

# Boosting 视角<sup>\*</sup>

涂承胜<sup>1,2</sup> 陆玉昌<sup>2</sup>

(重庆三峡学院计算机科学系 重庆万州404000)<sup>1</sup>

(清华大学计算机科学技术系 智能技术与系统国家重点实验室 北京100084)<sup>2</sup>

**摘要** AdaBoost 是 Boosting 家族中的最基础的代表算法。本文主要介绍了 AdaBoost 的泛化错误分析及其与结构风险最小化和 VC 维、支持向量机及 margin 理论的关系,并从游戏理论和统计学视点分别对 AdaBoost 进行了理解和解释,以期提供 Boosting 的一个较为全面的视角。

**关键词** 泛化错误,结构风险最小化,VC 维,支持向量机,游戏理论

## The Analge of View of Boosting

TU Cheng-Sheng<sup>1,2</sup> LU Yu-Chang<sup>2</sup>

(Dept. of Computer Science, Chongqing Three Gorges College, Chongqing Wanzhou 404000)<sup>1</sup>

(Dept. of Computer Science and Technology, Tsinghua University, The State Key Laboratory of

Intelligent Technology and System, Beijing 100084)<sup>2</sup>

**Abstract** AdaBoost is the most important fundamental algorithm in the family of Boosting algorithms. This paper analyzes the generalization error of boosting and its relationship with structural risk minimization and VC dimension, Support Vector Machine and margin theory. Besides, AdaBoost is interpreted from the view point of game theory and statistics respectively, in order to provide an over all view of Boosting for promoting its further study.

**Keywords** Generalization error, Structural risk minimization, VC dimension, Support vector machine, Game theory

## 1 引言

Boosting 是提高预测学习系统预测能力的有效工具,由 Freund 和 Schapire 于1990年提出<sup>[1]</sup>,其代表算法可分为 AdaBoost 和 Boost-by-majority 两个系列。Boosting 操纵训练例子以产生多个假设,从而建立通过投票结合的预测器集合。AdaBoost 在训练例子上维护一套概率分布,在每一回迭代中 Boost-by-majority 通过重取样(resampling)生成不同的训练集,而 AdaBoost 在每个例子上调整这种概率分布。具体的学习算法用以产生成员分类器,并计算出成员分类器在训练例子上的错误率。AdaBoost 用成员分类器在训练例子上的错误率来调整训练例子上的概率分布。权重改变的作用是在被误分的例子上放置更多的权重,在分类器正确的例子上减少其权重。通过单个分类器的加权投票建立最终分类器。每个分类器按照其在训练集上的精度而加权<sup>[2]</sup>。如果产生成员分类器的具体学习算法可以直接使用例子上的分布,则通常会有很好的结果。如果不能,则样本可以随机地有替代地按照概率分布从训练集中抽出。这使得 AdaBoost 更不确定,但实验表明该过程仍然很有效。Boosting 方法一般用于提高不稳定的学习器的性能。

Kearns 和 Valiant 首先提出:在 Valiant 的 PAC 模型<sup>[3]</sup>中一个“弱”学习器是否能被“提升”为一个具有任意精度的“强”学习算法<sup>[4,5]</sup>?1989年 Schapire 通过一个构造性方法<sup>[1]</sup>(其构造过程就是最初的 Boosting 算法)对此问题作了肯定回答。之后, Freund 设计了一个更有效的通过重取样或过滤运作的 Boost-by-majority 算法<sup>[6]</sup>。在某种意义上该算法是优化的,实践上却有一些缺陷。1995年 Freund 和 Schapire 介绍

了通过调整权重而运作的算法:AdaBoost (Adaptive Boost), AdaBoost.M1, AdaBoost.M2, AdaBoost.R, 解决了早期 Boosting 算法很多实践上的困难<sup>[2]</sup>。为解决类别数很大时的多类问题,1997年 Schapire 和 Singer 提出了 AdaBoost 与由 Dietterich 和 Bakiri 提出的 ECOC 算法<sup>[7]</sup>结合的 AdaBoost.OC<sup>[8]</sup>算法。1998年 Schapire 和 Singer 提出了 AdaBoost 的泛化形式,并引入自信率预测以改善 Boosting 的性能。他们还提出了基于汉明距离和损失排序的多类多标签 Boosting 算法 AdaBoost.MH, AdaBoost.MR, 并介绍了 AdaBoost.MH 与 ECOC 结合的 AdaBoost.MO<sup>[9]</sup>算法。1998年 Friedland 等提出了被称之为“Gentle AdaBoost”的 AdaBoost 的变种算法<sup>[10]</sup>,它较少地强调野点(Outliers, 训练样本中被标错或本身就难以分类的样本)。1999年 Freund 介绍了 Boost-by-majority 算法的一个自适应扩展版本 Brown-Boost<sup>[11]</sup>,该算法采用了更激进的方法—不强调那些太“困难”以至于无法正确分类的野点。

## 2 AdaBoost 泛化错误分析

### 2.1 泛化错误一致收敛的理论保证

最终假设  $H$  的泛化错误是它在分布为  $P$  的整个例子空间  $X$  上的错误,即  $\epsilon_f = \Pr_{(x,y) \sim P}[H(x) \neq y]$ 。为了使  $\epsilon_f$  接近训练集上的经验错误  $\hat{\epsilon}$ ,我们必须以某些方式限制  $H$  的选择。对于 Boosting 而言,就是限制弱学习器以使其从一些简单的函数类中选择其假设,且还要限制将要组合成  $H$  的弱假设的数目。弱假设的类别选择与特定的学习问题有关,它应当反映人们对未知概念属性的知识。可以设计出多种通用方法以选择  $T$ ,一流行方法是使用概念类的 VC 维上的一个上界。该方法

<sup>\*</sup>资助项目:重庆市教委科技项目(编号:031104)资助。涂承胜 副教授,研究方向为数据采掘与知识发现,机器学习。陆玉昌 教授,研究方向为数据采掘与知识发现,机器学习,知识工程。

有时也称之为“结构风险最小化”。这里引用 Vapnik 的定理<sup>[12]</sup>：

**定理1(Vapnik)** 设  $h$  是某域  $X$  上的二值函数类。设  $d$  是  $h$  的 VC 维。设  $P$  是  $X \times \{0, 1\}$  上的分布。对于  $h \in h$ ，定义关于  $P$  的  $h$  的(泛化)错误为： $\epsilon_r(h) \approx P_{r(x,y) \sim P}[H(x) \neq y]$ 。设  $S = \{(x_1, y_1), \dots, (x_N, y_N)\}$  是从  $X \times \{0, 1\}$  中按分布  $P$  抽出的  $N$  个独立的随机例子，定义关于  $S$  的  $h$  经验错误为： $\hat{\epsilon}(h) \approx \frac{|\{i: h(x_i) \neq y_i\}|}{N}$ 。于是，对任意的  $\delta > 0$ ，有：

$$Pr[\exists h \in h: |\hat{\epsilon}(h) - \epsilon_r(h)| > 2\sqrt{\frac{d(\ln \frac{2N}{d} + 1) + \ln \frac{9}{\delta}}{N}}] \leq \delta$$

设  $\theta: R \rightarrow \{0, 1\}$  定义为  $\theta(x) = \begin{cases} 1 & \text{若 } x \geq 0 \\ 0 & \text{其它} \end{cases}$ ；并且，设  $\Theta_T(h)$  ( $h$ ) 是  $h$  中所有定义为  $T$  个函数的线性阈值的函数类： $\Theta_T(h) = \{\theta(\sum_{i=1}^T a_i h_i - b) : b, a_1, \dots, a_T \in R; h_1, \dots, h_T \in h\}$ 。很显然，若由 WeakLearn 产生的所有假设都属于某类  $h$ ，则经过  $T$  次迭代后，AdaBoost 的最终假设属于  $\Theta_T(h)$ 。因此，下面的定理提供了由 AdaBoost 产生的最终假设类的 VC 维的上边界：

**定理2** 设  $h$  是 VC 维  $d \geq 2$  的二值函数类。则  $\Theta_T(h)$  的 VC 维最大是： $2(d+1)(T+1)\log_2(e(T+1))$  ( $e$  是自然对数的底)。因此，若由 WeakLearn 产生的假设是从  $d \geq 2$  的类中选出，则  $T$  回迭代后，AdaBoost 产生的最终假设属于一个 VC 维最多是  $2(d+1)(T+1)\log_2(e(T+1))$  的类。沿着结构风险最小的引导，可以这样(假设知道弱假设类的 VC 维的合理上界)：设  $H^T$  是经过 AdaBoost 的  $T$  回迭代后产生的假设，我们能为所有的  $T$  计算出  $H^T$  的泛化错误的上界，之后应选择最小化该上界的假设  $H^T$ 。

虽然结构风险最小化是在数学上较好的方法，这样产生的  $\epsilon_r$  的上界可能比实际值大，且因此  $T$  的选择也可能比最优值小得多，导致不好的性能。一个简单的替代是用交叉验证法：训练集的部分样本不用来产生  $H$ ，而是作为所谓的验证集。于是  $T$  值就被选为使验证集上最终假设错误最小化的  $T$  值。实验表明，AdaBoost 没有过学习(Over fitting)的趋势；在很多问题上，甚至经过数百回的 Boosting 后，泛化错误继续下降，至少不增加。

### 2.2 投票方法有效性的解释

Freund 和 Schapire 描述了如何约束最终假设之泛化错误的一个方法，用 VC 维从训练误差的角度分析 Boosting 的泛化误差(Generalization Error)，VC 维是学习算法的复杂度及其学习能力的度量<sup>[13]</sup>。此处，泛化错误的上界由训练错误  $\delta$ 、取样大小  $m$ 、弱假设空间的 VC 维  $d$  和 Boosting 回数(训练轮数)  $T$  所决定，即  $E \leq f(\delta, m, d, T)$ 。他们使用了来自 Baum 和 Haussler<sup>[14]</sup>的技术以展示有高概率的泛化错误，最多为：

$\hat{Pr}[H(x) \neq y] + O(\sqrt{\frac{Td}{m}})$ 。其中  $\hat{Pr}[\cdot]$  表示训练集的经验概率。该边界表明：若训练轮数过多(即  $T$  大时)，Boosting 将会过分学习。事实上，这确有发生。然而，在早期实验中，一些学者经验性地观察到 Boosting 通常并不过分学习，甚至在运行上万次时也是如此。他们也发现 AdaBoost 有时在训练错误达到零之后的很长一段时间内持续减小泛化错误。这明显与上面的边界相抵触。

为解释(respond)这些经验性发现，Schapire 等从边界(margin)的角度对泛化错误进行了分析<sup>[15]</sup>。边界 margin( $x$ ,

$$y)$$
 的 margin 被定义为： $\text{margin}(x, y) = \frac{y \sum_i a_i h_i(x)}{\sum_i |a_i|}$ 。这是一

个在  $[-1, 1]$  之间的数。当且仅当  $H$  正确地分类了这个例子时其为正。和以前一样，margin 的大小可以被理解为预测自信度的一种测量，较大的正边界表示可信度高的正确的预测，较大的负边界表示可信度高的错误的预测，较小的边界表示可信度低的预测。Schapire 等证明了：较大的训练集上的 margin 可被转换为泛化错误上更高的上边界。Schapire 具体给出了泛化错误的上限，最多为： $\hat{Pr}[\text{margin}_f(x, y) \leq \theta] + O(\sqrt{\frac{d}{m\theta^2}})$ ，对任意有高概率的  $\theta > 0$ 。该边界与  $T$  完全无关，一些实验证明了该理论<sup>[16]</sup>。

另外，Schapire 等证明：Boosting 从计量的角度看，在抵制减少 margin 方面是特别有用的，因为它集中于有最小 margin(不管是正或负)的实例。Boosting 在空间上的影响可以经验性地观测到。他展示了在“letter”数据集上训练样本 margin 的累积分布。在这种情况下，甚至在训练错误达到 0 后，Boosting 持续增加训练样本的 margin，导致测试错误的相应下降。

### 2.3 与统计学习理论、支持向量机的联系

Margin 理论指出了 Vapnik 等人的支持向量机<sup>[16, 17]</sup>和 Boosting 的联系。为此，假定已经找到了我们想要组合的弱假设，且仅对系数  $\alpha_i$  的选择感兴趣。由 AdaBoost 的泛化错误分析得出的一个合理的方法是选择使前面定义的泛化错误边界最小的系数。特别地，假定第一项是 0 使得我们注意第二项。这样，我们就有效地尝试最大化每个训练样本的最小 Margin。更精确地说，与样本  $(x, y)$  相联系的弱假设预测的向量为  $h(x) = \langle h_1(x), h_2(x), \dots, h_N(x) \rangle$ ，系数向量为  $\alpha = \langle \alpha_1, \alpha_2, \dots, \alpha_N \rangle$ ，分别称之为实例向量和权重向量。使用该向量记号和 margin 定义，可以给出最大化最小 margin 的表达式： $\max \min \frac{(\alpha \cdot h(x))y_i}{\|\alpha\| \|h(x)\|}$ 。其中，对 Boosting 来说，分母的范式定义为： $\|\alpha\| \equiv \sum_i |\alpha_i|$ ， $\|h(x)\|_\infty \equiv \max_i |h_i(x)|$ 。(当所有  $h_i$  都有范围  $\{-1, +1\}$  时， $\|h(x)\|$  直接等于 1)。与之相比较，支持向量机(SVM)的目的是最大化和 AdaBoost 形式一样的最小 margin，但其范式是欧氏的： $\|\alpha\|_2 \equiv \sqrt{\sum_i \alpha_i^2}$ ， $\|h(x)\|_2 \equiv \sqrt{\sum_i h_i(x)^2}$ 。因此，SVM 对实例向量和权重向量都使用  $l_2$  范式，而 AdaBoost 对实例向量使用  $l_\infty$  范式，对权重向量使用  $l_1$  范式。在用这种方法描述时，SVM 和 AdaBoost 是很相似的。然而，它们之间也有些较大的差别：

①高维空间中不同的范式可以导致差别很大的 margin。在低维空间中范式  $l_1, l_2$  和  $l_\infty$  的差别可能不是很重要。然而，不管是在 Boosting 还是 SVM 中，其维数通常都很高(常在百万或以上)。这种情况下，范式之间的差别能导致 margin 值间的巨大差异。在仅有少数几个相关变量使得  $\alpha$  很稀疏时尤其如此。例如，假定弱假设都有范围  $\{-1, +1\}$  且所有例子上的标签  $y$  都可以通过这  $k$  个弱假设进行多数投票计算出。这时，可以看出，如果相关弱假设的数目  $k$  是全部弱假设数目的一个小部分，则与 AdaBoost 相联系的 margin 比与 SVM 相联系的 margin 要大得多。

②计算需求不同。最大化 margin 的计算是数学可编程的，即给定一不等式集合最大化某数学表达式。在这方面两种方法的差别是：SVM 对应于二次编程，而 AdaBoost 对应于线

性编程。实际上,AdaBoost 与把它和游戏理论及在线学习联系起来的线性编程有很深的关系。

③空间中用不同的方法进行高效搜索。二次编程比线性编程有更多的计算要求。然而 SVM 和 Boosting 之间有另外的重要得多的计算差别。SVM 和 AdaBoost 有效的部分原因是它们为特别高维(有时甚至是无限维)的空间找到了线性分类器。虽然通过最大化 margin 解决了过于学习的问题,但在高维空间操作相联系的计算问题仍然存在。SVM 通过允许算法执行低维空间计算(这在数学上等价于高维虚拟空间上的内积)的“核”方法来解决该问题。Boosting 采用的是“贪婪搜索”,从这个角度看,弱假设是执行寻找与标签有不可忽略关系的  $H(x)$  的坐标的命令。例子的重新加权改变了分布,依靠该分布相关度得到测量,并引导弱学习器找到有不同相关情况的坐标。大多数应用 SVM 或 AdaBoost 到特定分类问题的实际工作都与选择合适的核函数(Kernel Function)或弱学习算法有关,因核函数和弱学习算法很不一样,导出的学习算法通常在不同的空间运作,产生的分类器也有很大的不同。

### 3 统计视点

#### 3.1 贝叶斯理解

由 AdaBoost 产生的最终假设同贝叶斯分析所建议的假设有关。假设例子  $(x, y)$  按  $X \times \{0, 1\}$  上某分布  $P$  产生。现给定  $\{0, 1\}$  值的一套假设集合  $h_1, \dots, h_T$ , 且我们的目的是以最佳的方式组合这些假设的预测。给定一个例子  $x$  及假设预测  $h_i(x)$ , 贝叶斯最佳决策规则表明,在给定假设值的情况下,应以最大似然预测标签。即:若  $Pr[y=1|h_1(x), \dots, h_T(x)] > Pr[y=0|h_1(x), \dots, h_T(x)]$ , 则预测为 1, 否则预测为 0。如果不同假设的错误是相互独立,且与目标概念独立,即如果我们假定事件  $h_i(x) \neq y$  是条件独立于真实标签  $y$  和所有其它假设  $h_1(x), \dots, h_{i-1}(x), h_{i+1}(x), \dots, h_T(x)$  的预测,则该规则很容易计算。在这种情况下,通过应用贝叶斯规则,能把贝叶斯最佳决策规则重写成一种特别简单的形式:若  $Pr[y=1] \prod_{i: h_i(x)=0} \epsilon_i \prod_{i: h_i(x)=1} (1-\epsilon_i) > Pr[y=0] \prod_{i: h_i(x)=0} (1-\epsilon_i) \prod_{i: h_i(x)=1} \epsilon_i$ , 则预测为 1, 否则为 0。此处  $\epsilon_i = Pr[h_i(x) \neq y]$ 。把其预测为 1 的平凡假设  $h_0$  加入假设集,于是可用  $\epsilon_0$  代替  $Pr[y=0]$ 。不等式两边取对数并整理,则贝叶斯最佳决策规则与 AdaBoost 产生的组合规则是一致的。

如果不同假设的错误相互依赖,贝叶斯最佳决策规则就会变得非常复杂。在实际中,甚至当假设独立性没有得到证实时,也常用上述描述的简单规则(有时称之为“朴素贝叶斯”)。一个更有原则性的替换选择是使用 AdaBoost 算法发现非平凡精度的组合规则。

#### 3.2 估计概率

分类通常是预测某样本  $x$  的标签  $y$  的问题,它希望最小化非正确预测的概率。然而,估计出某特定标签出现的概率通常是有用的。Friedman、Hastie 和 Tibshirani 于 1998 年提出了一种方法<sup>[10]</sup>, 它用 AdaBoost 的输出得到这种概率的合理估计。他们建议用一种 logistic 函数,并估计:  $Pr_f[y=+1|x] = \frac{e^{f(x)}}{e^{f(x)} + e^{-f(x)}}$ 。其中,  $f(x)$  是 AdaBoost 产生的弱假设的加权平均。这种选择的基本原理源于模型  $\sum_i \ln(1 + e^{-2y_i f(x)})$  的对数损失与前面提到过的 AdaBoost 试图最小化的函数  $\sum_i e^{-y_i f(x)}$  之间的紧密联系。可以证明  $\sum_i \ln(1 + e^{-2y_i f(x)})$  是  $\sum_i e^{-y_i f(x)}$  的上界。另外,如果加常数  $1 - \ln 2$  到  $\sum_i \ln(1 +$

$e^{-2y_i f(x)}$ ) 中(它并不影响最小化),可以证明结果与  $\sum_i e^{-y_i f(x)}$  有相同的 0 点附近的 2 阶泰勒(Thaler)展开式。因此,它们在 0 点附近的行为是相似的。最后,可以发现,对任意  $(x, y)$  对上的分布,期望  $E[\ln(1 + e^{-2y f(x)})]$  和  $E[e^{-y f(x)}]$  被同一个函数  $f$  最小化,即  $f(x) = \frac{1}{2} \ln\left(\frac{Pr[y=+1|x]}{Pr[y=-1|x]}\right)$ 。因此,最小化  $\sum_i e^{-y_i f(x)}$ , 如在 AdaBoost 中的那样,可被看作是近似最小化  $\sum_i \ln(1 + e^{-2y_i f(x)})$  中给出的负对数相似的一种方法。因此,可以预期  $Pr_f[y=+1|x] = \frac{e^{f(x)}}{e^{f(x)} + e^{-f(x)}}$  会给出合理的概率估计。

### 4 游戏理论理解

AdaBoost 的行为也可以像 Freund 和 Schapire<sup>[18~22]</sup> 研究的那样在游戏理论环境中得到理解。在经典的游戏理论中,可以把任意二人 0 和游戏以矩阵  $M$  的形式表示。为玩此游戏,其中一个玩家选择某行  $i$ , 另一个玩家则选择某列  $j$ 。行玩家的损失(与列玩家的一样)是  $M_{ij}$ 。双方能随机地玩,在行列上分别选择分布  $P$  和  $Q$ , 于是预期损失为  $P^T M Q$ 。Boosting 可被看作是某游戏矩阵的重复玩。假设弱假设是二值的,且让  $H = \{h_1, h_2, \dots, h_m\}$  为整个弱假设空间(暂时假设其有限)。对一固定训练集  $(x_1, y_2), \dots, (x_m, y_m)$ , 有  $m \times n$  的游戏矩阵  $M$ , 其中  $M_{ij} = \begin{cases} 1 & \text{若 } h_j(x_i) = y_i \\ 0 & \text{其他} \end{cases}$ 。现在行玩家就是 Boosting 算法,列玩家就是弱学习器。训练样本上的 Boosting 算法的分布  $D_t$  的选择变成了  $M$  的行分布  $P$ , 而弱学习器的弱假设  $h_i$  的选择变成了  $M$  的列  $j$  的选择。

作为 Boosting 和游戏理论联系的一个例子,考察 Von Neumann 著名的最小最大定理:  $\max_Q \min_P P^T M Q = \min_P \max_Q P^T M Q$  对任意矩阵  $M$ 。将之应用到刚定义的矩阵上且在 Boosting 环境下重新理解时,就会发现如下意义:如果对样本集上的任意分布,有错误率最多为  $1/2 - \gamma$  的弱假设存在,那么就存在一个弱假设的凸集合,它在所有训练样本上都有最多为  $2\gamma$  的 margin。AdaBoost 找到这样一个最终假设:它通过综合很多弱假设使其在所有样本上都有高的 margin。AdaBoost 至少有成功的可能,因为,给定一“好的”弱学习器,可以发现 AdaBoost 是玩重复游戏或近似解决矩阵游戏的更一般算法的一种特殊情况。这就表明,按游戏理论的理解,训练例上的分布和最终假设中各弱假设上的权重逐步近似于最小最大或最大最小策略。

### 5 多类扩展

有几个把 AdaBoost 扩展到多类情况的方法。最直接的泛化叫 AdaBoost.M1, 当弱学习器强到在 AdaBoost 产生的困难分布上也能取得合适的高精度时它足以解决多类问题。如果弱学习器不能在这些困难分布上至少获得 50% 的精度则该方法会失败。对此情形,常常通过简化多类问题至更大一些的二值问题来导出更高级的方法加以解决。AdaBoost.MH 通过对每个样本  $x$  和每个标签  $y$  产生一套如下形式的二值问题来运作:“对样本  $x$ , 正确标签是  $y$  还是其他?”, AdaBoost.M2 (AdaBoost.MR 的一种特殊情况) 对每个有正确标签  $y$  的样本  $x$  和每个非正确标签  $y'$  提出另一种形式的二值问题:“对  $x$ , 正确标签是  $y$  还是  $y'$ ”? 这些方法需要在设计弱学习器算法中付出额外的努力。AdaBoost.OC 是另一种解

决方案。它与 ECOC 结合,获得了与 AdaBoost.MH 和 AdaBoost.M2 相似的可证明边界,其特点是它可以与任意能处理简单二值标签的弱学习器相结合。

## 6 应用

关于 Boosting 的大量实验和应用证明了该方法的有效性。这使得在实际应用中,可以不再寻找通常很难获得的预测精度很高的强学习算法,只需找出一个精度略好于随机预测的弱学习算法,就可以通过 Boosting 方法大幅度提高预测算法的预测准确率,从而促进了机器学习成果的广泛应用。在文本挖掘中 Boosting 用于文本分类,这是 AdaBoost 的多类多标签情况。其中弱假设被用于测试某字或短语出现与否。它还可以用于文本过滤, Schapire 等将 Boosting 应用到文本过滤,并与从信息检索角度所提出的 Rocchio 算法进行比较,其实验结果表明在提供的训练文本集较大的情况下, AdaBoost 算法的效果比 Rocchio 算法好,但比 Rocchio 算法耗时。在网络挖掘中还可用于网络用户行为分析(如偏好排序),与决策树等学习方法的结合,如 Boosting 神经网络、决策树等。Quinlan 把 Boosting 应用于 C4.5 决策树。其实验结果表明,在一般情况下 Boosting 都能明显减少泛化错误。Boosting 在数据挖掘中也有了越来越多的应用。Boosting 在自然语言理解和多媒体信息挖掘中语音理解、语音识别和分类、图像数据库中图像检索和识别、手写体的字符识别等方面的应用也有了一定的进展。Boosting 可以用来产生出与 RIPPER、IREP 和 C4.5rules 等系统相似的规则列表。还可与“随机属性选择社团”结合以提高决策树性能。

**结束语** AdaBoost 具有快、简单、易于编程的优点,除了迭代次数  $T$  外不需调整参数;不需要弱学习器的先验知识,因此可以灵活地与任意方法结合寻找弱假设;给定足够数据和一个能够可靠地仅提供中等精度的弱学习器,它可以提供学习的一套理论保证。但 AdaBoost 在特定问题上的实际性能很明显依赖于数据和弱学习器;与理论相一致, AdaBoost 在不足的数据集,过于复杂的或太弱的弱假设上性能不好;对噪声是乎很敏感。

本文主要介绍了 AdaBoost 的泛化错误及其与结构风险最小化、VC 维、支持向量机及 margin 理论的关系,并从游戏理论和统计学视点分别对 AdaBoost 进行了理解和解释,以期提供 Boosting 的较为全面的视角。

(上接第136页)

堆栈集具有统一的结构模式和应用接口。

**结论** 本文对“软件人”的概念、概念模型和构造特征进行了论述,对其构造特征进行了分析,使其能在网上形成独立生存的“软件动态实体”。课题目前进行的“软件人”原型构造研究和受限(约束条件下)网上迁移机制与实现技术研究正是基于本文的研究结论。实践证明本文提出的软件人模型和构造架构是合理的,是可行的理论范式之一,但其完备性要在应用中补充。

## 参考文献

- 1 Ferber J. Multi-agent Systems. Addison-Wesley, Reading, MA, 1999
- 2 Labrou Y, Finin T, peng Y. Agent communication language: current landscape. IEEE Intelligent Systems, 1999, 14(2): 45~52
- 3 焦文品, 史忠植. 构造 MAS 的动态体系结构的模型. 计算机学报, 2000(7)

## 参考文献

- 1 Schapire R E. The strength of weak learnability. Machine Learning, 1990, 5(2): 197~227
- 2 Freund Y, Schapire R E. A decision-theoretic generalization of on-line learning and an application to boosting. Journal of Computer and System Science, 1997, 55(1): 119~139
- 3 Valiant L G. A theory of the learnable. Communications of the ACM, 1984, 27(11): 1134~1142
- 4 Kearns M J, Vazirani L G. Learning Boolean formulae or finite automata is as hard as factoring. [Technical Report TR-14-88]. Harvard University Aiken Computation Laboratory, Aug. 1988
- 5 Kearns M J, Vazirani L G. Cryptographic limitations on learning Boolean formulae and finite automata. Journal of the Association for Computing Machinery, 1994, 41(1): 67~95
- 6 Freund Y. Boosting a weak learning algorithm by majority. Information and computation, 1994, 141(2): 256~285
- 7 Dietterich T G, Bakiri G. Solving multiclass learning problems via error-correcting output codes. Journal of Artificial Intelligence Research, 1995, 2: 263~286
- 8 Schapire R E, Singer Y. Using output codes to boost multiclass learning problems. In: Machine Learning: Proc. of the Fourteenth Intl. Conf. 1997. 313~321
- 9 Schapire R E, Singer Y. Improved boosting algorithms using confidence-related predictions. In: Proc. of the eleventh Annual Conf. on Computational Learning Theory, 1998. 80~91
- 10 Friedman J, Hastie T, Tibshirani R. Additive logistic regression: a statistical view of boosting. [Technical Report]. 1998
- 11 Freund Y. An adaptive version of the boost by majority algorithm. In: Proc. of the Twelfth Annual Conf. on Computational Learning Theory, 1999
- 12 Vapnik V N. Estimation of Dependences Based on Empirical Data. Springer-Verlag, 1982
- 13 Schapire R E, Freund Y, Bartlett Y, et al. Boosting the margin: A New Explanation for the Effectiveness of Voting Methods. The Annals of Statistics, 1998, 26(5): 1651~1686
- 14 Baum E B, Haussler D. What size net gives valid generalization?. Neural Computation, 1989, 1(1): 151~160
- 15 Schapire R E. A Brief Introduction of Boosting. In: Proc. of the 16<sup>th</sup> Intl. Joint Conf. on Artificial Intelligence, 1999
- 16 Cortes C, Vapnik V. Support-vector networks. Machine Learning, 1995, 20(3): 273~297
- 17 Vapnik V N. The Nature of Statistical Learning Theory. Springer, 1995
- 18 Freund Y, Schapire R E. Game theory, on-line prediction and boosting. In: Proc. of the Ninth Annual Conf. on Computational Learning Theory, 1996. 325~332
- 19 Freund Y, Schapire R E. Adaptive game playing using multiplicative weights. Games and Economics Behavior, 1997
- 20 Schapire R E. Drifting games. In: Proc. of the Twelfth Annual Conf. on Computational Learning Theory, 1999
- 21 Breiman L. Prediction games and arcing classifiers. [Technical Report 504]. Statistics Department, University of California at Berkeley, 1997
- 22 Grove A J, Schuurmans D. Boosting in the limit: Maximizing the margin of learned ensembles. In: Proc. of the Fifteenth National Conf. on Artificial Intelligence, 1998

- 4 帅典勋, 顾静. 多 Agent 系统分布式问题求解的代数模型方法(2). 群体智能和社会动力学. 计算机学报, 2002(2)
- 5 朱鑫良. 基于多智能体的自主式移动机器人的体系结构. 见: 中国人工智能进展 2001年. 2001
- 6 蔡自兴, 郭军华. 基于多 Agent 的移动机器人导航进化控制的体系结构. 控制理论与应用, 2001, S1
- 7 钟义信. 意识机: 理论与模型. 电子学报, 2000(10)
- 8 刘新宇, 洪炳锐. 基于 BDI 框架的多 Agent 动态协作模型与应用研究. 计算机研究与发展, 2002(7)
- 9 陶先平, 吕建, 等. 一种移动 agent 结构化迁移机制的设计和实现. 软件学报, 2000(7)
- 10 涂序彦. 大系统控制论. 北京: 国防工业出版社, 1994
- 11 涂序彦, 等. 智能管理. 清华大学出版社, 1995
- 12 曾广平. 异质网智能隔离与交换技术及其应用. 通信学报, 2000(3)
- 13 曾广平, 等. 工程专家系统模型与开发方法的研究. 高技术通讯, 2000(9)
- 14 涂序彦. 广义人工生命. 见: 中国人工智能学会第九届全国人工智能学术年会论文集. 北京邮电大学出版社, 2001
- 15 Tu Xuyan. AI, AL, and Robotics. In: Proc. of FIRA World Congress (plenary speech), Korea, 2002
- 16 Tu Xuyan. Generalized Artificial Life Race and Model. In: Proc. of the 8th AROB, Japan, 2003