非平衡概念漂移数据流主动学习方法

李艳红 [1, 2] 王甜甜 [1, 2] 王素格 [1, 2] 李德玉 [1, 2]

摘 要 数据流分类研究在开放、动态环境中如何提供更可靠的数据驱动预测模型, 关键在于从实时到达且不断变化的数据流中检测并适应概念漂移. 目前, 为检测概念漂移和更新分类模型, 数据流分类方法通常假设所有样本的标签都是已知的, 这一假设在真实场景下是不现实的. 此外, 真实数据流可能表现出较高且不断变化的类不平衡比率, 会进一步增加数据流分类任务的复杂性. 为此, 提出一种非平衡概念漂移数据流主动学习方法 (Active learning method for imbalanced concept drift data stream, ALM-ICDDS). 定义基于多预测概率的样本预测确定性度量, 提出边缘阈值矩阵的自适应调整方法, 使得标签查询策略适用于类别数较多的非平衡数据流; 提出基于记忆强度的样本替换策略, 将难区分、少数类样本和代表当前数据分布的样本保存在记忆窗口中, 提升新基分类器的分类性能; 定义基于分类精度的基分类器重要性评价及更新方法, 实现漂移后的集成分类器更新. 在 7 个合成数据流和 3 个真实数据流上的对比实验表明, 提出的非平衡概念漂移数据流主动学习方法的分类性能优于 6 种概念漂移数据流学习方法.

关键词 数据流分类, 主动学习, 概念漂移, 多类不平衡

引用格式 李艳红, 王甜甜, 王素格, 李德玉. 非平衡概念漂移数据流主动学习方法. 自动化学报, 2024, **50**(3): 589−606

DOI 10.16383/j.aas.c230233

Active Learning Method for Imbalanced Concept Drift Data Stream

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

Abstract Data stream classification researchs how to provide more reliable data-driven prediction models in open and dynamic environment. The key is how to detect and adapt to concept drift from continuously changing data stream that arrive in real-time. Currently, in order to detect concept drift and update classification models, data stream classification methods usually assume that the labels of all samples are known, which is unrealistic in real scenarios. Additionally, real data stream may exhibit a high and constantly changing class imbalance ratios, further increasing the complexity of the data stream classification task. In this paper, we propose an active learning method for imbalanced concept drift data stream (ALM-ICDDS). Firstly, we define a sample prediction certainty measure based on multiple prediction probabilities and propose an adaptive adjustment method for the margin threshold matrix, which makes the label query strategy suitable for imbalanced data stream with a number of categories. Then, we propose a sample replacement strategy based on memory strength, which saves the samples that are difficult-to-distinguish, minority class and represent the current data distribution in the memory window, and improves the classification performance of new base classifier. Finally, we define the importance evaluation and update method of base classifier based on classification accuracy, which realizes the ensemble classifier update after drift. Comparative experiments on seven synthetic data streams and three real data streams show that the active learning method for imbalance concept drift data stream is better than six concept drift data stream learning methods in classification performance.

Key words Data stream classification, active learning, concept drift, multi-class imbalance

Citation Li Yan-Hong, Wang Tian-Tian, Wang Su-Ge, Li De-Yu. Active learning method for imbalanced concept drift data stream. *Acta Automatica Sinica*, 2024, **50**(3): 589−606

随着互联网和移动通讯技术的发展, 流数据变得越来越普遍, 如社交网络、超市交易、传感器网络、垃圾邮件过滤等领域的数据往往都是以流的形式出现, 具有实时、动态变化、潜在无穷等特点[1]. 为从复杂多变的数据流中挖掘有价值的信息, 迫切需要研究面向数据流的高效学习方法实时捕获数据流中

收稿日期 2023-04-24 录用日期 2023-10-12
Manuscript received April 24, 2023; accepted October 12, 2023
国家重点研发项目 (2022QY0300-01), 国家自然科学基金 (62076158), 山西省基础研究计划项目 (202203021221001) 资助
Supported by National Key Research and Development Program of China (2022QY0300-01), National Natural Science Foundation of China (62076158), and Fundamental Research Program of Shanxi Province (202203021221001)
本文责任编委 张敏灵
Recommended by Associate Editor ZHANG Min-Ling
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

变化的簇结构[2].

在实际应用中, 数据流的概念漂移、标签成本昂贵和多类不平衡特性给数据流分类任务带来挑战, 这些特性往往同时存在、相互影响, 使得数据流的分类任务变得更加复杂[3]. 数据流的概念漂移是指随着时间的推移其数据分布发生不可预见的变化, 从而使得之前训练的分类模型不再适用[4]. 目前概念漂移的处理包括主动检测[5] 和被动适应[6] 两种方法. 主动检测方法通常基于分类模型性能的下降或数据分布的变化检测概念漂移, 只有在检测到概念漂移时才会更新分类模型[7]; 被动适应方法通常以固定的时间间隔采样数据流中的样本并定期调整分类模型, 无需进行概念漂移检测[8]. 被动适应方法以恒定的速度更新模型, 不具有针对性, 时间开销较大[9], 因此本文采用主动检测方法来处理概念漂移. 根据数据分布随时间推移变化速度和形式的不同, 可将概念漂移分为 4 种类型: 突变型、重复型、增量型和逐渐型. 其中突变型概念漂移是指在某一时刻, 旧数据分布突然变化为新数据分布; 重复型概念漂移是指发生突然型概念漂移后, 新数据分布维持一段时间又突然变为旧数据分布; 增量型概念漂移是指旧数据分布在一段时间内缓慢变化为新数据分布, 缓慢变化过程的数据分布为新旧数据分布的混合; 逐渐型概念漂移是指在一段时间内新数据分布逐渐取代旧数据分布, 在变化过程中新旧数据分布交替出现[10]. 为检测概念漂移以及更新分类模型, 现有的方法通常假定可以不受限制地访问数据流中所有样本的标签 (即有监督的学习方法), 这在实际应用中是不现实的[11]. 而主动学习方法可以通过查询少量最有价值的样本标签构建分类模型, 进而解决标签成本昂贵的问题[12−13].

数据流的类不平衡特性会使分类模型对少数类样本的学习不够充分, 从而导致模型的分类准确率下降[14]; 而且类不平衡比率可能会随着时间的推移发生变化, 在设计标签查询策略时需要特别关注[15]. 类不平衡问题可以从数据[16] 和算法[17] 两个层面加以解决. 数据层面是指通过减少多数类样本或增加少数类样本来平衡类分布, 但这种方式会丢失有用信息或增加过拟合的风险. 算法层面是指通过给少数类样本增加额外的损失代价, 或者基于 Boosting 方法重新训练分错的样本, 使得分类模型更加关注少数类样本. 本文在标签查询和基分类器重构时分别使用基于算法层面和数据层面的类不平衡处理方式.

本文提出一种非平衡概念漂移数据流主动学习方法 (Active learning method for imbalanced concept drift data stream, ALM-ICDDS), 该方法包括初始化阶段和在线学习阶段. 在初始化阶段训练初始集成分类器. 在线学习阶段, 首先使用初始集成分类器预测样本并计算样本的预测确定性, 基于该确定性和边缘阈值矩阵查询样本标签; 然后为了将难区分、少数类和代表当前数据分布的样本保存在记忆窗口中, 用新查询到标签的样本替换窗口中记忆强度最低的样本; 最后当检测到概念漂移时, 利用记忆窗口中的样本构建新基分类器并替换重要性最低的已有基分类器. 本文的主要贡献有以下几点:

1) 提出基于多预测概率的样本预测确定性度量以及边缘阈值矩阵的自适应调整方法, 从而使标签查询策略适用于类别数较多的非平衡数据流.

2) 提出基于记忆强度的样本替换策略, 将难区分、少数类和代表当前数据分布的样本保存在记忆窗口中, 并用于漂移后基分类器的重构.

3) 定义基于分类精度的基分类器重要性评价及更新方法, 提出一种集成分类器更新机制.

1 相关工作

本节将分别介绍基于监督学习和主动学习的非平衡概念漂移数据流分类方法的研究现状.

1.1 数据流监督学习方法

在数据流监督学习方法中, 依据对数据流处理方式的不同, 可分为基于数据块的批处理方法和基于单个样本的在线学习方法.

基于数据块的批处理方法首先将数据流划分成固定大小的块, 然后对数据块采样来实现类平衡, 采样的样本用于训练分类模型. Gao 等[18] 提出一种基于数据块的类不平衡数据流集成分类方法, 通过累计历史数据块中的少数类样本和对当前数据块中的多数类样本随机欠采样来实现类平衡, 并用于训练新基分类器来适应概念漂移, 但该算法需要累积足够多的样本才可以新建基分类器, 因此缺乏快速适应新概念的能力. 基于此, Lu 等[19] 提出一种基于基分类器动态加权的集成分类方法, 该方法通过对每个块中的多数类样本欠采样来实现类平衡, 并为每个块创建一个新的基分类器来及时学习新概念, 同时淘汰权重小于阈值的基分类器来提高算法效率. 但是在真实数据流中概念漂移的位置是不可预知的, 因此, 此类方法面临的主要问题是难以确定合适的块大小[20].

基于单个样本的在线学习方法首先用最初的一部分样本建立初始分类模型, 然后在线学习阶段, 用每个新到来样本动态更新分类模型[21]. Wang 等[22]

提出在线过采样 (Oversampling online bagging, OOB) 和在线欠采样 (Undersampling online bagging, UOB) 两种方法来解决数据流中的类不平衡问题, 这两个方法基于一个时间衰减函数来实时计算当前的类不平衡比率, 当检测到类不平衡比率发生严重的倾斜时, 基于重采样的样本重新训练分类模型, 但这两种方法没有检测数据流中的概念漂移. Cano 等[23] 提出一种在线自调整集成分类方法 (Robust online self-adjusting ensemble, ROSE), 该方法为每类样本设置一个滑动窗口缓冲区来保存最新的样本, 并基于自适应滑动窗口 ADWIN 方法[24] 比较两个子窗口之间错误率的差异程度来检测概念漂移, 当检测到发生概念漂移时, 重构一个新的集成分类器, 并与旧的集成分类器组合选出一组性能最好的基分类器. 以上方法只能解决数据流分类中的二类不平衡问题, 对于多类不平衡问题, Barros 等[25] 基于 AdaBoost 提出 BOLE (Boosting-like online learning ensemble) 方法, 该方法根据类不平衡比率为样本设置初始化权重, 同时使用 AdaBoost 将弱分类器逐步迭代为强分类器, 并使用 DDM[26] 算法来检测概念漂移, 当检测到发生概念漂移则重构分类模型. 但基于 AdaBoost 的集成分类方法时间复杂度较高, 尤其当基分类器的误差率较高时, 基分类器需要迭代更多的次数, 同时也容易出现过拟合的现象[27]. Bifet 等[28] 基于 Bagging 提出 LB (Leverage bagging) 方法, 该方法对每个新到来样本都设置一个随机权重, 并基于 ADWIN 来检测概念漂移, 当发生概念漂移时, 新建一个基分类器来淘汰分类效果最差的基分类器. Ferreira 等[29] 基于随机森林进行改进提出 ARFRE (Adaptive random forest with resampling) 方法, 该方法通过计算数据流中每类样本出现的次数来评估当前的类不平衡比率, 然后与泊松分布的输出相结合作为样本的训练权重, 从而增加少数类样本用于训练的机会. 基于随机森林的方法在 Bagging 的基础上引入随机属性的选择, 进一步增强基学习器之间的差异和独立性[30].

以上数据流监督学习方法需要获取全部样本的标签, 然而在实际应用中获取全部样本标签的代价往往十分高, 此时该类方法难以得到很好的应用.

1.2 数据流主动学习方法

主动学习方法通过人工标注难区分样本, 并将其用于分类模型的更新, 通过将专家知识融入分类模型, 实现利用少量带标签样本完成数据流的分类任务[31]. 主动学习方法的性能很大程度上依赖于标签查询策略的好坏[32]. 目前标签查询策略方法有随机策略、不确定性策略, 以及混合策略[33−36]. 随机策略是指随机选择样本进行标签查询. 不确定性策略是指根据模型对样本的预测不确定性程度选择样本进行标签查询. 混合策略则是结合随机策略和不确定性策略的一种综合方法. 数据流主动学习需要全面考虑类不平衡、概念漂移和异常点等因素对分类器性能的影响, 设计高效的标签查询策略, 从而提高数据流主动学习方法的性能.

针对二分类数据流, Xu 等[33] 提出一种基于混合标签查询策略的数据流主动学习方法, 包含用于样本预测的静态分类器和检测概念漂移的动态分类器. 该方法使用混合标签查询策略查询到的样本更新静态分类器, 基于动态分类器预测样本的错误率变化来检测概念漂移. 当发生概念漂移时用动态分类器取代静态分类器, 并基于随机策略查询到的样本重新训练新的动态分类器. Liu 等[34] 提出一种二分类概念漂移数据流在线主动学习方法, 该方法除使用混合标签查询策略外, 还基于数据流样本的局部密度来查询标签 (局部密度值大的样本更具有代表性), 并使用查询到标签的样本更新分类模型以适应概念漂移.

上述两个主动学习方法没有考虑数据的平衡性, 不适用于多类不平衡数据流. Liu 等[35] 提出一种概念漂移多类不平衡数据流主动学习方法 (A comprehensive active learning method for multiclass imbalanced streaming data with concept drift, CALMID), 该方法使用一个非对称边缘阈值矩阵来存储样本预测的不确定性阈值, 该方法在调整边缘阈值矩阵时, 采用固定的比率, 没有考虑样本预测的不确定程度. 李艳红等[36] 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), 该方法定义基于不平衡比率的样本重要性度量, 使得 AdaBoost.M2 适用于非平衡数据流, 提出基于样本预测不确定程度的边缘阈值矩阵自适应调整方法, 并定义基于概念漂移指数和不平衡比率的样本重要性度量, 实现漂移后的模型重构. 方法 CALMID 和 OALM-IDS 基于样本预测的两个最大概率值之差进行标签查询, 不适用于数据流中样本类别数较多的情况; 并基于先进先出的样本滑动窗口保存查询到标签的样本, 没有考虑样本被“回忆”的次数和样本的预测误差.

集成分类方法常用于数据流在线学习, 当发生概念漂移时, 旧基分类器很可能不再适用, 需要被替换为适用于当前数据分布的新基分类器. 研究者

针对基分类器的选择、组合和动态更新开展相关研究. Zhao 等[37] 提出一种通过模型重用处理概念漂移的方法, 该方法基于当前数据块的分类性能自适应地为已有模型分配权重, 将已有模型的加权组合作为新分类模型从而实现模型重用并适应概念漂移. Karimi 等[38] 提出一种基于主动学习的预训练模型选择方法, 该方法基于标签查询策略选择信息较丰富的样本进行人工标注, 并根据标注样本的分类性能为分类模型分配权重. Zyblewski 等[39] 提出一种面向高度不平衡漂移数据流的预处理和分类器动态集成选择方法, 该方法结合欠采样和过采样对非平衡数据流进行预处理, 将基于 Bagging 生成的多个分类模型加入模型池, 根据当前数据块的分类性能选择分类模型以适应概念漂移.

本文的研究工作将综合考虑数据流分类任务中面临的概念漂移、标签成本昂贵和多类不平衡问题, 从标签查询、记忆窗口维护和集成分类器更新三方面开展研究, 从而提高数据流主动学习方法的性能.

2 问题的形式化定义

设本文要处理的数据流 $DS = \{x_1, x_2, \cdots, x_i, \cdots\}$, 其中 x_i 表示第 i 个样本, t_i 为第 i 个样本到达的时刻, Y 为样本的类别标签, $Y = \{y_1, y_2, \cdots, y_k\}$, k 为样本的类别数. 由于数据流中的样本是实时到达并且无穷的, 在初始化阶段选取最初到达的 S_l 个样本用于构建初始集成分类器 E, 同时将这 S_l 个样本的类别标签存入标签滑动窗口 $LW[S_l]$, 用于计算类不平衡比率 $imb^y_{t_i}$.

在线学习阶段, 用初始集成分类器 E 对新样本 x_i 进行类别预测, 同时定义样本预测的确定性度量 $D(x_i)$. 为筛选出难区分和少数类的样本进行人工标注, 定义 n_d 维边缘阈值矩阵 M, 用于保存集成分类器对样本预测的前 n_d 个可能类 (即前 n_d 大预测概率对应的类) 不同取值下的样本预测确定性阈值, 矩阵元素初值为 θ_0. 在新样本到达的同时, 采用先进先出的方式对 LW 进行更新, LW 用于保存最近 S_l 个连续样本的标签 (基于随机策略查询到的样本存放真实类别标签, 基于不确定性策略查询到的样本和没有查询标签的样本存放 $Null$ 标签). 将人工标记的样本信息存储在记忆窗口 $MW[k][S_m]$ 中, S_m 为每类样本存储的个数, 样本信息包括样本被“回忆”的次数 λ_x、最后一次被“回忆”的时刻 τ_x 以及预测误差 $f(x)$, MW 用于计算样本的记忆强度 $S(x)$. 为在记忆窗口中保存难区分、少数类和代表当前数据分布的样本, 用新查询到标签的样本替换窗口中记忆强度最低的样本. 每当基于标签查询策略为样本查询到真实标签后, 调用 ADWIN 算法判断是否发生概念漂移. 如果发生概念漂移, 则使用 MW 中的样本重新训练新的基分类器 C_{new}, 同时更新集成分类器 E 中每个基分类器的重要性, 并用 C_{new} 替换重要性最低的基分类器.

本文采用 Hoeffding 树作为基分类器, 利用由多个 Hoeffding 树构成的集成分类器 E 对样本进行预测, 并在查询到样本标签时更新基分类器. 本文的数据流学习框架适用于其他分类模型, 例如可以使用支持向量机、极限学习机等作为基分类器, 并设计相应的模型更新方法.

3 本文方法

3.1 样本预测的确定性度量

数据流的主动学习就是在分类过程中查询那些预测确定性低和代表性样本的真实标签, 并将人工标注的样本用于分类器更新, 使得分类器适应当前的数据分布. 因此如何度量样本预测的确定性直接关系到选择哪些样本用于更新分类器. 现有的研究工作中, 样本预测的确定性度量通常基于置信度、边缘抽样和熵等方法. 置信度是分类器对样本预测最大概率值, 置信度越低, 表明分类器对该样本预测的确定性越低; 边缘抽样根据两个样本预测最大概率值之间的差值选择样本, 差值越小, 表明样本预测的确定性越低; 基于熵的方法通过度量分类器对样本所有预测概率的混乱程度来选择样本, 越混乱, 熵越大, 表明样本预测的确定性越低.

基于置信度的方法只考虑样本预测的最大概率值, 而忽略其他类别的预测信息; 边缘抽样方法只考虑两个最大概率值之间的差值, 忽略样本预测的最大概率值; 基于熵的方法对不同大小的预测概率是同等对待的, 但在度量样本预测确定性时, 需要考虑预测概率的序关系, 通常预测概率大的值之间的差异要比预测概率小的值之间的差异更重要. 为提高标签查询策略的有效性, 本文在度量样本预测的确定性时考虑下面两方面因素:

1) 基于分类器对样本预测的最大概率值和两个最大概率值之间的差值综合评价样本预测的确定性 (二者之和越大, 样本预测确定性越高, 反之确定性越低).

2) 当类别数较多时, 还应考虑样本预测最大概率值与其余预测概率的差值.

基于此, 本文定义样本预测的确定性度量 $D(x_i)$ 如下

$$D(x_i) = \beta \times P\left(y_{c_1}|x_i\right) + (1-\beta) \times \mu(x_i) \tag{1}$$

其中, $P(y_{c_1}|x_i)$ 表示样本预测的最大可能类概率, 相应的类标签为 y_{c_1}, 调节参数 $\beta \in [0, 1]$. $\mu(x_i)$ 表示最大可能类与其余类预测概率的平均差值, 计算方式为

$$\mu(x_i) = \frac{1}{n-1} \times \sum_{j=2}^{n} \left(P(y_{c_1}|x_i) - P(y_{c_j}|x_i)\right) \quad (2)$$

其中, $P(y_{c_j}|x_i)$ 表示样本预测的第 c_j 大概率值, 相应的类标签为 y_{c_j}. n 为确定性度量所使用的预测概率的类别数.

为筛选出预测确定性低的样本进行人工标注, 需要指定样本预测确定性阈值. 不确定性标签查询策略[34–35] 通常使用唯一的样本预测确定性阈值 (固定阈值或可变阈值), 无法刻画多分类问题中类之间区分难易程度的差异. 为解决这一问题, 本文使用边缘阈值矩阵保存样本预测确定性阈值. 在边缘阈值矩阵的动态调整过程中, 难区分类之间的确定性阈值通常小于易区分类之间的确定性阈值, 多数类对少数类的确定性阈值通常小于少数类对多数类的确定性阈值. 在标签查询时, 这种样本预测确定性阈值的设置方式倾向于给难区分和少数类样本更多的机会, 而不是易区分和多数类样本.

基于上述样本预测确定性度量 $D(x_i)$ 和边缘阈值矩阵 M, 本文提出一种标签查询策略, 根据样本分类难度查询样本的真实标签, 并基于样本预测结果和类不平衡比率对边缘阈值矩阵 M 进行自适应调整, 具体方式如下:

1) 如果 $D(x_i) > M[c_1, c_2, \cdots, c_{n_d}]$, 则表明分类器对样本 x_i 的预测结果较为确定, 因此不需要查询 x_i 的真实标签, 也不需要调整边缘阈值矩阵.

2) 如果 $D(x_i) \leq M[c_1, c_2, \cdots, c_{n_d}]$, 则表明分类器对样本 x_i 的预测结果不确定, 需要查询样本的真实类别标签 y. 若样本预测的最大可能类与真实类别 y 一致, 表明该样本预测正确, 无需查询其标签, 此时应降低阈值 $M[c_1, c_2, \cdots, c_{n_d}]$.

阈值 $M[c_1, c_2, \cdots, c_{n_d}]$ 的降低程度应与类不平衡比率成正比, 类不平衡比率越高, 则表明该类样本数量越多, 阈值的降低程度应该越高, 反之阈值的降低程度应该越低. 据此, 本文提出边缘阈值矩阵自适应调整方式为

$$M[c_1, c_2, \cdots, c_{n_d}] = M[c_1, c_2, \cdots, c_{n_d}] \times (1 - \alpha \times imb_{t_i}^{y}) \quad (3)$$

其中, $0 < \alpha < 1$, 不平衡比率 $imb_{t_i}^{y}$ 基于标签滑动窗口 LW 中累积的标签进行计算, 计算方式为

$$imb_{t_i}^{y} = \frac{labelnum_y}{S_l - nullnum} \quad (4)$$

其中, $labelnum_y$ 为 LW 中 y 类的标签个数, $nullnum$ 为 LW 中空标签的个数.

除上述基于样本预测确定性的标签查询策略外, 本文还随机地从数据流中选择一些样本进行标签查询 (称为代表性样本), 用于刻画数据流的整体分布, 二者合称混合标签查询策略.

3.2　基于记忆强度的样本替换策略

如前所述, 记忆窗口用于保存查询到标签的样本, 包括难区分、少数类样本和当前数据分布的代表性样本. 随着时间的推移, 记忆窗口中的样本需要被新查询到标签的样本替换, 现有的替换方法通常用到达时间作为样本的重要性度量, 即认为后到达的样本比先到达样本更能代表当前的数据分布, 因而采用基于先进先出的替换策略. 为替换记忆窗口中的样本, 本文从以下三个方面来衡量样本的重要性:

1) 样本被“回忆”的次数 (如果数据流的当前样本落在记忆窗口中样本 x 的邻域范围内, 则称样本 x 被“回忆”一次). 样本被“回忆”的次数越多, 表明在数据流中该样本出现的次数越多, 其重要性应该越高.

2) 样本最后一次被“回忆”的时刻. 样本最后一次被“回忆”的时刻与当前时刻越接近, 表明该样本越能代表当前数据分布, 其重要性越高.

3) 集成分类器对样本的预测误差. 预测误差越大, 表明该样本与历史数据分布越不相符, 应具有更高的重要性.

基于此, 本文将记忆窗口 MW 中样本 x 的重要性, 即记忆强度 $S(x)$ 定义为

$$S(x) = \mathrm{e}^{-\frac{t_i - \tau_x}{(\lambda_x + 1) \times f(x)}} \quad (5)$$

其中, t_i 为当前样本 x_i 到达的时刻, τ_x 为样本 x 最后一次被“回忆”的时刻, λ_x 为样本 x 被“回忆”的次数, $f(x)$ 为样本 x 的预测误差. 其中 λ_x 和 $f(x)$ 的计算分别如式 (6) 和式 (7) 所示.

$$\lambda_x = \lambda_x + I\left(Dis(x_i, x) < MinDis(x)\right) \quad (6)$$

其中, $Dis(x_i, x)$ 为样本 x_i 与 x 的距离, $MinDis(x)$ 为样本 x 与记忆窗口中同类样本的最小距离. $I(\cdot)$ 为指示函数, 当“$\cdot$”为真时取值为 1, 为假时取值为 0.

$$f(x) = 1 - P(y|x) \quad (7)$$

其中, $P(y|x)$ 为集成分类器对样本 x 真实类别 y 的预测概率.

为解决数据流的多类不平衡问题, 本文为记忆窗口中的所有类分配等量的存储空间, 并且样本替换仅在同类样本中进行. 在记忆窗口维护过程中, 少数类和难区分的样本会频繁发生替换, 而多数类

和易区分的样本较少发生替换, 从而保证记忆窗口中的样本能够代表数据流的当前分布. 一旦发生概念漂移, 基于这些样本构建的基分类器可以更好地代表新的数据分布. 此外, 本文还将记忆强度作为样本的权重来训练新基分类器.

3.3 集成分类器的更新机制

在数据流分类过程中, 为适应数据流中的概念漂移, 需要不断更新集成分类器. 在对集成分类器更新时, 通常采用新基分类器替代重要性较低的旧基分类器的方式, 其关键在于基分类器的重要性评价. 现有的方法通常基于分类精度或创建时间对基分类器的重要性进行评价. 本文提出一种新的基分类器重要性评价及更新方法: 新基分类器 C_{new} 代表当前的数据分布, 具有最高的重要性; 旧基分类器对记忆窗口中样本的分类精度越高, 其重要性越高; 旧基分类器 C_d 的重要性随着新基分类器的加入不断降低.

据此, 基分类器重要性定义及更新方式为

$$W(C_{\text{new}}) = \frac{1}{D} \tag{8}$$

$$W(C_d) = W(C_d) \times \left(1 - \frac{1}{D}\right) \times \frac{\sum\limits_{\substack{1 \le i \le k \\ 1 \le j \le S_m}} I(h_d(MW[i, j]) = y)}{k \times S_m} \tag{9}$$

其中, $1 \le d \le D$, D 为集成分类器中基分类器的个数, $h_d(MW[i, j])$ 为基分类器 C_d 对样本 $MW[i, j]$ 的预测类别, y 为该样本的真实类别.

基于上述基分类器重要性评价及更新方式, 本文提出一种集成分类器更新机制: 当发生概念漂移时, 首先基于记忆窗口 MW 中的样本构建新基分类器 C_{new}, 并初始化其重要性为 $1/D$; 然后根据式 (9) 更新所有旧基分类器的重要性; 接下来用 C_{new} 替换重要性最小的旧基分类器, 并归一化所有基分类器的重要性.

在实际应用中数据流中会出现不同类型的概念漂移, 当发生突变型和重复型概念漂移时, 集成分类器中的基分类器往往会连续更替; 当发生增量型和逐渐型概念漂移时, 基分类器的更替速度相对较为缓慢. 可见, 本文提出的集成分类器更新机制适用于不同类型的概念漂移.

3.4 算法框架

本文提出一种非平衡概念漂移数据流的主动学习方法 ALM-ICDDS, 该方法包括初始化阶段和在线学习阶段, 如图 1 所示. 在初始化阶段, 使用最初到达的 S_l 个样本训练 D 个基分类器构成初始集成分类器 E, 同时将这 S_l 个样本的类别标签存入标签滑动窗口 LW 中. 在线学习阶段中包括标签查询、记忆窗口维护、概念漂移检测、集成分类器更新四个任务.

1) 标签查询. 首先集成分类器 E 对新到来样本 x_i 进行预测; 然后判断是否满足混合标签查询策略, 若满足则由专家标注该样本, 否则将预测结果作为样本标签, 集成分类器 E 继续预测下一个样本; 接下来更新标签滑动窗口 LW, 并计算当前数据流的类不平衡比率; 最后根据样本预测结果和类不平衡比率自适应调整边缘阈值矩阵 M.

2) 记忆窗口维护. 对于新查询到真实标签的样本 x_i, 首先要判断该样本是否为异常点, 若是异常点则继续预测下一个样本, 否则需要初始化样本 x_i 的相关"记忆"信息; 然后更新记忆窗口 MW 中与 x_i 同类样本 x 的被"回忆"次数 λ_x、最后一次被"回忆"时刻 τ_x, 并计算记忆强度 $S(x)$; 接下来用 x_i 替换记忆强度最低的样本 (若记忆窗口中 x_i 类样本数量未达到 S_m, 则直接将样本 x_i 的"记忆"信息存入记忆窗口 MW 中).

3) 概念漂移检测. 当基于混合标签查询策略为样本查询到真实标签后, 使用概念漂移检测算法 ADWIN 检测数据流中是否发生概念漂移, 若发生概念漂移则更新集成分类器; 否则使用样本 x_i 更新基分类器对应的 Hoeffding 树, 然后继续预测下一个样本. 更新 Hoeffding 树采用自顶向下预剪枝的方式, 首先更新样本 x_i 对应叶结点的统计信息并计算分类精度, 然后基于信息增益选择最优分裂属性, 如果叶结点分裂后的分类精度高于分裂前, 则对其进行分裂, 否则不分裂. ADWIN 算法基于两个相邻子窗口中预测正确率的差异程度来检测概念漂移. 算法的基本流程为: 首先用窗口 W 保存待检测数据流, 若数据流中的样本预测正确保存 1, 错误保存 0 (本文只保存查询到标签的样本的预测情况); 然后基于指数矩形图计算窗口 W 的切割点, 将窗口 W 切割成两个子窗口 W_0 和 W_1, 并比较 W_0 和 W_1 预测正确率的差异. 如果差异大于阈值 ϵ_{cut}, 则认为至少以 $1-\delta$ 的概率发生概念漂移, 抛弃较旧的子窗口 W_0, 否则认为至少没有以 $1-\delta$ 的概率发生概念漂移. 阈值 ϵ_{cut} 的定义方式为

$$\epsilon_{\text{cut}} = \sqrt{\frac{1}{2}\left(\frac{1}{n_0} + \frac{1}{n_1}\right) \times \ln \frac{4(n_0 + n_1)}{\delta}} \tag{10}$$

其中, n_0 为子窗口 W_0 的大小, n_1 为子窗口 W_1 的大小. 置信度 $\delta \in (0, 1)$, 在本文实验中, δ 设置为 0.002,

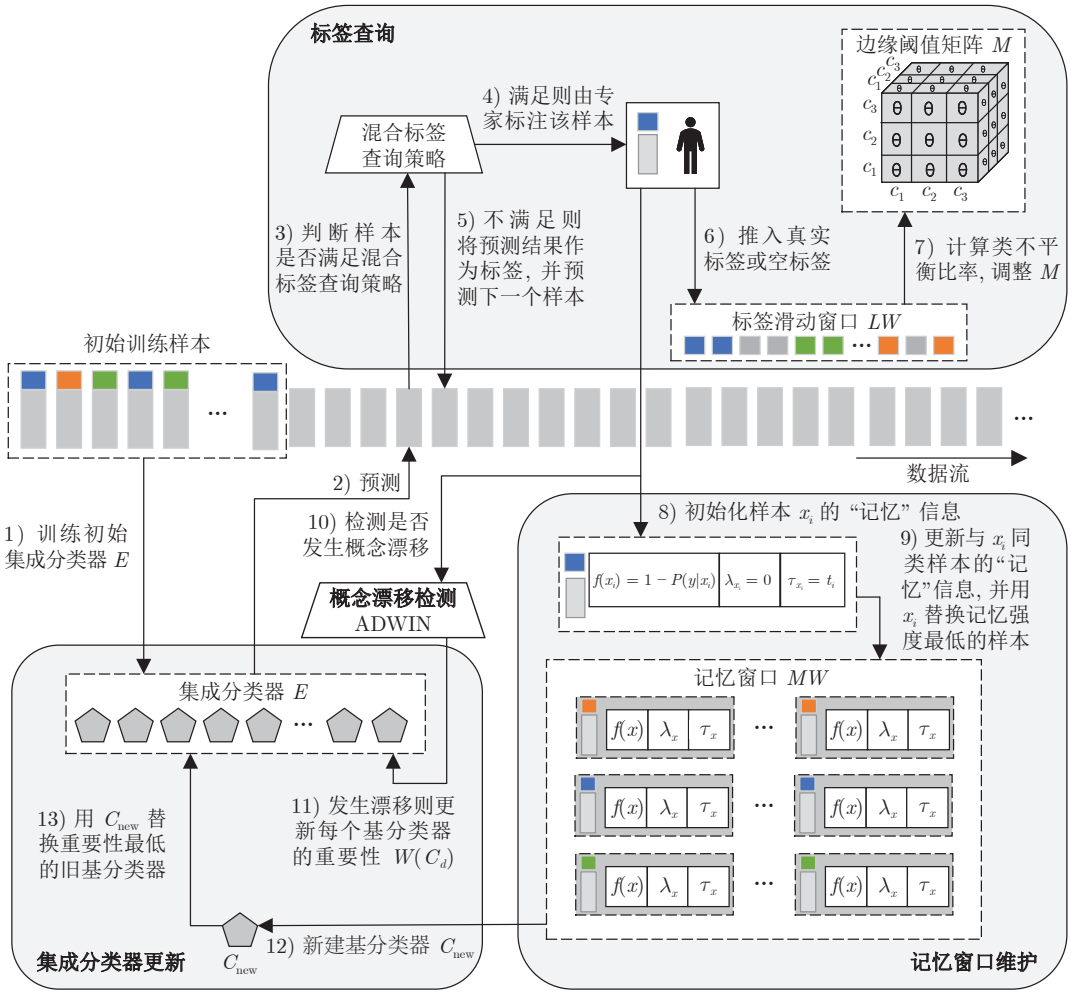

图 1　算法框架

Fig. 1　Algorithm framework

窗口 W、子窗口 W_0 和 W_1 的大小在检测过程中是动态变化的.

4) 集成分类器更新. 当检测到概念漂移后, 首先基于记忆窗口 MW 中的样本训练新基分类器 C_{new}, 然后根据式 (9) 更新旧基分类器的重要性, 最后使用新基分类器 C_{new} 替换重要性最小的旧基分类器.

3.5　算法伪代码

本节给出非平衡概念漂移数据流主动学习算法 ALM-ICDDS、基于样本预测确定性的标签查询 (Label query based on sample prediction certainty, LQ-SPC) 算法和基于记忆强度的样本替换 (Sample replace based on memory strength, SRMS) 算法的伪代码.

算法 1. ALM-ICDDS

输入. 数据流 DS, 已处理的样本数 p, 已查询标签的样本数 m, 标签预算 B, 调节参数 β, 随机选择比率 ε, 标签滑动窗口 $LW[S_l]$, 边缘阈值矩阵 M, 记忆窗口 $MW[k][S_m]$.

输出. 集成分类器 E 对 x_i 的预测结果.

1)　使用最初 S_l 个样本构建初始集成分类器 E;
2)　初始化 $p=0$, $m=0$;
3)　**while** (到达一个新样本 x_i) **do**
4)　$p++$;
5)　输出 E 对 x_i 的预测结果 $P(y_{c_j}|x_i)$, $1 \le j \le k$;
6)　$labelling =$ False;
7)　**if** ($m/p < B$) **then**
8)　　生成一个随机变量 ζ, $0 \le \zeta \le 1$;
9)　　**if** ($\zeta < \varepsilon$) **then**
10)　　$labelling =$ True;
11)　　为 x_i 查询真实标签 y, 并将 y 存入 LW 中;
12)　**else if** (LQ-SPC (x_i)) **then**
13)　　$labelling =$ True;
14)　　为 x_i 查询真实标签 y, 并将 $Null$ 存入 LW 中;

15) **else**
16) 将 $Null$ 存入 LW 中;
17) **end if**;
18) **end if**;
19) **if** ($labelling ==$ True and $y \in \{y_1, y_2, \cdots, y_k\}$) **then**
20) $m++$;
21) SR-MS (x_i);
22) $drifting =$ ADWIN (x_i);
23) **if** ($drifting ==$ True) **then**
24) 使用 MW 中样本训练 C_{new};
25) 根据式 (9) 更新每个基分类器 C_d 的重要性;
26) 用 C_{new} 替换重要性最小的基分类器;
27) **else**
28) 用 x_i 对所有的基分类器 C_d 进行更新;
29) **end if**;
30) **end if**;
31) 继续预测下一个样本;
32) **end while**.

算法 2. LQ-SPC

输入. 新到达样本 x_i.

输出. $bool \in$ {True, False}.

1) 根据式 (1) 计算样本 x_i 的确定性度量 $D(x_i)$;
2) **if** ($D(x_i) \leq M[c_1, c_2, \cdots, c_{n_d}]$) **then**
3) **return** True;
4) **if** ($y == y_{c_1}$) **then**
5) 根据式 (4) 计算类不平衡比率 $imb_{t_i}^{y}$;
6) 根据式 (3) 调整矩阵中阈值;
7) **end if**;
8) **else**
9) **return** False;
10) **end if**.

算法 3. SR-MS

输入. 新到达样本 x_i.

输出. 更新后的记忆窗口 MW.

1) 初始化 $\lambda_{x_i} = 0$, $\tau_{x_i} = t_i$, 根据式 (7) 计算 $f(x_i)$;
2) **if** (MW 中与 x_i 同类样本个数 $< S_m$) **then**
3) 将样本 x_i 及其"记忆"信息存入 MW 中;
4) **if** (MW 中与 x_i 同类样本个数 $== 0$) **then**
5) 初始化 $MinDis(x_i) = \infty$;
6) **else**
7) 计算 x_i 与同类样本的最小距离 $MinDis(x_i)$;
8) **end if**;
9) **else**
10) **for** (MW 中与 x_i 同类的每一个样本 x) **do**
11) **if** ($Dis(x_i, x) < MinDis(x)$) **then**
12) $\tau_x = t_i$;
13) λ_x++;
14) $MinDis(x) = Dis(x_i, x)$;
15) **end if**;
16) 根据式 (5) 计算 $S(x)$;
17) **end for**;
18) 计算 x_i 与同类样本的最小距离 $MinDis(x_i)$;
19) 用 x_i 替换 MW 中同类记忆强度最小的样本;
20) **end if**.

3.6 复杂度分析

下面对本文提出的算法 ALM-ICDDS 进行复杂度的分析, 由于初始化集成分类器 E 的复杂度主要取决于随机森林算法, 因此只分析在线学习阶段的复杂度.

假设数据流 DS 的样本个数为 N、标记成本为 B, 则集成分类器 E 对数据流中样本预测的时间复杂度为 $\mathrm{O}(D \cdot N \log N)$, 标签查询算法的时间复杂度为 $\mathrm{O}(N)$, 样本替换算法的时间复杂度为 $\mathrm{O}(N \cdot B \cdot S_m)$, 因此在线学习的时间复杂度为 $\mathrm{O}(D \cdot N \log N + N + N \cdot B \cdot S_m)$.

在线学习阶段需要使用边缘阈值矩阵 M、标签滑动窗口 $LW[S_l]$ 和记忆窗口 $MW[k][S_m]$, 因此算法的空间复杂度为 $\mathrm{O}(k^{n_d} + S_l + k \cdot S_m)$.

4 实验结果和分析

4.1 实验环境和数据

本文实验环境为 Window10, 处理器为 Intel Core i7-7 700 3.60 GHz, 内存 16 GB, 开发软件使用 IntelliJ IDEA (Community Edition), 所有实验均基于大规模在线分析平台 MOA (Massive online analysis)[40] 实现.

实验使用 11 个合成数据流和 3 个真实数据流, 数据流的特征如表 1 和表 2 所示 (表 2 中的 4 个数据流用于分析本文所提算法对不同类型概念漂移数据流的适用性). 合成数据流 DS_1, DS_2, $\mathrm{DS}_7 \sim \mathrm{DS}_{11}$ 为平衡数据流; DS_3 和 DS_4 为类不平衡比率固定的非平衡数据流; DS_5 和 DS_6 为类不平衡比率可变的非平衡数据流, 且在第 200 000 个样本处类不平衡比率发生变化. 数据流 DS_2, DS_4, DS_6, DS_7 中添加 5% 的异常点和 3 次概念漂移, 其中第 1 次和第 3 次为增量型概念漂移, 第 2 次为突变型概念漂移, 数据流 $\mathrm{DS}_8 \sim \mathrm{DS}_{11}$ 中添加 1 次概念漂移, 分别为突变型、重复型、增量型和逐渐型概念漂移. 实验采用 MOA 平台的数据流生成器 RandomRBF 和概念漂移合成工具 ConceptDriftStream 构造突变型和增量型概念漂移数据流, 具体步骤为: 首先使用

表 1　数据流特征
Table 1　Data stream feature

编号	数据流	样本数	特征数	类别数	类分布	异常点 (%)	漂移次数
1	DS_1	400000	25	15	类平衡	0	0
2	DS_2	400000	25	15	类平衡	5	3
3	DS_3	400000	25	15	(1/1/1/1/1/1/1/1/1/1/2/2/3/3/5)	0	0
4	DS_4	400000	25	15	(1/1/1/1/1/1/1/1/1/1/2/2/3/3/5)	5	3
5	DS_5	400000	25	15	(1/1/1/1/1/1/1/1/1/1/2/2/3/3/5), (2/2/3/3/5/1/1/1/1/1/1/1/1/1/1)	0	0
6	DS_6	400000	25	15	(1/1/1/1/1/1/1/1/1/1/2/2/3/3/5), (2/2/3/3/5/1/1/1/1/1/1/1/1/1/1)	5	3
7	DS_7	400000	25	50	类平衡	5	3
8	Kddcup99_10%	494000	42	23	—	—	—
9	Shuttle	570000	10	7	—	—	—
10	PokerHand	830000	10	10	—	—	—

表 2　概念漂移数据流特征
Table 2　Concept drift data stream feature

编号	数据流	概念漂移类型	样本数	特征数	类别数	漂移宽度
1	DS_8	突变型	400000	25	15	1
2	DS_9	重复型	400000	25	15	1
3	DS_{10}	增量型	400000	25	15	10000
4	DS_{11}	逐渐型	400000	25	15	10000

RandomRBF 对基准数据聚类, 并从聚类结果中选择部分簇作为具有特定分布的数据流 (当聚类个数不同时, 选出数据流的分布也不相同, 可作为概念漂移前后的数据流); 然后使用 ConceptDriftStream 将不同分布的数据流进行链接, 基于给定的概念漂移宽度, 以前一部分数据流的最后一个样本为中心向两边扩展, 并对这部分数据使用给定的扰动函数模拟概念漂移. 一次重复型概念漂移相当于两次突变型概念漂移, 逐渐型概念漂移在一段时间内用新数据分布逐渐取代旧数据分布[10], 本实验在 RandomRBF 生成不同分布数据流的基础上编写函数构造这两种类型的概念漂移. 漂移前后数据流聚类个数取值分别为 50 和 100, 突变型和重复型概念漂移宽度设置为 1, 增量型和逐渐型概念漂移宽度设置为 10000. 3 个真实数据流来源于公开数据集 UCIs (University of California at Irvine), 均为多类不平衡数据流, 且概念漂移的类型和数量都是未知的.

4.2　ALM-ICDDS 算法的分类性能评价

下面用本文提出的 ALM-ICDDS 算法在 7 个合成数据流和 3 个真实数据流上与 6 种数据流分类算法进行分类性能对比, 6 种算法包括 3 种监督学习算法 (LB、BOLE 和 ARFRE) 和 3 种主动学习算法 (CALMID、OALM-IDS 和 ALM-ICDDS-E), 其中主动学习算法 ALM-ICDDS-E 是将 ALM-ICDDS 算法中的样本预测确定性度量替换为基于熵的度量后得到的算法. 同时本文使用精确率 (P)、召回率 (R)、F1 值、Kappa 系数和 ROC (Receiver operating characteristic) 曲线作为评价指标. 所有算法在同一数据流上的实验结果均取 10 次实验的平均值.

为保证实验的公平性, 所有算法均使用集成分类器, 且均包含 11 个基分类器. 除 ARFRE 算法使用其构造的 ARFHoeffding 树作为基分类器之外, 其余算法均使用 Hoeffding 树作为基分类器. 4 种主动学习算法的标签预算 B 均设置为 20%. CALMID 和 OALM-IDS 算法中标签和样本滑动窗口的大小都设置为 500, ALM-ICDDS-E 和 ALM-ICDDS 算法中标签滑动窗口 LW 大小 S_l 设置为 500, 记忆窗口 MW 中每类样本存储个数 S_m 设置为 100. 综合考虑算法的性能和空间复杂度, 边缘阈值矩阵维数 n_d 取 2 或 3, 分别对应数据流中样本的类别数小于等于 10 或大于 10 的情况. 随机选择比率 ε 设置为 0.1, 调节参数 β 设置为 0.8, 边缘阈值矩阵元素的初始值 θ_0 设置为 0.75, 边缘阈值矩阵自适应调整参数 α 设置为 0.1. 参数 n 为样本预测确定性度量所使用

的预测概率的数目, 通过测试 n 的取值对数据流分类性能的影响, 给出如下所示的设置方法, 具体细节参见第 4.4 节.

$$n=\begin{cases}2, & 2\le k\le 10\\ 3, & 10<k\le 20\\ \left\lfloor\sqrt{k}\right\rfloor, & k>20\end{cases}\tag{11}$$

两种主动学习算法 CALMID、OALM-IDS 和本文所提算法 ALM-ICDDS 在线学习阶段的空间复杂度分别为 $\mathrm{O}(k^2+w_l+w_s)$、$\mathrm{O}(k^2+w_l+w_s)$ 和 $\mathrm{O}(k^{n_d}+S_l+k\cdot S_m)$, 其中 w_l 和 w_s 分别为标签滑动窗口与样本滑动窗口的大小. 可见当数据流中样本的类别数大于 10 时, 本文算法中边缘阈值矩阵的空间开销比 CALMID 和 OALM-IDS 算法大. 此外, 为处理非平衡数据, 本文算法在记忆窗口中等量存储每类的样本, 因此记忆窗口的空间开销为 $\mathrm{O}(k\cdot S_m)$. 实验结果如表 3 ~ 6 所示.

由表 3 和表 6 可知, ALM-ICDDS 算法在 6 个合成数据流和 3 个真实数据流上的 P 值和 Kappa 系数均为最高, 尤其在类别数较多的合成数据流 DS_7 和真实数据流 Kddcup99_10% 上优势更加明显. 在 DS_7 数据流上 P 值和 Kappa 系数比 OALM-IDS 算法分别高 3.16% 和 3.48%, 在 Kddcup99_10% 数据流上 P 值比 OALM-IDS 算法高 3.67%. 仅在合成数据流 DS_5 上, ALM-ICDDS 算法的 P 值和 Kappa 系数比 ARFRE 略低, 分别低 0.13% 和 0.05%, 这是由于 ARFRE 是监督学习算法, 在算法执行过程中需要使用所有样本的标签信息.

由表 4 和表 5 可知, ALM-ICDDS 算法在所有数据流上的 R 和 F1 值均优于其他的对比算法, 同样在类别数较多的两个数据流 (DS_7、Kddcup99_10%) 上优势更加明显. 在 DS_7 数据流上 R 和 F1 值比 OALM-IDS 算法分别高 2.84% 和 3.02%, 在 Kddcup99_10% 数据流上 R 和 F1 值分别比 OALM-IDS 算法高 5.63% 和 5.12%.

通过对表 3 ~ 6 的实验结果分析可知, 主动学习算法 CALMID、OALM-IDS 和 ALM-ICDDS 在 10 个数据流上的分类性能均优于监督学习算法 LB、

表 3 7 种算法的 P 值 (%)

Table 3 P value of seven algorithms (%)

数据流	LB	BOLE	ARFRE	CALMID	OALM-IDS	ALM-ICDDS-E	ALM-ICDDS
DS_1	96.89±0.31	96.36±0.11	98.07±0.43	98.01±0.41	98.03±0.25	97.18±0.48	**99.07±0.34**
DS_2	90.61±0.21	88.63±0.54	92.77±0.42	93.31±0.14	93.27±0.49	91.97±0.26	**94.64±0.15**
DS_3	94.41±0.11	96.07±0.23	96.74±0.45	96.64±0.34	96.75±0.56	96.46±0.61	**97.84±0.24**
DS_4	86.91±0.45	85.23±0.52	88.30±0.29	89.90±0.28	90.27±0.42	89.70±0.72	**92.06±0.28**
DS_5	93.60±0.48	94.04±0.52	**96.30±0.18**	94.65±0.49	95.47±0.32	94.24±0.35	96.17±0.19
DS_6	86.59±0.19	84.69±0.48	88.02±0.47	88.44±0.19	88.65±0.25	87.41±0.40	**90.86±0.37**
DS_7	88.25±0.86	87.21±0.79	90.16±0.92	90.49±0.47	90.51±0.53	89.32±0.38	**93.67±0.40**
Kddcup99_10%	83.85±0.59	81.10±0.15	85.56±0.54	92.12±0.45	92.13±0.31	91.24±0.51	**95.80±0.17**
Shuttle	64.63±0.42	63.85±0.27	79.07±0.31	85.35±0.14	85.70±0.32	83.48±0.25	**85.99±0.13**
PokerHand	51.63±0.39	50.36±0.35	52.51±0.56	53.93±0.28	54.57±0.50	52.90±0.18	**55.89±0.51**

表 4 7 种算法的 R 值 (%)

Table 4 R value of seven algorithms (%)

数据流	LB	BOLE	ARFRE	CALMID	OALM-IDS	ALM-ICDDS-E	ALM-ICDDS
DS_1	94.78±0.13	96.04±0.24	96.81±0.59	97.87±0.24	97.92±0.25	96.15±0.31	**98.63±0.17**
DS_2	88.65±0.25	87.86±0.53	90.35±0.30	91.54±0.54	91.84±0.58	90.78±0.70	**92.30±0.24**
DS_3	92.55±0.45	95.92±0.32	94.80±0.43	96.12±0.14	97.92±0.54	95.99±0.52	**98.55±0.29**
DS_4	87.03±0.49	87.08±0.39	88.23±0.31	90.50±0.30	91.07±0.52	90.13±0.43	**91.15±0.11**
DS_5	91.54±0.11	92.33±0.51	96.04±0.20	93.82±0.55	94.94±0.27	92.91±0.42	**96.53±0.42**
DS_6	86.56±0.50	85.48±0.24	87.83±0.49	89.43±0.18	88.85±0.36	88.39±0.34	**90.63±0.21**
DS_7	87.19±0.42	86.12±0.11	87.29±0.36	88.41±0.50	88.77±0.43	87.87±0.20	**91.61±0.78**
Kddcup99_10%	60.89±0.50	63.05±0.50	58.26±0.38	61.88±0.38	63.71±0.54	63.42±0.67	**69.34±0.57**
Shuttle	61.40±0.21	50.84±0.31	54.36±0.35	59.52±0.41	63.12±0.59	61.79±0.16	**64.59±0.29**
PokerHand	43.57±0.30	44.78±0.46	55.21±0.60	56.84±0.11	52.77±0.54	55.36±0.25	**59.57±0.43**

表 5 7 种算法的 F1 值 (%)
Table 5 F1 value of seven algorithms (%)

数据流	LB	BOLE	ARFRE	CALMID	OALM-IDS	ALM-ICDDS-E	ALM-ICDDS
DS_1	95.82±0.18	96.20±0.16	97.44±0.50	97.94±0.30	97.97±0.25	96.66±0.37	**98.85±0.23**
DS_2	89.62±0.23	88.24±0.53	91.54±0.35	92.42±0.22	92.55±0.53	91.37±0.43	**93.46±0.18**
DS_3	93.47±0.18	95.99±0.27	95.76±0.44	96.38±0.20	97.33±0.55	96.22±0.57	**98.19±0.26**
DS_4	86.97±0.47	86.15±0.45	88.26±0.30	90.20±0.29	90.67±0.46	89.91±0.59	**91.60±0.16**
DS_5	92.55±0.17	93.18±0.30	96.17±0.19	94.23±0.52	95.20±0.29	93.57±0.38	**96.35±0.26**
DS_6	86.57±0.27	85.08±0.32	87.92±0.48	88.93±0.18	88.75±0.30	87.90±0.35	**90.74±0.27**
DS_7	87.72±0.56	86.66±0.19	88.70±0.52	89.44±0.48	89.61±0.47	88.59±0.29	**92.63±0.40**
Kddcup99_10%	70.55±0.54	70.94±0.23	69.32±0.45	74.03±0.22	75.33±0.39	74.82±0.54	**80.45±0.49**
Shuttle	62.97±0.28	56.61±0.29	64.43±0.33	70.13±0.21	72.70±0.41	71.01±0.20	**73.77±0.18**
PokerHand	47.26±0.34	47.41±0.40	53.83±0.57	55.35±0.16	56.12±0.52	54.10±0.23	**57.67±0.72**

表 6 7 种算法的 Kappa 值 (%)
Table 6 Kappa value of seven algorithms (%)

数据流	LB	BOLE	ARFRE	CALMID	OALM-IDS	ALM-ICDDS-E	ALM-ICDDS
DS_1	95.09±0.43	95.47±0.26	97.11±0.33	97.84±0.18	97.52±0.50	96.31±0.53	**98.72±0.18**
DS_2	89.66±0.50	88.28±0.45	91.80±0.17	92.55±0.25	92.65±0.28	91.27±0.29	**93.56±0.46**
DS_3	93.08±0.13	95.68±0.22	95.62±0.53	96.50±0.46	96.46±0.60	96.05±0.36	**97.69±0.21**
DS_4	86.97±0.46	85.86±0.13	88.18±0.25	89.94±0.24	89.99±0.36	88.61±0.46	**90.19±0.57**
DS_5	92.32±0.37	94.18±0.45	**95.86±0.28**	94.40±0.50	95.52±0.14	94.29±0.20	95.81±0.35
DS_6	86.59±0.32	85.25±0.29	87.81±0.54	88.90±0.51	89.00±0.13	87.68±0.47	**89.80±0.25**
DS_7	88.28±0.46	87.51±0.97	89.93±0.71	90.01±0.92	90.19±0.40	89.51±0.59	**93.67±0.54**
Kddcup99_10%	80.94±0.22	75.68±0.25	79.36±0.35	83.32±0.24	85.83±0.50	84.87±0.16	**86.81±0.33**
Shuttle	58.73±0.39	61.54±0.22	73.78±0.20	79.39±0.43	80.11±0.53	80.97±0.24	**83.56±0.54**
PokerHand	50.34±0.58	49.86±0.40	50.36±0.16	51.24±0.21	51.39±0.16	50.55±0.41	**52.25±0.35**

BOLE 和 ARFRE; 主动学习算法 ALM-ICDDS-E 在 7 个合成数据流上的分类性能优于监督学习算法 LB 和 BOLE, 在 3 个真实数据流上分类性能优于监督学习算法 LB、BOLE 和 ARFRE; ALM-ICDDS 算法在 10 个数据流上的分类性能均优于其他 3 种主动学习算法; 所有算法在不包含概念漂移和异常点的数据流上, 分类性能优于包含概念漂移和异常点的数据流; 对于有无异常点和概念漂移的两类数据流, 所有算法在类平衡、类不平衡比率固定和类不平衡比率变化的数据流上的分类性能依次下降.

图 2 展示所有算法在 10 个数据流上的接受者操作特征曲线 ROC, ROC 曲线下面积能够直观地反映出分类性能的好坏. 由图 2 可知, 主动学习算法 CALMID、OALM-IDS 和 ALM-ICDDS 在 7 个合成数据流和 PokerHand 数据流上的 ROC 曲线下面积优于监督学习算法 LB、BOLE 和 ARFRE; 主动学习算法 ALM-ICDDS-E 在 7 个合成数据流上的 ROC 曲线下面积优于监督学习算法 LB 和 BOLE, 在 3 个真实数据流上 ROC 曲线下面积优于监督学习算法 LB、BOLE 和 ARFRE; ALM-ICDDS 算法除在 PokerHand 数据流上优于 ALM-ICDDS-E 算法但与 CALMID 和 OALM-IDS 算法的 ROC 曲线下面积相同外, 在其他数据流上均优于 6 种对比算法; 在类别数较多的合成数据流 DS_7 和真实数据流 Kddcup99_10% 上 ALM-ICDDS 算法优势更明显, 在这两个数据流上比 ROC 曲线下面积第二大的算法分别高 3% 和 4%.

图 3 展示所有算法在 2 个较为复杂的数据流 (DS_6 和 Kddcup99_10%) 上分类精确率随样本规模增加的变化曲线. 可知 ALM-ICDDS 算法在这 2 个数据流上的分类精确率均优于 3 种主动学习算法 (CALMID、OALM-IDS 和 ALM-ICDDS-E), 且明显优于 3 种监督学习算法 (LB、BOLE 和 ARFRE). ALM-ICDDS 算法在这 2 个数据流上均可以用最少的标签成本获得最高的精确率, 在 DS_6 数据流上的标签成本比 OALM-IDS 算法低 0.17%、比 CALMID 算法低 0.32%、比 ALM-ICDDS-E 算

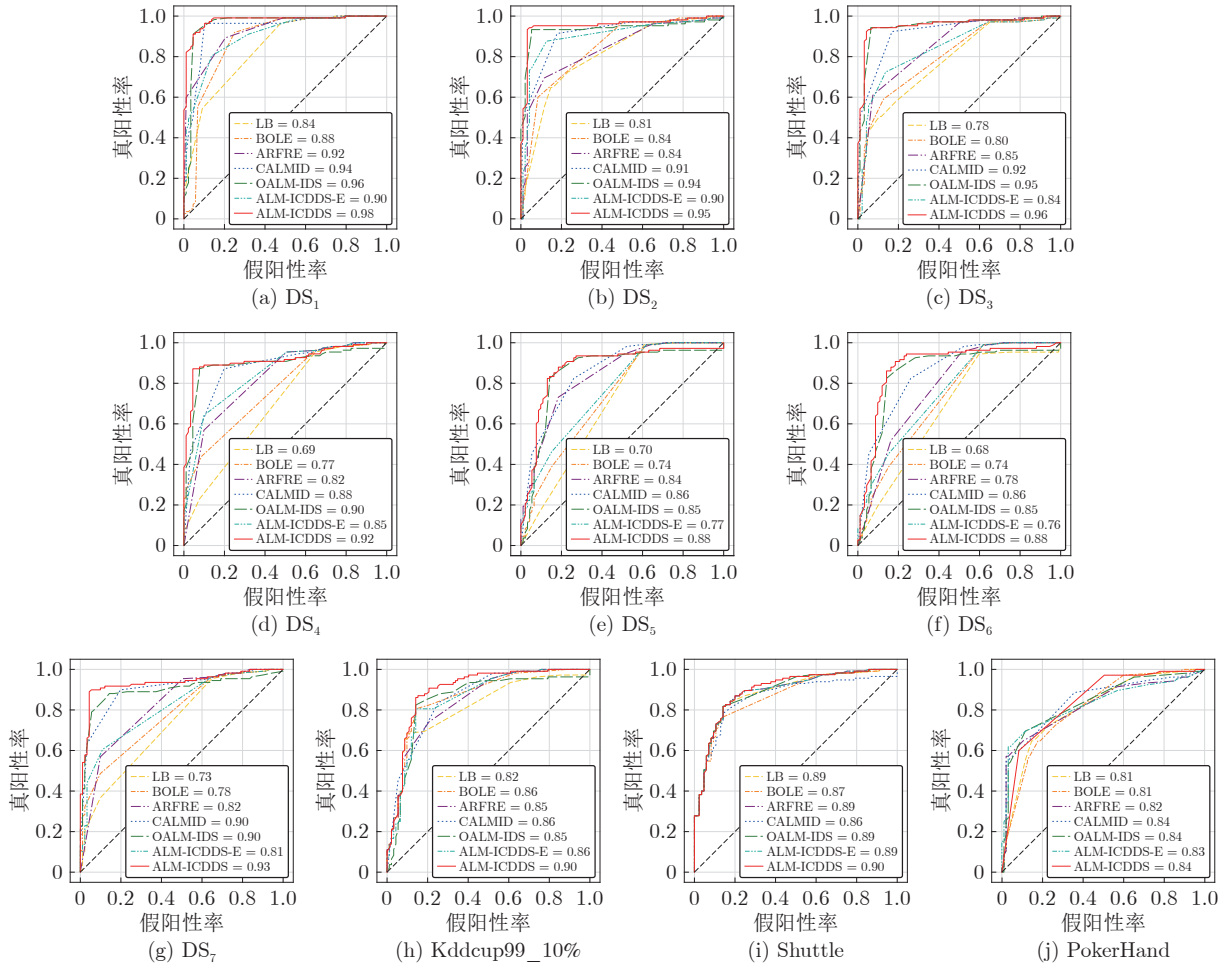

图 2 7 种算法的 ROC 曲线

Fig. 2 ROC curves of seven algorithms

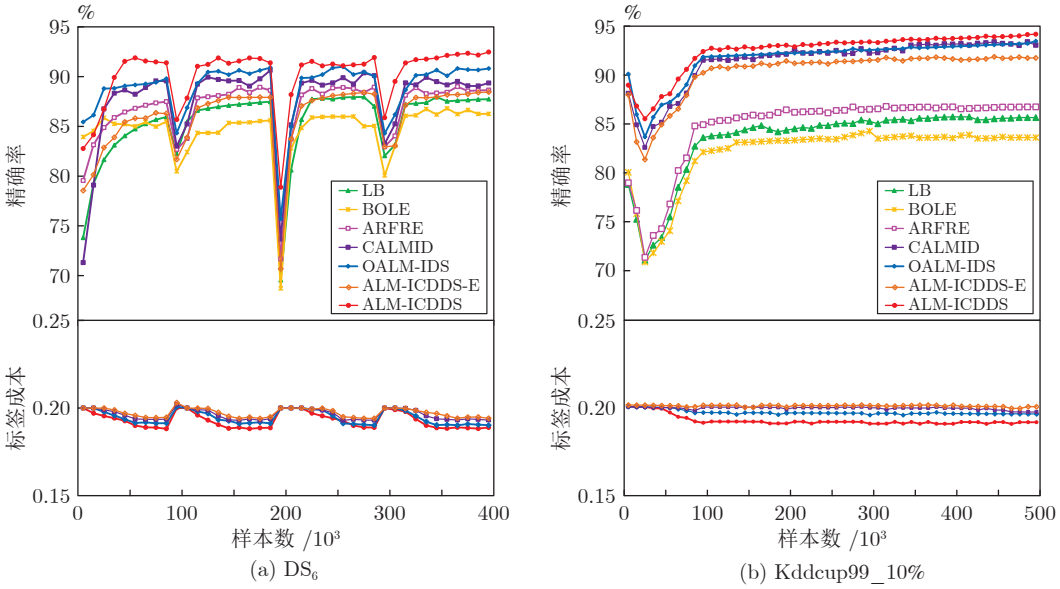

图 3 7 种算法的精确率曲线

Fig. 3 Precision rate curves of seven algorithms

法低 0.43%, 在 Kddcup99_10% 数据流上的标签成本比 OALM-IDS 算法低 0.45%、比 CALMID 算法低 0.69%、比 ALM-ICDDS-E 算法低 0.75%.

基于上述实验结果可以得知, 本文所提算法 ALM-ICDDS 在各项评价指标上整体优于其他对比算法, 这是由于 ALM-ICDDS 算法在 3 个方面进行改进. 首先, 定义基于多预测概率的样本预测确定性度量, 对难区分和少数类样本查询真实标签; 其次, 提出基于记忆强度的样本替换策略, 使得代表当前数据分布的样本能够保留在记忆窗口中, 提升新基分类器的分类性能; 此外, 定义基于分类精度的基分类器重要性评价及更新方法, 用于发生概念漂移后集成分类器的更新.

4.3　消融实验

为测试 ALM-ICDDS 算法引入混合标签查询策略、样本替换策略和集成分类器更新机制的有效性, 在类不平衡比率可变、含有异常点和概念漂移的数据流 DS_6 上进行 5 种消融实验. 1) 将 ALM-ICDDS 的样本预测确定性度量替换为只考虑两个最大概率值的差值, 得到 ALM-ICDDS-rfs; 2) 将 ALM-ICDDS 的样本预测确定性度量替换为只考虑最大概率值, 得到 ALM-ICDDS-rf 算法; 3) 将 ALM-ICDDS 的混合标签查询策略替换为随机标签查询策略, 得到 ALM-ICDDS-r 算法; 4) 将 ALM-ICDDS-r 中基于记忆强度的样本替换策略改为基于先进先出的替换策略, 得到 ALM-ICDDS-rs 算法; 5) 将 ALM-ICDDS-rs 的基分类器更新方法替换为只考虑分类精度, 得到 ALM-ICDDS-rse 算法.

通过对图 4 的实验结果分析可知, 只考虑两个最大概率值差值的 ALM-ICDDS-rfs 算法比 ALM-ICDDS 的精确率有所下降; 只考虑最大概率值的 ALM-ICDDS-rf 算法精确率比 ALM-ICDDS-rfs 更低; 只考虑随机标签查询策略的 ALM-ICDDS-r 算法精确率比 ALM-ICDDS 算法下降明显; 改为先进先出替换策略的 ALM-ICDDS-rs 算法精确率继续下降, 且适应概念漂移速度变慢; 基分类器更新方法只考虑分类精度的 ALM-ICDDS-rse 算法精确率进一步下降, 且适应概念漂移速度变得更慢, 尤其适应突变概念漂移的速度下降明显. 综上, 本文在混合标签查询策略、样本替换策略和集成分类器更新机制方面的改进能够有效提升数据流分类的性能.

4.4　参数取值对 ALM-ICDDS 算法分类性能的影响

样本预测确定性度量中的参数 β 用于调节最大可能类的概率和最大可能类与其余类预测概率平均

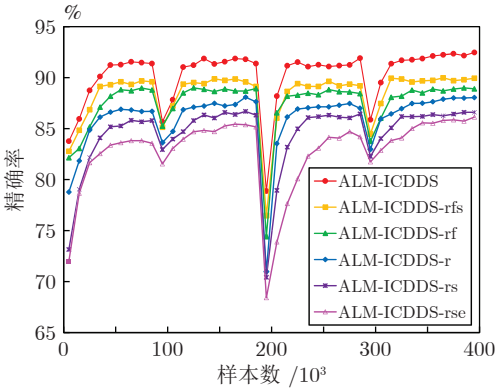

图 4　DS_6 上消融实验的结果

Fig. 4　Results of the ablation experiment on DS_6

差值所起的作用, θ_0 为边缘阈值矩阵元素初值, 参数 n 为样本预测确定性度量所使用的预测概率的数目, 参数 α 用于调节类不平衡比率在边缘阈值矩阵自适应调整中所起的作用, 参数 n_d 为边缘阈值矩阵的维数. 接下来测试参数 β, θ_0, n, α, n_d 对 ALM-ICDDS 算法分类性能的影响. 实验使用含有异常点和概念漂移的 4 个合成数据流和 2 个真实数据流, 采用 P 值、R 值、F1 值、Kappa 系数 4 个评价指标, 所有实验重复 10 次, 结果如图 5 ~ 9 所示.

图 5 展示参数 β 对 ALM-ICDDS 算法分类性能的影响, 可知当 β 取 0.8 时在 6 个数据流上的 P 值、R 值、F1 值、Kappa 系数都达到了最大值, 表明在度量样本预测确定性时, 最大可能类的概率非常重要. 当 β 大于 0.8 时, 算法分类性能下降, 可知最大可能类与其余类预测概率平均差值的作用也至关重要.

图 6 展示边缘阈值矩阵元素初值 θ_0 对 ALM-ICDDS 算法分类性能的影响, 可知当 θ_0 取 0.75 时在 6 个数据流上的 P 值、R 值、F1 值、Kappa 系数都达到最大值. 当 θ_0 小于 0.75 时, 样本被查询标签的可能性减小, 会导致难区分和少数类样本查询不到真实标签, 从而使得 ALM-ICDDS 算法的分类性能下降; 当 θ_0 大于 0.75 时, 样本被查询标签的可能性增大, 但由于标签预算是一定的, 同样会导致难区分和少数类样本查询不到足够的标签, 也会使得 ALM-ICDDS 算法的分类性能下降.

图 7 展示参数 n 对 ALM-ICDDS 算法分类性能的影响, 可知在 6 个数据流上随着 n 的增大, P 值、R 值、F1 值、Kappa 系数增大到一定程度后保持基本稳定. 在 $k = 15$ (数据流 DS_2, DS_4, DS_6)、$k = 50$ (数据流 DS_7)、$k = 23$ (数据流 Kddcup99_10%)、$k = 10$ (数据流 PokerHand) 时, 当 n 分别

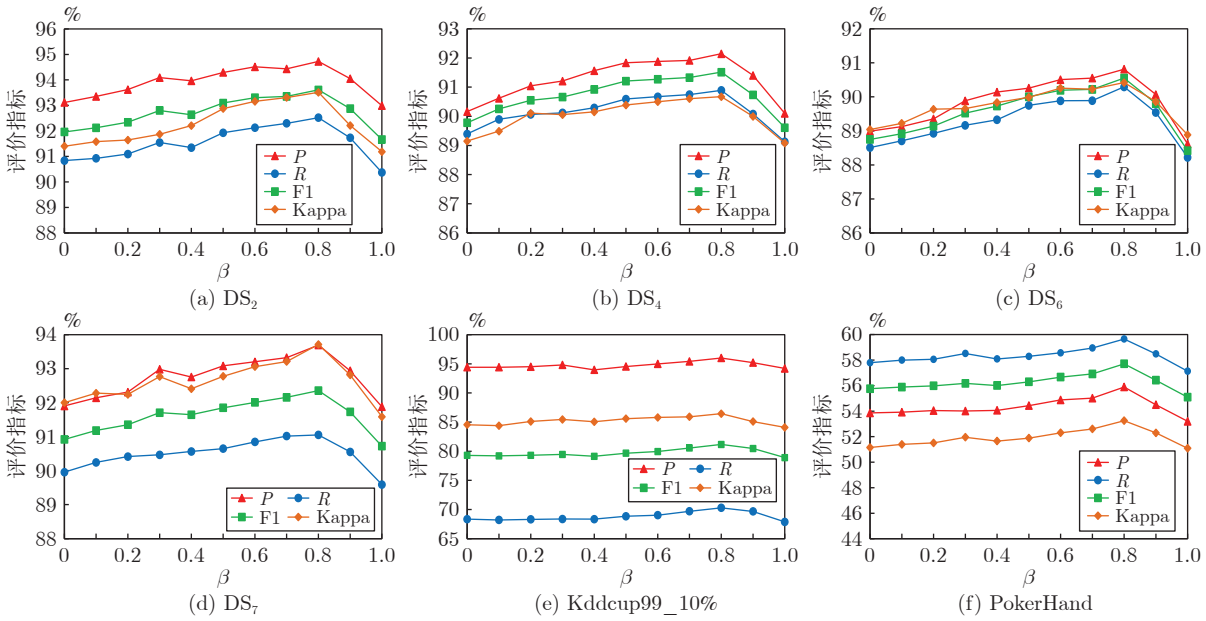

图 5 参数 β 对算法的影响

Fig. 5 Effect of the parameter β on the algorithm

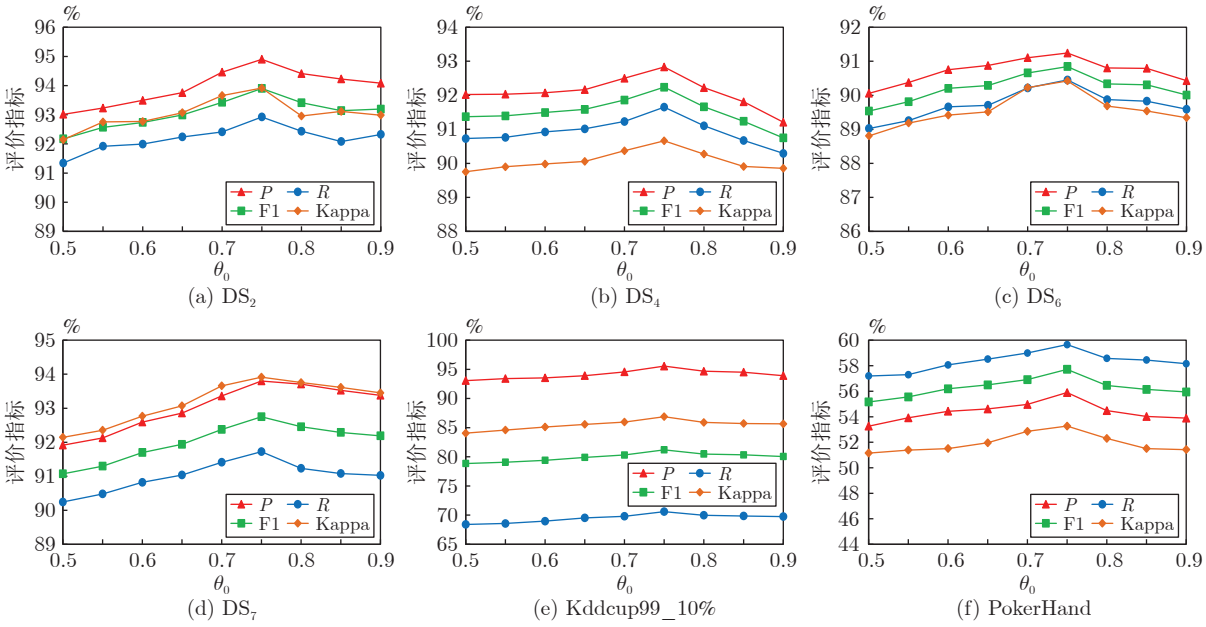

图 6 参数 θ_0 对算法的影响

Fig. 6 Effect of the parameter θ_0 on the algorithm

增加到 3, 7, 4, 2 时, P 值、R 值、F1 值、Kappa 系数基本稳定. 可以验证上述 n 的取值与式 (11) 一致.

图 8 展示参数 α 对 ALM-ICDDS 算法分类性能的影响, 可知当 α 取 0.1 时, 算法在 6 个数据流上的 P 值、R 值、F1 值、Kappa 系数都达到最大值.

图 9 展示参数 n_d 对 ALM-ICDDS 算法分类性能的影响, 可知在类别数 $k > 10$ 的 5 个数据流 (DS$_2$, DS$_4$, DS$_6$, DS$_7$, Kddcup99_10%) 上, 当 n_d 的取值由 2 增加为 3 时, P 值、R 值、F1 值、Kappa 系数明显提升, 当 n_d 继续增大时算法分类性能提升幅度较小. 在类别数 $k = 10$ 的数据流 (PokerHand) 上, 当 n_d 的取值由 2 增加为 9 时, P 值、R 值、F1 值、Kappa 系数提升幅度很小. 考虑到边缘阈值矩阵的空间开销为 O(k^{n_d}), 因此实验中在数据流的类

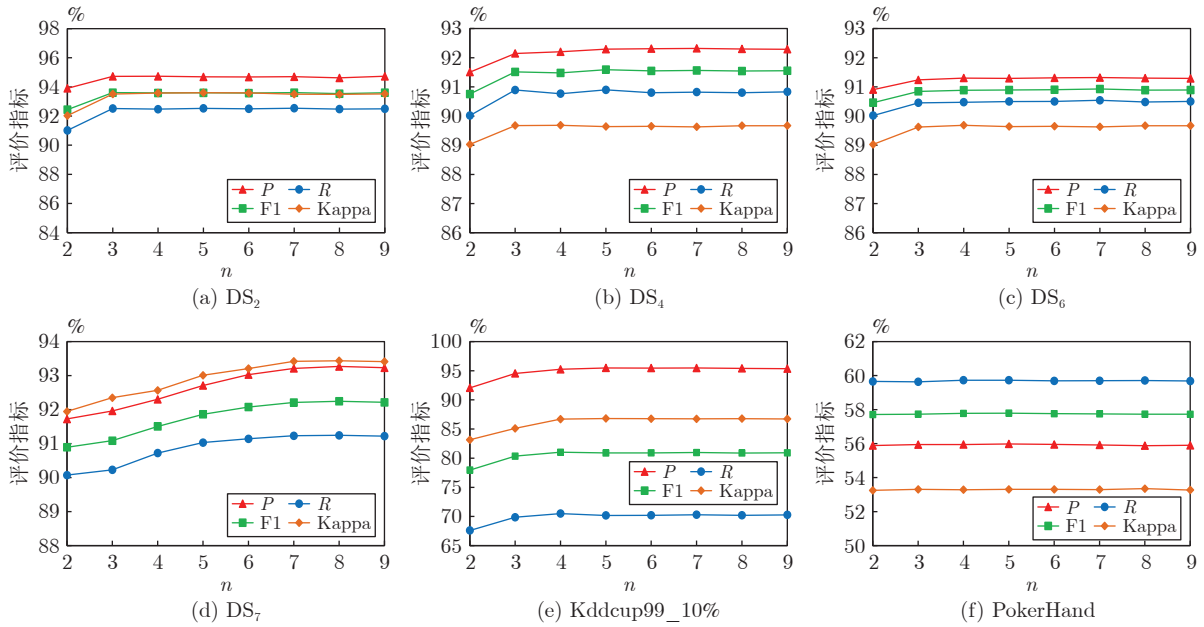

图 7　参数 n 对算法的影响

Fig. 7　Effect of the parameter n on the algorithm

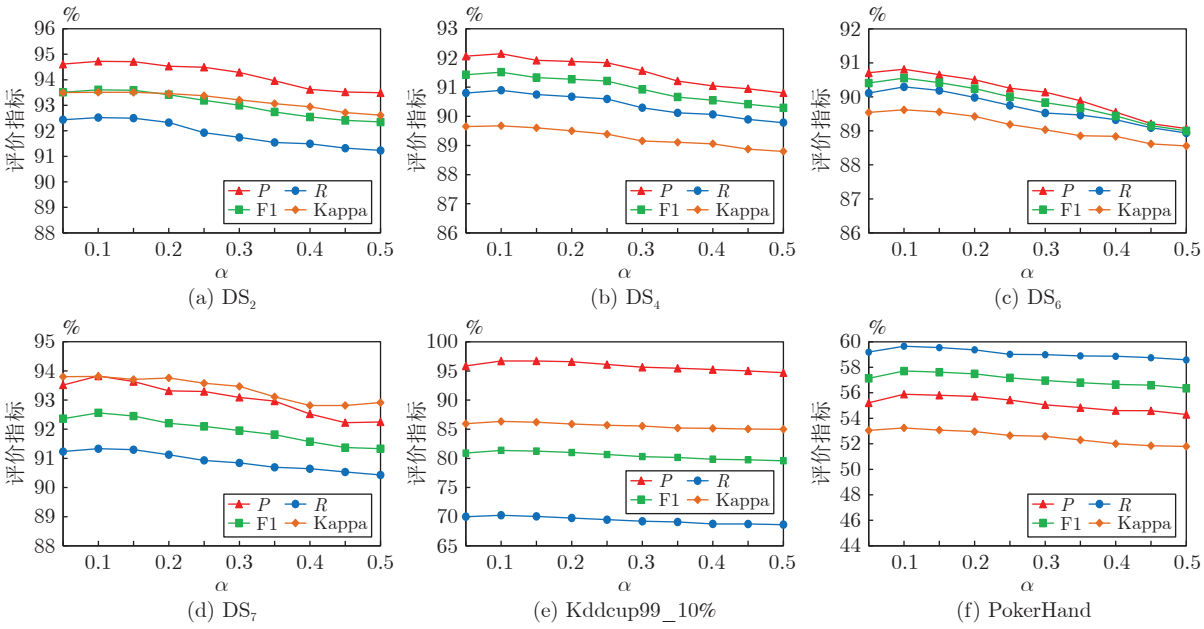

图 8　参数 α 对算法的影响

Fig. 8　Effect of the parameter α on the algorithm

别数 $k \leq 10$ 和 $k > 10$ 两种情况下, n_d 分别取 2 和 3.

4.5　ALM-ICDDS 算法对不同类型概念漂移的处理

图 10 展示本文所提算法和 5 个对比算法在 4 种不同类型概念漂移数据流上分类精确率随样本规模增加的变化曲线. 可知 ALM-ICDDS 算法在 4 种不同类型概念漂移数据流上的分类精确率整体优于 5 个对比算法. 由于突变型和重复型的概念漂移宽度远小于增量型和逐渐型的概念漂移宽度, 由图 10 可知, 相比于增量型和逐渐型漂移, 发生突变型和重复型概念漂移时, 分类精确率下降更快, 通过及时调整模型可以更快地适应概念漂移. 此外, 相比于增量型概念漂移, 发生逐渐型概念漂移时, 算

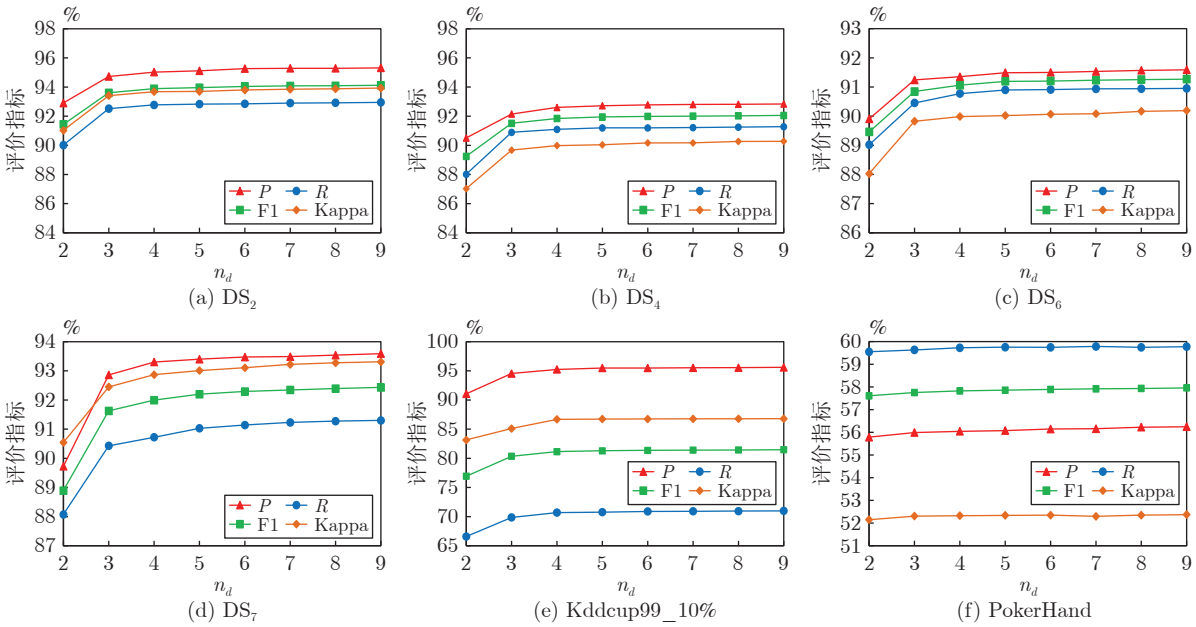

图 9 参数 n_d 对算法的影响

Fig. 9 Effect of the parameter n_d on the algorithm

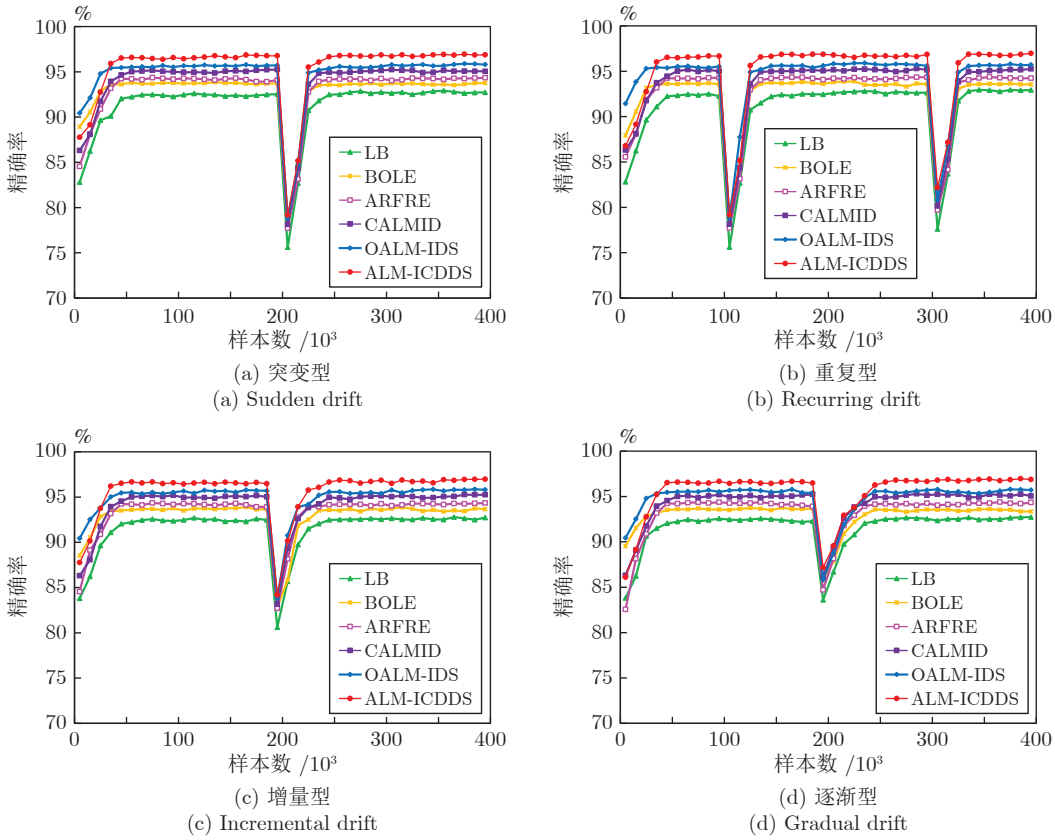

图 10 不同类型概念漂移数据流上的精确率曲线

Fig. 10 P curves on different types of concept drift data stream

法适应概念漂移的速度要慢, 这是由于发生逐渐型概念漂移时, 新数据分布在一段时间内逐渐取代旧数据分布, 而旧数据分布的出现不利于概念漂移的检测.

5 结束语

本文研究概念漂移、标签成本昂贵和多类不平衡数据流的主动学习方法, 定义基于多预测概率的样本预测确定性度量, 使得标签查询策略适用于类别数较多的不平衡数据流; 提出基于记忆强度的样本替换策略, 将难区分、少数类和代表当前数据分布的样本保存在记忆窗口中; 定义基于分类精度的基分类器重要性评价及更新方法, 用于集成分类器更新.

为增强概念漂移数据流主动学习方法的适应性, 在未来工作中, 我们将关注以下问题. 首先, 已有的概念漂移检测方法通常假定数据流中样本的标签都是已知的, 而这在真实应用中是不现实的, 需要研究无监督或半监督场景下的概念漂移检测方法. 其次, 现有的大多数数据流学习模型通常假设样本类别是固定的, 只能泛化到训练集中出现的类别, 因此需要研究适用于类别增量出现的数据流学习模型.

References

1 Liao G, Zhang P, Yin H, Luo T, Lin J. A novel semi-supervised classification approach for evolving data streams. *Expert Systems With Applications*, 2023, **215**: Article No. 119273

2 Zhu Fei, Zhang Xu-Yao, Liu Cheng-Lin. Class incremental learning: A review and performance evaluation. *Acta Automatica Sinica*, 2023, **49**(3): 1−26
(朱飞, 张煦尧, 刘成林. 类别增量学习研究进展和性能评价. 自动化学报, 2023, **49**(3): 1−26)

3 Zhou Z H. Open-environment machine learning. *National Science Review*, 2022, **9**(8): 211−221

4 Wang P, Jin N, Woo W L, Woodward J R, Davies D. Noise tolerant drift detection method for data stream mining. *Information Sciences*, 2022, **609**: 1318−1333

5 Yu H, Liu W, Lu J, Wen Y, Luo X, Zhang G. Detecting group concept drift from multiple data streams. *Pattern Recognition*, 2023, **134**: Article No. 109113

6 Suárez-Cetrulo A L, Quintana D, Cervantes A. A survey on machine learning for recurring concept drifting data streams. *Expert Systems With Applications*, 2022, **213**: Article No. 118934

7 Yang L, Shami A. A lightweight concept drift detection and adaptation framework for IoT data streams. *IEEE Internet of Things Magazine*, 2021, **4**(2): 96−101

8 Bayram F, Ahmed B S, Kassler A. From concept drift to model degradation: An overview on performance-aware drift detectors. *Knowledge-Based Systems*, 2022, **245**: Article No. 108632

9 Karimian M, Beigy H. Concept drift handling: A domain adaptation perspective. *Expert Systems With Applications*, 2023, **224**: Article No. 119946

10 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*, 2018, **31**(12): 2346−2363

11 Shahraki A, Abbasi M, Taherkordi A, Jurcut A D. Active learning for network traffic classification: A technical study. *IEEE Transactions on Cognitive Communications and Networking*, 2021, **8**(1): 422−439

12 Pham T, Kottke D, Sick B, Krempl G. Stream-based active learning for sliding windows under the influence of verification latency. *Machine Learning*, 2022, **111**(6): 2011−2036

13 Khowaja S A, Khuwaja P. Q-learning and LSTM based deep active learning strategy for malware defense in industrial IoT applications. *Multimedia Tools and Applications*, 2021, **80**(10): 14637−14663

14 Wang S, Luo H, Huang S, Li Q, Liu L, Su G, et al. Counterfactual-based minority oversampling for imbalanced classification. *Engineering Applications of Artificial Intelligence*, 2023, **122**: Article No. 106024

15 Malialis K, Panayiotou C G, Polycarpou M M. Nonstationary data stream classification with online active learning and siamese neural networks. *Neurocomputing*, 2022, **512**: 235−252

16 Du H, Zhang Y, Gang K, Zhang L, Chen Y. Online ensemble learning algorithm for imbalanced data stream. *Applied Soft Computing*, 2021, **107**(1): Article No. 107378

17 Wang W, Sun D. The improved AdaBoost algorithms for imbalanced data classification. *Information Sciences*, 2021, **563**: 358−374

18 Gao J, Fan W, Han J, Yu P. A general framework for mining concept-drifting data streams with skewed distributions. In: Proceedings of the International Conference on Data Mining. Minnesota, USA: 2007. 3−14

19 Lu Y, Cheung Y, Tang Y Y. Dynamic weighted majority for incremental learning of imbalanced data streams with concept drift. In: Proceedings of the International Joint Conference on Artificial Intelligence. Melbourne, Australia: AAAI, 2017. 2393−2399

20 Jiao B, Guo Y, Gong D, Chen Q. Dynamic ensemble selection for imbalanced data streams with concept drift. *IEEE Transactions on Neural Networks and Learning Systems*, 2024, **35**(1): 1278−1291

21 Guo H S, Zhang S, Wang W J. Selective ensemble-based online adaptive deep neural networks for streaming data with concept drift. *Neural Networks*, 2021, **142**: 437−456

22 Wang S, Minku L L, Yao X. Resampling-based ensemble methods for online class imbalance learning. *IEEE Transactions on Knowledge and Data Engineering*, 2014, **27**(5): 1356−1368

23 Cano A, Krawczyk B. ROSE: Robust online self-adjusting ensemble for continual learning on imbalanced drifting data streams. *Machine Learning*, 2022, **111**(7): 2561−2599

24 Bifet A, Gavalda R. Learning from time-changing data with adaptive windowing. In: Proceedings of the International Conference on Data Mining. Minnesota, USA: 2007. 443−448

25 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: 2016. 1871−1878

26 Gama J, Medas P, Castillo G, Rodrigues P. Learning with drift detection. In: Proceedings of the Advances in Artificial Intelligence. Maranhao, Brazil: Springer, 2004. 286−295

27 Zhang Yong-Qing, Lu Rong-Zhao, Qiao Shao-Jie, Han Nan, Gutierrez Louis Alberto, Zhou Ji-Liu. A sampling method of imbalanced data based on sample space. *Acta Automatica Sinica*, 2022, **48**(10): 2549−2563
(张永清, 卢荣钊, 乔少杰, 韩楠, Gutierrez Louis Alberto, 周激流. 一种基于样本空间的类别不平衡数据采样方法. 自动化学报, 2022, **48**(10): 2549−2563)

28 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: Springer, 2010. 135−150

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

30 Gu Q, Tian J, Li X, Song J. A novel random forest integrated

model for imbalanced data classification problem. *Knowledge-Based Systems*, 2022, **250**: Article No. 109050

31 Martins V E, Cano A, Junior S B. Meta-learning for dynamic tuning of active learning on stream classification. *Pattern Recognition*, 2023, **138**: Article No. 109359

32 Yin C Y, Chen S S, Yin Z C. Clustering-based active learning classification towards data stream. *ACM Transactions on Intelligent Systems and Technology*, 2023, **14**(2): 1−18

33 Xu W H, Zhao F F, Lu Z C. Active learning over evolving data streams using paired ensemble framework. In: Proceedings of the 8th International Conference on Advanced Computational Intelligence. Chiang Mai, Thailand: 2016. 180−185

34 Liu S X, Xue S, Wu J, Zhou C, Yang J, Li Z, et al. Online active learning for drifting data streams. *IEEE Transactions on Neural Networks and Learning Systems*, 2023, **34**(1): 186−200

35 Liu W K, Zhang H, Ding Z Y, Liu Q B, Zhu C. A comprehensive active learning method for multiclass imbalanced data streams with concept drift. *Knowledge-Based Systems*, 2021, **215**: Article No. 106778

36 Li Yan-Hong, Ren Lin, Wang Su-Ge, Li De-Yu. Online active learning method for imbalanced data stream. *Acta Automatica Sinica*, DOI: 10.16383/j.aas.c211246
(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, DOI: 10.16383/j.aas.c211246)

37 Zhao P, Cai L W, Zhou Z H. Handling concept drift via model reuse. *Machine learning*, 2020, **109**: 533−568

38 Karimi M R, Gürel N M, Karlas B, Rausch J, Zhang C, Krause A. Online active model selection for pre-trained classifiers. In: Proceedings of the International Conference on Artificial Intelligence and Statistics. San Diego, California, USA: 2021. 307−315

39 Zyblewski P, Wozniak M, Sabourin R. Preprocessed dynamic classifier ensemble selection for highly imbalanced drifted data streams. *Information Fusion*, 2021, **66**: 138−154

40 Moraes M, Gradvohl A. MOAFS: A massive online analysis library for feature selection in data streams. *The Journal of Open Source Software*, 2020, **5**: Article No. 1970

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.)

王甜甜 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: wttstu@163.com
(**WANG Tian-Tian** Master student at the School of Computer and Information Technology, Shanxi University. Her 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.)