



模式识别与机器学习

集成学习

邹腊梅

QQ: 156685941

科技楼1008



前言

12.1 Bagging算法

12.2 AdaBoost算法

12.3 Bagging 和boosting的比较与实例分析

12.4 决策树（见下一ppt）

12.4 随机森林（见下一ppt）

分类

- ❖ 决策树分类:
 - ID3
 - C4.5
- ❖ 贝叶斯分类
- ❖ 后向传播分类
- ❖ 其它分类

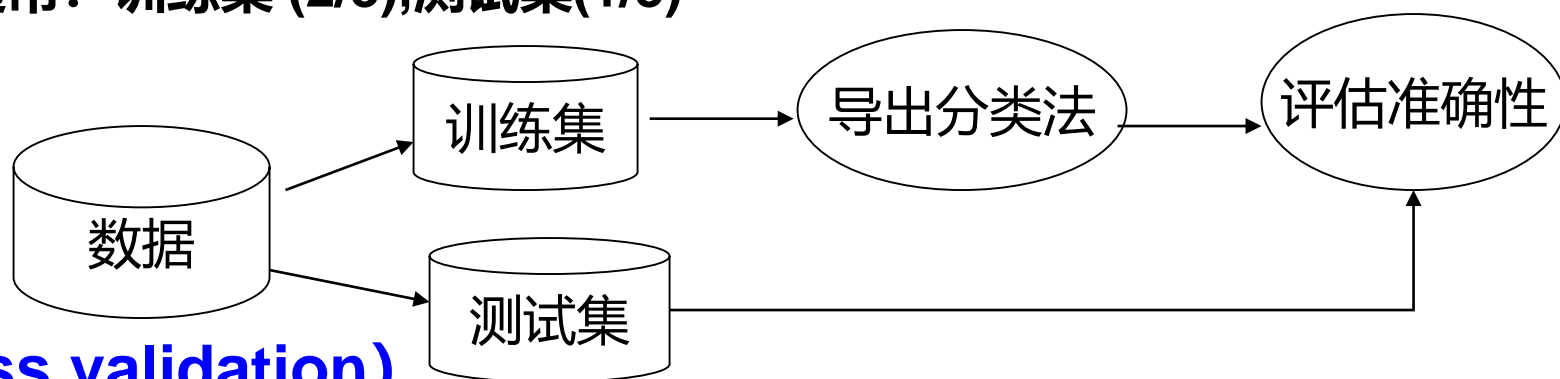
分类法的准确性

❖ 评估分类法的准确率

■ 保持 (holdout) / 留出法

(1) 划分为两个独立的数据集, 通常: 训练集 (2/3), 测试集 (1/3)

(2) 变形: 随机子选择

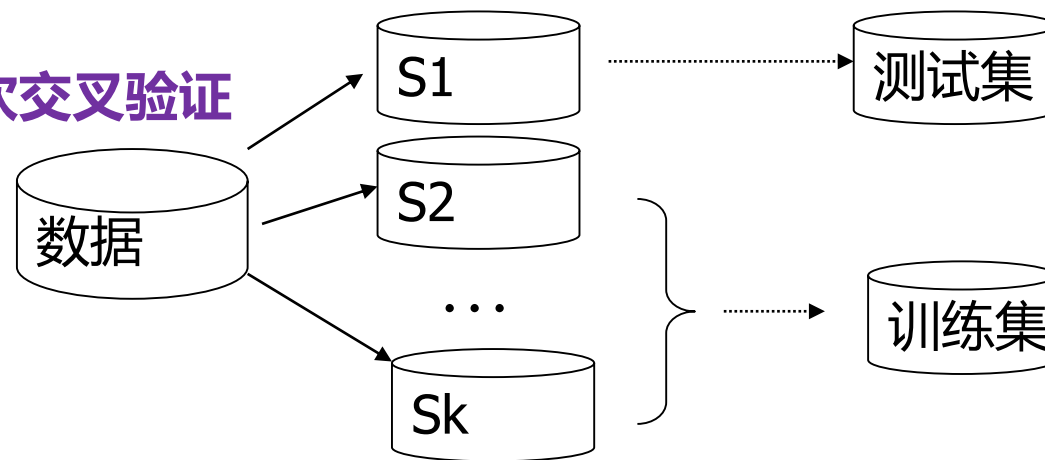


■ K-次交叉验证 (k-fold cross validation)

(1) 将数据集分为k个子集;

(2) 用 $k-1$ 个子集作训练集, 1 个子集作测试集, 然后 **k次交叉验证**

根据训练集训练出模型或假设函数, 把这个模型放到测试集上获得分类率, 计算k次分类率的平均值, 作为真实分类率。



分类法的准确性

提高分类法的准确率

❖ Bagging

❖ Boosting

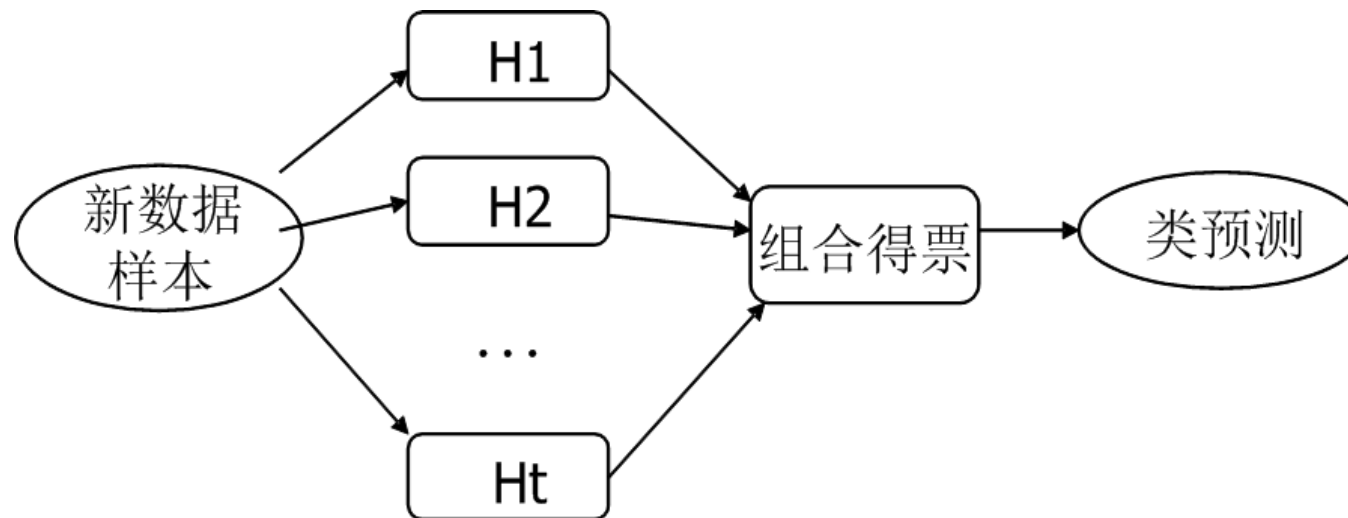
都是将已有的分类或回归算法通过一定的方式组合起来，形成一个性能更加强大的分类器，更准确的说是一种分类算法的组装方法，即将弱分类器组装成强分类器。

三个臭皮匠顶个诸葛亮

❖ Bagging基本思想:

- 给定一个弱学习算法，和一个训练集；
- 单个弱学习算法准确率不高；
- 将该学习算法使用多次，得出预测函数序列,进行投票；
- 最后结果准确率将得到提高.

12.1 Bagging



❖ 算法:

For $t = 1, 2, \dots, T$ *Do*

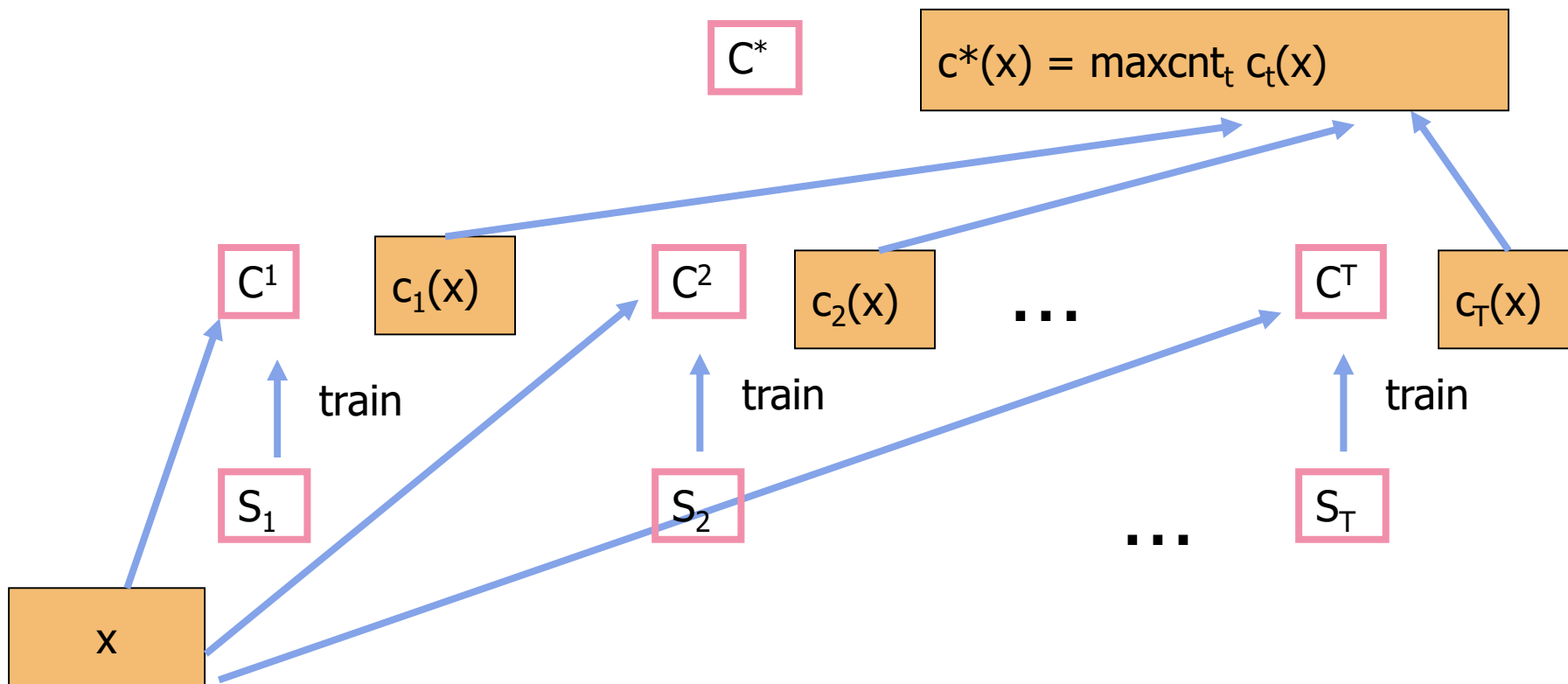
从数据集S中取样 (放回选样)

训练得到模型 H_t

对未知样本X分类时,每个模型 H_t 都得出一个分类, 得票最高的即为未知样本X的分类

❖ 也可通过得票的平均值用于连续值的预测

12.1 Bagging



训练各子分类器模型 $C_i(x)$ 时，各样本子集采用“放回选样”的方式

12.1 Bagging



- ❖ **Bagging要求“不稳定”的分类方法；
比如：决策树，神经网络算法**
- ❖ **不稳定：数据集的小的变动能够使得分类结果显著的变动。**
- ❖ **“The vital element is the instability of the prediction method. If perturbing the learning set can cause significant changes in the predictor constructed, then bagging can improve accuracy.” (Breiman 1996)**

12.2 Boosting

❖ 背景

probably approximately correct, 概率近似正确

来源于: PAC-Learning Model, Kearns & Valiant 1984 -11

提出问题:

- 强学习算法: 正确率很高的学习算法
- 弱学习算法: 正确率不高, 仅比随机猜测略好 (>0.5)
- 是否可以将弱学习算法提升为强学习算法

最初的boosting算法, Schapire 1989

AdaBoost算法, Freund and Schapire 1995

❖ 基本思想:

- 每个样本都赋予一个权重
- T次迭代, 每次迭代后, 对分类错误的样本加大权重, 使得下一次的迭代更加关注这些样本

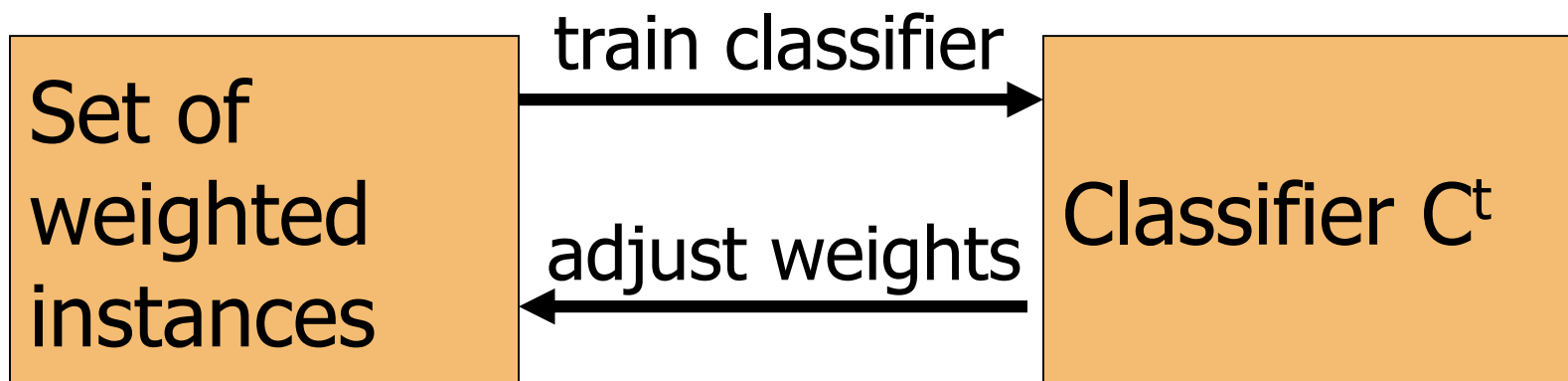
Boosting也要求“不稳定”的分类方法

12.2 Boosting



❖ 过程:

- 在一定的权重条件下训练数据，得出分类器 C^t
- 根据 C^t 的错误率调整权重



❖ AdaBoost

❖ AdaBoost.M1

❖ AdaBoost.M2...

一. 样本

Given: m examples $(x_1, y_1), \dots, (x_m, y_m)$

where $x_i \in X, y_i \in Y = \{-1, +1\}$

x_i 表示 X 中第 i 个元素,

y_i 表示与 x_i 对应元素的属性值, $+1$ 表示 x_i 属于某个分类,
 -1 表示 x_i 不属于某个分类

二. 初始化训练样本 x_i 的权重分布 $D(i) : i=1, \dots, m$;

(1) 若正负样本数目一致, $D_1(i) = 1/m$

(2) 若正负样本数目 m_+ 和 m_- , 则正样本 $D_1(i) = 1/(2m_+)$

负样本 $D_1(i) = 1/(2m_-)$

12.2 Boosting: AdaBoost

Schapire Adaboost Algorithm (续)



三. 训练弱分类器

输入M个样本, 以及基学习算法, 训练轮数T。初始化样本权值分布 D_1

for $t=1, \dots, T$

1. Train learner h_t with **min error** $\varepsilon_t = \Pr_{i \sim D_t}[h_t(x_i) \neq y_i]$

利用分布 D_t 从数据集D中训练分类器 h_t 。计算 h_t 的误差 ε_t

若划分正确, 则不计入误差, 若所有元素都被正确划分, 则误差为0;

若划分错误, 则计入误差。

2. If $\varepsilon_t \geq 0.5$, then break

3. 确定分类器 h_t 权重

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

The weight **Adapts**.
The bigger ε_t becomes,
the smaller α_t becomes.

$$d_t = \sqrt{(1 - \varepsilon_t) / \varepsilon_t}$$

$$\alpha_t = \ln \sqrt{(1 - \varepsilon_t) / \varepsilon_t}$$

4. 更新样本权重值

$$D_{t+1}(i) = D_t(i) \times \begin{cases} e^{-\alpha_t} & \text{if } h_t(x_i) = y_i \\ e^{\alpha_t} & \text{if } h_t(x_i) \neq y_i \end{cases}$$

错分样本的权重变大

$$\varepsilon_t < 0.5, \quad \frac{1 - \varepsilon_t}{\varepsilon_t} > 1$$

5. 更新样本权重分布

(样本权重 D_{t+1} 归一化)

$$D_{t+1}(i) = \frac{D_{t+1}(i)}{Z_t} \quad \text{其中, } Z_t = \sum_i D_{t+1}(i)$$

or

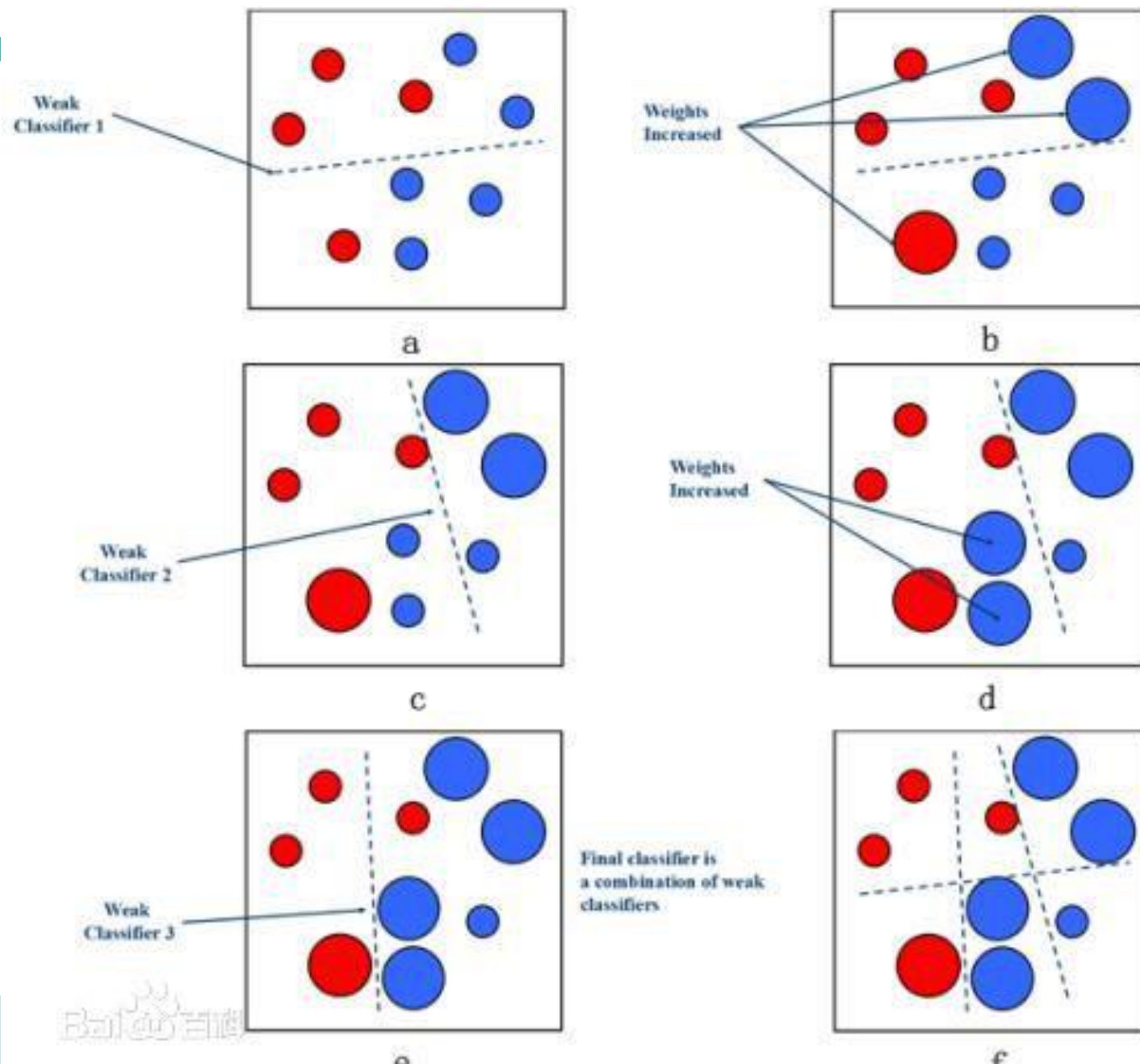
$$D_{t+1}(i) = \begin{cases} D_t(i) & \text{if } h_t(x_i) = y_i \\ D_t(i) \frac{1 - \varepsilon_t}{\varepsilon_t} & \text{if } h_t(x_i) \neq y_i \end{cases}$$

end for

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

终止条件: 1、组合分类器H(x)在训练集上无错分样本;
2、达到学习轮数

12.2 Boosting: AdaBoost



AdaBoost的特点

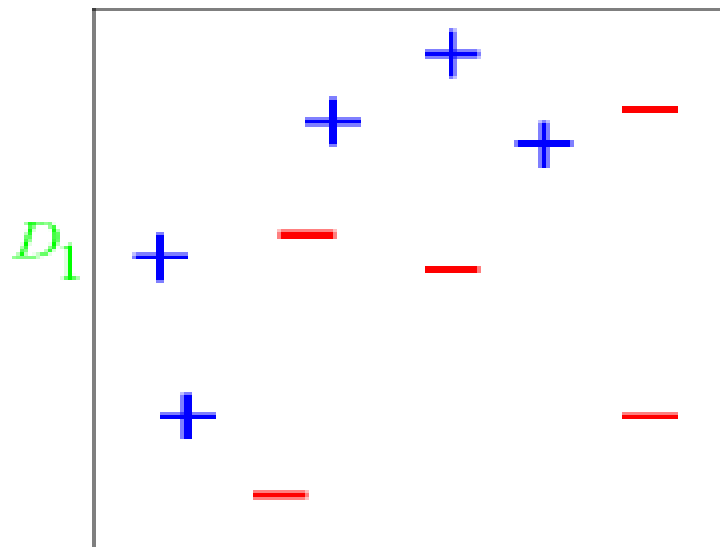
- 1) 每次迭代改变的是样本的分布 (re-weight), 而不是重复采样
- 2) 样本分布的改变取决于样本是否被正确分类
总是分类正确的样本权值低, 分类错误的样本权值高(通常是边界附近的样本)
- 3) 最终的结果是弱分类器的加权组合
弱分类器的权值表示该弱分类器的性能

AdaBoost的优点

- 1) adaboost是一种高精度分类器
- 2) adaboost算法提供的是框架, 可以使用各种方法构建子分类器
- 3) 当使用简单分类器时, 计算出的结果是可以理解的。而且弱分类器构造极其简单
- 4) 简单, 不用做特征筛选

Adaboost 实例详解

例：adaboost 的实现过程示例



图中，“+”和“-”分别表示两种类别的训练数据。

假设弱分类器由水平或者垂直的直线（ $x < v$ 或 $x > v$ 或 $y < v$ 或 $y > v$ ）产生，具体的直线及阈值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。

两类样本数相同。算法**初始化样本权值分布**为均匀分布 \mathbf{D}_1 。每个点的初始权值为**0.1**。

遍历不同直线进行划分时分类的误差率。选取使训练数据集上分类误差率最低的直线作为 **h_1** 划分。采用 **h_1** 划分后，有三个点划分错误，此时**分类误差率**为：

$$\varepsilon_{t=1} = \sum_{i=1}^m D_t \| y_i \neq h_t(x_i) \| = 0.1 + 0.1 + 0.1 = 0.3$$

$$\varepsilon_1 = 0.3$$

$$\alpha_1 = 0.42$$

确定分类器 h_t 权重： $\alpha_1 = \frac{1}{2} \ln \left(\frac{1 - \varepsilon_1}{\varepsilon_1} \right) = \frac{1}{2} \ln \left(\frac{1 - 0.3}{0.3} \right) = 0.42$

$$\alpha_t = \ln \sqrt{(1 - \varepsilon_t) / \varepsilon_t}$$

更新样本权重值：对于分类正确的**7**个点，其权值保持不变，为**0.1**；对于分类错误的**3**个点，将其权值变大，其权值为

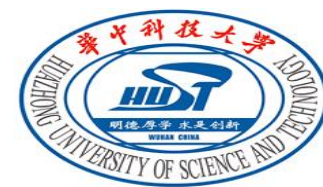
$$D_{2(i)} \frac{1 - \varepsilon_1}{\varepsilon_1} = 0.1 \left(\frac{1 - 0.3}{0.3} \right) = 0.2333$$

$$D_{t+1}(i) = \begin{cases} D_t(i) & \text{if } h_t(x_i) = y_i \\ D_t(i) \frac{1 - \varepsilon_t}{\varepsilon_t} & \text{if } h_t(x_i) \neq y_i \end{cases}$$

样本权值归一化，得到新的**样本权值分布 D_2**

$$D_{t+1}(i) = D_{t+1}(i) / \sum_i D_{t+1}(i)$$

Adaboost 实例详解 (续)



简化的权值更新策略

$$\alpha_t = \ln \sqrt{(1 - \varepsilon_t) / \varepsilon_t}$$

$$D_{t+1}(i) = \begin{cases} D_t(i) & \text{if } h_t(x_i) = y_i \\ D_t(i) \frac{1 - \varepsilon_t}{\varepsilon_t} & \text{if } h_t(x_i) \neq y_i \end{cases}$$

$$\varepsilon_1 = 0.3$$

$$\alpha_1 = 0.42$$

更新权值

权值归一化
更新权值分布

$$\begin{aligned} \text{权值归一化: } 0.233 / (0.233 * 3 + 0.1 * 7) &= 0.1665 \\ 0.1 / (0.233 * 3 + 0.1 * 7) &= 0.0714 \end{aligned}$$

$$D_{t+1}(i) = D_t(i) / \sum_i D_t(i)$$

Adaboost 实例详解(续)

t=2, 选取使训练数据集D₂上分类误差率最低的直线作为h₂划分。利用h₂划分后, 有三个点划分错了, 计算**分类误差率**: $\varepsilon_2 = (0.0714 + 0.0714 + 0.0714) = 0.2142$

$$\alpha_2 = \frac{1}{2} \ln \left(\frac{1 - \varepsilon_2}{\varepsilon_2} \right) = \frac{1}{2} \ln \left(\frac{1 - 0.2142}{0.2142} \right) = 0.65$$

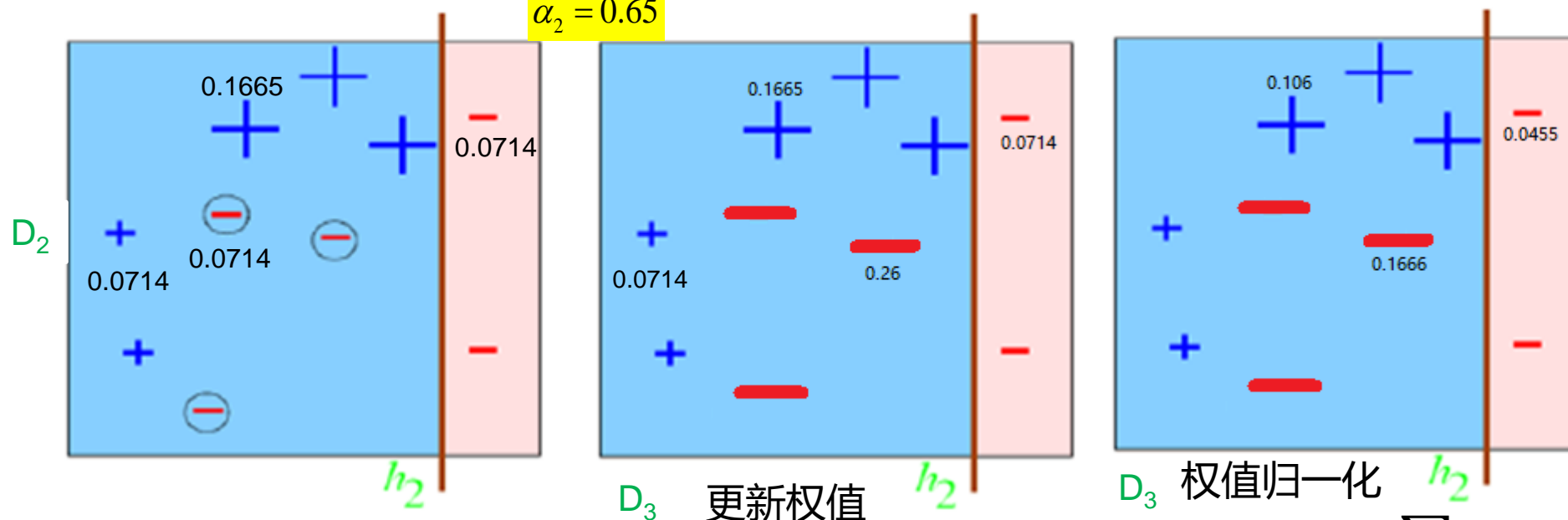
对于分类错误的点, 其权值为: $D_2(i) \frac{1 - \varepsilon_2}{\varepsilon_2} = 0.0714 \left(\frac{1 - 0.2142}{0.2142} \right) = 0.26$

$$\varepsilon_2 = 0.21$$

$$\alpha_2 = 0.65$$

$$\alpha_t = \ln \sqrt{(1 - \varepsilon_t) / \varepsilon_t}$$

$$D_{t+1}(i) = \begin{cases} D_t(i) & \text{if } h_t(x_i) = y_i \\ D_t(i) \frac{1 - \varepsilon_t}{\varepsilon_t} & \text{if } h_t(x_i) \neq y_i \end{cases}$$



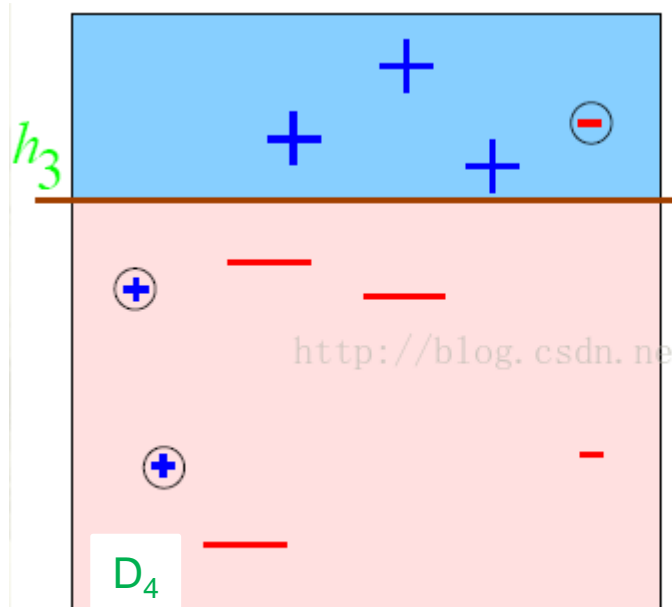
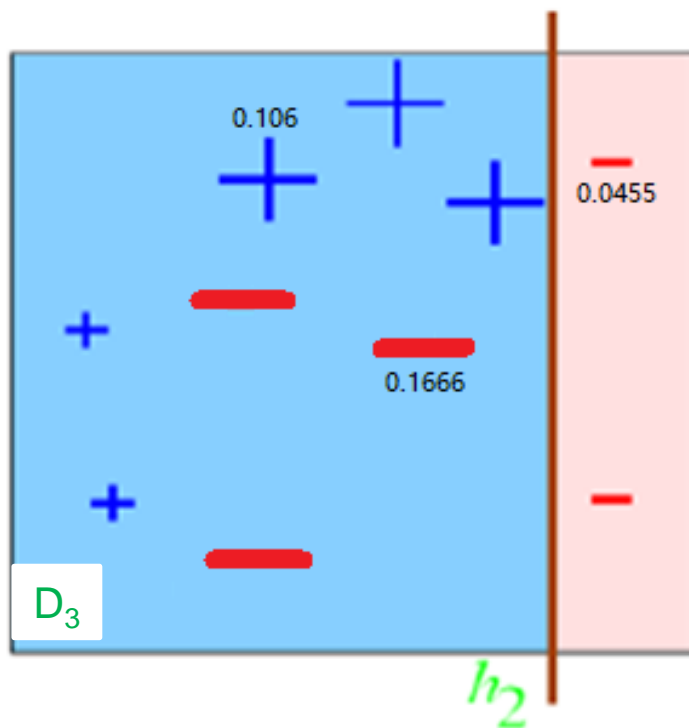
$$D_{t+1}(i) = D_t(i) / \sum_i D_t(i)$$

Adaboost 实例详解

$t=3$, 选取使训练数据集 D_3 上分类误差率最低的直线作为 h_3 划分。利用 h_3 进行划分后, 有三个点划分错了, 计算**分类误差率**: $\varepsilon_3=(0.0455+0.0455+0.0455)=0.1365$

$$\alpha_3 = \frac{1}{2} \ln \left(\frac{1 - \varepsilon_3}{\varepsilon_3} \right) = \frac{1}{2} \ln \left(\frac{1 - 0.1365}{0.1365} \right) = 0.9225$$

对于分类错误的点, 其权值为: $D_3(i) \frac{1 - \varepsilon_3}{\varepsilon_3} = 0.0455 \left(\frac{1 - 0.1364}{0.1364} \right) = 0.2879$



$$\varepsilon_3 = 0.14$$

$$\alpha_3 = 0.92$$

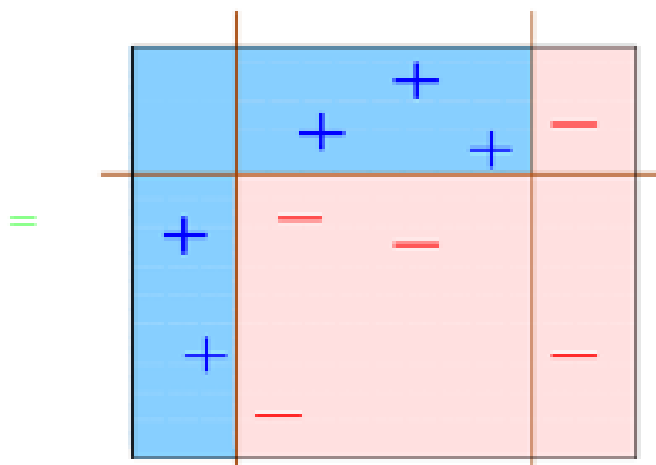
终止条件：

1. 组合分类器 $H(x)$ 在训练集上无错分样本；
2. 达到学习轮数

$$H_{\text{final}} = \text{sign} \left(0.42 \begin{array}{|c|c|} \hline \text{blue} & \text{red} \\ \hline \end{array} + 0.65 \begin{array}{|c|c|} \hline \text{blue} & \text{red} \\ \hline \end{array} + 0.92 \begin{array}{|c|c|} \hline \text{blue} & \text{red} \\ \hline \end{array} \right)$$

$$H(x) = \text{sign} \left(\sum_{t=1}^3 \alpha_t h_t(x) \right)$$

此时，训练集样本都正确分类



每个区域是属于哪个属性，由这个区域所在分类器的权值综合决定。

如：左下角区域，属于蓝色分类区的权重为**h1** 中的**0.42**和**h2** 中的**0.65**，其和为**1.07**；属于淡红色分类区域的权重为**h3** 中的**0.92**；属于淡红色分类区的权重小于属于蓝色分类区的权值，因此左下角属于蓝色分类区。

从结果图中看，即使是简单的分类器，组合起来也能获得很好的分类效果。

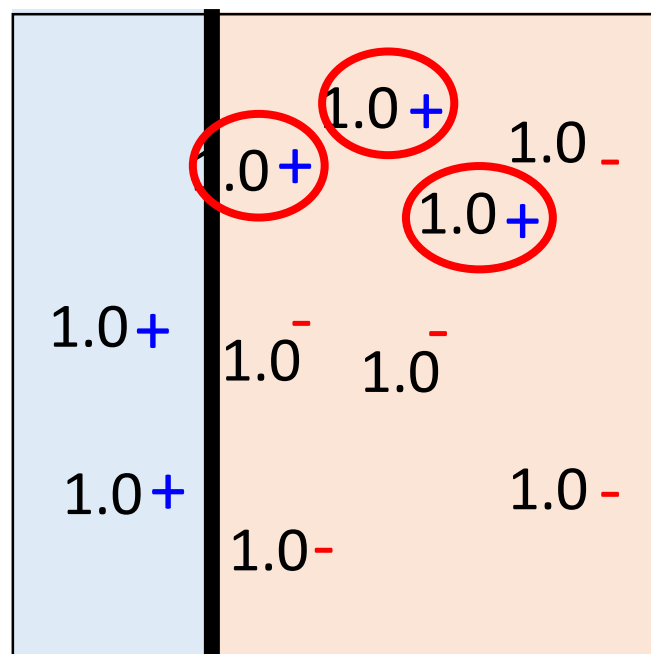
Adaboost 实例详解(续)

采用标准的权值更新规则

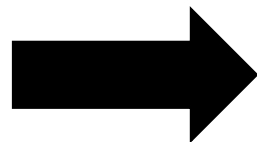
Toy Example

T=3, weak classifier = decision stump

• t=1



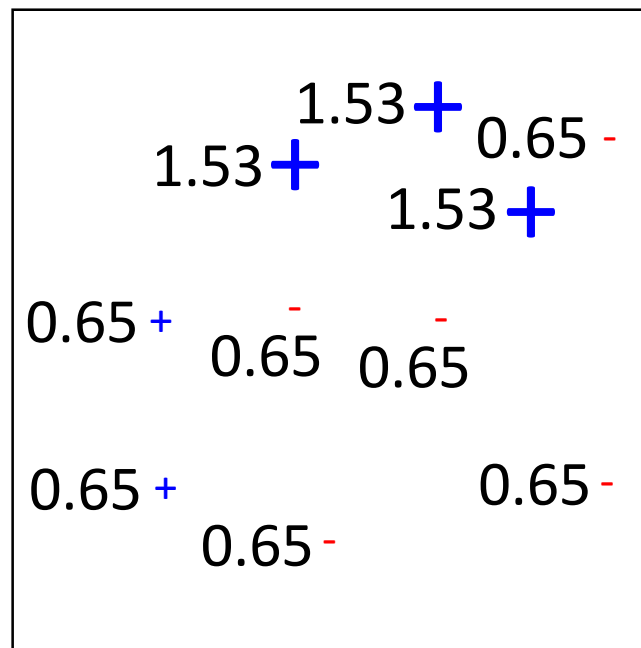
$h_1(x)$



$$\varepsilon_1 = 0.30$$

$$d_1 = 1.53$$

$$\alpha_1 = 0.42$$



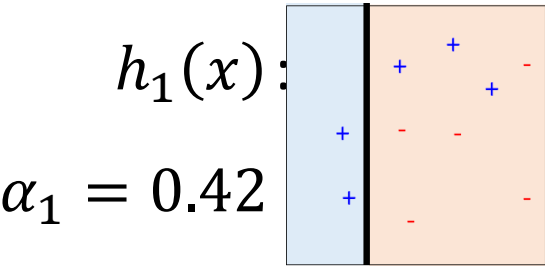
$$D_{t+1}(i) = D_t(i) \times \begin{cases} e^{-\alpha_t} & \text{if } h_t(x_i) = y_i \\ e^{\alpha_t} & \text{if } h_t(x_i) \neq y_i \end{cases}$$

$$D_{t+1}(i) = D_{t+1}(i) / \sum_i D_{t+1}(i)$$

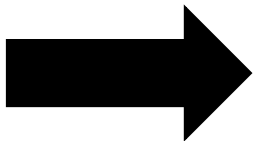
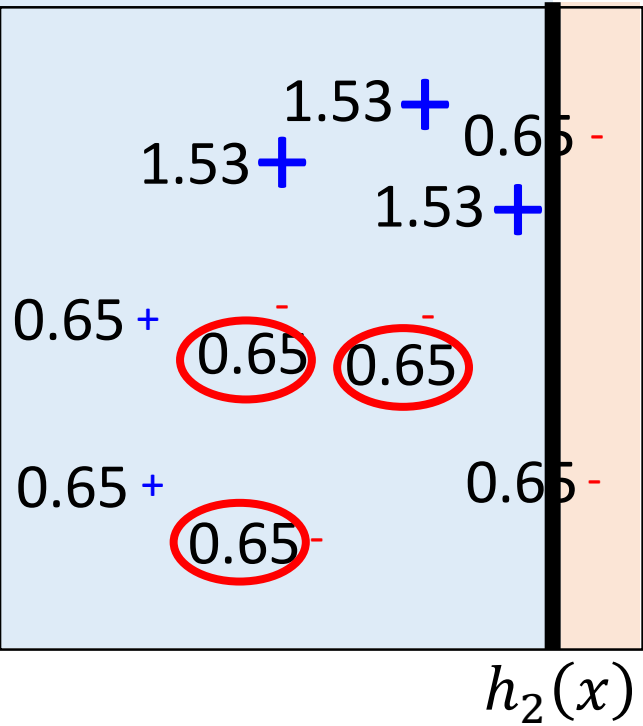
Toy Example

T=3, weak classifier = decision stump

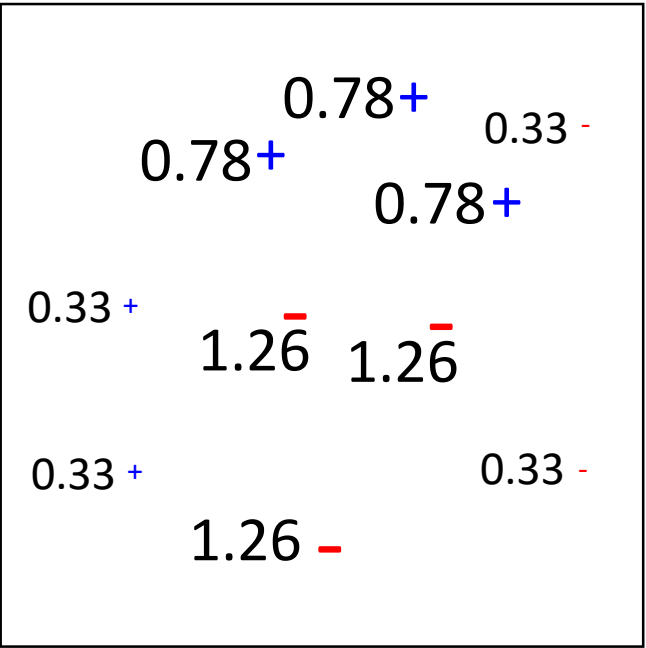
• t=2



$$d_t = \sqrt{(1 - \varepsilon_t) / \varepsilon_t}$$
$$\alpha_t = \ln \sqrt{(1 - \varepsilon_t) / \varepsilon_t}$$



$\varepsilon_2 = 0.21$
 $d_2 = 1.94$
 $\alpha_2 = 0.66$



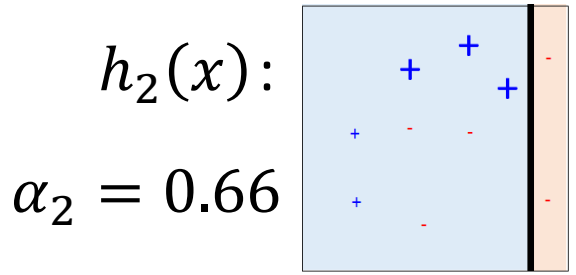
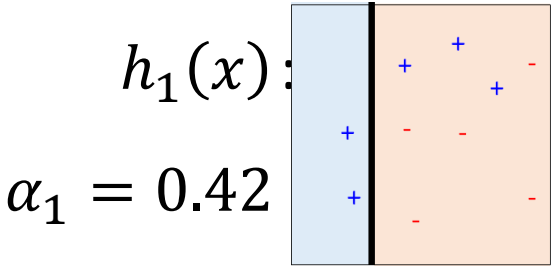
$$D_{t+1}(i) = D_t(i) \times \begin{cases} e^{-\alpha_t} & \text{if } h_t(x_i) = y_i \\ e^{\alpha_t} & \text{if } h_t(x_i) \neq y_i \end{cases}$$

$$D_{t+1}(i) = D_{t+1}(i) / \sum_i D_{t+1}(i)$$

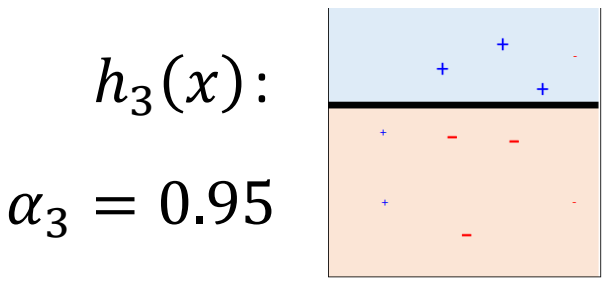
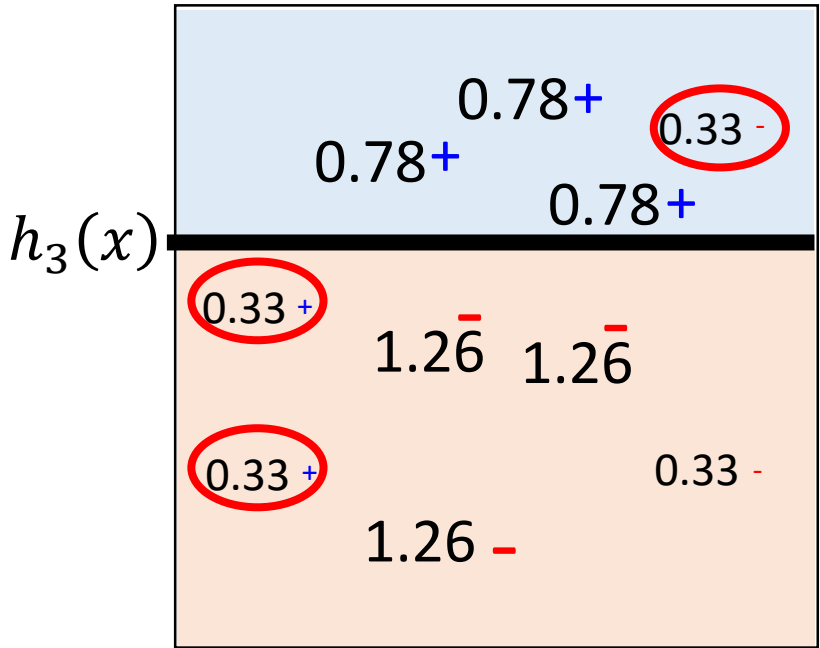
Toy Example

T=3, weak classifier = decision stump

• t=3



$$d_t = \sqrt{(1 - \varepsilon_t) / \varepsilon_t}$$
$$\alpha_t = \ln \sqrt{(1 - \varepsilon_t) / \varepsilon_t}$$



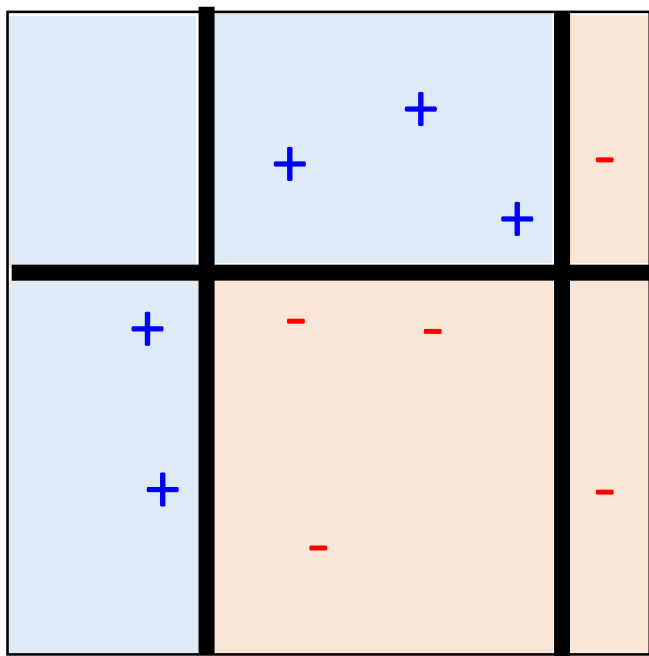
$$D_{t+1}(i) = D_t(i) \times \begin{cases} e^{-\alpha_t} & \text{if } h_t(x_i) = y_i \\ e^{\alpha_t} & \text{if } h_t(x_i) \neq y_i \end{cases}$$

$$D_{t+1}(i) = D_{t+1}(i) / \sum_i D_{t+1}(i)$$

Toy Example

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

$$\text{sign}(0.42 \begin{array}{|c|} \hline \text{[Diagram 1]} \\ \hline \end{array} + 0.66 \begin{array}{|c|} \hline \text{[Diagram 2]} \\ \hline \end{array} + 0.95 \begin{array}{|c|} \hline \text{[Diagram 3]} \\ \hline \end{array})$$



Adaboost权值调整的原因



注意到算法最后的表到式为 $H(x) = \text{sign}(\sum_{t=1}^T \alpha_t h_t(x))$ ，其中 α_t 表示的权值，是由 $\alpha_t = \ln \sqrt{(1 - \varepsilon_t) / \varepsilon_t}$ 得到的，是关于误差的表达式。

提高错误点的权值，当下一次分类器再次分错了这些点之后，会提高整体的错误率，这样就导致 α_t 变的很小，最终导致这个分类器在整个混合分类器的权值变低。

即是，这种算法让优秀的分类器占整体的权值更高，而“挫”的分类器权值更低。符合常理。

另外， $h(x)$ 是1和-1，不是1和0

AdaBoost总结

AdaBoost算法的应用:

- 1) 用于二分类或多分类的应用场景
- 2) 用于做分类任务的**baseline**, 无脑化, 简单, 不会严重**overfitting**, 不用调分类器
- 3) 用于特征选择 (**feature selection**)
- 4) **Boosting** 框架用于对**badcase**的修正

AdaBoost算法实现简单, 应用也很简单方便的算法, 只需增加新的分类器, 不需变动原有分类器。

$$\alpha_t = \ln \sqrt{(1 - \varepsilon_t) / \varepsilon_t}$$

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

As we have more and more h_t (T increases), $H(x)$ achieves smaller and smaller error rate on training data.

Adaboost算法通过组合弱分类器而得到强分类器, 具有**分类错误率上界随着训练增加而稳定下降 (可证)**, **不容易过拟合**等性质, 适合于在各种分类识别场景下的应用。

12.3 Bagging 和Boosting的比较与实例分析

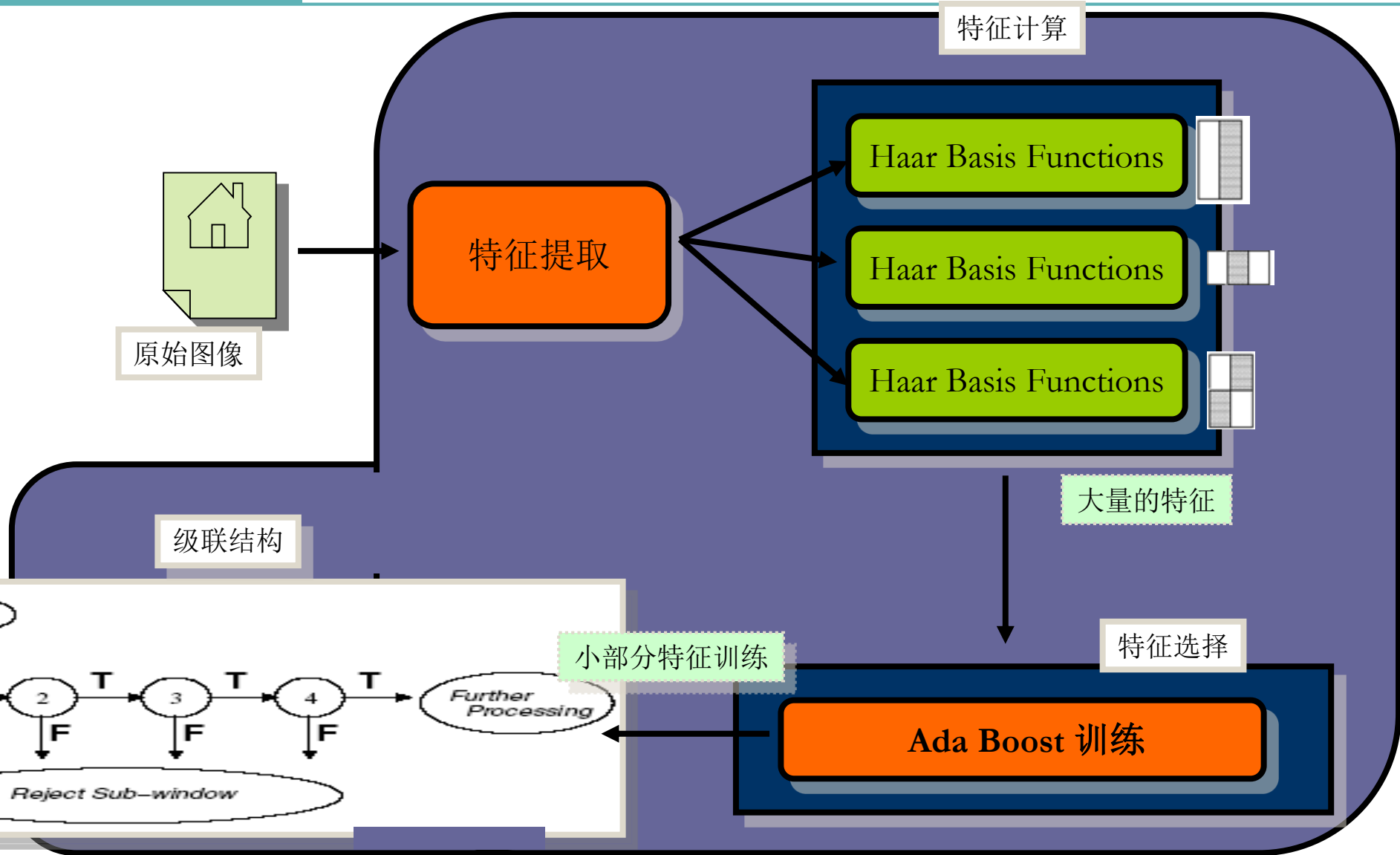
❖ 训练集:

- Bagging: 随机选择,各轮训练集相互独立
- Boosting:各轮训练集并不独立,它的选择与前轮的学习结果有关

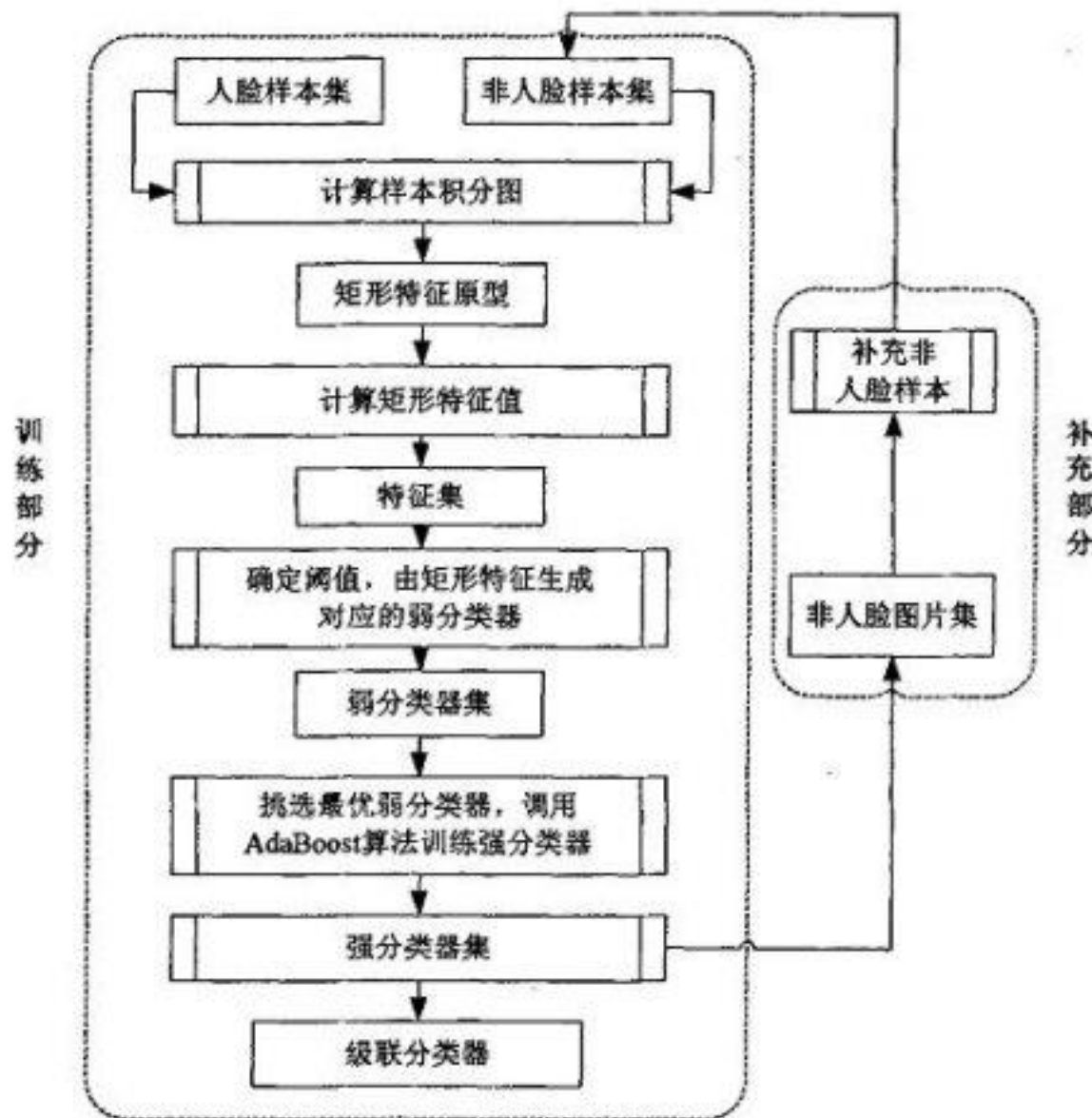
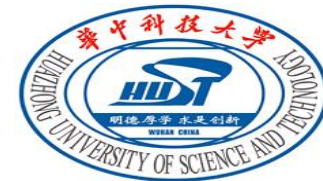
❖ 预测函数:

- Bagging: 没有权重;可以并行生成
- Boosting:有权重;只能顺序生成

12.3 实例分析



12.3 实例分析



训练系统分为“训练部分”和“补充部分”，1-4为训练部分，5为补充部分。

1、以样本集为输入，在给定的矩形特征原型下，计算并获得矩形特征集；

2、以特征集为输入，根据给定的弱学习算法，确定阈值，将特征与弱分类器一一对应，获得弱分类器集；

3、以弱分类器集为输入，在训练检出率和误判率限制下，使用AdaBoost算法挑选最优的弱分类器构成强分类器；

4、以强分类器集为输入，将其组合为级联分类器；

5、以非人脸图片集为输入，组合强分类器为临时的级联分类器，筛选并补充非人脸样本。

12.3 实例分析—矩形特征

- ❖ 2001年，Viola和Jones将Adaboost应用于人脸检测，在保证检测率的同时，首次使得人脸检测达到了**实时的速度**。
- ❖ 为保证Adaboost分类器的分类能力，选择的弱分类器一般都应该尽可能简单。
- ❖ 在基于Adaboost的人脸检测系统中，**每个弱分类器是对图像某个特征值的判断**，常用的特征是一种基于积分图计算的**Haar-like特征**。
- ❖ 在Viola的方法中，使用矩形特征作为分类的依据，称为**Haar特征**。典型的矩阵特征由2到4个矩形组成，分别对应于边界、细线/棒或者对角线特征。

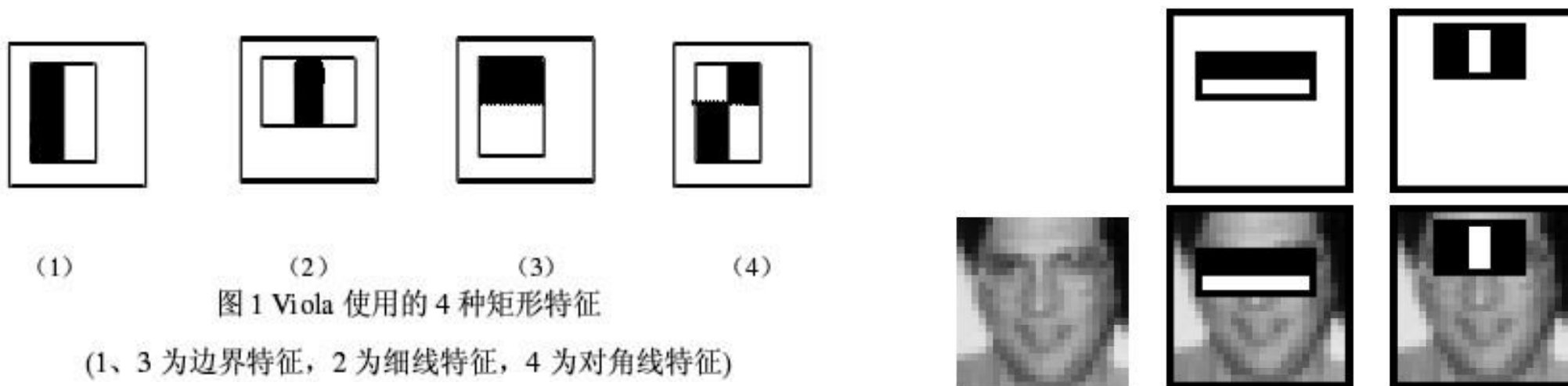


图1 Viola 使用的4种矩形特征

(1、3 为边界特征，2 为细线特征，4 为对角线特征)

[1]. Viola P, Jones M. Rapid object detection using a boosted cascade of simple features[C]//Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on. IEEE, 2001, 1: 1-511-1-518 vol. 1.

12.3 实例分析—矩形特征

- ❖ 后来，Lienhart等人提出扩展的Haar-like特征下图所示，每个特征由2~3个矩形组成，分别检测边界、细线、中心特征等。

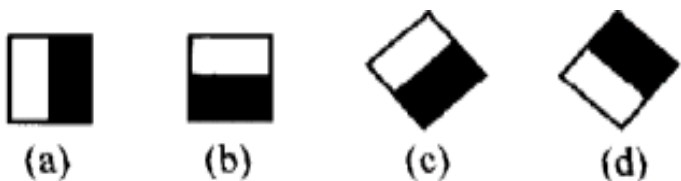


图2 边界特征

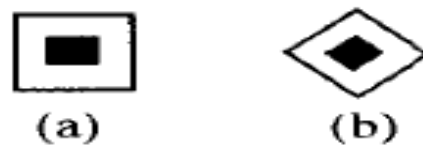


图4 中心特征

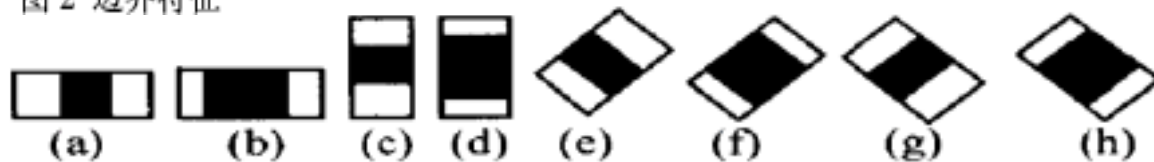


图3 线特征

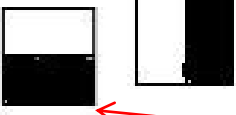
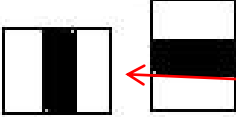

- ❖ 在确定了特征形式后，Harr-like特征的数量就取决于训练样本图像矩阵的大小。

$$\sum_{(s,t)}^{(m,n)} = \sum_{x=1}^{m-s+1} \left[\frac{m-x+1}{s} \right] * \sum_{y=1}^{n-t+1} \left[\frac{n-y+1}{t} \right]$$

m,n样本图像的宽和高，s,t表示haar矩形特征宽和高。公式表示样本图像逐渐缩小，滑动，得到的某一haar特征的总数。

表 2.1 Viola 四类特征在 24×24 子窗口中数量

特征类型	w/h	X/Y	特征数量
A, B	2/1; 1/2	12/24; 24/12	43200×2
C	3/1	8/24	27600
D	2/2	12/12	20736

特征模板	数量
	86400
	55200
	20736
总数	162336

假设训练或检测窗口大小为 $W \times H$ 个像素， w, h 分别为特征原型的长、宽，所示四种特征原型对应的 w/h 分别为：

2/1,
1/2,
3/1,
2/2。

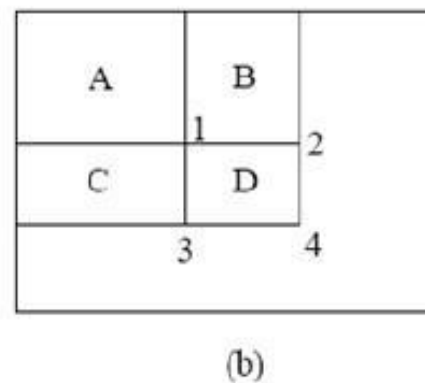
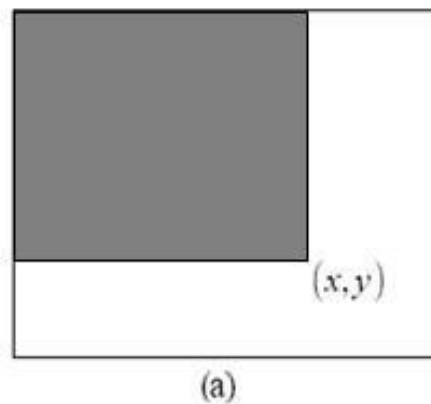
Adaboost 算法通过从大量的 haar 特征中挑选出最优的特征，并将其转换成对应的弱分类器进行分类使用，从而达到对目标进行分类的目的。

12.3 实例分析—积分图

- ❖ 利用矩形特征来计算选取人脸的特征有一种非常快速的算法，称之为积分图。在一张积分图上，点 $i(x,y)$ 的积分值 $ii(x,y)$ 是原图像上该点的上方和左方所有点的亮度值的和。即：

$$ii(x, y) = \sum_{x' < x, y' < y} i(x', y')$$

其中 $ii(x,y)$ 为积分图， $i(x,y)$ 为原始图像， x, y 表示图像的像素坐标。上式表示对 (x,y) 左上角像素求和。



$$D = (4) - (2) - (3) + (1)$$

图6 原始矩形特征的积分图

12.3 实例分析--弱分类器及其选取

- ❖ 一个弱分类器 $h(x, f, p, \theta)$ 由一个特征 f , 阈值 θ 和指示不等号方向的 p 组成:

$$h(x, f, p, \theta) = \begin{cases} 1 & pf(x) < p\theta \\ 0 & \text{其他} \end{cases}$$

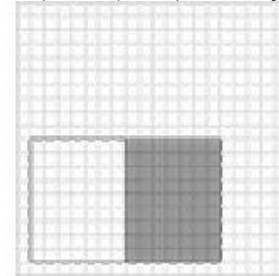
- ❖ 训练一个弱分类器 (特征 f) 就是在当前权重分布的情况下, 确定 f 的最优**阈值**, 使得这个弱分类器 (特征 f) 对所有训练样本的分类误差最低。
- ❖ 对于每个特征 f , 计算所有训练样本的特征值 $f(x)$, 并将其排序。通过扫描一遍排好序的特征值, 可以为这个特征确定一个最优的阈值, 从而训练成一个弱分类器 h 。
- ❖ 选取一个最佳弱分类器, 即是选择那个对所有训练样本 x 的**分类误差**在所有弱分类器 h 中**最低**的那个弱分类器 (所对应的**特征 f**)。

12.3 实例分析--弱分类器及其选取

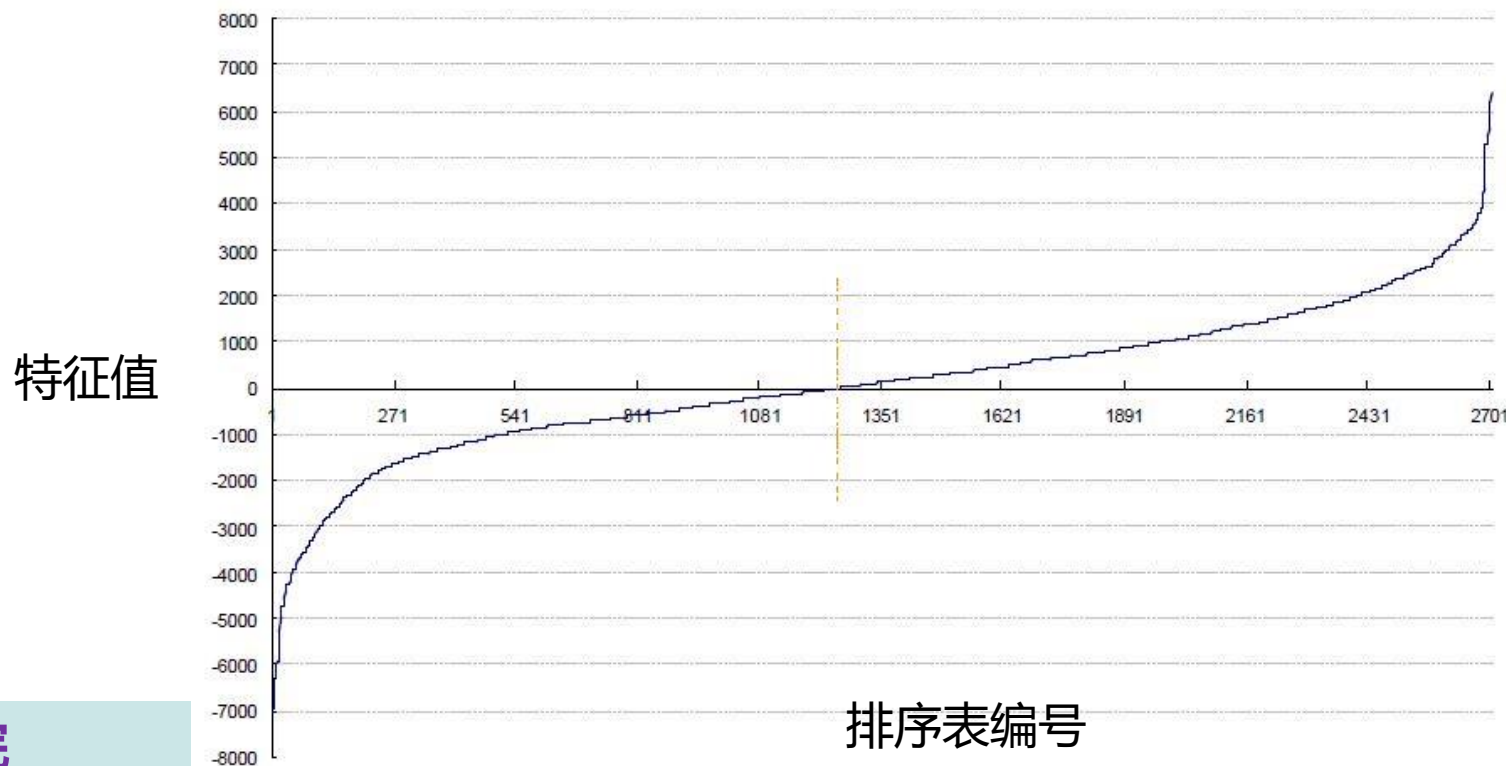
从矩形特征中随机抽取两个特征A和B，这两个特征遍历2,706 个人脸样本和4,381 个非人脸样本，计算每张图像对应的特征值，最后将特征值进行从小到大的排序，并按新的顺序表绘制分布图如下所示：

矩形特征 A 在 20×20 子窗口中位置如右图

参数： 坐标 $(2, 11)-(15, 19)$ (s, t) 条件 $(2, 1)$



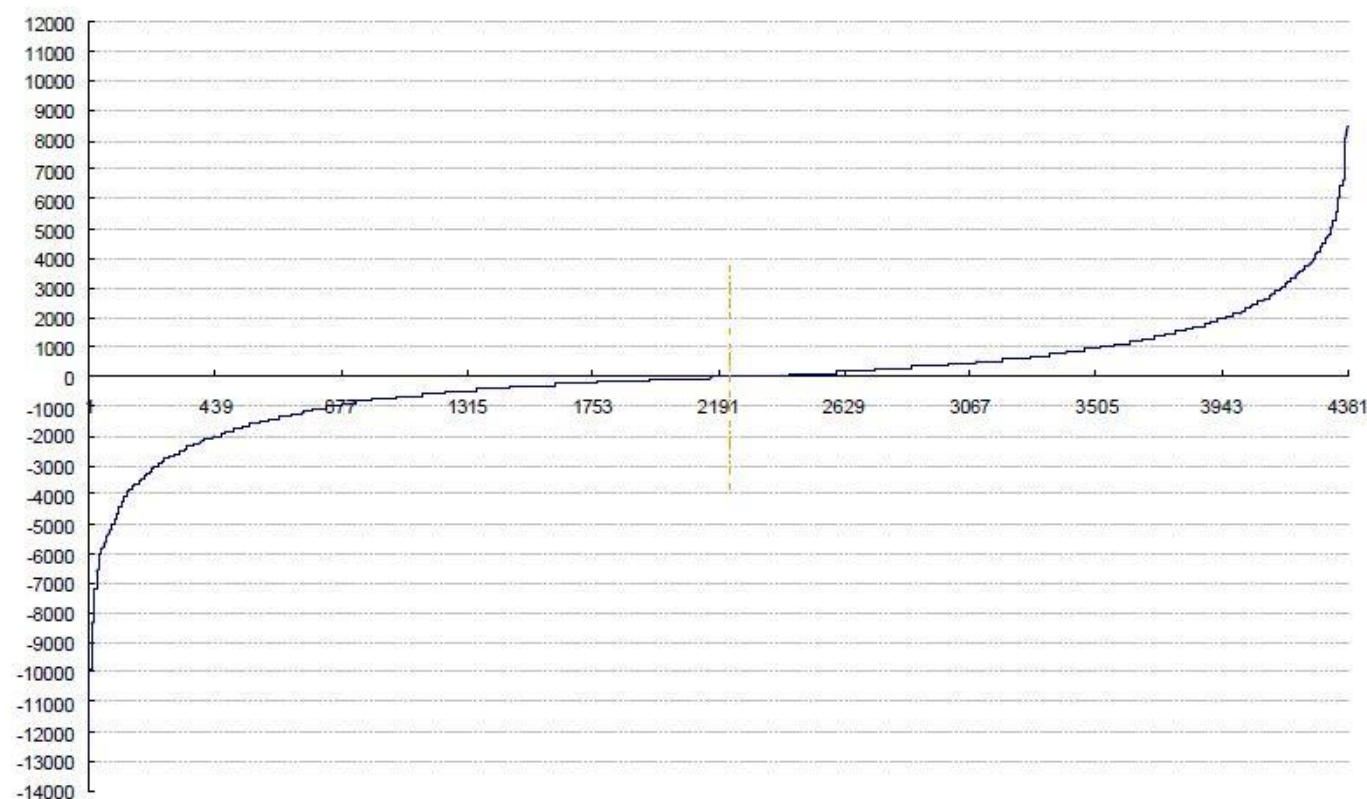
人脸图像特征值分布



2,706 个
人脸样本

12.3 实例分析--弱分类器及其选取

非人脸图像特征值分布



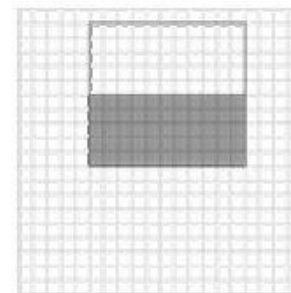
4,381 个
非人脸样本

图 17 矩形特征 A 对人脸和非人脸图像的特征值分布 (横坐标为排序表编号)。这里看不出 A 有区分能力。

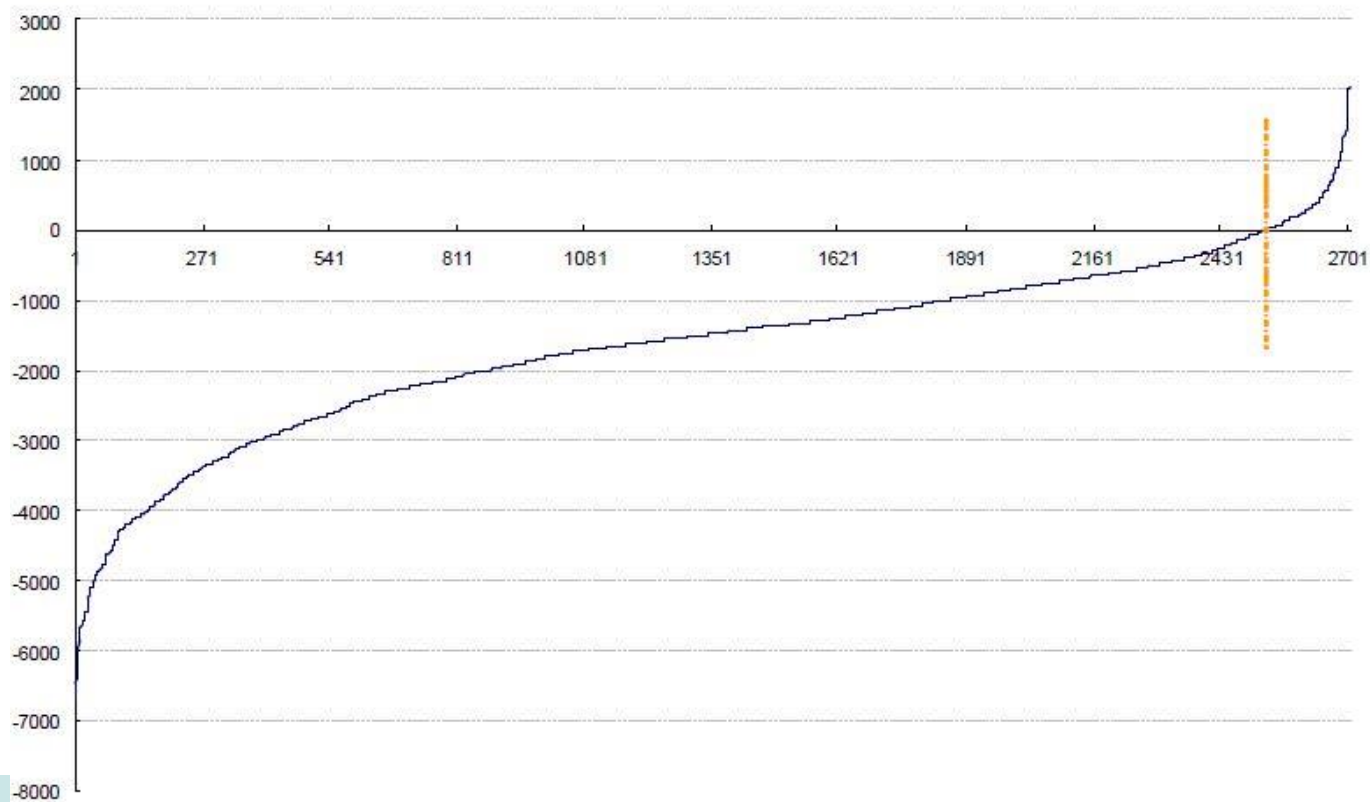
12.3 实例分析--弱分类器及其选取

矩形特征 B 在 20×20 子窗口中位置如右图

参数: 坐标 $(6, 2)-(16, 11)$ (s, t) 条件 $(1, 2)$

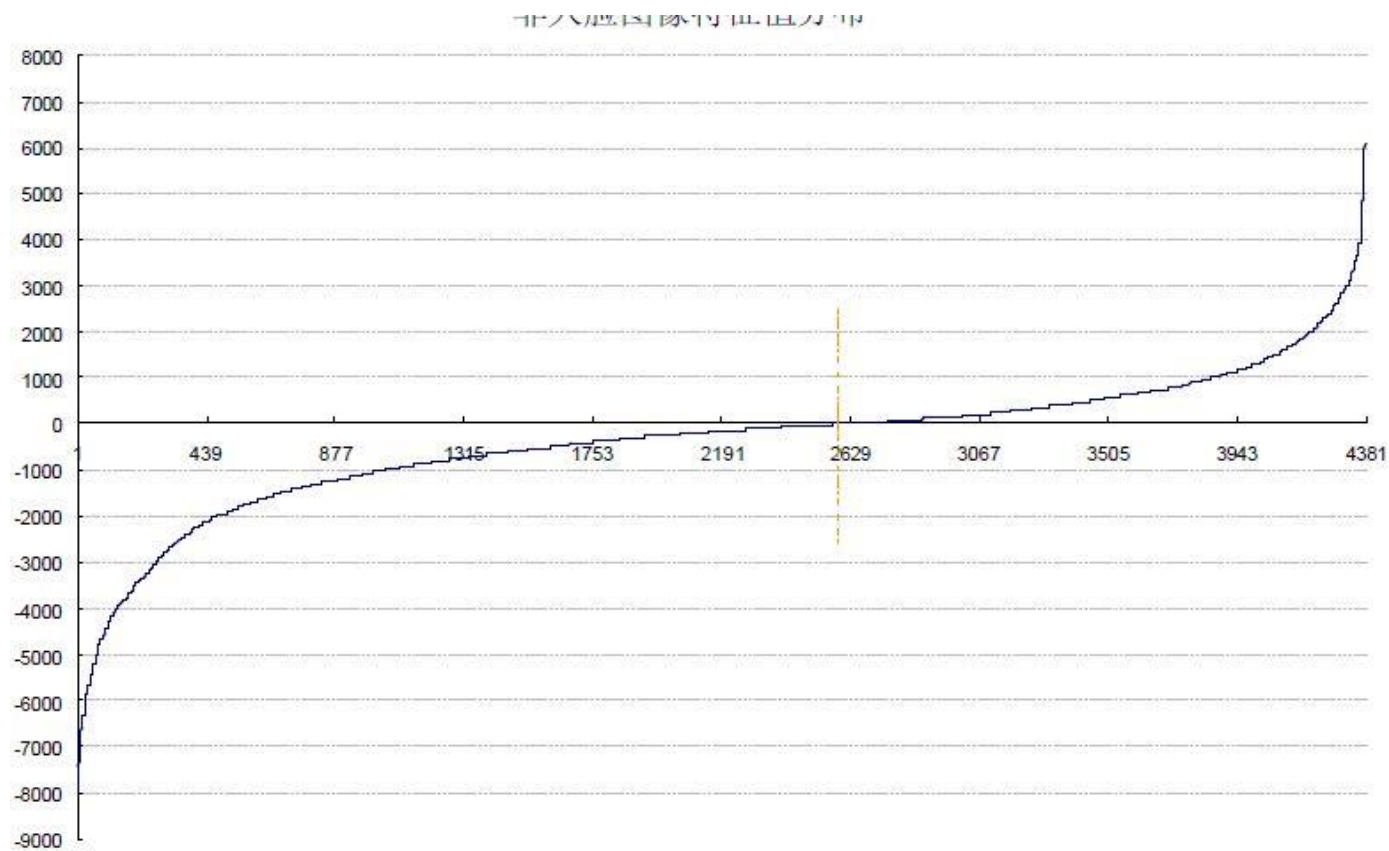


人脸图像特征值分布



2,706 个
人脸样本

12.3 实例分析--弱分类器及其选取



4,381 个
非人脸样
本

图 18 矩形特征 B 对人脸和非人脸图像的特征值分布 (横坐标为排序表编号)。这里 B 表现了很强的分辨能力。

12.3 实例分析--弱分类器及其选取

具体来说，对排好序的表中的每个元素，计算下面四个值：

- 1) 全部人脸样本的权重的和 T^+ ; $=0.5$, B特征每一个 $1/2706$
- 2) 全部非人脸样本的权重的和 T^- ; $=0.5$, 每一个 $1/4381$
- 3) 在此元素之前的人脸样本的权重的和 S^+ ; 阈值=0时, $=2500/2706$
- 4) 在此元素之前的非人脸样本的权重的和 S^- ; 阈值=0时, $=2600/4381$

这样，当选取当前元素的特征值 F_j 和它前面的一个特征值 F_{j-1} 之间的数作为阈值时，对应的弱分类器在当前元素处把样本分开。统计利用该阈值进行分类时的分类误差：

$$\varepsilon = \min(\underbrace{S^+ + (T^- - S^-)}_{\text{阈值前的元素判为非人脸, 阈值后(含)的判为人脸}}, \underbrace{S^- + (T^+ - S^+)}_{\text{阈值前的元素判为人脸, 阈值后(含)的判为非人脸}})$$

阈值前的元素判为非人脸，阈值后（含）的判为人脸 阈值前的元素判为人脸，阈值后（含）的判为非人脸

从头到尾遍历扫描排序表，为弱分类器选择使分类误差最小的阈值（最优阈值 F_{\min} ），即是为对应的特征选取了一个最佳弱分类器。

对于所有特征，应用以上寻找阈值的方法，就得到了所有特征对应的弱分类器，组成一个弱分类器集，作为训练的输入。

12.3 实例分析—训练强分类器

- ❖ 在弱分类器训练中，“每个特征 f ”指的是在 20×20 大小的训练样本中所有的可能出现的矩形特征，大概有80,000种，所有的这些特征都要进行计算。也就是要计算80,000个左右的弱分类器，再选择性能好的分类器。
- ❖ 特别说明：在前期准备训练样本的时候，需要将样本归一化和灰度化到 20×20 的大小，这样每个样本都是大小一致的灰度图像，保证了每一个Haar特征（描述的是特征及其位置）都在每一个样本中出现。

12.3 实例分析—训练强分类器

输入： 一组训练集： $(x_1, y_1), \dots, (x_n, y_n)$ ，其中 x_i 为样本描述， y_i 为样本标识， $y_i \in (0, 1)$ ；其中 0, 1 分别表示正例子和反例。在人脸检测中，定义：0-非人脸，1-人脸。

初始化： 初始样本权值设为 $w_{1,j} = \frac{1}{n}$ （可能会导致正样本比例很小，所以常用正 m 个，负 n 个，则正的权重为 $1/2m$ ，负的权重 $1/2n$ ，使得正负比例分别为 $1/2$ ）。

对 $t = 1, 2, \dots, T$ ，（ T 为循环数，即找到 T 个弱分类器）循环执行下面的步骤：

1. 归一化权重：

$$q_{t,i} = \frac{w_{t,i}}{\sum_{j=1}^n w_{t,j}}$$

12.3 实例分析—训练强分类器

寻找最优 θ

2. 对每个特征 f ，训练一个弱分类器 $h(x, f_t)$ ；计算每个特征的弱分类器的（全部样本的）加权错误率

$$\varepsilon_f = \sum_i q_i |h(x_i, f_t) - y_i|$$

3. （在所有特征中）选取具有最小错误率的最佳弱分类器 $h_t(x)$

$$\varepsilon_t = \min_f \sum_i q_i |h(x_i, f_t) - y_i| = \sum_i q_i |h(x_i, f_t) - y_i|$$

$$h_t(x) = h(x, f_t)$$

4. 按照这个最佳弱分类器，调整每个样本的权重：

$$w_{t+1,i} = w_{t,i} \beta_t^{1-e_i} \quad \beta_t = \frac{\varepsilon_t}{1 - \varepsilon_t}$$

减少正确分类样本的权重

其中 $e_i = 0$ 表示 x_i 被正确分类， $e_i = 1$ 表示 x_i 被错误分类。

12.3 实例分析—训练强分类器

5. 经过 T 次迭代后，获得了 T 个最佳弱分类器 $h_1(x), \dots, h_T(x)$ ，可以按照下面的方式组合成一个强分类器：

$$C(x) = \begin{cases} 1 & \sum_{t=1}^T \alpha_t h_t(x) \geq \frac{1}{2} \sum_{t=1}^T \alpha_t \\ 0 & \text{其他} \end{cases} \quad \text{其中} \quad \alpha_t = \log \frac{1}{\beta_t}。$$

那么，这个强分类器对一幅待检测图像时，相当于让所有弱分类器投票，再对投票结果按照弱分类器的错误率加权求和，将投票加权求和的结果与平均投票结果比较得出最终的结果。

注：另一种终止方法：

不用循环 T 次，而是用识别率与误识别率是否达到来进行循环控制。在每一次循环完之后，运用步骤5对已得到的弱分类器加权组合后的组合分类器，判断组合分类器的识别率与误识别率是否在预定范围内，若在：停止循环，输出分类器；不在：继续。

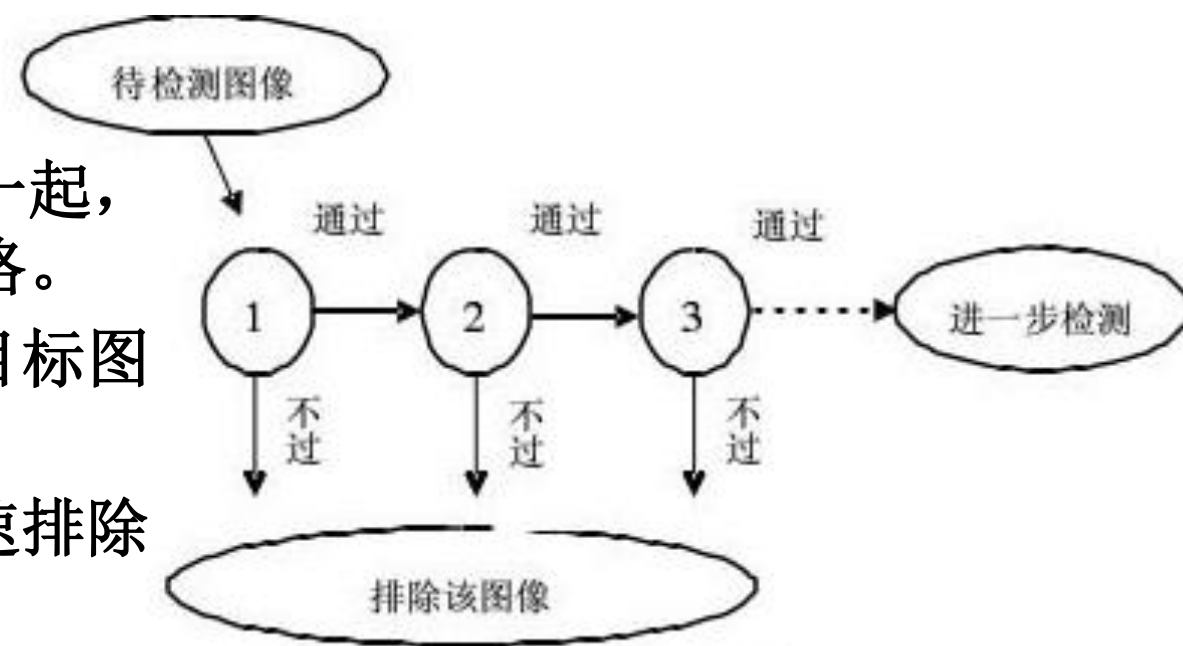
12.3 实例分析—级联分类器

❖ 利用训练过程得到的弱分类器，使用上式将部分弱分类器组合得到若干强分类器，各强分类器对目标都有较强的检测能力。**如果将多个强分类器级联在一起，那么能够通过各级强分类器检测的对象是人脸的可能性也最大。**根据这一原理，Adaboost算法引入了一种瀑布型的分类器——级联分类器。（分类器误识别率不断降低。确定不是正样本的被排除，不确定的到下一个分类器中）

❖ 级联分类器将若干个强分类器分级串联在一起，强分类器一级比一级复杂，一级比一级严格。

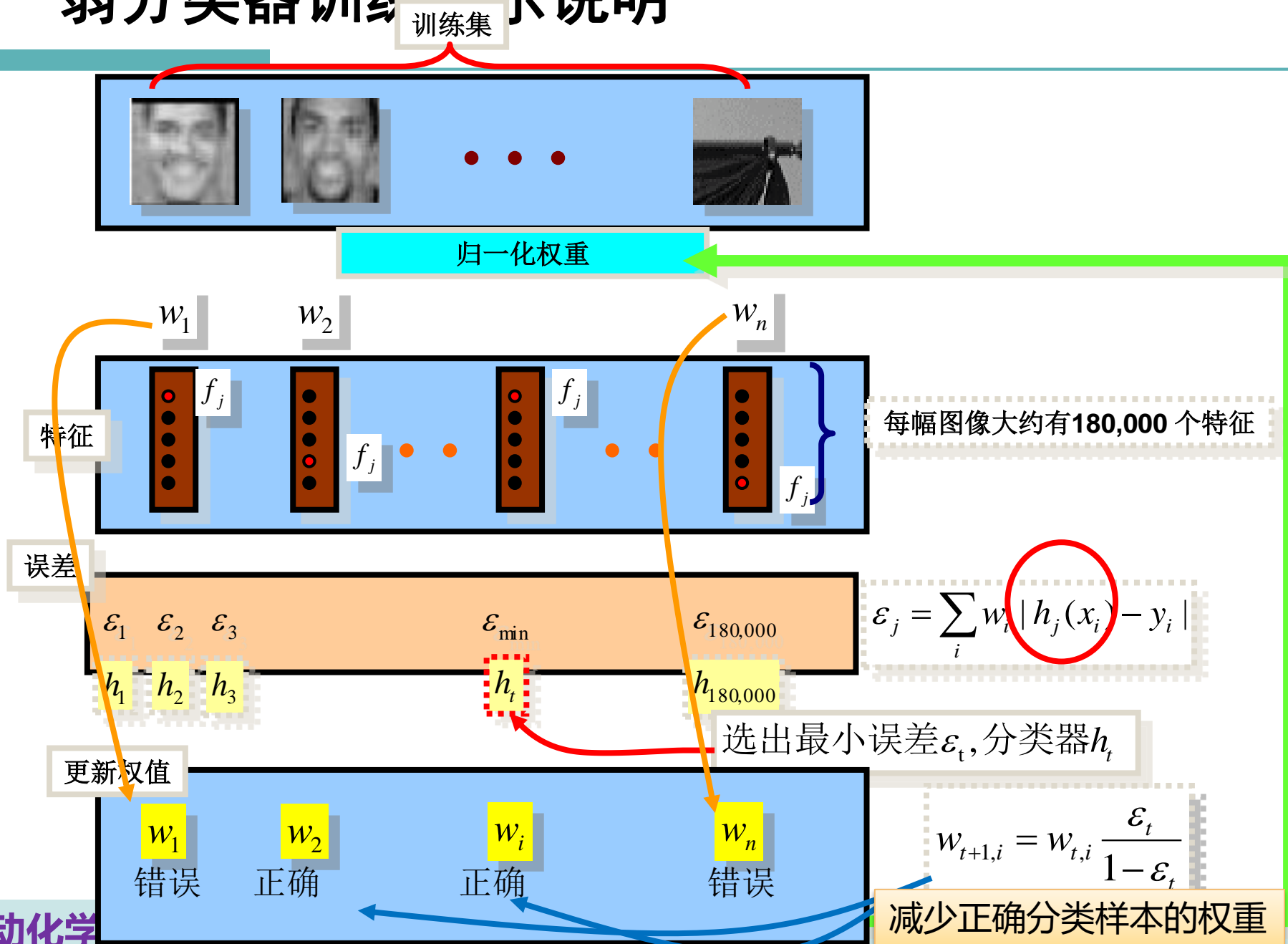
检测中非目标图像会在前端被排除掉，只有目标图像才能通过各级强分类器的检测。

由于非目标图像会被级联分类器的前几级迅速排除掉，从而加快了Adaboost算法的检测速度。



级联分类器的检测示意图

弱分类器训练图示说明



- ❖ 初始赋予每个样本相等的权重 $1/N$;
- ❖ *For* $t = 1, 2, \dots, T$ *Do*
 - 学习得到分类法 C_t ;
 - 计算该分类法的错误率 E_t
 E_t = 所有被错误分类的样本的权重和;
 - $\beta_t = E_t / (1 - E_t)$
 - 根据错误率更新样本的权重;
 - 正确分类的样本: $W_{\text{new}} = W_{\text{old}} * \beta_t$
 - 错误分类的样本: $W_{\text{new}} = W_{\text{old}}$
 - 调整使得权重和为1;
- ❖ 每个分类法 C_t 的投票价值为 $\log [1/\beta_t]$

减少正确分类
样本的权重值

Algorithm 4 AdaBoost.M1

Input: Training set $S = \{\mathbf{x}_i, y_i\}$, $i = 1, \dots, N$; and $y_i \in \mathbb{C}$, $\mathbb{C} = \{c_1, \dots, c_m\}$; T : Number of iterations; I : Weak learner

Output: Boosted classifier:

$$H(x) = \arg \max_{y \in \mathbb{C}} \sum_{t=1}^T \ln \left(\frac{1}{\beta_t} \right) [h_t(x) = y] \text{ where } h_t, \beta_t$$

are the induced classifiers (with $h_t(x) \in \mathbb{C}$) and their assigned weights, respectively

- 1: $D_1(i) \leftarrow 1/N$ for $i = 1, \dots, N$
- 2: **for** $t = 1$ to T **do**
- 3: $h_t \leftarrow I(S, D_t)$
- 4: $\varepsilon_t \leftarrow \sum_{i=1}^N D_t(i) [h_t(\mathbf{x}_i) \neq y_i]$
- 5: **if** $\varepsilon_t > 0.5$ **then**
- 6: $T \leftarrow t - 1$
- 7: **return**
- 8: **end if**
- 9: $\beta_t = \frac{\varepsilon_t}{1 - \varepsilon_t}$
- 10: $D_{t+1}(i) = D_t(i) \cdot \beta_t^{1 - [h_t(\mathbf{x}_i) \neq y_i]}$ for $i = 1, \dots, N$
- 11: Normalize D_{t+1} to be a proper distribution
- 12: **end for**

T1、强分类器的公式，权重的选取怎么来的？

一共n个样本， $Y=\{-1, +1\}$ ，-1代表负样本

错误率： $\varepsilon_f = \sum_i D_t(i) I(h(x, f) \neq y_i)$

假设的权重： $\alpha_t = \frac{1}{2} \ln \left(\frac{1 - \varepsilon_t}{\varepsilon_t} \right)$

从1到M循环,样本权重： $D_{t+1}(i) = \frac{D_t(i)}{Z_t} \times \begin{cases} e^{-\alpha_t} & \text{if } h_t(x_i) = y_i \\ e^{\alpha_t} & \text{if } h_t(x_i) \neq y_i \end{cases}$

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

adaboost最终强分类器的错误率上限

自习

为什么要设置 $\alpha_t = \frac{1}{2} \ln \left(\frac{1 - \varepsilon_t}{\varepsilon_t} \right)$

首先证明adaboost最终强分类器的错误率上限是：

$$\frac{1}{n} \sum_{i=1}^n I(H(x_i) \neq y_i) \leq \prod_{t=1}^T Z_t$$

↑ 个数

← adaboost训练误差界

证明：

由于
$$D_{t+1}(i) = \frac{D_t(i)}{Z_t} \times \begin{cases} e^{-\alpha_t} & \text{if } h_t(x_i) = y_i \\ e^{\alpha_t} & \text{if } h_t(x_i) \neq y_i \end{cases}$$

通过迭代公式倒推回去，一直迭代到 $D_1=1/n$ 。公式变为：

$$D_{t+1}(i) = \frac{1}{n \prod_{t=1}^T Z_t} \times \begin{cases} e^{-\sum_{t=1}^T \alpha_t} & \text{if } h_t(x_i) = y_i \\ e^{\sum_{t=1}^T \alpha_t} & \text{if } h_t(x_i) \neq y_i \end{cases}$$

adaboost最终强分类器的错误率上限

$$D_{t+1}(i) = \frac{1}{n \prod_t Z_t} \times \begin{cases} e^{-\sum_t \alpha_t} & \text{if } h_t(x_i) = y_i \\ e^{\sum_t \alpha_t} & \text{if } h_t(x_i) \neq y_i \end{cases}$$

将上式写成一个式子，如下：

（定义：h与y相等，两者乘积为1； h与y不等，两者乘积为-1）

$$D_{t+1}(i) = \frac{\exp\left(-\sum_t \alpha_t y_i h_t(x_i)\right)}{n \prod_t Z_t} = \frac{\exp\left(-y_i \left(\sum_t \alpha_t h_t(x_i)\right)\right)}{n \prod_t Z_t}$$

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

如果 $H(x_i) \neq y_i$ ，有 $y_i \left(\sum_{t=1}^T \alpha_t h_t(x_i)\right) \leq 0$ ，因此 $\exp\left(-y_i \left(\sum_t \alpha_t h_t(x_i)\right)\right) \geq 1$

因此有： $I(H(x_i) \neq y_i) \leq \exp\left(-y_i \left(\sum_t \alpha_t h_t(x_i)\right)\right)$

$$\frac{1}{n} \sum_{i=1}^n I(H(x_i) \neq y_i) \leq \frac{1}{n} \sum_{i=1}^n \exp\left(-y_i \left(\sum_t \alpha_t h_t(x_i)\right)\right) = \sum_{i=1}^n \left(\prod_t Z_t\right) D_{t+1}(i) = \prod_t Z_t$$

即证：adaboost最终强分类器的错误率上限

为了使得adaboost算法精确，就要使错误率最小，就是使 $\prod_t Z_t$ 最小，而

$$Z_t = \sum_i D_t(i) \exp(-\alpha_t y_i h_t(x_i))$$

是 α_t 、 h_t 的函数，又因为 h 的值域是 $\{-1, 1\}$ ，所以只要求另一个参数使得 Z 最小。（在原始的AdaBoost算法中采用贪婪算法，每次的 Z_t 都是最小的保证 $\prod_t Z_t$ 收敛到满意的结果。）

求 Z_t 偏导数，令其为0。

$$\frac{\partial Z_t}{\partial \alpha_t} = \sum_i D_t(x_i) \exp(-y_i h_t \alpha_t) \cdot (-y_i h_t) = 0$$

其中设 A 是 $h=y$ 的集合，就有以下式子：

$$\frac{\partial Z_t}{\partial a_t} = \sum_t D_t(x_i) \exp(-y_i h_t a_t) \cdot (-y_i h_t) = 0$$

因为，如果 $y=h$, $yh=1$; 否则 $yh=-1$ 。A类为 $y=h$ 的样本集。 \bar{A} 为其他样本

$$\frac{\partial Z_t}{\partial \alpha_t} = 0 \Rightarrow \sum_{x_i \in A} D_t(i) \exp(-\alpha_t) = \sum_{x_i \in \bar{A}} D_t(i) \exp(\alpha_t)$$

$$\Rightarrow \sum_{x_i \in A} D_t(i) \exp(-\alpha_t) = \sum_{x_i \in \bar{A}} D_t(i) \exp(\alpha_t), \text{ 两边同乘以 } \exp(\alpha_t)$$

$$\sum_{x_i \in A} D_t(i) = \exp(2\alpha_t) \sum_{x_i \in \bar{A}} D_t(i)$$

$$\text{正确率} = \sum_{x_i \in A} D_t(i) = 1 - \varepsilon_t, \quad \text{错误率} = \sum_{x_i \in \bar{A}} D_t(i) = \varepsilon_t,$$

$$\text{所以 } 1 - \varepsilon_t = \varepsilon_t \exp(2\alpha_t)$$

$$\text{所以 } \alpha_t = \frac{1}{2} \ln \left(\frac{1 - \varepsilon_t}{\varepsilon_t} \right)$$

即证：权重的选取

二分类问题adaboost的训练误差（错误率）上限

将 $\gamma_t = \frac{1}{2} - \varepsilon_t$;

自习

Freund and Schapire 证明了最大错误率为:

$$\prod Z_t = \prod [2\sqrt{\varepsilon_t(1-\varepsilon_t)}] = \prod \sqrt{1-4\gamma_t^2}$$

即训练错误率随 γ_t 的增大呈指数级的减小。

证明: $Z_t = \sum_{i=1}^n D_t(i) \exp(-a_t y_i h_t(x_i))$

$$= \sum_{i=1, y=h}^n D_t(i) \exp(-a_t) + \sum_{i=1, y \neq h}^n D_t(i) \exp(a_t)$$

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

$$= (1-\varepsilon_t) \exp(-a_t) + \varepsilon_t \exp(a_t) = 2\sqrt{(1-\varepsilon_t)\varepsilon_t} = \sqrt{1-4\gamma_t^2}$$

由泰勒展开式在点x=0处 $e^{0.5x} = 1 + \frac{0.5}{1!}x + \frac{0.5*0.5}{2!}x^2 + \frac{0.5*0.5*0.5}{3!}x^3 + o(x^3)$

$$(1+x)^{0.5} = 1 + \frac{0.5}{1!}x + \frac{0.5(0.5-1)}{2!}x^2 + \frac{0.5(0.5-1)(0.5-2)}{3!}x^3 + o(x^3)$$

假设 $x = -4\gamma_t^2$ 有 $x \leq 0$, 且 $x > -1$

所以有 $(1+x)^{0.5} \leq e^{0.5x}$

$$\prod_{t=1}^T \sqrt{1-4\gamma_t^2} \leq \exp \left(-2 \sum_{t=1}^T \gamma_t^2 \right)$$

To learn more ...



❖ Introduction of Adaboost:

- Freund; Schapire (1999). "A Short Introduction to Boosting"

❖ Multiclass/Regression

- Y. Freund, R. Schapire, "A Decision-Theoretic Generalization of on-Line Learning and an Application to Boosting", 1995.
- Robert E. Schapire and Yoram Singer. Improved boosting algorithms using confidence-rated predictions. In Proceedings of the Eleventh Annual Conference on Computational Learning Theory, pages 80–91, 1998.

❖ Gentle Boost

- Schapire, Robert; Singer, Yoram (1999). "Improved Boosting Algorithms Using Confidence-rated Predictions".

❖ 人脸检测

- [1]. Viola P, Jones M. Rapid object detection using a boosted cascade of simple features[C] //Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on. IEEE, 2001, 1: 1-511-1-518 vol. 1.



Next:

集成学习（续）——决策树&随机森林