

# 一种加入动态权重的 AdaBoost 算法\*

熊霖, 唐万梅

(重庆师范大学 计算机与信息科学学院, 重庆 401331)

**摘要:**【目的】由于 AdaBoost 算法弱分类器对各待测样本其投票权重是固定不变的,这会导致使用该算法进行分类时准确性不高,于是针对该问题提出了一种结合静态权重与动态权重的学习算法 AdaBoost\_DW。【方法】该算法保留了 AdaBoost 算法训练阶段产生的全部弱分类器,在预测阶段加入了能度量待测样本与各个弱分类器适应性的动态权重,通过双权重共同决定该弱分类器的投票权重。【结果】实验结果表明,AdaBoost\_DW 算法的分类精度绝大多数情况下高于原 AdaBoost 算法。【结论】证明了动态权重的加入能够提高原 AdaBoost 算法在各数据集上的分类准确率。

**关键词:** AdaBoost 算法;局部区域;适应性;动态权重

**中图分类号:** TP391

**文献标志码:** A

**文章编号:** 1672-6693(2019)05-0117-06

集成学习的基本思想是对同一问题使用若干个学习器进行学习,然后按照一定的策略将多个不同的学习结果进行整合,利用学习器之间的互补性,从而获得比单个学习器更好的效果。在分类问题中集成学习分为两个阶段,即弱分类器的训练阶段和弱分类器的组合阶段。其中弱分类器训练阶段是训练多个有差异的弱分类器,组合阶段主要是采用更有效的方式融合生成一个强分类器,以达到最优的学习效果。Breiman<sup>[1]</sup>在重复取样技术(Bootstrap sampling)的基础上,提出了著名的 Bagging 方法,该方法对原始训练集进行有放回的随机重复采样,得到多个有差异的训练集,用这些训练集分别训练分类器,在待测样本预测阶段,所有的分类器通过投票的方式来决定待测样本的预测结果。另一种集成学习方法是 Boosting(提升方法)<sup>[2]</sup>,它的思想来源于 Valiant<sup>[3]</sup>提出的 PAC(Probably approximately correct)学习模型。1997年 Freund 和 Schapire<sup>[4]</sup>提出的 AdaBoost(Adaptive boosting)算法是当前集成学习中最具有代表性且最流行的 Boosting 算法。AdaBoost 算法初始时对每个训练样本赋予相同的权重,每轮迭代训练一个弱分类器,降低能被该轮弱分类器正确分类样本的权重,提升被错误分类样本的权重,使后面训练的弱分类器把分类重点集中在那些较难分的样本上。每一轮弱分类器的投票权重都来源于该弱分类器对整个训练集的分类能力。将多轮学习获得的若干个弱分类器通过加权组合得到最终的分类器。然而随着迭代次数的增多,分类器会过多的关注那些较难分的样本,产生的分类器多样性逐渐消失,大量的分类器不但会降低训练和预测速度,而且会出现过拟合问题。

近几年的研究表明,对 AdaBoost 算法的改进主要集中在 3 个方面:1)改进样本权值更新的策略,大多是抑制权值过快偏向于难分的样本<sup>[5-7]</sup>;2)改进 AdaBoost 算法的训练过程以提升算法性能<sup>[8-10]</sup>;3)与其他算法结合以提升 AdaBoost 算法分类精度<sup>[11-12]</sup>。针对数据集分布的具体问题,文献[13]用 AdaBoost 算法集成多个遗传规划分类器,很大程度提高了地震预测系统的预测精度。文献[14]提出可以用不同分类算法在同一训练集上训练多个候选分类器,预测阶段选择与待测样本适应性好的基分类器做预测,经证实可以提升分类准确率。文献[15]提出的 AdaBoost SVM 算法,用 AdaBoost 训练多个 SVM 弱分类器然后组合成一个强分类器,得到了更好的分类效果。文献[16]首先用 AdaBoost 算法训练若干个 SVM 弱分类器,用 K-means 算法剔除冗余的分类器,然后选择待测样本周围最近的  $k$  个训练样本点,选出对这  $k$  个样本点分类准确率最高的弱分类器来预测待测样本的类别。该方法虽然很多时候可以获得比原 AdaBoost 算法更好的性能,但是需选用 SVM 这种分类精度很高的基分类算法。选择单个分类器不能像集成分类器那样形成互补,且寻找待测样本近邻时会花费很多时间,因

\* 收稿日期:2018-11-13 修回日期:2019-04-18 网络出版时间:2019-09-26 11:24

资助项目:国家自然科学基金面上项目(No. 11671062);重庆市教育规划重点项目(No. 2017-GX-116)

第一作者简介:熊霖,男,研究方向为机器学习与数据挖掘,E-mail:921147163@qq.com;通信作者:唐万梅,女,教授,博士,E-mail:1093895431@qq.com

网络出版地址: <http://kns.cnki.net/kcms/detail/50.1165.N.20190926.1124.026.html>

此它的适用范围会大打折扣。

为了提高 AdaBoost 算法在分布各异的数据集上的适应性,本文提出了一种结合静态权重和动态权重的分类算法(AdaBoost\_Dual Weights, AdaBoost\_DW)。该算法首先计算弱分类器对待测样本周围多个近邻点的分类准确率,以此来度量该弱分类器和待测样本的适应性,从而使各个弱分类器获得针对待测样本的动态权重。静态权重来自于该弱分类器对整个训练集的分类能力。将静态权重和动态权重相乘得到弱分类器的组合权重,最后用多个弱分类器的组合权重进行投票得到待测样本的类别。

### 1 AdaBoost\_DW 算法

Adaboost 算法的多个弱分类器一旦训练完成,弱分类器的权重在预测不同的待测样本时便固定下来。由于分类器具有偏好,有些弱分类器虽然在整个训练集上分类能力好,但有可能缺少对某一局部区域的学习,造成对这一区域分类能力较低,特别是对于某些呈现多个簇分布的数据集,如图 1a 所示。

为了解决这个问题,本文提出了 AdaBoost\_DW 算法,该算法基于以下假设:如果一个分类器对与待测样本相似的数据有较好的分类能力,那么对待测数据同样有较高的分类准确性<sup>[17]</sup>。算法在弱分类器拥有静态权重的基础上,加入能度量该弱分类器与待测样本适应性的动态权重。在预测阶段,将静态权重和动态权重相乘得到弱分类器的组合权重来预测待测数据类别。算法如图 2 所示。左边的正方形代表数据集,正方形中长条宽度代表每个样本的权重,用  $D_{w1}$  训练一个弱分类器  $h_1$ ,然后更新样本权重分布得到  $D_{w2}$ ,再用  $D_{w2}$  训练分类器  $h_2$ 。这样迭代  $T$  次,得到  $T$  个拥有静态权重( $\alpha$ )的弱分类器。 $\beta$  为弱分类器对待测样本周围多个训练样本的分类准确率。组合权重的方法有很多,多次试验发现不同组合权值的方法对分类的结果影响并不大,所以本文采用了最简单的乘法归一化方法,每个弱分类器的投票权重为  $\alpha \times \beta$ 。

#### 1.1 局部敏感哈希寻找近邻

在 AdaBoost 算法的研究<sup>[5,12,16-17]</sup>中,寻找与待测样本的近邻大多是直接计算样本间的欧氏距离,对于海量高维数据,这种方法会花费大量的时间。可以采用类似索引技术来加快查找过程,在提出的 AdaBoost\_DW 算法中,采用局部敏感哈希 LSH<sup>[18-19]</sup>(Locality sensitive hashing)的方式来查找待测样本的近邻,该方法基本思想是:高维空间中相邻的数据经哈希映射到低维空间仍相邻的概率很高;高维空间不相邻的数据映射到低维空间相邻的概率很低。通常对高维数据进行多次哈希处理,这样相邻数据有很大的可能性落到同一个桶中。将待测样本同样哈希到桶中,该桶中的数据与该待测数据相邻的可能性较高。Indyk 等人<sup>[18]</sup>提出固定哈希桶的大小可将 LSH 时间复杂度由  $O(dn^{1/\epsilon})$  降低到  $O(dn^{1/(1+\epsilon)})$ ,其中  $d$  为数据的维数, $n$  为数据量, $\epsilon$  越大则时间复杂度越小,但准确率会稍低。LSH 查搜索范围低于线性复杂度,故被大量运用于相似性搜索。

#### 1.2 计算弱分类器动态权重

AdaBoost 算法前几轮训练的弱分类器可能在数据集某一局部区域分类能力差,但因为在整个训练集上分类

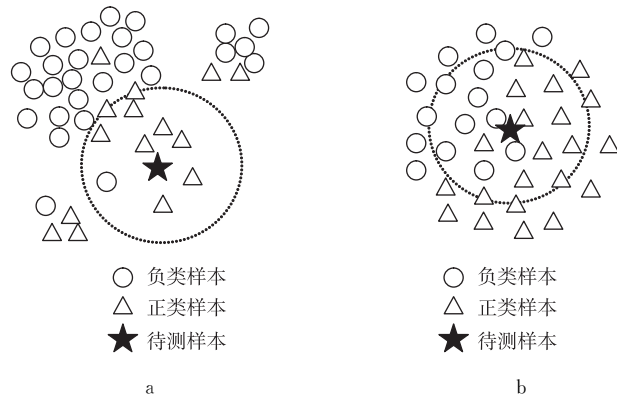


图 1 不同分布的数据集

Fig. 1 Type of data distributions

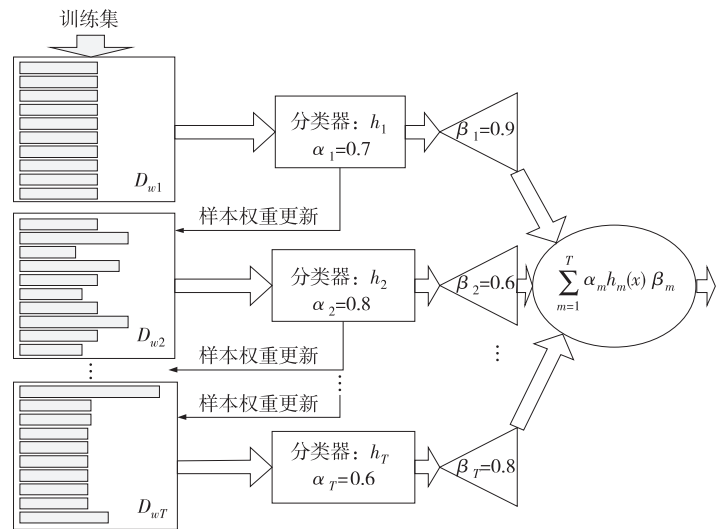


图 2 AdaBoost\_DW 算法示意图

Fig. 2 The diagram of the AdaBoost\_DW

能力好而获得了较大的权重;而后期训练的弱分类器恰恰相反。加入动态权重是为了在预测阶段对弱分类器投票权重进行一个调整。Adaboost\_DW 算法将待测样本哈希到一个桶中,它的近邻样本即是在这个桶中的其他数据。然后用每个弱分类器对这些数据所组成的临时测试集进行分类测试,以度量各弱分类器与该待测样本的适应能力,分类准确率作为该弱分类器在预测该待测样本时的动态权重,用

$$\beta = \frac{k \text{ 个近邻点中分类正确的样本数量}}{\text{近邻样本数 } k} \quad (1)$$

计算每个弱分类器的动态权重。

### 1.3 算法实现

AdaBoost 算法在训练阶段随着迭代次数的增加,训练的弱分类器逐渐倾向于关注那些难分的样本点,这可能使得后续训练的许多弱分类器在整个数据集上的分类错误率大于 0.5,从而不会被加入到最终的组合分类器中,但这些弱分类器在某些局部区域却能达到很好的分类效果。Adaboost\_DW 算法训练弱分类器过程与原 Adaboost 算法相比,保留了所训练的全部弱分类器;在分类器的组合阶段,加入了能度量待测样本与各个弱分类器适应性的动态权重,将各个弱分类器的静态权重与针对待测样本所获得的动态权重相乘得到弱分类器的整体权重,用整体权重投票判定待测样本的类别。AdaBoost\_DW 算法伪代码如下所示。

AdaBoost\_DW 算法流程:

输入:训练集  $D = \{(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)\}$ , 其中  $x_i \in \mathcal{X} \subseteq \mathbf{R}^n$ ,  $y_i \in Y = \{-1, +1\}$ ; 迭代次数  $T$ ; 弱学习算法; 测试数据  $(x_{\text{test}}, \cdot)$ ;

输出:带双权重的分类器集合  $f(x)$ 。

训练阶段:

步骤 1, 初始化每个样本权重  $D_1 = (w_{11}, \dots, w_{1i}, \dots, w_{1N})$ ,  $w_{1i} = \frac{1}{N}, i = 1, 2, \dots, N$ 。

步骤 2, 对  $m = 1 \sim T$  作如下操作:

a) 用具有权重分布  $D_m$  的训练集学习得到弱分类器:  $h_m(x): \mathcal{X} \rightarrow \{-1, +1\}$ ;

b) 计算  $h_m(x)$  在训练集上的分类误差:

$$e_m = P(h_m(x_i) \neq y_i) = \sum_{i=1}^N w_{mi} I(h_m(x_i) \neq y_i); \quad (2)$$

c) 计算  $h_m$  的静态权重:

$$\alpha_m = \frac{1}{2} \ln \frac{1 - e_m}{e_m}; \quad (3)$$

d) 更新训练集权重分布:

$$D_{m+1} = (w_{m+1,1}, \dots, w_{m+1,i}, \dots, w_{m+1,N}),$$

$$w_{m+1,i} = \begin{cases} \frac{w_{mi}}{Z_m} e^{-\alpha_m}, & h_m(x_i) = y_i; \\ \frac{w_{mi}}{Z_m} e^{\alpha_m}, & h_m(x_i) \neq y_i. \end{cases} \quad (4)$$

其中,  $Z_m = \sum_{i=1}^N w_{mi} \exp(-\alpha_m y_i h_m(x_i))$ ,  $Z_m$  是规范因子。

步骤 3, 得到  $T$  个弱分类器  $H = \{h_1, h_2, \dots, h_T\}$ 。

预测阶段:

步骤 4, 用 1.1 节的 LSH 方法从训练集中找到与待测样本  $(x_{\text{test}}, \cdot)$  相似的多个样本组成临时测试集  $D$ ;

步骤 5, 对  $i = 1, 2, \dots, T$  作如下操作:

用弱分类器  $h_i$  分别对  $D$  进行分类测试, 用(1)式计算分类器的动态权重  $\beta_i$ ;

结果, 得到对于待测样本  $(x_{\text{test}}, \cdot)$  的线性组合分类器:

$$f(x) = \sum_{m=1}^T \alpha_m h_m(x) \beta_m \quad (5)$$

## 2 实验结果与分析

### 2.1 实验数据集

本文提出的算法采用 Python 2.7 在 Windows 7 操作系统上实现。为了验证本文提出的算法的有效性,在 10 个数据集上进行了仿真实验。由于提出的 AdaBoost\_DW 算法没有考虑多分类问题,所以在 UCI 数据库中随机选择多个领域的二分类数据集,数据集的信息描述如表 1 所示。

### 2.2 实验参数设置

选择的对比算法有 AdaBoost<sup>[20]</sup> 算法、支持向量机 (SVM) 算法<sup>[21]</sup> 和随机森林 (RandomForest)<sup>[22]</sup>。其中 AdaBoost 算法与 AdaBoost\_DW 算法都选择了单层决策树作为基分类算法,算法的迭代次数都设置为 10 次(即训练 10 个弱分类器);实验过程中 SVM 算法的实现是直接调用当前最为流行的 LIBSVM 工具包,经过多次试验,确定参数设置为默认参数可以达到非常好的分类效果;根据文献<sup>[22-23]</sup>的建议,随机森林所采用的参数

为:森林里包含 10 棵决策树;随机选择特征数为 $\sqrt{d}$ ;用基尼系数来度量不纯度)。

在查找待测样本近邻及计算弱分类器动态权重时,采用余弦度量相似性的局部敏感哈希,其中哈希表尺寸凭经验设置为: $\text{hash\_size} = \frac{\text{训练集数据总量}}{100}$ ,因为 AdaBoost\_DW 算法不需要精确查找最近邻,以兼顾算法的分类精度和性能。

### 2.3 实验结果与分析

实验过程中每个数据集随机选择 80% 的数据用作训练集用于分类器的训练和局部敏感哈希表的构建;剩下 20% 的数据用作测试数据,用于测试分类器的泛化能力。为了使得实验数据更加可靠,重复实验 10 次,表 2 给出了 4 种算法在实验数据集上的预测准确率(平均准确率 $\pm$ 标准差)。

表 2 4 种算法在实验数据集上的预测准确率

Tab. 2 The classification accuracy of four algorithms on experimental data sets

数据集	准确率/%			
	Random Forest	SVM	AdaBoost	AdaBoost_DW
seeds_dataset	92.14 $\pm$ 5.27	95.71 $\pm$ 4.39	96.43 $\pm$ 3.76*	96.43 $\pm$ 3.76*
Colic	73.57 $\pm$ 2.04	65.77 $\pm$ 2.87	73.12 $\pm$ 3.13	74.32 $\pm$ 3.45*
Liver_Patient	71.51 $\pm$ 3.63	71.98 $\pm$ 3.30*	70.47 $\pm$ 3.39	71.70 $\pm$ 2.85
blood_transfusion	75.78 $\pm$ 3.26	76.53 $\pm$ 3.68	78.04 $\pm$ 3.42*	77.20 $\pm$ 3.38
german	69.39 $\pm$ 2.63	70.06 $\pm$ 2.77	75.28 $\pm$ 4.03	75.89 $\pm$ 2.80*
messidor	63.42 $\pm$ 2.98	59.34 $\pm$ 3.11	63.24 $\pm$ 3.32	64.11 $\pm$ 3.01*
Credit	72.95 $\pm$ 1.99	71.25 $\pm$ 2.54	73.40 $\pm$ 3.17	74.00 $\pm$ 2.19*
wilt	94.11 $\pm$ 0.59	94.24 $\pm$ 1.74	94.11 $\pm$ 0.59	94.24 $\pm$ 0.82*
Dota2S	53.63 $\pm$ 1.50	56.61 $\pm$ 0.55*	54.28 $\pm$ 1.10	54.47 $\pm$ 1.04
skin	90.27 $\pm$ 0.74	91.92 $\pm$ 0.29	87.35 $\pm$ 1.68	96.58 $\pm$ 1.97*

注:\* 表示同数据集中准确率最高的算法。

表 1 实验数据集信息

Tab. 1 Properties of data sets

数据集	样本数	特征数	领域
Seeds_dataset	139	7	农业物理学
Colic	366	21	生物学
Liver_Patient	583	10	医学
blood_transfusion	748	5	商学
german	1 000	24	金融学
messidor	1 151	19	医学
Credit	1 000	12	金融学
wilt	4 839	5	生命科学
Dota2S	7 760	116	游戏
skin	8 635	3	计算机



通过表2的实验结果可以看出:本文提出的算法只有在 blood\_transfusion 数据集上预测准确率稍低于原 AdaBoost 算法,经过在多个人工数据集上实验进行了验证,这是因为该数据集分布没有明显的局部特征,即数据集中所有数据呈现集中分布趋势(如图1b),导致动态权重的设置出现偏差,但与原 AdaBoost 预测精度相差很小。在另外的9个数据集上 AdaBoost\_DW 算法相对于 AdaBoost 算法预测准确率有不同程度的提高,这应该是这些数据集数据分布离散程度比较大,具有类似于图1a这样的局部特征,动态权重的加入能让各弱分类器针对待测样本获得更加合理的投票权重,进而提升预测的准确率。在 skin 数据集上,AdaBoost\_DW 算法的分类准确率比原 AdaBoost 算法高出许多,这是因为该数据集中的数据呈现明显局部分布;而在局部分布不是非常明显的数据集上,针对待测样本设置的动态权重值和静态权重差别很小,这就导致 AdaBoost\_DW 算法的分类准确率相对于原 AdaBoost 算法没有多大的提升。

整体上看,AdaBoost\_DW 算法在7个数据集上预测准确率都排在第1位,只有在 Liver\_Patient、blood\_transfusion 和 Dota2S 数据集上预测准确率在第2位。与3种对比算法比较,AdaBoost\_DW 算法还是具有很大的优势。实验结果表明,对于有明显局部特征的数据集,针对待测样本设置的动态权重,可以解决某些弱分类器在整个数据上表现好但在某个局部分类不佳引起的预测准确率不高的问题。对于没有明显局部分布差异的数据集,动态权重的加入不会产生大的影响,所以本文提出的方法是有效的。

### 3 结束语

本文在原 AdaBoost 算法基础上,加入了能衡量每个弱分类器与待测样本适应性的动态权重,结合弱分类器在整个训练集上获得的静态权重,提出的双权重算法 AdaBoost\_DW,在多个经典的数据集上预测准确率相比于原 AdaBoost 算法有所提升。实验表明,在集成学习中,对于分布具有明显局部特征的数据集,在弱分类器组合投票阶段,加入能度量待测样本与弱分类器适应性的动态权重,可以提升算法预测准确率。AdaBoost\_DW 算法花费的时间与原 AdaBoost 算法相差不大,多花费的时间用于局部敏感哈希表的构建和待测样本近邻的查找、弱分类器动态权重的计算。本文提出的 AdaBoost\_DW 算法仅仅考虑了二分类问题,后续工作可将其推广到多类别分类问题。

#### 参考文献:

- [1] BREIMAN L. Bagging predictors[J]. Machine Learning, 1996,24(2):123-140.
- [2] SCHAPIRE R E. The strength of weak learnability [J]. Machine Learning,1990,5(2):197-227.
- [3] VALIANT L G. A theory of the learnable [J]. Communications of ACM,1984,27(11):1134-1142.
- [4] YOAV F, ROBERT E S. A decision-theoretic generalization of online learning and an application to boosting [J]. Journal of Computer and System Sciences,1997,55(1):23-37.
- [5] 金龙. 基于集成半监督学习的标签噪声研究[D]. 西安:西安电子科技大学,2013.  
JIN L. Research on label noisy based ensemble semi-supervised learning[D]. Xi'an: Xidian University,2013.
- [6] LU J N, HU H P, BAI Y P. Generalized radial basis function neural network based on an improved dynamic particle swarm optimization and AdaBoost algorithm[J]. Neurocomputing, 2015,152:305-315.
- [7] LIAO S W, CHEN Y. A kind of improved AdaBoost algorithm[C]// International Conference on Intelligent Computation Technology and Automation. Changsha, China: IEEE, 2015:16-18.
- [8] ZHANG T, LI J J, JIA W J. Fast and robust occluded face detection in ATM surveillance[J]. Pattern Recognition Letters, 2018,107(1):33-40.
- [9] WANG L L, FU Z L, TAO P. Four-chamber plane detection in cardiac ultrasound images based on improved imbalanced AdaBoost algorithm[C]// IEEE International Conference on Cloud Computing and Big Data Analysis. Chengdu, China: IEEE, 2016:299-303.
- [10] WU S, NAGAHASHI H. Parameterized AdaBoost: introducing a parameter to speed up the training of real AdaBoost[J]. IEEE Signal Processing Letters, 2014, 21(6): 687-691.
- [11] TIAN Y, WANG X. SVM ensemble method based on improved iteration process of Adaboost algorithm [C]// 2017 29th Chinese Control and Decision Conference. Chongqing, China: IEEE, 2017:4026-4032.
- [12] GAO Y L, GAO F. Edited AdaBoost by weighted KNN [J]. Neurocomputing, 2010,73(16):3079-3088.
- [13] ASIM K M, IDRIS A, IQBAL T, et al. Seismic indicators based earthquake predictor system using genetic programming and AdaBoost classification[J]. Soil Dynamics & Earthquake Engineering, 2018,111:1-7.

- [14] 熊霖,唐万梅. 基于 k-means++ 的多分类器选择分类研究[J]. 重庆师范大学学报(自然科学版),2018,35(6):88-96.  
XIONG L, TANG W M. Multiple Classifiers Selection Classification Based on k-means++[J]. Journal of Chongqing Normal University (Natural Science),2018,35(6):88-96.
- [15] LI X, WANG L, SUNG E. AdaBoost with SVM-based component classifiers [J]. Engineering Applications of Artificial Intelligence,2008,21(5):785-795.
- [16] 林思阳. 选择性 AdaBoost SVM 语音情感识别算法的研究[D]. 广州:华南理工大学,2014.  
LIN S Y. Research of selective AdaBoost SVM speech emotion recognition algorithm[D]. Guangzhou:South China University of Technology,2014.
- [17] DIDACI L, GIACINTO G, ROLI F. A study on the performances of dynamic classifier selection based on local accuracy estimation [J]. Pattern Recognition,2005,38(11):2188-2191.
- [18] INDYK P, MOTWANNI R. Approximate nearest neighbor: towards removing the curse of dimensionality [EB/OL]. [2019-09-25]. <https://www.cs.princeton.edu/courses/archive/spring04/cos598B/bib/IndykM-curse.pdf>.
- [19] ANDONI A, INDYK P, LAARHOVEN T, et al. Practical and optimal LSH for angular distance [EB/OL]. (2015-09-09)[2019-09-25]. <https://arxiv.org/abs/1509.02897>.
- [20] HASTIE T, ROSSET S. Multi-class AdaBoost [J]. Statistics & Its Interface,2009,2(3):349-360.
- [21] CHANG C C, LIN C J. LIBSVM: a library for support vector machines. ACM Transactions on Internet Systems and Technology,2011,2:1-27.
- [22] BREIMAN L. Random Forests [J]. Machine Learning,2001,45(1):5-32.
- [23] BERNARD S, ADAM S, HEUTTE L. Using random forests for handwritten digit recognition[C]//International Conference on Document Analysis and Recognition, Parana, Brazil: IEEE,2009:1043-1047.

## An Improved Adaboost Algorithm by Add Dynamic Weight

XIONG Lin, TANG Wanmei

(College of Computer Science, Chongqing Normal University, Chongqing 401331, China)

**Abstract:** [Purposes] In order to solve the problem that the voting weight of weak classifier of AdaBoost algorithm is fixed and leads to low classification accuracy for different samples, a learning algorithm AdaBoost\_DW is proposed. [Methods] This algorithm retains all the weak classifiers generated in the training phase of the AdaBoost algorithm. In the prediction phase, add the dynamic weight that can measure the adaptability of the test sample and the weak classifier, the voting weight of the weak classifier is determined by two weights. [Findings] The experimental result on many data sets on UCI databases show that AdaBoost\_DW classification accuracy was higher than the AdaBoost algorithm in the most cases. [Conclusions] This proved that the dynamic weight can improve the classification accuracy of AdaBoost algorithm on each data set.

**Keywords:** AdaBoost algorithm; local region; adaptability; dynamic weight

(责任编辑 许 甲)