面向概念漂移和类不平衡数据流的在线分类算法

陆克中,陈超凡,蔡 桓,吴定明
(深圳大学计算机与软件学院,广东深圳518060)

摘 要: 数据流是大数据的重要形式,数据流分类是数据挖掘的重要任务之一,该任务在现实生活中有着巨大的应用前景,因此得到了研究者们的广泛关注. 概念漂移和类不平衡是影响数据流分类性能的两个核心问题,但目前大多数算法都只考虑处理两者之一,并且大多数算法过于理想,只能在人工设置的数据流上才能发挥较好的性能,无法适用于复杂的真实数据流. 针对这一问题,提出了一种同时处理概念漂移和类不平衡复杂数据流的算法——具有自适应遗忘因子的加权在线顺序极限学习机集成算法. 该算法首先融合加权机制和遗忘机制,初步提出具有遗忘机制的加权在线顺序极限学习机算法. 为了更好地适应复杂数据流,进一步以初步算法为基分类器,设计包含自适应遗忘因子和概念漂移检测机制的在线集成策略. 大量仿真实验表明,所提算法在所有数据集上都取得了最佳的Gmean值,具有更好的概念漂移和类不平衡适应能力,表现出了更稳定、更平衡以及更准确的分类效果.

关键词: 数据流分类; 概念漂移; 类不平衡; 在线学习; 极限学习机
中图分类号: TP391 **文献标识码:** A **文章编号:** 0372-2112(2022)03-0585-13
电子学报URL:http://www.ejournal.org.cn **DOI:**10.12263/DZXB.20210094

Online Classification Algorithm for Concept Drift and Class Imbalance Data Stream

LU Ke-zhong, CHEN Chao-fan, CAI Huan, WU Ding-ming
(*College of Computer Science and Software Engineering of Shenzhen University, Shenzhen, Guangdong* 518060, *China*)

Abstract: Data stream is an important form of big data, and data stream classification is one of the most important tasks in data mining. This task finds wide application in our life, so it has been attracting great attention of researchers. Concept drift and class imbalance are two main issues that affect the performance of data stream classification algorithms. However, most solutions only address one of these two issues. Even worse, most algorithms can only achieve good performance on data streams under manual settings and cannot be applied to real complex data streams. To solve this problem, an ensemble algorithm of weighted online sequential extreme learning machine with adaptive forgetting factor is proposed to deal with both conceptual drift and imbalance on complex data streams. The proposed algorithm is a weighted online sequential limit learning machine that integrates a weighting mechanism and a forgetting mechanism. In order to adapt to complex data streams, an online integration strategy including adaptive forgetting factor and concept drift detection mechanism was designed as a classifier. Extensive simulation experiments show that the proposed algorithm achieves the best Gmean value on all data sets, has the ability to deal with concept drift and class imbalance, and presents stable, balanced and accurate classification effects.

Key words: data stream classification; concept drift; class imbalance; online learning; extreme learning machine

1 引言

随着信息技术的发展,各行各业中的数据出现爆炸性的增长,产生了海量数据,并且还在不断增加. 电子邮件、网络监控、股票预测、交通控制、传感器检测、信用卡交易和网络点击流等应用程序产生的数据流是一种新型的数据形式,具有快速、连续、多变、无限等特性[1]. 而且数据流的数据分布可能随时间发生改变,即存在概念漂移,因此模型需要不断地更新以适应新的

收稿日期:2021-01-12;修回日期:2021-12-14;责任编辑:梅志强
基金项目:国家自然科学基金项目(No.61502310);广东省自然科学基金(No.2019A1515011721, No.2019A1515011064);深圳市基础研究资助项目(No.20200806102941001)

数据流环境[2]. 概念漂移最早由 Jeffrey 在文献[3]中提出,近年来,对概念漂移数据流进行在线学习、实时分析吸引了研究人员的兴趣[4]. 而类不平衡问题则加剧了对概念漂移数据流进行学习分类的困难,并且可能导致在线学习(包括概念漂移检测)的性能严重降低[5]. 类不平衡在真实数据流中很常见,例如癌症诊断、垃圾邮件过滤、欺诈检测、计算机安全、图像识别、风险管理和故障诊断等往往存在多数类和少数类[6]. 由于数据的偏斜分布,传统的机器学习算法无法正确预测可能携带有用信息的少数类示例[7]. 因此,必须开发一种新的算法来解决概念漂移和类不平衡同时存在的问题.

近些年来,随着神经网络的迅速发展,研究人员已开始开发基于神经网络的数据流分类方法. 由新加坡南洋理工大学黄广斌教授团队提出的在线顺序极限学习机(Online Sequential Extreme Learning Machine, OS-ELM)[8]是一种增量学习神经网络算法,是黄教授之前提出的极限学习机(Extreme Learning Machine, ELM)算法[9]的在线学习方法. 该算法可以逐步更新分类模型而无需重新训练,相对于其他算法具有速度快、分类性能好的优势,完美满足数据流分类的要求. 因此,基于OSELM算法进行优化成为解决数据流分类问题的一个热门方向. 针对数据流的类不平衡问题,Mirza 等人于2013 年提出加权在线顺序极限学习机算法 WOSELM(Weight Online Sequential Extreme Learning Machine)[10],该算法基于代价敏感学习方法,根据不平衡率 IR(Imbalance Rate)对少数类进行加权,从而使分类器具有类不平衡适应能力. 2015 年 Mirza 等人进一步提出基于投票的加权在线顺序极限学习机算法 VWOSELM[11],该算法以 WOSELM 算法为基分类器,同时可以应用于多分类问题,实验表明相比原始的WOSELM 算法分类性能更好. 除了基于优化 OSELM 更新公式和集成方法外,也有一些研究使用数据采样、进化算法优化等. 由 Klikowski 等人于 2019 年提出的多采样随机子空间集合算法 MSRS(Multi Sampling Random Subspace)[12]是一种可以用于类不平衡非平稳数据流分类的基于块的集成方法. 2020 年 Zhu 等人则使用三种进化算法优化加权极限学习机来解决类不平衡问题[13],但仍是传统的批学习方式.

现有的基于数据采样的顺序学习方法通常以块的形式学习数据,即所谓的逐块学习,它们以类似于批处理学习方法的方式来处理类不平衡问题. 数据采样方法的局限性在于需要访问旧数据而不能轻松直接应用于一对一学习[14]. 因此,通常需要延迟分类模型的更新,直到接收到完整的数据块为止. 此外,在面对类不平衡数据流时,传统分类器的决策边界容易过度靠近少数类,在学习过程中容易忽略少数类的分类误差,而错分类一个少数类的代价往往高于错分类一个多数类的代价[15]. 大多数针对类不平衡的研究都只关注类不平衡问题,而忽略了类不平衡和概念漂移往往同时存在. 而且在进行分类任务前我们一般无法提前获知数据流不平衡率大小,不平衡率也可能发生改变. 特别是在复杂的真实数据集中,这大大增加了分类的难度. 因此如何提高复杂数据流的分类性能是论文的研究重点.

针对上面提到的问题,在FROSELM(Extreme Learning Machine based on Regularization and Forgetting factor)[16]和WOSELM[10]等算法的启发下,本文首先提出了一种具有遗忘因子的加权在线顺序极限学习机算法 FWOSELM(Weighted Online Sequential Extreme Learning Machine with Forgetting Factor),该算法融合了WOSELM 算法的加权机制和 FOSELM 算法的遗忘机制,从而能够同时具备概念漂移和类不平衡适应能力. 集成学习往往能在复杂分布的数据流上取得更优秀的分类性能[17],因此本文进一步提出了一种具有自适应遗忘因子的加权在线顺序极限学习机集成算法(Ensemble of Weighted Online Sequential Extreme Learning Machine with Adaptive Forgetting Factor, EAFWOSELM). 该算法首先使用FWOSELM作为基分类器,然后引入自适应遗忘因子和概念漂移检测机制以及自适应更新权重修正项,并将遗忘因子与混淆矩阵相结合,从而使分类器更关注最近到达的数据流,最后设计了在线集成策略. 算法使用递推公式实时增量更新模型,更符合在线学习的需求.

EAFWOSELM算法的创新性体现在三个方面:

(1)将 WOSELM 算法和 FOSELM 算法两者相融合,提出FWOSELM算法,推导出一个简单有效的在线学习阶段更新公式,可以同时处理数据流的概念漂移和类不平衡问题;

(2)将遗忘因子引入混淆矩阵,从而可以更关注最近到达的数据流,防止历史累积数据过度影响分类模型;

(3)引入自适应遗忘因子和概念漂移检测机制,使模型根据分类性能自适应更新遗忘因子以及基分类器的权重修正项、投票权重以及类别权重. 从而能够更少依赖人工干预,可以根据数据流的变化自适应学习.

2 相关工作

在本节中,我们将重点介绍 OSELM、FROSELM 和 WOSELM 三种算法. 为了简单起见,三种算法都考虑用于二分类问题,即只有单个输出节点.

2.1 OSELM 算法

OSELM 算法[8]分为两个阶段,在初始化阶段,假设

有 N_0 个训练样本 $\left(\boldsymbol{X}_i, y_i\right)$,其中 $\boldsymbol{X}_i=[x_{i1}, x_{i2}, \cdots, x_{iL}]^{\mathrm{T}}$,$L$ 表示数据流特征数,即 OSELM 的输入层节点数. 利用极限学习机 ELM 算法的思想,希望求得满足 $\|\boldsymbol{H}_0\boldsymbol{\beta}-\boldsymbol{Y}_0\|$ 最小的输出层权重 $\boldsymbol{\beta}_0$,其中

$$\boldsymbol{H}_0=\begin{bmatrix} g(a_1,b_1,x_1) & \cdots & g(a_L,b_L,x_1) \\ \vdots & \ddots & \vdots \\ g(a_1,b_1,x_{N_0}) & \cdots & g(a_L,b_L,x_{N_0}) \end{bmatrix} \tag{1}$$

而 $\boldsymbol{Y}_0=[y_1,\cdots,y_{N_0}]^{\mathrm{T}}$. 根据广义逆的计算方法,可以计算得 $\boldsymbol{\beta}_0$

$$\boldsymbol{\beta}_0=\boldsymbol{P}_0\boldsymbol{H}_0^{\mathrm{T}}\boldsymbol{Y}_0 \tag{2}$$

其中,$\boldsymbol{P}_0=(\boldsymbol{H}_0^{\mathrm{T}}\boldsymbol{H}_0)^{-1}$. 在线学习阶段,数据流被逐条处理,更新公式为

$$\boldsymbol{P}_{k+1}=\boldsymbol{P}_k-\frac{\boldsymbol{P}_k\boldsymbol{h}_{k+1}^{\mathrm{T}}\boldsymbol{h}_{k+1}\boldsymbol{P}_k}{1+\boldsymbol{h}_{k+1}\boldsymbol{P}_k\boldsymbol{h}_{k+1}^{\mathrm{T}}} \tag{3}$$

$$\boldsymbol{\beta}_{k+1}=\boldsymbol{\beta}_k+\boldsymbol{P}_{k+1}\boldsymbol{h}_{k+1}^{\mathrm{T}}(y_{k+1}-\boldsymbol{h}_{k+1}\boldsymbol{\beta}_k) \tag{4}$$

其中 $\boldsymbol{h}_{k+1}=\left[g\left(a_1,b_1,\boldsymbol{X}_{k+1}\right),\cdots,g\left(a_L,b_L,\boldsymbol{X}_{k+1}\right)\right]$

OSELM 具备了 ELM 的速度和泛化能力上的优点,从式(3)和式(4)可以看出,该模型的输出权重是根据最后一次的结果和新到达的数据进行迭代更新的,并且可以随着新数据的到来不断更新模型,而不是重新训练模型,数据一旦使用完毕即可丢弃;这种数据处理方式可以极大地降低算法的计算开销和内存,十分符合在线学习处理方式的要求. 因此,基于 OSELM 算法进行优化成为数据流分类研究的一个热门方向.

2.2　FROSELM 算法

FROSELM 算法[16]是由 Du 等人于 2015 年提出,该算法将将遗忘因子 FF(Forgetting Factor)方法和正则化技术引入 OSELM,根据实例的时间顺序分别为每个样本分配不同的权重,为最近到来的数据分配较高的权重,更加关注最近到来的数据,使得算法可以适应数据流的状态变化.

初始化阶段输出层权重 $\boldsymbol{\beta}_0$ 为

$$\boldsymbol{P}_0=(\boldsymbol{H}_0^{\mathrm{T}}\boldsymbol{H}_0+C\boldsymbol{I})^{-1} \tag{5}$$

$$\boldsymbol{\beta}_0=\boldsymbol{P}_0\boldsymbol{H}_0^{\mathrm{T}}\boldsymbol{Y}_0 \tag{6}$$

其中,C 为惩罚项系数,$\boldsymbol{I}$ 为单位矩阵. 在线学习阶段更新递推公式为

$$\boldsymbol{P}_{k+1}=\frac{\boldsymbol{P}_k}{\lambda}-\frac{\boldsymbol{P}_k\boldsymbol{h}_{k+1}^{\mathrm{T}}\boldsymbol{h}_{k+1}\boldsymbol{P}_k}{\lambda(\lambda+\boldsymbol{h}_{k+1}\boldsymbol{P}_k\boldsymbol{h}_{k+1}^{\mathrm{T}})} \tag{7}$$

$$\boldsymbol{\beta}_{k+1}=\boldsymbol{\beta}_k+\boldsymbol{P}_{k+1}\boldsymbol{h}_{k+1}^{\mathrm{T}}(y_{k+1}-\boldsymbol{h}_{k+1}\boldsymbol{\beta}_k) \tag{8}$$

其中 λ 为遗忘因子,当 λ 为 1 且 C 为 0 时,FROSELM 退化为原始 OSELM.

2.3　WOSELM 算法

加权在线顺序极限学习机 WOSELM 算法[10]是由 Mirza 于 2013 年提出. 该算法基于代价敏感学习方法,根据不平衡率 IR(Imbalance Rate)对少数类进行加权,从而保证分类器不会过度关注多数类. 初始化阶段输出权重 $\boldsymbol{\beta}_0$ 为

$$\boldsymbol{P}_0=(\boldsymbol{H}_0^{\mathrm{T}}\boldsymbol{W}_0\boldsymbol{H}_0)^{-1} \tag{9}$$

$$\boldsymbol{\beta}_0=\boldsymbol{P}_0\boldsymbol{H}_0^{\mathrm{T}}\boldsymbol{W}_0\boldsymbol{Y}_0 \tag{10}$$

其中,$\boldsymbol{W}_0=\mathrm{diag}\left(\left[w_1,w_2,\cdots,w_N\right]\right)$,$N$ 为训练阶段样本数量. 当 $y_k=0$ 时,类别权重 w_k 为不平衡率 IR. 当 $y_k=1$ 时,w_k 为 1. 在线学习阶段更新递推公式为

$$\boldsymbol{P}_{k+1}=\boldsymbol{P}_k-\frac{\boldsymbol{P}_k\boldsymbol{h}_{k+1}^{\mathrm{T}}\boldsymbol{h}_{k+1}\boldsymbol{P}_k}{w_{k+1}^{-1}+\boldsymbol{h}_{k+1}\boldsymbol{P}_k\boldsymbol{h}_{k+1}^{\mathrm{T}}} \tag{11}$$

$$\boldsymbol{\beta}_{k+1}=\boldsymbol{\beta}_k+\boldsymbol{P}_{k+1}\boldsymbol{h}_{k+1}^{\mathrm{T}}w_{k+1}(y_{k+1}-\boldsymbol{h}_{k+1}\boldsymbol{\beta}_k) \tag{12}$$

其中,当 y_{k+1} 为 0 时,w_{k+1} 为更新后计算得到的 IR,当 y_{k+1} 为 1 时,w_{k+1} 保持为 1. WOSELM 算法给少数类分类更高的权重,从而使分类器能够有效处理类不平衡数据流分类问题.

3　EAFWOSELM 算法

在面对类不平衡数据流时,传统分类器的决策边界容易过度靠近少数类,在学习过程中容易忽略少数类的分类误差,而错分类一个少数类的代价往往高于错分类一个多数类的代价. 此外,大多数针对类不平衡的研究都只关注类不平衡问题,而忽略了类不平衡和概念漂移往往同时存在. 针对这些问题,本文提出了 EAFWOSELM 算法. 本节将对 EAFWOSELM 算法的基本思想及其实现过程进行详细介绍.

3.1　算法基本思想

针对概念漂移和类不平衡同时存在的问题,首先融合 WOSELM 算法的加权机制和 FOSELM 算法的遗忘机制,提出了一种具有遗忘因子的加权在线顺序极限学习机算法 FWOSELM,从而能够同时具备概念漂移和类不平衡适应能力. 然后以所提出的 FWOSELM 算法作为基分类器,引入自适应遗忘因子和概念漂移检测机制,进一步提出了具有自适应遗忘因子的加权在线顺序极限学习机集成分类算法 EAFWOSELM. 该算法将遗忘因子与混淆矩阵相结合,集成分类器的混淆矩阵可以用作概念漂移检测和计算类别权重,而基分类器的混淆矩阵可以用来确定基分类器的投票权重和类别权重修正项. 根据类别权重和类别权重修正项可以确定每个基分类器的更新权重 W. 图 1 展示了本文所提出的 EAFOSELM 算法的整体框架.

3.2　基分类器 FWOSELM

针对同时具有概念漂移和类不平衡数据流,首先融合 WOSELM 算法的加权机制和 FOSELM 算法的遗忘机制提出 FWOSELM 算法,推导出一个基于 OSELM 的隐含层输出权重 $\boldsymbol{\beta}$ 更新公式,从而保证算法能同时处理

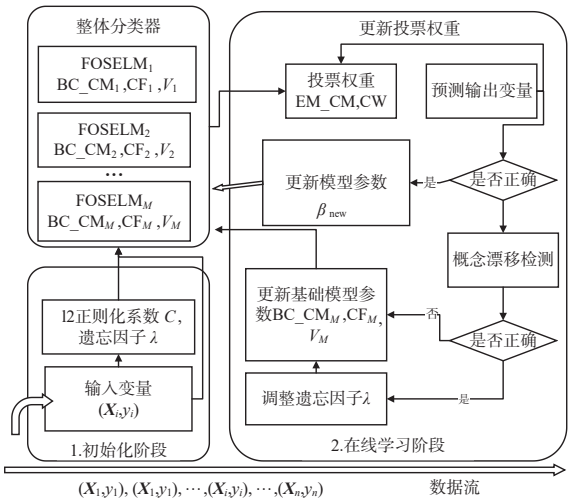


图1 EAFWOSELM算法的整体框架

概念漂移和类不平衡问题.

FWOSELM算法的预测公式可以表示为

$$\sum_{i}^{T}\boldsymbol{\beta}_i g(\boldsymbol{W}_i\boldsymbol{X}+b_i)=y \tag{13}$$

其中T为隐含层节点数

FWOSELM算法的代价函数可以表示为

$$\begin{aligned}J\left(\boldsymbol{\beta}_k\right)&=\sum_{i=1}^{k}w_i\lambda^{k-i}\left|e_i\right|^2+C\lambda^k\left\|\boldsymbol{\beta}\right\|_2^2\\&=\sum_{i=1}^{k}w_i\lambda^{k-i}\left|y_i-\boldsymbol{h}_i\boldsymbol{\beta}_k\right|^2+C\lambda^k\left\|\boldsymbol{\beta}\right\|_2^2\end{aligned} \tag{14}$$

其中w_i、λ和C分别为更新权重、遗忘因子以及正则化参数.w_i用来对类别进行加权,λ用来对新旧样本进行加权,C则是用来提高解的泛化能力.为了满足在线学习的要求,接下来进一步推导$\boldsymbol{\beta}_k$的递推公式.运用递归最小二乘法[18]对式(14)求解,可得

$$\boldsymbol{\beta}_k=\left(\boldsymbol{H}_k^{\mathrm{T}}\boldsymbol{W}_k\boldsymbol{H}_k+C\boldsymbol{I}\right)^{-1}\boldsymbol{H}_k^{\mathrm{T}}\boldsymbol{W}_kY_k \tag{15}$$

其中,$\boldsymbol{W}_k=\mathrm{diag}\left(\left[w_1,w_2,\cdots,w_k\right]\right)$.因此,初始化阶段输出权重$\boldsymbol{\beta}_0$为

$$\boldsymbol{P}_0=(\boldsymbol{H}_0^{\mathrm{T}}\boldsymbol{W}_0\boldsymbol{H}_0+C\boldsymbol{I})^{-1} \tag{16}$$

$$\boldsymbol{\beta}_0=\boldsymbol{P}_0\boldsymbol{H}_0^{\mathrm{T}}\boldsymbol{W}_0\boldsymbol{Y}_0 \tag{17}$$

其中,C为惩罚项系数,$\boldsymbol{W}_0$为初始权重矩阵.在线学习阶段,当新实例$\left(\boldsymbol{X}_{k+1},y_{k+1}\right)$到达时,计算$\boldsymbol{h}_{k+1}$:

$$\boldsymbol{h}_{k+1}=\left[g\left(a_1,b_1,\boldsymbol{X}_{k+1}\right),\cdots,g\left(a_L,b_L,\boldsymbol{X}_{k+1}\right)\right] \tag{18}$$

于是输出层权值可以表示为

$$\begin{aligned}\boldsymbol{\beta}_{k+1}&=\left(\boldsymbol{H}_{k+1}^{\mathrm{T}}\boldsymbol{W}_{k+1}\boldsymbol{H}_{k+1}+C\boldsymbol{I}\right)^{-1}\boldsymbol{H}_{k+1}^{\mathrm{T}}\boldsymbol{W}_{k+1}\boldsymbol{Y}_{k+1}\\&=\left(\boldsymbol{H}_k^{\mathrm{T}}\boldsymbol{W}_k\boldsymbol{H}_k+\boldsymbol{h}_{k+1}^{\mathrm{T}}w_{k+1}\boldsymbol{h}_{k+1}+C\boldsymbol{I}\right)^{-1}\\&\quad\left(\boldsymbol{H}_k^{\mathrm{T}}\boldsymbol{W}_k\boldsymbol{Y}_k+\boldsymbol{h}_{k+1}^{\mathrm{T}}w_{k+1}y_{k+1}\right)\end{aligned} \tag{19}$$

其中,$\boldsymbol{H}_{k+1}=\begin{bmatrix}\boldsymbol{H}_k\\\boldsymbol{h}_{k+1}\end{bmatrix}$,$\boldsymbol{W}_{k+1}=\begin{bmatrix}\boldsymbol{W}_k&0\\0&w_{k+1}\end{bmatrix}$,$\boldsymbol{Y}_{k+1}=\begin{bmatrix}\boldsymbol{Y}_k\\y_{k+1}\end{bmatrix}$.对式(19)中的$\boldsymbol{H}_k^{\mathrm{T}}\boldsymbol{W}_k\boldsymbol{H}_k$和$\boldsymbol{H}_k^{\mathrm{T}}\boldsymbol{W}_k\boldsymbol{Y}_k$引入遗忘因子$\lambda\in[0,1]$,从而是使新旧样本具有不同权重,则式(19)变为:

$$\begin{aligned}\boldsymbol{\beta}_{k+1}&=\left(\lambda\boldsymbol{H}_k^{\mathrm{T}}\boldsymbol{W}_k\boldsymbol{H}_k+\boldsymbol{h}_{k+1}^{\mathrm{T}}w_{k+1}\boldsymbol{h}_{k+1}+\lambda C\boldsymbol{I}\right)^{-1}\\&\quad\left(\lambda\boldsymbol{H}_k^{\mathrm{T}}\boldsymbol{W}_k\boldsymbol{Y}_k+\boldsymbol{h}_{k+1}^{\mathrm{T}}w_{k+1}y_{k+1}\right)\end{aligned} \tag{20}$$

根据Woodbury公式,计算过渡矩阵$\boldsymbol{P}_{k+1}$的更新递推公式为

$$\begin{aligned}\boldsymbol{P}_{k+1}&=\left(\lambda\boldsymbol{H}_k^{\mathrm{T}}\boldsymbol{W}_k\boldsymbol{H}_k+\boldsymbol{h}_{k+1}^{\mathrm{T}}w_{k+1}\boldsymbol{h}_{k+1}+\lambda C\boldsymbol{I}\right)^{-1}\\&=\left(\lambda\boldsymbol{P}_k^{-1}+\boldsymbol{h}_{k+1}^{\mathrm{T}}w_{k+1}\boldsymbol{h}_{k+1}\right)^{-1}\\&=\frac{\boldsymbol{P}_k}{\lambda}-\frac{\boldsymbol{P}_k}{\lambda}\boldsymbol{h}_{k+1}^{\mathrm{T}}\left(w_{k+1}^{-1}+\boldsymbol{h}_{k+1}\frac{\boldsymbol{P}_k}{\lambda}\boldsymbol{h}_{k+1}^{\mathrm{T}}\right)^{-1}\boldsymbol{h}_{k+1}\frac{\boldsymbol{P}_k}{\lambda}\\&=\frac{\boldsymbol{P}_k}{\lambda}-\frac{\boldsymbol{P}_k\boldsymbol{h}_{k+1}^{\mathrm{T}}\boldsymbol{h}_{k+1}\boldsymbol{P}_k}{\lambda\left(\lambda/w_{k+1}+\boldsymbol{h}_{k+1}\boldsymbol{P}_k\boldsymbol{h}_{k+1}^{\mathrm{T}}\right)}\end{aligned} \tag{21}$$

则隐含层输出权重的$\boldsymbol{\beta}_{k+1}$更新递推公式为

$$\begin{aligned}\boldsymbol{\beta}_{k+1}&=\boldsymbol{P}_{k+1}\left(\lambda\boldsymbol{H}_k^{\mathrm{T}}\boldsymbol{W}_k\boldsymbol{Y}_k+\boldsymbol{h}_{k+1}^{\mathrm{T}}w_{k+1}y_{k+1}\right)\\&=\boldsymbol{P}_{k+1}\left(\lambda\boldsymbol{P}_k^{-1}\boldsymbol{\beta}_k+\boldsymbol{h}_{k+1}^{\mathrm{T}}w_{k+1}y_{k+1}\right)\\&=\boldsymbol{P}_{k+1}\left(\left(\boldsymbol{P}_{k+1}^{-1}-\boldsymbol{h}_{k+1}^{\mathrm{T}}w_{k+1}\boldsymbol{h}_{k+1}\right)\boldsymbol{\beta}_k+\boldsymbol{h}_{k+1}^{\mathrm{T}}w_{k+1}y_{k+1}\right)\\&=\boldsymbol{\beta}_k+\boldsymbol{P}_{k+1}\boldsymbol{h}_{k+1}^{\mathrm{T}}w_{k+1}\left(y_{k+1}-\boldsymbol{h}_{k+1}\boldsymbol{\beta}_k\right)\end{aligned} \tag{22}$$

整理式(21)和式(22)可得FWOSELM算法在线学习阶段更新递推公式为:

$$\boldsymbol{P}_{k+1}=\frac{\boldsymbol{P}_k}{\lambda}-\frac{\boldsymbol{P}_k\boldsymbol{h}_{k+1}^{\mathrm{T}}\boldsymbol{h}_{k+1}\boldsymbol{P}_k}{\lambda(\lambda/w_{k+1}+\boldsymbol{h}_{k+1}\boldsymbol{P}_k\boldsymbol{h}_{k+1}^{\mathrm{T}})} \tag{23}$$

$$\boldsymbol{\beta}_{k+1}=\boldsymbol{\beta}_k+\boldsymbol{P}_{k+1}\boldsymbol{h}_{k+1}^{\mathrm{T}}w_{k+1}(y_{k+1}-\boldsymbol{h}_{k+1}\boldsymbol{\beta}_k) \tag{24}$$

其中λ为遗忘因子,w_{k+1}为更新后的类别权重.当λ为1且正则化参数C为0时,FWOSELM退化为原始WOSELM.

FWOSELM算法为最近的样本分配较高的权重,而为旧的样本分配较低的权重,以表示它们对学习模型的不同贡献,因此使模型能够适应数据流的动态变化.同时,在学习时又为少数类分配更高的权重,提高少数类的分类错误损失,从而避免分类决策边界过度靠近少数类,具有更好的类不平衡适应能力.

虽然理论上FWOSELM算法可以同时适应概念漂移和类不平衡问题,但仍然存在以下四点不足:(1)固定的遗忘因子无法在遗忘历史数据和学习历史数据之间取得平衡.(2)当不平衡率太大时,简单地按照不平衡率IR进行加权可以无法取得最好的分类性能.(3)当历史数据累积太多时,新到达的数据影响太小.(4)仍然没有解决大多数真实数据集都存在的复杂分布问题.

针对以上问题,本文进一步提出的EAFWOSELM算法将给出对应解决方案:(1)引入自适应遗忘因子和概念漂移检测机制,使遗忘因子能随概念漂移程度升高而自适应变小.(2)基于各类的分类准确率引入类别权重修正项,防止类别加权权值过大或过小.(3)将遗忘因子引入混淆矩阵,降低历史累积数据对现在的影响.(4)将FWOSELM算法作为基分类器,设计集成学习策略,从而更好地适应复杂数据流.

3.3 自适应遗忘因子和概念漂移检测机制

自适应遗忘因子和概念漂移检测机制采用Gmean作为概念漂移检测的观测值,将遗忘因子引入混淆矩阵,只需保存历史最大的$\text{Gmean}_{\max}$和当前的Gmean值.

通常,概念漂移检测机制需要检测的概念漂移有两种类型,即突发式漂移和渐进式漂移.由于OSELM算法能根据最新数据不断更新模型,FROSELM算法在加入了遗忘因子后也更加关注最近到达的样本,因此具有较好的渐进式漂移适应能力.但发生突发式漂移时,OSELM算法和固定遗忘因子的FROSELM算法适应缓慢,分类性能下降明显,特别是面临概念反转型数据流时,分类模型很难更新成功.因此在EAFWOSELM算法中,如果集成分类器的性能下降到某个阈值τ及以下,则判断发生概念漂移,并自适应地调整遗忘因子.实验表明,将τ设置为历史最佳Gmean值的90%,可以使得实时Gmean值的方差和标准差获得较好的表现,即实时Gmean值较为稳定,可以较好地适应概念漂移.定义概念漂移指数$\text{CDI}=\text{Gmean}/\text{Gmean}_{\max}$.EAFWOSELM算法中默认的遗忘因子$\lambda$设置为0.999,当CDI小于或等于0.9时,将遗忘因子更新为

$$\lambda = 0.9 + \text{CDI} \times 0.1 \tag{25}$$

因此,概念漂移下遗忘因子的取值范围是[0.9,0.99].当发生概念漂移导致分类器性能下降时,概念漂移指数CDI会变小,此时遗忘因子λ也会自适应地减小,从而加快遗忘历史数据,适应新数据.使用Gmean代替平均分类准确率作为检测指标的优势是在数据流存在类不平衡时,可能少数类分类性能的显著下降并不会带来分类器整体分类准确率的显著下降,但会引起Gmean值的显著下降,从而避免概念漂移检测机制由于数据流存在类不平衡而出现漏报问题.

3.4 在线集成算法设计

为了增强FWOSELM算法的鲁棒性和应对更复杂的真实数据流,我们以FWOSELM算法为基分类器,引入自适应遗忘因子和概念漂移检测机制,进一步提出了在线集成算法EAFWOSELM.本节我们将介绍在线集成算法的3个核心策略,即多样性策略、组件管理策略和结构更新策略[19].

3.4.1 多样性策略

基分类器的多样性对集成分类器的性能起着至关重要的影响[20].EAFWOSELM算法在初始化阶段,首先构建一个存储基分类器FWOSELM的结构体BC_struct.struct of Base Classifiers)然后生成M个(实验中设置为12个)基分类器,通过采用不同模型参数的方式使基分类器之间存在差异,12个基分类器采用四种隐含层节点数[inputs, 2×inputs, 3×inputs, 4×inputs]和三种激活函数[sigmod, softplus, tanh]的交叉组合方式.隐含层节点数和激活函数直接影响着模型的复杂度,极大地影响着FWOSELM算法的学习性能,因此能够具有多样性.

3.4.2 组件管理策略

EAFWOSELM算法采用加权多数投票来进行分类预测,具体而言,当需要预测新到达样本$\boldsymbol{X}_i$的类标变量$\hat{y}_i$时,EAFWOSELM算法采用以下公式聚合组件的预测:

$$\hat{y}_i = \sum_{k=1}^{M} f_k\left(\boldsymbol{X}_i\right) \times V_k \tag{26}$$

其中,M为基分类器个数,V_k为第k个基分类器本轮的投票权重,而$f_k\left(\boldsymbol{X}_i\right)$则是第$k$个基分类器本轮的投票结果,且已经处理为类别标签.最后选择得分最多的类别标签作为本轮的预测分类结果.

3.4.3 结构更新策略

结构更新策略是对组件进行更新,让集成分类器能够更好地适应数据流的变化,从而在充分利用和快速遗忘旧知识之间取得一个更好的平衡.在EAFWOSELM算法结构更新策略中,首先在学习过程中更新集成分类器的类别权重CW(Combine Weight).然后更新每个基分类器的投票权重V和类别权重修正项CF(Correction Factor),进一步得到基分类器的更新权重W.最后通过自适应遗忘因子和概念漂移检测机制更新分类模型整体的遗忘因子λ.其中的核心是将遗忘因子引入混淆矩阵,集成分类器的混淆矩阵EC_CM(Confusion Matrix for Ensemble Classifiers)可以用作概念漂移检测和计算类别权重CW,而基分类器的混淆矩阵BC_CM(Confusion Matrix for Base Classifiers)可以同时用来确定每个基分类器的投票权重V和权重修正项CF,再根据类别权重CW和权重修正项CF可以确定每个基分类器的更新权重W.具体更新机制如图2所示.

其中每一轮学习中混淆矩阵更新公式为

$$\text{EC\_CM} = \lambda \times \text{EC\_CM} \tag{27}$$

$$\text{EC\_CM}\left[y\right]\left[\hat{y}\right] += 1 \tag{28}$$

基于集成分类器的混淆矩阵,可以得到'0'类权重CW的更新公式为

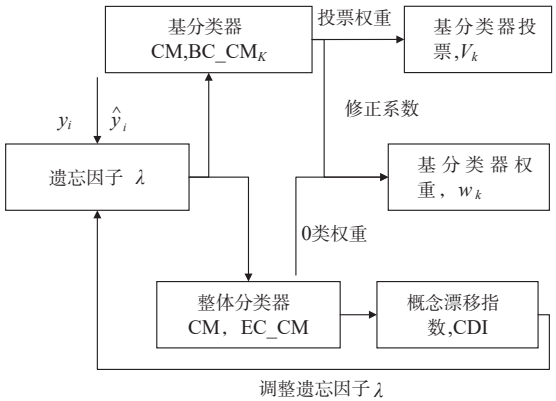


图2 自适应组件更新机制

$$\mathrm{CW}=\frac{\mathrm{FN}+\mathrm{TP}}{\mathrm{TN}+\mathrm{FP}} \tag{29}$$

遗忘因子λ的更新公式为

$$\lambda=\begin{cases}0.9+\mathrm{CDI}\times 0.1, & \mathrm{CDI}\le 0.9(\mathrm{Drift})\\ 0.999, & \mathrm{CDI}>0.9(\mathrm{Stability})\end{cases} \tag{30}$$

第k个基分类器的类别权重修正项CF_k更新公式为

$$\mathrm{CF}_k=\begin{cases}\mathrm{CF}_k\times 0.99, & \mathrm{Spe}-\mathrm{Rec}\ge\mu\\ \mathrm{CF}_k\times 1.01, & \mathrm{Rec}-\mathrm{Spe}\ge\mu\\ \mathrm{CF}_k, & \mathrm{Otherwise}\end{cases} \tag{31}$$

其中,我们将Spe–Rec的绝对值的界限定义为μ,这里取μ为0.1.

第k个基分类器的更新权重w的计算公式为

$$\mathrm{BC\_W}_k=\begin{cases}1, & y_i=1\\ \mathrm{CW}\times\mathrm{CF}_k, & y_i=0\end{cases} \tag{32}$$

将更新后的遗忘因子λ和基分类器的更新权重w代入式(23)和式(24)中,可以得到最新的输出权重$\boldsymbol{\beta}_{i+1}$,至此结束本轮学习.

3.5 算法伪代码

EAFWOSELM算法通过融合加权机制和遗忘机制以及采用在线集成学习方式形成较优分类模型,使其更好地适应概念漂移和类不平衡,算法伪代码见算法1所示.

算法1 EAFWOSELM

输入:数据流($\boldsymbol{X}$, y),初始化训练数量m
输出: Acc,Rec, Spe,Gmean,D(Rec,Spe)
1. // 1. 初始化阶段
2. 初始化基分类器的结构;
3. 生成M个不同的基分类器(BC, base classifier);
4. 随机生成每个基分类器的输入层权重和偏置;
5. 通过式(16)和式(17),使用($\boldsymbol{X}$, y)计算$\boldsymbol{P}$和$\boldsymbol{\beta}_0$;
6. 保存$\boldsymbol{P}$和$\boldsymbol{\beta}_0$到基分类器结构(BC_struct)中;
7. // 结束初始化阶段
8. // 2. 在线学习阶段
9. // 2. 1. 分类阶段
10. 通过式(13),使用$\boldsymbol{X}_i$计算每一个基分类器的预测值y_k;
11. 通过式(26)计算集成分类器的预测值y_pre;
12. 更新Acc, Rec, Spe, Gmean,D(Rec,Spe);
13. // 结束该轮分类;
14. // 2. 2. 在线学习与更新阶段;
15. 更新集成分类器的混淆矩阵;
16. 更新0类权重;
17. 计算概念漂移指数CDI;
18. 如果CDI <= 0.9 则更新遗忘因子λ;
19. for k=1 to M;
20. 更新$\mathrm{BC_CM}_k$;
21. 更新投票权重V_k;
22. 更新类别权重修正项CF_k;
23. 计算更新权重w_k;
24. 更新$\boldsymbol{P}_k$和$\boldsymbol{\beta}_k$;
25. // 结束本轮学习

3.6 复杂度分析

本小节将从时间复杂度与空间复杂度两个层面分析FWOSELM和EAFWOSELM算法的计算复杂性. 由于实验中初始化阶段实例数占总实例数的比值均小于3%,且现实中数据流往往不断产生,因此我们更关心在线学习阶段的时间和空间复杂度.

首先对FWOSELM算法而言,该算法与FROSELM和WOSELM两种算法类似,都只是对OSELM算法的更新公式进行了优化. FROSELM算法的遗忘机制增加了一个额外的遗忘因子λ,在计算时额外消耗的时间和空间可以忽略不计. 而WOSELM算法的加权机制虽然也只增加了一个权重项w,但每轮都需要对W进行更新,因此需要额外消耗更多的时间,不过这种更新的时间开销对于OSELM算法本身更新的时间开销而言,仍然可以忽略不计. 因此,FWOSELM、FROSELM、WOSELM和OSELM四种算法的时间复杂度基本相同,与数据流的实例数N成正比,假设OSELM算法每轮的预测和更新时间为T_1,则算法整体时间复杂度为$O(T_1\times N)$. 而且由于四种算法都属于在线学习方式,即都不需要保留历史数据,因此空间复杂度都为$O(1)$.

而对于EAFWOSELM算法而言,由于采用进行集成学习方式,使用M个基分类器进行加权投票作为最终的分类结果. 因此,在时间开销上主要分为三个部分,基分类器FWOSELM预测和更新的时间开销T_1、加权投票的时间开销T_2以及更新集成分类器的时间开销T_3. 因此,EAFWOSELM算法的整体时间开销为$O((M\times T_1+T_2+T_3)\times N)$,而由于每一轮在线学习的$T_2+T_3$远小于$M\times T_1$,所以EAFWOSELM算法的时间复杂度为$O(M\times T_1\times N)$. EAFWOSELM算法同样采用在线学习方式,不保留历史数据,只需要额外保留M个存储基分类器的结构体,因此空间复杂度为$O(1)$.

4　实验与结果

为了验证本文提出的EAFWOSELM和FWOSELM算法的性能以及其对概念漂移和类不平衡数据流的适应能力,本文在理论研究的基础上进行了大量的实验.本节主要介绍实验环境和数据集、参数敏感性分析、参数选择验证、时间复杂度验证以及算法性能对比.

4.1　实验数据集

本实验运行于单机环境,所有分类算法均基于python实现.为了验证EAFWOSELM和FWOSELM算法的性能,实验数据集选取12个人工合成数据集和2个真实数据集,为了简单起见,仿真实验只考虑用于二分类问题,实验数据集简要信息如表1.实验中默认所有算法的训练集实例数为500,惩罚参数C为0.1.

表1　实验数据集信息

数据集	实例数	特征数	IR	漂移类型
Sine_2	20 k	4	2	反转、突变
Sine_4	20 k	4	4	反转、突变
Sine_9	20 k	4	9	反转、突变
Sea_s_2	20 k	3	2	突变型
Sea_s_4	20 k	3	4	突变型
Sea_s_9	20 k	3	9	突变型
Sea_g_2	20 k	3	2	渐变型
Sea_g_4	20 k	3	4	渐变型
Sea_g_9	20 k	3	9	渐变型
Elec	45312	6	-	真实数据集
Weather	18159	8	-	真实数据集
Sine_100k	100 k	4	4	反转、突变
Sine_1M	1 M	4	4	反转、突变
Sine_10M	10 M	4	4	反转、突变

(1)Sine_IR数据集:其中IR为不平衡率,即数据集中'1'类样本数和'0'类样本数的比值,实验选择了2、4和9三种.数据集中含有4个属性,其中只有两个属性是相关的.数据集包含20000个实例,且在5000、10000和15000三个位置发生突变型反转,即概念漂移前后目标值刚好相反.

(2)Sea_s_IR数据集:其中IR为不平衡率,包含2、4和9三种.SEA生成器在SEA算法[21]中被提出,通过改变阈值,可以模拟概念漂移.数据集中含有3个属性,其中只有两个属性是相关的.通过使用SEA生成器生成了三个数据集,每个数据集包含20000个实例,并添加了3%的噪声.另外,三个数据集均包含2次概念漂移,且都发生在实例编号为5000和15000的位置.其中,Sea_s_IR数据集包含两次突变型概念漂移.

(3)Sea_g_IR数据集:其中IR为不平衡率,包含2、4和9三种.同上,Sea_g_IR数据集包含两次渐变型概念漂移.

(4)Elec数据集:是广泛应用于数据流学习中的真实数据集.该数据集是来自澳大利亚新南威尔士州电力市场1995年至1998年的部分数据,包含45312个实例.数据集一共包含6个相关属性,由于那里的电力价格不是固定的,而是根据供求关系而变化,因此目标是预测每天电力价格的变化(1=上升或0=下降).

(5)Weather数据集:包含1949年至1999年在内布拉斯加州Bellevue收集的天气信息,包含18159个实例.数据集一共包含8个相关属性,目的是预测给定日期是否下雨.

(6)Sine_N数据集:其中N为样本数量,选取了20 k(Sine_4),100 k,1 M,10 M的数据验证时间复杂度以及算法对大数据集的表现.

4.2　对比算法

实验重点关注基于优化OSELM更新公式和集成方法的在线学习算法,将本文提出的EAFWOSELM和FWOSELM两种算法与其他6种数据流在线分类算法进行性能比较.它们分别是VWOSELM、WOSELM、FROSELM、OSELM、LPP和SRP,具体介绍如下:

(1)OSELM算法:在线顺序极限学习机[8],由新加坡南洋理工大学黄广斌教授团队于2006年提出.该算法是黄教授之前提出的极限学习机ELM的在线学习方法,

(2)FROSELM算法:具有遗忘机制的正则在线顺序极限学习机[16],由Du等人于2015年提出.该算法将遗忘因子FF(Forgetting Factor)方法和正则化技术引入OSELM,根据实例的时间顺序分别为每个样本分配不同的权重.也就是说,为最近的样本分配较高的权重,而为旧的样本分配较低的权重,以表示它们对学习模型的不同贡献.

(3)WOSELM算法:加权在线顺序极限学习机[10],由Mirza等人于2013年提出.该算法基于代价敏感学习方法,根据不平衡率IR对少数类进行加权,从而使分类器具有类不平衡适应能力.

(4)VWOSELM算法:基于投票的加权在线顺序极限学习机[11],由Mirza等人于2015年提出.该算法以WOSELM算法为基分类器,同时可以应用于多分类问题,实验表明相比原始的WOSELM算法分类性能更好.

(5)LPP算法:Learn++.NSE算法[20]是Learn++系列算法中最受关注的算法之一.该算法由Elwell等人于2011年提出,其具有独特的多分类器投票机制,是一种可以从非平稳环境(NSE)中进行增量学习的集成分类器.

(6)SRP算法:流随机补丁(SRP)集成方法[21]模拟了装袋和随机子空间,由Gomes等人于2019年提出.该-算法默认的基分类器是Hoeffding树,但它可以使用

任何其他基分类器,此外算法默认采用ADWIN方法进行概念漂移检测.

4.3 评价指标

机器学习的主要目标是学习到性能更好的模型,因此用来评估模型性能的评价指标在机器学习过程中起到至关重要的作用.对于分类任务而言,分类准确率(Accuracy)是使用最广泛的性能评价指标,它可以衡量算法对整体样本的分类性能,但当数据流存在类不平衡问题时,分类准确率并不是最理想的评价指标.Kubat等人提出的几何均值Gmean[22]指标反映了分类器的总体性能,是衡量类不平衡数据流分类性能最重要的指标.大多数分类性能评价指标都是从表2的混淆矩阵中计算得出.

表2 二分类中的混淆矩阵

	预测为′0′类	预测为′1′类
实际为′0′类	TN	FP
实际为′1′类	FN	TP

本文实验使用以下5种指标来衡量类不平衡数据流的分类性能,具体计算公式如下:

(1)分类准确率(Accuracy,记为Acc):

$$\mathrm{Acc}=\frac{\mathrm{TN}+\mathrm{TP}}{\mathrm{TN}+\mathrm{FP}+\mathrm{FN}+\mathrm{TP}} \tag{33}$$

(2)召回率(Recall,记为Rec):

$$\mathrm{Rec}=\frac{\mathrm{TP}}{\mathrm{FN}+\mathrm{TP}} \tag{34}$$

(3)特异度(Specificity,记为Spe):

$$\mathrm{Spe}=\frac{\mathrm{TN}}{\mathrm{TN}+\mathrm{FP}} \tag{35}$$

(4)几何均值Gmean[22]:

$$\mathrm{Gmean}=\sqrt{\mathrm{Rec}\times\mathrm{Spe}} \tag{36}$$

(5)分类距离指标D(Rec,Spe):

$$D\left(\mathrm{Rec},\mathrm{Spe}\right)=\mathrm{abs}\left(\mathrm{Rec}-\mathrm{Spe}\right) \tag{37}$$

4.4 参数敏感性分析

为了解释融合WOSELM算法的加权机制和FOSELM算法的遗忘机制以及引入权重修正项CF的动机,本节设计了参数敏感性分析实验.首先测试OSELM、FROSELM和WOSELM三种算法是否能同时适应概念漂移和类不平衡,此外还测试WOSELM算法简单地采用不平衡率IR进行加权的方式在不同IR下的性能表现.

本次实验主要测试不同IR值下四种算法(包含OSELM、FROSELM、WOSELM和FWOSELM)在Sine_IR和Sea_s_IR数据集上的Gmean值,实验结果如图3所示.实验结果表明WOSELM算法不具备概念漂移适应能力,因此在Sine_IR数据集上性能表现很差;而FROSELM算法不具备类不平衡适应能力,因此当不平衡率IR增大时,Gmean值严重下降.原始的OSELM算法则两种能力都不具备,因此表现最差.而本文初步提出的融合了WOSELM加权机制和FROSELM遗忘机制的FWOSELM算法,在实验中Gmean值表现是四种对比算法中最好的,而且具有很好的概念漂移和类不平衡适应能力.

接下来进一步测试不同IR下WOSELM算法的性能表现,实验结果如图4所示.结果表明WOSELM算法简单地采用不平衡率IR进行加权无法取得最佳的效果.在Sine_IR数据集上,分类器按IR加权后过度关注′0′类,且IR值越大,倾向程度越高.而在Sea_s_IR数据集上,当IR小于8时,分类器按IR加权后过度关注′1′类;当IR大于10时,分类器按IR加权后过度关注′0′类;只有当IR为8或9时,WOSELM算法按IR加权的方式才取得较好的性能.由于在进行分类任务前我们无法提前获知不平衡率IR,而且不平衡率IR也可能不是一直固定不变的.特别是在真实数据集中,在不同时期不平衡率IR往往一直在改变,甚至多数类和少数类出现反转.因此,本文提出在分类过程中对权重引入一个自适应的修正项,从而使多数类和少数类分类性能更加平衡.

4.5 参数选择实验

4.5.1 τ选择实验

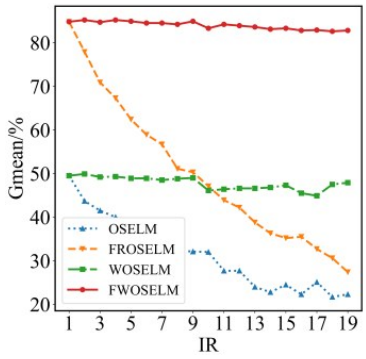
在探究τ值选择的实验中,我们选取了除Sine_N外的数据集进行实验;τ定义为发生概念漂移的阈值,则意味着选择正确的τ值可以使得算法在数据集上可以更好地适应概念漂移,即表现为Gmean的分时变化更为稳定,本实验采取了计算分时变化的Gmean的方差和标准差来探究τ的选择.

如图5所示,τ在90%左右的时候,方差和标准差都可获得较为良好的表现,意味着算法在τ选值为90%时可以更好地判断概念漂移的数据.由于本实验采用的数据集涵盖了概念漂移的所有类型,且本算法在使用τ为90%时在所有采用的数据集上都有良好表现,故可认为该选值对所有发生漂移的数据集具有普适性.

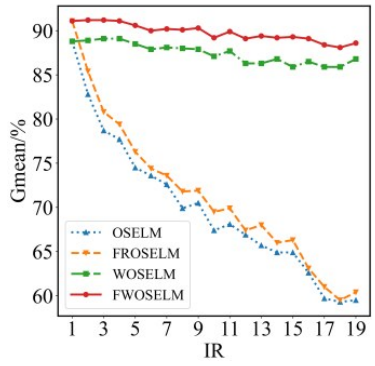
4.5.2 临界值μ值选择实验

为了探究类不平衡在什么时候发生,我们将μ值作为判断是否发生类不平衡变化的临界值,并在Sine_IR数据集和其他数据集上进行μ值选择的实验,实验选择范围为[0,0.2].

由图6可以看出,μ值过小则会让其过度反应比例的变化,抖动过大;μ值过大则会反馈迟钝,无法正确地反应最近到来数据的比例变化;在μ值为0.1的时候可以使得D(Rec,Spe)值在各数据集上表现良好,故取μ值为0.1.

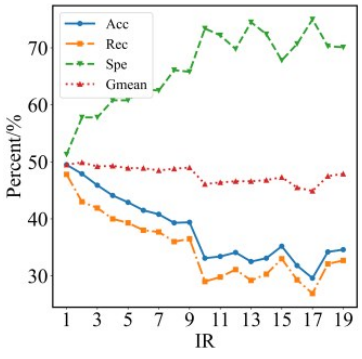


（a） 在Sine_IR上的性能

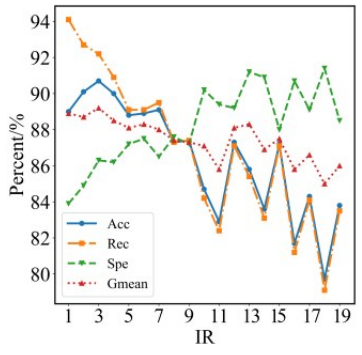


（b） 在Sea_s_IR上的性能

图3 不同IR下四种算法在Sine_IR和Sea_s_IR上的Gmean

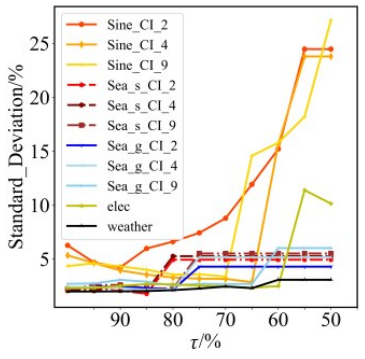


（a） 在Sine_IR上的性能

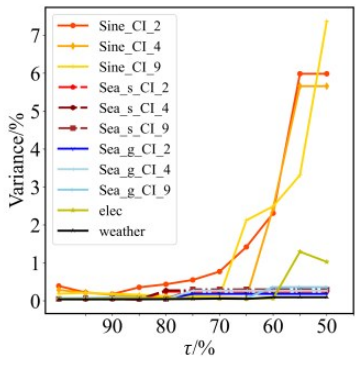


（b） 在Sea_s_IR上的性能

图4 不同IR值下WOSELM在Sine_IR和Sea_s_IR上的性能



（a） 分时Gmean标准差



（b） 分时Gmean方差

图5 分时Gmean标准差和方差在各数据集上的表现

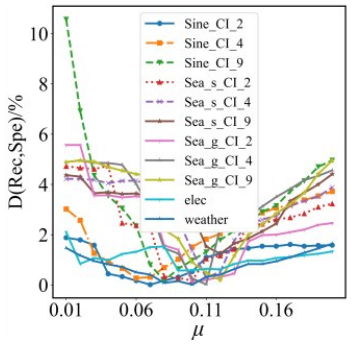


图6 不同μ值下D(Rec,Spe)在各数据集上的表现

4.6 时间复杂度验证

为了验证EAFWOSELM算法的时间复杂度为$O(M\times T_1\times N)$,其中T_1为算法每轮的预测和更新时间,M为基分类器的数量,N为数据集大小;因此,我们使用了大小分别为20 k,100 k,1 M,10 M的数据集进行实验,得到的数据如表3.

从表3可以看出,算法对数量不同的20 k、100 k、1 M、10 M数据集的运算时间与数据集的大小N成正比,故可得出EAFWOSELM算法的时间复杂度为$O(M\times T_1\times N)$.

表3 算法在不同大小数据集上的表现

	Sine 20 k	Sine 100 k	Sine 1 M	Sine 10 M
Acc	94.38	94.39	94.33	94.32
Rec	94.08	94.22	94.13	94.12
Spe	95.59	95.1	95.12	95.12
Gmean	94.83	94.66	94.62	94.62
D(Rec,Spe)	1.51	0.88	0.99	1.00
Time/s	25.47	134.55	1291.56	12784.87

与此同时,算法在面对大数据集的时候也有着较为稳定和优异的表现.

4.7 对比实验

在对比实验中,将本文提出的EAFWOSELM和FWOSELM两种算法与其他6种数据流在线分类算法进行性能比较. 其它6种分别是VWOSELM、WOSELM、FROSELM、OSELM、LPP和SRP. 其中,LPP和SRP是当前应用广泛的数据流在线分类算法. 而VWOSELM、WOSELM、FROSELM以及本文所提出来的EAFWOSELM和FWOSELM均是基于OSELM优化的算法. 实验采用五种性能评价指标——Acc、Rec、Spe、Gmean和D(Rec,Spe)来综合评估每个算法的分

类性能.

图7和图8分别展示了对比算法人工数据集Sine_4上的分时和累计Gmean以及综合性能指标.从图7可以看出,在[5000,1000,15000]这三个概念漂移发生点,EAFWOSELM算法性能都能最快恢复,具有最好的概念漂移适应能力.同时从图8可以看出EAFWOSELM算法全程都具有最佳的Gmean值,因此具有更好的类不平衡适应能力.

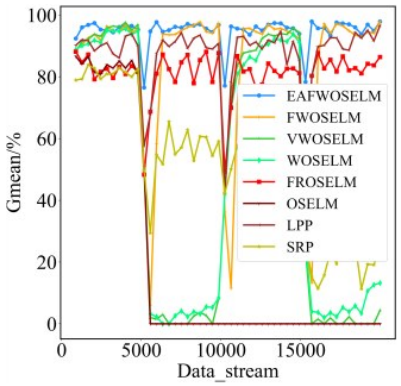


图7 对比算法在Sine_4上的分时Gmean

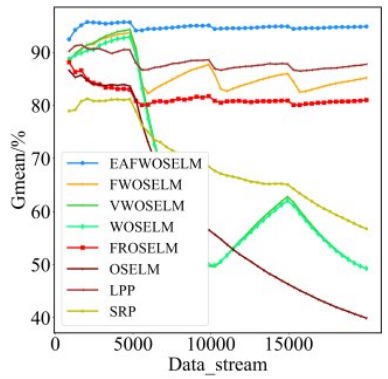


图8 对比算法在Sine_4上的实时Gmean

数据流分类最大的难题之一是如何应对复杂的真实数据流.因此,我们更加关注所提算法在Elec和Weather两个真实数据集上的表现.图9、图10分别展示了对比算法在真实数据集Elec上的分时和累计Gmean.从图9可以看出,VWOSELM、WOSELM和OSELM三种算法在[25000,35000]区间内Gmean表现非常差,无法适应复杂数据流,而EAFWOSELM可以很好地适应实时复杂数据.从图9可以看出,EAFWOSELM算法全程都保持着最佳的Gmean值.证明了EAFWOSELM算法在真实数据集上具有更好的分类性能,可以更好地适应复杂的数据流.

表4展示了八种对比算法在11个数据集上的综

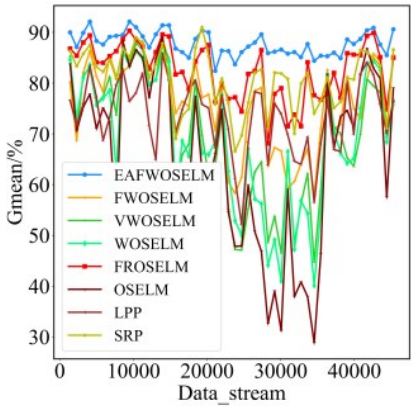


图9 对比算法在Elec上的分时Gmean

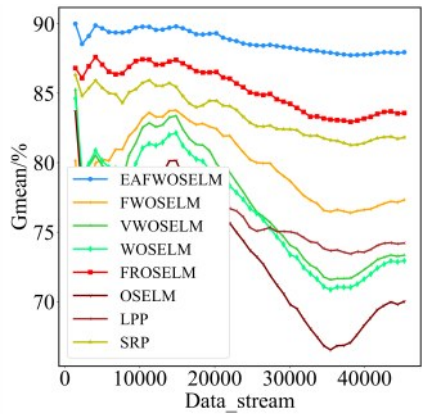


图10 对比算法在Elec上实时Gmean

合性能指标.其中EAFWOSELM算法在所有数据集上均取得了最佳的Gmean值,而且在其它指标上也取得了多项领先.在9个人工数据集上,可以观察到FROSELM、OSELM、LPP和SRP四种算法在Sine_IR、Sea_s_IR和Sea_g_IR这三类数据集上,随着数据流不平衡率IR增大,少数类分类准确率Spe均出现快速下滑,D(Rec, Spe)明显变大,从而导致Gmean值也迅速下降,因此可以断定这四种算法不具备类不平衡适应能力.此外,在Sine_IR数据集上,EAFWOSELM算法取得了巨大的领先优势,说明该算法可以快速适应反转型概念漂移,其它算法在面对这种概念漂移时,性能表现不佳.在Sea_s_IR和Sea_g_IR数据集上,EAFWOSELM和本文初步提出的FWOSELM算法性能相当,都取得了很好的少数类分类准确率,Gmean值也至少领先其它六种对比算法2个百分点,说明初步提出的FWOSELM算法已经具有较好的概念漂移和类不平衡适应能力.而在最后的2个真实数据集上,EAFWOSELM算法则取得显著领先的分类性能,Gmean值相比其它算法均至少提高了5个百分点,而在平均分类准确率上也提高了2个百分点左右.

表4　对比算法在不同数据集上的综合性能表现(%)

Datasets	Evaluation	EAFW OSELM	FW OSELM	VW OSELM	W OSELM	FR OSELM	OSELM	LPP	SRP
Sine_2	Acc	**95.07**	84.93	48.05	47.89	91.78	67.59	90.14	78.99
	Rec	94.89	84.37	44.28	42.95	**97.38**	90.92	92.08	92.49
	Spe	**95.42**	86.07	55.58	57.75	80.59	20.96	86.25	52.00
	Gmean	**95.15**	85.22	49.61	49.80	88.59	43.65	89.12	69.35
	D(Rec,Spe)	**0.53**	1.70	11.30	14.80	16.79	69.96	5.83	40.49
Sine_4	Acc	**94.38**	84.85	44.89	44.14	92.44	80.72	92.05	84.55
	Rec	94.08	84.62	41.49	39.99	**98.97**	96.76	94.69	97.41
	Spe	**95.59**	85.76	58.48	60.76	66.26	16.47	81.45	33.05
	Gmean	**94.83**	85.19	49.26	49.29	80.98	39.92	87.82	56.74
	D(Rec,Spe)	1.51	**1.14**	16.99	20.77	32.71	80.29	13.24	64.36
Sine_9	Acc	94.08	83.53	40.82	39.45	94.32	90.28	**94.49**	91.23
	Rec	93.98	83.19	38.24	36.49	**99.79**	99.22	96.91	99.48
	Spe	**94.96**	86.56	63.82	65.80	45.50	10.43	72.88	17.61
	Gmean	**94.47**	84.86	49.40	49.00	67.38	32.17	84.04	41.86
	D(Rec,Spe)	**0.98**	3.37	25.58	29.31	54.29	88.79	24.03	81.87
Sea_s_2	Acc	**92.25**	91.64	90.16	90.53	90.28	88.71	90.15	90.76
	Rec	92.3	92.57	93.66	93.59	**97.51**	98.03	93.24	95.67
	Spe	**92.13**	89.76	83.11	84.36	75.70	69.92	83.93	80.88
	Gmean	**92.21**	91.15	88.23	88.86	85.92	82.79	88.46	87.96
	D(Rec,Spe)	**0.17**	2.81	10.55	9.23	21.81	28.11	9.31	14.79
Sea_s_4	Acc	**92.5**	91.54	91.28	90.82	91.69	91.24	91.56	90.15
	Rec	92.77	91.86	93.02	91.95	98.57	**98.74**	95.15	97.68
	Spe	**91.41**	90.28	84.31	86.29	64.06	61.10	77.13	59.89
	Gmean	**92.09**	91.07	88.56	89.08	79.46	77.67	85.67	76.49
	D(Rec,Spe)	**1.36**	1.58	8.71	5.66	34.51	37.64	18.02	37.79
Sea_s_9	Acc	93.68	90.62	90.34	87.68	**94.30**	94.08	93.47	92.18
	Rec	93.97	90.68	90.86	87.63	98.91	98.95	96.77	**99.44**
	Spe	**91.12**	89.99	85.68	88.19	52.72	50.21	63.71	26.85
	Gmean	**92.53**	90.33	88.23	87.91	72.21	70.49	78.52	51.67
	D(Rec,Spe)	2.85	0.69	5.18	**0.56**	46.19	48.74	33.06	72.59
Sea_g_2	Acc	**91.59**	90.82	89.77	90.07	89.84	88.63	89.34	83.64
	Rec	91.55	91.62	93.36	92.66	97.17	**97.94**	92.49	93.14
	Spe	**91.66**	89.22	82.63	84.90	75.24	70.09	83.09	64.73
	Gmean	**91.6**	90.41	87.83	88.70	85.50	82.85	87.66	77.65
	D(Rec,Spe)	**0.11**	2.40	10.73	7.76	21.93	27.85	9.40	28.41
Sea_g_4	Acc	**91.54**	90.68	90.68	90.02	91.48	90.97	90.89	86.64
	Rec	91.59	91.02	92.40	90.93	98.48	**98.76**	94.80	98.09
	Spe	**91.53**	89.26	83.56	86.24	62.62	58.89	74.78	39.43
	Gmean	**91.46**	90.14	87.87	88.55	78.53	76.26	84.20	62.19
	D(Rec,Spe)	**0.26**	**1.76**	8.84	4.69	35.86	39.87	20.02	58.66
Sea_g_9	Acc	91.44	89.49	89.01	87.38	**94.01**	93.81	92.65	91.83
	Rec	91.53	89.64	89.56	87.40	98.76	98.95	96.23	**99.51**
	Spe	**90.63**	88.15	84.10	87.24	51.80	48.15	60.91	23.59
	Gmean	**91.08**	88.89	86.79	87.32	71.52	69.02	76.56	48.45
	D(Rec,Spe)	0.9	1.49	5.46	**0.16**	46.96	50.80	35.32	75.92

续表

Datasets	Evaluation	EAFW OSELM	FW OSELM	VW OSELM	W OSELM	FR OSELM	OSELM	LPP	SRP
Elec	Acc	**87.9**	77.78	75.84	75.17	84.63	75.01	75.08	84.29
	Rec	**88.24**	74.64	62.43	63.54	78.03	54.26	69.49	78.02
	Spe	87.66	80.09	85.71	83.74	89.50	**90.30**	79.19	88.91
	Gmean	**87.95**	77.32	73.15	72.94	83.57	70.00	74.18	83.29
	D(Rec,Spe)	**0.58**	5.45	23.28	20.20	11.47	36.04	9.70	10.89
Weather	Acc	**80.12**	73.10	72.20	70.94	78.81	76.76	69.43	75.38
	Rec	80.11	84.57	85.43	**86.53**	59.72	62.19	54.33	54.99
	Spe	80.12	67.83	66.11	63.77	**87.59**	83.46	76.38	84.75
	Gmean	**80.11**	75.74	75.15	74.28	72.32	72.04	64.42	68.27
	D(Rec,Spe)	**0.01**	16.74	19.32	22.76	27.87	21.27	22.05	29.76

5 总结

本文首先提出了一种具有遗忘因子的加权在线顺序极限学习机(FWOSELM)算法,该算法融合了WOSELM算法的加权机制和FOSELM算法的遗忘机制,从而能够同时具备概念漂移和类不平衡适应能力.此外,为了应对更复杂的真实数据流,本文进一步提出了一种具有自适应遗忘因子的加权在线顺序极限学习机集成算法(EAFWOSELM).该算法以FOSELM算法为基分类器,设计包含自适应遗忘因子和概念漂移检测机制的在线集成算法.

在仿真实验部分,通过参数敏感性分析验证了OSELM、FROSELM和WOSELM三种算法无法同时适应概念漂移和类不平衡,此外还验证了WOSELM算法简单地采用样本比例对′0′类进行加权无法在不同IR下都取得最佳的效果,从而解释了融合WOSELM和FOSELM两种算法以及引入权重修正项CF的动机.然后在对比实验中,通过将EAFWOSELM和FWOSELM两种算法与其他6种数据流在线分类算法进行性能比较,验证了EAFWOSELM算法的有效性,在9个人工数据集和2个真实数据集上,EAFWOSELM算法都取得了最优的Gmean值,且在大多数数据集上也获得了最高的分类准确率.从分类过程中可以看出,EAFWOSELM算法具有更好的概念漂移和类不平衡适应能力,表现出了更稳定、更平衡以及更准确的分类效果.尤其在两个真实数据集上,EAFWOSELM算法相比其它算法在Gmean和分类准确率都有显著的提高.另外值得一提的是,本文初步提出的FWOSELM算法也取得了很好的性能,在大多数情况下分类性能都超过另外六种对比算法.未来,我们将使用EAFWOSELM算法应用在多分类数据集上,以此来观察其分类性能的泛化能力.

参考文献

[1] PRIYA S, UTHRA RA. Comprehensive analysis for class imbalance data with concept drift using ensemble based classification [J]. J Ambient Intell Human Comput, 2020.

[2] WANKHADE K K, DONGRE S S, JONDHALE K C. Data stream classification: A Review[J]. Iran J Comput, 2020, 3(2): 239-260.

[3] SCHLIMMER JEFFREY C, GRANGER RICHARD H. Incremental learning from noisy data[J]. 1986, 1(3): 317-354.

[4] KHANDEKAR V S, SRINATH P. Non-stationary data stream analysis: State-of-the-Art challenges and solutions [C]//Proceeding of International Conference on Computational Science and Applications, Algorithms for Intelligent Systems. Singapore: Springer, 2020: 67-80.

[5] MAO W, WANG J, HE L, et al. Online sequential prediction of imbalance data with two-stage hybrid strategy by extreme learning machine[J]. Neurocomputing, 2017, 261: 94-105.

[6] NITIN M, ANANT V N. A survey on effects of class imbalance in data pre-processing stage of classification problem[J]. International Journal of Computational Systems Engineering, 2020, 6(2): 63−75.

[7] ZHANG B, CHEN Y. Research on detection and integration classification based on concept drift of data stream [J]. EURASIP Journal on Wireless Communications and Networking, 2019, 86. DOI: 10. 1186/s13638-019-1408-2.

[8] LIANG N, HUANG G, SARACHANDREN P, SUNDARARAJAN N. A fast and sccurate online sequential learning algorithm for feedforward networks[J]. IEEE Trans Neur Netw, 2006, 17(6): 1411-1423.

[9] HUANG G, ZHU Q, SIEW C. Extreme learning machine : Theory and applications[J]. Neurocomputing, 2006, 70(1-3): 489-501.

[10] MIRZA B, LIN Z, TOH KA. Weighted online sequential

extreme learning machine for class imbalance learning[J]. Neural Process Lett, 2013, 38: 465-486.

[11] MIRZA B, LIN Z, CAO J. Voting based weighted online sequential extreme learning machine for imbalance multi-Class classification[C]//2015 IEEE International Symposium on Circuits and Systems(ISCAS), Lisbon: IEEE, 2015: 565-568.

[12] KLIKOWSKI J, WONIAK M. Multi sampling random subspace ensemble for imbalanced data stream classification[C]//Progress of Computer Recognition Systems, CORES 2019. Berlin: Springer, 2020: 360-369.

[13] ZHU H, LIN G, ZHOU M, et al. Optimizing weighted extreme learning machines for imbalanced classification and application to credit card fraud detection[J]. Neurocomputing, 2020, 407: 50-62.

[14] HU W, ZHANG B. Study of sampling techniques and algorithms in data stream environments[C]//2012 9th International Conference on Fuzzy Systems and Knowledge Discovery. Chongqing, China: IEEE, 2012: 1028-1034.

[15] NGUYEN H M, COOPER E W, KAMEI K. Online learning from imbalanced data streams[C]//2011 International Conference of Soft Computing and Pattern Recognition (SoCPaR). Dalian, China: IEEE, 2011: 347-352.

[16] DU Z, LI X, ZHENG Z, ZHANG G, MAO Q. Extreme learning machine based on regularization and forgetting factor and its application in fault prediction[J]. Yi Qi Yi Biao Xue Bao/Chinese Journal of Scientific Instrument, 2015, 36(7): 1546-1553.

[17] YANG R, XU S, FENG L. An ensemble extreme learning machine for data stream classification[J]. Algorithms, 2018, 11(7). DOI: 10. 3390/a11070107.

[18] SIMON H. Adaptive Filter Theory[M]. Upper Saddle River: Prentice Hall, 2002.

[19] ZHAI T, GAO Y, WANG H, CAO L. Classification of high-dimensional evolving data streams via a resource-efficient online ensemble[J]. Data Mining and Knowledge Discovery, 2017, 31(5): 1242-1265

[20] MINKU L L, WHITE A P, YAO X. The impact of diversity on online ensemble learning in the presence of concept Drift[J]. IEEE Transactions on Neural Networks, 2011, 22(10): 1517-1531.

[21] GOMES H M, READ J, BIFET A. Streaming random patches for evolving data stream classification[C]//IEEE International Conference on Data Mining (ICDM), Beijing, China: IEEE, 2019: 240-249.

[22] KUBAT M, HOLTE R, MATWIN S. Learning when Negative Examples Abound[M]. Berlin: Springer, 1997.

作者简介

陆克中　男,1982年生. 博士,教授. 主要研究方向为大数据计算、并行计算和数据挖掘.

陈超凡　男,1998年生. 硕士研究生在读. 主要研究方向为机器学习,深度学习以及数据分析.

蔡　桓　男,1993年生. 硕士研究生. 主要研究方向为数据挖掘和机器学习.

吴定明(通讯作者)　女,1982年生. 博士,副教授. 主要研究方向为大数据计算和数据挖掘. E-mail1: dingming@szu. edu. cn