

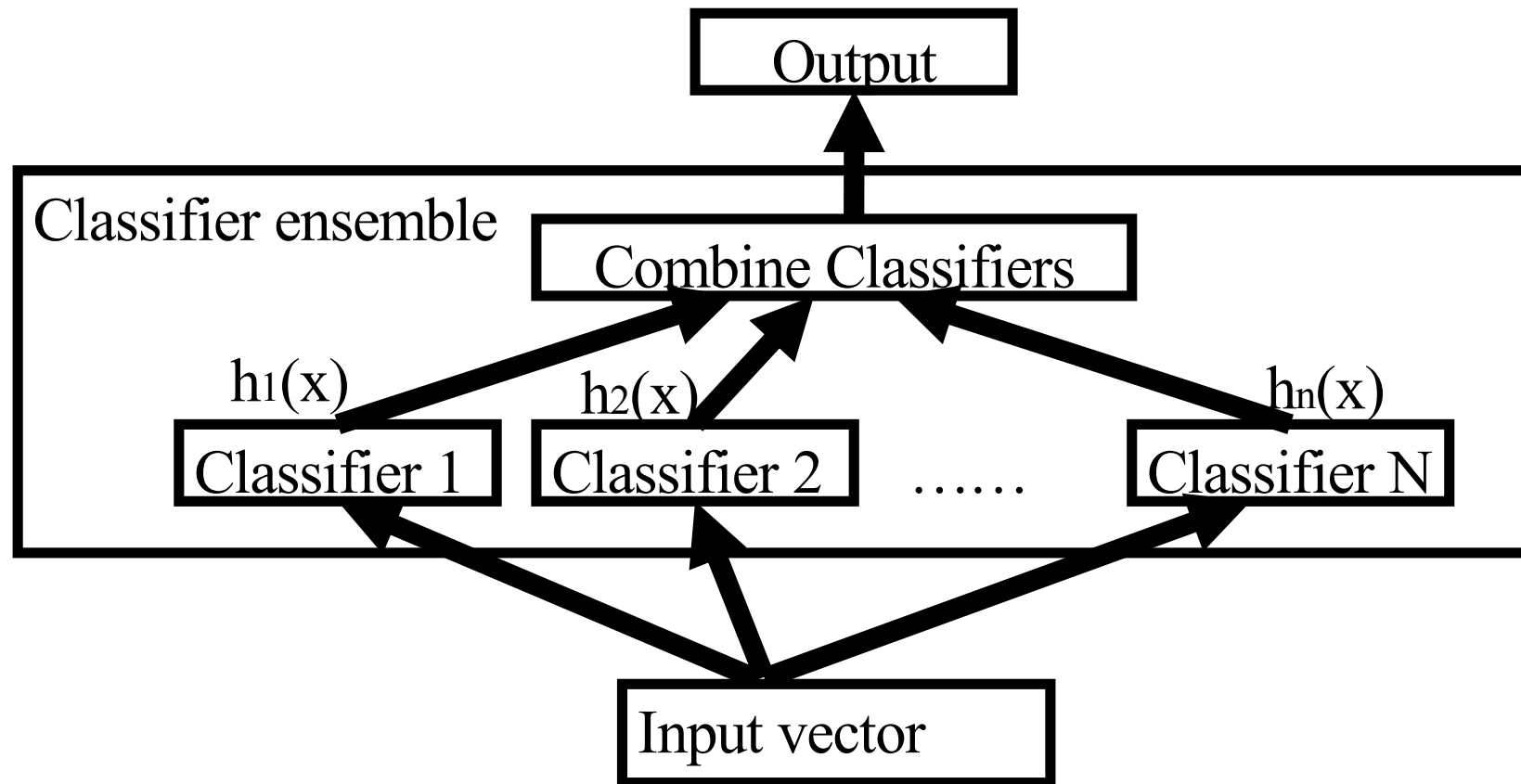
集成学习

机器学习算法之七

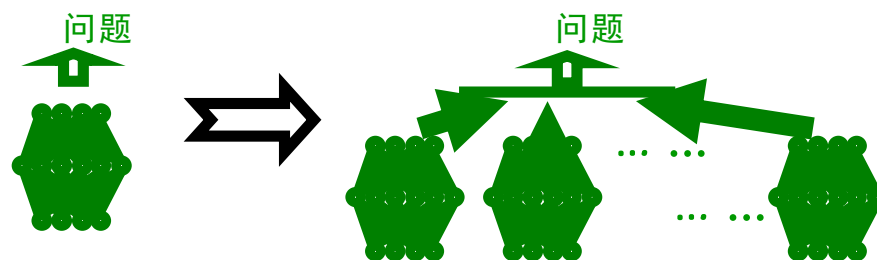
- 1.什么是集成学习
- 2.为什么要做集成学习
- 3.集成学习的方式
- 4常用的集成学习算法

- 什么是集成学习?
- 集成学习(Ensemble learning)是使用一系列学习器进行学习, 并使用某种规则把各个学习结果进行整合从而获得比单个学习器效果更好的一种机器学习方法。对于训练数据, 我们通过若干个个体学习器, 通过一定的聚合策略, 就可以形成一个强学习器, 以达到博采众长的目的。


















































- 动机:
- 在机器学习中，直接建立一个高性能的分类器是很困难的。
- 如果能找到一系列性能较差的分类器，并把它们集成起来的话，也许就能得到更好的分类器。



集成学习(Ensemble Learning)是一种机器学习范式，
它使用多个学习器来解决同一个问题



直观上来看，集成学习可以有效地提高学习系统的泛化能力。

Reality							
1		 X		X 			X 
2	X 			X 			X 
3			X 		X 	X 	
4			X 		X 		
5		X 				X 	
Combine							

- 真的能够博采众长么？

	测试例1	测试例2	测试例3
h_1	✓	✓	×
h_2	×	✓	✓
h_3	✓	×	✓
集成	✓	✓	✓

(a) 集成提升性能

	测试例1	测试例2	测试例3
h_1	✓	✓	×
h_2	✓	✓	×
h_3	✓	✓	×
集成	✓	✓	×

(b) 集成不起作用

	测试例1	测试例2	测试例3
h_1	✓	×	×
h_2	×	✓	×
h_3	×	×	✓
集成	×	×	×

(c) 集成起负作用

1. 一定准确性
2. 差异性

投票

5个独立的分类器：（多样性）

每个精度为70%

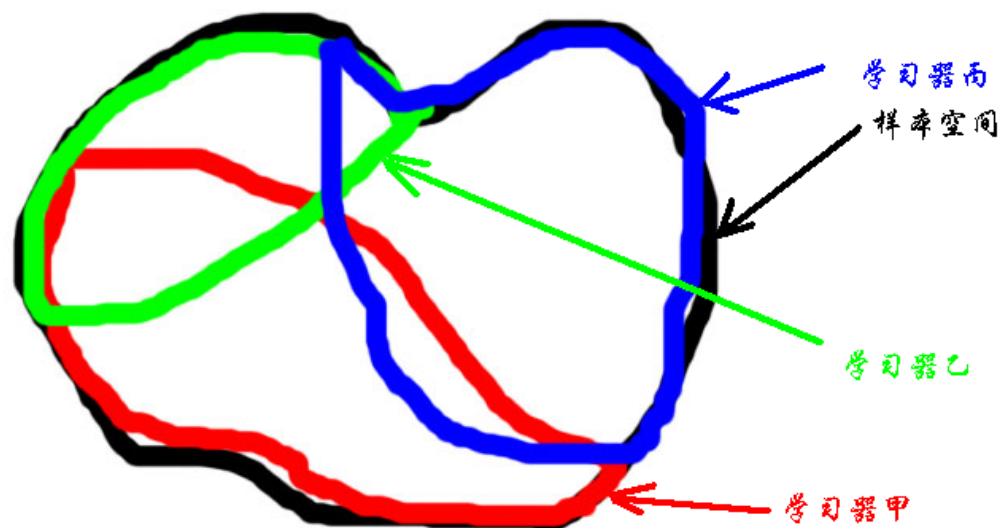
$$10 (0.7^3)(0.3^2)+5(0.7^4)(0.3)+(0.7^5) =83.7\%$$

100棵这样的数

- 能够达到99%准确率

【集成学习：如何构造？】

- 办法就是改变训练集。
- 通常的学习算法，根据训练集的不同，会给出不同的学习器。这时就可以通过改变训练集来构造不同的学习器。然后再把它们集成起来。



- 集成学习方式
- 1. bagging- (RF)
- 2. boosting - (GBDT/Adaboost/ XGBOOST)
- 3. stacking

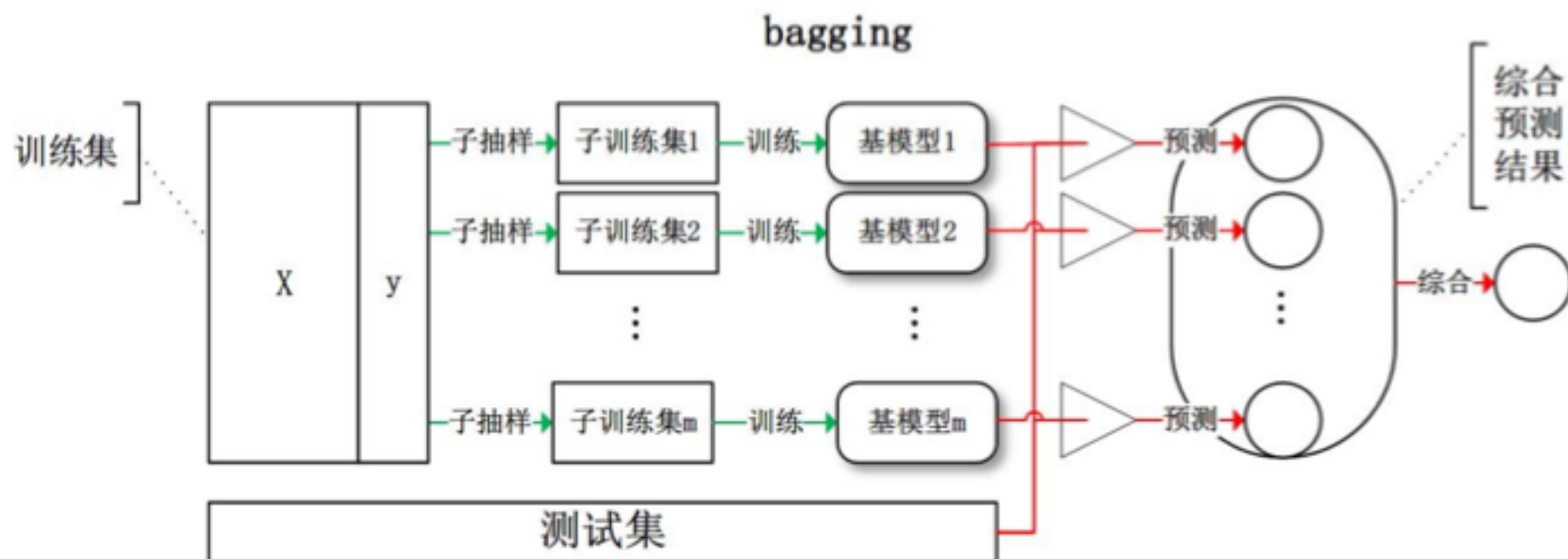
- Bagging
 - 1) 从原始样本集中抽取训练集。
 - 每轮从原始样本集中使用Bootstrapping（有放回）的方法抽取 n 个训练样本。共进行 k 轮抽取，得到 k 个训练集。（我们这里假设 k 个训练集之间是相互独立的，事实上不是完全独立）
 - 2) 每次使用一个训练集得到一个模型， k 个训练集共得到 k 个模型。将上步得到的 k 个模型采用**投票的方式**得到分类结果；对**回归**问题，计算上述模型的**均值**作为最后的结果

A, B, C, D, E



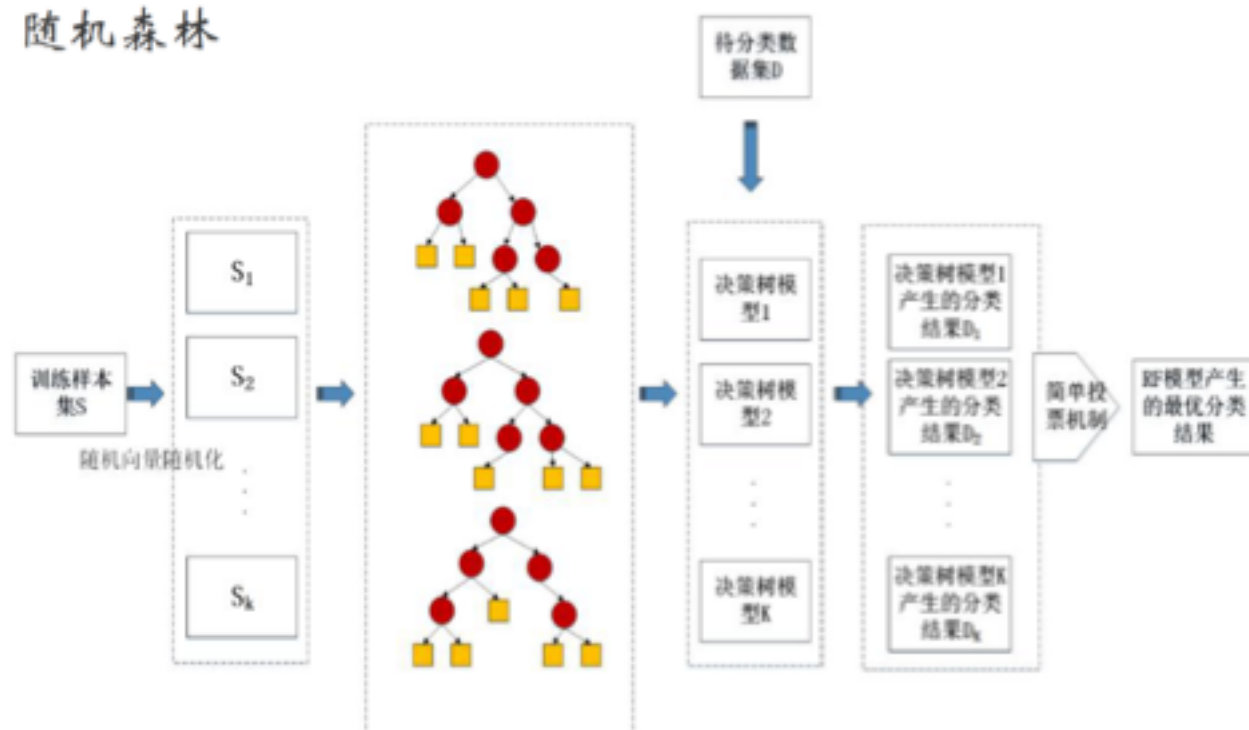
D, E, E, A, B, C, B, A, E

- 有没有可能样本始终没被取到？
- 每次大约有36.8%的数据没有被抽到。
 - $(1-1/n)^m$
 - $\lim_{n \rightarrow \infty} [(1-1/n)^m]$



基模型：同质/强分类器

随机森林

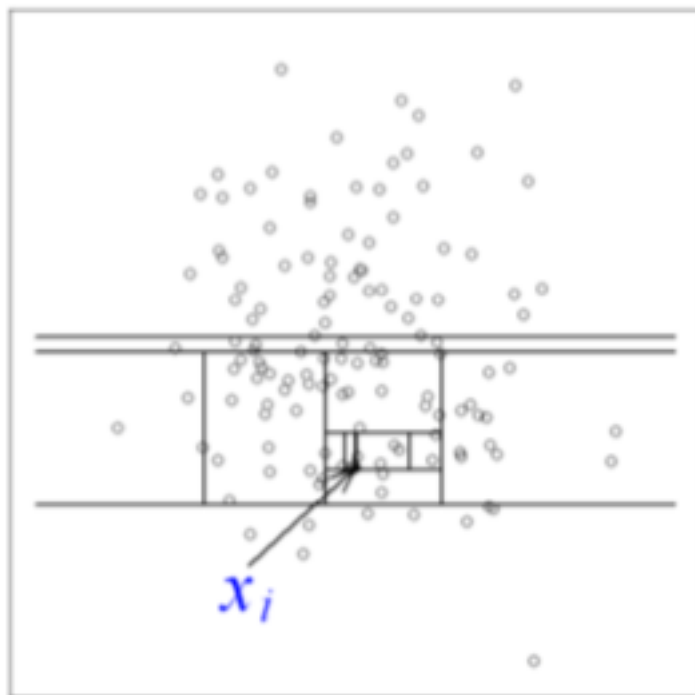


- 两种随机过程:
 - 样本随机
 - 特征随机

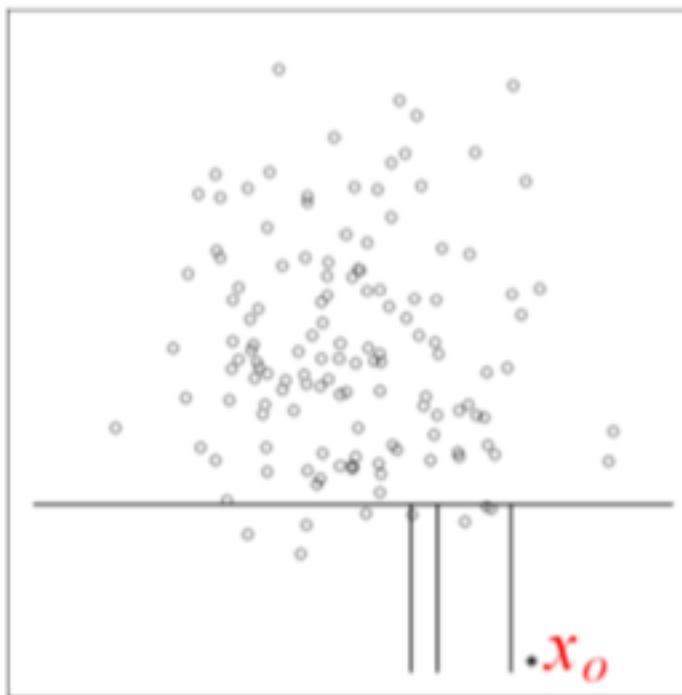
- 随机森林优缺点：
 - 训练可以高度并行化，对于大数据时代的大样本训练速度有优势。
 - 由于采用了随机采样，泛化能力强。
 - 对部分特征缺失不敏感。
- 对于噪音较大的样本，容易过拟合
- 可解释性较决策树弱

- 随机的艺术:
 - 特征选择
 - 被选择频次
 - 离根节点更近
 - 异常检验 (**Isolation Forest**)
 - 较早被分出来的最有可能是异常值

- Isolation Forest (孤立森林)



(a) Isolating x_i



(b) Isolating x_o

<http://25231283>

- Boosting

Boosting有很多种，比如AdaBoost(Adaptive Boosting)，Gradient Boosting等，这里以AdaBoost为典型讲。

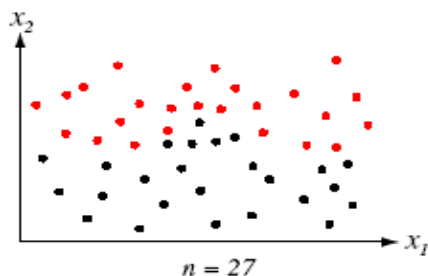
Boosting也是集合了多个决策树，但是Boosting的每棵树是顺序生成的，每一棵树都依赖于前一颗树。

对新的样本 x 进行分类，如果 C_1 和 C_2 判别结果相同，则将 x 判别为此类别，否则以 C_3 的结果作为 x 的类别；

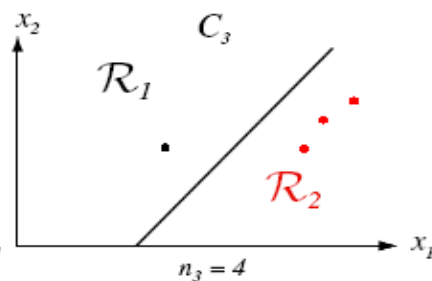
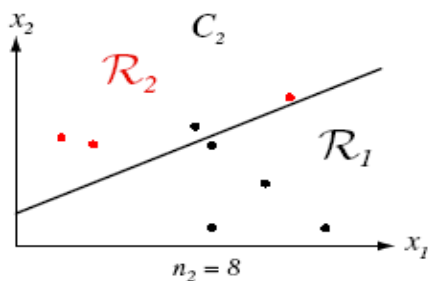
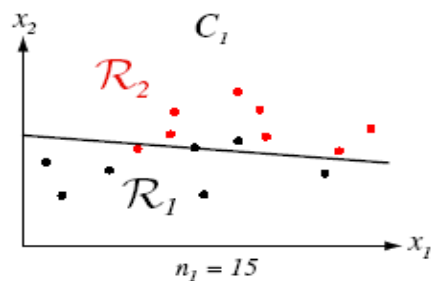
• 3个分类器，2分类数据

1. 在数量为 n 的原始样本集 D 中随机选取 n_1 个样本构成 D_1 ，利用 D_1 训练出一个分类器 C_1 ；
2. 在样本集 $D-D_1$ 中选择被 C_1 正确分类和错误分类的样本各一半组成样本集 D_2 ，用 D_2 训练出一个分类器 C_2 ；
3. 将样本集 $D-D_1-D_2$ 中所有 C_1 和 C_2 分类结果不同的样本组成样本集 D_3 ，训练出一个分类器 C_3 ；

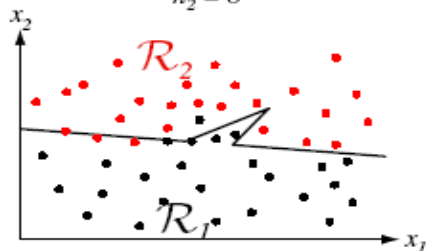
原始样本集



分量分类



组合分类器



Adaboost

$$\left. \begin{array}{l} h_1(x) \in \{-1, +1\} \\ h_2(x) \in \{-1, +1\} \\ \vdots \\ h_T(x) \in \{-1, +1\} \end{array} \right\} H_T(x) = \text{sign} \left(\sum_{t=1}^T \alpha_t h_t(x) \right)$$

弱分类器

强分类器

步骤:

样本: $(x_1, y_1), \dots, (x_m, y_m)$ where $x_i \in X, y_i \in \{-1, +1\}$

初始化权值: $D_1(i) = \frac{1}{m}, i = 1, \dots, m$

对于 $t = 1, \dots, T$:

• 分类器 $h_t : X \rightarrow \{-1, +1\}$

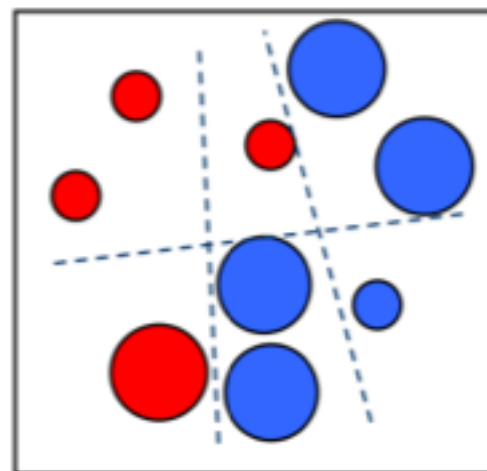
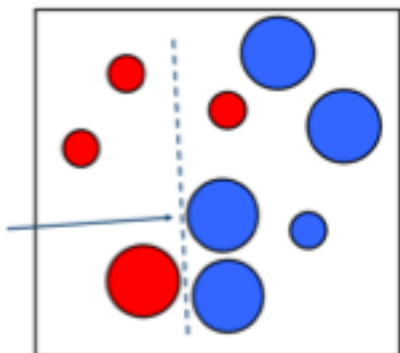
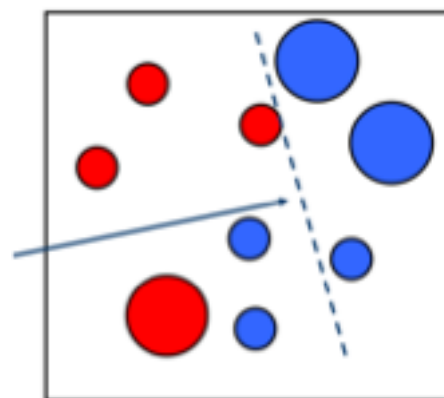
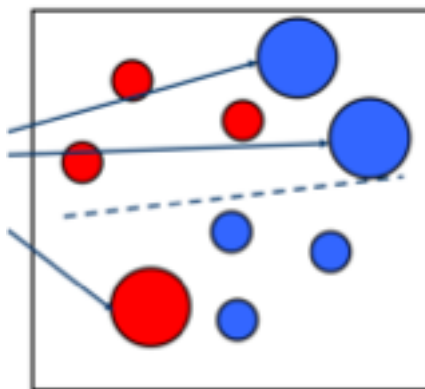
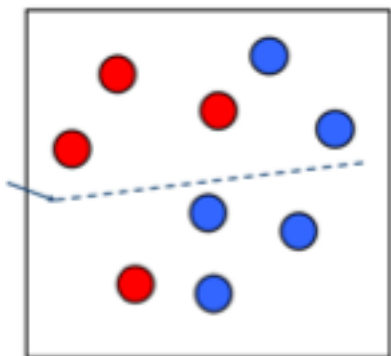
$$\varepsilon_j = \sum_{i=1}^m D_t(i) [y_i \neq h_j(x_i)]$$

• 分类器权重: $\alpha_t = \frac{1}{2} \ln \frac{1 - \varepsilon_t}{\varepsilon_t}$

• 更新样本权重: $D_{t+1}(i) = \frac{D_t(i) \exp[-\alpha_t y_i h_t(x_i)]}{Z_t}$

输出强分类器: $\text{sign} \left(H(x) = \sum_{t=1}^T \alpha_t h_t(x) \right)$

3个分类器的综合



- 如何理解:

$$\varepsilon_j = \sum_{i=1}^m D_i(i)[y_i \neq h_j(x_i)]$$

事实上是错误分类率

$$\alpha_t = \frac{1}{2} \ln \frac{1 - \varepsilon_t}{\varepsilon_t}$$

错误分类率越小，模型权重越大

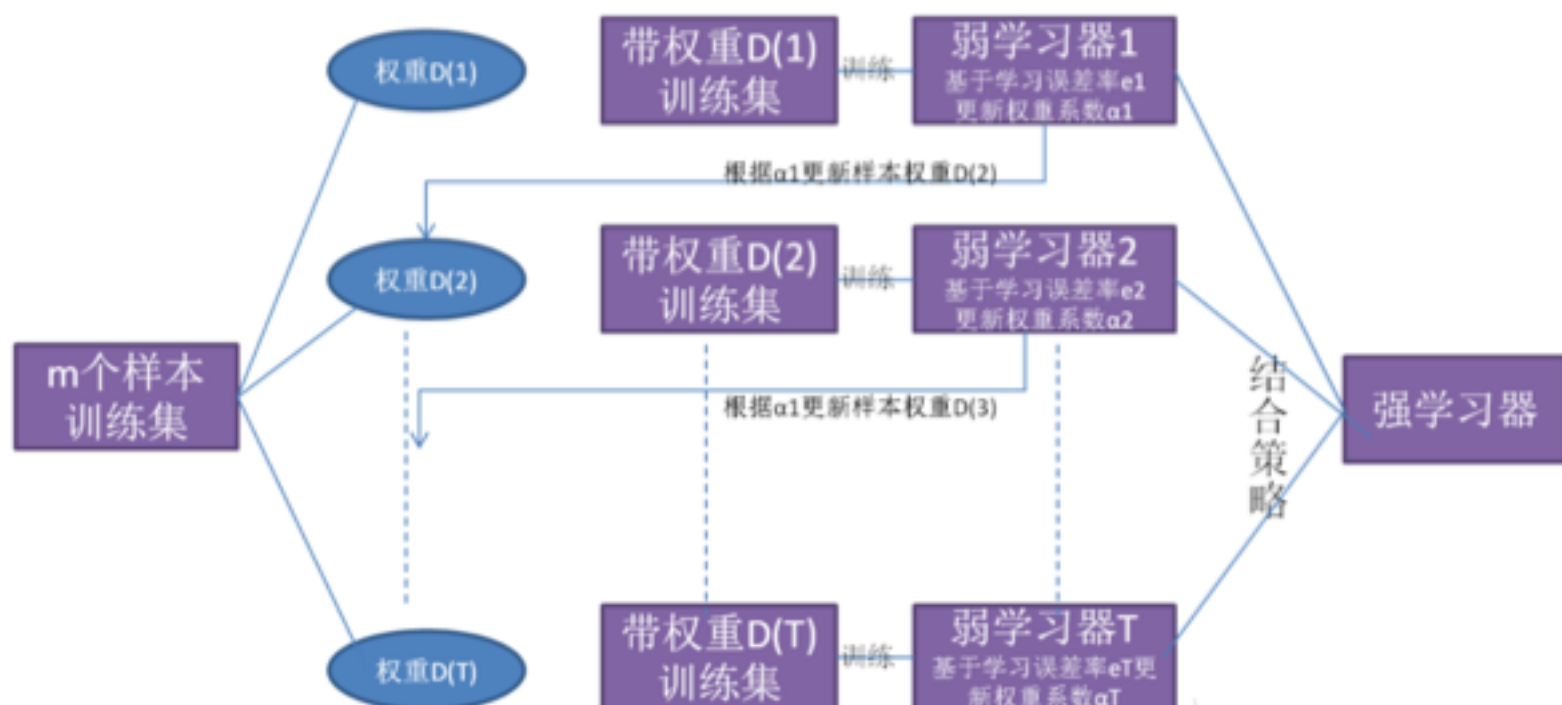
- 权重更新

$$D_{t+1}(i) = \frac{D_t(i) \exp[-\alpha_t y_i h_t(x_i)]}{Z_t}$$

样本分类错误，权重越大，反之越小

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

强分类器为多个分类器的加和。串行



- 统计机器学习例子:

例 8.1 给定如表 8.1 所示训练数据. 假设弱分类器由 $x < v$ 或 $x > v$ 产生, 其阈值 v 使该分类器在训练数据集上分类误差率最低. 试用 AdaBoost 算法学习一个强分类器.

表 8.1 训练数据表

序号	1	2	3	4	5	6	7	8	9	10
x	0	1	2	3	4	5	6	7	8	9
y	1	1	1	-1	-1	-1	1	1	1	-1

表 8.1 训练数据表

序号	1	2	3	4	5	6	7	8	9	10
x	0	1	2	3	4	5	6	7	8	9
y	1	1	1	-1	-1	-1	1	1	1	-1

(a) 在权值分布为 D_1 的训练数据上, 阈值 ν 取 2.5 时分类误差率最低, 故基本分类器为

$$G_1(x) = \begin{cases} 1, & x < 2.5 \\ -1, & x > 2.5 \end{cases}$$

表 8.1 训练数据表

序号	1	2	3	4	5	6	7	8	9	10
x	0	1	2	3	4	5	6	7	8	9
y	1	1	1	-1	-1	-1	1	1	1	-1

(b) $G_1(x)$ 在训练数据集上的误差率 $e_1 = P(G_1(x_i) \neq y_i) = 0.3$.

(c) 计算 $G_1(x)$ 的系数: $\alpha_1 = \frac{1}{2} \log \frac{1-e_1}{e_1} = 0.4236$.

(d) 更新训练数据的权值分布:

$$D_2 = (w_{21}, \dots, w_{2i}, \dots, w_{210})$$

$$w_{2i} = \frac{w_{1i}}{Z_1} \exp(-\alpha_1 y_i G_1(x_i)), \quad i = 1, 2, \dots, 10$$

$$D_2 = (0.0715, 0.0715, 0.0715, 0.0715, 0.0715, 0.0715, \\ 0.1666, 0.1666, 0.1666, 0.0715)$$

$$f_1(x) = 0.4236 G_1(x)$$

X	0	1	2	3	4	5	6	7	8	9
Y	1	1	1	-1	-1	-1	1	1	1	-1
w	0.0715	0.0715	0.0715	0.0715	0.0715	0.0715	0.1666	0.1666	0.1666	0.0715

对 $m=2$,

(a) 在权值分布为 D_2 的训练数据上, 阈值 v 是 8.5 时分类误差率最低, 基本分类器为

$$G_2(x) = \begin{cases} 1, & x < 8.5 \\ -1, & x > 8.5 \end{cases}$$

(b) $G_2(x)$ 在训练数据集上的误差率 $e_2 = 0.2143$.

(c) 计算 $\alpha_2 = 0.6496$.

(d) 更新训练数据权值分布:

$$D_3 = (0.0455, 0.0455, 0.0455, 0.1667, 0.1667, 0.1667, \\ 0.1060, 0.1060, 0.1060, 0.0455)$$

$$f_2(x) = 0.4236G_1(x) + 0.6496G_2(x)$$

分类器 $\text{sign}[f_2(x)]$ 在训练数据集上有 3 个误分类点.

对 $m=3$,

(a) 在权值分布为 D_3 的训练数据上, 阈值 v 是 5.5 时分类误差率最低, 基本分类器为

$$G_3(x) = \begin{cases} 1, & x > 5.5 \\ -1, & x < 5.5 \end{cases}$$

(b) $G_3(x)$ 在训练样本集上的误差率 $e_3 = 0.1820$.

(c) 计算 $\alpha_3 = 0.7514$.

(d) 更新训练数据的权值分布:

$$D_4 = (0.125, 0.125, 0.125, 0.102, 0.102, 0.102, 0.065, 0.065, 0.065, 0.125)$$

$$f_3(x) = 0.4236G_1(x) + 0.6496G_2(x) + 0.7514G_3(x)$$

分类器 $\text{sign}[f_3(x)]$ 在训练数据集上误分类点个数为 0.

于是最终分类器为

$$G(x) = \text{sign}[f_3(x)] = \text{sign}[0.4236G_1(x) + 0.6496G_2(x) + 0.7514G_3(x)]$$