第一章 学习问题

1.1 学习与机器学习

机器学习定义

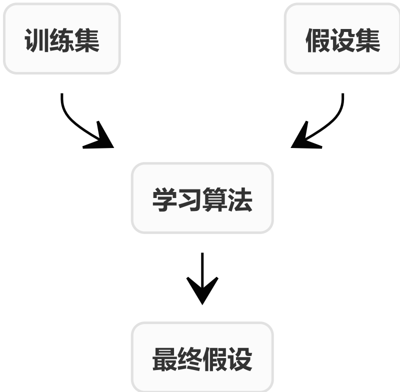
通过从数据中获取经验来提高性能

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

- 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： $f : \mathcal{X} \to \mathcal{Y}$
- 数据 $\Leftrightarrow$ 训练集： $\mathcal{D} = \{(\boldsymbol{x}_1, y_1), \cdots, (\boldsymbol{x}_N, y_N)\}$
- 假设 $\Leftrightarrow$ 使计算机获得良好性能的能力： $g : \mathcal{X} \to \mathcal{Y}$

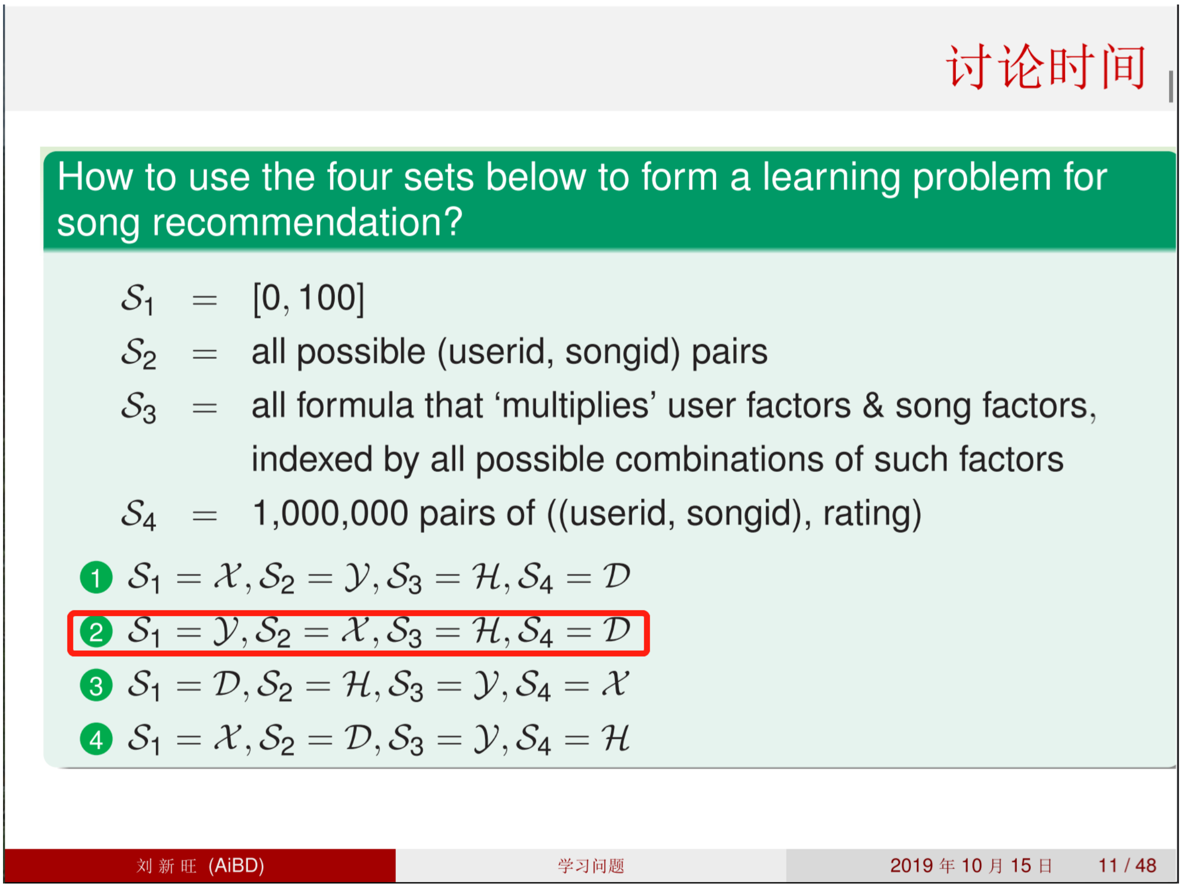
学习模型



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:

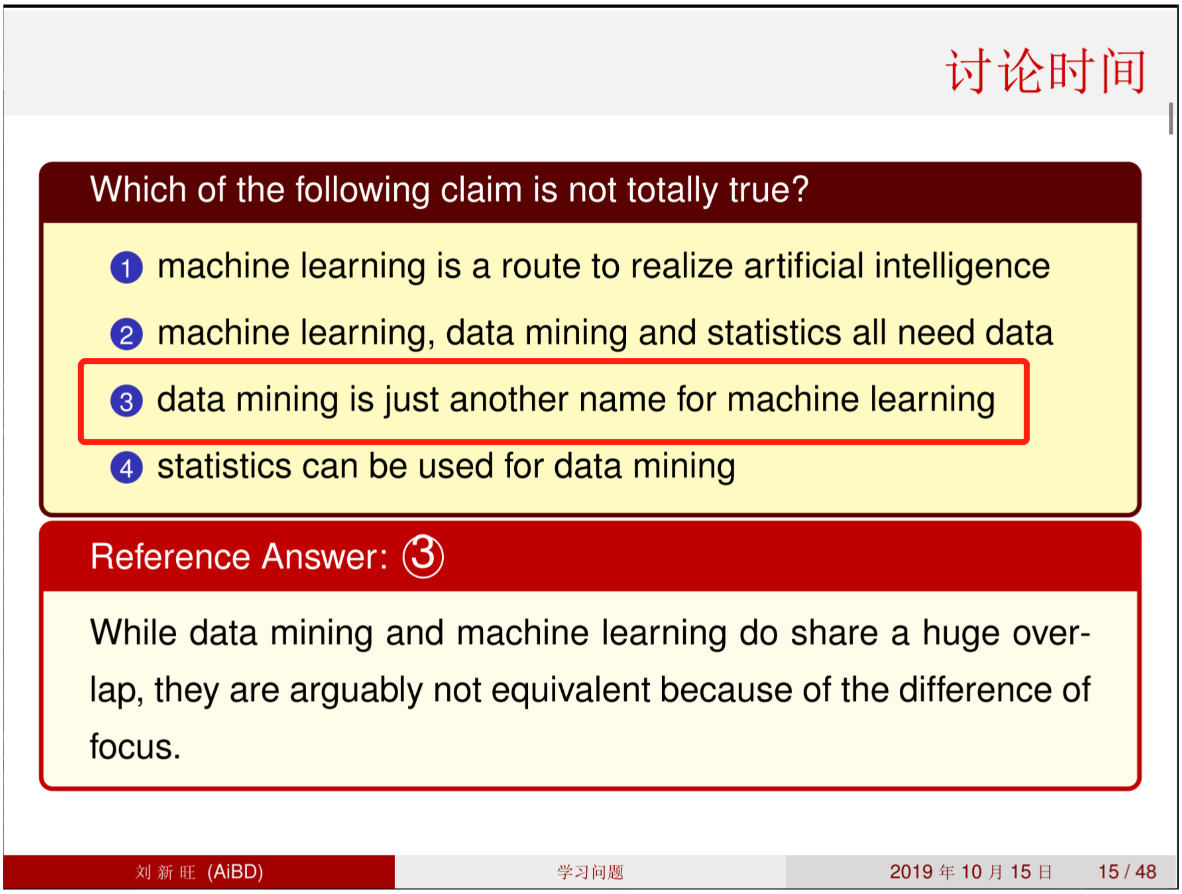


机器学习与数据挖掘、人工智能、统计学的关系

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



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”

- 对于$\boldsymbol{x} = \{x_1, x_2, \cdots, x_d\}$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：

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

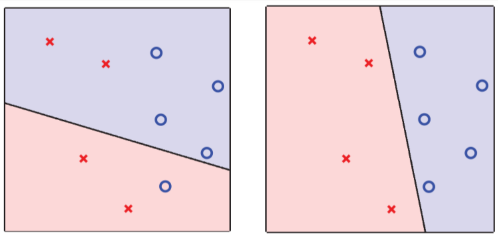
这被称为感知机假设

感知机假说的向量形式

$$
\begin{aligned}
h(x) &= sign((\sum_{i=1}^{d} w_i x_i) - threshold) & (2)\\
&= sign((\sum_{i=1}^{d} w_i x_i) + \underbrace{(-threshold)}_{w_0} \cdot \underbrace{(+1)}_{x_0}) & (3)\\
&= sign(\sum_{i=0}^{d} w_i x_i) & (4)\\
&= sign(\boldsymbol{w}^T \boldsymbol{x}) & (5)
\end{aligned}
$$

$\mathbb{R}^2$空间中的感知机

$$
h(\boldsymbol{x}) = sign(w_0 + w_1 x_1 + w_2 x_2) \qquad (6)
$$



- 特征 $\boldsymbol{x}$：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⇔线性（二）分类

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 D 上 $g \simeq f$， $g(\boldsymbol{x}_n) = f(\boldsymbol{x}_n) = y_n$
- 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

感知机学习算法

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

For $t = 0, 1, \ldots$

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

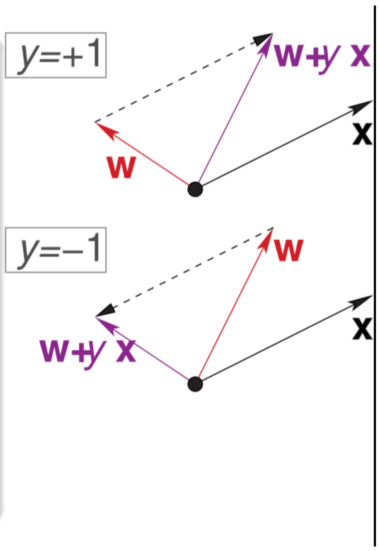
$$\text{sign}\left(\mathbf{w}_t^T \mathbf{x}_{n(t)}\right) \neq y_{n(t)}$$

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 $\mathbf{w}_{\text{PLA}}$) as g



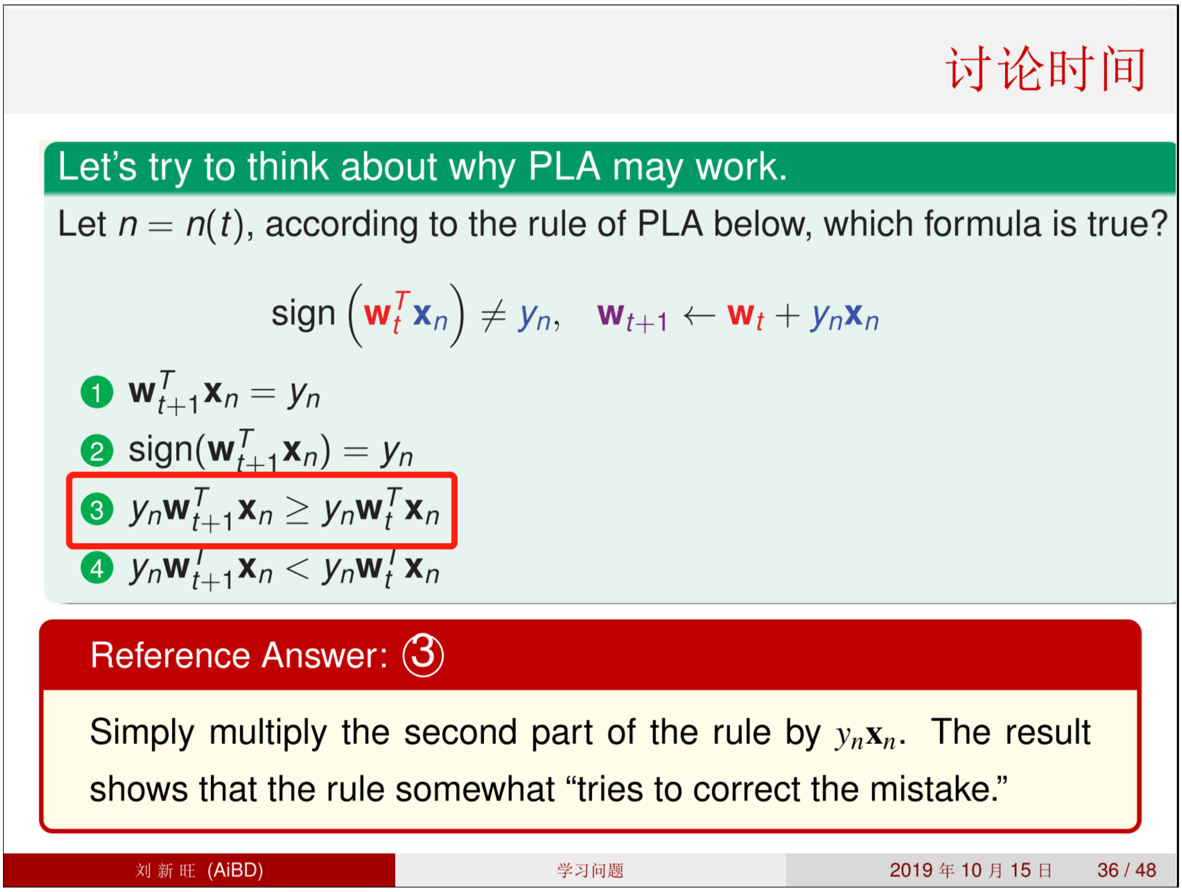
代码示例:

```
# 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
```

感知机算法的问题

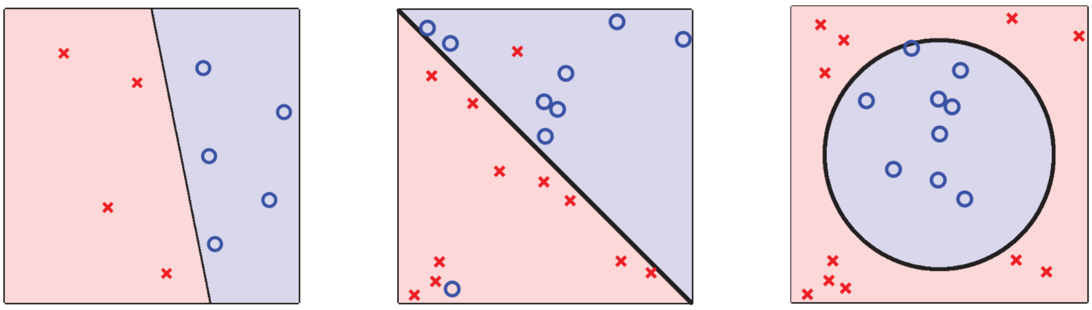
要将g做到在$\mathcal{D}$上无分类错误

- 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?



1.4 感知机学习算法的理论保证

线性可分性

- 如果PLA收敛，要能找到$\boldsymbol{w}$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$\Leftrightarrow \exists \boldsymbol{w}_f \quad s.t \quad y_n = sign(\boldsymbol{w}_f^T \boldsymbol{x}_n)$
  - 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$\boldsymbol{w}_{t+1}$比当前的参数$\boldsymbol{w}_t$更加接近目标参数$\boldsymbol{w}_f$，即需要证明$\boldsymbol{w}_f^T \boldsymbol{w}_{t+1} > \boldsymbol{w}_f^T \boldsymbol{w}_t$

    证明

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

$$\therefore \text{由更新规则知：} \tag{8}$$

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

$$\geq \boldsymbol{w}_f^T \boldsymbol{w}_t + \min_n y_n \boldsymbol{w}_f^T \boldsymbol{x}_n \tag{10}$$

$$> \boldsymbol{w}_f^T \boldsymbol{w}_t + 0 \tag{11}$$

得证

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

  证明

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

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

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

$$\leq \|\boldsymbol{w}_t\|^2 + \|y_n \boldsymbol{x}_n\|^2 \tag{15}$$

  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** [1] 设训练数据集$T = \{(x_1, y_1), (x_2, y_2), \cdots, (x_N, y_N),\}$是线性可分的，其中$x_i \in \mathcal{X} = \boldsymbol{R}^n, y_i = \{-1, +1\}, i = 1, 2, \cdots, N$，记$\hat{\boldsymbol{w}} = (\boldsymbol{w}^T, b)^T$，$\hat{\boldsymbol{x}} = (\boldsymbol{x}, 1)^T$，则$\hat{\boldsymbol{x}}, \hat{\boldsymbol{w}} \in \boldsymbol{R}^{n+1}$，显然$\hat{\boldsymbol{w}} \cdot \hat{\boldsymbol{x}} = \boldsymbol{w} \cdot \boldsymbol{x} + b$
  - 存在满足条件$\|\hat{\boldsymbol{w}}_{opt}\| = 1$的超平面$\hat{\boldsymbol{w}}_{opt} \cdot \hat{\boldsymbol{x}} = \boldsymbol{w}_{opt} \cdot \boldsymbol{x} + b_{opt} = 0$将数据集完全正确分开，且存在$\gamma > 0$，对所有$i = 1, 2, \cdots, N$

$$y_I (\hat{\boldsymbol{w}} \cdot \hat{x}_i) = y_i (\boldsymbol{w}_{opt} \cdot x_i + b_{opt}) \geq \gamma \tag{16}$$

  - 令$R = \max_{1 \leq i \leq N} \|\hat{\boldsymbol{x}}\|$，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

$$k \leq (\frac{R}{\gamma})^2 \tag{17}$$

    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$\hat{\boldsymbol{w}}_{opt} \cdot \hat{\boldsymbol{x}} = \boldsymbol{w}_{opt} \cdot \boldsymbol{x} + b_{opt} = 0$，使$\|\hat{\boldsymbol{w}}_{opt}\| = 1$，由于对有限的$i = 1, 2, \cdots, N$，均有

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

      所以存在

$$\gamma = \min_i \{y_i (w_{opt} \cdot x_i + b_{opt})\} \tag{19}$$

      使

$$y_I (\hat{\boldsymbol{w}} \cdot \hat{x}_i) = y_i (\boldsymbol{w}_{opt} \cdot x_i + b_{opt}) \geq \gamma \tag{20}$$

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

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

则第k个误分类实例的条件是

$$y_i(\hat{\boldsymbol{w}}_{k-1} \cdot \hat{\boldsymbol{x}}_i) = y_i(\boldsymbol{w}_{k-1} \cdot \boldsymbol{x}_i + b_{k-1}) \le 0 \tag{22}$$

则此时进行更新

$$\boldsymbol{w}_k \leftarrow \boldsymbol{w}_{k-1} + \eta y_i x_i \tag{23}$$
$$b_k \leftarrow b_{k-1} + \eta y_i \tag{24}$$

即

$$\hat{\boldsymbol{w}}_k = \hat{\boldsymbol{w}}_{k-1} + \eta y_i \hat{\boldsymbol{x}}_i \tag{25}$$

下面证明两个不等式

  - $$\hat{\boldsymbol{w}}_k \cdot \hat{\boldsymbol{w}}_{opt} \ge k\eta\gamma \tag{26}$$

    由25及20可知

    $$\begin{aligned} \hat{\boldsymbol{w}}_k \cdot \hat{\boldsymbol{w}}_{opt} &= \hat{\boldsymbol{w}}_{k-1} \cdot \hat{\boldsymbol{w}}_{opt} + \eta y_i \hat{\boldsymbol{w}}_{opt} \cdot \hat{\boldsymbol{x}}_i && (27)\\ &\ge \hat{\boldsymbol{w}}_{k-1} \cdot \hat{\boldsymbol{w}}_{opt} + \eta\gamma && (28) \end{aligned}$$

    由此递推即可证得26

  - $$\|\hat{\boldsymbol{w}}_k\|^2 \le k\eta^2 R^2 \tag{29}$$

    由式25及22可得

    $$\begin{aligned} \|\hat{\boldsymbol{w}}_k\|^2 &= \|\hat{\boldsymbol{w}}_{k-1}\|^2 + 2\eta y_i \hat{\boldsymbol{w}}_{k-1} \cdot \hat{x}_i + \eta^2 \|\hat{\boldsymbol{x}}_i\|^2 && (30)\\ &\le \|\hat{\boldsymbol{w}}_{k-1}\|^2 + \eta^2 \|\hat{\boldsymbol{x}}_i\|^2 && (31)\\ &\le \|\hat{\boldsymbol{w}}_{k-1}\|^2 + \eta^2 R^2 && (32)\\ &\le \|\hat{\boldsymbol{w}}_{k-2}\|^2 + 2\eta^2 R^2 && (33)\\ &\vdots && (34)\\ &\le k\eta^2 R^2 && (35) \end{aligned}$$

    结合不等式26及不等式29即可证得

    $$\begin{aligned} k\eta\gamma \le \hat{\boldsymbol{w}}_k \cdot \hat{\boldsymbol{w}}_{opt} &\le \|\hat{\boldsymbol{w}}_k\|\|\hat{\boldsymbol{w}}_{opt}\| \le \sqrt{k}\eta R && (36)\\ \Rightarrow k^2\gamma^2 &\le kR^2 \\ \Rightarrow k &\le (\frac{R}{\gamma})^2 \end{aligned}$$

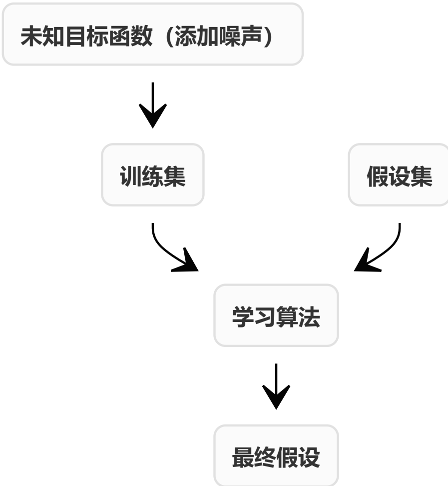
PLA的其他信息

  - $\langle \boldsymbol{w}_f, \boldsymbol{w}_t \rangle$ 增长迅速， $\boldsymbol{w}_t$ 的长度增长缓慢

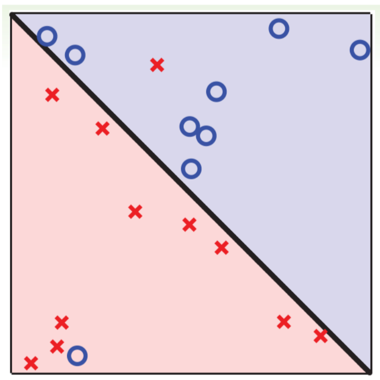
- PLA算法实现简单，对于任何维度d的线性可分数据都有效
  - 实际上 $\mathcal{D}$ 是否线性可分提前不知道
  - 不能完全确定需要多久才能收敛， γ 依赖于 $\boldsymbol{w}_k$ （谱半径）

1.5 非可分数据

噪声数据学习



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

Pocket算法

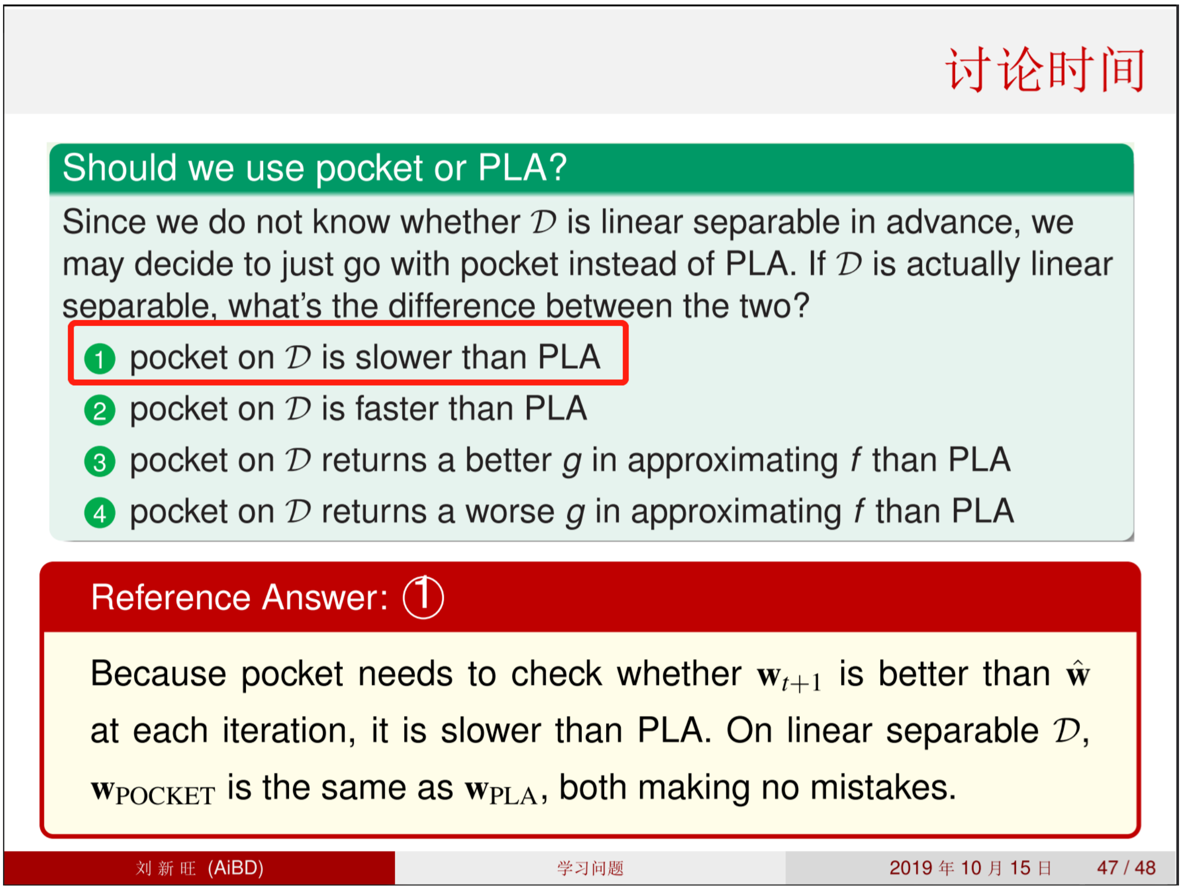
将PLA算法改进为将当前最佳权重存在pocket中

```
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
```



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

参考文献

1. 李航. 统计学习方法 第2版[M]. 清华大学出版社, 2019. ↩