

基于多输出极限学习机的快速一致性分类器

王 迪[†], 王 萍, 石君志

(天津大学 电气自动化与信息工程学院, 天津 300072)

摘 要: 一致性分类器是建立在一致性预测基础上的分类器, 其输出结果具有很高的可靠性, 但由于计算框架的限制, 学习的时间往往较长. 为了加快学习速度, 首次将一致性预测与多输出极限学习机相结合, 提出基于两者的快速一致性分类算法. 该算法利用了极限学习机, 能够快速计算样本标签的留一交叉估计的特性, 极大地加快了学习速度. 算法复杂度分析表明, 所提算法的计算复杂度与多输出极限学习机的算法复杂度相同, 该算法继承了一致性预测的可靠性特征, 即预测的错误率能够被显著性水平参数所控制. 在 10 个公共数据集上的对比实验表明, 所提算法具有极快的计算速度, 且与其他常用一致性分类器相比, 该算法的平均预测标签个数在某些数据集上更少, 预测结果更有效.

关键词: 一致性预测; 刀切法一致性预测; 一致性分类器; 神经网络; 多输出极限学习机; 快速学习
中图分类号: TP273 **文献标志码:** A

A fast conformal classifier based on multi-output extreme learning machine

WANG Di[†], WANG Ping, SHI Jun-zhi

(School of Electrical and Information Engineering, Tianjin University, Tianjin 300072, China)

Abstract: A conformal classifier is a conformal prediction based classifier. Although the prediction is highly valid, the learning time of conformal classifiers is often very long due to the limitation of the computational framework. To make the conformal classifier learn faster, this paper firstly proposes an algorithm combining the conformal prediction with the multi-output extreme learning machine whose leave-one-out predictions on the training set can be computed efficiently. From the analysis of algorithm complexity, the computational complexity of the proposed algorithm is equivalent to that of the multi-output extreme learning machine. The experiments on ten public data sets show that the proposed our algorithm has fast computation speed and inherits the property of validity from conformal prediction, whose prediction error can be controlled by the significance level. The average number of labels per prediction of the proposed algorithm is lower than that of other common conformal classifiers on some data sets, which shows that of is more efficient in some applications.

Keywords: conformal prediction; jackknife conformal prediction; conformal classifier; neural network; multi-output extreme learning machine; fast learning

0 引 言

现如今,在大数据和计算机科学发展的推动下,机器学习与数据挖掘技术得到了蓬勃的发展,越来越多的应用领域开始应用甚至依赖机器学习与数据挖掘方法为人类服务,例如搜索引擎技术、商品推荐服务和垃圾邮件识别等.但对于诸如故障检测、医疗和金融等有着高风险低容错的应用,很少有成熟的具有独立决策能力的机器学习系统,因为在这些领域中,

预测的可信度是需要得到保证的.对于每一个样本,都需要机器学习算法给出高可信度的预测结果,但传统的机器学习算法对这样的要求无能为力,因为它们预测时往往只给出一个标签,且对预测结果没有可靠的保证.然而,作为近些年来新提出的机器学习框架,一致性预测(Conformal prediction)很好地解决了这一问题.

一致性预测框架的萌芽起源于 20 世纪 90 年代,

收稿日期: 2017-09-06; 修回日期: 2018-04-04.

基金项目: 天津市自然科学基金项目(14JCYBJC21800).

责任编辑: 曹进德.

作者简介: 王迪(1989—),男,博士生,从事模式识别、神经网络、一致性预测的研究;王萍(1955—),女,教授,博士生导师,从事模式识别、图像识别、运动对象跟踪等研究.

[†]通讯作者. E-mail: wangdi2015@tju.edu.cn.

是在柯尔莫戈洛夫复杂性和半监督支持向量机的启发下提出的,其主要的理论框架成型于2005年,之后在理论和应用中得到了研究和发展^[1-2]. 一致性预测作为一种学习框架,可以建立在传统的监督学习方法(称为底层算法)之上,可以看作是对底层算法的补充和提升. 为了保证预测的可靠性,一致性预测器在预测时并不是只给出一个预测标签,而是给出一个预测的标签集合. 例如对于分类问题,一致性预测器的预测结果是类别的集合,集合中可能包含多个类别;对于回归问题,预测结果可以是区间. 在样本是独立同分布的假设下,用一致性预测方法得到的标签集合包含真实标签的概率可以被人为指定,这使得预测结果有极高的可信度,对于高风险行业具有重要的参考价值.

虽然一致性预测器有很好的应用价值,但由于原始计算框架的限制,其计算过程较繁琐,且整体计算速度与底层算法是否高效息息相关,使得目前的一致性预测器很难有效地应用到需要快速处理数据的场合. 为解决一致性预测计算速度慢的问题,已有的工作主要从改进一致性预测的计算框架入手,代表性的工作有归纳一致性预测、交叉一致性预测^[3]和刀切法一致性预测^[4]等. 解决一致性预测计算问题的另一种方法是采用快速且学习能力强的底层算法. 本文即从这方面入手,首次尝试将一致性预测与多输出极限学习机(Multi-output extreme learning machine, M-ELM)^[5-6]相结合,并专注于解决分类问题,提出基于多输出极限学习机的快速一致性分类器算法. 该算法充分结合了刀切法一致性预测的特性以及极限学习机能够快速计算留一交叉验证估计的优点,在保持预测可靠性和有效性的基础上,大幅度提高了一致性分类的计算速度.

1 一致性预测

本节介绍原始一致性预测框架及其变种——刀切法一致性预测框架^[4].

设训练集为 $z^l = \{(\mathbf{x}_i, \mathbf{y}_i), i = 1, 2, \dots, l\}$. 其中: $\mathbf{x}_i \in \mathbf{R}^n$ 为输入向量, $\mathbf{y}_i \in \mathbf{R}$ 为其对应的标签. 记 $z^l_{(-m)} = \{(\mathbf{x}_i, \mathbf{y}_i), i = 1, 2, \dots, m-1, m+1, \dots, l\}$.

一致性预测的目标是对于一个新来的测试输入 \mathbf{x}_{l+1} , 输出一个标签的集合, \mathbf{y}_{l+1} 属于该集合的概率大于 $1 - \epsilon$, 其中 ϵ 为人为指定的参数, 用来控制预测的错误率, 称为显著性水平.

1.1 原始一致性预测框架

原始一致性预测构造预测标签集合的方法如下: 首先对每一个可能的标签 $\tilde{\mathbf{y}}$, 生成增广数据集

$z^l \cup \{(\mathbf{x}_{l+1}, \tilde{\mathbf{y}})\}$, 然后计算 $l+1$ 个一致性得分

$$\alpha_i^{\tilde{\mathbf{y}}} = A(\{z^l_{(-i)}, (\mathbf{x}_{l+1}, \tilde{\mathbf{y}})\}, (\mathbf{x}_i, \mathbf{y}_i)),$$

$$\alpha_{l+1}^{\tilde{\mathbf{y}}} = A(z^l, (\mathbf{x}_{l+1}, \tilde{\mathbf{y}})),$$

其中 $i = 1, 2, \dots, l$. $A(S, z)$ 是一个度量数据 z 与数据集 S 的一致性函数, 被称为一致性得分^[2]. 该函数值越大, 说明 z 与数据集 S 的一致性越强, $A(S, z)$ 的选取往往与具体的应用有关.

对于每一个 $\tilde{\mathbf{y}}$, 计算完一致性得分后还要计算其 p 值, 公式如下:

$$p^{\tilde{\mathbf{y}}} = \frac{|\{i \in \{1, 2, \dots, l\} | \alpha_i^{\tilde{\mathbf{y}}} \leq \alpha_{l+1}^{\tilde{\mathbf{y}}}\}| + 1}{l + 1}.$$

对于给定的显著性水平 ϵ , 一致性预测输出的预测集合为 $\{\tilde{\mathbf{y}} | p^{\tilde{\mathbf{y}}} > \epsilon\}$. 根据一致性预测的理论结果, 若数据满足独立同分布的假设, 则 \mathbf{x}_{l+1} 的标签在 $\{\tilde{\mathbf{y}} | p^{\tilde{\mathbf{y}}} > \epsilon\}$ 中的概率大于 $1 - \epsilon$.

一致性预测的计算过程实际上运用了假设检验的思想. 对于每一个可能的 $\tilde{\mathbf{y}}$, $(\mathbf{x}_{l+1}, \tilde{\mathbf{y}})$ 和训练数据集 z^l 通过函数 $A(S, z)$ 计算得到了一个与 $(\mathbf{x}_{l+1}, \tilde{\mathbf{y}})$ 有关一致性得分的分布. 因为数据是满足独立同分布的, 且 $\alpha_{l+1}^{\tilde{\mathbf{y}}}$ 作为一致性得分应该越大越好, 所以 $\alpha_{l+1}^{\tilde{\mathbf{y}}}$ 在一致性得分分布中的前 $100(1 - \epsilon)\%$ 时, 一致性预测不拒绝 $\mathbf{y}_{l+1} = \tilde{\mathbf{y}}$ 的假设, 反之则拒绝. 这样, 最终得到的预测集合 $\{\tilde{\mathbf{y}} | p^{\tilde{\mathbf{y}}} > \epsilon\}$ 为所有通过检验的 $\tilde{\mathbf{y}}$.

对于原始一致性预测框架, 每计算一次 $A(S, z)$, 则需要训练一次底层算法, 因此当可能的标签数较多甚至为无穷时, 原始计算框架的计算过程将非常耗时. 为了解决原始计算框架计算性能低下的问题, 一些改进的计算框架被提出, 包括归纳一致性预测、交叉一致性预测和刀切法一致性预测. 由于刀切法一致性预测的计算框架用到了训练样本的留一交叉估计值, 而极限学习机可以快速地计算留一交叉估计, 两者结合可以使整个算法的计算速度得到巨大的提升, 因此本文将两者结合, 并用于分类问题. 下面介绍刀切法一致性预测.

1.2 刀切法一致性预测

刀切法一致性预测是最近被提出的原始一致性预测的变种, 其计算速度快于原始计算框架. 该计算框架得到预测集合的过程用到了底层算法对训练数据的留一交叉验证估计, 故被称为刀切法. 刀切法一致性预测的计算过程^[4]如下.

算法1 刀切法一致性预测.

输入: 训练数据 z^l , 显著性水平 ϵ , 分类算法 μ , 函数 $\hat{A}(m, n)$, 其中 $m, n \in \mathbf{R}^k$;

输出: 对于每个 $\mathbf{x} \in \mathbf{R}^n$, 给出预测标签集合

$C(\mathbf{x})$.

Step 1: 在训练集上用分类算法 μ 得到分类器 $\hat{\mu}$;

Step 2: 对每个训练样本 \mathbf{z}_i , 在训练集上计算其留一交叉估计 $\hat{\mu}_i$, 并利用该值得到该样本的一致性得分估计 $A_i = \hat{A}(\hat{\mu}_i, \mathbf{y}_i)$;

Step 3: 找出集合 $\{A_i, i = 1, 2, \dots, l\}$ 中第 q 个最大的值并记为 d , 其中 $q = \lceil l(1 - \epsilon) \rceil$;

Step 4: 对于每个 $\mathbf{x} \in \mathbf{R}^n$, 预测的标签集合为 $C(\mathbf{x}) = \{\tilde{\mathbf{y}} | \hat{A}(\hat{\mu}(\mathbf{x}), \tilde{\mathbf{y}}) \leq d\}$.

算法1中的 $\hat{A}(\mathbf{m}, \mathbf{n})$ 是将标签估计 \mathbf{m} 和标签 \mathbf{n} 转换成一致性得分估计的函数, 其形式可以根据分类器的特点而定.

刀切法一致性预测不同于原始一致性预测的主要地方在于, 前者不是对每个可能的 $\mathbf{y}_{l+1} = \tilde{\mathbf{y}}$, 给出一个与 $(\mathbf{x}_{l+1}, \tilde{\mathbf{y}})$ 有关的一致性得分分布进而计算 $p\tilde{\mathbf{y}}$, 而是在得到新数据自变量 \mathbf{x}_{l+1} 之前, 先利用训练数据集、留一交叉估计和函数 $\hat{A}(\mathbf{m}, \mathbf{n})$ 计算一致性得分的经验分布, 该分布只与训练集有关. 当预测 \mathbf{x}_{l+1} 的标签集合时, 对于每一个 $\mathbf{y}_{l+1} = \tilde{\mathbf{y}}$, 用 $\hat{A}(\hat{\mu}(\mathbf{x}_{l+1}), \tilde{\mathbf{y}})$ 计算一致性得分的估计, 然后保留那些使得该估计排在分布前 $100(1 - \epsilon)\%$ 的 $\tilde{\mathbf{y}}$.

2 多输出极限学习机分类器

极限学习机是一种快速的训练单隐层前馈神经网络的算法. 其主要特点是隐层神经元个数很多, 且隐层参数是随机产生的, 隐层到输出层的参数一般用最小二乘方法求解. 由于ELM的隐层参数不参与训练, 且需要人为调整的超参数较少, 使得其比BP神经网络和支持向量机的学习速度更快, 需要的人为干预更少, 且在一些应用中与后两者有相同甚至更好的泛化能力^[7-8]. 这些优点吸引了许多学者对其进行了相关的理论研究、模型扩展和应用实践^[9-12].

将极限学习应用到分类问题时, 主要有单输出和多输出两种模型. 由于多输出的极限学习机分类器(M-ELM)在许多数据集都有很好的表现, 本文使用多输出极限学习分类器作为一致性预测的底层学习算法.

设分类问题的类别数目为 k , 则数据 $\mathbf{z}_i = (\mathbf{x}_i, \mathbf{y}_i)$ 的标签 \mathbf{y}_i 为一 k 维向量. 若该数据属于第 j 类, 则 \mathbf{y}_i 的第 j 个元素 $\mathbf{y}_i^{(j)}$ 为1, 其余元素为0.

多输出极限学习机分类器的输出函数的形式为

$$\mathbf{f}(\mathbf{x}) = \mathbf{h}(\mathbf{x})\boldsymbol{\beta},$$

其中

$$\mathbf{h}(\mathbf{x}) = \frac{1}{\sqrt{L}} [g(\mathbf{w}_1^T \mathbf{x} + b_1), \dots, g(\mathbf{w}_L^T \mathbf{x} + b_L)],$$

$g(\mathbf{w}^T \mathbf{x} + b)$ 是参数为 \mathbf{w} 和 b 的激活函数, $\boldsymbol{\beta}$ 是 $L \times k$ 维的输出权重矩阵.

根据文献[5], 对于给定的 L 、 λ 和 \mathbf{z}^l , M-ELM 首先随机生成参数 \mathbf{w}_i 和 b_i , 然后通过求解如下的优化问题得到 $\boldsymbol{\beta}$:

$$\begin{aligned} \min \quad & \frac{1}{2} \|\boldsymbol{\beta}\|^2 + \lambda \frac{1}{2} \sum_{i=1}^l \|\boldsymbol{\xi}_i\|^2; \\ \text{s.t.} \quad & \mathbf{h}(\mathbf{x}_i)\boldsymbol{\beta} = \mathbf{y}_i - \boldsymbol{\xi}_i. \end{aligned}$$

求解以上问题之后, 便可得到M-ELM的输出函数

$$\hat{\mathbf{f}}(\mathbf{x}) = \mathbf{h}(\mathbf{x}) \left(\frac{\mathbf{I}}{\lambda} + \mathbf{H}^T \mathbf{H} \right)^{-1} \mathbf{H}^T \mathbf{Y}. \quad (1)$$

其中

$$\mathbf{H} = \begin{bmatrix} \mathbf{h}(\mathbf{x}_1) \\ \mathbf{h}(\mathbf{x}_2) \\ \vdots \\ \mathbf{h}(\mathbf{x}_l) \end{bmatrix}, \quad \mathbf{Y} = \begin{bmatrix} \mathbf{y}_1^T \\ \mathbf{y}_2^T \\ \vdots \\ \mathbf{y}_l^T \end{bmatrix}.$$

记 $\hat{\mathbf{f}}^{(j)}(\mathbf{x})$ 为M-ELM的输出函数 $\hat{\mathbf{f}}(\mathbf{x})$ 对测试输入 \mathbf{x} 的第 j 个输出值, 则 \mathbf{x} 被M-ELM分到第 $\arg \max_j (\hat{\mathbf{f}}^{(j)}(\mathbf{x}))$ 类.

在学习过程中, M-ELM 首先将样本转换到一个 L 维的向量空间, 然后利用线性回归中的岭回归方法求解在该 L 维空间下的输出权重矩阵 $\boldsymbol{\beta}$. 由于使用了岭回归, 使得M-ELM对于训练样本的留一交叉估计存在闭式解, 并可以很快得到.

记 $\hat{\mathbf{f}}_i$ 为M-ELM在训练集得到的分类器和第 i 个样本的留一交叉估计向量, hat_{ii} 为 $\mathbf{H} \left(\frac{\mathbf{I}}{\lambda} + \mathbf{H}^T \mathbf{H} \right)^{-1} \mathbf{H}^T$ 的第 i 行第 i 列的值, 则由文献[13]可得如下公式:

$$\hat{\mathbf{f}}_i = \frac{\hat{\mathbf{f}}(\mathbf{x}_i) - \mathbf{y}_i \times \text{hat}_{ii}}{1 - \text{hat}_{ii}}. \quad (2)$$

由式(2)可以看出, M-ELM 只需在训练集中训练一次即可得到 l 个训练数据的留一交叉估计. 此公式和M-ELM自身的快速学习能力使得本文提出的一致性分类器具有很快的学习速度.

3 基于多输出极限学习机的快速一致性分类器

为了将算法1与M-ELM相结合, 还需要确定函数 $\hat{A}(\mathbf{m}, \mathbf{n})$ 的类型. 由于M-ELM是最小化最小二乘误差, 根据文献[14], 对于回归问题的单输出ELM而言, 随着数据量的增大, ELM的输出函数逐渐逼近并收敛到 y 对 \mathbf{x} 的条件期望函数 $f_\rho(\mathbf{x}) = \mathbb{E}(y|\mathbf{x})$. 因此, 当 y 的取值为0或1时, ELM的输出函数将收敛到 $y = 1$ 对 \mathbf{x} 的条件概率, 即 $f_\rho(\mathbf{x}) = P(y = 1|\mathbf{x})$.

由于本文M-ELM的类别标签采用了0和1进行

二值编码,对于式(1)的第 j 个函数 $\hat{f}^{(j)}(\mathbf{x})$,实际上相当于只用 \mathbf{Y} 的第 j 列训练得到的单输出ELM,并且当训练数据属于第 j 类时,对应的标签为1,其余情况下为0.这样,根据上文的分析,当样本足够大时, $\hat{f}^{(j)}(\mathbf{x})$ 的值近似等于 $P(\mathbf{y}^{(j)} = 1|\mathbf{x})$,即近似等于给定 \mathbf{x} 的情况下,该数据属于第 j 类的条件概率.

根据文献[15],当样本足够大时,使用 $P(\mathbf{y}^{(j)} = 1|\mathbf{x})$ 作为一致性得分是有效的,可以使得输出集合所含标签数的平均个数尽可能的少(Theorem 1).

由以上分析,本文利用M-ELM的第 j 个输出近似等于 $P(\mathbf{y}^{(j)} = 1|\mathbf{x})$,且使用 $P(\mathbf{y}^{(j)} = 1|\mathbf{x})$ 作为一致性得分具有一定的最优性的特点,构造 $\hat{A}(\mathbf{m}, \mathbf{n})$.即令 $\hat{A}(\mathbf{m}, \mathbf{n}) = \mathbf{m}^{(\arg \max_j (\mathbf{n}^{(j)}))}$,其中 $\mathbf{n}^{(j)}$ 为 \mathbf{n} 的第 j 个分量.该式的含义是:若 \mathbf{n} 的第 j 个元素最大,则 \hat{A} 的值即为 $\mathbf{m}^{(j)}$.如此,对于数据 $(\mathbf{x}_i, \mathbf{y}_i)$, $\hat{A}(\hat{\mu}(\mathbf{x}_i), \mathbf{y}_i) = \hat{\mu}(\mathbf{x}_i)^{(\arg \max_j (\mathbf{y}^{(j)}))}$,当 $\mathbf{y}^{(j_1)} = 1$ 时, $\hat{A}(\hat{\mu}(\mathbf{x}_i), \mathbf{y}_i) = \hat{\mu}(\mathbf{x}_i)^{(j_1)}$,即近似为 $P(\mathbf{y}^{(j_1)} = 1|\mathbf{x})$.

综上所述,将M-ELM与刀切法一致性预测相结合,并令 $\hat{A}(\mathbf{m}, \mathbf{n}) = \mathbf{m}^{(\arg \max_j (\mathbf{n}^{(j)}))}$,即为本文提出的快速一致性预测算法.具体细节见如下算法.

算法2 快速一致性预测算法.

输入:训练数据 \mathbf{z}^l ,显著性水平 ϵ ,隐节点个数 L ,正则化参数 λ ;

输出:对于每个 $\mathbf{x} \in \mathbf{R}^n$,给出预测标签集合 $C(\mathbf{x})$.

Step 1:在训练集上训练M-ELM,得到输出函数(1);

Step 2:利用式(2)得到训练样本的留一交叉估计 \hat{f}_i ,进而得到一致性得分估计 $A_i = \hat{A}(\hat{f}_i, \mathbf{y}_i) = \hat{f}_i^{(\arg \max_j (\mathbf{y}_i^{(j)}))}$;

Step 3:找出集合 $\{A_i, i = 1, 2, \dots, l\}$ 中第 q 个最大的值并记为 d ,其中 $q = \lceil l(1 - \epsilon) \rceil$;

Step 4:对于每个 $\mathbf{x} \in \mathbf{R}^n$,预测的标签集合为 $C(\mathbf{x}) = \{\tilde{\mathbf{y}} | \hat{A}(\hat{f}(\mathbf{x}), \tilde{\mathbf{y}}) \geq d\}$.

确定隐节点个数 L 和正则化参数 λ 后,算法2前两步的计算复杂度为 $\Theta(l)$.若使用线性时间排序算法^[16],则Step 3的时间复杂度为 $\Theta(l)$.故对于算法2的学习阶段,总计算复杂度为 $\Theta(l)$,与计算式(1)的计算复杂度相同.这说明算法2有很快的学习速度,克服了现有的一致性预测学习速度慢的缺点.

4 实验

一致性分类器的性能好坏主要通过两个方面来衡量,即可靠性和有效性^[2].一致性分类器的错误率是指预测的标签集合不包含真实标签的概率,若该

错误率不大于或接近显著性水平,则称该一致性分类器是可靠的,即其错误率可以被显著性水平参数所控制.对于输出标签集合的预测器,可靠性是被优先考虑的.在可靠性条件满足的情况下,有效性用于衡量预测器输出的集合是否包含足够的信息,如果一个集合预测器预测的标签个数越少,则该预测器给出的信息越多,越有效.有效性一般用测试集上预测标签平均个数来衡量,该值越小,预测器越有效^[15].下面通过两个实验考察本文算法.

4.1 显著性水平参数在本文算法中的作用

为了解显著性水平 ϵ 对本文提出的一致性分类器的可靠性和有效性的影响,使用UCI数据库中的Ionosphere数据集进行实验.实验取 $L = 1000, \lambda = 1$.将该数据集随机分成两份,其中70%的数据作为训练集,余下的数据作为测试集.图1为本文算法在测试集上的错误率随 ϵ 变化的关系图.从图1可以看出,两者关系曲线在对角线的下方或非常靠近对角线,这说明 ϵ 可以很好地控制预测的错误率.

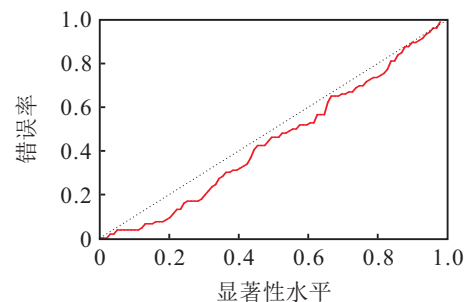


图1 错误率与显著性水平的关系

图2是本文算法在测试集上预测标签的平均个数随 ϵ 变化的图像.当显著性水平很低时,为了保证预测错误率也很低,算法倾向于输出多个标签;而随着显著性水平的升高,预测标签的平均个数有所减少,预测得到的结果更清晰,但预测错误的风险更高.

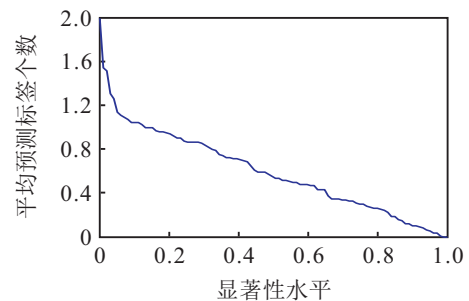


图2 平均预测标签个数和显著性水平的关系

图2的特性说明,通过调节显著性水平参数,可以使得最终预测的集合为单标签集合,即让一致性分类器只输出单一标签.因此一致性分类器也可用于传统分类问题,此时即将单标签集合中的标签作为测

试输入的分类结果. 对于该分类结果, 一致性分类器可以给出分类的可信度指标, 具体的指标构造方法参见文献[2].

4.2 与基于决策树和支持向量机的一致性分类器的对比实验

本小节实验旨在验证本文算法的可靠性, 并与文献[17]中提出的基于CART的原始一致性分类器(CART-TCP)、基于CART的归纳一致性分类器(CART-ICP)、基于SVM的原始一致性分类器(SVM-TCP)和基于SVM的归纳一致性分类器(SVM-ICP)进行比较, 以证明本文算法在加快了计算速度的同时, 并不影响预测的有效性.

实验在10个UCI公共数据集^[18]上进行. 实验设计方案和数据集的使用参考了文献[17]. 所有数据集上的实验结果都来源于 5×10 重交叉验证的平均结果.

实验首先测试了在显著性水平 $\epsilon = 0.05$ 和 $\epsilon = 0.01$ 的情况下, 本文提出的一致性预测算法2的错误率是否被 ϵ 所控制, 实验结果见表1. 结果表明, 参数 ϵ 确实可以有效地控制预测的错误率, 使得预测结果的可靠性得到保证.

表1 本文算法在各数据集上的错误率

数据集	$\epsilon = 0.05$	$\epsilon = 0.01$
breast-w	0.051	0.011
credit-a	0.043	0.011
credit-g	0.044	0.006
diabetes	0.047	0.010
haberman	0.048	0.004
heart-s	0.053	0.011
Ionosphere	0.050	0.013
kr-vs-kp	0.049	0.009
liver-disorders	0.045	0.009
sonar	0.042	0.011

接下来的实验对比了算法2与基于CART和SVM的一致性分类器的平均预测标签个数和在各个数据集上的平均运行时间. 在同样的错误率前提下, 平均预测标签个数越小, 预测的结果信息量越大, 预测器的性能越好, 预测越有效. 实验结果见表2~表4. 其中: 各表中每行的最小值被加粗; 各表的倒数第二行是各一致性分类器在10个数据集上的错误率或平均运行时间的平均值. 对于表2, 在每个数据集上, 首先通过将错误率由小到大排序, 可以得到各一致性分类器在该数据集上的序数, 即秩; 然后将每个一致性分类器在各数据集上的所有秩取平均, 即为表2最后一行的平均秩. 同理也可计算表3和表4的平均秩, 平均秩即各一致性分类器在各数据集表现性能排序

的平均值, 可用于衡量一个一致性分类器的整体性能^[17].

表2 各算法的平均预测标签个数($\epsilon = 0.05$)

	CART-TCP	CART-ICP	SVM-TCP	SVM-ICP	算法2
breast-w	1.518	1.257	0.990	0.985	1.034
credit-a	1.898	1.743	1.776	1.765	1.713
credit-g	1.905	1.848	1.850	1.824	1.749
diabetes	1.895	1.844	1.864	1.819	1.698
haberman	1.841	1.851	1.606	1.678	1.772
heart-s	1.894	1.816	1.898	1.902	1.668
Ionosphere	1.892	1.588	1.038	1.056	1.179
kr-vs-kp	1.892	0.948	1.016	1.033	1.038
liver-disorders	1.862	1.879	1.848	1.860	1.816
sonar	1.900	1.830	1.647	1.721	1.650
均值	1.541	1.384	1.294	1.304	1.276
平均秩	4.500	3.333	2.500	2.667	2.000

表3 各算法的平均预测标签个数($\epsilon = 0.01$)

	CART-TCP	CART-ICP	SVM-TCP	SVM-ICP	算法2
breast-w	1.588	1.841	1.229	1.255	1.336
credit-a	1.982	1.953	1.935	1.931	1.885
credit-g	1.980	1.969	1.972	1.957	1.932
diabetes	1.979	1.968	1.982	1.964	1.881
haberman	1.956	1.968	1.920	1.934	1.976
heart-s	1.979	1.952	1.977	1.982	1.936
Ionosphere	1.976	1.917	1.331	1.543	1.510
kr-vs-kp	1.981	1.070	1.255	1.205	1.229
liver-disorders	1.961	1.976	1.947	1.970	1.951
sonar	1.981	1.970	1.875	1.921	1.902
均值	1.614	1.549	1.452	1.472	1.462
平均秩	4.250	3.500	2.333	2.750	2.167

表4 各算法在数据集上的平均运行时间 s

	CART-TCP	CART-ICP	SVM-TCP	SVM-ICP	算法2
breast-w	236.7	0.006	9976	0.237	0.110
credit-a	304.2	0.007	1.203×10^4	0.281	0.107
credit-g	885.5	0.010	3.946×10^4	0.439	0.138
diabetes	432.1	0.008	9955	0.188	0.176
haberman	45.89	0.005	977.8	0.115	0.074
heart-s	39.81	0.006	603.0	0.092	0.064
Ionosphere	68.06	0.006	1380	0.122	0.065
kr-vs-kp	12010	0.013	2.158×10^6	2.346	0.454
liver-disorders	71.14	0.007	1017	0.094	0.065
sonar	46.25	0.012	430.7	0.109	0.057
均值	1178	0.007	1.8615×10^5	0.335	0.109
平均秩	3.917	1.167	4.833	3.000	2.083

表2和表3的结果表明, 本文算法的平均预测标签个数在一些数据集上小于前4种算法, 说明在这些数据集上本文算法比前4种算法更有效. 从表2和表3还可以看出, 本文算法在显著性水平参数为0.05和0.01的情况下, 平均秩均为最小, 说明本文算法在整

体上表现最优. 表4中, 本文算法在各个数据集上的平均运行时间只高于 CART-ICP 算法, 而小于其他算法, 说明本文算法有极快的计算速度. 考虑到表2和表3中 CART-ICP 算法的预测标签平均个数一般较大, 故本文算法在同时考虑计算速度和预测标签平均个数上具有比其他算法更优的性能.

综合以上实验结果可知, 本文算法在加快一致性分类器计算速度的同时, 保持了可靠性和有效性, 可以在快速处理数据的同时给出可信度很高的预测结果.

5 结论

本文在多输出极限学习机和刀切法一致性预测的基础上提出了基于极限学习机的快速一致性分类器. 通过算法复杂度的分析和实际数据集的检验, 验证了本文算法在保持了一致性预测可靠性和有效性的前提下, 大幅度加快了一致性分类器的学习速度. 对于如故障检测、量化交易等需要实时数据处理且对预测结果的可信度要求很高的应用领域, 本文的算法具有极高的应用价值.

参考文献(References)

- [1] Vovk V, Gammerman A, Shafer G. Algorithmic learning in a random world[M]. New York: Springer Science Business Media, 2005: 1-16.
- [2] Balasubramanian V, Ho S S, Vovk V. Conformal prediction for reliable machine learning: Theory, adaptations and applications[M]. Waltham: Morgan Kaufmann Publishers Inc, 2014: 1-46.
- [3] Vovk V. Cross-conformal predictors[J]. Annals of Mathematics and Artificial Intelligence, 2015, 74(1/2): 9-28.
- [4] Lei J, G'Sell M, Rinaldo A, et al. Distribution-free predictive inference for regression[J]. J of the American Statistical Association, 2017, DOI: 10.1080/01621459.2017.1307116.
- [5] Huang G B, Zhou H, Ding X, et al. Extreme learning machine for regression and multiclass classification[J]. IEEE Trans on Systems, Man, and Cybernetics, Part B (Cybernetics), 2012, 42(2): 513-529.
- [6] Huang G B. What are extreme learning machines? filling the gap between frank rosenblatts dream and john von neumanns puzzle[J]. Cognitive Computation, 2015, 7(3): 263-278.
- [7] Huang G B, Zhu Q Y, Siew C K. Extreme learning machine: Theory and applications[J]. Neurocomputing, 2006, 70(1): 489-501.
- [8] Chorowski J, Wang J, Zurada J M. Review and performance comparison of SVM- and ELM-based classifiers[J]. Neurocomputing, 2014, 128(5): 507-516.
- [9] Huang G, Huang G B, Song S, et al. Trends in extreme learning machines[J]. Neural Networks, 2015, 61(C): 32-48.
- [10] 徐嘉明, 张卫强, 杨登舟, 等. 基于流形正则化极限学习机的语种识别系统[J]. 自动化学报, 2015, 41(9): 1680-1685.
(Xu J M, Zhang W Q, Yang D Z, et al. Manifold regularized extreme learning machine for language recognition[J]. Acta Automatica Sinica, 2015, 41(9): 1680-1685.)
- [11] 杜占龙, 李小民, 席雷平, 等. 改进的灵敏度剪枝极限学习机[J]. 控制与决策, 2016, 31(2): 249-255.
(Du Z L, Li X M, Xi L P, et al. Improved sensitivity-analysis based pruning extreme learning machine[J]. Control and Decision, 2016, 31(2): 249-255.)
- [12] 王超, 王建辉, 顾树生, 等. 改进式混合增量极限学习机算法[J]. 控制与决策, 2015, 30(11): 1981-1986.
(Wang C, Wang J H, Gu S S, et al. Improved hybrid incremental extreme learning machine algorithm[J]. Control and Decision, 2015, 30(11): 1981-1986.)
- [13] Shao Z, Meng J E. Efficient leave-one-out cross-validation-based regularized extreme learning machine[J]. Neurocomputing, 2016, 194: 260-270.
- [14] Liu X, Lin S, Fang J, et al. Is extreme learning machine feasible? A theoretical assessment (part I)[J]. IEEE Trans on Neural Networks and Learning Systems, 2014, 26(1): 21-34.
- [15] Vovk V, Fedorova V, Nouretdinov I, et al. Criteria of efficiency for conformal prediction[C]. Symposium on Conformal and Probabilistic Prediction with Applications. Switzerland: Springer Int Publishing, 2016: 23-39.
- [16] Cormen T H, Leiserson C E, Rivest R L, et al. Introduction to algorithms[M]. 3rd ed. London: The MIT Press, 2009: 191-200.
- [17] Linusson H, Johansson U, Boström H, et al. Efficiency comparison of unstable transductive and inductive conformal classifiers[M]. Heidelberg: Springer Berlin, 2014: 261-270.
- [18] Bache K, Lichman M. UCI machine learning repository[EB/OL]. (2013-01-01)[2014-03-25]. <http://archive.ics.uci.edu/ml>.

(责任编辑: 齐 霖)