电子科技大学
University of Electronic Science and Technology of China

《机器学习》
第八章：集成学习

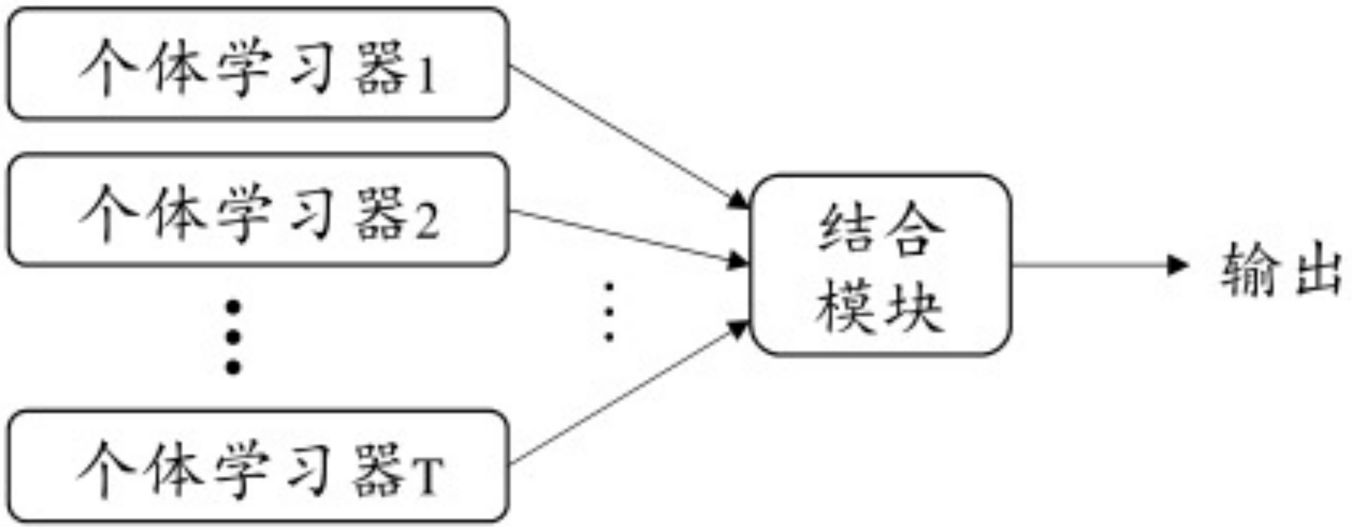
任亚洲
yazhou.ren@uestc.edu.cn
计算机科学与工程学院

集成学习

- 个体与集成
- Boosting
  - Adaboost
- Bagging与随机森林
- 结合策略
  - 平均法
  - 投票法
  - 学习法
- 多样性
  - 误差-分歧分解
  - 多样性度量
  - 多样性扰动

个体与集成

- **集成学习(ensemble learning)通过构建并结合多个学习器来提升性能**



课堂讨论

- 集成学习中，个体学习器应如何选择？

个体与集成

- 考虑一个简单的例子，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图所示，其中√ 表示分类正确，X 号表示分类错误，集成的结果通过投票产生。

	测试例1	测试例2	测试例3
h_1	√	√	×
h_2	×	√	√
h_3	√	×	√
集群	√	√	√

(a) 集群提升性能

	测试例1	测试例2	测试例3
h_1	√	√	×
h_2	√	√	×
h_3	√	√	×
集群	√	√	×

(b) 集群不起作用

	测试例1	测试例2	测试例3
h_1	√	×	×
h_2	×	√	×
h_3	×	×	√
集群	×	×	×

(c) 集群起负作用

- 集成个体应：好而不同

个体与集成 – 简单分析

- 考虑二分类问题，假设基分类器的错误率为：

$$P\left(h_i\left(\boldsymbol{x}\right) \neq f\left(\boldsymbol{x}\right)\right) = \epsilon$$

- 假设集成通过简单投票法结合T个分类器，若有超过半数的基分类器正确则分类就正确

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

个体与集成 – 简单分析

- 假设基分类器的错误率相互独立，则由Hoeffding霍夫丁不等式可得集成的错误率为：

$$
\begin{aligned}
P(H(\boldsymbol{x}) \neq f(\boldsymbol{x})) &= \sum_{k=0}^{\lfloor T/2 \rfloor} \binom{T}{k} (1-\epsilon)^k \epsilon^{T-k} \\
&\leq \exp\left(-\frac{1}{2} T (1-2\epsilon)^2\right)
\end{aligned}
$$

- 上式显示，在一定条件下，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

个体与集成 – 简单分析

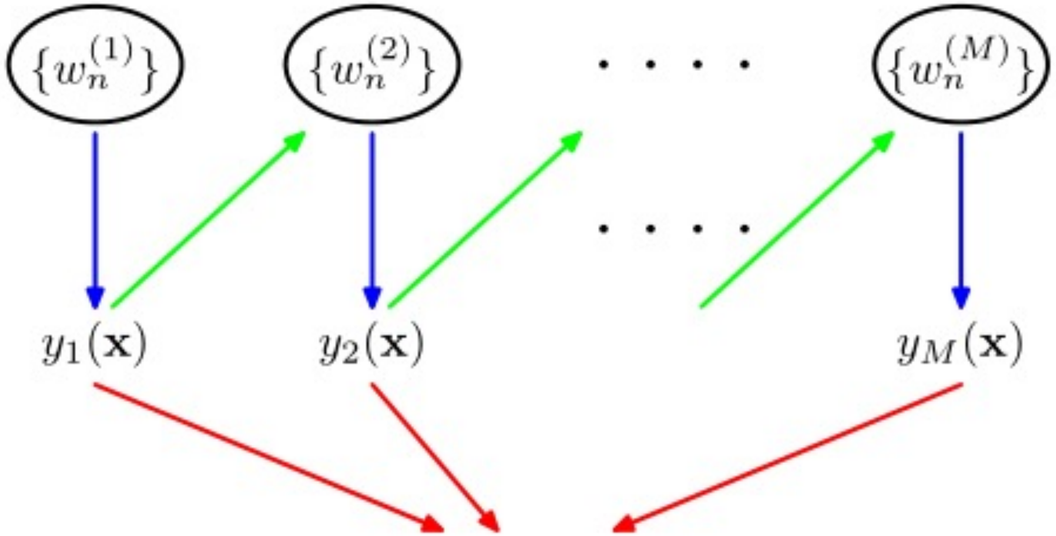
- 上面的分析有一个关键假设：基学习器的误差相互独立
- 现实任务中，个体学习器是为解决同一个问题训练出来的，显然**不可能互相独立**
- 事实上，个体学习器的“**准确性**”和“**多样性**”本身就存在冲突
- 如何产生“好而不同”的个体学习器是集成学习研究的**核心**
- 集成学习大致可分为两大类
  - Boosting:串行，个体（基）学习器**存在强依赖**关系
  - Bagging和随机森林：个体（基）学习器**不存在强依赖**关系

集成学习

- 个体与集成
- Boosting
  - **Adaboost**
- Bagging与随机森林
- 结合策略
  - 平均法
  - 投票法
  - 学习法
- 多样性
  - 误差-分歧分解
  - 多样性度量
  - 多样性扰动

Boosting

- 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



$$Y_M(\mathbf{x}) = \operatorname{sign}\left(\sum_{m}^{M} \alpha_m y_m(\mathbf{x})\right)$$

Boosting - Boosting算法

Input: Sample distribution $\mathcal{D}$;
Base learning algorithm $\mathfrak{L}$;
Number of learning rounds T.

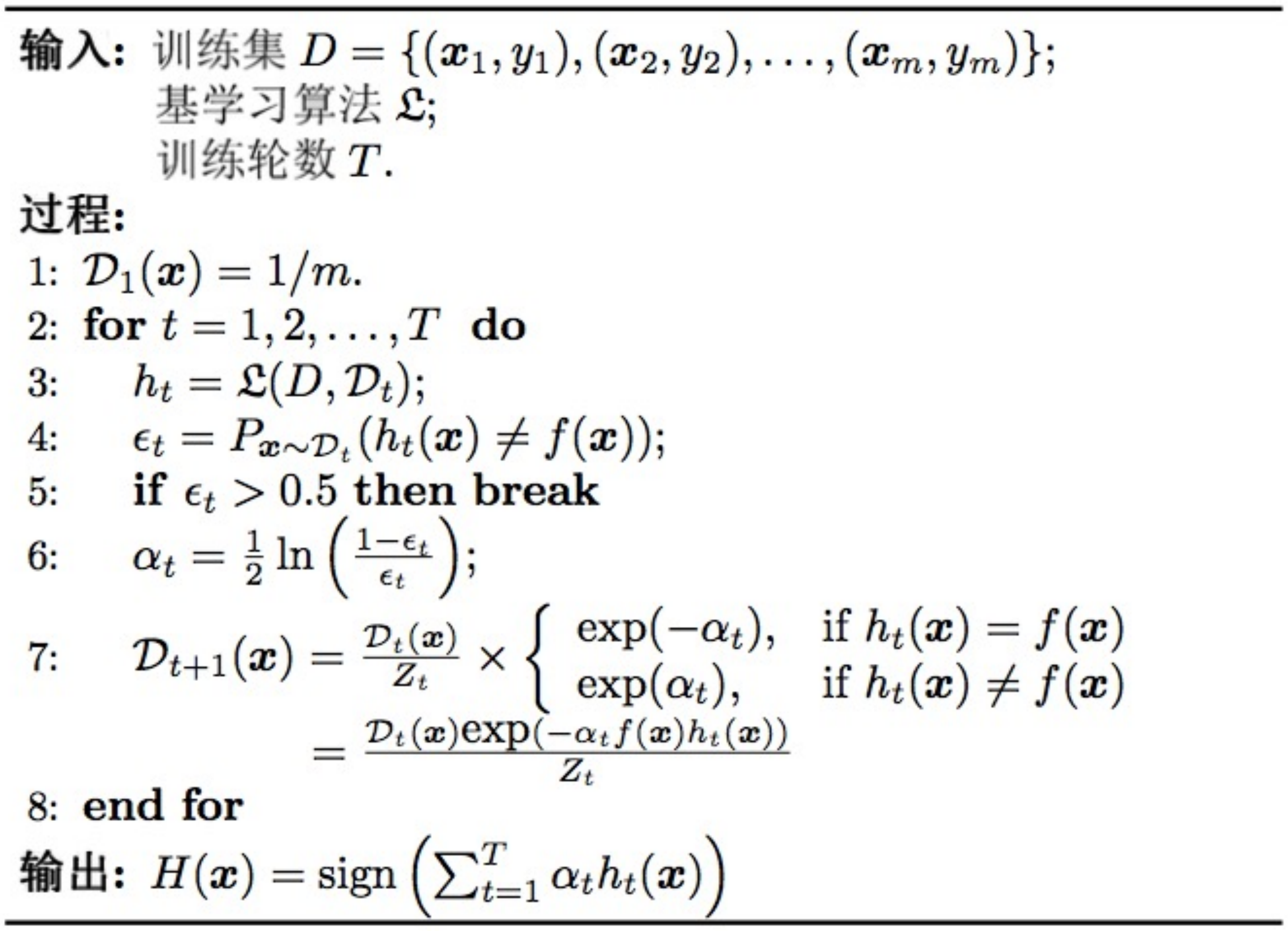
Process:

1. $\mathcal{D}_1 = \mathcal{D}$. % Initialize distribution
2. **for** $t = 1, \ldots, T$:
3. $h_t = \mathfrak{L}(\mathcal{D}_t)$; % Train a weak learner from distribution $\mathcal{D}_t$
4. $\epsilon_t = P_{\boldsymbol{x} \sim D_t}(h_t(\boldsymbol{x}) \neq f(\boldsymbol{x}))$; % Evaluate the error of h_t
5. $\mathcal{D}_{t+1} = Adjust\_Distribution(\mathcal{D}_t, \epsilon_t)$
6. **end**

Output: $H(\boldsymbol{x}) = Combine\_Outputs(\{h_1(\boldsymbol{x}), \ldots, h_t(\boldsymbol{x})\})$

Boosting – AdaBoost算法

- Boosting族算法最著名的代表是AdaBoost

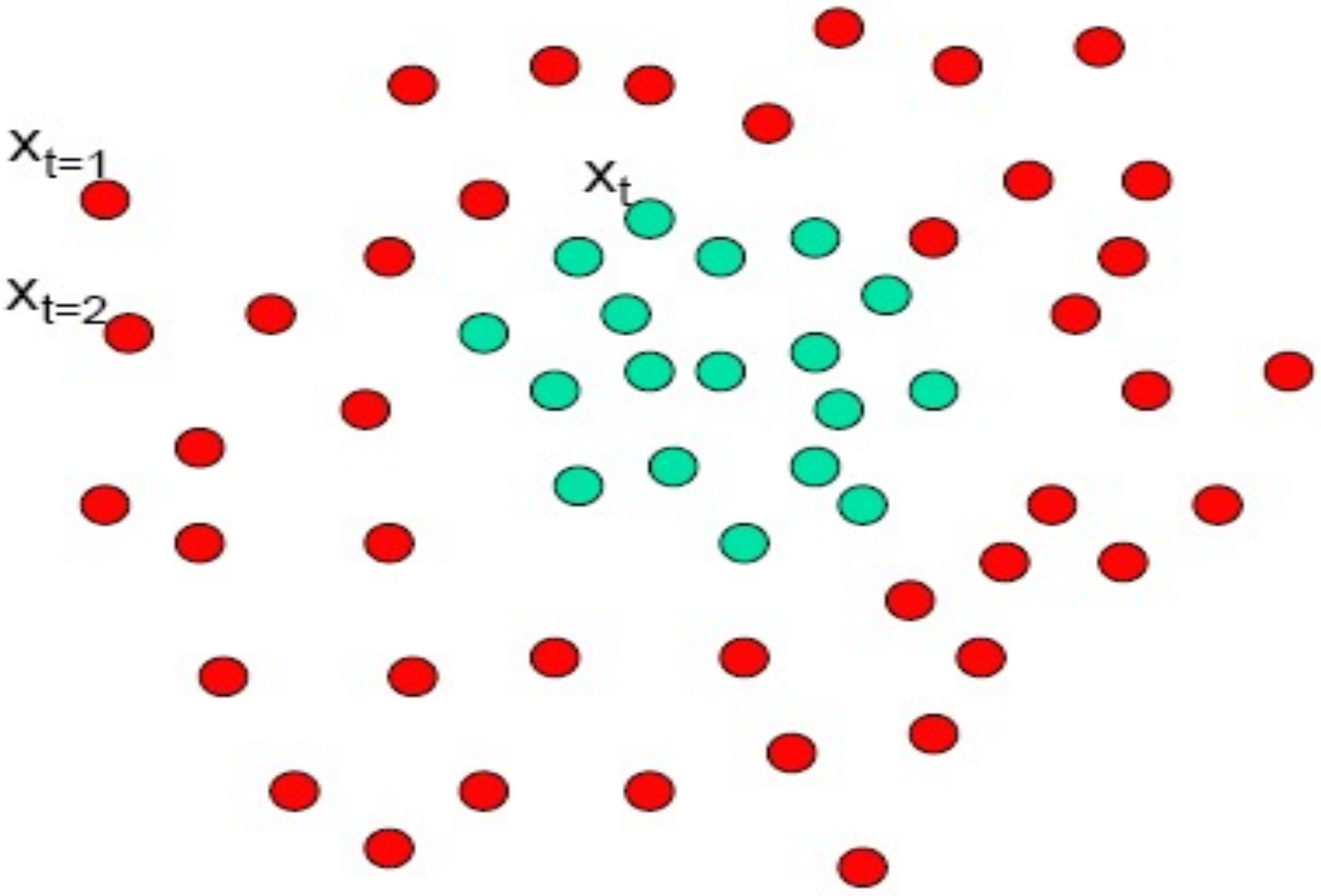


Boosting – AdaBoost注意事项

- 数据分布的学习
  - 重赋权法
  - 重采样法

- 利用“重采样法”实现重启动，避免训练过程过早停止

Boosting Example



Each data point has

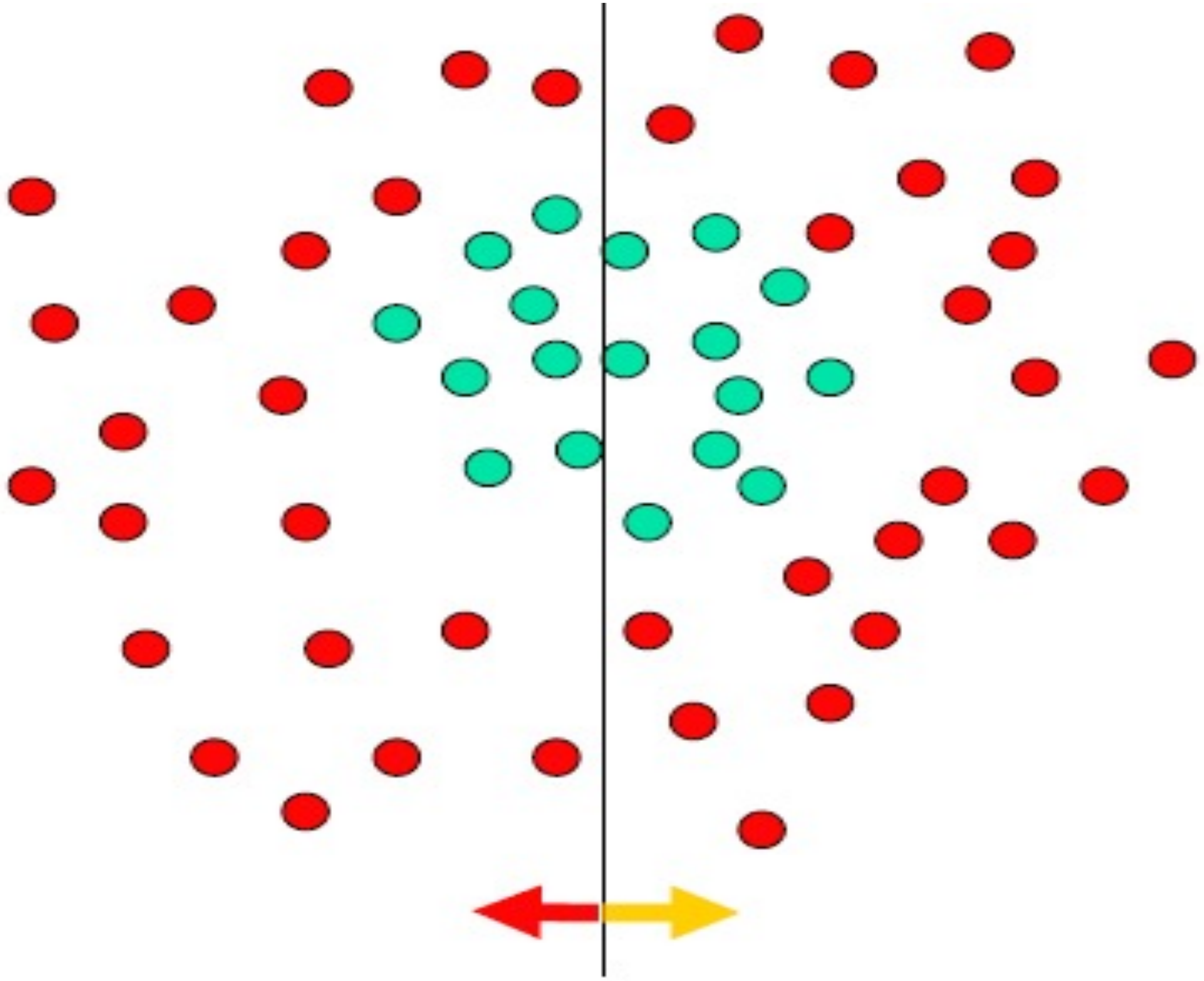
a class label:

$$y_t = \begin{cases} +1 \ (\text{●}) \\ -1 \ (\text{●}) \end{cases}$$

and a weight:

$w_t = 1$

Boosting Example



Each data point has

a class label:

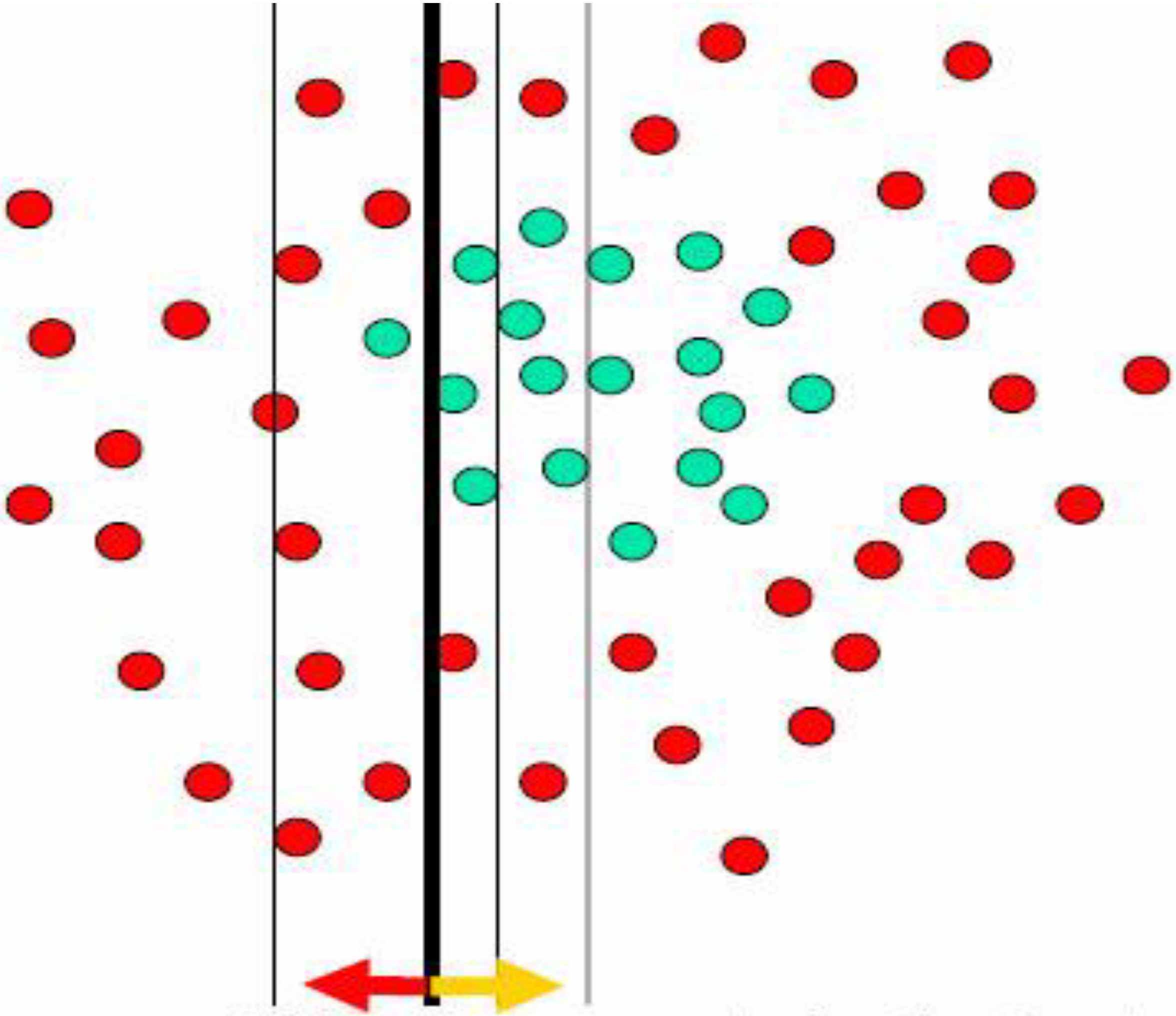
$$y_t = \begin{cases} +1 & (\text{red}) \\ -1 & (\text{green}) \end{cases}$$

and a weight:

$w_t = 1$

h => p(error) = 0.5 it is at chance

Boosting Example



Each data point has

a class label:

$$y_t = \begin{cases} +1 \\ -1 \end{cases}$$

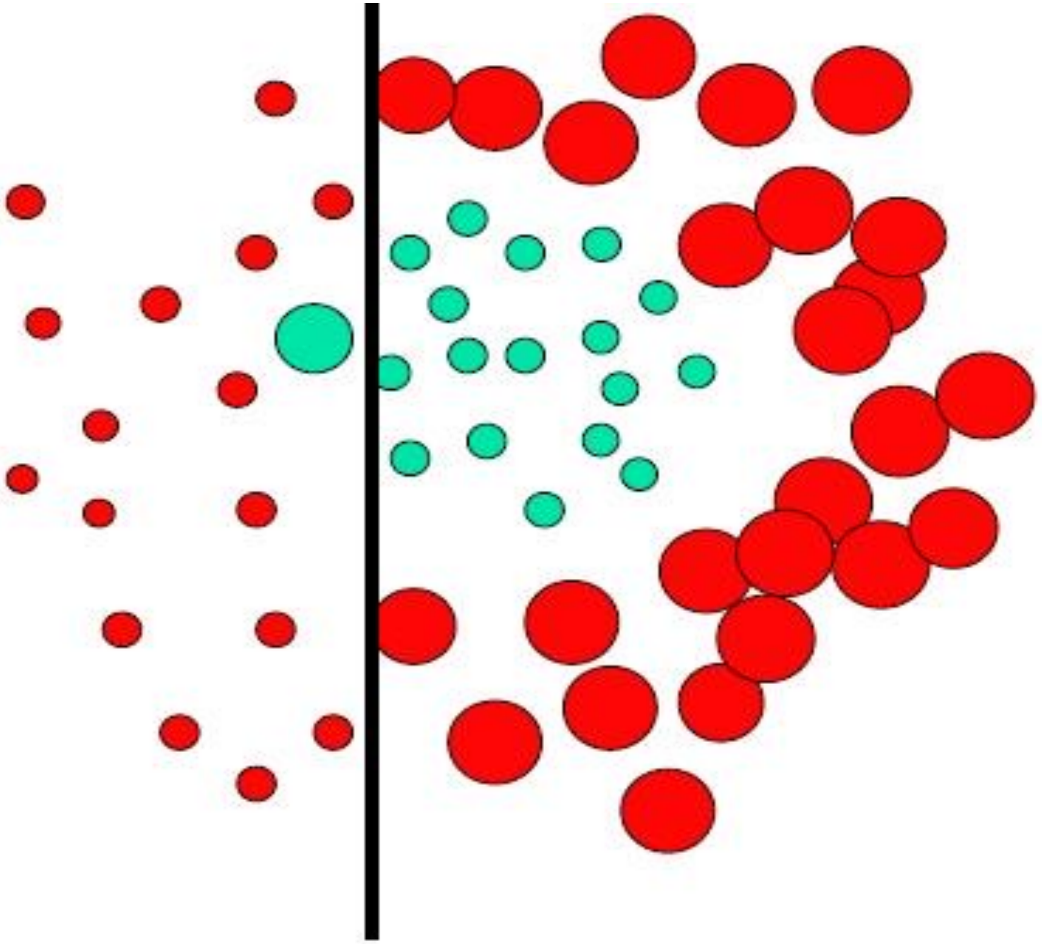
and a weight:

$w_t = 1$

This one seems to be the best

This is a '**weak classifier**': It performs slightly better than chance.

Boosting Example



Each data point has

a class label:

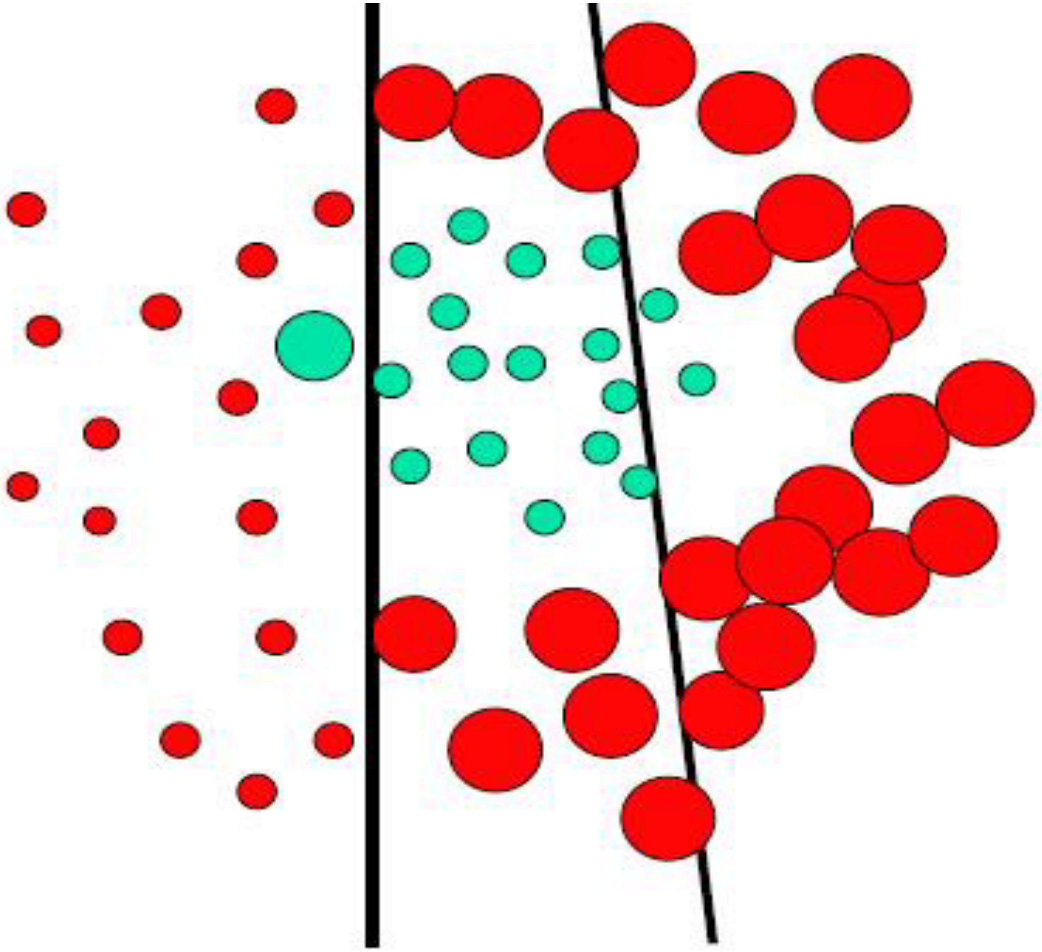
$$y_t = \begin{cases} +1 & (\bullet) \\ -1 & (\bullet) \end{cases}$$

We update the weights:

$$w_t \leftarrow w_t \exp\{-y_t H_t\}$$

We set a new problem for which the previous weak classifier performs at chance again

Boosting Example



Each data point has

a class label:

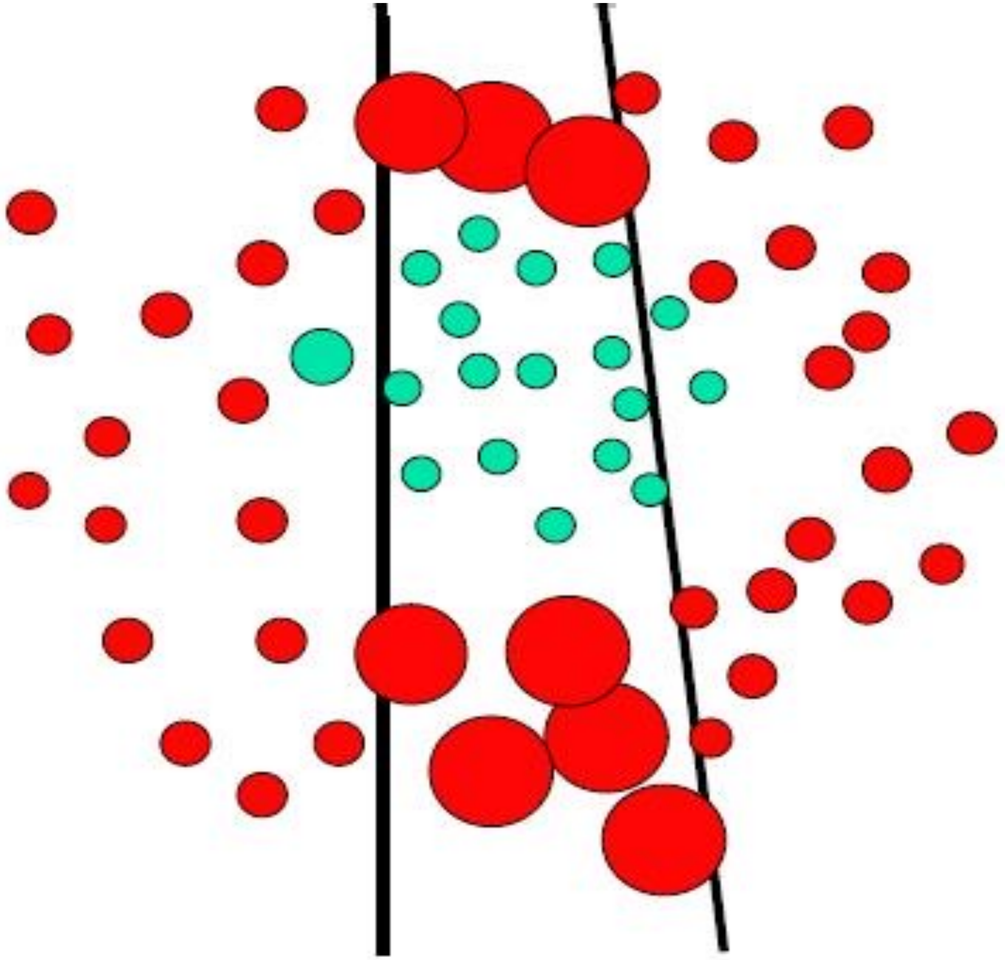
$$y_t = \begin{cases} +1 & (\text{red}) \\ -1 & (\text{green}) \end{cases}$$

We update the weights:

$$w_t \leftarrow w_t \exp\{-y_t H_t\}$$

We set a new problem for which the previous weak classifier performs at chance again

Boosting Example



Each data point has

a class label:

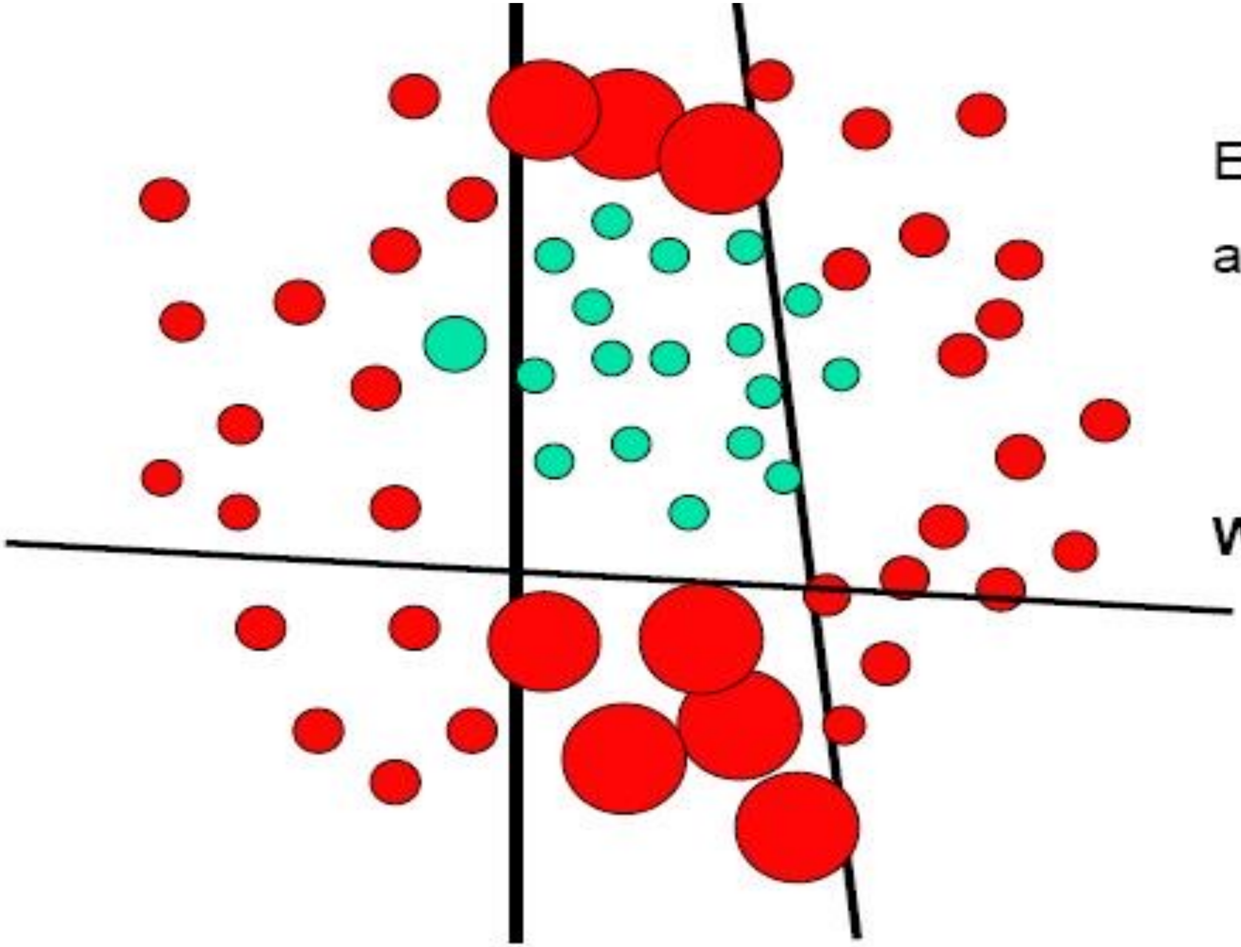
$$y_t = \begin{cases} +1 \ (\bullet) \\ -1 \ (\bullet) \end{cases}$$

We update the weights:

$$w_t \leftarrow w_t \exp\{-y_t H_t\}$$

We set a new problem for which the previous weak classifier performs at chance again

Boosting Example



Each data point has a class label:

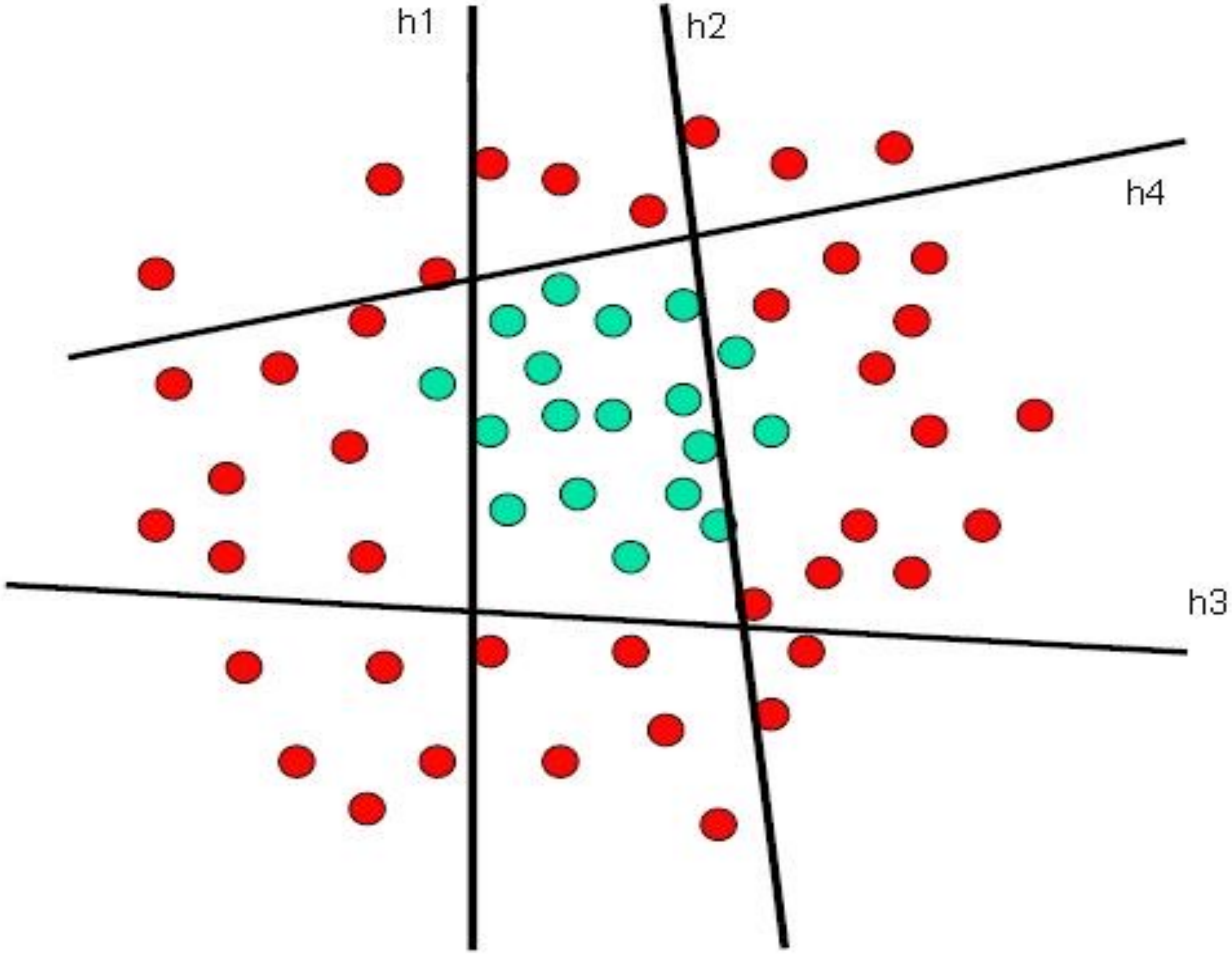
$$y_t = \begin{cases} +1 & (\text{red}) \\ -1 & (\text{green}) \end{cases}$$

We update the weights:

$$w_t \leftarrow w_t \exp\{-y_t H_t\}$$

We set a new problem for which the previous weak classifier performs at chance again

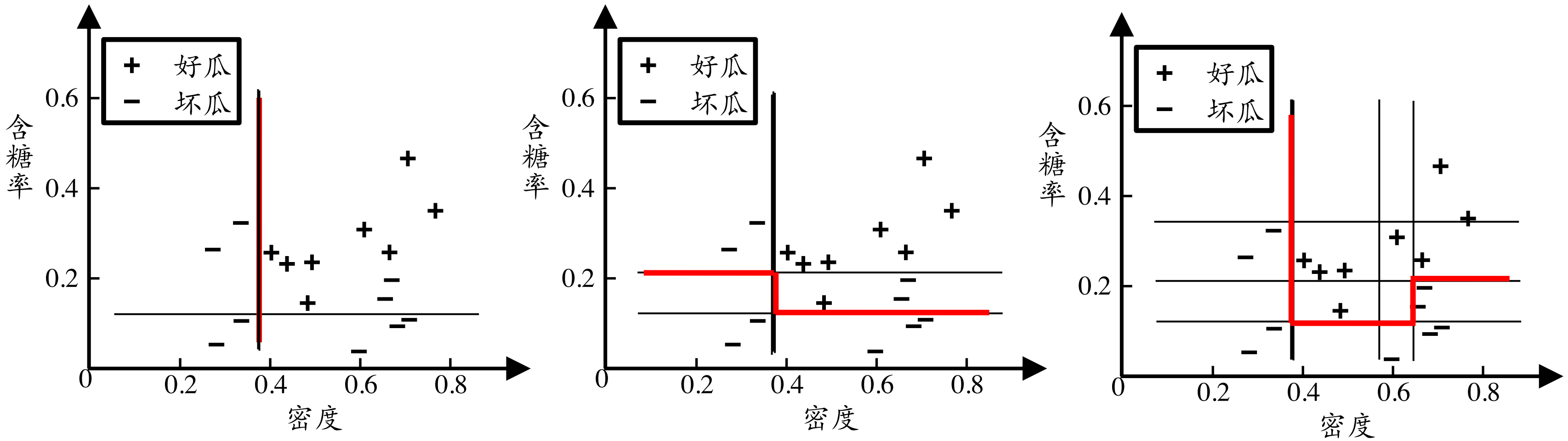
Boosting Example



The strong (non- linear) classifier is built as the combination of all the weak (linear) classifiers.

Boosting – AdaBoost实验

基分类器：决策树桩（单层决策树）



(a) 3个基学习器　(b) 5个基学习器　(c) 11个基学习器

- 从偏差-方差的角度：降低偏差，可对泛化性能相当弱的学习器构造出很强的集成 [偏差/方差：参考西瓜书2.5]

集成学习

- 个体与集成
- Boosting
  - Adaboost
- **Bagging与随机森林**
- 结合策略
  - 平均法
  - 投票法
  - 学习法
- 多样性
  - 误差-分歧分解
  - 多样性度量
  - 多样性扰动

Bagging与随机森林

- 个体学习器不存在强依赖关系
- 并行化生成
- 自助采样法

Bagging与随机森林 – Bagging算法

输入：训练集 $D = \{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \ldots, (\boldsymbol{x}_m, y_m)\}$;
基学习算法 $\mathfrak{L}$;
训练轮数 T.

过程：

1: **for** $t = 1, 2, \ldots, T$ **do**
2: $\quad h_t = \mathfrak{L}(D, \mathcal{D}_{bs})$
3: **end for**

输出：$H(\boldsymbol{x}) = \underset{y \in \mathcal{Y}}{\arg\max} \sum_{t=1}^{T} \mathbb{I}(h_t(\boldsymbol{x}) = y)$

对于一个样本，它在某一次含m个样本的训练集的随机采样中，每次被采集到的概率是$1/m$，不被采集到的概率为$\left(1-\frac{1}{m}\right)$，如果$m$次采样都没有被采集中的概率是$\left(1-\frac{1}{m}\right)^m$,当$m \to \infty$，$\left(1-\frac{1}{m}\right)^m \to \frac{1}{e} \cong 0.368$

- **回归：取T个结果的均值**

Bagging与随机森林 – Bagging算法特点

- **时间复杂度低**
  - 假定基学习器的计算复杂度为O(m)，采样与投票/平均过程的复杂度为O(s)，则bagging的复杂度大致为T(O(m)+O(s))
  - 由于O(s)很小且T是一个不大的常数
  - 因此训练一个bagging集成与直接使用基学习器的复杂度同阶

- **可使用包外估计**

Bagging与随机森林 – 包外估计

- $H^{oob}(x)$表示对样本x的包外预测，即仅考虑那些未使用样本x训练的基学习器在x上的预测

$$H^{oob}(\boldsymbol{x}) = \mathop{\arg\max}_{y \in \mathcal{Y}} \sum_{t=1}^{T} \mathbb{I}(h_t(\boldsymbol{x}) = y) \cdot \mathbb{I}(\boldsymbol{x} \notin D_t)$$

□ Bagging泛化误差的包外估计为：

$$\epsilon^{oob} = \frac{1}{|D|} \sum_{(\boldsymbol{x},y) \in D} \mathbb{I}(H^{oob}(\boldsymbol{x}) \neq y)$$

Bagging与随机森林- Bagging实验

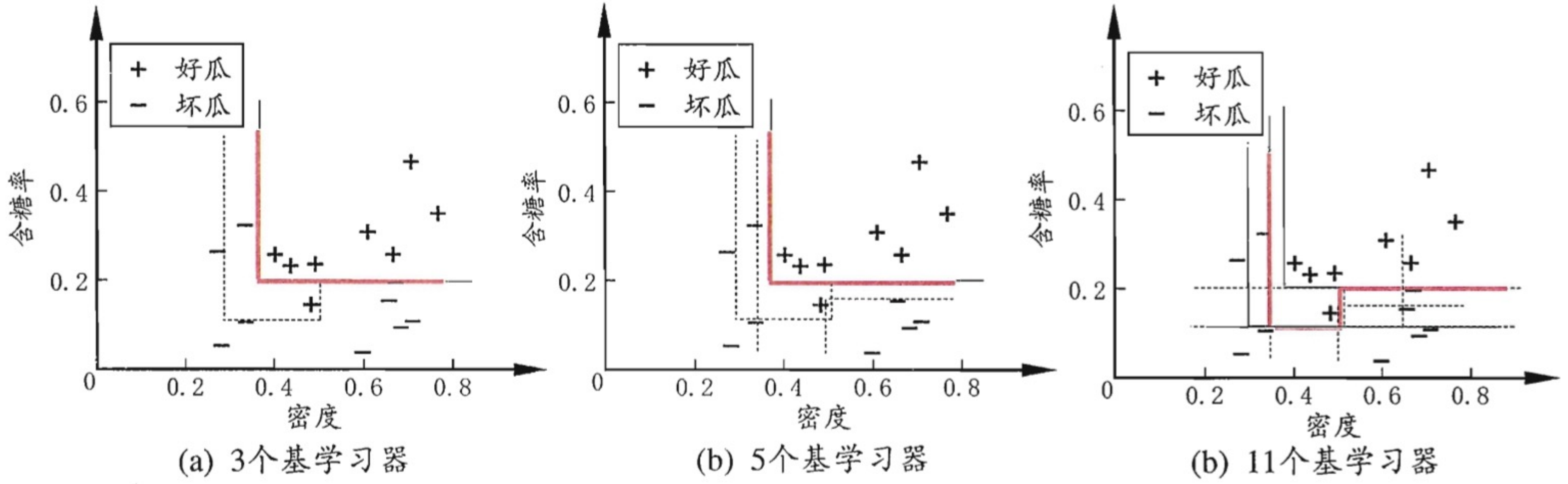


图 8.6 西瓜数据集 3.0α 上 Bagging 集成规模为 3、5、11 时，集成(红色)与基学习器(黑色)的分类边界.

- 从偏差-方差的角度：降低方差，在不剪枝的决策树、神经网络等易受样本影响的学习器上效果更好

Bagging与随机森林-随机森林

- 随机森林(Random Forest，简称RF)是bagging的一个扩展变种
  - 被誉为“代表集成学习技术水平的方法”
- 采样的随机性：
  - 与bagging一致
- 特征选择的随机性：
  - 随机选择一个特征子集，并从中计算最优分裂特征

随机森林算法

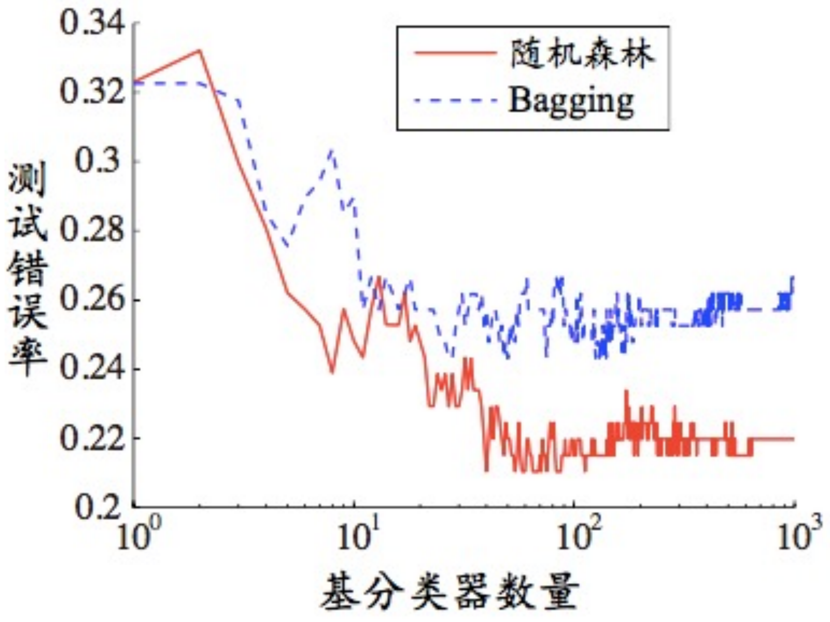
Input: Data set $D = \{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \ldots, (\boldsymbol{x}_m, y_m)\}$;
Feature subset size K.

Process:

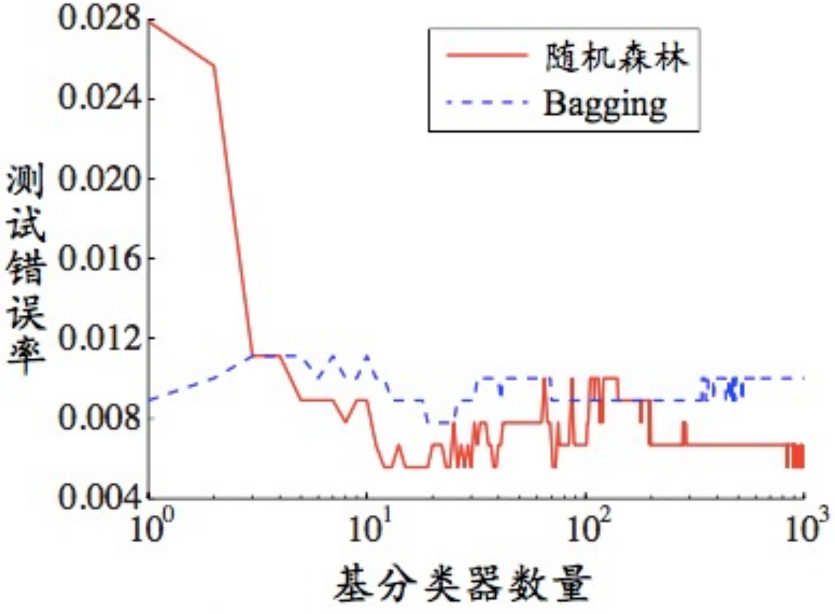
1. $N \leftarrow$ create a tree node based on D;
2. **if** *all instances in the same class* **then return** N
3. $\mathcal{F} \leftarrow$ the set of features that can be split further;
4. **if** $\mathcal{F}$ *is empty* **then return** N
5. $\tilde{\mathcal{F}} \leftarrow$ select K features from $\mathcal{F}$ randomly;
6. $N.f \leftarrow$ the feature which has the best split point in $\tilde{\mathcal{F}}$;
7. $N.p \leftarrow$ the best split point on $N.f$;
8. $D_l \leftarrow$ subset of D with values on $N.f$ smaller than $N.p$;
9. $D_r \leftarrow$ subset of D with values on $N.f$ no smaller than $N.p$;
10. $N_l \leftarrow$ call the process with parameters (D_l, K);
11. $N_r \leftarrow$ call the process with parameters (D_r, K);
12. **return** N

Output: A random decision tree

Bagging与随机森林 – 随机森林实验



(a) glass 数据集



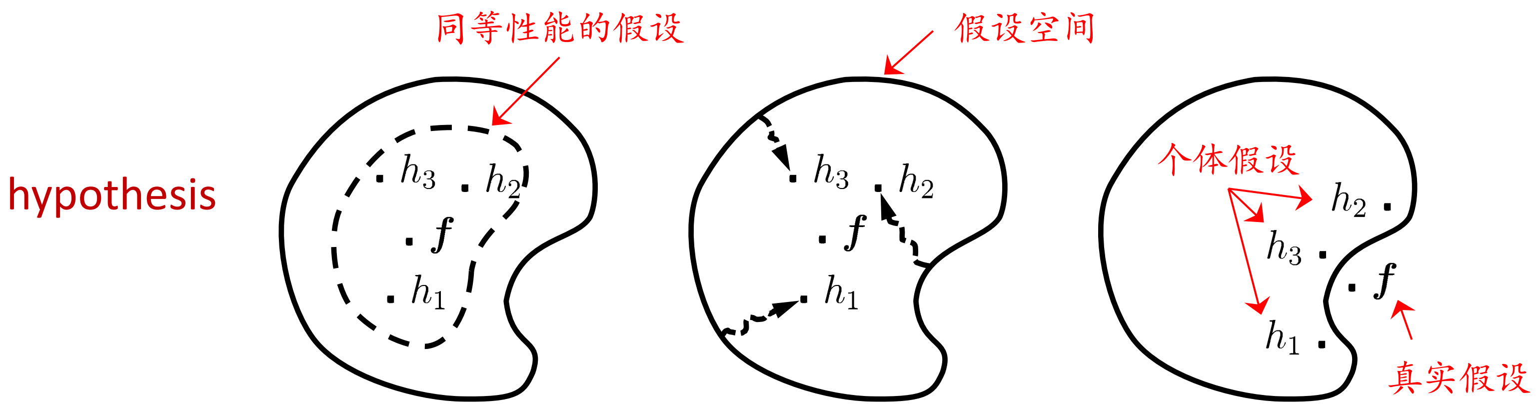
(b) auto-mpg 数据集

集成学习

- 个体与集成
- Boosting
  - Adaboost
- Bagging与随机森林
- 结合策略
  - 平均法
  - 投票法
  - 学习法
- 多样性
  - 误差-分歧分解
  - 多样性度量
  - 多样性扰动

结合策略

- **学习器的组合可以从三个方面带来好处**



(a) 统计的原因 (b) 计算的原因 (c) 表示的原因

➢ **统计的原因**：多个假设在训练集上达到同等性能，若使用单个，因为误选，导致泛化性能不佳；结合多个学习器会降低这个风险

➢ **计算的原因**：学习算法往往会陷入局部极小，有的局部极小点对应的泛化性能可能很糟，经过多次运行之后进行结合，可降低陷入糟糕的局部极小点的概率

➢ **表示的原因**：假设空间变大，有可能学到更好的近似

结合策略 – 平均法

- **简单平均法**

$$H(\boldsymbol{x}) = \frac{1}{T}\sum_{i=1}^{T} h_i(\boldsymbol{x}).$$

- **加权平均法**

$$H(\boldsymbol{x}) = \sum_{i=1}^{T} w_i h_i(\boldsymbol{x}) \quad w_i \geq 0 \ \text{ and } \ \sum_{i=1}^{T} w_i = 1 \,.$$

结合策略 – 平均法

- 简单平均法是加权平均法的特例
- 加权平均法在二十世纪五十年代被广泛使用
- 集成学习中的各种结合方法都可以看成是加权平均法的变种或特例
- 加权平均法可认为是集成学习研究的基本出发点
- 加权平均法未必一定优于简单平均法

个体性能差距大　　　　个体性能差距小

结合策略 – 投票法

- **绝对多数投票法（majority voting）**

$$H(\boldsymbol{x})=\begin{cases} c_j & \text{if } \sum\limits_{i=1}^{T} h_i^j(\boldsymbol{x}) > \frac{1}{2}\sum\limits_{k=1}^{l}\sum\limits_{i=1}^{T} h_i^k(\boldsymbol{x}) \\ \text{rejection} & \text{otherwise}. \end{cases}$$

- **相对多数投票法（plurality voting）**

$$H(\boldsymbol{x})=c_{\underset{j}{\arg\max} \sum_{i=1}^{T} h_i^j(\boldsymbol{x})}$$

- **加权投票法（weighted voting）**

$$H(\boldsymbol{x})=c_{\underset{j}{\arg\max} \sum_{i=1}^{T} w_i h_i^j(\boldsymbol{x})}$$

结合策略 – 学习法

- **Stacking是学习法的典型代表**

输入: 训练集 $D=\{(\boldsymbol{x}_1,y_1),(\boldsymbol{x}_2,y_2),\ldots,(\boldsymbol{x}_m,y_m)\}$;
初级学习算法 $\mathfrak{L}_1,\mathfrak{L}_2,\ldots,\mathfrak{L}_T$;
次级学习算法 $\mathfrak{L}$.

过程:

for $t=1,2,\ldots,T$ **do**
  $h_t=\mathfrak{L}_t(D)$;
end for
$D'=\varnothing$;
for $i=1,2,\ldots,m$ **do**
  for $t=1,2,\ldots,T$ **do**
    $z_{it}=h_t(\boldsymbol{x}_i)$;
  end for
  $D'=D'\cup((z_{i1},z_{i2},\ldots,z_{iT}),y_i)$;
end for
$h'=\mathfrak{L}(D')$;

输出: $H(\boldsymbol{x})=h'(h_1(\boldsymbol{x}),h_2(\boldsymbol{x}),\ldots,h_T(\boldsymbol{x}))$

集成学习

- 个体与集成
- Boosting
  - Adaboost
- Bagging与随机森林
- 结合策略
  - 平均法
  - 投票法
  - 学习法
- 多样性
  - 误差-分歧分解
  - 多样性度量
  - 多样性扰动

多样性 – 误差–分歧分解

- **定义学习器h_i的分歧(ambiguity)：**

$$A(h_i \mid \boldsymbol{x}) = \left(h_i(\boldsymbol{x}) - H(\boldsymbol{x})\right)^2$$

□ **集成的分歧：**

$$\begin{aligned}\overline{A}(h \mid \boldsymbol{x}) &= \sum_{i=1}^{T} w_i A(h_i \mid \boldsymbol{x}) \\ &= \sum_{i=1}^{T} w_i \left(h_i(\boldsymbol{x}) - H(\boldsymbol{x})\right)^2\end{aligned}$$

多样性 – 误差–分歧分解

- 分歧项代表了个体学习器在样本x上的不一致性，即在一定程度上反映了个体学习器的多样性，个体学习器h_i和集成H的平方误差分别为

$$E(h_i \mid \boldsymbol{x}) = \left(f(\boldsymbol{x}) - h_i(\boldsymbol{x})\right)^2$$

$$E(H \mid \boldsymbol{x}) = \left(f(\boldsymbol{x}) - H(\boldsymbol{x})\right)^2$$

多样性 – 误差–分歧分解

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

$$\begin{aligned} \overline{A}(h \mid \boldsymbol{x}) &= \sum_{i=1}^{T} w_i E(h_i \mid \boldsymbol{x}) - E(H \mid \boldsymbol{x}) \\ &= \overline{E}(h \mid \boldsymbol{x}) - E(H \mid \boldsymbol{x}) \, . \end{aligned}$$

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

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

多样性 – 误差-分歧分解

- 个体学习器h_i在全样本上的泛化误差和分歧项分别为：

$$E_i = \int E(h_i \mid \boldsymbol{x}) p(\boldsymbol{x}) d\boldsymbol{x}$$

$$A_i = \int A(h_i \mid \boldsymbol{x}) p(\boldsymbol{x}) d\boldsymbol{x}$$

- 集成的泛化误差为：

$$E = \int E(H \mid \boldsymbol{x}) p(\boldsymbol{x}) d\boldsymbol{x}$$

- 令 $\overline{E} = \sum_{i=1}^{T} w_i E_i$ 表示个体学习器泛化误差的加权均值，$\overline{A} = \sum_{i=1}^{T} w_i A_i$ 表示个体学习器的加权分歧值，有

$$E = \overline{E} - \overline{A}$$

多样性 – 误差–分歧分解

- 这个漂亮的式子显示:**个体学习器精确性越高、多样性越大，则集成效果越好**。称为误差-分歧分解

- 为什么不能直接把$\bar{E}-\bar{A}$作为优化目标来求解？
  - 现实任务中很难直接对$\bar{E}-\bar{A}$进行优化，
    - 它们定义在整个样本空间上
    - $\bar{A}$不是一个可直接操作的多样性度量
    - 上面的推导过程只适用于回归学习，难以直接推广到分类学习任务上去

多样性 – 多样性度量

- 多样性度量(diversity measure)用于度量集成中个体学习器的多样性

□ 对于二分类问题，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 － 多样性度量

常见的多样性度量

- **不合度量(Disagreement Measure)**

$$dis_{ij} = \frac{b+c}{m}$$

- **相关系数(Correlation Coefficient)**

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

多样性 – 多样性度量

□ 常见的多样性度量

- **Q-统计量(Q-Statistic)**

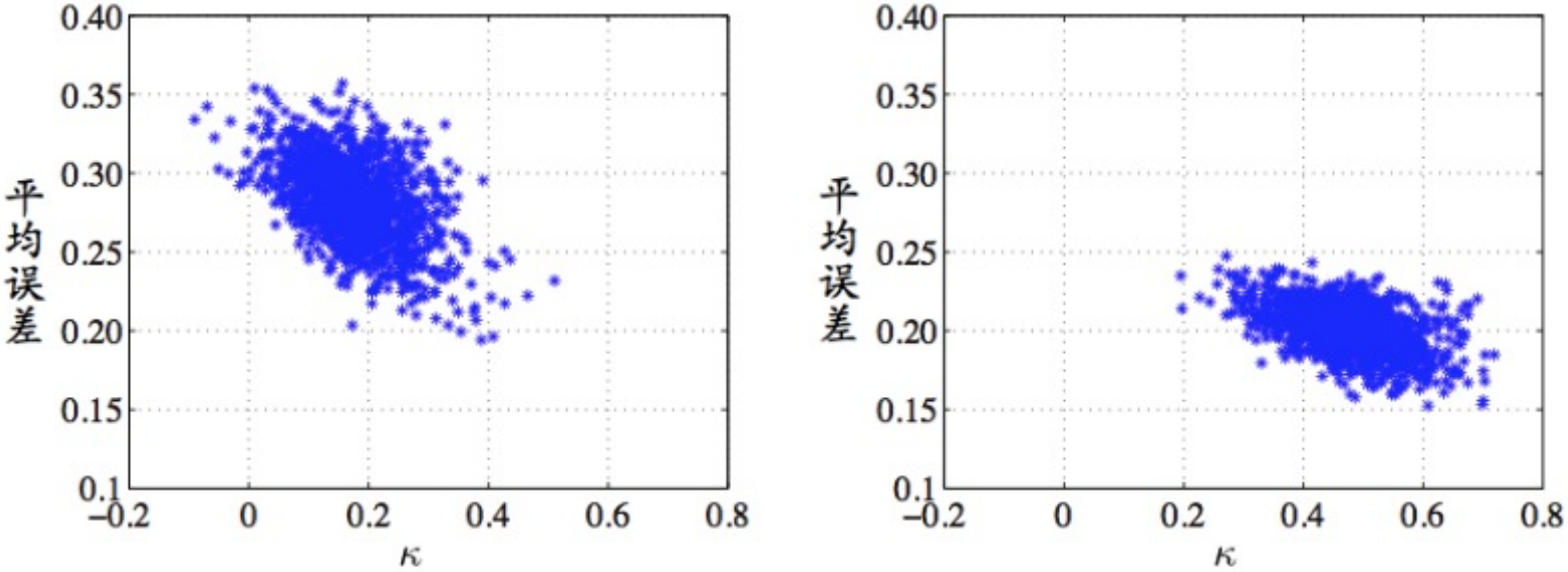
$$Q_{ij} = \frac{ad - bc}{ad + bc} \quad |Q_{ij}| \leq |\rho_{ij}|$$

- k **-统计量(Kappa-Statistic)**

$$\kappa = \frac{p_1 - p_2}{1 - p_2} \qquad \begin{aligned} p_1 &= \frac{a+d}{m}, \\ p_2 &= \frac{(a+b)(a+c)+(c+d)(b+d)}{m^2} \end{aligned}$$

多样性 – 多样性度量

□ k –误差图



(a) AdaBoost 集成 (b) Bagging 集成

课堂测验

- $k-$误差图，如点云出现在右下角，则代表（　）

A　个体学习器精度高，个体学习器多样性大

B　个体学习器精度高，个体学习器多样性小

C　个体学习器精度低，个体学习器多样性大

D　个体学习器精度低，个体学习器多样性小

多样性 – 多样性增强

- 常见的增强个体学习器的多样性的方法

  - ✓ 数据样本扰动
  - ✓ 输入属性扰动
  - ✓ 输出表示扰动
  - ✓ 算法参数扰动

多样性 – 多样性增强 – 数据样本扰动

- 数据样本扰动通常是基于采样法
  - Bagging中的自助采样法
  - Adaboost中的序列采样

数据样本扰动对“不稳定基学习器”很有效

- 对数据样本的扰动敏感的基学习器(不稳定基学习器)
  - 决策树，神经网络等
- 对数据样本的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

多样性 – 多样性增强 – 输入属性扰动

□ 随机子空间算法(random subspace)

输入: 训练集 $D=\{(\boldsymbol{x}_1,y_1),(\boldsymbol{x}_2,y_2),\cdots,(\boldsymbol{x}_m,y_m)\}$;
基学习算法 $\mathfrak{L}$;
基学习器数 T;
子空间属性数 d'.

过程:

1: **for** $t=1,2,\ldots,T$ **do**
2: $\quad \mathcal{F}_t=\mathrm{RS}(D,d')$
3: $\quad D_t=\mathrm{Map}_{\mathcal{F}_t}(D)$
4: $\quad h_t=\mathfrak{L}(D_t)$
5: **end for**

输出: $H(\boldsymbol{x})=\underset{y\in\mathcal{Y}}{\arg\max}\sum_{t=1}^{T}\mathbb{I}\left(h_t\left(\mathrm{Map}_{\mathcal{F}_t}(\boldsymbol{x})\right)=y\right)$

多样性 – 多样性增强 – 输出表示扰动

- □翻转法(Flipping Output)
- □输出调剂法(Output Smearing)
- □ECOC法

多样性 – 多样性增强 – 算法参数扰动

- 负相关法
- 不同的多样性增强机制同时使用

阅读材料

- 集成学习方面的主要推荐读物是[Zhou,2012]，本章提及的所有内容在该书中都有更深入的详细介绍。[Kuncheva,2004;Rockach,2010b]可供参考，[Schapire and Freund,2012]则是专门关于Boosting的著作，集成学习方面有一些专门性的会议MCS(International Workshop on Multiple Classifier System).

- Boosting源于[Schapire,1990]对[Kearns and Valiant,1989]提出的”弱分类器是否等价于强学习”这个重要理论问题的构造性证明。最初的Boosting算法仅有理论意义，经数年努力后[Freund and Schapire,1997]提出Adaboost，并因此或得理论计算机科学方面的重要奖项—哥德尔奖。关于Boosting和Bagging已有的很多理论研究结果课参阅[Zhou,2012]第2-3章。

- 作业：网上下载或自己编程实现随机森林算法，在西瓜数据集3.0a（p.89表4.5）上测试算法性能（如参数敏感性、训练精度、收敛速度、运行时间等），并分析。提交实验报告。

谢　　谢！