

基于 K 近邻统计的非线性 AdaBoost 算法

苟 富, 郑 凯*

(华东师范大学 计算中心, 上海 200062)

(* 通信作者电子邮箱 kzheng@cs.ecnu.edu.cn)

摘 要: AdaBoost 是数据挖掘领域最常见的提升算法之一。对传统 AdaBoost 将各个基分类器线性相加所存在的不足进行分析, 并针对 AdaBoost 各个弱分类器的加权方式提出新的改进, 将传统的线性相加改为非线性组合, 把从学习过程得到的固定不变的权重系数改为由预测阶段的具体实例决定的动态参数, 该参数基于待测实例 K 近邻的分类结果统计, 从而使各个基分类器的权重更贴近当前待测实例的实际可靠度。实验结果表明, 与传统 AdaBoost 相比, 提出的非线性改进算法对不同数据集均有不同程度提升, 提升最高的达到了 7 个百分点。由此证明, 提出的改进是一种更加准确的分类算法, 对绝大多数数据集均能得到更高的分类准确率。

关键词: AdaBoost; 数据挖掘; 分类器; 非线性; K 近邻

中图分类号: TP181; TP391.4 **文献标志码:** A

Nonlinear AdaBoost algorithm based on statistics for K -nearest neighbors

GOU Fu, ZHENG Kai

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

Abstract: AdaBoost is one of the most popular boosting algorithms in the area of data mining. By analyzing the disadvantages of the traditional AdaBoost using linear combination of the basic classifiers, a new algorithm was proposed, which changed the traditional linear addition into a nonlinear combination, and replaced the constant weights acquired in the training stage by a series of dynamic parameters based on the statistics of the K -nearest neighbors and decided by the instances in the predicting stage. In this way, the weight of each basic classifier was closer to reality. The experimental results show that, compared to the traditional AdaBoost, the new algorithm can increase the prediction accuracy nearly seven percentage points at most. The new algorithm is more accurate and it can achieve higher classification accuracy for most data sets.

Key words: AdaBoost; data mining; classifier; nonlinear; K -nearest neighbor

0 引言

数据分类^[1]是数据挖掘的一个重要研究方向, 是一种重要的数据分析形式, 其一般分为学习和分类两个阶段。学习阶段分类算法通过已知的训练样本集构造分类器, 分类阶段使用前一阶段得到的分类器预测给定数据的类别。在学习阶段用来构造分类器的算法有很多, 如决策树归纳、朴素贝叶斯等, 在这些分类算法日趋成熟的同时, 又出现了诸如装袋、提升和随机森林等提高分类准确率的技术。AdaBoost 作为一种最常见的提升算法, 受到了研究者的广泛关注与研究, 并得到了各种各样的改进。文献[2]针对数据不平衡问题提出一种基于正负类损失函数的 AdaBoost 改进算法, 用训练好的基分类器对每次训练子集的补集进行损失估计, 根据该损失更新分类错误的样本权重, 从而有效避免数据不平衡的问题; 文献[3]针对 AdaBoost 存在的训练消耗大的问题提出一种基于特征裁剪的 FPAdaBoost (Feature Pruning AdaBoost) 算法, 通过裁剪掉一部分分类误差较大的特征来提高算法的训练速度; 文献[4]通过设定样本权重阈值来防止样本更新时某些错误样本权重过大, 一定程度上避免了过度拟合的问题。

目前研究者主要从数据集的权重更新和训练速度方面进

行改进, 对 AdaBoost 组合分类器的加权方式研究较少。本文首先对传统的 AdaBoost 算法进行简单的介绍, 并阐述了 AdaBoost 的算法流程, 然后针对 AdaBoost 组合分类器加权方式存在的不足进行分析, 并提出自己的改进, 将从学习过程得到的固定不变的权重系数改为由预测阶段的具体实例决定的动态参数, 将传统的线性相加改为非线性组合, 从而使各个基分类器的投票对最终结果的作用更加合理, 进而提升组合分类器的预测准确率。由于该方法与文献[5]提出的改进算法 WBTI (Weighting Based on Test Instances) 具有一定相似度, 随后对 WBTI 与本文的非线性 AdaBoost 分别进行实验验证, 通过对比证明了本文的方法具有更高的准确率。最后, 为更进一步地提高 AdaBoost 准确率提出一种可能的研究方向。

1 AdaBoost 算法概述

对于分类问题, 通过给定的样本数据要得到一个准确率一般的分类器 (弱分类器) 比得到一个精确的分类器容易得多, AdaBoost 的主要思想就是反复利用最普通的学习方法 (如决策树、朴素贝叶斯等) 得到一系列的弱分类器, 然后让这些弱分类器进行加权投票。AdaBoost 算法过程主要有两大特点: 一是在迭代训练过程中加大了上一轮训练中分类错误的

收稿日期: 2015-04-20; 修回日期: 2015-05-26。 基金项目: 国家 863 计划项目 (2013AA01A211)。

作者简介: 苟富 (1989 -), 男, 山西大同人, 硕士研究生, 主要研究方向: 数据挖掘、机器学习; 郑凯 (1968 -), 男, 浙江宁波人, 副教授, 博士, 主要研究方向: 计算机网络、云计算。

样本权重,使得下一轮训练更加关注这些分类错误的样本;另一个特点是根据每个分类器在训练过程中的分类误差率计算出一个权重赋给对应的分类器,这样,越准确的分类器在最终的表决中起的作用越大^[6]。

1.1 AdaBoost 算法基本原理

AdaBoost 是一种迭代算法,其核心思想是针对同一个训练集训练不同的弱分类器,然后把这些弱分类器组合起来,构成一个更强的最终分类器。其每一个不同的弱分类器是通过改变训练样本集中数据分布来得到的,它根据每次训练集中每个样本的分类是否正确,以及上次的总体分类的准确率,来确定下一轮中每个样本的权值。将修改过权值的新数据集送给下层分类器进行训练,最后将每次训练得到的不同分类器组合在一起,作为最后的决策分类器。

1.2 AdaBoost 算法流程

1) 准备训练样本数据集,并初始化训练数据的权重分布。这里的权重决定了对应的样本在下一轮训练中受到的重视程度。刚开始一般采用均匀分布:

$$D_1 = (w_{1,1}, w_{1,2}, \dots, w_{1,N}) \quad (1)$$

其中: $w_{1,i} = 1/N (i = 1, 2, \dots, N)$ 。

设 M 为总迭代次数, m 为当前迭代轮数。

2) 使用带权重的样本集进行学习(任选一种分类算法),得到一个弱分类器 $G_m(x)$ 。这里的弱分类器只要优于胡乱猜测就是合格的分类器。

3) 计算弱分类器在训练数据集上的分类误差率:

$$e_m = P(G_m(x_i) \neq y_i) = \frac{\sum_{i=1}^N w_{m,i} I(G_m(x_i) \neq y_i)}{\sum_{i=1}^N w_{m,i}} \quad (2)$$

4) 根据弱分类器的分类误差率计算该弱分类器在最终决策中的投票权重。该权重代表了该弱分类器在训练过程中的总体准确度。

$$\alpha_m = \ln \frac{1 - e_m}{e_m} \quad (3)$$

5) 更新样本数据集的权重分布。由式(4)可以看出,分类错误的样本在下一轮的权重会增大;反之则减小。

$$D_{m+1} = (w_{m+1,1}, w_{m+1,2}, \dots, w_{m+1,N}) \quad (4)$$

其中: $w_{m+1,i} = w_{m,i} e^{-\alpha_m y_i G_m(x_i)}$ ($i = 1, 2, \dots, N, m = 1, 2, \dots, M$)。

6) 重复2)~5)步,直到满足一定的准确率或达到指定的迭代次数。

7) 将得到的弱分类器进行线性组合,得到最终的分器:

$$G(x) = \sum_{m=1}^M \alpha_m G_m(x) \quad (5)$$

由以上过程^[7]可以看出,分类器 $G_m(x)$ 在训练中表现出色,其系数 α_m 越大,在最终分类器 $G(x)$ 中的影响力也越大,有效利用了每个弱分类器在训练阶段表现出来的优劣性。

在传统方法中,AdaBoost 最终将各个弱分类器进行线性组合,由式(3)可以看出,每个弱分类器的系数 α_m 大于0,而且误差率 e_m 越低, α_m 越大。所以, α_m 代表了对应分类器在训练过程中分类正确的把握。这样的加权线性组合类似于生活上的投票中,德高望重的人说话分量就重一些。但是, α_m 代表

的是整个训练过程中正确分类的把握,是从之前大量分类过程中对分类可靠性的总结和平均,而真正的投票应该以当前具体问题正确判断的把握作为权重,对不同的待测实例不能等同对待,正如德高望重的人对自己不擅长的问题投票,那么权重理所当然应该偏低一点。

2 非线性 AdaBoost 改进算法

2.1 非线性 AdaBoost 算法改进分析

针对弱分类器线性组合的不足,本文提出一种非线性 AdaBoost 改进算法,将原来固定不变的基分类器权重改为由具体待测实例决定的动态系数,这一想法的出发点和文献[5]不谋而合。文献[5]提出把通过训练过程误差率计算出的权重与对待测实例正确预测的概率作乘积,得到的新算法记为 WBTI(Weighting Based on Test Instances)。该算法中,把每个基分类器单独来看,基于训练过程学习到的特征规律可以输出待测实例属于各类别的权重大小,权重最大的就是该实例所属的类别,对待测实例正确预测的概率就是该类别的权重占总权重的百分比。WBTI 既利用了训练过程整体的优劣性,又结合了具体待测实例的个体性。WBTI 的分类准确率确实有所提升,但对于训练过程计算出来的权重,训练样本是相当丰富的,在具体预测的时候,对于给定的待测实例,在训练样本中只有满足一定相似度的一小部分和该实例相关,大多数训练样本对个体预测的可靠性估计其实是起负面的误导作用。

比如一个人脸和非人脸的图片库,采集某 Haar 特征值作为样本的唯一属性,其特征值分布如图1所示。

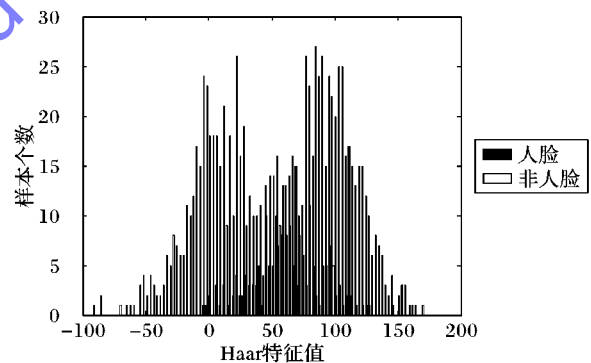


图1 人脸与非人脸 Haar 特征值分布

用 H 表示该特征值,图中: $-5 < H < 5$ 的样本中人脸有10个,非人脸90个; $70 < H < 80$ 中有人脸60个,非人脸40个。假设某分类器把特征值 $H < 60$ 的样本全部判断为非人脸, $H \geq 60$ 判断为人脸。显然, $-5 < H < 5$ 的样本只有10个人脸被误判为非人脸,而 $40 < H < 50$ 的样本中有40个人脸被误判为非人脸。可靠性 $p = N_{\text{正}} / (N_{\text{正}} + N_{\text{反}})$ ($N_{\text{正}}$ 、 $N_{\text{反}}$ 分别表示判断正确与错误的样本数),那么当一个待测实例的特征值 $H = 0$ 时,该分类器判断为非人脸的可靠性就大约为90%,因为在训练样本中凡是特征值接近0的样本有90%都判断正确了。而距离 $H = 0$ 较远的部分,如 $H = 75$ 附近的样本,其特征值相对 $H = 0$ 已经改变很大,表现出来的准确率往往不再接近90%(恰好接近90%的情况属于巧合)。所以,对 $H = 0$ 的待测实例,该分类器正确判断的可靠性为 $90 / (90 + 10) = 90\%$,将 $H = 75$ 附近的样本考虑在内的话,会将可靠性扭曲为 $(90 + 60) / (90 + 10 + 60 + 40) = 75\%$ 。

所以本文主张彻底废弃通过训练过程的整体误差率计算的权重,改为完全由待测实例决定的动态参数,该参数可以继续使用 WBTI 算法中对待测实例正确预测的概率,也可以从待测实例最相似的若干样本中统计局部准确率。前者(把这种方法称作基于基分类器的非线性 AdaBoost)所用的概率来源于基分类器自身的输出,一个训练好的基分类器可以根据样本特征输出其判为各个类别的可能性大小,也就是该基分类器作出正确判断的把握大小。对于无噪声的理想数据,一个基分类器往往可以很好地学习其特征,并输出相对满意的结果,但遇到噪声数据的误导时,学习效果就会大打折扣。所以,基于基分类器的非线性 AdaBoost 算法对理想化的数据应该是一种很好的改进,提升效果也会优于 WBTI,但现实中大多数数据集或多或少会存在些噪声,如果有少数几个弱分类器以较高的预测概率而预测错误了,则有可能造成整个模型的预测错误,因此鲁棒性会降低。

本文选择第二种做法,通过计算样本集中各个样本和具体待测实例的相似度,找出最相似的 K 个样本,统计弱分类器在这 K 个与待测实例最相似的样本中表现出来的分类误差率(把这种方法称作基于 K 近邻统计的非线性 AdaBoost 算法)。目前已有多种计算样本相似度的算法,比如对连续型属性的样本可通过计算样本间的欧氏距离来衡量相异程度;对离散型属性的样本,一种简单的处理方法是按照属性值相同的属性个数把样本集分为若干子集,比如一个拥有 1000 个样本、20 个属性的样本集,要获取与待测实例最相似的局部样本集,假设与待测实例属性值相同的属性个数为 20, 19, ..., 14 的样本个数依次为 3, 8, 15, 20, 18, 30, 38, 那么相同属性个数大于 15 的样本已经有 94 个,可以把这 94 个样本作为局部样本集来统计局部分类误差率,万一其中出现噪声样本,一个噪声给基分类器权重带来的误差仅为 $1/94$ 。局部样本集容量越大,统计越精确,但是相对待测实例的总体偏移程度也越大,所以局部样本个数的选取是一种折中的做法。

$$e_m = \left(\sum_{i=1}^K I(G_m(x_i) \neq y_i) \right) / K \quad (6)$$

式(6)是局部分类误差率的计算公式,其中: y_i 表示 x_i 的实际类别, $G_m(x_i)$ 表示第 m 个基分类器对样本 x_i 的判断类别。这样的误差率 e_m 是从与待测实例最接近的若干样本中统计出来的,相比整个样本集的统计,更能代表分类器对于当前待测实例的实际误差率,如果其中存在少数噪声数据,其他非噪声数据还可以起到纠正作用,使得最终误差率偏离不会太大。基分类器在最终分类器中的权重采用式(7)计算:

$$\alpha_m = \ln \frac{1 - e_m}{e_m} = \ln \left(\frac{K}{\sum_{i=1}^K I(G_m(x_i) \neq y_i)} - 1 \right) \quad (7)$$

最终的组合分类器为:

$$G(x) = \sum_{m=1}^M \alpha_m G_m(x) \quad (8)$$

2.2 时间复杂度分析

由于训练样本数往往非常大,得到一个弱分类器非常耗时,而训练下一个弱分类器前又要更新训练样本的权重分布,这样每一个基分类器都要完全重新训练,所以 AdaBoost 算法最耗时的部分就在于各个基分类器的训练。本文的改进从分类阶段入手,在预测具体待测实例之前需要得到与待测实例最接近的 K 个样本,统计基分类器在这 K 个样本中表现出来

的误差率。因此本文的方法会增加一定的时间负担,增加的时间花费主要在于获取这 K 个最相样本,可以采用堆排序算法来实现,该过程平均时间复杂度为 $O(N \lg K)$ 。

2.3 非线性 AdaBoost 算法流程

算法:基于 K 近邻统计的非线性 AdaBoost。

输入: S : 类标记的训练元组集。

M : 迭代次数(每轮产生一个分类器)。

一种分类学习算法。

输出: 一个复合模型。

方法:

// 训练阶段, N 为训练样本总数

1) 将 S 中每个元组的权重初始化为 $1/N$;

2) for $m = 1$ to M do

3) 根据元组的权重从 S 中有放回抽样,得到 S_m ;

4) 使用训练集 S_m 训练分类器 $G_m(x)$;

5) 计算 $G_m(x)$ 的错误率 e_m ;

6) if $e_m > 1/C$ then // C 为样本类别数

7) 转步骤 3) 重试;

8) endif

9) for S_m 的每个被正确分类的元组 do

10) 元组的权重乘以 $e_m/(1 - e_m)$;

11) endfor

12) 规范化每个元组的权重;

13) endfor

// 预测阶段,使用组合分类器对元组 x 分类:

1) 计算 S 中每个元组与 x 的相似度;

2) 取相似度最大的 K 个元组;

3) for $m = 1$ to M do

$$4) \alpha_{m,x} = \ln \left(\frac{K}{\sum_{i=1}^K I(G_m(x_i) \neq y_i)} - 1 \right)$$

5) $G(x) += \alpha_{m,x} G_m(x)$

6) endfor

7) 返回具有最大权重的类;

3 实验及结果分析

3.1 实验数据

实验所用数据来自著名的 UCI 机器学习数据库^[8],这个数据库常被机器学习与模式识别研究者用来对算法进行测试和分析,从中随机选择 10 个数据集对本文提到的几种方法进行测试对比。表 1 为所用数据集的相关信息描述。其中: Dataset 为数据集名称, Samples 为样本数, Features 为属性数, Classes 为类别数。

表 1 各数据集信息描述

Dataset	Samples	Features	Classes
anneal	898	38	6
connect-4	67 557	42	3
arrhythmia	452	279	16
hypothyroid	3 772	30	4
house-votes-84	435	17	2
adult	48 842	15	2
glass	214	10	7
satimage	6 430	36	6
eucalyptus	736	20	5
breast-cancer	286	10	2

3.2 实验设计

实验通过 J48(一种决策树算法)、朴素贝叶斯树(Naive

Bayes Tree, NBTree)、朴素贝叶斯算法(Naive Bayes, NB)等不同算法作为基分类器,分别对传统 AdaBoost 算法、刘雪莲^[5]的 WBTI 算法以及本文提出的基于 K 近邻统计的非线性 AdaBoost 作对比。为了验证前文对基于基分类器的非线性 AdaBoost 降低了鲁棒性的猜想,将该方法也加入到对比实验中。迭代次数设定为默认值 10,采用十折交叉验证方式,预测阶段与待测实例最相似的样本数 K 根据样本集中样本数目大小分别设定为对应训练集中总样本数的 10%、5%、1% 或 0.5%,坚持的原则是 K 值尽量小,但又必须保证 K 个近似样本足够统计出局部范围内较准确的平均概率值。利用 Java 语言进行设计验证,weka^[9] 提供了机器学习相关的一系列开源代码,各个基分类器以及传统 AdaBoost 均直接使用 weka 提供的代码和默认参数,weka 提供的方法 distributionForInstance (Instance) 返回基分类器预测实例 Instance 为每个类别对应概率的数组, WBTI 可以利用该方法计算预测准确率。AdaBoostM1 中最终组合分类器的关键伪代码(各方法的实现代码中不同的部分)为:

```
传统 AdaBoost:
for m = 1 to M do
    G(x) +=  $\alpha_m G_m(x)$ 
endfor

WBTI:
for m = 1 to M do
    G(x) +=  $\alpha_m P_m(x) G_m(x)$ 
    //  $P_m(x)$  为  $G_m(x)$  判断  $x$  为预测类的概率
endfor
```

```
基于基分类器的非线性 AdaBoost:
for m = 1 to M do
    G(x) +=  $P_m(x) G_m(x)$ 
endfor

基于 K 近邻统计的非线性 AdaBoost:
for m = 1 to M do
    for i = 1 to K do
        // 对于 K 个最相似样本
        if ( $G_m(x_i) \neq y_i$ ) then
            // 判断  $x_i$  的类别不是实际类别  $y_i$  时
            // error 为判断错误的样本个数
            error ++
        endif
    endfor
     $e_m = \text{error}/K$ 
    if ( $e_m \neq 0$  and  $e_m < 0.5$ ) then
         $\alpha_m = \ln((1 - e_m)/e_m)$ 
    endif
    G(x) +=  $\alpha_m G_m(x)$ 
endfor
```

为了验证预测阶段增加 K 近邻统计过程引起的效率下降,手动从一些样本集中取出一定数目的样本作为测试集,剩余的为训练集,不再使用十折交叉验证,这样使两个阶段的运行更加分明,记录预测阶段完成所有样本的测试花费的总时间。

3.3 实验结果

表 2 是不同数据集与不同基分类算法产生的实验结果。

表 2 基分类算法、传统 AdaBoost、WBTI、非线性 AdaBoost 预测效果

数据集	基分类算法	预测错误数				预测准确率/%			
		传统 Adaboost	WBTI	基于基分类器	基于 K 近邻统计	传统 Adaboost	WBTI	基于基分类器	基于 K 近邻统计
breast-cancer	J48	87	91	85	67	69.5804	68.1818	70.2797	76.5734
connect-4	J48	11 662	11 498	11 340	11 144	82.7375	82.9803	83.2142	83.5043
anneal	NaiveBayes	57	59	57	40	93.6526	93.4298	93.6526	95.5457
satimage	J48	592	594	583	547	90.7932	90.7621	90.9331	91.4930
hypothyroid	J48	16	17	16	14	99.5758	99.5493	99.5758	99.6288
adult	NaiveBayes	8 181	8 162	8 263	7 599	83.2501	83.2890	83.0822	84.4417
house-votes-84	NBTree	18	18	17	13	95.8621	95.8621	96.0920	97.0115
anneal	J48	4	4	2	2	99.5546	99.5546	99.7773	99.7773
anneal	NBTree	4	4	4	4	99.5546	99.5546	99.5546	99.5546
hypothyroid	NBTree	16	15	18	16	99.5758	99.6023	99.5228	99.5758
adult	RandomForest	7 913	7 847	7 818	7 736	83.7988	83.9339	83.9933	84.1612
arrhythmia	J48	133	132	134	129	70.5752	70.7965	70.3540	71.4602
adult	J48	7 806	7 625	7 552	7 144	84.0179	84.3884	84.5379	85.3732
breast-cancer	NBTree	89	87	82	72	68.8811	69.5804	71.3287	74.8252
eucalyptus	NBTree	302	295	292	292	58.9674	59.9185	60.3261	60.3261
house-votes-84	RandomForest	25	22	29	22	94.2529	94.9425	93.3333	94.9425
satimage	NaiveBayes	1 314	1 314	1 305	1 117	79.5645	79.5645	79.7045	82.6283
glass	J48	55	53	51	45	74.2991	75.2336	76.1682	78.9720
glass	NBTree	52	51	50	40	75.7009	76.1682	76.6355	81.3084
connect-4	NaiveBayes	18 820	18 806	17 357	17 311	72.1420	72.1628	74.3076	74.3757

从表 2 数据看,20 个不同数据集和分类算法中,基本上都有不同程度的提升。

以上方法提升幅度对比曲线如图 2 所示。

和传统 AdaBoost 算法相比,文献[5]提出的 WBTI 与本文的基于局部统计的非线性 AdaBoost 算法以及基于基分类

器的非线性 AdaBoost 算法在多数情况下均有不同程度提升,而基于 K 近邻统计的非线性 AdaBoost 效果最为明显,最高的提升了将近 7 个百分点,只有两个是准确率保持不变的情况。其余方法均出现若干反例,这一点说明了基分类器的概率预测模型引起了鲁棒性的下降。

预测阶段运行效率对比如表3所示,其中 KAdaBoost 和 AdaBoost 分别表示基于 K 近邻统计的非线性 AdaBoost 算法和传统 AdaBoost 花费的预测时间,“KAdaBoost(ms)/测试样本数”表示使用 K 近邻 AdaBoost 算法平均每个样本花费的预测时间。

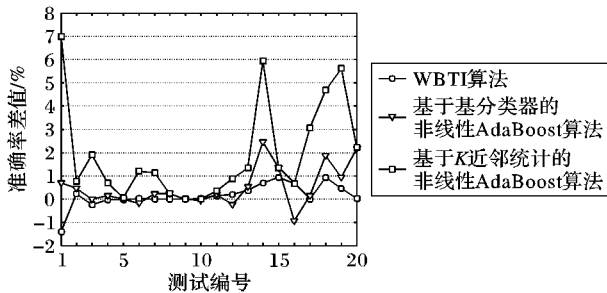


图2 改进算法与传统 AdaBoost 准确率差值曲线

表3 K 近邻 AdaBoost 与传统 AdaBoost 预测效率

dataset	训练 样本数	测试 样本数	KAdaBoost/ ms	AdaBoost/ ms	KAdaBoost(ms)/ 测试样本数
eucalyptus	536	200	312	16	1.56
anneal	698	200	499	16	2.495
hypothyroid	2972	800	7349	31	9.186 25
satimage	6230	200	7222	16	36.11
adult	38 842	10 000	1 028 794	468	102.879 4
connect-4	57 557	10 000	3 175 279	343	317.527 9

由于传统 AdaBoost 预测阶段只是把待测实例代入最终分类器中输出结果,时间复杂度为 $O(1)$;而基于 K 近邻 AdaBoost 预测阶段要遍历整个训练集来获取 K 近邻,时间复杂度为 $O(N \ln K)$,所以预测效率较低,实验结果也验证了这一点。所以本文提出的方法在现实生活中更适合于对单个个体分类的场合,算法执行时间基本都处于毫秒级别,而需要一次性进行批量分类的场合则效率问题会比较明显。另外,对训练集样本太多的情况可以通过减少遍历训练集样本个数来提高预测效率,比如:通过聚类方法把训练集预先分为 n 个聚类,找出待测实例最可能属于的 2 ~ 4 个聚类,最相似的 K 个样本主要集中在这几个聚类中,只需从这些聚类中再遍历找出最相似的 K 个样本。

4 结语

本文详细分析了 AdaBoost 算法中加权方式的不足之处,并以各个基分类器在具体待测实例最相似的若干样本中表现出来的准确率作为新的权重系数,使得加权方式更为合理。实验表明,新的加权方式使 AdaBoost 的准确率得到了进一步提升。

本文提出的非线性 AdaBoost 算法和传统方法相比,添加了获取 K 近邻的环节,时间复杂度有所增加。随着训练集样本个数和 K 值的不同,增加的时间也会不同,准确度提升效果也会不同,本文暂且采用一个总体把握的原则进行实验,更合适的 K 值获取方法还有待进一步研究,到时候相信会有更显著的提升效果。另外,AdaBoost 算法本身是一种效果比较显著的提升算法,目前的相关改进大多针对训练速度或与其他算法结合,本文通过修改算法本身使准确率更进一步提高,为进一

步的改进打开一个新的方向,今后可以朝这个方向寻找效率更高的改进方法。

参考文献:

- [1] HAN J W, KAMBER M. Data mining: concepts and techniques [M]. FAN M, MENG X, translated. Beijing: China Machine Press, 2012: 211 - 249. (HAN J W, KAMBER M. 数据挖掘: 概念与技术[M]. 范明, 孟小峰, 译. 北京: 机械工业出版社, 2012: 211 - 249.)
- [2] LEI L, WANG X. Improved AdaBoost ensemble approach based on loss function [J]. Journal of Computer Applications, 2012, 32(10): 2916 - 2919. (雷蕾, 王晓丹. 基于损失函数的 AdaBoost 改进算法[J]. 计算机应用, 2012, 32(10): 2916 - 2919.)
- [3] MENG Z, JIANG H, CHEN J, et al. Feature pruning based AdaBoost and its application in face detection [J]. Journal of Zhejiang University: Engineering Science, 2013, 47(5): 906 - 911. (孟子博, 姜虹, 陈婧, 等. 基于特征裁剪的 AdaBoost 算法及在人脸检测中的应用[J]. 浙江大学学报: 工学版, 2013, 47(5): 906 - 911.)
- [4] GE J, LU D, FANG Y. A revised training mechanism for AdaBoost algorithm [C]// Proceedings 2010 IEEE International Conference on Software Engineering and Service Sciences. Piscataway: IEEE, 2010: 491 - 494.
- [5] LIU X. The improvement of the weighting method in AdaBoost [D]. Beijing: Beijing Jiaotong University, 2010. (刘雪莲. AdaBoost 中加权方式的改进[D]. 北京: 北京交通大学, 2010.)
- [6] CAO Y, MIAO Q, LIU J, et al. Advance and prospects of AdaBoost algorithm [J]. Acta Automatica Sinica, 2013, 39(6): 745 - 758. (曹莹, 苗启广, 刘家辰, 等. AdaBoost 算法研究进展与展望[J]. 自动化学报, 2013, 39(6): 745 - 758.)
- [7] FAN Y. Research on face detection based on AdaBoost algorithm [D]. Hangzhou: Zhejiang University of Technology, 2008. (范一峰. 基于 AdaBoost 算法的人脸检测研究[D]. 杭州: 浙江工业大学, 2008.)
- [8] BLAKE C, KEOGH E, MERZ C. UCI machine learning repository [EB/OL]. [2014-10-24]. <http://www.ics.uci.edu/~mlearn/Mlrepository.html>.
- [9] WITTEN I H, FRANK E. Data mining: practical machine learning tools and techniques with Java implementations [M]. Beijing: China Machine Press, 2003: 265 - 296. (WITTEN I H, FRANK E. 数据挖掘: 实用机器学习技术及 Java 实现[M]. 北京: 机械工业出版社, 2003: 265 - 296.)
- [10] WANG W, NIU H. Face detection based on improved AdaBoost algorithm in e-learning [C]// Proceedings of 2012 IEEE 2nd International Conference on Cloud Computing and Intelligence Systems. Piscataway: IEEE, 2012, 2: 924 - 927.
- [11] GUO Q, LI L, LI N. Novel modified AdaBoost algorithm for imbalanced data classification [J]. Computer Engineering and Applications, 2008, 44(21): 217 - 221. (郭乔进, 李立斌, 李宁. 一种用于不平衡数据分类的改进 AdaBoost 算法[J]. 计算机工程与应用, 2008, 44(21): 217 - 221.)
- [12] LI R, LI C. Pruning AdaBoost algorithm based on covariance feature [J]. Application Research of Computers, 2014, 31(11): 3517 - 3520. (李睿, 李长风. 基于协方差特征的裁剪 AdaBoost 算法[J]. 计算机应用研究, 2014, 31(11): 3517 - 3520.)