感知器

Part 2　2025/03/14　凤维杰

回顾：分类器

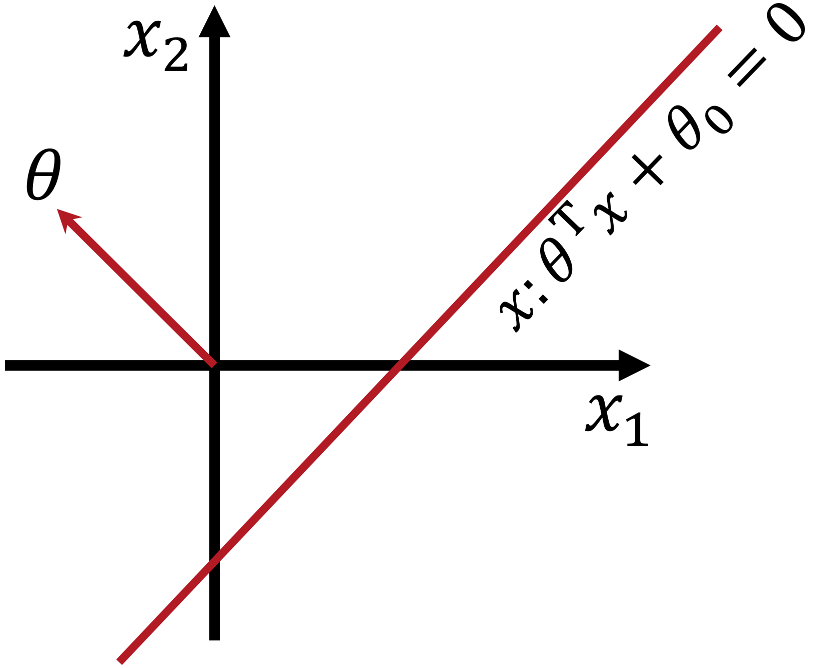
- 线性分类器：

$$h(x;\theta,\theta_0) = \mathrm{sign}(\theta^{\mathrm{T}}x+\theta_0) = \begin{cases} +1, & \text{if } \theta^{\mathrm{T}}x+\theta_0>0 \\ -1, & \text{if } \theta^{\mathrm{T}}x+\theta_0\le 0 \end{cases}$$

回顾：分类器

- 线性分类器：

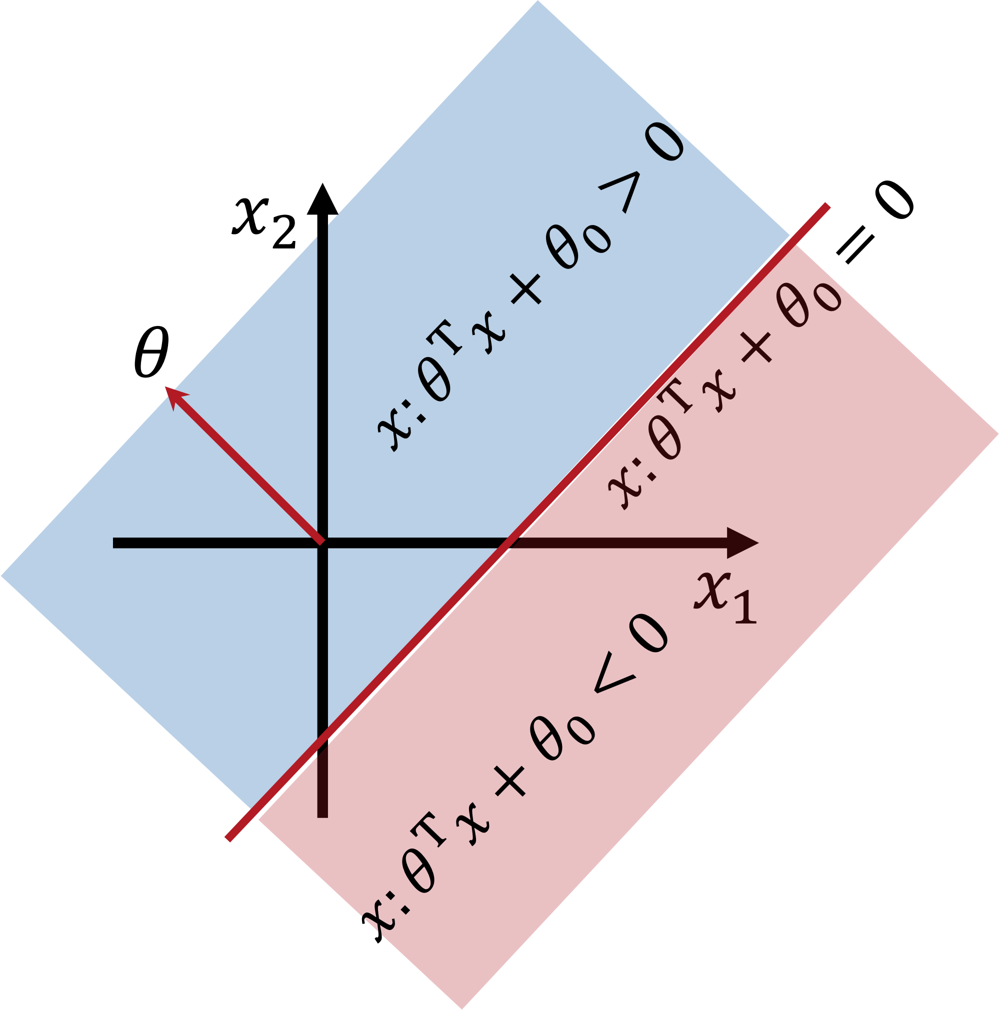
$$h(x;\theta,\theta_0) = \text{sign}(\theta^{\mathrm{T}}x+\theta_0) = \begin{cases} +1, & \text{if } \theta^{\mathrm{T}}x+\theta_0>0 \\ -1, & \text{if } \theta^{\mathrm{T}}x+\theta_0\le 0 \end{cases}$$



回顾：分类器

- 线性分类器：

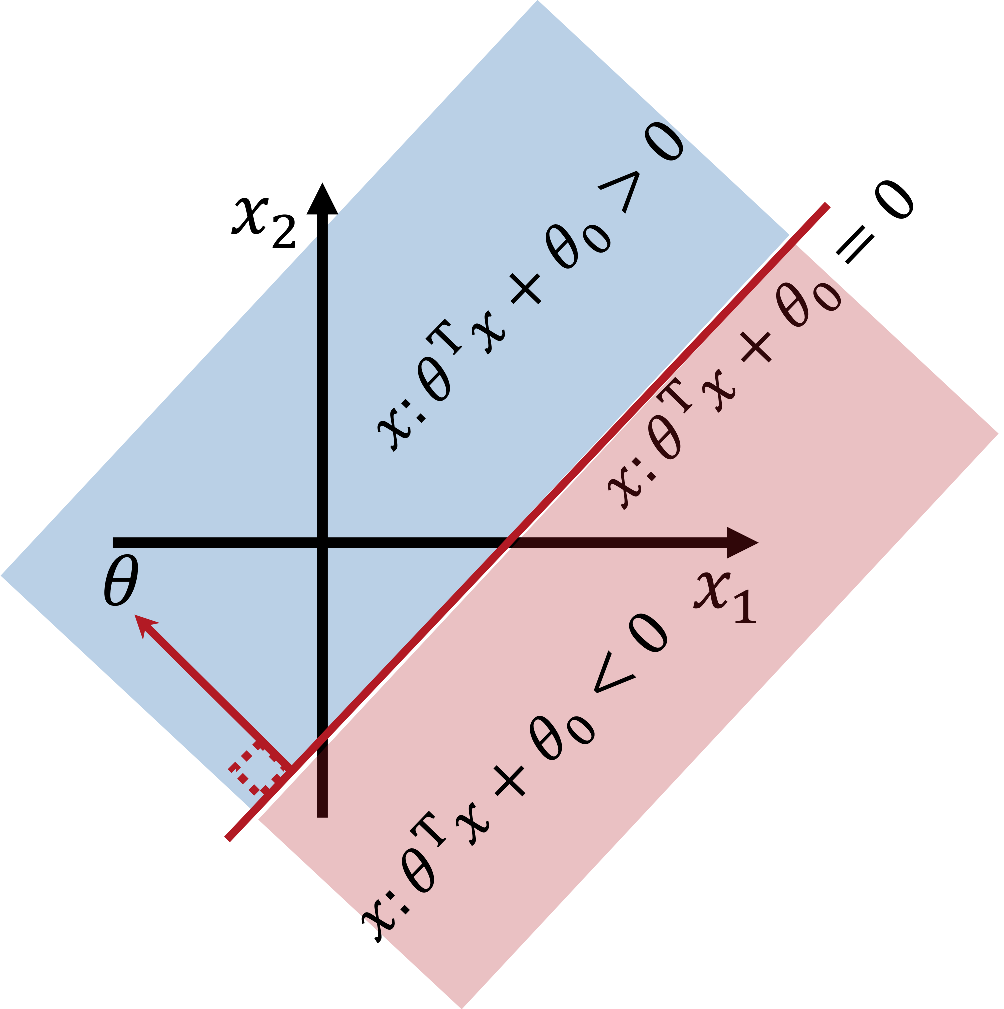
$$
\begin{aligned}
h(x;\theta,\theta_0) &= \mathrm{sign}(\theta^{\mathrm{T}}x+\theta_0) \\
&= \begin{cases} +1, & \text{if } \theta^{\mathrm{T}}x+\theta_0>0 \\ -1, & \text{if } \theta^{\mathrm{T}}x+\theta_0\le 0 \end{cases}
\end{aligned}
$$



回顾：分类器

- 线性分类器：

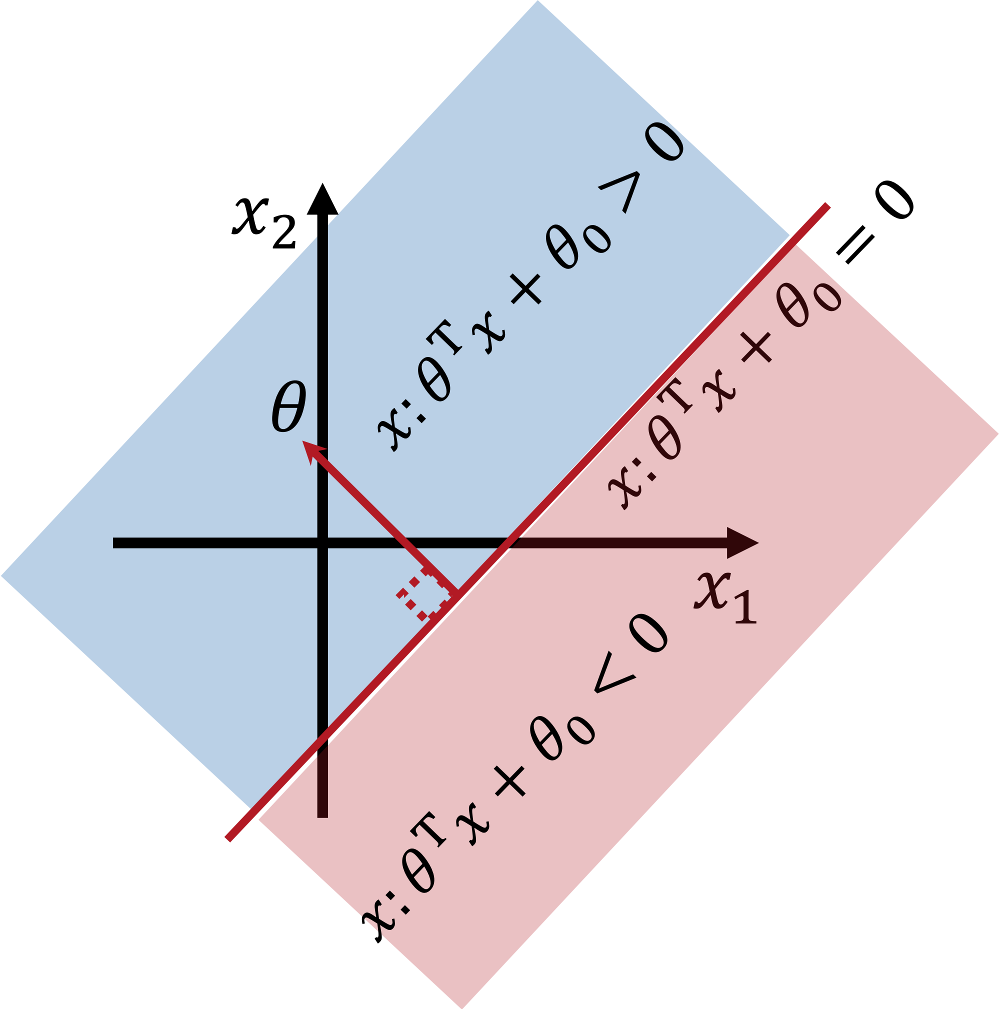
$$\begin{aligned} h(x;\theta,\theta_0) &= \mathrm{sign}(\theta^{\mathrm{T}}x+\theta_0) \\ &= \begin{cases} +1, & \text{if } \theta^{\mathrm{T}}x+\theta_0>0 \\ -1, & \text{if } \theta^{\mathrm{T}}x+\theta_0\le 0 \end{cases} \end{aligned}$$



回顾：分类器

- 线性分类器：

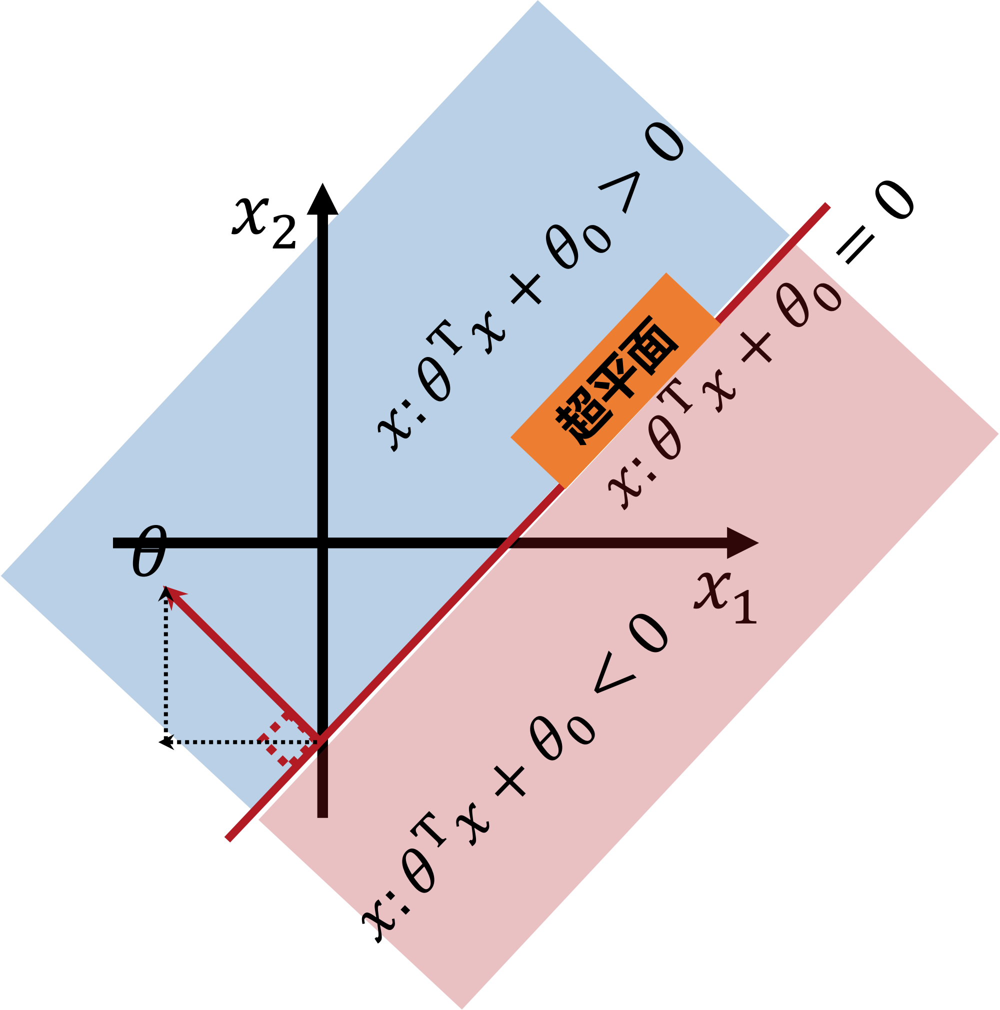
$$
\begin{aligned}
h(x;\theta,\theta_0) &= \operatorname{sign}(\theta^{\mathrm{T}}x+\theta_0) \\
&= \begin{cases} +1, & \text{if } \theta^{\mathrm{T}}x+\theta_0 > 0 \\ -1, & \text{if } \theta^{\mathrm{T}}x+\theta_0 \le 0 \end{cases}
\end{aligned}
$$



回顾：分类器

- 线性分类器：

$$
\begin{aligned}
h(x;\theta,\theta_0) &= \operatorname{sign}(\theta^{\mathrm{T}}x+\theta_0) \\
&= \begin{cases} +1, & \text{if } \theta^{\mathrm{T}}x+\theta_0>0 \\ -1, & \text{if } \theta^{\mathrm{T}}x+\theta_0\le 0 \end{cases}
\end{aligned}
$$

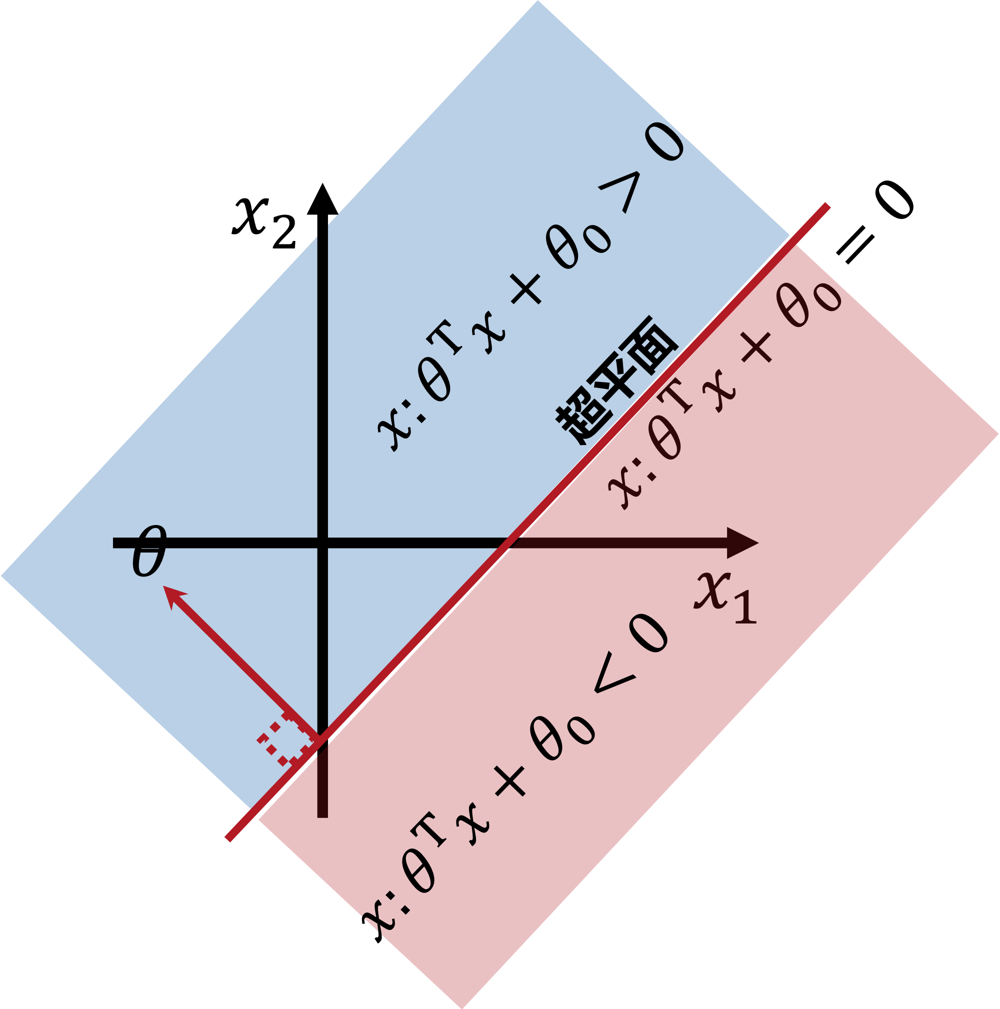


回顾：分类器

- 线性分类器：

$$h(x;\theta,\theta_0) = \mathrm{sign}(\theta^{\mathrm{T}}x+\theta_0) = \begin{cases} +1, & \text{if } \theta^{\mathrm{T}}x+\theta_0>0 \\ -1, & \text{if } \theta^{\mathrm{T}}x+\theta_0\le 0 \end{cases}$$

- 假设类$\mathcal{H}$：所有分类器的集合
- 0-1损失：$L(g,a)=\begin{cases} 0, & \text{if } g=a \\ 1, & \text{else} \end{cases}$
- 训练误差：$\mathcal{E}_n(h)=\frac{1}{n}\sum_{i=1}^{n}L(h(x^{(i)}),y^{(i)})$



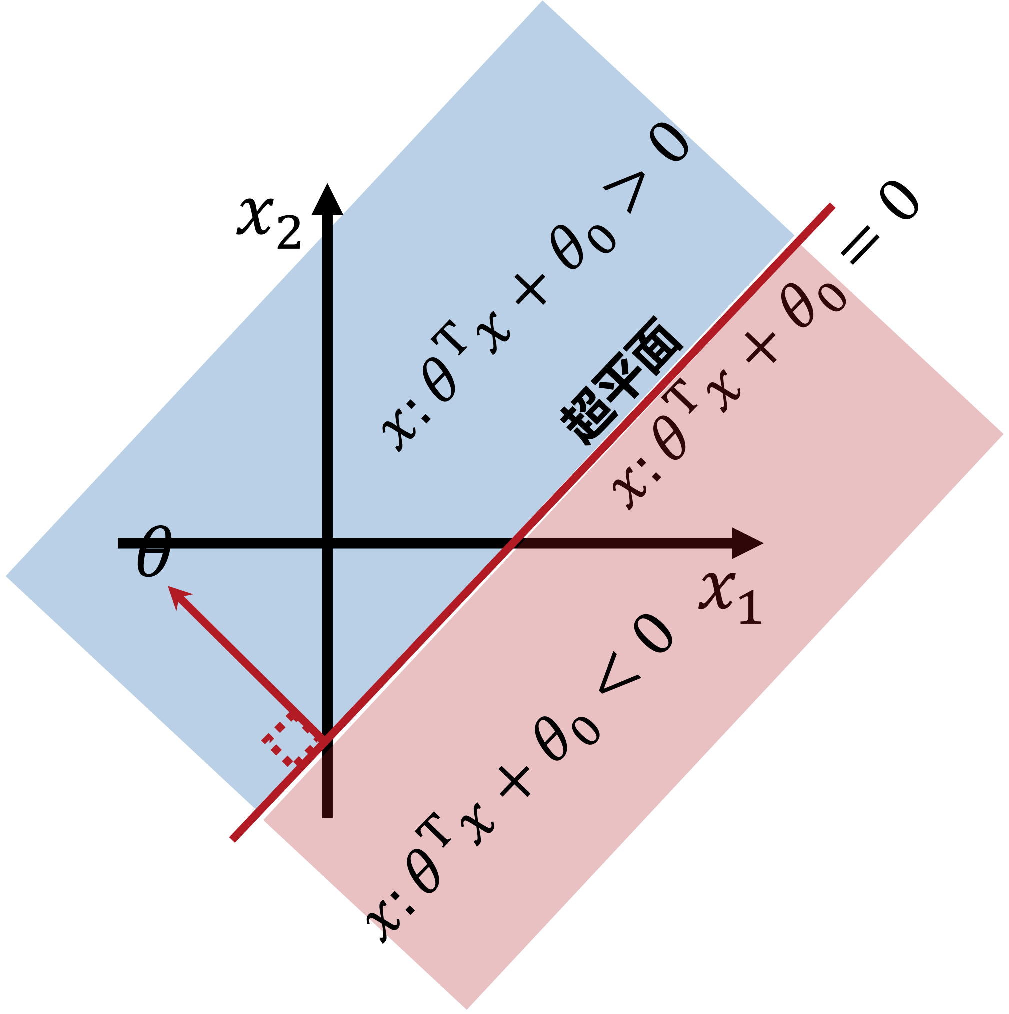
回顾：分类器

- 线性分类器：

$$
\begin{aligned}
h(x;\theta,\theta_0) &= \operatorname{sign}(\theta^{\mathrm{T}}x+\theta_0)\\
&= \begin{cases} +1, & \text{if } \theta^{\mathrm{T}}x+\theta_0>0\\ -1, & \text{if } \theta^{\mathrm{T}}x+\theta_0\le 0\end{cases}
\end{aligned}
$$

- 假设类$\mathcal{H}$：所有分类器的集合
- 0-1损失：$L(g,a)=\begin{cases}0, & \text{if } g=a\\ 1, & \text{else}\end{cases}$
- 训练误差：$\mathcal{E}_n(h)=\frac{1}{n}\sum_{i=1}^{n}L(h(x^{(i)}),y^{(i)})$
- 学习算法Ex：

```
Ex_learning_alg(𝒟_n; k)
    set j* = argmin_{j∈{1,…,k}} ℰ_n(h^(j))
    return h^(j*)
```



感知器算法

感知器算法

Perceptron($\mathcal{D}_n; \tau$)

Initialize $\theta = [0\ 0\ \ldots 0]^T$

Initialize $\theta_0 = 0$

感知器算法

Perceptron($\mathcal{D}_n; \tau$)

    Initialize $\theta = [0\ 0\ \ldots 0]^T$

    Initialize $\theta_0 = 0$

多少个0?

感知器算法

```
Perceptron(𝒟_n; τ)
    Initialize θ = [0 0 … 0]^T
    Initialize θ_0 = 0
    for t = 1 to τ
        changed = False
        for i = 1 to n
            if y^(i)(θ^T x^(i) + θ_0) ≤ 0
```

Perceptron($\mathcal{D}_n;\tau$)

Initialize $\theta = [0\ 0\ \ldots 0]^T$

Initialize $\theta_0 = 0$

for t = 1 to τ

changed = False

for i = 1 to n

if $y^{(i)}\left(\theta^T x^{(i)} + \theta_0\right) \le 0$

多少个0?

感知器算法

```
Perceptron(𝒟_n; τ)
    Initialize θ = [0 0 … 0]^T
    Initialize θ_0 = 0
    for t = 1 to τ
        changed = False
        for i = 1 to n
            if y^(i)(θ^T x^(i) + θ_0) ≤ 0
```

$\text{Perceptron}(\mathcal{D}_n; \tau)$

Initialize $\theta = [0\ 0\ \dots 0]^T$

Initialize $\theta_0 = 0$

for t = 1 to τ

changed = False

for i = 1 to n

if $y^{(i)}\left(\theta^T x^{(i)} + \theta_0\right) \leq 0$

多少个0?

感知器算法

```
Perceptron(𝒟_n; τ)
    Initialize θ = [0 0 … 0]^T
    Initialize θ_0 = 0
    for t = 1 to τ
        changed = False
        for i = 1 to n
            if y^(i)(θ^T x^(i) + θ_0) ≤ 0
```

多少个0?

处于以下三种情况之一时为真：

A. 点不在线上且预测错误

B. 点在线上

感知器算法

```
Perceptron(𝒟_n; τ)
    Initialize θ = [0 0 … 0]^T
    Initialize θ_0 = 0
    for t = 1 to τ
        changed = False
        for i = 1 to n
            if y^(i)(θ^T x^(i) + θ_0) ≤ 0
```

Perceptron($\mathcal{D}_n;\tau$)

Initialize $\theta = [0\ 0\ \dots 0]^T$

Initialize $\theta_0 = 0$

for t = 1 to τ

changed = False

for i = 1 to n

if $y^{(i)}(\theta^T x^{(i)} + \theta_0) \leq 0$

多少个0?

处于以下三种情况之一时为真:

A. 点不在线上且预测错误

B. 点在线上

C. 初始化

感知器算法

```
Perceptron(𝒟_n; τ)
    Initialize θ = [0 0 … 0]^T
    Initialize θ_0 = 0
    for t = 1 to τ
        changed = False
        for i = 1 to n
            if y^(i)(θ^T x^(i) + θ_0) ≤ 0
                Set θ = θ + y^(i) x^(i)
                Set θ_0 = θ_0 + y^(i)
                changed = True
        if not changed
            break
    Return θ, θ_0
```

多少个0?

处于以下三种情况之一时为真:

A. 点不在线上且预测错误

B. 点在线上

C. 初始化

感知器算法

```
Perceptron(𝒟_n; τ)
    Initialize θ = [0 0 … 0]^T
    Initialize θ_0 = 0
    for t = 1 to τ
        changed = False
        for i = 1 to n
            if y^(i)(θ^T x^(i) + θ_0) ≤ 0
                Set θ = θ + y^(i) x^(i)
                Set θ_0 = θ_0 + y^(i)
                changed = True
        if not changed
            break
    Return θ, θ_0
```

多少个0?

处于以下三种情况之一时为真：

A. 点不在线上且预测错误

B. 点在线上

C. 初始化

如何更新?

感知器算法

```
Perceptron(𝒟_n; τ)
    Initialize θ = [0 0 … 0]^T
    Initialize θ_0 = 0
    for t = 1 to τ
        changed = False
        for i = 1 to n
            if y^(i)(θ^T x^(i) + θ_0) ≤ 0
                Set θ = θ + y^(i) x^(i)
                Set θ_0 = θ_0 + y^(i)
                changed = True
        if not changed
            break
    Return θ, θ_0
```

多少个0?

处于以下三种情况之一时为真:

A. 点不在线上且预测错误

B. 点在线上

C. 初始化

如何更新?

$$y^{(i)}\left(\theta_{\text{updated}}^{T} x^{(i)} + \theta_{0,\text{updated}}\right)$$

感知器算法

```
Perceptron(𝒟_n; τ)
    Initialize θ = [0 0 … 0]^T
    Initialize θ_0 = 0
    for t = 1 to τ
        changed = False
        for i = 1 to n
            if y^(i)(θ^T x^(i) + θ_0) ≤ 0
                Set θ = θ + y^(i) x^(i)
                Set θ_0 = θ_0 + y^(i)
                changed = True
        if not changed
            break
    Return θ, θ_0
```

多少个0?

处于以下三种情况之一时为真:

A. 点不在线上且预测错误

B. 点在线上

C. 初始化

如何更新?

$$y^{(i)}\left(\left(\theta + y^{(i)}x^{(i)}\right)^T x^{(i)} + \left(\theta_0 + y^{(i)}\right)\right)$$

$$= y^{(i)}\left(\theta^T x^{(i)} + \theta_0\right) + \left(y^{(i)}\right)^2\left(x^{(i)T}x^{(i)} + 1\right)$$

感知器算法

```
Perceptron(𝒟_n; τ)
    Initialize θ = [0 0 … 0]^T
    Initialize θ_0 = 0
    for t = 1 to τ
        changed = False
        for i = 1 to n
            if y^(i)(θ^T x^(i) + θ_0) ≤ 0
                Set θ = θ + y^(i) x^(i)
                Set θ_0 = θ_0 + y^(i)
                changed = True
        if not changed
            break
    Return θ, θ_0
```

多少个0?

处于以下三种情况之一时为真:

A. 点不在线上且预测错误

B. 点在线上

C. 初始化

如何更新?

$$y^{(i)}\left(\left(\theta + y^{(i)}x^{(i)}\right)^T x^{(i)} + \left(\theta_0 + y^{(i)}\right)\right)$$

$$= y^{(i)}\left(\theta^T x^{(i)} + \theta_0\right) + \left(y^{(i)}\right)^2\left(x^{(i)T}x^{(i)} + 1\right)$$

感知器算法

Perceptron($\mathcal{D}_n$; τ)
    Initialize $\theta = [0\ 0\ \dots 0]^T$
    Initialize $\theta_0 = 0$
    for t = 1 to τ
        changed = False
        for i = 1 to n
            if $y^{(i)}\left(\theta^T x^{(i)} + \theta_0\right) \le 0$
                Set $\theta = \theta + y^{(i)} x^{(i)}$
                Set $\theta_0 = \theta_0 + y^{(i)}$
                changed = True
        if not changed
            break
    Return θ, θ_0

多少个0?

处于以下三种情况之一时为真：

A. 点不在线上且预测错误
B. 点在线上
C. 初始化

如何更新?

$$y^{(i)}\left(\left(\theta + y^{(i)} x^{(i)}\right)^T x^{(i)} + \left(\theta_0 + y^{(i)}\right)\right)$$

$$= y^{(i)}\left(\theta^T x^{(i)} + \theta_0\right) + \left(y^{(i)}\right)^2\left(x^{(i)T} x^{(i)} + 1\right)$$

感知器算法

```
Perceptron(𝒟_n; τ)
    Initialize θ = [0 0 … 0]^T
    Initialize θ_0 = 0
    for t = 1 to τ
        changed = False
        for i = 1 to n
            if y^(i)(θ^T x^(i) + θ_0) ≤ 0
                Set θ = θ + y^(i) x^(i)
                Set θ_0 = θ_0 + y^(i)
                changed = True
        if not changed
            break
    Return θ, θ_0
```

多少个0?

处于以下三种情况之一时为真:

A. 点不在线上且预测错误

B. 点在线上

C. 初始化

如何更新?

$$y^{(i)}\left(\left(\theta + y^{(i)}x^{(i)}\right)^T x^{(i)} + \left(\theta_0 + y^{(i)}\right)\right)$$

$$= y^{(i)}\left(\theta^T x^{(i)} + \theta_0\right) + \left(y^{(i)}\right)^2\left(x^{(i)T}x^{(i)} + 1\right)$$

感知器算法

```
Perceptron(𝒟_n; τ)
    Initialize θ = [0 0 … 0]^T
    Initialize θ_0 = 0
    for t = 1 to τ
        changed = False
        for i = 1 to n
            if y^(i)(θ^T x^(i) + θ_0) ≤ 0
                Set θ = θ + y^(i) x^(i)
                Set θ_0 = θ_0 + y^(i)
                changed = True
        if not changed
            break
    Return θ, θ_0
```

多少个0?

处于以下三种情况之一时为真:

A. 点不在线上且预测错误

B. 点在线上

C. 初始化

如何更新?

$$y^{(i)}\left(\left(\theta + y^{(i)}x^{(i)}\right)^T x^{(i)} + \left(\theta_0 + y^{(i)}\right)\right)$$

$$= y^{(i)}\left(\theta^T x^{(i)} + \theta_0\right) + \left(y^{(i)}\right)^2\left(x^{(i)T}x^{(i)} + 1\right)$$

感知器算法

```
Perceptron(𝒟_n; τ)
    Initialize θ = [0 0 … 0]^T
    Initialize θ_0 = 0
    for t = 1 to τ
        changed = False
        for i = 1 to n
            if y^(i)(θ^T x^(i) + θ_0) ≤ 0
                Set θ = θ + y^(i) x^(i)
                Set θ_0 = θ_0 + y^(i)
                changed = True
        if not changed
            break
    Return θ, θ_0
```

多少个0?

处于以下三种情况之一时为真:

A. 点不在线上且预测错误

B. 点在线上

C. 初始化

如何更新?

$$y^{(i)}\left(\left(\theta+y^{(i)}x^{(i)}\right)^{T}x^{(i)}+\left(\theta_0+y^{(i)}\right)\right)$$

$$=y^{(i)}\left(\theta^{T}x^{(i)}+\theta_0\right)+\left(y^{(i)}\right)^{2}\left(x^{(i)T}x^{(i)}+1\right)$$

感知器算法

```
Perceptron(𝒟_n; τ)
    Initialize θ = [0 0 … 0]^T
    Initialize θ_0 = 0
    for t = 1 to τ
        changed = False
        for i = 1 to n
            if y^(i)(θ^T x^(i) + θ_0) ≤ 0
                Set θ = θ + y^(i) x^(i)
                Set θ_0 = θ_0 + y^(i)
                changed = True
        if not changed
            break
    Return θ, θ_0
```

多少个0?

处于以下三种情况之一时为真:

A. 点不在线上且预测错误

B. 点在线上

C. 初始化

如何更新?

$$y^{(i)}\left(\left(\theta + y^{(i)}x^{(i)}\right)^T x^{(i)} + \left(\theta_0 + y^{(i)}\right)\right)$$

$$= y^{(i)}\left(\theta^T x^{(i)} + \theta_0\right) + \left(y^{(i)}\right)^2 \left(x^{(i)T} x^{(i)} + 1\right)$$

感知器算法

```
Perceptron(𝒟_n; τ)
    Initialize θ = [0 0 … 0]^T
    Initialize θ_0 = 0
    for t = 1 to τ
        changed = False
        for i = 1 to n
            if y^(i)(θ^T x^(i) + θ_0) ≤ 0
                Set θ = θ + y^(i) x^(i)
                Set θ_0 = θ_0 + y^(i)
                changed = True
        if not changed
            break
    Return θ, θ_0
```

多少个0?

处于以下三种情况之一时为真:

A. 点不在线上且预测错误

B. 点在线上

C. 初始化

如何更新?

$$y^{(i)}\left(\left(\theta + y^{(i)}x^{(i)}\right)^T x^{(i)} + \left(\theta_0 + y^{(i)}\right)\right)$$

$$= y^{(i)}\left(\theta^T x^{(i)} + \theta_0\right) + \left(y^{(i)}\right)^2\left(x^{(i)T}x^{(i)} + 1\right)$$

$$= y^{(i)}\left(\theta^T x^{(i)} + \theta_0\right) + \left(\left\|x^{(i)}\right\|^2 + 1\right)$$

感知器算法

```
Perceptron(𝒟_n; τ)
    Initialize θ = [0 0 … 0]^T
    Initialize θ_0 = 0
    for t = 1 to τ
        changed = False
        for i = 1 to n
            if y^(i)(θ^T x^(i) + θ_0) ≤ 0
                Set θ = θ + y^(i) x^(i)
                Set θ_0 = θ_0 + y^(i)
                changed = True
        if not changed
            break
    Return θ, θ_0
```

多少个0?

处于以下三种情况之一时为真:

A. 点不在线上且预测错误

B. 点在线上

C. 初始化

如何更新?

$$y^{(i)}\left(\left(\theta + y^{(i)}x^{(i)}\right)^T x^{(i)} + \left(\theta_0 + y^{(i)}\right)\right)$$

$$= y^{(i)}\left(\theta^T x^{(i)} + \theta_0\right) + \left(y^{(i)}\right)^2\left(x^{(i)T}x^{(i)} + 1\right)$$

$$= y^{(i)}\left(\theta^T x^{(i)} + \theta_0\right) + \left(\left\|x^{(i)}\right\|^2 + 1\right)$$

数据评估

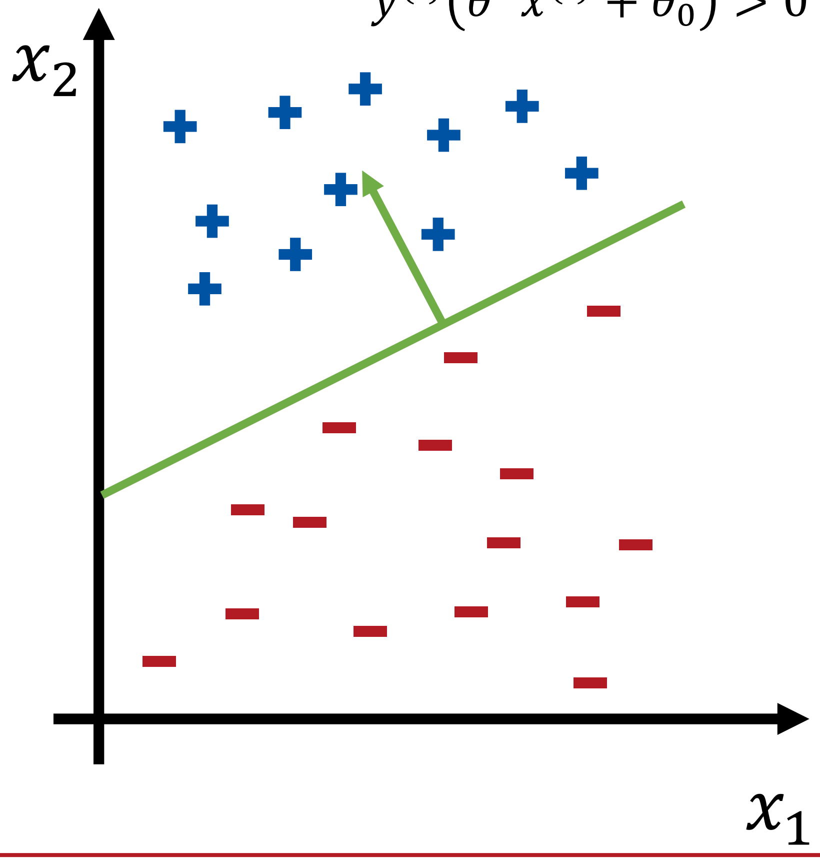
- 定义：对于任意一个训练集$\mathcal{D}_n$，如果存在(θ, θ_0)，使得数据集中的每个点$i \in \{1, \dots, n\}$，都满足下式，则称训练集$\mathcal{D}_n$是**线性可分**的

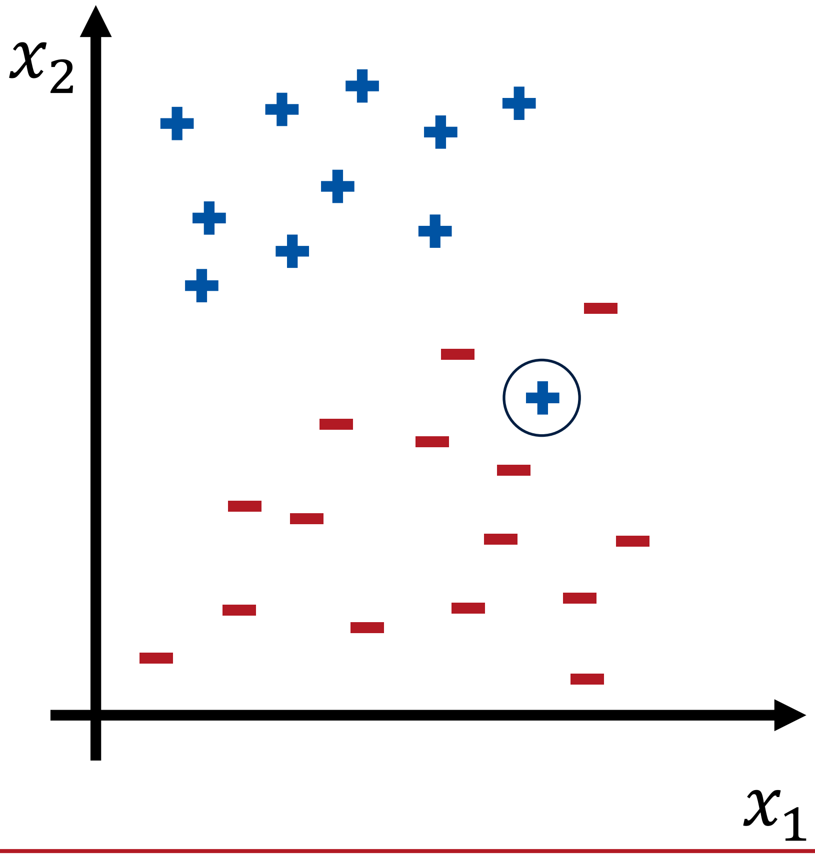
$$y^{(i)}\left(\theta^T x^{(i)} + \theta_0\right) > 0$$

数据评估

- 定义：对于任意一个训练集$\mathcal{D}_n$，如果存在(θ, θ_0)，使得数据集中的每个点$i \in \{1, \dots, n\}$，都满足下式，则称训练集$\mathcal{D}_n$是**线性可分**的

$$y^{(i)}\left(\theta^T x^{(i)} + \theta_0\right) > 0$$

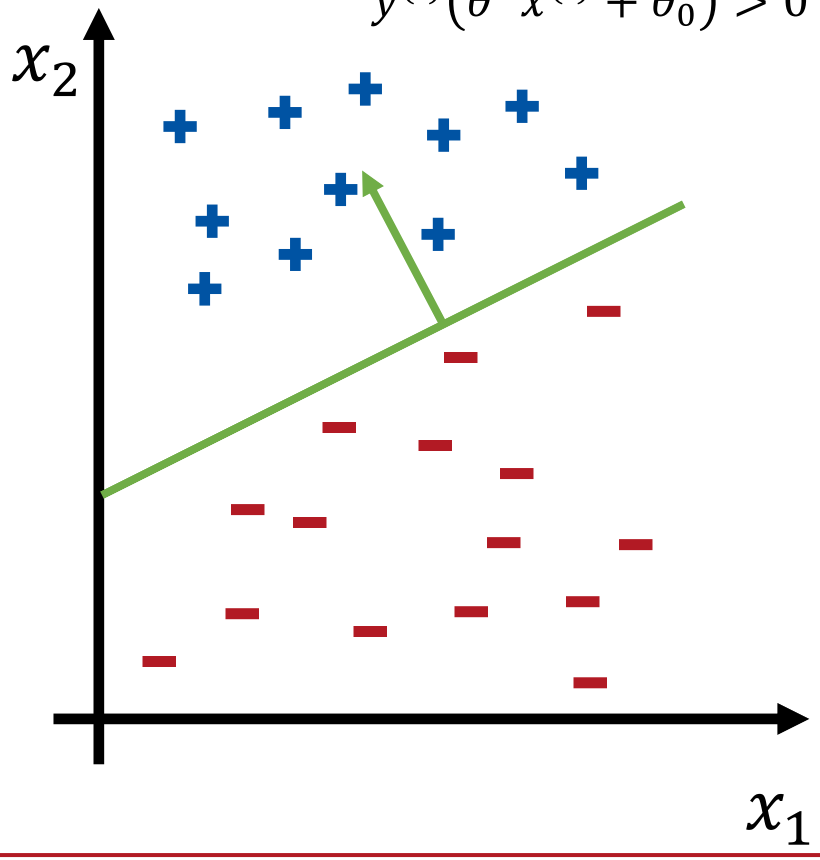


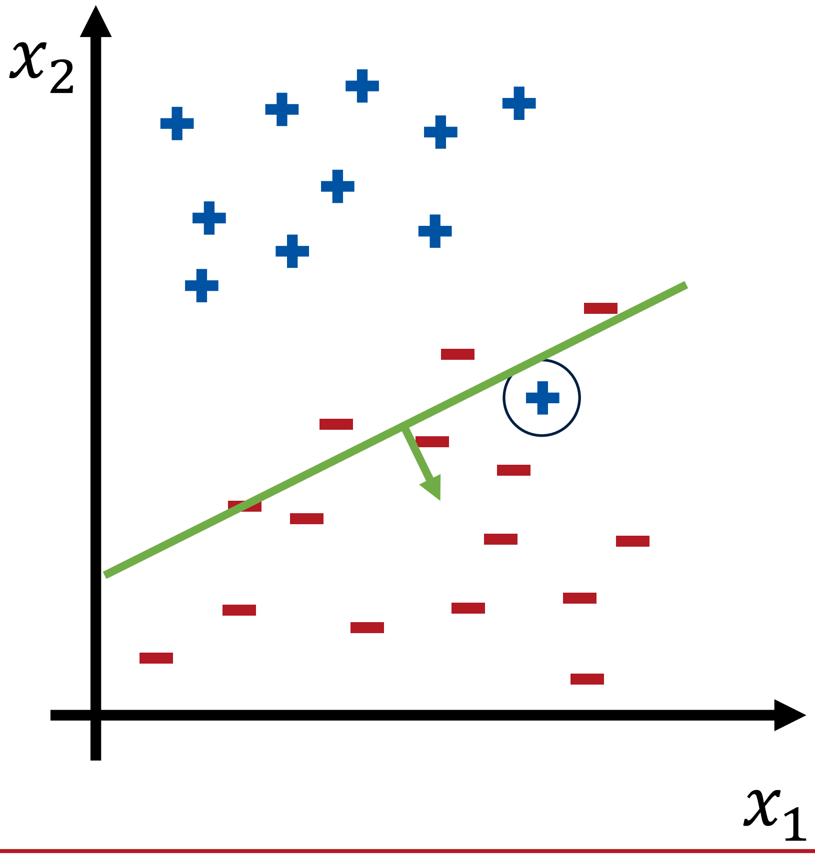


数据评估

- 定义：对于任意一个训练集$\mathcal{D}_n$，如果存在(θ, θ_0)，使得数据集中的每个点$i \in \{1, \dots, n\}$，都满足下式，则称训练集$\mathcal{D}_n$是**线性可分**的

$$y^{(i)}\left(\theta^T x^{(i)} + \theta_0\right) > 0$$

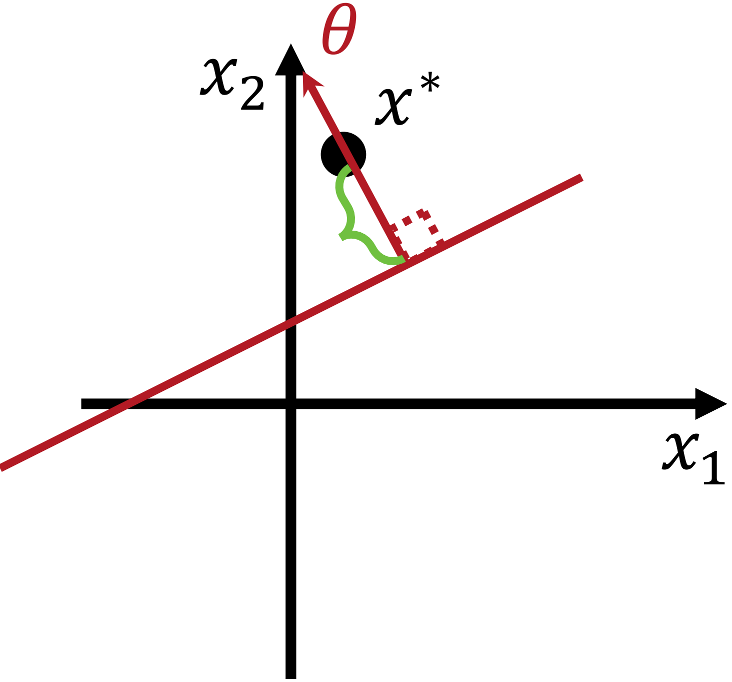




数据评估

数学视角

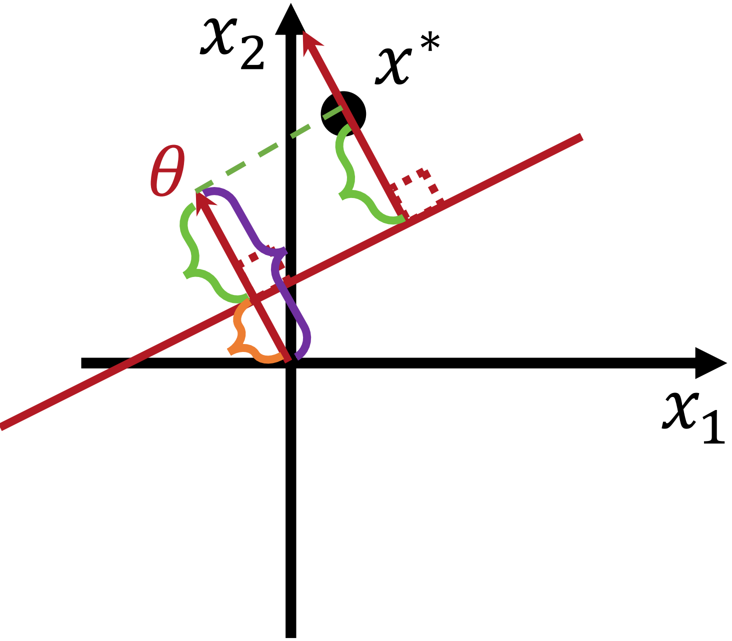
- 从超平面(θ,θ_0)到点x^*的距离



数据评估

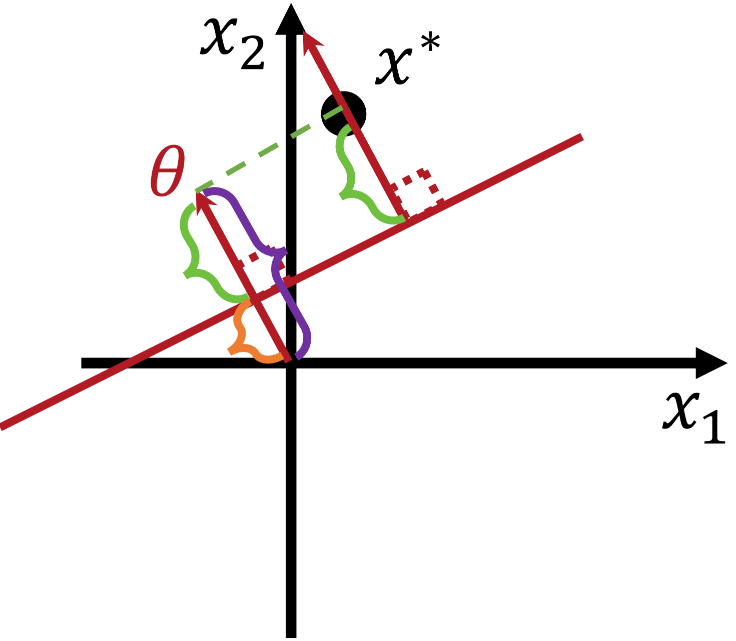
数学视角

- 从超平面(θ,θ_0)到点x^*的距离



数据评估

数学视角



- 从超平面(θ, θ_0)到点x^*的距离

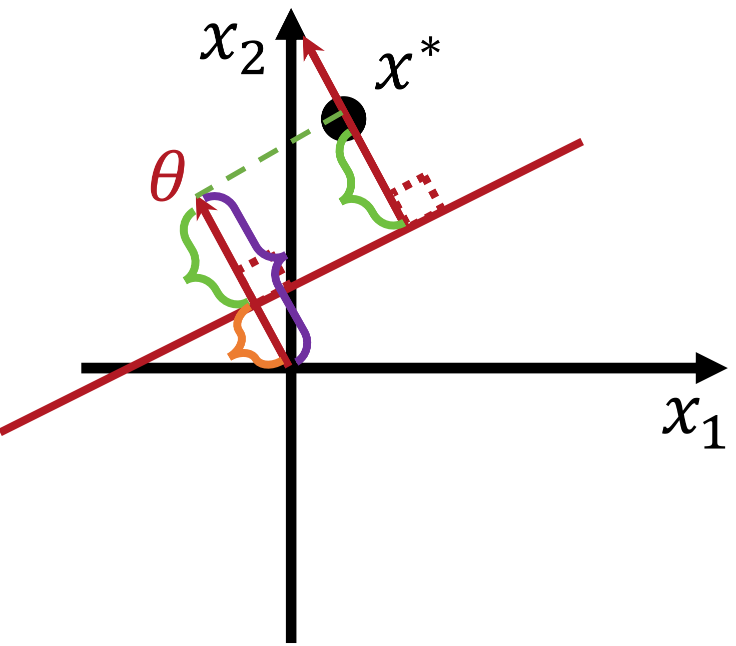
  = x^*在θ上的映射

  − 超平面到原点的距离

$$= \frac{\theta^T x^*}{\|\theta\|} - \frac{-\theta_0}{\|\theta\|} = \frac{\theta^T x^* + \theta_0}{\|\theta\|}$$

数据评估

数学视角



- 从超平面(θ,θ_0)到点x^*的距离

  $= x^*$在θ上的映射

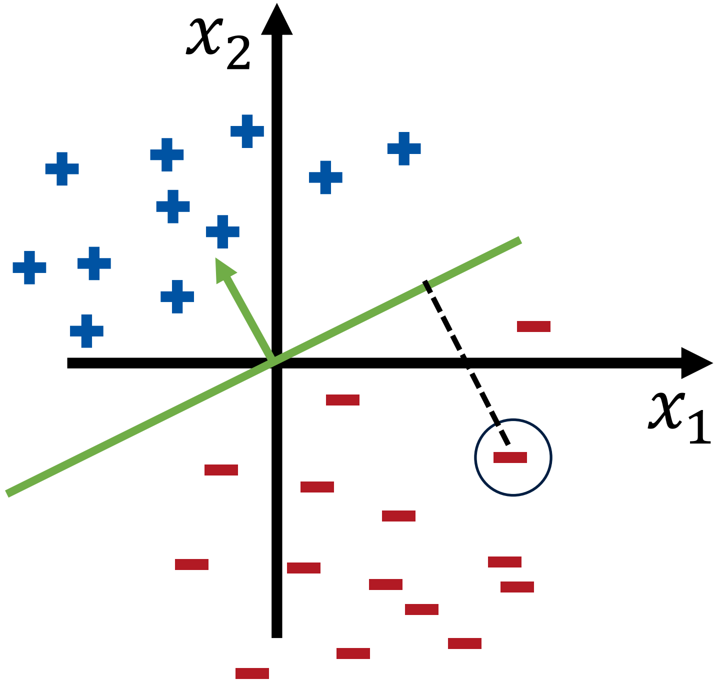
  $-$ 超平面到原点的距离

  $$= \frac{\theta^T x^*}{\|\theta\|} - \frac{-\theta_0}{\|\theta\|} = \frac{\theta^T x^* + \theta_0}{\|\theta\|}$$

- 定义：对于由(θ,θ_0)定义的超平面，带标签的样本点(x^*,y^*)到超平面的**边界(margin)**定义为：

$$y^*\left(\frac{\theta^T x^* + \theta_0}{\|\theta\|}\right)$$

数据评估



- 从超平面(θ,θ_0)到点x^*的距离

$$= x^*\text{在}\theta\text{上的映射}$$

$$- \text{超平面到原点的距离}$$

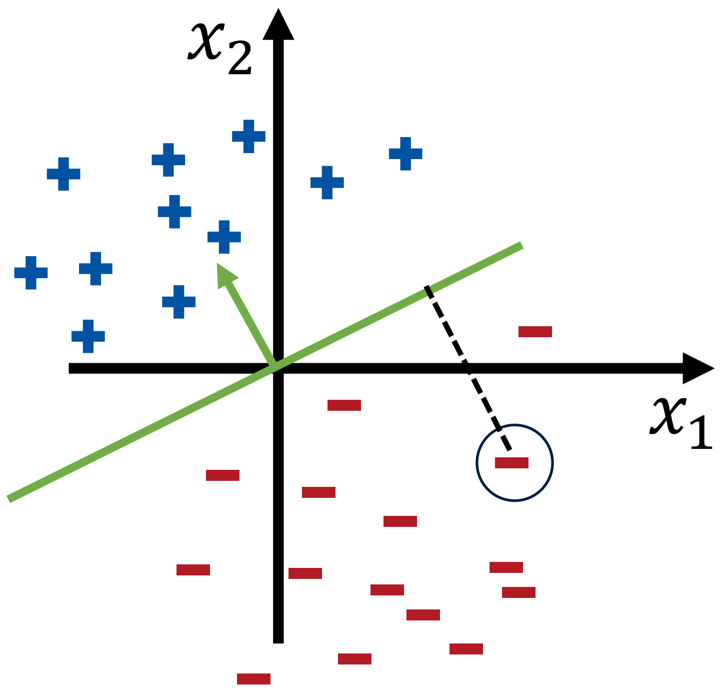
$$= \frac{\theta^T x^*}{\|\theta\|} - \frac{-\theta_0}{\|\theta\|} = \frac{\theta^T x^* + \theta_0}{\|\theta\|}$$

- 定义：对于由(θ,θ_0)定义的超平面，带标签的样本点(x^*,y^*)到超平面的**边界(margin)**定义为：

$$y^*\left(\frac{\theta^T x^* + \theta_0}{\|\theta\|}\right)$$

数据评估

数学视角



- 从超平面(θ,θ_0)到点x^*的距离

  $= x^*$在θ上的映射

  $-$ 超平面到原点的距离

  $= \frac{\theta^T x^*}{\|\theta\|} - \frac{-\theta_0}{\|\theta\|} = \frac{\theta^T x^* + \theta_0}{\|\theta\|}$

- 定义：对于由(θ,θ_0)定义的超平面，带标签的样本点(x^*, y^*)到超平面的**边界(margin)**定义为：

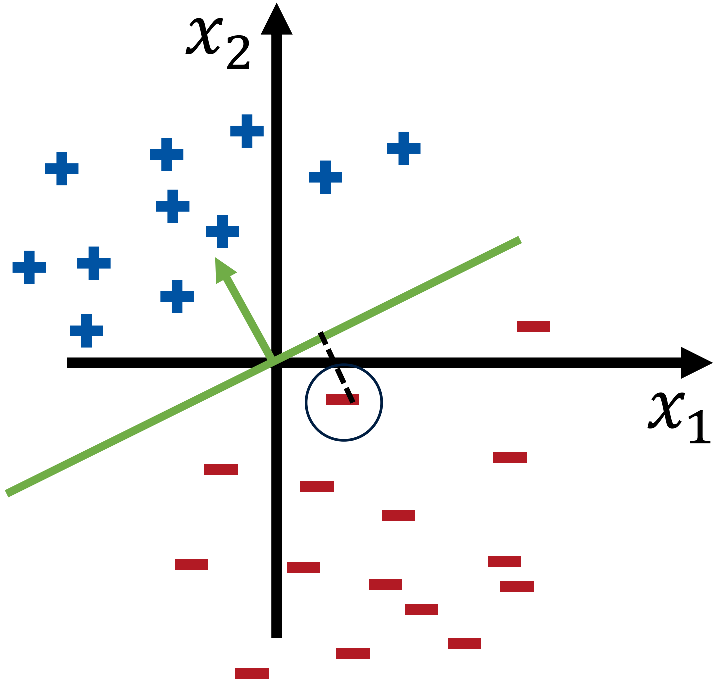
$$y^* \left(\frac{\theta^T x^* + \theta_0}{\|\theta\|} \right)$$

- 定义：对于由(θ,θ_0)定义的超平面，训练集$\mathcal{D}_n$到超平面的边界(margin)定义为：

$$\min_{i \in \{1,\ldots,n\}} y^{(i)} \left(\frac{\theta^T x^{(i)} + \theta_0}{\|\theta\|} \right)$$

数据评估

数学视角



- 从超平面(θ,θ_0)到点x^*的距离

  $= x^*$在θ上的映射

  $-$ 超平面到原点的距离

  $$= \frac{\theta^T x^*}{\|\theta\|} - \frac{-\theta_0}{\|\theta\|} = \frac{\theta^T x^* + \theta_0}{\|\theta\|}$$

- 定义：对于由(θ,θ_0)定义的超平面，带标签的样本点(x^*,y^*)到超平面的**边界(margin)**定义为：

$$y^*\left(\frac{\theta^T x^* + \theta_0}{\|\theta\|}\right)$$

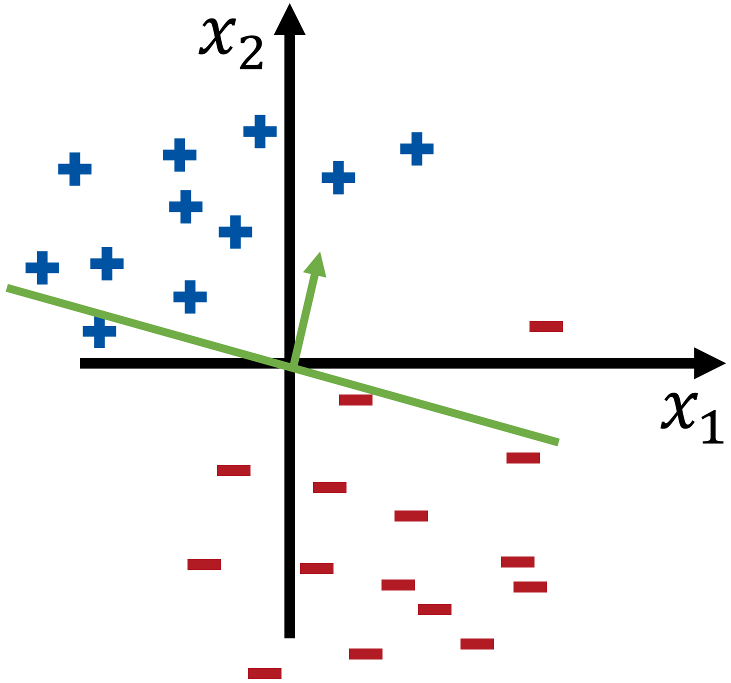
- 定义：对于由(θ,θ_0)定义的超平面，训练集$\mathcal{D}_n$到超平面的边界(margin)定义为：

$$\min_{i\in\{1,\ldots,n\}} y^{(i)}\left(\frac{\theta^T x^{(i)} + \theta_0}{\|\theta\|}\right)$$

定理：感知器性能

- 假设：

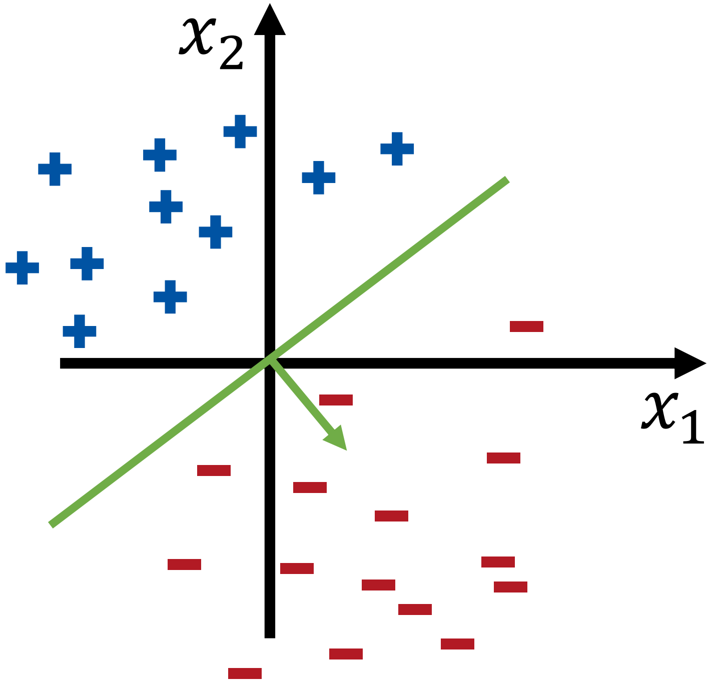
  A. 假设类 = 由穿过原点的超平面所定义的分类器集合（$\theta_0 = 0$）



定理：感知器性能

- 假设：

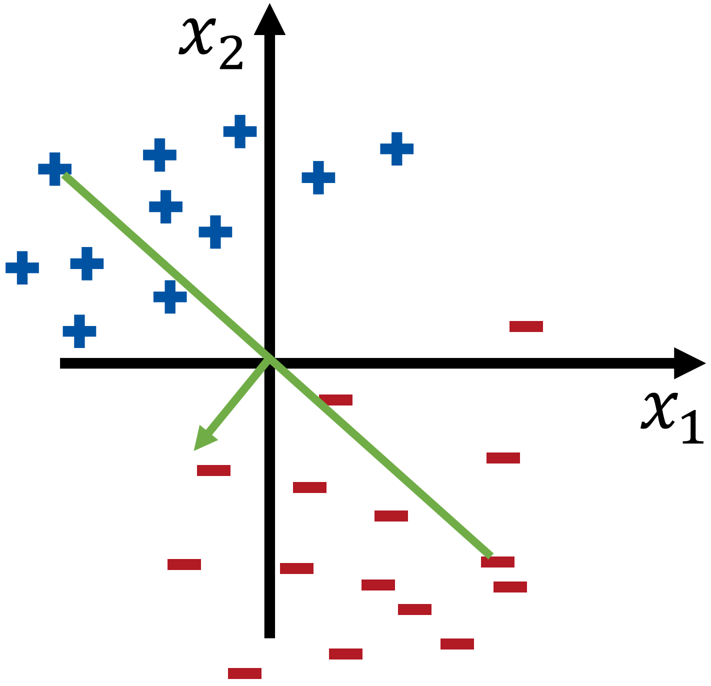
  A. 假设类 = 由穿过原点的超平面所定义的分类器集合（$\theta_0 = 0$）



定理：感知器性能

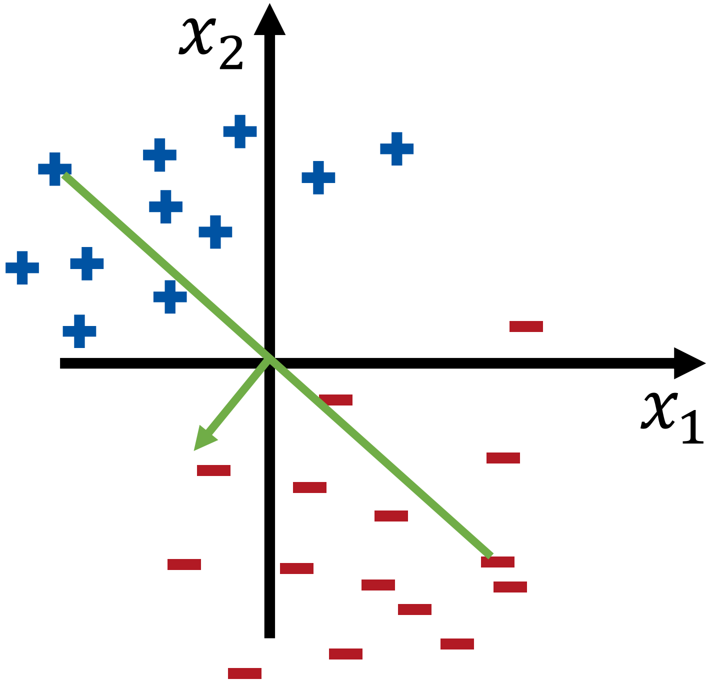
- **假设：**

  A. 假设类 = 由穿过原点的超平面所定义的分类器集合（$\theta_0 = 0$）



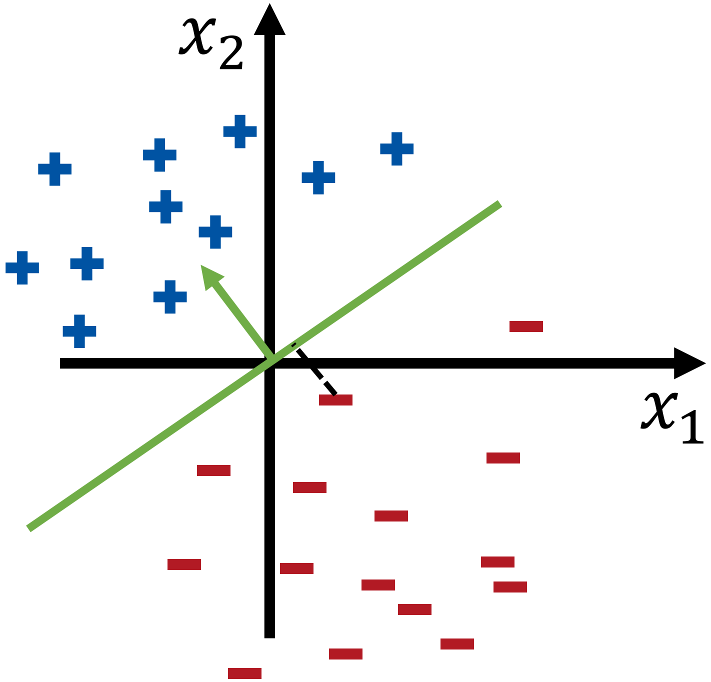
定理：感知器性能

- **假设：**
  A. 假设类 = 由穿过原点的超平面所定义的分类器集合（$\theta_0 = 0$）
  B. 存在θ^*和$\gamma > 0$，对于所有的样本点$i \in \{1, \dots, n\}$，满足$y^{(i)}\left(\frac{\theta^{*T} x^{(i)}}{\|\theta\|}\right) > \gamma$



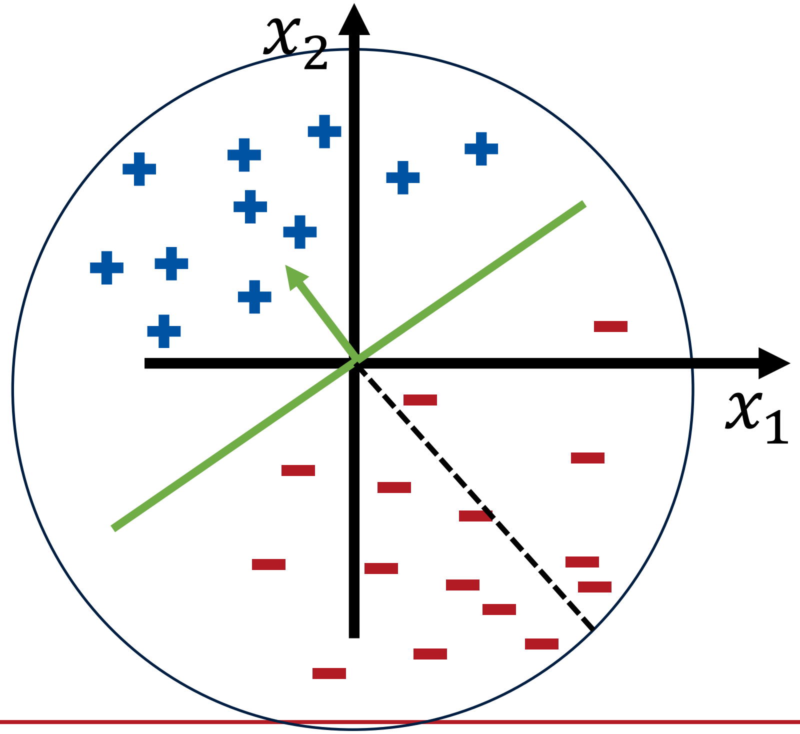
定理：感知器性能

- **假设：**
  A. 假设类 = 由穿过原点的超平面所定义的分类器集合（$\theta_0 = 0$）
  B. 存在θ^*和$\gamma > 0$，对于所有的样本点$i \in \{1, \dots, n\}$，满足$y^{(i)}\left(\frac{\theta^{*T} x^{(i)}}{\|\theta\|}\right) > \gamma$



定理：感知器性能

- **假设：**
  A. 假设类 = 由穿过原点的超平面所定义的分类器集合（$\theta_0 = 0$）
  B. 存在θ^*和$\gamma > 0$，对于所有的样本点$i \in \{1, \dots, n\}$，满足$y^{(i)}\left(\frac{\theta^{*T} x^{(i)}}{\|\theta\|}\right) > \gamma$
  C. 存在R使得对于所有的样本点$i \in \{1, \dots, n\}$，满足$\left\|x^{(i)}\right\| \leq R$



定理：感知器性能

- **假设：**

A. 假设类 = 由穿过原点的超平面所定义的分类器集合（$\theta_0 = 0$）

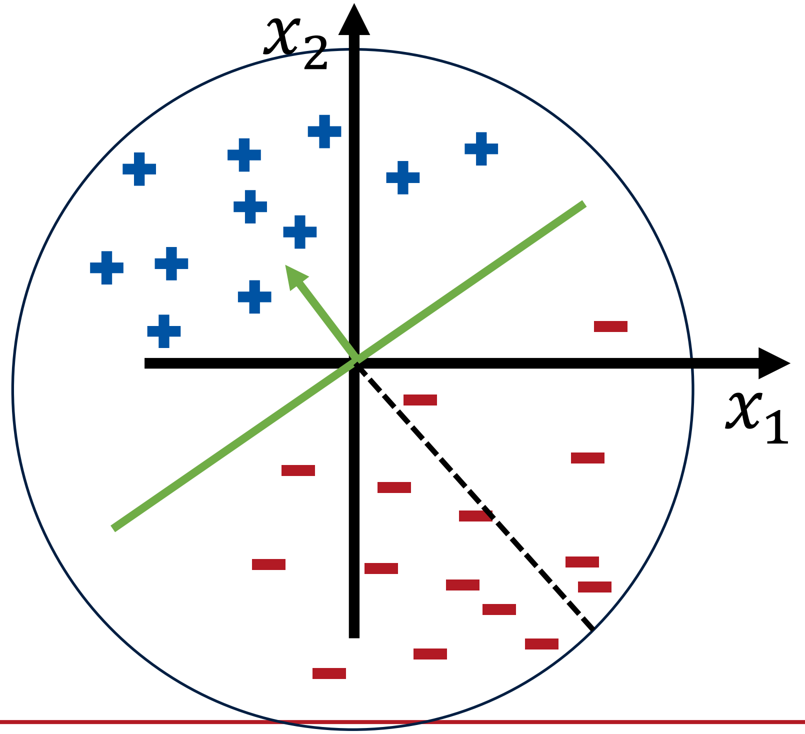
B. 存在θ^*和$\gamma > 0$，对于所有的样本点$i \in \{1, \dots, n\}$，满足$y^{(i)} \left(\frac{\theta^{*T} x^{(i)}}{\|\theta\|} \right) > \gamma$

C. 存在R使得对于所有的样本点$i \in \{1, \dots, n\}$，满足$\left\| x^{(i)} \right\| \leq R$

- **结论：**

感知器算法对参数θ的更新次数最多为$\left(\frac{R}{\gamma}\right)^2$，

若对所有样本点的遍历中没有更新，则该算法所学到的假设在训练集上的训练误差为0



感知器与原点

- 有偏置项的分类器

  $x \in \mathbb{R}^d$， $\theta \in \mathbb{R}^d$， $\theta_0 \in \mathbb{R}$

  $x: \theta^T x + \theta_0 = 0$

感知器与原点

- 有偏置项的分类器

  $x \in \mathbb{R}^d$， $\theta \in \mathbb{R}^d$， $\theta_0 \in \mathbb{R}$

  $x: \theta^T x + \theta_0 = 0$

- 无偏置项的分类器

  $x_{new} \in \mathbb{R}^{d+1}$， $\theta_{new} \in \mathbb{R}^{d+1}$

  $x_{new} = [x_1, x_2, \dots, x_d, 1]$， $\theta_{new} = [\theta_1, \theta_2, \dots, \theta_d, \theta_0]$

  $x_{new,1:d}: \theta_{new}^T \cdot x_{new} = 0$

感知器与原点

- 有偏置项的分类器

  $x \in \mathbb{R}^d,\quad \theta \in \mathbb{R}^d,\quad \theta_0 \in \mathbb{R}$

  $x: \theta^T x + \theta_0 \leq 0$

- 无偏置项的分类器

  $x_{new} \in \mathbb{R}^{d+1},\quad \theta_{new} \in \mathbb{R}^{d+1}$

  $x_{new} = [x_1, x_2, \dots, x_d, 1]$，$\theta_{new} = [\theta_1, \theta_2, \dots, \theta_d, \theta_0]$

  $x_{new,1:d}: \theta_{new}^T \cdot x_{new} \leq 0$

感知器与原点

- 有偏置项的分类器

  $x \in \mathbb{R}^d, \quad \theta \in \mathbb{R}^d, \quad \theta_0 \in \mathbb{R}$

  $x: \theta^T x + \theta_0 \underset{>}{\overset{<}{=}} 0$

- 无偏置项的分类器

  $x_{new} \in \mathbb{R}^{d+1}, \quad \theta_{new} \in \mathbb{R}^{d+1}$

  $x_{new} = [x_1, x_2, \dots, x_d, 1], \quad \theta_{new} = [\theta_1, \theta_2, \dots, \theta_d, \theta_0]$

  $x_{new,1:d}: \theta_{new}^T \cdot x_{new} \underset{>}{\overset{<}{=}} 0$

感知器与原点

- 有偏置项的分类器

  $x \in \mathbb{R}^d$， $\theta \in \mathbb{R}^d$， $\theta_0 \in \mathbb{R}$

  $x: \theta^T x + \theta_0 \overset{<}{\underset{>}{=}} 0$

- 无偏置项的分类器

  $x_{new} \in \mathbb{R}^{d+1}$， $\theta_{new} \in \mathbb{R}^{d+1}$

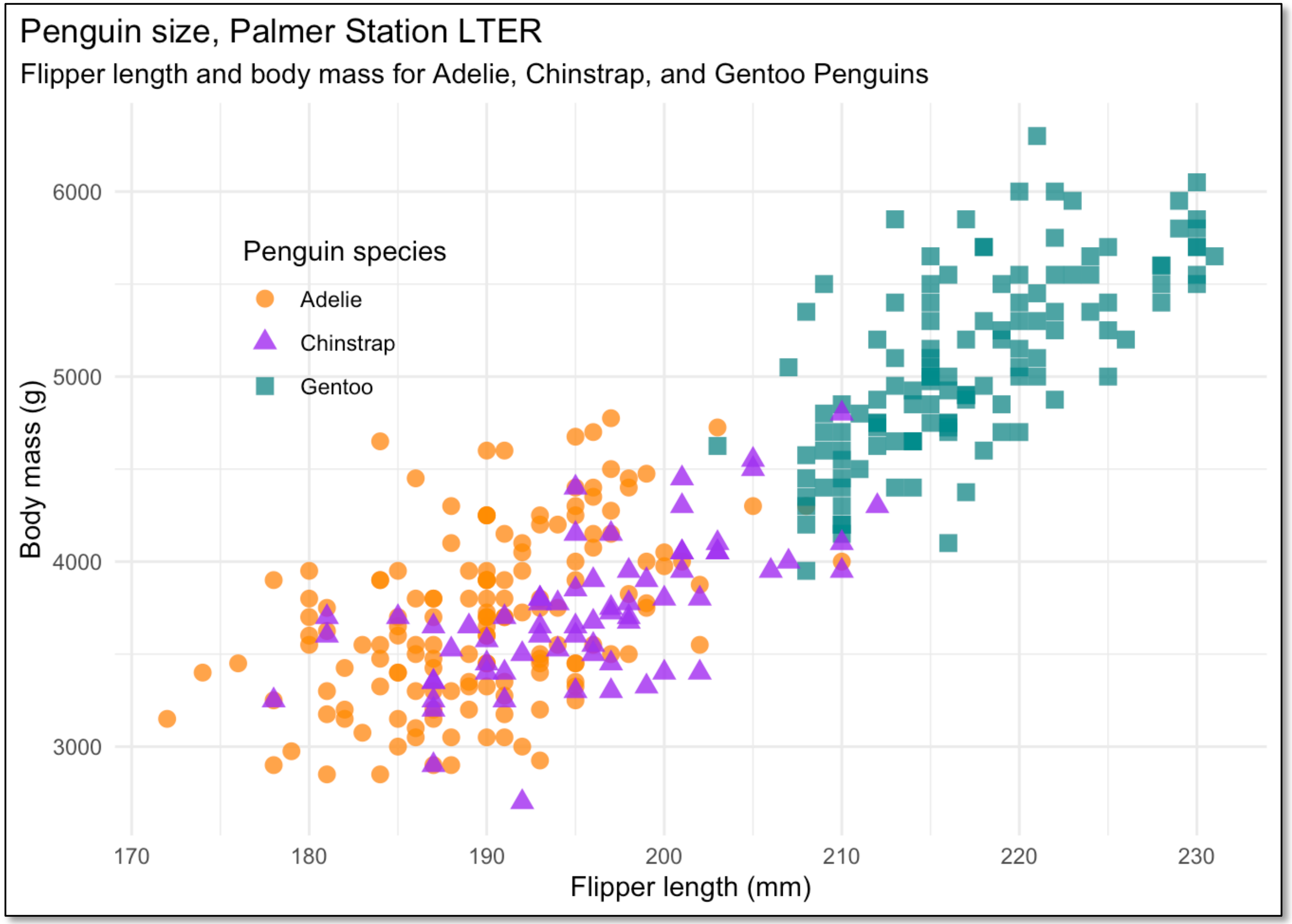
  $x_{new} = [x_1, x_2, \dots, x_d, 1]$， $\theta_{new} = [\theta_1, \theta_2, \dots, \theta_d, \theta_0]$

  $x_{new,1:d}: \theta_{new}^T \cdot x_{new} \overset{<}{\underset{>}{=}} 0$

- 可以将特征转换到扩展的特征空间，然后再应用感知器算法

线性不可分问题

- 现实场景中大多数数据集线性不可分
- 如何使用分类器求解?



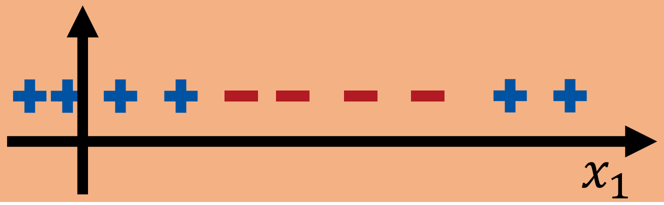
https://github.com/allisonhorst/palmerpenguins

机器学习任务

- **二分类任务：**

  学习映射：$\mathbb{R}^d \to \{-1, +1\}$

  - Ex：**线性分类**

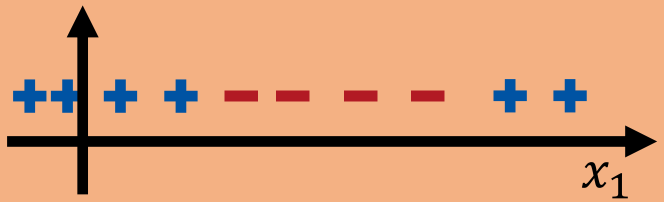


机器学习任务

- **二分类任务：**

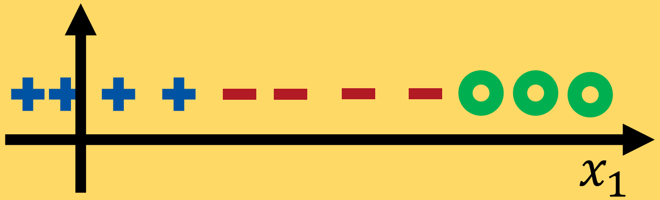
  学习映射：$\mathbb{R}^d \to \{-1,+1\}$

  - Ex：**线性分类**



- **多分类：**

  标签类别> 2

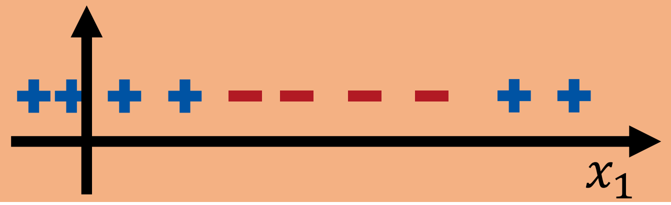


机器学习任务

- **二分类任务：**

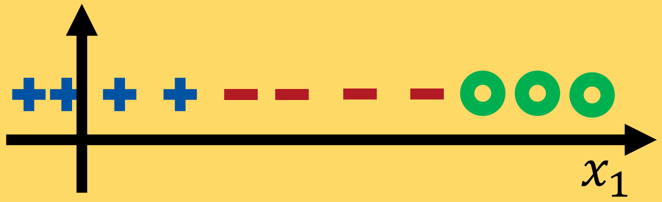
  学习映射：$\mathbb{R}^d \rightarrow \{-1,+1\}$

  - Ex：**线性分类**



- **多分类：**

  标签类别>2



- **分类：**

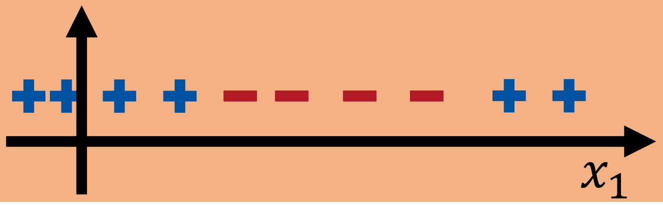
  学习一个到离散集合上的映射

机器学习任务

- **二分类任务：**

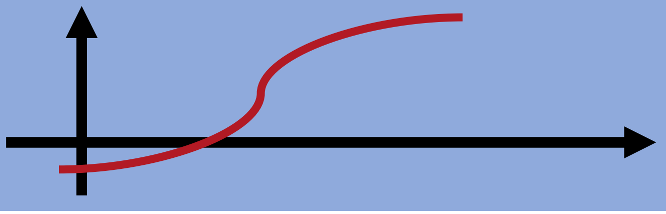
  学习映射：$\mathbb{R}^d \to \{-1,+1\}$

  - Ex：**线性分类**



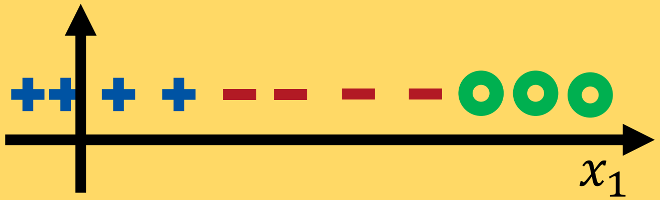
- **回归：**

  学习一个到连续值上的映射



- **多分类：**

  标签类别>2



- **分类：**

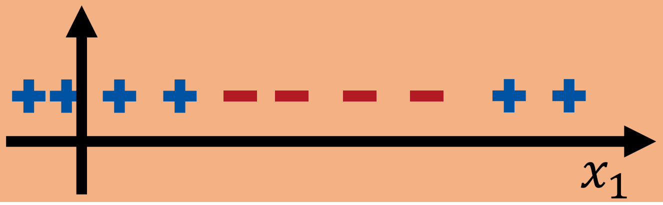
  学习一个到离散集合上的映射

机器学习任务

- **二分类任务：**

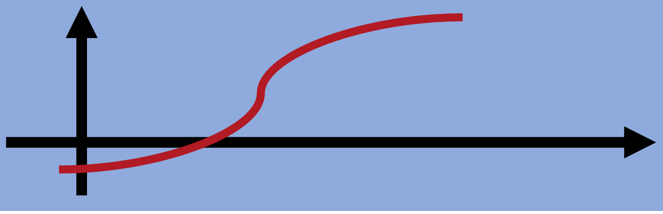
  学习映射：$\mathbb{R}^d \to \{-1,+1\}$

  - Ex：**线性分类**



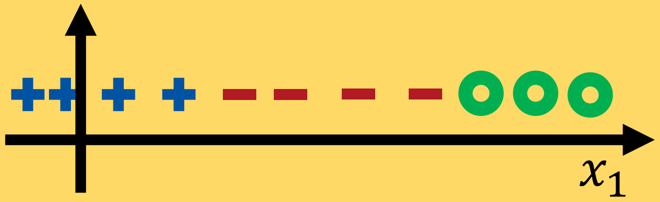
- **回归：**

  学习一个到连续值上的映射



- **多分类：**

  标签类别> 2



- **分类：**

  学习一个到离散集合上的映射

- **监督学习：**

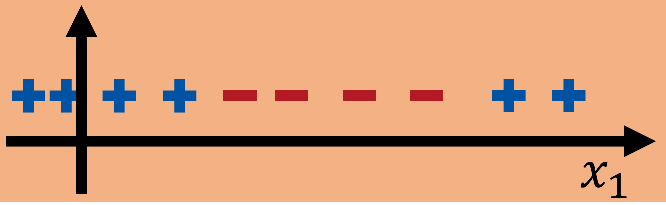
  学习从特征到标签的映射

机器学习任务

- **二分类任务：**

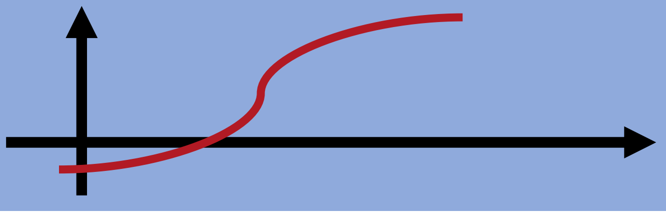
  学习映射：$\mathbb{R}^d \rightarrow \{-1,+1\}$

  - Ex：**线性分类**



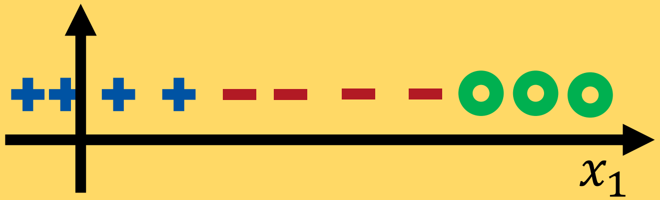
- **回归：**

  学习一个到连续值上的映射



- **多分类：**

  标签类别> 2



- **分类：**

  学习一个到离散集合上的映射

- **监督学习：**

  学习从特征到标签的映射

- **无监督学习：**

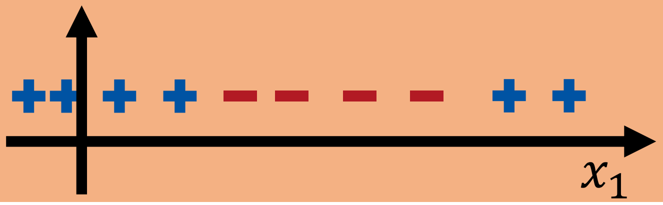
  没有标签，自主发掘模式

机器学习任务

- **二分类任务：**

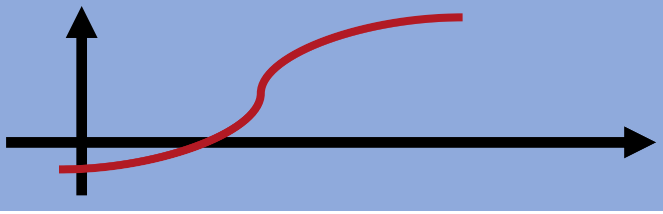
  学习映射：$\mathbb{R}^d \to \{-1,+1\}$

  - Ex：**线性分类**



- **回归：**

  学习一个到连续值上的映射

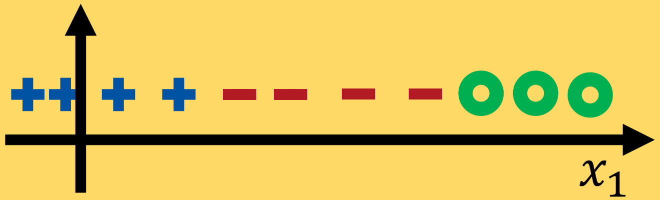


- **半监督学习：**

  少量带标签样本+大量无标签样本

- **多分类：**

  标签类别> 2



- **分类：**

  学习一个到离散集合上的映射

- **监督学习：**

  学习从特征到标签的映射

- **无监督学习：**

  没有标签，自主发掘模式