

原

Adaboost算法原理分析和实例+代码（简明易懂）

置顶

2017年05月02日 08:52:31

pan\_jinquan

阅读数：46032

标签：

Adaboost算法原理

Adaboost代码

Adaboost例子

Adaboost优点和缺点

Boosting

更多

个人分类：

机器学习

版权声明：本文为博主原创文章，未经博主允许不得转载（pan\_jinquan）https://blog.csdn.net/guyuealian/article/details/70995333

Adaboost算法原理分析和实例+代码（简明易懂）

【尊重原创，转载请注明出处】http://blog.csdn.net/guyuealian/article/details/70995333

本人最初了解AdaBoost算法着实是花了几天时间，才明白他的基本原理。也许是自己能力有限吧，很多资料也是看得懵懵懂懂。网上找了一下关于AdaBoost算法原理分析，是你复制我，我摘抄你，反正我也搞不清谁是原创。有些资料给出的Adaboost实例，要么是没有代码，要么省略很多步骤，让初学者很难看懂AdaBoost过程。

本博客将会详细介绍AdaBoost算法过程，并给出了一个Adaboost例子的详细求解过程，当然也给出了Matlab代码求解过程。碍于太多复杂公式，文章是在电脑Word文档写好制上博客的，为了排版好看，有些地方给出了截图。

下面给出几个我认为不错的博客资料：

- 【1】[http://blog.csdn.net/v\\_july\\_v/article/details/40718799](http://blog.csdn.net/v_july_v/article/details/40718799) 感谢这位博主给出了 Adaboost 算法的原理与推导，本文章很多地方都参考了他的内容
- 【2】[http://blog.csdn.net/m0\\_37407756/article/details/67637400](http://blog.csdn.net/m0_37407756/article/details/67637400) 该博客有一个Adaboost 算法的例子，但其过程简略太多，初学者很难看懂。本文章的Adaboost 算法例子也是与之相对应的，但本人给出了详细的步骤和分析过程。话说，图都是我一个一个画上去，心疼我用了两天时间！！

（1）鉴于很多网友评论留言说，为啥子第三次迭代时，“**这个强分类器Hfinal对训练样本的错误率为0！**”。这里要说明一下，如果你只是用眼去看，而不假思索一下，你是绝逼看不出结果来的！！！要计算，公式就是： $H_{final} = sign(f_3(x))$ 这个函数，其中是sign符号函数（大于0，返回1；小于0，则返回-1），把样本点X代入公式计算就可以得到预测结果了

（2）为了照顾大家，我又重新绘制了表格，把每一步 $sign(f_i(x))$ 的预测结果显示出来，详见每一次迭代的表格的 $sign(f_i(x))$ 值

（3）附带了Matlab源代码，自己运行一下，就OK了

（4）这份博客，花了我好大心血，记得点赞哈哈！

（5）娃哈哈~！

一、AdaBoost简介

Boosting, 也称为增强学习或提升法，是一种重要的集成学习技术， 能够将预测精度仅比随机猜度略高的弱学习器增强为预测精度高的强学习器，这在直接构造强学习器非常困难的情况下，为学习算法的设计提供了一种有效的新思路和新方法。其中最为成功应用的是，Yoav Freund和Robert Schapire在1995年提出的AdaBoost算法。

AdaBoost是英文"Adaptive Boosting"（自适应增强）的缩写，它的自适应在于：前一个基本分类器被错误分类的样本的权值会增大，而正确分类的样本的权值会减小，并再次用来训练下一个基本分类器。同时，在每一轮迭代中，加入一个新的弱分类器，直到达到某个预定的足够小的错误率或达到预先指定的最大迭代次数才确定最终的强分类器。

Adaboost算法可以简述为三个步骤：

- （1）首先，是初始化训练数据的权值分布D1。假设有N个训练样本数据，则每一个训练样本最开始时，都被赋予相同的权值：w1=1/N。
- （2）然后，训练弱分类器hi。具体训练过程中是：如果某个训练样本点，被弱分类器hi准确地分类，那么在构造下一个训练集中，它对应的权值要减小；相反，如果某个训练样本点被错误分类，那么它的权值就应该增大。权值更新过的样本集被用于训练下一个分类器，整个训练过程如此迭代地进行下去。
- （3）最后，将各个训练得到的弱分类器组合成一个强分类器。各个弱分类器的训练过程结束后，加大分类误差率小的弱分类器的权重，使其在最终的分类函数中起着较大的决定作用，而降低分类误差率大的弱分类器的权重，使其在最终的分类函数中起着较小的决定作用。
- 换言之，误差率低的弱分类器在最终分类器中占的权重较大，否则较小。

二、AdaBoost算法过程

给定训练数据集： $(x_1, y_1), \dots, (x_N, y_N)$ ，其中 $y_i \in \{1, -1\}$ 用于表示训练样本的类别标签， $i=1, \dots, N$ 。Adaboost的目的就是从训练数据中学习一系列弱分类器或基本分类器，然后将这些弱分类器组合成一个强分类器。

相关符号定义：

$D_i(i)$ ：训练样本集的权值分布；  
 $w_i$ ：每个训练样本的权值大小；  
 $h$ ：弱分类器，  
 $H$ ：基本分类器；  
 $H_{final}$ =最终的强分类器  
 $e$ ：误差率；  
 $\alpha$ ：弱分类器的权重；

https://blog.csdn.net/guyuealian

Adaboost的算法流程如下：

- （1）首先，初始化训练数据的权值分布。每一个训练样本最开始时都被赋予相同的权值： $w_i=1/N$ ，这样训练样本集的初始权值分布  $D_1(i)$ ：

$$D_1(i) = (w_1, w_2, \dots, w_N) = (\frac{1}{N}, \dots, \frac{1}{N})$$

(2) 进行迭代  $t=1,\dots,T$

(a) 选取一个当前误差率最低的弱分类器  $h$  作为第  $t$  个基本分类器  $H_t$ ，并计算弱分类器  $h_t: X \rightarrow \{-1,1\}$ ，该弱分类器在分布  $D_t$  上的误差为：

$$e_t = P(H_t(x_i) \neq y_i) = \sum_{i=1}^N w_{it} I(H_t(x_i) \neq y_i)$$

PS：由上述式子可知， $H_t(x)$  在训练数据集上的误差率  $e_t$  就是被  $H_t(x)$  误分类样本的权值之和。

(b) 计算该弱分类器在最终分类器中所占的权重（弱分类器权重用  $\alpha$  表示）：

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

(c) 更新训练样本的权值分布  $D_{t+1}$ ：

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

其中  $Z_t$  为归一化常数  $Z_t = 2\sqrt{e_t(1-e_t)}$

(3) 最后，按弱分类器权重  $\alpha$  组合各个弱分类器，即

$$f(x) = \sum_{t=1}^T \alpha_t H_t(x)$$

通过符号函数  $\text{sign}$  的作用，得到一个强分类器为：

$$H_{\text{final}} = \text{sign}(f(x)) = \text{sign} \left( \sum_{t=1}^T \alpha_t H_t(x) \right)$$

相关说明：

因为权重更新依赖于  $\alpha$ ，而  $\alpha$  又依赖于误差率  $e$ ，所以我们可以直接将权重更新公式用  $e$  表示。样本权重更新公式： $D_{t+1}(i) = \frac{D_t(i) \exp(-\alpha_t y_i H_t(x_i))}{Z_t}$ ，其中

$$Z_t = 2\sqrt{e_t(1-e_t)}$$

(1) 当样本分错时， $y_i H_t(x_i) = -1$

$$D_{t+1}(i) = \frac{D_t(i)}{Z_t} \exp(\alpha_t) = \frac{D_t(i)}{Z_t} \exp\left(\frac{1}{2} \ln\left(\frac{1-e_t}{e_t}\right)\right) = \frac{D_t(i)}{Z_t} \sqrt{\frac{1-e_t}{e_t}} = \frac{D_t(i)}{2\sqrt{e_t(1-e_t)}} \sqrt{\frac{1-e_t}{e_t}} = \frac{D_t(i)}{2e_t}$$

(2) 当样本分对时， $y_i H_t(x_i) = 1$

$$D_{t+1}(i) = \frac{D_t(i)}{Z_t} \exp(-\alpha_t) = \frac{D_t(i)}{Z_t} \exp\left(-\frac{1}{2} \ln\left(\frac{1-e_t}{e_t}\right)\right) = \frac{D_t(i)}{Z_t} \sqrt{\frac{e_t}{1-e_t}} = \frac{D_t(i)}{2\sqrt{e_t(1-e_t)}} \sqrt{\frac{e_t}{1-e_t}} = \frac{D_t(i)}{2(1-e_t)}$$

综合上面的推导，可得样本分错与分对时，其权值更新的公式为：

$$\text{错误分类样本，权值更新：} D_{t+1}(i) = \frac{D_t(i)}{2e_t}$$

$$\text{正确分类样本，权值更新：} D_{t+1}(i) = \frac{D_t(i)}{2(1-e_t)}$$

1、鉴于很多网友反应，归一化常数  $Z_t$  是怎么来的，这个我就不做推导，直接给个截图吧：

截图来自：[https://blog.csdn.net/m0\\_37407756/article/details/67637400](https://blog.csdn.net/m0_37407756/article/details/67637400)

• **Step 3:**  $Z_t = 2\sqrt{\epsilon_t(1-\epsilon_t)}$

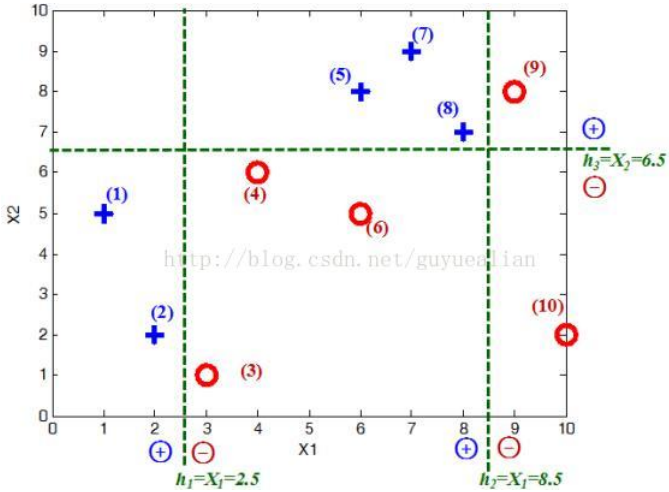
• **Proof:**

$$\begin{aligned} Z_t &= \sum_i D_t(i) \exp(-\alpha_t y_i h_t(x_i)) \\ &= \sum_{i: y_i \neq h_t(x_i)} D_t(i) e^{\alpha_t} + \sum_{i: y_i = h_t(x_i)} D_t(i) e^{-\alpha_t} \\ &= \epsilon_t e^{\alpha_t} + (1 - \epsilon_t) e^{-\alpha_t} \\ &= 2\sqrt{\epsilon_t(1-\epsilon_t)} \end{aligned}$$

### 三、AdaBoost实例讲解

例：给定如图所示的训练样本，弱分类器采用平行于坐标轴的直线，用Adaboost算法的实现强分类过程。

样本序号	1	2	3	4	5	6	7	8	9	10
样本点 $X$	(1,5)	(2,2)	(3,1)	(4,6)	(6,8)	(6,5)	(7,9)	(8,7)	(9,8)	(10,2)
类别 $Y$	1	1	-1	-1	1	-1	1	1	-1	-1



数据分析：

将这10个样本作为训练数据，根据  $X$  和  $Y$  的对应关系，可把这10个数据分为两类，图中用“+”表示类别1，用“O”表示类别-1。本例使用水平或者垂直的直线作为分类器，图中已经给出了三个弱分类器，即：

$$h_1 = \begin{cases} 1, & X_1 < 2.5 \\ -1, & X_1 \geq 2.5 \end{cases}, \quad h_2 = \begin{cases} 1, & X_1 < 8.5 \\ -1, & X_1 \geq 8.5 \end{cases}, \quad h_3 = \begin{cases} 1, & X_2 > 6.5 \\ -1, & X_2 \leq 6.5 \end{cases}$$

初始化：

首先需要初始化训练样本数据的权值分布，每一个训练样本最开始时都被赋予相同的权值： $w_i=1/N$ ，这样训练样本集的初始权值分布 $D1(i)$ ：

令每个权值 $w_i = 1/N = 0.1$ ，其中， $N = 10$ ， $i = 1, 2, \dots, 10$ ，然后分别对于 $t=1, 2, 3, \dots$ 等值进行迭代（ $t$ 表示迭代次数，表示第 $t$ 轮），下表已经给出训练样本的权值分布情况：

样本序号	1	2	3	4	5	6	7	8	9	10
样本点 $X$	(1,5)	(2,2)	(3,1)	(4,6)	(6,8)	(6,5)	(7,9)	(8,7)	(9,8)	(10,2)
类别 $Y$	1	1	-1	-1	1	-1	1	1	-1	-1
权值分布 $D_t$	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1

第1次迭代 $t=1$ ：

初试的权值分布 $D1$ 为 $1/N$ （10个数据，每个数据的权值皆初始化为0.1），

$$D1=[0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1]$$

在权值分布 $D1$ 的情况下，取已知的三个弱分类器 $h1$ 、 $h2$ 和 $h3$ 中误差率最小的分类器作为第1个基本分类器 $H1(x)$ （三个弱分类器的误差率都是0.3，那就取第1个吧）

$$H_1 = \begin{cases} 1, & X_1 < 2.5 \\ -1, & X_1 \geq 2.5 \end{cases}$$

PS：某个分类器的误差率等于该分类器的被错分类样本的权值之和

在分类器 $H1(x)=h1$ 情况下，样本点“5 7 8”被错分，因此基本分类器 $H1(x)$ 的误差率为：

误差率为：

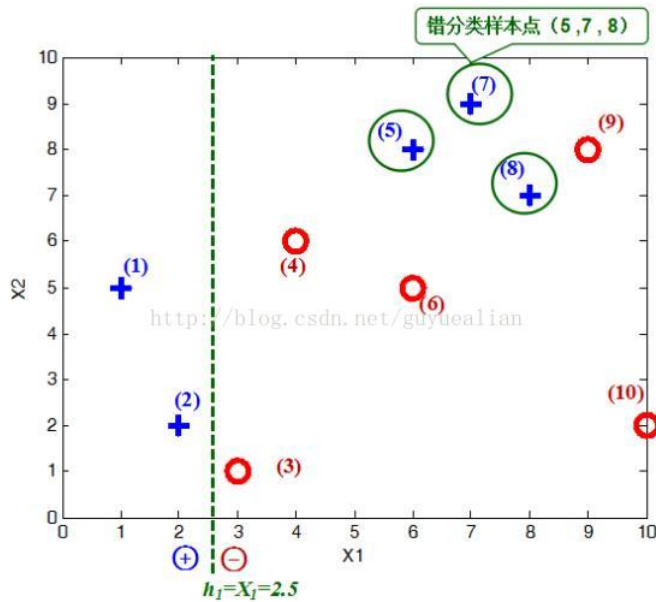
$$e_1 = (0.1 + 0.1 + 0.1) = 0.3$$

根据误差率  $e_1$  计算  $H_1$  的权重：

$$\alpha_1 = \frac{1}{2} \ln\left(\frac{1-e_1}{e_1}\right) = \frac{1}{2} \ln\left(\frac{1-0.3}{0.3}\right) = 0.4236$$

PS：这个 $\alpha_1$ 代表  $H_1(x)$ 在最终的分类函数中所占的权重为 0.4236

可见，被误分类样本的权值之和影响误差率 $e$ ，误差率 $e$ 影响基本分类器在最终分类器中所占的权重 $\alpha$ 。



然后，更新训练样本数据的权值分布，用于下一轮迭代，对于正确分类的训练样本“1 2 3 4 6 9 10”（共7个）的权值更新为：

$$D_2 = \frac{D_1}{2(1-\varepsilon_1)} = \frac{1}{10} \times \frac{1}{2 \times (1-0.3)} = \frac{1}{14}$$

PS：可见，正确分类的样本的权值由原来 1/10 减小到 1/14。

对于所有错误分类的训练样本“5 7 8”（共 3 个）的权值更新为：

$$D_2(i) = \frac{D_1(i)}{2\varepsilon_1} = \frac{1}{10} \times \frac{1}{2 \times 0.3} = \frac{1}{6}$$

PS：可见，错误分类的样本的权值由原来 1/10 增大到 1/6。

这样，第1轮迭代后，最后得到各个样本数据新的权值分布：

$D_2 = [1/14, 1/14, 1/14, 1/14, 1/6, 1/14, 1/6, 1/6, 1/14, 1/14]$

由于样本数据“5 7 8”被 $H_1(x)$ 分错了，所以它们的权值由之前的0.1增大到1/6；反之，其它数据皆被分正确，所以它们的权值皆由之前的0.1减小到1/14，下表给出了权值分布的变换情况：

样本序号	1	2	3	4	5	6	7	8	9	10
样本点 $X$	(1,5)	(2,2)	(3,1)	(4,6)	(5,8)	(6,5)	(7,9)	(8,7)	(9,8)	(10,2)
类别 $Y$	1	1	-1	-1	1	-1	1	1	-1	-1
权值分布 $D_1$	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1
权值分布 $D_2$	1/14	1/14	1/14	1/14	1/6	1/14	1/6	1/6	1/14	1/14
$sign(f_1(x))$	1	1	-1	-1	-1	-1	-1	-1	-1	-1

被  $H_1(x)$  分错的样本

PS：用浅绿色底纹标记的表格，是被  $H_1(x)$  分错的样本“5 7 8”；没有底纹（白色的）是正确分类的样本

可得分类函数： $f_1(x) = \alpha_1 H_1(x) = 0.4236 H_1(x)$ 。此时，组合一个基本分类器  $sign(f_1(x))$  作为强分类器在训练数据集上有3个误分类点（即5 7 8），此时强分类器的训练错误为：0.3

第二次迭代  $t=2$ ：

在权值分布  $D_2$  的情况下，再取三个弱分类器  $h_1$ 、 $h_2$  和  $h_3$  中误差率最小的分类器作为第2个基本分类器  $H_2(x)$ ：

① 当取弱分类器  $h_1 = X_1 = 2.5$  时，此时被错分的样本点为“5 7 8”：

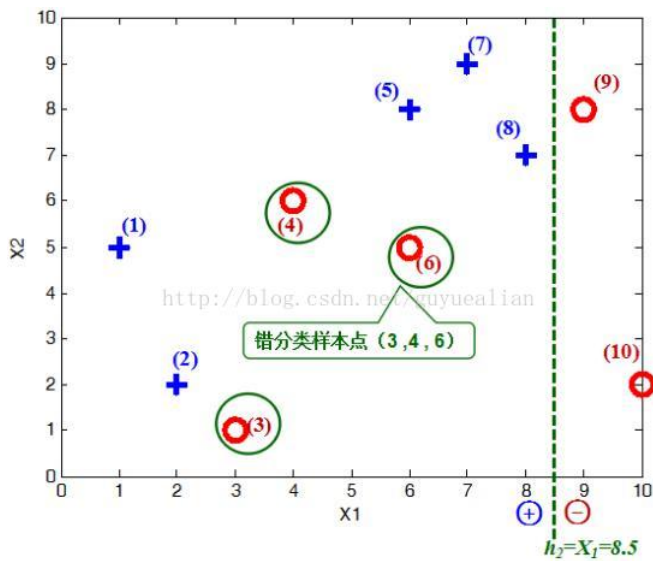
误差率  $e = 1/6 + 1/6 + 1/6 = 3/6 = 1/2$ ；

② 当取弱分类器  $h_2 = X_1 = 8.5$  时，此时被错分的样本点为“3 4 6”：

误差率  $e = 1/14 + 1/14 + 1/14 = 3/14$ ；

③ 当取弱分类器  $h_3 = X_2 = 6.5$  时，此时被错分的样本点为“1 2 9”：

误差率  $e = 1/14 + 1/14 + 1/14 = 3/14$ ；



因此，取当前最小的分类器 $h_2$ 作为第2个基本分类器 $H_2(x)$

$$H_2 = \begin{cases} 1, & X_1 < 8.5 \\ -1, & X_1 > 8.5 \end{cases}$$

显然， $H_2(x)$ 把样本“3 4 6”分错了，根据 $D_2$ 可知它们的权值为 $D_2(3)=1/14$ ， $D_2(4)=1/14$ ， $D_2(6)=1/14$ ，所以 $H_2(x)$ 在训练数据集上的误差率：

$$e_2 = P(H_2(x_i) \neq y_i) = 3 \times \frac{1}{14} = \frac{3}{14} \quad (\text{即权值之和})$$

根据误差率 $e_2$ 计算 $H_2$ 的权重：

$$\alpha_2 = \frac{1}{2} \ln\left(\frac{1-e_2}{e_2}\right) = 0.6496$$

更新训练样本数据的权值分布，对于正确分类的样本权值更新为：

$$D_3(i) = \frac{D_2(i)}{2(1-e_2)} = \frac{7}{11} D_2(i),$$

对于错误分类的权值更新为：

$$D_3(i) = \frac{D_2(i)}{2e_2} = \frac{7}{3} D_2(i)$$

这样，第2轮迭代后，最后得到各个样本数据新的权值分布：

$$D_3 = [1/22, 1/22, 1/6, 1/6, 7/66, 1/6, 7/66, 1/22, 1/22]$$

下表给出了权值分布的变换情况：

序号	1	2	3	4	5	6	7	8	9	10
样本点 $x$	(1,5)	(2,2)	(3,1)	(4,6)	(5,8)	(6,5)	(7,9)	(8,7)	(9,8)	(10,2)
类别 $Y$	1	1	-1	-1	1	-1	1	1	-1	-1
权值分布 $D_1$	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1
权值分布 $D_2$	1/14	1/14	1/14	1/14	1/6	1/14	1/6	1/6	1/14	1/14
$sign(f_1(x))$	1	1	-1	-1	-1	-1	-1	-1	-1	-1
权值分布 $D_3$	1/22	1/22	1/6	1/6	7/66	1/6	7/66	7/66	1/22	1/22
$sign(f_2(x))$	1	1	1	1	1	1	1	1	-1	-1

被  $H_2(x)$  分错的样本

可得分类函数： $f_2(x)=0.4236H_1(x) + 0.6496H_2(x)$ 。此时，组合两个基本分类器 $sign(f_2(x))$ 作为强分类器在训练数据集上有3个误分类点（即3 4 6），此时强分类器的训练错误为：0.3

第三次迭代 $t=3$ ：

在权值分布 $D_3$ 的情况下，再取三个弱分类器 $h_1$ 、 $h_2$ 和 $h_3$ 中误差率最小的分类器作为第3个基本分类器 $H_3(x)$ ：

① 当取弱分类器 $h_1=X_1=2.5$ 时，此时被错分的样本点为“5 7 8”：

误差率 $e=7/66+7/66+7/66=7/22$ ；

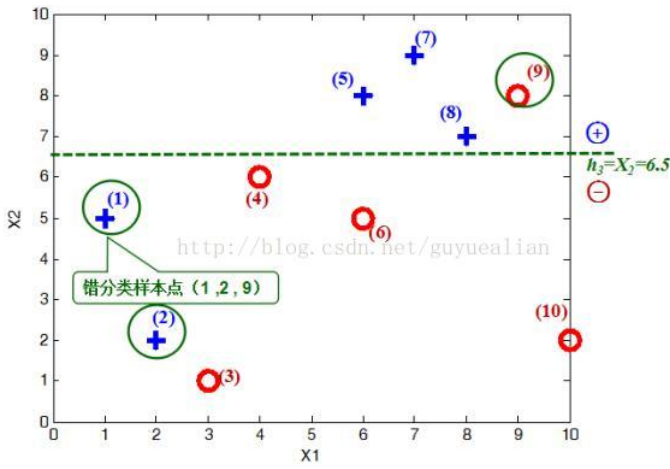
② 当取弱分类器 $h_2=X_1=8.5$ 时，此时被错分的样本点为“3 4 6”：

误差率 $e=1/6+1/6+1/6=1/2=0.5$ ；

③ 当取弱分类器 $h_3=X_2=6.5$ 时，此时被错分的样本点为“1 2 9”：

误差率 $e=1/22+1/22+1/22=3/22$ ；





因此，取当前最小的分类器 $h_3$ 作为第3个基本分类器 $H_3(x)$ ：

$$H_3(x) = \begin{cases} 1, & X_2 > 6.5 \\ -1, & X_2 < 6.5 \end{cases}$$

此时被 $H_3(x)$ 误分类的样本是“1 2 9”，根据 $D_3$ 可知它们的权值为 $D_3(1)=1/22$ ， $D_3(2)=1/22$ ， $D_3(9)=1/22$ ，所以 $H_3(x)$ 在训练数据集上的误差率：

$$e_3 = P(H_3(x_i) \neq y_i) = 3 \times \frac{1}{22} = \frac{3}{22} \quad (\text{即权值之和})$$

根据误差率 $e_3$ 计算 $H_3$ 的权重：

$$\alpha_3 = \frac{1}{2} \ln\left(\frac{1-e_3}{e_3}\right) = 0.9229$$

更新训练样本数据的权值分布，对于**正确分类**的样本权值更新为：

$$D_4(i) = \frac{D_3(i)}{2(1-e_3)} = \frac{11}{19} D_3(i)$$

对于**错误分类**的权值更新为：

$$D_4(i) = \frac{D_3(i)}{2e_3} = \frac{11}{3} D_3(i)$$

这样，第3轮迭代后，得到各个样本数据新的权值分布为：

$$D_4 = [1/6, 1/6, 11/114, 11/114, 7/114, 11/114, 7/114, 11/114, 7/114, 1/6, 1/38]$$

下表给出了权值分布的变换情况：

序号	1	2	3	4	5	6	7	8	9	10
样本点 $X$	(1,5)	(2,2)	(3,1)	(4,6)	(6,8)	(6,5)	(7,9)	(8,7)	(9,8)	(10,2)
类别 $Y$	1	1	-1	-1	1	-1	1	1	-1	-1
权值分布 $D_1$	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1
权值分布 $D_2$	1/4	1/4	1/4	1/4	1/6	1/4	1/6	1/6	1/4	1/4
$\text{sign}(f_1(x))$	1	1	-1	-1	-1	-1	-1	-1	-1	-1
权值分布 $D_3$	1/22	1/22	1/6	1/6	7/66	1/6	7/66	7/66	1/22	1/22
$\text{sign}(f_2(x))$	1	1	1	1	1	1	1	1	-1	-1
权值分布 $D_4$	1/6	1/6	11/114	11/114	7/114	11/114	7/114	7/114	1/6	1/38
$\text{sign}(f_3(x))$	1	1	-1	-1	-1	-1	-1	-1	-1	-1

可得分类函数： $f_3(x) = 0.4236H_1(x) + 0.6496H_2(x) + 0.9229H_3(x)$ 。此时，组合三个基本分类器 $\text{sign}(f_3(x))$ 作为强分类器，在训练数据集上有0个误分类点。至此，整个训练过程结束。

整合所有分类器，可得最终的强分类器为：

$$H_{\text{final}} = \text{sign}\left(\sum_{i=1}^T \alpha_i H_i(x)\right) = \text{sign}(0.4236H_1(x) + 0.6496H_2(x) + 0.9229H_3(x))$$

**这个强分类器 $H_{\text{final}}$ 对训练样本的错误率为0！**

本例Matlab代码，如下：

先建立Matlab函数文件，定义h1,h2和h3三个弱分类器

```

1 function kind = wcH1( X,TH )
2 %h1弱分类器
3 X1=X(1);
4 X2=X(2);
5 if X1<TH
6     kind=1;
7 else
8     kind=-1;
9 end

```

```

1 function kind = wcH2( X,TH )
2 %h2弱分类器
3 X1=X(1);
4 X2=X(2);
5 if X1<TH
6     kind=1;
7 else
8     kind=-1;
9 end
10 end

```

```

1 function kind = wcH3( X,TH )
2 %h3弱分类器
3 X1=X(1);
4 X2=X(2);
5 if X2<TH
6     kind=-1;
7 else
8     kind=1;
9 end
10 end
11

```

主程序Matlab代码:

```

1 clc,clear all;
2 %% 训练样本数据
3 xData=[1 5;2 2;3 1;4 6;6 8;6 5;7 9;8 7;9 8;10 2] %样本数据点，对应编号为1,2,...10
4 Y=[1 1 -1 -1 1 -1 1 1 -1 -1]'; %对应的样本类别，用1和-1表示
5 xNum=1:10; %编号
6 format rat
7 %% 绘制样本分布图
8 L1=find(Y==1);
9 x=xData(L1,1);y=xData(L1,2);
10 plot(x,y,'b+', 'LineWidth',3, 'MarkerSize',12);
11 hold on;
12 L2=find(Y==-1);
13 x=xData(L2,1);y=xData(L2,2);
14 plot(x,y,'ro', 'LineWidth',3, 'MarkerSize',12);
15 xlabel('X1');ylabel('X2');axis([0 10 0 10])
16 %% *****初试过程*****
17 H1=zeros(10,1);H2=H1;H3=H1
18 for i=1:10
19     X=xData(i,:);
20     H1(i) = wcH1( X,2.5 );%弱分类器h1
21     H2(i) = wcH2( X,8.5 );%弱分类器h2
22     H3(i) = wcH3( X,6.5 );%弱分类器h3
23 end
24 errDataH1=find(H1~=Y);%找到被h1错分的样本点的序号
25 errDataH2=find(H2~=Y);%找到被h2错分的样本点的序号
26 errDataH3=find(H3~=Y);%找到被h3错分的样本点的序号
27 accDataH1=find(H1==Y);%找到被h1正确分的样本点的序号
28 accDataH2=find(H2==Y);%找到被h2正确分的样本点的序号
29 accDataH3=find(H3==Y);%找到被h3正确分的样本点的序号
30 errDataAll=[errDataH1,errDataH2,errDataH3];
31 accDataAll=[accDataH1,accDataH2,accDataH3];
32
33 N=10;
34 D1=zeros(10,1)+1/N % 初始化权值分布
35 %% *****第一次迭代*****
36 err1=sum(D1(errDataH1,:));%所有被错分类的样本点的权值之和即为误差率
37 err2=sum(D1(errDataH2,:));%所有被错分类的样本点的权值之和即为误差率
38 err3=sum(D1(errDataH3,:));%所有被错分类的样本点的权值之和即为误差率
39 errAll=[err1,err2,err3];
40 [minErr,minIndex]=min(errAll);
41 %根据误差率e1计算H1的系数:
42 a1=0.5*log((1-minErr)/minErr)
43 minErrData=errDataAll(:,minIndex);
44 minAccData=accDataAll(:,minIndex);
45 D2=D1;

```

```

48 | end
49 | for i=minErrData'
50 |     D2(i)=D2(i)/(2*minErr);
51 | end
52 | D2
53 | %分类函数
54 | f1=a1.*H1;
55 | kindFinal=sign(f1)%此时强分类器的分类结果
56 |
57 | %% *****第二次迭代*****
58 | err1=sum(D2(errDataH1,:));%所有被错分类的样本点的权值之和即为误差率
59 | err2=sum(D2(errDataH2,:));%所有被错分类的样本点的权值之和即为误差率
60 | err3=sum(D2(errDataH3,:));%所有被错分类的样本点的权值之和即为误差率
61 | errAll=[err1,err2,err3];
62 | [minErr,minIndex]=min(errAll);
63 | % 根据误差率e2计算H2的系数:
64 | a2=0.5*log((1-minErr)/minErr)
65 | minErrData=errDataAll(:,minIndex);
66 | minAccData=accDataAll(:,minIndex);
67 | D3=D2;
68 | for i=minAccData'
69 |     D3(i)=D3(i)/(2*(1-minErr));
70 | end
71 | for i=minErrData'
72 |     D3(i)=D3(i)/(2*minErr);
73 | end
74 | D3
75 | % 分类函数
76 | f2=a1.*H1+a2.*H2;
77 | kindFinal=sign(f2)%此时强分类器的分类结果
78 |
79 | %% *****第三次迭代*****
80 | err1=sum(D3(errDataH1,:));%所有被错分类的样本点的权值之和即为误差率
81 | err2=sum(D3(errDataH2,:));%所有被错分类的样本点的权值之和即为误差率
82 | err3=sum(D3(errDataH3,:));%所有被错分类的样本点的权值之和即为误差率
83 | errAll=[err1,err2,err3];
84 | [minErr,minIndex]=min(errAll);
85 | % 根据误差率e3计算G3的系数:
86 | a3=0.5*log((1-minErr)/minErr)
87 | minErrData=errDataAll(:,minIndex);
88 | minAccData=accDataAll(:,minIndex);
89 | D4=D3;
90 | for i=minAccData'
91 |     D4(i)=D4(i)/(2*(1-minErr));
92 | end
93 | for i=minErrData'
94 |     D4(i)=D4(i)/(2*minErr);
95 | end
96 | D4
97 | % 分类函数
98 | f3=a1.*H1+a2.*H2+a3.*H3;
99 | kindFinal=sign(f3)%此时强分类器的分类结果
100 | %%

```

110

55

Adaboost算法的某些特性是非常好的，这里主要介绍Adaboost的两个特性。（1）是训练的错误率上界，随着迭代次数的增加，会逐渐下降；（2）是Adaboost算法即使训练次数很多，也不会出现过拟合的问题。关于这两方面的研究和分析，我建议各大网友，还是看看大神的博客：[http://blog.csdn.net/v\\_july\\_v/article/details/40718799](http://blog.csdn.net/v_july_v/article/details/40718799)

#### 四、AdaBoost的优点和缺点

##### 优点

- （1）Adaboost提供一种框架，在框架内可以使用各种方法构建子分类器。可以使用简单的弱分类器，不用对特征进行筛选，也不存在过拟合的现象。
- （2）Adaboost算法不需要弱分类器的先验知识，最后得到的强分类器的分类精度依赖于所有弱分类器。无论是应用于人造数据还是真实数据，Adaboost都能显著的提高学习精度。
- （3）Adaboost算法不需要预先知道弱分类器的错误率上限，且最后得到的强分类器的分类精度依赖于所有弱分类器的分类精度，可以深挖分类器的能力。Adaboost可以根据弱分类器的反馈，自适应地调整假定的错误率，执行的效率高。
- （4）Adaboost对同一个训练样本集训练不同的弱分类器，按照一定的方法把这些弱分类器集合起来，构造一个分类能力很强的强分类器，即“三个臭皮匠赛过一个诸葛亮”。

##### 缺点：

在Adaboost训练过程中，Adaboost会使得难于分类样本的权值呈指数增长，训练将会过于偏向这类困难的样本，导致Adaboost算法易受噪声干扰。此外，Adaboost依赖于弱分类器，而弱分类器的训练时间往往很长。







如果你觉得该帖子帮到你，还望贵人多多支持，鄙人会再接再厉，继续努力的~




110
55

想对作者说点什么？[我来说一句](#)

- **qq\_42067384**： 很详细，通俗易懂，感谢博主 （2天前 #38楼）
- **cwang\_running**： 博主你好，请问你的附图是用什么工具绘制的？ （3周前 #37楼） [查看回复\(1\)](#)
- **上官玉瑾**： 我也想问一下那个Zt是怎么算出来的？按理说zt应该是和判决后的正确与错误的个数有关吧，怎么能直接就等于一个固定得表达式呢 （3周前 #36楼） [查看回复\(3\)](#)
- **一慕寒衣入画**： 感觉第二次迭代计算第五个样本时有点问题。根据H i（x）的结果，0.4236\*（-1）+0.6496\*1+0.9229\*（-1）计算sian之后不是-1吗？请博主指点一下谢谢 （1个月前 #35楼）  
[查看 53 条热评](#)

**AdaBoost--从原理到实现（Code：Python）**  7497

对于Adaboost，可以说是久闻大名，据说在Deep Learning出来之前，SVM和Adaboost是效果最好的 两个算法，而Adaboost是提升树(boosting tree)，...

**Adaboost算法原理分析和实例+代码**  425

【尊重原创，转载请注明出处】 <http://blog.csdn.net/guyuealian/article/details/70995333> 本人最初了解AdaBoost算法着实是花了几天...

**机器学习算法原理与实践（四）、AdaBoost算法详解与实战**  5535


【原创】Liu\_LongPo 转载请注明出处 【CSDN】 <http://blog.csdn.net/llp1992AdaBoost>算法是基于单层决策树等弱分类算法的强学习分类算法。单层决...

**Hoeffding不等式**  1.4万

Hoeffding不等式是关于一组随机变量均值的概率不等式。如果X1,X2,...,Xn为一组独立同分布的参数为p的伯努利分布随机变量，n为随机变量的个数。...

[下载](#) **Adaboost**的几个Python例子 12-17

adaboost的几个Python例子，包含训练数据，测试数据，Python代码，下载即可运行

**VS2013无法打开源文件stdafx.h**  5191


第一次用VS2013，刚开始就不是很顺利，移植一个CvImage类（OpenCv用的），总是报错，显示无法打开源文件stdafx.h，项目中也有stdafx.h这个文...

**数据挖掘领域十大经典算法之一AdaBoost算法（超详细附代码）**  3655

相关文章：数据挖掘领域十大经典算法之一C4.5算法（超详细附代码） 数据挖掘领域十大经典算法之一K-Means算法（超详细附代码） 数据挖掘领域十...

**Adaboost入门教程——最通俗易懂的原理介绍（图文实例）**  1.8万

Adaboost入门教程——最通俗易懂的原理介绍（图文实例）

**AdaBoost（Adaptive Boosting）算法详解**  1615

AdaBoost（Adaptive Boosting）算法详解 主要内容 AdaBoost算法详解 AdaBoost算法实例讲解 1、AdaBoost算法详解 假设有一位患者，存在某些...

VS2013无法打开源文件stdafx.h

博主推荐



打工是不可能打工滴

关注

154 篇文章



mjiansun

关注



358 篇文章




Liu\_LongPo

关注

106 篇文章

**AdaBoost(自适应增强算法)**   2528



AdaBoost(自适应增强算法)AdaBoost的目标是通过一个弱分类器构建一个强分类器，AdaBoost的大致运行过程：训练数据中的每一个样本，并赋予其一...

**机器学习之AdaBoost元算法（七）**   1582


主要内容：

- 组合相似的分类器来提高分类器性能
- 应用AdaBoost算法
- 处理非均衡问题分类问题



打个比方， 做重要决定的时候， 大家可能会汲取多...

**adaboost原理（包含权重详细解释）**   1.7万



1.1 Adaboost是什么 Adaboost，是英文"Adaptive Boosting"（自适应增强）的缩写，由Yoav Freund和Robert Schapire在1995年提出。它的自...

**深入浅出的 Adaboost算法**   340



转自知乎：<https://zhuanlan.zhihu.com/p/32960452> 通俗易懂 <https://zhuanlan.zhihu.com/p/306762...>

**Adaboost 算法的原理与推导**   9.4万



Adaboost 算法的原理与推导 0 引言 一直想写Adaboost来着，但迟迟未能动笔。其算法思想虽然简单：听取多人意见，最后综合决策，但一般书上...

**机器学习--Adaboost 算法的原理与推导**   4645


0 引言 一直想写Adaboost来着，但迟迟未能动笔。其算法思想虽然简单：听取多人意见，最后综合决策，但一般书上对其算法的流程描述实在是过于...

**机器学习(20)之Adaboost算法原理小结**   1710



微信公众号 关键字全网搜索最新排名 【机器学习算法】：排名第一 【机器学习】：排名第二 【Python】：排名第三 【算法】：排名第四 前言 在（机...

**随机森林算法学习(RandomForest)**   6.1万

随机森林算法学习最近在做kaggle的时候，发现随机森林这个算法在分类问题上效果十分的好，大多数情况下效果远要比svm，log回归，knn等算法效果...

**极大似然估计详解**   9.7万

极大似然估计 以前多次接触过极大似然估计，但一直都不太明白到底什么原理，最近在看贝叶斯分类，对极大似然估计有了新的认识，总结如下： ...

**详解最大似然估计（MLE）、最大后验概率估计（MAP），以及贝叶斯公式的理解**   4.4万

声明：本文为原创文章，发表于nebulaf91的csdn博客。欢迎转载，但请务必保留本信息，注明文章出处。本文作者: nebulaf91 本文原始地址：最大似...



pan\_jinquan

关注

 博客专家

向TA提问

原创	粉丝	喜欢	评论
132	604	452	330


等级：

博客 5

访问：99万+

积分：9136

排名：2955

勋章：

- 最新文章
- [bilateral filter双边滤波器的通俗理解](#)
  - [YUV420与YUV444互转，YUV420与YUV444读取和保存，YUV的显示和播放功能](#)
  - [tensorflow实现将ckpt转pb文件](#)
  - [Android使用NDK OpenGL ES3.0绘制一个三角形](#)

博主专栏



图像处理

阅读量：154939

26 篇



机器学习

阅读量：13746

20 篇

个人分类

图像处理	18篇
TensoFlow	7篇
机器学习	41篇
OpenCV	28篇
OpenGL	1篇

展开

归档

2018年10月	1篇
2018年9月	3篇
2018年8月	5篇
2018年7月	2篇
2018年6月	4篇

展开

最新评论

- Adaboost算法原理分析和实例...

Kevin\_LeeLee：[reply]guyuealian[/reply] 这个推导公式成立是有条件的，因为公式是大于等于...
- Tomcat 输入http://l...

weixin\_41660213：我来补充一个我自己遇到情况，就是 虽然localhost:8080访问报404，但是实际的服务器是没...
- String内容不能改变的理解 S...

qq\_29778641：很好，很清晰的逻辑
- OpenCV Mat类详解和用法

qwe1234asdz：写的很全，真好
- Adaboost算法原理分析和实例...

qq\_42067384：很详细，通俗易懂，感谢博主

置顶文章

热门文章

- Java单链表反转 详细过程

阅读量：49213
- 简单操作只需10秒破解PDF加密文件

阅读量：46234
- Adaboost算法原理分析和实例+代码（简明易懂）

阅读量：45960
- Java实现数组反转翻转的方法

阅读量：45841
- Eclipse配置Tomcat服务器，通用方法

阅读量：45559

联系我们



京ICP证09002463号 百度提供站内搜索  
©2018 CSDN版权所有  
kefu@csdn.net 400-660-0108  
QQ客服 客服论坛  
经营性网站备案信息 网络110报警服务  
中国互联网举报中心 北京互联网违法和不良信息举报中心



110
55