

# 混合 Boost 算法实现的行人检测技术

陈 超

(内江师范学院数学与信息科学学院 四川 内江 641110)

**摘要** 传统 AdaBoost 存在一定的局限,比如训练分类器时对训练样本自身所带的噪声过于敏感,产生的分类器泛化能力不强和导致分类器过拟合问题,在训练分类器时只能静态分配分类器权重而不能自适应地对每个训练样本动态调整权重等问题。提出一种基于 SBoost 算法和 PBoost 算法,引入样本权重调节器、非平衡的样本采样、误差纠偏方法来检测潜在的样本。模拟实验表明:改进后的技术有效的提高了分类器的精确度且防止过拟合问题。

**关键词** SBoost 算法 PBoost 算法 混合 Boost 算法 动态权重调整 非平衡的样本采样 误差纠偏方法

**中图分类号** TP3 **文献标识码** A **DOI**:10.3969/j.issn.1000-386x.2019.06.035

## PEDESTRIAN DETECTION TECHNOLOGY BASED ON MIXTURE BOOST ALGORITHM

Chen Chao

(School of Maths and Informations Science, Neijiang Normal University, Neijiang 641110, Sichuan, China)

**Abstract** Traditional AdaBoost has some limitations, such as too sensitive to the noise of training samples, weak generalization ability of classifiers leading over-fitting. When training classifier, the weight of classifier can only be allocated statically, but the weight of each training sample cannot be dynamically adjusted adaptively. To solve this problem, this paper proposed a new method based on SBoost algorithm and Boost algorithm, and introduced sample weight adjuster, unbalanced sample sampling and the error correction method to detect potential samples. The simulation results show that the improved method can effectively improve the accuracy of the classifier and prevent over-fitting.

**Keywords** SBoost algorithm PBoost algorithm Mixture Boost algorithm Dynamic weight adjustment Unbalanced sample sampling Error correction method

## 0 引言

在现代智能小区中实现对行人的自动检测有着及其重要的意义。随着我国城市化进程的推进,一个小区的业主往往来自四面八方,彼此之间不认识,存在多个地下车库入口、门禁系统的不完善等问题,物业管理公司形似摆设,实现对行人的自动检测和识别有着举足轻重的作用。视频图像中检测和识别感兴趣目标成为研究热点和难点。不同于刚性目标有固定的形状和空间结构,行人目标存在着形变、部分遮挡和观察角度变化等问题会影响检测效果。当前较为经典的行人检测方法有 Wen 等<sup>[1-4]</sup>提出的一种对于汽车分类器的

快速学习算法、惩罚式 AdaBoost 算法、RBoost 算法、基于 SBoost 算法等。Shen 提出误差纠偏方法可以检测到潜在的样本有效的避免过拟合问题<sup>[5-6]</sup>。Chen 提出选择性组合弱分类器, Frenay 提出噪声标签分类器来提高分类器精准度<sup>[7]</sup>。Nguyen 提出了在二分分类器中的丢失性训练算法<sup>[8-11]</sup>。Patel 提出了渐进式学习分类器<sup>[12]</sup>。有学者提出 k-均值聚类算法初始化分类器等<sup>[13]</sup>。Zhu 提出对姿态估计等算法来改进 Boost 算法取得了较好效果,但是精准度和过拟合问题任然存在<sup>[14-15]</sup>。分别使用 MIT 行人数据库、Caltech 行人数据库和 INRIA 行人数据库上做对比实验。实验结果表明,提出的方法可以有效提高行人检测的精度和效率,同时有效防止分类器过拟合问题。

## 1 AdaBoost 算法原理及实现步骤

### 1.1 经典 AdaBoost 算法

作为自适应提升式 AdaBoost 算法,每次迭代得到的弱分类器组合成一个分类功能较大的强分类器,最后利用决策树生成目标分类器<sup>[1-3]</sup>。

### 1.2 基本训练分类器步骤

1) 输入为  $N$  个训练样本:

$$(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$$

$i = 1, 2, \dots, N, y_i = \{-1, 1\}$ , 其中  $-1$  表示非目标训练样本,  $+1$  表示目标训练样本;

2) 对于  $y_i = \{-1, 1\}$  分别进行初始化对应的权值:

$$w_{1i} = \frac{1}{2m} \quad w_{2i} = \frac{1}{2l}$$

式中:  $m$  表示集中训练的非目标样本的数目,  $l$  表示训练时候设定的目标训练样本的数目<sup>[1-3]</sup>。

3) 进行循环 (while  $t = 1, 2, \dots, T$ ):

(1) 权值归一化  $\bar{w}_{ii} = \frac{w_{ii}}{\sum_{j=1}^n w_{ij}}$ , 其中,  $n$  表示采取

总特征数;

(2) 为每个特征弱分类器  $h_i$ , 计算该当前权值的错误率:

$$\varepsilon_j = \sum_{i=1}^n \bar{w}_{ii} \cdot |h_j(x_i) - y_i|$$

(3) 选择最小的错误率  $\varepsilon$ , 所对应的判决函数作为本次迭代得到的弱分类器;

(4) 根据  $h_i$  对每个训练样本的判断情况, 修改样本的权值, 即:  $w_{i+1} = \bar{w}_{ii} \cdot \beta_i^{1-\varepsilon_i}$ , 其中, 若  $h_i$  对第  $i$  样本判断正确, 则  $\varepsilon_i = 0$ ; 反之  $\varepsilon_i = 1$ 。

4) 生成强分类器:

$$h(x) = \begin{cases} 1 & \sum_{i=1}^{T_i} \alpha_i h_i(x) \geq 0.5 \sum_{i=1}^{T_i} \alpha_i \\ 0 & \text{其他} \end{cases}$$

### 1.3 分类器设计

图像分类就是用 AdaBoost 算法实现目标和非目标的分类<sup>[11-14]</sup>。设图像样本的特征为  $X$ , 对应的分类标签为  $Y$ , 其中  $y = \{+1, -1\}$  分别对应正负样本。训练样本集  $S = \{(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)\}$ , 其中  $x_i \in X, y_i \in Y$ 。  $N$  是样本个数。在算法实现中, 取正样本为行人即没有背景的行人或者是背景很单一的行人; 负样本为没有行人的任何样本, 要求尺度大于正样

本的归一化尺度的灰度图像, 其算法具体步骤如下<sup>[1-3,7]</sup>:

1) 初始化权重, 对于每个  $(x_i, y_i) \in S$ , 权重初始化为:

$$D(x, y) = \begin{cases} \frac{1}{2m} & y = -1 \\ \frac{1}{2l} & y = +1 \end{cases}$$

2) 令  $t = 1$ :

(1) 选择弱分类器:  $h_i(x_i) = \begin{cases} 1 & \lambda_i x_i < \lambda_i \theta_i \\ 0 & \text{otherwise} \end{cases}$

式中:  $\theta_i$  是阈值, 默认取该类特征值的中值,  $\lambda_i \in \{-1, +1\}$  表示不等号的偏执方向, 根据样本权重分布  $D_i$  进行学习, 获得弱分类器  $h_i: X \rightarrow Y$ 。计算过检率:  $\varepsilon_i =$

$$\sum_{i: y_i \neq h_i(x_i)} D_i(x_i, y_i);$$

(2) 若  $\varepsilon_i < 0.5$ , 选择  $\alpha_i = \frac{1}{2} \ln \left[ \frac{1 - \varepsilon_i}{\varepsilon_i} \right]$ , 若  $\varepsilon_i \geq$

$0.5$ , 删除本次生成的弱分类器,  $t = t + 1$ , 返回(1)。

(3) 更新样本权重:

$$D_{t+1}(x_i, y_i) = \frac{D_t(x_i, y_i) e^{-\alpha_i h_i(x_i)}}{Z_t}$$

式中:  $Z_t$  是归一化因子, 使得总体权重之和为 1;

(4)  $t = t + 1$ , 设  $T$  为弱分类器最大训练的次数, 假如  $t = T$ , 则训练结束, 假如  $t < T$ , 返回(1)。

3) 组成强分类器:

使用倒置的决策树级联成具备图像目标分类功能的超级级联分类器。

## 2 混合 Boost 算法

### 2.1 PBoost 算法

一种带惩罚因子的 AdaBoost 算法通过概率中的边缘分布来优化精度 AdaBoost 算法的泛化误差<sup>[2]</sup>。其中样本权重如下:

$$\text{Margin}_T(x_i) = y_i \sum_{i=1}^T \alpha_i h_i(x) / \sum_{i=1}^T \alpha_i h_i(x) \quad (1)$$

要实现不平衡采样, 必须保证  $m$  和  $n$  要相差较大, 权重初始化<sup>[2]</sup>为:

$$w_{t-}^j = \frac{1}{2m} \quad y_i = -1$$

$$w_{t+}^j = \frac{1}{2n} \quad y_i = +1 \quad (2)$$

反馈因子:

$$m_{i,t} = \exp(-\text{Margin}_{t-1}(x_i)) / U_i \quad (3)$$

式中:  $U_i = \sum_i \exp(-Margin_{i-1}(x_i))$  对于每个  $(x_i, y_i) \in S$ , 假如负样本有  $m$  个, 正样本有  $n$  个, 所有样本之和为  $m+n$  个, 然后分别求和:

$$\begin{aligned} M_{i-}^j &= \sum_{y_i=-1} m_{i,t} \quad y_i = -1 \\ M_{i+}^j &= \sum_{y_i=+1} m_{i,t} \quad y_i = +1 \end{aligned} \quad (4)$$

弱分类器为:

$$f_i^j(x) = \begin{cases} (W_{i+}^j - W_{i-}^j)(1 - M_{i-}^j) & W_{i+}^j > W_{i-}^j \\ (W_{i+}^j + W_{i-}^j)(1 - M_{i+}^j) & \text{其他} \end{cases} \quad (5)$$

弱分类器的权重更新公式:

$$\omega_{i,t+1} = 1 \quad \sum_{p=1}^t f_p(x_i) = 0 \quad \omega_{i,t+1} > Q_{t+1} \wedge Margin_i(x_i) < 0 \quad (6)$$

其中:

$$Q_{t+1} = \max_i \left\{ w_{i,t+1} - \frac{\max_i \{w_{i,t+1}\} - \min_i \{w_{i,t+1}\}}{\gamma} \right\} \quad (7)$$

式中:  $\gamma$  为设置的一个整型参数, 本文取 50。

### 2.2 SBoost 算法

一个选择性的 Boost 算法<sup>[3]</sup>的重点在于提高分类器的泛化性能而不是在暴力训练的样本上, 暴力训练的惩罚因子被用来降低相关联的噪声程度。同时引入误差纠偏方法有效地检测到潜在清晰样本, 以致阻止了误分类器来避免过拟合问题。

### 2.3 混合 Boost 算法

**理论 1:** 在文献[3]中给出了权重训练误差  $\varphi_t = \sum_{i=1}^N L(i) [[h_t(x_i) \neq y_i]]$ , 对于任何的  $\delta, \bar{\rho}_t \leq \delta$  的概率为:

$$P(\bar{\rho}_t \leq \delta) \leq 2^T \prod_{i=1}^T \sqrt{\varphi_i^{1-\delta} (1 - \varphi_i)^{(1+\delta)}} \quad (8)$$

当  $\delta=0$  时,  $P(\bar{\rho}_t \leq 0)$  表示了样本被误分的概率, 我们现在的目的就是最小化  $P(\bar{\rho}_t \leq 0)$ <sup>[3]</sup>。

**理论 2:** 给出样本权重调节器, 使用  $\beta_t$  代替经典 AdaBoost 算法<sup>[3]</sup>中的  $h_t$ , 样本所占比例可以重新定义为:

$$\varphi_t = \frac{y_i \sum_{i=1}^T \beta_i h_t(x_i)}{\sum_{i=1}^T \beta_i} \quad (9)$$

第  $i$  个样本的累积权重计算为:

$$k_t(i) = \frac{\exp(\rho_t(x_i))}{\sum_{i=1}^N \exp(\rho_t(x_i))} \quad (10)$$

$$\varepsilon_t(i) = \sum_{j=1}^l (\beta_j k_t(i)) \quad (11)$$

$$\psi_t(i) = \frac{\varepsilon_t(i)}{\max\{\varepsilon_t(i)\}} \quad (12)$$

由于训练样本中可能存在含有噪声容易导致过拟合, 在此引入了  $K$ -领域方法来标记噪声样本和评价噪声程度。找出在训练集中一个特定的目标样本最近的  $K$  个样本, 在这  $K$  个样本中有  $P$  个样本含有同样的标签, 其中  $0 \leq P \leq K$ , 这些样本的噪声程度定义为:

$$\theta_i = 1 - \frac{P}{K} \quad (13)$$

其中噪声程度独立于循环次数, 每次迭代时使用式(14)来更新样本权重:

$$0 \leq \omega_t(i) \leq \beta_t \quad (14)$$

其中,

$$\begin{aligned} L_{t+1}(x_i, y_i) &= \frac{L_t^{(i)} \exp(-\beta_t h_t(x_i) y_i - \omega_t(i))}{J'_t} = \\ &= \frac{1}{N} \frac{\exp\left\{-\sum_{i=1}^T [y_i \beta_i h_t(x_i) y_i + \omega_t(i)]\right\}}{\prod_{i=1}^T J'_i} \end{aligned} \quad (15)$$

为 Sboost 算法的代价函数<sup>[3]</sup>。

$$J'_t = \varepsilon_t \exp(-\beta_t) + \tau_t \exp(\beta_t) \quad (16)$$

令式(16)等于 0, 并相对于  $\beta_t$  求得:

$$\beta_t = \frac{1}{2} \ln\left(\frac{\varepsilon_t}{\tau_t}\right) \quad (17)$$

把式(17)代入式(16)得到下式:

$$J'_t = 2\sqrt{\varepsilon_t \tau_t} \quad (18)$$

改进后的概率为<sup>[3]</sup>:

$$\begin{aligned} P[\varphi_t(x_i) \leq \gamma] &= \\ E\left[\exp\left(\gamma \sum_{i=1}^T \beta_i - \sum_{i=1}^T [y_i \beta_i h_t(x_i) + \omega_t(x_i)]\right)\right] &= \\ 2^T \prod_{i=1}^T \left(\sqrt{\frac{\varepsilon_i}{\tau_i} \varepsilon_i \tau_i}\right) & \quad (19) \end{aligned}$$

式中:  $\gamma$  是一个  $(-1, 1)$  之间的一个实参数。

**理论 3:** 误差纠偏方法表示如下:

$$K_t^{(j)} = \sum_{i \in S_j} (y_i / q) \quad j = 1, 2, \dots, N_t \quad (20)$$

式中:  $y_i \in \{-1, 1\}$  是训练数据集的类标签,  $S_j$  表示欧几里德距离最近包含了  $q$  个样本的一个集合, 很明显  $K_t^{(j)} \leq 1$ , 误差纠偏有三个情况:

- 1) 假如  $K_t^{(j)} = 1$  表示为确切的正样本  $y_i = 1$ ;
- 2) 假如  $K_t^{(j)} = -1$  表示为确切的负样本  $y_i = -1$ ;
- 3) 假如  $K_t^{(j)} < 1$ , 训练集合中数据的权重将会由

样本权重调节器  $L_{t+1}(x_i, y_i)$  来给定。

### 2.4 混合 boost 算法的主要步骤

**Step 1** 输入为  $m+n$  个训练样本,其中  $m$  为负样本个数,  $n$  为正样本个数,非平衡采样时要保证  $m$  和  $n$  有较大差异:  $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n), i = 1, 2, \dots, N, y_i = \{-1, 1\}$ , 其中  $-1$  表示非目标训练样本,  $1$  表示目标训练样本;

**Step 2** 正负样本初始化权值  $w_{1l} = \frac{1}{2 \times m}, w_{2l} = \frac{1}{2 \times n}$ , 其中  $m$  表示集中训练的非目标样本的数目<sup>[1-3]</sup>。

**Step 3** 进行循环(while  $t = 1, \dots, T$ ):

(1) 权值归一化  $\bar{w}_i = \frac{w_i}{\sum_{j=1}^n w_j}$ , 其中,  $n$  为采取的总

特征数;

(2) 为每个特征弱分类器  $h_t$ , 计算该当前权值的

错误率  $\varepsilon_j = \sum_{i=1}^n \bar{w}_i \cdot |h_j(x_i) - y_i|$ ;

(3) 使用误差纠偏方法  $K_i^{(j)} = \sum_{i \in S_j} (y_i/q), j = 1,$

$2, \dots, N_i$  其中  $y_i \in \{-1, 1\}$  是训练数据集的类标签,  $S_j$  表示欧几里德距离最近包含了  $q$  个样本的一个集合, 很明显  $K_i^{(j)} \leq 1$ , 误差纠偏有三个情况: ① 假如  $K_i^{(j)} = 1$  表示为确切的正样本  $y_i = 1$ ; ② 假如  $K_i^{(j)} = -1$  表示为确切的负样本  $y_i = -1$ ; ③ 假如  $K_i^{(j)} < 1$ , 训练集合中数据的权重将由样本权重调节器来给定。

(4) 使用下式来更新弱分类器的权重:

$$\begin{cases} L_{t+1} = 1 & \sum_{p=1}^t f_p(x_i) = 0 \quad \omega_{i,t+1} < Q_{t+1} \wedge Margin_i(x_i) < 0 \\ L_{t+1}(x_i, y_i) = \frac{1}{N} \frac{\exp\left\{-\sum_{i=1}^T [y_i \beta_i h_i(x_i) y + \omega_i(i)]\right\}}{\prod_{i=1}^T J'_i} & \text{其他} \end{cases} \quad (21)$$

(5) 生成强分类器为:

$$H(x) = \text{sign}[f_t(x_i)] \quad (22)$$

## 3 实验

### 3.1 实验数据信息

本实验环境是 Window 7 系统、MATLAB R2013A、英特奔腾 CPU 3.00 GHz (2 CPUs)、内存 4 GB。为了验证改进后的算法的效率, 使用来自文献[3]中 KEEL

数据集的 12 个参考数据集做对比性仿真实验, 如表 1 所示。

表 1 KEEL 数据集相关信息

数据集名称	样本个数	特征数量
Phoneme	5 404	5
Phoneme	5 404	5
Appendicitis	106	7
Pima	768	8
Breast	277	9
Tic-tac-toe	958	9
Magic	19 020	10
Heart	270	13
Australian	690	14
German	1 000	20
Twonorm	7 400	20
Wdbc	569	30
Sonar	208	60

### 3.2 平均测试误差分析

样本数量从 106 到 19 020 且特征数量从 5 到 60<sup>[3]</sup>。分别使用 AdaBoost 算法、P\_Boost 算法、S\_Boost 算法、M\_Boost 算法从不同噪声度下成功检测率误差、平均测试误差等方面进行对比实验。

分别对噪声为 0%、5%、10% 的训练样本和测试样本进行对比实验, 威尔科克森标识排名测试<sup>[3]</sup> 显示如表 2、表 3、表 4 所示。

表 2 0% (噪声度) 时平均测试误差

噪声数据 相应算法	0% (噪声度)			
	AB	PB	SB	MB
Phoneme	0.193	0.195	0.179	<b>0.176</b>
Appendicits	0.192	0.185	0.189	<b>0.184</b>
Pima	0.258	0.252	0.230	<b>0.211</b>
Breast	0.299	0.292	0.281	<b>0.272</b>
Tic-tac-toe	0.245	0.168	0.254	<b>0.232</b>
Magic	<b>0.172</b>	0.192	0.182	0.184
Heart	0.207	0.183	<b>0.176</b>	0.179
Australian	0.163	0.143	0.141	<b>0.138</b>
German	0.260	0.254	0.258	<b>0.249</b>
Twonorm	0.032	0.030	0.035	<b>0.027</b>
Wdbc	0.050	0.031	0.028	<b>0.026</b>
Sonar	0.208	0.210	0.195	<b>0.189</b>

表 3 5% (噪声度) 时平均测试误差

噪声数据 相应算法	5% (噪声度)			
	AB	PB	SB	MB
Phoneme	<b>0.173</b>	0.198	0.185	0.182
Appendicits	0.212	<b>0.191</b>	0.199	0.194
Pima	0.270	0.264	0.245	<b>0.241</b>
Breast	0.307	0.320	0.293	<b>0.284</b>
Tic-tac-toe	0.254	<b>0.174</b>	0.262	0.245
Magic	<b>0.179</b>	0.198	0.192	0.197
Heart	0.245	0.193	0.187	<b>0.188</b>
Australian	0.174	0.165	0.153	<b>0.146</b>
German	0.283	0.274	0.272	<b>0.256</b>
Twonorm	0.037	0.038	0.041	<b>0.035</b>
Wdbc	0.059	0.041	<b>0.035</b>	0.038
Sonar	0.315	<b>0.240</b>	0.350	0.390

表 4 10% (噪声度) 时平均测试误差

噪声数据 相应算法	10% (噪声度)			
	AB	PB	SB	MB
Phoneme	0.223	0.32	0.225	<b>0.312</b>
Appendicits	0.254	0.231	0.239	<b>0.254</b>
Pima	0.280	0.304	0.315	<b>0.341</b>
Breast	0.387	0.370	0.353	<b>0.344</b>
Tic-tac-toe	0.285	<b>0.244</b>	0.342	0.285
Magic	<b>0.199</b>	0.258	0.222	0.257
Heart	<b>0.29</b>	0.287	0.257	0.306
Austalian	<b>0.34</b>	0.575	0.553	0.387
German	0.313	0.295	0.299	<b>0.274</b>
Twonorm	0.117	0.091	0.061	<b>0.041 1</b>
Wdbc	0.079	<b>0.061</b>	0.046	0.108
Sonar	0.425	0.360	<b>0.360</b>	0.410

AdaBoost、P\_Boost、S\_Boost、M\_Boost 分别简写为 AB、PB、SB、MB,加粗部分是本次试验误差最小者,本次实验为 200 个弱分类器时的平均检测误差是指每种算法和当前检测的平均值的绝对差。

### 3.3 成功检测率误差分析

成功检测率误差是指每种算法的正检率与真实行人目标的差异,结果如表 5、表 6、表 7 所示。

表 5 0% (噪声度) 时成功检测率误差

噪声数据 相应算法	0% (噪声度)			
	AB	PB	SB	MB
Phoneme	0.810	0.820	0.832	<b>0.845</b>
Appendicits	0.851	0.847	0.812	<b>0.913</b>
Pima	0.834	0.825	0.834	<b>0.874</b>

续表 5

噪声数据 相应算法	0% (噪声度)			
	AB	PB	SB	MB
Breast	0.849	0.902	0.875	<b>0.901</b>
Tic-tac-toe	0.825	0.904	<b>0.932</b>	0.904
Magic	0.903	0.904	0.912	<b>0.915</b>
Heart	<b>0.827</b>	0.814	0.826	0.801
Australian	0.801	0.807	<b>0.851</b>	0.836
German	0.807	0.764	0.752	<b>0.815</b>
Twonorm	0.821	0.831	0.805	<b>0.854</b>
Wdbc	0.850	0.812	<b>0.873</b>	0.854
Sonar	0.832	0.820	<b>0.841</b>	0.794

表 6 5% (噪声度) 时成功检测率误差

噪声数据 相应算法	5% (噪声度)			
	AB	PB	SB	MB
Phoneme	0.830	0.851	0.879	<b>0.898</b>
Appendicits	0.931	0.862	0.825	<b>0.954</b>
Pima	0.856	<b>0.923</b>	0.854	0.914
Breast	0.879	0.922	0.915	<b>0.924</b>
Tic-tac-toe	0.855	0.914	0.954	<b>0.963</b>
Magic	0.923	0.924	<b>0.952</b>	0.945
Heart	0.852	0.834	0.856	<b>0.873</b>
Australian	0.835	0.887	0.931	<b>0.945</b>
German	0.859	0.824	0.842	<b>0.865</b>
Twonorm	0.867	0.880	0.835	<b>0.897</b>
Wdbc	0.870	0.831	0.923	<b>0.915</b>
Sonar	0.894	0.910	0.893	<b>0.924</b>

表 7 10% (噪声度) 时成功检测率误差

噪声数据 相应算法	10% (噪声度)			
	AB	PB	SB	MB
Phoneme	0.620	0.710	0.78	<b>0.802</b>
Appendicits	<b>0.845</b>	0.802	0.762	0.813
Pima	<b>0.800</b>	0.783	0.782	0.793
Breast	0.802	0.873	0.825	<b>0.831</b>
Tic-tac-toe	0.755	<b>0.875</b>	0.862	0.874
Magic	0.821	0.764	<b>0.852</b>	0.815
Heart	0.802	<b>0.804</b>	0.803	0.733
Australian	0.724	<b>0.757</b>	0.721	0.730
German	0.765	0.714	0.722	<b>0.805</b>
Twonorm	0.801	<b>0.812</b>	0.745	0.767
Wdbc	<b>0.810</b>	0.768	0.753	0.754
Sonar	0.751	0.715	<b>0.760</b>	0.734

表中加粗部分是本次试验正检率最大者;由此可见改进后的算法对大多数实验数据集在误差程度增加的时候较其他算法有一定的稳定优势。

### 3.4 检测目标的时间复杂度分析

计算机算法复杂性的衡量标准是该算法完成相应功能时所需要消耗的计算机资源的多少,主要包括时间和空间两个部分,即时间复杂性和空间复杂性两个部分构成<sup>[16]</sup>。改进后的算法在检测目标时,没有增加额外时间开销,在普通的计算机上即可完成实验。在上述数据集检测时没有出现数量级的增加,所以没有增加时间复杂度。

### 3.5 检测目标的空间复杂度分析

空间复杂度是对一个算法在运行过程中临时占用存储空间大小的量度。一个算法在计算机存储器上所占用的存储空间,包括存储算法本身所占的存储空间,算法的输入输出数据所占用的存储空间和算法在运行过程中临时占用的存储空间这三个方面<sup>[16]</sup>。混合 AdaBoost 算法没有添加太多的代码,只是在算法内部进行了修改,因此存储算法本身所占用的存储空间没有增加。输入输出数据所占用的存储空间和其他算法完全一样,都是输入同样的图像集,输出都是检测到的图像目标,所以输入输出数据所占用的存储空间没有增加。运行过程中临时占用的存储空间与其他算法相比较没有增加。总体来说混合 AdaBoost 算法没有增加空间复杂度。

### 3.6 鲁棒性分析

在以上仿真实验中,所有数据在实现行人检测的过程中没有出现不稳定现象,没有出现拟合现象导致检测出其他类似行人的目标,因此改进后的算法具有较好的鲁棒性。

## 4 结 语

结合文献[2-3]最近的重要科研成果,提出混合 Boost 算法,主要贡献在于:第一,在采样的过程中使用非平衡的样本采样,保证了算法的泛化能力。第二,设置了样本权重自适应调节器更新分类器的权重。第三,结合 PBoost 算法引入误差纠偏方法有效地检测到潜在清晰样本,阻止了误分类器来避免过拟合问题。

## 参 考 文 献

- [1] Wen X, Shao L, Xue Y, et al. A rapid learning algorithm for vehicle classification[J]. Information Sciences, 2015, 295: 395-406.
- [2] Wu S, Nagahashi H. Penalized AdaBoost: Improving the Generalization Error of Gentle AdaBoost through a Margin Distribution[J]. IEEE Transactions on Information & Systems, 2015, E98. D(11):1906-1915.
- [3] Miao Q, Cao Y, Xia G, et al. RBoost: Label Noise-Robust Boosting Algorithm Based on a Nonconvex Loss Function and the Numerically Stable Base Learners[J]. IEEE Transactions on Neural Networks & Learning Systems, 2015, 27(11): 2216-2228.
- [4] Li D Z, Wang W, Ismail F. A selective boosting technique for pattern classification[J]. Neurocomputing, 2015, 156(C): 186-192.
- [5] Shen C, Li H, Van d H A. Fully corrective boosting with arbitrary loss and regularization[J]. Neural Networks the Official Journal of the International Neural Network Society, 2013, 48:44-58.
- [6] Chen T. A selective ensemble classification method on microarray data[J]. Journal of Chemical & Pharmaceutical Research, 2014, 6(6): 2860-2866.
- [7] Frenay B, Verleysen M. Classification in the Presence of Label Noise: A Survey[J]. IEEE Transactions on Neural Networks and Learning Systems, 2014, 25(5):845-869.
- [8] Nguyen T T, Sanner S. Algorithms for direct 0-1 loss optimization in binary classification[C]//International Conference on Machine Learning. 2013:1085-1093.
- [9] Cao J, Kwong S, Wang R. A noise-detection based AdaBoost algorithm for mislabeled data[J]. Pattern Recognition, 2012, 45(12):4451-4465.
- [10] Zhai S, Xia T, Tan M, et al. Direct 0-1 loss minimization and margin maximization with boosting[C]//Advances in Neural Information Processing Systems, 2013:872-880.
- [11] Liu L, Shao L, Rockett P. Boosted key-frame selection and correlated pyramidal motion-feature representation for human action recognition[J]. Pattern Recognition, 2013, 46(7): 1810-1818.
- [12] Patel A J, Patel J S. Ensemble systems and incremental learning[C]//International Conference on Intelligent Systems and Signal Processing. IEEE, 2013:365-368.
- [13] Celebi M E, Kingravi H A, Vela P A. A comparative study of efficient initialization methods for the k-means clustering algorithm[J]. Expert Systems with Applications, 2013, 40(1):200-210.
- [14] Zhu X, Ramanan D. Face detection, pose estimation, and landmark localization in the wild[C]//IEEE Conference on Computer Vision and Pattern Recognition. IEEE Computer Society, 2012:2879-2886.
- [15] Du C H, Zhu H, Luo L M, et al. Face detection in video based on AdaBoost algorithm and skin model[J]. The Journal of China Universities of Posts and Telecommunications, 2013, 20(S1):6-9,24.
- [16] 王晓东. 计算机算法设计与分析[M]. 电子工业出版社, 2012.