

AdaBoost 算法研究进展与展望

曹莹¹ 苗启广¹ 刘家辰¹ 高琳¹

摘要 AdaBoost 是最优秀的 Boosting 算法之一, 有着坚实的理论基础, 在实践中得到了很好的推广和应用. 算法能够将比随机猜测略好的弱分类器提升为分类精度高的强分类器, 为学习算法的设计提供了新的思想和新的方法. 本文首先介绍 Boosting 猜想提出以及被证实的过程, 在此基础上, 引出 AdaBoost 算法的起源与最初设计思想; 接着, 介绍 AdaBoost 算法训练误差与泛化误差分析方法, 解释了算法能够提高学习精度的原因; 然后, 分析了 AdaBoost 算法的不同理论分析模型, 以及从这些模型衍生出的变种算法; 之后, 介绍 AdaBoost 算法从二分类到多分类的推广. 同时, 介绍了 AdaBoost 及其变种算法在实际问题中的应用情况. 本文围绕 AdaBoost 及其变种算法来介绍在集成学习中有着重要地位的 Boosting 理论, 探讨 Boosting 理论研究的发展过程以及未来的研究方向, 为相关研究人员提供一些有用的线索. 最后, 对今后研究进行了展望, 对于推导更紧致的泛化误差界、多分类问题中的弱分类器条件、更适合多分类问题的损失函数、更精确的迭代停止条件、提高算法抗噪声能力以及从子分类器的多样性角度优化 AdaBoost 算法等问题值得进一步深入与完善.

关键词 集成学习, Boosting, AdaBoost, 泛化误差, 分类间隔, 多分类

引用格式 曹莹, 苗启广, 刘家辰, 高琳. AdaBoost 算法研究进展与展望. 自动化学报, 2013, 39(6): 745–758

DOI 10.3724/SP.J.1004.2013.00745

Advance and Prospects of AdaBoost Algorithm

CAO Ying¹ MIAO Qi-Guang¹ LIU Jia-Chen¹ GAO Lin¹

Abstract AdaBoost is one of the most excellent Boosting algorithms. It has a solid theoretical basis and has made great success in practical applications. AdaBoost can boost a weak learning algorithm with an accuracy slightly better than random guessing into an arbitrarily accurate strong learning algorithm, bringing about a new method and a new design idea to the design of learning algorithm. This paper first introduces how Boosting, just a conjecture when proposed, was proved right, and how this proof led to the origin of AdaBoost algorithm. Second, training and generalization error of AdaBoost are analyzed to explain why AdaBoost can successfully improve the accuracy of a weak learning algorithm. Third, different theoretical models to analyze AdaBoost are given. Meanwhile, many variants derived from these models are presented. Fourth, extensions of binary-class AdaBoost to multiclass AdaBoost are described. Besides, applications of AdaBoost algorithm are also introduced. Finally, interested directions which need to be further studied are discussed. For Boosting theory, these directions include deducing a tighter generalization error bound and figuring out a more precise weak learning condition in a multiclass problem. For AdaBoost, the stopping conditions, the way to enhance anti-noise capability and how to improve the accuracy by optimizing the diversity of the base classifiers, are good questions to be in-depth researched.

Key words Ensemble learning, Boosting, AdaBoost, generalization error, classification margin, multiclass classification

Citation Cao Ying, Miao Qi-Guang, Liu Jia-Chen, Gao Lin. Advance and prospects of AdaBoost algorithm. *Acta Automatica Sinica*, 2013, 39(6): 745–758

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. 作为一种元算法框架, Boosting 几乎可以应用于所有目前流行的机器学习算法以进一步加强原算法的预测精度, 应用十分广泛, 产生了极大的影响. 而 AdaBoost 正是其中最成功的代表, 被评为数据挖掘十大算法之一^[1]. 在 AdaBoost 提出至今的十几年间, 机器学习领域的诸多知名学者不断投入到算法相关理论的研究中去, 扎实的理论为 AdaBoost 算法的成功应用打下了坚实的基础. AdaBoost 的成功不仅仅在于它是一种有效的学习算法, 还在于: 1)

收稿日期 2012-07-09 录用日期 2012-09-29
Manuscript received July 9, 2012; accepted September 29, 2012
国家自然科学基金 (61072109, 61272280, 41271447, 61272195),
教育部新世纪优秀人才支持计划 (NCET-12-0919), 中央高校基本科研
业务费专项资金 (K5051203020, K5051203001, K5051303018) 资助
Supported by National Natural Science Foundation of China
(61072109, 61272280, 41271447, 61272195), the Program for
New Century Excellent Talents in University (NCET-12-0919),
and the Fundamental Research Funds for the Central Universi-
ties (K5051203020, K5051203001, K5051303018)
本文责任编辑 张学工
Recommended by Associate Editor ZHANG Xue-Gong
1. 西安电子科技大学计算机学院 西安 710071
1. School of Computer Science and Technology, Xidian Univer-
sity, Xi'an 710071

它让 Boosting 从最初的猜想变成一种真正具有实用价值的算法; 2) 算法采用的一些技巧, 如: 打破原有样本分布, 也为其他统计学习算法的设计带来了重要的启示; 3) 相关理论研究成果极大地促进了集成学习的发展。

本文以 AdaBoost 及其变种算法为线索, 介绍 Boosting 理论的起源、发展, 并探讨一些可能研究的方向: 1) 介绍 Boosting 猜想被证实的过程, 从中引出 AdaBoost 算法的提出以及算法最初的设计思想, 这为后文介绍最新研究方向的变化与优点进行铺垫; 2) 对 AdaBoost 训练误差与泛化误差进行分析, 解释算法能够提高学习精度的原因; 3) 介绍随着研究的发展而出现的几种重要 AdaBoost 理论分析模型, 以及从这些模型衍生出的各种变种算法, 新的分析理论与最初分析理论形成对比, 揭示了 AdaBoost 算法研究的新思路以及尚未解决的一些问题; 4) 介绍 AdaBoost 从二分类到多分类的推广; 5) 介绍 AdaBoost 在实际问题中的应用, 并探讨 AdaBoost 算法及 Boosting 理论未来的一些研究方向。

1 AdaBoost 算法概述

1.1 Boosting 问题起源

在可能近似正确 (Probably approximately correct, PAC) 学习框架^[2]中, X 是样本空间, 概念类 C 是学习器要学习的目标概念的集合, C 中的每个概念 c 对应于 X 的某个子集或一个等效的布尔函数 $c: X \rightarrow \{0, 1\}$. $\mathcal{D}$ 是定义于样本空间 X 上固定但未知的分布, 训练实例集合 S 依分布 $\mathcal{D}$ 从 X 中独立随机抽取. 为了考察学习器的输出假设 h 对真实目标概念类 C 中概念 c 的逼近程度, 首先定义 h 对应于目标概念 c 和样本分布 $\mathcal{D}$ 的真实错误率 $error_{\mathcal{D}}(h) = \Pr_{x \in \mathcal{D}}[c(x) \neq h(x)]$. PAC 学习模型弱化对学习器的要求, 不要求学习器输出一个零错误率假设, 只要求错误率被限制在一个很小的常数 ε 范围内; 不要求学习器对任何随机抽取的训练样本都能成功, 只要求失败的概率在一个很小的常数 δ 范围内。

考虑定义在长度为 n 的训练实例集合 S 上的一个概念类 C , 对 $\forall c \in C$, S 上的任意样本分布 $\mathcal{D}$, $0 < \varepsilon < 1/2$, $0 < \delta < 1/2$, 如果存在 $1/\varepsilon$, $1/\delta$, n 和 $size(c)$ 的多项式函数复杂性的学习算法 A , 能够以 $1 - \delta$ 的概率输出假设 h , 且真实错误率 $error_{\mathcal{D}}(h) \leq \varepsilon$, 则称目标概念类 C 是 PAC 强可学习的, 称算法 A 是一个强学习算法. 如果只需存在某对 ε 和 δ 使以上结论成立, 则称目标概念类 C 是 PAC 弱可学习的, 称算法 A 是一个弱学习算法。

1988 年, Kearns 等在研究 PAC 学习模型时提出了一个有趣的问题: “弱可学习是否等价于强可学习”^[3], 即 Boosting 问题. 如果这一问题有肯定的回答, 意味着只要找到比随机猜测略好的弱学习算法, 就可以将其提升为强学习算法, 而不必直接去寻找通常情况下很难获得的强学习算法, 这对学习算法的设计有着重要的意义。

Schapire 通过构造性方法最终证明: 一个概念是弱可学习的, 当且仅当它是强可学习的, 对 Boosting 问题作出了肯定的回答^[4]. 算法 1 便是 Schapire 设计的第一个 Boosting 算法, 其过程是: 首先依原样本分布 $\mathcal{D}$ 产生训练样本集的一个子集 EX_1 , 调用学习算法 $WeakLearn$ 训练得到分类器 h_1 , 只要求 h_1 是一个弱分类器; 接着, 构造样本集 EX_2 , 其中的样本一半被 h_1 正确分类, 一半被 h_1 错误分类, 训练得到分类器 h_2 ; 最后, 选择 h_1 和 h_2 分类结果不一致的那些样本构成样本集 EX_3 , 调用算法 $WeakLearn$ 训练得到分类器 h_3 . 对新来样本分类的结果, 由 h_1 、 h_2 和 h_3 通过投票表决给出。

算法 1. 强学习算法 Learn

输入. 训练样本集 EX , 弱学习算法 $WeakLearn$, $WeakLearn$ 错误率上界 ε , 置信度参数 δ .

- 1) if ($\varepsilon \geq (1/2 - \gamma)$)
 - 2) return $WeakLearn(EX)$
 - 3) $\alpha = g^{-1}(\varepsilon)$
 - 4) $h_1 = Learn(\alpha, \delta/k, EX_1)$
 - 5) $h_2 = Learn(\alpha, \delta/k, EX_2)$
 - 6) $h_3 = Learn(\alpha, \delta/k, EX_3)$
- 输出. $h = \text{sgn}(h_1 + h_2 + h_3)$.

可以证明, 若 h_1 , h_2 和 h_3 在任意分布下的错误率 α 均小于等于 $1/2$, 则 $h = \text{sgn}(h_1 + h_2 + h_3)$ 在任意分布下的错误率 $g(\alpha)$ 小于等于 $3\alpha^2 - 2\alpha^3$, 这时 $g(\alpha) \leq \alpha$ (仅在 $\alpha = 1/2$ 时取等号), 这一结论意味着算法 1 能够得到一个有着更低错误率的集成分类器. 继续有限次地递归调用算法 1 能得到任意低错误率的强分类器. 在上述构造过程中, 子分类器的误差置信度是 $1 - \delta/k$, 最终分类器的置信度为 $(1 - \delta/k)^k > 1 - \delta$. 算法每次递归的性能增益与弱学习算法的最大错误率 ε 成多项式关系, 所以递归层数是多项式复杂性的. 至此, 证明了弱学习算法可以被提升到以 $1 - \delta$ 概率输出错误率小于任意 ε 的假设, 即得到了强学习算法。

弱可学习与强可学习等价这一定理奠定了 Boosting 算法的理论基础, 同时为构造强分类器提供了重要的启示: 打破分类器在已有样本分布上的优势, 重新采样使接下来的分类器聚焦于难分类的样本. 然而, 尽管这一证明过程给出了一个具体的

Boosting 算法,但是由于要求提前预知分类器错误率上限,因此难以在实际应用问题中使用。

1.2 从在线分配问题到 AdaBoost 算法

在更深入的研究中, Freund 等发现: 在线分配问题与 Boosting 问题之间存在着很强的相似性, 引入在线分配算法的设计思想, 有助于设计出更实用的 Boosting 算法。他们将加权投票的相关研究成果^[5]与在线分配问题结合, 并在 Boosting 问题框架下进行对应推广, 得到了著名的 AdaBoost^[6-7] 算法。AdaBoost 算法不再要求预知弱学习算法的任何先验知识, 在实践中获得了极大的成功。

在线分配算法是一种动态资源分配策略, 分配代理 A 使用 N 个子策略进行决策。在 $t = 1, 2, \dots, T$ 的每一步决策中, A 产生一个定义于 N 个策略之上的权值分布 $\mathbf{p}^t$, $p_i^t \geq 0$ 是分配给策略 i 的权值, 并保证 $\sum p_i^t = 1$ 。第 t 轮, 策略 i 受到由环境决定的损失 $\mathbf{l}_i^t$, 若基于分布 $\mathbf{p}^t$ 使用所有策略进行决策, 则受到混合损失 $\sum_{i=1}^N p_i^t l_i^t = \mathbf{p}^t \mathbf{l}^t$ 。 T 轮后, N 个策略受到的累积混合损失为 $L_A = \sum_{t=1}^T \mathbf{p}^t \mathbf{l}^t$ 。若仅使用单一策略决策, 则个策略中的最优决策受到累积损失 $\min_i L_i$, $L_i = \sum_{t=1}^T l_i^t$ 。分配代理的目标是找到一种最优的权值分布 $\mathbf{p}^t$, 使得 $L_A - \min_i L_i$ 最小。

算法 2. Hedge(β) 算法

输入. 参数 $\beta \in [0, 1]$, 迭代次数 $\sum_{i=1}^N \omega^1 = 1$ 。
初始化: 权值向量 $\omega^1 = [0, 1]^N$, 迭代次数 T 。

- 1) for $t = 1, \dots, T$
- 2) 产生分布 $\mathbf{p}^t = \omega^t / \sum_{i=1}^N \omega_i^t$
- 3) 计算损失向量 $\mathbf{l}^t \in [0, 1]^N$
- 4) 计算混合损失 $\mathbf{p}^t \mathbf{l}^t$
- 5) 更新 $\omega_i^{t+1} = \omega_i^t \cdot \beta^{l_i^t}$
- 6) end

输出. 分布 $\mathbf{p}^t$ 。

为更好地解决在线分配问题, Freund 等设计出了 Hedge(β) 算法^[6] (算法 2)。算法通过综合考虑每一个子策略在历史决策中的累积表现来调整分配给每个策略的权值, 但算法关键参数 β 依赖于问题的先验知识, 需要提前设定。

AdaBoost.M1 (算法 3) 是对 Hedge(β) 的直接对应推广, 仅从表面形式上比较这两个算法, 容易发现它们之间有着惊人的相似性, 进一步验证了在线分配与 Boosting 问题之间存在的深层联系。

从 Hedge(β) 到 AdaBoost, 一种最直接的对应关系是: 在线分配问题中的策略对应 AdaBoost 算法中的子分类器, 分配给每一个策略的权值对应子分类器的投票权值。然而, AdaBoost 使用的却是另一种更为巧妙的对应关系, 如表 1 所示。与直接对应相比, 这种对应方法能够得到更小的训练错误率上

界, 避免了必须提前预知子分类器分类精度相关的先验知识的问题^[6]。

算法 3. AdaBoost.M1 算法

输入. 训练样本集 $S\langle(\mathbf{x}_1, y_1) \cdots (\mathbf{x}_m, y_m)\rangle$, 迭代次数 T , 弱分类器 *WeakLearn*。

初始化: 样本权值 $\omega_i^1 = D(i)$, $i = 1, \dots, m$ 。

- 1) for $t = 1, \dots, T$
 - 2) 产生分布 $\mathbf{p}^t = \omega^t / \sum_{i=1}^m \omega_i^t$
 - 3) 基于分布 $\mathbf{p}^t$ 调用 *WeakLearn*, 得到假设: $h_t: X \rightarrow [0, 1]$
 - 4) 计算假设 h_t 的错误率: $\varepsilon_t = \sum_{i=1}^m p_i^t |h_t(\mathbf{x}_i) - y_i|$
 - 5) 计算 $\beta_t = \varepsilon_t / (1 - \varepsilon_t)$
 - 6) 更新 $\omega_i^{t+1} = \omega_i^t \times \beta_t^{1 - |h_t(\mathbf{x}_i) - y_i|}$
 - 7) end
- 输出.

$$H_f(\mathbf{x}) = \begin{cases} 1, & \text{若 } \sum_{t=1}^T \left(\log \frac{1}{\beta_t} \right) h_t(\mathbf{x}) \geq \\ & \frac{1}{2} \sum_{t=1}^T \left(\log \frac{1}{\beta_t} \right) \\ 0, & \text{否则} \end{cases}$$

表 1 在线分配模型与 AdaBoost 算法各要素之间的对应关系

Table 1 Correspondences of elements between online allocation and AdaBoost algorithm

在线分配问题	AdaBoost 算法
策略	训练样本
使用策略 i 决策受到的损失	第 t 轮迭代中, 样本被当前迭代生成的子分类器 h_i 分类, 若样本正确分类, 损失为 0, 反之, 损失为 1
第 t 轮迭代的混合损失	第 t 轮迭代完成后得到集成分类器 $H_t = \sum_{i=1}^t \alpha_i h_i$. 使用 H_t 对训练样本分类的加权错误率
分配给策略 i 的权值 p_i^t	训练样本权 ω_i^t
每一轮决策	调用弱学习算法生成子分类器

对在线分配算法的对应推广是 AdaBoost 算法最初的设计思想, 一系列的算法理论分析也都建立在在线分配和加权投票研究的结果之上。算法的核心思想是维护一个定义于训练样本集上的样本权值分布, 每一轮迭代加重错分样本的权值, 迫使下一个子分类器聚焦于当前难以被分类样本, 同时使用加权投票策略, 为那些分类准确率高的子分类器赋予更重的投票权值, 同时解决了子分类器生成和集成两大问题。

2 AdaBoost 算法的误差分析

2.1 训练误差分析

在 AdaBoost 提出之初, 研究关注对算法训练误差的分析. Freund 等在文献 [6] 中首次证明了: 若 AdaBoost 每一轮迭代中生成的子分类器错误率分别为: $\varepsilon_1, \varepsilon_2, \dots, \varepsilon_T$, 则集成分类器 H 的训练错误率 ε 有上界: $\varepsilon \leq 2^T \prod_{t=1}^T \sqrt{\varepsilon_t(1-\varepsilon_t)}$.

证明方法是将加权投票研究中已有结论和证明方法^[5]加以推广, 过程具有一定的技巧性和独特性, 不完全适合用于在 Boosting 框架下设计新的算法, 因此, 在后续研究中很少再被采用. 随后, Schapire 等又给出了另一种更简洁地推导训练误差界的方法^[8], 得出:

$$\varepsilon = \frac{1}{N} \sum_i [H(\mathbf{x}_i) \neq y_i] \leq \frac{1}{m} \sum_i \exp(-y_i f(\mathbf{x}_i)) = \prod_{t=1}^T Z_t \quad (1)$$

其中, Z_t 是第 t 轮迭代的样本权归一化因子. 尽管式 (1) 给出的训练误差界更松弛, 但指导算法优化有着更强的可操作性, 只要选择子分类器 h_t 以及 h_t 的加权投票权值 α_t 使 $\prod_{t=1}^T Z_t$ 最小化, 便可得到对训练样本集拟合精度高的强分类器. 然而, 要实现 $\min \prod_{t=1}^T Z_t$ 的精确全局最小化, 每加入一个新的子分类器, 可能都需要修改已有子分类器的集成方式, 复杂度较高. 为了避免解这个复杂的优化问题, AdaBoost 使用贪心策略, 不修改已有分类器形式, 以线性加和方式加入新的子分类器, 只最小化当前迭代的样本权归一化因子 Z_t , 使 $\prod_{t=1}^T Z_t$ 尽可能小. 对训练误差的进一步分析可得到推论 1^[9].

推论 1. 令 $\gamma_t = 1/2 - \varepsilon_t$, 如果存在 $\gamma > 0$, 对所有 t , 都有 $\gamma_t \geq \gamma$ 成立, 则:

$$\frac{1}{N} \sum_{i=1}^N [H(\mathbf{x}_i) \neq y_i] \leq \exp(-2T\gamma^2)$$

推论 1 表明 AdaBoost 的训练误差是以指数速率下降, 并且不需要知道 γ 的下界, 这正是 AdaBoost 设计时所考虑的. 与早期一些 Boosting 算法不同, AdaBoost 具有适应性, 即它能适应弱分类器各自的训练误差, 这也是 Adaptive Boost (AdaBoost) 名称的由来.

2.2 AdaBoost 算法泛化误差分析的发展

在 AdaBoost 提出之初, 作者曾对算法的泛化能力进行过初步分析, 得到泛化误差估计: $\Pr_S[H(\mathbf{x}) \neq y] + \tilde{O}(\sqrt{Td/m})$, 指出算法的泛化错误率与迭代次数和子分类器的复杂度有关, 为了使最终得到

的集成分类器具有更好的泛化性能, 应该按照所掌握的先验知识尽可能选择形式简单的子分类器, 同时适当限制迭代次数 T , 否则当 T 很大时有可能出现过拟合^[6]. 然而, 许多实验都表明 AdaBoost 在迭代次数很高时“似乎并不出现”过拟合, 特别是在训练错误率降低到 0 以后, AdaBoost 的测试错误率仍在降低^[10-11]. 图 1^[12] 是调用 Bagging^[13] 和 AdaBoost 对 UCI 数据集 Letter 测试时的训练错误率和测试错误率变化图, 从中可以看出, 对于 Bagging 算法, 当训练错误率不再变化后, 测试错误率也停止降低; 而对于 AdaBoost 算法, 在最初几轮训练错误率已降至 0 后的很长一段时间内, 测试错误率仍持续降低, 这正是 AdaBoost 的迷人之处. 从表面上看, 随着更多子分类器的加入, 集成分类器形式趋于复杂, 分类精度却仍在提高, 似乎违背了“奥卡姆剃刀”原理^[14]. 为了解释这一现象, 回答“AdaBoost 算法的泛化能力从何而来”, AdaBoost 作者 Schapire 将统计学习中分类间隔分析的相关理论引入 AdaBoost 算法分析, 这也成为了目前最流行的 AdaBoost 分析方法.

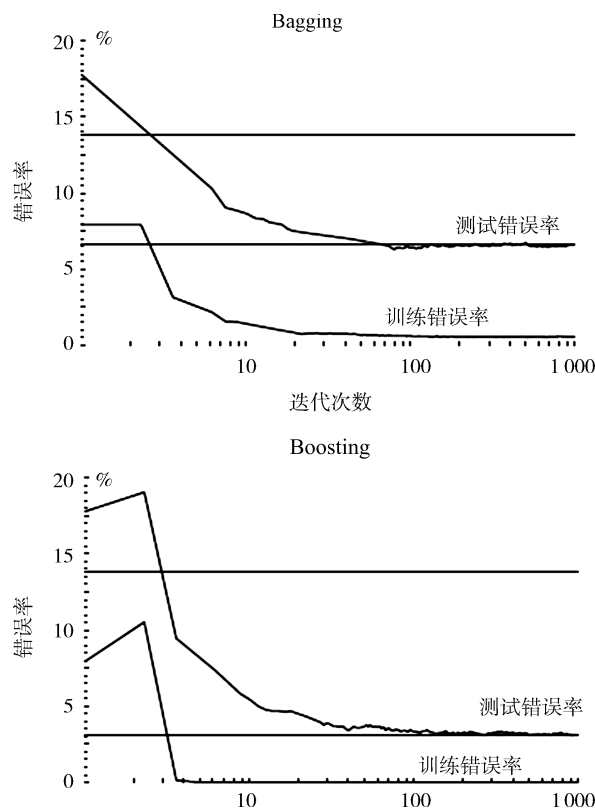


图 1 AdaBoost 算法对 UCI 数据集 Letter 测试的训练错误率与测试错误率

Fig.1 Training error and testing error of AdaBoost on UCI dataset Letter

定义训练样本 $(\mathbf{x}, y)$ 对应于集成分类器

$\sum_i \alpha_i h_i(\mathbf{x})$ 的分类间隔为: $\text{margin}_f(\mathbf{x}, y) = y \sum_t \alpha_t h_t(\mathbf{x}) / \sum_t |\alpha_t| \in [-1, +1]$. 分类间隔的符号为正表示正确分类, 反之则为错误分类, 其取值反映了分类结果的可信程度.

定理 1. 设 $\mathcal{D}$ 是定义在 $X \times \{-1, 1\}$ 上的分布, 训练样本集 S 依分布 $\mathcal{D}$ 独立随机抽取. 若子分类器假设空间 $\mathcal{H}$ 有限, $C = \{f: \mathbf{x} \rightarrow \sum_{h \in \mathcal{H}} \alpha_h h(\mathbf{x}) | \alpha_h \geq 0, \sum_h \alpha_h = 1\}$, 对 $\theta > 0, \delta > 0$, 每一个 $f \in C$, 以 $1 - \delta$ 的概率满足如下错误率限界:

$$P_D[\text{margin}_f(\mathbf{x}, y) \leq 0] \leq P_S[\text{margin}_f(\mathbf{x}, y) \leq \theta] + O\left(\frac{1}{\sqrt{m}} \left(\frac{\log m \log |\mathcal{H}|}{\theta^2} + \log \frac{1}{\delta} \right)^{\frac{1}{2}}\right)$$

定理 1 给出了假设空间 $\mathcal{H}$ 有限时 AdaBoost 的泛化误差界, 指出算法的泛化能力不依赖于集成分类器系统所含有子分类器的个数 T , 只依赖于训练样本集的分类间隔分布 $P_S[\text{margin}_f(\mathbf{x}, y) \leq \theta]$ (Margin distribution)、训练样本数目 m 以及子分类器假设空间的大小 $|\mathcal{H}|$ (通常以 VC 维度量). 文献 [12] 同时证明了 AdaBoost 以贪心方式增大难分类样本的分类间隔, 图 2^[15] 是调用 AdaBoost 算法对 UCI 数据集 Letter 进行测试, 迭代 5 轮 (短虚线)、10 轮 (长虚线) 以及 1000 轮 (实线) 后的累计分类间隔分布, 直观地反映了算法对分类间隔的影响, 与图 1 结合可看出当子分类器数目逐渐增多, 训练错误率已降低为 0 时, 算法依然对分类间隔分布进行提升, 使训练样本的分类间隔分布向大分类间隔样本数目增多的方向发展.

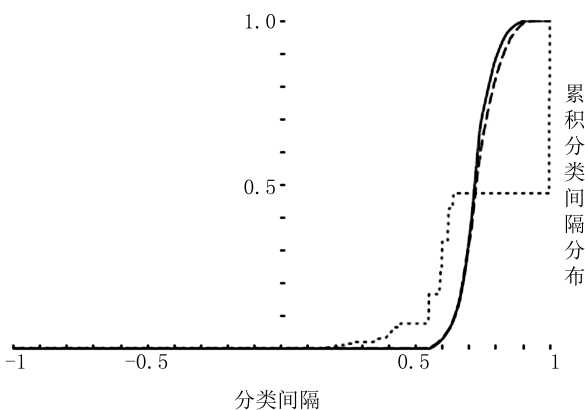


图 2 AdaBoost 算法对 UCI 数据集 Letter 测试 5、10、1000 轮迭代后的累积分类间隔分布图

Fig. 2 Accumulative distribution of margin for AdaBoost on UCI dataset Letter after 5, 10, and 1000 iterations

然而, 实际上定理 1 给出的泛化误差界非常宽松, 并且不具有渐近性, 只有当训练样本集中的样本数目很大 (千至万以上) 时才有意义^[12], 实际问题通

常不能满足此条件, 当样本数目较少时, 这一误差界并不准确. 因此, AdaBoost 的实际运行效果往往优于理论分析结果. 文献 [12] 也讨论了假设空间无限时的泛化误差界, 这依然是一个非常松弛的估计. 可以说, 推导更紧致的泛化误差界以指导算法的设计与优化, 至今依然是一个具有挑战性的问题.

在以上分析提出之后很短的时间内, Breiman 给出了另一种基于最小分类间隔 (Minimum margin) 的分析方法, 得到了比定理 1 更紧致的泛化误差界, 并针对理论分析结果, 设计出了一种直接优化最小分类间隔的 Boosting 算法: Arc-gv^[16]. 与 AdaBoost 相比, Arc-gv 能够得到更大的最小分类间隔以及更好的分类间隔分布, 按照理论分析, 应该具有更强的泛化能力, 然而实验却显示 Arc-gv 的测试错误率总是高于 AdaBoost, 因此, Breiman 对 Schapire 的分析方法提出了质疑, 认为分类间隔分布不是决定 AdaBoost 泛化能力的关键因素, 使得 AdaBoost 的分类间隔分析方法都受到了极大的挑战.

随后, 许多学者独立地对 AdaBoost 算法的泛化能力进行了研究, 给出了一些更紧致的泛化误差界公式^[17-20], 这些研究都表明是分类间隔分布, 而不是最小分类间隔影响着 AdaBoost 的泛化误差, 但都没有正面回答 Breiman 的质疑, 没有解释 Arc-gv 比 AdaBoost 有着更好分类间隔分布的事实. 2006 年, Schapire 部分修正了定理 1 的结论, 对 Breiman 的质疑做出了直接回应, 指出 AdaBoost 的泛化误差还依赖于子分类器的复杂度, 因此, 研究分类间隔对泛化误差的影响必须固定子分类器复杂度. Breiman 在实验中以 CART 为子分类器, 通过固定叶节点数目来控制子分类器复杂度, 比较发现, 当叶节点数目相同时, Arc-gv 生成的树总是比 AdaBoost 生成的树更深, 复杂度更高, 在这种情况下比较 Arc-gv 与 AdaBoost 是不公平的^[21]. 当采用 Decision Stump 作为子分类器 (复杂度完全固定) 后, Arc-gv 不再产生更好的分类间隔分布, 但依然有着更大的最小分类间隔, 这一修正结论最终依然没有完全解决 Breiman “最小分类间隔影响泛化能力” 的分析.

为了解决以上争论, 许多学者一直致力于寻找基于分类间隔分布的更紧致的泛化误差界, 这个泛化误差界应该比 Breiman 给出的基于最小分类间隔的泛化误差界更紧致, 这样才能够完整地证明尽管 Arc-gv 有着更大的最小分类间隔, 但却不足以证明算法会有更好的泛化能力, 因为是分类间隔分布, 而不是最小分类间隔决定了 AdaBoost 算法的泛化能力. 最终, 文献 [22] 构造了一种与分类间隔分布有关, 与最小分类间隔几乎无关的平衡分类间

隔 (Equilibrium margin, Emargin), 给出了基于平衡分类间隔更紧致的泛化误差界, 并通过实验证明: 与 Arc-gv 相比, AdaBoost 有着更大的平衡分类间隔以及更低的测试错误率, 实现了实验测试与理论分析结果的一致. 但设计一种直接优化平衡分类间隔的算法目前依然是一个开放性问题^[23], 如果可以证实与 AdaBoost 相比, 直接优化平衡分类间隔的算法有着更强的泛化能力, 将会更好地证实理论分析的正确性和价值.

很长时期内, AdaBoost 泛化误差研究的核心是解释算法的泛化能力从何而来, 并不能直接指导算法的设计与优化. 现有研究已证明 AdaBoost 是一种间接优化分类间隔分布的算法, 而间接优化已经使算法具有如此优良的特性, 有理由相信直接优化分类间隔分布会得到具有更强泛化能力的分类器^[24]. 这种直接优化的思想已被越来越多的研究者接受, 设计出了不同的 AdaBoost 变种算法^[19, 25-26].

3 AdaBoost 算法的不同理论分析模型

AdaBoost 算法起源于在线分配与加权投票研究, 但随着研究的发展, 最初用于算法分析的理论模型与方法由于具有很强的技巧性以及独特性, 已很少再被采用, 不同学者提出了多种新的理论模型, 越来越多地将 Boosting 算法设计与优化理论联系起来, 在从不同角度解释 AdaBoost 算法原理和有效性的同时, 也为新算法的设计提供了更广阔的思路, 使得 “Boosting” 已经变成一种有效提高分类器分类精度的通用技术, 只要在相应的理论框架下根据问题的实际特点调整相关因素便可设计出不同的变种算法.

3.1 偏差方差分解

在二次回归 (Quadratic regression) 中, 人们发现将多个独立训练的回归方程的输出结果进行平均从不增加期望错误, 平均能够在不影响偏差的情况下降低方差, 从而使期望错误降低. 从贝叶斯决策角度看, 使用多个分类器投票, 相当于对后验概率估计进行平滑和平均. 很自然地联想到, 如果这种特性也存在于分类问题中, 那么投票可以降低期望错误, 这就是 Bagging 思想的由来. 然而, 分类问题的偏差方差有多种不同的定义方式, 并不像回归中那么直接和自然^[27], 若不是每一个 Bootstrap 采样样本集都很好地逼近了真实样本分布, 多个分类器进行投票有时反而会增加分类错误^[28]. 偏差方差分解理论最初认为 Bagging 和 AdaBoost 都是通过减小方差来降低泛化错误率^[11], 随后 AdaBoost 作者对这种观点进行了修正, 指出 AdaBoost 能够同时减小方差和偏差^[12]. Bagging 对错误率的降低主要来自于

减小方差, 适用于如 C4.5、CART 这样低偏差、高方差的分类算法. AdaBoost 能够同时减小方差和偏差, 且方差的减小来自于自适应采样而不是集成方法, 适用于 Decision Stump 这样高方差、偏差稳定的算法, 算法对泛化错误率降低的核心是子分类器错误的无关性^[29], 即子分类器的多样性.

然而, 偏差方差分解理论对 Bagging 和 Boosting 算法的解释并不充分, 实际上这两种算法并不总是减小方差, 存在方差在期望错误率中占主导作用, 但 Bagging 算法依然不能提高分类精度的反例; 也存在 Boosting 算法同时减小方差和偏差, 但泛化错误率没有任何改变的反例. 这说明在使用投票的集成算法中, 除方差偏差外, 还有其他因素影响着泛化错误率 (子分类器以及集成算法的复杂度是其中的核心因素). 同时, 偏差方差分解理论通常只从实验分析角度对 AdaBoost 算法的性质进行定性解释^[29], 很难建立起偏差和方差与泛化错误率之间的有效量化关系, 在指导算法设计和算法优化上作用有限, 因此随着研究的发展这种分析方法很少再有所发展.

3.2 函数空间梯度下降模型

回归和分类都可以形式化地看作是函数估计问题: 设 $\widehat{F(\mathbf{x})}$ 是输入特征空间到输出空间的一个待估计映射, 给定一组训练样本 $\{y_i, \mathbf{x}_i\}_1^N$, 求 $\widehat{F(\mathbf{x})}$ 的一个近似估计 $F^*(\mathbf{x})$ 使训练样本集在某个特定损失函数下的期望损失最小.

经典的参数估计方法假定函数形式已知, 基于训练样本集通过最大似然估计等方法求解模型参数. 参数估计方法简洁稳定, 但是模型选择是否能够很好地近似真实情况往往直接决定了预测结果的好坏. 与参数方法不同, 非参数方法以数据为驱动, 不对模型形式进行任何假定, 将样本点 $\mathbf{x}_i$ 在估计函数 $F^*(\mathbf{x}_i)$ 处的取值看作参数, 在函数空间进行数值优化. 研究证明, AdaBoost 可看作是在函数空间优化分类间隔的指数损失函数 $L(F) = E(e^{-yF(\mathbf{x})})$ 的算法, 通过最大化训练样本的分类间隔来提高分类器的泛化能力^[30].

每一轮迭代, 算法在已有分类器 $F(\mathbf{x})$ 的基础上, 求一个新的估计 $F(\mathbf{x}) + \alpha_t f_t(\mathbf{x})$ 使得损失函数在给定训练样本集上下降最快, 即解优化问题: $\min \text{Loss}(F(\mathbf{x}) + \alpha_t f_t(\mathbf{x}))$. 对每个固定的 α_t 和 $f_t(\mathbf{x})$, 将损失函数在 $f_t(\mathbf{x}) = 0$ 处二阶展开, 得到:

$$\begin{aligned} E[e^{-y(F+\alpha_t f_t)}] &\approx \\ E[e^{-yF(\mathbf{x})}(1 - y\alpha_t f_t(\mathbf{x}) + \frac{1}{2}\alpha_t^2 y^2 f_t(\mathbf{x})^2)|\mathbf{x}] &= \\ E[e^{-yF(\mathbf{x})}(1 - y\alpha_t f_t(\mathbf{x}) + \frac{1}{2}\alpha_t^2)|\mathbf{x}] \end{aligned}$$

对 $f(\mathbf{x}) \in \{-1, +1\}$ 逐点求最小值得:

$$f(\mathbf{x}) = \arg \min_f E_{\varpi} \left(1 - y\alpha_t f_t(\mathbf{x}) + \frac{1}{2}\alpha_t^2 |\mathbf{x}| \right) \quad (2)$$

其中, $E_{\varpi}(\cdot|\mathbf{x})$ 表示加权条件期望, 令 $\omega = \varpi(\mathbf{x}, y) = e^{-yF(\mathbf{x})}$, 并定义:

$$E_{\varpi}[g(\mathbf{x}, y)] = \frac{E[\varpi(\mathbf{x}, y)g(\mathbf{x}, y)|\mathbf{x}]}{E[\varpi(\mathbf{x}, y)|\mathbf{x}]}$$

对 $\alpha_t > 0$, 最小化式 (2) 等价于最小化 $E_{\varpi}[yf_t(\mathbf{x})]$, 其解为

$$f_t(\mathbf{x}) = \begin{cases} 1, & \text{若 } E_{\varpi}(y|\mathbf{x}) = P_{\varpi}(y=1|\mathbf{x}) - \\ & P_{\varpi}(y=0|\mathbf{x}) > 0 \\ -1, & \text{否则} \end{cases}$$

由于 $-E_{\varpi}[yf_t(\mathbf{x})] = \frac{1}{2}E_{\varpi}[y - f_t(\mathbf{x})]^2 - 1$, 于是解优化问题 (2) 可以使用拟牛顿法. 给定 $f_t(\mathbf{x})$, 令损失函数的导数为 0, 可求得:

$$\alpha_t = \arg \min_{\alpha} E_{\varpi} e^{-\alpha_t y f_t(\mathbf{x})} = \frac{1}{2} \log \frac{1 - \varepsilon}{\varepsilon} \varepsilon = (E_{\varpi}[1_{[y \neq f(\mathbf{x})]})]$$

则 $F(\mathbf{x})$ 的迭代规则为

$$F(\mathbf{x}) \leftarrow F(\mathbf{x}) + \frac{1}{2} \log \frac{1 - \varepsilon}{\varepsilon} f_t(\mathbf{x})$$

权值更新策略为

$$\varpi(\mathbf{x}, y) \leftarrow \varpi(\mathbf{x}, y) \exp(\log(\frac{1 - \varepsilon}{\varepsilon}) 1_{[y \neq f(\mathbf{x})]})$$

使用贪心策略以迭代方式在函数空间优化特定损失函数的分析理论不但有效地解释了 AdaBoost 算法的原理和泛化能力, 同时也建立起一类 AdaBoost 变种算法设计的通用框架^[31], 如 AnyBoost 算法优化任意可导损失函数^[32], MarginBoost 算法优化基于分类间隔的损失函数^[32], 文献 [33] 正是在 MarginBoost 算法框架下设计出的非对称代价 Boosting 算法, 用于垃圾邮件检测. 文献 [34] 的研究结果给出了一组通用的约束条件对损失函数加以约束, 保证了只要按照标准优化步骤优化对满足约束条件的损失函数进行优化, 得到的算法一定具有 Boosting 性质, 这使得 “Boosting” 成为一种简单易行的算法设计技术.

3.3 前向可加分步模型

分类与回归是机器学习研究的两大基本问题, AdaBoost 最初为解决分类问题设计. Schapire 也曾尝试将回归转化为特定形式的分类, 再按照解分类问题的标准做法将 Boosting 技术应用于回归分析,

得到了 AdaBoost.R 算法^[6], 然而这种推广是一种曲折迂回的方法^[30]. 为在回归问题中直接使用 Boosting 技术, Friedman 等从统计学角度, 在非参数回归基础上, 用前向可加分步模型解释了 AdaBoost 算法^[30], 尽管这种解释受到了一些质疑, 认为不能有效解释 AdaBoost 的泛化能力, 但在此模型框架下衍生出了多种重要且具有影响力的 AdaBoost 变种算法, 包括: LogitBoost^[30], GradientBoost^[35], Gradient-boost-tree^[36], 以及用于解决概率回归的 ProbitBoost^[37].

$f(\mathbf{x}) = \sum_{t=1}^T \beta_t b(\mathbf{x}; \gamma_t)$ 是一个泛化可加模型, 其中, $b(\mathbf{x}; \gamma_t)$ 是基函数, β_t 是基函数的系数, γ_t 是基函数的参数. AdaBoost 求一组分类器的加权组合, 是一个典型的加法模型. 在给定训练数据和损失函数 $L(y, f(\mathbf{x}))$ 的情况下, 学习加法模型 $f(\mathbf{x})$ 就是求解经验风险最小化式 (3) 的优化问题:

$$\min_{\beta_t, \gamma_t} \sum_{i=1}^N L \left(y_i, \sum_{t=1}^T \beta_t b(\mathbf{x}_i; \gamma_t) \right) \quad (3)$$

通常情况下, 求解式 (3) 是一个复杂的优化问题. 而前向分步可加算法的思想是不修改已生成分类器的形式, 在上一个分类器残差减小的梯度方向建立下一个分类器, 使损失函数下降最快以逐步逼近最优解, 降低优化问题的复杂度.

算法 4. 求解前向分步加法模型的通用算法框架

输入. 训练样本集 $S\langle(\mathbf{x}_1, y_1) \cdots (\mathbf{x}_m, y_m)\rangle$, 损失函数 $L(y, f(\mathbf{x}))$, 基函数集合 $\{b(\mathbf{x}; \gamma)\}$.

初始化: $f_0(\mathbf{x}) = 0$.

1) for $t = 1, \dots, T$

2) $(\beta_m, \gamma_m) = \arg \min_{\beta, \gamma} \sum_{i=1}^N L(y_i, f_{t-1}(\mathbf{x}_i) + \beta b(\mathbf{x}_i; \gamma))$

3) 更新 $f_m(\mathbf{x}) = f_{m-1}(\mathbf{x}) + \beta_t b(\mathbf{x}_i; \gamma_t)$

4) end

输出. $f(\mathbf{x}) = f_T(\mathbf{x}) = \sum_{t=1}^T \beta_t b(\mathbf{x}_i; \gamma_t)$.

算法 4 是求解前向分步加法模型的通用算法框架, 可以证明 AdaBoost 是它的一个具体实现: 基函数是子分类器, 损失函数是分类间隔的指数函数^[30]. 求解算法 4 第 2) 步的优化问题时, 使用不同的准则函数便得到不同的变种算法. 如, 对于回归问题, 可得到使用最小平方误差的 LS_Boost, 使用绝对损失的 LAD_Boost 和 LAD_TreeBoost, 使用 Huber-M 损失函数的 M_TreeBoost^[30] 等. 对于分类问题, 可得到使用二项对数损失的 L_k _TreeBoost 和 LogitBoost^[38].

泛化可加模型将 Boosting 思想与统计学中的最大似然估计联系在一起, 引出了一种重要的数学模型: 梯度下降 Boosting. 该模型下推导出的算法

多用于解决数值问题,或是将分类转化为逻辑斯谛回归,算法中不再出现样本权分布,即使出现也不用于重采样(如 LogitBoost).

3.4 AdaBoost 算法的其他分析方法

除去以上几种有着重要影响力的理论模型,AdaBoost 发展至今,研究者们将算法与各种理论联系起来,衍生出了不同的分析方法和相应的变种算法.

1) 与熵投影相结合. 熵投影与 AdaBoost 算法的结合从信息论角度很好地解释了样本权更新策略在算法中的作用,为样本权更新策略的设计提供有效的设计原则,并提出了“完全正确更新”的概念^[39]. 式(1)指出 AdaBoost 的训练误差界为 $\prod_{t=1}^T Z_t$, 算法以贪心策略最小化每一轮迭代的 Z_t , 以实现提高训练精度的目的,算法的设计可看作是解无约束优化问题 $\max_{\alpha \in R} (-\ln Z_t(\alpha))$. 在信息论中,相对熵用于度量两个分布 $\mathbf{d}$ 和 $\tilde{\mathbf{d}}$ 之间的距离,其定义为: $\Delta(\mathbf{d}, \tilde{\mathbf{d}}) = \sum_{i=1}^m (d_i \ln d_i / \tilde{d}_i)$. 可以证明,AdaBoost 的样本权值调整策略与式(4)的相对熵最小化问题是对偶问题,可通过标准线性规划法求解^[40].

$$d_{t+1} = \arg \min_{\mathbf{d} \in P_m} \Delta(\mathbf{d}, \mathbf{d}_t) \quad \text{s.t.} \quad \mathbf{d} \cdot \mathbf{u}_t = 0 \quad (4)$$

AdaBoost 算法的权值更新策略使本轮生成的子分类器在新的样本权之下,加权错误率恰好等于随机猜测错误率 0.5,即新的样本分布与当前子分类器分类结果向量正交. 若将这种策略进一步深化,选择新的样本分布与到目前为止生成的所有分类器的分类结果向量都正交,就是完全正确更新算法^[41-43]. 理论上,与 AdaBoost 算法使用的正确更新策略相比,完全正确更新能够更快收敛,得到的集成分类器更简单,但算法更复杂,计算量更大,因此,不是任何时候都是必要的^[40].

2) 与线性规划相结合. 在最大化分类间隔理论的指导下,大的分类间隔意味着更好的泛化能力. 在这一思想下,LPBoost 将最大化训练样本的软间隔与线性规划联系起来,用求解大规模优化问题而设计的列生成技术求解算法关键参数^[39, 44-45]. 与梯度下降 Boosting 算法只在迭代次数趋于无穷的极限情况下收敛不同,LPBoost 能够在有限次迭代收敛到全局最优解,实验结果显示 LPBoost 对真实数据的测试效果很好,通常情况下比其他变种算法更快收敛,但存在错误率随迭代次数线性下降的情况. 为了克服这一缺点,只要向 LPBoost 的约束目标中引入相对熵规则化项,得到的熵规则化 ERLPBoost 算法便有着指数误差界^[46],并且与另一种同样优化软分类间隔的 SoftBoost 算法^[47]相比,泛化错误率更低.

3) 与博弈论相结合. AdaBoost 与博弈理论的联系由 Breiman、Freund 和 Schapire 等提出^[16, 48-50], 算法可看作是弱分类器和主算法反复进行零和博弈的过程. 零和博弈用博弈矩阵 M 表示,游戏中 A 依分布 P 随机选择矩阵的一行 i , B 依分布 Q 随机选择矩阵的一列 j , 定义 A 受到的损失等于 B 的收益,为 M_{ij} , A 受到的期望损失为 $P^T M Q$.

对应到分类问题,若假设空间 $\mathcal{H} = \{h_1, \dots, h_n\}$ 含 n 个假设,训练样本集 $\{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_m, y_m)\}$ 含有 m 个样本,博弈矩阵 M 的 m 行 n 列,对应了分类器对训练样本分类的结果;分布 P 对应了训练样本分布 $\mathcal{D}_t$; 分布 Q 对应归一化后的子分类器投票权值. 若第 j 轮迭代从假设空间中选择的分类器 h_i 正确分类第 j 个样本,即 $h_j(\mathbf{x}_i) = y_i$, 则 $M_{i,j} = 1$, 反之 $M_{i,j} = 0$. 可以证明,AdaBoost 迭代最终选择的样本权分布和分类器投票权值是零和博弈最优解(最大最小策略或是最小最大策略)的一个近似.

4) 与 PoE 概率模型相结合. 在最新研究中,研究者使用 PoE 概率模型来解释 AdaBoost 算法. PoE 模型(Product of expert)是一种将多个基于同一组数据建立起的不同概率模型进行合成的方法,合成结果是每一个子模型输出概率的归一化积^[51], 适合用于描述满足多个不同低维约束条件的高维数据. 在 PoE 概率模型与 Boosting 思想的结合中,Boosting 框架下的弱分类器被看作是独立的子概率分量,增强学习过程被看作是增量式的模型更新过程,每一轮迭代新加入的子分类器必须保证不降低训练样本上的似然估计. 这一约束条件等价于 Boosting 框架下的弱分类器条件,会对子分类器的模型参数产生影响,从中可推导出每一轮迭代的样本权分布更新规则.

PoE 概率模型与 AdaBoost 的结合将 Boosting 框架下的各种学习问题假设转化为概率模型中对似然估计函数的约束,提供了一种简单的、以数据为驱动的学习模型建立方法. 通过替换底层所使用的子概率模型,能够将 Boosting 思想无缝地应用于包括分类问题在内的各种不同学习问题,同时也使得在建立学习模型过程中使用各种成熟的统计推断方法变为可能. 在这一分析模型下提出的变种算法有 POEBoost.DS 和 POEBoost.CS^[52].

4 多分类 AdaBoost 算法

AdaBoost 最初设计用于解决二分类,而实际问题以多分类居多. 对二分类,弱分类器条件要求对任意样本分布,分类器的分类准确率只需略高于 0.5,对分类数目为 K 的多分类,这一条件过强,然而,要求比随机猜测准确率 $1/K$ 略高,又是一个过弱的条

件. 精确的弱分类器条件, 能够帮助算法设计者选择更简洁的子分类器以防止过拟合, 防止集成后分类器准确率退化, 以及用于判断是否终止迭代, 是多分类 Boosting 算法设计中的一个重要问题.

若将二分类 AdaBoost 算法直接应用于多分类问题, 可能会由于弱分类器条件过强而失败. 按照解决这一问题的不同处理方法, 多分类 AdaBoost 算法可分为二分类拆解与直接多分类两大类. 拆解法将多分类拆解为多个二分类, 这种拆解中隐式地假定了使用与二分类相同的弱分类器条件, 从而避免求解复杂的多分类下的弱分类器条件. 不同的拆解方法又分为经典的一对一 (One against one)、一对其余 (One against all) 策略, 纠错输出编码 (Error correcting output code, ECOC), 以及层次分解法等.

AdaBoost.M2 是一对一拆解方法中的典型代表, AdaBoost.MH, AdaBoost.MO, AdaBoost.MR^[8] 则采用一对其余拆分思想. 在 ECOC 方法中, 分类问题被看作是一个通信任务, 使用数据通信中的纠错编码对输出类别标签进行变换, 从而将多分类转化而多个二分类. 其中, 多分类到二分类的分解对应了编码过程, 二分类到多分类的合成对应了解码过程, 算法的核心思想是通过编码矩阵的纠错能力来提高分类准确率^[53-54], 代表算法包括 AdaBoost.ECC^[55], AdaBoost.ERP^[56], AdaBoost.SECC^[57], HingeBoost^[58] 等. 然而, ECOC 方法的一个缺点是没有有效利用每一个原始类别的结构信息, 对先验结构信息的利用能够进一步提高算法的性能^[59].

实际上, 无论是一对一、一对其余, 还是 ECOC 拆解法都可以统一到编码矩阵 (Coding matrix) 算法框架下^[60]. 编码矩阵 M 是一个 k 行 l 列的矩阵, 形式如图 3 所示, 每一行对应一个类别, 每一列对应一个子分类器, 矩阵元素 $M(i, j) \in \{-1, 0, +1\}$ 表示训练第 j 个子分类器时第 i 类样本的类别标签编码, 若 $M(y_i, s) = 0$, 则真实类别标签是 y_i 的样本不用作训练.

分类时, 所有子分类器的输出会构成一个输出行向量, 计算该向量与编码矩阵 M 每一个行向量 $M(r)$, $r = 1, \dots, k$ 之间的距离, 最近的那一行对应的类别作为预测标签. 在编码矩阵算法框架下, 可以证明通过二分类拆解得到的多分类 AdaBoost 训练错误率上界为 $E \leq l\varepsilon/\rho L(0)$, $\varepsilon = \sum_{i=1}^m \sum_{s=1}^l L(M(y_i, s) f_s(\mathbf{x}_i))/ml$, 是平均二分类损失, $\rho = \min \{\Delta(M(r_1), M(r_2)) : r_1 \neq r_2\}$ 是编码矩阵任意两个行向量之间的最小距离^[60].

在以上方法之外的层次拆解法按照类别之间的相似性, 将 K 分类问题拆解为以二叉树形式组

织的 $K - 1$ 个二分类问题^[61], 典型代表有 AdaBoost.asBHC^[62]. 层次拆解法可以避免一对其余方法造成的数据集偏斜, 避免一对一方法造成的子分类器数目过多, 但也存在错误累积的问题, 设计时需要仔细构造问题拆解二叉树.

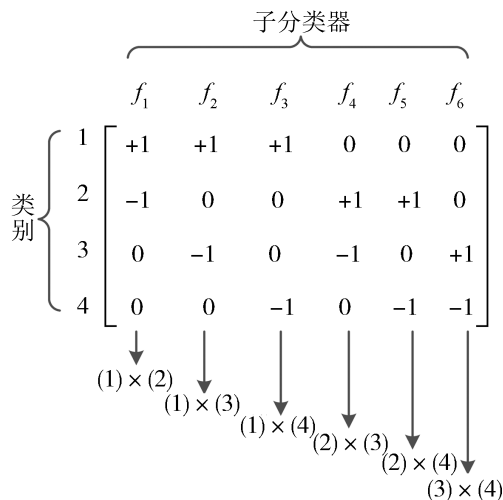


图3 编码矩阵的一个实例

Fig. 3 An example of the coding matrix

在 Boosting 类算法设计中, 将多分类拆解为多个二分类的一个优点是: 直接使用与二分类相同的弱分类器条件, 只要求每个子分类器的分类准确率高于 0.5, 便能保证集成分类器分类的准确率不出现退化, 从而避免了确定弱分类器条件这一难题. 然而, 通过问题拆解隐式地使用二分类下的弱分类器条件可能依然过强 (如 AdaBoost.MH), 因此, 拆解法对多分类 Boosting 问题不一定总是最合适的^[63]. 直接多分类 AdaBoost 算法一方面通过选择如 CART、C4.5 这样直接支持多分类的算法作为子分类器, 对原有二分类 AdaBoost 算法直接推广, 另一方面研究更适合多分类的精确弱分类器条件, 如 AdaBoost.SAMME 算法, 以及文献 [63] 中提出一族最优的多分类下的弱分类器条件, 以及在此条件下统一的多分类算法框架. 但总体上, 目前研究提出的直接多分类算法只占所有多分类 AdaBoost 算法中很少的一部分.

相比二分类, 多分类 Boosting 理论的研究并不完善, 还有许多有待解决的问题, 其中的核心是寻找精确的弱分类器条件, 并在此条件下设计能够更有效减小训练误差、提高分类器泛化能力的直接多分类算法.

5 AdaBoost 算法的应用

AdaBoost 算法具有能够显著改善子分类器预

测精度、不需要先验知识、理论扎实等优点,一经提出就受到不同领域研究人员的关注,在解决各行业应用问题中都获得了极大的成功。

手写字体识别是 AdaBoost 最早也是最为成功的实际应用之一。在早期流行的手写字体识别算法中,研究者使用多层前向神经网络作为弱分类器。与单个神经网络相比,由 AdaBoost 算法增强的神经网络识别能力更强,不仅能够极大地降低识别错误率,还能够识别更多不同书写风格的字体^[64]。随着研究的深入,能够使用的弱分类器也已经扩展到决策树、KNN 以及 SVM 等多种流行的学习算法。相关的应用场景也扩展到手写文档识别与检索^[65]、从印刷文档中分割与识别手写注释^[66]以及手写签名验证^[67]等。

Viola 等提出的基于 AdaBoost 的人脸检测算法^[68],在提高人脸识别准确率的同时,极大地加快了检测速度,一经提出便迅速发展为该领域的主流算法^[69],目前,该算法已被固化到芯片中,应用于全球数以万计的数码相机和手机^[70-71]。同时,各种基于 AdaBoost 及其变种算法的人脸检测以及虹膜检测算法也被扩展应用到移动设备安全验证领域^[72-73],在检测速度和检测率上,与传统的方法相比都有很大的提高。

近年来,随着互联网发展,互联网产生、承载和传递的数据迅速增长,文本分类、文本过滤与信息检索成为炙手可热的研究方向,AdaBoost 算法与 Boosting 技术在这些领域的应用也最为频繁^[74-75]。由 Schapire 和 Singer 开发的 BoosTexter 系统,是第一个基于 Boosting 思想的文本分类系统。BoosTexter 使用单层决策树 Decision Stump 作为弱分类器去识别一个单词或简单词组,再通过对弱分类器的集成和增强最终实现对整个文本意思的理解^[76]。在这之后,Boosting 技术在文本分类中的应用被不同研究者发展与完善,目前已经成为构建文本分类器,提高文本分类准确率的一种重要方法^[77-78]。基于 AdaBoost 的信息检索算法不仅可以用于文本信息检索与分类,同时也广泛的应用于图像、视频与音频等多媒体信息检索的研究中^[79]。

代价敏感学习以及不平衡数据集学习是机器学习研究的重要内容。目标检测、医疗诊断、入侵检测以及经济预测等领域的应用问题常伴随着天生的不对称性。不同分类错误引起的代价差异较大,或是关注的样本难以获得高造成训练和检测样本类分布严重不平衡。研究指出 AdaBoost 有着内在的非对称性,可以直接应用于非对称学习^[80],算法使用的改变样本分布的策略,能够很自然地与经典代价敏感学习方法以及不平衡数据集学习方法相结合,使得 AdaBoost 及其变种算法已发展成为实现非对称学

习的一种重要方法^[81-84]。

除此之外,AdaBoost 及其变种算法还被广泛应用于人体动作识别^[85]、自然语言处理^[86-87];在机器视觉领域中进行目标检测^[88]、车辆识别^[89]、视频文字定位^[90]等;在计算机安全领域进行垃圾邮件分类^[33,91]、钓鱼网站检测^[92]等;在计算生物学中进行 DNA 序列分析^[93]等。C4.5 算法的商业化改进版本 C5.0 也默认使用 AdaBoost 进一步增强决策树的预测精度。这些成功应用无一不证明了 AdaBoost 算法的优秀性能。

6 总结与展望

以 AdaBoost 为代表的增强学习算法在过去十几年间取得了巨大的成功,已经发展为一种理论完善的机器学习技术。本文对 Boosting 问题的提出,AdaBoost 算法的起源与发展进行了总结与分析,以理论研究为主线,介绍了算法的误差分析与不同理论模型,同时探讨了 AdaBoost 的一些应用实例,希望能够为相关研究人员提供一些有用的线索。

AdaBoost 算法的重要意义在于为研究和实际问题的解决带来了新的思想,在绝大多数集成学习算法通关构造越来越复杂的分类器来提高预测精度时,AdaBoost 却追求将最简单的、比随机猜测略好的弱分类器组合得到强分类器。在训练子分类器的方法上,AdaBoost 提供了重要启示:打破已有样本分布,重新采样使分类器更多地关注难学习的样本。在算法使用上,仅需要指定迭代次数,不需要任何先验知识,一切运行过程中的参数由算法自适应地调整,因此被评价为最接近“拿来即用”的算法^[36]。

与此同时,AdaBoost 算法也存在缺点以及一些尚未解决的问题。1) 推导更紧致的泛化误差界以及算法收敛速度的证明^[94]始终是理论研究所追求的,这对算法优化,新算法设计有着重要的意义;2) 与二分类研究相比,多分类 Boosting 理论并不完善,寻找更适合多分类问题的损失函数,以及多分类下的精确弱分类器条件都是值得继续研究的问题;3) AdaBoost 算法的不易过拟合是有条件的,对噪声数据敏感始终是 AdaBoost 算法被人们熟知的缺点,在有噪声情况下算法很快出现过拟合,这与算法选择的指数损失函数有很大的关系。指数损失函数过度地聚焦于始终难以被正确分类的样本,这些样本通常是噪声和奇异值点。基于错误率的 Boosting 算法抗噪声能力更强,但非凸性、导数不连续性等特点决定了 0-1 损失不适合用于数值优化方法的设计,不同研究者提出的不同的优化目标和各种抗噪声损失函数实际上都是对 0-1 损失的逼近^[36],如何设计新的优化方法,对如平滑 0-1 损失函数这样理论上更优的损失函数进行优化,提高算法抗噪声能力,是

进一步研究的重要问题;同时,从另一个角度出发,利用 AdaBoost 算法对噪声敏感的特性设计噪声检测算法也是一个值得探索的研究题目^[95];4) 运行 AdaBoost 算法只需设定迭代次数这个唯一的参数,人们通常依赖 AdaBoost 不易过拟合(相对、有条件的)的特性,为在得到精确的分类模型之前不停止算法,不惜设置很大的迭代次数。但研究已证明,即使在理想情况下,不断增大迭代次数,AdaBoost 最终会出现过拟合^[96],并且迭代后期生成的子分类器对提高分类器泛化性能的作用非常微小,进行过多次迭代不但存在过拟合风险,而且也是计算能力的浪费,因此,停止迭代条件是进一步研究不可避免的问题,已受到许多研究人员的关注^[97-98];5) 多样性是集成学习的基石,尽管分类器多样性研究目前依然是一个没有统一理论和太多有效结论的研究题目,但是研究者普遍认可改善子分类器的多样性能够帮助提高集成学习系统的分类精度和泛化能力这一观点。对 AdaBoost 算法生成的子分类器多样性进行分析发现:在前几轮迭代中,由于每一个新的子分类器都关注于上一个子分类器难以正确分类的样本,因此有着较高的多样性。然而几轮迭代之后,难以被正确分类的样本很可能始终被错分,导致后生成的子分类器越来越关注同一批不能被正确分类的样本,多样性急剧下降。一些相关研究在遵循 AdaBoost 基本设计思想的基础上,增加子分类器多样性,实验结果都表明算法性能得到了改善^[99-100],但有关分类器多样性的研究还有待进一步深入与完善。

References

- 1 Zhou Z H, Yang Y, Wu X D, Kumar V. *The Top Ten Algorithms in Data Mining*. New York, USA: CRC Press, 2009, 127-149
- 2 Valiant L G. A theory of the learnable. *Communications of the ACM*, 1984, **27**(11): 1134-1142
- 3 Kearns M, Valiant L. Cryptographic limitations on learning boolean formulae and finite automata. *Journal of the ACM*, 1994, **41**(1): 67-95
- 4 Schapire R E. The strength of weak learnability. *Machine Learning*, 1990, **5**(2): 197-227
- 5 Littlestone N, Warmuth M K. The weighted majority algorithm. *Information and Computation*, 1994, **108**(2): 212-261
- 6 Freund Y, Schapire R E. A decision-theoretic generalization of on-line learning and an application to Boosting. *Journal of Computer and System Sciences*, 1997, **55**(1): 119-139
- 7 Freund Y, Schapire R E. Experiments with a new Boosting algorithm. In: *Proceedings of the 13th Conference on Machine Learning*. San Francisco, USA: Morgan Kaufmann, 1996. 148-156
- 8 Schapire R E, Singer Y. Improved Boosting algorithms using confidence-rated predictions. *Machine Learning*, 1999, **37**(3): 297-336
- 9 Li Hang. *Methods of Statistical Learning*. Beijing: Tsinghua University Press, 2012. 143
(李航. 统计学习方法. 北京: 清华大学出版社, 2012. 143)
- 10 Quinlan J R. Bagging, Boosting, and C4.5. In: *Proceedings of the 13th National Conference on Artificial Intelligence*. California, USA: AAAI Press, 1996. 725-730
- 11 Breiman L. Arcing classifier (with discussion and a rejoinder by the author). *Annals of Statistics*, 1998, **26**(3): 801-849
- 12 Bartlett P, Freund Y, Lee W S, Schapire R E. Boosting the margin: a new explanation for the effectiveness of voting methods. *Annals of Statistics*, 1998, **26**(5): 1651-1686
- 13 Breiman L. Bagging predictors. *Machine Learning*, 1996, **24**(2): 123-140
- 14 Blumer A, Ehrenfeucht A, Haussler D, Warmuth M K. Occam's razor. *Information Processing Letters*, 1987, **24**(6): 377-380
- 15 Schapire R E. Theoretical views of Boosting. In: *Proceedings of the 4th European Conference on Computational Learning Theory*. Berlin, Germany: Springer-Verlag, 1999. 1-10
- 16 Breiman L. Prediction games and arcing algorithms. *Neural Computation*, 1999, **11**(7): 1493-1517
- 17 Grove A J, Schuurmans D. Boosting in the limit: maximizing the margin of learned ensembles. In: *Proceedings of the 15th National/10th Conference on Artificial Intelligence/Innovative Applications of Artificial Intelligence*. San Francisco, USA: John Wiley & Sons, 1998. 692-699
- 18 Koltchinskii V, Panchenko D. Empirical margin distributions and bounding the generalization error of combined classifiers. *Annals of Statistics*, 2002, **30**(1): 1-50
- 19 Rätsch G, Warmuth M K. Maximizing the margin with Boosting. In: *Proceedings of the 15th Annual Conference on Computational Learning Theory*. Berlin, Germany: Springer-Verlag, 2002. 334-350
- 20 Koltchinskii V, Panchenko D. Complexities of convex combinations and bounding the generalization error in classification. *Annals of Statistics*, 2005, **33**(4): 1455-1496
- 21 Reyzin L, Schapire R E. How Boosting the margin can also boost classifier complexity. In: *Proceedings of the 23rd International Conference on Machine Learning*. New York, USA: ACM, 2006. 753-760
- 22 Wang L W, Sugiyama M, Yang C, Zhou Z H, Feng J F. On the margin explanation of Boosting algorithms. In: *Proceedings of the 21st Annual Conference on Learning Theory*. Madison, USA: Omni Press, 2008. 479-490
- 23 Feng J F, Wang L W, Sugiyama M, Yang C, Zhou Z H, Zhang C C. Boosting and margin theory. *Frontiers of Electrical and Electronic Engineering*, 2012, **7**(1): 127-133
- 24 Mason L, Bartlett P L, Baxter J. Direct optimization of margins improves generalization in combined classifiers. In: *Proceedings of the 1998 Conference on Advances in Neural Information Processing Systems*. Cambridge, MA: The MIT Press, 1999. 288-294
- 25 Mason L, Bartlett P L, Baxter J. Improved generalization through explicit optimization of margins. *Machine Learning*, 2000, **38**(3): 243-255

- 26 Shen C H, Li H X. Boosting through optimization of margin distributions. *IEEE Transactions on Neural Networks*, 2010, **21**(4): 659–666
- 27 Geurts P. Bias vs variance decomposition for regression and classification. *Data Mining and Knowledge Discovery Handbook*. New York, USA: Springer, 2005. 749–763
- 28 Friedman J H. On bias, variance, 0/1-loss, and the curse-of-dimensionality. *Data Mining and Knowledge Discovery*, 1997, **1**(1): 55–77
- 29 Whewey V. Variance reduction trends on “boosted” classifiers. *Journal of Applied Mathematics & Decision Sciences*, 2004, **8**(3): 141–154
- 30 Friedman J, Hastie T, Tibshirani R. Additive logistic regression: a statistical view of Boosting (with discussion and a rejoinder by the authors). *Annals of Statistics*, 2000, **28**(2): 337–407
- 31 Mason L, Baxter J, Bartlett P L, Frean M. Functional gradient techniques for combining hypotheses. *Advances in Large Margin Classifiers*. Cambridge, MA: MIT Press, 2000. 221–247
- 32 Mason L, Baxter J, Bartlett P, Frean M. Boosting algorithms as gradient descent. *Advances in Neural Information Processing Systems*, 1999, **12**: 512–518
- 33 He J R, Thiesson B. Asymmetric gradient Boosting with application to spam filtering. [Online], available: <http://ceas.cc/2007/>, June 7, 2013
- 34 Duffy N, Helmbold D. Potential boosters. *Advances in Neural Information Processing Systems*, 1999, **12**: 258–264
- 35 Friedman J H. Greedy function approximation: a gradient boosting machine. *Annals of Statistics*, 2001, **29**(5): 1189–1232
- 36 Friedman J, Hastie T, Tibshirani R. *The Elements of Statistical Learning*. New York, USA: Springer, 2001. 337–384
- 37 Zheng S F, Liu W X. Functional gradient ascent for probit regression. *Pattern Recognition*, 2012, **45**(12): 4428–4437
- 38 Collins M, Schapire R E, Singer Y. Logistic regression, AdaBoost and Bregman distances. *Machine Learning*, 2002, **48**(1–3): 253–285
- 39 Warmuth M K, Liao J, Rätsch G. Totally corrective Boosting algorithms that maximize the margin. In: Proceedings of the 23rd International Conference on Machine Learning. New York, USA: ACM, 2006. 1001–1008
- 40 Kivinen J, Warmuth M K. Boosting as entropy projection. In: Proceedings of the 12th Annual Conference on Computational Learning Theory. New York, USA: ACM, 1999. 134–144
- 41 Wang P, Shen C H, Barnes N, Zheng H, Ren Z. Asymmetric totally-corrective Boosting for real-time object detection. In: Proceedings of the 10th Asian conference on Computer Vision. Berlin, Heidelberg: Springer-Verlag, 2011. 176–188
- 42 Shen C H, Hao Z H. A direct formulation for totally-corrective multi-class Boosting. In: Proceedings of the 2001 IEEE Conference on Computer Vision and Pattern Recognition. New York, USA: IEEE, 2011. 2585–2592
- 43 Hao Z H, Shen C H, Barnes N, Wang B. Totally-corrective multi-class Boosting. *Computer Vision-ACCV 2010*, 2011, **6495**: 269–280
- 44 Demiriz A, Bennett K P, Shawe-Taylor J. Linear programming Boosting via column generation. *Machine Learning*, 2002, **46**(1–3): 225–254
- 45 Li H X, Shen C H. Boosting the minimum margin: LPBoost vs. AdaBoost. In: Proceedings of the 2008 Digital Image Computing: Techniques and Applications. New York, USA: IEEE, 2008. 533–539
- 46 Warmuth M K, Glocer K A, Vishwanathan S V N. Entropy regularized LPBoost. In: Proceedings of the 2008 Algorithmic Learning Theory: 19th International Conference. Berlin, Germany: Springer-Verlag, 2008. 256–271
- 47 Warmuth M K, Glocer K A, Rätsch G. Boosting algorithms for maximizing the soft margin. *Advances in Neural Information Processing Systems*, 2007, **20**: 1585–1592
- 48 Freund Y, Schapire R E. Game theory, on-line prediction and Boosting. In: Proceedings of the 9th Annual Conference on Computational Learning Theory. New York, USA: ACM, 1996. 325–332
- 49 Freund Y, Schapire R E. Adaptive game playing using multiplicative weights. *Games and Economic Behavior*, 1999, **29**(1–2): 79–103
- 50 Rezek I, Leslie D S, Reece S, Roberts S J, Rogers A, Dash R K, Jennings N R. On similarities between inference in game theory and machine learning. *Journal of Artificial Intelligence Research*, 2008, **33**(1): 259–283
- 51 Hinton G E. Training products of experts by minimizing contrastive divergence. *Neural Computation*, 2002, **14**(8): 1771–1800
- 52 Edakunni N U, Brown G, Kovacs T. Boosting as a product of experts. In: Proceedings of the 27th Conference on Uncertainty in Artificial Intelligence. Virginia, USA: AUAI Press, 2011. 187–194
- 53 Dietterich T G, Bakiri G. Solving multiclass learning problems via error-correcting output codes. *Journal of Artificial Intelligence Research*, 1995, **2**(1): 263–286
- 54 Bautista M Á, Escalera S, Baró X, Radeva P, Vitriá J, Pujol O. Minimal design of error-correcting output codes. *Pattern Recognition Letters*, 2012, **33**(6): 693–702
- 55 Guruswami V, Sahai A. Multiclass learning, Boosting, and error-correcting codes. In: Proceedings of the 12th Annual Conference on Computational Learning Theory. New York, USA: ACM, 1999. 145–155
- 56 Li L. Multiclass Boosting with repartitioning. In: Proceedings of the 23rd International Conference on Machine Learning. New York, USA: ACM, 2006. 569–576
- 57 Sun Y J, Todorovic S, Li J. Unifying multi-class AdaBoost algorithms with binary base learners under the margin framework. *Pattern Recognition Letters*, 2007, **28**(5): 631–643
- 58 Gao T S, Koller D. Multiclass Boosting with hinge loss based on output coding. In: Proceedings of the 28th International Conference on Machine Learning. New York, USA: ACM, 2011. 569–576

- 59 Wang Y Y, Chen S C, Xue H. Can under-exploited structure of original-classes help ECOC-based multi-class classification? *Neurocomputing*, 2012, **85**: 158–167
- 60 Allwein E L, Schapire R E, Singer Y. Reducing multiclass to binary: a unifying approach for margin classifiers. *Journal of Machine Learning Research*, 2001, **1**: 113–141
- 61 Kumar S, Ghosh J, Crawford M M. Hierarchical fusion of multiple classifiers for hyperspectral data analysis. *Pattern Analysis & Applications*, 2002, **5**(2): 210–220
- 62 Jun G, Ghosh J. Multi-class Boosting with class hierarchies. In: Proceedings of the 8th International Workshop on Multiple Classifier Systems. Berlin, Germany: Springer-Verlag, 2009. 32–41
- 63 Mukherjee I, Schapire R E. A theory of multiclass Boosting. *Advances in Neural Information Processing Systems*, 2010, **23**: 1714–1722
- 64 Schwenk H, Bengio Y. Training methods for adaptive Boosting of neural networks for character recognition. *Advances in Neural Information Processing Systems*, 1998, **10**: 647–653
- 65 Howe N R, Rath T M, Manmatha R. Boosted decision trees for word recognition in handwritten document retrieval. In: Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. New York, USA: ACM, 2005. 377–383
- 66 Peng X J, Setlur S, Govindaraju V, Ramachandru S. Using a boosted tree classifier for text segmentation in hand-annotated documents. *Pattern Recognition Letters*, 2012, **33**(7): 943–950
- 67 Hu J, Chen Y B. Writer-independent off-line handwritten signature verification based on real AdaBoost. In: Proceedings of the 2nd International Conference on Artificial Intelligence, Management Science and Electronic Commerce. New York, USA: IEEE, 2011. 6095–6098
- 68 Viola P, Jones M J. Robust real-time face detection. *International Journal of Computer Vision*, 2004, **57**(2): 137–154
- 69 Yang Zhi-Guang, Ai Hai-Zhou. Cluster-based face image retrieval and its relevance feedback. *Acta Automatica Sinica*, 2008, **34**(9): 1033–1039
(杨之光, 艾海舟. 基于聚类的人脸图像检索及相关反馈. 自动化学报, 2008, **34**(9): 1033–1039)
- 70 Sun H L, Shen J, Chen B. LBP based fast face recognition system on symbian platform. *AASRI Procedia*, 2012, **1**: 276–281
- 71 Lei Z, Li S Z. Fast multi-scale local phase quantization histogram for face recognition. *Pattern Recognition Letters*, 2012, **33**(13): 1761–1767
- 72 Ren J F, Jiang X D, Yuan J S. A complete and fully automated face verification system on mobile devices. *Pattern Recognition*, 2013, **46**(1): 45–56
- 73 Wang Q, Zhang X D, Li M Q, Dong X P, Zhou Q H, Yin Y. AdaBoost and multi-orientation 2D gabor-based noisy iris recognition. *Pattern Recognition Letters*, 2012, **33**(8): 978–983
- 74 Ren S K, Hou Y X, Zhang P, Liang X R. Importance weighted AdaRank. In: Proceedings of the 7th International Conference on Advanced Intelligent Computing. Berlin, Heidelberg: Springer-Verlag, 2012, **6838**: 448–455
- 75 Lee J J, Lee P H, Lee S W, Yuille A, Koch C. AdaBoost for text detection in natural scene. In: Proceedings of the 11th International Conference on Document Analysis and Recognition. New York, USA: IEEE, 2011. 429–434
- 76 Schapire R E, Singer Y. BoosTexter: a Boosting-based system for text categorization. *Machine Learning*, 2000, **39**(2–3): 135–168
- 77 Bloehdorn S, Hotho A. Boosting for text classification with semantic features. *Advances in Web Mining and Web Usage Analysis*, 2006, **3932**: 149–166
- 78 Batal I, Hauskrecht M. Boosting KNN text classification accuracy by using supervised term weighting schemes. In: Proceedings of the 18th ACM Conference on Information and Knowledge Management. New York, USA: ACM, 2009. 2041–2044
- 79 Wu X, Ngo C W, Zhu Y M, Peng Q. Boosting web video categorization with contextual information from social web. *World Wide Web*, 2012, **15**(2): 197–212
- 80 Landesa-Vázquez I, Alba-Castro J L. Shedding light on the asymmetric learning capability of AdaBoost. *Pattern Recognition Letters*, 2011, **33**(3): 247–255
- 81 Masnadi-Shirazi H, Vasconcelos N. Cost-sensitive Boosting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2011, **33**(2): 294–309
- 82 Ge J F, Luo Y P. A comprehensive study for asymmetric AdaBoost and its application in object detection. *Acta Automatica Sinica*, 2009, **35**(11): 1403–1409
- 83 Sun Y M, Kamel M S, Wong A K C, Wang Y. Cost-sensitive Boosting for classification of imbalanced data. *Pattern Recognition*, 2007, **40**(12): 3358–3378
- 84 Jan T K, Wang D W, Lin C H, Lin H T. A simple methodology for soft cost-sensitive classification. In: Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. New York, USA: ACM, 2012. 141–149
- 85 Yan X S, Luo Y P. Recognizing human actions using a new descriptor based on spatial temporal interest points and weighted output classifier. *Neurocomputing*, 2012, **87**: 51–61
- 86 Saon G, Soltau H. Boosting systems for large vocabulary continuous speech recognition. *Speech Communication*, 2012, **54**(2): 212–218
- 87 Park J, Diehl F, Gales M J F, Tomalin M, Woodland P C. The efficient incorporation of MLP features into automatic speech recognition systems. *Computer Speech & Language*, 2011, **25**(3): 519–534
- 88 Chen S, Wang J Q, Ouyang Y, Wang B, Xu C S, Lu H Q. Boosting part-sense multi-feature learners toward effective object detection. *Computer Vision and Image Understanding*, 2011, **115**(3): 364–374

- 89 Rios-Cabrera R, Tuytelaars T, Van Gool L. Efficient multi-camera vehicle detection, tracking, and identification in a tunnel surveillance application. *Computer Vision and Image Understanding*, 2012, **116**(6): 742–753
- 90 Cheng Hao, Huang Lei, Liu Chang-Ping, Tan Nu-Tao. A two-level video text localization algorithm based on strokes and AdaBoost. *Acta Automatica Sinica*, 2008, **34**(10): 1312–1318
(程豪, 黄磊, 刘昌平, 谭怒涛. 基于笔画和 AdaBoost 的两层视频文字定位算法. 自动化学报, 2008, **34**(10): 1312–1318)
- 91 Nizamani S, Memon N, Wiil U K. Detection of illegitimate emails using Boosting algorithm. *Counter Terrorism and Open Source Intelligence*, 2011, **2**: 249–264
- 92 Ramanathan V, Wechsler H. Phishing website detection using latent dirichlet allocation and AdaBoost. In: *Proceedings of the 2012 IEEE International Conference on Intelligence and Security Informatics*. New York, USA: IEEE, 2012. 102–107
- 93 Goli B, Nair A S. The elusive short gene — an ensemble method for recognition for prokaryotic genome. *Biochemical and Biophysical Research Communications*, 2012, **422**(1): 36–41
- 94 Mukherjee I, Rudin C, Schapire R E. The rate of convergence of AdaBoost. In: *Proceedings of the 24th Annual Conference on Learning Theory*. Budapest, Hungary: JMLR, 2011. 559–594
- 95 Cao J J, Kwong S, Wang R. A noise-detection based AdaBoost algorithm for mislabeled data. *Pattern Recognition*, 2012, **45**(12): 4451–4465
- 96 Bylander T, Tate L. Using validation sets to avoid overfitting in AdaBoost. In: *Proceedings of the 19th International FLAIRS Conference*. California, USA: AAAI Press, 2006. 544–549
- 97 Mahmood A, Khan S. Early terminating algorithms for AdaBoost based detectors. In: *Proceedings of the 16th IEEE International Conference on Image Processing*. New York, USA: IEEE, 2009. 1209–1212
- 98 Chang Y C I, Huang Y F, Huang Y P. Early stopping in L2 Boosting. *Computational Statistics & Data Analysis*, 2010, **54**(10): 2203–2213
- 99 Meddouri N, Khoufi H, Maddouri M S. Diversity analysis on boosting nominal concepts. *Advance in Computational Intelligence*, 2012, **7301**: 306–317
- 100 Thammasiri D, Meesad P. AdaBoost ensemble data classification based on diversity of classifiers. *Advanced Materials Research*, 2012, **403–408**: 3682–3687



曹莹 西安电子科技大学计算机学院
博士研究生. 主要研究方向为机器学习,
模式识别, 网络安全.

E-mail: yingcao@stu.xidian.edu.cn

(CAO Ying Ph.D. candidate at the
School of Computer Science and Tech-
nology, Xidian University. Her research
interest covers machine learning, pat-
tern recognition, and network security.)



苗启广 西安电子科技大学计算机学院
教授. 主要研究方向为智能图像处理, 机
器学习与模式识别, 恶意程序分析, 数
据/信息融合, 网络计算与嵌入式软件.
本文通信作者.

E-mail: qgmiao@mail.xidian.edu.cn

(MIAO Qi-Guang Professor at the
School of Computer Science and Tech-
nology, Xidian University. His research
interest covers intelligent image processing,
machine learning and pattern
recognition, malware analysis, data/information fu-
sion, network computing, and embedded software. Cor-
responding author of this paper.)



刘家辰 西安电子科技大学计算机学院
博士研究生. 主要研究方向为机器学习,
模式识别, 网络安全.

E-mail: jcliu@stu.xidian.edu.cn

(LIU Jia-Chen Ph.D. candidate at
the School of Computer Science and
Technology, Xidian University. His re-
search interest covers machine learning,
pattern recognitions, and network security.)



高琳 西安电子科技大学计算机学院
教授. 主要研究方向为计算生物信息学,
数据挖掘, 图论与组合优化算法及应用.

E-mail: lgao@mail.xidian.edu.cn

(GAO Lin Professor at the School
of Computer Science and Technology,
Xidian University. Her research inter-
est covers computational biology, data
mining, theory and applications of graph theory and com-
binatorial optimization algorithm.)