

文章编号:1001-9081(2010)02-0465-04

## 基于属性组合的集成学习算法

付忠良,赵向辉,苗青,姚宇

(中国科学院成都计算机应用研究所,成都 610041)

(fzliang@netease.com)

**摘要:**针对样本由数字属性构成的分类问题,在 AdaBoost 算法流程基础上,改传统的基于单属性分类器构造方法为基于组合属性分类器构造方法,提出了一种基于样本属性线性组合的集成学习算法。对属性组合系数的构造,提出了一般性的构造思路,按照该思路,提出了几种具体的组合系数构造方法,并对构造方法的科学合理性进行了分析。利用 UCI 机器学习数据集中的数据对提出的方法进行了实验与分析,结果表明,基于属性组合的集成学习算法不仅有是有效的,而且比传统 AdaBoost 算法好。

**关键词:** AdaBoost 算法;属性组合;集成学习;分类器组合

**中图分类号:** TP181; TP391.41 **文献标志码:** A

## Ensemble learning algorithm on attribute combination

FU Zhong-liang, ZHAO Xiang-hui, MIAO Qing, YAO Yu

(Chengdu Institute of Computer Application, Chinese Academy of Sciences, Chengdu Sichuan 610041, China)

**Abstract:** Concerning the classification of samples being composed of digital attributes, an ensemble learning algorithm based on linear combination of samples attributes was proposed. It constructed classifiers based on combined attributes instead of single attribute by traditional AdaBoost algorithm. The general construction idea for attribute combination coefficients was put forward. In accordance with the idea, several concrete construction methods for combination coefficients were given and analyzed to be scientific and reasonable. The experimental results on UCI machine learning dataset illustrate that the ensemble learning algorithm based on attribute combination is effective and outperforms AdaBoost algorithm.

**Key words:** AdaBoost algorithm; attribute combination; ensemble learning; classification combination

### 0 引言

AdaBoost(Adaptive Boosting)算法<sup>[1]</sup>是目前机器学习领域中被广泛研究并得到大量应用的集成学习算法,算法先训练多个分类精度一般的分类器,然后对其进行线性组合(加权投票)以提升分类精度,在一定条件下,组合分类器的分类精度可以无限制地得到提升<sup>[2-4]</sup>。AdaBoost 算法的有效性在文本分类<sup>[5]</sup>、人脸检测<sup>[6-8]</sup>中得到了很好的验证,其中 Viola 和 Jones 提出的基于 AdaBoost 的人脸检测器<sup>[6]</sup>已成为人脸检测中的主流算法。

在 AdaBoost 算法中,各个分类器需要通过一种弱学习算法基于训练样本集训练获得,常用的弱学习算法有判定树、支持向量机(Support Vector Machine, SVM)、神经网络等方法。当样本以多个数字属性表示时,最简单且最常用的训练方法就是判定树方法,它针对样本的每个属性构造一个单属性分类器,分类阈值取正类样本属性均值与负类样本属性均值的平均值。所谓训练一个分类器就是有条件地选取一个分类器,使得新选取的分类器和已选取的分类器组合后,得到的组合分类器的分类精度能有所提高<sup>[4]</sup>。这种方法针对每个属性只能构造一个分类器,即训练时可供选择的分类器个数与属性个数一样多,当属性个数不多时,AdaBoost 的通过更多分类器组合来提升分类精度这一重要特点就难以得到体现,此时,怎样才能构造任意多的单属性分类器问题值得探讨。

另一方面,当把具有数字属性的样本看成向量,分类问题就是高维空间的一个划分问题,仿照支持向量机思想,对其进行映射转换,在新的空间来构造并选取分类器是否会有更好的结果,也是一个值得研究与探讨的问题。

本文就上述两个问题进行了探讨,对样本带数字属性的分类问题进行了深入研究。对样本属性进行线性组合时,通过组合系数的变化可以构造任意多的组合属性分类器,当新增组合属性分类器时,组合系数的选择始终面向最终组合分类器的分类精度提升这一目标就有望得到好的集成学习算法。本文对提出的基于属性组合的集成学习算法的合理性进行了分析,并通过 UCI 实验数据对提出的集成学习算法进行了验证。目前针对 AdaBoost 算法的研究多集中在样本权值和分类器组合权值的改进上<sup>[9-12]</sup>,而对其起关键作用的弱训练方法研究得却不多,本文进行了相应研究。

### 1 单属性分类的 AdaBoost 算法

设训练样本集  $S = \{(x_1, y_1), (x_2, y_2), \dots, (x_m, y_m)\}$ ,  $y_i \in \{-1, +1\}$ , 每个样本含  $d$  个属性,即  $x_i = (x_{i1}, x_{i2}, \dots, x_{id})$ 。记弱学习算法为  $P$ ,则 AdaBoost 算法<sup>[1]</sup>流程为:

1) 初始化权值:  $\omega_i^1 = 1/m, i = 1, \dots, m$ ;

2) DO FOR  $t = 1, \dots, T$

① 用弱学习算法  $P$ ,基于带权值  $\omega_i^t$  的训练集  $S$  得到弱分类器:  $h_t(x): x \rightarrow \{-1, +1\}$ 。

收稿日期:2009-08-26;修回日期:2009-10-27。 基金项目:四川省科技支撑计划基金项目(2008SZ0100;2009SZ0214)。

**作者简介:**付忠良(1967-),男,重庆人,研究员,博士生导师,主要研究方向:计算机视觉、机器学习;赵向辉(1982-),男,河南长葛人,博士研究生,主要研究方向:机器学习、数据挖掘;苗青(1982-),男,四川成都人,博士研究生,主要研究方向:计算机视觉、模式识别;姚宇(1980-),男,四川宜宾人,博士,主要研究方向:机器学习、模式识别。

② 计算  $h_t(x)$  的错误率:  $\varepsilon_t = \sum_{i=1}^m \omega_i^t [h_t(x_i) \neq y_i]$ , 令  $\alpha_t = \frac{1}{2} \ln((1 - \varepsilon_t)/\varepsilon_t)$ 。

③ 调整样本权值 ( $Z_t$  为归一化因子):

$$\omega_i^{t+1} = \begin{cases} \frac{\omega_i^t}{Z_t} \times e^{-\alpha_t}, & h_t(x_i) = y_i \\ \frac{\omega_i^t}{Z_t} \times e^{\alpha_t}, & h_t(x_i) \neq y_i \end{cases}$$

3) 循环结束后集成分类器为:

$$H(x) = \text{sign}(f(x)); f(x) = \sum_{t=1}^T \alpha_t h_t(x)$$

AdaBoost 算法给错分样本更大权值, 集成分类器采取加权组合 (投票), 错误率小的投票权值大。

当弱学习算法采取判定树方法时, 上述算法中的①将包含以下步骤:

第1步 构造  $d$  个单属性分类器:

$$g_j(x_i) = \begin{cases} x_i \rightarrow +1, & x_{ij} > v_j; j = 1, \dots, d \\ x_i \rightarrow -1, & x_{ij} \leq v_j \end{cases} \quad (1)$$

其中分类阈值  $v_j = \frac{1}{2} \left( \frac{\sum_{i: y_i=1} x_{ij}}{\sum_{i: y_i=1} 1} + \frac{\sum_{i: y_i=-1} x_{ij}}{\sum_{i: y_i=-1} 1} \right)$ , 即正类样本和负类样本第  $j$  个属性均值的平均值。

第2步 统计  $g_j(x)$  的训练错误率:  $\varepsilon_j = \sum_{i=1}^m \omega_i^t [g_j(x_i) \neq y_i]$ , 如果  $\varepsilon_j > 1/2$ , 用  $-g_j(x)$  代替  $g_j(x)$ 。

第3步 选取  $h_t(x)$ :  $h_t(x)$  选取  $\varepsilon_j$  最小对应的  $g_j(x)$ 。

上述 AdaBoost 算法中的弱学习算法采用了单属性分类器, 简称为单属性分类的 AdaBoost 算法。

## 2 属性组合的集成学习算法

单属性分类器依据一个属性只能构造一个分类器, 属性之间的互补性只能在最终分类器组合时得到体现, 如果在构造分类器时就考虑属性之间的互补性, 可以对属性进行组合, 然后在组合属性上构造单属性分类器。比如第  $i$  个样本组合属性为  $s_i = \sum_{j=1}^d c_j x_{ij}$ , 在组合属性  $s_i (i = 1, \dots, m)$  上可以构造单属性分类器, 其分类阈值仍然用正类样本和负类样本的组合属性均值的平均值, 当用于测试时, 对待分类样本或目标先进行同样的组合, 然后与该阈值比较完成分类。

显然, 不同的组合系数可以得到不同的组合属性, 进而得到不同的分类器, 要得到  $T$  个组合属性分类器, 相当于作如下的线性变换或映射, 在变换后的属性上构造单属性分类器。

$$\begin{pmatrix} s_{11} & \dots & s_{1T} \\ \vdots & & \vdots \\ s_{m1} & \dots & s_{mT} \end{pmatrix} = \begin{pmatrix} x_{11} & \dots & x_{1d} \\ \vdots & & \vdots \\ x_{m1} & \dots & x_{md} \end{pmatrix} \cdot \begin{pmatrix} c_{11} & \dots & c_{1T} \\ \vdots & & \vdots \\ c_{d1} & \dots & c_{dT} \end{pmatrix} \quad (2)$$

记成矩阵形式为  $S_{m \times T} = X_{m \times d} C_{d \times T}$ 。

于是, 组合属性分类器问题可简单描述为: 含  $d$  个属性的  $m$  个样本  $X_{m \times d}$ , 通过线性变换  $C_{d \times T}$  而成为含  $T$  个属性的  $m$  个样本  $S_{m \times T}$ , 在  $S_{m \times T}$  上构造  $T$  个单属性分类器, 再对其进行加权组合。

当最终分类器的加权组合系数仍然用  $\alpha_t = \frac{1}{2} \ln((1 -$

$\varepsilon_t)/\varepsilon_t)$ , 其中  $\varepsilon_t$  为第  $t$  个组合属性的单属性分类器的训练错误率, 则前面介绍的单属性分类的 AdaBoost 算法的实质, 便是求取变换矩阵  $C_{d \times T}$ , 使得在变换后属性上得到的单属性分类器之加权组合可以有更低的分类错误率, 只是限制  $C_{d \times T}$  中每列仅一个元素为 1 其余为 0, 即列向量为单位向量。当对  $C_{d \times T}$  不作这样的限制来求取  $C_{d \times T}$ , 则变换矩阵  $C_{d \times T}$  就从有限的可选对象中选取变成从无限的可选对象中选取, 也就是说, 前者只是后者的特例。从这一点分析可见, 采取属性线性组合方法, 应该有更好的结果, 因为传统的 AdaBoost 算法只是一种非常特殊的属性线性组合方法, 一种限制其组合系数向量  $(c_{1t}, \dots, c_{dt})$  为单位向量的特殊方法。

基于属性组合的一般集成学习算法问题可以描述为:

对带  $d$  个属性的  $m$  个样本  $X_{m \times d}$ , 寻找线性变换  $C_{d \times T}$ , 使得在带  $T$  个新属性的  $m$  个样本  $S_{m \times T} = X_{m \times d} C_{d \times T}$  上的  $T$  个单属性分类器的组合分类器分类精度最小。

于是, 有基于属性组合的一般性集成学习算法:

1) 初始化权值:  $\omega_i^1 = 1/m, i = 1, \dots, m$ ;

2) DO FOR  $t = 1, \dots, T$

① 构造组合系数  $c_{1t}, \dots, c_{dt}$ , 基于组合属性  $s_{it} = \sum_{j=1}^d c_{jt} (\omega_j^t x_{ij}), i = 1, \dots, m$ , 构造单属性分类器:  $h_t(x): x \rightarrow \{-1, +1\}$ 。

② 计算  $h_t(x)$  的错误率:  $\varepsilon_t = \sum_{i=1}^m \omega_i^t [h_t(x_i) \neq y_i]$ , 令  $\alpha_t = \frac{1}{2} \ln((1 - \varepsilon_t)/\varepsilon_t)$ 。

③ 调整样本权值 ( $Z_t$  为归一化因子):

$$\omega_i^{t+1} = \begin{cases} \frac{\omega_i^t}{Z_t} \times e^{-\alpha_t}, & h_t(x_i) = y_i \\ \frac{\omega_i^t}{Z_t} \times e^{\alpha_t}, & h_t(x_i) \neq y_i \end{cases}$$

3) 循环结束后集成分类器为:

$$H(x) = \text{sign}(f(x)); f(x) = \sum_{t=1}^T \alpha_t h_t(x)$$

在上述算法中, 当对求取的组合系数向量  $(c_{1t}, \dots, c_{dt})$  限制为单位向量时, 就变成传统的基于单属性分类的 AdaBoost 算法, 而上述算法显然极大地扩展了 AdaBoost 算法。

上述算法对组合系数向量  $(c_{1t}, \dots, c_{dt})$  不作任何限制, 则可选空间由有限变成无限, 从理论上讲, 上述方法包含传统 AdaBoost 算法, 最坏结果也是 AdaBoost 算法结果, 因此, 只要有好的求取  $C_{d \times T}$  的方法, 都应好于传统的 AdaBoost 算法。不仅如此, 上述方法因为对组合系数向量  $(c_{1t}, \dots, c_{dt})$  的无限限制, 理论上可以得到无限多的组合, 进而可以构造出无限多的组合属性分类器, 从中选择一些来加权组合得到组合分类器, 正好可以发挥弱学习定理的一个重要特性: 利用越来越多的弱分类器来加权组合使得组合分类器分类精度越来越高, 而传统的 AdaBoost 算法只能构造与属性个数一样多的分类器, 上述弱学习定理特性在属性个数较少时, 往往很难得到充分体现。

到此, 基于属性组合构造新的集成学习算法问题, 按照上述思路, 就转变成如何求取变换矩阵  $C_{d \times T}$  的问题, 而要求取矩阵  $C_{d \times T}$ , 一种可行思路就是参照 AdaBoost 算法思路逐级构造, 即先构造第一列, 得到一个组合属性, 进而得到一个组合属性分类器, 然后按照如下原则构造第二列第三列等后续组

合系数:新的组合属性分类器与已有的组合属性分类器采取加权组合后的分类精度能得到逐步提升。这种完全面向最终目标(分类精度提升)的组合系数构造方法,也正是 AdaBoost 算法思路。

下面就几种  $C_{d \times T}$  构造具体方法进行分析。

### 3 属性组合系数的构造

#### 3.1 组合系数构造基本思路

先分析单属性分类的 AdaBoost 算法,其第一个组合系数,其实质是第一个单属性选择,AdaBoost 算法为选取单属性分类器训练错误率最小对应的属性,然后根据其是否正确分类样本而调整样本权值,让被错分的样本权值增大。第二个组合系数,其实质是第二个属性选择,AdaBoost 算法选取以样本权值为加权系数的加权训练错误率最小对应的属性为第二个属性,因为权值关系,这样选取的结果是:让选取的属性对应的分类器向错分样本聚焦,从而使最终的组合分类器的分类错误率降低。采用类似的权值调整策略和属性选取方法,直到选取  $T$  个属性。这种方法的合理性在文献[3-4]中有很好的说明。

以完全类似的思路,组合系数构造的基本思路如下:

第一个组合系数  $c_{11}, \dots, c_{d1}$ , 可以选取使组合属性  $s_{11} = \sum_{j=1}^d c_{j1}(\omega_j^1 x_{ij})$  上的单属性分类器的训练错误率最小者,然后采取类似的样本权值调整策略,增大错分样本的权值。第二个组合系数  $c_{12}, \dots, c_{d2}$ , 可以选取使带新样本权值的组合属性  $s_{12} = \sum_{j=1}^d c_{j2}(\omega_j^2 x_{ij})$  上的单属性分类器训练错误率最小者,然后用类似的样本权值调整策略和组合系数构造方法,直到构造出  $T$  个组合系数  $C_{d \times T}$ 。这便是基于组合属性集成学习算法的组合系数构造一般方法。

由单属性分类器构造公式(1)知,  $(x_{ij} - v_j)$  或者  $-(x_{ij} - v_j)$  的符号正好为第  $j$  个单属性分类器分类结果(假设值为 0 时符号取负),当某个属性的正类属性均值小于负类属性均值时,  $-(x_{ij} - v_j)$  的符号正好为第  $j$  个单属性分类器的分类结果,此时可以对该属性值取反,在用于测试时也先对该属性值取反再利用分类器分类。因此,处理带数字属性的分类问题时,在已知的训练样本集上,可以对正类属性均值小于负类属性均值的属性值先取反,反映在组合系数上,便是对组合系数取反。基于这样的分析,不失一般性,在下面的讨论中,可以假定  $(x_{ij} - v_j)$  符号正好为第  $j$  个单属性分类器分类结果。

#### 3.2 组合系数的构造方法一

第  $t$  轮组合属性为  $s_{it} = \sum_{j=1}^d c_{jt} \omega_j^t x_{ij}$ , 基于该属性的单属性分类器的阈值可以用相同的组合,即分类阈值取  $v_i = \sum_{j=1}^d c_{jt} \omega_j^t v_j$ , 类似上面的分析,希望基于组合属性  $s_{it}$  的单属性分类器的分类正确率最大,相当于希望式(3)中大于零的比率最大。

$$L_i = (s_{it} - v_i) y_i; i = 1, \dots, m \quad (3)$$

直接求解该问题是困难的,但因为最终是多个分类器组合来使用,每个分类器不一定非得得到最优,因此不一定求最大值点,求取极大值点或近似于极大值点也是可行的。

$m$  个样本的第  $j$  个属性值,可以认为是得到的  $m$  个观测值,即  $\omega_j^t(x_{ij} - v_j) y_i (i = 1, \dots, m)$ , 它可以被认为是某个随机

变量  $z_j$  的  $m$  个观测值,于是式(3)得到的  $m$  个值就相当于组合随机变量  $z = \sum_{j=1}^d c_{jt} z_j$  的  $m$  个观测值,求  $c_{1t}, \dots, c_{dt}$  使得式(3)中大于零的比率极大,可转变为求  $c_{1t}, \dots, c_{dt}$ , 使得随机变量  $z < 0$  的概率最小。

可以证明:随机变量  $z_j$  之间相互独立,且  $d$  很大,  $c_{jt} = \mu_j / \sigma_j^2$  时,可以使得  $z = \sum_{j=1}^d c_{jt} z_j < 0$  的概率最小<sup>[4]</sup>, 其中  $\mu_j, \sigma_j^2$  分别为随机变量  $z_j$  的均值和方差,即:  $\mu_j = \frac{1}{m} \sum_{i=1}^m \omega_j^t(x_{ij} - v_j) y_i, \sigma_j^2 = \frac{1}{m} \sum_{i=1}^m (\omega_j^t(x_{ij} - v_j) y_i - \mu_j)^2$ 。

样本的  $d$  个属性取值可以近似假设具有一定的独立性,根据上面的分析,于是有组合系数构造方法 1:

$$c_{jt} = \mu_j / \sigma_j^2 \quad (4)$$

其中  $\mu_j = \frac{1}{m} \sum_{i=1}^m \omega_j^t(x_{ij} - v_j) y_i, \sigma_j^2 = \frac{1}{m} \sum_{i=1}^m (\omega_j^t(x_{ij} - v_j) y_i - \mu_j)^2$ 。

上面从概率的角度分析了组合系数构造公式(4)的合理性,下面还可以从另一角度来说明组合系数构造公式(4)的合理性:

为使式(3)中大于零的比率极大,一种简单近似的办法是使这  $m$  个元素之和极大,于是有:

$$\begin{aligned} \sum_{i=1}^m L_i &= \sum_{i=1}^m \sum_{j=1}^d c_{jt} (\omega_j^t(x_{ij} - v_j) y_i) = \\ &= \sum_{j=1}^d c_{jt} \sum_{i=1}^m (\omega_j^t(x_{ij} - v_j) y_i) \leq \\ &= \left\{ \sum_{j=1}^d c_{jt}^2 \sum_{i=1}^m \left[ \sum_{i=1}^m (\omega_j^t(x_{ij} - v_j) y_i) \right]^2 \right\}^{1/2} \end{aligned} \quad (5)$$

根据不等式定理,当且仅当  $c_{jt} \propto \sum_{i=1}^m (\omega_j^t(x_{ij} - v_j) y_i)$  时,左边取到极大值。

取值变化范围小的属性在组合中将会被取值变化范围大的属性所掩盖,此时组合系数的作用将受到影响,因此,在构造组合属性前,用属性的方差对属性进行归一化处理是合理的,这个归一化系数当然可以放在组合系数中,这样,我们就得到与式(4)一样的结论。

#### 3.3 组合系数的构造方法二

类似于文献[4]中讨论分类错误相互独立的分类器  $h_i(x)$  的最佳组合问题时,得到分类器最佳组合系数为  $(1 - 2\varepsilon_i) / (4(1 - \varepsilon_i)\varepsilon_i)$  ( $\varepsilon_i$  为  $h_i(x)$  的训练错误率),它正好是按照式(6)定义的随机变量的均值与方差之比。

$$z_i = \begin{cases} +1, & h_i(x_i) = y_i \\ -1, & h_i(x_i) \neq y_i \end{cases} \quad (6)$$

在文献[4]中讨论了最佳组合系数与 AdaBoost 算法组合系数  $\alpha_i = \frac{1}{2} \ln((1 - \varepsilon_i) / \varepsilon_i)$  之间的关系,完全对照到这里,我们便可以得到另一种组合系数构造方法,于是有组合系数构造方法二:

$$\begin{aligned} c_{jt} &= \frac{1}{2} \ln(\theta_j^+ / \theta_j^-); \\ \theta_j^+ &= \sum_{i: (x_{ij} - v_j) y_i > 0} \omega_j^t(x_{ij} - v_j) y_i \\ \theta_j^- &= \sum_{i: (x_{ij} - v_j) y_i \leq 0} \omega_j^t(x_{ij} - v_j) y_i \end{aligned} \quad (7)$$

### 3.4 组合系数的构造方法三

为了使式(3)中大于零的比率极大,一种间接的办法是考虑正类组合属性与负类组合属性的距离尽量大,当两类之间的距离很大时,则更容易通过一个分类阈值来把两类分开。基于这样的思路,只要能很好地度量两类之间的距离,然后按照两类距离最大化原则来构造组合系数,就可以得到相应的组合系数构造方法。

一种最简单的度量方法便是二者中心之差,而均值可以近似代表一个类的中心。类似上面的分析思路,可以让  $c_{jk}$  取正类样本第  $j$  个属性的均值与负类样本第  $j$  个属性的均值之差,同时为了克服属性取值范围差异带来的影响,可以考虑用属性方差来进行修正与克服,因此,让  $c_{jk}$  与正类样本第  $j$  个属性的方差和负类样本第  $j$  个属性的方差之和成反比是合理的。于是有组合系数构造方法三:

$$\begin{cases} c_{jk} = (\mu_j^+ - \mu_j^-) / (\sigma_j^+ + \sigma_j^-); \\ \mu_j^+ = \sum_{i: y_i > 0} \omega_i^+ x_{ij} / \sum_{i: y_i > 0} 1 \\ \mu_j^- = \sum_{i: y_i \leq 0} \omega_i^- x_{ij} / \sum_{i: y_i \leq 0} 1 \\ (\sigma_j^+)^2 = \sum_{i: y_i > 0} (\omega_i^+ x_{ij} - \mu_j^+)^2 / \sum_{i: y_i > 0} 1 \\ (\sigma_j^-)^2 = \sum_{i: y_i \leq 0} (\omega_i^- x_{ij} - \mu_j^-)^2 / \sum_{i: y_i \leq 0} 1 \end{cases} \quad (8)$$

需要说明的是,上述几种组合系数构造方法,尽管考虑了方差,但在具体使用时,还存在一些不足。当两个属性值相差很大时,则在进行属性组合时组合系数的变化将会被属性值的差异所掩盖,为此,在进行训练前,有必要对各个属性取值进行归一化处理,如第  $j$  个属性除以其最大值与最小值之差,其反映在组合系数上,便是对得到的组合系数,除以对应属性最大值与最小值之差。在实际应用时,则不需要事先归一化,只需要在得到组合系数后再用属性最大值与最小值之差来调整即可。

## 4 实验结果与分析

前面提出的基于属性组合的集成学习算法,对应为几种组合系数构造方法,从理论分析似乎应该比 AdaBoost 算法好,因为 AdaBoost 算法只是其特例,但实际情况究竟如何则需要用实际数据来验证。

本文选取数据挖掘或集成学习方法公用的 UCI 数据集中的 Ionosphere、Sonar、Pima、Wdbc、breastCancer 数据集,它们都只有两类标签,属性个数有多有少。具体测试方法是:训练集和测试集按照正(负)类样本同比例的方式随机多次划分,即按照比例从正类样本中随机选取一些数据为训练集,剩余的为测试集,按照同样的比例选取负类样本训练集和测试集。基于训练集训练出组合分类器,用其对测试集数据进行分类判别,统计分类错误个数,从而得到组合分类器的测试错误率。显然,仅仅这样一次的验证不能说明问题,为此,我们重复 20 次,每次训练集和测试集的选取都是随机的,只是比例不变。计算这 20 次得到的组合分类器的测试错误率的平均值和方差。错误率平均值可以反映算法总体效果情况,而错误率方差可以反映算法是否稳定,如果方差过大,则说明算法是不稳定的。

在具体实验时,训练集与测试集划分比例选为 6:4,即 60% 数据用于训练,40% 数据用于测试。每次训练多少个弱

分类器来进行加权组合也是重要和关键的。本文试验时,每次都训练出 30 个弱分类器,然后进行加权组合。显然,对于样本属性个数小于 30 的数据集,按照我们的方法也可以轻松得到 30 个弱分类器,对在同样情况下的 AdaBoost 算法,则当构造 30 个弱分类器时,有些弱分类器将是重复的,新增加这些重复的弱分类器,体现在加权组合时,其实质表现为组合系数发生变化。

采取同比例随机划分训练集和测试集是科学合理的。训练集和测试集划分的随机性可以反映算法的适应性,而按照同比例划分,正好反映了学习算法的适用条件。因为一般的学习算法都有一个潜在的假设,即训练集、测试集、整个目标空间的正类和负类应该有相同的分布和比例,也只有这样,通过不断降低训练错误率才能降低测试错误率,用组合分类器来分类目标时,其分类错误率才能认为等同于测试错误率。如果训练集和测试集中正负类样本的分布不一样,则所有的结论都需要用两者的分布函数来修正,这是集成学习算法中另一大类问题。

实验结果如表 1 和 2。

表 1 测试错误率平均值对比

数据集	AdaBoost	方法一	方法二	方法三
Ionosphere	0.161 4	0.124 6	0.131 8	0.135 0
Sonar	0.265 1	0.248 2	0.238 6	0.245 2
Pima	0.263 4	0.251 0	0.247 7	0.255 2
Wdbc	0.050 7	0.046 9	0.050 9	0.044 3
breastCancer	0.086 9	0.079 6	0.052 9	0.078 2

表 2 测试错误率方差值对比

数据集	AdaBoost	方法一	方法二	方法三
Ionosphere	0.025 5	0.023 6	0.032 2	0.023 8
Sonar	0.040 9	0.047 8	0.031 3	0.048 5
Pima	0.015 3	0.016 2	0.017 0	0.016 9
Wdbc	0.013 1	0.014 8	0.017 6	0.010 5
breastCancer	0.018 0	0.100 5	0.015 5	0.099 2

本文提出的三种属性组合方法在表中对应为“方法一”、“方法二”、“方法三”,表中“AdaBoost”对应单属性分类的传统 AdaBoost 算法。具体就是“方法一”按照式(4)构造组合系数,“方法二”按照式(7)构造组合系数,“方法三”按照式(8)构造组合系数。由于公式中包含样本权值,系数构造是递进的,即  $C_{dxT}$  中后一列值依赖于前面各列结果对样本权值的调整结果。四种方法是在完全相同的测试集和训练集划分上进行训练和测试的,20 次随机同比例划分后,对各自得到的测试错误率进行平均后对应表中的“平均错误率”。20 次随机同比例划分得到测试错误率后,计算这 20 次测试错误率的方差得到表 2 中的“错误率方差”。分析上述测试数据,除了在 Wdbc 数据集上,方法二的平均错误率 0.050 9 比 AdaBoost 的 0.050 7 稍差以外,其他情况下,提出的三种属性组合集成算法的平均测试错误率,均比 AdaBoost 算法低,特别是对 Ionosphere 数据集,本文算法比 AdaBoost 算法明显好,测试错误率由 AdaBoost 的 0.161 4 降低到方法一的 0.124 6、方法二的 0.131 8、方法三的 0.135 0,降低幅度约 20% 左右。而 Wdbc 数据集上方法二的例外,也仅仅为 0.4%,可以忽略。因此实验数据支撑了属性组合集成算法好于传统的 AdaBoost 算法的结论。

(下转第 475 页)

表3 AV-PSO与参考文献[8]性能比较

维数	函数	平均最优值				成功率/%				平均迭代次数			
		AV-PSO	PSO-IIW	BPSO	PSO-TVIW	AV-PSO	PSO-IIW	BPSO	PSO-TVIW	AV-PSO	PSO-IIW	BPSO	PSO-TVIW
80	$F_1$	0	0	13 600	146	100	100	24	8	139.6	282.3	468.9	1 925.4
	$F_3$	0	105	191	237	100	20	0	0	113.8	3 759.1	2 200.8	3 106.2
	$F_4$	0	0	152	102	100	100	22	16	117.8	286.1	1 249.5	2 204.3
120	$F_1$	0	0	30 400	323	100	100	8	2	129.6	328.4	608.0	2 420.6
	$F_3$	0	176	347	394	100	8	0	0	111.7	3 851.0	2 553.1	3 306.4
	$F_4$	0	0	269	258	100	100	6	4	119.5	346.0	1 805.0	2 371.8
160	$F_1$	0	0	41 200	38 100	100	100	2	0	137.2	370.1	962.7	2 935.5
	$F_3$	0	232	516	519	100	8	0	0	108.6	3 974.9	2 730.1	3 582.0
	$F_4$	0	0	444	373	100	100	2	0	113.5	376.7	1 964.0	2 964.7

## 5 结语

新改进的 AV-PSO 算法主要是:根据鸟群觅食的规律,利用粒子前两代的中心位置以寻找类似于当前全局最优位置的有用信息,并在速度更换式子新增  $c_3 r_3 (Avg_i - x_i(t))$  部分,表示了粒子综合利用前两次的信息对自己下一步行为的影响。经简单分析鸟群在最优区域觅食规律及试验仿真表明: $c_2$  取值较大  $c_3$  取值较小并且其范围为  $[0.1, 0.4]$ , 可以解决复杂高维优化问题。本文仅重点讨论该部分同社会部分  $c_2 r_2 (p_g(t) - x_i(t))$  之间的关系,并没有从整体上考虑各个部分间的关系,如  $c_1$ 、 $c_2$  和  $c_3$  等系数取值范围与相互间关系、收敛性分析及其在工程等其他方面的应用,将是下一步研究工作重点。

### 参考文献:

- [1] KENNEDY J, EBERHART R. Particle swarm optimization [C]// Proceedings of the 1995 IEEE International Conference on Neural Networks. Washington, DC: IEEE Computer Society, 1995: 1942–1948.
- [2] SEO J H, IM C H, HEO C G, *et al.* Multimodal function optimization based on particle swarm optimization [J]. IEEE Transactions on

Magnetics, 2006, 42(4): 1095–1098.

- [3] YI DA, GE XIU-YUN. An improved PSO-based ANN with simulated annealing technique [J]. Neurocomputing, 2005, 63(11): 527–533.
- [4] SOUSA T, SILVA A, NEVES A. A particle swarm data miner [C]// EPIA'03: Proceedings of the 11th Portuguese Conference on Artificial Intelligence, LNAI 2902. Berlin: Springer-Verlag, 2003: 43–53.
- [5] GAING Z L. A particle swarm optimization approach for optimum design of PID controller in AVR system [J]. IEEE Transactions on Energy Conversion, 2004, 19(2): 384–391.
- [6] FRANKEN N, ENGELBRECHT A P. Particle swarm optimization approaches to coevolve strategies for the iterated prisoner's dilemma [J]. IEEE Transactions on Evolutionary Computation, 2005, 9(6): 562–579.
- [7] SOUSA T, SILVA A, NEVES A. Particle swarm based data mining algorithms for classification tasks [J]. Parallel Computing, 2004, 30(5/6): 767–783.
- [8] 李剑, 王乘. 一种改进的自适应微粒群优化算法[J]. 华中科技大学学报: 自然科学版, 2008, 36(3): 118–121.

(上接第 468 页)

从表 2 中可见,本文提出的几种算法的测试错误率的方差也与 AdaBoost 算法的测试错误率方差差不多。除在数据集 breastCancer 中,方法一和方法三的错误率方差比 AdaBoost 算法的错误率方差大,即使这样,错误率方差也很小。因此,测试数据也表明,提出的方法与 AdaBoost 算法一样,是稳定的。

## 5 结语

针对样本由数字属性构成的分类问题,在 AdaBoost 算法流程基础之上,提出了一种基于样本属性线性组合的集成学习算法,仍然采用 AdaBoost 算法的样本权值调整策略和分类器组合策略,对带样本权值的属性进行组合来构造单属性分类器,通过样本权值的调整来形成集成学习算法。本文提出的一般性属性组合集成学习算法思路具有一定的通用性,可以在此基础上,采取不同的方法来构造组合系数,从而得到不同的集成学习算法。本文给出了三种具体的属性组合系数构造方法,并对其构成的集成学习算法的有效性进行了分析,通过 UCI 机器学习数据集中的数据对提出的方法进行了实验验证,结果表明,基于属性组合的集成学习算法思路是正确的,方法是有效的。

### 参考文献:

- [1] FREUND Y, SCHAPIRE R E. A decision-theoretic generalization of on-line learning and an application to boosting [J]. Journal of Com-

puter and System Sciences, 1997, 55(1): 119–139.

- [2] SCHAPIRE R E. The strength of weak learnability [J]. Machine Learning, 1990, 5(2): 197–227.
- [3] 付忠良. 关于 AdaBoost 有效性的分析[J]. 计算机研究与发展, 2008, 45(10): 1747–1755.
- [4] 付忠良. 分类器线性组合的有效性和最佳组合问题的研究[J]. 计算机研究与发展, 2009, 46(7): 1206–1216.
- [5] SCHAPIRE R E, SINGER Y. BoostTexter: A boosting-based system for text categorization [J]. Machine Learning, 2000, 39(2): 135–168.
- [6] VIOLA P, JONES M. Robust real-time face detection [J]. International Journal of Computer Vision, 2004, 57(2): 137–154.
- [7] 梁路宏, 艾海舟, 徐光祐, 等. 人脸检测研究综述[J]. 计算机学报, 2002, 25(5): 449–458.
- [8] 武勃, 黄畅, 艾海舟, 等. 基于连续 AdaBoost 算法的多视角人脸检测[J]. 计算机研究与发展, 2005, 42(9): 1612–1621.
- [9] 蒋焰, 丁晓青. 基于多步校正的改进 AdaBoost 算法[J]. 清华大学学报: 自然科学版, 2008, 48(10): 1609–1612.
- [10] 赵春晖, 张洪才, 陆朝霞. 基于 AdaBoost 的选择性样本权重更新算法[J]. 计算机应用研究, 2008, 25(10): 2943–2945.
- [11] 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.
- [12] 张健沛, 程丽丽, 杨静, 等. 基于全信息相关度的动态多分类器融合[J]. 计算机科学, 2008, 35(3): 188–190.