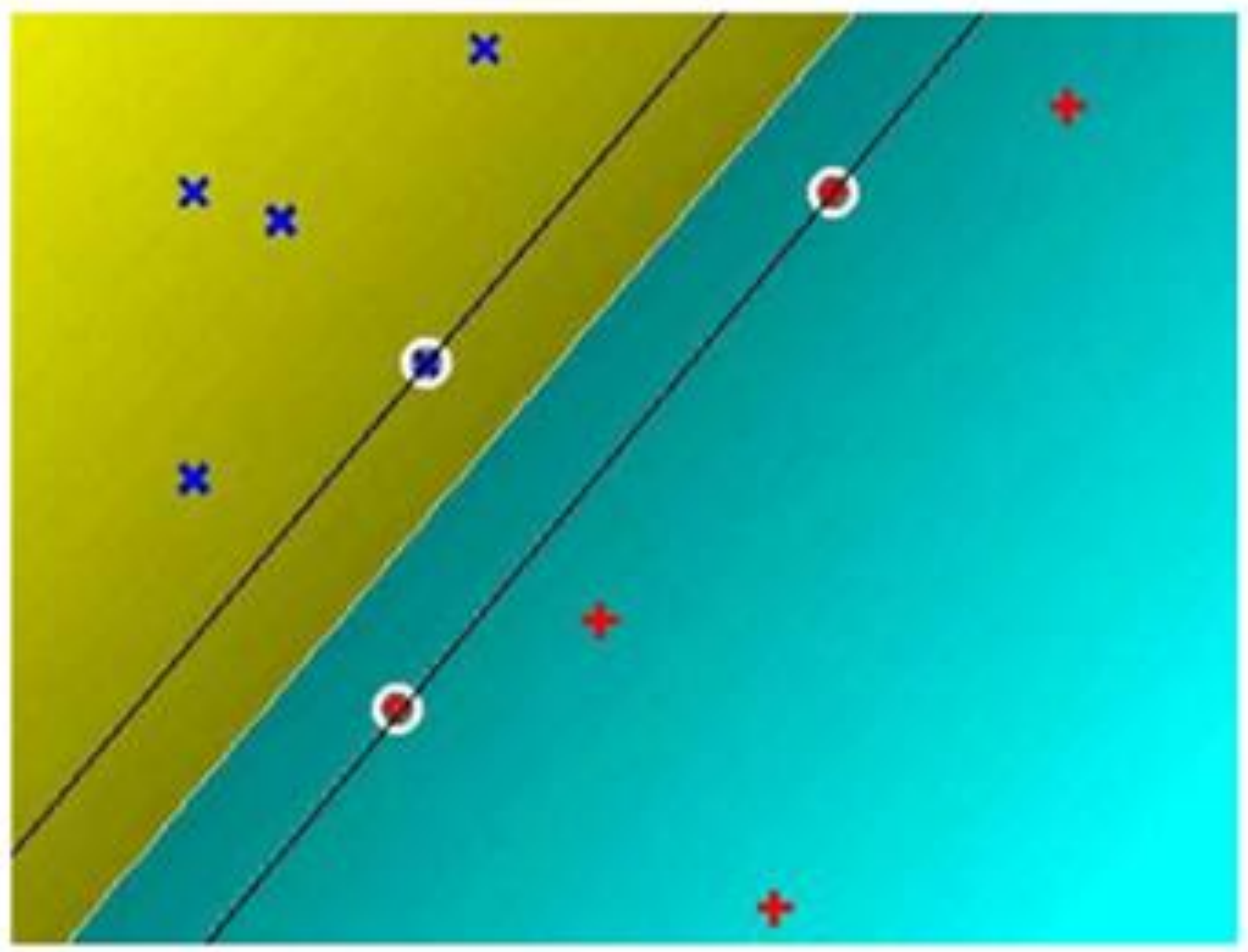
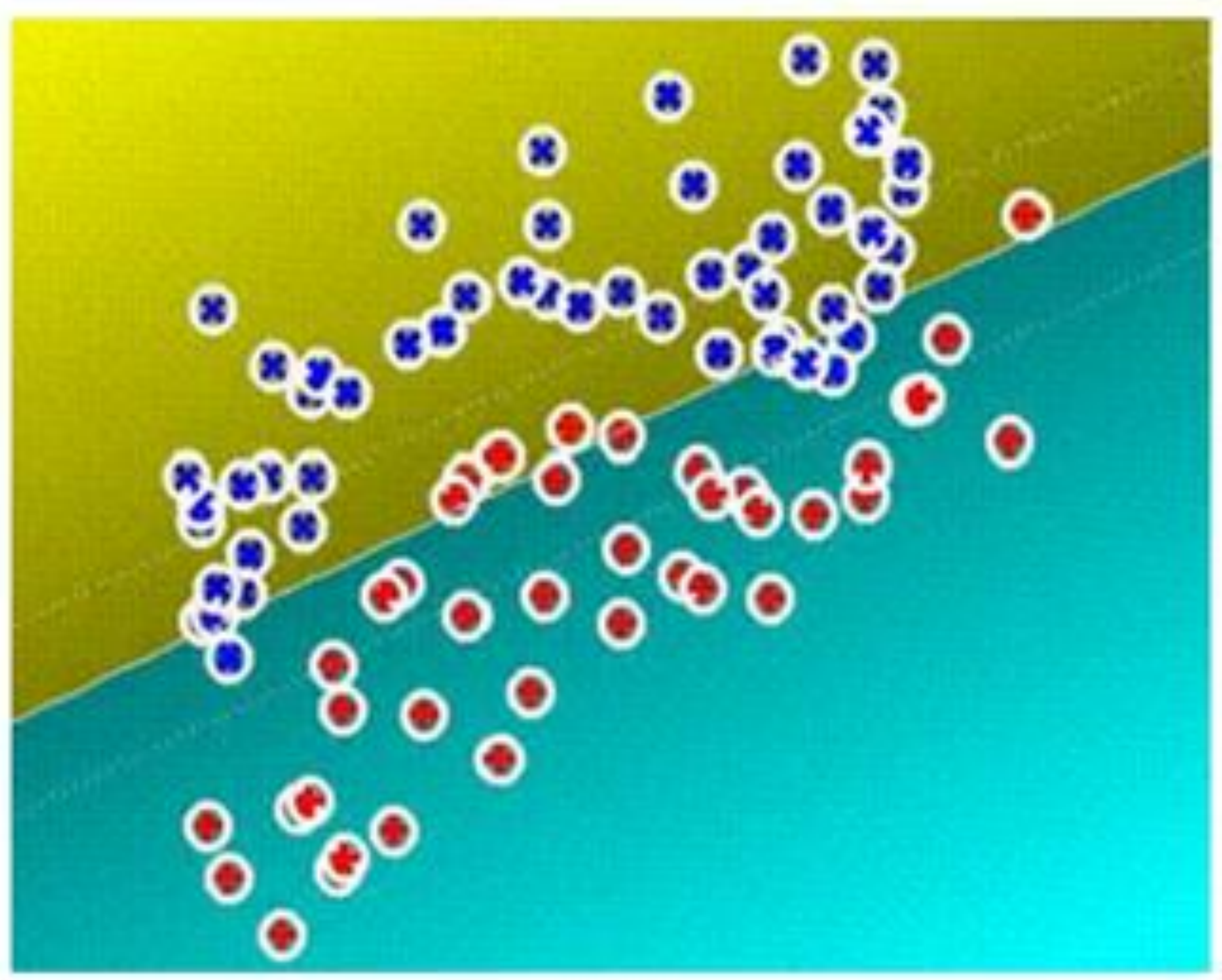
Python
机器学习实战

支持向量机

支持向量机的三种情况

- 线性可分
- 近似线性可分
- 非线性可分

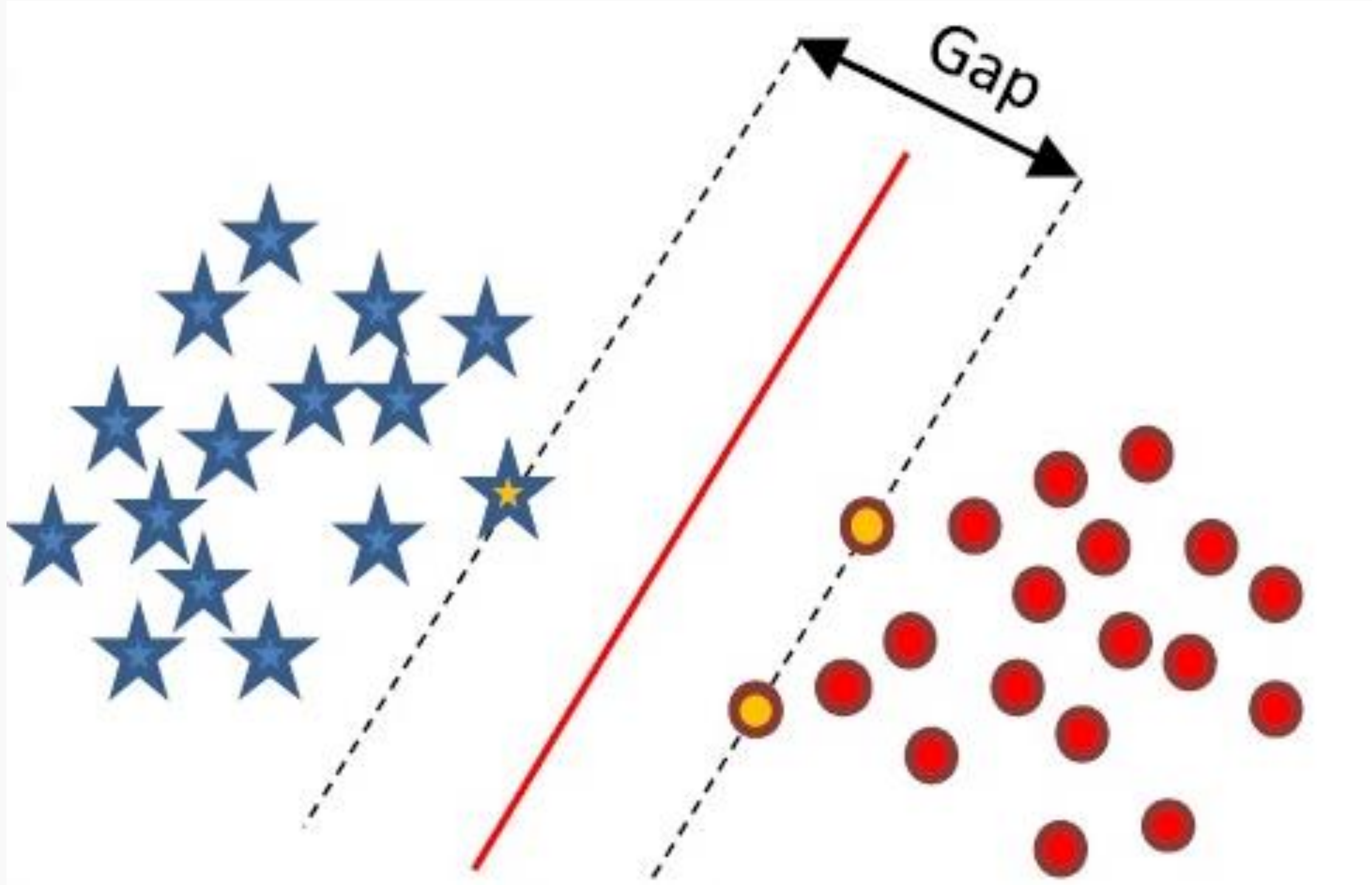




支持向量机

重要概念

- 支持向量
- 分隔面
- 间隔(margin)



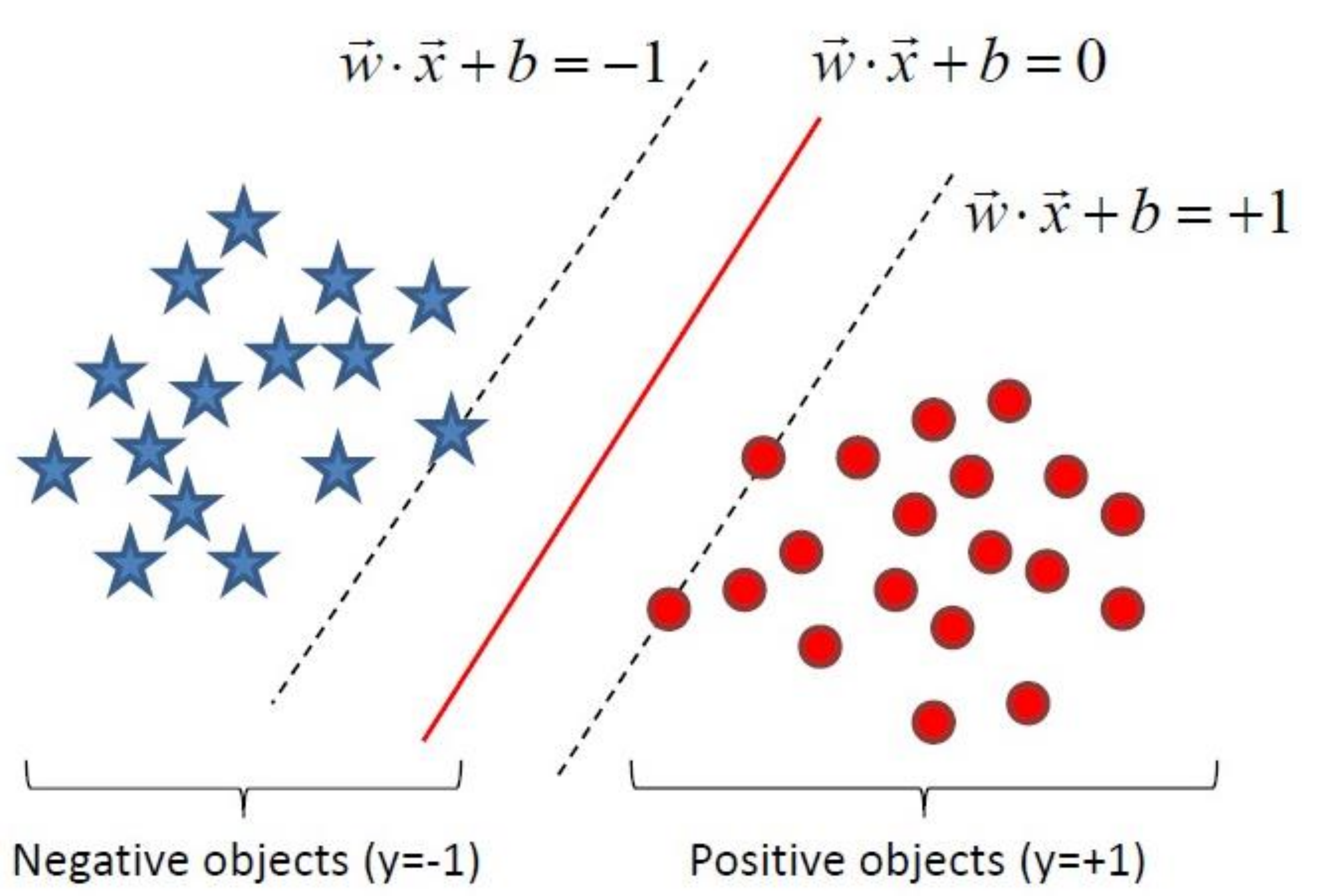
支持向量机

三个平面公式

W是平面的法向量

两个平面之间的距离

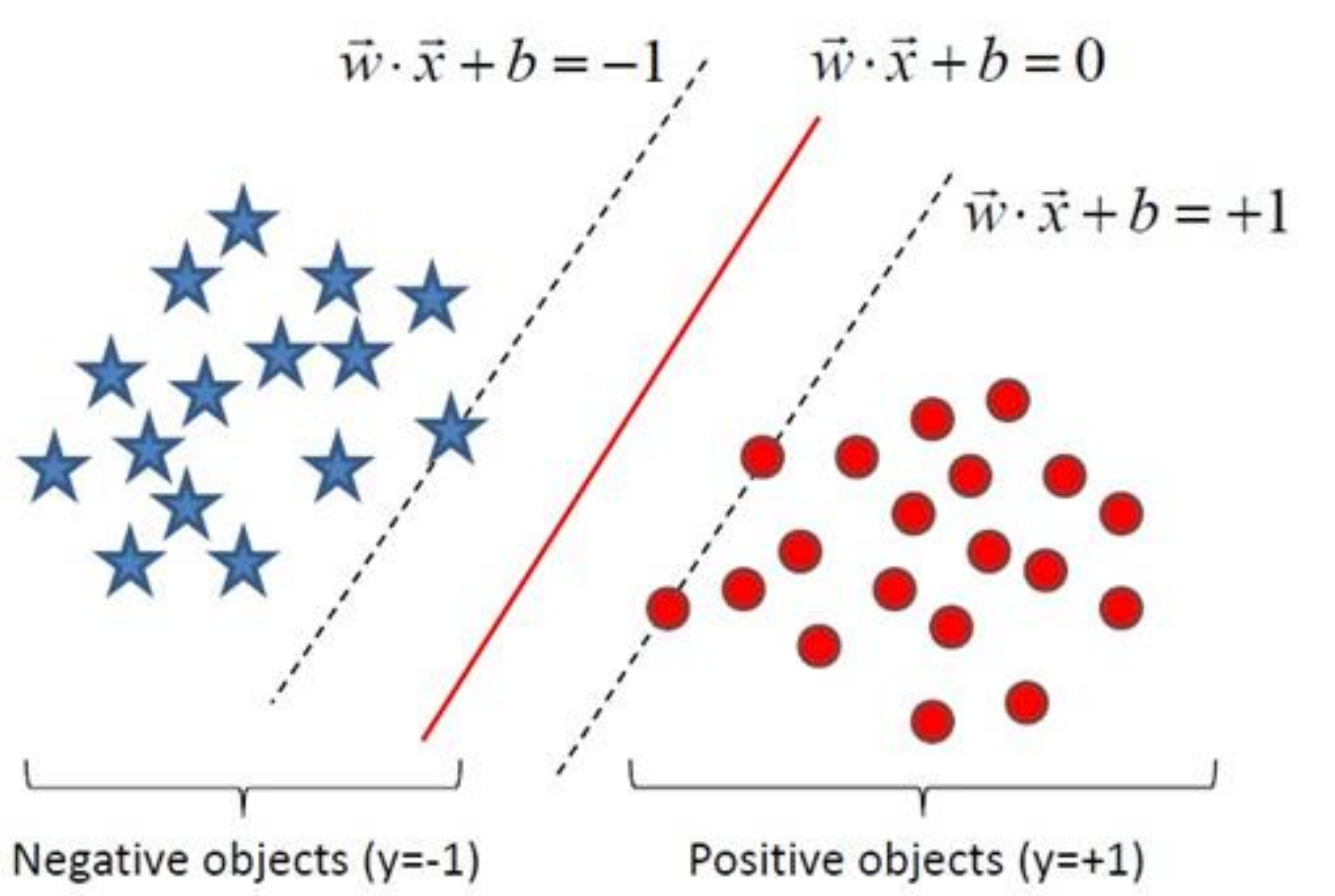
$$\frac{2}{\|\mathbf{w}\|}$$



支持向量机

$$\max \frac{1}{\|w\|}, \quad s.t., y_i(w^T x_i + b) \geq 1, i = 1, \ldots, n$$

$$\min \frac{1}{2}\|w\|^2 \quad s.t., y_i(w^T x_i + b) \geq 1, i = 1, \ldots, n$$

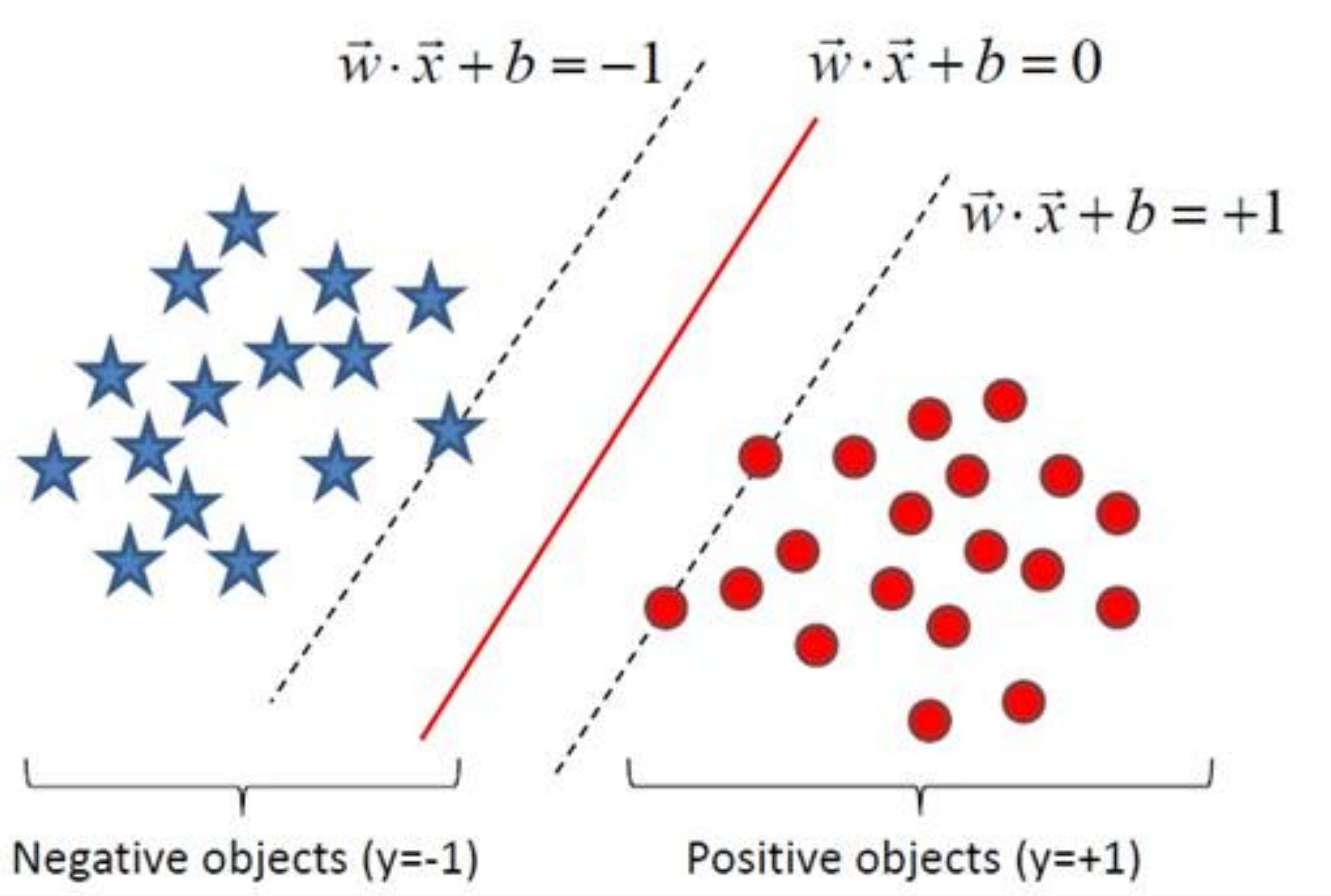


线性可分

支持向量机

$$\max \frac{1}{\|w\|}, \quad s.t., y_i(w^T x_i + b) \geq 1, i = 1, \ldots, n$$

$$\min \frac{1}{2}\|w\|^2 \quad s.t., y_i(w^T x_i + b) \geq 1, i = 1, \ldots, n$$



拉格朗日乘子法

要找函数$z=f(x,y)$在条件$\varphi(x,y)=0$下的可能极值点，

构造函数$F(x,y)=f(x,y)+\lambda\varphi(x,y)$，其中$\lambda$为某一常数

令 $\frac{\partial F}{\partial x}=0, \quad \frac{\partial F}{\partial y}=0, \quad \frac{\partial F}{\partial \lambda}=0$

$$\begin{cases} f_x(x,y)+\lambda\varphi_x(x,y)=0, \\ f_y(x,y)+\lambda\varphi_y(x,y)=0, \\ \varphi(x,y)=0. \end{cases}$$

解出x,y,λ，其中x,y就是可能的极值点

例 7　将正数 12 分成三个正数 x, y, z 之和 使得 $u = x^3 y^2 z$ 为最大.

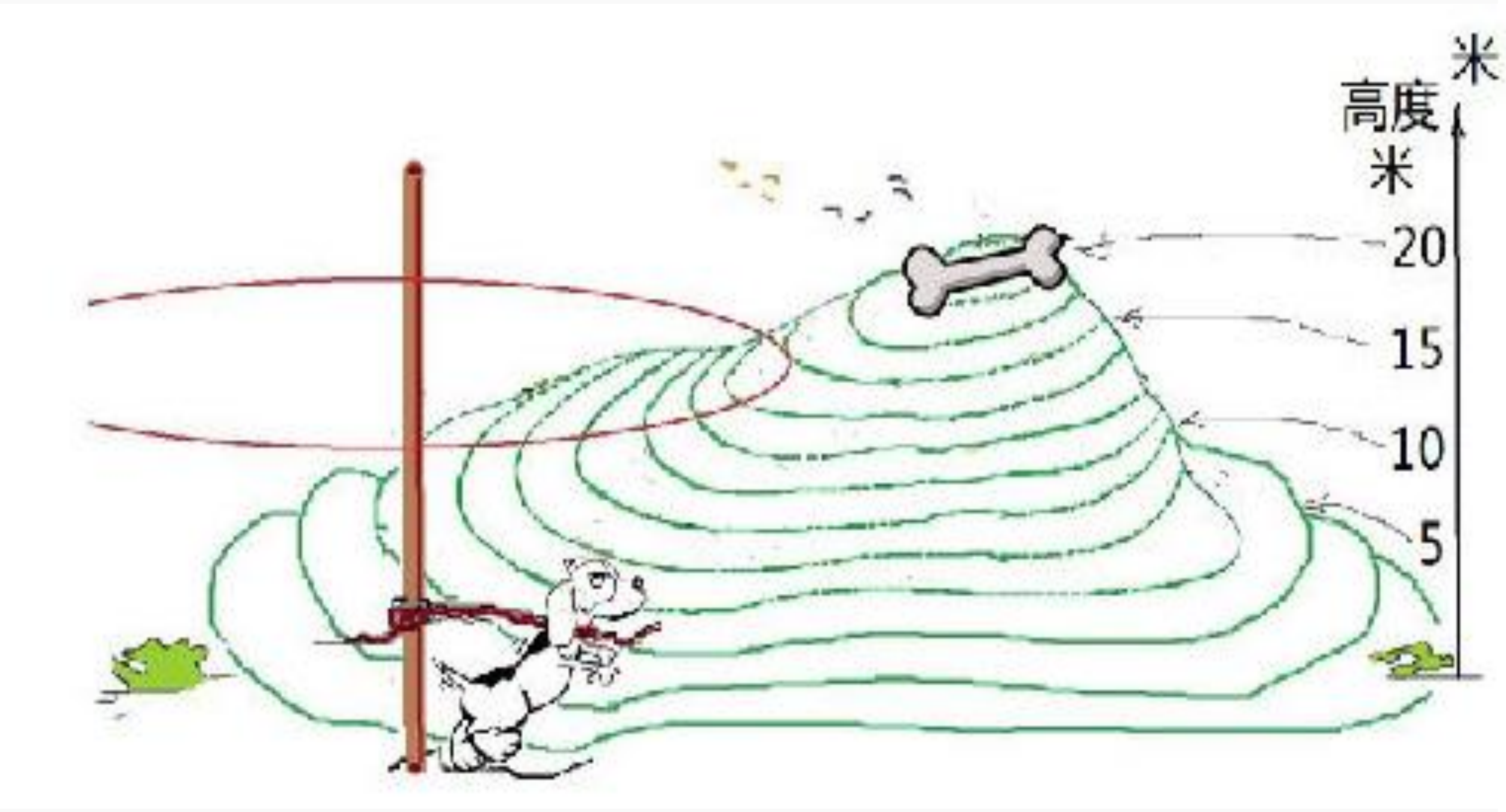
解　令 $F(x, y, z) = x^3 y^2 z + \lambda(x + y + z - 12)$,

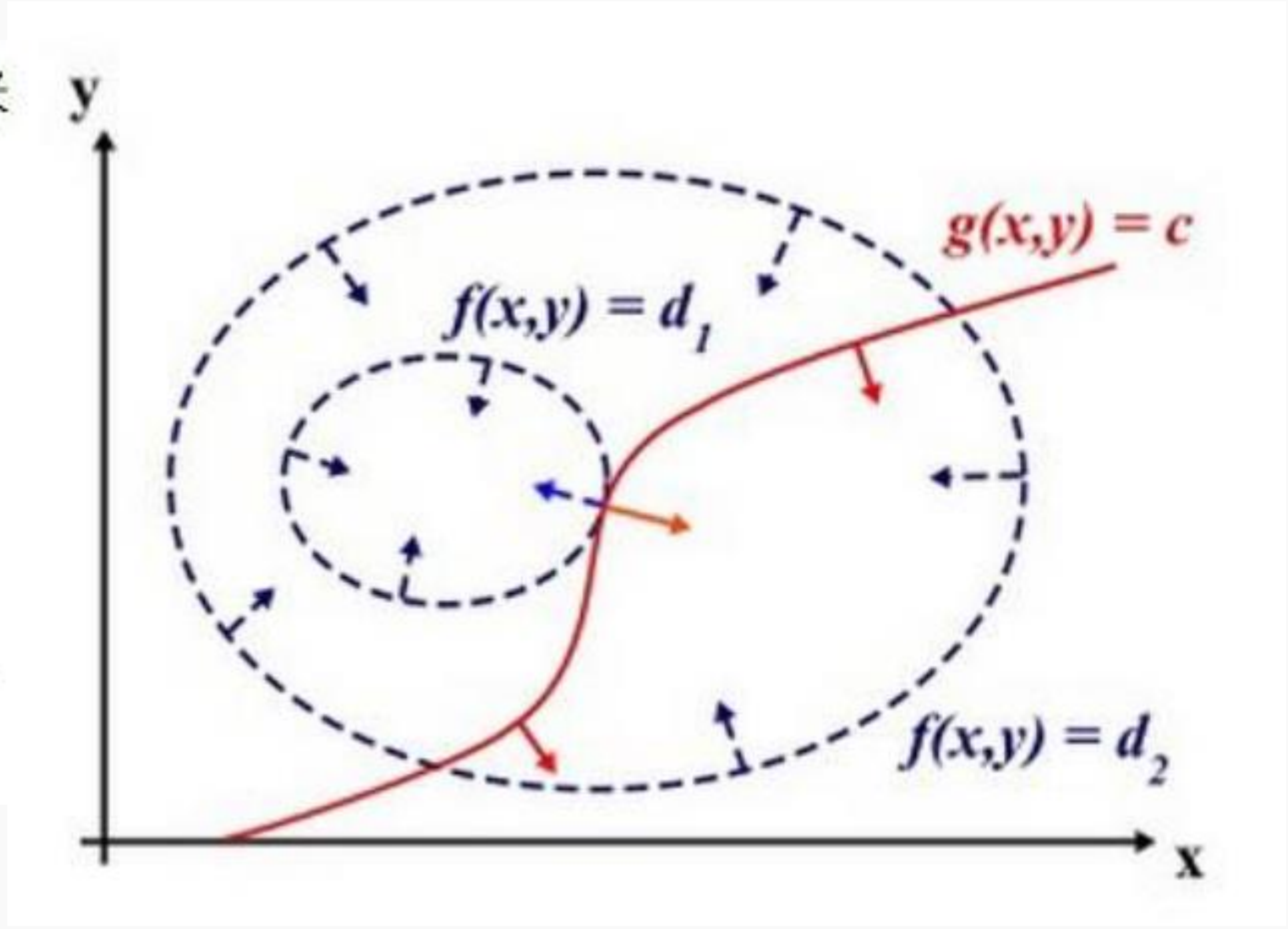
则
$$\begin{cases} F_x' = 3x^2 y^2 z + \lambda = 0 \\ F_y' = 2x^3 yz + \lambda = 0 \\ F_z' = x^3 y^2 + \lambda = 0 \\ x + y + z = 12 \end{cases} \Rightarrow 2x=3y,\ y=2z$$

解得唯一驻点 $(6,4,2)$,

故最大值为 $u_{\max} = 6^3 \cdot 4^2 \cdot 2 = 6912.$

拉格朗日乘子法的几何解释





广义拉格朗日乘子法

- 极值问题的一般形式

$$\begin{aligned} \text{minimize} \quad & f_0(x), \quad x \in \mathbf{R}^n \\ \text{subject to} \quad & f_i(x) \le 0, \quad i = 1, \ldots, m \\ & h_j(x) = 0, \quad j = 1, \ldots, p \end{aligned}$$

- 广义的拉格朗日函数

$$L(x, \lambda, \nu) = f_0(x) + \sum_{i=1}^{m} \lambda_i f_i(x) + \sum_{j=1}^{p} \nu_j h_j(x)$$

KKT条件

$$L(x,\lambda,\nu) = f_0(x) + \sum_{i=1}^{m} \lambda_i f_i(x) + \sum_{j=1}^{p} \nu_j h_j(x)$$

$$\begin{aligned}
f_i(x^\star) &\le 0, \quad i = 1,\dots,m \\
h_i(x^\star) &= 0, \quad i = 1,\dots,p \\
\lambda_i^\star &\ge 0, \quad i = 1,\dots,m \\
\lambda_i^\star f_i(x^\star) &= 0, \quad i = 1,\dots,m
\end{aligned}$$

$$\nabla f_0(x^\star) + \sum_{i=1}^{m} \lambda_i^\star \nabla f_i(x^\star) + \sum_{i=1}^{p} \nu_i^\star \nabla h_i(x^\star) = 0$$

支持向量机

拉格朗日函数

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

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

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

拉格朗日对偶函数

$$\begin{aligned}\mathcal{L}(w,b,\alpha)&=\frac{1}{2}\sum_{i,j=1}^{n}\alpha_i\alpha_j y_i y_j x_i^T x_j-\sum_{i,j=1}^{n}\alpha_i\alpha_j y_i y_j x_i^T x_j-b\sum_{i=1}^{n}\alpha_i y_i+\sum_{i=1}^{n}\alpha_i\\&=\sum_{i=1}^{n}\alpha_i-\frac{1}{2}\sum_{i,j=1}^{n}\alpha_i\alpha_j y_i y_j x_i^T x_j\end{aligned}$$

原问题的对偶问题

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

$$s.t.\ \alpha_i \geq 0, i = 1, 2, \cdots, n$$

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

近似线性可分

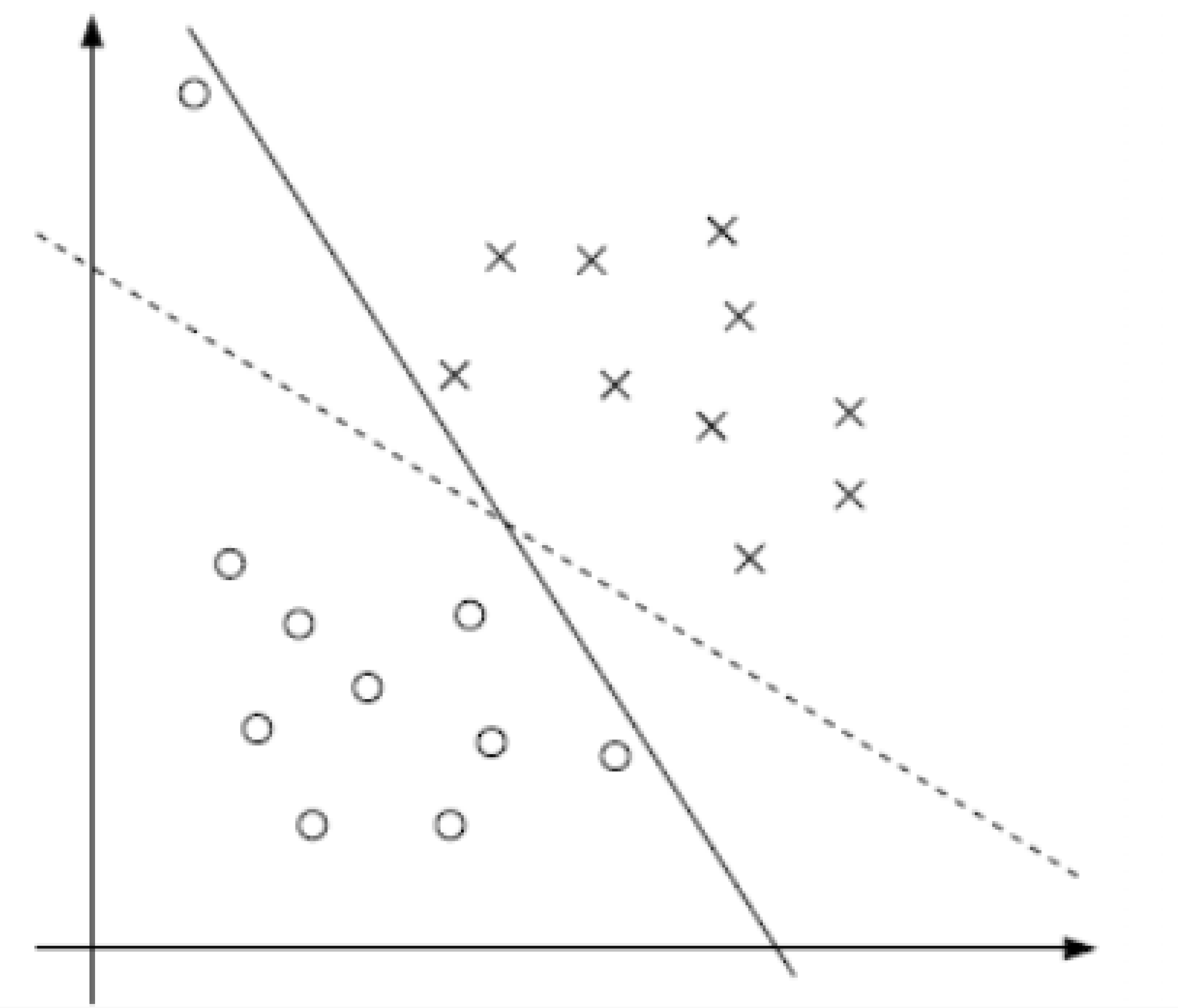
支持向量机

间隔的含义

- 代表健壮性
- 硬间隔和软间隔

完美分离的平面未必最好

数据本身不是线性可分



支持向量机

松弛变量和惩罚因子C

C代表对分错的数据点的惩罚，C越大，惩罚越重，分错的数据点就越少，容易过拟合；C越小，惩罚越小，分错的数据点越多，精度下降

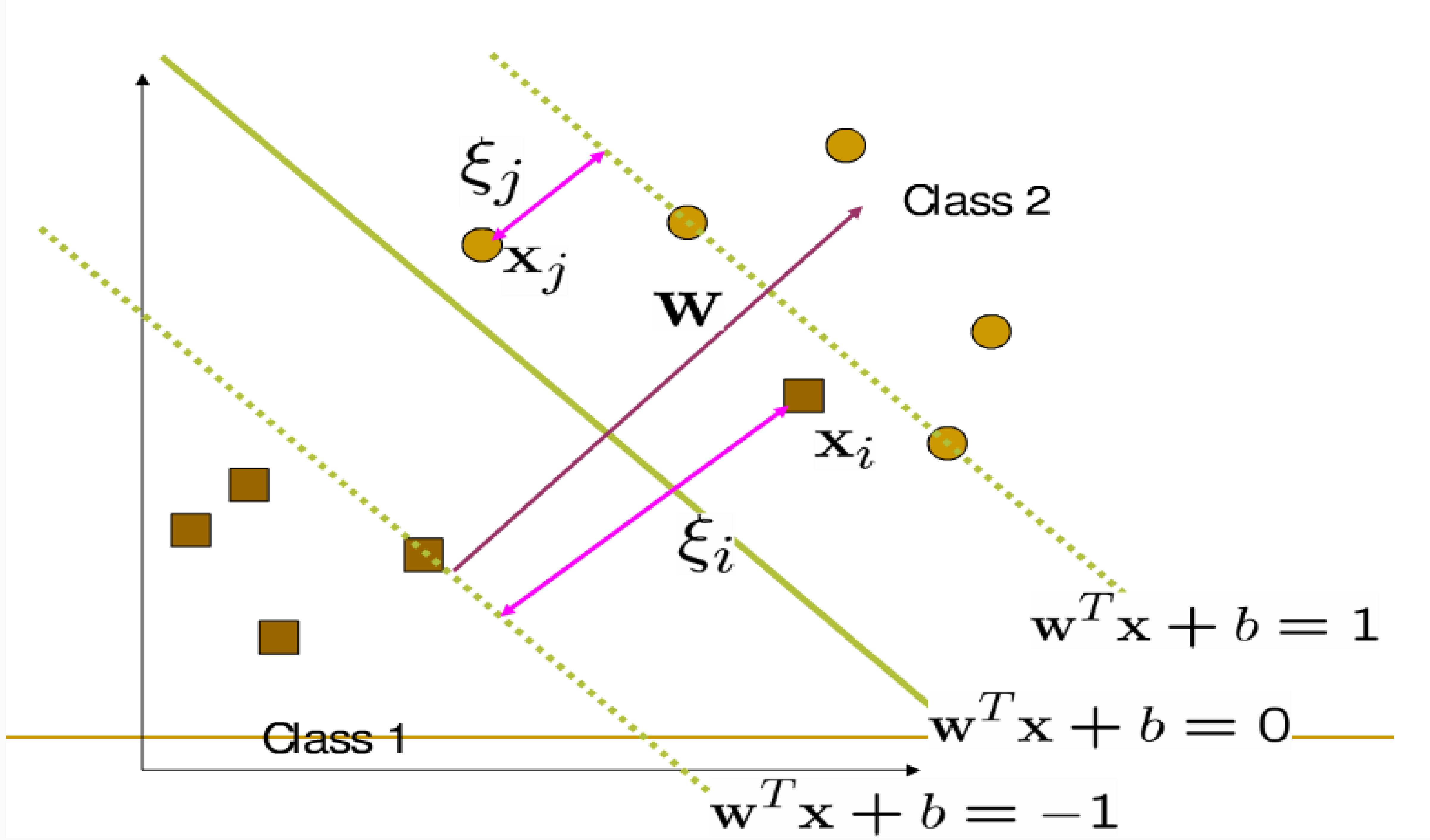
C->∞等价于硬间隔；

目标函数和约束条件

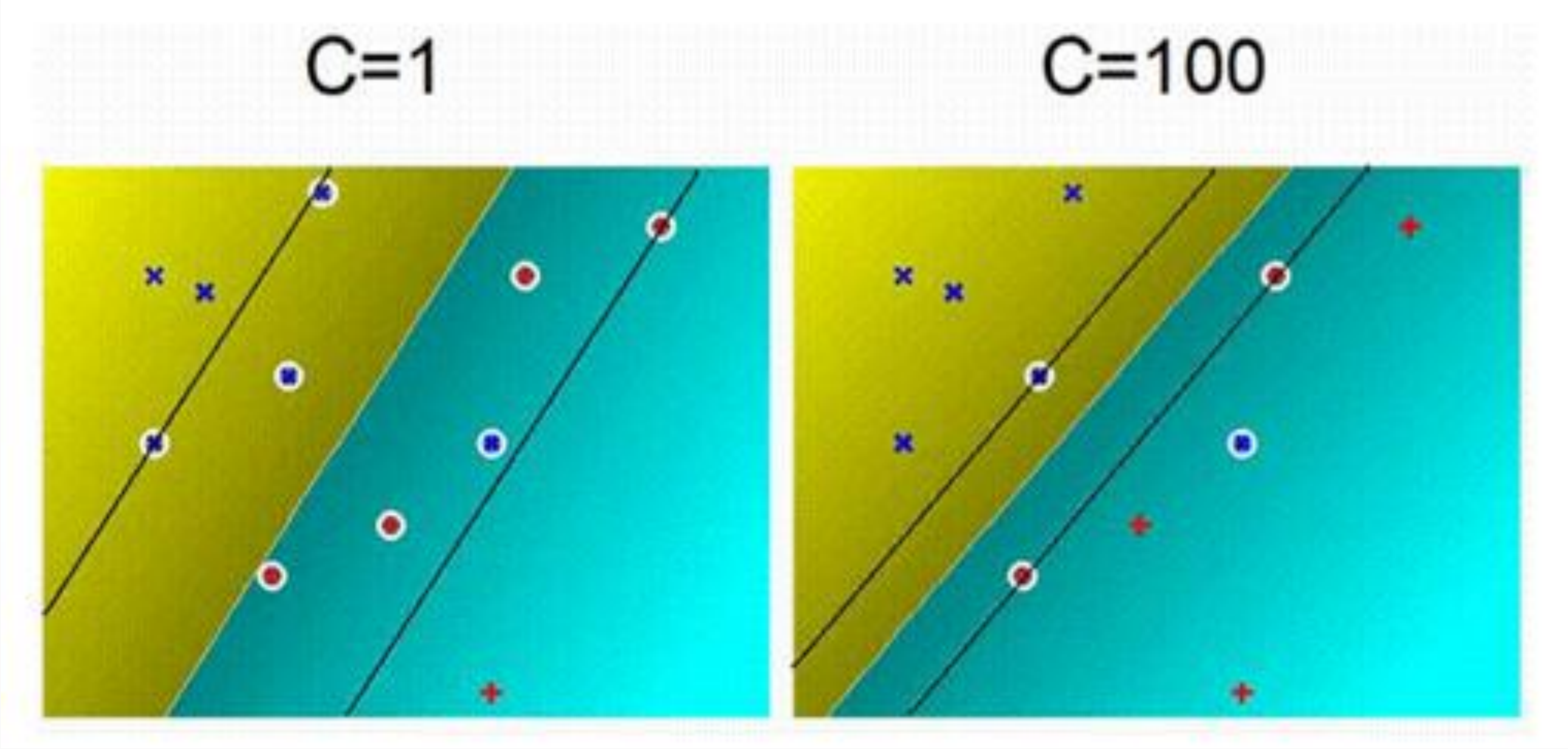
$$\min \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n}\xi_i$$

$$s.t., y_i(w^T x_i + b) \geq 1 - \xi_i, i = 1, \ldots, n$$

$$\xi_i \geq 0, i = 1, \ldots, n$$



不同惩罚因子的含义



硬间隔和软间隔

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

$$s.t.\ \alpha_i \geq 0, i = 1, 2, \cdots, n$$

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

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

$$s.t., 0 \leq \alpha_i \leq C, i = 1, \ldots, n$$

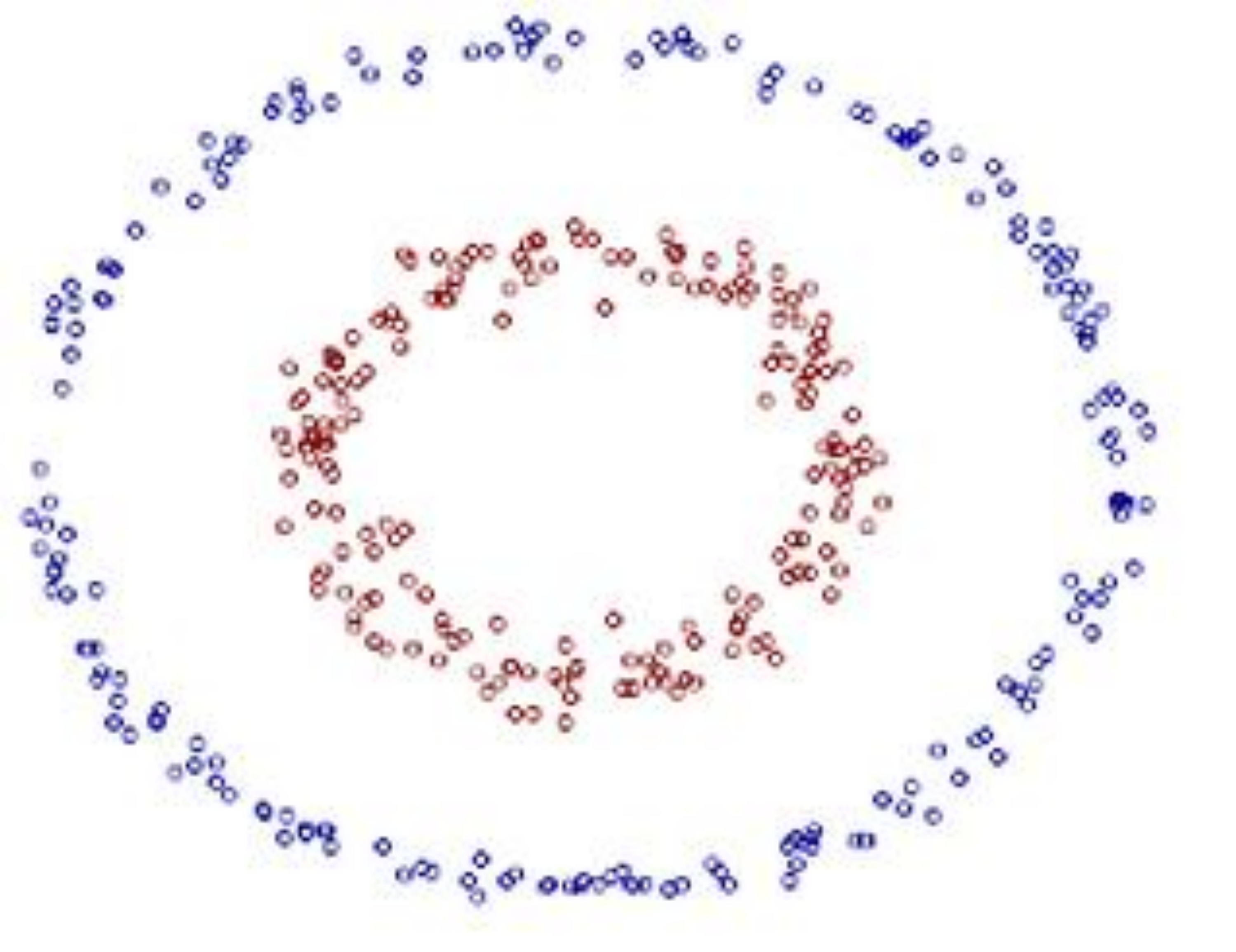
$$\sum_{i=1}^{n} \alpha_i y_i = 0$$

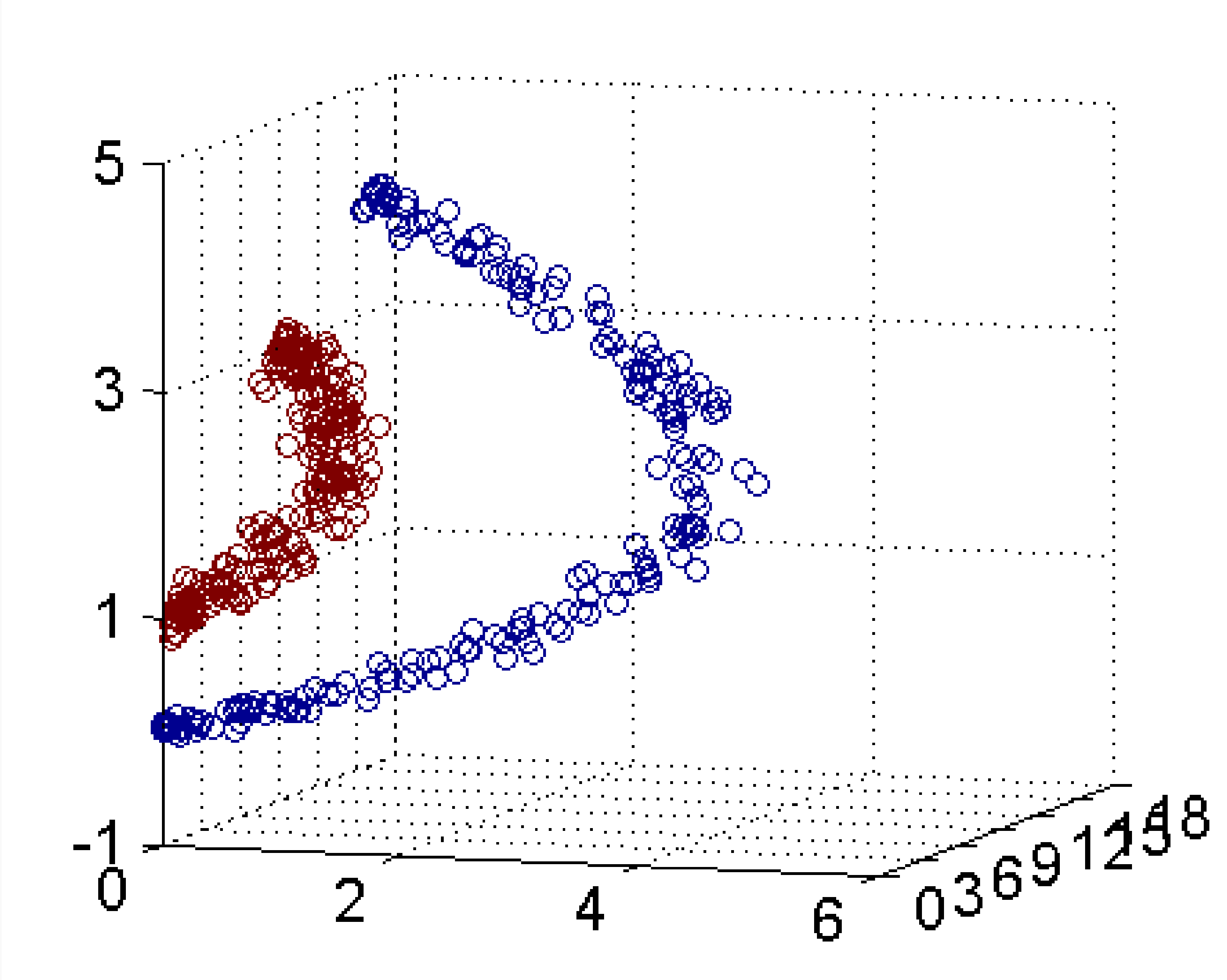
非线性可分和核函数

核函数和映射

低维空间线性不可分，映射到高维空间变成线性可分

寻找映射函数?





支持向量机

高维空间的向量点积

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

$$s.t.\ \alpha_i \geq 0, i = 1, 2, \cdots, n$$

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

$$\max_{\alpha} \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j \kappa(\boldsymbol{x}_i, \boldsymbol{x}_j)$$

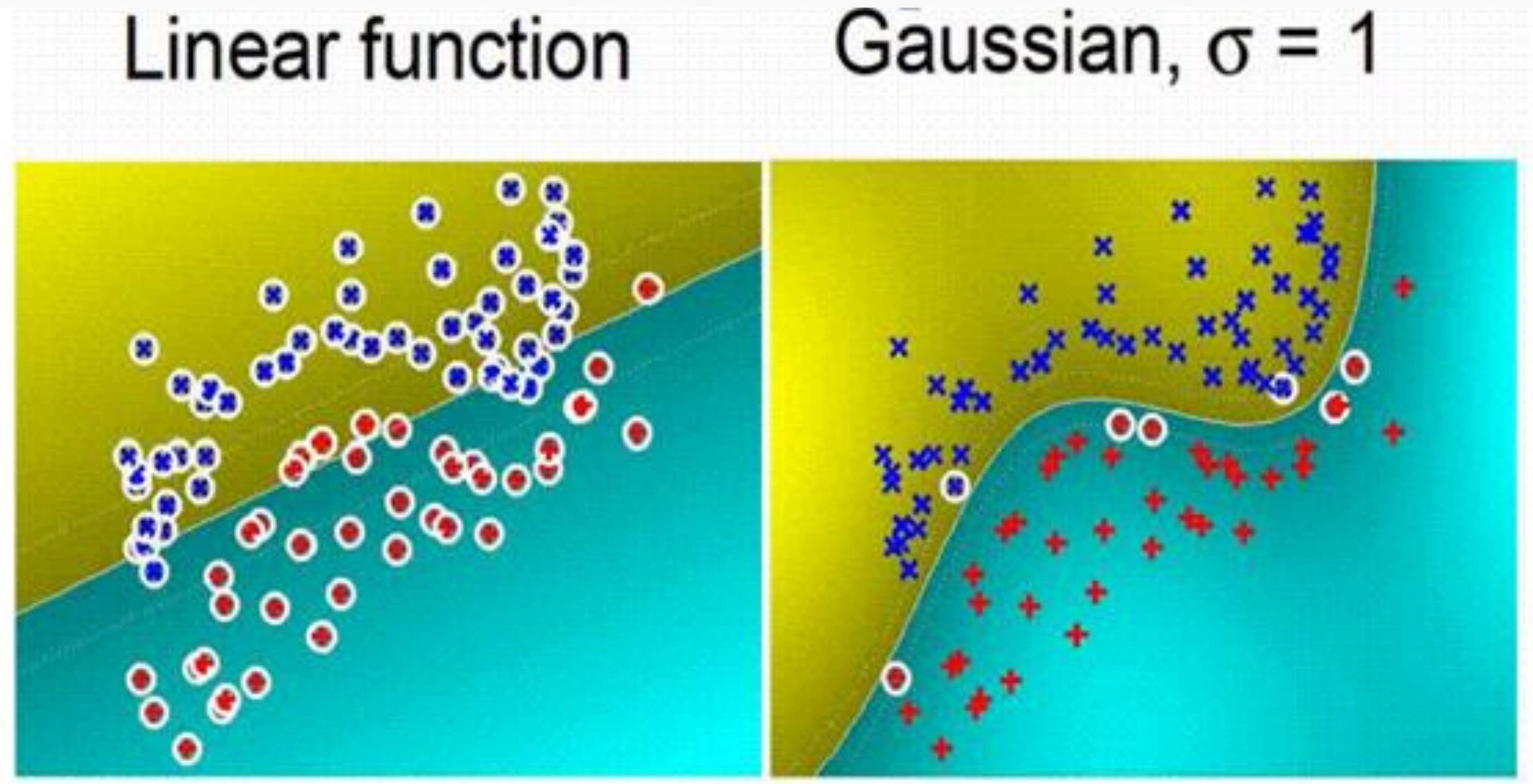
$$s.t.,\ \alpha_i \geq 0, i = 1, \ldots, n$$

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

支持向量机

核函数和映射没有关系

核函数只是一种运算技巧，用来计算映射到高维空间之后的内积的一种简便方法。实际中我们根本就不知道映射到底是什么形式的。



常见核函数

高斯核（径向基核，RBF）

指数核

多项式核

http://crsouza.com/2010/03/kernel-functions-for-machine-learning-applications/#linearhttp://www.zhihu.com/question/24627666

支持向量机

引入核之后的分类

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

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

$$\begin{aligned} w^T x + b &= \left(\sum_{i=1}^{m} \alpha_i y^{(i)} x^{(i)} \right)^T x + b \\ &= \sum_{i=1}^{m} \alpha_i y^{(i)} \langle x^{(i)}, x \rangle + b. \end{aligned}$$

SMO

优化问题

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

$$s.t.\ \alpha_i \geq 0, i = 1, 2, \cdots, n$$

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

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

$$s.t.,\ \alpha_i \geq 0, i = 1, \ldots, n$$

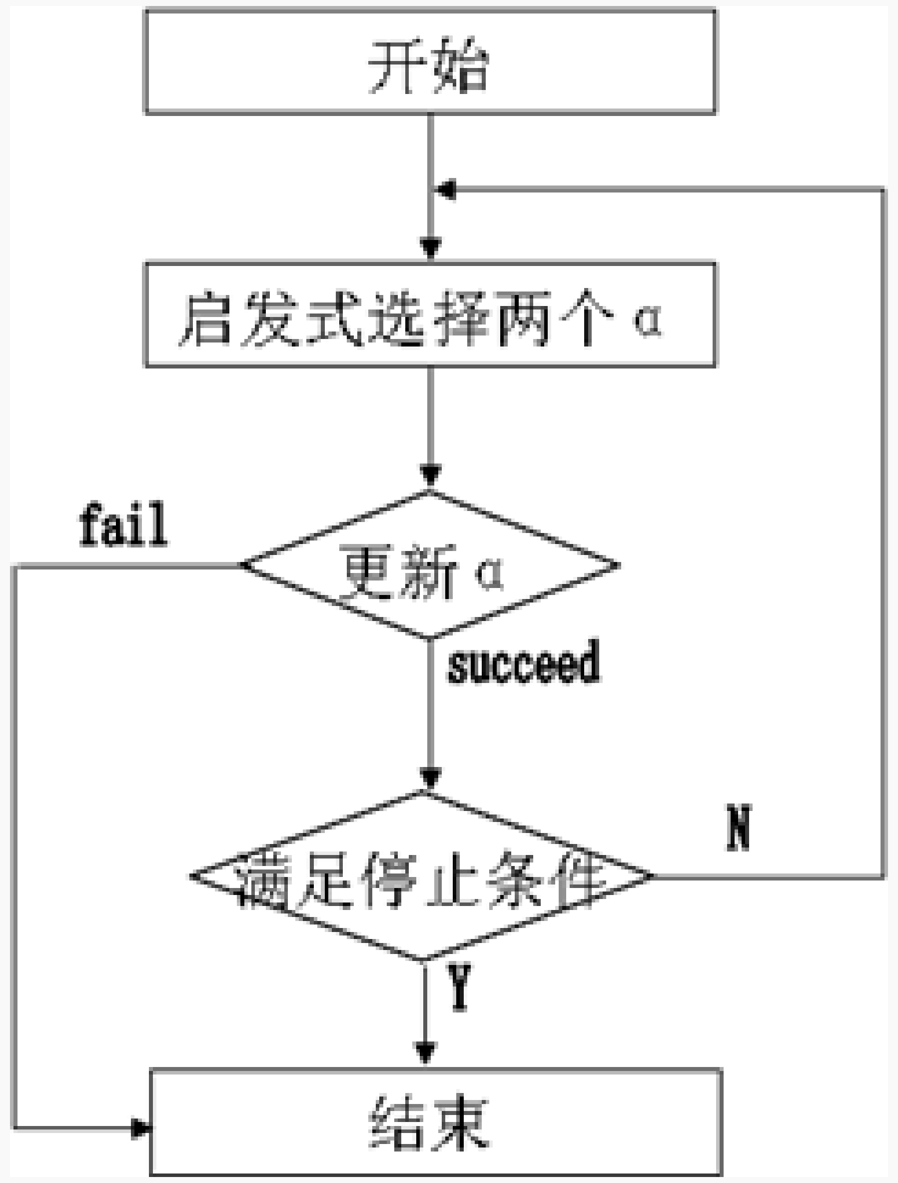
$$\sum_{i=1}^{n} \alpha_i y_i = 0$$

梯度法?

支持向量机

SMO算法由Microsoft Research的John C. Platt在1998年提出，并成为最快的二次规划优化算法，特别针对线性SVM和数据稀疏时性能更优。

第一步选取一对参数，选取方法使用启发式方法（Maximal violating pair）。第二步，固定除被选取的参数之外的其他参数，确定W极值。



坐标法上升法原理

$$\max_{\alpha} W(\alpha_1, \alpha_2, \ldots, \alpha_m).$$

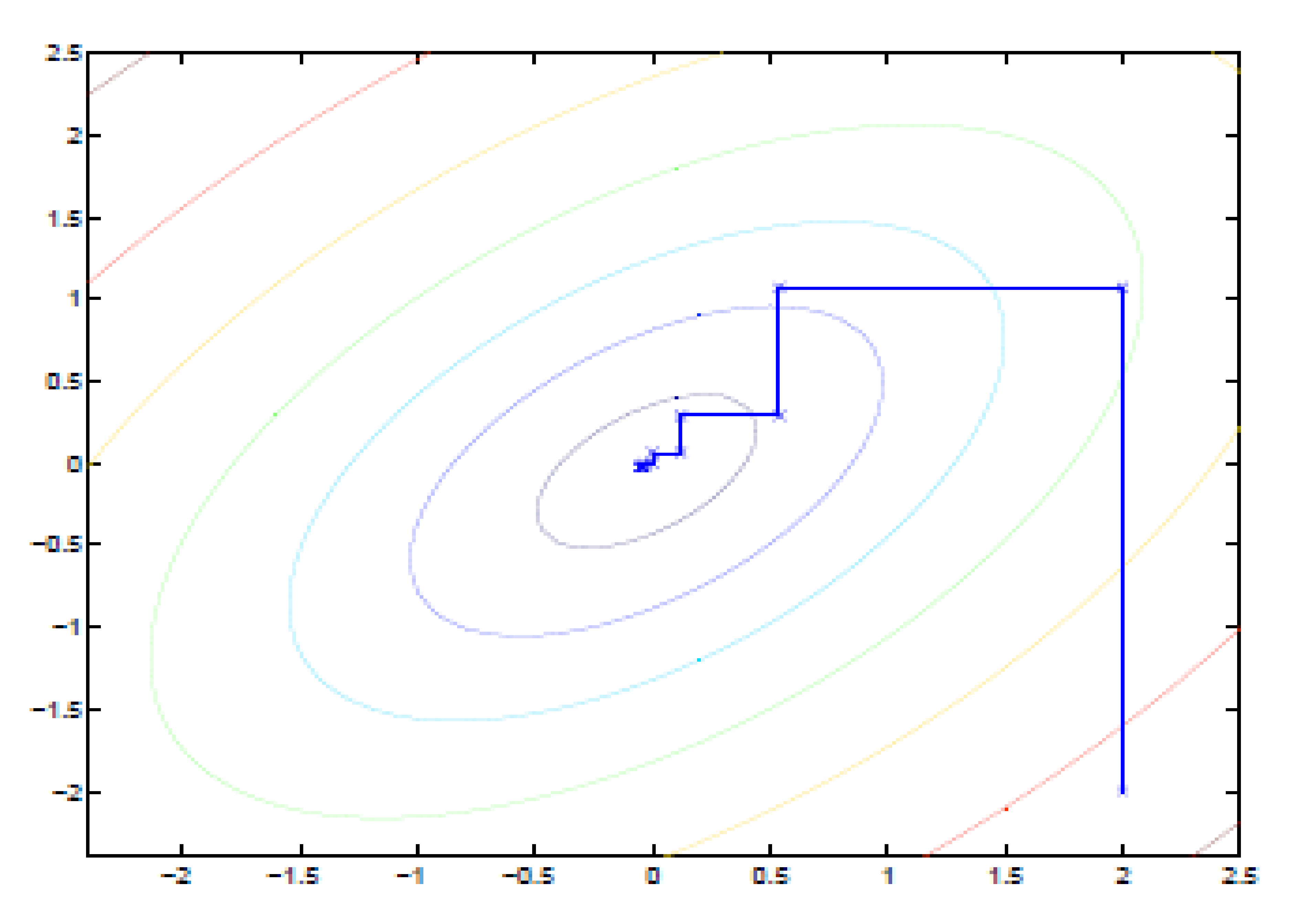
Loop until convergence: {

For $i = 1, \ldots, m$, {

$\alpha_i := \arg\max_{\hat{\alpha}_i} W(\alpha_1, \ldots, \alpha_{i-1}, \hat{\alpha}_i, \alpha_{i+1}, \ldots, \alpha_m).$

}

}



SMO算法

一次选取两个参数做优化，比如α_i和α_j，此时可以由和其他参数表示出来。

$$\sum_{i=1}^{N} y_i \alpha_i = 0. \quad \Rightarrow \quad \alpha_1 y^{(1)} = -\sum_{i=2}^{m} \alpha_i y^{(i)}.$$

伪代码

重复下面过程直到收敛{

选择两个拉格朗日乘子 α_i和α_j；

固定其他拉格朗日乘子α_k(k不等于i和j)，只对α_i和α_j优化，w(α);

根据优化后的α_i和α_j，更新截距b的值；

}

迭代的停止条件是a_j基本没有改变，或者总的迭代次数达到了迭代次数上限

启发式选择α_i和α_j

违反KKT条件

$$
\begin{aligned}
\alpha_i = 0 \quad &\Rightarrow \quad y^{(i)}(w^T x^{(i)} + b) \geq 1 \\
\alpha_i = C \quad &\Rightarrow \quad y^{(i)}(w^T x^{(i)} + b) \leq 1 \\
0 < \alpha_i < C \quad &\Rightarrow \quad y^{(i)}(w^T x^{(i)} + b) = 1.
\end{aligned}
$$

- 两个支撑平面外的点，对应前面的系数$\alpha_{j\ i}$为0，
- 两个支撑平面之间的点，对应α_j为C，
- 两个支撑平面上的点，对应的α_j在0和C之间

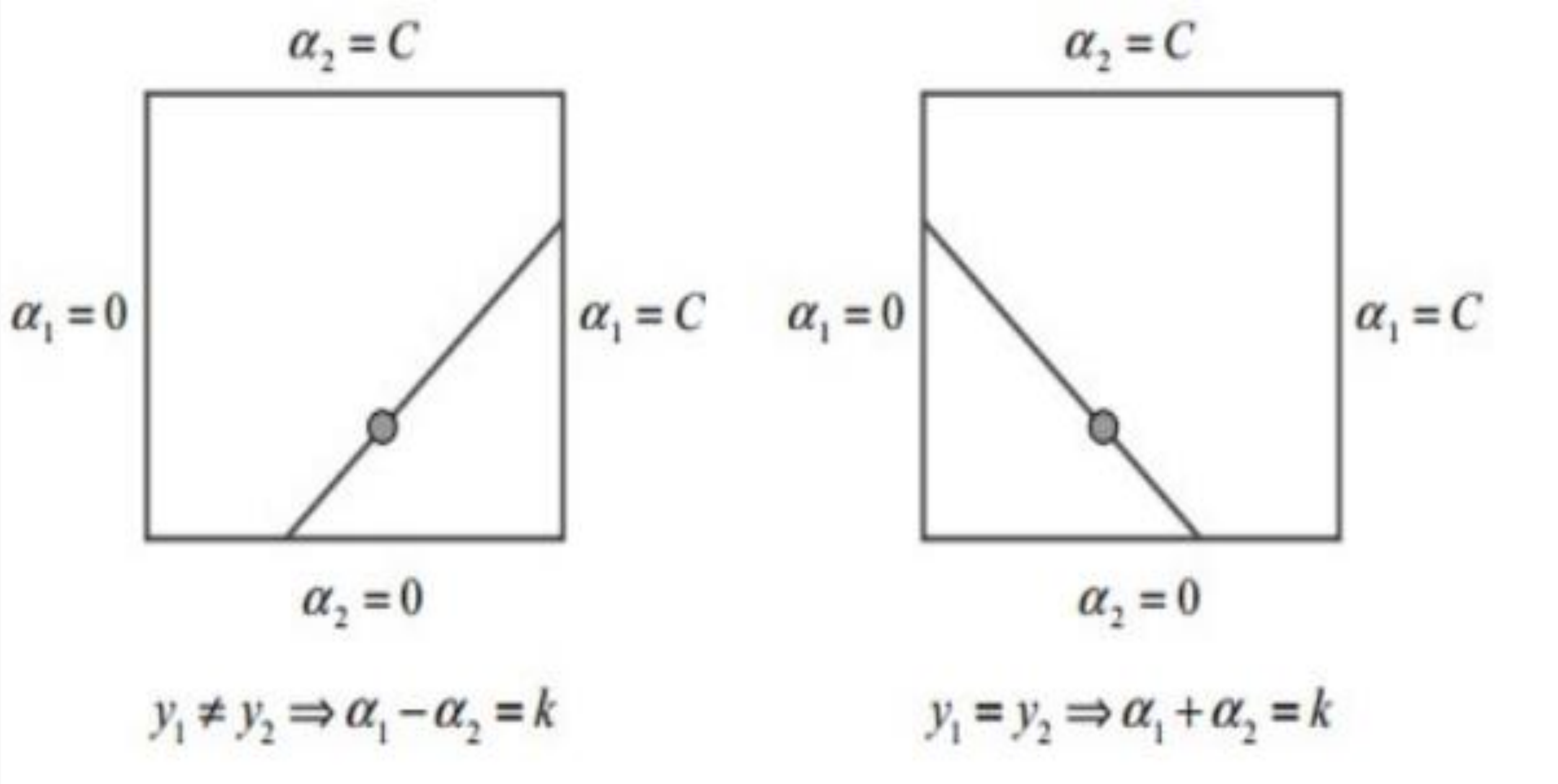
支持向量机

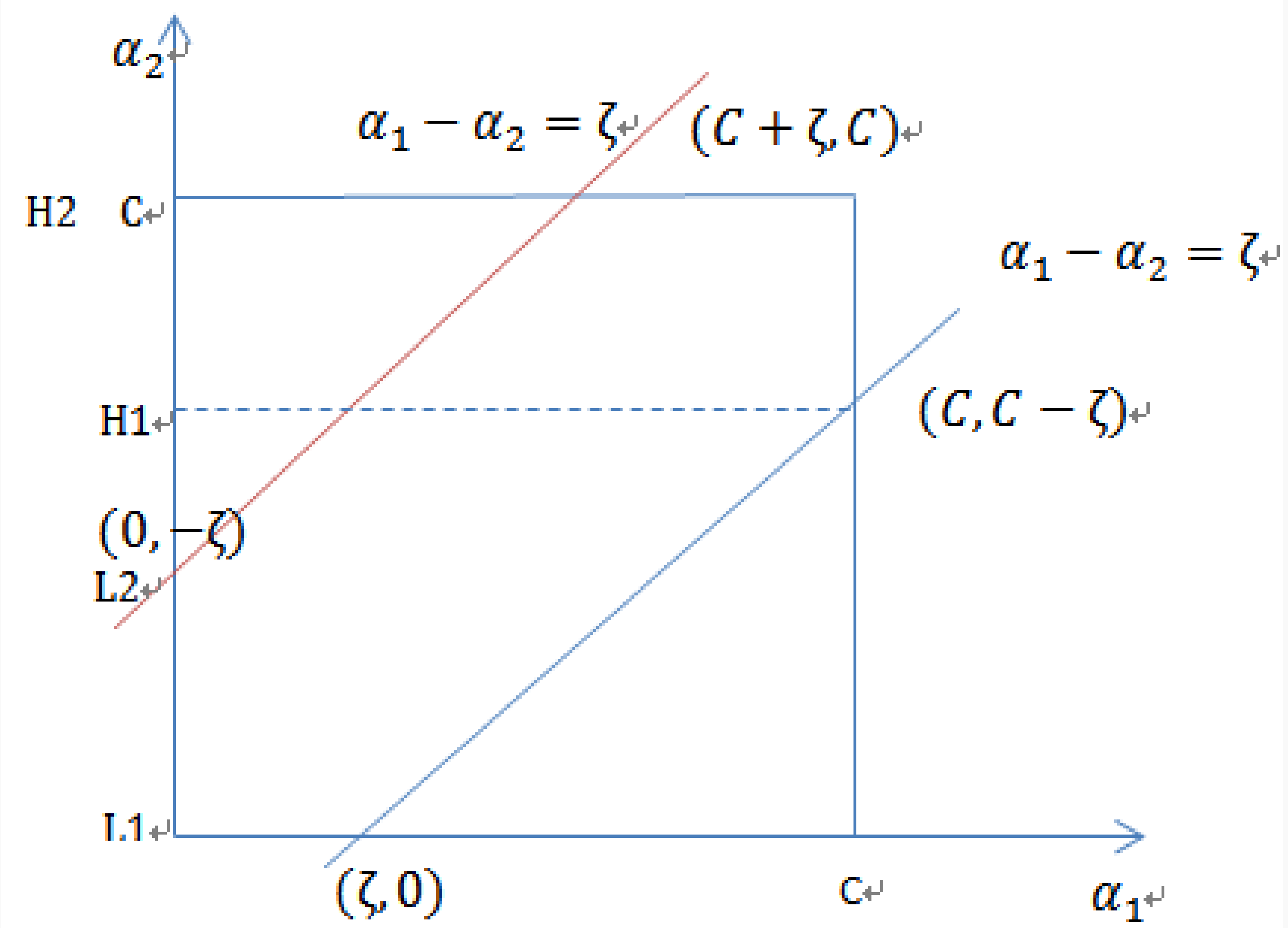
计算的α_i边界

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

$$\alpha_1^{new} y_1 + \alpha_2^{new} y_2 = \alpha_1^{old} y_1 + \alpha_2^{old} y_2 = \zeta$$

$$L = \max(0, \alpha_2^{old} - \alpha_1^{old}), H = \min(C, C + \alpha_2^{old} - \alpha_1^{old}) \quad \text{if } y_1 \neq y_2$$
$$L = \max(0, \alpha_2^{old} + \alpha_1^{old} - C), H = \min(C, \alpha_2^{old} + \alpha_1^{old}) \quad \text{if } y_1 = y_2$$





求解

$$\alpha_2^{\text{new}} = \alpha_2 + \frac{y_2(E_1 - E_2)}{\eta}$$

$$\eta = K(\vec{x}_1, \vec{x}_1) + K(\vec{x}_2, \vec{x}_2) - 2K(\vec{x}_1, \vec{x}_2).$$

$$\alpha_2^{\text{new,clipped}} = \begin{cases} H & \text{if} \quad \alpha_2^{\text{new}} \geq H; \\ \alpha_2^{\text{new}} & \text{if} \quad L < \alpha_2^{\text{new}} < H; \\ \mathrm{L} & \text{if} \quad \alpha_2^{\text{new}} \leq L. \end{cases}$$

$$\alpha_1^{new} = \alpha_1^{old} + y_1 y_2(\alpha_2^{old} - \alpha_2^{new})$$

求解b

$$b = \begin{cases} b_1 & \text{if } 0 < \alpha_1^{new} < C \\ b_2 & \text{if } 0 < \alpha_2^{new} < C \\ (b_1 + b_2)/2 & \text{otherwise} \end{cases}$$

$$b_1^{new} = b^{old} - E_1 - y_1(\alpha_1^{new} - \alpha_1^{old})k(x_1, x_1) - y_2(\alpha_2^{new} - \alpha_2^{old})k(x_1, x_2)$$

$$b_2^{new} = b^{old} - E_2 - y_1(\alpha_1^{new} - \alpha_1^{old})k(x_1, x_2) - y_2(\alpha_2^{new} - \alpha_2^{old})k(x_2, x_2)$$