ISSN 1000-9825, CODEN RUXUEW
Journal of Software, Vol.21, No.8, August 2010, pp.1889−1897
doi: 10.3724/SP.J.1001.2010.03611

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel/Fax: +86-10-62562563

使用聚类来加速 AdaBoost 并实现噪声数据探测*

谢元澄[1,2+], 杨静宇[2]

[1](南京农业大学 信息科学技术学院,江苏 南京 210095)

[2](南京理工大学 计算机科学与技术学院,江苏 南京 210094)

Using Clustering to Speed up AdaBoost and Detecting Noisy Data

XIE Yuan-Cheng[1,2+], YANG Jing-Yu[2]

[1](College of Information Science and Technology, Nanjing Agricultural University, Nanjing 210095, China)

[2](School of Computer Science and Technology, Nanjing University of Science and Technology, Nanjing 210094, China)

+ Corresponding author: E-mail: xych.em@163.com

Xie YC, Yang JY. Using clustering to speed up AdaBoost and detecting noisy data. *Journal of Software*, 2010, 21(8):1889−1897. http://www.jos.org.cn/1000-9825/3611.htm

Abstract: According as the main factor deciding the performance of ensemble learning is the diversity of component learners, clustering technology is used to speed up AdaBoost in this paper. The performance of the new algorithm is very close to the AdaBoost in learning deferent noise levels data sets. The new algorithm can be used to detect and eliminate noisy data quickly, and could achieve rapid learning on data sets after eliminating noise. It overcomes the noise-sensitive shortcoming of AdaBoost. The general performance and efficiency of the new algorithm are much better than AdaBoost in processing data sets containing noise.

Key words: AdaBoost; clustering; component learner; BP neural network; speed up; noise detection

摘 要: 决定集成学习性能的主要因素是集成的个体学习器之间的差异性.使用聚类技术来加速 AdaBoost.在不同噪声水平环境下,新算法的性能都接近 AdaBoost.对 AdaBoost 噪声敏感问题提出了新的解决思路,使用该项技术可以实现快速的噪声探测和噪声剔除后的再学习,从而在对含噪声数据基进行处理时,在综合性能和效率上都明显优于 AdaBoost.

关键词: AdaBoost;聚类;个体学习器;BP 神经网络;加速;噪声检测

中图法分类号: TP181 **文献标识码**: A

机器学习的一个重要目标就是对新的测试样本尽可能地给出最精确的估计.构造一个高精度估计是一件困难的事情,但是产生数个只比随机猜测好的粗糙估计却相对容易[1].Kearns 和 Valiant 证明只要有足够的数据,弱学习算法就能通过集成的方式生成任意高精度的估计[2],这个证明就是 Boosting.1995 年,Freund 和 Schapire 提出了 AdaBoost(adaptive boosting)算法[3].这种算法不需要任何关于弱学习器性能的先验知识.实验结果表明,使用 AdaBoost 算法对个体学习器进行集成可以极大地提高弱学习器的泛化性能[4].AdaBoost 存在的一个问题是,虽然其具有优异的性能,但是存在训练时间非常漫长,以致其在实际问题中的应用受到了限制.特别是在大

* Supported by the National Natural Science Foundation of China under Grant No.60632050 (国家自然科学基金)
Received 2008-06-25; Revised 2008-12-10; Accepted 2009-03-30

样本多特征的情况,这一缺点更是明显.2001 年,Viola 等人[5]首先提出基于 Cascade 方法的 AdaBoost 人脸检测技术.该方法通过采用积分图实现了 Harr-like 特征的快速计算,并且每一个特征都可以快速学习为一个基分类器.这一技术从基分类器快速训练的角度实现了 AdaBoost 的快速训练.2007 年,郭志波[6]采用 Walsh 特征取代 Harr-like 特征,特征数目大幅度约减,从而进一步实现了 AdaBoost 人脸检测训练的加速.但我们也知道,所谓弱学习是相对而言,对于神经网络,决策树等学习器相比以上的单一特征分类器,其训练是非常耗时的.所以,Viola 等人的方法只能是 AdaBoost 针对人脸检测这一实际应用问题的专门的加速优化方法.

1995 年,Korgh[7]认为弱学习器之间的差异化决定了集成学习的泛化误差.2002 年,周志华的"many could be better than all"[8]揭示了通过选择部分学习器来组成集成更好.2001 年,Giacinto 和 Roli[9]通过分级聚类算法来选择精度高和差异度大的弱学习器来构建集成子集.2003 年,Bakker 和 Heskes[10]使用聚类技术来对相似弱学习器进行分组,并通过从每组中选出一个弱学习器代表作为个体学习器来进行集成,聚类技术在这里实现了在保留学习器差异化基础上的选择性集成.在该文中提出了使用模型输出取代模型的参数来度量模型间的差异性,实验结果显示,基于聚类的选择性集成可以进一步提升集成的性能.2004 年,李国正[11]采用最大信息压缩指数作为聚类的相似度标准来进行基于聚类的选择性集成,在性能上取得了与 GASEN[8]相当的精度和稳定性,并且其学习效率较后者平均提高了 25 倍.本文研究的内容就是考虑如何把集成技术应用到 AdaBoost 的个体学习器生成上,去除 AdaBoost 训练中的冗余过程,实现 AdaBoost 算法的训练加速,新算法称为基于聚类的 AdaBoost 算法(CLU_AdaBoost).

另外,实验结果也表明,AdaBoost 具有一些明显的缺陷,例如对噪声敏感[4].1998 年,Friedman 等人提出了"泛化 AdaBoost"[12],1999 年,Freund 提出了 Brown-Boost[13]来解决这一问题.2000 年,Virginia Wheway 证明可以使用 Boosting 技术来探测噪声[14].使用本文的算法可以实现对数据集的快速噪声探测和剔除,间接弥补了 AdaBoost 的这一缺陷,从而大幅度提高了 AdaBoost 和本文所提出新算法的性能.

1 AdaBoost 算法耗时分析

AdaBoost 是序列生成算法,每步迭代生成的个体学习器必须要考虑先前生成的个体学习器的性能.每一轮集成的个体分类器都通过以下方法诱导生成:第 t 轮训练集中的每个样本都安排权重 ($D^t=\{d_1^t,d_2^t,...,d_{N_{train}}^t\}$),使用一个固定的学习算法按照此权重进行训练,并生成相应的被集成个体学习器.每当一轮训练结束时,样本权重都要进行调整,正确(错误)分类的样本权重将减小(增加),采用这一方法,使得后继的学习器将偏向那些难以分类的样本.该过程直至满足以下条件之一时停止:得到 T 个个体学习器、学习器权重误差达到 0 或学习器权重误差大于 50%.

AdaBoost 算法流程简单而耗时,其原因主要在第 1 步.按照调整后的权重分布训练一个新的弱学习器进行集成,这个弱学习器必须满足对应于新的权重分布其预测误差小(最小权重误差).传统的训练方法一般分为两类:一类是使用可以按照样本权值分布训练的学习器作为基学习器;另一类是找到符合样本权重分布 D^t 的采样样本集,将在此样本集上训练得到的学习器作为基学习器[3],算法 1 列出了后者的伪代码.具体的做法是,通过重复采样训练出弱学习器集合 H,里面满足最小权重误差的那个学习器就被认为与算法中要求的分布是相符的.理论上,弱学习器集合 H 应该是无限大的,但实际的操作中其必定是有限集.例如,在基于 AdaBoost 人脸检测中生成的矩形特征有上万个,每一个特征都对应一个弱学习器,我们需要从庞大的弱学习器集合中挑选出最合适的一个;并且随着样本权重的改变,需要重新训练一批新的弱学习器以供挑选.AdaBoost 算法耗时的原因可以归结为以下几点:

(1) 如在人脸检测中随着样本数量、特征或数据的空间分布复杂度的增加,其单个弱学习器的训练时间也越长.

(2) 弱学习器集合 H 越大,挑选出的弱学习器就越符合分布条件,所以为获得一个较优的弱学习器,必须先训练出较大规模的 H,规模越大,则训练时间越长.

(3) 每轮结束,权重更新后上一轮生成的所有弱学习器被抛弃,需重新生成新的弱学习器集合以供挑选,即

需要更多的弱学习器.

算法 1. AdaBoost.

Input:训练集 $L=\{(x_1,y_1),\ldots,(x_N,y_N)\}$;$x_i\in X$;$y_i\in\{-1,1\}$.

Output:$C^*(x)=\arg\max_y \sum_{t:h_t(x)=y}^{T} \log(1/\beta_t)$.

初始化权重 $d_i^1=1/N$

For t=1,…,T:

1. 按照权重分布 D^t,调用 Weaklearn,从弱学习器集合 $H=\{h(x)\}$中选择 $h_t:X\to\{-1,1\}$,使其满足最小权重误差

$$h_t=\arg\min_{h_j\in H}\varepsilon_j=\sum_{i=1}^{N} d_i^t\,|\,h_j(x_i)-y_i\,|$$

2. ε_t=*WeightedError*(h_t,L,D^t)
3. if (ε_t>0.5) {删除 h_t,中断当前循环}
4. $\beta_t=\varepsilon_t/(1-\varepsilon_t)$
5. for j=1 to N {
6. if ($h_t(x_j)=y_j$) then $d_j^t=d_j^t\cdot\beta_t$}; 更新权重
7. $D^{t+1}=D^t\Big/\sum_{i=1}^{N} d_i^t$

2 使用聚类加速 AdaBoost

2.1 AdaBoost算法加速途径分析

针对第 1 节分析 AdaBoost 训练算法耗时的原因,分析相应的 AdaBoost 加速途径.

第 1 个原因.我们知道,一般的机器学习问题首先面对的就是样本问题,如人脸检测我们常常使用大量的样本来学习非人脸和人脸,其正、负样本越丰富,其训练效果也越好.可以通过精简样本的方式(在人脸检测中往往希望获得更多的非人脸样本,因此对特定的问题,该方法受到约束)、特征压缩的方式和寻找合适且更快速的弱学习算法(如人脸检测中基于矩形特征阈值分割的简单分类器)来加速 AdaBoost 训练.

第 2 个原因.弱学习器集合 H 显然是越大越好,实际应用中也必须要保持一定的规模才能满足算法精度,因此很难进一步压缩.

第 3 个原因.由于每一轮的权重发生改变,需要每一轮都重新训练出弱学习器集合 H 以供挑选出相应个体学习器.如果保留前一轮挑选后剩余的弱学习器集合,那后果将是后面的弱学习器集合快速增大,反而影响训练的速度;如果限制新的弱学习器集合并不再生成新的,那必将导致算法性能下降并可能提前终止.解决这一矛盾的途径应该是:在保留旧的弱学习器集合同时还要加入新的弱学习器,并且不能无限增大弱学习器集合的规模;或者说找到保留旧弱学习器和接受新弱学习器的折中方法,通过减少弱学习器的总数量来提速.

2.2 聚类作用分析

1995 年,Korgh[7]认为个体学习器之间的差异性决定了集成学习的泛化误差.当集成差异性较大的一组个体学习器时,我们就可以获得较好的集成效果.根据这一思想,可以这样来理解 AdaBoost:每一轮中的样本权重都发生了改变,得到的个体学习器都是针对当前训练中较难分类的一些样本训练而得来的,所以相邻几轮的个体学习器所侧重的分类样本对象是逐步变化的.也就是说,这一策略的使用,必然使得学习器之间在预测结果上存在差异性.针对这一分析可以判定,那些与当前个体学习器有差异的原有弱学习器可以保留到下一轮,而与当前轮个体学习器相似度较高的弱学习器需要删除.

寻求差异性的逆向思考可以理解为从弱学习器集合中挑选出同质化的弱学习器,并根据同质化进行分类.2003 年,Bakker 和 Heskes[10]基于聚类的选择性集成方法即是这一思想的体现,从分类后的弱学习器子集中挑选

出一个代表作为个体学习器加入集成.因此,考虑使用聚类方法来解决上面的弱学习器挑选问题.

聚类的首要问题就是如何度量聚类对象之间的相似程度(或距离),判断两个随机变量之间的相似度,我们通常考虑它们之间的线性相关性.2002 年,Mitra[15]引入了 Maximal information compression index(λ_2)来度量随机变量 x 和 y 的之间的相似性.满足以下等式:

$$2\lambda_2(x,y)=\mathrm{var}(x)+\mathrm{var}(y)-\sqrt{(\mathrm{var}(x)+\mathrm{var}(y))^2-4\,\mathrm{var}(x)\,\mathrm{var}(y)(1-\rho(x,y)^2)} \tag{1}$$

其中,$\rho(x,y)$为相关系数.该方法被证明在处理特征选择问题上优于相关系数法和最小二乘回归误差法.λ_2 值越小,说明两者之间的相似程度越高.从一定的角度上来说,AdaBoost 算法中的弱学习器可以被理解为是特征,因此,基于该相似度量对新旧弱学习器集合进行聚类选择应该是可行的.将与 AdaBoost 每轮中集成的个体学习器属于一类的所有弱学习器挑选出来进行剔除,不属一类的进行保留,并保持弱学习器删减与增加的平衡,这样就可以满足第 2.1 节第 3 个原因分析中提出的要求了.

2.3 基于聚类的AdaBoost加速算法

聚类算法是耗费时间的,每一轮都使用聚类显然不能起到加速的作用.注意到只需要挑出需要删除的弱学习器,至于保留下的其他弱学习器如何聚类并不重要,因此,只需要对与被挑选出来的个体学习器相似的弱学习器进行一类聚类.考虑到基于聚类的选择性集成通常采用每个聚类的中心作为其代表,因此逆向思考,可以将按 Boosting 方法挑选的弱分类器理解为某个聚类的中心代表,因此,根据相似度函数采用 K-NN 方法将与其最相似的若干个弱学习器一起选出,构成一个聚类,将这个聚类整体删除.使用 AdaBoost 算法从弱学习器集合中挑选出当前轮个体学习器;向弱学习器中添加若干新的弱学习器,以当前轮个体学习器为基准进行一类聚类;将删除该聚类后的弱分类器集合作为为下一轮的 AdaBoost 备选集.通过这一方法的反复迭代,有效地对 AdaBoost 算法训练过程中生成的中间弱学习器进行了选择和抛弃,控制了弱学习器集合的规模,从而实现了算法的提速.设计新算法——聚类 AdaBoost(clustering AdaBoost,简称 CLU_AdaBoost),见算法 2.

算法 2. Clustering AdaBoost.

Input:训练集 $L=\{(x_1,y_1),\ldots,(x_N,y_N)\}$;$x_i\in X$;$y_i\in\{-1,1\}$;每一轮弱学习器集合规模 $size(H^t)=M$;

调用 Weaklearn,训练出弱学习器集合 H^1;用于剔除的弱学习器数目 S.

Output:$C^*(x)=\arg\max_y \sum_{t:h_t(x)=y}^{T}\log(1/\beta_t)$.

初始化权重 $d_i^1=1/N$

For t=1,…,T:

1. 按照权重分布 D^t,调用 Weaklearn,从弱学习器集合 $H^t=\{h(x)\}$中选择 $h_t:X\to\{-1,1\}$,使其满足最小权重误差 $h_t=\arg\min_{h_j\in H}\varepsilon_j=\sum_{i=1}^{N}d_i^t\,|y_i-h_j(x_i)|$
2. ε_t=*WeightedError*(h_t,L,D^t)
3. if (ε_t>0.5) {删除 h_t,恢复初始权重分布,令 $D^t=D^1$,goto 1}
4. $\beta_t=\varepsilon_t/(1-\varepsilon_t)$
5. for j=1 to N {
6. if ($h_t(x_j)=y_j$) then $d_j^t=d_j^t\cdot\beta_t$ };更新权重
7. $D^{t+1}=D^t\Big/\sum_{i=1}^{N}d_i^t$
8. 调用 Weaklearn,训练出 S 个弱学习器加入到 H^t,构成 $\hat{H}^t$
9. 使用 K 近邻法从 $\hat{H}^t$ 选择出 S–1 个与 h_t 最相似的弱学习器,并将它们与 h_t 一起从 $\hat{H}^t$ 中删除,从而建立新的弱学习器集合 H^{t+1}

2.4 基于Clustering AdaBoost实现快速噪声检测和剔除

研究表明,Boosting 在处理那些较难分类的数据时总有保持边界平衡的趋势[14].为了使那些难以分类的对象被正确分类,一些先前分类正确的在随后的 Boosting 迭代中被错判,这导致随着 Boosting 迭代轮数的增加,每个对象最终被错分的比例趋向一致.这一特性可被用来对训练数据进行噪声探测.假设在 Boosting 算法的 t 轮迭代中,生成个体分类器产生假设 $h_t(x)$以及一个错误指示函数 $I_t(x_i)=I(h_t(x_i)\neq y_i)$.定义 c_t 为每轮迭代中生成个体分类器的投票权重,并使 $\sum_t c_t=1$,那么可以定义边缘函数[16]:

$$edge_i(t,c)=\sum_{j=1}^{t}c_j I_j(x_i) \tag{2}$$

随着 Boosting 迭代轮数的增加,该边缘函数的值趋于一致.噪声和野点数据难以进行分类,因此在边缘函数值上会与其他数据发生背离,因为错分它们会有较高的边缘值.绘制出边缘函数值图,根据训练样本在不同值区域内的分布可以设定分割阈值,根据该阈值来进行噪声剔除.

由于 CLU_AdaBoost 与 AdaBoost 相比具有明显的速度优势,因此使用该新算法可以快速实现噪声检测和剔除.对噪声剔除后的样本再次使用 CLU_AdaBoost 进行学习,可以大幅度地提高学习的精度和效率.

3 实 验

3.1 数据集设计

双螺旋数据集是模式识别领域公认的标准问题,因其难度而经常被用作检验模式识别算法性能的试金石.构造双螺旋线结构样本集如下:

定义随机数 $n_1,n_2\in[0,N]$,随机弧度 $\phi_1,\phi_2\in[0,2\pi]$,局部随机半径 $r_1,r_2\in[0,k]$,$k>0$,螺旋线每圈间距 D,则有:

正样本:

$$\begin{cases}x_p=(D\cdot n_1+1)\cdot\cos(2\pi\cdot n_1)+r_1\cos\phi_1\\ y_p=(D\cdot n_1+1)\cdot\sin(2\pi\cdot n_1)+r_1\sin\phi_1\end{cases} \tag{3}$$

负样本:

$$\begin{cases}x_n=(D\cdot n_2+1)\cdot\cos(2\pi\cdot n_2+\pi)+r_2\cos\phi_2\\ y_n=(D\cdot n_2+1)\cdot\sin(2\pi\cdot n_2+\pi)+r_2\sin\phi_2\end{cases} \tag{4}$$

调节 D 和 k 可以决定正、负样本是否有重叠,调节 N(螺旋线的圈数)可以决定样本集的复杂度.公式(3)、(4)的最后一项为加在双螺旋曲线数据集上的随机扰动.正、负样本交叉后可以生成含噪声数据.如图 1 所示为含 10%噪声的双螺旋数据集,那些处于浅色星号中的深色点和处在深色点中的浅色星号为噪声数据.

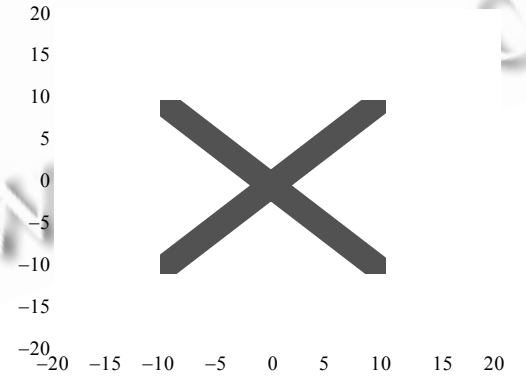

Fig.1 Two spiral data set include 10% noise

图 1 含 10%噪声的双螺旋数据集

3.2 算法Bagging,AdaBoost和CLU_AdaBoost在不同噪声水平上的对比实验

采用双螺旋数据 N 为 4,正、负样本都为 500 个,采用正、负样本交叉的方法分别生成含 1%,5%,10%和 20%噪声的样本集进行学习,另生成含有 0%噪声的数据作为验证集.分别使用 Bagging,AdaBoost 和 CLU_AdaBoost 算法进行对比测试,误差曲线如图 2 所示.其中,使用的弱学习器都是包含 25 个隐层节点的三层 BP 神经网络.从实验中可以看出:在不同噪声水平下,新算法都优于 Bagging 算法,且都非常接近 AdaBoost;而时间代价都只是 AdaBoost 的 1/10 左右.新算法充分体现了 Boosting 和聚类的优点.

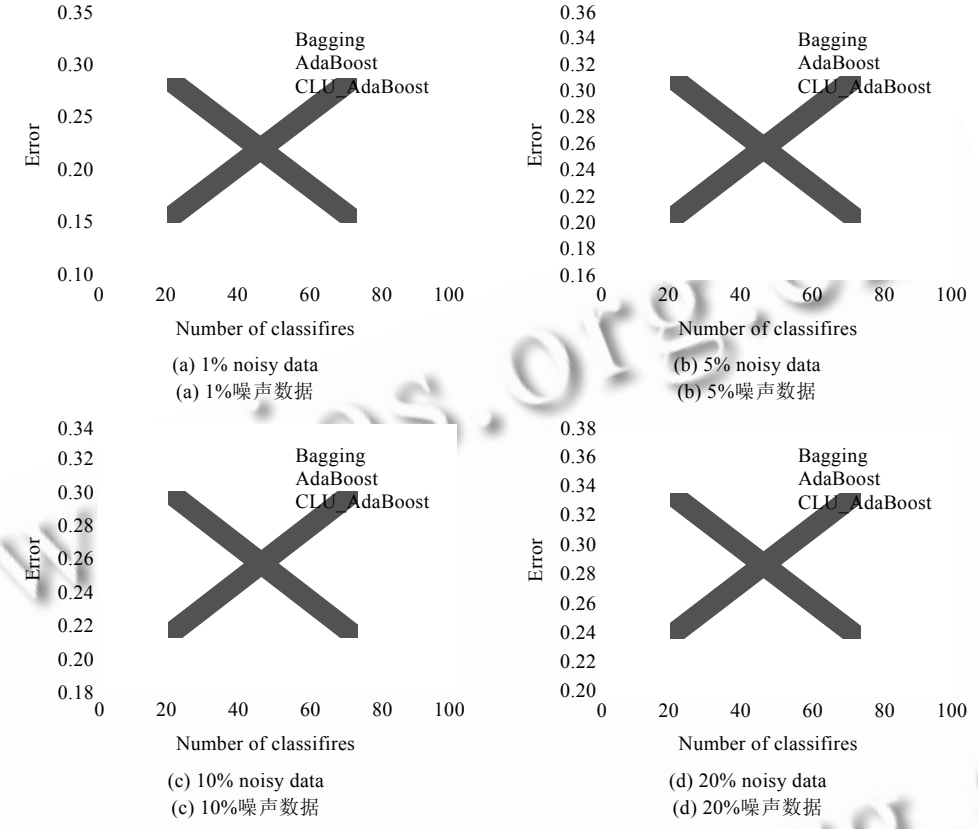

Fig.2 Comparison of Bagging, AdaBoost and CLU_AdaBoost on different noise levels test sets

图 2 不同噪声水平测试集上 Bagging,AdaBoost 和 CLU_AdaBoost 对比

3.3 基于Cluster AdaBoost实现快速噪声检测和剔除以提高算法性能

设计 N 为 1 的双螺旋训练数据 300 个,其中正、负样本各 150 个,并生成相应的 300 个测试数据;生成 UCI waveform[17]训练数据 600 个(为简化问题,将 waveform 修改为二分类问题,其中:1 类 waveform 为正类,2 类和 3 类 waveform 共同组成负类),生成对应的测试数据 400 个.对以上两个数据集的训练集分别按 5%,10%和 20%比例进行正、负样本交换生成噪声数据.根据以上方法,对每个数据集在每种噪声比例下随机生成 5 组对应训练集和测试集,分别采用 AdaBoost 和 CLU_AdaBoost 进行噪声剔除前测试、噪声检测和剔除以及噪声剔除后测试,将平均结果列于表 1.图 3(a)和图 3(b)分别为双螺旋数据和 waveform 数据在噪声比例为 20%状况下,其中一次实验中的边缘函数值分布图.图 3(a)中的噪声生成位置为横坐标 1~30 和 151~180,图 3(b)中的噪声生成位置为 0~120.可以明显看出,在这两个区域,数据的边缘函数值是明显偏高的,正是利用这一特点实现了数据集中的噪声检测.

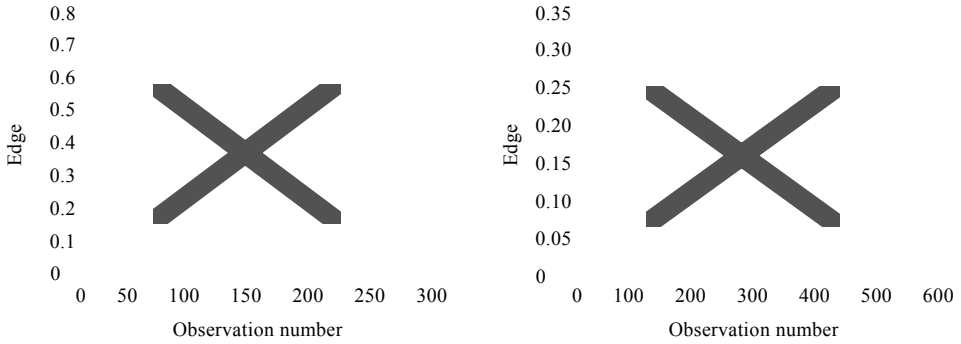

Fig.3 Edge function value figures of two spiral data set and UCI waveform data set, include 20% noise

图 3 含有 20%噪声的双螺旋数据集和 UCI waveform 数据集边缘函数值分布图

Table 1 Comparisons of Bagging and AdaBoost and CLU_AdaBoost before and after removing noise

表 1 Bagging,AdaBoost 和 CLU_AdaBoost 在剔除噪声前后的对比

Data sets	Algorithm	Before removing noise		After removing noise		Total time-consuming (s)
		Noise ratio (%)	Error rate	Noise ratio (%)	Error rate	
Two spiral (*M*=100, *S*=5, *T*=100)	AdaBoost	5	0.033 5	1.1	0.034 2	59 358.4
	CLU_AdaBoost		0.031 3	0.7	0.032 7	24 97.3
	AdaBoost	10	0.066 4	2.5	0.046 6	60 768.5
	CLU_AdaBoost		0.047 9	2.2	0.040 3	2 868.9
	AdaBoost	20	0.188 3	16.9	0.135 8	63 954.3
	CLU_AdaBoost		0.175 2	15.5	0.133 6	2 630.7
Waveform (*M*=300, *S*=7, *T*=100)	AdaBoost	5	0.135 2	2.8	0.122 5	51 530.1
	CLU_AdaBoost		0.122 4	2.8	0.122 6	5 464.8
	AdaBoost	10	0.130 1	4.8	0.127 4	66 329.4
	CLU_AdaBoost		0.127 6	3.7	0.124 9	6 752.9
	AdaBoost	20	0.151 2	10.7	0.131 1	78 629.2
	CLU_AdaBoost		0.156 3	10.4	0.131 3	7 675.4

文献[4]使用 Waveform 数据集通过交叉样本的方法生成噪声,并以此来说明 AdaBoost 方法对噪声的敏感问题.从表 1 中可以看出,噪声剔除前后,CLU_AdaBoost 算法的性能在大多数情况下都优于 AdaBoost,这说明这一算法对噪声的敏感性要低于 AdaBoost.AdaBoost 由于其近乎完美的学习能力对噪声也进行了充分的学习,因此对噪声具有敏感性,噪声为 20%时,它的错误率 0.188 3;噪声为 16.9%时,其错误率为 0.135 8.CLU_AdaBoost 也充分体现了 Boosting 的学习能力,并强化了其对非噪声数据的学习.与第 3.2 节实验相比,由于数据集的空间结构简单了,使得其只需要花费 AdaBoost 所用时间的大约 1/24 就可以获得相同的性能.因此可以看到,在噪声剔除前后,AdaBoost 和 CLU_AdaBoost 的性能都是非常接近的.在实验的大多数情况下,CLU_AdaBoost 优于 AdaBoost,这可能是由于算法中的聚类作用强迫相邻几轮集成的个体学习器在较难学习的样本上不连续保持一致(AdaBoost 可以),从而部分削弱了 CLU_AdaBoost 对噪声的学习能力,也即削弱了噪声的影响.当数据集空间分布较为简单时,这一特点便凸显出来,从而使其性能优于 AdaBoost.同时也可以看出,对噪声的检测能力前者也优于后者,其原因也在于此.

在不同比例噪声含量下的噪声检测实验中,使用 Boosting 算法进行噪声检测,双螺旋在低噪声含量下效果非常明显.定义噪声正确检出率为:检测算法判断为噪声且判断正确的样本数目与数据集中所有生成噪声样本数目的比值.定义噪声错误检出率为:检测算法判断为噪声但判断错误的样本数目与数据集中所有生成噪声样本数目的比值.对含有 5%噪声的双螺旋数据集,本文算法噪声正确检出率为 93.3%;将所有检测出被认为是噪声的样本全体(包含被正确和不正确判定的噪声样本)剔除后,噪声含量降为 1.1%.对含有 5%噪声的 waveform 数据,其噪声正确检出率为 53.33%;将所有检测出被认为是噪声的样本全体剔除后,噪声率降为 2.8%.这是由于根

据公式(3)、(4)生成的双螺旋数据是完全干净的,其噪声比例是真实的;而 waveform 数据集本身是在三角形波形上添加随机噪声得到的,不排除在数据生成的过程中已经有噪声存在.因此,waveform 数据集在噪声含量为 20%时仍然得到了 53.2%的噪声正确检出率,而相对于双螺旋数据集则为 48.0%.

4 结 论

本文以 AdaBoost 为基础,结合基于聚类的选择性集成方法的优点,设计出了基于聚类技术的 AdaBoost 加速算法(CLU_AdaBoost).该算法充分结合了二者的优点,使得在算法训练的时间耗费比 AdaBoost 低一个数量级的基础上获得与其接近的性能.同时,本文利用新算法速度的优势取代 AdaBoost 算法,实现了快速的噪声探测和剔除,从而在快速噪声剔除基础上实现了新算法对 AdaBoost 的完全胜出,而所用耗费时间仍比 AdaBoost 低一个数量级.对于 CLU_AdaBoost 中的参数,一般情况下,M 与 S–1 的比值在 15~50 倍之间都可以获得较好的效果.表 1 中,双螺旋数据为 25 倍,waveform 为 50 倍.值得注意的是,增大 M 值往往可以进一步提升算法的学习性能,但却不一定获得更优的测试性能.

我们还注意到,在噪声剔除过程中,边缘函数值的阈值选取是较为困难的.当阈值选取过小时,往往在删除噪声的同时还删除大量正确样本;当阈值选取过大时,删除的噪声又较少,这影响了新算法性能提高的幅度.实验中,一般按噪声比例来确定删除噪声的阈值,或者按原始样本总数的 1/20~1/5 来确定边缘函数阈值,都取得了较好的效果.下一步的工作主要研究如何自适应地选择这个阈值,以获得更好的算法性能提升.

References:

[1] Schapire RE. The boosting approach to machine learning: An overview. In: Denison DD, Hansen MH, Holmes C, Mallick B, Yu B, eds. Proc. of the Mathematical Sciences Research Institute (MSRI) Workshop on Nonlinear Estimation and Classification. Berlin, Heidelberg: Springer-Verlag, 2002. 149−172.

[2] Kearns M, Valiant L. Cryptographic limitations on learning Boolean formulae and finite automata. Journal of the ACM, 1994,4l(1): 67−95. [doi: 10.1145/174644.174647]

[3] Freund Y, Schapire RE. A decision-theoretic generalization of on-line learning and an application to Boosting. Journal of Computer and System Sciences, 1997,55(1):11−139.

[4] Dietterich TG. An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting, and randomization. Machine Learning, 1999,40(2):139−157. [doi: 10.1023/A:1007607513941]

[5] Viola P, Jones M. Rapid object detection using a boosted cascade of simple features. In: Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition. New York: IEEE Press, 2001. 511−518.

[6] Guo ZB. Research on the algorithm of fast face detection and feature extraction [Ph.D. Thesis]. Nanjing: Nanjing University of Science and Technology, 2007 (in Chinese with English abstract).

[7] Krogh A, Vedelsby J. Neural network ensembles, cross validation, and active learning. In: Tesauro G, Touretzky DS, Leen TK, eds. Advances in Neural Information Processing Systems 7. Cambridge: MIT Press, 1995. 231−238.

[8] Zhou ZH, Wu JX, Tang W. Ensembling neural networks: Many could be better than all. Artificial Intelligence, 2002,137(1-2): 239−263. [doi: 10.1016/S0004-3702(02)00190-X]

[9] Giacinto G, Roli F. An approach to the automatic design of multiple classifier systems. Pattern Recognition Letters, 2001,22(1): 25−33. [doi: 10.1016/S0167-8655(00)00096-9]

[10] Bakker B, Heskes T. Clustering ensembles of neural network models. Neural Networks, 2003,16(2):261−269. [doi: 10.1016/S0893-6080(02)00187-9]

[11] Li GZ, Yang J, Kong AS, Chen NY. Clustering algorithm based on selective ensemble. Journal of Fudan University (Natural Science), 2004,43(5):689−691 (in Chinese with English abstract).

[12] Friedman J, Hastie T, Tibshirani R. Additive logistic regression: A statistical view of boost. Annals of Statistics, 2000,28(2): 337−407. [doi: 10.1214/aos/1016218223]

[13] Freund Y. An adaptive version of the boost by majority algorithm. Machine Learning, 2001,43(3):293−318. [doi: 10.1023/A:1010852229904]

[14] Wheway V. Using boosting to detect noisy data. In: Kowalczyk R, ed. Advance in Artificial intelligence, PRICAI 2000 Workshop Reader. LNAI 2112, Berlin, Heidelberg: Springer-Verlag, 2001. 123−130.

[15] Mitra P, Murthy CA, Pal SK. Unsupervised feature selection using feature similarity. IEEE Trans. on Pattern Analysis and Machine Intelligence, 2002,24(3):301−312. [doi: 10.1109/34.990133]

[16] Breiman L. Arcing the edge. Technical Report, 486, Berkeley: University of California, 1997.

[17] Asuncion A, Newman D. UCI machine learning repository. 2007. http://archive.ics.uci.edu/ml/datasets.html

附中文参考文献:

[6] 郭志波.人脸快速检测和特征抽取方法的研究[博士学位论文].南京:南京理工大学,2007.

[11] 李国正,杨杰,孔安生,陈念贻.基于聚类算法的选择性神经网络集成.复旦学报(自然科学版),2004,43(5):194−196.

谢元澄(1976－),男,江苏无锡人,博士,主要研究领域为模式识别,图像处理,视频压缩.

杨静宇(1941－),男,教授,博士生导师,CCF 高级会员,主要研究领域为模式识别,智能机器人,数据融合.