpha_0 + \alpha_1 K(x, x_1) + \alpha_2 K(x, x_2) + \alpha_3 K(x, x_3) \geq 0,$

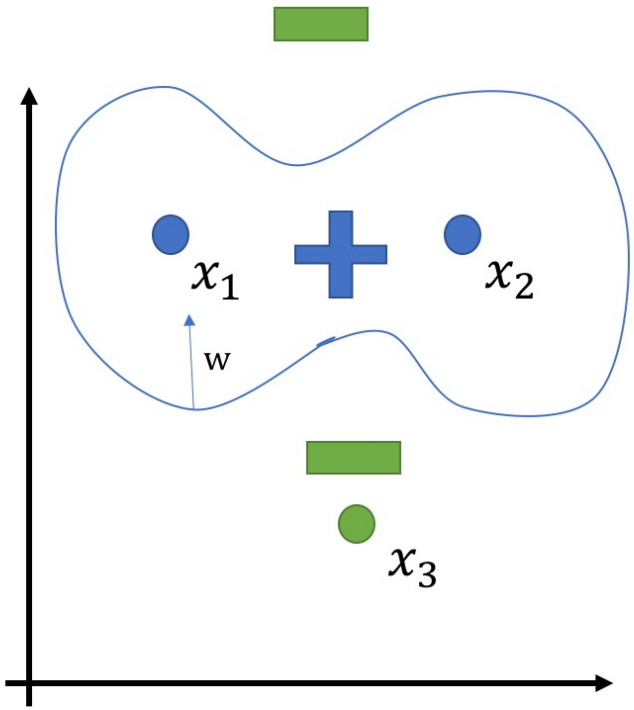
认为输出1

因为 x' 接近 x_3, 所以 $K(x', x_3) \approx 1$, 其他情况≈ 0

$\alpha_0 + \alpha_1 K(x, x_1) + \alpha_2 K(x, x_2) + \alpha_3 K(x, x_3)$

$= -0.5 + 1 * 0 + 1 * 0 + 0 * 1 = -0.5 \leq 0$

$$K(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|_2^2}{2\sigma^2}\right)$$



高斯核举例

$\alpha = [-0.5, 1.0, 1.0, 0.0]$

如果

$\alpha_0 + \alpha_1 K(x, x_1) + \alpha_2 K(x, x_2) + \alpha_3 K(x, x_3) \geq 0,$

认为输出1

大致边界

$$K(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|_2^2}{2\sigma^2}\right)$$

Sigmoid Kernel

- 此时的SVM等价于一个没有隐含层(Hidden Layer)的简单神经网络

$$K(\mathbf{x}_i, \mathbf{x}_j) = \tanh\left(\alpha \mathbf{x}_i^\intercal \mathbf{x}_j + c\right)$$

Cosine Similarity Kernel

- 常用于衡量两段文字的相似性
- 相当于衡量两个向量的余弦相似度 (向量夹角的余弦值)

$$K(\mathbf{x}_i, \mathbf{x}_j) = \frac{\mathbf{x}_i^\top \mathbf{x}_j}{\|\mathbf{x}_i\| \, \|\mathbf{x}_j\|}$$

Chi-squared Kernel

- 常用于计算机视觉
- 衡量两个概率分布的相似性
- 输入数据必须是非负的, 并且使用了L1 归一化 (L1 Normalized)

$$K(\mathbf{x}_i, \mathbf{x}_j) = \frac{\mathbf{x}_i^\top \mathbf{x}_j}{\|\mathbf{x}_i\| \ \|\mathbf{x}_j\|}$$

总结

- SVM 专注于找最优分界线, 用于减小过拟合
- Kernel Trick的应用使得 SVM 可以高效的用于非线性可分的情况

- 优势
  - 理论非常完美
  - 支持不同Kernel, 用于调参

- 缺点
  - 当数据量特别大时, 训练比较慢

实例演示

贪心科技 GREEDY TECHNOLOGY

贪心科技　让每个人享受个性化教育服务

THANKS

贪心学院讲师：袁源