非平衡数据流在线主动学习方法

李艳红[1, 2] 任 霖[1, 2] 王素格[1, 2] 李德玉[1, 2]

摘 要 数据流分类是数据流挖掘领域一项重要研究任务, 目标是从不断变化的海量数据中捕获变化的类结构. 目前, 几乎没有框架可以同时处理数据流中常见的多类非平衡、概念漂移、异常点和标记样本成本高昂问题. 基于此, 提出一种非平衡数据流在线主动学习方法 (Online active learning method for imbalanced data stream, OALM-IDS). AdaBoost 是一种将多个弱分类器经过迭代生成强分类器的集成分类方法, AdaBoost.M2 引入了弱分类器的置信度, 此类方法常用于静态数据. 定义了基于非平衡比率和自适应遗忘因子的训练样本重要性度量, 从而使 AdaBoost.M2 方法适用于非平衡数据流, 提升了非平衡数据流集成分类器的性能. 提出了边际阈值矩阵的自适应调整方法, 优化了标签请求策略. 将概念漂移程度融入模型构建过程中, 定义了基于概念漂移指数的自适应遗忘因子, 实现了漂移后的模型重构. 在 6 个人工数据流和 4 个真实数据流上的对比实验表明, 提出的非平衡数据流在线主动学习方法的分类性能优于其他 5 种非平衡数据流学习方法.

关键词 主动学习, 数据流分类, 多类非平衡, 概念漂移

引用格式 李艳红, 任霖, 王素格, 李德玉. 非平衡数据流在线主动学习方法. 自动化学报, 2024, **50**(7): 1389−1401

DOI 10.16383/j.aas.c211246

Online Active Learning Method for Imbalanced Data Stream

LI Yan-Hong[1, 2] REN Lin[1, 2] WANG Su-Ge[1, 2] LI De-Yu[1, 2]

Abstract Data stream classification is an important research task in the field of data stream mining, which aims to capture changing class structures from the ever-changing massive data. At present, almost no frameworks can simultaneously address the common problems in data stream, such as multi-class imbalance, concept drift, outlier and the exorbitant costs associated with labeling the unlabeled samples. In this paper, we propose an online active learning method for imbalanced data stream (OALM-IDS). AdaBoost is an ensemble classification method that iteratively generates a strong classifier from multiple weak classifiers. AdaBoost.M2 further introduces the confidence degree of weak classifiers, which is suitable for static data. In the method, we firstly define an importance measure of training sample based on imbalanced ratio and adaptive forgetting factor, which makes the AdaBoost.M2 method applying for imbalanced data stream and improves the performance of ensemble classifier. Then, we propose an adaptive adjustment method of marginal threshold matrix, which optimizes the label request strategy. Finally, we define an adaptive forgetting factor based on the concept drift index by bringing the degree of concept drift into the construction process of model, which realizes the model reconstruction after drift. Comparative experiments on six artificial data streams and four real data streams show that the classification performance of the online active learning method is better than those of the existing five learning methods for imbalance data stream.

Key words Active learning, data stream classification, multi-class imbalance, concept drift

Citation Li Yan-Hong, Ren Lin, Wang Su-Ge, Li De-Yu. Online active learning method for imbalanced data stream. *Acta Automatica Sinica*, 2024, **50**(7): 1389−1401

收稿日期 2021-12-29 录用日期 2022-04-07
Manuscript received December 29, 2021; accepted April 7, 2022
国家自然科学基金 (62076158, 62072294, 41871286), 山西省重点研发计划 (201903D421041) 资助
Supported by National Natural Science Foundation of China (62076158, 62072294, 41871286) and Shanxi Key Research and Development Program (201903D421041)
本文责任编委 黎铭
Recommended by Associate Editor LI Ming
1. 山西大学计算机与信息技术学院 太原 030006 2. 山西大学计算智能与中文信息处理教育部重点实验室 太原 030006
1. School of Computer and Information Technology, Shanxi University, Taiyuan 030006 2. Key Laboratory of Computational Intelligence and Chinese Information Processing of Ministry of Education, Shanxi University, Taiyuan 030006

随着信息行业的高速发展, 大量数据以数据流的形式呈现, 如超市交易记录、网络搜索请求、电信呼叫记录和传感器网络数据等[1]. 在这些数据流中, 有许多需要处理的重要信息. 因此, 从这些海量数据流中发现和挖掘有价值信息已成为一项重要且艰巨的任务[2]. 与传统数据相比, 数据流具有实时性、连续性、时序性、变化性和无限性等特点. 因此, 数据流分类问题更为复杂[3].

数据流通常可以分为静态数据流和动态数据

流, 静态数据流以固定数据分布形式出现, 动态数据流数据分布会随着时间变化而变化[4]. 不同数据分布称为不同概念, 这种概念的变化称为概念漂移[5]. 概念漂移将导致之前训练好的分类模型不再适用于目前数据流环境, 从而严重影响数据流分类的准确率. 同时, 在数据流中, 还会存在一些异常点, 这些异常点的出现会导致分类模型的决策边界受到影响. 因此, 数据流分类模型中概念漂移和异常点的处理成为数据流分类研究的重要内容.

非平衡数据流分类研究包括二类非平衡分类(即存在一个多数类和一个少数类) 和多类非平衡分类[6]. 目前, 多数研究工作只关注二类非平衡分类[7−12]. 多类非平衡分类有 2 种研究思路, 第 1 种是直接处理, 第 2 种是将其转换成多个二类非平衡数据来研究. 采用第 1 种方法时, 由于分类取决于它与哪类进行比较, 因此类与类间关系更为复杂; 采用第 2 种方法时, 由于多个类间是相互联系的, 这种转换方法将丢失有价值信息[12]. 因此, 多类非平衡数据流分类的研究更具挑战性.

由此可见, 多类非平衡、概念漂移和异常点都会影响数据流分类模型性能, 当它们同时发生时, 会互相影响[13], 使数据流分类更为复杂. 例如, 随着时间的推移, 数据流的非平衡比率有可能会发生变化, 与之相应的采样机制应该随非平衡比率的变化而改变[14]. 目前, 众多学者对非平衡数据流的分类问题进行了研究. 如 Bifet 等[15] 提出的 LB (Leverage bagging) 方法是解决概念漂移和多类非平衡问题的经典算法之一, 该方法通过被动适应来应对概念漂移; Mirza 等[16] 提出基于极限学习机的多类非平衡数据流分类算法 (Meta-cognitive online sequential extreme learning machine, MOS-ELM), 是首次用于解决概念漂移和多类非平衡问题的方法; Barros 等[17] 提出一种在线学习集成算法 (Boosting-like online learning ensemble, BOLE) 将 AdaBoost[18] 引入数据流分类问题中, 用于解决多类非平衡问题. 此后, Ferreira 等[19] 提出自适应重采样随机森林算法 (Adaptive random forests with resampling, ARF_{RE}), 用于解决概念漂移和非平衡比率变化的非平衡数据流在线分类的问题.

然而, 上述方法都是有监督的学习方法 (即假定在训练期间不受限制地访问类标签), 而在真实数据流中获取所有样本的真实标签非常困难或者代价很高. 因此, 近年来主动学习方法[20] 备受关注, 原因是其有望用最少样本标签构建预测模型. 目前, 将主动学习方法与在线分类技术相结合, 已成为数据流分类的有效方法之一.

本文提出一种非平衡数据流在线主动学习方法 (Online active learning method for imbalanced data stream, OALM-IDS). 该方法由初始化阶段、在线学习阶段和概念漂移检测阶段构成. 在初始化阶段, 提出一种基于非平衡比率和自适应遗忘因子的样本初始权重定义方法, 使 AdaBoost.M2 方法适用于非平衡数据流, 提升了非平衡数据流集成分类器的性能; 在在线学习阶段, 为了适应数据分布的变化, 提出了边际阈值矩阵的自适应调整方法, 使标签请求策略可以选出难分和少数类的样本, 用于概念漂移后重新训练分类器; 在概念漂移检测阶段, 定义了基于集成分类器分类偏差的概念漂移指数, 并基于概念漂移指数, 定义了自适应遗忘因子, 从而将概念漂移程度融入模型重构.

本文主要贡献有以下 3 点:

1) 针对非平衡漂移数据流分类任务, 提出一个在线主动学习框架;

2) 提出含有自适应遗忘因子的样本初始权重定义方法, 使 AdaBoost.M2 方法适用于非平衡数据流, 并可以根据概念漂移程度实现分类模型的重构;

3) 提出基于样本分类不确定程度的边际阈值矩阵自适应调整方法, 构建了基于混合标签请求策略的主动学习模型.

1 相关工作

本文主要研究非平衡数据流的自适应分类方法, 该方法是一种主动学习方法. 下面对非平衡数据流分类方法和数据流主动学习方法的研究现状进行回顾.

1.1 非平衡数据流分类方法

目前, 面向具有概念漂移的非平衡数据流分类方法主要有基于数据块的分类方法和非基于数据块的分类方法 2 种研究思路.

1) 基于数据块的分类方法. 首先, 将数据流划分为长度等长的数据块; 然后, 通过对历史数据块和当前数据块的欠采样得到平衡的训练集, 训练样本大多数采样自当前数据块, 少数采样自历史数据块; 最后, 训练集成分类器, 并且在分类过程对概念漂移进行检测. MOS-ELM 是首次用于解决概念漂移和多类非平衡问题的方法, 该方法采用了基于块的研究思路. 这类方法研究思路简单, 但由于数据流被分割为等长的数据块, 会导致一个概念被划分到多个数据块中, 或一个数据块中包含多个概念, 而目前尚没有有效确定块长的方法.

2) 非基于数据块的分类方法可以避免上述提

到的块长难以确定问题. 这类方法多采用集成分类模型, 主要包括基于 bagging[21] 的方法和基于 boosting[22] 的方法. 基于 bagging 的方法, 随着非平衡数据的到来, 不断对样本采样, 从而得到平衡的训练集, 然后训练集成分类器, 并在分类过程对概念漂移进行检测. 但对异常点较多的数据流, 该类方法极易造成过拟合和准确率降低. 2010 年, Bifet 等[15] 在 bagging 方法基础上提出了 LB 方法, 该方法通过对预测准确率的检测, 来应对概念漂移, 当预测准确率下降时, 模型进行被动调整. 针对非平衡数据流的二分类问题, 自适应随机森林 (Adaptive random forest, ARF)[23] 是将随机森林应用到数据流上的一种改进方法. 该方法在概念漂移检测阶段设立警告阈值, 当达到警告值时, 分类模型开始训练新树; 当漂移发生时, 训练好的新树替换掉森林中最差的旧树, 从而提升了模型的更新速度. Ferreira 等[19] 提出 ARF_{RE} 方法, 旨在解决多类非平衡和概念漂移的混合问题. ARF_{RE} 继承了 ARF 中的概念漂移检测机制, 该方法在采样过程中, 通过记录样本到达时间来估算非平衡比率, 并将泊松分布的输出作为训练样本的权值, 以反映样本被训练的不同可能性.

目前, 有少数研究工作是基于 boosting 的方法, 其中最具代表性的是基于 AdaBoost 的方法[18]. AdaBoost 算法最初是基于静态数据提出的, 其核心思想在于将弱分类器经过多次迭代形成强分类器, 并且在迭代过程中, 样本权重由上次迭代该样本的分类难度来确定. 与基于 bagging 的方法相比, 基于 AdaBoost 的方法将 AdaBoost 作为集成分类器, 充分考虑了每个基分类器的权重, 提高了集成分类器的分类精度. Barros 等[17] 提出一种在线学习集成算法用于解决多类非平衡问题, 该方法一方面降低了 AdaBoost 中只保留准确率高于 50% 的分类器限制, 另一方面使用了基于分类错误率及标准差变化的概念漂移检测方法[24].

为了检测概念漂移, 在分类过程中, 以上方法需要利用真实标签计算准确率, 因此都是有监督的学习方法.

1.2 数据流主动学习方法

在样本标签有限情况下, 主动学习方法可通过少量的样本标签构建预测模型, 从而有效节约标签成本[25]. 然而, 主动学习方法的性能很大程度上依赖于标签请求策略的优劣. 典型的标签请求策略包括不确定性策略、随机策略和混合策略[20]. 不确定性策略根据模型对样本的预测不确定性程度选择样本, 而随机策略从数据流中随机选择样本, 混合策略则是结合了随机策略和不确定性策略的一种综合方法.

对于多类非平衡、概念漂移和异常点并存的数据流, 主动学习任务将更具挑战性. 一种简单方法是将标签请求策略添加到上述监督学习算法中. 但是, 这对有些方法 (如 MOS-ELM) 是不适用的, 因为方法要求已知所有样本标签, 另外一些方法 (如 LB、BOLE 和 ARF_{RE}) 由于没有综合考虑标签请求策略与分类方法, 使分类性能受到影响.

另一种更为有效的方法是结合分类方法设计标签请求策略, 即针对在线学习、多类非平衡、异常点和概念漂移这些分类问题设计标签请求策略. Shekhar 等[26] 提出二类非平衡数据流分类的集成分类器综合框架, 此框架采用根据非平衡比率请求标签的策略. Shan 等[27] 提出一种漂移数据流的在线主动学习集成 (Online active learning ensemble, OALE) 方法, 该方法采用不确定性策略和随机策略组成的混合标签请求策略. 当检测到概念漂移时, 优先请求最不确定的样本标签, 以减少标签请求的数量. 但该方法并没有考虑类非平衡比率可变问题. 之后, Liu 等[28] 提出一种基于概念漂移的多类非平衡数据流综合在线主动学习 (Comprehensive active learning method for multi-class imbalanced streaming data with concept drift, CALMID) 方法. 该方法采用基于非对称边际阈值矩阵的不确定性策略和随机策略相结合的混合策略作为标签请求策略, 在调整边际阈值矩阵时, 采用固定比率, 而没有区分样本分类的不确定程度.

本文综合考虑在线学习、多类非平衡、概念漂移和异常点这些分类问题来设计标签请求策略, 从而提高主动学习方法的性能.

2 问题的形式化定义

将所处理的非平衡数据流记作 $DS = \{(x_1, y_1), (x_2, y_2), \cdots, (x_t, y_t), \cdots\}$, 其中 (x_t, y_t) 为 t 时刻的数据样本 x_t 及其类别标签 y_t, x_t 为 d 维特征向量, $y_t \in \{C_1, C_2, \cdots, C_k\}$, k 为数据流中样本的类别数. 本文采用滑动窗口技术处理非平衡数据流的分类问题, 滑动窗口的大小反映了在多大粒度上对数据流进行分析. 在初始化阶段, 滑动窗口 $S_1[size]$ 用于存放数据流最初的 $size$ 个样本及其类别标签用于训练集成分类器 E, 标签滑动窗口 $S_2[size]$ 仅存放数据流最初的 $size$ 个样本的类别标签, 用于计算当前数据流的非平衡比率 $imb_t = (imb_t^1, imb_t^2, \cdots, imb_t^k)$. 在线学习过程中, 基于标签请求策略可得到

部分样本的真实标签, 将这些样本及其真实标签通过先进先出的方式更新滑动窗口 S_1. S_1 中的样本用于概念漂移时重新训练集成分类器; 滑动窗口 S_2 用于保存最近的 $size$ 个样本的标签 (如采用随机样本标签请求策略对应的样本保存其真实标签; 采用不确定性样本标签请求策略对应的样本保存 $Null$ 标签; 未请求标签的样本也保存 $Null$ 标签). 基于 S_2, 可计算当前数据流的非平衡比率 imb_t.

3 本文方法

3.1 训练样本的重要性度量

AdaBoost 算法是通过将弱分类器多次迭代训练形成强分类器, 在静态数据的二分类问题上具有很高的精度. AdaBoost.M2 算法是 AdaBoost 算法的改进, 将二分类问题扩展到多分类问题, 并充分考虑每个弱分类器的权重, 以提高模型的学习和泛化能力, 较 AdaBoost 算法, 取得了更好的分类效果[18]. AdaBoost.M2 算法适用于静态数据分类, 具体步骤[29]如下.

算法 1. AdaBoost.M2

输入. k 类 n 个样本, 记为 (x_i, y_i) $(i = 1, 2, \cdots, n)$, 其中 x_i 表示第 i 个样本, y_i 表示第 i 个样本对应的类别.

输出. 集成分类器 $h_{f(x)} = \arg\max_{y\in Y} \sum_{t=1}^{T} \lg\{(1/\eta_t) \times h_t(x, y)\}$, $Y = \{1, 2, \cdots, k\}$.

1) 初始化第 i 个样本的权重 $O_1(i) = 1/n$, 初始化样本 i 的某个错误标签 y 的权重 $\omega_{i,y}^1 = O_1(i)/(k-1)$.

2) 循环迭代 T 次, $t = 1, 2, \cdots, T$.

a) 计算第 t 次迭代中的标签权重和样本权重. 样本权重为 $O_t(i) = M_i^t / \sum_{i=1}^{n} M_i^t$. 标签权重为 $q_t(i, y) = \omega_{i,y}^t / M_i^t$, 其中 $M_i^t = \sum_y \omega_{i,y}^t$, y 表示除正确标签 y_i 以外的其他类别标签.

b) 基于 $O_t(i)$ 和 $q_t(i, y)$, 训练弱分类器 h_t.

c) 计算 h_t 的伪损失 $\varepsilon_t = (1/2)\sum_{i=1}^{n} O_t(i)[1 - h_t(x_i, y_i) + \sum_y q_t(i, y) \times h_t(x_i, y)]$, 其中 $h_t(x_i, y_i)$ 表示样本 x_i 属于类别 y_i 的概率.

d) 重置权重更新系数 $\eta_t = \varepsilon_t/(1 - \varepsilon_t)$.

e) 计算新的权重 $\omega_{i,y}^{t+1} = \omega_{i,y}^t \times \eta_t^{(1/2)[1-h_t(x_i,y_i)+h_t(x_i,y)]}$, 该权重用于第 $t+1$ 次训练.

由 AdaBoost.M2 算法步骤可知, 训练样本的初始权重是相同的. 在多类非平衡数据流环境下使用 AdaBoost.M2 算法, 为了提高算法分类性能, 需要为样本赋予不同的初始权重, 本文在度量训练样本的重要性时考虑以下 2 个因素:

1) 根据非平衡比率对不同类别的训练样本赋予不同的初始权重 (非平衡比率高的类别相应的样本权重较小, 反之应该较大);

2) 检测到概念漂移时需要重新训练分类器, 此时需要区分样本滑动窗口中样本的到达时间和概念漂移指数 (见式 (1)), 后到达的样本应该具有较高权重, 概念漂移指数大的样本应该具有较高的权重.

因此, 本文将训练样本的重要性, 即初始权重定义如下:

$$W(x) = \alpha(x) \times \lg\left\{1 + \frac{1}{imb_t^j}\right\} \tag{1}$$

式中, α 为自适应遗忘因子, 定义如下:

$$\alpha(x) = \begin{cases} 1, & \text{初始训练} \\ \exp\left\{-\dfrac{t - t'}{I(x) + 1}\right\}, & \text{重训练} \end{cases} \tag{2}$$

本文将概念漂移指数 I 引入时间衰减机制, 提出了自适应遗忘因子 α. 式 (2) 中, t 表示当前时刻, t' 表示窗口 S_1 中样本到达窗口的时间, I 表示该样本的概念漂移指数. 式 (2) 表明自适应遗忘因子在概念漂移后, 取决于概念漂移指数和样本到达窗口的时间 2 个因素. 因此, 概念漂移指数越大, 样本到达窗口的时间和当前时刻越近, 自适应遗忘因子越大.

imb_t^j 是数据流中 C_j 类在 t 时刻的非平衡比率, 通过滑动窗口 S_2 中积累的标签计算:

$$imb_t^j = \frac{labelnum_j}{\frac{size - Nullnum}{k}} \tag{3}$$

式中, $labelnum_j$ 是标签滑动窗口中 C_j 类的标签个数, $Nullnum$ 是标签滑动窗口中空标签 $Null$ 的个数, $Null$ 是在标签请求过程中存放的.

3.2 自适应标签请求算法

本文提出一种基于边际阈值矩阵的自适应标签请求 (Label request base on adaptive threshold matrix, LR-ATM) 算法, 基于样本分类的不确定程度, 对边际阈值矩阵进行自适应调整, 并根据分类难度请求样本的真实标签.

设 $P(y_{c_1}|x_t)$ 为集成分类器 E 对样本 x_t 的第 1 预测概率; $P(y_{c_2}|x_t)$ 为集成分类器 E 对样本 x_t 的第 2 预测概率. 边际阈值矩阵 $(m_{ij})_{k\times k}$, 记为 M, 其中 m_{ij} 表示 $Margin(x_t) = P(y_{c_1}|x_t) - P(y_{c_2}|x_t)$ 的阈值, 初值均为 θ. 在线学习过程中, M 是不断调整的, 具体方式如下:

1) 当 $Margin(x_t) > M[y_{c_1}][y_{c_2}]$ 时, 表明 x_t 的第 1 预测概率和第 2 预测概率的差值在限定范围内, 即分类器对 x_t 的分类是有效的, 此时不请求 x_t 的标签, 也不调整 M 中相应的阈值.

2) 当 $Margin(x_t) \le M[y_{c_1}][y_{c_2}]$ 时, 表明分类器对 x_t 的分类是不确定的, 需要请求 x_t 的真实标签 y_t

进一步验证, 若 $y_{c_1}=y_t$, 表明阈值 $m_{c_1c_2}$ 对 x_t 过于严格, 需要减小. 又因为在样本标签请求时, 应该给少数类更多机会, 与之相应的多数类应该有较少机会. 因此若 $imb_t^{y_t}>0.5$ 时, 需进一步减小阈值 $m_{c_1c_2}$.

由于阈值 $m_{c_1c_2}$ 减小的程度应与 x_t 的分类不确定程度成正比, 因此本文提出边际阈值矩阵的自适应调整方法如下:

$$m_{c_1c_2}=m_{c_1c_2}\times(1-\beta\times Margin(x_t)) \tag{4}$$

式中, $Margin(x_t)$为 x_t的分类不确定程度, $\beta\in[0,1]$.

本文提出的自适应标签请求策略基于样本的分类不确定性和类非平衡比率实现对难分类样本筛选, 由人工对这些样本进行确认, 并将人工标注得到的数据保存在样本滑动窗口中, 用于概念漂移后新的集成分类器的训练, 从而提升分类模型性能.

此外, 本文还使用随机标签请求策略, 随机从数据流中选取样本, 并请求其真实标签. 这些被随机选取的样本被认为是有代表性的样本, 反映了某时刻数据流的状态.

3.3　概念漂移检测机制

概念漂移指的是随时间推移, 数据流中的数据分布发生不可预测的变化. 假设目标概念可表示为联合概率分布, 即样本在 t 时刻的概率分布为 $P_t(x,y)$, 如果在 $t+1$ 时刻, 分布发生变化, 即 $P_{t+1}(x,y)\neq P_t(x,y)$, 则认为发生了概念漂移. 概念漂移可分为虚拟、真实和混合 3 种类型. 令 $P_t(x,y)=P_t(x)\times P_t(y|x)$.

1) 虚拟概念漂移是指决策边界未变化, 即 $P_t(y|x)=P_{t+1}(y|x)$, 而样本空间分布变化, 即 $P_t(x)\neq P_{t+1}(x)$.

2) 真实概念漂移是指决策边界变化, 即 $P_t(y|x)\neq P_{t+1}(y|x)$, 而样本空间分布未变化, 即 $P_t(x)=P_{t+1}(x)$.

3) 混合概念漂移是指决策边界和样本空间分布都发生变化, 即 $P_t(y|x)\neq P_{t+1}(y|x)$, 且 $P_t(x)\neq P_{t+1}(x)$.

如果进一步考虑数据流非平衡比率的变化, 即 $P_t(y)$ 的变化, 可令 $P_t(x,y)=P_t(y)\times P_t(x|y)$. 此时的概念漂移可分为以下 2 种情况:

1) $P_t(y)=P_{t+1}(y)$、$P_t(x|y)\neq P_{t+1}(x|y)$, 此时类非平衡比率并未发生变化.

2) $P_t(y)\neq P_{t+1}(y)$、$P_t(x|y)\neq P_{t+1}(x|y)$, 此时类非平衡比率发生变化, 因此在检测概念漂移时需要综合考虑非平衡比率的影响.

样本空间分布的变化并不会影响决策边界, 因此, 对数据流中的这类变化无需检测和处理, 只需检测由决策边界变化引起的概念漂移和数据流中非平衡比率的变化. 为此, 本文基于边际阈值矩阵和标签滑动窗口实现概念漂移检测, 通过度量当前样本的预测概率与标签滑动窗口中样本的平均预测概率的不相似性来判断是否发生了概念漂移.

边际阈值矩阵用于表示集成分类器对样本第 1 预测概率和第 2 预测概率的差异程度. 当 $Margin(x_t)\leq M[y_{c_1}][y_{c_2}]$ 时, 表示对于当前分类器, x_t 是一个难分的样本, 这种难分可能是由决策边界的变化或者非平衡比率变化引起的. 因此, 需要请求这些难分样本的标签, 并将其保存在 S_1 中, 用于分类器的更新.

当数据流中的新样本 x_t 进入分类器后, 得到一个预测标签 y_{c_1}, 如果满足样本标签请求机制的条件, 就会得到 x_t 的真实标签 y_t. 若 y_t 属于集合 (即 x_t 不是异常点), 此时需要判断数据流中是否出现概念漂移.

由于 S_1 中累积了难分或有代表性样本, 如果 x_t 预测概率与 S_1 中样本的平均预测概率相差较大, 则认为 t 时刻发生概念漂移的可能性较大. 因此, 本文定义了如下的概念漂移指数 I:

$$I(x_t)=\frac{P_{\max}-(P_t-V_t)}{V_{\min}}-1 \tag{5}$$

式中, $P_{\max}$ 是 S_1 中样本预测概率的最大值; P_t 和 V_t 分别表示 x_t 的预测概率和标准差; $V_{\min}$ 是 S_1 中样本预测概率的最小标准差. 标准差 V 的计算如下:

$$V=\sqrt{P\times(1-P)} \tag{6}$$

基于概念漂移指数 I, 可判断是否发生了概念漂移. 当 $I\leq 0$ 时, 表明未检测到概念漂移; 当 $I\geq 1$ 时, 表明发生概念漂移, 需要使用 S_1 中的样本重新训练集成分类器 E; 当 $1>I>0$ 时, 表明数据流处于漂移警告状态, 需要使用 x_t 在线更新集成分类器 E. 在线更新集成分类器的过程是对基分类器自下而上的“剪枝”过程, 对基分类器上的每个非叶子树 Tr, 依次用 Tr 上深度从大到小的叶子尝试替代 Tr. 如果 Tr 被某个叶子替代后得到的基分类器对 x_t 的误差不变或变小, 则用该叶子替代 Tr.

3.4　算法框架

本文提出一种非平衡漂移数据流在线主动学习方法 OALM-IDS, 该方法分为 3 个阶段, 如图 1 所示. 第 1 阶段为初始化阶段, 为了训练集成分类器, 首先请求数据流前 $size$ 个样本的真实标签, 并将带有真实标签 y_t 的样本 x_t 存入 S_1 中, 将真实标签存入 S_2 中. 然后, 利用 S_2 中的标签计算类非平衡率

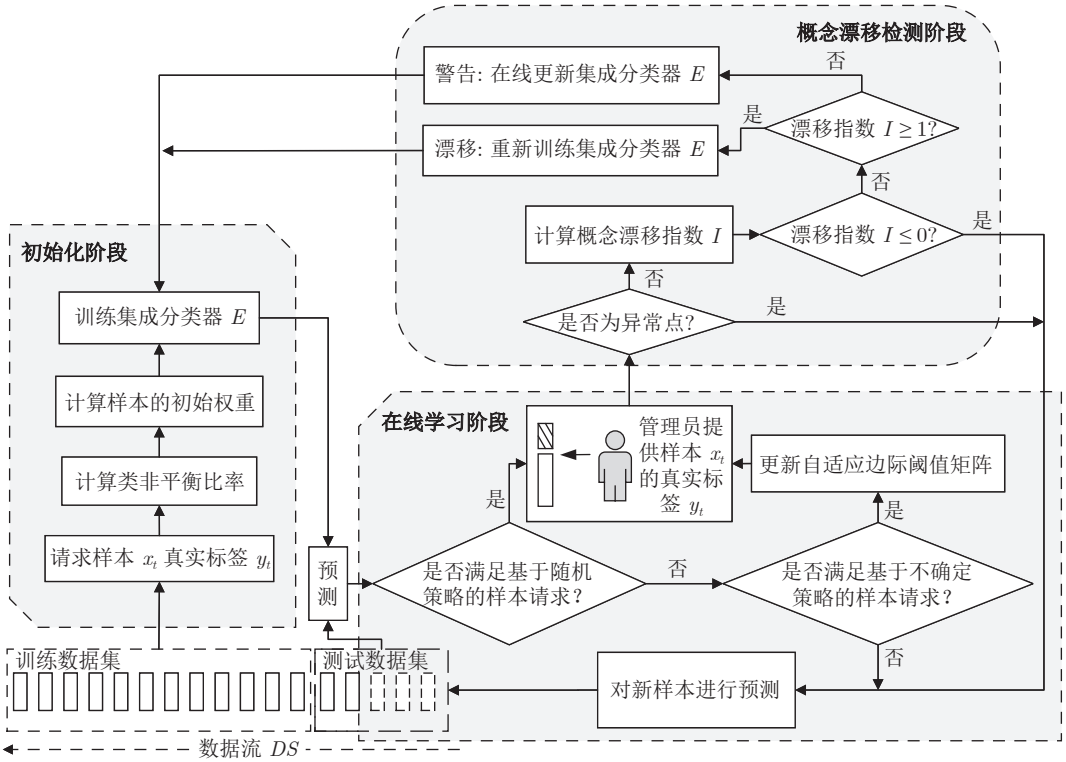

图 1 算法框架

Fig. 1 Algorithm framework

imb_t, 并通过式 (1) 计算 S_1 中每个样本的初始权重. 最后, 将 S_1 中加权的样本作为训练集, 训练集成分类器 E.

第 2 阶段为在线学习阶段, 首先利用集成分类器 E 对当前样本 x_t 预测. 然后, 基于随机标签请求策略判断是否需要请求 x_t 的标签, 如果需要, 则请求; 否则, 进一步基于不确定标签请求策略再次判断是否需要请求 x_t 的标签. 如果需要, 则请求, 更新边际阈值矩阵内的阈值; 若不需要, 则对下一个样本进行预测.

第 3 阶段为概念漂移检测阶段, 对于请求了真实标签的样本, 判断是否为异常点, 若是, 则继续预测下一个样本; 若不是, 则基于式 (5) 计算概念漂移指数 $I(x_t)$. 若 $I \leq 0$ 说明没有检测到概念漂移, 继续对下个样本进行预测; 若 $1 > I > 0$ 表明进入警告状态, 在线更新集成分类器 E; 若 $I \geq 1$ 表明检测到概念漂移, 更新 S_1 中样本的权重并重新训练集成分类器 E.

3.5 算法伪代码

本节给出完整的学习算法 OALM-IDS 和 LR-ATM 的伪代码.

算法 2. OALM-IDS 的伪代码

输入. 数据流 DS, 已经处理的样本数 g, 带标签的样本数 l, 标签预算 b ($b < \varepsilon$), 随机选择比率 ε, 滑动窗口大小 $size$, 样本滑动窗口 $S_1[size]$, 标签滑动窗口 $S_2[size]$.

输出. 集成分类器 E 的预测结果.

1) while (输入一个样本 x_t) do;
2) 集成分类器 E 对当前样本 x_t 预测;
3) $g = g + 1$;
4) $labelling =$ False;
5) 生成一个随机变量 ζ, $\zeta \in [0, 1]$;
6) if ($\zeta < \varepsilon$ 或 $g < size$) then;
7) $labelling =$ Ture;
8) 将真实标签 y_t 赋予样本 x_t;
9) 将 y_t 推入 S_2;
10) else if (LR-ATM() and $(l/g) < b$) then;
11) 将 $Null$ 推入 S_2;
12) $labelling =$ Ture;
13) else;
14) 将 $Null$ 推入 S_2;
15) 输出样本 x_t 的预测结果;
16) end if;
17) if ($labelling =$ Ture) then;
18) $l = l + 1$;
19) if ($y_{c_1} \notin \{C_1, C_2, \cdots, C_k\}$) then;

20) 则该点为异常点, 继续对下一个样本进行预测;
21) else;
22) 将带标签的样本推入 S_1;
23) 根据式 (5), 计算概念漂移指数 I;
24) if ($I > 0$) then;
25) if ($I \geq 1$) then;
26) 根据式 (2), 计算 α;
27) 利用 S_2 中的样本, 重新训练集成分类器 E;
28) end if;
29) 在线更新集成分类器 E;
30) end if;
31) end if;
32) end if;
33) end while.

算法 3. LR-ATM 的伪代码

输入. 集成分类器 E 对样本 x_t 的第 1 预测概率 $P(y_{c_1}|x_t)$, 集成分类器 E 对样本 x_t 的第 2 预测概率 $P(y_{c_2}|x_t)$, 自适应阈值矩阵 $M[k][k]$, 矩阵内阈值的初始值 θ.

输出. $labelling \in$ {Ture, False}.

1) 计算样本预测难度 $Margin(x_t)$;
2) if ($Margin(x_t) \leq M[y_{c_1}][y_{c_2}]$) then;
3) $labelling =$ Ture;
4) 计算 y_t 的类非平衡比率;
5) if ($y_{c_1} = y_t$) then;
6) 根据式 (4), 调整矩阵参数;
7) if ($imb_t^{y_t} > 0.5$) then;
8) 根据式 (4), 调整矩阵参数;
9) end if;
10) else
11) 不做调整;
12) end if;
13) else
14) 不做调整;
15) end if.

3.6　算法复杂度分析

由于初始化阶段和概念漂移检测阶段的时间成本和空间成本取决于 AdaBoost.M2 算法, 因此只分析在线学习阶段的复杂度.

由算法 2 可知, 假设目前数据流中有 N 个样本, 混合标签请求算法的时间复杂度是 O (N); 含有 D 个基分类器的集成分类器 E 预测的时间复杂度为 O ($D \cdot N \log_2 N$), 因此 N 个样本在线学习的时间复杂度为 O ($N + D \cdot N \log_2 N$).

在线学习阶段需要 2 个滑动窗口 $S_1[size]$ 和 $S_2[size]$. 因此, 算法的空间复杂度为 O($size$).

4　实验结果与分析

4.1　实验环境及数据

本文实验环境为 Windows10 操作系统, CPU 为 Intel Core i7-10750H 2.6 GHz, 内存 16 GB. 本文实验均在大规模在线分析平台 MOA (Massive online analysis)[30] 上实现, 开发软件使用 IntelliJ IDEA.

实验使用 6 个人工数据流和 4 个真实数据流, 数据流的特征如表 1 所示. 人工数据流均通过 MOA 平台生成, 其中 DS_1 和 DS_2 为平衡数据流, DS_3 和 DS_4 为非平衡比率固定的非平衡数据流, DS_5 和 DS_6 为非平衡比率可变的非平衡数据流. 在 DS_1、DS_3 和 DS_5 中没有设置概念漂移和异常点, 在 DS_2、DS_4 和 DS_6 中设置了 3 次概念漂移和 10 个异

表 1　数据流的特征
Table 1　Data stream feature

数据流	样本数	特征数	类别数	类分布	漂移次数	异常点
DS_1	200 000	21	5	(0.2, 0.2, 0.2, 0.2, 0.2)	0	0
DS_2	200 000	21	5	(0.2, 0.2, 0.2, 0.2, 0.2)	3	10
DS_3	200 000	21	5	(0.1, 0.3, 0.4, 0.2, 0.1)	0	0
DS_4	200 000	21	5	(0.1, 0.3, 0.4, 0.2, 0.1)	3	10
DS_5	200 000	21	5	(0.1, 0.3, 0.4, 0.2, 0.1), (0.4, 0.2, 0.1, 0.1, 0.2)	0	0
DS_6	200 000	21	5	(0.1, 0.3, 0.4, 0.2, 0.1), (0.4, 0.2, 0.1, 0.1, 0.2)	3	10
Kddcup 99_10%	494 000	42	23	—	—	—
Statlog	570 000	10	7	—	—	—
IoT	663 000	115	11	—	—	—
HAR	10 299	561	6	—	—	—

常点. 真实数据流 Kddcup 99_10%、Statlog (Shuttle)、IoT (IoT botnet attack) 和 HAR (Human activity recognition) 来源于公开数据集 UCIs (University of California at irvine), 在这 4 个数据流中, 概念漂移的类型和异常点的数量是未知的.

4.2 OALM-IDS 算法分类性能评价

本节将本文提出的分类算法 OALM-IDS 与 LB、BOLE、ARF$_{RE}$、OALE、CALMID 五个分类算法在 6 个人工数据流和 4 个真实数据流上进行性能比较, 其中 LB、BOLE 和 ARF$_{RE}$ 是监督学习算法, OALE 和 CALMID 是主动学习算法. 使用准确率、召回率、F1 值、Kappa 系数值和接受者操作特征 (Receiver operating characteristic, ROC) 曲线作为评价指标.

为了确保实验的公平性, 除了 ARF$_{RE}$ 使用其构造的 ARFHoeffding 树作为基分类器之外, 其他算法均使用 Hoeffding 树作为基分类器; OALE 中数据块大小和 CALMID 中滑动窗口的大小都设置为 500; OALE 根据学习条件按需使用真实标签. 此外所有主动学习算法的标签预算均设置为 20%. 所有算法在同一数据集上均重复实验 10 次.

由表 2 可知, OALM-IDS 的准确率在 5 个人工数据流和 4 个真实数据流上均为最高, 仅在 DS$_1$ 上比 BOLE 低 0.1, 这是由于 BOLE 为监督学习算法, 需要使用所有样本的标签信息, 并且使用了先进的集成分类器 AdaBoost.M2.

由表 3 ~ 5 可知, OALM-IDS 的召回率、F1 值和 Kappa 系数值在 6 个人工数据流和 4 个真实数据流上都优于对比算法, 且 F1 值有明显提高.

通过对表 2 ~ 5 实验结果分析可知, 主动学习算法 OALE、CALMID 和 OALM-IDS 的分类性能

表 2 6 种算法的准确率

Table 2 Precision values of six algorithms

数据流	LB	BOLE	ARF$_{RE}$	OALE	CALMID	OALM-IDS
DS$_1$	94.56 ± 0.12	**95.61 ± 0.11**	93.54 ± 0.13	89.78 ± 0.21	94.76 ± 0.16	95.48 ± 0.15
DS$_2$	92.27 ± 0.17	92.44 ± 0.14	91.04 ± 0.19	88.31 ± 0.23	92.81 ± 0.13	**93.94 ± 0.12**
DS$_3$	88.39 ± 0.22	89.52 ± 0.14	90.95 ± 0.13	88.83 ± 0.16	92.57 ± 0.13	**93.72 ± 0.13**
DS$_4$	86.55 ± 0.31	88.68 ± 0.26	89.89 ± 0.23	86.29 ± 0.29	91.31 ± 0.18	**92.18 ± 0.21**
DS$_5$	85.64 ± 0.29	87.04 ± 0.34	89.61 ± 0.51	88.83 ± 0.21	91.13 ± 0.21	**92.92 ± 0.16**
DS$_6$	82.10 ± 0.69	83.15 ± 0.73	86.54 ± 0.72	83.42 ± 0.55	90.64 ± 0.42	**92.41 ± 0.21**
Kddcup 99_10%	83.87 ± 0.43	81.09 ± 0.56	85.48 ± 0.65	81.01 ± 0.36	92.06 ± 0.19	**92.07 ± 0.18**
Statlog	64.55 ± 0.31	63.78 ± 0.61	79.97 ± 0.39	73.78 ± 0.43	85.40 ± 0.34	**85.68 ± 0.33**
IoT	64.03 ± 0.48	61.54 ± 0.43	66.66 ± 0.53	55.81 ± 0.51	70.85 ± 0.54	**73.12 ± 0.38**
HAR	61.63 ± 0.53	59.76 ± 0.46	63.22 ± 0.49	55.16 ± 0.69	68.64 ± 0.71	**69.98 ± 0.51**

表 3 6 种算法的召回率

Table 3 Recall values of six algorithms

数据流	LB	BOLE	ARF$_{RE}$	OALE	CALMID	OALM-IDS
DS$_1$	95.37 ± 0.18	95.96 ± 0.13	93.39 ± 0.11	90.13 ± 0.13	95.91 ± 0.11	**96.14 ± 0.12**
DS$_2$	92.39 ± 0.21	92.28 ± 0.35	91.35 ± 0.26	89.45 ± 0.18	92.51 ± 0.15	**94.08 ± 0.14**
DS$_3$	87.55 ± 0.19	88.19 ± 0.22	86.14 ± 0.21	88.52 ± 0.22	90.55 ± 0.13	**92.52 ± 0.13**
DS$_4$	84.57 ± 0.36	86.73 ± 0.29	87.47 ± 0.28	83.05 ± 0.31	89.89 ± 0.21	**92.44 ± 0.18**
DS$_5$	84.14 ± 0.43	86.44 ± 0.49	87.26 ± 0.69	83.26 ± 0.36	90.25 ± 0.18	**91.16 ± 0.13**
DS$_6$	83.98 ± 1.13	81.87 ± 0.91	84.56 ± 1.31	78.87 ± 0.69	90.46 ± 0.13	**90.71 ± 0.21**
Kddcup 99_10%	60.82 ± 0.71	62.75 ± 0.64	58.17 ± 1.32	58.44 ± 1.63	61.88 ± 0.43	**63.71 ± 0.37**
Statlog	61.39 ± 0.91	50.92 ± 1.32	54.36 ± 1.11	51.20 ± 1.34	59.52 ± 0.63	**63.12 ± 0.39**
IoT	40.73 ± 2.14	42.29 ± 1.58	39.35 ± 1.89	40.42 ± 2.15	48.04 ± 1.04	**51.26 ± 0.81**
HAR	61.64 ± 1.18	60.57 ± 0.97	57.91 ± 1.43	54.11 ± 1.36	65.53 ± 0.76	**66.57 ± 0.46**

表 4　6 种算法的 F1 值
Table 4　F1 values of six algorithms

数据流	LB	BOLE	ARF_{RE}	OALE	CALMID	OALM-IDS
DS_1	94.96 ± 0.11	**95.80 ± 0.10**	93.42 ± 0.13	89.93 ± 0.15	95.33 ± 0.11	**95.80 ± 0.10**
DS_2	92.32 ± 0.16	92.34 ± 0.13	91.18 ± 0.15	88.85 ± 0.21	92.65 ± 0.13	**94.01 ± 0.12**
DS_3	87.91 ± 0.20	88.81 ± 0.24	88.11 ± 0.36	88.67 ± 0.20	91.50 ± 0.16	**93.07 ± 0.14**
DS_4	85.35 ± 0.42	87.38 ± 0.36	88.42 ± 0.51	84.50 ± 0.33	90.51 ± 0.21	**92.29 ± 0.20**
DS_5	84.85 ± 0.41	86.67 ± 0.43	88.30 ± 0.46	85.36 ± 0.48	90.62 ± 0.21	**91.93 ± 0.18**
DS_6	82.97 ± 0.87	82.43 ± 0.71	85.35 ± 0.91	80.59 ± 0.63	90.46 ± 0.39	**91.53 ± 0.31**
Kddcup 99_10%	73.12 ± 0.55	72.47 ± 0.63	72.01 ± 0.46	72.81 ± 0.51	73.56 ± 0.33	**74.65 ± 0.20**
Statlog	66.18 ± 0.83	54.32 ± 1.91	63.85 ± 1.03	63.42 ± 0.98	74.42 ± 0.36	**75.19 ± 0.31**
IoT	47.01 ± 1.24	48.40 ± 0.96	47.34 ± 1.89	44.94 ± 1.36	54.26 ± 0.65	**56.73 ± 0.67**
HAR	59.93 ± 0.91	58.81 ± 1.21	58.52 ± 0.79	54.43 ± 1.13	65.43 ± 0.63	**67.76 ± 0.58**

表 5　6 种算法的 Kappa 系数值
Table 5　Kappa coefficient values of six algorithms

数据流	LB	BOLE	ARF_{RE}	OALE	CALMID	OALM-IDS
DS_1	90.17 ± 0.12	91.18 ± 0.14	90.59 ± 0.16	85.47 ± 0.21	90.48 ± 0.19	**91.31 ± 0.12**
DS_2	88.85 ± 0.19	88.14 ± 0.23	87.91 ± 0.39	83.18 ± 0.56	89.97 ± 0.31	**90.66 ± 0.23**
DS_3	85.25 ± 0.22	85.86 ± 0.38	86.68 ± 0.29	83.91 ± 0.39	88.91 ± 0.26	**89.93 ± 0.21**
DS_4	84.15 ± 0.55	86.04 ± 0.63	87.14 ± 0.66	83.42 ± 0.71	88.92 ± 0.33	**89.33 ± 0.36**
DS_5	83.85 ± 0.77	85.83 ± 0.69	86.45 ± 0.81	86.67 ± 0.70	88.57 ± 0.31	**89.12 ± 0.29**
DS_6	81.49 ± 1.12	82.98 ± 1.69	84.15 ± 1.87	79.92 ± 1.48	89.01 ± 0.41	**89.73 ± 0.28**
Kddcup 99_10%	80.93 ± 0.67	75.62 ± 1.13	79.32 ± 1.32	78.31 ± 0.91	83.32 ± 0.26	**85.83 ± 0.18**
Statlog	58.71 ± 1.42	61.43 ± 1.18	73.72 ± 0.93	71.21 ± 1.24	79.39 ± 0.46	**80.11 ± 0.19**
IoT	67.53 ± 1.54	65.02 ± 1.89	68.99 ± 2.14	59.53 ± 2.12	71.65 ± 0.71	**73.29 ± 0.68**
HAR	60.49 ± 1.12	60.01 ± 1.38	61.86 ± 1.13	56.75 ± 2.03	68.52 ± 0.76	**69.64 ± 0.71**

整体优于监督学习算法 LB、BOLE 和 ARF_{RE}; 所有算法在平衡数据流上的分类性能优于非平衡比率固定的非平衡数据流, 且在非平衡比率固定的非平衡数据流上的分类性能优于非平衡比率可变的非平衡数据流; 所有算法在不包含概念漂移和异常点的数据流上的分类性能优于包含概念漂移和异常点的数据流; 所有算法在高维数据流上的分类性能均有不同程度的下降.

ROC 曲线可以直观地通过图示分析算法分类性能的优劣, 实验结果如图 2 所示. ROC 曲线下区域的面积可以反映算法分类性能的优劣, 面积越大, 表示分类性能越好. 由图 2 可知, OALM-IDS 算法的 ROC 曲线下区域的面积在 4 个人工数据流 (DS_3 ~ DS_6) 和 4 个真实数据流中均为最大. 仅在数据流 DS_1 和 DS_2 上的面积比 ARF_{RE} 略小, 但也极为接近. 这是由于 ARF_{RE} 算法为监督学习算法, 即需要使用所有样本的标签信息, 并且该算法仅在对平衡数据流分类时 ROC 曲线下区域的面积最大. 所有算法在高维数据流上的 ROC 曲线下区域面积均偏小.

图 3 ~ 5 展示了主动学习算法 OALM-IDS、OALE 和 CALMID 在 3 个较为复杂的数据流 (DS_6、Kddcup 99_10% 和 Statlog) 上的准确率随样本规模增加的变化曲线. 由图 3 可知, OALM-IDS 算法在处理非平衡比率可变、含有异常点和概念漂移的数据流 DS_6 时, 可以用最少的标记样本获得最高的准确率, 明显优于算法 OALE 和 CALMID, 并且优于监督学习算法 LB、BOLE 和 ARF_{RE}. 由图 4 和图 5 可知, OALM-IDS 算法优于主动学习算法 OALE 和 3 种监督学习算法, 与 CALMID 算法性能较为接近, 但当数据流中出现概念漂移时, 本文提出的 OALM-IDS 算法可以用更少的标记样本获取更高

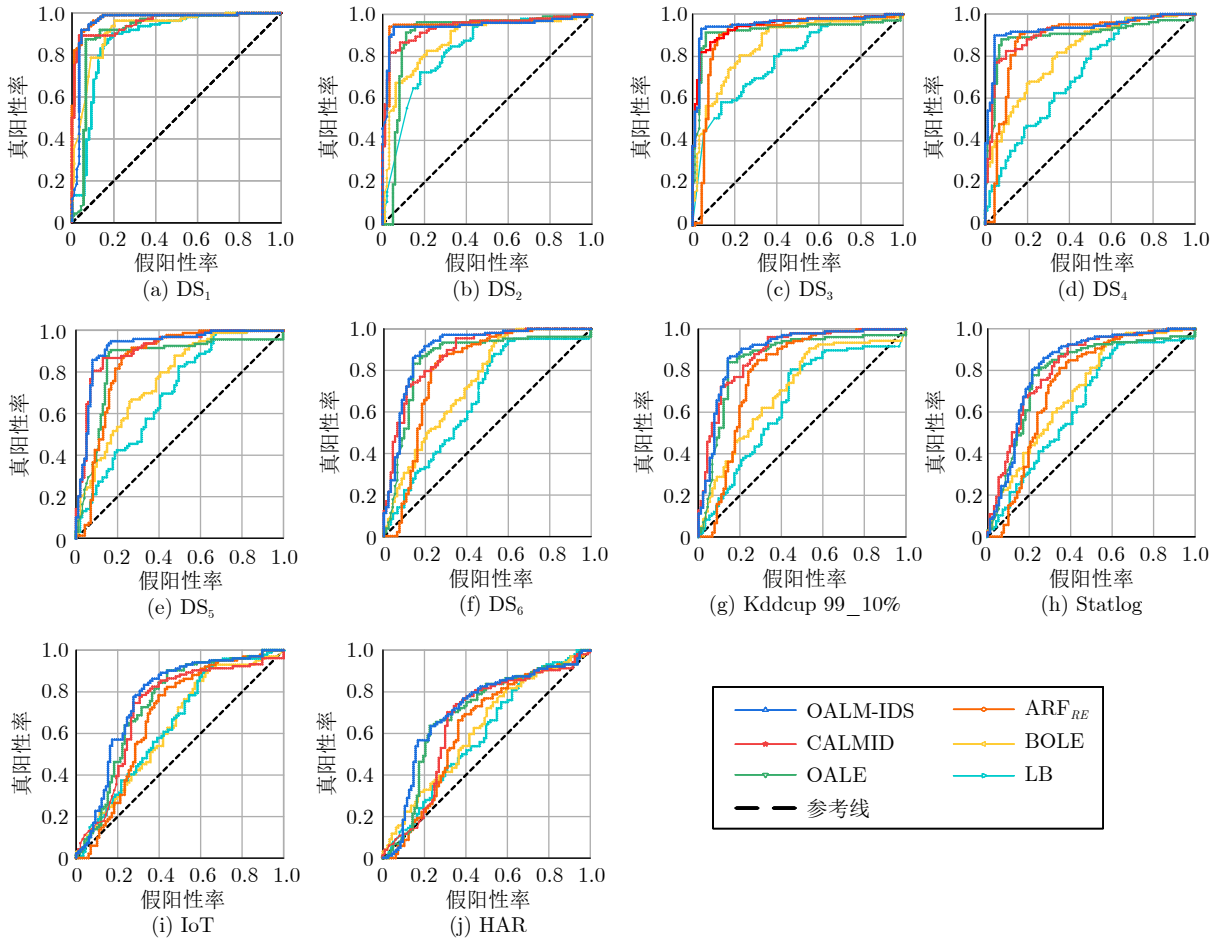

图 2 6 种算法的 ROC 曲线

Fig. 2 ROC curve of six algorithms

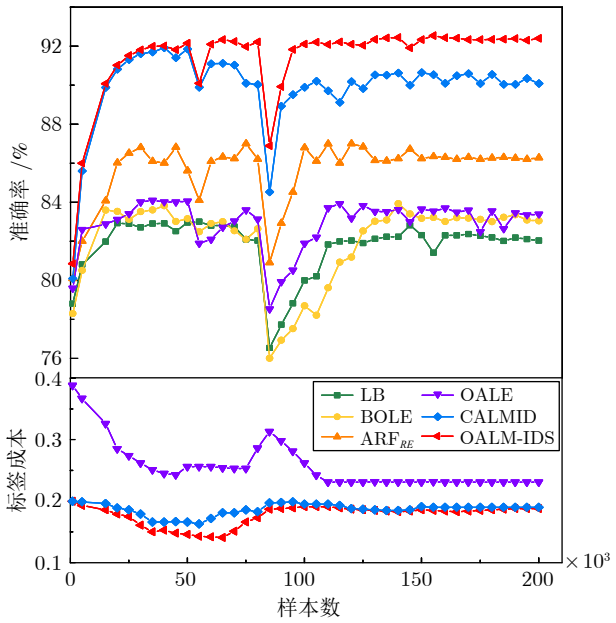

图 3 DS_6 的准确率曲线

Fig. 3 Precision curve of the DS_6

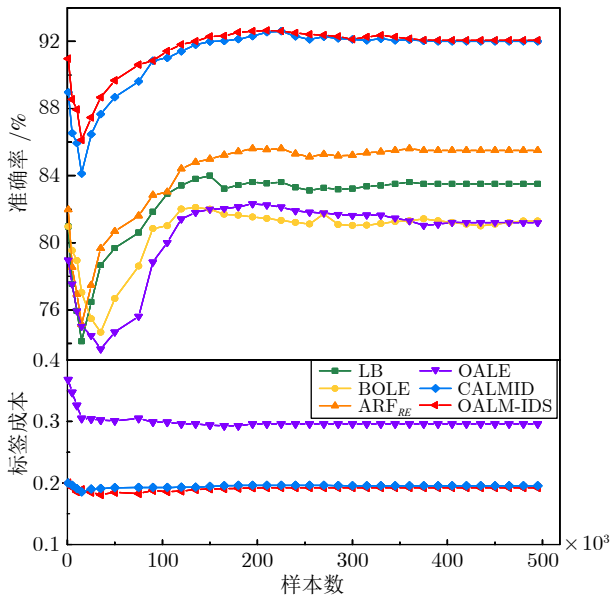

图 4 Kddcup 99_10% 的准确率曲线

Fig. 4 Precision curve of the Kddcup 99_10%

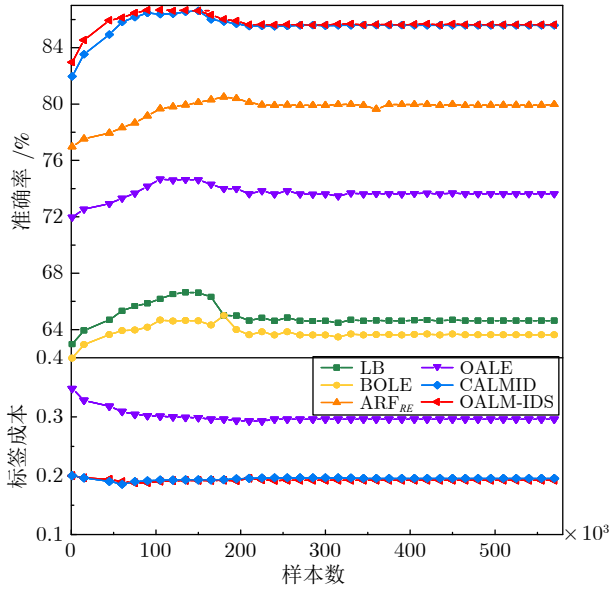

图 5　Statlog 的准确率曲线

Fig. 5　Precision curve of the Statlog

准确率, 优于 CALMID 算法.

基于上述实验结果可知, 本文提出的 OALM-IDS 算法在 3 个主动学习算法中分类性能是最优的, 这是由于 OALM-IDS 算法在以下 3 个方面做了改进: 1) 将 AdaBoost.M2 方法用于非平衡数据流的分类; 2) 引入基于边际阈值矩阵的自适应标签请求策略, 用于解决训练过程中由数据分布改变引起的非平衡比率的变化; 3) 基于概念漂移指数定义了自适应遗忘因子, 使得分类模型可以更快地适应新数据.

4.3　消融实验

为了测试 OALM-IDS 中引入 AdaBoost.M2 集成分类算法、自适应遗忘因子 α 和异常点检测机制的作用, 进行了以下 3 种消融实验: 1) 从 OALM-IDS 中去掉异常点检测机制, 得到 OALM-IDS-o 算法; 2) 在 OALM-IDS-o 算法的基础上, 去除自适应遗忘因子 α, 形成 OALM-IDS-oa 算法; 3) 将 OALM-IDS-oa 算法中的 AdaBoost.M2, 替换成普通的决策树集成分类器, 形成 OALM-IDS-oab.

本文实验使用非平衡比率可变的且包含概念漂移和异常点的人工数据流 DS_6, 实验结果如图 6 所示. 可见, 去掉异常点检测机制后的 OALM-IDS-o 算法比 OALM-IDS 算法的准确率有所降低, 进一步去掉自适应遗忘因子 α 后的 OALM-IDS-oa 算法的准确率继续下降. 而将 OALM-IDS-oa 算法中的 AdaBoost.M2 替换成普通的决策树集成分类器后, 算法 OALM-IDS-oab 的准确率下降明显, 可见

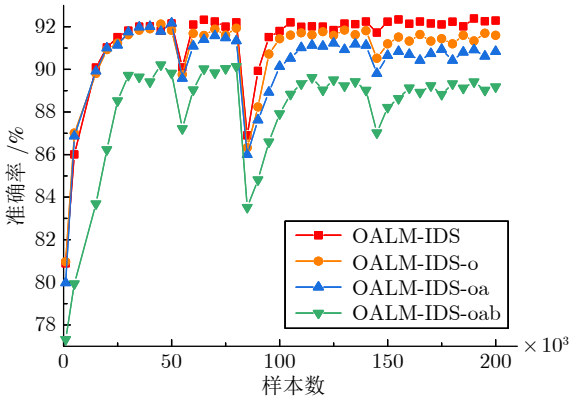

图 6　消融实验结果

Fig. 6　Result of the ablation experiment

AdaBoost.M2 的引入对整个算法准确率的提升贡献很大.

4.4　参数 θ 对 OALM-IDS 算法的影响

边际阈值矩阵的阈值 θ 对应请求样本标签的可能性, 下面测试初始阈值 θ 对 OALM-IDS 算法分类性能的影响, 重复实验 10 次, 实验结果如表 6 所示. 可见当初始阈值 $\theta = 0.5$ 时, 算法的准确率、召回率、F1 值和 Kappa 系数值最高.

当 $\theta = 0.4$ 时, 请求样本标签的可能性减小, 这会导致那些难分、少数类样本请求不到真实标签, 最终使得 OALM-IDS 算法性能下降; 当 $\theta = 0.6$ 时, 请求样本标签的可能性增大, 但由于标签预算 b 是一定的, 这也会导致难分、少数类样本请求不到足够标签, 也会使得 OALM-IDS 算法性能下降.

5　结束语

本文研究多类非平衡、概念漂移和异常点并存的数据流在少量真实标签情况下的在线分类问题, 提出一种非平衡漂移数据流在线主动学习方法. 定义了基于非平衡比率和自适应遗忘因子的训练样本重要性度量, 使得 AdaBoost.M2 适用于非平衡数据流环境; 提出基于样本分类不确定程度的自适应标签请求策略, 使得难分和少数类样本可以获得更多的训练机会; 定义了基于分类偏差的概念漂移指数, 并将其引入时间衰减机制, 用于模型的重构.

为了增强非平衡数据流在线主动学习方法的鲁棒性, 在未来工作中, 将关注以下问题. 首先, 在有新类的流数据中, 要考虑如何评估异常点的分布情况. 其次, 除了通过主动学习解决样本标签稀缺问题外, 还可以尝试结合迁移学习研究更有效的方法.

表 6 参数 θ 对 OALM-IDS 的影响

Table 6 Effect of parameter θ to OALM-IDS

数据流	θ	b	准确率	召回率	F1 值	Kappa 系数值
DS_1	0.4	0.17143	94.21 ± 0.16	93.18 ± 0.12	94.13 ± 0.11	90.11 ± 0.12
	0.5	**0.180 26**	**95.48 ± 0.15**	**96.14 ± 0.12**	**95.80 ± 0.10**	**91.31 ± 0.12**
	0.6	0.19782	95.03 ± 0.15	93.19 ± 0.12	95.16 ± 0.10	91.01 ± 0.12
DS_2	0.4	0.17136	93.01 ± 0.12	92.81 ± 0.16	93.04 ± 0.13	89.09 ± 0.26
	0.5	**0.191 78**	**93.94 ± 0.12**	**94.08 ± 0.14**	**94.01 ± 0.12**	**90.66 ± 0.23**
	0.6	0.20000	93.18 ± 0.13	93.16 ± 0.14	93.75 ± 0.12	90.07 ± 0.23
DS_3	0.4	0.17821	93.24 ± 0.13	92.05 ± 0.13	92.54 ± 0.16	88.56 ± 0.22
	0.5	**0.195 12**	**93.72 ± 0.13**	**92.52 ± 0.13**	**93.07 ± 0.14**	**89.93 ± 0.21**
	0.6	0.20000	93.43 ± 0.13	92.24 ± 0.13	92.10 ± 0.14	88.71 ± 0.21
DS_4	0.4	0.18423	91.63 ± 0.21	91.34 ± 0.18	91.76 ± 0.20	88.54 ± 0.38
	0.5	**0.198 77**	**92.18 ± 0.21**	**92.44 ± 0.18**	**92.29 ± 0.20**	**89.33 ± 0.36**
	0.6	0.20000	91.06 ± 0.21	91.56 ± 0.19	91.80 ± 0.21	88.63 ± 0.36
DS_5	0.4	0.18002	92.01 ± 0.16	90.46 ± 0.13	90.76 ± 0.18	88.42 ± 0.29
	0.5	**0.197 22**	**92.92 ± 0.16**	**91.16 ± 0.13**	**91.93 ± 0.18**	**89.12 ± 0.29**
	0.6	0.20000	92.50 ± 0.16	90.76 ± 0.13	91.21 ± 0.19	88.56 ± 0.30
DS_6	0.4	0.18331	91.02 ± 0.21	89.03 ± 0.22	90.32 ± 0.31	88.12 ± 0.28
	0.5	**0.199 23**	**92.41 ± 0.21**	**90.71 ± 0.21**	**91.53 ± 0.31**	**89.73 ± 0.28**
	0.6	0.20000	91.01 ± 0.21	89.92 ± 0.22	90.12 ± 0.31	89.13 ± 0.28
Kddcup 99_10%	0.4	0.18188	90.59 ± 0.18	63.51 ± 0.37	73.35 ± 0.20	83.14 ± 0.18
	0.5	**0.199 61**	**92.07 ± 0.18**	**63.71 ± 0.37**	**74.65 ± 0.20**	**85.83 ± 0.18**
	0.6	0.20000	91.63 ± 0.18	63.63 ± 0.37	74.43 ± 0.21	85.61 ± 0.18
Statlog	0.4	0.19022	84.75 ± 0.33	62.19 ± 0.39	74.85 ± 0.31	78.86 ± 0.19
	0.5	**0.199 94**	**85.68 ± 0.33**	**63.12 ± 0.39**	**75.19 ± 0.31**	**80.11 ± 0.19**
	0.6	0.20000	85.66 ± 0.33	63.01 ± 0.39	75.19 ± 0.31	79.89 ± 0.19
IoT	0.4	0.19113	71.21 ± 0.38	49.86 ± 0.81	51.21 ± 0.67	71.61 ± 0.68
	0.5	**0.196 84**	**73.12 ± 0.38**	**51.26 ± 0.81**	**56.73 ± 0.67**	**73.29 ± 0.68**
	0.6	0.20000	72.11 ± 0.39	50.06 ± 0.81	54.33 ± 0.67	71.34 ± 0.68
HAR	0.4	0.18634	66.54 ± 0.52	64.32 ± 0.48	65.05 ± 0.59	66.81 ± 0.72
	0.5	**0.195 47**	**69.98 ± 0.51**	**66.57 ± 0.46**	**67.76 ± 0.58**	**69.64 ± 0.71**
	0.6	0.20000	64.32 ± 0.52	65.14 ± 0.46	66.11 ± 0.58	64.32 ± 0.71

References

1 Yu Hong, He De-Niu, Wang Guo-Yin, Li Jie, Xie Yong-Fang. Big data for intelligent decision making. *Acta Automatica Sinica*, 2020, **46**(5): 878−896
(于洪, 何德牛, 王国胤, 李劼, 谢永芳. 大数据智能决策. 自动化学报, 2020, **46**(5): 878−896)

2 Lu J, Liu A, Dong F, Gu F, Gama J, Zhang G. Learning under concept drift: A review. *IEEE Transactions on Knowledge and Data Engineering*, 2020, **31**(12): 2346−2363

3 Liu W, Zhang H, Liu Q. An air quality grade forecasting approach based on ensemble learning. In: Proceedings of the International Conference on Artificial Intelligence and Advanced Manufacturing. Dublin, Ireland: AIAM, 2019. 87−91

4 Cano A, Krawczyk B. Kappa updated ensemble for drifting data stream mining. *Machine Learning*, 2020, **109**(1): 175−218

5 Liu A, Lu J, Zhang G. Concept drift detection via equal intensity k-means space partitioning. *IEEE Transactions on Cybernetics*, 2020, **51**(6): 3198−3211

6 Wang Jin-Jia, Zhang Yu-Zhen, Xia Jing, Wang Feng-Pin. Multi-layer local block coordinate descent algorithm and unfolding classification and reconstruction networks. *Acta Automatica Sinica*, 2020, **46**(12): 2647−2661
(王金甲, 张玉珍, 夏静, 王凤嫔. 多层局部块坐标下降法及其驱动的分类重构网络. 自动化学报, 2020, **46**(12): 2647−2661)

7 Lu Y, Cheung M Y, Tang Y Y. Adaptive chunk-based dynamic weighted majority for imbalanced data stream with concept drift. *IEEE Transactions on Neural Networks and Learning Systems*, 2020, **31**(8): 2764−2778

8 Grzyb J, Klikowski J, Woźniak M. Hellinger distance weighted ensemble for imbalanced data stream classification. *Journal of Computational Science*, 2021, **51**: Article No. 101314

9 Kim T, Park C H. Anomaly pattern detection for streaming data. *Expert Systems With Applications*, 2020, **149**: Article No. 113252

10 Wankhade K K, Dongre S S, Jondhale K C. Data stream classification: A review. *Iran Journal of Computer Science*, 2020, **3**: 239−260

11 Bahri M, Bifet A, Gama J, Gomes H M, Maniu S. Data stream analysis: Foundations, major tasks and tools. *Wiley Interdiscip-*

linary Reviews: Data Mining and Knowledge Discovery, 2021, **11**(3): Article No. e1405

12 Kontopoulos I, Chatzikokolakis K, Tserpes K, Zissis D. Classification of vessel activity in streaming data. In: Proceedings of the 14th ACM International Conference on Distributed and Event-based Systems. Jerusalem, Israel: ACM, 2020. 153−164

13 Wang S, Minku L L. Auc estimation and concept drift detection for imbalanced data streams with multiple classes. In: Proceedings of the International Joint Conference on Neural Networks. Glasgow, UK: IJCNN, 2020. 1−8

14 Fan S, Zhang X, Song Z. Reinforced knowledge distillation: Multi-class imbalanced classifier based on policy gradient reinforcement learning. *Neurocomputing*, 2021, **463**: 422−436

15 Bifet A, Holmes G, Pfahringer B. Leveraging bagging for evolving data stream. In: Proceedings of the Joint European Conference on Machine Learning and Knowledge Discovery in Databases. Barcelona, Spain: PKDD, 2010. 135−150

16 Mirza B, Lin Z. Meta-cognitive online sequential extreme learning machine for imbalanced and concept drifting data classification. *Neural Networks*, 2016, **80**: 79−94

17 Barros R S M, Carvalho-Santos S G T, Júnior P M G. A boosting-like online learning ensemble. In: Proceedings of the International Joint Conference on Neural Networks. Vancouver, Canada: IJCNN, 2016. 1871−1878

18 Carvalho-Santos S G T, Barros R S M. Online AdaBoost-based methods for multi-class problems. *Artificial Intelligence Review*, 2020, **53**(2): 1293−1322

19 Ferreira L E B, Gomes H M, Bifet A, Oliveira L S. Adaptive random forests with resampling for imbalanced data stream. In: Proceedings of the International Joint Conference on Neural Networks. Budapest, Hungary: IJCNN, 2019. 1−6

20 Ren P Z, Xiao Y, Chang X J, Huang P Y, Li Z, Gupta B B, et al. A survey of deep active learning. *ACM Computing Surveys*, 2021, **54**(9): 1−40

21 Yousaf M S, Ahmad I, Khurshid A, Ikram M. Machine assisted classification of chicken, beef and mutton tissues using optical polarimetry and bagging model. *Photodiagnosis and Photodynamic Therapy*, 2020, **31**: Article No. 101779

22 Wang Y, Feng L. An adaptive boosting algorithm based on weighted feature selection and category classification confidence. *Applied Intelligence*, 2021, **51**(10): 1−22

23 Gomes H M, Bifet A, Read J, Barddal J P, Enembreck F, Pfharinger B, et al. Adaptive random forests for evolving data stream classification. *Machine Learning*, 2017, **106**(9): 1469−1495

24 Babüroğlu E S, Durmuşoğlu A, Dereli T. Novel hybrid pair recommendations based on a large-scale comparative study of concept drift detection. *Expert Systems With Applications*, 2021, **163**: Article No. 113786

25 Liu Zi-Ang, Jiang Xue, Wu Dong-Rui. Unsupervised pool-based active learning for linear regression. *Acta Automatica Sinica*, 2021, **47**(12): 2771−2783
(刘子昂, 蒋雪, 伍冬睿. 基于池的无监督线性回归主动学习. 自动化学报, 2021, **47**(12): 2771−2783)

26 Shekhar S, Ghavamzadeh M, Javidi T. Active learning for classification with abstention. *IEEE Journal on Selected Areas in Information Theory*, 2021, **2**(2): 705−719

27 Shan J, Zhang H, Liu W, Liu Q. Online active learning ensemble framework for drifted data stream. *IEEE Transactions on Neural Networks and Learning Systems*, 2018, **30**(2): 486−498

28 Liu W, Zhang H, Ding Z, Liu Q, Zhu C. A comprehensive active learning method for multi-class imbalanced data stream with concept drift. *Knowledge-based Systems*, 2021, **215**: Article No. 106778

29 Gu X, Angelov P P. Multi-class fuzzily weighted adaptive boosting-based self-organising fuzzy inference ensemble systems for classification. *IEEE Transactions on Fuzzy Systems*, 2021, **30**(9): 3722−3735

30 Bifet A, Holmes G, Kirkby R, Pfahringer B. MOA: Massive online analysis. *Journal of Machine Learning Research*, 2010, **11**: 1601−1604

李艳红 山西大学计算机与信息技术学院副教授. 主要研究方向为数据挖掘, 机器学习. 本文通信作者.
E-mail: liyh@sxu.edu.cn
(**LI Yan-Hong** Associate professor at the School of Computer and Information Technology, Shanxi University. Her research interest covers data mining and machine learning. Corresponding author of this paper.)

任 霖 山西大学计算机与信息技术学院硕士研究生. 主要研究方向为数据挖掘, 机器学习.
E-mail: renlinssdx@163.com
(**REN Lin** Master student at the School of Computer and Information Technology, Shanxi University. His research interest covers data mining and machine learning.)

王素格 山西大学计算机与信息技术学院教授. 主要研究方向为自然语言处理, 机器学习.
E-mail: wsg@sxu.edu.cn
(**WANG Su-Ge** Professor at the School of Computer and Information Technology, Shanxi University. Her research interest covers natural language processing and machine learning.)

李德玉 山西大学计算机与信息技术学院教授. 主要研究方向为数据挖掘, 人工智能. E-mail: lidy@sxu.edu.cn
(**LI De-Yu** Professor at the School of Computer and Information Technology, Shanxi University. His research interest covers data mining and artificial intelligence.)