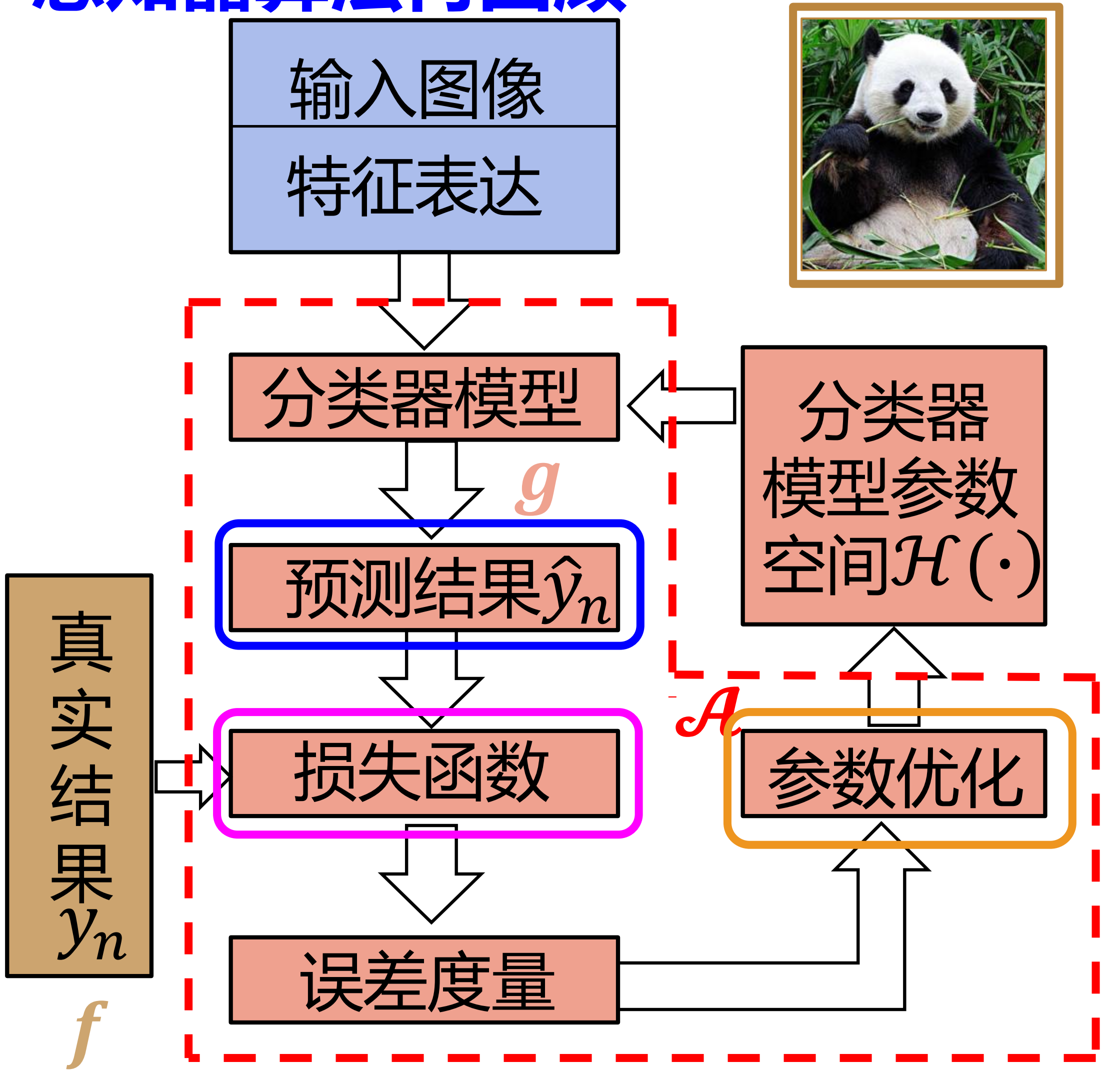
第七讲 线性支撑向量机（*Linear Support Vector Machine*）

7.1 最大间隔分类面(*Large-Margin Separating Hyperplane*)

7.2 标准的最大间隔问题(*Standard Large-Margin Problem*)

7.3 支撑向量机(*Support Vector Machine*)

感知器算法再回顾



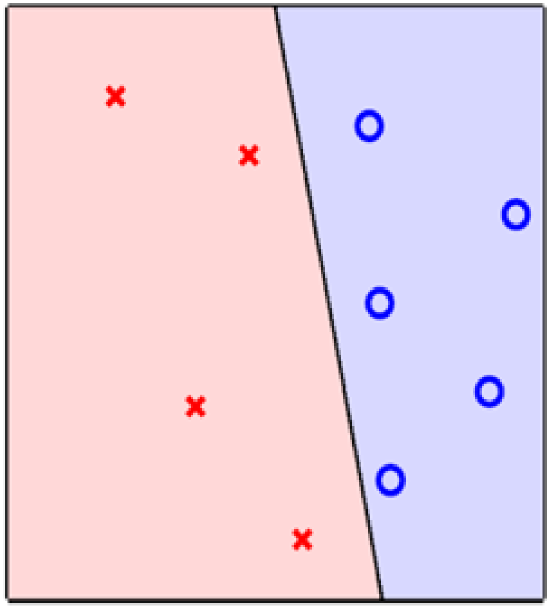
$$\hat{y}_{n(t)} = \text{sign}(\mathbf{w}_t^T \mathbf{x}_{n(t)})$$

算法收敛：

$$L_{in} = \sum_{n=1}^{N} [\![y_n \neq \hat{y}_n]\!] = 0$$

$$\mathbf{w}_{t+1} = \mathbf{w}_t + y_n \mathbf{x}_{n(t)}$$

线性可分

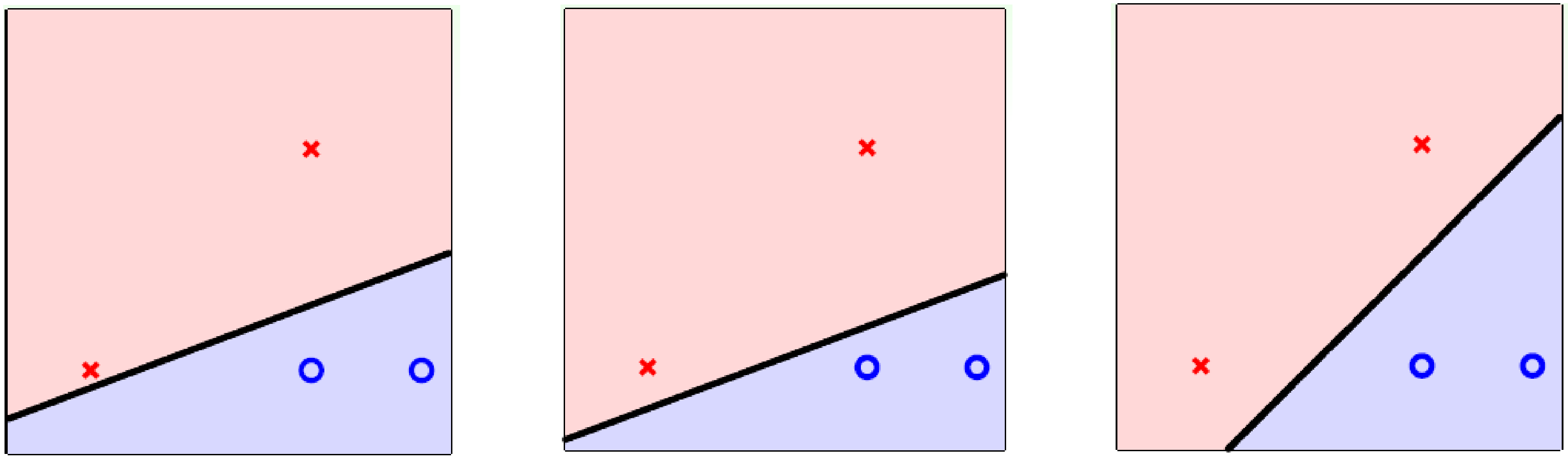


- 设置初始分类面(权重)$\mathbf{w}_0$
- 如果有样本分错，就修正权重

Ref.: NTU-LIN

7.1 最大间隔分类面

哪一个分类面最佳?



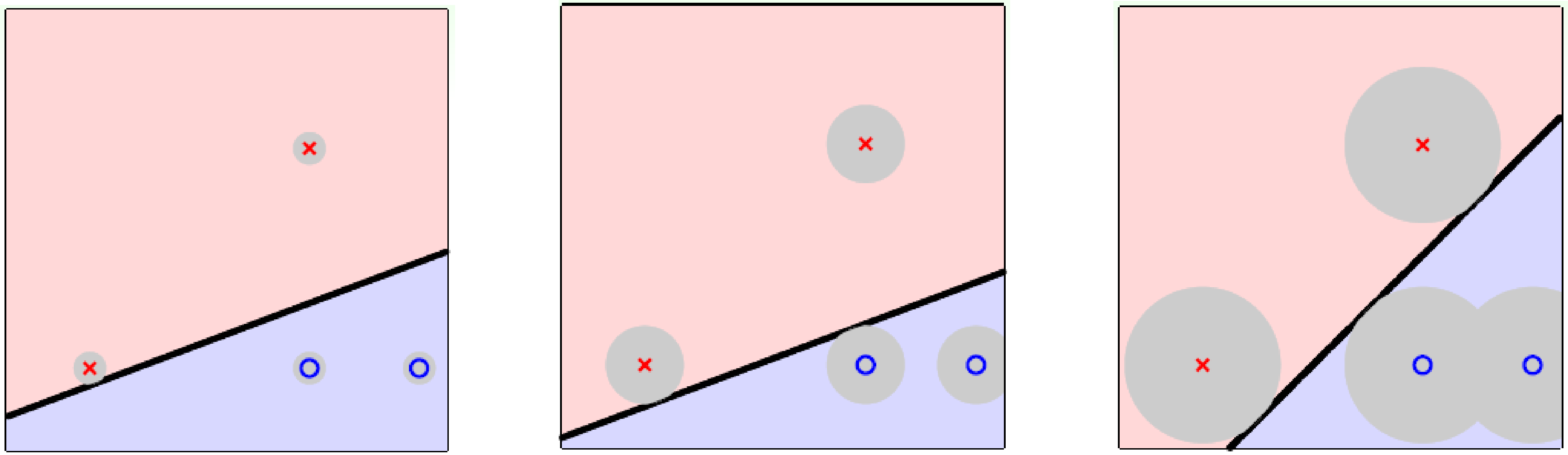
- 感知器算法能找出吗?
- VC bound?

$$E_{\text{out}}(\mathbf{w}) \le \underbrace{E_{\text{in}}(\mathbf{w})}_{0} + \underbrace{\Omega(\mathcal{H})}_{d_{\text{VC}}=d+1}$$

Ref.: NTU-LIN

7.1 最大间隔分类面

为什么会认为最右边的分类面最佳?



假设 $\mathbf{x} \approx \mathbf{x}_n + \Delta\mathbf{x}_n$, $\Delta\mathbf{x}_n \sim N(\mathbf{x}_n, \sigma_n)$

$\mathbf{x}_n$ 离分类面越远 ⇔ 容许σ_n越大，噪声的容忍度越大⇔不易出现过拟合

⇔ 更鲁棒分类面⇔对噪声的容忍度更大⇔离分类面最近的$\mathbf{x}_n$ 到分类面距离

最右边的最佳----因为离分类面最近的$\mathbf{x}_n$ 到分类面距离**最大**，对噪声最鲁棒

7.1 最大间隔分类面

对噪声鲁棒的分类面

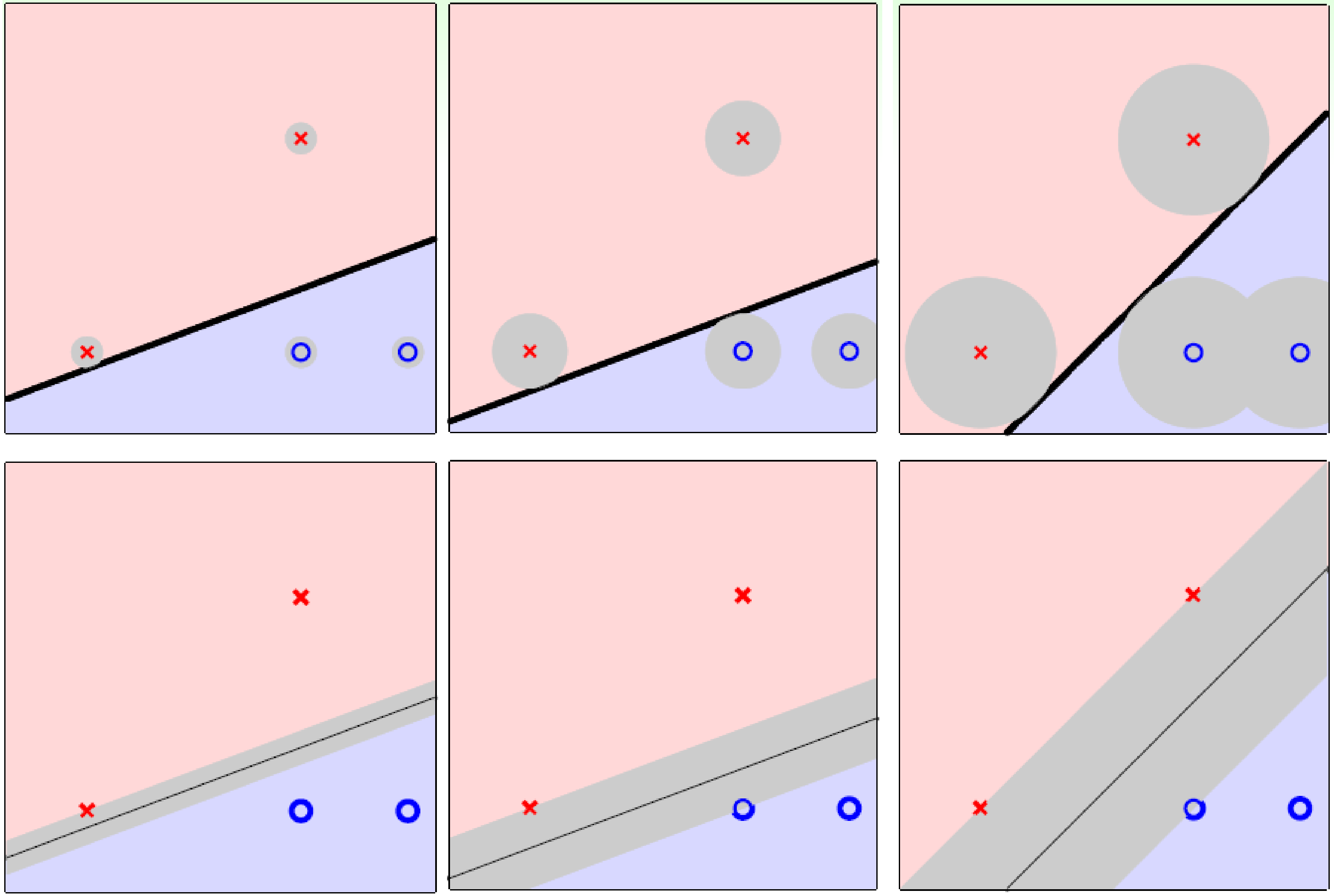
⇕

分类面到两边样本的距离要大

⇕

“胖胖”的分类面

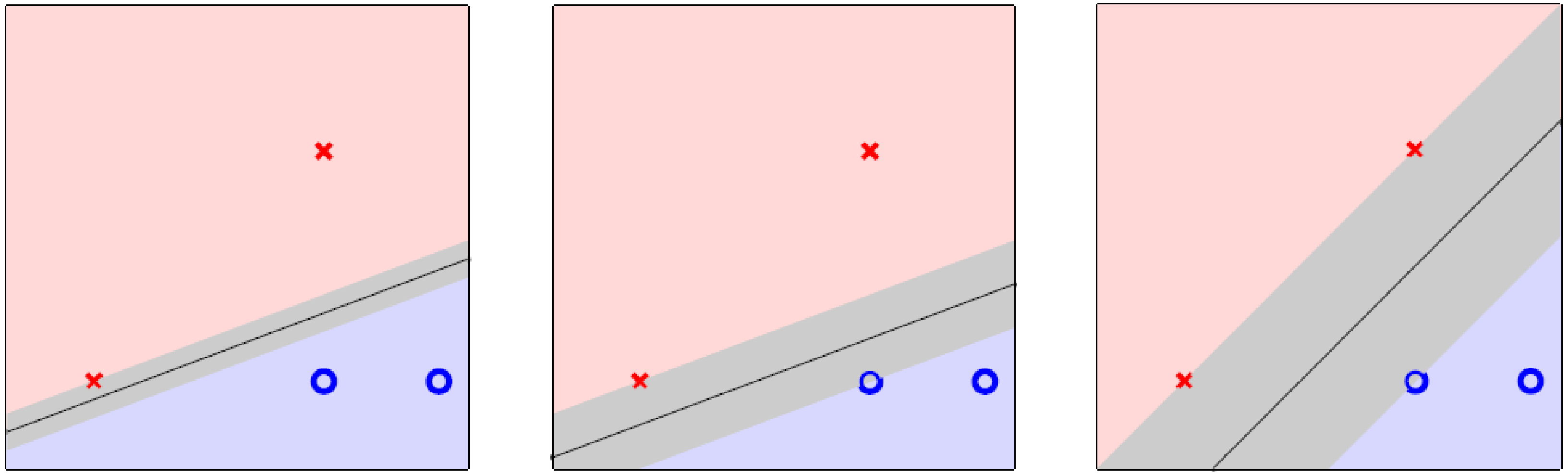
算法的目的是如何找到“胖胖”的分类面



Ref.: NTU-LIN

7.1 最大间隔分类面

"胖胖"的分类面



$$\max_{\mathbf{w}} \quad \text{fatness}(\mathbf{w})$$

$$\text{subject to} \quad \mathbf{w} \text{ classifies every } (\mathbf{x}_n, y_n) \text{ correctly}$$

$$\text{fatness}(\mathbf{w}) = \min_{n=1,\ldots,N} \text{distance}(\mathbf{x}_n, \mathbf{w})$$

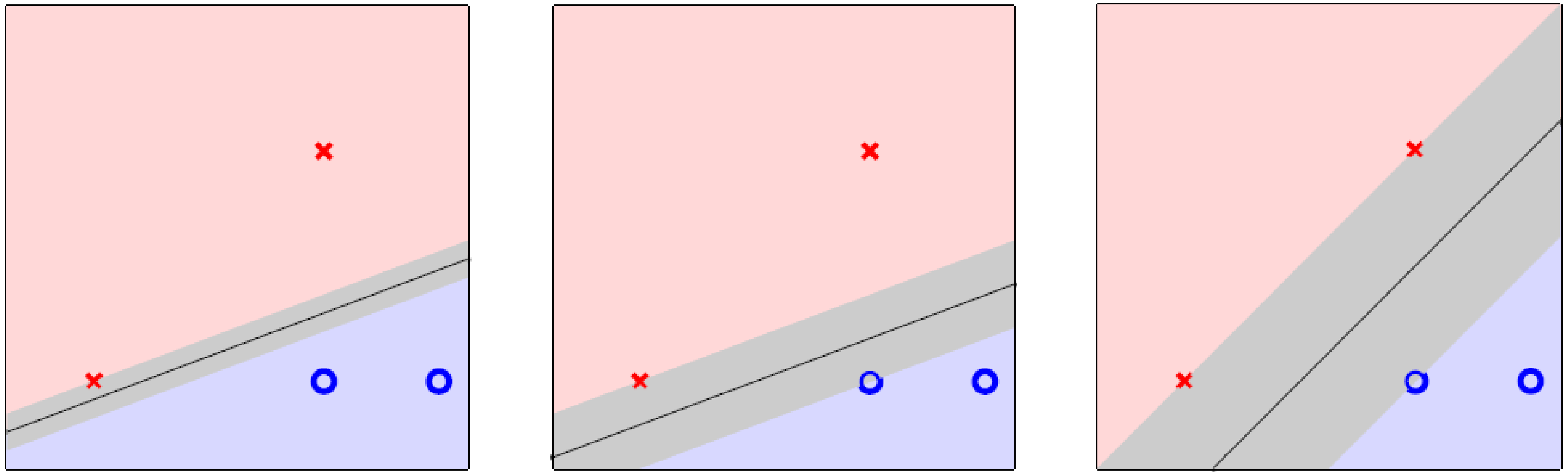
Ref.: NTU-LIN

7.1 最大间隔分类面

“胖胖”的分类面

⇕

最大间隔分类面



所有样本正确分类

⇕

$y_n = \mathrm{sign}(\boldsymbol{w}^T \mathbf{x}_n)$

$$\max_{\mathbf{w}} \quad \mathrm{margin}(\mathbf{w})$$

subject to $\mathbf{w}$ classifies every $(\mathbf{x}_n, y_n)$ correctly

$$\mathrm{margin}(\mathbf{w}) = \min_{n=1,\ldots,N} \mathrm{distance}(\mathbf{x}_n, \mathbf{w})$$

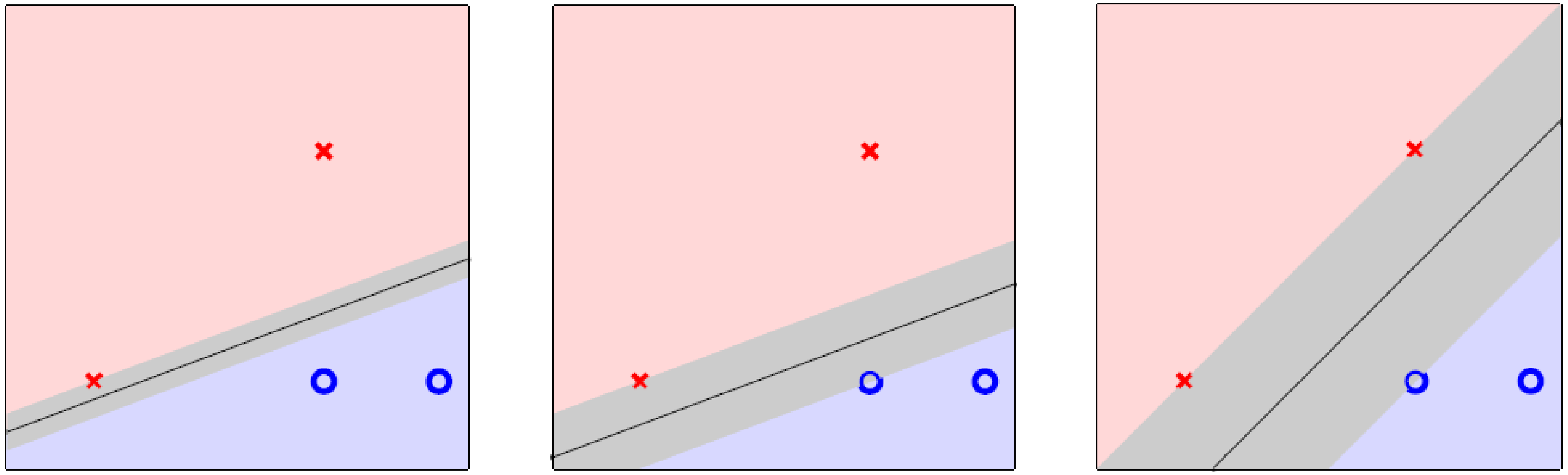
Ref.: NTU-LIN

7.1 最大间隔分类面

“胖胖”的分类面

⇕

最大间隔分类面



所有样本正确分类

⇕

$y_n = \text{sign}(\boldsymbol{w}^T \mathbf{x}_n)$

$$\max_{\mathbf{w}} \quad \text{margin}(\mathbf{w})$$

$$\text{subject to} \quad \text{every } y_n \mathbf{w}^T \mathbf{x}_n > 0$$

$$\text{margin}(\mathbf{w}) = \min_{n=1,\ldots,N} \text{distance}(\mathbf{x}_n, \mathbf{w})$$

算法的目的是：如何找到“最大间隔”的分类面(*find largest-margin separating hyperplane*)

Ref.: NTU-LIN

7.1 最大间隔分类面(*Large-Margin Separating Hyperplane*)

7.2 标准的最大间隔问题(*Standard Large-Margin Problem*)

7.3 支撑向量机(*Support Vector Machine*)

7.2 标准的最大间隔问题

样本到分类面的距离

$$\max_{\mathbf{w}} \quad \text{margin}(\boldsymbol{w})$$

$$\textit{Subject to} \quad \textit{every} \;\; y_n \mathbf{w}^T \mathbf{x}_n > 0$$

$$\text{margin}(\boldsymbol{w}) = \min_{n=1,\ldots,N} \text{distance}(\mathbf{x}_n, \boldsymbol{w})$$

为了便于了解分类器特性，支撑向量机的分析过程中样本向量不做增广

$$b = w_0 \qquad\qquad \cancel{x_0 = 1}$$

$$\begin{bmatrix} | \\ \mathbf{w} \\ | \end{bmatrix} = \begin{bmatrix} w_1 \\ \vdots \\ w_d \end{bmatrix} ; \quad \begin{bmatrix} | \\ \mathbf{x} \\ | \end{bmatrix} = \begin{bmatrix} x_1 \\ \vdots \\ x_d \end{bmatrix}$$

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

Ref.: NTU-LIN

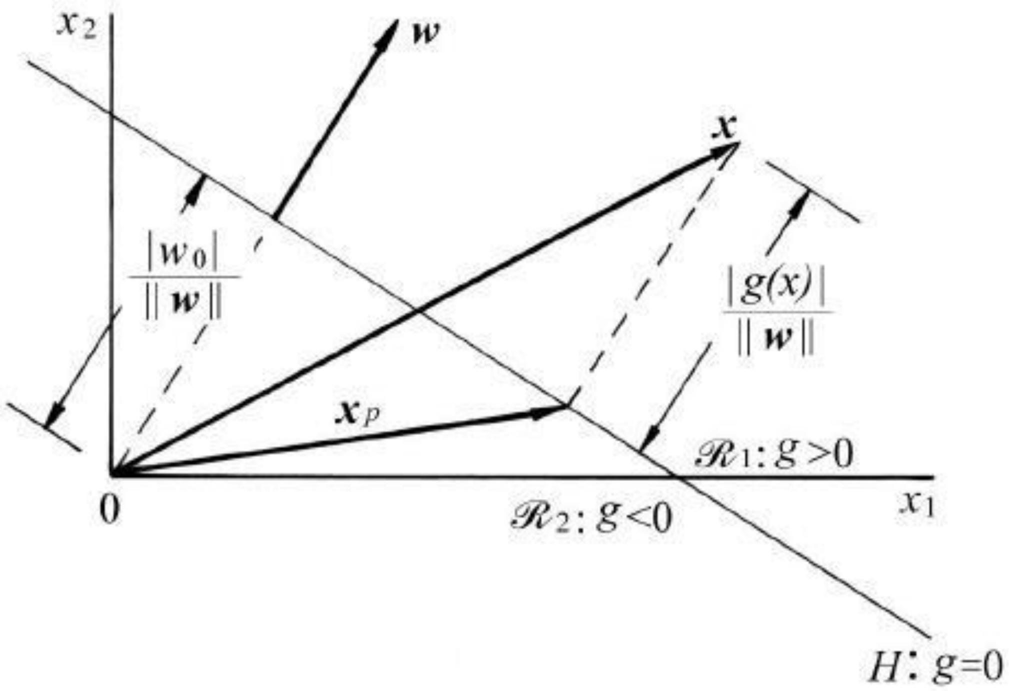
7.2 标准的最大间隔问题

样本到分类面的距离

$$\max_{\mathbf{w}} \quad \text{margin}(\boldsymbol{w})$$

$$\textit{Subject to} \quad \textit{every}\ \ y_n(\boldsymbol{w}^T\mathbf{x}_n + b) > 0$$

$$\text{margin}(\boldsymbol{w}) = \min_{n=1,\dots,N} \text{distance}(\mathbf{x}_n, \boldsymbol{w})$$



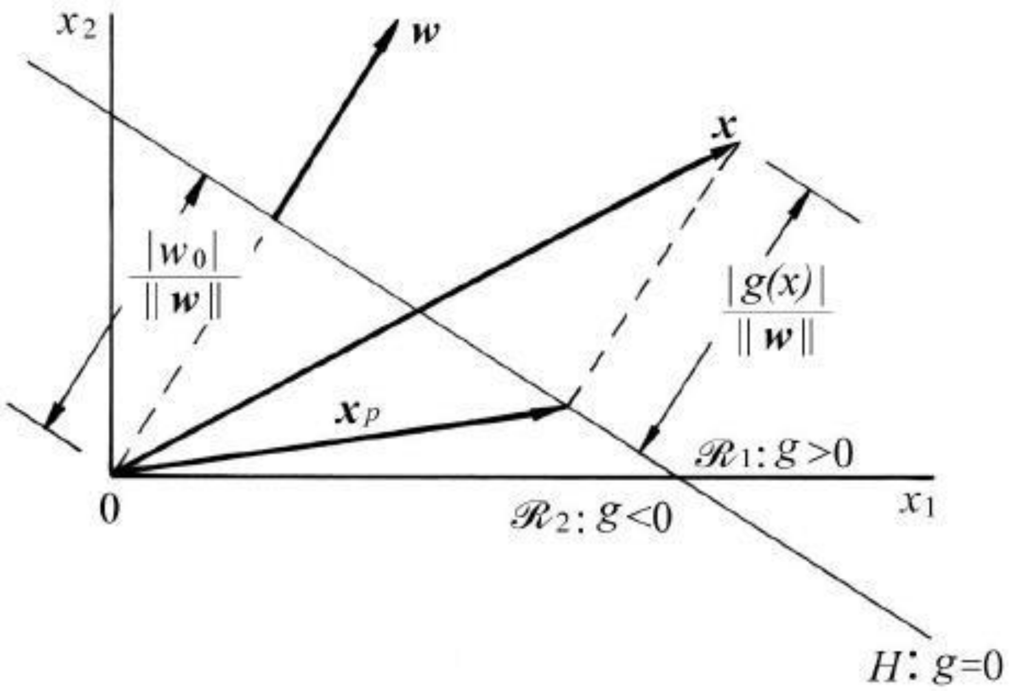
$$
\begin{aligned}
g(\mathbf{x}) &= \boldsymbol{w}^T\mathbf{x} + w_0 \\
&= \boldsymbol{w}^T\left(\mathbf{x}_p + r\frac{\boldsymbol{w}}{\|\boldsymbol{w}\|}\right) + w_0 \\
&= \boldsymbol{w}^T\ \mathbf{x}_p + w_0 + r\frac{\boldsymbol{w}^T\boldsymbol{w}}{\|\boldsymbol{w}\|} \\
&= r\|\boldsymbol{w}\|
\end{aligned}
$$

样本到分类面的距离

$$\max_{\mathbf{w}} \quad \text{margin}(\boldsymbol{w})$$

$$\textit{Subject to} \quad \textit{every} \ \ y_n(\boldsymbol{w}^T\mathbf{x}_n + b) > 0$$

$$\text{margin}(\boldsymbol{w}) = \min_{n=1,\ldots,N} \text{distance}(\mathbf{x}_n, \boldsymbol{w})$$



$$g(\mathbf{x}) = \boldsymbol{w}^T\mathbf{x} + b$$
$$= \boldsymbol{w}^T\left(\mathbf{x}_p + r\frac{\boldsymbol{w}}{\|\boldsymbol{w}\|}\right) + b$$
$$= \boldsymbol{w}^T\ \mathbf{x}_p + b + r\frac{\boldsymbol{w}^T\boldsymbol{w}}{\|\boldsymbol{w}\|}$$
$$= r\|\boldsymbol{w}\|$$

$$|r| = \frac{|g(\mathbf{x})|}{\|\boldsymbol{w}\|} \quad \Rightarrow \quad \text{distance}(\mathbf{x}_n, \boldsymbol{w}) = \frac{1}{\|\boldsymbol{w}\|}\left|\boldsymbol{w}^T\mathbf{x} + b\right|$$

样本到分类面的距离

$$\text{distance}(\mathbf{x}_n, \boldsymbol{w}) = \frac{1}{\|\boldsymbol{w}\|}\left|\boldsymbol{w}^T\mathbf{x} + \boldsymbol{b}\right|$$

$$\because \quad \textit{every} \;\; y_n(\boldsymbol{w}^T\mathbf{x}_n + b) > 0$$

$$\text{distance}(\mathbf{x}_n, \boldsymbol{w}) = \frac{\mathbf{1}}{\|\boldsymbol{w}\|} y_n(\boldsymbol{w}^T\mathbf{x}_n + b)$$

$$\max_{\mathbf{w}} \quad \text{margin}(\boldsymbol{w})$$

$$\textit{Subject to} \quad \textit{every} \;\; y_n(\boldsymbol{w}^T\mathbf{x}_n + b) > 0$$

$$\text{margin}(\boldsymbol{w}) = \min_{n=1,\ldots,N} \text{distance}(\mathbf{x}_n, \boldsymbol{w})$$

$$\max_{\boldsymbol{w}} \quad \text{margin}(\boldsymbol{w})$$

$$\textit{Subject to} \quad \textit{every} \;\; y_n(\boldsymbol{w}^T\mathbf{x}_n + b) > 0$$

$$\text{margin}(\boldsymbol{w}) = \min_{n=1,\ldots,N} \frac{1}{\|\boldsymbol{w}\|} y_n(\boldsymbol{w}^T\mathbf{x}_n + b)$$

Ref.: NTU-LIN

7.2 标准的最大间隔问题

分类面的尺度缩放：

$$\mathbf{w}^T\mathbf{x}_n + b = 0$$

$$3\mathbf{w}^T\mathbf{x}_n + 3b = 0$$

$$\therefore \quad \min_{n=1,\dots,N} y_n(\mathbf{w}^T\mathbf{x}_n + b) = 1$$

$$\Downarrow$$

$$\text{margin}(\mathbf{w}) = \frac{1}{\|\mathbf{w}\|}$$

$$\max_{\mathbf{w}} \quad \text{margin}(\mathbf{w})$$

$$\textit{Subject to} \quad \textit{every} \;\; y_n(\mathbf{w}^T\mathbf{x}_n + b) > 0$$

$$\text{margin}(\mathbf{w}) = \min_{n=1,\dots,N} \frac{1}{\|\mathbf{w}\|} y_n(\mathbf{w}^T\mathbf{x}_n + b)$$

$$\Downarrow$$

$$\max_{\mathbf{w}} \quad \frac{1}{\|\mathbf{w}\|}$$

$$\textit{Subject to} \quad \textit{every} \;\; y_n(\mathbf{w}^T\mathbf{x}_n + b) > 0$$

$$\min_{n=1,\dots,N} y_n(\mathbf{w}^T\mathbf{x}_n + b) = 1$$

Ref.: NTU-LIN

标准的最大间隔问题：

条件松弛后的值域

$y_n(\mathbf{w}^T\mathbf{x}_n + b) \geq 1, for\ all\ n$

原始约束条件下的值域

$\min_{n=1,\ldots,N} y_n(\mathbf{w}^T\mathbf{x}_n + b) = 1$

条件松弛后的解
仍然在紫色值域

$$\max_{\mathbf{w}} \quad \frac{1}{\|\mathbf{w}\|}$$

$$Subject\ to \quad \min_{n=1,\ldots,N} y_n(\mathbf{w}^T\mathbf{x}_n + b) = 1$$

如果$(\mathbf{w}^*, b^*)$ 位于蓝色值域，即对所有样本：

$$y_n({\mathbf{w}^*}^T\mathbf{x}_n + b^*) \geq 1.6$$

根据分类面$(\mathbf{w}^*, b^*)$ 取值的尺度不变：

$$y_n\left(\frac{{\mathbf{w}^*}^T}{1.6}\mathbf{x}_n + \frac{b^*}{1.6}\right) \geq 1$$

$\left(\frac{{\mathbf{w}^*}^T}{1.6}, \frac{b^*}{1.6}\right)$为最佳解 ➡ 矛盾！！！

Ref.: NTU-LIN

7.2 标准的最大间隔问题

标准的最大间隔问题：

条件松弛后的值域

$y_n(\mathbf{w}^T\mathbf{x}_n + b) \geq 1, for\ all\ n$

原始约束条件下的值域

$\min_{n=1,\ldots,N} y_n(\mathbf{w}^T\mathbf{x}_n + b) = 1$

条件松弛后的解
仍然在紫色值域

$$\max_{\mathbf{w}} \quad \frac{1}{\|\mathbf{w}\|}$$

$$Subject\ to \quad \min_{n=1,\ldots,N} y_n(\mathbf{w}^T\mathbf{x}_n + b) = 1$$

$$\min_{\mathbf{w}} \quad \frac{1}{2}\mathbf{w}^T\mathbf{w}$$

$$Subject\ to \quad y_n(\mathbf{w}^T\mathbf{x}_n + b) \geq 1, for\ all\ n$$

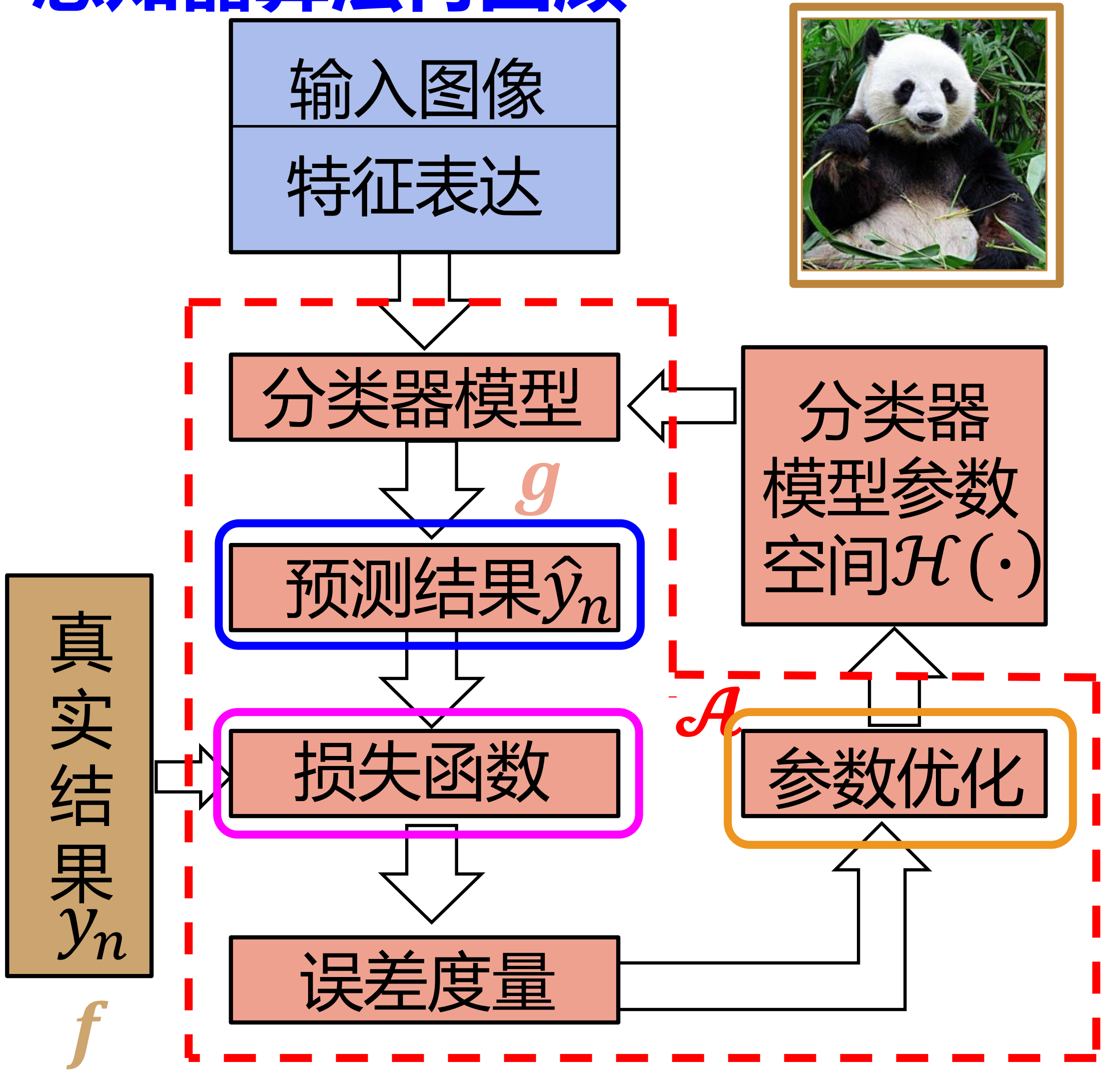
Ref.: NTU-LIN

7.1 最大间隔分类面(*Large-Margin Separating Hyperplane*)

7.2 标准的最大间隔问题(*Standard Large-Margin Problem*)

7.3 支撑向量机(*Support Vector Machine*)

感知器算法再回顾



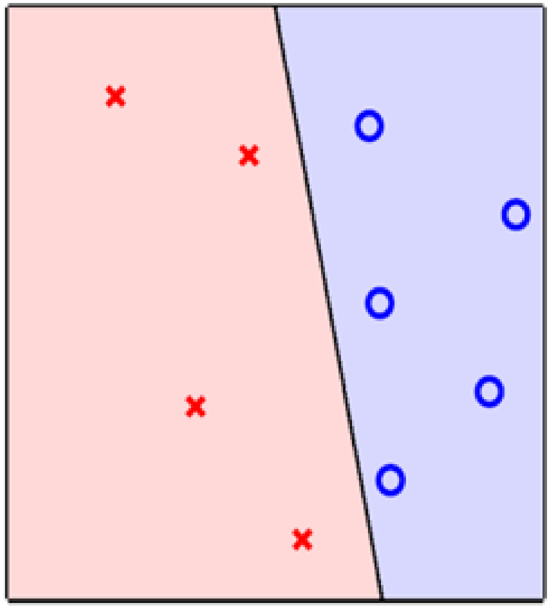
$$\hat{y}_{n(t)} = \operatorname{sign}(\mathbf{w}_t^T \mathbf{x}_{n(t)})$$

算法收敛：

$$L_{in} = \sum_{n=1}^{N} [\![y_n \neq \hat{y}_n]\!] = 0$$

$$\mathbf{w}_{t+1} = \mathbf{w}_t + y_n \mathbf{x}_{n(t)}$$

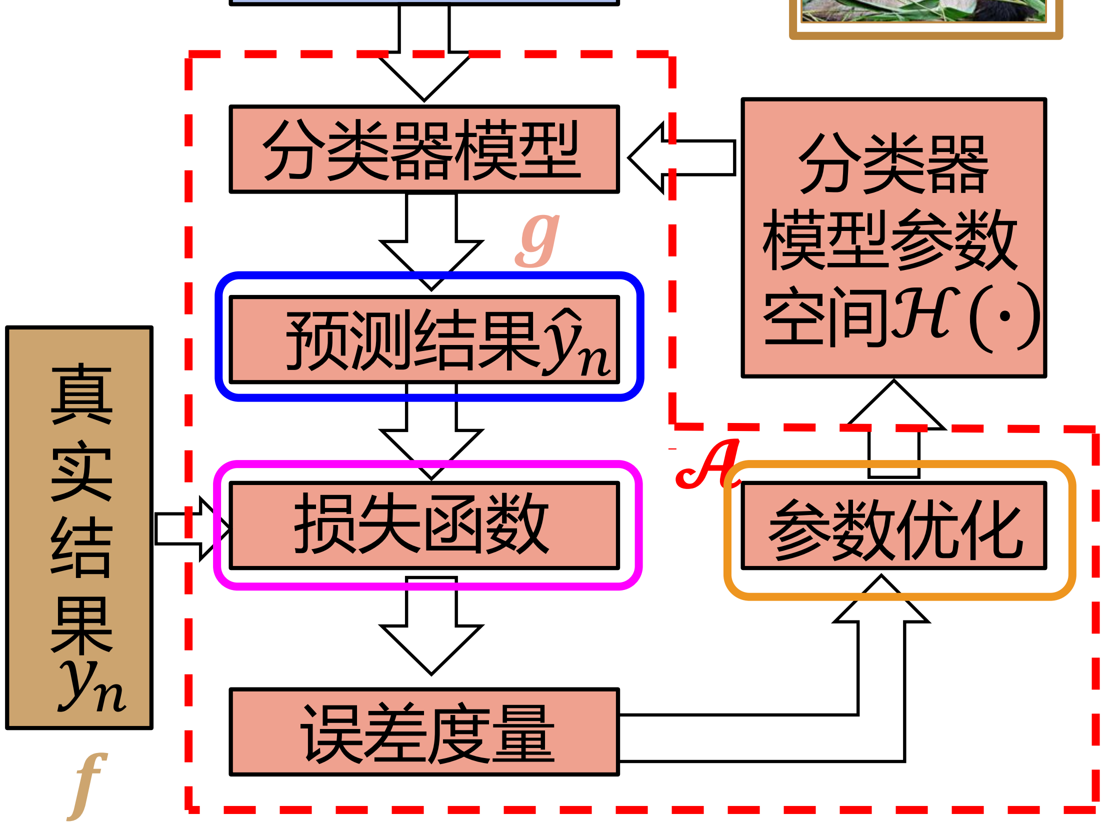
线性可分



- 设置初始分类面(权重)$\mathbf{w}_0$
- 如果有样本分错，就修正权重

Ref.: NTU-LIN

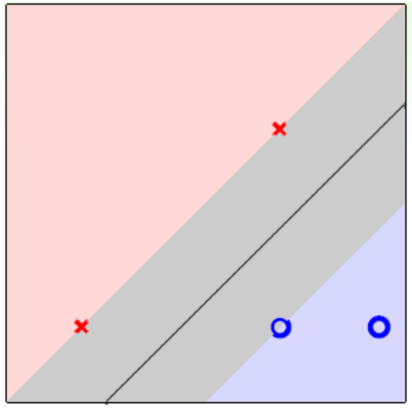
支撑向量机算法



$$\min_{\boldsymbol{w}} \quad \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w}$$

$$Subject\ to \quad y_n(\boldsymbol{w}^T\mathbf{x}_n + b) \geq 1, for\ all\ n$$

$$\hat{y}_n = \mathrm{sign}(\boldsymbol{w}^T\mathbf{x}_n + b)$$

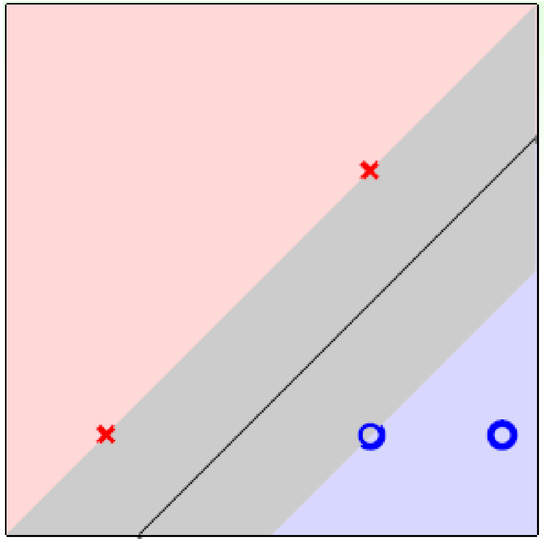


通过求解目标函数的最优解找到最大间隔作为分类面

Ref.: NTU-LIN

7.3 支撑向量机

最大间隔面的求解示例：



$$\mathbf{X}=\begin{bmatrix}0 & 0\\ 2 & 2\\ 2 & 0\\ 3 & 0\end{bmatrix}\quad \mathbf{y}=\begin{bmatrix}-1\\ -1\\ +1\\ +1\end{bmatrix}$$

$$\min_{\boldsymbol{w}} \quad \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w}$$

$$Subject\ to \qquad y_n(\boldsymbol{w}^T\mathbf{x}_n+b)\geq 1, for\ all\ n$$

由约束条件可得：

$$\begin{cases} \qquad\qquad\qquad -b\geq 1 & (1)\\ -2w_1-2w_2-b\geq 1 & (2)\\ 2w_1 \qquad\quad +b\geq 1 & (3)\\ 3w_1 \qquad\quad +b\geq 1 & (4)\end{cases}$$

对约束条件计算可得：

$$(1)+(3)\Rightarrow w_1\geq +1$$

$$(2)+(3)\Rightarrow w_2\leq -1$$

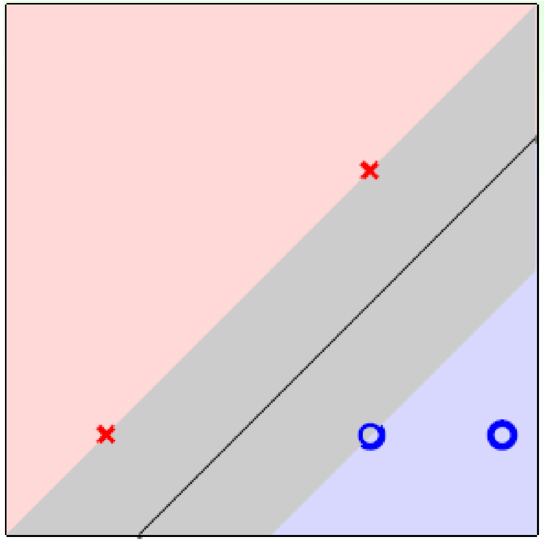
由目标函数$\frac{1}{2}\boldsymbol{w}^T\boldsymbol{w}$取最小值可得：

$$w_1=1,\quad w_2=-1$$

代入约束条件：

$b\leq -1$， $b\leq -1$， $b\geq -1$， $b\geq -2$；

最大间隔面的求解示例:



$$\mathbf{X} = \begin{bmatrix} 0 & 0 \\ 2 & 2 \\ 2 & 0 \\ 3 & 0 \end{bmatrix} \quad \mathbf{y} = \begin{bmatrix} -1 \\ -1 \\ +1 \\ +1 \end{bmatrix}$$

$$\min_{\boldsymbol{w}} \quad \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w}$$

$$Subject\ to \quad y_n(\boldsymbol{w}^T\mathbf{x}_n + b) \geq 1, for\ all\ n$$

由约束条件可得:

$$\begin{cases} \quad\quad\quad\quad\quad\quad -b \geq 1 & (1) \\ -2w_1 - 2w_2 - b \geq 1 & (2) \\ 2w_1 \quad\quad\quad + b \geq 1 & (3) \\ 3w_1 \quad\quad\quad + b \geq 1 & (4) \end{cases}$$

对约束条件计算可得:

$$(1) + (3) \Rightarrow w_1 \geq +1$$

$$(2) + (3) \Rightarrow w_2 \leq -1$$

由目标函数$\frac{1}{2}\boldsymbol{w}^T\boldsymbol{w}$取最小值可得:

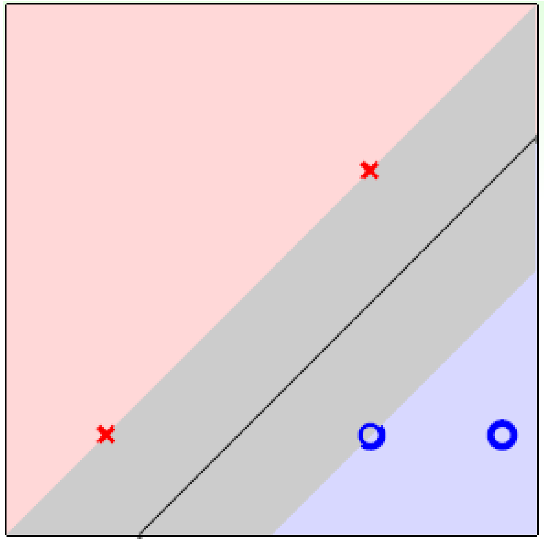
$$w_1 = 1, \quad w_2 = -1$$

代入约束条件:

$$b \leq -1, \quad b \leq -1, \quad b \geq -1, \quad b \geq -2;$$

Ref.: NTU-LIN

最大间隔面的求解示例：



$$\mathbf{X} = \begin{bmatrix} 0 & 0 \\ 2 & 2 \\ 2 & 0 \\ 3 & 0 \end{bmatrix} \quad \mathbf{y} = \begin{bmatrix} -1 \\ -1 \\ +1 \\ +1 \end{bmatrix}$$

$$\min_{\boldsymbol{w}} \quad \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w}$$

$$Subject\ to \quad y_n(\boldsymbol{w}^T\mathbf{x}_n + b) \geq 1, for\ all\ n$$

由约束条件可得：

$$\begin{cases} \qquad\qquad\qquad -b \geq 1 & (1) \\ -2w_1 - 2w_2 - b \geq 1 & (2) \\ 2w_1 \qquad\qquad + b \geq 1 & (3) \\ 3w_1 \qquad\qquad + b \geq 1 & (4) \end{cases}$$

对约束条件计算可得：

$$(1) + (3) \Rightarrow w_1 \geq +1$$

$$(2) + (3) \Rightarrow w_2 \leq -1$$

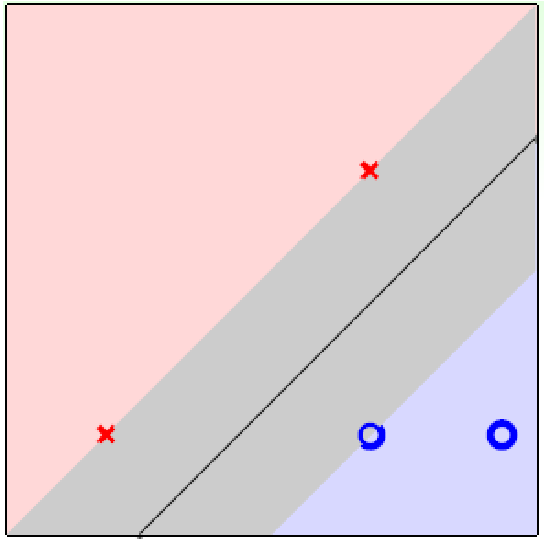
由目标函数$\frac{1}{2}\boldsymbol{w}^T\boldsymbol{w}$取最小值可得：

$$w_1 = 1，\ w_2 = -1$$

代入约束条件：$b = -1$

$$\hat{y}_n = \text{sign}(\boldsymbol{w}^T\mathbf{x}_n + b)$$

最大间隔面的求解示例：



$$\mathbf{X}=\begin{bmatrix}0 & 0\\ 2 & 2\\ 2 & 0\\ 3 & 0\end{bmatrix}\quad \mathbf{y}=\begin{bmatrix}-1\\ -1\\ +1\\ +1\end{bmatrix}$$

$$\min_{\boldsymbol{w}} \quad \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w}$$

$$Subject\ to \quad y_n(\boldsymbol{w}^T\mathbf{x}_n+b)\geq 1, for\ all\ n$$

由约束条件可得：

$$\begin{cases} \qquad\qquad\qquad -b\geq 1 & (1)\\ -2w_1-2w_2-b\geq 1 & (2)\\ 2w_1 \qquad\quad +b\geq 1 & (3)\\ 3w_1 \qquad\quad +b\geq 1 & (4)\end{cases}$$

对约束条件计算可得：

$$(1)+(3)\Rightarrow w_1\geq +1$$

$$(2)+(3)\Rightarrow w_2\leq -1$$

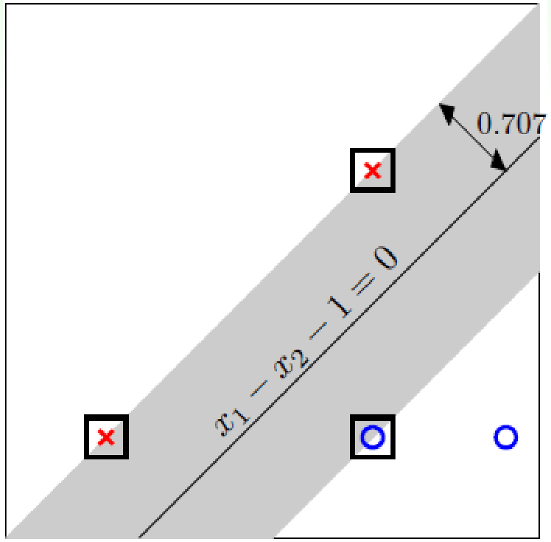
由目标函数$\frac{1}{2}\boldsymbol{w}^T\boldsymbol{w}$取最小值可得：

$$w_1=1，\ w_2=-1$$

代入约束条件：$b=-1$

$$\boxed{\hat{y}=g(\mathbf{x})=\mathrm{sign}(x_1-x_2-1)}$$

为什么叫支撑向量机?



$$\mathbf{X} = \begin{bmatrix} 0 & 0 \\ 2 & 2 \\ 2 & 0 \\ 3 & 0 \end{bmatrix} \quad \mathbf{y} = \begin{bmatrix} -1 \\ -1 \\ +1 \\ +1 \end{bmatrix}$$

$$\min_{\boldsymbol{w}} \quad \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w}$$

$$Subject\ to \quad y_n(\boldsymbol{w}^T\mathbf{x}_n + b) \geq 1, for\ all\ n$$

优化得到的解为:

$$w_1 = 1,\ w_2 = -1,\quad b = -1$$

$$g(\mathbf{x}) = \operatorname{sign}(x_1 - x_2 - 1)$$

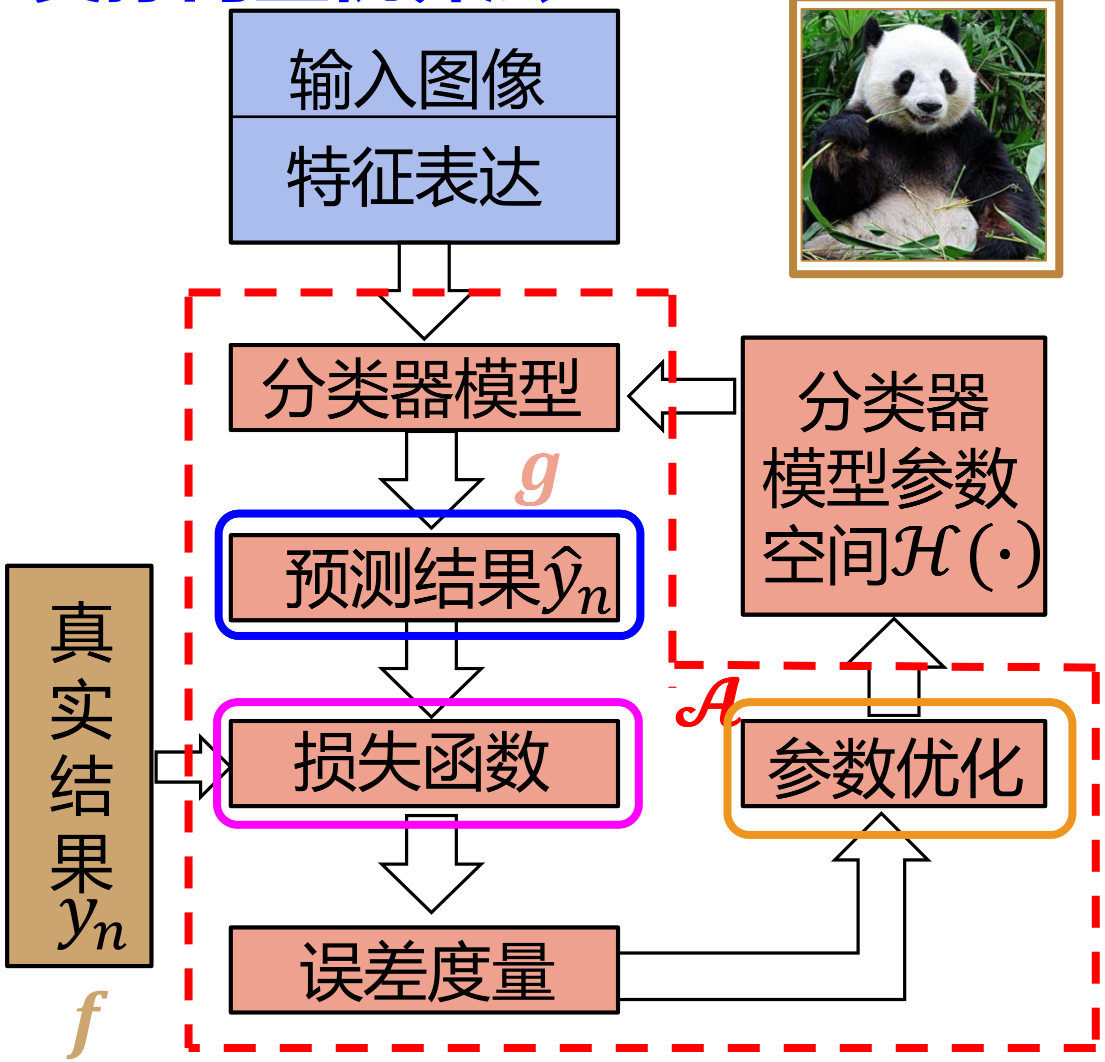
$$\operatorname{margin}(\boldsymbol{w}) = \frac{1}{\|\boldsymbol{w}\|} = \frac{1}{\sqrt{2}}$$

- 分类面由边界上的样本确定，其他样本不起作用
- 边界上的样本被称为支撑向量(候选)

支撑向量机(SVM)—Support Vector Machine
----借助支撑向量学到间隔最大分类面

Ref.: NTU-LIN

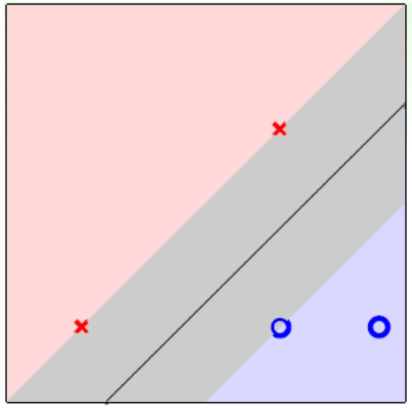
支撑向量机算法



$$\min_{\boldsymbol{w}} \quad \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w}$$

$$Subject\ to \quad y_n(\boldsymbol{w}^T\mathbf{x}_n + b) \geq 1, for\ all\ n$$

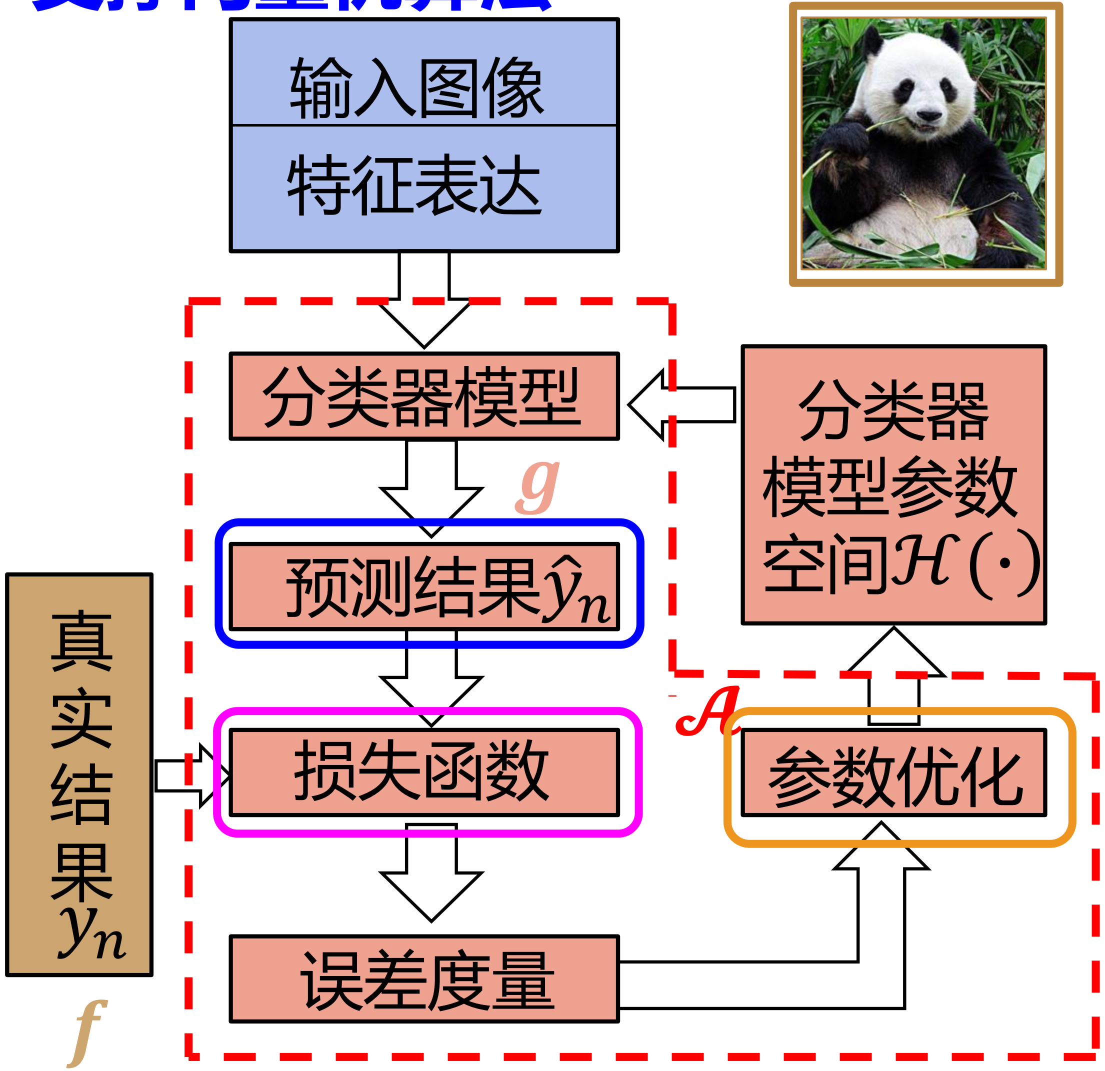
$$\hat{y}_n = \mathrm{sign}(\boldsymbol{w}^T\mathbf{x}_n + b)$$



通过求解目标函数的最优解找到最大间隔作为分类面

Ref.: NTU-LIN

支撑向量机算法



$$\min_{w} \quad \frac{1}{2}\mathbf{w}^T\mathbf{w}$$

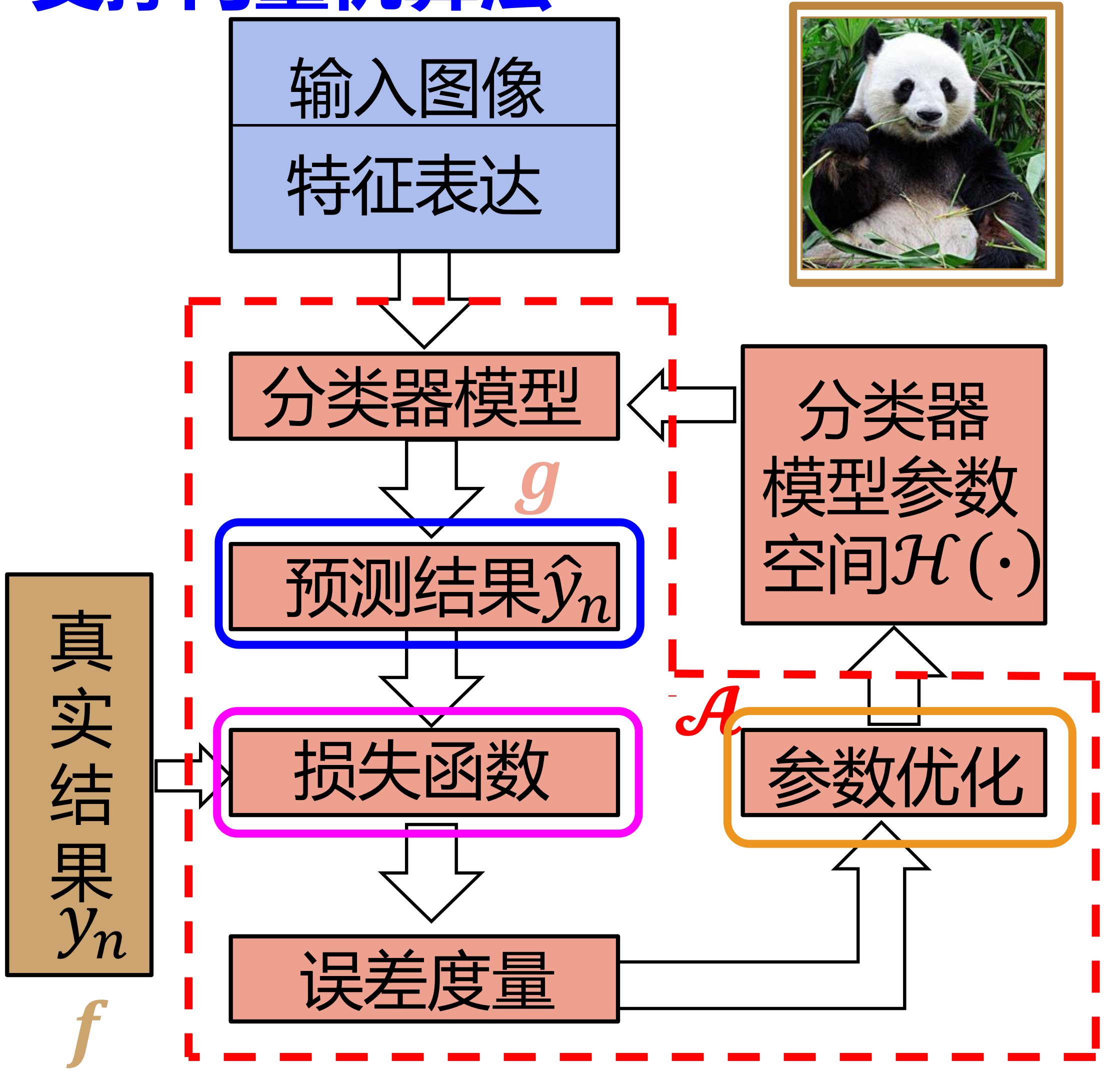
$$Subject\ to \quad y_n(\mathbf{w}^T\mathbf{x}_n + b) \geq 1, for\ all\ n$$

感知器算法的损失函数:

$$L_{in} = \sum_{n=1}^{N} [\![y_n \neq \hat{y}_n]\!] = 0$$

$$\hat{y}_n = \mathrm{sign}(\mathbf{w}^T\mathbf{x}_n + b)$$

支撑向量机算法



$$\min_{w} \quad \frac{1}{2} w^T w$$

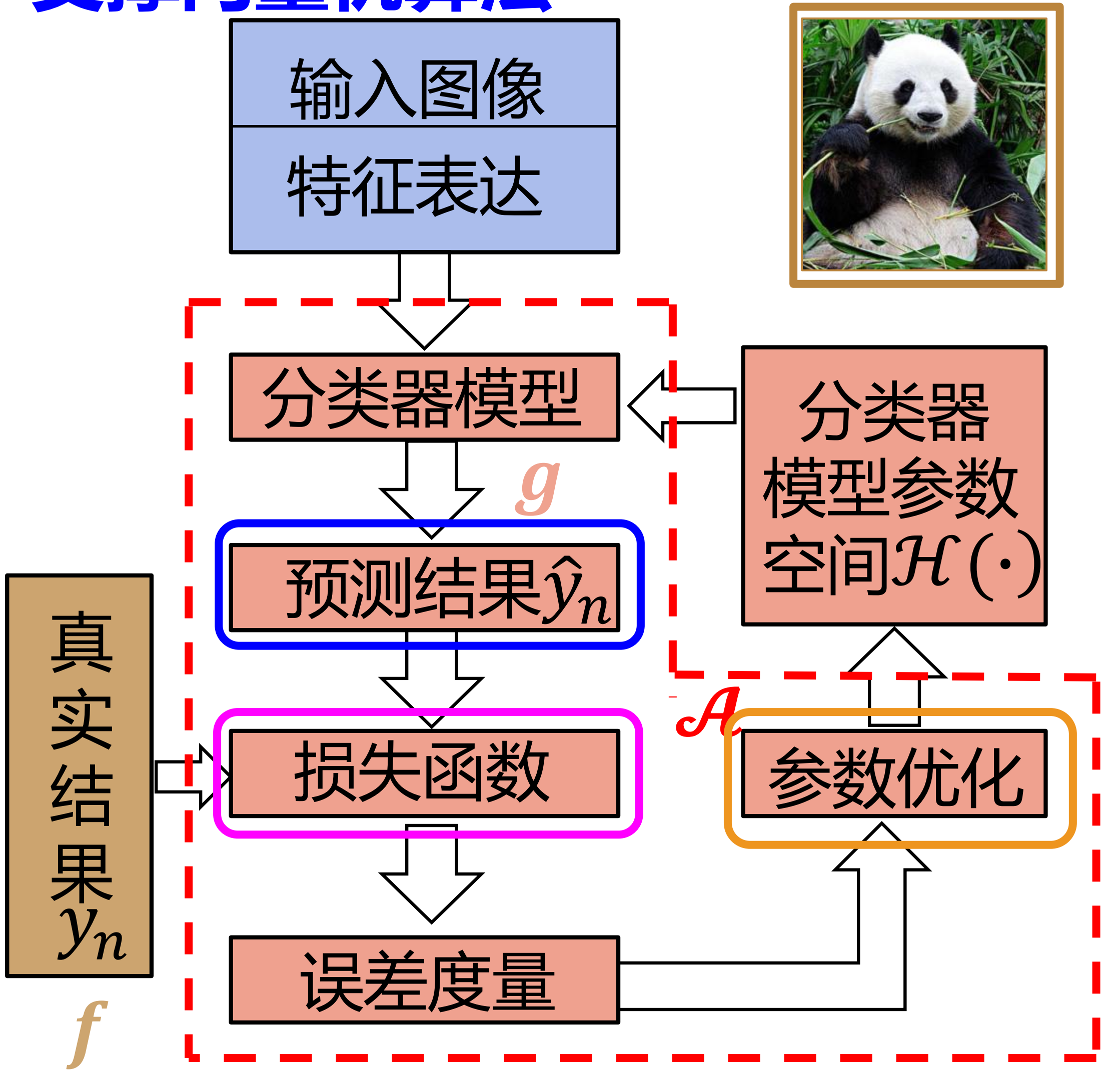
$$\text{Subject to} \quad y_n(\mathbf{w}^T \mathbf{x}_n + b) \geq 1, for\ all\ n$$

支撑向量机算法的损失函数?

$$L_{SVM} = ?$$

$$\hat{y}_n = \mathrm{sign}(\mathbf{w}^T \mathbf{x}_n + b)$$

支撑向量机算法



$$y_n(\mathbf{w}^T\mathbf{x}_n + b) \geq 1$$

$$y_n s_n \geq 1$$

$$1 - y_n s_n \leq 0$$

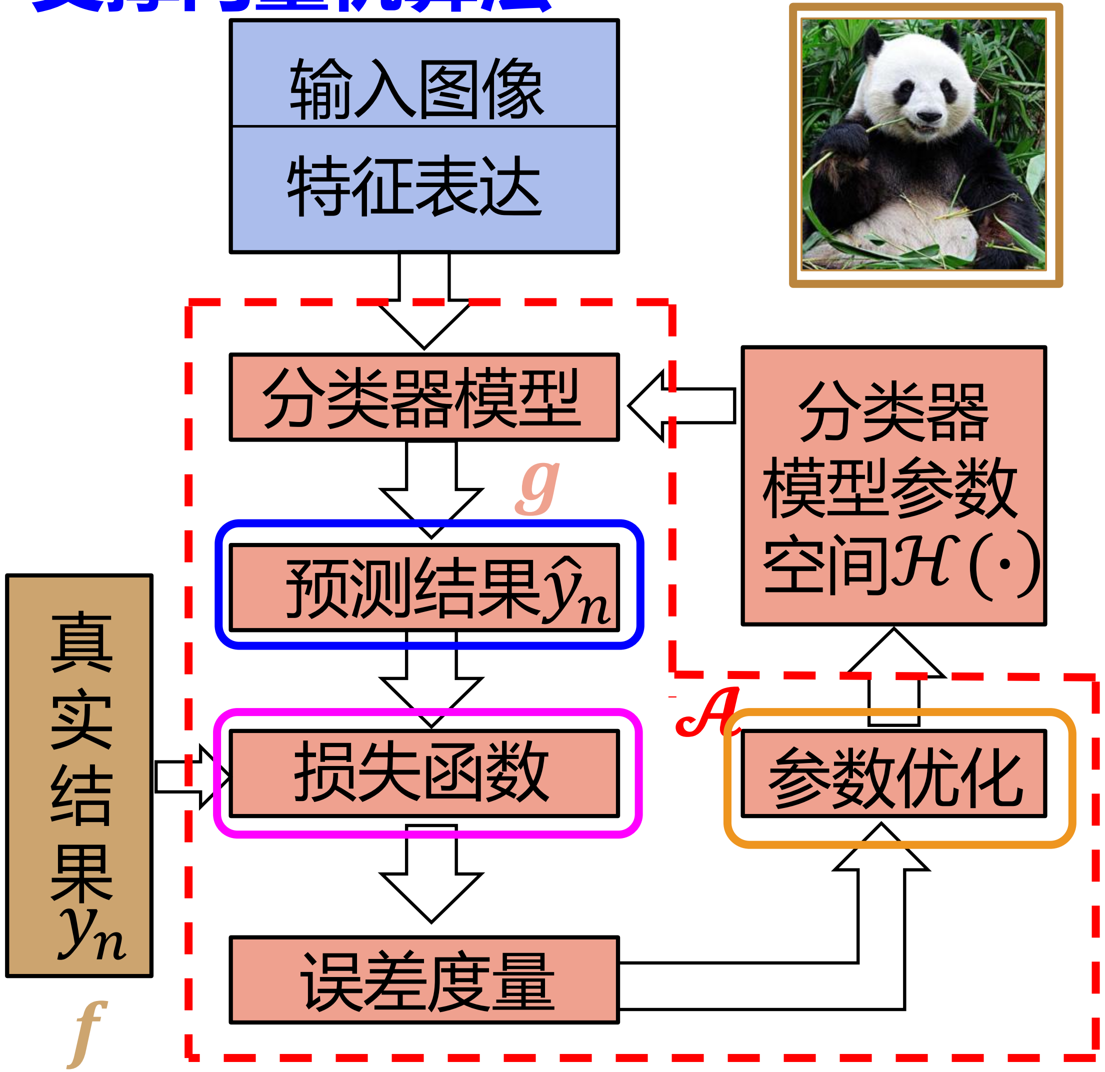
支撑向量机算法的损失函数?

$$L_{SVM} = ?$$

$$\hat{y}_n = \mathrm{sign}(\mathbf{w}^T\mathbf{x}_n + b)$$

7.3 支撑向量机

支撑向量机算法



$$y_n(\mathbf{w}^T\mathbf{x}_n + b) \geq 1$$

$$y_n s_n \geq 1$$

$$1 - y_n s_n \leq 0$$

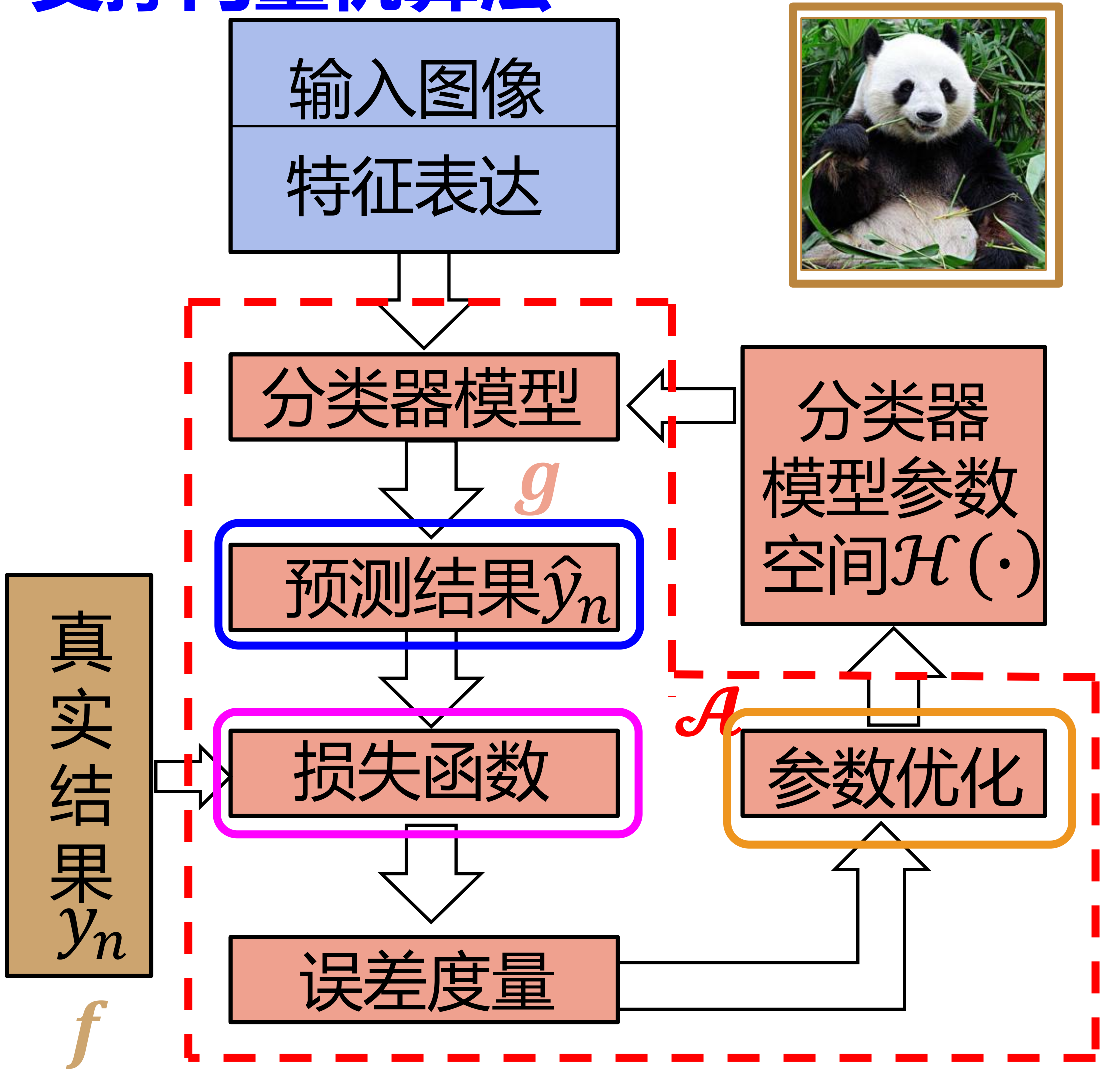
支撑向量机算法的损失函数

$$L_{SVM} = \max(0, 1 - ys)$$

Hinge Loss

$$\hat{y}_n = \mathrm{sign}(\mathbf{w}^T\mathbf{x}_n + b)$$

支撑向量机算法



$$y_n(\mathbf{w}^T\mathbf{x}_n + b) \geq 1$$

$$y_n s_n \geq 1$$

$$1 - y_n s_n \leq 0$$

支撑向量机算法的损失函数

$$L_{SVM} = \max(0,1 - ys)$$ *Hinge Loss*

$$\frac{\partial L_{SVM}(\mathbf{w})}{\partial \mathbf{w}} = [\![1 - y_n(\mathbf{w}^T\mathbf{x}_n) > 0]\!](-y_n\mathbf{x}_n)$$ (增广后)

$$\hat{y}_n = \mathrm{sign}(\mathbf{w}^T\mathbf{x}_n + b)$$

梯度下降法实现支撑向量机

- 初始化权向量$\boldsymbol{w}_0$ **Stochastic Gradient Descent(SGD)**
- $for\ t=0,1,2,\ldots$ (t 代表迭代次数)

① 计算梯度：$\nabla L_{SVM}(\boldsymbol{w}_t)=\frac{1}{B}\sum_{n=1}^{B}[\![1-y_n(\boldsymbol{w}^T\mathbf{x}_n)>0]\!](-y_n\mathbf{x}_n)$

② 对权向量$\boldsymbol{w}_t$进行更新：$\boldsymbol{w}_{t+1}\leftarrow\boldsymbol{w}_t-\eta\nabla L_{SVM}(\boldsymbol{w}_t)$

…直到对任意$\mathbf{x}_n$满足$1-y_n(\boldsymbol{w}_{t+1}^T\mathbf{x}_n)\leq 0$，或者迭代足够多次数

返回最终的$\boldsymbol{w}_{t+1}$作为学到的g

3. 3 梯度下降法

训练(Training)

输入图像

特征表达

回归模型

g

预测结果$\hat{y}_n$

损失函数

误差度量

真实结果y_n

f

回归模型参数空间$\mathcal{H}(\cdot)$

$\mathcal{A}$

参数优化

随机梯度下降法(SGD):

$$\nabla L_{in}(\mathbf{w}) = \sum_{n=1}^{B} (\mathbf{w}^T \mathbf{x}_n - y_n)\mathbf{x}_n$$

$$\mathbf{m}_{i,t+1} = \lambda \mathbf{m}_{i,t} \quad \eta$$

$$\mathbf{w}_{i,t+1} \leftarrow \mathbf{w}$$

第五讲
再讨论

- 问题1：学习率
- 问题2：梯度为 ... 佳解?
- 问题3：训练样本批量大小的影响?
- 问题4：损失函数的影响?

5. 2 逻辑斯蒂回归损失

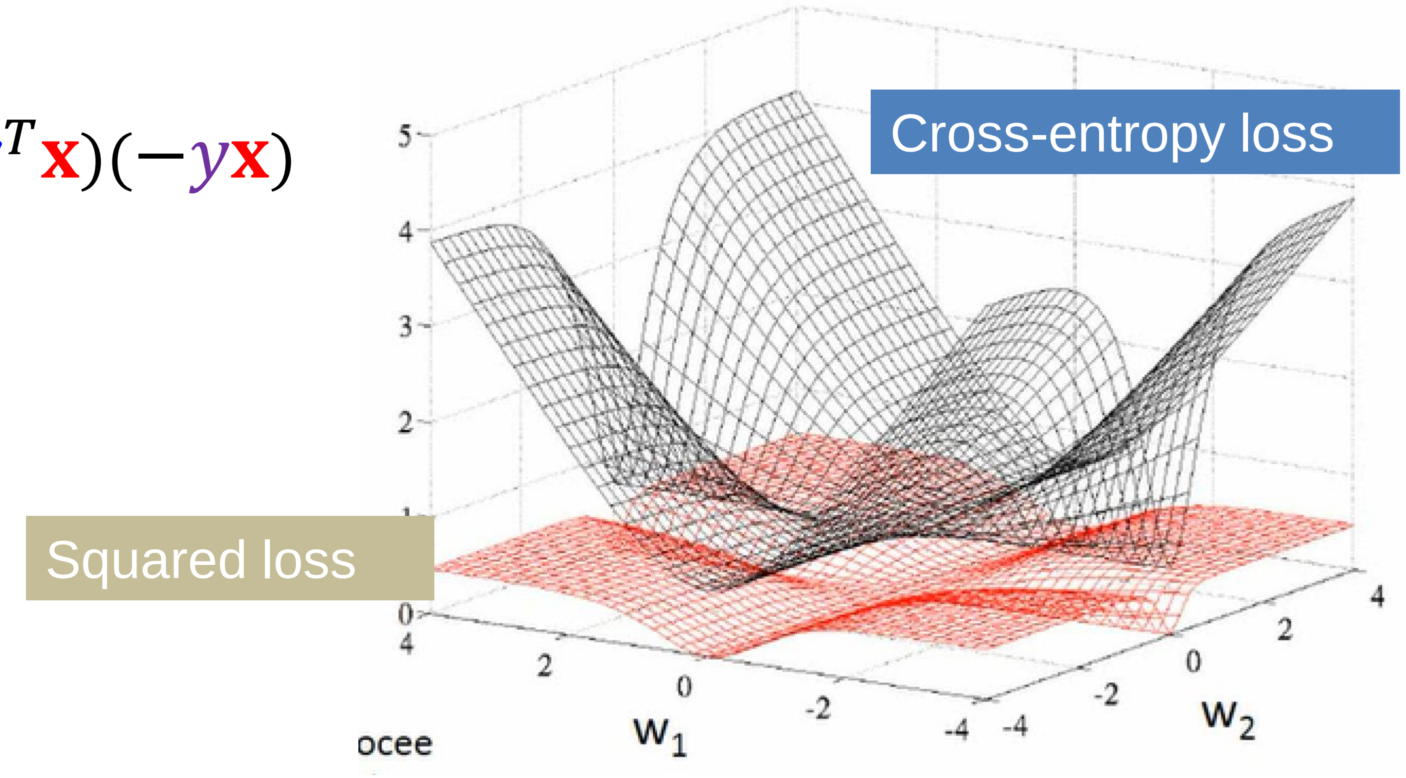
交叉熵损失与平方损失的梯度对比：

平方损失的梯度：

$$\nabla L_{in}(\mathbf{w},\mathbf{x},y) = 2(\theta(y\mathbf{w}^T\mathbf{x}) - 1)\theta(y\mathbf{w}^T\mathbf{x})(1\text{-}\theta(y\mathbf{w}^T\mathbf{x}))y\mathbf{x}$$

交叉熵损失的梯度：

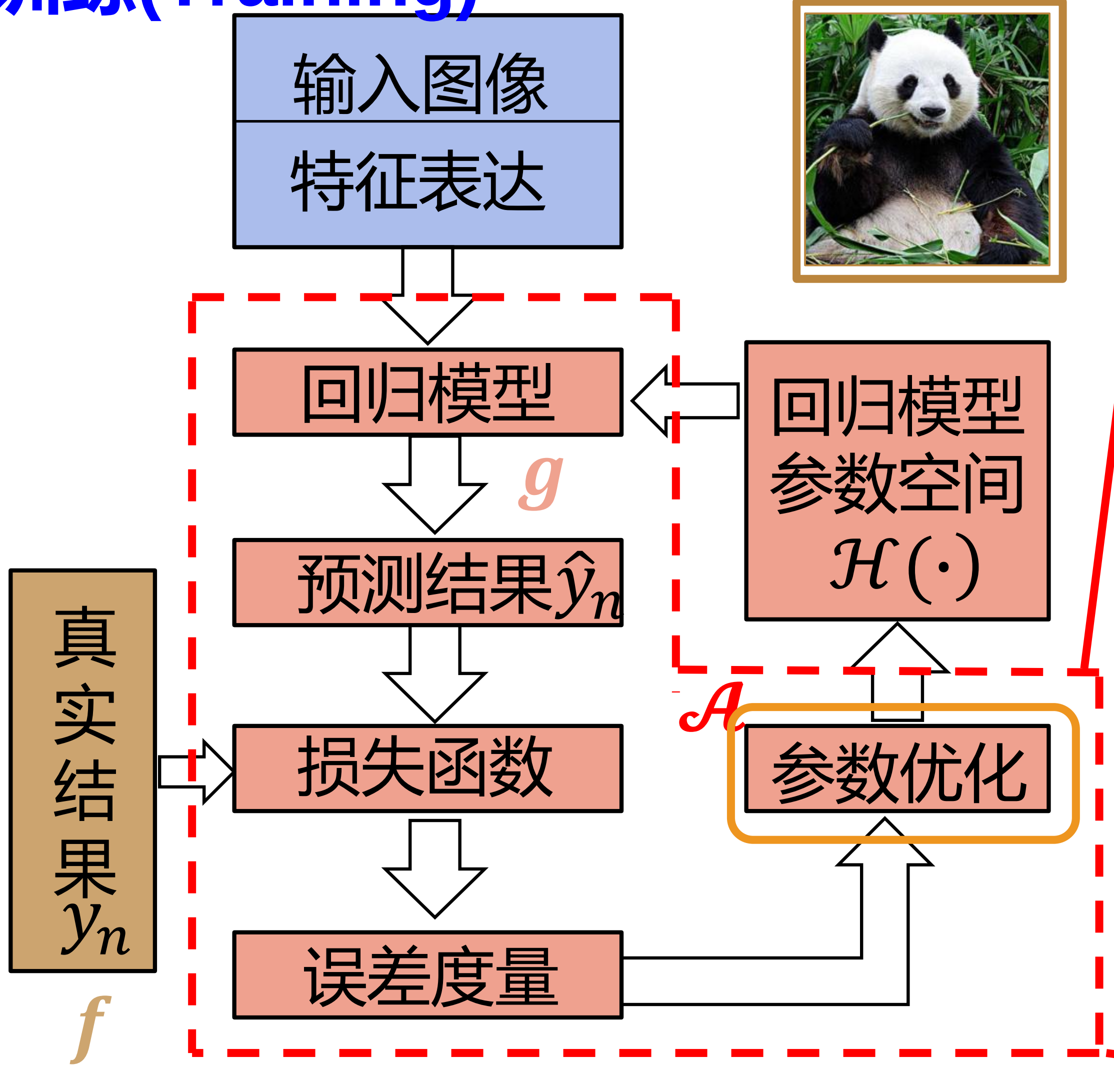
$$\nabla L_{in}(\mathbf{w},\mathbf{x},y) = \theta(-y\mathbf{w}^T\mathbf{x})(-y\mathbf{x})$$



Ref.: NTU-LEE

3. 3 梯度下降法

训练(Training)



随机梯度下降法(SGD)：

$$\nabla L_{in}(\boldsymbol{w}) = \sum_{n=1}^{B} (\boldsymbol{w}^T \mathbf{x}_n - y_n)\mathbf{x}_n$$

$\boldsymbol{m}_{i,t+1} = \lambda \boldsymbol{m}_{i,t}$ … η …

$\boldsymbol{w}_{i,t+1} \leftarrow \boldsymbol{w}$ …

不同损失函数在梯度下降时的速度不同

- 问题1：学习率 …
- 问题2：梯度为 … 佳解？
- 问题3：训练样本批量大小的影响？
- 问题4：损失函数的影响？

3.3 梯度下降法

训练(Training)

输入图像

特征表达

回归模型 g

预测结果$\hat{y}_n$

真实结果 y_n f

损失函数

误差度量

参数优化 $\mathcal{A}$

回归模型参数空间 $\mathcal{H}(\cdot)$

随机梯度下降法(SGD)：

$$\nabla L_{in}(\mathbf{w}) = \sum_{n=1}^{B} (\mathbf{w}^T \mathbf{x}_n - y_n)\mathbf{x}_n$$

$$\mathbf{m}_{i,t+1} = \lambda \mathbf{m}_{i,t}$$

$$\mathbf{w}_{i,t+1} \leftarrow \mathbf{w}$$

第五讲再讨论

第七讲还讨论

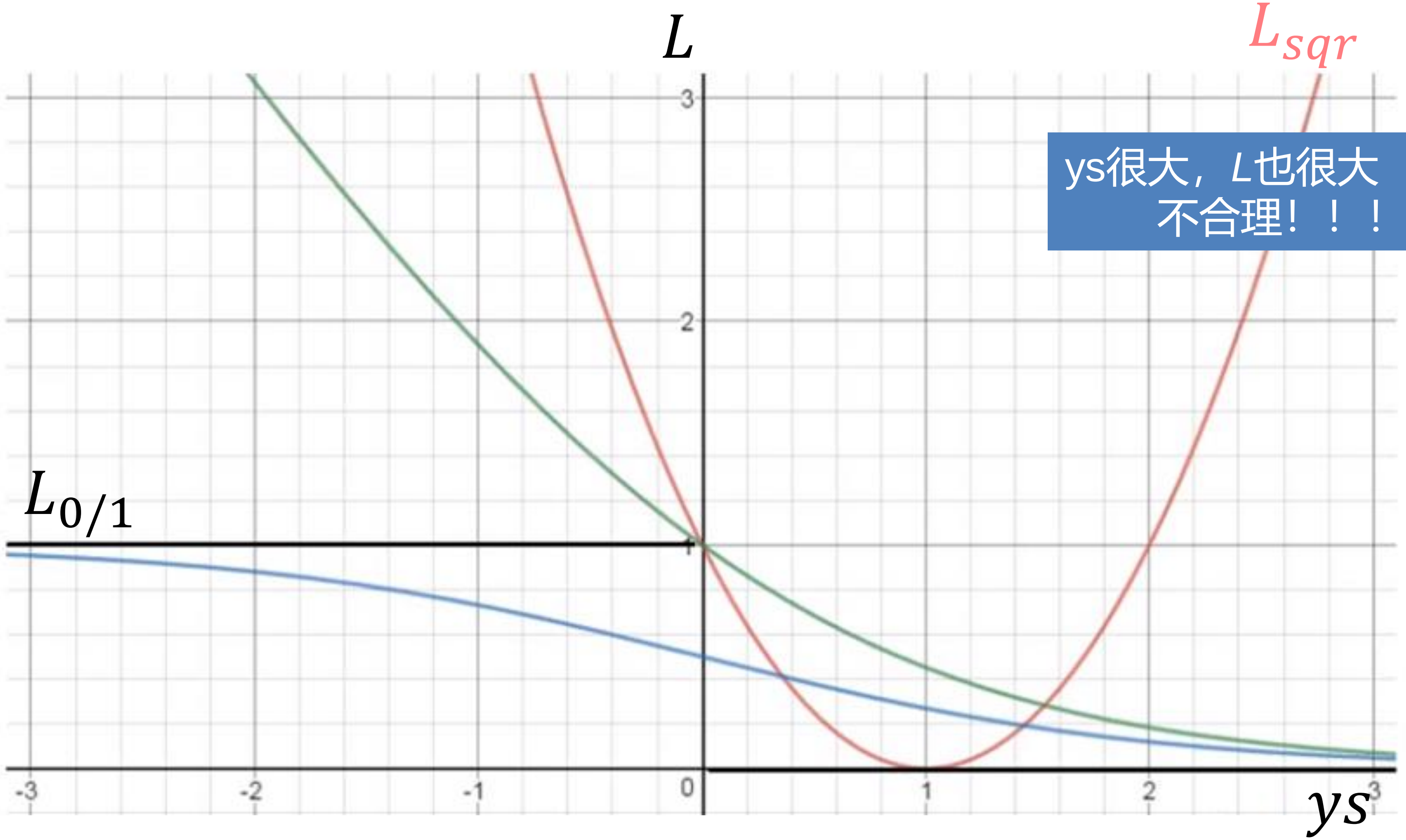
- 问题1：学习
- 问题2：梯度为 佳解？
- 问题3：训练样本批量大小的影响？
- 问题4：损失函数的影响？

7.3 支撑向量机

$L_{0/1} = [\![\hat{y} \neq y]\!]$

$(s-y)^2 \quad = y^2(s-y)^2 = (ys-y^2)^2 = (ys-1)^2$

$L_{sqr} = (ys-1)^2$



ys很大，L也很大
不合理！！！

7.3 支撑向量机

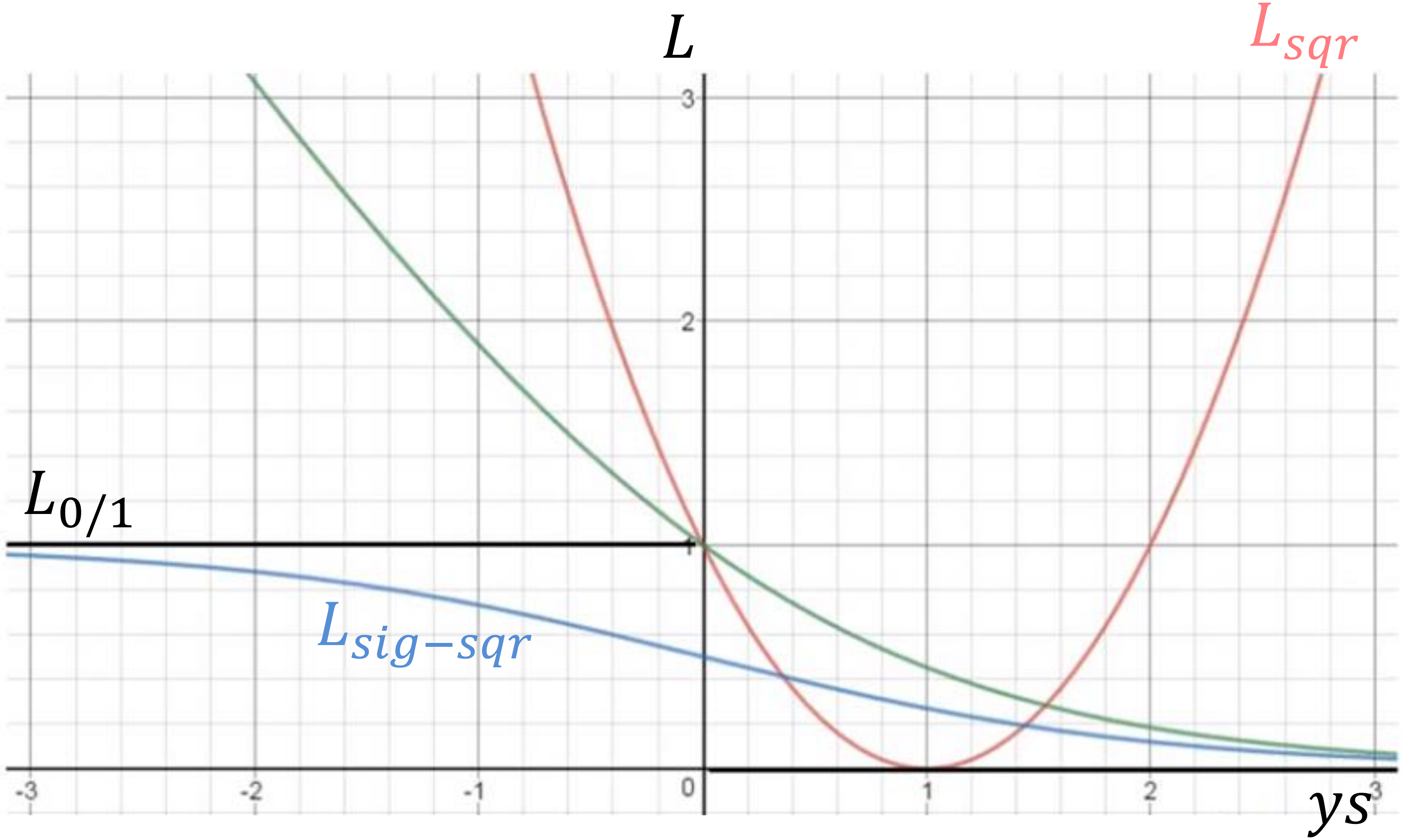
$y=1, L_{sig-sqr}=(\theta(s)-1)^2$

$y=-1, L_{sig-sqr}=(\theta(-s)-1)^2=(1-\theta(s)-1)^2=(\theta(s))^2$

$L_{0/1}=[\![\hat{y}\neq y]\!]$

$L_{sqr}=(ys-1)^2$

$L_{sig-sqr}=(\theta(ys)-1)^2$



Ref.: NTU-LEE

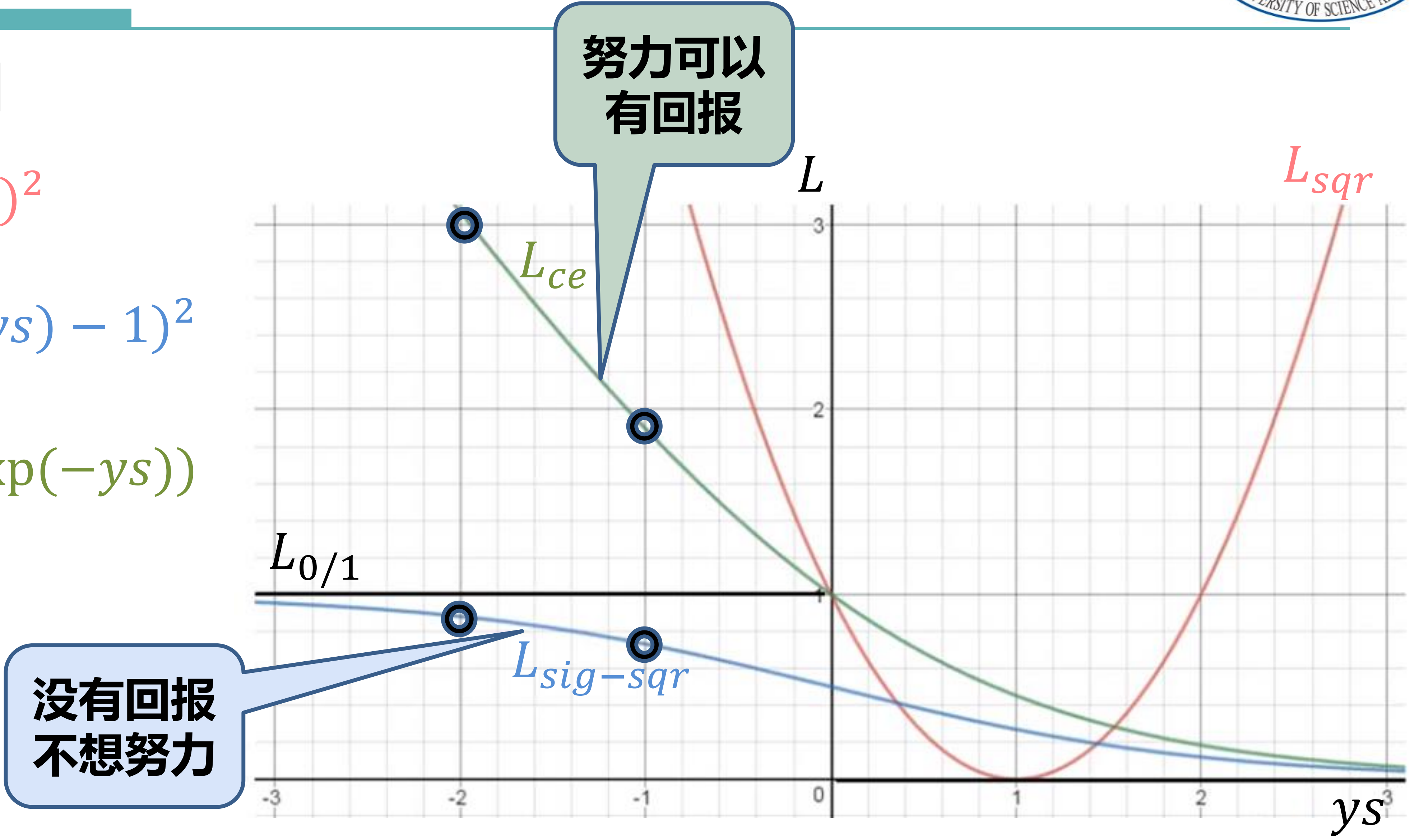
7.3 支撑向量机

$$L_{0/1} = [\![\hat{y} \neq y]\!]$$

$$L_{sqr} = (ys - 1)^2$$

$$L_{sig-sqr} = (\theta(ys) - 1)^2$$

$$L_{ce} = \ln(1 + \exp(-ys))$$



Ref.: NTU-LEE

7.3 支撑向量机

$L_{0/1} = [\![\hat{y} \neq y]\!]$

$L_{sqr} = (ys - 1)^2$

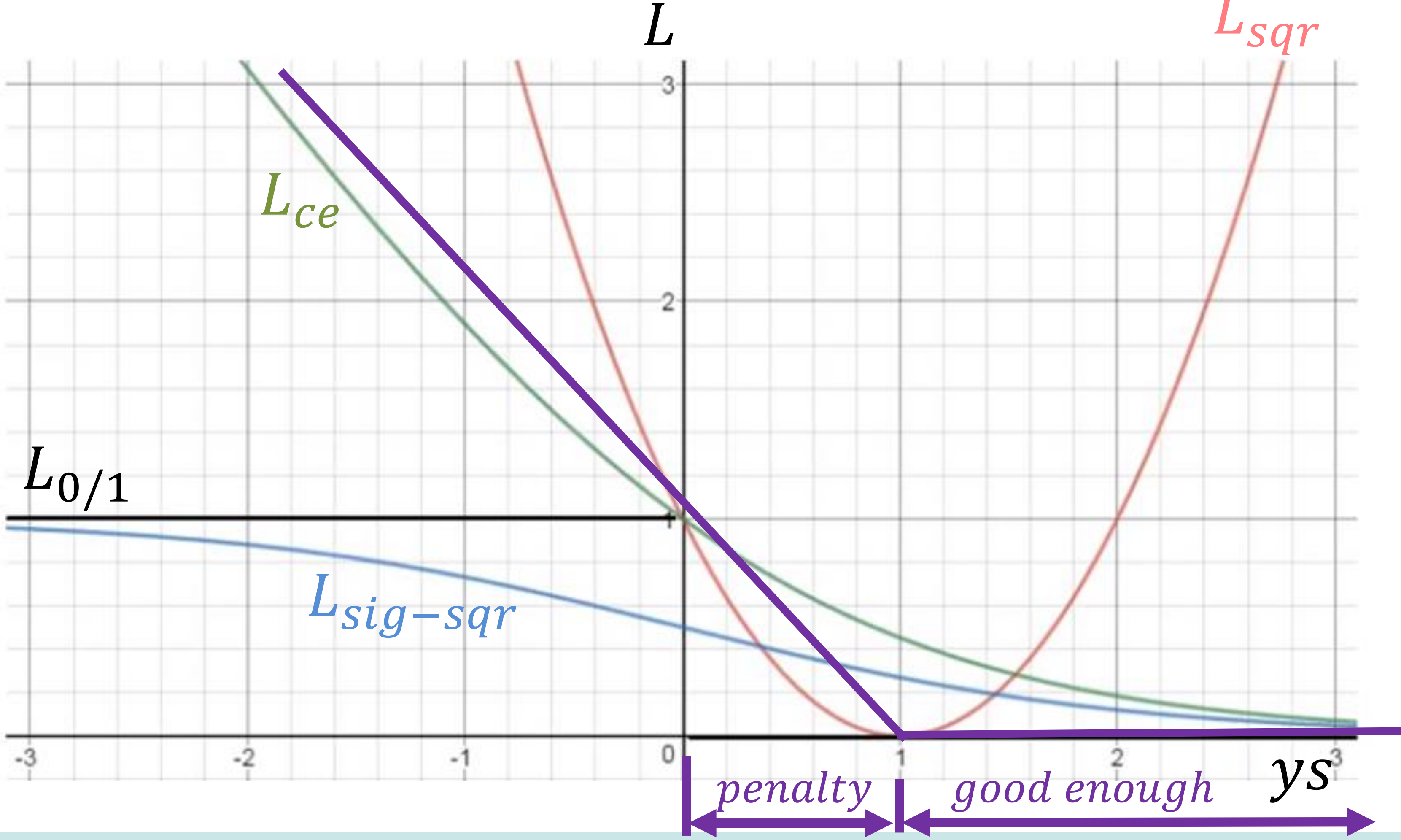
$L_{sig-sqr} = (\theta(ys) - 1)^2$

$L_{ce} = \ln(1 + \exp(-ys))$

$L_{SVM} = \max(0,1 - ys)$

$y = 1, L_{SVM} = \max(0,1 - s), \rightarrow 1 - s < 0 \rightarrow s > 1$

$y = -1, L_{SVM} = \max(0,1 + s), \rightarrow 1 + s < 0 \rightarrow s < -1$



7.3 支撑向量机

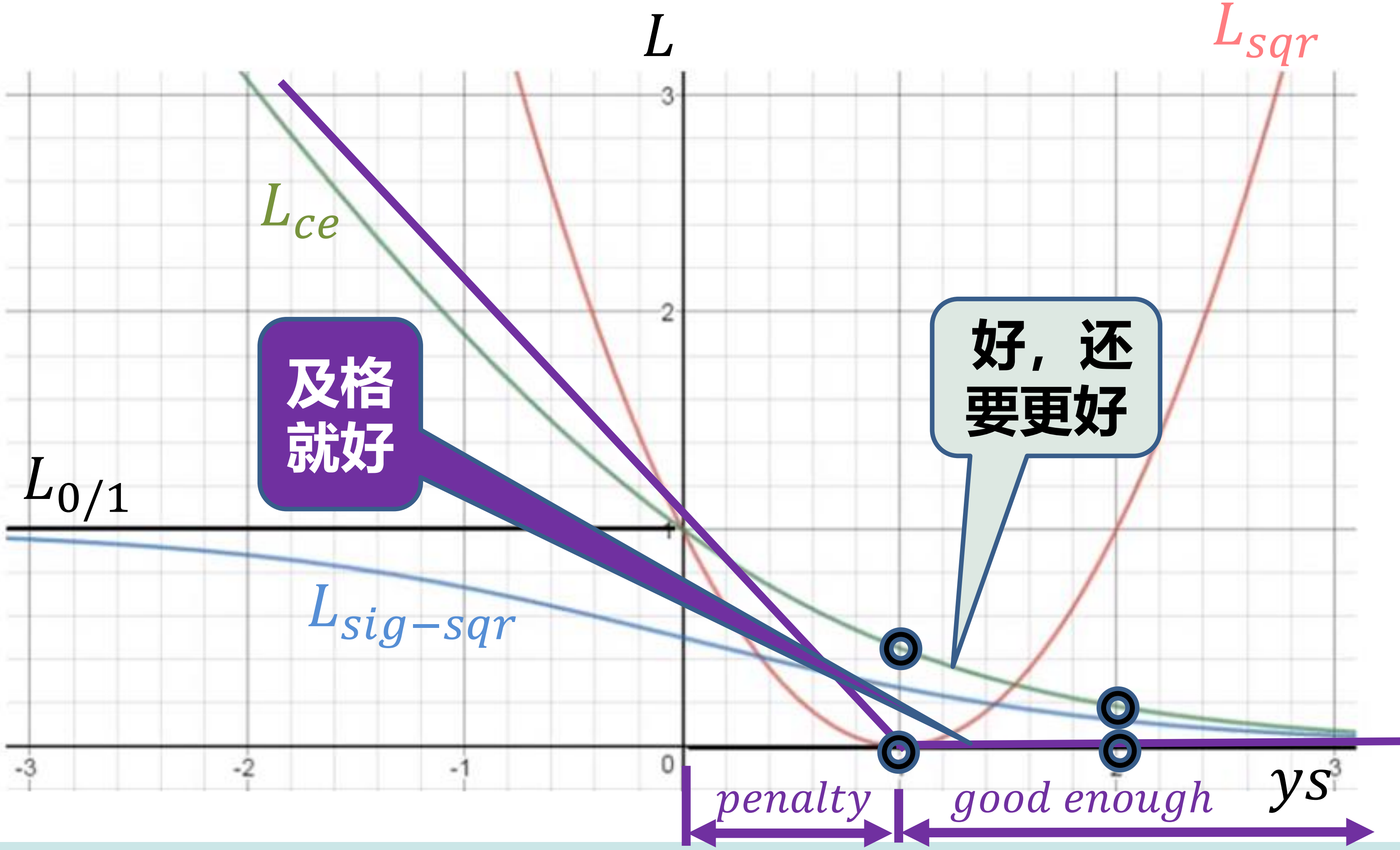
$L_{0/1} = [\![\hat{y} \neq y]\!]$

$L_{sqr} = (ys - 1)^2$

$L_{sig-sqr} = (\theta(ys) - 1)^2$

$L_{ce} = \ln(1 + \exp(-ys))$

$L_{SVM} = \max(0,1 - ys)$



Ref.: NTU-LEE

SVM的一般求解：

$$\min_{\boldsymbol{w}} \quad \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w}$$

$$Subject\ to \quad y_n(\boldsymbol{w}^T\mathbf{x}_n + b) \geq 1, for\ all\ n$$

- 一般情况下，"手工"求解不容易！
- 用梯度下降法？如何处理约束条件？

SVM求解模型的特点：

- $(\boldsymbol{w}, b)$的目标函数是二次函数，是凸函数！
- $(\boldsymbol{w}, b)$的约束条件是线性函数！

----二次规划(*quadratic programming*)问题！

二次规划(QP)
有成熟方便的办法求优化解！

SVM的一般求解：

最佳的(w, b) =?

$$\min_{w} \quad \frac{1}{2} w^T w$$

二次规划(QP)的求解：

最佳的$\mathbf{u} \leftarrow \mathrm{QP}(\mathbf{Q}, \mathbf{p}, \mathbf{A}, \mathbf{c})$

$$\min_{\mathbf{u}} \quad \frac{1}{2} \mathbf{u}^T \mathbf{Q} \mathbf{u} + \mathbf{p}^T \mathbf{u}$$

$$Subject\ to \quad \mathbf{a}_m^T \mathbf{u} \geq c_m$$

$$for\ m = 1,2, \dots M$$

$$\mathbf{u} = \begin{bmatrix} b \\ w \end{bmatrix} \quad \mathbf{Q} = \begin{bmatrix} 0 & \mathbf{0}_d^T \\ \mathbf{0}_d & \boldsymbol{I}_d \end{bmatrix} \quad \mathbf{p} = \mathbf{0}_{d+1}$$

$$\frac{1}{2} \mathbf{u}^T \mathbf{Q} \mathbf{u} + \mathbf{p}^T \mathbf{u} = \frac{1}{2} [b, w] \begin{bmatrix} 0 & \mathbf{0}_d^T \\ \mathbf{0}_d & \boldsymbol{I}_d \end{bmatrix} \begin{bmatrix} b \\ w \end{bmatrix} + \mathbf{0}_{d+1}^T \begin{bmatrix} b \\ w \end{bmatrix} = \frac{1}{2} [0, w] \begin{bmatrix} b \\ w \end{bmatrix} + 0 = \frac{1}{2} w^T w$$

SVM的一般求解：

最佳的(w, b) =?

$$\min_{w} \quad \frac{1}{2} w^T w$$

$$\text{Subject to} \quad y_n(w^T \mathbf{x}_n + b) \geq 1$$

$$for\ n = 1,2, \ldots N$$

$$\mathbf{a}_n^T \mathbf{u} = y_n \begin{bmatrix} 1 & \mathbf{x}_n^T \end{bmatrix} \begin{bmatrix} b \\ w \end{bmatrix} = y_n \left(b + \mathbf{x}_n^T w \right) = y_n (w^T \mathbf{x}_n + b)$$

$$\mathbf{a}_m^T \mathbf{u} \geq c_m \Rightarrow y_n (w^T \mathbf{x}_n + b) \geq 1$$

二次规划(QP)的求解：

最佳的$\mathbf{u} \leftarrow \mathrm{QP}(\mathbf{Q}, \mathbf{p}, \mathbf{A}, \mathbf{c})$

$$\min_{\mathbf{u}} \quad \frac{1}{2} \mathbf{u}^T \mathbf{Q} \mathbf{u} + \mathbf{p}^T \mathbf{u}$$

$$\text{Subject to} \quad \mathbf{a}_m^T \mathbf{u} \geq c_m$$

$$for\ m = 1,2, \ldots M$$

$$\mathbf{u} = \begin{bmatrix} b \\ w \end{bmatrix}, \quad \mathbf{Q} = \begin{bmatrix} 0 & \mathbf{0}_d^T \\ \mathbf{0}_d & \boldsymbol{I}_d \end{bmatrix}, \quad \mathbf{p} = \mathbf{0}_{d+1}$$

$$\mathbf{a}_n^T = y_n \begin{bmatrix} 1 & \mathbf{x}_n^T \end{bmatrix}, \quad c_n = 1, \quad M = N$$

SVM的一般求解：

最佳的$(\boldsymbol{w}, b)$ =?

$$\min_{\boldsymbol{w}} \quad \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w}$$

$$\text{Subject to} \quad y_n(\boldsymbol{w}^T\mathbf{x}_n + b) \geq 1$$

$$for\ n = 1,2,\ldots N$$

二次规划(QP)的求解：

最佳的$\mathbf{u} \leftarrow \text{QP}(\mathbf{Q}, \mathbf{p}, \mathbf{A}, \mathbf{c})$

$$\min_{\mathbf{u}} \quad \frac{1}{2}\mathbf{u}^T\mathbf{Q}\mathbf{u} + \mathbf{p}^T\mathbf{u}$$

$$\text{Subject to} \quad \mathbf{a}_m^T\mathbf{u} \geq c_m$$

$$for\ m = 1,2,\ldots M$$

$$\mathbf{u} = \begin{bmatrix} b \\ \boldsymbol{w} \end{bmatrix}, \quad \mathbf{Q} = \begin{bmatrix} 0 & \mathbf{0}_d^T \\ \mathbf{0}_d & \boldsymbol{I}_d \end{bmatrix}, \quad \mathbf{p} = \mathbf{0}_{d+1}$$

$$\mathbf{a}_n^T = y_n[1 \quad \mathbf{x}_n^T], \quad c_n = 1, \quad M = N$$

通过调用二次规划(QP)的求解函数就能得到SVM的最优解

利用二次规划(QP)实现支撑向量机

① $\mathbf{Q}=\begin{bmatrix}0 & \mathbf{0}_d^T\\ \mathbf{0}_d & I_d\end{bmatrix}$， $\mathbf{p}=\mathbf{0}_{d+1}$， $\mathbf{a}_n^T=y_n[1\quad \mathbf{x}_n^T]$， $c_n=1$，

② $\begin{bmatrix}b\\ \mathbf{w}\end{bmatrix}\leftarrow \mathrm{QP}(\mathbf{Q},\mathbf{p},\mathbf{A},\mathbf{c})$

③ 返回最终的b和$\mathbf{w}$作为学到的g_{SVM}

线性硬间隔SVM算法(*Linear Hard-Margin SVM Algorithm*)

- *Hard-Margin*：没有任何样本会落入到“胖胖的”间隔区域里！
- *Linear*：样本$\mathbf{x}_n$是线性可分的！*如果不是线性可分？* $\mathbf{z}_n=\boldsymbol{\phi}(\mathbf{x}_n)$

7.1 最大间隔分类面(*Large-Margin Separating Hyperplane*)

直觉上对噪声更为鲁棒

7.2 标准的最大间隔问题(*Standard Large-Margin Problem*)

最小化 $\mathbf{w}$ *的模长，使得对所有样本都正确分类*

7.3 支撑向量机(*Support Vector Machine*)

利用二次规划获得最佳解