# 基于自适应权值裁剪的Adaboost快速训练算法

余陆斌 杜启亮* 田联房
(华南理工大学自动化科学与工程学院 广州 510640)

**摘 要**：Adaboost是一种广泛使用的机器学习算法，然而Adaboost算法在训练时耗时十分严重。针对该问题，该文提出一种基于自适应权值的Adaboost快速训练算法AWTAdaboost。该算法首先统计每一轮迭代的样本权值分布，再结合当前样本权值的最大值和样本集规模计算出裁剪系数，权值小于裁剪系数的样本将不参与训练，进而加快了训练速度。在INRIA数据集和自定义数据集上的实验表明，该文算法能在保证检测效果的情况下大幅加快训练速度，相比于其他快速训练算法，在训练时间接近的情况下有更好的检测效果。

**关键词**：目标检测；Adaboost算法；快速训练；自适应；权值分布
**中图分类号**：TP391 **文献标识码**：A **文章编号**：1009-5896(2020)11-2742-07
**DOI**: 10.11999/JEIT190473

# Fast Training Adaboost Algorithm Based on Adaptive Weight Trimming

YU Lubin DU Qiliang TIAN Lianfang
(*College of Automation Science and Engineering, South China University of Technology, Guangzhou* 510640, *China*)

**Abstract**: The Adaboost algorithm provides noteworthy benefits over the traditional machine algorithms for numerous applications, including face recognition, text recognition, and pedestrian detection. However, it takes a lot of time during the training process that affects the overall performance. Adaboost fast training algorithm based on adaptive weight (Adaptable Weight Trimming Adaboost, AWTAdaboost) is proposed in this work to address the aforementioned issue. First, the algorithm counts the current sample weight distribution of each iteration. Then, it combines the maximum value of current sample weights with data size to calculate the adaptable coefficients. The sample whose weight is less than the adaptable coefficients is discarded, that speeds up the training. The experimental results validate that it can significantly speed up the training speed while ensuring the detection effect. Compared with other fast training algorithms, the detection effect is better when the training time is close to each other.

**Key words**: Object detection; Adaboost algorithm; Fast traing; Adaptive; Weight distribution

## 1 引言

Boosting是一种重要的机器学习算法，其基本思想来源于PAC(Probably Approximately Correct)学习模型[1]。Kearns等人[2]首次提出了弱学习算法能否等价于强学习算法的问题。Schapire[3]通过构造性方法最终证明：一个概念是弱可学习的等价于它是强可学习的，在此基础上提出了最初的Boosting算法。但是Boosting算法要求预知分类器错误率上限，因此难以在实际应用问题中应用。Freund等人[4]对Boosting算法进行改进，提出Hedge($\beta$)算法。但算法关键参数$\beta$依赖于先验知识。Freund等人[5]提出AdaBoost(Adaptive Boosting)算法，该算法无需知道分类器错误率上限，结构简单，得到广泛应用。

目前Adaboost已经在很多领域有着广泛应用，如人脸识别[6]、文字识别[7]、行人检测[8]等等。然而训练样本和特征较多时，Adaboost算法训练

收稿日期：2019-06-27；改回日期：2020-04-19；网络出版：2020-08-31
*通信作者： 杜启亮 qldu@scut.edu.cn
基金项目：海防公益类项目(201505002)，广东省重点研发计划-新一代人工智能(20180109)，广州市产业技术重大攻关计划(2019-01-01-12-1006-0001)，广东省科学技术厅重大科技计划项目(2016B090912001)，中央高校基本科研业务费专项资金(2018KZ05)
Foundation Items: The Coast defence Public Welfare Project (201505002), Guangdong Province Key R&D Program-A New Generation of Artificial Intelligence (20180109), Guangzhou City Industrial Technology Major Research Project (2019-01-01-12-1006-0001), The Major Science and Technology Plan Project of Guangdong Science and Technology Department (2016B090912001), The Special Fund for Basic Scientific Research in Central Colleges and Universities (2018KZ05)

时间过长。针对上述特点，Friedman等人[9]提出静态权重裁剪法SWTAdaboost(Static Weight Trimming Adaboost)。SWTAdaboost的改进在于调整Adaboost算法迭代过程中的样本分布，设每轮迭代不参与训练的样本权值之和为裁剪系数$\beta$，根据$\beta$计算裁剪阈值$T(\beta)$，样本权值小于$T(\beta)$的样本不参与本轮弱分类器训练。但是针对不同训练集，SWTAdaboost需要选择合适的$\beta$，若$\beta$过大容易导致训练提前结束，影响最终强分类器的效果。贾慧星等人[10]在SWTAdaboost的基础上提出动态权重裁剪法DWTAdaboost(Dynamic Weight Trimming Adaboost)。当训练提前停止时，DWTAdaboost减小$\beta$并重新训练弱分类器。但是DWTAdaboost仍需预先设定$\beta$，若预设$\beta$不当会出现加速效果不明显、算法性能退化等问题。Seyedhosseini等人[11]提出WNS-Adaboost算法。WNS-Adaboost算法在Adaboost算法训练前进行样本权值选择，筛选训练集中样本间距离最大的样本得到新的样本集$X^R$来加速训练。然而样本权值选择过程需要消耗时间，而且在高维样本空间中根据样本间距离筛选样本容易减少对分类器生成有关键影响的样本。Tolba等人[12]提出GAdaboost算法。在正式训练开始前，GAdaboost先进行预训练。预训练使用遗传算法通过数轮迭代选择合适分类的特征。在正式训练开始时，根据预训练的结果选择部分特征参与训练，达到加速训练的效果。GAdaboost的一个问题是预训练过程迭代次数过少难以筛选出具有区分度的特征。利用预训练筛选出的特征进行正式训练往往会导致测试集上错误率偏高。袁双等人[13,14]提出PCA+DRAdaboost算法，DRAdaboost算法通过优化搜索阈值结合PCA降维样本特征来加速训练过程，同时限制权值扩张来避免过拟合的问题，但是限制权值扩张和特征降维会降低对疑难样本的分类能力。

本文提出自适应权值裁剪算法AWTAdaboost，在训练过程中统计每一轮迭代的样本权值分布，结合当前样本权值的最大值和样本集规模计算出裁剪系数，训练时使权值小于裁剪系数的样本不参与训练来加快训练速度，同时保证检测效果。

## 2　Adaboost算法原理分析

### 2.1　Adaboost算法基本原理

Adaboost算法首先对训练集的所有样本都赋予初始权值，在每轮迭代中基于当前样本分布调用弱学习算法生成弱分类器。在迭代中降低被正确分类样本的权值，增加被错误分类的样本权值。迭代完成后，将弱分类器按错误率加权组合成一个强分类器。

Adaboost算法流程如下：

输入：训练集$D=\{(x_1,y_1),(x_2,y_2),\cdots,(x_m,y_m)\}$，其中$x_i\in X$, $X$为训练样本集合，$y_i\in Y=\{-1,1\}$；弱学习算法WeakLearn；迭代训练轮数$N$。

初始化：令每个样本的初始权值为$d_1(i)=1/m$。

(1) For $n=1,2,\cdots,N$

(2) 基于当前分布$D_n$调用WeakLearn，得到弱分类器$h_n$。

(3) 计算弱分类器$h_n$在当前分布下的错误率：

$$\varepsilon_n=\sum_{i:h_n(x_i)\neq y_i} d_n(i) \tag{1}$$

如果$\varepsilon_n>0.5$，则令$N=n-1$，同时停止迭代。

(4) 计算分类器$h_n$在最终分类器集合中的加权系数：

$$\alpha_n=\frac{1}{2}\ln\left(\frac{1-\varepsilon_n}{\varepsilon_n}\right) \tag{2}$$

(5) 更新样本分布：

$$\begin{aligned}D_{n+1}(i)&=\frac{D_n(i)}{Z_n}\times\begin{cases}\exp(-\alpha_n), & h_n(x_i)=y_i\\ \exp(\alpha_n), & h_n(x_i)\neq y_i\end{cases}\\&=\frac{D_n(i)\exp(-\alpha_n y_i h_n(x_i))}{Z_n}\end{aligned} \tag{3}$$

(6) End

输出：

$$H(x)=\mathrm{sign}\left(\sum_{n=1}^{N}\alpha_n h_n(x)\right) \tag{4}$$

### 2.2　改进的快速Adaboost算法分析

Adaboost算法在每轮迭代后都会降低正确分类样本的权值，增加错误分类样本的权值。因此多次迭代后每次都正确分类的样本对分类器生成起到的作用很小。Friedman等人[9]提出SWTAdaboost算法，裁剪对分类器影响很小的样本。贾慧星等人[10]针对SWTAdaboost容易提前停止迭代的问题，提出DWTAdaboost。

DWTAdaboost算法的改进在于Adaboost算法第(2)，第(3)步中选取样本的规则：

对于第(2)步中，利用当前的样本分布$D_n$和裁剪比例$\beta$计算裁剪阈值$T(\beta)$，抽取权值大于$T(\beta)$的样本形成集合$D_n^{\beta}$，基于新的样本集合$D_n^{\beta}$调用WeakLearn生成弱分类器$h_n$。$T(\beta)$的计算规则为

$$\sum_{i=1}^{m}[d_n(i)<T(\beta)]=\beta \tag{5}$$

对于第(3)步中，计算分类器$h_n$在当前分布$D_n$下的错误率

$$\varepsilon_n=\sum_{i:h_n(x_i)\neq y_i} d_n(i) \tag{6}$$

当$\varepsilon_n \geq 0.5$且$D^\beta = D$，则停止迭代，令$N = n-1$。当$\varepsilon_n \geq 0.5$且$D^\beta \neq D$，则令$\beta = \beta/2$，转至(2)。

DWTAdaboost在迭代提前结束时令$\beta = \beta/2$，并重新进行训练使迭代能继续进行。但仍没解决需要根据不同数据集设定不同$\beta$的问题。当预设$\beta$过大时，DWTAdaboost算法需要多次减小$\beta$重新开始迭代，导致训练速度降低。

## 3 AWTAdaboost算法

### 3.1 AWTAdaboost算法实现

针对DWTAdaboost仍需根据不同数据集选择不同$\beta$的问题，本文提出根据每轮样本权值分布自适应选择参与训练样本的改进Adaboost算法——AWTAdaboost算法。对于大规模样本集，样本的权值分布分散，平均样本权值较小，因此裁剪阈值应较小；对于小规模样本集，样本的权值分布集中，平均样本权值较大，因此裁剪阈值应较大。随着迭代进行，样本权值分布会发生很大变化，易分类样本权值降低，难分类样本权值升高，因此迭代初期应保留较多样本参与训练，随着迭代进行逐渐减少保留样本。

AWTAdaboost算法的改进主要在于Adaboost算法的第(2)，第(3)步，每当一轮迭代开始时，寻找所有样本权值中的最大值$\max(d_n)$，根据最大值$\max(d_n)$和训练集规模系数$K/m$计算裁剪阈值$T(\max_n)$。训练时根据$T(\max_n)$自适应裁剪参与训练的样本，保留影响较大的样本参与每轮迭代，保证了分类器的检测效果，避免了裁剪过度导致的训练提前停止和裁剪不足导致的加速效果不明显。

AWTAdaboost算法流程如下：

输入：训练集$D$=$\{(x_1,y_1),(x_2,y_2),\cdots,(x_m,y_m)\}$，其中$x_i \in X$, $X$为训练样本集合，$y_i \in Y = \{-1,1\}$；弱学习算法WeakLearn；迭代训练轮数$N$。

初始化：令每个样本的初始权值为$d_1(i) = 1/m$。

(1) For $n = 1,2,\cdots,N$

(2) 利用当前的样本分布$D_n$和训练集样本个数$m$，计算裁剪阈值$T(\max_n)$，$T(\max_n)$的计算规则为

$$T(\max_n) = K/m \times \max(d_n) \tag{7}$$

其中$\max(d_n)$为第$n$轮迭代中所有样本权值的最大值；$K$为调节系数，取5。

(3) 抽取权值大于$T(\max_n)$的样本形成新分布$D_n^{\max}$，基于$D_n^{\max}$调用WeakLearn得到弱分类器$h_n^{\max}$。

(4) 计算分类器$h_n^{\max}$在当前分布$D_n$下的错误率

$$\varepsilon_n^{\max} = \sum_{i:h_n^{\max}(x_i)\neq y_i} d_n(i) \tag{8}$$

当$\varepsilon_n^{\max} \geq 0.5$且$D^{\max} = D$，则令$N = n-1$，停止迭代。当$\varepsilon_n^{\max} \geq 0.5$且$D^{\max} \neq D$，则令$T(\max_n)$=0，转至步骤(3)。

(5) 计算分类器$h_n^{\max}$在最终分类器集合中的加权系数

$$\alpha_n^{\max} = \frac{1}{2}\ln\left(\frac{1-\varepsilon_n^{\max}}{\varepsilon_n^{\max}}\right) \tag{9}$$

(6) 更新样本分布

$$\begin{aligned} D_{n+1}(i) &= \frac{D_n(i)}{Z_n} \times \begin{cases} \exp(-\alpha_n^{\max}), & h_n^{\max}(x_i) = y_i \\ \exp(\alpha_n^{\max}), & h_n^{\max}(x_i) \neq y_i \end{cases} \\ &= \frac{D_n(i)\exp(-\alpha_n^{\max} y_i h_n^{\max}(x_i))}{Z_n} \end{aligned} \tag{10}$$

(7) END

输出：

$$H(x) = \mathrm{sign}\left(\sum_{n=1}^{N} \alpha_n^{\max} h_n^{\max}(x)\right) \tag{11}$$

式(7)中的$K$根据统计大量样本分布得到的。$K$的取值范围一般在5～10, $K$越大，裁剪的样本越多，训练时间越短，但会稍稍降低准确率；$K$越小，裁剪的样本越少，训练时间越长，准确率稍有提高。

### 3.2 AWTAdaboost算法错误率分析

下面分析AWTAdaboost算法的错误率能否满足要求。

AWTAdaboost算法在训练集上的错误率$\varepsilon$为

$$\varepsilon = \frac{1}{m}\sum_{i=1}^{m} \|H(x_i) \neq y_i\| \tag{12}$$

其中记号$\|\bullet\|$表示：当$\bullet$成立时$\|\bullet\| = 1$，否则$\|\bullet\| = 0$。

令 $f(x) = \sum_{n=1}^{N} \alpha_n^{\max} h_n^{\max}(x)$，有$H(x) = \mathrm{sign}(f(x))$。

当$H(x_i) \neq y_i$时，有$y_i f(x_i) \leq 0$，此时$\exp(-y_i f(x_i)) \geq 1$，有

$$\|H(x_i) \neq y_i\| \leq \exp(-y_i f(x_i)) \tag{13}$$

根据式(10)有

$$\begin{aligned} D_{n+1}(i) &= \frac{D_n(i)\exp(-\alpha_n^{\max} y_i h_n^{\max}(x_i))}{Z_n} \\ &= \frac{\exp\left(-\sum_n \alpha_n^{\max} y_i h_n^{\max}(x_i)\right)}{m\prod_n Z_n} \\ &= \frac{\exp(-y_i f(x_i))}{m\prod_n Z_n} \end{aligned} \tag{14}$$

根据式(13)，式(14)以及样本分布定义$\sum_{i=1}^{m} D_{n+1}(i) = 1$，有

$$\frac{1}{m}\sum_{i=1}^{m}\|H(x_i)\neq y_i\|\leq\frac{1}{m}\sum_i\exp(-y_if(x_i))$$
$$=\frac{1}{m}\sum_i D_{n+1}(i)m\prod_n Z_n=\prod_n Z_n \tag{15}$$

在$D_n(i)$更新后为使$D_{n+1}(i)$成为一个新的概率分布有

$$D_{n+1}(i)=\frac{D_n(i)}{\sum_{i=1}^{N}D_n(i)} \tag{16}$$

联系式(14)和式(16)，有

$$D_{n+1}(i)=\frac{D_n(i)}{\sum_{i=1}^{N}D_n(i)}=\frac{D_n(i)\exp(-\alpha_n^{\max}y_ih_n^{\max}(x_i))}{Z_n} \tag{17}$$

根据式(17)有

$$\begin{aligned}Z_n&=\sum_{i=1}^{N}D_n(i)\exp(-\alpha_n^{\max}y_ih_n^{\max}(x_i))\\&=\sum_{y_i=h_n^{\max}(x_i)}D_n(i)\mathrm{e}^{-\alpha_n^{\max}}+\sum_{y_i\neq h_n^{\max}(x_i)}D_n(i)\mathrm{e}^{\alpha_n^{\max}}\\&=(1-\varepsilon_n^{\max})\mathrm{e}^{-\alpha_n^{\max}}+\varepsilon_n^{\max}\mathrm{e}^{\alpha_n^{\max}}\\&=2\sqrt{\varepsilon_n^{\max}(1-\varepsilon_n^{\max})}\\&=\sqrt{1+2r}\end{aligned} \tag{18}$$

其中$r=-2\left(\frac{1}{2}-\varepsilon_n^{\max}\right)^2$

比较$\sqrt{1+2r}$和$\mathrm{e}^{-r}$的泰勒展开可以得出

$$Z_n=\sqrt{1+2r}\leq\mathrm{e}^{-r} \tag{19}$$

根据式(17)和式(19)得AWTAdaboost算法在训练集上的错误率$\varepsilon$为

$$\begin{aligned}\varepsilon&=\frac{1}{m}\sum_{i=1}^{m}\|H(x_i)\neq y_i\|\\&=\prod_n Z_n=n\sqrt{1+2r}\leq\mathrm{e}^{-nr}\end{aligned} \tag{20}$$

根据式(20)可得AWTAdaboost算法得到的分类器在训练集上的错误率有上界$\mathrm{e}^{-nr}$，该上界会随着迭代次数增加呈指数减小，因此随着迭代次数增加，AWTAdaboost算法错误率最终能满足需求。

## 4　实验研究与结果分析

行人检测是一个典型的目标检测问题。基于该问题对本文提出的AWTAdaboost算法进行对比测试。测试实验在INRIA数据集和自定义数据集上进行。INRIA数据集是Dalal等人[15]建立的行人检测通用数据库。自定义数据集如图1所示。自定义数据集是为了检测自动扶梯上的乘客而建立的数据库，其中正样本为自动扶梯上的乘客，负样本为自动扶梯环境中不包含乘客的其他元素。因自动扶梯场景复杂，自定义数据集比现有通用行人检测数据集包含了更多疑难样本。自定义数据集训练集包含1000个正样本，2000个负样本，测试集包含1000个正样本，2000个负样本。测试用的机器为3GHz CPU, 4GB内存，采用的MATLAB版本为2015a。在实验进行前已对数据集中的样本进行了检查与清理，确保不存在异常样本。

### 4.1　AWTAdaboost算法错误率试验研究

本节实验分别用Adaboost，SWTAdaboost算法，DWTAdaboost算法，WNS-Adaboost算法，GAdaboost算法，PCA+DRAdaboost算法和AWTAdaboost算法在INRIA数据集和自定义数据集上训练分类器，然后在测试集上进行检测得到错误率。其中SWTAdaboost算法，DWTAdaboost算法的$\beta$为0.2。训练时样本为64×128的归一化图像，选用特征为HOG特征[15]，总共维数是3780维。训练时弱分类器为二值分类器。在INRIA数据集上训练迭代次数为200次，在自定义数据集上训练迭代次数为100次。

**实验1**：各算法的错误率对比

图2和表1表明：在INRIA数据集上，AWTAdaboost算法测试集错误率比其他改进算法低。AWTAdaboost算法测试集错误率为0.0302，仅比Adaboost算法的错误率高0.0017。SWTAdaboost和DWTAdaboost的错误率高于AWTAdaboost算法。WNS-Adaboost算法错误率比AWTAdaboost算法高0.0054。GAdaboost算法错误率远高于其他算法。PCA+DRAdaboost算法错误率比AWTAdaboost算法高0.0111。

(a) 自定义数据集的正样本　　(b) 自定义数据集的负样本

图 1 自定义数据集样本示例

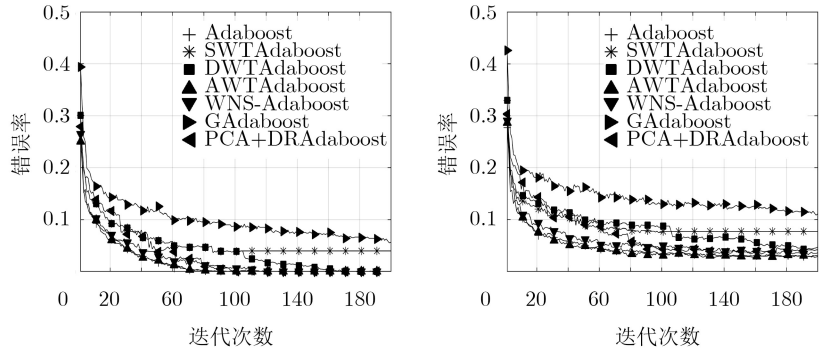
图3和表1表明：在自定义数据集上，AWTAdaboost算法的测试集错误率比其他改进算法低。AWTAdaboost算法的测试集错误率为0.0324，仅比Adaboost算法的错误率高0.0028。SWTAdaboost和DWTAdaboost仍高于本文提出的AWTAdaboost算法。WNS-Adaboost算法错误率比AWTAdaboost算法高0.0115。GAdaboost算法错误率有明显上升。PCA+DRAdaboost算法错误率比AWTAdaboost算法高0.0215。

### 4.2 AWTAdaboost算法训练时间损耗试验研究

本节实验分别用Adaboost算法，SWTAdaboost算法，DWTAdaboost算法，WNS-Adaboost算法，GAdaboost算法，PCA+DRAdaboost算法和AWTAdaboost算法在INRIA数据集和自定义数据集上训练分类器，然后记录每个算法的训练耗时。统计训练时间时，以Adaboost算法的实际训练时间作为1单位时间，其他算法训练的实际训练时间除以Adaboost算法的实际训练时间得到对应的相对训练时间。

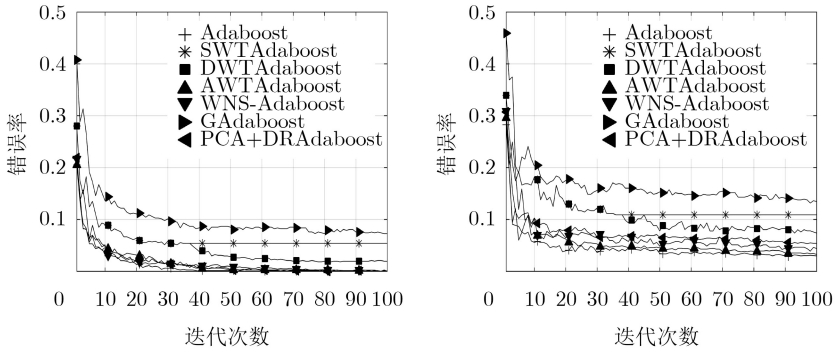
**实验2**：各算法训练时间对比

图4和表2表明：在INRIA数据集上，AWTAdaboost算法训练时间为Adaboost算法的55.70%。SWTAdaboost算法和DWTAdaboost算法的训练速度均比AWTAdaboost算法慢。WNS-Adaboost算法的训练速度稍慢于AWTAdaboost算法。GAdaboost算法的训练速度虽然快于AWTAdaboost算




(a) 训练集错误率 (b) 测试集错误率

图 2 各算法在INRIA数据集上的错误率

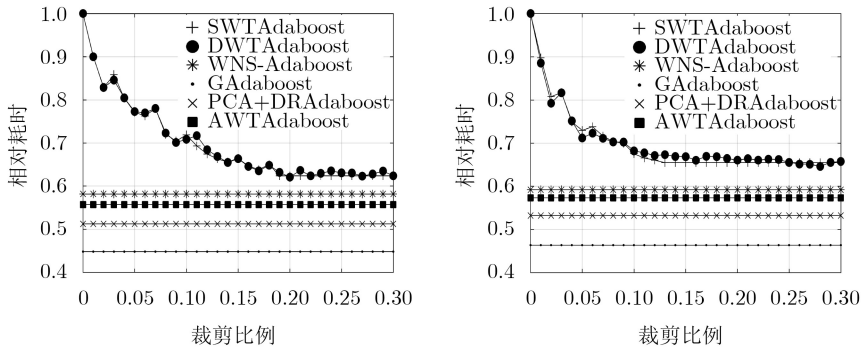



(a) 训练集错误率 (b) 测试集错误率

图 3 各算法在自定义数据集上的错误率

**表 1 各算法在两个数据集上的错误率**

| | INRIA数据集 | | 自定义数据集 | |
|---|---|---|---|---|
| | 训练集错误率 | 测试集错误率 | 训练集错误率 | 测试集错误率 |
| Adaboost | 0.0000 | 0.0285 | 0.0000 | 0.0296 |
| SWTAdaboost | 0.0395 | 0.0768 | 0.0538 | 0.1089 |
| DWTAdaboost | 0.0000 | 0.0466 | 0.0194 | 0.0735 |
| WNS-Adaboost | 0.0000 | 0.0356 | 0.0006 | 0.0439 |
| GAdaboost | 0.0563 | 0.1108 | 0.0724 | 0.1345 |
| PCA+DRAdaboost | 0.0000 | 0.0413 | 0.0000 | 0.0539 |
| AWTAdaboost | 0.0000 | 0.0302 | 0.0000 | 0.0324 |




(a) INRIA数据集训练时间　　(b) 自定义数据集训练时间

图 4 各算法的训练时间

**表 2　各算法训练时间对比**

| 算法 | INRIA数据集相对训练时间 | 自定义数据集相对训练时间 |
|---|---|---|
| Adaboost | 1.0000 | 1.0000 |
| SWTAdaboost | 0.6237 | 0.6547 |
| DWTAdaboost | 0.6347 | 0.6551 |
| WNS-Adaboost | 0.5814 | 0.5919 |
| GAdaboost | 0.4482 | 0.4636 |
| PCA+DRAdaboost | 0.5124 | 0.5324 |
| AWTAdaboost | 0.5570 | 0.5732 |

注：表中只记录了SWTAdaboost提前停止迭代前的训练时间和相同$\beta$下DWTAdaboost的训练时间。

法，但是其错误率远高于AWTAdaboost算法。PCA+DRAdaboost算法的训练速度虽然比AWTAdaboost算法快0.0446，但其错误率比AWTAdaboost算法高0.0111。

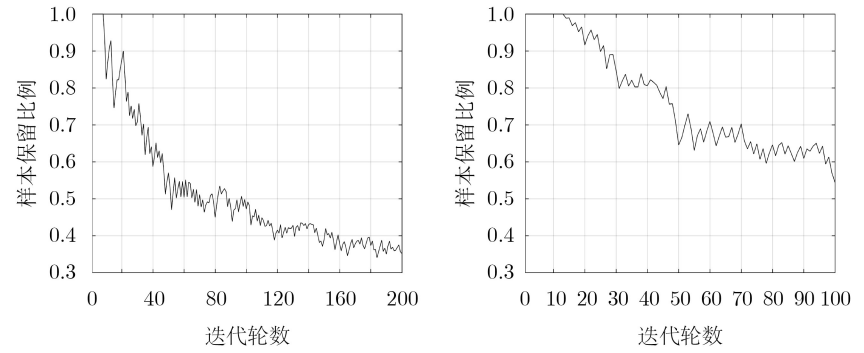
在自定义数据集上，因疑难样本较多，各个算法训练速度均有下降。AWTAdaboost算法训练时间为Adaboost算法的57.32%。SWTAdaboost和DWTAdaboost的训练速度均比AWTAdaboost算法慢。WNS-Adaboost算法的训练速度稍慢于AWTAdaboost算法。GAdaboost算法的训练速度虽然快于AWTAdaboost算法，但是其错误率远高于AWTAdaboost算法。虽然PCA+DRAdaboost算法的训练速度比AWTAdaboost算法快0.0408，但其错误率比AWTAdaboost算法高0.0215。

**实验3**：AWTAdaboost算法每轮训练保留样本比例统计

图5显示AWTAdaboost算法在不同数据集上每一轮训练所保留的样本比例。图5表明在没有预设裁剪系数的情况下，AWTAdaboost算法能够根据不同样本集，不同训练阶段合适地选择保留样本数目，在保证加速效果的同时不降低分类准确率。

### 4.3　实验结果分析

实验结果表明：AWTAdaboost算法的检测效果与Adaboost算法一致，同时大幅加快了训练速度，这是因为AWTAdaboost算法在每轮迭代时根据当前样本分布保留了合适样本，在加快训练速度的同时保证了检测效果。与SWTAdaboost算法和DWTAdaboost算法相比，AWTAdaboost算法的错误率更低，训练时间也更短。WNS-Adaboost算法减少了部分距离紧密但对分类器生成有重要影响的样本导致错误率上升，相比之下AWTAdaboost算法在错误率和训练时间方面都更有优势。虽然GAdaboost算法能大幅加快训练速度，但GAdaboost未筛选出适合分类的特征，导致错误率高于其他算法。PCA+DRAdaboost训练速度稍快于




(a) INRIA数据集上保留比例　　(b) 自定义数据集上保留比例

图 5 AWTAdaboost算法在训练时保留样本比例

AWTAdaboost，但是PCA+DRAdaboost为了避免过拟合，限制了样本权值的扩张，降低了对特殊样本的分类能力，导致错误率较高。

## 5 结论

为了加快Adaboost算法分类器训练速度同时避免检测性能下降，本文提出了一种新的自适应权值裁剪算法AWTAdaboost。AWTAdaboost在训练过程中统计每一轮迭代的样本权值分布，结合当前样本权值的最大值和样本集规模计算出裁剪系数，训练时使样本权值小于裁剪系数的样本不参与训练，保证了训练样本的有效性。AWTAdaboost的优势在于显著加快训练速度的同时能保证分类器检测效果，解决了现有算法在加快训练速度时往往会导致检测效果下降的问题。在INRIA数据集和自定义数据集上的实验结果表明：AWTAdaboost算法的错误率接近Adaboost算法，训练时间大幅减少。与SWTAdaboost算法，DWTAdaboost算法和WNS-Adaboost算法相比，AWTAdaboost的错误率更低，训练时间更短。与GAdaboost算法和PCA+DRAdaboost相比，AWTAdaboost的训练速度稍慢，但在错误率方面有较大优势。

## 参 考 文 献


[1] VALIANT L G. A theory of the learnable[C]. The 16th Annual ACM Symposium on Theory of Computing, New York, USA, 1984: 436–445.

[2] KEARNS M and VALIANT L. Cryptographic limitations on learning Boolean formulae and finite automata[J]. *Journal of the ACM*, 1994, 41(1): 67–95. doi: 10.1145/174644.174647.

[3] SCHAPIRE R E. The strength of weak learnability[J]. *Machine Learning*, 1990, 5(2): 197–227.

[4] FREUND Y and SCHAPIRE R E. A decision-theoretic generalization of on-line learning and an application to boosting[J]. *Journal of Computer and System Sciences*, 1997, 55(1): 119–139. doi: 10.1006/jcss.1997.1504.

[5] FREUND Y and SCHAPIRE R E. Experiments with a new boosting algorithm[C]. International Conference on Machine Learning, Bari, Italy, 1996: 148–156.

[6] ZHANG Xingqiang and DING Jiajun. An improved Adaboost face detection algorithm based on the different sample weights[C]. The 20th IEEE International Conference on Computer Supported Cooperative Work in Design, Nanchang, China, 2016: 436–439.

[7] CHO H, SUNG M, and JUN B. Canny text detector: Fast and robust scene text localization algorithm[C]. 2016 IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, USA, 2016: 3566–3573.

[8] GAO Chenqiang, LI Pei, ZHANG Yajun, *et al*. People counting based on head detection combining Adaboost and CNN in crowded surveillance environment[J]. *Neurocomputing*, 2016, 208: 108–116. doi: 10.1016/j.neucom.2016.01.097.

[9] FRIEDMAN J, HASTIE T, and TIBSHIRANI R. Additive logistic regression: A statistical view of boosting (With discussion and a rejoinder by the authors)[J]. *Annals of Statistics*, 2000, 28(2): 337–407.

[10] 贾慧星, 章毓晋. 基于动态权重裁剪的快速Adaboost训练算法[J]. 计算机学报, 2009, 32(2): 336–341. doi: 10.3724/SP.J.1016.2009.00336.

JIA Huixing and ZHANG Yujin. Fast Adaboost training algorithm by dynamic weight trimming[J]. *Chinese Journal of Computers*, 2009, 32(2): 336–341. doi: 10.3724/SP.J.1016.2009.00336.

[11] SEYEDHOSSEINI M, PAIVA A R C, and TASDIZEN T. Fast AdaBoost training using weighted novelty selection[C]. 2011 International Joint Conference on Neural Networks, San Jose, USA, 2011: 1245–1250.

[12] TOLBA M F and MOUSTAFA M. GAdaBoost: Accelerating adaboost feature selection with genetic algorithms[C]. The 8th International Joint Conference on Computational Intelligence, Porto, Portugal, 2016: 156-163.

[13] YUAN Shuang and LÜ Cixing. Fast adaboost algorithm based on weight constraints[C]. 2015 IEEE International Conference on Cyber Technology in Automation, Control, and Intelligent Systems, Shenyang, China, 2015: 825–828.

[14] 袁双, 吕赐兴. 基于PCA改进的快速Adaboost算法研究[J]. 科学技术与工程, 2015, 15(29): 62–66. doi: 10.3969/j.issn.1671-1815.2015.29.011.

YUAN Shuang and LÜ Cixing. Fast adaboost algorithm based on improved PCA[J]. *Science Technology and Engineering*, 2015, 15(29): 62–66. doi: 10.3969/j.issn.1671-1815.2015.29.011.

[15] DALAL N and TRIGGS B. Histograms of oriented gradients for human detection[C]. 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, San Diego, USA, 2005: 886–893.



余陆斌：男，1994年生，博士生，主要研究方向为机器学习、机器视觉.

杜启亮：男，1980年生，副研究员，博士，主要研究方向为机器人、机器视觉.

田联房：男，1968年生，教授，博士，主要研究方向为模式识别、人工智能.


责任编辑：陈 倩