

H1 第一章 学习问题

第一章 学习问题

1.1 学习与机器学习

机器学习定义

1.2 机器学习组成要素及与其他领域的关系

学习问题形式化

学习模型

机器学习的实用定义

机器学习与数据挖掘、人工智能、统计学的关系

1.3 感知机假说集及感知机学习算法

一类简单的假说集合：“感知机”

感知机假说的向量形式

\mathbb{R}^2 空间中的感知机

从 \mathcal{H} 中选择 g

感知机学习算法

感知机算法的问题

1.4 感知机学习算法的理论保证

线性可分性

1.5 非可分数据

噪声数据学习

容忍噪声的线性分类器

Pocket 算法

总结

H2 1.1 学习与机器学习

H3 机器学习定义

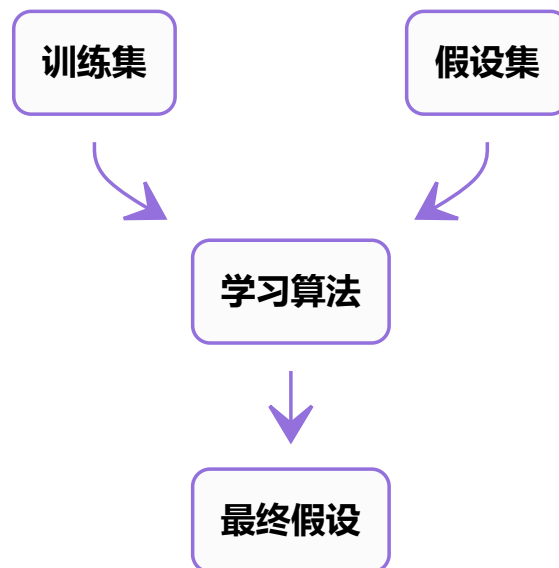
通过从数据中获取经验来提高性能

H2 1.2 机器学习组成要素及与其他领域的关系

H3 学习问题形式化

- 输入: $x \in \mathcal{X}$
- 输出: $y \in \mathcal{Y}$
- 需要学习的未知模式 \Leftrightarrow 目标函数: $f: \mathcal{X} \rightarrow \mathcal{Y}$
- 数据 \Leftrightarrow 训练集: $\mathcal{D} = \{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_N, y_N)\}$
- 假设 \Leftrightarrow 使计算机获得良好性能的能力: $g: \mathcal{X} \rightarrow \mathcal{Y}$

H3 学习模型



H3 机器学习的实用定义

使用数据来计算一个拟合目标 f 的假设 g

- 例题：

讨论时间

How to use the four sets below to form a learning problem for song recommendation?

$S_1 = [0, 100]$
 $S_2 =$ all possible (userid, songid) pairs
 $S_3 =$ all formula that 'multiplies' user factors & song factors, indexed by all possible combinations of such factors
 $S_4 =$ 1,000,000 pairs of ((userid, songid), rating)

① $S_1 = \mathcal{X}, S_2 = \mathcal{Y}, S_3 = \mathcal{H}, S_4 = \mathcal{D}$
 ② $S_1 = \mathcal{Y}, S_2 = \mathcal{X}, S_3 = \mathcal{H}, S_4 = \mathcal{D}$
 ③ $S_1 = \mathcal{D}, S_2 = \mathcal{H}, S_3 = \mathcal{Y}, S_4 = \mathcal{X}$
 ④ $S_1 = \mathcal{X}, S_2 = \mathcal{D}, S_3 = \mathcal{Y}, S_4 = \mathcal{H}$

刘新旺 (AiBD)
学习问题
2019 年 10 月 15 日 11 / 48

H3 机器学习与数据挖掘、人工智能、统计学的关系

| 机器学习 | 数据挖掘 | 人工智能 | 统计学 |
|---------------------------|-----------------|-------------|---------------|
| 使用数据来计算一个拟合目标 f 的假设 g | 使用大量数据来找到感兴趣的属性 | 使计算机能够表现出智能 | 使用数据对未知过程进行推断 |

- 如果感兴趣的属性与目标假设相同，则 机器学习=数据挖掘

- 如果感兴趣的属性与目标假设相关，则数据挖掘可以辅助机器学习
- 传统数据挖掘同时关注大数据上高性能的计算
- $g \simeq f$ — 机器学习可以实现AI (机器学习是实现AI的一个途径)
- g 是推断结果， f 是未知的 — 统计可以用来实现机器学习
- 传统统计同时关注具有数学假设的可证明结果，对计算不怎么关注

讨论时间

Which of the following claim is not totally true?

- ① machine learning is a route to realize artificial intelligence
- ② machine learning, data mining and statistics all need data
- ③ data mining is just another name for machine learning
- ④ statistics can be used for data mining

Reference Answer: ③

While data mining and machine learning do share a huge overlap, they are arguably not equivalent because of the difference of focus.

H2 1.3 感知机假说集及感知机学习算法

H3 一类简单的假说集合：“感知机”

- 对于 $\mathbf{x} = \{x_1, x_2, \dots, x_d\}$ 的特征，计算一个权重分数 $score$ ，且
 - 若 $\sum_{i=1}^d w_i x_i > threshold$ ，则判定为正例
 - 若 $\sum_{i=1}^d w_i x_i < threshold$ ，则判定为负例
- $\mathcal{Y} \{+1, -1\}$ ，线性式 $h \in \mathcal{H}$ 为：

$$h(\mathbf{x}) = \text{sign}((\sum_{i=1}^d w_i x_i) - threshold) \quad (1)$$

这被称为感知机假设

H3 感知机假说的向量形式

$$h(x) = \text{sign}\left(\sum_{i=1}^d w_i x_i - \text{threshold}\right) \quad (2)$$

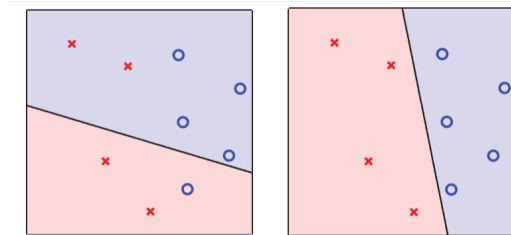
$$= \text{sign}\left(\sum_{i=1}^d w_i x_i + \underbrace{(-\text{threshold})}_{w_0} \cdot \underbrace{(+1)}_{x_0}\right) \quad (3)$$

$$= \text{sign}\left(\sum_{i=0}^d w_i x_i\right) \quad (4)$$

$$= \text{sign}(\mathbf{w}^T \mathbf{x}) \quad (5)$$

H3 \mathbb{R}^2 空间中的感知机

$$h(\mathbf{x}) = \text{sign}(w_0 + w_1 x_1 + w_2 x_2) \quad (6)$$



- 特征 \mathbf{x} ：空间中的点
- 标签 y ：+1, -1
- 假设 h ：平面中的直线（或 \mathbb{R}^d 中的超平面）

感知机 \Leftrightarrow 线性（二）分类

H3 从 \mathcal{H} 中选择 g

\mathcal{H} = 所有可能的感知机, $g = ?$

- 目标: $g \simeq f$ (f 未知是很难)
- 可行方案: 在 \mathcal{D} 上 $g \simeq f$, $g(\mathbf{x}_n) = f(\mathbf{x}_n) = y_n$
- 难点: \mathcal{H} 无穷多
- 想法: 从某个 g_0 开始, 不断修正它在 \mathcal{D} 上的结果

H3 感知机学习算法

从某个 \mathbf{w}_0 开始 (通常设为 $\mathbf{0}$), 不断修正它在 \mathcal{D} 上的错误

For $t = 0, 1, \dots$

- 1 find a mistake of \mathbf{w}_t called $(\mathbf{x}_{n(t)}, y_{n(t)})$

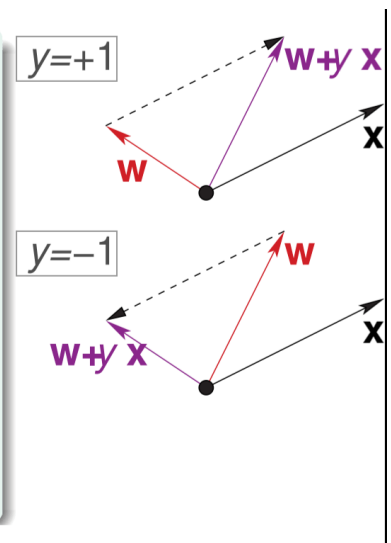
$$\text{sign}(\mathbf{w}_t^T \mathbf{x}_{n(t)}) \neq y_{n(t)}$$

- 2 (try to) correct the mistake by

$$\mathbf{w}_{t+1} \leftarrow \mathbf{w}_t + y_{n(t)} \mathbf{x}_{n(t)}$$

... until no more mistakes

return last \mathbf{w} (called \mathbf{w}_{PLA}) as g



代码示例:

```
# 更新参数
def update(self, label_i, data_i):
```

```

        tmp = label_i * data_i
        tmp = tmp.reshape(self.w.shape)
        # 更新w和b
        self.w = tmp + self.w
        self.b = self.b + label_i
# 感知机算法
def pla(self):
    isFind = False
    num = 0
    while not isFind:
        count = 0
        for i in range(self.num_samples):
            tmp_y = self.sign(self.w, self.b, self.x[i, :])
            if tmp_y * self.y[i] <= 0:
                count += 1
                num += 1
                self.update(self.y[i], self.x[i, :])
        if count == 0:
            isFind = True

```

H3 感知机算法的问题

要将 g 做到在 \mathcal{D} 上无分类错误

- 算法上, 要达到收敛
 - 初始循环次数?
 - 随机循环次数?
 - 其他?
- 学习: $g \simeq f$?
 - 在 \mathcal{D} 上, 能够收敛
 - 在 \mathcal{D} 之外呢?
 - 如果不能收敛呢?

Let's try to think about why PLA may work.

Let $n = n(t)$, according to the rule of PLA below, which formula is true?

$$\text{sign}(\mathbf{w}_t^T \mathbf{x}_n) \neq y_n, \quad \mathbf{w}_{t+1} \leftarrow \mathbf{w}_t + y_n \mathbf{x}_n$$

- ① $\mathbf{w}_{t+1}^T \mathbf{x}_n = y_n$
- ② $\text{sign}(\mathbf{w}_{t+1}^T \mathbf{x}_n) = y_n$
- ③ $y_n \mathbf{w}_{t+1}^T \mathbf{x}_n \geq y_n \mathbf{w}_t^T \mathbf{x}_n$
- ④ $y_n \mathbf{w}_{t+1}^T \mathbf{x}_n < y_n \mathbf{w}_t^T \mathbf{x}_n$

Reference Answer: ③

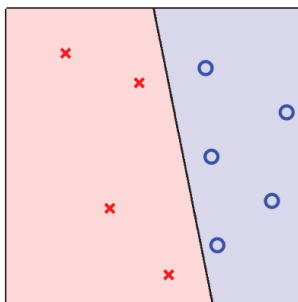
Simply multiply the second part of the rule by $y_n \mathbf{x}_n$. The result shows that the rule somewhat “tries to correct the mistake.”

H2 1.4 感知机器学习算法的理论保证

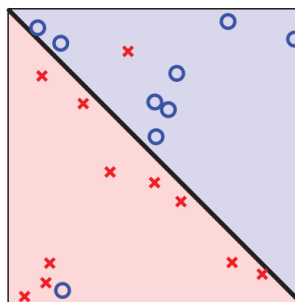
H3 线性可分性

- 如果PLA收敛，要能找到 \mathbf{w} 使得在 \mathcal{D} 无分类错误
- 满足的话，称 \mathcal{D} 线性可分

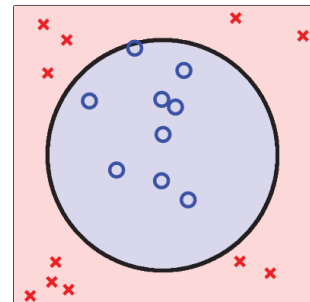
•



(linear separable)



(not linear separable)



(not linear separable)

- 假设 \mathcal{D} 线性可分，PLA一定收敛吗？

- 线性可分 $\Leftrightarrow \exists \mathbf{w}_f \text{ s.t. } y_n = \text{sign}(\mathbf{w}_f^T \mathbf{x}_n)$
- 在更新算法的过程中，每一步都比上一步更好

需要证明更新后的参数 \mathbf{w}_{t+1} 比当前的参数 \mathbf{w}_t 更加接近目标参数 \mathbf{w}_f ，即需要证明

$$\mathbf{w}_f^T \mathbf{w}_{t+1} > \mathbf{w}_f^T \mathbf{w}_t$$

证明

$$\because y_{n(t)} \mathbf{w}_f^T \mathbf{x}_{n(t)} \geq \min_n y_n \mathbf{w}_f^T \mathbf{x}_n > 0 \quad (7)$$

\therefore 由更新规则知: (8)

$$\mathbf{w}_t^T \mathbf{w}_{t+1}^T = \mathbf{w}_t^T (\mathbf{w}_t + y_{n(t)} \mathbf{x}_{n(t)}) \quad (9)$$

$$\geq \mathbf{w}_f^T \mathbf{w}_t + \min_n y_n \mathbf{w}_f^T \mathbf{x}_n \quad (10)$$

$$> \mathbf{w}_f^T \mathbf{w}_t + 0 \quad (11)$$

得证

- 以上证明内容并不能保证是 \mathbf{w}_t 与 \mathbf{w}_f 角度更一致, 故需要证明 \mathbf{w}_t 的模长并不会太大, 由于 \mathbf{w}_t 只在发生错误的时候按照更新规则 $\text{sign}(\mathbf{w}_t^T \mathbf{x}_{n(t)}) \neq y_{n(t)} \Leftrightarrow y_{n(t)} \mathbf{w}_t^T \mathbf{x}_{n(t)} \leq 0$ 改变, 所以错误的分类限制了 $\|\mathbf{w}_t\|^2$ 的增长, 需要证明每一步的增长都有上界

证明

$$\|\mathbf{w}_{t+1}\|^2 = \|\mathbf{w}_t + y_{n(t)} \mathbf{x}_{n(t)}\|^2 \quad (12)$$

$$= \|\mathbf{w}_t\|^2 + 2y_{n(t)} \mathbf{w}_t^T \mathbf{x}_{n(t)} + \|y_{n(t)} \mathbf{x}_{n(t)}\|^2 \quad (13)$$

$$\leq \|\mathbf{w}_t\|^2 + 0 + \|y_{n(t)} \mathbf{x}_{n(t)}\|^2 \quad (14)$$

$$\leq \|\mathbf{w}_t\|^2 + \|y_n \mathbf{x}_n\|^2 \quad (15)$$

则每一步的增长都有上界

- 上面两步可以从直觉上表明PLA可以收敛, 下面给出更严格的证明
- **Novikoff定理** 设训练数据集 $T = \{(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)\}$ 是线性可分的, 其中 $x_i \in \mathcal{X} = \mathbf{R}^n, y_i = \{-1, +1\}, i = 1, 2, \dots, N$, 记 $\hat{\mathbf{w}} = (\mathbf{w}^T, b)^T, \hat{\mathbf{x}} = (\mathbf{x}, 1)^T$, 则 $\hat{\mathbf{x}}, \hat{\mathbf{w}} \in \mathbf{R}^{n+1}$, 显然 $\hat{\mathbf{w}} \cdot \hat{\mathbf{x}} = \mathbf{w} \cdot \mathbf{x} + b$

- 存在满足条件 $\|\hat{\mathbf{w}}_{opt}\| = 1$ 的超平面 $\hat{\mathbf{w}}_{opt} \cdot \hat{\mathbf{x}} = \mathbf{w}_{opt} \cdot \mathbf{x} + b_{opt} = 0$ 将数据集完全正确分开, 且存在 $\gamma > 0$, 对所有 $i = 1, 2, \dots, N$

$$y_i(\hat{\mathbf{w}} \cdot \hat{\mathbf{x}}_i) = y_i(\mathbf{w}_{opt} \cdot \mathbf{x}_i + b_{opt}) \geq \gamma \quad (16)$$

- 令 $R = \max_{1 \leq i \leq N} \|\hat{\mathbf{x}}\|$, 则感知机算法在训练数据集上的误分类次数 k 满足不等式

$$k \leq \left(\frac{R}{\gamma}\right)^2 \quad (17)$$

证明

- 由于训练数据是线性可分的, 故存在超平面可将数据集完全正确分开, 取此超平面为 $\hat{\mathbf{w}}_{opt} \cdot \hat{\mathbf{x}} = \mathbf{w}_{opt} \cdot \mathbf{x} + b_{opt} = 0$, 使 $\|\hat{\mathbf{w}}_{opt}\| = 1$, 由于对有限的 $i = 1, 2, \dots, N$, 均有

$$y_i(\hat{\mathbf{w}}_{opt} \cdot \hat{\mathbf{x}}_i) = y_i(\mathbf{w}_{opt} \cdot \mathbf{x}_i + b_{opt}) > 0 \quad (18)$$

所以存在

$$\gamma = \min_i \{y_i(\mathbf{w}_{opt} \cdot \mathbf{x}_i + b_{opt})\} \quad (19)$$

使

$$y_i(\hat{\mathbf{w}} \cdot \hat{\mathbf{x}}_i) = y_i(\mathbf{w}_{opt} \cdot \mathbf{x}_i + b_{opt}) \geq \gamma \quad (20)$$

- 感知机算法从 $\hat{\mathbf{w}}_0 = 0$ 开始, 如果示例被误分类, 则更新权重。令 $\hat{\mathbf{w}}_{k-1}$ 是第 k 个误分类实例之前的扩充权重向量, 即

$$\hat{\mathbf{w}}_{k-1} = (\mathbf{w}_{k-1}^T, b_{k-1})^T \quad (21)$$

则第 k 个误分类实例的条件是

$$y_i(\hat{\mathbf{w}}_{k-1} \cdot \hat{\mathbf{x}}_i) = y_i(\mathbf{w}_{k-1} \cdot \mathbf{x}_i + b_{k-1}) \leq 0 \quad (22)$$

则此时进行更新

$$\mathbf{w}_k \leftarrow \mathbf{w}_{k-1} + \eta y_i \mathbf{x}_i \quad (23)$$

$$b_k \leftarrow b_{k-1} + \eta y_i \quad (24)$$

即

$$\hat{\mathbf{w}}_k = \hat{\mathbf{w}}_{k-1} + \eta y_i \hat{\mathbf{x}}_i \quad (25)$$

下面证明两个不等式

$$\bullet \quad \hat{\mathbf{w}}_k \cdot \hat{\mathbf{w}}_{opt} \geq k\eta\gamma \quad (26)$$

由25及20可知

$$\hat{\mathbf{w}}_k \cdot \hat{\mathbf{w}}_{opt} = \hat{\mathbf{w}}_{k-1} \cdot \hat{\mathbf{w}}_{opt} + \eta y_i \hat{\mathbf{w}}_{opt} \cdot \hat{\mathbf{x}}_i \quad (27)$$

$$\geq \hat{\mathbf{w}}_{k-1} \cdot \hat{\mathbf{w}}_{opt} + \eta\gamma \quad (28)$$

由此递推即可证得26

$$\bullet \quad \|\hat{\mathbf{w}}_k\|^2 \leq k\eta^2 R^2 \quad (29)$$

由式25及22可得

$$\|\hat{\mathbf{w}}_k\|^2 = \|\hat{\mathbf{w}}_{k-1}\|^2 + 2\eta y_i \hat{\mathbf{w}}_{k-1} \cdot \hat{\mathbf{x}}_i + \eta^2 \|\hat{\mathbf{x}}_i\|^2 \quad (30)$$

$$\leq \|\hat{\mathbf{w}}_{k-1}\|^2 + \eta^2 \|\hat{\mathbf{x}}_i\|^2 \quad (31)$$

$$\leq \|\hat{\mathbf{w}}_{k-1}\|^2 + \eta^2 R^2 \quad (32)$$

$$\leq \|\hat{\mathbf{w}}_{k-2}\|^2 + 2\eta^2 R^2 \quad (33)$$

$$\vdots \quad (34)$$

$$\leq k\eta^2 R^2 \quad (35)$$

结合不等式26及不等式29即可证得

$$k\eta\gamma \leq \hat{\mathbf{w}}_k \cdot \hat{\mathbf{w}}_{opt} \leq \|\hat{\mathbf{w}}_k\| \|\hat{\mathbf{w}}_{opt}\| \leq \sqrt{k}\eta R \quad (36)$$

$$\Rightarrow k^2 \gamma^2 \leq kR^2$$

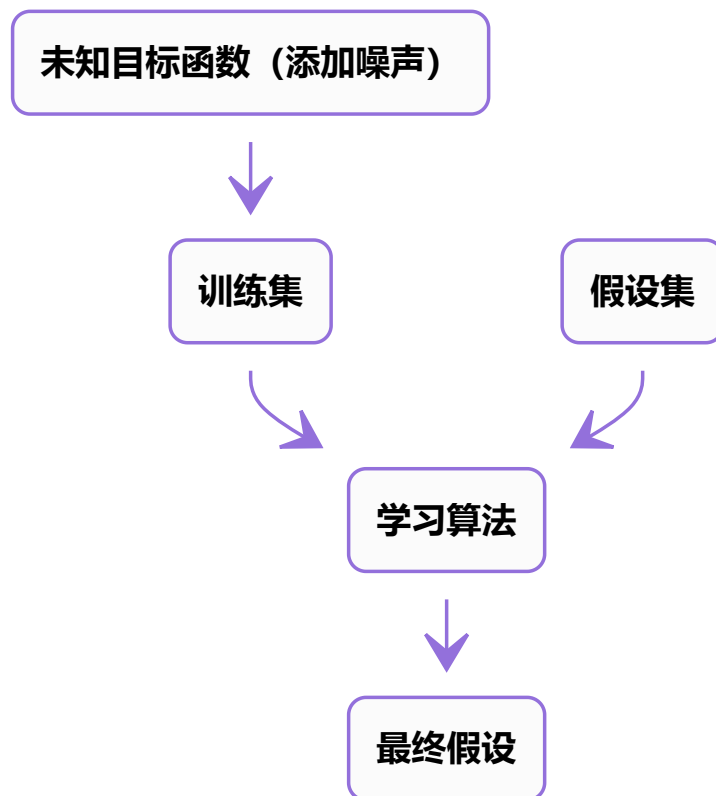
$$\Rightarrow k \leq \left(\frac{R}{\gamma}\right)^2$$

H3 PLA的其他信息

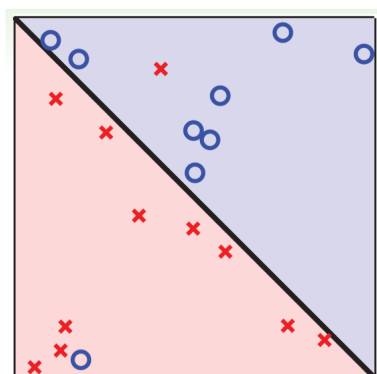
- $\langle \mathbf{w}_f, \mathbf{w}_t \rangle$ 增长迅速, \mathbf{w}_t 的长度增长缓慢
- PLA算法实现简单, 对于任何维度 d 的线性可分数据都有效
- 实际上 \mathcal{D} 是否线性可分提前不知道
- 不能完全确定需要多久才能收敛, γ 依赖于 \mathbf{w}_k (谱半径)

H2 1.5 非可分数据

H3 噪声数据学习



H3 容忍噪声的线性分类器



要满足 $y_n = f(x)$ 是不能直接线性分割

H3 Pocket算法

将PLA算法改进为将当前最佳权重存在pocket中

```

def update(self, label_i, data_i):
    tmp = label_i * data_i
    # 更新w和b
    tmp_w = tmp.reshape(self.w.shape) + self.w
    tmp_b = self.b + label_i
    if len(self.classify(tmp_w, tmp_b)) <= len(self.classify(self.w,
self.b)):
        self.best_w = tmp_w
        self.best_b = tmp_b
    self.w = tmp_w
    self.b = tmp_b

def classify(self, w, b):

```

```

mistakes = []
for i in range(self.num_samples):
    tmp_y = self.sign(w, b, self.x[i, :])
    if tmp_y * self.y[i] <= 0:
        mistakes.append(i)
return mistakes

def pocket(self):
    iters = 0
    isFind = False
    while not isFind:
        iters += 1
        mistakes = self.classify(self.w, self.b)
        if len(mistakes) == 0:
            break
        elif len(mistakes) > 1:
            i = mistakes[np.random.randint(0, len(mistakes)-1)]
        else:
            i = 0
        update = self.update(self.y[i], self.x[i, :])
        if iters == self.max_iters:
            isFind = True
    print("Pocket totally iter:", iters)
    return self.best_w, self.best_b

```

讨论时间

Should we use pocket or PLA?

Since we do not know whether \mathcal{D} is linear separable in advance, we may decide to just go with pocket instead of PLA. If \mathcal{D} is actually linear separable, what's the difference between the two?

- ① pocket on \mathcal{D} is slower than PLA
- ② pocket on \mathcal{D} is faster than PLA
- ③ pocket on \mathcal{D} returns a better g in approximating f than PLA
- ④ pocket on \mathcal{D} returns a worse g in approximating f than PLA

Reference Answer: ①

Because pocket needs to check whether \mathbf{w}_{t+1} is better than $\hat{\mathbf{w}}$ at each iteration, it is slower than PLA. On linear separable \mathcal{D} , $\mathbf{w}_{\text{POCKET}}$ is the same as \mathbf{w}_{PLA} , both making no mistakes.

H2 总结

- 感知机假设集：在 \mathbb{R}^d 中将数据分开的超平面
- PLA算法：在改正分类错误的过程中提高性能
- PLA算法停下来的条件：所有的数据都被正确分类
- 线性不可分数据：将最佳权重保存下来对比更新