·人工智能及识别技术· 文章编号:1000-3428(2017)05-0163-06 文献标志码:A 中图分类号:TP301.6

基于噪声自检测的分段非线性组合 Adaboost 改进算法

张 才,陈优广

(华东师范大学 计算中心,上海 200062)

摘 要: 针对传统 Adaboost 算法对有噪声样本敏感的问题以及线性相加基分类器的不合理性,提出一种噪声自检测的分段非线性组合 Adaboost 算法(NDK Adaboost)。NDK Adaboost 利用传统 Adaboost 算法的训练误差率随迭代次数呈指数下降的特点直接构造检测噪声模型来识别噪声,并且在预测阶段将预测样本映射到训练样本的相对位置,根据其邻近的样本分布决定基分类器的权重,从而使算法在不同的样本分布中具有较高的分类准确率。实验结果表明,与传统 Adaboost 算法以及 Adaboost 相关的改进算法相比,该算法具有较高的分类准确率。

关键词: 噪声检测;传统 Adaboost;分段;基分类器;邻近样本;权重

中文引用格式: 张 才,陈优广. 基于噪声自检测的分段非线性组合 Adaboost 改进算法[J]. 计算机工程,2017,43(5):163-168,173.

英文引用格式: Zhang Cai,Chen Youguang. Improved Piecewise Nonlinear Combinatorial Adaboost Algorithm Based on Noise Self-detection[J]. Computer Engineering,2017,43(5):163-168,173.

Improved Piecewise Nonlinear Combinatorial Adaboost Algorithm Based on Noise Self-detection

ZHANG Cai,CHEN Youguang

(Computing Center,East China Normal University,Shanghai 200062,China)

【Abstract】 As traditional Adaboost algorithm is sensitive to noisy sample and the linear combination of base classifiers is irrational,a piecewise nonlinear Adaboost algorithm based on noise self-detection called NDK Adaboost is proposed. NDK Adaboost,drawing on traditional Adaboost algorithm whose error rate in training set decreases with iteration times exponentially,establishes directly a noise detection model to recognize noise,and maps the prediction samples to the relative positions of the training set. According to the neighbor samples' distribution,it determines the weight of the base classifier. A higher classification accuracy rate of the algorithm can be drawn out among the different sample distribution. Experimental results show that,compared with the traditional Adaboost algorithm and the related improved algorithms,NDK Adaboost has a higher classification accuracy rate.

【Key words】 noise detection; traditional Adaboost; piecewise; base classifier; neighbor sample; weight

DOI: 10.3969/j.issn.1000-3428.2017.05.026

0 概述

在机器学习和数据挖掘领域里,分类是一个关键问题,有各种分类算法被提出,如决策树、k 邻近算法、贝叶斯网络、支持向量机等。后来出现了比较有代表性的集成提升算法 Adaboost,它是在 PAC[1] 框架下提出的一种算法,并且在机器学习中得到了广泛运用[2]。PAC 框架中提出了弱可学习和强可学习的概念,并且证明了弱可学习可以转化为强可学习。文献[3]利用这个特点提出 Adaboost 算法,并从理论和实践中取得了良好的分类效果[4-5]。特别在人脸的检测、文本分类中取得了很大成功[6]。Adaboost 最有吸引力的特点是通过反复迭代,训练误差以指数速率下降,训练集上的错误率可以不断逼近零[7]。但在有噪声的情况下会出现过拟合现象[8],这是因为 Adaboost 对有噪声很敏感,由于不断的迭代,噪声样本权重会不断增加,最终导致因权重过大而使分类器的准确率降低。目前提出了一些方法来减弱噪声的影响,一类方法通过修改损失函数[9],使噪声点在不断迭代过程中权重下降,著名的改进算法就是 LogitBoost。它将损失函数改成对数似然函数[10]。另一类方法直接设置阈值,限制样本

作者简介:张 才(1991—),男,硕士研究生,主研方向为数据挖掘、图形处理;陈优广(通信作者),副教授、博士。

收稿日期:2016-05-18 修回日期:2016-06-20 E-mail:doWolf123@126.com

权重的增长[11]。这 2 类方法虽然在一定程度上取得了效果,但是也限制了正常训练样本的权重。近年来有各种去噪方法,但对于不平衡数据效果往往不佳[12]。本文承接这种思想,利用传统 Adaboost 算法特性定义不同种类的噪声,并且做相应处理。另外对于一个测试样本,以往的很多算法都是直接用训练得到的分类器进行分类[13-16]。文献[17]提出利用基分类器对测试样本的概率作为基分类器的系数,但是它没有考虑待测样本所在训练样本的分布情况。本文进一步扩充并修改它的方法,先分析待测样本映射到训练样本的位置,再由其 K 邻近样本计算基分类器的系数。

除了以上介绍的方法外,还有研究者将 Adaboost 和 bagging[18]算法结合起来,形成 mutiboosting[19],后文将这些算法和本文提出的 NDK-Adaboost 算法进行实验比较。

1 Adaboost 算法

Adaboost 中对于基分类器的训练,可以使用一些常见的学习算法,例如决策树等。

1.1 Adaboost 算法流程

输入 1)给定含有 N 个样本的训练集 $S=\{(\boldsymbol{x}_1y_1),(\boldsymbol{x}_2y_2),\cdots,(\boldsymbol{x}_Ny_N)\}$,其中,$\boldsymbol{x}_i$ 为属性向量;$y_i\in\{+1,-1\}$;2)任意一种学习算法

输出 最终分类器:$G(x)$

训练方法:

步骤 1 初始化训练集的权重,默认使用均匀分布,即每一个样本的权重相等,表示为:$D_1=(w_{11},w_{12},\cdots,w_{1i},\cdots w_{1N})$,$w_{1i}=1/N$,$i=1,2,\cdots,N$。

步骤 2 记 T 为算法训练迭代的总次数,t 为第 t 次迭代,D_t 为第 t 次迭代的样本权重分布,对于 $t=1,2,\cdots,T$,循环以下操作:

1)使用带有权重 D_t 的训练集进行学习,学习算法为输入的学习算法,得到基分类器 $h_t(x)\in\{-1,1\}$。

2)计算该分类前在训练数据集上的分类误差率,记分类器 $h_t(x)$ 所对应的误差率为 e_t,有:

$$e_t=P(h_t(x_i)\neq y_i)=\sum_{i=1}^{N}w_{ti}I(h_t(x_i)\neq y_i)$$

3)根据该分类器的误差率,计算其权重 $\alpha_t=1/2\times\ln(1-e_t/e_t)$,该权重即作为最终分类器的系数,体现该分类器在总分类器的比重。

4)更新训练样本权重,$D_{t+1}=((w_{t+1,1},w_{t+1,2},\cdots,w_{t+1,i},\cdots,w_{t+1,N}))$,其中:

$$w_{t+1,i}=w_{t,i}\times e^{-\alpha_t y_i h_t(x_i)}/Z_t$$

$$Z_t=\sum_{i=1}^{N}w_{t.i}\times e^{-\alpha_t y_i h_t(x_i)}$$

Z_t 为规范因子,目的是保证每一轮的样本权重之和为 1。

步骤 3 通过上面 T 次的迭代,产生了 T 个分类器和相应的系数,将其得到的分类器进行线性组合得到最终分类器:$G(x)=sign(\sum_{t=1}^{T}\alpha_t\times h_t(x))$。

1.2 Adaboost 优缺点分析

Adaboost 算法在训练集上通过迭代,会不断降低在训练集上的误差率,并且下文的定理给出了其最终误差下界[5]:

定理 1(Adaboost 训练误差界)

$$1/N\times\sum_{i=1}^{N}I(G(x_i)\neq y_i)\leqslant\prod_t Z_t$$

其中,$G(x)$ 和 Z 均为上文提到的最终分类器和规范化因子。如果是二分类问题,进一步有下面的结论:

定理 2(二分类问题的训练误差界[20])

$$\prod_{t=1}^{T}Z_t=\prod_{t=1}^{T}\sqrt{(1-4(1/2-e_t)^2)}\leqslant e^{-2\sum_{t=1}^{T}(1/2-e_m)}$$

进一步可得到推论:

推论 1(指数下降推论)

$$1/N\times\sum_{i=1}^{N}I(G(x_i)\neq y_i)\leqslant\prod_t Z_t\leqslant e^{-2T\gamma^2}$$

其中,$\gamma=\min\{(1/2-e_1),(1/2-e_2),\cdots,(1/2-e_T)\}$;$T$ 为迭代次数。

由此可直接看出,Adaboost 的训练误差率随着迭代的次数呈指数下降,从而 Adaboost 可以快速地训练数据。随着研究的深入,当在训练集上误差率趋向于零时,在测试集上也趋向于一个稳定值,但是不是 0。这样就产生一个想法,既然随着训练次数已经无法明显提升测试准确率,那是否还有什么办法。本文猜想要提升准确率,就要从降低噪声和产生基分类器系数两方面入手。下文将围绕这两点展开。

2 改进的 Adaboost 算法

2.1 基本概念和名词定义

对于噪声有很多定义,从训练集中任意取出的一个样本,记为 $p(\boldsymbol{x},\boldsymbol{y})$,$\boldsymbol{x}$ 为属性向量,$\boldsymbol{y}$ 为标记量。由于在现实生活中因技术、人工错误操作等原因,都会出现样本实例中的 $\boldsymbol{x}$,或者 $\boldsymbol{y}$ 与真实值之间出现偏差。例如,是否能够办理信用卡有很多的条件,比如年龄、月收入、信用度等,表 1 给出了 p1,p2 2 个抽样实例。当 $\boldsymbol{y}$ 出现误标记(如:是否发卡由“是”变成了“否”),那么这就产生了噪声,在此称为第 1 类噪声。

当输入错误,也就是当 $\boldsymbol{x}$ 出现了错误(如:月收入少了一个 0,性别写反了等),在此称为第 2 类噪声。

表 1　2 个信用卡用户样本

变量	条件	p1	p2
	年龄	25	17
	性别	男	女
$\boldsymbol{x}$	月收入	20 000	1 000
	信用度	良好	一般
	工作年限	2 年	0 年
$\boldsymbol{y}$	是否发卡	是	否

在现实生活中,特别是大量抽样的数据,很可能会有类似的噪声,但不同的噪声对分类器的影响程度是不一样的。比如,在上文的例子中,将性别写反对于最终是否发卡影响不大,但是如果工作年限,或者月收入多加了一个零,那么对是否发卡影响很大,像这种样本直观看上去就很“异常”。用敏感的 Adaboost 来训练它,会对分类的准确率造成很大影响。为了区别不同噪声的影响,直接利用 Adaboost 算法特性划分不同的噪声。在 Adaboost 算法中,最终得到的分类器为:$G(\boldsymbol{x}) = sign(\sum_{t=1}^{T}\alpha_t \times h_t(\boldsymbol{x}))$,观察到,$-\sum_{t=1}^{T}\alpha_t \leqslant \sum_{t=1}^{T}\alpha_t \times h_t(\boldsymbol{x}) \leqslant \sum_{t=1}^{T}\alpha_t$,为了将每个基分类器的系数进行规范化,在这里加入参数 $\alpha = \sum_{t=1}^{T}\alpha_t$,使基分类器的系数之和为 1。这样对于一个样本点 $P(\boldsymbol{x}_{样本},\boldsymbol{y}_{样本})$,计算 $\boldsymbol{y}_{样本} \times (\sum_{t=1}^{T}\frac{\alpha_t}{\alpha} \times h_t(\boldsymbol{x}_{样本}))$,它的值都在[-1,1]区间内。由于进过训练,Adaboost 在训练集上误差率已经很低了,因此越接近于 -1,说明它是噪声的可能性越大,越接近于 1 说明它是噪声的可能性越小。前文叙述到,Adaboost 的训练误差率是指数下降,并且一般在测试集上都能显示出较高的分类准确率。所以,当将训练后的分类器重新作用于训练集后,计算 $\boldsymbol{y}_{样本} \times (\sum_{t=1}^{T}\frac{\alpha_t}{\alpha} \times h_t(\boldsymbol{x}_{样本}))$,只需要它小于零,就很有可能是噪声。对于[0,1]期间内,则给定一个阈值,在小于这个阈值的区间,可以认为它是噪声的可能性为百分之五十,大于阈值的点,认为它是噪声的可能性为零(正常样本),下面给出上述描述的数学定义。

定义 1　Adaboost 框架下的噪声样本分类

对于给定训练样本集 S 中的一个样本,记为 $P(\boldsymbol{x}_{样本},\boldsymbol{y}_{样本})$,当给定一个值 λ,其中,$0 \leqslant \lambda \leqslant 1$,计算 $m = \boldsymbol{y}_{样本} \times (\sum_{t=1}^{T}\frac{\alpha_t}{\alpha} \times h_t(\boldsymbol{x}_{样本}))$,若 $m \in [-1,0)$,则称 $P(\boldsymbol{x}_{样本},\boldsymbol{y}_{样本})$ 为强噪声样本(点),若 $m \in [0,\lambda]$,则称为模糊地带样本(点),若 $m \in (\lambda,1]$,则称为正常样本(点)。

图 1 给出了上述定义的形象化表示,图中虚线之内为模糊地带样本,虚线之外为正常样本,其中含有 2 个强噪声样本。

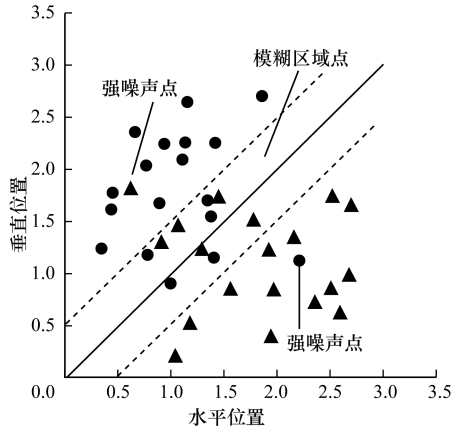

图 1　噪声分布示意图

根据上文的分析,本文算法将训练集中的样本,通过传统 Adaboost 算法本身得到 3 种不同种类的样本,从而检测噪声,并作为新算法 NDK-Adaboost 的数据预处理阶段。对于强噪声样本,这类样本是噪声的可能性极大,这类点可以直接从训练集中移除,对于正常样本,这类样本是噪声的可能性很小,一定要作为接下来的训练集样本,现在的问题在于最后的模糊地带样本,因为这类样本没有明显趋向于噪声的特性。对于这种情况,假设总体分布为 D,并假设其中的强噪声样本已经去除,将它拆分为 $D1$ 分布和 $D2$ 分布,$D1$ 影响着正常样本,$D2$ 分布影响着模糊地带样本。在总体样本中用传统的 Adaboost 算法得到最终分类器后,对于待测样本,当测试样本属于 $D1$ 分布时,分类准确率就会比较高,因为 $D1$ 包含大量正常样本;而当属于 $D2$ 分布的样本时,分类准确率相对不会太高,因为 $D2$ 包含大量模糊地带样本。因此,对于一个待测样本,首先要确定它是属于 $D1$ 分布还是 $D2$ 分布。对于一个待测样本,它所属于哪一个分布一般是难以知道的,但是通常情况下它邻近的样本可以准确率很高地反映它所属的分布。例如与它邻近的 10 个样本都属于 $D1$ 分布,那么本文推测该样本也属于 $D1$ 分布。本文算法先判断待测样本的所属分布,采取的判断方法为直接判断待测样本的 K 邻近样本是否全部属于 $D1$ 分布,如果是,那么待测样本判为 $D1$ 分布样本,否则 $D2$,后面的实验也表明,用这种判定方法能很好地提升分类准确率。通过判定结果决定选用传统 Adaboost 基分类器系数计算方式还是新的基分类器系数计算方式。对于新系数的计算方式先进行如下分析:一个样本属于模糊地带样本,它邻近的点的特性往往反而能更

好地反映出其特性。所以,在算法中将基分类器的系数改为由待测样本的 K 邻近样本和原来系数两方面决定,这是因为,对于传统 Adaboost 计算出来的系数,虽然在这样的样本下分类准确率不太高,但是它在一定程度上反映了数据情况,所以应当保留下来;另一方面,本文算法用每一个基分类器来直接计算 K 个邻近训练样本的分类误差率,进而得到的系数(计算系数的具体方法在下文算法中给出)能从另一个角度反映待测样本的相似程度,这一点应当引入,本文算法最终采取这 2 种系数的几何平均数作为 $D2$ 分布样本的最终基分类器系数。这样,对于 $D2$ 分布的点就不会因为权重分布问题而导致其分类准确率降低,并且还保证了基分类器的可用。下面将叙述 NDK-Adaboost 算法流程。

2.2 NDK-Adaboost 算法流程

2.2.1 NDK-Adaboost 的主算法

输入 S:训练数据集(一种训练基分类器的算法)

T:总迭代次数

λ:判断模糊地带样本的阈值

K:K 邻近的参数

输出 最终分类器

训练方法:

步骤 1 将 S 用传统 Adaboost 进行训练得到最终分类器 $G_1(\boldsymbol{x})=sign(\sum_{t=1}^{T}\alpha_t\times h_t(\boldsymbol{x}))$。

步骤 2 对于训练集中的每一个样本计算 $m=\boldsymbol{y}_{样本}\times(\sum_{t=1}^{T}\frac{\alpha_t}{\alpha}\times h_t(\boldsymbol{x}_{样本}))$。

步骤 3 根据上一步所得的结果,将 $m\in[-1,0)$的训练样本集合记为 S^1,将 $m\in[0,\lambda]$的训练样本集合记为 S^2。

步骤 4 对样本$(S-S^1)$(代表在最初样本中使用除去 S^1 的样本)重新使用 Adaboost 算法得到新的分类器:$G_2(x)=sign(\sum_{t=1}^{T}\beta_t\times h_t(\boldsymbol{x}))$。

//预测阶段

给定一个预测样本 $p_{预测}(\boldsymbol{x}_{预测},\boldsymbol{y}_{待预测})$,其中,$\boldsymbol{x}_{预测}$已知,$\boldsymbol{y}_{待预测}$为待预测的值。

步骤 5 计算 $\boldsymbol{x}_{预测}$与样本集$(S-S^1)$$K$ 个最相似的样本。

若(这 K 个样本中有样本属于 S^2),那么用 γ_t 计算算法来计算每个基分类器的系数,用 γ_t 表示,得到 $G_3(\boldsymbol{x})=sign(\sum_{t=1}^{T}\gamma_t\times h_t(\boldsymbol{x}))$,并用它对该样本进行预测。

若(这 K 个样本中没有样本属于 S^2),那么直接使用 $G_2(\boldsymbol{x})=sign(\sum_{t=1}^{T}\beta_t\times h_t(\boldsymbol{x}))$对该样本进行预测,即最终分类器可以分段表示为:

$$G(\boldsymbol{x})=\begin{cases}G2(\boldsymbol{x})//与\ X\ 最相近的\ K\ 个样本中没有存在于\ S^2\ 中\\ G3(\boldsymbol{x})//与\ X\ 最相近的\ K\ 个样本中有存在于\ S^2\ 中\end{cases}$$

2.2.2 γ_t 计算算法

输入 K 个最相近的样本

基分类器序列 $h_1(\boldsymbol{x}),h_2(\boldsymbol{x}),\cdots,h_T(\boldsymbol{x})$

NDK-Adaboost 的主算法中步骤 4 计算出来的基分类器系数系列 $\beta_1,\beta_2,\cdots,\beta_T$

输出 各个基分类器的系数 $\gamma_1,\gamma_2,\cdots,\gamma_T$

计算方法:

步骤 1 对于 $t=1,2,\cdots,T$,循环以下操作:

1)计算 $h_t(\boldsymbol{x})$在这 K 个样本中的误差率 $e_t=\frac{1}{K}\sum_{i=1}^{K}I(h_t(x_i)\neq y_i)$,并计算对应系数,用 A_t 表示:

$$A_t=\frac{1}{2}\ln((1-e_t)/e_t)=\frac{1}{2}\ln\left(\frac{1}{\frac{1}{K}\sum_{i=1}^{K}I(h_t(x_i)\neq y_i)}-1\right)$$

2)计算 $\gamma_t=\sqrt{(A_t\times\beta_t)}$。

步骤 2 得到的 $\gamma_1,\gamma_2,\cdots,\gamma_T$ 即为所求。

3 实验结果与分析

3.1 实验方法

本文提出的算法主要用在二分类问题上,所以本文的实验数据从著名的 UCI[15] 数据库中选出大量不同的二分类数据进行实验。UCI 数据库是加州大学欧文分校提出的用于数据分析、数据挖掘和机器学习的数据库,广泛用于分类和聚类问题,是行业一个比较常用的标准测试数据集。从中随机选取了 13 个数据集,这些数据集的特征如表 2 所示。

表 2 各种数据集信息

编号	数据集名称	属性数	样本总数
1	breast-cancer	10	286
2	breast-w	10	699
3	colic	23	368
4	credit-a	16	690
5	credit-g	21	1 000
6	diabetes	9	768
7	heart-statlog	14	270
8	hepatitis	20	155
9	kr-vs-kp	37	3 196
10	labor	17	57
11	sick	30	3 772
12	sonar	61	208
13	vote	17	435

将本文提出的 NDK-Adaboost 算法与传统的 Adaboost 算法在基本相同的条件下进行比较,同时

对于上文提到的其他著名的 Adaboost 改进算法、LogitBost 和 MutiboostAB 和参考文献[17]中的 Adaboost 改进算法也一并参与比较,最终得出结论。为了使实验结果具有可靠性,采取了十折交叉验证的方式进行实验,为了测试在有噪声情况下的分类效果,之后按一定比例加入噪声到训练集中。编程语言采用 Java,其他算法的结果直接采用 weka[16] 提供的开源代码进行操作。对于本文算法和其他比较算法的一些参数,若有相同类型和作用的参数,如迭代次数,都一并采用相同的值(迭代次数默认都为 10),其余参数都采用 weka 默认参数。对于 NDK-Adaboost,默认 λ 为 0.3,默认 K 为样本总数的 5%,这 2 个参数目前是靠经验来设定的。但是基本的原则是,λ 尽量小于 0.5。K 按照总样本数量的比例进行计算,一般尽可能小,但又要能足够统计出范围内的样本相似度。

3.2　NDK-Adaboost 与其他算法的比较结果

表 3 是 NDK-Adaboost 与其他算法的比较结果。

表 3　5 种算法的预测准确率　%

编号	数据集名称	NDK-Adaboost 算法	Adaboost 算法	LogitBost 算法	MutiboostAB 算法	WBIT 算法
1	breast-cancer	**74.475 5**	69.230 8	73.426 6	72.727 3	68.181 8
2	breast-w	95.565 1	94.849 8	**95.708 2**	94.992 8	95.094 3
3	colic	82.065 2	81.250 0	81.521 7	81.521 7	**83.967 4**
4	credit-a	**85.507 2**	84.637 7	84.927 5	**85.507 2**	84.637 7
5	credit-g	**72.500 0**	69.500 0	70.800 0	70.600 0	71.600 0
6	diabetes	72.916 7	**74.349 0**	74.088 5	72.526 0	72.395 8
7	heart-statlog	80.000 0	80.000 0	82.222 2	**82.592 6**	79.259 3
8	hepatitis	83.225 8	82.580 6	81.935 5	82.580 6	**86.451 6**
9	kr-vs-kp	**99.562 0**	93.836 0	93.804 8	99.342 9	99.499 4
10	labor	87.719 3	87.719 3	**89.473 7**	78.947 4	87.719 3
11	sick	**98.807 0**	97.189 8	97.905 6	96.553 6	98.204 7
12	sonar	**82.211 5**	71.634 6	79.326 9	74.519 2	78.846 2
13	vote	**96.092 0**	95.402 3	95.402 3	95.632 2	95.632 2

在表 3 中,对于每一个数据集,5 种算法中准确率最高的用粗体标出,准确率最低的用下划线标出(当 2 个并列相等时,2 个同时标出),从中可以看出,NDK-Adaboost 在超过一半的数据集(7 个数据集)上取得了最高准确率,并且始终没有沦为最低准确率的(其余 4 种都有成为最低准确率的情况)。

当在 NDK-Adaboost 不是最高准确率的数据集上,它与最高准确率相比,始终没有超过 4 个百分比(差距最大的为 3.2258 个百分比),综合起来都比其他算法有明显优势。

3.3　不同算法在各种噪声比例条件下的准确率

根据前文的叙述,该算法在有噪声的情况下,能够保持良好的分类准确率。为了验证,从上述数据集中挑选出样本个数最多的 3 个数据集(credit-g,kr-vs-kp,sick),分别进行噪声实验,从上述数据集中,对每一个数据集人工随机加入 5%,10%,20%,30% 的噪声,以测定在不同噪声的条件下,不同算法的分类准确率。表 4 ~ 表 8 分别记录了 NDK-Adaboost, Adaboost, LogitBost, MutiboostAB 和 WBIT 算法在不同噪声条件下的分类准确率。

表 4　NDK-Adaboost 各种噪声条件下的分类准确率　%

编号	数据集名称	噪声比例 5%	噪声比例 10%	噪声比例 20%	噪声比例 30%
5	credit-g	69.60	70.50	68.40	57.90
9	kr-vs-kp	98.37	97.03	93.62	84.79
11	sick	97.83	96.66	93.24	91.44

表 5　Adaboost 各种噪声条件下的分类准确率　%

编号	数据集名称	噪声比例 5%	噪声比例 10%	噪声比例 20%	噪声比例 30%
5	credit-g	66.90	64.70	60.10	56.70
9	kr-vs-kp	89.49	89.52	87.45	71.75
11	sick	91.91	85.02	72.96	61.74

表 6　LogitBost 各种噪声条件下的分类准确率　%

编号	数据集名称	噪声比例 5%	噪声比例 10%	噪声比例 20%	噪声比例 30%
5	credit-g	69.00	67.80	62.70	61.30
9	kr-vs-kp	89.55	89.24	81.16	80.13
11	sick	93.21	85.79	78.66	65.83

表 7 MutiboostAB 各种噪声条件下的分类准确率 %

编号	数据集名称	噪声比例 5%	噪声比例 10%	噪声比例 20%	噪声比例 30%
5	credit-g	67.40	66.50	59.60	58.70
9	kr-vs-kp	93.46	93.52	80.94	81.79
11	sick	93.27	84.12	67.71	56.20

表 8 WBIT 各种噪声条件下的分类准确率 %

编号	数据集名称	噪声比例 5%	噪声比例 10%	噪声比例 20%	噪声比例 30%
5	credit-g	68.40	69.90	65.30	59.80
9	kr-vs-kp	94.93	91.43	80.48	69.90
11	sick	97.22	93.98	87.30	84.89

从上面 5 张表中可以发现,NDK-Adaboost 算法在有噪声的条件下仍能保持较高的分类准确率,并且在相同噪声条件下,常常优于其他 4 种算法。当加入噪声不断增加的情况下,分类准确率虽然有所下降,大部分情况下没有出现急剧下降(除了 credit-g 中,噪声从 20% ~30%),而其他 3 种算法当噪声变得越来越大的时候,分类准确率下降很快。图 2 ~图 4 分别展示了在 3 个数据集上的各种算法比较的折线图。可以看出,在 sick 数据集上使用 Adaboost 算法,当噪声从 5% 增加到 30% ,分类准确率从 91.91% 降到了 61.74% ,而在相同的条件下,NDK-Adaboost 算法依然保持分类准确率在 90% 以上。这也证明在有噪声的情况下,Adaboost 算法会受很大影响,同时也表明,在有噪声的情况下,NDK-Adaboost 算法能够克服传统 Adaboost 这一弱点。

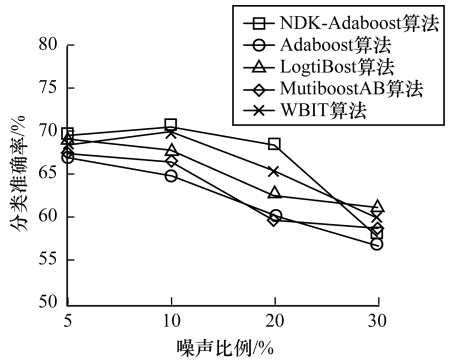

图 2 credit-g 数据集上各个算法的分类准确率

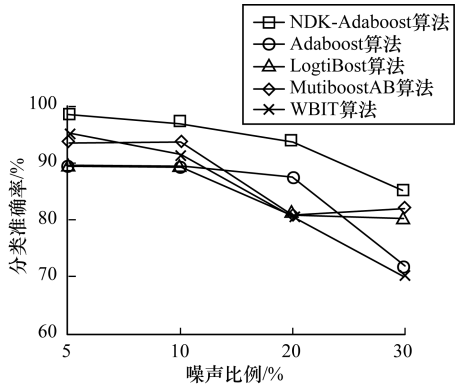

图 3 kr-vs-kp 数据集上各个算法的分类准确率

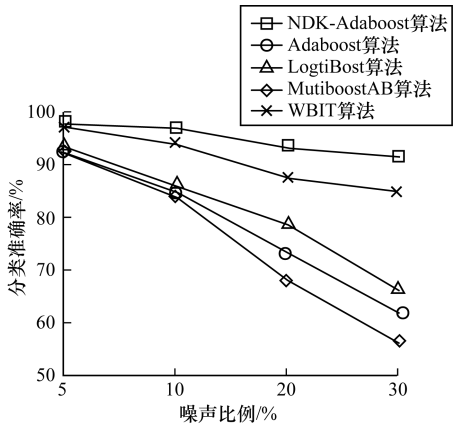

图 4 sick 数据集上各个算法的分类准确率

4 结束语

针对传统 Adaboost 算法在有噪声情况下分类准确率低的问题,本文给出一种基于噪声自检测的分段非线性组合 Adaboost 改进算法,根据不同样本分类采用不同的基分类器系数,使用大量 UCI 的数据,从多个角度进行了实验,结果都表明本文提出的改进算法有更好的分类准确率。现有的噪声检测模型只针对于二分类问题,对于多分类问题,模型将不适用。将二分类问题的噪声检测模型推广到多分类问题,这将要从数学上重新修改模型,使其更好地检测噪声,同时,将进一步研究 λ 和 K 对算法的影响以及如何根据训练集选取最佳数值。

参考文献

[1] Valiant L G. A Theory of the Learnable[J]. Communications of the ACM,1984,27(11):1134-1142.

[2] Tenenbaum J B, De S V, Langford J C. A Global Geometric Framework for Nonlinear Dimensionality Reduction[J]. Science,2000,290(5500):2319-2341.

[3] Freund Y, Schapire R E. A Decision-theoretic Generalization of On-line Learning and an Application to Boosting[J]. Journal of Computer & System Sciences, 2010,55(1):119-139.

[4] 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.

[5] 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.

[6] Zhou Shuisheng,Warmuth M K,Dong Yinli,et al. New Combination Coefficients for AdaBoost Algorithms[C]// Proceedings of International Conference on Natural Computation. Washington D. C., USA: IEEE Press,2010:3194-3198.

[7] Romero E, Màrquez L, Carreras X. Margin Maximization with Feed-forward Neural Networks: A Comparative Study with SVM and AdaBoost[J]. Neurocomputing,2004,57(1):313-344.

[8] Jiang Wenxin. Does Boosting Overfit: Views from an Exact Solution [EB/OL]. (2001-01-01). https://academic.microsoft.com/#/detail/172026072.

(下转第 173 页)

法不但保持了 HOG-SVM 算法的高效性,同时有效提升了行人检测的准确率与召回率。本文侧重于增加算法对检测区域多样性的判定,特征提取仅采用了 HOG 一种特征,因此,改进特征提取算法是下一步研究的内容。

参考文献

[1] Dalal N, Triggs B. Histograms of Oriented Gradients for Human Detection [C]//Proceedings of CVPR'05. San Diego, USA: [s. n.], 2005: 886-893.

[2] Li Jianan, Liang Xiaodan, Shen Shengmei, et al. Scale-aware Fast R-CNN for Pedestrian Detection[EB/OL]. (2015-10-28). https://arxiv.org/pdf/1510.08160v1.pdf.

[3] 苏松志,李绍滋,陈淑媛,等. 行人检测技术综述[J]. 电子学报,2012,40(4):814-820.

[4] Viola P, Jones M J, Snow D. Detecting Pedestrians Using Patterns of Motion and Appearance [J]. International Journal of Computer Vision, 2005, 63(2): 153-161.

[5] Ojala T, Pietikainen M, Harwood D. Performance Evaluation of Texture Measures with Classification Based on Kullback Discrimination of Distributions[C]//Proceedings of ICPR'94. Jerusalem, Israel: [s. n.], 1994: 582-585.

[6] Dollár P, Appel R, Belongie S, et al. Fast Feature Pyramids for Object Detection[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2014, 36(8): 1532-1545.

[7] Yann Lecun, Bottou L, Bengio Y, et al. Gradient-based Learning Applied to Document Recognition [J]. Proceedings of the IEEE, 1998, 86(11): 2278-2324.

[8] Krizhevsky A, Sutskever I, Hinton G E. ImageNet Classification with Deep Convolutional Neural Networks[C]//Proceedings of NIPS'12. California, USA: [s. n.], 2012: 1097-1105.

[9] Zeiler M D, Fergus R. Visualizing and Understanding Convolutional Networks[C]//Proceedings of ECCV'14. Zurich, Switzerland: [s. n.], 2014: 818-833.

[10] Simonyan K, Zisserman A. Very Deep Convolutional Networks for Large-scale Image Recognition[EB/OL]. (2014-09-04). https://arxiv.org/pdf/1409.1556.pdf.

[11] Szegedy C, Liu Wei, Jia Yangqing, et al. Going Deeper with Convolutions [C]//Proceedings of CVPR'15. Boston, USA: [s. n.], 2015: 1-9.

[12] He Kaiming, Zhang Xiangyu, Ren Shaoqing, et al. Deep Residual Learning for Image Recognition [C]// Proceedings of CVPR'16. Las Vegas, USA: [s. n.], 2016: 770-778.

[13] 杨　航,张鑫淼,杨　冲. 基于卷积神经网络的公路限速牌识别方法[J]. 地理空间信息,2016,14(1):31-33.

[14] 胡　丹,周兴社,许婉君,等. 基于深度特征与 LBP 纹理融合的视觉跟踪[J]. 计算机工程,2016,42(9): 220-225.

[15] 徐　渊,许晓亮,李才年,等. 结合 SVM 分类器与 HOG 特征提取的行人检测[J]. 计算机工程,2016,42(1): 56-60.

[16] 李　航. 统计学习方法[M]. 北京:清华大学出版社,2012.

编辑　刘　冰

(上接第 168 页)

[9] Servedio R A. Smooth Boosting and Learning with Malicious Noise [C]//Proceedings of the 10th International Conference on Human Computer Interaction. New York, USA: ACM Press, 2003: 473-489.

[10] Friedman J, Hastie T, Tibshirani R. Additive Logistic Regression: A Statistical View of Boosting[J]. Annals of Statistics, 2000, 28(2): 374-376.

[11] Ge Junwei, Lu Daobing, Fang Yiqiu. A Revised Training Mechanism for AdaBoost Algorithm [C]//Proceedings of IEEE International Conference on Software Engineering and Service Sciences. Washington D. C., USA: IEEE Press, 2010: 491-494.

[12] Yang Zhihai, Xu Lin, Cai Zhongmin. Re-scale AdaBoost for Attack Detection in Collaborative Filtering Recommender Systems[J]. Knowledge-based Systems, 2015(6): 74-88.

[13] Guo Haixiang, Li Yijing, Li Yanan, et al. BPSO-Adaboost-KNN Ensemble Learning Algorithm for Multi-class Imbalanced Data Classification [J]. Engineering Applications of Artificial Intelligence, 2015, 49(3): 176-193.

[14] 丁天怀. 基于 AdaBoost 算法的快速虹膜检测与定位[J]. 清华大学学报(自然科学版),2008,48(11): 1923-1926.

[15] 严　超,王元庆,李久雪,等. AdaBoost 分类问题的理论推导[J]. 东南大学学报(自然科学版),2011, 41(4): 700-705.

[16] Domingo C, Watanabe O. MadaBoost: A Modification of AdaBoost[C]//Proceedings of the 13th Conference on Computational Learning Theory. [S. l.]: Morgan Kaufmann Publishers Inc., 2000: 180-189.

[17] 刘雪莲. AdaBoost 中加权方式的改进[D]. 北京:北京交通大学,2010.

[18] Breiman L, Breiman L. Bagging Predictors" Machine Learning[J]. Lecturenotes in Computer Science Ai, 1996, 20(1-2): 35-61.

[19] Webb G I. MultiBoosting: A Technique for Combining Boosting and Wagging[J]. Machine Learning, 2000, 40(2): 159-196.

[20] Freund Y, Schapire R E. A Decision-theoretic Generalization of On-line Learning and an Application to Boosting[J]. Journal of Computer & System Sciences, 1999, 55(7): 119-139.

编辑　顾逸斐