模式识别
Pattern Recognition

——与计算机视觉
and Computer Vision

樊超

大数据系统计算技术国家工程实验室@深圳大学

E-mail： chaofan996@szu.edu.cn

上周回顾：线性判别函数

线性可分训练集：$\{(\boldsymbol{x}_i, y_i)\}$，

其中$\boldsymbol{x}_i \in \mathbb{R}^{d\times 1}$，$y_i$={-1, +1}

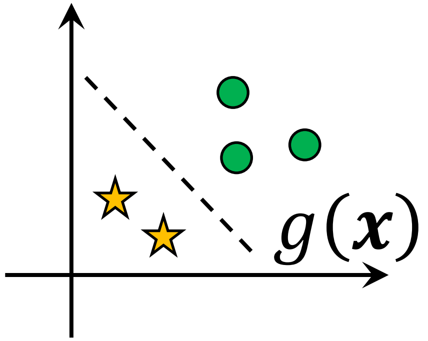
问题：

寻找分类器$g(\boldsymbol{x}) = \boldsymbol{w}^T\boldsymbol{x} + b$，要求

$y = +1$时$g(\boldsymbol{x}) > 0$；

$y = -1$时$g(\boldsymbol{x}) < 0$

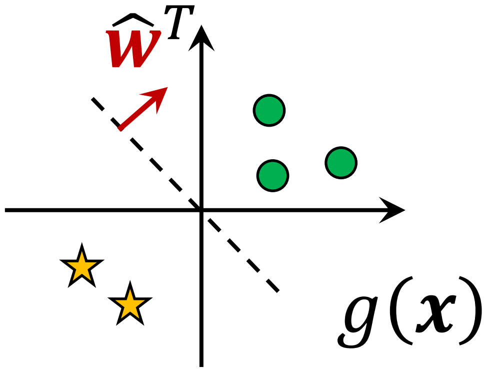
$$g(\boldsymbol{x}) = \boldsymbol{w}^T\boldsymbol{x} + b$$



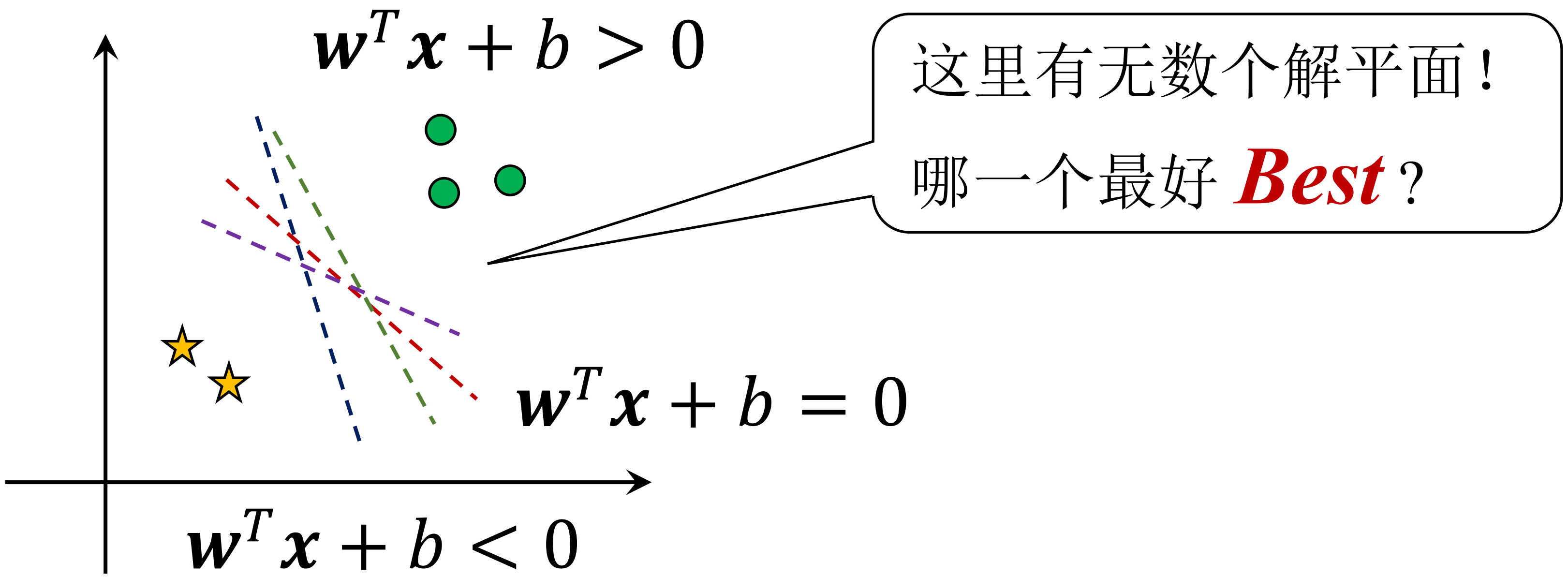
$$\widehat{\boldsymbol{w}}^T = [b, w_1, \ldots]$$

$$\widehat{\boldsymbol{x}} = [1, x_1, \ldots]$$

$$g(\boldsymbol{x}) = \widehat{\boldsymbol{w}}^T\widehat{\boldsymbol{x}}$$

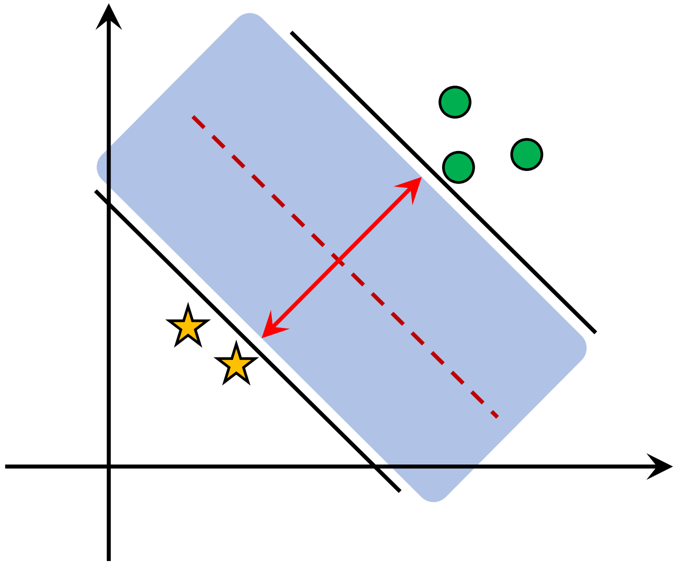


支持向量机：动机



$\boldsymbol{w}^T\boldsymbol{x} + b > 0$
这里有无数个解平面！
哪一个最好 Best？
$\boldsymbol{w}^T\boldsymbol{x} + b = 0$
$\boldsymbol{w}^T\boldsymbol{x} + b < 0$

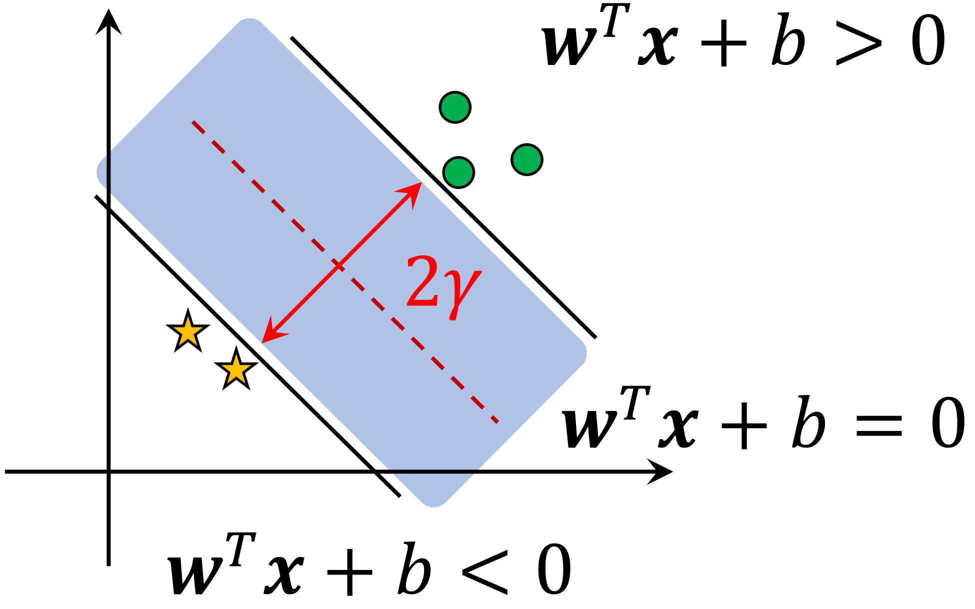
支持向量机：最大间隔思想



V. Vapnik’s Principle (1992)

The best classifier is the hyperplane with the maximum margin between two classes of data

支持向量机：最大间隔目标

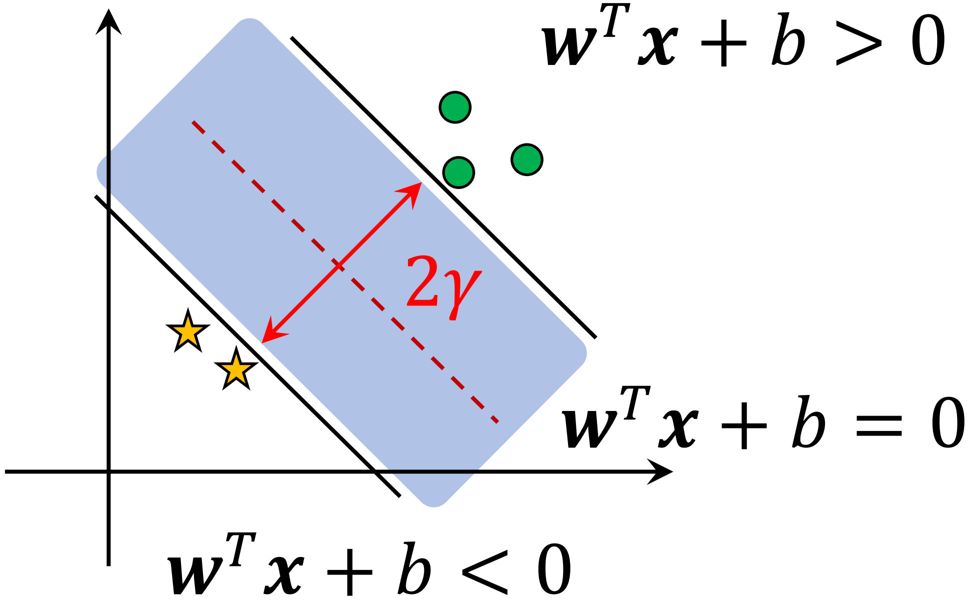


$$\max_{\boldsymbol{w},b,\gamma} \ 2\gamma$$

$$s.t. \quad \frac{\boldsymbol{w}^T\boldsymbol{x}_i + b}{\|\boldsymbol{w}\|_2} \geq \gamma \quad \forall i, y_i = 1$$

$$\frac{\boldsymbol{w}^T\boldsymbol{x}_i + b}{\|\boldsymbol{w}\|_2} \leq -\gamma \quad \forall i, y_i = -1$$

支持向量机：最大间隔目标



$$\max_{\boldsymbol{w},b,\gamma} \quad 2\gamma$$

$$s.t. \quad y_i \frac{\boldsymbol{w}^T\boldsymbol{x}_i + b}{\|\boldsymbol{w}\|_2} \geq \gamma$$

支持向量机：最大间隔目标

$$\max_{\boldsymbol{w},b,\gamma} \ 2\gamma$$

$$s.t. \ y_i \frac{\boldsymbol{w}^T \boldsymbol{x}_i + b}{\|\boldsymbol{w}\|_2} \geq \gamma$$

$$\Rightarrow$$

$$\max_{\boldsymbol{w},b,\gamma} \ 2\gamma$$

$$y_i(\boldsymbol{w}^T \boldsymbol{x}_i + b) \geq 1$$

$$\|\boldsymbol{w}\|_2 \gamma = 1$$

$$\Rightarrow$$

$$\max_{\boldsymbol{w},b} \frac{2}{\|\boldsymbol{w}\|_2}$$

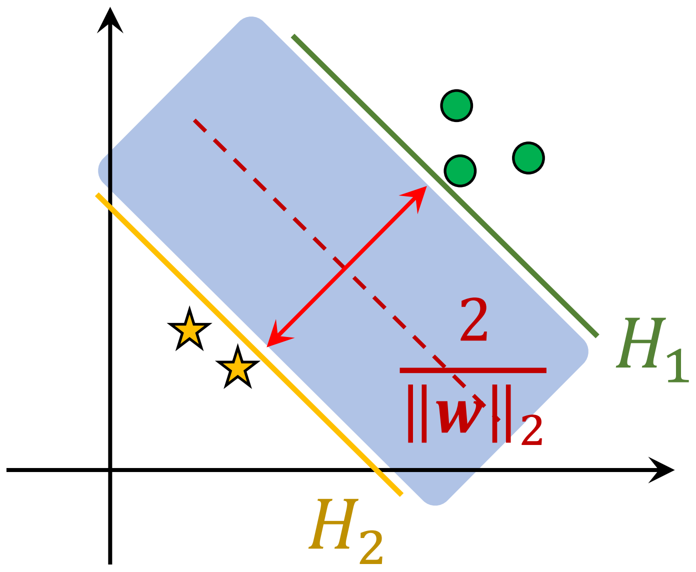
$$y_i(\boldsymbol{w}^T \boldsymbol{x}_i + b) \geq 1$$

支持向量机：最大间隔目标

$$\max_{\boldsymbol{w},b} \frac{2}{\|\boldsymbol{w}\|_2}$$

$$s.t.\quad y_i(\boldsymbol{w}^T\boldsymbol{x}_i + b) \geq 1$$

几何解释



H_1: $\boldsymbol{w}^T\boldsymbol{x} + b = 1$

H_2: $\boldsymbol{w}^T\boldsymbol{x} + b = -1$

支持向量机：最大间隔目标

$$\max_{\boldsymbol{w},b} \frac{2}{\|\boldsymbol{w}\|_2}$$

$$s.t. \quad y_i(\boldsymbol{w}^T\boldsymbol{x}_i + b) \geq 1$$

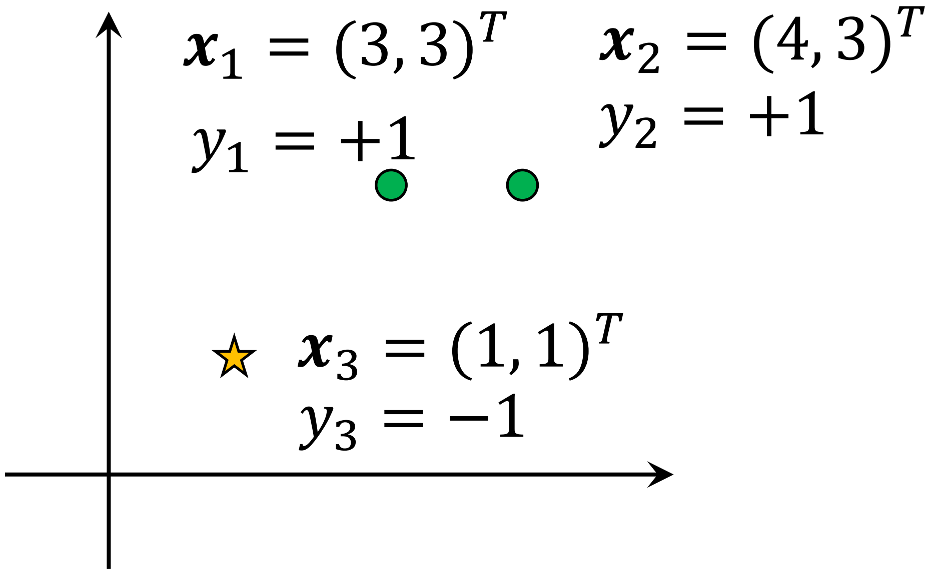
$\Rightarrow$

$$\min_{\boldsymbol{w},b} \frac{1}{2}\|\boldsymbol{w}\|_2$$

$$s.t. \quad y_i(\boldsymbol{w}^T\boldsymbol{x}_i + b) \geq 1$$

常见形式

支持向量机：算一个简单例子！



已知如图数据集，

试求最大间隔SVM $(\boldsymbol{w}, b)$.

$$\min_{\boldsymbol{w},b} \frac{1}{2} \|\boldsymbol{w}\|_2$$

$$s.t. \quad y_i(\boldsymbol{w}^T \boldsymbol{x}_i + b) \geq 1$$

支持向量机：学习的对偶算法

$$\min_{\boldsymbol{w},b} \frac{1}{2}\|\boldsymbol{w}\|_2$$

$$s.t.\quad y_i(\boldsymbol{w}^T\boldsymbol{x}_i + b) \geq 1$$

$$\Downarrow$$

构造拉格朗日函数

$$L(\boldsymbol{w}, b, \alpha) = \frac{1}{2}\|\boldsymbol{w}\|_2 - \sum_i \alpha_i(y_i(\boldsymbol{w}^T\boldsymbol{x}_i + b) - 1)$$

$$s.t.\quad \alpha_i \geq 0$$

支持向量机：学习的对偶算法

原问题

$$\min_{\boldsymbol{w}} \frac{1}{2}\|\boldsymbol{w}\|_2$$

$$s.t. \quad y_i(\boldsymbol{w}^T\boldsymbol{x}_i + b) \geq 1$$

$\Rightarrow$

对偶问题

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

$$s.t. \quad \alpha_i \geq 0$$

支持向量机：学习的对偶算法

$$\max_{\alpha} \min_{\boldsymbol{w},b} L(\boldsymbol{w}, b, \alpha) = \frac{1}{2}\|\boldsymbol{w}\|_2 - \sum_{i} \alpha_i \boxed{(y_i(\boldsymbol{w}^T \boldsymbol{x}_i + b) - 1)}$$

$$s.t.\ \ \alpha_i \geq 0$$

- 若满足原问题的约束条件（红色虚线框），则$\alpha_i \to 0$，即对偶问题等价于原问题
- 否则， $\alpha_i \to +\infty$，即无解

支持向量机：学习的对偶算法

$$\max_{\alpha} \min_{\boldsymbol{w},b} L(\boldsymbol{w}, b, \alpha) = \frac{1}{2}\|\boldsymbol{w}\|_2 - \sum_i \alpha_i(y_i(\boldsymbol{w}^T\boldsymbol{x}_i + b) - 1)$$

$$s.t.\ \ \alpha_i \geq 0$$

（1）先求$\min_{\boldsymbol{w},b} L(\boldsymbol{w}, b, \alpha)$

令 $\nabla_{\boldsymbol{w}} L(\boldsymbol{w}, b, \alpha) = \boldsymbol{w} - \sum_i \alpha_i y_i \boldsymbol{x}_i = 0 \Rightarrow \boldsymbol{w} = \sum_i \alpha_i y_i \boldsymbol{x}_i$

令 $\nabla_b L(\boldsymbol{w}, b, \alpha) = \sum_i \alpha_i y_i = 0$

支持向量机：学习的对偶算法

$$\max_{\alpha} \min_{\boldsymbol{w},b} L(\boldsymbol{w}, b, \alpha) = \frac{1}{2}\|\boldsymbol{w}\|_2 - \sum_i \alpha_i (y_i(\boldsymbol{w}^T \boldsymbol{x}_i + b) - 1)$$

$$s.t. \ \ \alpha_i \geq 0$$

$$\boldsymbol{w} = \sum_i \alpha_i y_i \boldsymbol{x}_i$$

$$\sum_i \alpha_i y_i = 0$$

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

$$L(\boldsymbol{w}, b, \alpha) = \frac{1}{2}\|\boldsymbol{w}\|_2 - \sum_i \alpha_i (y_i(\boldsymbol{w}^T \boldsymbol{x}_i + b) - 1)$$

$$= \frac{1}{2}\sum_i \sum_j \alpha_i \alpha_j y_i y_i \boldsymbol{x}_i{}^T \boldsymbol{x}_j - \sum_i \alpha_i \left(y_i (\sum_j \alpha_j y_j \boldsymbol{x}_j{}^T \boldsymbol{x}_i + b\right) - 1)$$

$$= -\frac{1}{2}\sum_i \sum_j \alpha_i \alpha_j y_i y_i \boldsymbol{x}_i{}^T \boldsymbol{x}_j + \sum_i \alpha_i$$

支持向量机：学习的对偶算法

$$\max_{\alpha}\min_{\boldsymbol{w},b} L(\boldsymbol{w},b,\alpha)=\frac{1}{2}\|\boldsymbol{w}\|_2-\sum_i \alpha_i(y_i(\boldsymbol{w}^T\boldsymbol{x}_i+b)-1)$$

$$s.t.\ \ \alpha_i \ge 0$$

（2）再求$\max\limits_{\alpha}\min\limits_{\boldsymbol{w},b} L(\boldsymbol{w},b,\alpha)$

$$\max_{\alpha}-\frac{1}{2}\sum_i\sum_j \alpha_i\,\alpha_j y_i y_j {\boldsymbol{x}_i}^T\boldsymbol{x}_j+\sum_i \alpha_i$$

$$s.t.\ \ \sum\nolimits_i \alpha_i y_i=0,\qquad \alpha_i \ge 0$$

支持向量机：学习的对偶算法

$$\max_{\alpha} \min_{\boldsymbol{w},b} L(\boldsymbol{w}, b, \alpha) = \frac{1}{2}\|\boldsymbol{w}\|_2 - \sum_i \alpha_i (y_i(\boldsymbol{w}^T \boldsymbol{x}_i + b) - 1)$$

$$s.t.\ \ \alpha_i \geq 0$$

（2）再求$\max\limits_{\alpha} \min\limits_{\boldsymbol{w},b} L(\boldsymbol{w}, b, \alpha)$

$$\min_{\alpha} \frac{1}{2}\sum_i \sum_j \alpha_i \alpha_j y_i y_j \boldsymbol{x}_i{}^T \boldsymbol{x}_j - \sum_i \alpha_i$$

$$s.t.\ \ \sum_i \alpha_i y_i = 0, \quad \alpha_i \geq 0$$

支持向量机：学习的对偶算法

$$\max_{\alpha} \min_{\boldsymbol{w},b} L(\boldsymbol{w}, b, \alpha) = \frac{1}{2}\|\boldsymbol{w}\|_2 - \sum_i \alpha_i (y_i(\boldsymbol{w}^T \boldsymbol{x}_i + b) - 1)$$

$$s.t.\ \ \alpha_i \geq 0$$

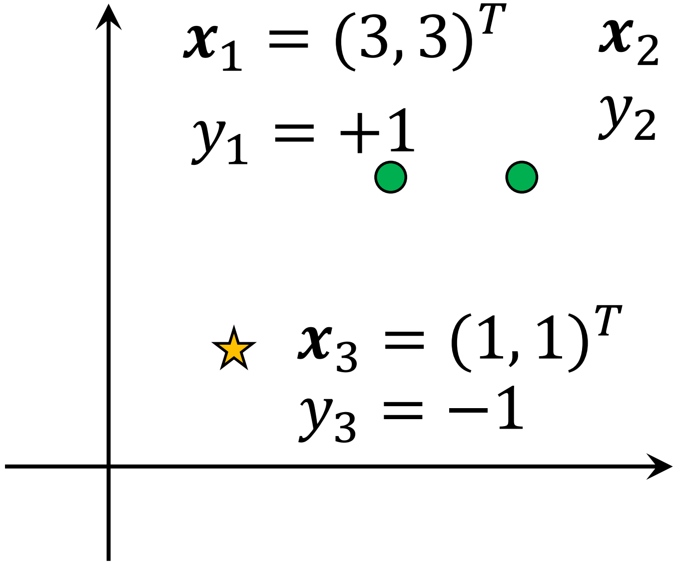
（3）最后找$\boldsymbol{w}^*$和b^*

令（2）中解为α^*，则：

$$\boldsymbol{w}^* = \sum_i \alpha_i^* y_i \boldsymbol{x}_i$$

$$b^* = y_j - \sum_i \alpha_i^* y_i ({\boldsymbol{x}_i}^T \boldsymbol{x}_j)$$

支持向量机：再算一下那个例子！



已知如图数据集，

试求最大间隔SVM $(\boldsymbol{w}, b)$.

$$\min_{\alpha} \frac{1}{2}\sum_i \sum_j \alpha_i \alpha_j y_i y_j \boldsymbol{x}_i{}^T \boldsymbol{x}_j - \sum_i \alpha_i$$

$$s.t. \ \sum_i \alpha_i y_i = 0, \quad \alpha_i \geq 0$$

支持向量机：学习的对偶算法

KKT（ Kuhn-Tucker ）条件：

$$\alpha_i(y_i(\boldsymbol{w}^T\boldsymbol{x}_i + b)) - 1) = 0$$

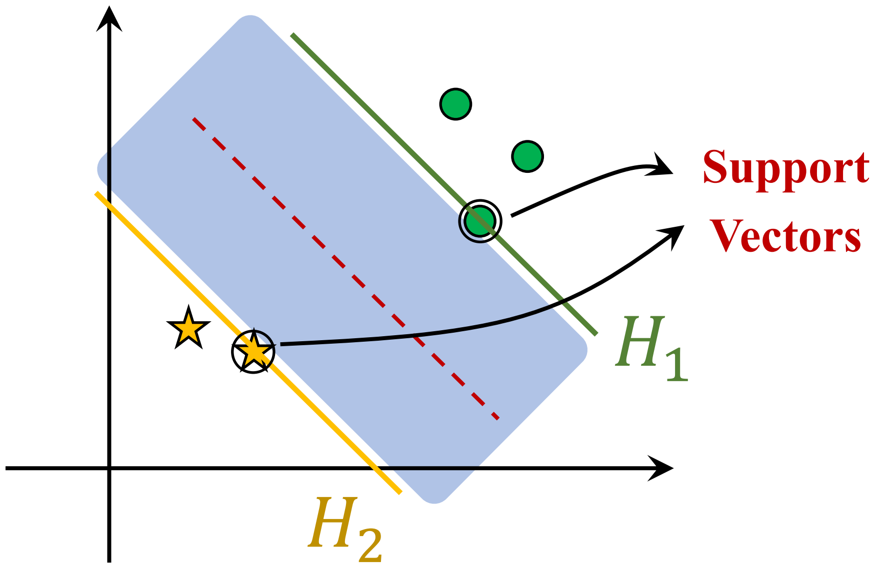
- $y_i(\boldsymbol{w}^T\boldsymbol{x}_i + b) > 1$，则$\alpha_i = 0$，

  此时$\boldsymbol{x}_i$远离决策面，不对发挥作用

- $y_i(\boldsymbol{w}^T\boldsymbol{x}_i + b) = 1$，则$\alpha_i > 0$，

  此时$\boldsymbol{x}_i$位于决策面上，直接决定决策面，

此时称$\boldsymbol{x}_i$为支持向量



H_1: $\boldsymbol{w}^T\boldsymbol{x} + b = 1$

H_2: $\boldsymbol{w}^T\boldsymbol{x} + b = -1$

支持向量机：学习的对偶算法

SVM的分类判别函数：

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

$$s.t.\quad \alpha_i \geq 0$$

$$g(\boldsymbol{x}) = \boldsymbol{w}^T\boldsymbol{x} + b$$

$$\boldsymbol{w}^* = \sum_i \alpha_i^* y_i \boldsymbol{x}_i$$

$$b^* = y_j - \sum_i \alpha_i^* y_i \boxed{(\boldsymbol{x}_i{}^T \boldsymbol{x}_j)}$$

$$g(\boldsymbol{x}) = \sum_i \alpha_i^* \boxed{\boldsymbol{x}_i^T\,\boldsymbol{x}} + b^*$$

与数据点的内积紧密相关！

松弛条件下的支持向量机

Hard Margin

$$\min_{\boldsymbol{w},b} \frac{1}{2}\|\boldsymbol{w}\|_2$$

$s.t.$

$$y_i(\boldsymbol{w}^T\boldsymbol{x}_i + b) \geq 1$$

$\Rightarrow$

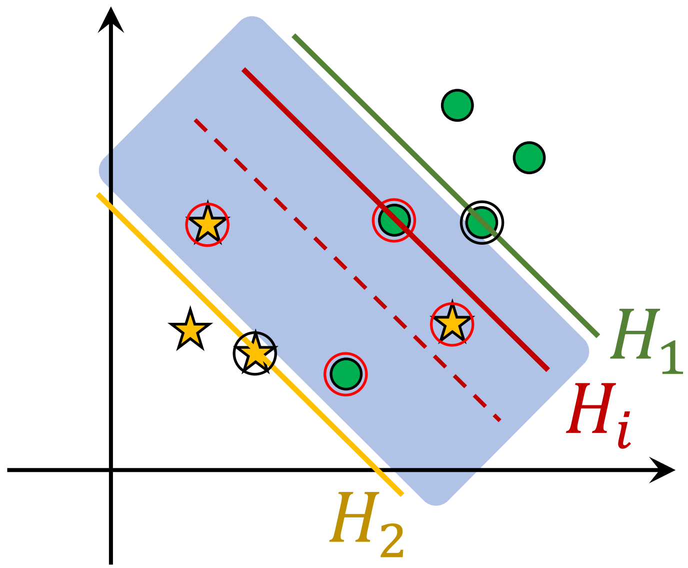
Soft Margin

$$\min_{\boldsymbol{w},b} \frac{1}{2}\|\boldsymbol{w}\|_2 + C\sum_i \xi_i$$

$s.t.$

$$y_i(\boldsymbol{w}^T\boldsymbol{x}_i + b) \geq 1 - \xi_i$$

$$\xi_i \geq 0$$



H_1: $\boldsymbol{w}^T\boldsymbol{x} + b = 1$

H_i: $\boldsymbol{w}^T\boldsymbol{x} + b = 1 - \xi_i$

H_2: $\boldsymbol{w}^T\boldsymbol{x} + b = -1$

ξ_i ：松弛变量（容忍限度，越大，则边界更宽松）
C ：惩罚系数（容忍意志，越大，则越不能容忍）

松弛条件下的支持向量机：学习的对偶算法

$$\min_{\boldsymbol{w},b} \frac{1}{2}\|\boldsymbol{w}\|_2 + C\sum_i \xi_i$$

$$s.t.\quad y_i(\boldsymbol{w}^T\boldsymbol{x}_i + b) \geq 1 - \xi_i$$

$$\xi_i \geq 0$$

⇩ 构造拉格朗日函数

$$L(\boldsymbol{w}, b, \xi, \alpha, \mu) = \frac{1}{2}\|\boldsymbol{w}\|_2 + C\sum_i \xi_i - \sum_i \alpha_i(y_i(\boldsymbol{w}^T\boldsymbol{x}_i + b) - 1 + \xi_i) - \mu_i\sum_i \xi_i$$

$$s.t.\ \alpha_i \geq 0,\quad \mu_i \geq 0$$

松弛条件下的支持向量机：学习的对偶算法

$$\max_{\alpha,\mu} \min_{\boldsymbol{w},b,\xi} L(\boldsymbol{w},b,\xi,\alpha,\mu) = \frac{1}{2}\|\boldsymbol{w}\|_2 + C\sum_i \xi_i - \sum_i \alpha_i(y_i(\boldsymbol{w}^T\boldsymbol{x}_i + b) - 1 + \xi_i) - \mu_i \sum_i \xi_i$$

$$s.t.\ \alpha_i \geq 0, \quad \mu_i \geq 0$$

令 $\nabla_{\boldsymbol{w}} L(\boldsymbol{w},b,\xi,\alpha,\mu) = 0$

令 $\nabla_b L(\boldsymbol{w},b,\xi,\alpha,\mu) = 0$

令 $\nabla_\xi L(\boldsymbol{w},b,\xi,\alpha,\mu) = 0$

$$\min_{\alpha} \frac{1}{2}\sum_i\sum_j \alpha_i \alpha_j y_i y_j {\boldsymbol{x}_i}^T \boldsymbol{x}_j - \sum_i \alpha_i$$

$$s.t.\ \ \textstyle\sum_i \alpha_i y_i = 0, \quad \boxed{0 \leq \alpha_i \leq C}$$

$$\Rightarrow \alpha^* \Rightarrow$$

$$\boldsymbol{w}^* = \sum_i \alpha_i^* y_i \boldsymbol{x}_i$$

$$b^* = y_j - \sum_i \alpha_i^* y_i({\boldsymbol{x}_i}^T \boldsymbol{x}_j)$$

松弛条件下的支持向量机：学习的对偶算法

KKT（ Kuhn-Tucker ）条件：

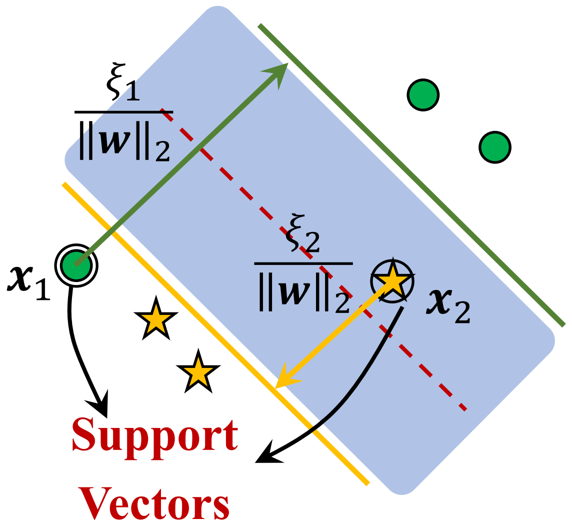
$$\alpha_i(y_i(\boldsymbol{w}^T\boldsymbol{x}_i + b)) - 1 + \xi_i) = 0$$

- $0 < \alpha_i < C$，则 $\xi_i = 0$，

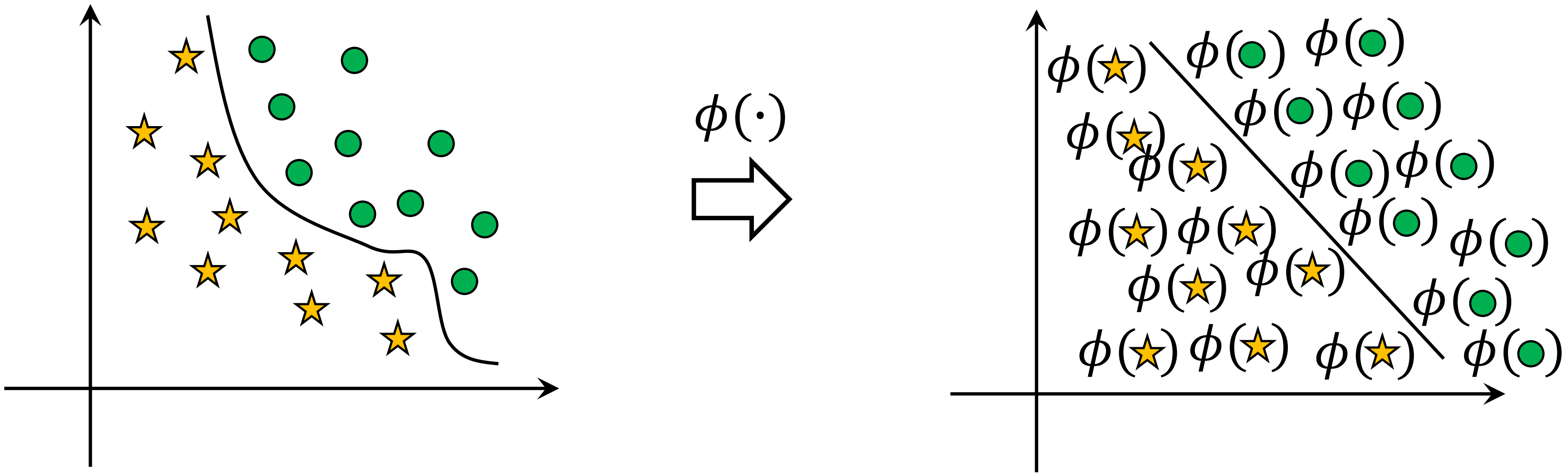
  此时$\boldsymbol{x}_i$为正确分类的支撑向量（位于决策边界）

- $\alpha_i = C$，则$\xi_i > 0$，

  此时$\boldsymbol{x}_i$为错误分类的支撑向量

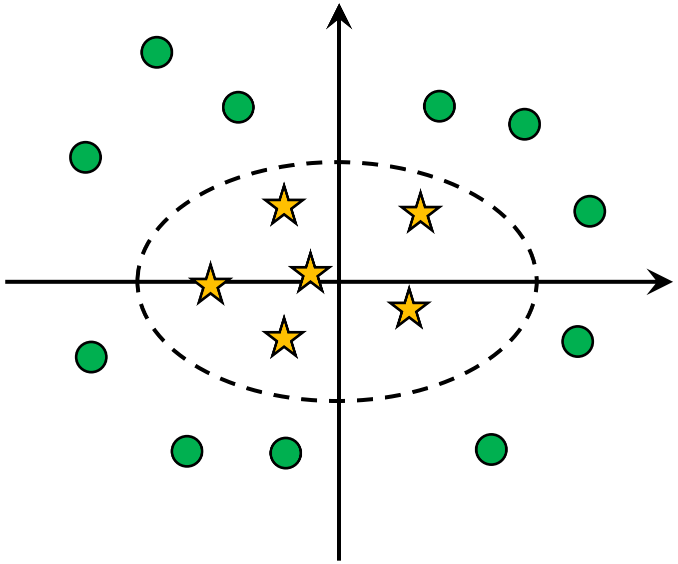


线性不可分条件下的支持向量机

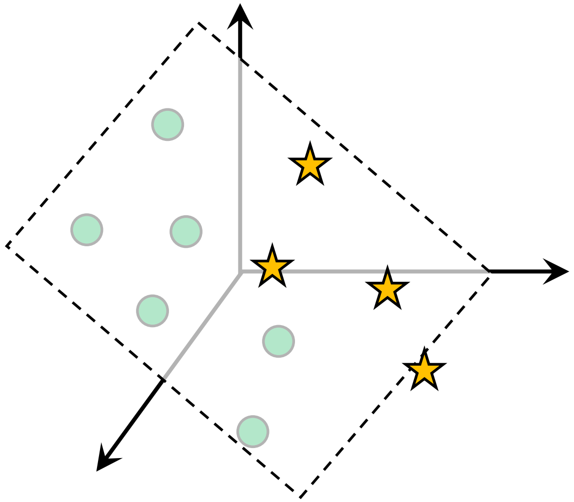


线性不可分条件下的支持向量机：核技巧

核心思路：低维不可分，则映射至高维试试



$\Phi: \boldsymbol{x} \rightarrow \phi(\boldsymbol{x})$



线性不可分条件下的支持向量机：核技巧

学习的对偶算法

$$g(\boldsymbol{x}) = \boldsymbol{w}^T \boldsymbol{x} + b \quad \Rightarrow \quad g(\boldsymbol{x}) = \sum_i \alpha_i^* \boldsymbol{x}_i^T \boldsymbol{x} + b^*$$

$\Downarrow$ 核技巧ϕ

$$g(\boldsymbol{x}) = \boldsymbol{w}^T \phi(\boldsymbol{x}) + b \quad \Rightarrow \quad g(\boldsymbol{x}) = \sum_i \alpha_i^* \boxed{\phi(\boldsymbol{x}_i^T)\,\phi(\boldsymbol{x})} + b^*$$

$$\Downarrow$$

$$\mathcal{K}(x_i, x_j) = \phi(x_i)\phi(x_j)$$

核函数：无需显式的ϕ，直接定义内积$\mathcal{K}$，避免（a）低维向高维投射的计算开销；（b）无限维的特征空间无法映射

线性不可分条件下的支持向量机：核技巧

显式映射 $\phi(\cdot)$ **VS.** 核技巧 $\mathcal{K}(\cdot,\cdot)$

第一步：2D 到 6D

$$\phi(\boldsymbol{x}) = \phi\left(\begin{bmatrix} x_1, \\ x_2 \end{bmatrix}\right) = \begin{bmatrix} 1, \\ \sqrt{2}x_1, \\ \sqrt{2}x_2, \\ x_1^2, \\ x_2^2, \\ \sqrt{2}x_1x_2 \end{bmatrix} \qquad \phi(\boldsymbol{y}) = \phi\left(\begin{bmatrix} y_1, \\ y_2 \end{bmatrix}\right) = \begin{bmatrix} 1, \\ \sqrt{2}y, \\ \sqrt{2}y_2, \\ y_1^2, \\ y_2^2, \\ \sqrt{2}y_1y_2 \end{bmatrix}$$

第二步：算内积

$$\phi(\boldsymbol{x})^{\boldsymbol{T}}\phi(\boldsymbol{y}) = (1 + x_1y_1 + x_2y_2)^2$$

直接定义
一步到位

$$\mathcal{K}(\boldsymbol{x},\boldsymbol{y}) = (1 + x_1y_1 + x_2y_2)^2$$

线性不可分条件下的支持向量机：核技巧

几个常见核函数

$\mathcal{K}(\boldsymbol{x}_1, \boldsymbol{x}_2) = \boldsymbol{x}_1^T \boldsymbol{x}_2$ 线性核

$\mathcal{K}(\boldsymbol{x}_1, \boldsymbol{x}_2) = \boldsymbol{x}_1^T \Sigma^{-1} \boldsymbol{x}_2$ （Σ^{-1} 为正定矩阵）

$\mathcal{K}(\boldsymbol{x}_1, \boldsymbol{x}_2) = (\boldsymbol{x}_1^T \boldsymbol{x}_2 + 1)^d$ 多项式核

$\mathcal{K}(\boldsymbol{x}_1, \boldsymbol{x}_2) = \exp(-\|\boldsymbol{x}_1 - \boldsymbol{x}_2\|_2^2 / 2\sigma^2)$ Gaussian核

线性不可分条件下的支持向量机

$$\min_{\alpha} \frac{1}{2}\sum_{i}\sum_{j} \alpha_i \alpha_j y_i y_j \mathcal{K}(\boldsymbol{x}_i, \boldsymbol{x}_j) - \sum_{i} \alpha_i$$

$$s.t. \quad \sum_i \alpha_i y_i = 0, \qquad 0 \le \alpha_i \le C$$

$$\Downarrow$$

$$g(\boldsymbol{x}) = \sum_{i} \alpha_i^* \mathcal{K}(\boldsymbol{x}_i, \boldsymbol{x}) + b^*$$

支持向量机：再回顾

$$\min_{\boldsymbol{w},b} \frac{1}{2}\|\boldsymbol{w}\|_2 + C\sum_i \xi_i$$

$$s.t. \quad y_i(\boldsymbol{w}^T\boldsymbol{x}_i + b) \geq 1 - \xi_i$$

$$\xi_i \geq 0$$

等价 $\Longleftrightarrow$

正则项　经验损失

$$\min_{\boldsymbol{w},b} \lambda\|\boldsymbol{w}\|_2 + \sum_i [1 - y_i(\boldsymbol{w}^T\boldsymbol{x}_i + b)]_+$$

其中：

$$[z]_+ = \begin{cases} z, z > 0 \\ 0, z \leq 0 \end{cases}$$

支持向量机：优化算法

经典的序列最小化最优化（Sequential Minimal Optimization，SMO）

同时，可以用梯度下降法：

$$\min_{\boldsymbol{w},b} L(\boldsymbol{w}, b) = \frac{1}{2}\|\boldsymbol{w}\|_2 + C\sum_i [1 - y_i(\boldsymbol{w}^T\boldsymbol{x}_i + b)]_+$$

$$\nabla_{\boldsymbol{w}} L_i = \begin{cases} \boldsymbol{w} - Cy_i\boldsymbol{x}_i, & 若 1 - y_i(\boldsymbol{w}^T\boldsymbol{x}_i + b) > 0 \\ \boldsymbol{w}, & 否则 \end{cases}$$

$$\nabla_b L_i = \begin{cases} -Cy_i, & 若 1 - y_i(\boldsymbol{w}^T\boldsymbol{x}_i + b) > 0 \\ 0, & 否则 \end{cases}$$

⇨

大小为 N 的数据集$\{(\boldsymbol{x}_i, y_i)\}$，
随机初始 $\boldsymbol{w}$，b，总迭代次数 E

for e in range(E):

  for i in range(N):

    if $1 - y_i(\boldsymbol{w}^T\boldsymbol{x}_i + b) > 0$:

      $\boldsymbol{w} \leftarrow \boldsymbol{w} - \eta\nabla_{\boldsymbol{w}} L_i$

      $b \leftarrow b - \eta\nabla_{\boldsymbol{b}} L_i$

return $\boldsymbol{w}, b$