

机器学习

Machine Learning

北京航空航天大学计算机学院

School of Computer Science and Engineering, Beihang University

刘庆杰 陈佳鑫

2025年春季学期

Spring 2025

第14章：集成学习

Chapter 14: Ensemble Learning

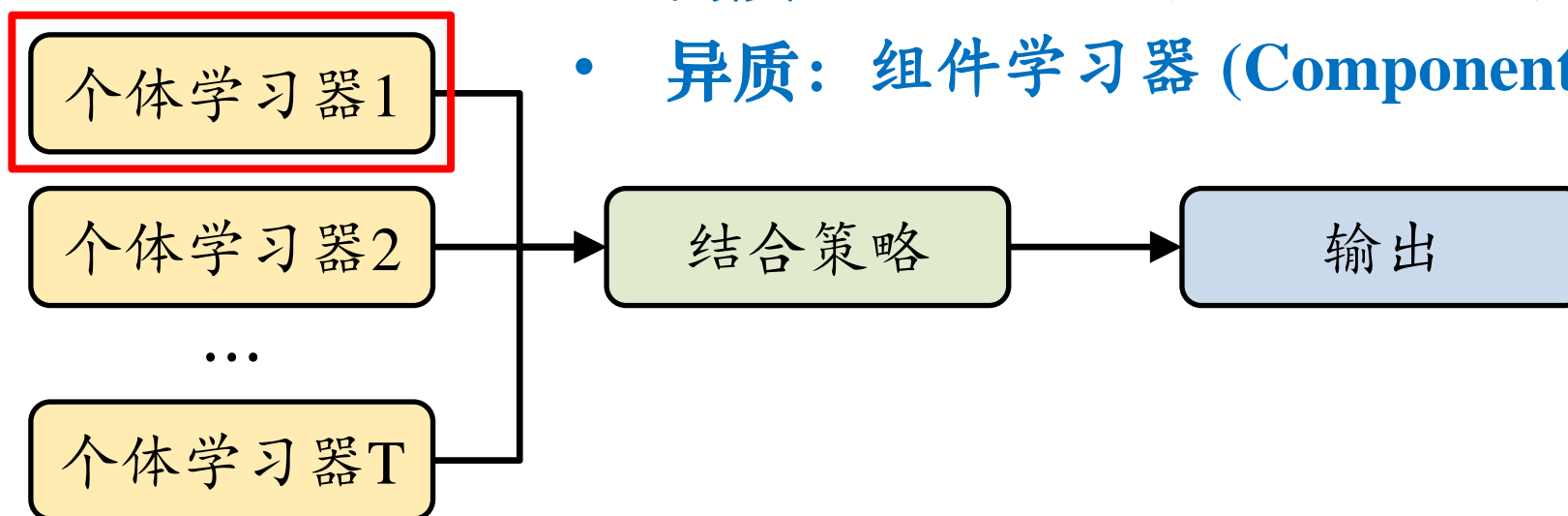
14.1 什么是集成学习?

- 集成学习的基本概念
- 个体与集成

集成学习的基本概念

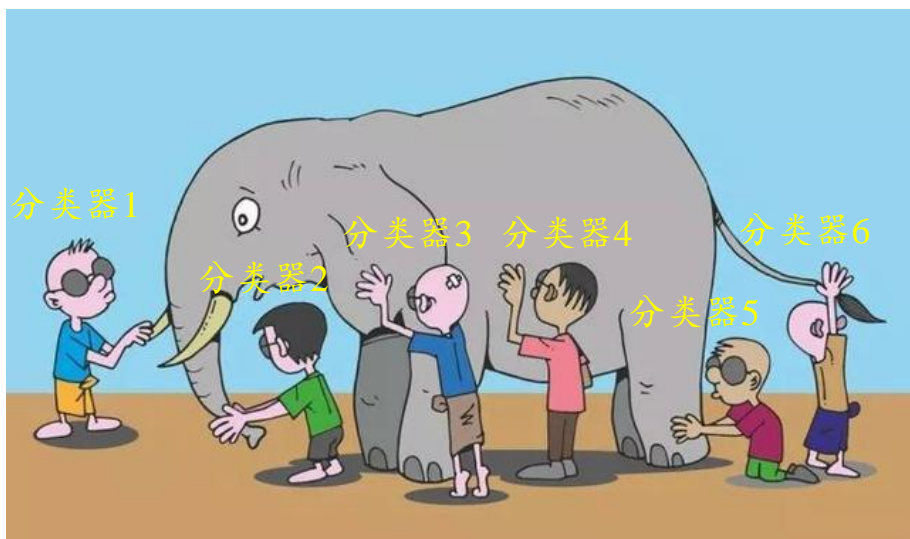
● 集成学习 (Ensemble Learning)

- 通过构建并结合多个分类器完成学习任务
- 也称为多分类器系统 (Multi-Classifier System)、基于委员会的学习 (Committee based Learning) 等



集成学习的基本概念

- 通过将多个学习器进行集成，常可获得比单一学习器显著优越的泛化性能，**这对弱学习器尤为明显**
 - **弱学习器**：准确率仅比随机猜测略高的学习器
 - **强学习器**：准确率高并能在多项式时间内完成的学习器



弱学习器 $\xrightarrow{\text{集成}}$ 强学习器

强学习器 $\xrightarrow{\text{集成}}$ 更强的学习器

集成学习的基本概念

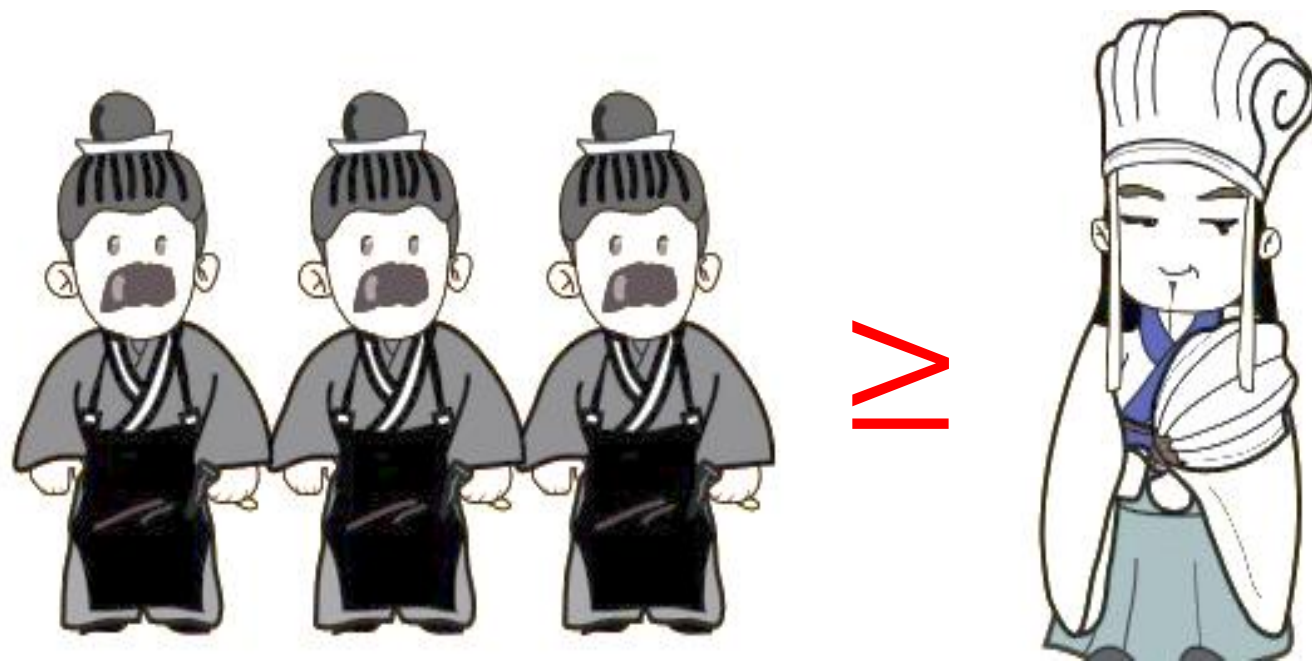
● 我国古代“集成学习”的智慧

- ✓ 《周易》：二人同心，其利断金
- ✓ 孔子：君子周而不比
- ✓ 孔子：三人行必有我师焉
- ✓ 孟子：天时不如地利，地利不如人和
- ✓ 荀子：民齐者强
- ✓ 文子：积力之所举，则无不胜也；众智之所为，则无不成也
- ✓ 孙武：上下同欲者胜



集成学习的基本概念

- 我国古代“集成学习”的智慧



三个臭皮匠，顶个(赛过)诸葛亮

个体与集成

● 多个学习器一定比单一学习器性能好吗？



简单示例：在二分类问题中，假定3个分类器在三个样本中的表现如下所示，其中√表示分类正确，×号表示分类错误，集成结果通过**投票 (Voting)**产生

	样本1	样本2	样本3
分类器1	√	×	√
分类器2	√	×	√
分类器3	√	×	√
集成	√	×	√

(a) 集成不起作用

	样本1	样本2	样本3
分类器1	×	√	×
分类器2	×	×	√
分类器3	√	×	×
集成	×	×	×

(b) 集成起负作用

	样本1	样本2	样本3
分类器1	√	√	×
分类器2	√	×	√
分类器3	×	√	√
集成	√	√	√

(c) 集成提升性能

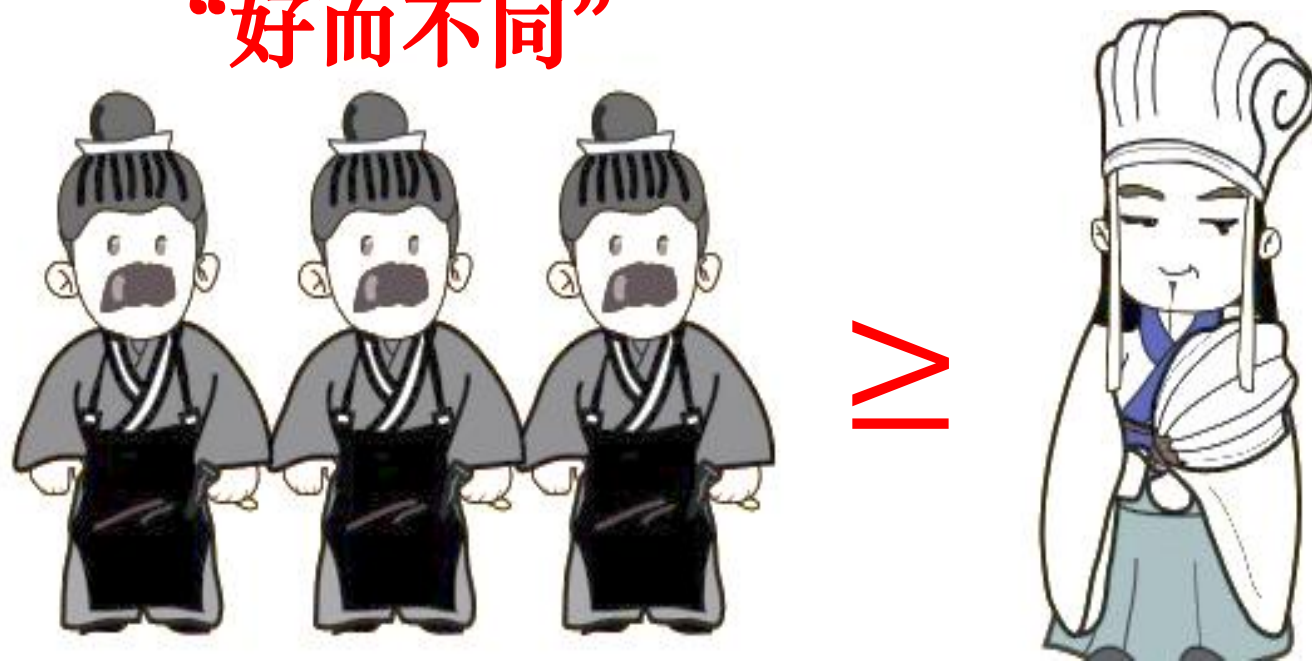
如何获得比最好的单一学习器更好的性能？

集成个体：好而不同

集成学习的基本概念

- 我国古代“集成学习”的智慧

“好而不同”



三个臭皮匠(裨将), 顶个(赛过)诸葛亮

“裨将”在古代指“副将”，原意是指三个副将的智慧能顶一个诸葛亮

个体与集成

● 简单分析

- 考虑二分类问题 $y \in \{-1, +1\}$ 和真实函数 f , 假定基分类器的错误率为 ϵ , 即对每个基分类器 h_i 有: $P(h_i(x) \neq f(x)) = \epsilon$
- 假设集成通过简单投票法结合 T 个基分类器 $\{h_i\}_{i=1}^T$, 若有超过半数的基分类器正确则分类就正确:

$$H(x) = \text{sign} \left(\sum_{i=1}^T h_i(x) \right)$$

个体与集成

● 简单分析

➤ 考虑二分类问题 $y \in \{-1, +1\}$ 和真实函数 f , 假定基分类器的错误率为 ϵ , 即对每个基分类器 h_i 有: $P(h_i(x) \neq f(x)) = \epsilon$

➤ 假设基分类器的错误率相互独立, 则由Hoeffding不等式可知:

至多 $\lfloor T/2 \rfloor$ 个基分类器分类正确

集成的错误率: $P(H(x) \neq f(x)) = \sum_{k=0}^{\lfloor T/2 \rfloor} \binom{T}{k} (1 - \epsilon)^k \epsilon^{T-k}$

恰好有 k 个基分类器
分类正确的概率

随着集成分类器数目的增加, 集成的
错误率将指数级下降, 最终趋向于0

$$\leq \exp\left(-\frac{1}{2}T(1 - 2\epsilon)^2\right)$$

个体与集成

● 简单分析

- **关键假设**：基学习器的误差相互独立
- 现实任务中，个体学习器是为解决同一个问题训练出来的，显然不可能互相独立！
- 个体学习器的“**准确性**”和“**多样性**”存在冲突

集成学习的研究核心：

如何产生并结合“**好而不同**”的个体学习器



集成学习方法

- 根据个体学习器生成方式不同，形成两大类方法

- 串行化方法：个体学习器间存在强依赖关系

- 典型算法：Boosting、Adaboost

- 并行化方法：个体学习器间不存在强依赖关系

- 典型算法：Bagging、随机森林 (Random Forest)

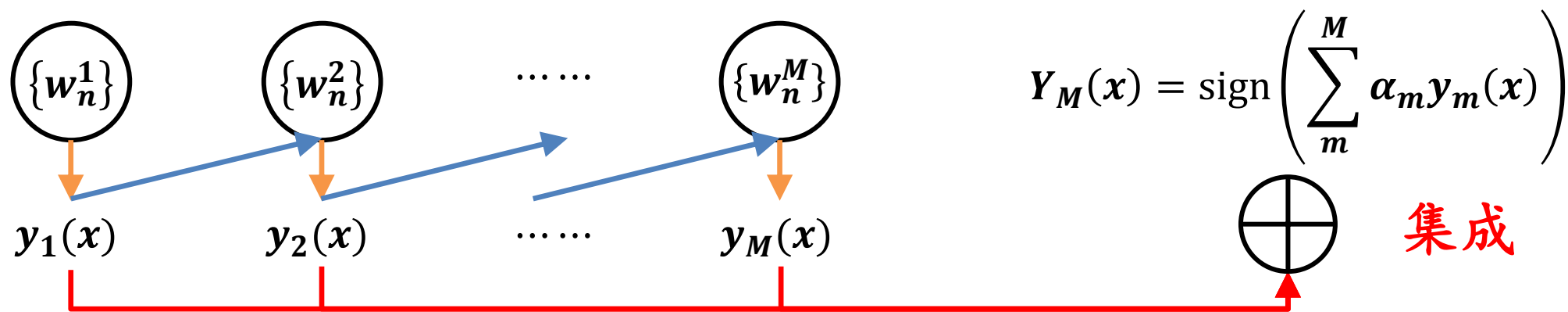
14.2 串行化集成学习算法

- Boosting算法
- AdaBoost算法

Boosting算法

- 串行式集成学习代表性方法：一族可将弱学习器提升为强学习器的算法

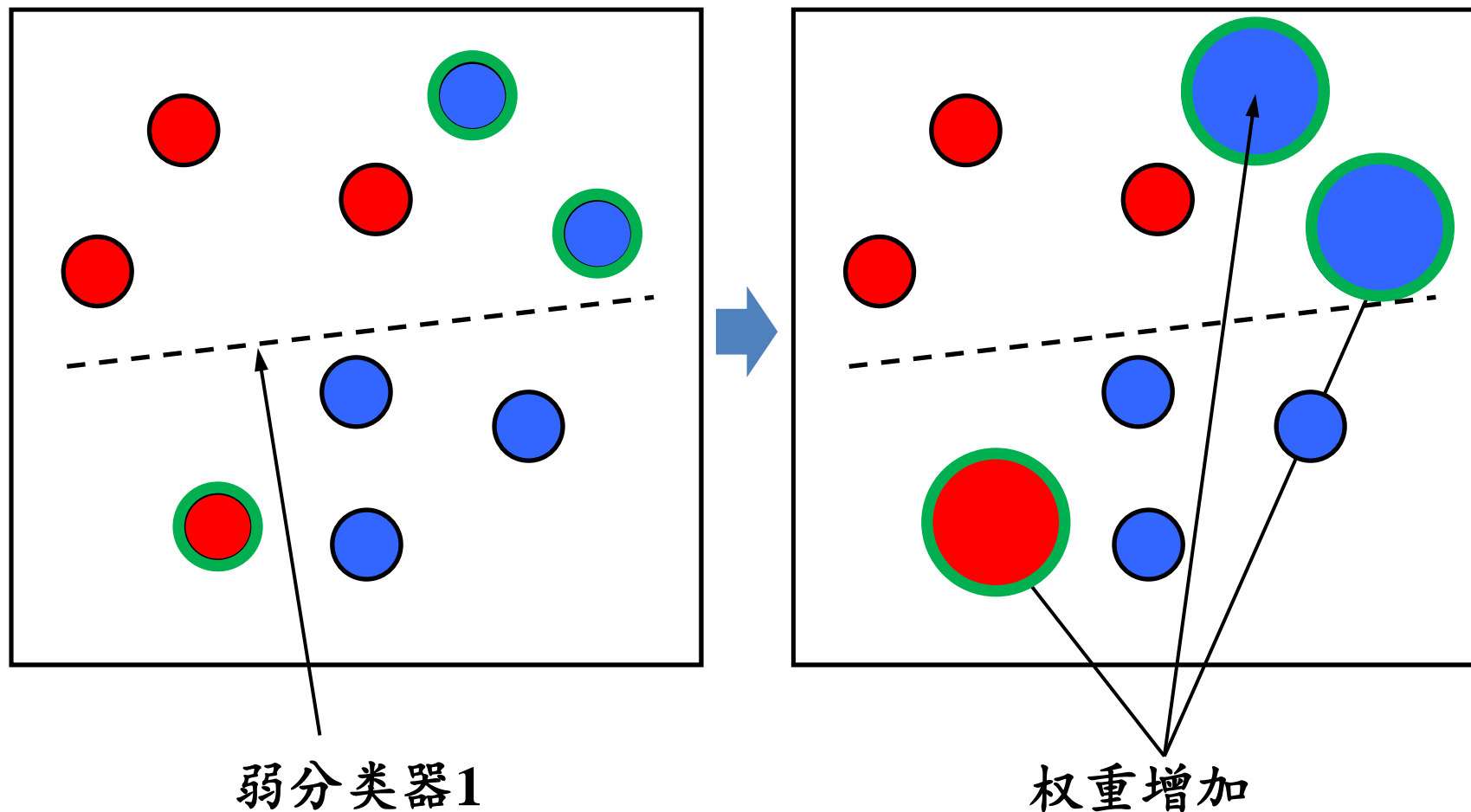
基本思想：迭代训练多个弱学习器，每次着重改进前一次的错误，从而将多个弱学习器组合成一个强学习器



- 先从初始数据集训练一个基学习器；
- 再根据其**对训练样本分布进行调整**，使**先前错分样本在后续受到更多关注**；
- 然后基于调整后的样本分布训练下一个基学习器；
- 重复进行直至基学习器数目达到预先指定值，最终将这些基学习器**加权结合**。

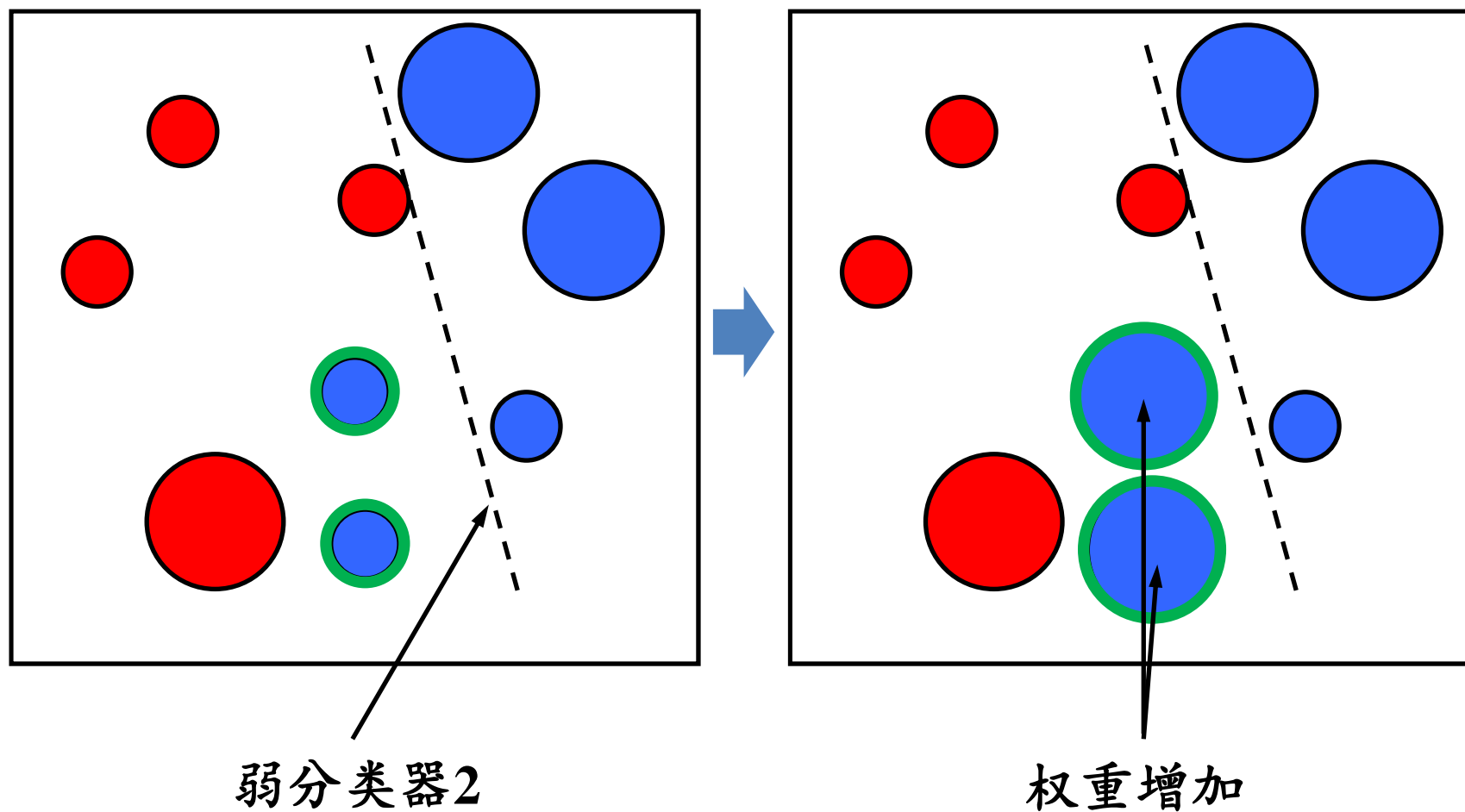
Boosting算法

● Boosting算法基本思想示意



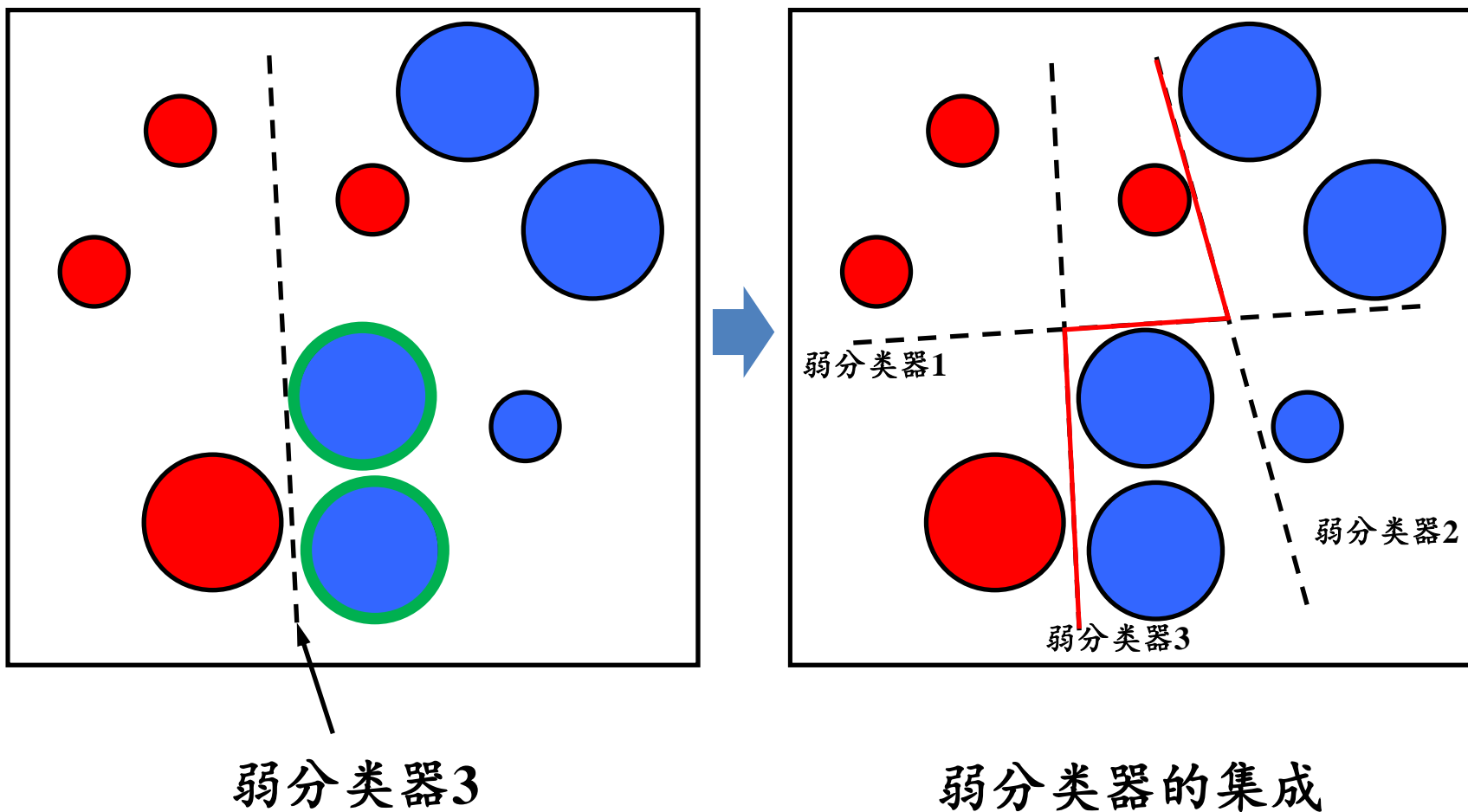
Boosting算法

● Boosting算法基本思想示意



Boosting算法

● Boosting算法基本思想示意



AdaBoost算法

● Boosting族算法最著名的代表

输入：训练集 $D = \{(x_1, y_1), (x_2, y_2), \dots, (x_m, y_m)\}$;
基学习算法 \mathcal{Q} ; 训练轮数 T .

$$1: \mathcal{D}_1(x) = \frac{1}{m}$$

2: **for** $t = 1, 2, \dots, T$ **do**:

$$3: \quad h_t = \mathcal{Q}(D, \mathcal{D}_t);$$

$$4: \quad \epsilon_t = P_{x \sim \mathcal{D}_t}(h_t(x) \neq f(x));$$

5: **if** $\epsilon_t > 0.5$ **then break**

$$6: \quad \alpha_t = \frac{1}{2} \ln \left(\frac{1 - \epsilon_t}{\epsilon_t} \right);$$

$$7: \quad \mathcal{D}_{t+1}(x) = \frac{\mathcal{D}_t(x)}{Z_t} \times \begin{cases} \exp(-\alpha_t), & \text{if } h_t(x) = f(x) \\ \exp(\alpha_t), & \text{if } h_t(x) \neq f(x) \end{cases} \\ = \frac{\mathcal{D}_t(x) \exp(-\alpha_t f(x) h_t(x))}{Z_t}$$

输出： $H(x) = \text{sign}(\sum_{t=1}^T \alpha_t h_t(x))$

【1997年Freund和Schapire提出】

f : 真实函数; $y \in \{-1, +1\}$

初始化样本权值分布

基于分布 \mathcal{D}_t 训练分类器 h_t

估计 h_t 误差

确定分类器 h_t 的权重

更新样本分布 (Z_t 规范化因子)

集成分类器 $h_{t=1,2,\dots,T}$

AdaBoost算法

● AdaBoost算法推导 – 问题定义

- 基于“加性模型” (Additive Model), 即基学习器线性组合:

$$H(x) = \sum_{t=1}^T \alpha_t h_t(x)$$

- 最小化指数损失函数(Exponential Loss Function) : [\[Friedman et al. 2000\]](#)

$$\ell_{\text{exp}}(H \mid \mathcal{D}) = \mathbb{E}_{x \sim \mathcal{D}}[e^{-f(x)H(x)}]$$

- 若 $H(x)$ 能令指数损失函数最小化, 考虑上式对 $H(x)$ 的偏导:

$$\frac{\partial \ell_{\text{exp}}(H \mid \mathcal{D})}{\partial H(x)} = -e^{-H(x)} P(f(x) = 1 \mid x) + e^{H(x)} P(f(x) = -1 \mid x)$$

AdaBoost算法

● AdaBoost算法推导 – 问题定义

➤ 令其为零求得: $H(x) = \frac{1}{2} \ln \frac{P(f(x) = 1 | x)}{P(f(x) = -1 | x)}$

➤ 因此有: $\text{sign}(H(x)) = \text{sign} \left(\frac{1}{2} \ln \frac{P(f(x) = 1 | x)}{P(f(x) = -1 | x)} \right)$

$$= \begin{cases} 1, & P(f(x) = 1 | x) > P(f(x) = -1 | x) \\ -1, & P(f(x) = 1 | x) < P(f(x) = -1 | x) \end{cases}$$

$$= \arg \max_{y \in \{-1, 1\}} P(f(x) = y | x)$$

$\text{sign}(H(x))$ 达到了贝叶斯最优错误率

说明指数损失函数是分类任务原来0/1损失函数的一致的替代损失函数

AdaBoost算法

$$\alpha_t = \frac{1}{2} \ln \left(\frac{1 - \epsilon_t}{\epsilon_t} \right)$$

● AdaBoost算法推导 - α_t 求解

- 第一个分类器 h_1 是直接将基学习算法用于初始数据分布得到的，此后迭代生成 h_t 和 α_t ，当基分类器 h_t 基于分布 \mathcal{D}_t 产生后，其权重 α_t 应使得 $\alpha_t h_t$ 最小化指数损失函数：

$$\begin{aligned} \ell_{\text{exp}}(\alpha_t h_t \mid \mathcal{D}_t) &= \mathbb{E}_{x \sim \mathcal{D}_t} [e^{-f(x) \alpha_t h_t(x)}] \\ &= \mathbb{E}_{x \sim \mathcal{D}_t} [e^{-\alpha_t} \mathbb{I}(f(x) = h_t(x)) + e^{\alpha_t} \mathbb{I}(f(x) \neq h_t(x))] \\ &= e^{-\alpha_t} P_{x \sim \mathcal{D}_t}(f(x) = h_t(x)) + e^{\alpha_t} P_{x \sim \mathcal{D}_t}(f(x) \neq h_t(x)) \\ &= e^{-\alpha_t} (1 - \epsilon_t) + e^{\alpha_t} \epsilon_t \end{aligned}$$

- 其中 $\epsilon_t = P_{x \sim \mathcal{D}_t}(f(x) \neq h_t(x))$ ，令指数损失函数的导数为0：

$$\frac{\partial \ell_{\text{exp}}(\alpha_t h_t \mid \mathcal{D}_t)}{\partial \alpha_t} = -e^{-\alpha_t} (1 - \epsilon_t) + e^{\alpha_t} \epsilon_t \quad \longrightarrow \quad \alpha_t = \frac{1}{2} \ln \left(\frac{1 - \epsilon_t}{\epsilon_t} \right)$$

AdaBoost算法

$$\mathcal{D}_{t+1}(x) = \frac{\mathcal{D}_t(x) \exp(-\alpha_t f(x) h_t(x))}{Z_t}$$

● AdaBoost算法推导 – $\mathcal{D}_{t+1}(x)$ 求解

- 获得 H_{t-1} 之后对样本分布进行调整，使下一轮基学习器 h_t 能纠正 H_{t-1} 的一些错误，理想的 h_t 能纠正全部错误，即最小化：

$$\begin{aligned}\ell_{\exp}(H_{t-1} + h_t | \mathcal{D}) &= \mathbb{E}_{x \sim \mathcal{D}}[e^{-f(x)(H_{t-1}(x) + h_t(x))}] \\ &= \mathbb{E}_{x \sim \mathcal{D}}[e^{-f(x)H_{t-1}(x)} e^{-f(x)h_t(x)}]\end{aligned}$$

- 由于 $f^2(x) = h_t^2(x) = 1$ ，上式用 $e^{-f(x)h_t(x)}$ 的泰勒展开式近似：

$$\begin{aligned}\ell_{\exp}(H_{t-1} + h_t | \mathcal{D}) &\simeq \mathbb{E}_{x \sim \mathcal{D}} \left[e^{-f(x)H_{t-1}(x)} \left(1 - f(x)h_t(x) + \frac{f^2(x)h_t^2(x)}{2} \right) \right] \\ &= \mathbb{E}_{x \sim \mathcal{D}} \left[e^{-f(x)H_{t-1}(x)} \left(1 - f(x)h_t(x) + \frac{1}{2} \right) \right]\end{aligned}$$

AdaBoost算法

$$\mathcal{D}_{t+1}(x) = \frac{\mathcal{D}_t(x) \exp(-\alpha_t f(x) h_t(x))}{Z_t}$$

● AdaBoost算法推导 - $\mathcal{D}_{t+1}(x)$ 求解

➤ 理想的基学习器:

$$\begin{aligned} h_t(x) &= \arg \min_h \ell_{\exp}(H_{t-1} + h \mid \mathcal{D}) \\ &= \arg \min_h \mathbb{E}_{x \sim \mathcal{D}} \left[e^{-f(x)H_{t-1}(x)} \left(1 - f(x)h(x) + \frac{1}{2} \right) \right] \\ &= \arg \max_h \mathbb{E}_{x \sim \mathcal{D}} [e^{-f(x)H_{t-1}(x)} f(x)h(x)] \\ &= \arg \max_h \mathbb{E}_{x \sim \mathcal{D}} \left[\frac{e^{-f(x)H_{t-1}(x)}}{\mathbb{E}_{x \sim \mathcal{D}}[e^{-f(x)H_{t-1}(x)}]} f(x)h(x) \right] \end{aligned}$$

➤ 注意到 $\mathbb{E}_{x \sim \mathcal{D}}[e^{-f(x)H_{t-1}(x)}]$ 是一个常数

➤ 令 \mathcal{D}_t 表示一个分布:

$$\mathcal{D}_t = \frac{\mathcal{D}(x) e^{-f(x)H_{t-1}(x)}}{\mathbb{E}_{x \sim \mathcal{D}}[e^{-f(x)H_{t-1}(x)}]}$$

AdaBoost算法

$$\mathcal{D}_{t+1}(x) = \frac{\mathcal{D}_t(x) \exp(-\alpha_t f(x) h_t(x))}{Z_t}$$

● AdaBoost算法推导 – $\mathcal{D}_{t+1}(x)$ 求解

➤ 根据数学期望的定义，这等价于令：

$$\begin{aligned} h_t(x) &= \arg \max_h \mathbb{E}_{x \sim \mathcal{D}} \left[\frac{e^{-f(x)H_{t-1}(x)}}{\mathbb{E}_{x \sim \mathcal{D}}[e^{-f(x)H_{t-1}(x)}]} f(x)h(x) \right] \\ &= \arg \max_h \mathbb{E}_{x \sim \mathcal{D}_t} [f(x)h(x)] \end{aligned}$$

➤ 由 $f(x), h(x) \in \{-1, +1\}$ ，有： $f(x)h(x) = 1 - 2\mathbb{I}(f(x) \neq h(x))$

➤ 则理想的基学习器： $h_t(x) = \arg \min_h \mathbb{E}_{x \sim \mathcal{D}_t} [\mathbb{I}(f(x) \neq h(x))]$


理想的 h_t 将在分布 \mathcal{D}_t 下最小化分类误差

AdaBoost算法

$$\mathcal{D}_{t+1}(x) = \frac{\mathcal{D}_t(x) \exp(-\alpha_t f(x) h_t(x))}{Z_t}$$

● AdaBoost算法推导 – $\mathcal{D}_{t+1}(x)$ 求解

- 弱分类器基于分布 \mathcal{D}_t 训练，且针对 \mathcal{D}_t 的分类误差应小于 0.5。这在一定程度上类似“残差逼近”的思想。考虑到 \mathcal{D}_t 和 \mathcal{D}_{t-1} 的关系，有：

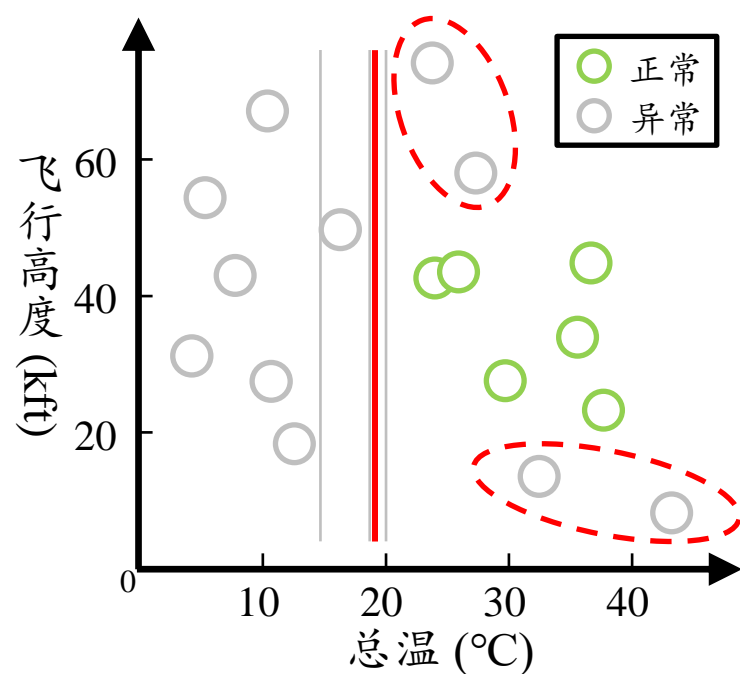
$$\begin{aligned}\mathcal{D}_{t+1}(x) &= \frac{\mathcal{D}(x) e^{-f(x) H_t(x)}}{\mathbb{E}_{x \sim \mathcal{D}}[e^{-f(x) H_t(x)}]} \\ &= \frac{\mathcal{D}(x) e^{-f(x) H_{t-1}(x)} e^{-f(x) \alpha_t h_t(x)}}{\mathbb{E}_{x \sim \mathcal{D}}[e^{-f(x) H_t(x)}]} \\ &= \mathcal{D}_t(x) \cdot e^{-f(x) \alpha_t h_t(x)} \frac{\mathbb{E}_{x \sim \mathcal{D}}[e^{-f(x) H_{t-1}(x)}]}{\mathbb{E}_{x \sim \mathcal{D}}[e^{-f(x) H_t(x)}]}\end{aligned}$$

$$\mathcal{D}_t = \frac{\mathcal{D}(x) e^{-f(x) H_{t-1}(x)}}{\mathbb{E}_{x \sim \mathcal{D}}[e^{-f(x) H_{t-1}(x)}]}$$

$$\mathcal{D}_{t+1}(x) = \frac{\mathcal{D}_t(x)}{Z_t} \times \begin{cases} \exp(-\alpha_t), & \text{if } h_t(x) = f(x) \\ \exp(\alpha_t), & \text{if } h_t(x) \neq f(x) \end{cases} = \frac{\mathcal{D}_t(x) \exp(-\alpha_t f(x) h_t(x))}{Z_t}$$

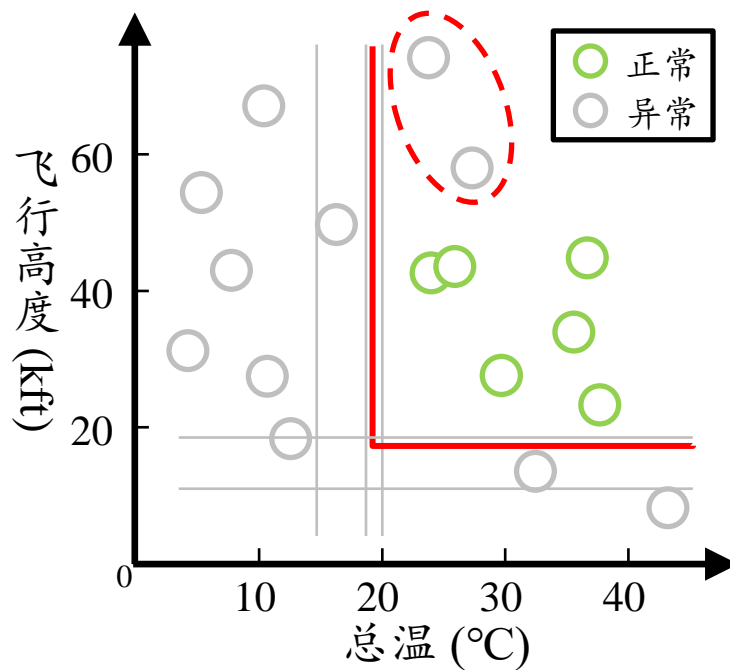
Boosting算法应用

● 航空发动机监控系统

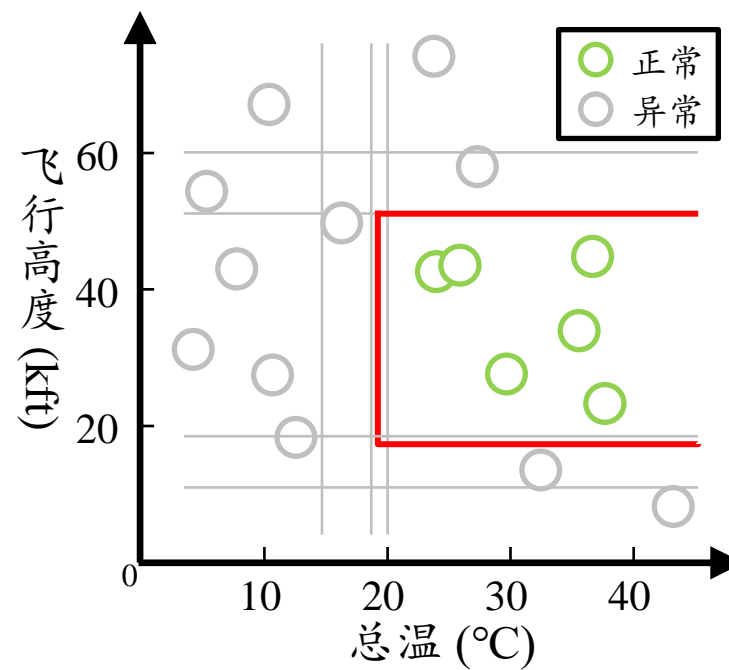
➤ 以决策树为基学习器，AdaBoost算法为集成学习算法



(a) 3个基学习器



(b) 5个基学习器



(c) 7个基学习器

Boosting算法

- 基学习器学习特定的数据分布

- 重赋权 (Re-weighting)

在每轮根据样本分布为每个训练样本重新赋予权重

- 重采样 (Re-sampling)

在每轮根据样本分布对训练集重新采样形成新的训练集

注意：Boosting每轮检查当前生成的基学习器是否满足优于随机猜测的基本条件，若不满足，此基学习器被抛弃，学习过程停止。如果与预先设定的学习轮数差距较大，会导致整体性能不佳。采用重采样策略，则可“重新启动”避免训练过早停止。

Boosting算法

● 特点总结

- Boosting中每个模型是弱模型，偏差高，方差低
- 个体学习器存在强依赖关系，串行生成，每次调整训练数据的样本分布
- Boosting的基本思想是用贪心法最小化损失函数，主要关注降低偏差
- 但是由于模型的相关性很强，因此不能显著降低方差

14.3 并行化集成学习算法

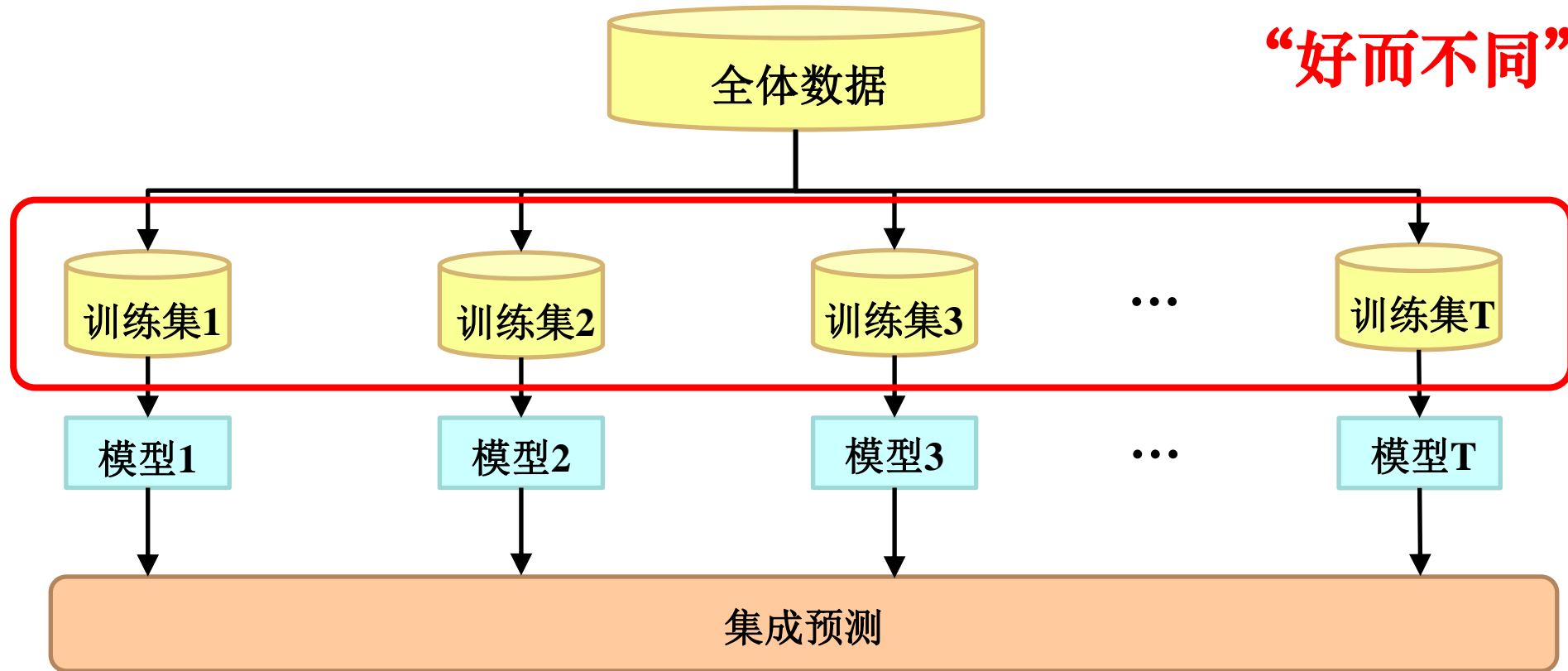
- Bagging算法
- 随机森林算法

Bagging算法

- 并行式集成学习最著名的代表性方法

【1993年Efron和Tibshirani提出】

➤ 名字由**B**ootstrap **AGG**regat**ING**缩写而来



Bagging算法

- 并行式集成学习最著名的代表性方法

- 基于自助采样法 (Bootstrap Sampling)

增加学习器的多样性：为不同学习器构建不同训练数据集

自助采样法流程

给定包含 m 个样本的数据集 D ，对其进行采样产生数据集 D' ：

每次随机从 D 中挑选一个样本，将其拷贝至 D' ，这个过程重复执行 m 次后，就得到了包含 m 个样本的数据集 D' 。显然， D 中有一部分样本会在 D' 中多次出现，而有一部分样本则不会出现。

样本在 m 次采样中始终不被采到的概率是 $\left(1 - \frac{1}{m}\right)^m$ ，其极限： $\lim_{m \rightarrow \infty} \left(1 - \frac{1}{m}\right)^m \mapsto \frac{1}{e} \approx 0.368$

将 D' 用做训练集； $D \setminus D'$ 用作测试集

Bagging算法

● 并行式集成学习最著名的代表性方法

基本思想：

利用自助法采样可构造 T 个含 m 个训练样本的采样集，基于每个采样集训练出一个基学习器，再将它们进行结合

在对预测输出结合时：

- 分类任务使用简单投票法
- 回归任务使用简单平均法

输入：

训练集 $D = \{(x_1, y_1), \dots, (x_m, y_m)\}$

基学习器 \mathcal{L} 学习器数量 T

过程：

1: **for** $t = 1, 2, \dots, T$ **do**:

2: $h_t = \mathcal{L}(D, D_{bs})$

3: **end for**

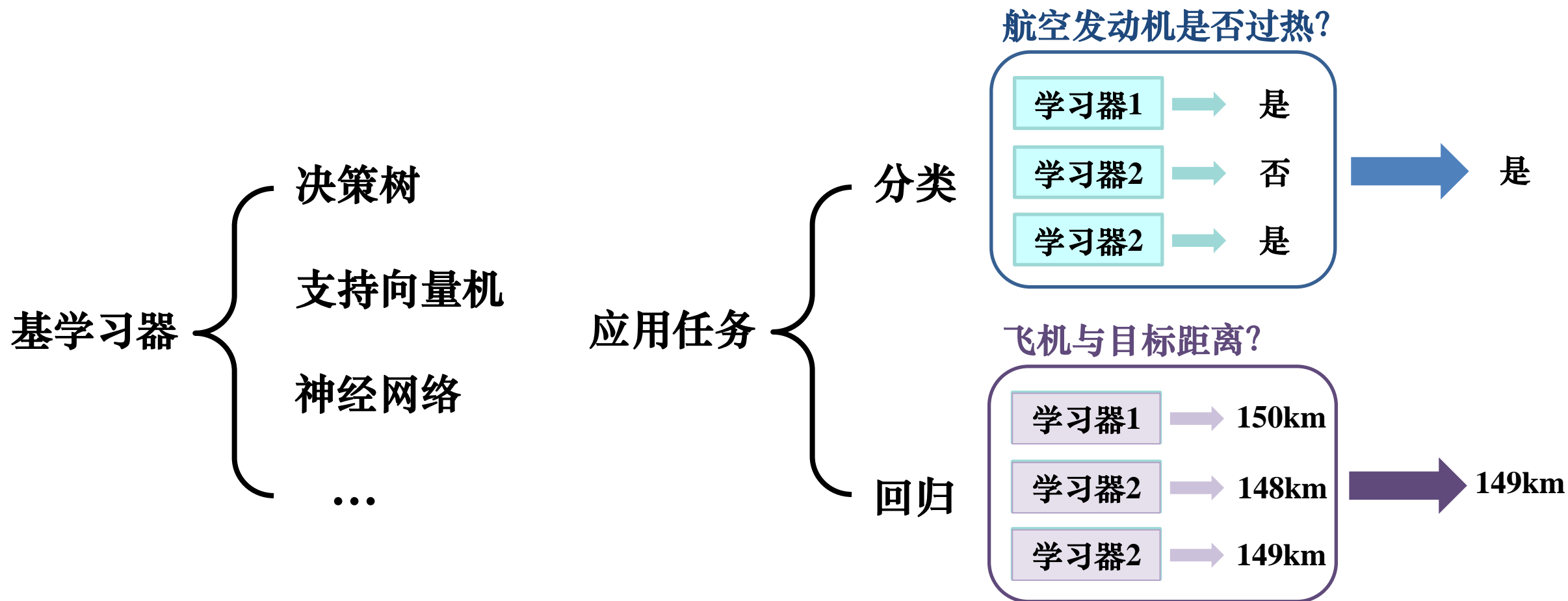
输出：

$$H(x) = \arg \max_{y \in \mathcal{Y}} \sum_{t=1}^T \mathbb{I}(h_t(x) = y)$$

Bagging算法

● 算法应用

➤ Bagging算法可应用于任何机器学习模型，具体场景取决于基学习器的选择



Bagging算法

● 特点总结

- **方差低**：集成多个独立训练的基学习器可以有效**降低方差**，在易受样本扰动的学习器上效用更为明显(如不剪枝的决策树、神经网络等)
- **易于并行**：由于各个基学习器互相独立，训练与推理都可并行化处理
- **无法降低偏差**：利用**样本扰动**构建的基学习器的偏差与方差近似相同，但互相间相关性不高，因此一般不能降低偏差，只能在一定程度上能降低方差
- **资源需求大**：训练与推理阶段都需运行多个基学习器，对计算资源要求较高

随机森林算法

- Bagging方法的一种扩展变体

【2001年Breiman提出】

- Bagging仅进行数据随机选择
- 训练过程引入随机属性选择
- 随机森林（Random Forest）以决策树为基学习器



决策树

● 决策树 (Decision Tree)

➤ 决策树是一种**树型结构**，由结点和有向边组成

➤ 结点

■ **内部结点**表示一个属性或特征

■ **叶结点**代表一种**类别**

➤ 有向边/分支

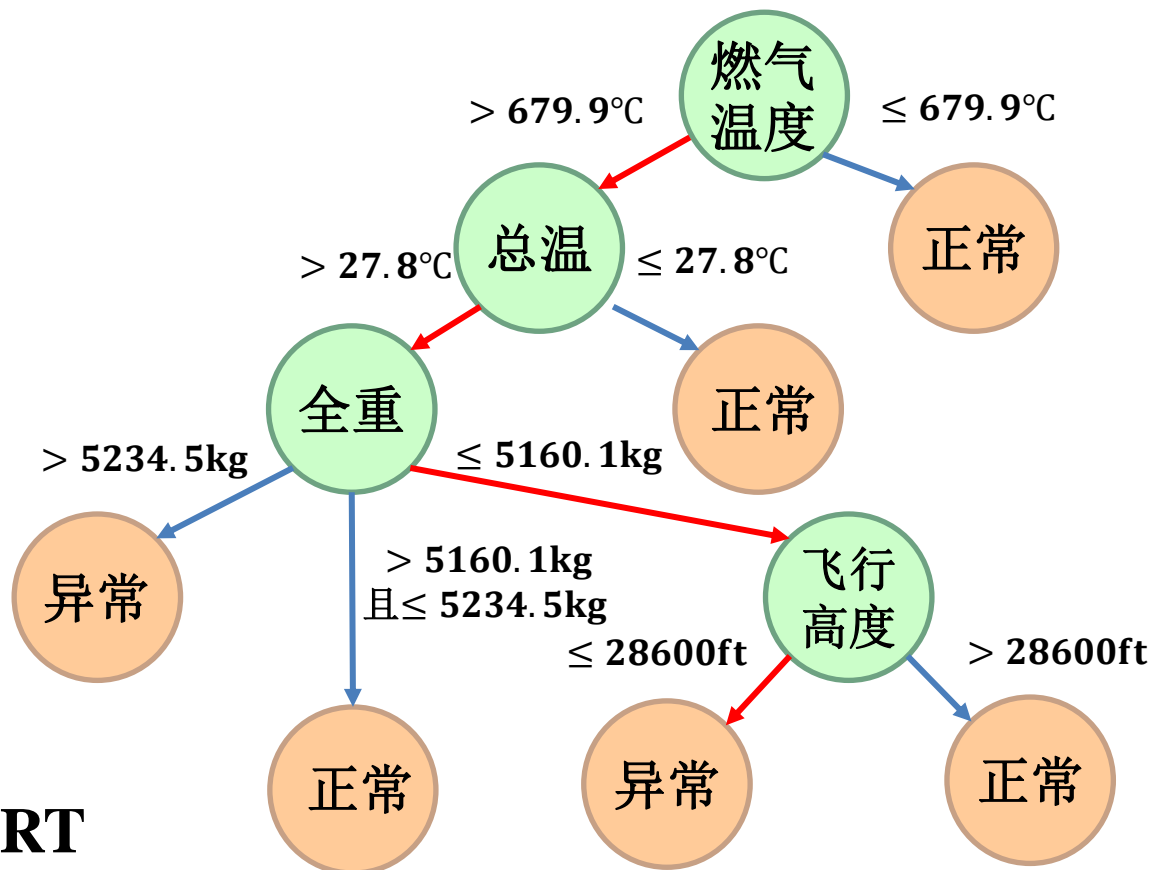
■ **分支**代表一个测试**输出**

● 决策树构建

➤ 核心：属性选取 → 节点划分

➤ 不同的属性选取策略：ID3、C4.5、CART

航空发动机过热分析



随机森林算法

● Bagging方法的一种扩展变体

【2001年Breiman提出】

- Bagging仅进行数据随机选择
- 训练过程引入随机属性选择
- 随机森林 (Random Forest) 以决策树为基学习器



● 基本思想：

- 结构组成：若干**决策树**组成，通过Bagging算法和随机属性选择策略实现
- 属性选择：对每个决策树结点，先从该结点的(d 个)属性集合中随机选择包含 k 个属性的子集，再选择一个最优属性用于划分
- 属性子集大小：一般情况下推荐 $k=\log_2 d$

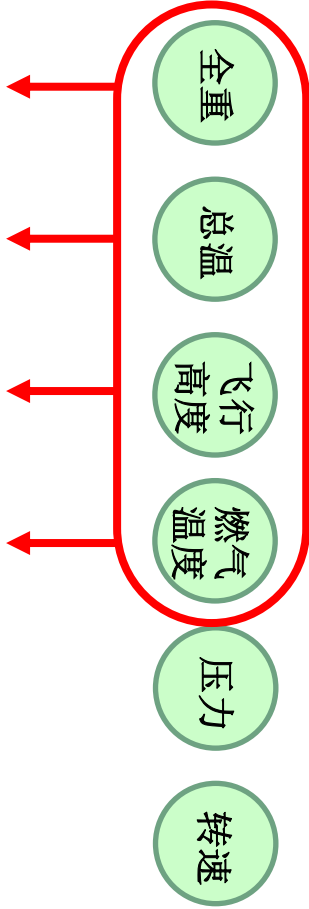
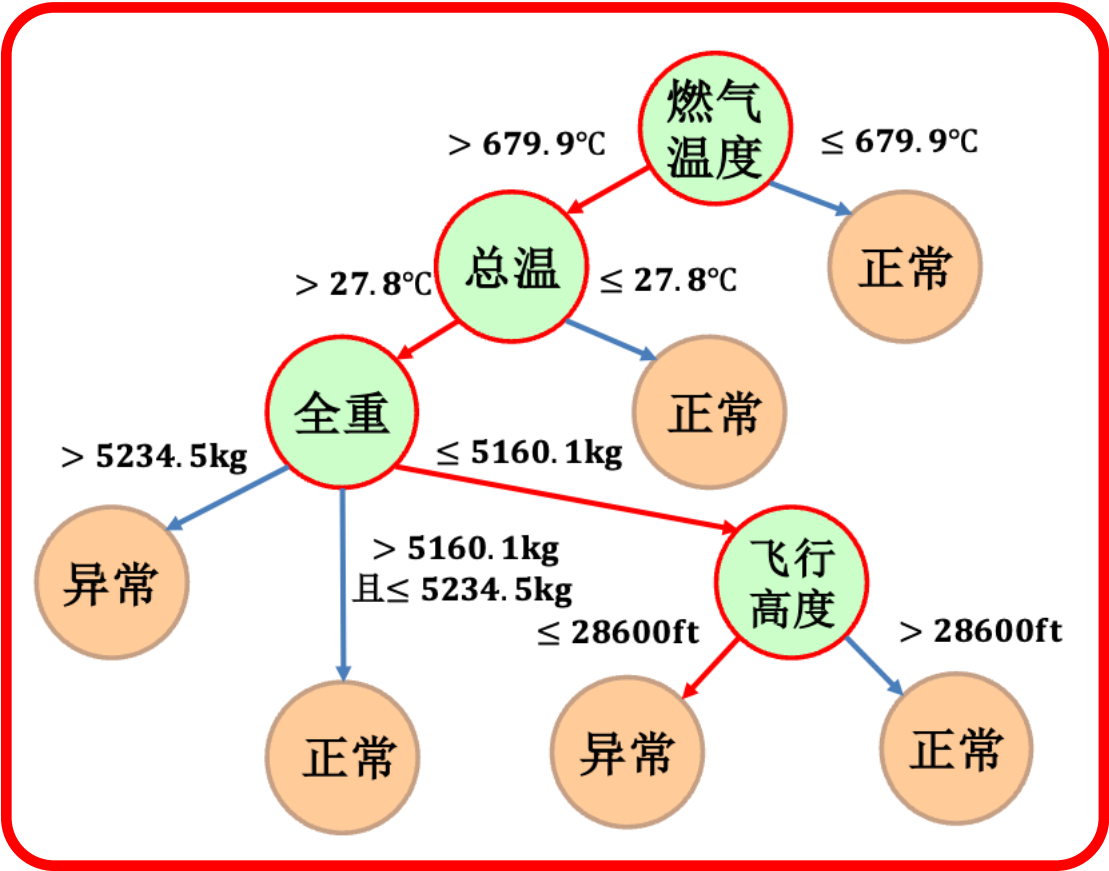
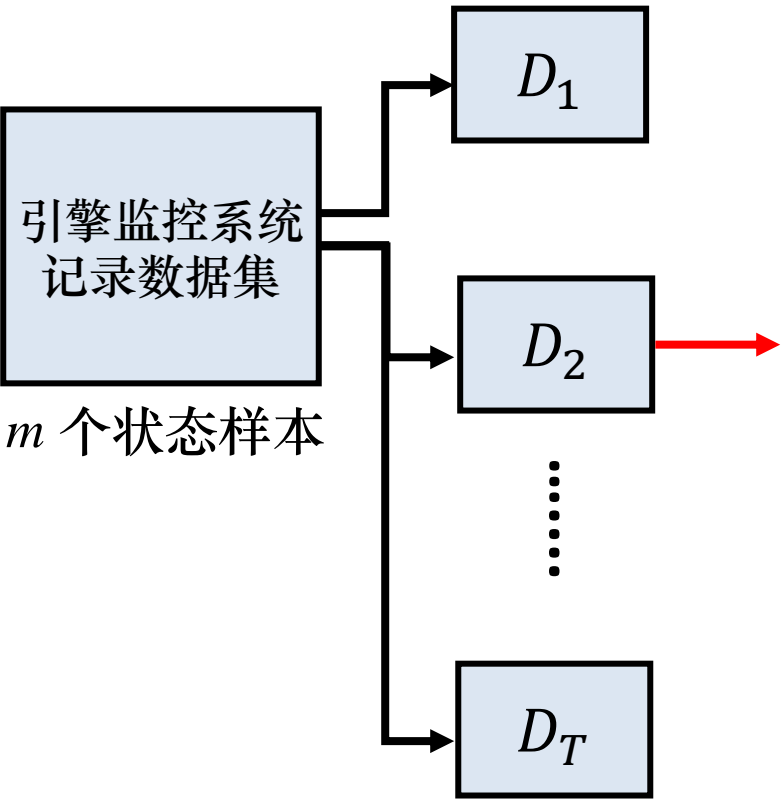
属性扰动 → 好而不同

随机森林算法

自助采样法构建数据集

航空发动机监控系统

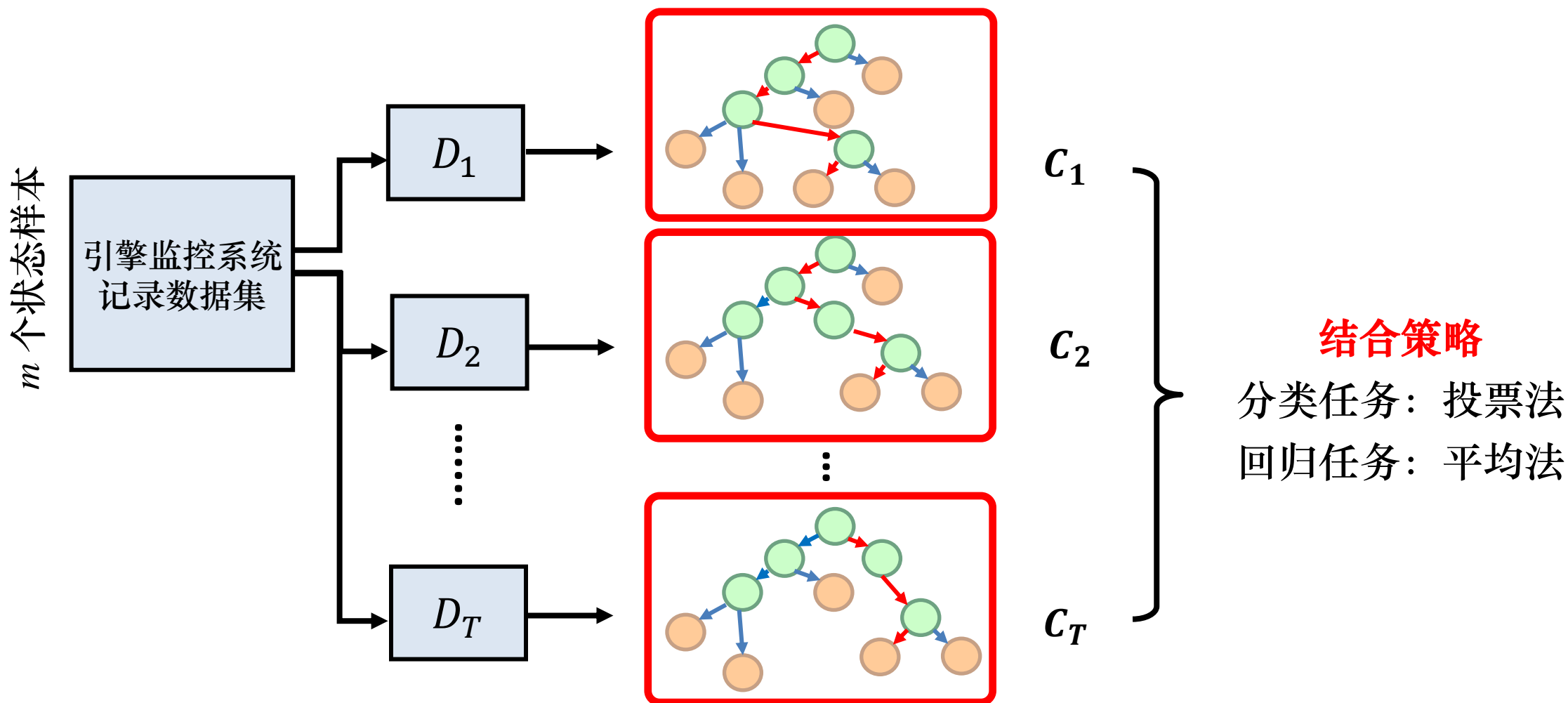
随机属性选择选择可分属性



d 个属性

随机森林算法

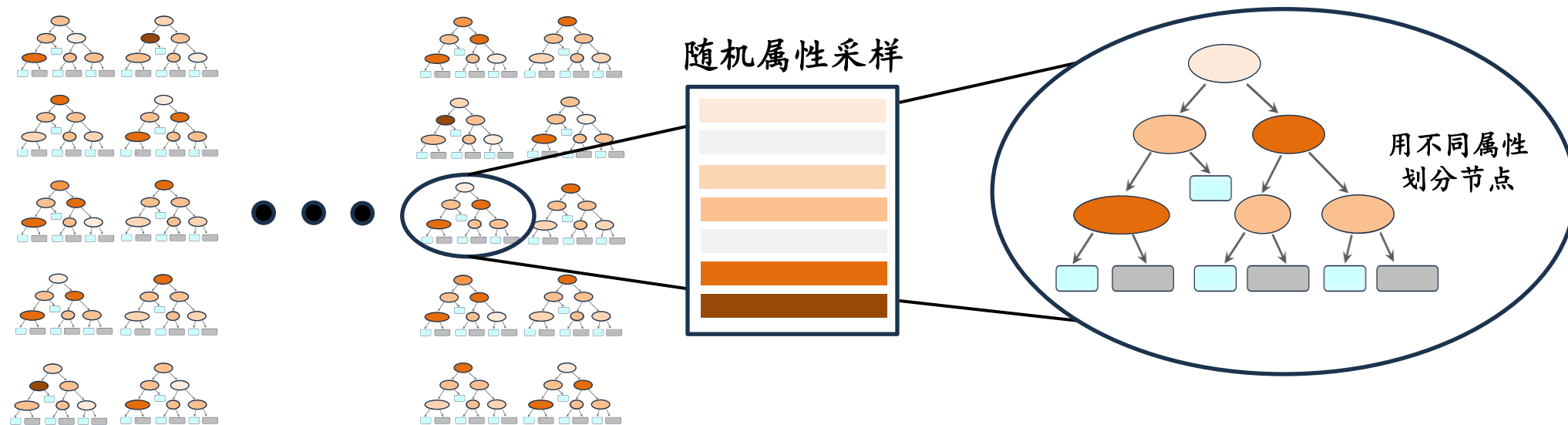
航空发动机监控系统



随机森林算法

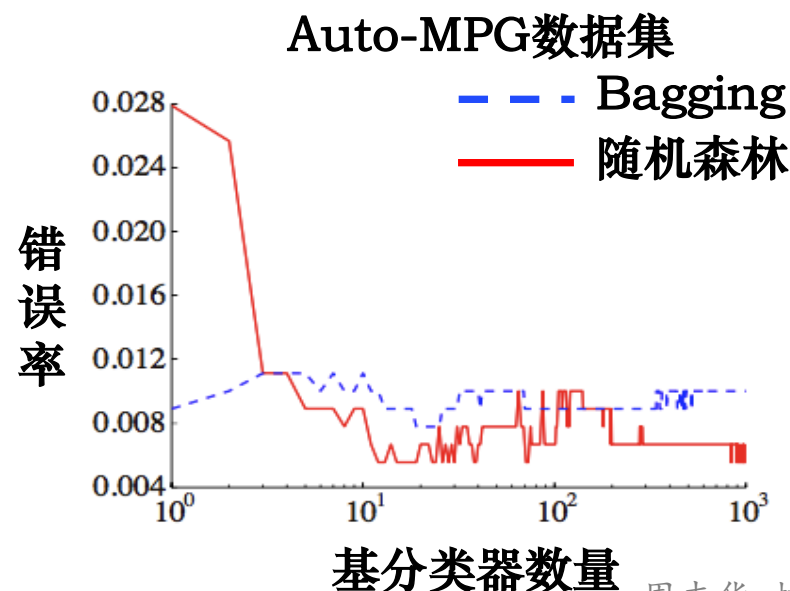
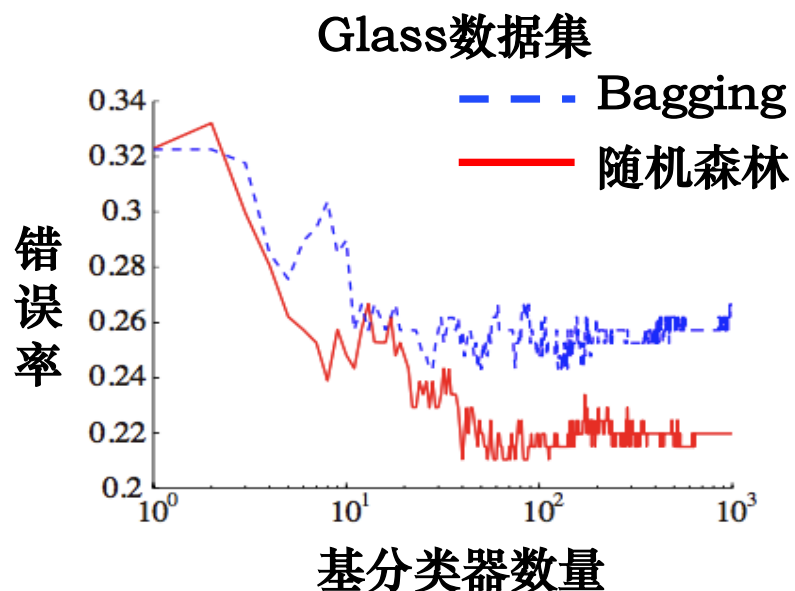
● 算法特点

- 基学习器多样性通过**样本扰动**和**属性扰动**实现
- 算法简单、容易实现、计算开销小
- 与Bagging方法相比性能更强，被誉为“**代表集成学习技术水平的方法**”



Bagging vs. 随机森林

- 随机森林与Bagging收敛性相似
 - 错误率随着基分类器数量增加最终趋于稳定
- 随机森林训练效率优于Bagging
 - 随机森林使用少量基分类器即可取得比Bagging方法更低的错误率



14.4 集成学习的结合策略及多样性

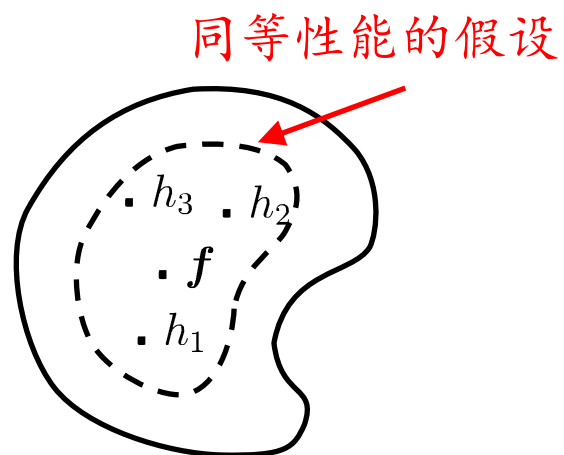
- 结合策略
- 多样性

结合策略

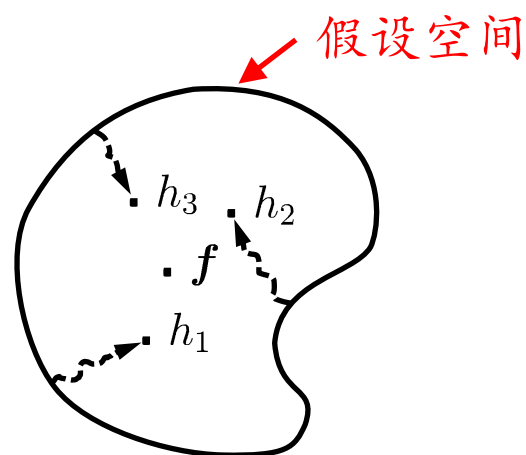
● 学习器的组合有三个方面的好处

【Dietterich, 2000】

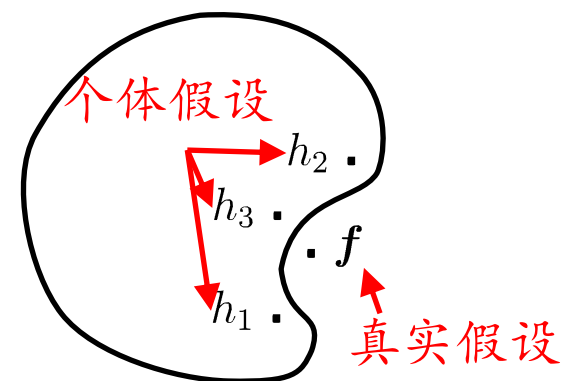
- 统计方面：减小误选假设空间导致泛化性能不佳的几率
- 计算方面：降低陷入坏局部极小点影响泛化性能的风险
- 表示方面：扩大假设空间学习对于真实空间更好的近似



(a) 统计的原因



(b) 计算的原因



(c) 表示的原因

结合策略——平均法

- 平均法 (Averaging) 是数值型输出最常见的结合策略

➤ 简单平均法 (Simple Averaging): $H(x) = \frac{1}{T} \sum_{i=1}^T h_i(x)$ 个体学习器性能相近时适用

➤ 加权平均法 (Weighted Averaging): 【1993年Perrone和Cooper正式将其用于集成学习】

$$H(x) = \sum_{i=1}^T w_i h_i(x), \quad w_i \geq 0 \text{ and } \sum_{i=1}^T w_i = 1 \quad \text{个体学习器性能迥异时适用}$$

加权平均法是集成学习的基本出发点，各种结合方法都可视为其特例或变体，不同的集成学习方法是通过不同的方式确定加权平均法中基学习器的权重

结合策略—投票法

- 投票法 (Voting) 是**标签型输出**最常见的结合策略

硬投票 (类标签)

软投票 (类概率)

标记集合 $\{c_1, c_2, \dots, c_N\}$, h_i 在样本 x 上的预测 $\{h_i^1(x), h_i^2(x), \dots, h_i^N(x)\}$

➤ 绝对多数投票法 (Majority Voting): $H(x) = \begin{cases} c_j & \text{if } \sum_{i=1}^T h_i^j(x) > \frac{1}{2} \sum_{k=1}^l \sum_{i=1}^T h_i^k(x) \\ \text{rejection} & \text{otherwise} \end{cases}$

得票超半数

➤ 相对多数投票法 (Plurality Voting): $H(x) = c_{\arg \max_j \sum_{i=1}^T h_i^j(x)}$

得票最多

➤ 加权投票法 (Weighted Voting): $H(x) = c_{\arg \max_j \sum_{i=1}^T w_i h_i^j(x)}$

加权后得票最多

结合策略—学习法

● 当训练数据很多时采用另一个学习器进行结合

初级学习器 vs. 次级学习器或元学习器 (Meta-learner)

➤ Stacking是学习法的典型代表

【1992年Wolpert提出】

输入：训练集 $D = \{(x_1, y_1), (x_2, y_2), \dots, (x_m, y_m)\}$;
初级学习算法 $\mathcal{Q}_1, \mathcal{Q}_2, \dots, \mathcal{Q}_T$; 次级学习算法 \mathcal{Q} .

1: **for** $t = 1, 2, \dots, T$ **do**:

2: $h_t = \mathcal{Q}_t(D)$;

3: $D' = \emptyset$;

4: **for** $i = 1, 2, \dots, m$ **do**:

5: **for** $t = 1, 2, \dots, T$ **do**:

6: $z_{it} = h_t(x_i)$;

7: $D' = D' \cup ((z_{i1}, z_{i2}, \dots, z_{iT}), y_i)$;

8: $h' = \mathcal{Q}(D')$;

输出： $H(x) = h'(h_1(x), h_2(x), \dots, h_T(x))$

从初始数据集训练初级学习器

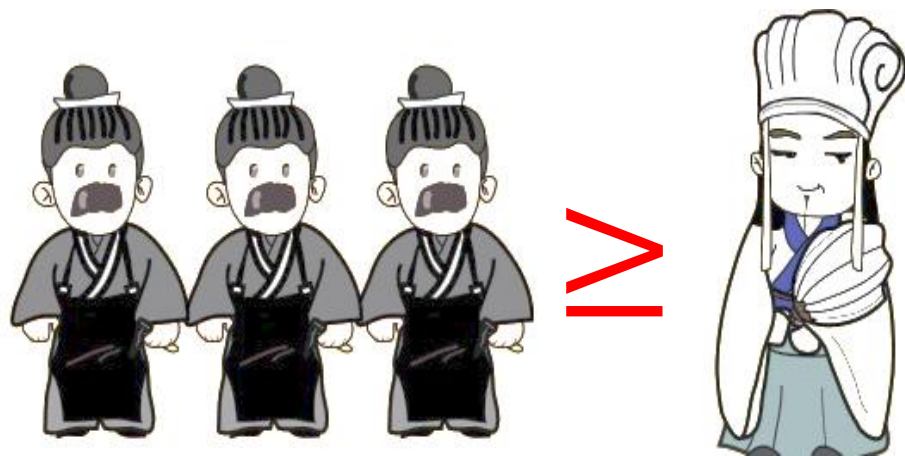
生成次级数据集：初级学习器的输出被当作样例输入特征，继承初始样本标记。

从次级数据集训练次级学习器

多样性

● 简单分析

- **关键假设**：基学习器的误差相互独立
- 现实任务中，个体学习器是为解决同一个问题训练出来的，显然不可能互相独立！
- 个体学习器的“**准确性**”和“**多样性**”存在冲突



集成学习的研究核心：

如何产生并结合“好而不同”的个体学习器



多样性

- 误差-分歧分解：对个体学习器“好而不同”的理论分析

- 假设个体学习器 $\{h_1, h_2, \dots, h_T\}$ 通过加权平均法结合产生集成，用于回归学习任务 $f: \mathbb{R}^d \mapsto \mathbb{R}$ ，对示例 x ：

- **分歧** – 定义学习器 h_i 的分歧 (Ambiguity): $A(h_i | x) = (h_i(x) - H(x))^2$

- 表征了个体学习器在样本 x 上的**不一致性**
- 在一定程度上反映了个体学习器的**多样性**

- **集成分歧**: $\bar{A}(h | x) = \sum_{i=1}^T w_i A(h_i | x) = \sum_{i=1}^T w_i (h_i(x) - H(x))^2$

多样性

- 误差-分歧分解：对个体学习器“好而不同”的理论分析

- 假设个体学习器 $\{h_1, h_2, \dots, h_T\}$ 通过加权平均法结合产生集成，用于回归学习任务 $f: \mathbb{R}^d \mapsto \mathbb{R}$ ，对示例 x ：

- **误差** – 个体学习器 h_i 和集成 H 的平方误差分别为：

$$E(h_i | x) = (f(x) - h_i(x))^2$$

$$E(H | x) = (f(x) - H(x))^2$$

- 令 $\bar{E}(h | x) = \sum_{i=1}^T w_i \cdot E(h_i | x)$ 表示个体学习器**误差的加权均值**，有：

$$\bar{A}(h | x) = \sum_{i=1}^T w_i E(h_i | x) - E(H | x) = \bar{E}(h | x) - E(H | x)$$

多样性

- 误差-分歧分解：对个体学习器“好而不同”的理论分析

- 上式对所有样本 x 均成立，令 $p(x)$ 表示样本概率密度，则在全样本上有：

$$\sum_{i=1}^T w_i \int A(h_i | x) p(x) dx = \sum_{i=1}^T w_i \int E(h_i | x) p(x) dx - \int E(H | x) p(x) dx$$

- 个体学习器 h_i 在全样本上的泛化误差和分歧项分别为：

$$E_i = \int E(h_i | x) p(x) dx \quad A_i = \int A(h_i | x) p(x) dx$$

- 集成的泛化误差为： $E = \int E(H | x) p(x) dx$

多样性

- 误差-分歧分解：对个体学习器“好而不同”的理论分析

- 个体学习器泛化**误差**的加权均值： $\bar{E} = \sum_{i=1}^T w_i E_i$

- 个体学习器的加权**分歧**值： $\bar{A} = \sum_{i=1}^T w_i A_i$

多样性

- 误差-分歧分解：对个体学习器“好而不同”的理论分析

➤ 上式对所有样本 x 均成立，令 $p(x)$ 表示样本概率密度，则在全样本上有：

$$\sum_{i=1}^T w_i \int A(h_i | x) p(x) dx = \sum_{i=1}^T w_i \int E(h_i | x) p(x) dx - \int E(H | x) p(x) dx$$

➤ 个体学习器 h_i 在全样本上的泛化误差和分歧项分别为：

$$E_i = \int E(h_i | x) p(x) dx \quad A_i = \int A(h_i | x) p(x) dx$$

代入

➤ 集成的泛化误差为： $E = \int E(H | x) p(x) dx$

多样性

- 误差-分歧分解：对个体学习器“好而不同”的理论分析

- 个体学习器泛化误差的加权均值： $\bar{E} = \sum_{i=1}^T w_i E_i$

- 个体学习器的加权分歧值： $\bar{A} = \sum_{i=1}^T w_i A_i$

- 综上： $E = \bar{E} - \bar{A}$

误差-分歧分解 (Error Ambiguity Decomposition):

“个体学习器精确性越高、多样性越大，则集成效果越好”

【1995年Krogh和Vedelsby给出】

多样性

- 多样性度量 (Diversity Measure)

- 用于度量集成中个体学习器的多样性
- 考虑个体学习器的两两相似/不相似性

给定数据集 $D = \{(x_1, y_1), (x_2, y_2), \dots, (x_m, y_m)\}$, 对二分类任务 分类器 h_i 与 h_j 的预测结果联立表 (Contingency Table) 为:

	$h_i = +1$	$h_i = -1$
$h_j = +1$	a	c
$h_j = -1$	b	d

$$a + b + c + d = m$$

多样性度量

➤ 不合度量 (Disagreement Measure)

$$\text{dis}_{ij} = \frac{b + c}{m}$$

值域 $[0, 1]$ ，值越大多样性越大

➤ 相关系数 (Correlation Coefficient)

$$\rho_{ij} = \frac{ad - bc}{\sqrt{(a + b)(a + c)(c + d)(b + d)}}$$

值域 $[-1, 1]$ ，学习器无关值为0；正相关值为正，否则为负

➤ Q -统计量 (Q-Statistic)

$$Q_{ij} = \frac{ad - bc}{ad + bc}$$

Q_{ij} 与相关系数 ρ_{ij} 符号相同，且 $|Q_{ij}| \leq |\rho_{ij}|$

➤ κ -统计量 (Kappa-Statistic)

$$\kappa = \frac{p_1 - p_2}{1 - p_2}$$

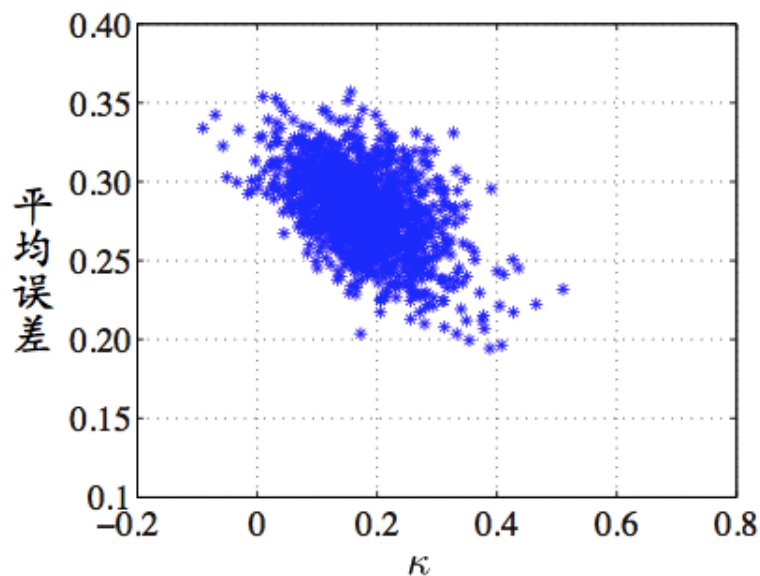
一致概率： $p_1 = (a + d)/m$ 偶然一致概率： $p_2 = ((a + b)(a + c) + (c + d)(b + d))/m^2$

学习器完全一致值为1；偶然一致值为0，一致概率低于偶然取负值

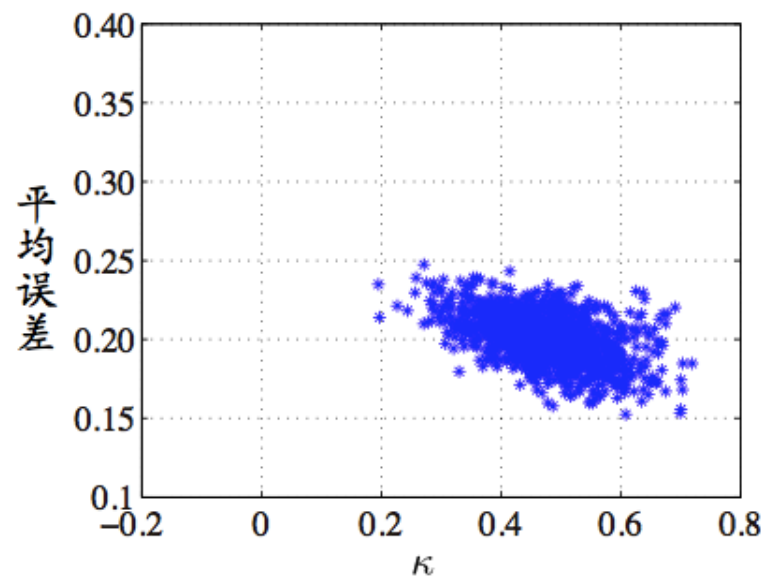
多样性度量

➤ k -误差图

UCI数据tic-tac-toe图，每个集成含50棵C4.5决策树。



(a) AdaBoost 集成



(b) Bagging 集成

横坐标是学习器的 κ 值，纵坐标是其平均误差。

数据点云位置越高，个体学习器准确性越低；点云位置越靠右，个体学习器多样性越小。

多样性

- 多样性增强：在学习过程引入随机性

- 数据样本扰动
- 输入属性扰动
- 输出表示扰动
- 算法参数扰动

不同的多样性增强机制也可一起使用

- AdaBoost：加入了数据样本扰动
- 随机森林：同时加入了数据样本扰动和输入属性扰动

多样性增强

- **数据样本扰动**：通常是基于采样法

- Bagging中的自助采样
- AdaBoost中的序列采样

对数据样本扰动敏感的基学习器（不稳定基学习器）

决策树，神经网络等

效果明显

对数据样本扰动不敏感的基学习器（稳定基学习器）

线性学习器，支持向量机，朴素贝叶斯， K 近邻等

效果不明显

多样性增强

● 输入属性扰动：不同子空间提供观察数据的不同视角

➤ 典型算法：随机子空间算法 [Ho, 1998]

输入：训练集 $D = \{(x_1, y_1), (x_2, y_2), \dots, (x_m, y_m)\}$;
基学习器 \mathcal{L} ; 训练轮数 T ; 子空间属性数 d' .

1: **for** $t = 1, 2, \dots, T$ **do**:

2: $\mathcal{F}_t = \text{RS}(D, d')$

3: $D_t = \text{Map}_{\mathcal{F}_t}(D)$

4: $h_t = \mathcal{L}(D_t)$

5: **end for**

输出： $H(x) = \arg \max_{y \in \mathcal{Y}} \sum_{t=1}^T \mathbb{I}(h_t(\text{Map}_{\mathcal{F}_t}(x)) = y)$

d' 小于初始属性数 d

\mathcal{F}_t 包含 d' 个随机选取的属性

D_t 仅保留 \mathcal{F}_t 中的属性

对包含大量冗余属性数据，可产生多样性大的个体学习器，
还因属性数减少会大幅节省时间开销；若数据只含少量属性或冗余属性较少，则不宜使用。

多样性增强

● 输出表示扰动：操纵输出表示

- 翻转法 (Flipping Output): 对训练样本的类标记稍作变动 **【2000年Brieman提出】**
随机改变一些训练样本的标记
- 输出调制法 (Output Smearing): 对输出表示进行转化 **【2000年Brieman提出】**
将分类输出转为回归输出构建个体学习器
- ECOC法: 将原任务拆解为多个可同时求解的子任务 **【1995年Dietterich和Bakiri提出】**
利用纠错输出码将多分类任务拆解为一系列二分类任务训练基学习器

多样性增强

- **算法参数扰动**：随机设置不同的参数或环节

- 负相关法 (Negative Correlation): **【1999年Liu和Yao提出】**

参数较多时 显式地通过正则化项强制个体神经网络使用不同参数

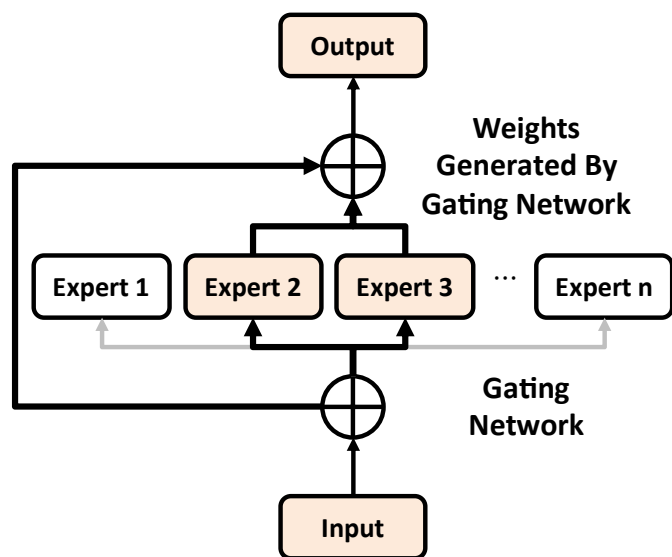
参数较少时 例如，可将决策树使用的属性选择机制用其他类似方式代替

单一学习器利用交叉验证对参数寻优，这事实上相当于使用了不同的参数训练学习器，最后仅选择了一个；而集成学习相当于把所有学习器都利用起来。

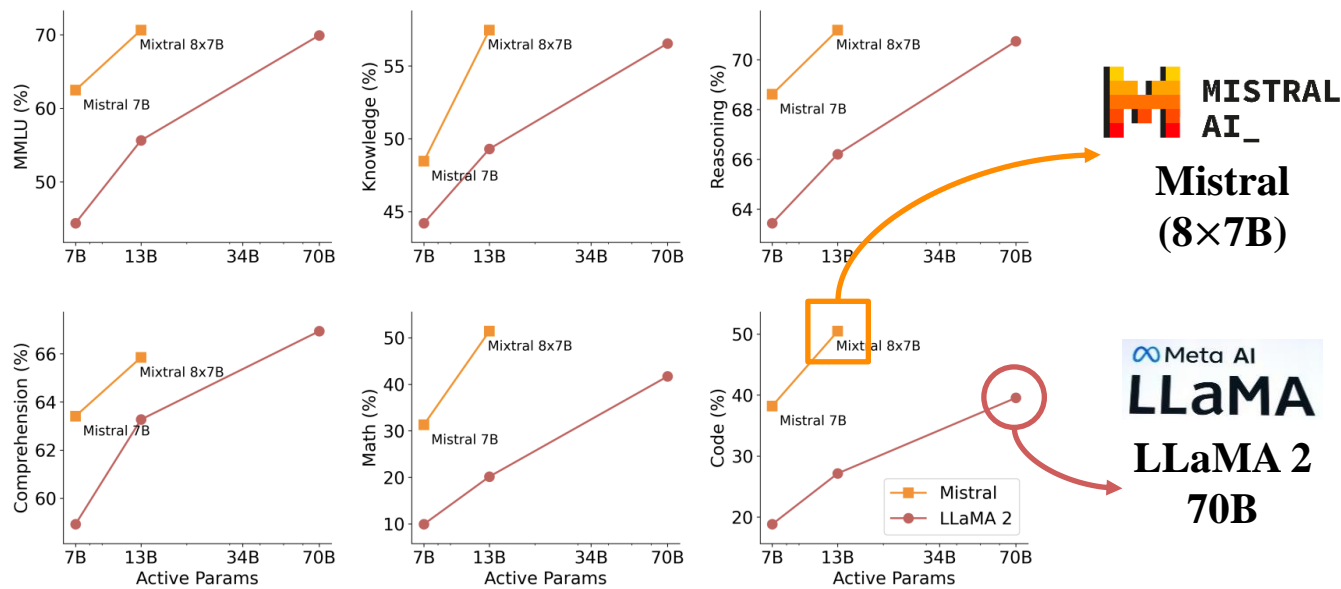
集成思想

● 大模型中的集成思想：混合专家模型 (Mixture of Experts, MoE)

- 混合专家模型 (Mixture of Experts, MoE)，或译为多专家模型，是一种机器学习技术。通过门控 (Gating) 模型将单一任务空间划分为多个子任务，再由多个专家网络分别处理特定的子任务，最终得到整体的预测结果



混合专家模型



引入MoE的Mixtral使用约 $\frac{1}{5}$ (13B) 的激活参数达到甚至超越LLaMA 2 (70B) 的性能

集成思想

● 现代科学和工程中的集成思想：复杂，多学科交叉融合、多方面人才协作

“两弹一星”精神：“热爱祖国、无私奉献，自力更生、艰苦奋斗，大力协同、勇于登攀”



“两弹一星”



“两弹一星”元勋

第15章：半监督学习

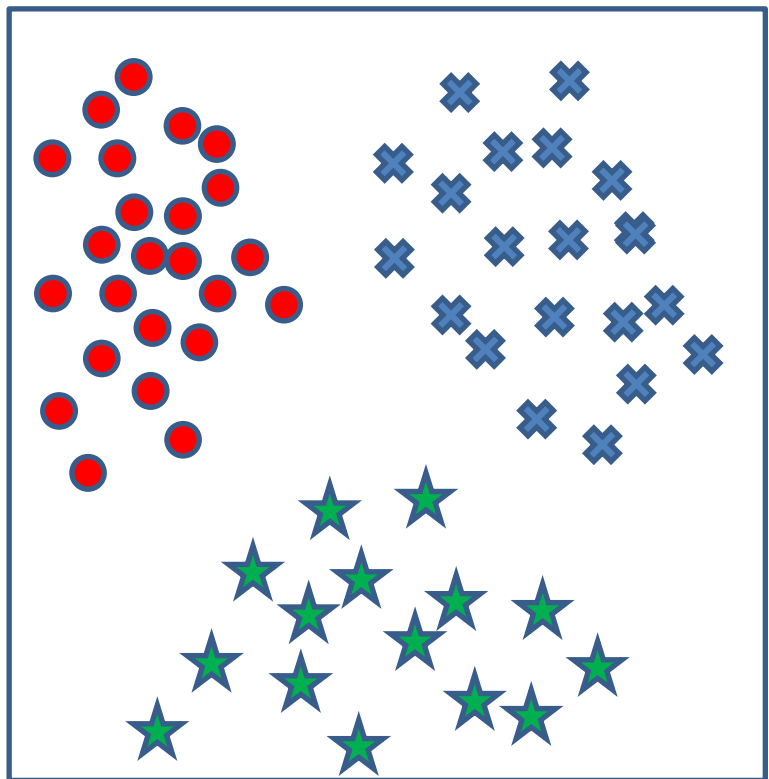
Chapter 15: Semi-Supervised Learning

15.1 什么是半监督学习?

- 半监督学习的定义
- 数据分布假设
- 半监督学习的应用

监督学习回顾

- 利用标记样本进行学习生成学习器，再对未标记样本进行测试



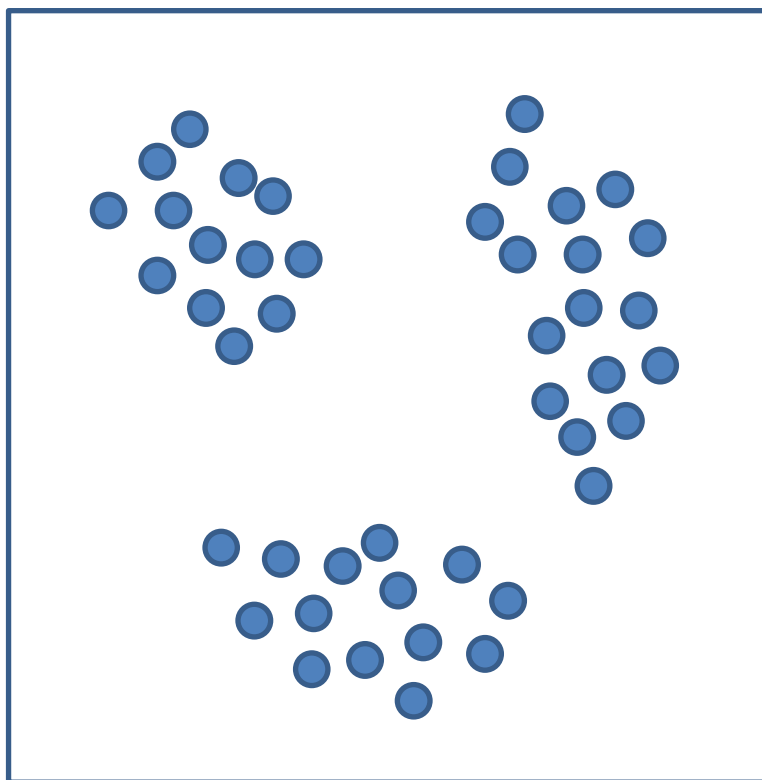
- 问题：需要大量标记样本

- 标注数据复用性较弱
- 大规模人工标注耗时、成本高昂



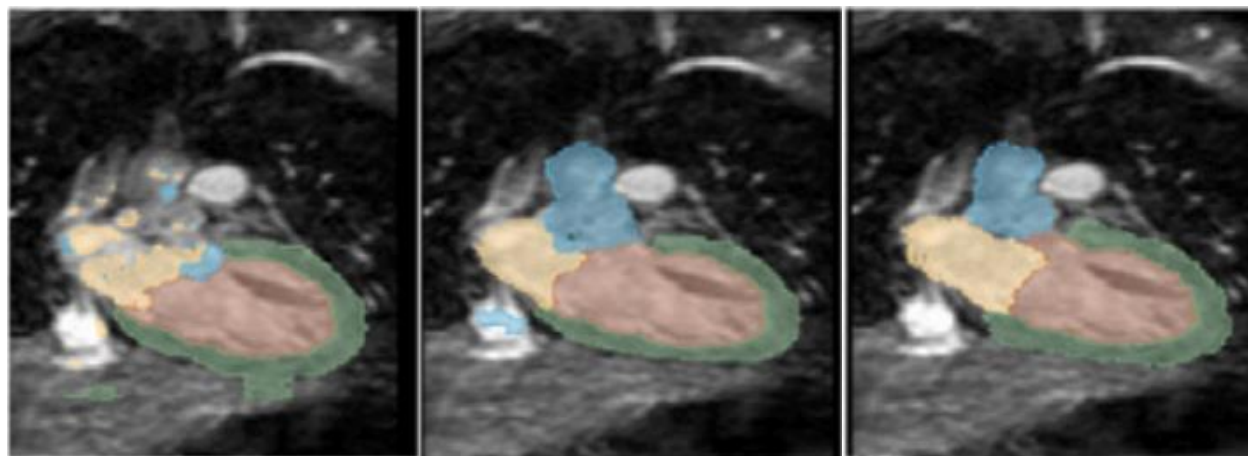
无监督学习回顾

- 使用未标记样本，基于样本间的相似度，归纳样本的类别



- 问题：完全基于未标记样本学习

➤ 精度难以保证



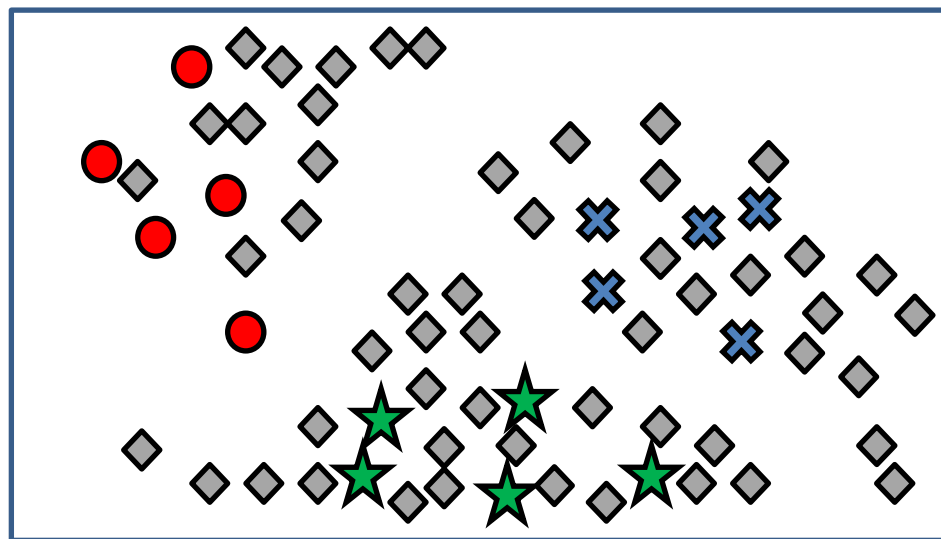
无监督学习

监督学习

人工标注

实际应用中面临的问题

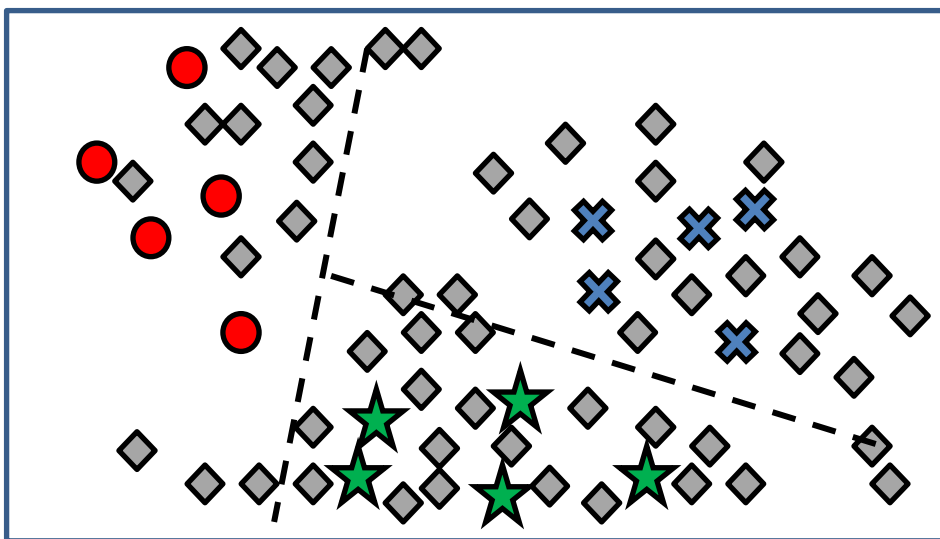
- 只有少量有标记样本，但只使用少量有标记样本，训练出来的学习系统往往难以具有良好的泛化性能
- 往往有大量未标记样本，仅使用少量“昂贵的”标记样本而不利用“廉价的”未标记样本是对数据资源的浪费



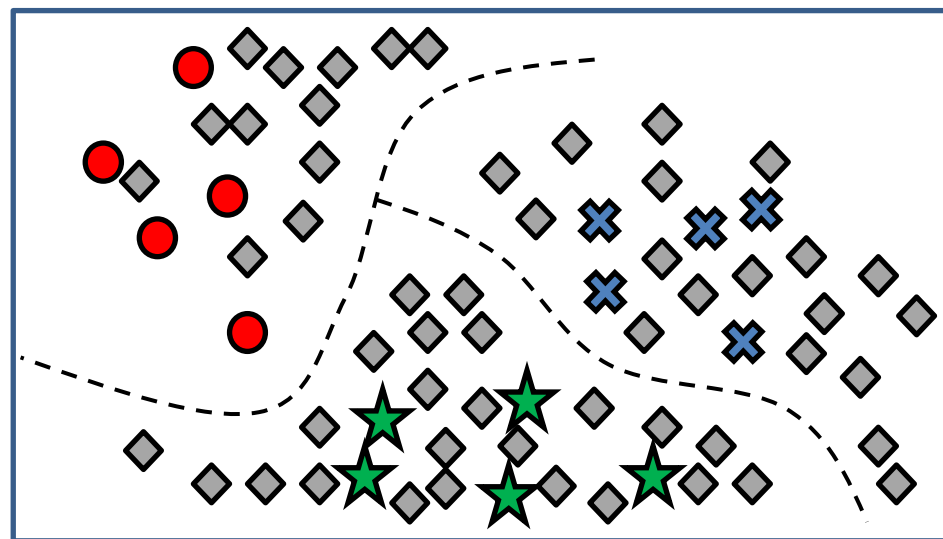
半监督学习

● 半监督学习的目标

- 在训练样本的部分信息缺失，特别是样本数据的类别标记缺失的情况下，获得具有良好泛化性能的学习器，即利用大量的未标记样本辅助标记样本建立一个更好的学习器



使用标记样本进行监督学习



使用标记和未标记样本进行半监督学习

半监督学习的定义

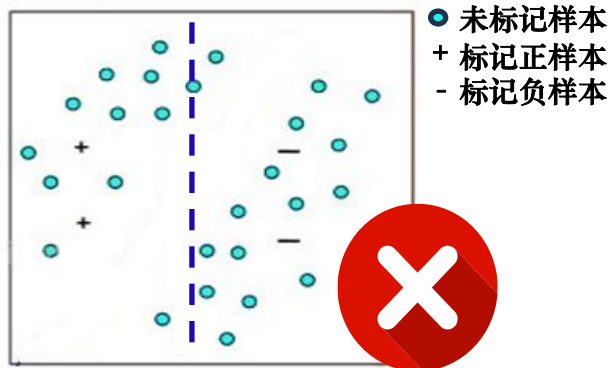
- 给定一个来自未知分布的样本集 $S = D_l \cup D_u$
 - D_l 是 **标记样本集** $D_l = \{(x_1, y_1), \dots, (x_l, y_l)\}$, D_u 是**未标记样本集** $D_u = \{x_{l+1}, \dots, x_{l+u}\}$ 。其中, x 为 d 维向量, $y_i \in Y$ 为 D_l 中样本 x_i 的标记
- 半监督学习就是在样本集 $S = D_l \cup D_u$ 上寻找最优学习器, 即函数 $f: X \rightarrow Y$, 可以准确地对样本 x 预测其标记 y

这个函数可以是含参数的 (如支持向量机等)
可以是不含参数的 (如最近邻法等)

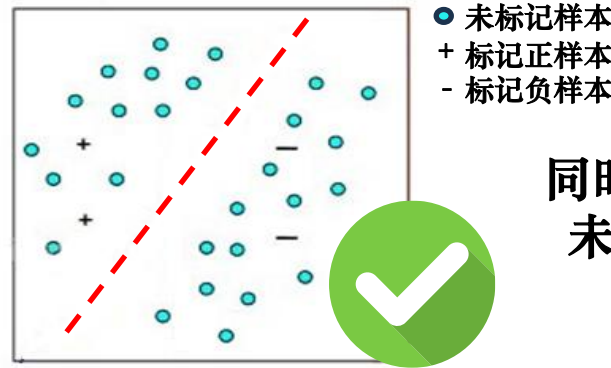
数据分布假设

- 如何利用未标记样本的之间的联系---数据分布假设
- 常用的数据分布假设：聚类假设
 - **聚类假设**：假设数据存在**簇结构**，同一个簇的样本属于同一个类别
 - 决策边界应通过数据较稀疏的地方，避免把同一簇中的样本分到决策边界两侧
 - 大量未标记样本的作用：帮助探明样本空间中数据分布的稠密和稀疏区域

仅使用标记样本，
不使用未标记样本



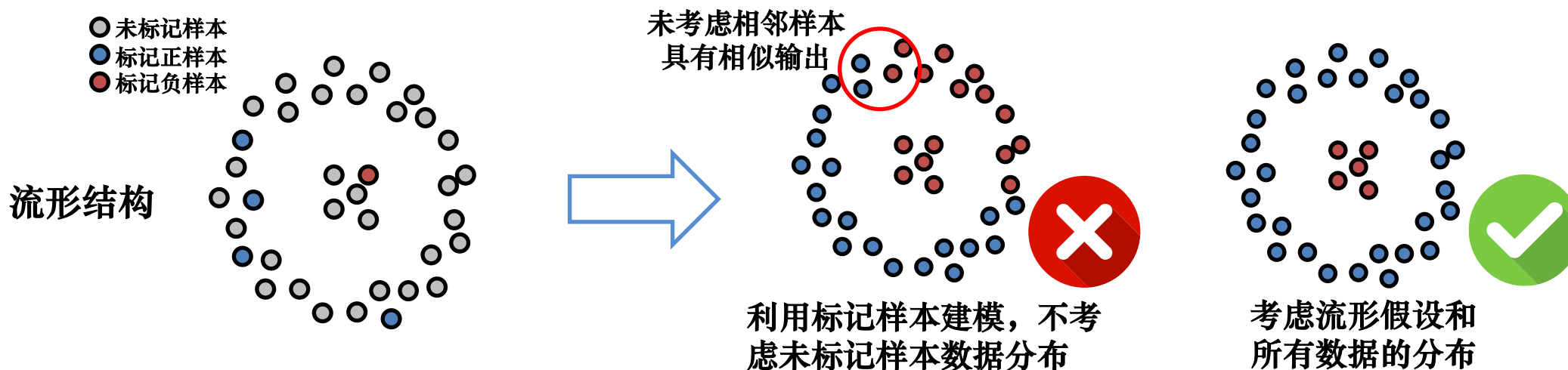
同时用标记和
未标记样本



数据分布假设

● 常用的数据分布假设：流形假设

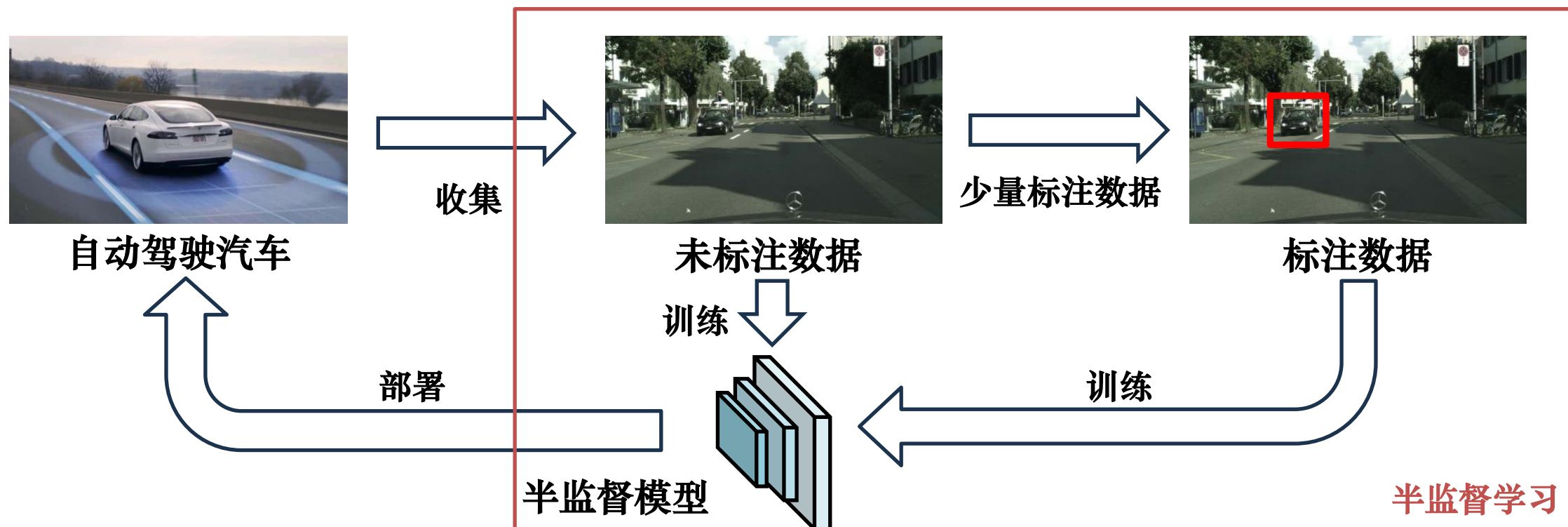
- **流形假设**：数据分布在一个**流形结构**上，邻近的样本拥有相似的输出值
- 主要考虑模型的局部特性，反映决策函数的**局部平滑性**
- 大量未标记示例的作用：让数据空间变得更加稠密，有助于**更加准确地刻画局部区域**的特性，使决策函数更好地拟合数据



半监督学习的应用

● 自动驾驶

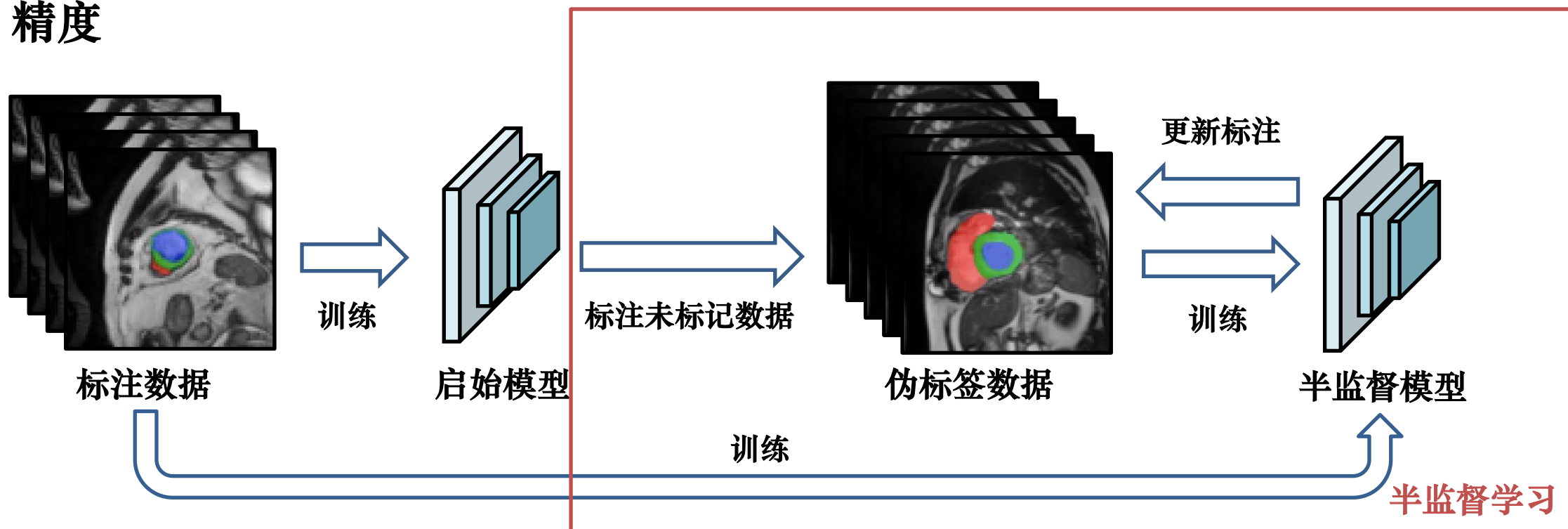
- 问题：公司收集大量数据，大规模数据难以全量标注
- 半监督学习：选取典型的边界案例进行标注，使模型覆盖更多的真实场景



半监督学习的应用

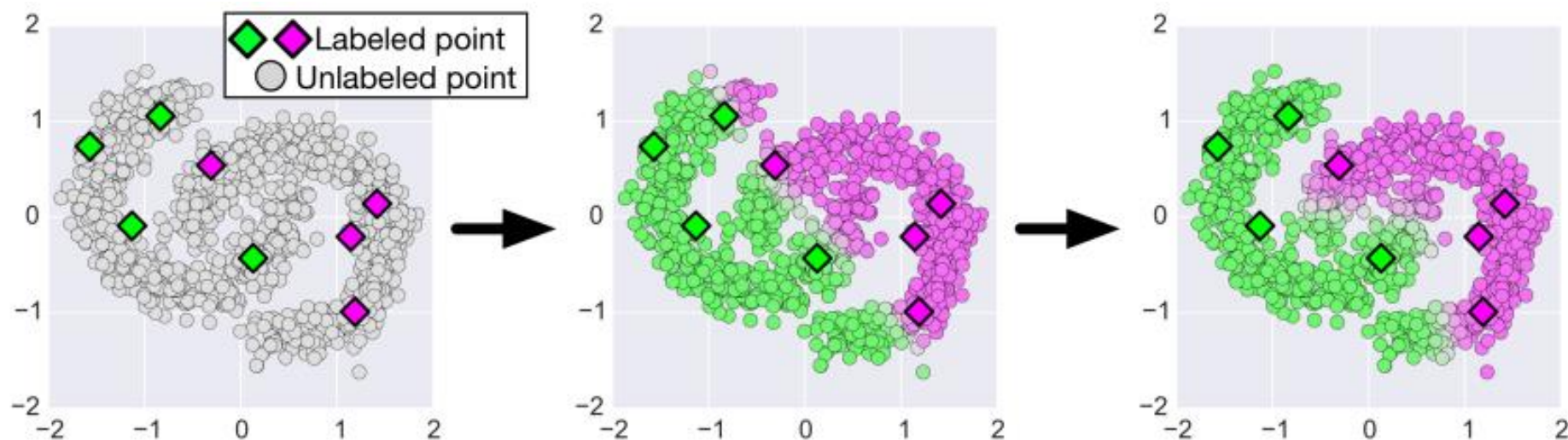
● 医学图像

- 问题：医学图像**标注昂贵**，医疗机构**难以大量标注数据**
- 半监督学习：通过对未标记样本标注伪标签并利用伪标签样本训练，提高模型精度



经典半监督算法

- 自学习 (Self-training)
- 直推式支持向量机 (Transductive SVM)
- 图半监督学习 (Graph Semi-Supervised Learning)
- 半监督聚类 (Semi-Supervised Clustering)
- ...



15.2 自学习

- 自学习的定义
- 自学习的核心步骤
- 自学习的典型方法

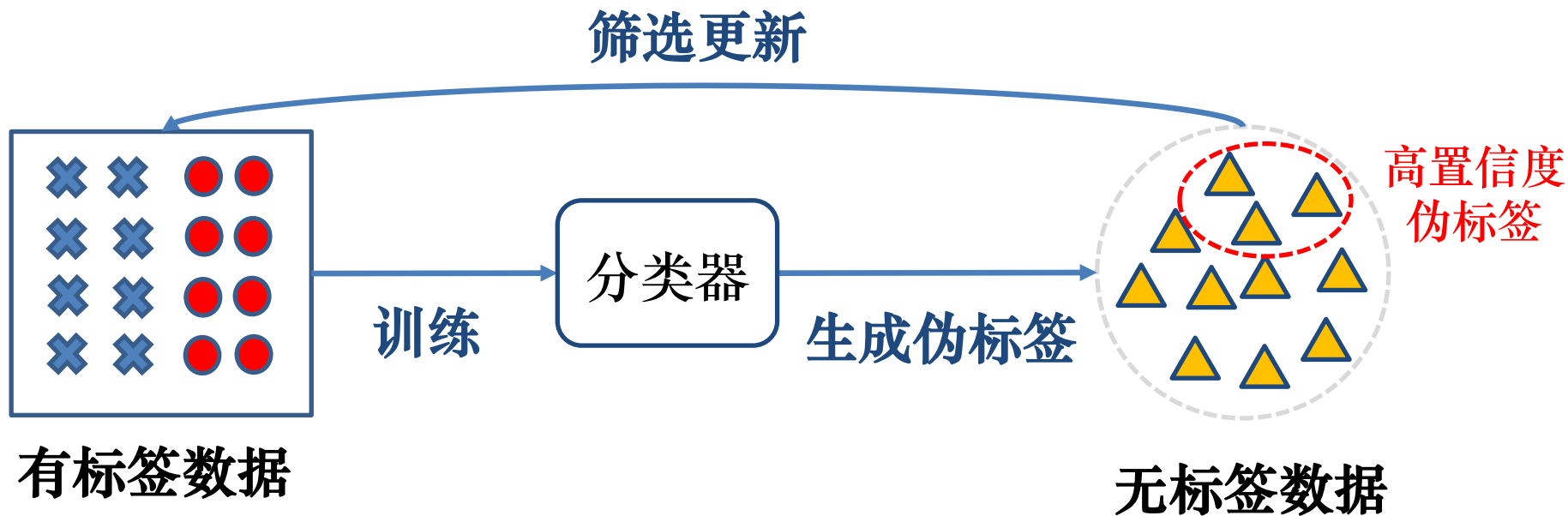
什么是自学习?

- 自学习 (Self-training)

- 一种**启发式**的半监督学习方法：用有标签数据训练机器学习模型，然后在无标签数据上进行推理生成伪标签 (Pseudo-label)，通过样本选择策略筛选高质量标签，不断迭代训练模型
- 自学习方法的优点：模型简单，容易与多种机器学习模型相结合而不改变其内部工作方式，包括从简单的kNN模型到复杂的神经网络分类器等
- 自学习方法的样本选择策略：每次递归仅将满足设定置信度阈值，即**置信度高**的**样本**纳入已标记样本集中，参与递归拟合

自学习的核心步骤

● 自学习的核心步骤



步骤1：用已标记的样本来训练得到一个初始分类器

步骤2：用初始分类器对未标记样本进行分类，将标记置信度高的未标记样本进行标记

步骤3：对所有样本进行重新训练，直到将所有未标记样本都标记为止

自学习的典型方法

● 最近邻自学习算法

- 在自学习算法中采用**最近邻的样本筛选策略**，每次选择离已标记样本最近的无标记样本进行标记

步骤1：用已标记样本 D_l 生成分类器 f

步骤2：选择 $x = \operatorname{argmin} d(x, x_0)$, $x \in D_u$, $x_0 \in D_l$ ，也就是选择离已标记样本最近的无标记样本

步骤3：用 f 给 x 确定一个类别 $f(x)$ ，并将 $(x, f(x))$ 加入 D_l 中

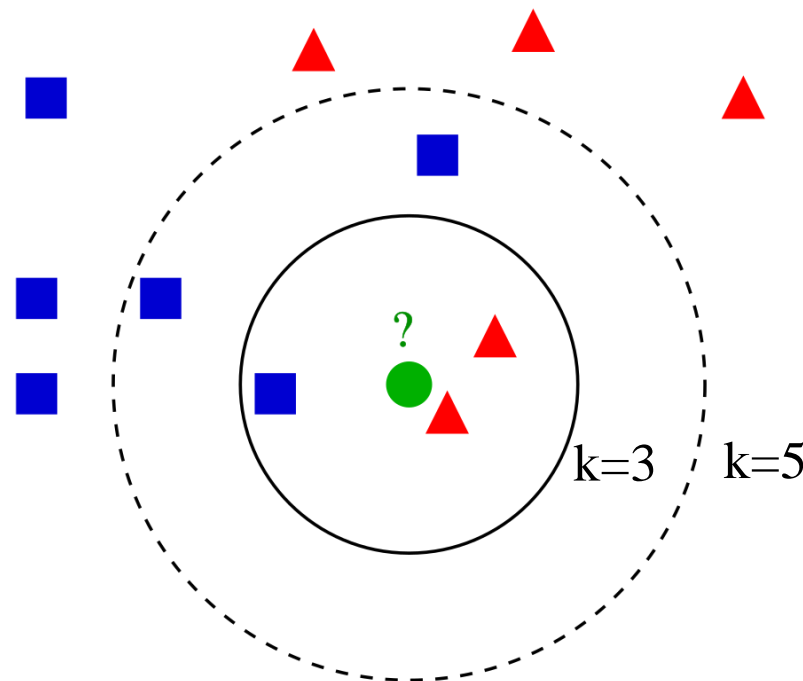
步骤4：重复上述步骤1-3，直到 D_u 为空集

其中 $d(x_1, x_2)$ 为两个样本的欧式距离

自学习的典型方法

● k-近邻算法

- k-近邻算法是用于分类和回归任务的**非参数统计的监督**方法，使用邻近度对单个数据点的分组进行分类或预测，是最简单的机器学习算法之一
- 分类任务中，样本的分类结果是由其邻居的“多数表决”确定的；回归任务中，样本的属性值为k个邻居的值的平均值
- 常用的距离度量：欧氏距离、曼哈顿距离、汉明距离

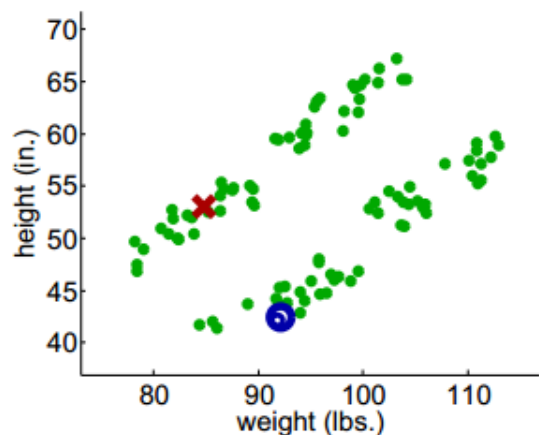


k=1: 最近邻算法

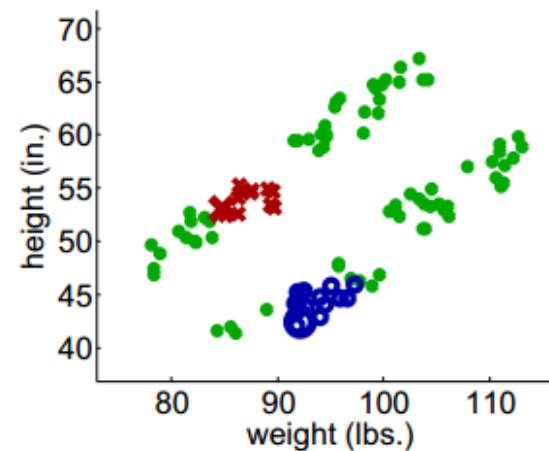
自学习的典型方法

● 最近邻自学习算法的示意图

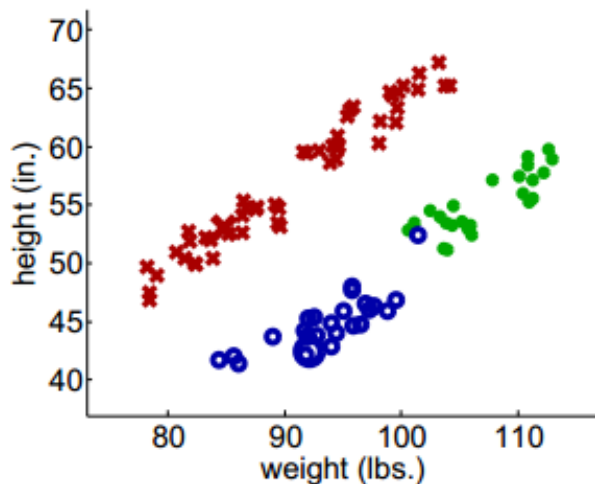
初始情况



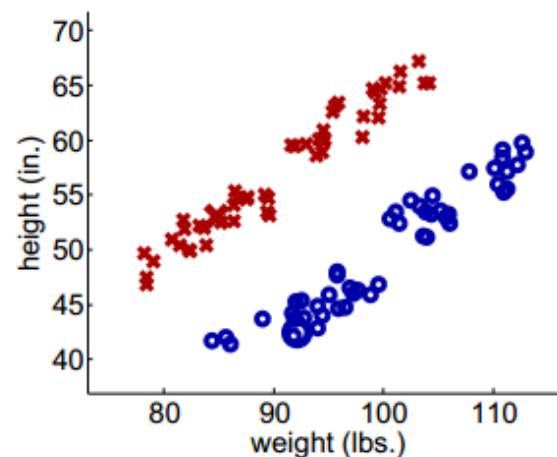
第25轮迭代



第75轮迭代



最终结果



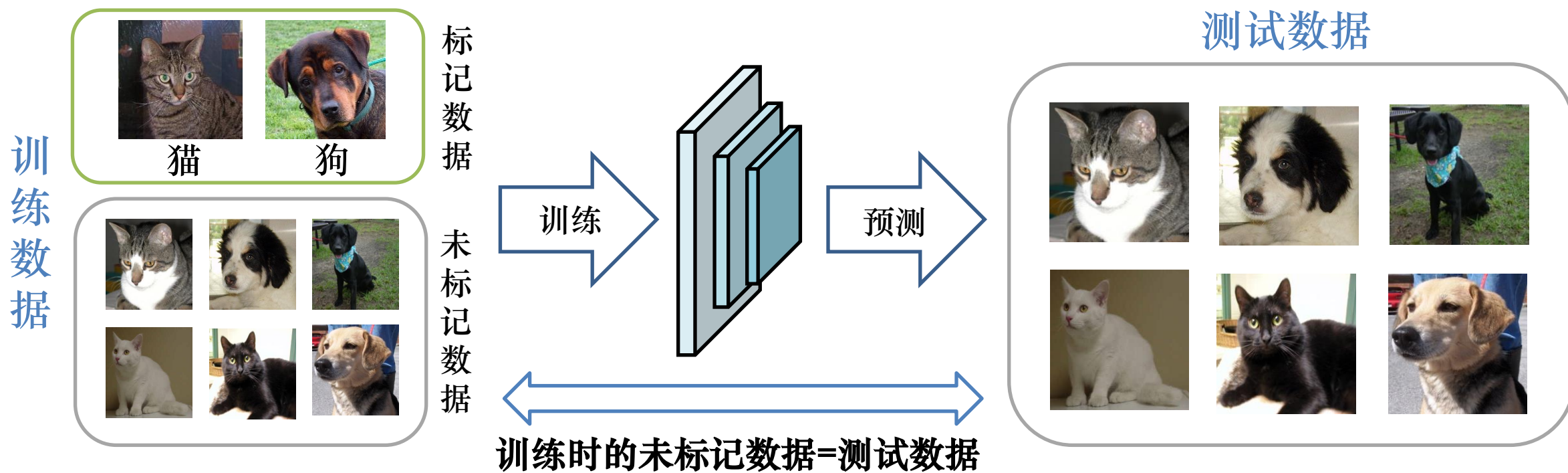
15.3 直推半监督学习

- 直推半监督学习的定义
- 直推支持向量机

直推半监督学习

● 直推半监督学习

- 基于“封闭世界”的假设，仅试图对学习过程中观察到的未标记数据进行预测
- 代表性方法：直推支持向量机、标签传播算法等



直推式支持向量机

- **直推式支持向量机** (Transductive SVM, T-SVM)

- 直推支持向量机 (T-SVM) 是一种结合了直推学习和支持向量机 (SVM) 思想的分类方法。通过同时考虑已知的标记样本和待分类的未标记样本，优化分类超平面，提高分类的准确性和泛化性能

- **基本思想**

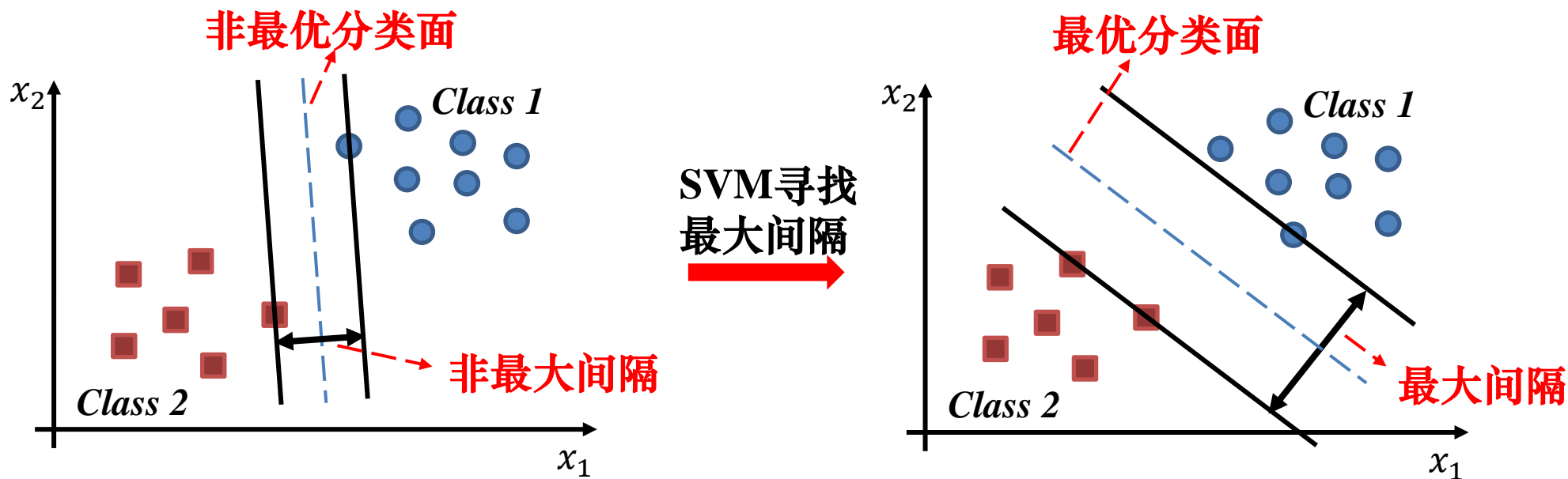
- 针对二分类问题，同时利用标记和未标记样本，**通过尝试将每个未标记样本分别作为正例和反例来寻找最优分类边界**，得到在原始数据中具有最大分类间隔的分类超平面

直推式支持向量机

● 标准SVM

➤ 标准SVM从线性可分情况下的**最优分类面**发展而来

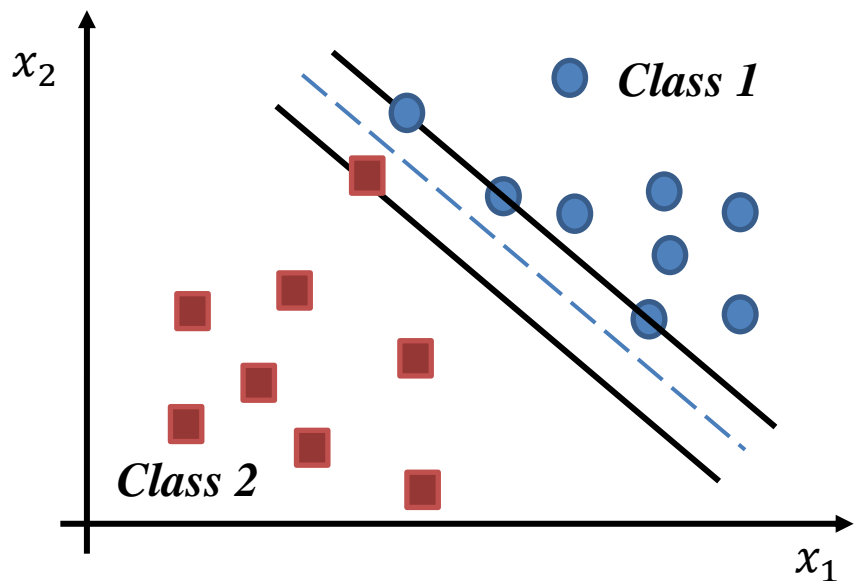
最优分类面：要求分类面不但能将两类正确分开，且使分类**间隔**最大



直推式支持向量机

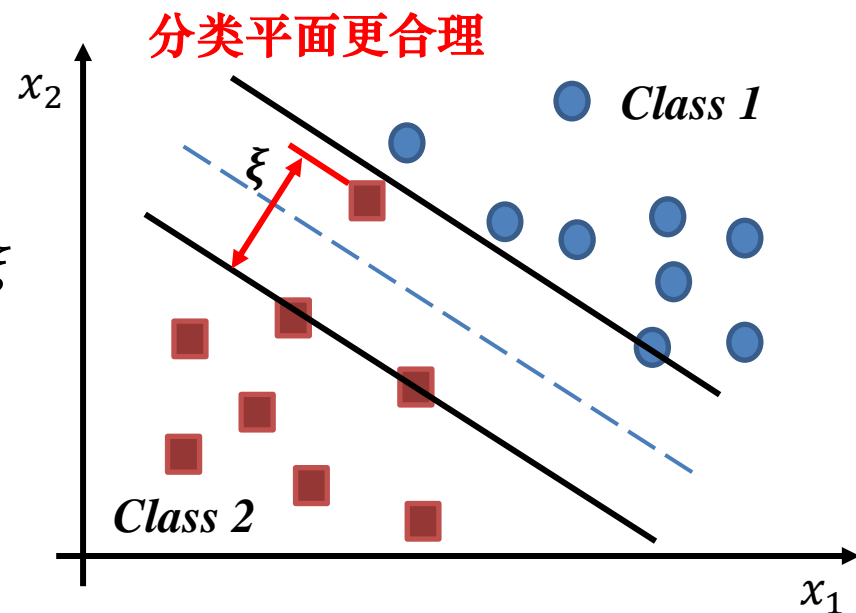
● 标准SVM

- 只考虑最优分类面可能导致**泛化性能差**，标准SVM引入松弛变量 ξ 处理噪声和离群点，增强模型的泛化能力



无错分但分界区域较小的硬间隔分类面

引入松弛变量 ξ



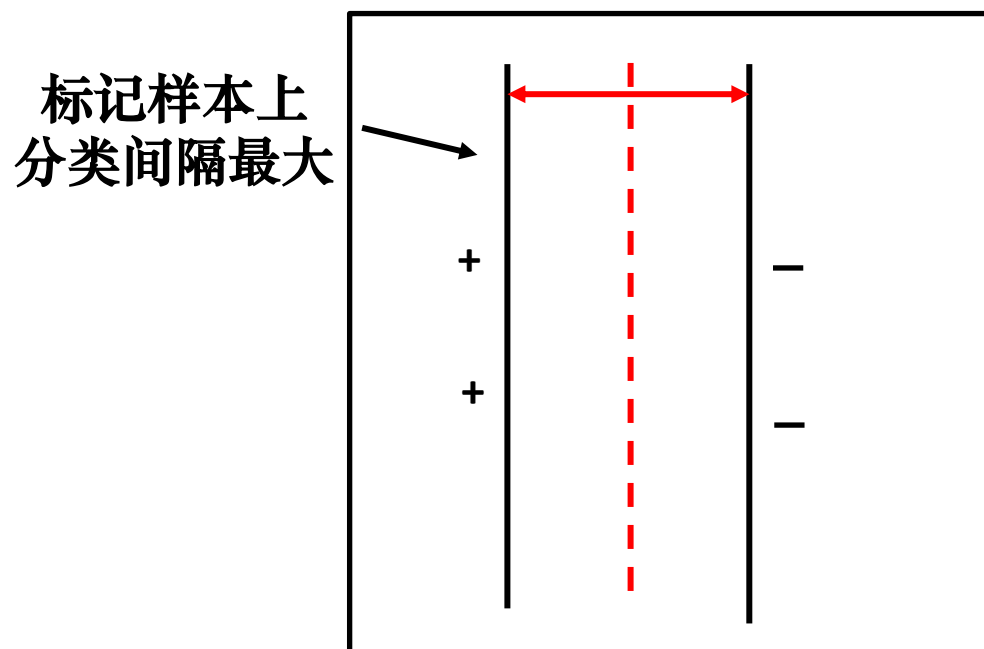
分类平面更合理

有错分但分界区域较大的广义最优分类面

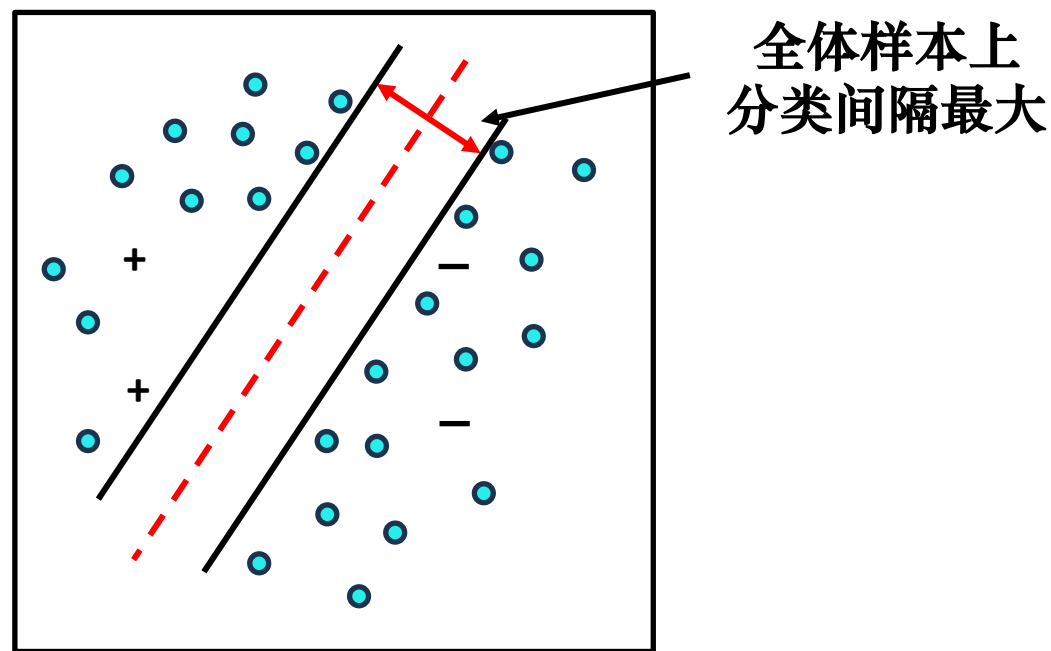
直推式支持向量机

● 标准SVM→半监督SVM(Semi-Supervised SVM)

- 相较于标准SVM，半监督SVM考虑未标记样本，要求划分超平面不仅能分隔标记样本，还力求在全体样本 $(D_l \cup D_u)$ 上使分类间隔最大



标准SVM只考虑
标记样本



无监督SVM考虑所有标记和
未标记样本

直推式支持向量机

- 半监督支持向量机中最著名的是直推式支持向量机
- 直推式支持向量机T-SVM的形式化定义

未标记样本
松弛变量

$$\min_{w, b, \hat{y}, \xi} \frac{1}{2} \|w\|_2^2 + C_l \sum_{i=1}^l \xi_i + C_u \sum_{i=l+1}^m \xi_i \quad (16-3-1)$$

分类间隔最大

$$s. t. \quad y_i(w^T x_i + b) \geq 1 - \xi_i, \quad i = 1, \dots, l,$$

$$\hat{y}_i(w^T x_i + b) \geq 1 - \xi_i, \quad i = l + 1, \dots, m,$$

未标记样本分类

未标记样本预测值

$$\xi_i \geq 0, \quad i = 1, \dots, m,$$

C_l 和 C_u 分别表示标记样本和未标记样本的**惩罚因子**，用于调整不同样本的权重， ξ 为**松弛变量**，用于调整对错分样本的容忍程度

直推式支持向量机

● 直推式支持向量机 (T-SVM)

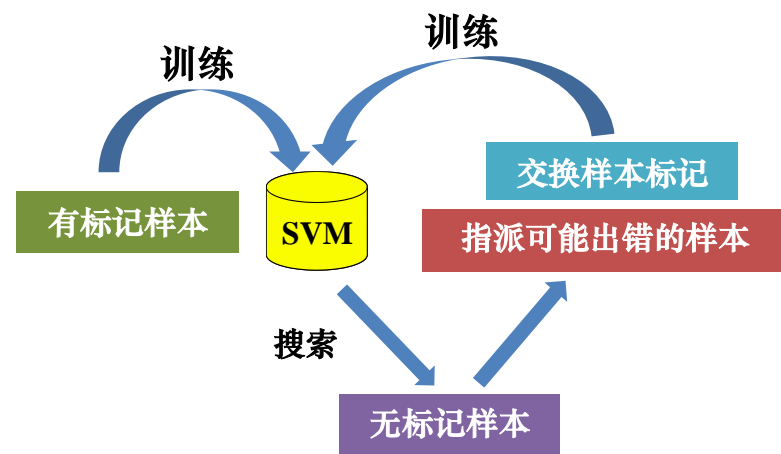
- 基本思想：对未标记样本的类别 \hat{y} 进行各种可能的指派，寻求一个在所有样本上间隔最大化的划分超平面
- 对未标记样本的各种标记指派是一个穷举遍历过程，未标记样本数太大时不能直接求解
- 采用局部搜索迭代求近似解，通过局部搜索和调整指派为异类且可能错误的标记指派，使目标函数值不断下降

步骤1：局部搜索 x_i, x_j

步骤2：判断 x_i, x_j 是否类别不同，即 $y_i y_j < 0$

步骤3：判断 x_i, x_j 是否发生指派错误，即 $\xi_i + \xi_j < 0$

步骤4：若步骤2、3判断为真，互换 x_i, x_j 标记



直推式支持向量机

● 直推式支持向量机的详细步骤

输入：有标记样本集 $D_l = \{(x_1, y_1), \dots, (x_l, y_l)\}$

未标记样本集 $D_u = \{x_{l+1}, x_{l+2}, \dots, x_{l+u}\}$

惩罚因子 C_l, C_u

过程：

1: 用 D_l 训练一个 SVM_1 ;

2: 用 SVM_1 对 D_u 中样本进行预测，得到 $\hat{y} = (\hat{y}_{l+1}, \hat{y}_{l+2}, \dots, \hat{y}_{l+u})$;

3: 初始化 $C_u \ll C_l$;

4: while $C_u < C_l$ do

5: 基于 $D_l, D_u, \hat{y}, C_l, C_u$ 求解式 (16-3-1)，得到 $(w, b), \xi$;

6: while $\exists \{i, j \mid (\hat{y}_i \hat{y}_j < 0) \wedge (\xi_i > 0) \wedge (\xi_j > 0) \wedge (\xi_i + \xi_j > 2)\}$ do

7: $\hat{y}_i = -\hat{y}_i$;

8: $\hat{y}_j = -\hat{y}_j$;

9: 基于 $D_l, D_u, \hat{y}, C_l, C_u$ 重新求解式(1)，得到 $(\omega, b), \xi$;

10: end while

11: $C_u = \min\{2C_u, C_l\}$

12: end while

输出：未标记样本的预测结果: $\hat{y} = (\hat{y}_{l+1}, \hat{y}_{l+2}, \dots, \hat{y}_{l+u})$;

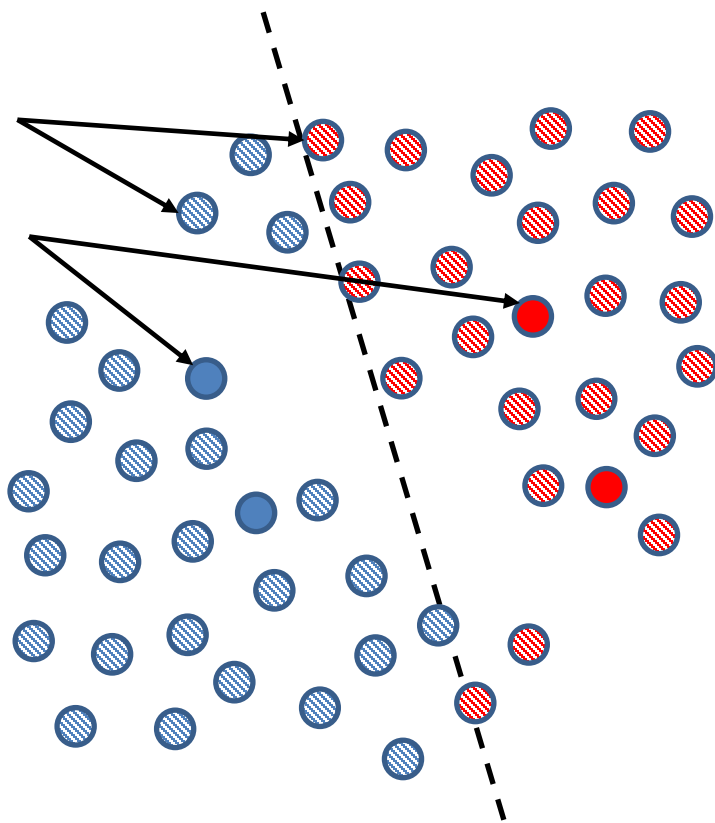
未标记样本的伪标记不准确

逐渐增大 C_u 提高未标记样本贡献

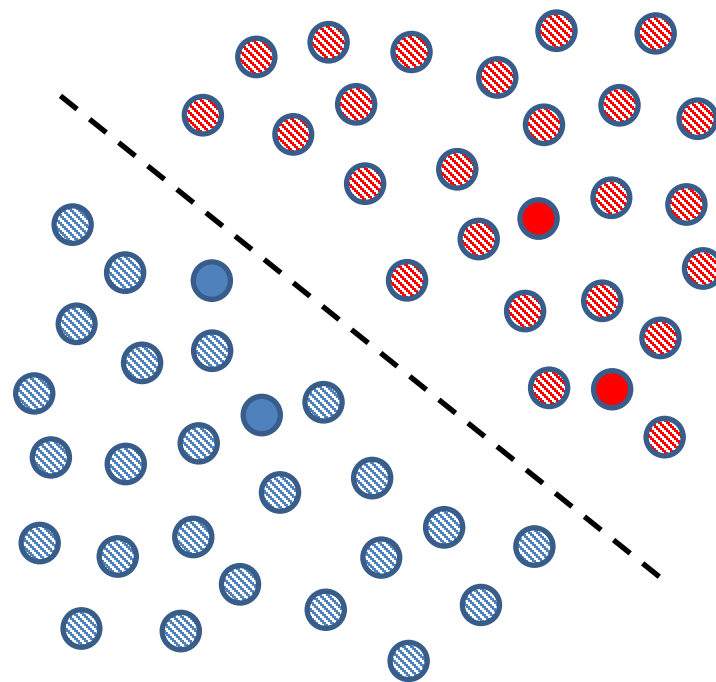
T-SVM和SVM算法对比示例

未标记数据

标记数据



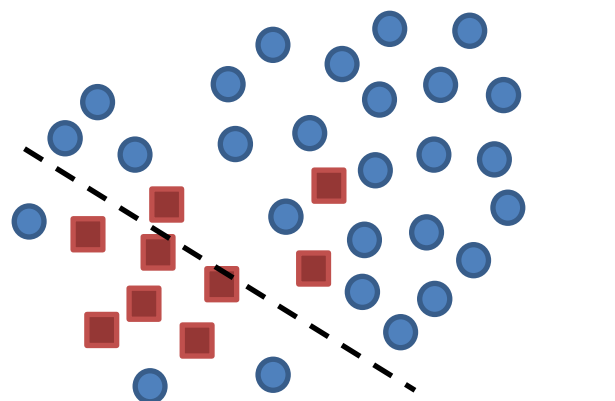
SVM划分超平面



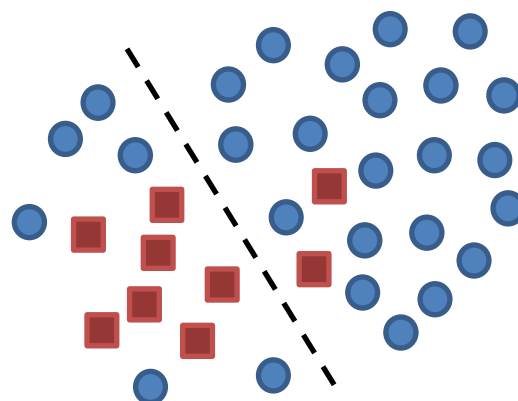
T-SVM划分超平面

直推式支持向量机的改进

- 存在的问题一：无标记样本进行标记指派及调整的过程中，有可能出现**类别不平衡**问题，即某类的样本远多于另一类
- 解决方法：设 u_+ 和 u_- 分别为未标记样本预测为正例和负例的样本数。为了减轻类别不平衡的影响，可将优化目标中的 C_u 拆分为 C_u^+ 与 C_u^- 两项，并在初始化时令 $C_u^+ = (u_-/u_+)C_u^-$



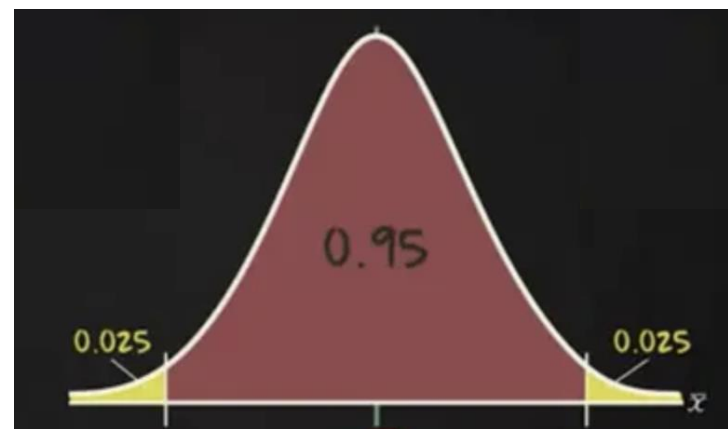
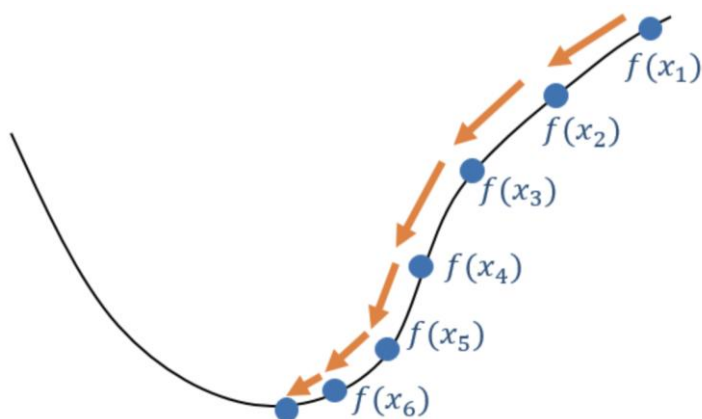
数据不平衡时，SVM 的分类超平面会靠近少数类样本



对惩罚因子加权后，SVM 的分类超平面更合理

直推式支持向量机的改进

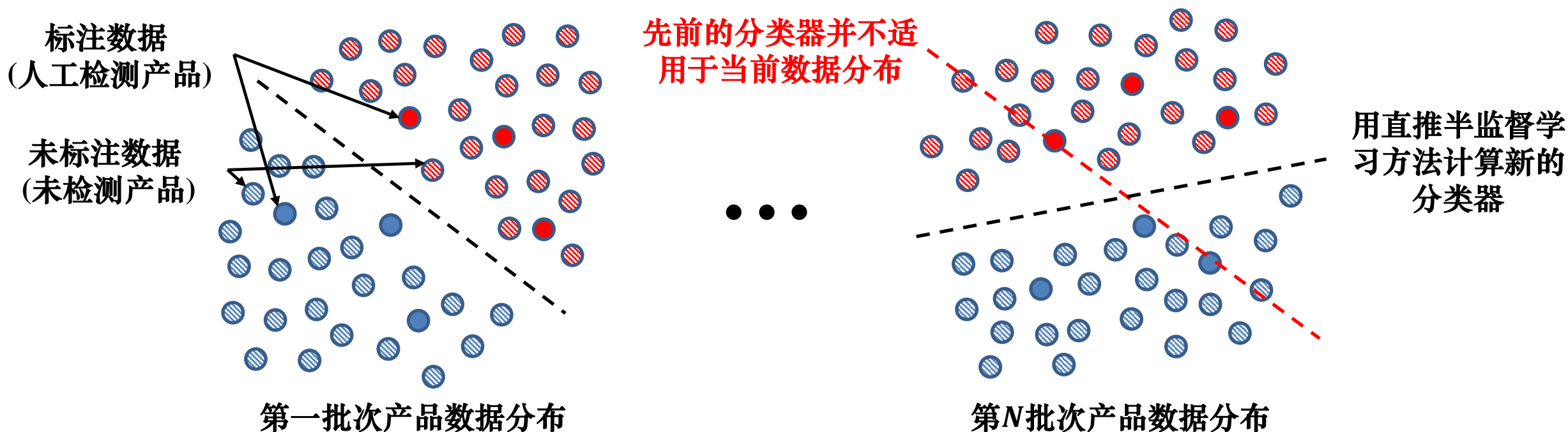
- 存在的问题二：对每一对标记错误的未标记样本进行调整复杂度高
- 解决方法：高效优化求解算法
 - 基于图和梯度下降的LDS算法 [O. Chapelle et al., 2005]
 - 基于标记均值估计的MeanS3VM [Y. Li et al., 2009]



直推半监督学习的应用

● 工业残次品检测

- 问题：每批次产品的**数据分布有所不同**，通用模型难以在所有批次上准确预测
- 直推半监督学习：每个批次上标注少量样本并重新训练，取得更高精度的预测



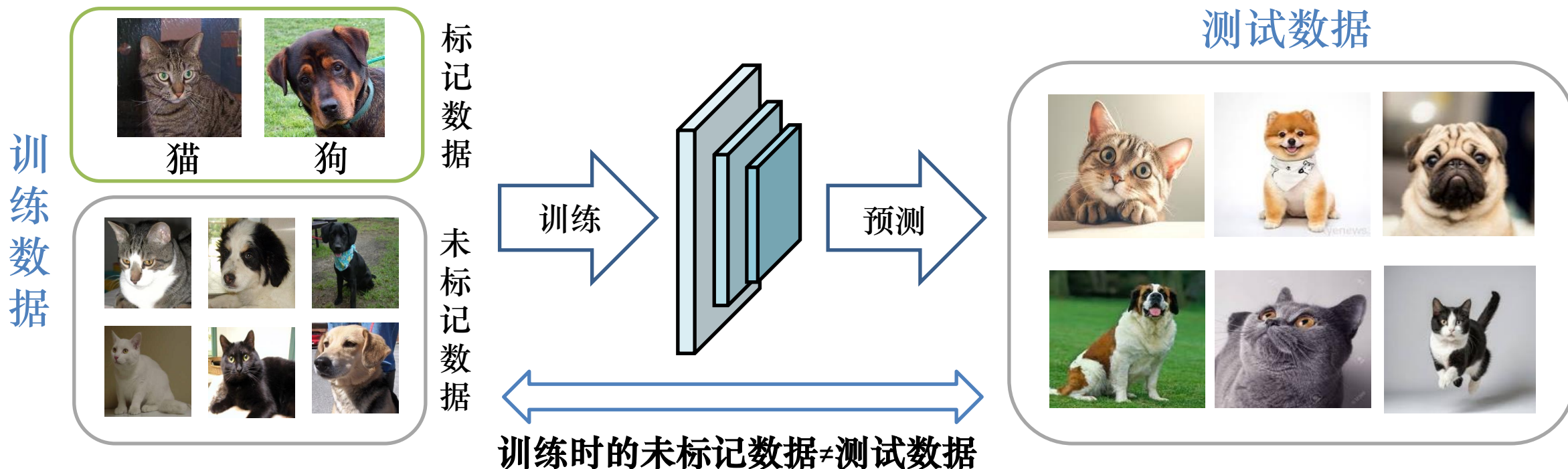
15.4 归纳半监督学习

- 归纳半监督学习的定义
- 协同学习算法

归纳半监督学习

● 归纳半监督学习

- 归纳半监督学习基于“开放世界”的假设，构建能够推广到未观察到的新数据上的模型。



归纳半监督学习的形式化描述

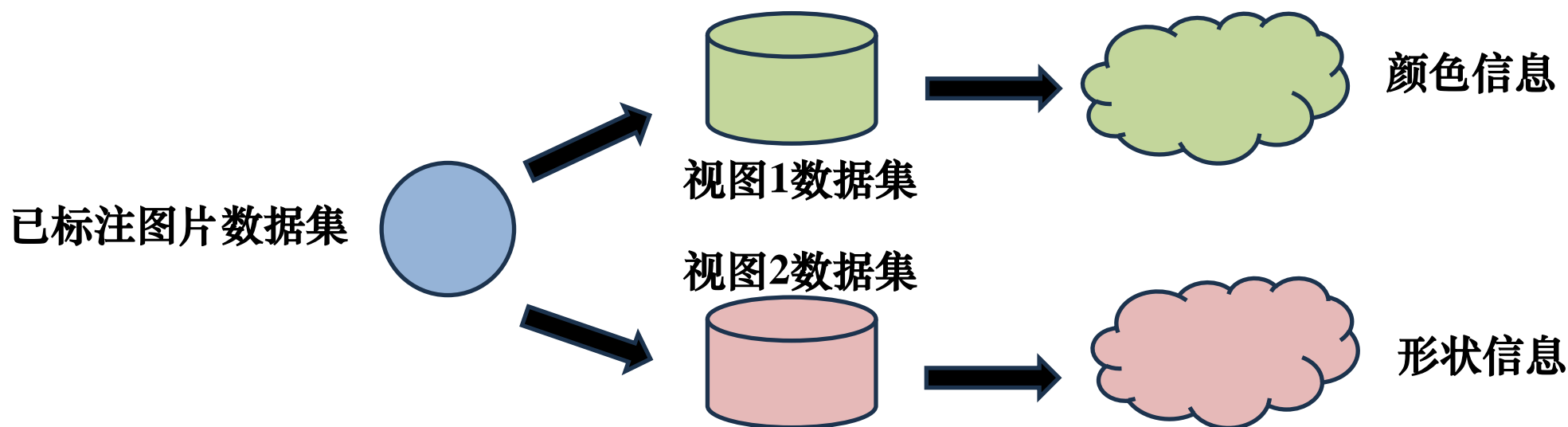
- 给定一个来自某未知分布的样本集 $S = D_l \cup D_u$ 。 D_l 是**已标记样本集** $D_l = \{(x_1, y_1), \dots, (x_l, y_l)\}$ ， D_u 是**未标记样本集** $D_u = \{x_{l+1}, \dots, x_{l+u}\}$ 。
其中, x 为 d 维向量, $y_i \in Y$ 为 D_l 中样本 x_i 的标记
- 归纳半监督学习就是在样本集 $S = D_l \cup D_u$ 上学习一个**推广的预测函数** $f: X \rightarrow Y$, 这个函数可以应用于任何**未见过的数据** x
- 经典的归纳半监督学习方法有：协同学习、图半监督学习等

协同学习

● 协同学习的基本概念

【1998年A. Blum和T. Mitchell提出】

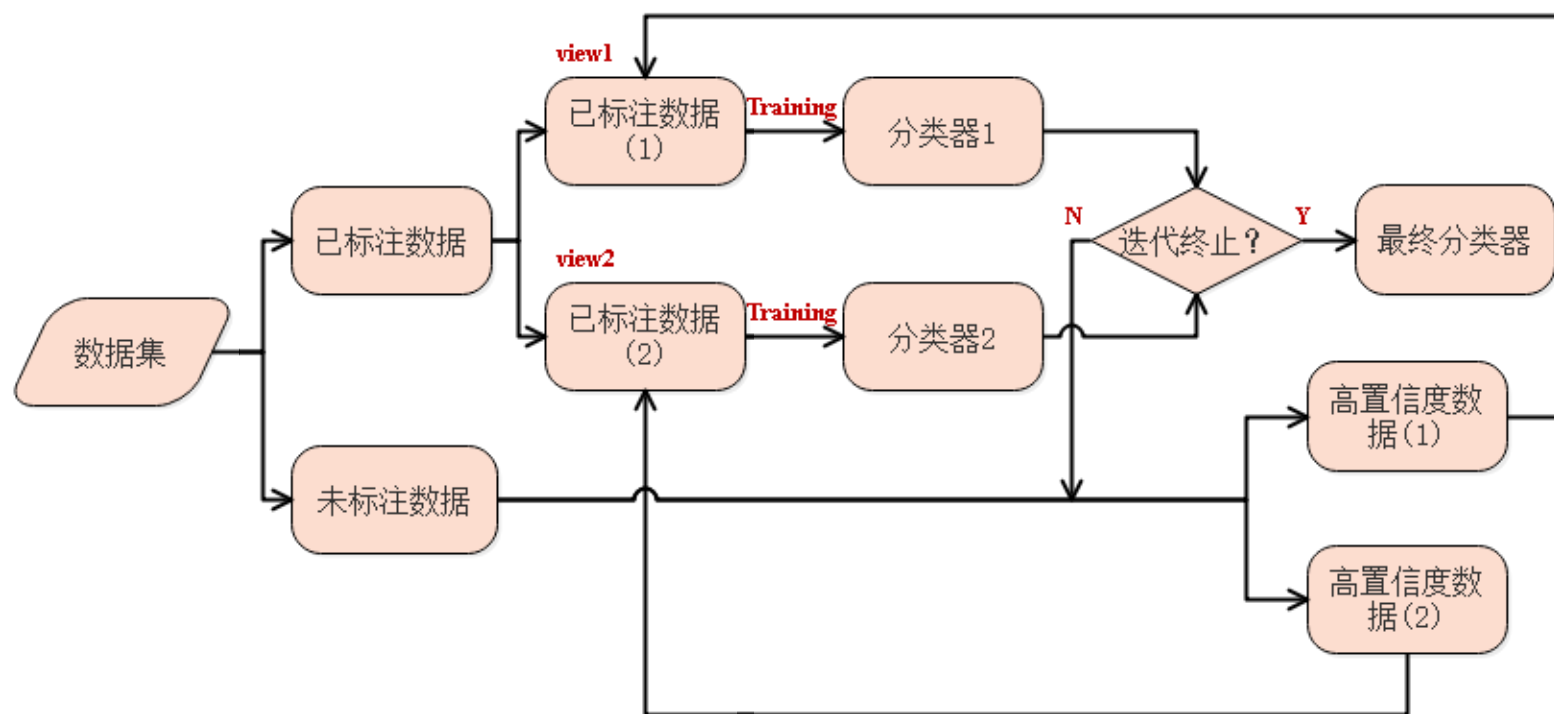
- 协同学习（Co-Training）是自学习的一种改进方法，通过两个基于不同视图（view）的分类器来互相促进
- 基于不同视图的条件独立性假设，在不同视图的数据集上训练出来的模型就相当于从不同视图来理解问题，具有一定的互补性，协同训练利用这种互补性来进行学习



协同学习

● 协同学习的算法流程

【1998年A. Blum和T. Mitchell提出】

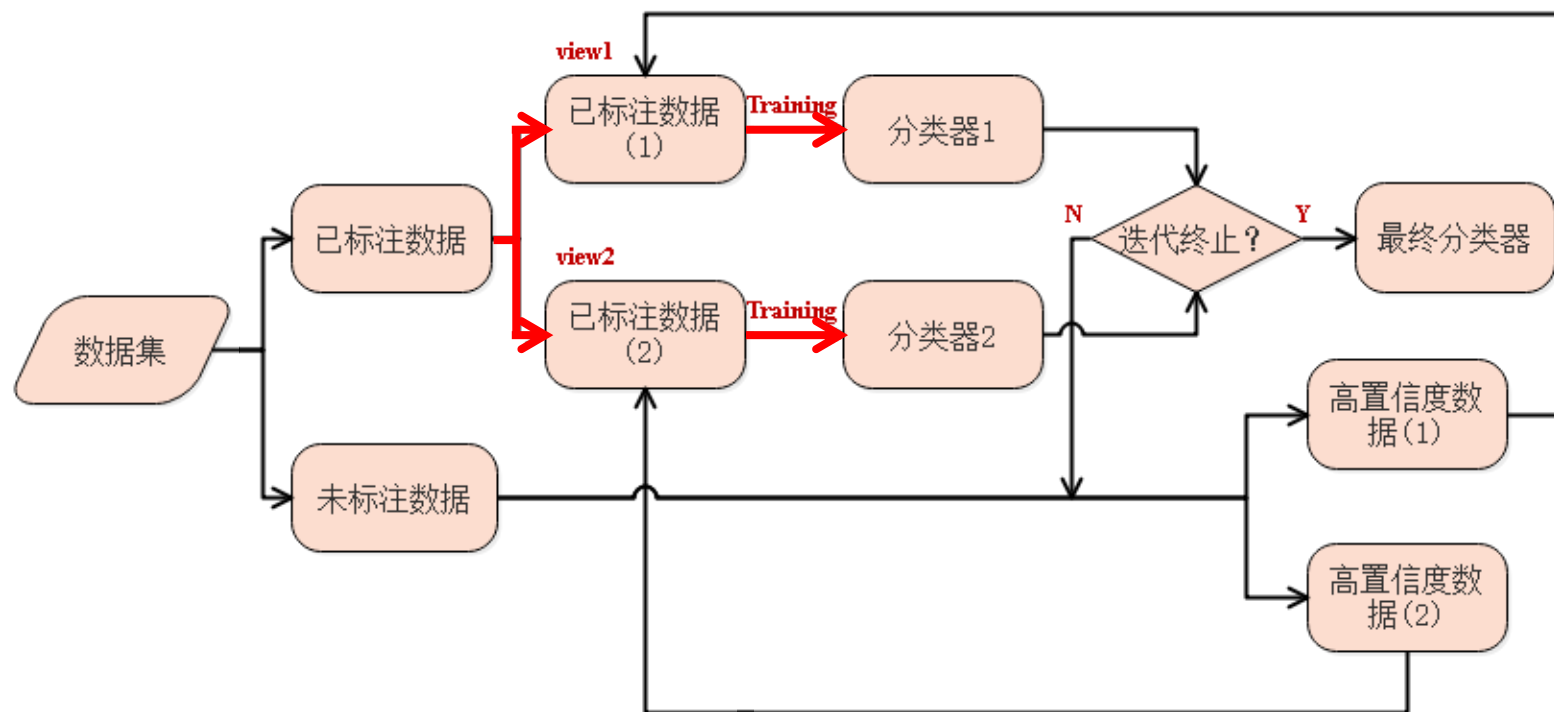


- 协同学习的的算法流程分为以下几个部分：初始阶段、迭代训练、模型更新、终止阶段

协同学习

● 协同学习的算法流程

【1998年A. Blum和T. Mitchell提出】

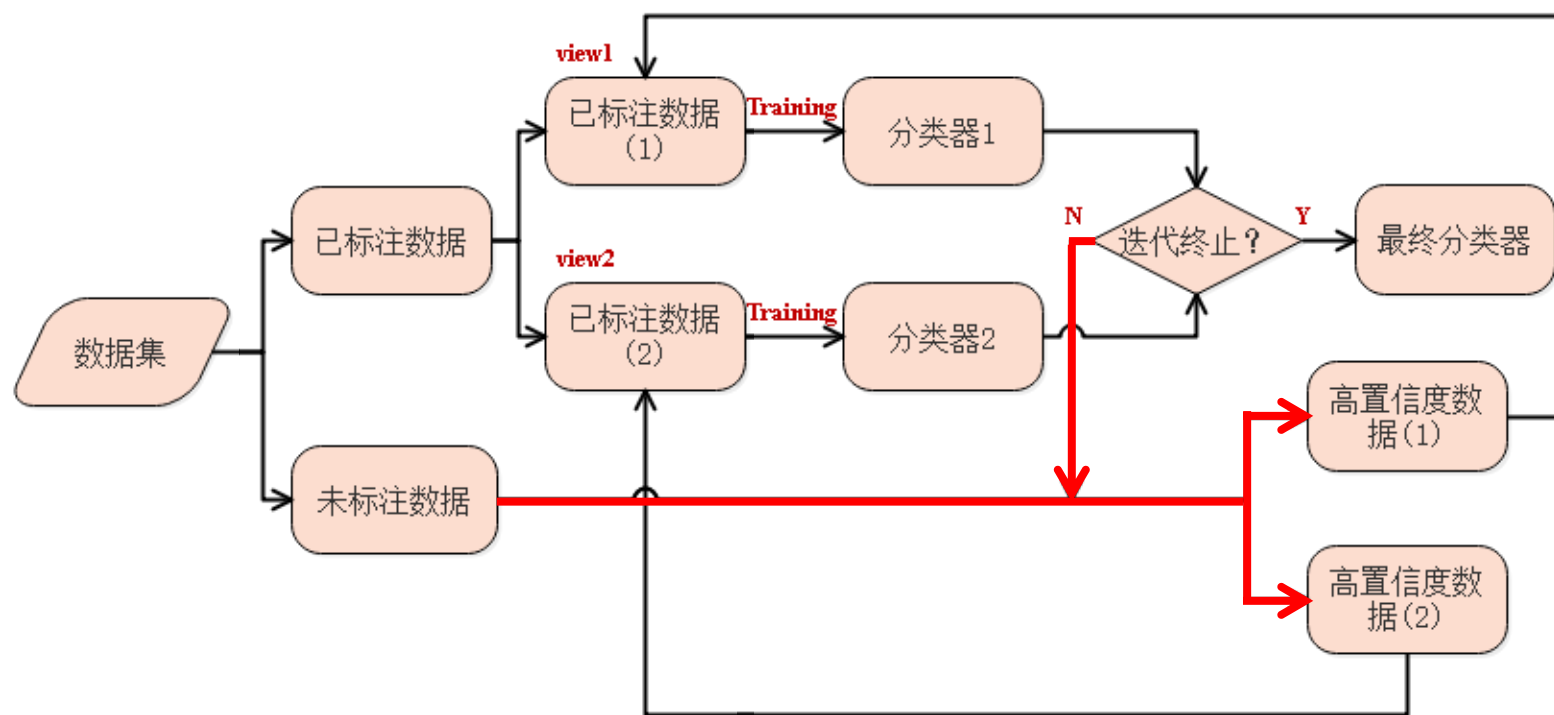


- 初始阶段：将有标签数据集 D_l 随机分成两个子集，分别视图1和视图2。分别训练分类器1和分类器2

协同学习

● 协同学习的算法流程

【1998年A. Blum和T. Mitchell提出】

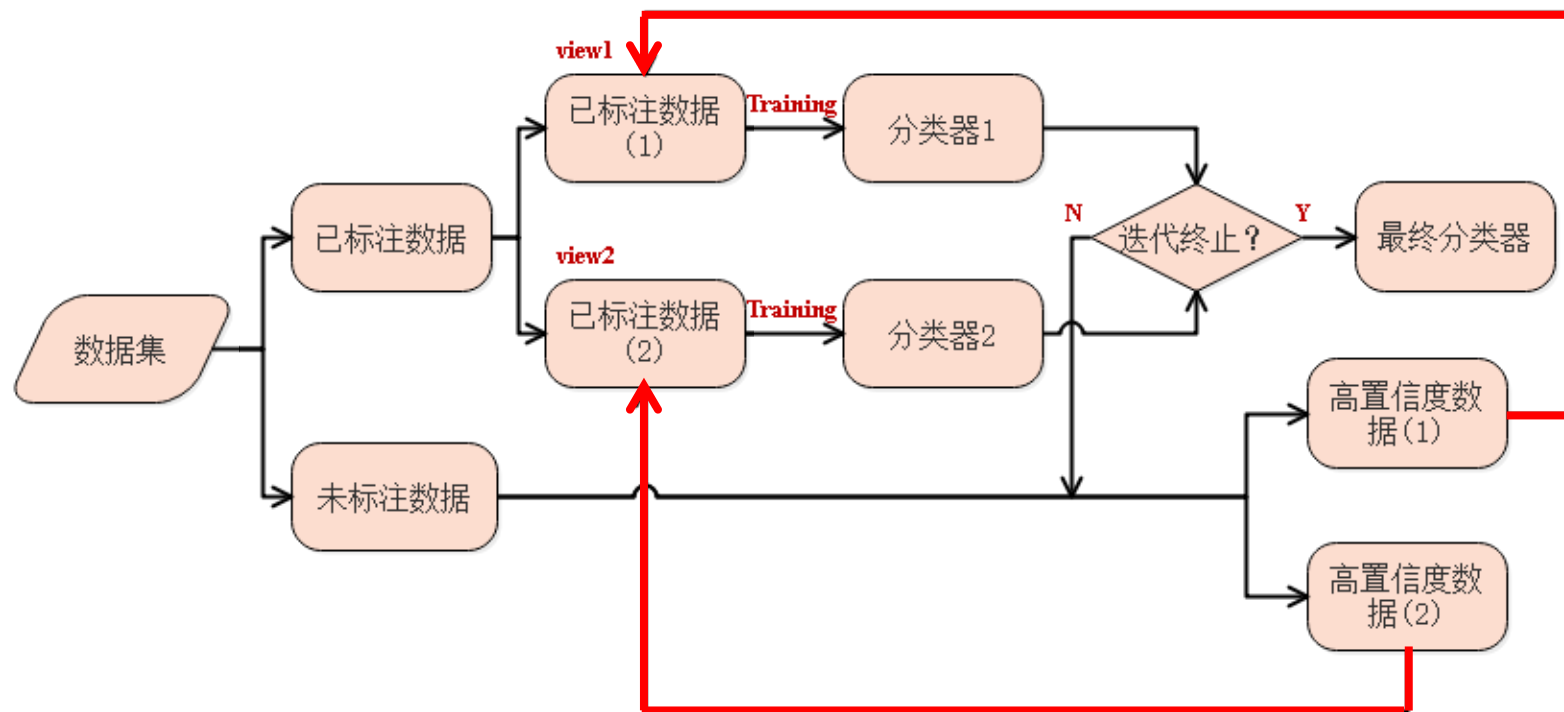


- 迭代训练：在每次迭代中，综合分类器1和分类器2对无标签数据 D_u 进行预测，并选择置信度较高的样本加入到相应视图的有标签数据集 D_l 中

协同学习

● 协同学习的算法流程

【1998年A. Blum和T. Mitchell提出】

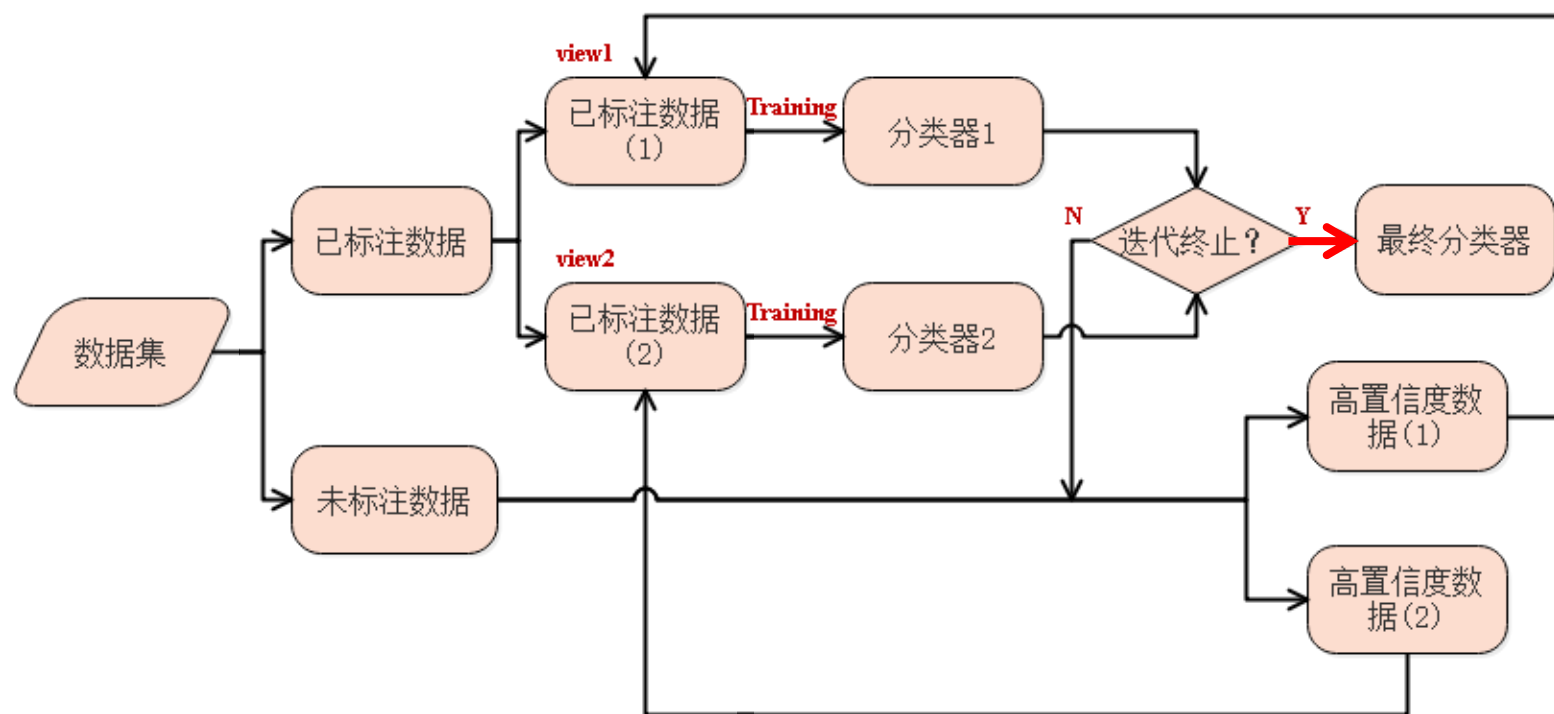


➤ 模型更新：使用**扩充后的有标签数据集** D_l 训练模型1和模型2

协同学习

● 协同学习的算法流程

【1998年A. Blum和T. Mitchell提出】



- 终止阶段：重复步骤2和步骤3，直到满足停止条件（如达到最大迭代次数或模型性能不再提升）

归纳半监督学习

● 归纳半监督学习 vs. 直推半监督学习

➤ 直推半监督学习 (Transductive Semi-Supervised Learning)

- 仅为训练集中的未标记数据生成标签

- 学习到的模型不是预测模型

- 关注如何有效地利用有限的有标签和无标签数据

➤ 归纳半监督学习 (Inductive Semi-Supervised Learning)

- 不仅为训练集中的未标记数据生成标签，还能够为**新数据**生成标签

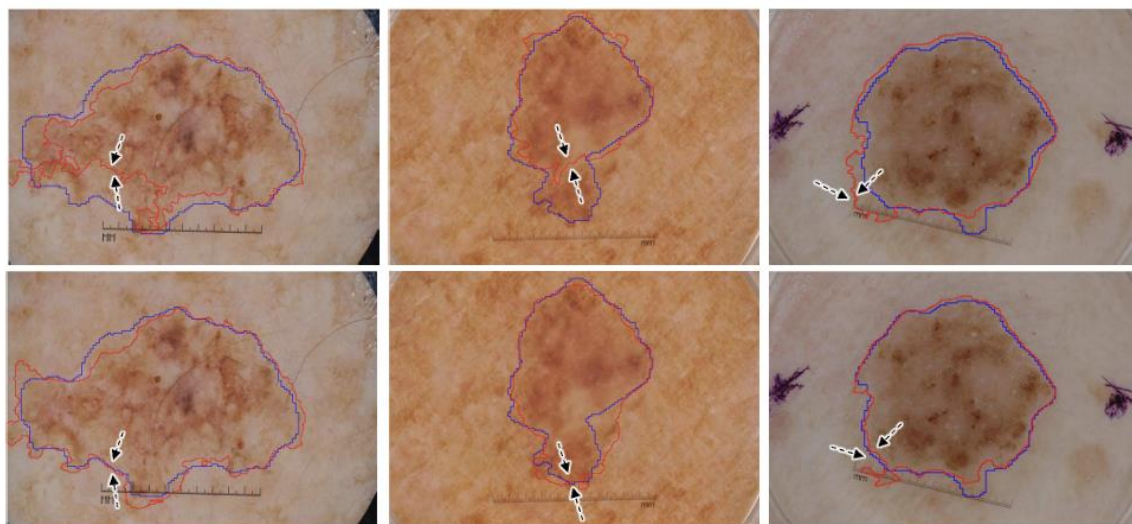
- 学习到的模型是预测模型

- 需要考虑模型的**泛化能力**

归纳半监督学习的应用

● 医学领域的辅助疾病诊断

- 皮肤镜图像上的自动皮肤病变分割是**计算机辅助诊断黑色素瘤**的重要组成部分
- 基于监督学习的方法需要**大量标注数据**，标注过程**非常耗时且成本高昂**
- 归纳半监督学习仅通过少量标注数据引导模型学习，利用**未标注数据**增强模型的学习效果



仅使用监督学习



使用归纳半监督学习

15.5 图半监督学习

- 图半监督学习的定义
- 二分类标记传播算法
- 多分类标记传播算法

图半监督学习

● 基本概念

- 图半监督学习是一类**针对图结构数据**的半监督学习方法
- 图半监督学习适合解决那些数据点间存在**复杂关系网络**的问题，如社交网络分析、推荐系统等问题，通过利用**少量标记数据**和**大量未标记数据**的**图结构信息**，有效提升对未标记样本分类的准确性



社交网络一类的数据**天然具有图的结构**：
社交网络可以被表示为图，其中**节点代表用户**，**边代表用户之间的关系**（如朋友关系、关注关系等）

图半监督学习

● 图半监督学习形式化

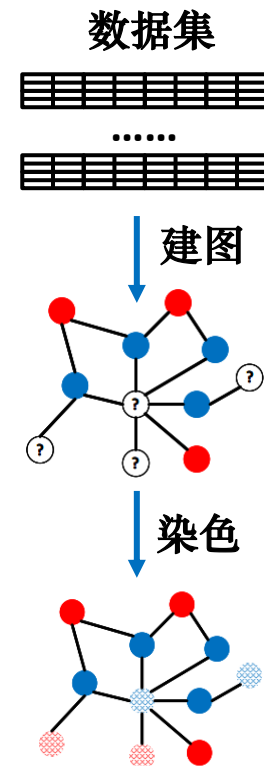
➤ 首先，需要将数据集构建为图

- 带标记的数据集表示为 D_l ，无标记的数据集为 D_u
- 基于 $D_l \cup D_u$ 构建一个图 $G = (V, E)$ ，其中结点集合 $V = \{x_1, \dots, x_l, x_{l+1}, \dots, x_{l+u}\}$
- 边集合 E 可表示为一个相似性矩阵(Affinity Matrix)可定义为

$$W_{ij} = \begin{cases} \exp\left(\frac{-\|x_i - x_j\|_2^2}{2\sigma^2}\right), & \text{if } i \neq j \\ 0, & \text{else} \end{cases}$$

相似性矩阵 W 是基于高斯函数定义。可以看出，**两个样本越近，相似度越大**

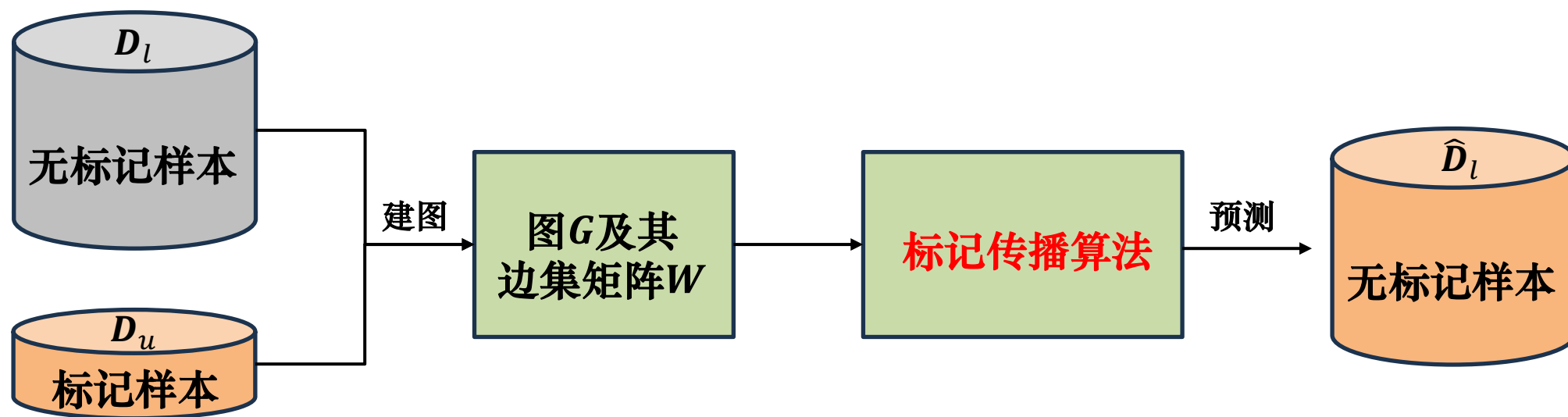
- 假设标记样本 $x \in D_l$ 所对应的结点为染过色，无标记样本 $x \in D_u$ 所对应的结点未染色，半监督学习就对应于“颜色”在图 G 上**扩散**或**传播**的过程
- 图 G 可以表示成一个**矩阵形式 W** ，可以使用矩阵运算来进行半监督学习算法的**推导与分析**



图半监督学习

● 图半监督学习核心问题

➤ 在构建的图中，如何使用少量标记样本信息预测无标记样本



➤ 代表性方法：二分类标记传播算法、多分类标记传播算法

二分类标记传播

● 二分类标记传播的形式化

➤ 目标：求解出无标记样本的类别指派函数 f_u 与标记样本的类别指派函数 f_l 的关系

➤ 基于图的类别指派函数即是**输入为样本输出为样本标记**的函数

实值函数 $f: V \rightarrow \mathbb{R}$ ，其对应的分类规则为： $y_i = \text{sign}(f(x_i))$, $y_i \in \{-1, +1\}$

➤ 核心思想：**相似的样本指派相似的标记**，具体可以得到关于 f 的“能量函数” (Energy Function)

$$E(f) = \frac{1}{2} \sum_{i=1}^m \sum_{j=1}^m w_{ij} (f(x_i) - f(x_j))^2$$

直接求导复杂，需要先化简

最小化能量函数，使 x_i 与 x_j 越接近时， $f(x_i)$ 与 $f(x_j)$ 也越接近

二分类标记传播

● 能量函数化简

$$\begin{aligned} E(f) &= \frac{1}{2} \sum_{i=1}^m \sum_{j=1}^m W_{ij} (f(x_i) - f(x_j))^2 \\ &= \frac{1}{2} \left(\sum_{i=1}^m \sum_{j=1}^m W_{ij} f^2(x_i) + \sum_{i=1}^m \sum_{j=1}^m W_{ij} f^2(x_j) - 2 \sum_{i=1}^m \sum_{j=1}^m W_{ij} f(x_i) f(x_j) \right) \\ &= \sum_{i=1}^m \sum_{j=1}^m W_{ij} f^2(x_i) - \sum_{i=1}^m \sum_{j=1}^m W_{ij} f(x_i) f(x_j) \\ &= \sum_{i=1}^m \sum_{j=1}^m d_i f^2(x_i) - \sum_{i=1}^m \sum_{j=1}^m W_{ij} f(x_i) f(x_j) \\ &= \mathbf{f}^T (\mathbf{D} - \mathbf{W}) \mathbf{f} \end{aligned}$$

展开

W 为对称矩阵

$d_i = \sum_{j=1}^{l+u} W_{ij}$

需要分离 f_u 和 f_l 以求解 f_u

■ 其中 $\mathbf{f} = (f_l^T; f_u^T)$ ；有标记样本的预测结果 $f_l = (f(x_1); f(x_2); \cdots; f(x_l))$

■ 无标记样本的预测结果 $f_u = (f(x_{l+1}); f(x_{l+2}); \cdots; f(x_{l+u}))$ ； $D = \text{diag}(d_1, d_2, \cdots, d_{l+u})$

二分类标记传播

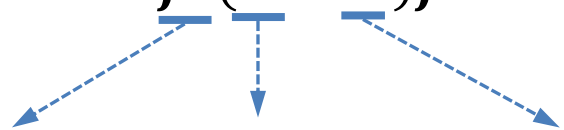
● 类别指派函数 f_u 的求解

➤ 求解“能量函数”最小时， f_l 与 f_u 的关系

➤ 将变量 f_l 与 f_u 进行分离

■ W 可用分块矩阵表示 $W = \begin{bmatrix} W_{ll} & W_{lu} \\ W_{ul} & W_{uu} \end{bmatrix}$

■ 能量函数表示为

$$f^T(D - W)f$$

$$E(f) = (f_l^T f_u^T) \left(\begin{bmatrix} D_{ll} & \mathbf{0}_{lu} \\ \mathbf{0}_{ul} & D_{uu} \end{bmatrix} - \begin{bmatrix} W_{ll} & W_{lu} \\ W_{ul} & W_{uu} \end{bmatrix} \right) \begin{bmatrix} f_l \\ f_u \end{bmatrix}$$

f_l 已知， f_u 未知，需要继续化简分离，以获得 f_l 和 f_u 的关系

二分类标记传播

- 类别指派函数 f_u 的求解

- 将该上述公式化简可得

$$E(f) = f_l^T (D_{ll} - W_{ll}) f_l - 2f_u^T W_{ul} f_l + f_u^T (D_{uu} - W_{uu}) f_u$$

- 令偏导数 $\frac{\partial E(f)}{\partial f_u} = 0$ 可得

$$2(D_{ll} - W_{ll})f_u - 2W_{ul}f_l = 0$$

化简

$$f_u = (D_{ll} - W_{ll})^{-1} W_{ul} f_l$$

- 将已知 f_l 带入这个公式即可，求得 f_u

多分类标记传播

● 多分类标记传播的形式化

- 二分类 “**单步式**” 标记传播拓展到多分类 “**迭代式**” 标记传播
- 多分类标记传播基本思想: **使用相似性矩阵 W 扩散或传播标记的过程**
- 定义一个 $(l + u) \times |\mathcal{Y}|$ 的非负标记矩阵 $F = (F_1^T, \dots, F_{|\mathcal{Y}|}^T)$, $|\mathcal{Y}|$ 为类别数量
 - 其第 i 行元素 $F_i = (F_{i1}, \dots, F_{i|\mathcal{Y}|})$ 为样本 x_i 的标记向量, 相应的分类规则为

$$y_i = \operatorname{argmax}_{1 \leq j \leq |\mathcal{Y}|} F_{ij}$$

$F(0)$ 标记矩阵 **标记样本的标记向量**

■ F 的初始化: $F(0) = \begin{cases} 1 & , \text{if } (1 \leq i \leq l) \wedge (y_i = j) \\ 0, & \text{else} \end{cases}$

	类别1	类别2	类别3	……	类别 $ \mathcal{Y} $
x_1	0	0	1		0
x_2	1	0	0		0
...
x_l	0	0	1		0
x_{l+1}	0	0	0		0
...
x_{l+u}	0	0	0		0

■ 定义矩阵 $Y = F(0)$

无标记样本的标记向量

多分类标记传播

● 多分类标记传播算法的流程

➤ 步骤1: 基于 W 构造一个标记传播矩阵 $S = D^{-\frac{1}{2}}W D^{-\frac{1}{2}}$, 其中

$$D^{-\frac{1}{2}} = \left(\frac{1}{\sqrt{d_1}}, \dots, \frac{1}{\sqrt{d_{l+u}}} \right) \quad \text{归一化}W$$

➤ 步骤2: “迭代式” 标记传播方法

核心迭代式

$$F(t+1) = \alpha S F(t) + (1-\alpha)Y$$

$$F^* = \alpha S F^* + (1-\alpha)Y$$

$$F^* = \lim_{t \rightarrow \infty} (1-\alpha)(I - \alpha S)^{-1}Y$$

$$\lim_{t \rightarrow \infty} F(t+1) = \lim_{t \rightarrow \infty} F(t) = F^*$$

多分类标记传播

● 多分类标记传播算法

输入:

有标记样本集 $D_l = \{(x_1, y_1), \dots, (x_l, y_l)\}$

未标记样本集 $D_u = \{x_{l+1}, x_{l+2}, \dots, x_{l+u}\}$

构图参数 σ 折中参数 α 亲和矩阵 W

过程:

1: $S = D^{-\frac{1}{2}} W D^{-\frac{1}{2}}$

2: 初始化 $F(0)$

3: $t = 0$

4: *repeat*

核心迭代式

5: $F(t+1) = \alpha S F(t) + (1 - \alpha) Y$

6: $t = t + 1$

7: *until* 迭代收敛至 F^*

8: *for* $i = l+1, l+2, \dots, l+u$ *do*

9: $y_i = \operatorname{argmax}_{1 \leq j \leq y} (F^*)_{ij}$

10: *end for*

输出:

未标记样本集的预测结果:

$\hat{y} = (\hat{y}_{l+1}, \hat{y}_{l+2}, \dots, \hat{y}_{l+u})$

多分类标记传播

● 多分类传播方法算法等价性分析

➤ “迭代式”传播方法算法是下列正则化框架的解

相似的样本指派相似的标记

有标记样本的标记逼近真实值

$$\min_F \frac{1}{2} \left(\sum_{i,j=1}^{l+u} W_{ij} \left\| \frac{1}{\sqrt{d_i}} F_i - \frac{1}{\sqrt{d_j}} F_j \right\|^2 \right) + \mu \sum_{i=1}^{l+u} \|F_i - Y_i\|^2$$

➤ 当 $\mu = \frac{1-\alpha}{\alpha}$ 时，最优解恰为迭代算法的收敛解 F^*

小结

● 标记传播算法

➤ 优势：

- **算法性质清晰**：标记传播方法在概念上相当清晰，易于理解
- **易于分析**：通过对所涉及的矩阵运算的分析，可以探索算法的性质

➤ 缺点：

- **存储开销大**：如果样本数为 $O(m)$ ，则算法中所涉及的矩阵规模为 $O(m^2)$ ，这使得算法难以处理大规模数据
- **难以适应新样本**：由于构图过程仅能考虑训练样本集，难以预测新样本在图中的位置。在接收到新样本时，需要重新进行标记传播或者引入额外的预测机制

15.6 半监督聚类

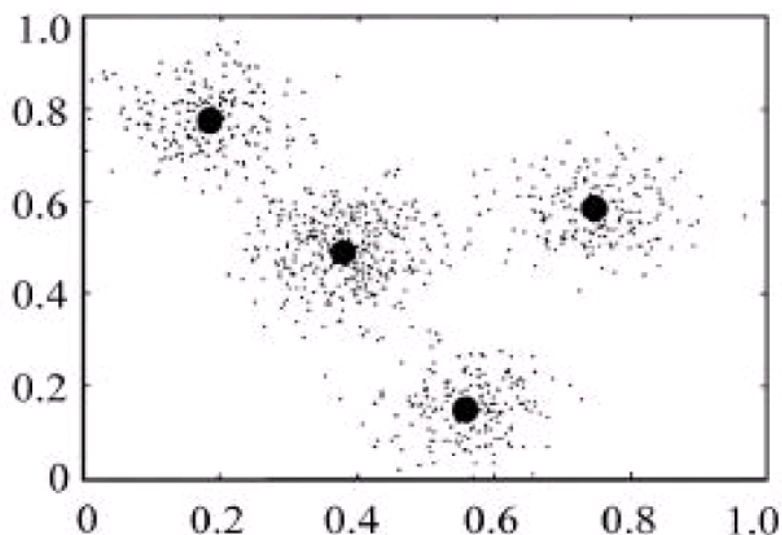
- 半监督聚类的定义
- 约束K均值算法
- 约束种子K均值算法
- 半监督学习方法讨论

聚类

● 聚类回顾

➤ 以K均值算法为例

- 核心思想是将无标记样本划分为K个簇，使得每个样本与其所属簇的中心的距离之和最小
- 形式化：给定D维空间的数据集 $\{x_1, x_2, \dots, x_N\}$ ，不知道这些样本所对应标签，通过聚类方法将这些数据集划分成K类
- 对于K个聚类中的每一类，分别建立一个代表点 μ_k ，将每一个样本划归到离该样本最近的 μ_k 所代表的聚类



聚类

- 聚类的问题

- 数据方面

- **数据利用不足**，无法利用数据中的少量标记信息

- 性能方面

- 性能不如监督学习，**准确率较低**

- 训练速度

- 作为数据处理的一步，**需要更高的效率**

需要一种能利用额外数据信息提升性能和效率的聚类方法

半监督聚类

● 半监督聚类定义

➤ 利用少量标记信息来提升聚类的准确性和效率

➤ 少量标记信息

■ 第一种类型是“必连” (Must-Link) 与 “勿连” (Cannot-Link) 约束

“必连”：样本**必属于**同一个簇

“勿连”：样本**必不属于**同一个簇

■ 第二种类型的标记信息则是少量的有标记样本

➤ 代表性算法：约束K均值 (Constrained K-means) 算法、约束种子K均值 (Constrained K-means) 算法

半监督聚类

● 约束K均值算法

【2001年K. Wagstaff等人提出】

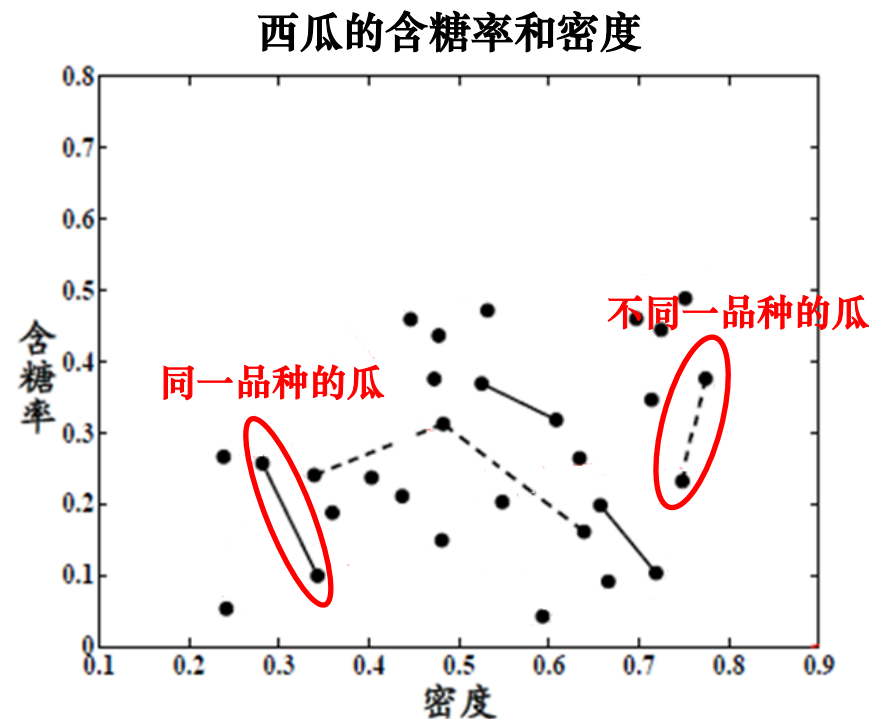
➤ 该算法是K均值算法的扩展

■ 在聚类过程中要检查每个样本在“必连”关系集合与“勿连”关系集合中的约束是否得以满足，否则将重新对该样本聚类

■ 一个例子：西瓜是否成熟

假设有一个包含大量西瓜信息的数据集，目标是根据西瓜的成熟度将它们分类，以便可以决定何时收获。现在已知的信息如下

“必连”：同一品种的瓜； “勿连”：不同品种的瓜



半监督聚类

● 约束K均值算法：基本思路

- 步骤1. 初始化：选择K个初始聚类中心
- 步骤2. 分配：将每个样本分配给最近的聚类中心 μ_k
- 步骤3. 检查：对有约束信息的样本，如果将其分配到最近的聚类中心不违反任何约束，则进行分配；如果违反约束，则尝试分配到次近的聚类中心，再执行步骤3；若没有能分配的聚类中心，则报错
- 步骤4. 更新：重新计算聚类中心 μ_k
- 步骤5. 迭代：重复分配和更新步骤2-4，直到满足终止条件
 - 聚类中心不再发生显著变化
 - 达到最大迭代次数

半监督聚类

● 约束K均值算法

输入:

样本集 $D_u = \{x_1, x_2, \dots, x_m\}$

必连约束集合 \mathcal{M} 勿连约束集合 \mathcal{C}

聚类簇数 k

过程:

1: 从 D 中取初始均值向量 $\{\mu_1, \mu_2, \dots, \mu_k\}$

2: **repeat**

3: $C_j \neq \emptyset \{1 \leq j \leq k\}$

4: **for** $i = 1, 2, \dots, m$ **do**

5: $d_{ij} = \|x_i - \mu_j\|_2$

6: $\mathcal{K} = \{1, 2, \dots, k\}$;

7: $is_{merged} = false$

8: **while** $\neg is_{merged}$ **do**

9: $r = \operatorname{argmin}_{j \in \mathcal{K}} d_{ij}$

不冲突, 选择最近的簇

冲突, 尝试次近的簇

10:

11:

13:

14:

15:

16:

18:

19:

20:

21:

22:

23:

24:

25:

输出:

簇划分:

x_i 划入 C_r 中是否违背 \mathcal{M} 与 \mathcal{C} 中约束

if $\neg is_{violated}$ **do**

$C_r = C_r \cup \{x_i\}$

$is_{merged} = true$

else

$\mathcal{K} = \mathcal{K} \setminus \{r\}$

if $\mathcal{K} \neq \emptyset$ **then**

break 并返回错误提示

end if

end if

end while

end for

for $i = 1, 2, \dots, k$ **do**

$\mu_j = \frac{1}{|C_j|} \sum_{x_i \in C_j} x$

end for

until 均值向量未更新

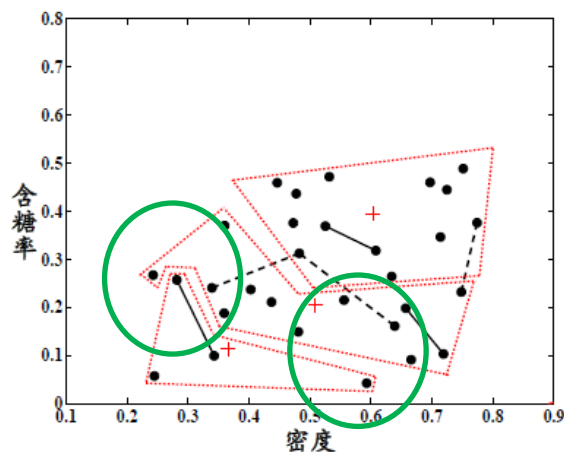
输出:

簇划分: $\{C_1, C_2, \dots, C_k\}$

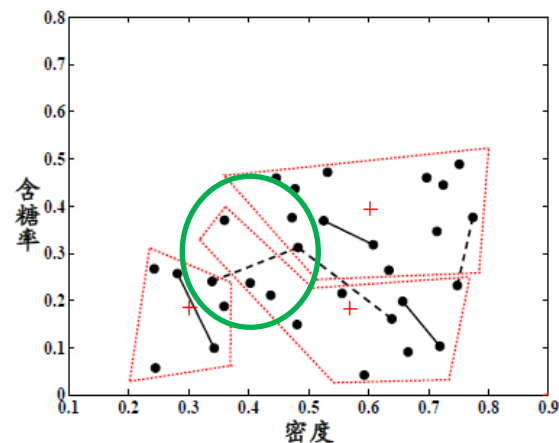
半监督聚类

● 约束K均值算法示例

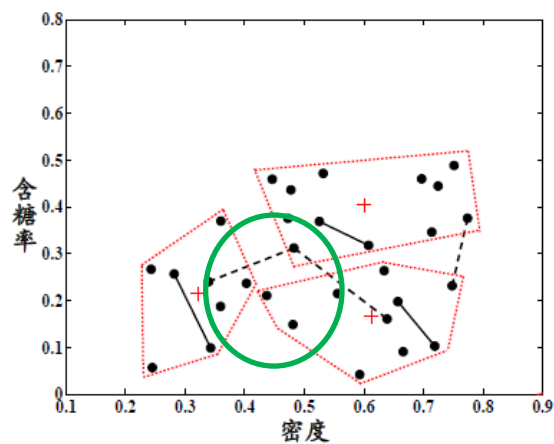
➤ 迭代过程



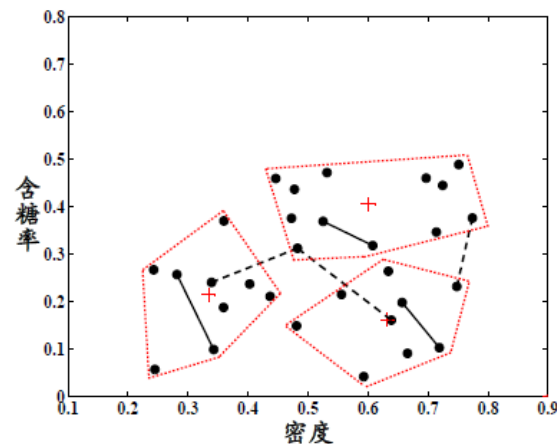
第1轮迭代



第2轮迭代



第3轮迭代



第4轮迭代

半监督聚类

● 约束种子K均值

【2002年S. Basu等人提出】

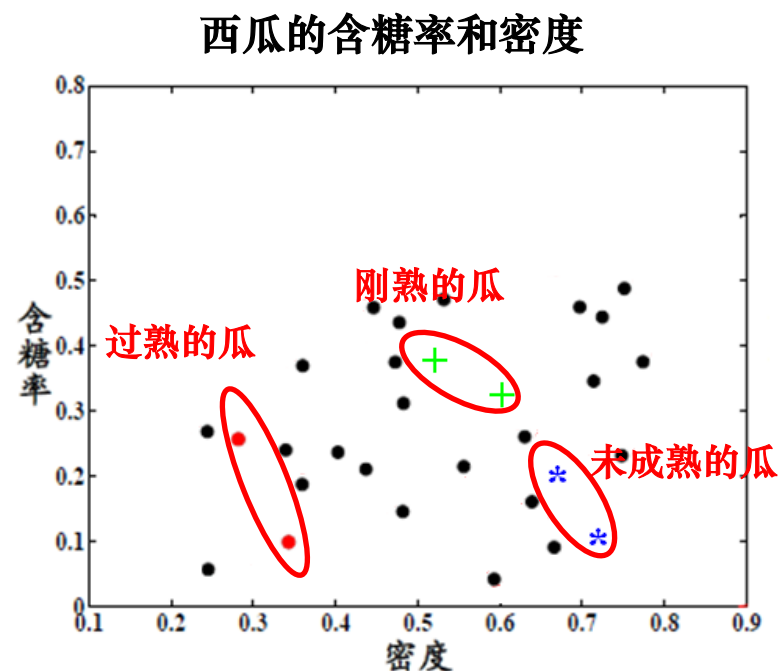
➤ 该算法是K均值算法的扩展

- 假设少量标记样本属于K个聚类簇直接将它们作为“种子”，用它们初始化K均值算法的K个聚类中心，并且在聚类簇迭代更新过程中不改变种子样本的簇隶属关系

■ 一个例子：西瓜是否成熟

现在已知的信息如下

已知少量的西瓜成熟度



半监督聚类

● 约束种子K均值算法：基本思路

- 步骤1. **初始化**：使用**标记样本初始化**K个初始聚类中心
- 步骤2. **标记样本分配**：将**标记样本**分配给对应分类的聚类中心
- 步骤3. 分配：将其余的每个样本分配给最近的聚类中心 μ_k
- 步骤4. 更新：重新计算聚类中心 μ_k
- 步骤5. 迭代：重复分配和更新步骤2-4，直到满足终止条件
 - 聚类中心不再发生显著变化
 - 达到最大迭代次数

半监督聚类

● 约束种子K均值算法

输入:

样本集 $D_u = \{x_1, x_2, \dots, x_m\}$

少量有标记样本 $S = \bigcup_{j=1}^k S_j$

聚类簇数 k

过程:

1: **for** $i = 1, 2, \dots, k$ **do**

2: $\mu_j = \frac{1}{|S_j|} \sum_{x_i \in S_j} x$

3: **end for**

4: **repeat**

5: $C_j \neq \emptyset \{1 \leq j \leq k\}$

6: **for** $i = 1, 2, \dots, k$ **do**

7: **for all** $x_i \in S_j$ **do**

8: $C_j = C_j \cup \{x\}$

9: **end for**

10: **end for**

11: **for all** $x_i \in D/S$ **do**

12: $d_{ij} = \|x_i - \mu_j\|_2$

13: $r = \operatorname{argmin}_{j \in \mathcal{K}} d_{ij}$

14: $C_r = C_r \cup \{x_i\}$

15: **end for**

16: **for** $i = 1, 2, \dots, k$ **do**

17: $\mu_j = \frac{1}{|C_j|} \sum_{x_i \in C_j} x$

18: **end for**

19: **until** 均值向量未更新

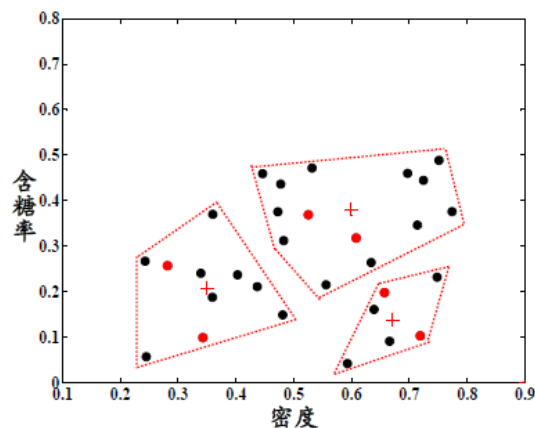
输出:

簇划分: $\{C_1, C_2, \dots, C_k\}$

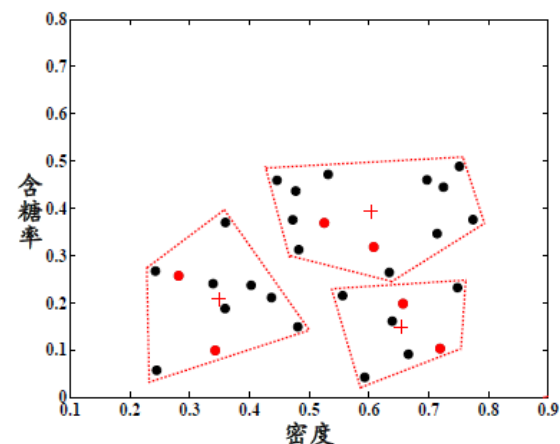
半监督聚类

● 约束种子K均值算法示例

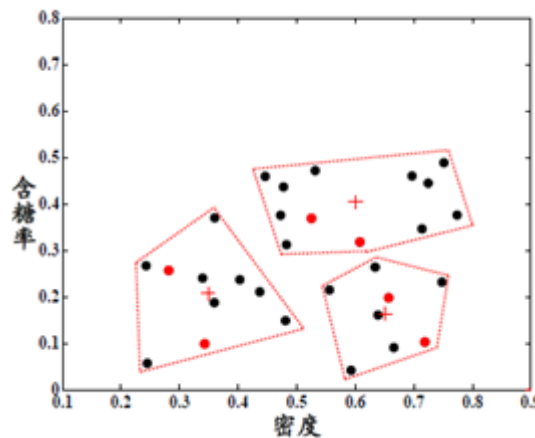
➤ 迭代过程



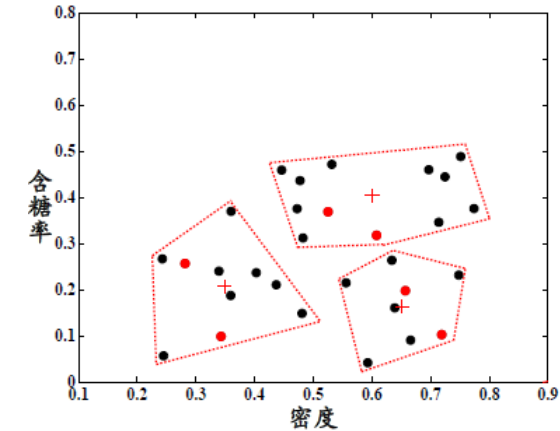
第1轮迭代



第2轮迭代



第3轮迭代



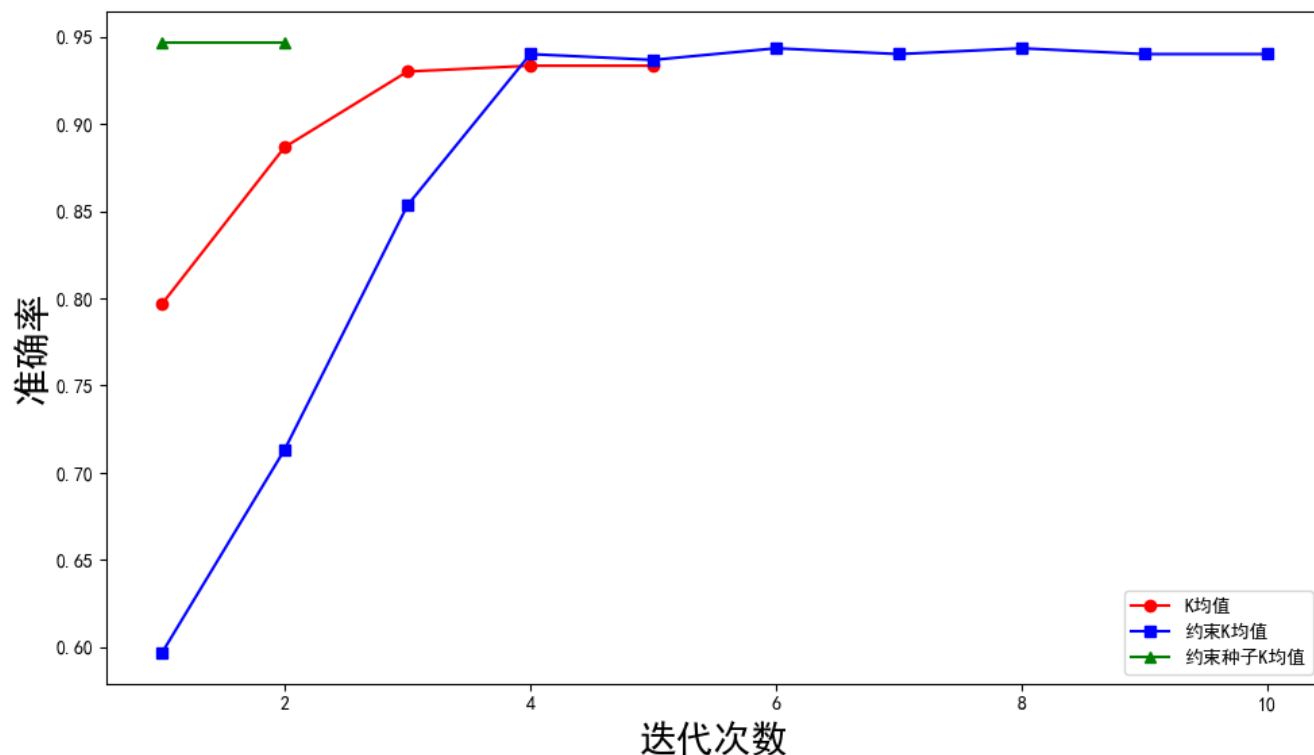
第4轮迭代

半监督聚类 VS 聚类

● 半监督聚类和聚类的实验对比图

➤ 半监督聚类**准确率高于**聚类

➤ 约束种子K均值可以有效**减少迭代次数**，而约束K均值需要的迭代次数**较多**

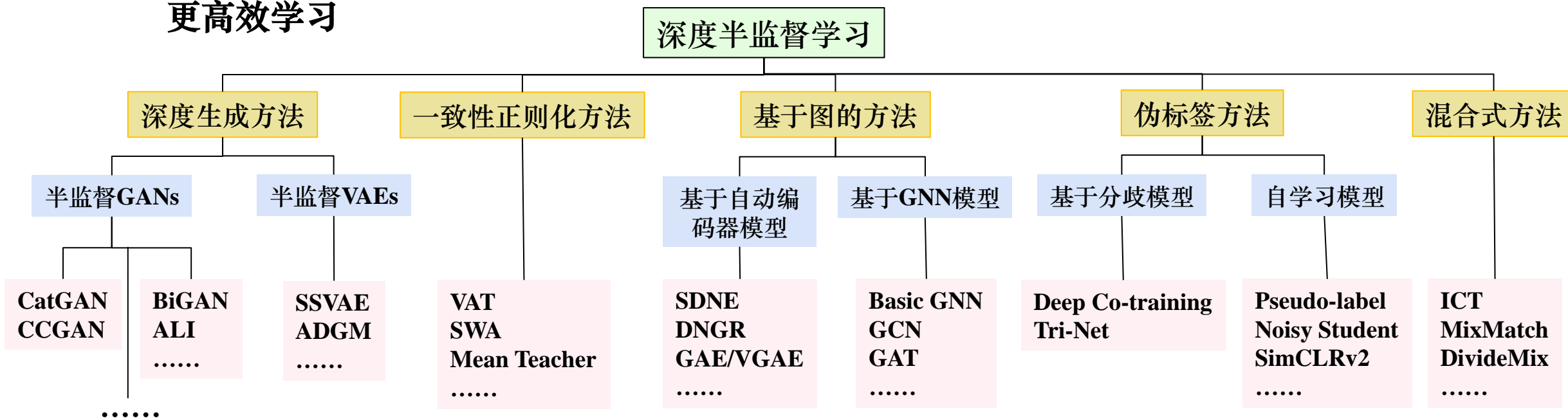


半监督学习发展近况

● 半监督学习发展

➤ 发展近况:

- 随着深度学习的成功, 出现了结合了生成对抗网络、图神经网络、自动编码器等新方法的深度半监督学习策略
- 深度半监督学习进一步结合了深度神经网络的**强大特征提取能力**, 实现了在更复杂数据上的更高效学习



自监督学习 (Self-Supervised Learning)

● 自监督学习概念

- 自监督学习 (Self-Supervised Learning) 是一种利用**无标记样本**训练算法模型的机器学习方法，它通过构造辅助任务从大规模无监督数据中挖掘监督信息，提高模型性能和泛化能力
- 自监督学习可以被视为**自学习**的一种具体实现形式
- 代表性方法：对比学习、MAE (Masked Autoencoders)、SimCLR等
- 自监督学习流程

