软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software, 2022,33(4):1315−1325 [doi: 10.13328/j.cnki.jos.006480]

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel: +86-10-62562563

特征演化的置信-加权学习方法*

刘艳芳[1,2], 李文斌[1], 高 阳[1]

[1](计算机软件新技术国家重点实验室(南京大学), 江苏 南京 210023)

[2](龙岩学院 数学与信息工程学院, 福建 龙岩 364012)

通信作者: 高阳, E-mail: gaoy@nju.edu.cn

摘 要: 与研究固定特征空间的传统在线学习相比, 特征演化学习通常假设特征不会以任意方式消失或出现, 而是随着收集数据特征的硬件设备更换旧特征消失、新特征出现. 然而, 已有的特征演化学习方法仅利用数据流的一阶信息, 而忽略可以挖掘特征之间相关性和显著提高分类性能的二阶信息. 提出了一种特征演化的置信-加权学习算法来解决上述问题: 首先, 引入二阶置信-加权来更新数据流的预测模型; 接着, 为了充分利用已学习的模型, 在重叠时期学习线性映射来恢复旧特征; 随后, 用恢复的旧特征更新已有模型; 同时, 用新特征学习新的预测模型; 继而, 运用两种集成方法来利用这两种模型; 实验研究表明, 所提算法优于已有的特征演化学习算法.

关键词: 机器学习; 二阶置信-加权; 在线学习; 演化特征; 分类

中图法分类号: TP18

中文引用格式: 刘艳芳, 李文斌, 高阳. 特征演化的置信-加权学习方法. 软件学报, 2022, 33(4): 1315–1325. http://www.jos.org.cn/1000-9825/6480.htm

英文引用格式: Liu YF, Li WB, Gao Y. Confidence-weighted Learning for Feature Evolution. Ruan Jian Xue Bao/Journal of Software, 2022, 33(4): 1315–1325 (in Chinese). http://www.jos.org.cn/1000-9825/6480.htm

Confidence-weighted Learning for Feature Evolution

LIU Yan-Fang[1,2], LI Wen-Bin[1], GAO Yang[1]

[1](State Key Laboratory for Novel Software Technology (Nanjing University), Nanjing 210023, China)

[2](College of Mathematics and Information Engineering, Longyan University, Longyan 364012, China)

Abstract: Compared with traditional online learning for fixed features, feature evolvable learning usually assumes that features would not vanish or appear in an arbitrary way, while the old features and new features gathered by the hardware devices will disappear and emerge at the same time along with the devices exchanging simultaneously. However, the existing feature evolvable algorithms merely utilize the first-order information of data streams, regardless of the second-order information which explores the correlations between features and significantly improves the classification performance. A confidence-weighted learning for feature evolution (CWFE) algorithm is proposed to solve the aforementioned problem. First, second-order confidence-weighted learning for data streams is introduced to update the prediction model. Next, in order to benefit the learned model, linear mapping during the overlap period is learned to recover the old features. Then, the existing model is updated with the recovered old features, and at the same time, a new predictive model is learned with the new features. Furthermore, two ensemble methods are introduced to utilize these two models. Finally, empirical studies show superior performance over state-of-the-art feature evolvable algorithms.

Key words: machine learning; second-order confidence-weighted; online learning; evolvable features; classification

在现实世界的许多任务中, 数据通常是从开放和动态的环境中收集的, 且以流的形式出现. 因此, 流数

* 基金项目: 国家重点研发计划(2018AAA0100905); 中央引导地方科技发展资金(2021Szvup056); 江苏省重点研发计划(产业前瞻与关键核心技术) (BE2021028); 国家电网公司科学技术项目(SGJS0000DKJS2000952); 龙岩市科技计划(2019LYF13002)
本文由“面向开放场景的鲁棒机器学习”专刊特约编辑陈恩红教授、李宇峰副教授、邹权教授推荐.
收稿时间: 2021-05-31; 修改时间: 2021-07-16; 采用时间: 2021-08-27; jos 在线出版时间: 2021-10-26

据[1,2]具有天然的演化性. 尤其是演化特征空间(evolvable feature space)[3], 它的特征空间可以随着时间的推移而演化, 即以前的特征空间消失而新的特征空间出现. 例如在生态系统中部署用来收集数据的传感器, 每个传感器返回的数据信号对应一个特征, 由于每个传感器的使用寿命有限, 需要用新的传感器替换老化的传感器, 则先前传感器对应的特征(旧特征)空间消失, 而当前传感器对应的特征(新特征)空间出现.

为了利用旧特征空间的历史数据和已学习的预测模型, 探索新特征空间与旧特征空间之间的关系是至关重要. 演化特征空间往往假设特征空间不会任意改变, 且在特征空间演化之前有一个旧特征和新特征同时存在的重叠阶段[3]. 基于这个假设, 特征演化流学习算法(feature evolvable streaming learning, FESL)[3]用在线梯度下降(online gradient descent, OGD)[4]来更新旧特征的模型和新特征的模型; 同时, 在重叠阶段学习一个线性映射, 用新特征空间下的特征来恢复旧特征; 进而在新特征空间下继续使用或更新旧特征已学习的模型; 最后, 将更新后的旧模型和新模型作为两个基模型, 提出了两种集成方法.作为 FESL 的扩展工作, Hou 等人[5]在假设重叠阶段相对较长的情况, 为旧特征和新特征学习一个更为复杂的非线性映射.通常情况下, 集成方法将一系列相对较弱的基模型组合起来, 可以得到比单个基模型效果更好的强模型, 但在集成方法学习中要求每个基模型的分类性能不能太差[6].而在线梯度下降作为基模型更新的方法, 仅在样本分类错误时更新模型, 分类性能不能很好的提升.基于此, 基于被动-主动的特征演化流学习(passive-aggressive learning with feature evolvable streams, PAFE)[7]运用在线被动-主动(online passive-aggressive, PA)[8]算法来更新新、旧特征空间的模型;同时, 在重叠阶段不仅学习了恢复旧特征的映射, 也学习了旧特征到新特征的映射, 以便利用旧模型来对新模型进行初始化, 进而加快模型的收敛和提高模型的性能. 尽管这些特征演化学习算法已获得不错的分类性能, 但这些算法采用的是基于一阶信息的在线学习方法作为预测模型, 而模型更新时所有维度共享相同的学习率, 其分类性能通常会受到限制.

为了解决以上问题, 本文提出了一种特征演化的置信-加权学习算法(confidence-weighted learning for feature evolution, CWFE). 该算法采用基于二阶信息的在线学习方法来更新新旧特征空间的预测模型, 其中, 基于二阶信息的在线学习方法不仅包含了一阶权重信息, 也学习了捕获权重之间相互作用的置信矩阵信息. 在重叠阶段, 学习了从新特征到旧特征的线性映射. 接下来, 在只有新特征空间的情况下, 用线性映射来恢复旧特征, 进而继续更新旧特征空间下已学习的模型和置信矩阵. 继而, 利用集成方法来融合继续更新的旧模型和正在更新的新模型, 以提升当前特征空间(新特征空间)的分类性能. 最后, 实验验证了所提算法的分类性能优于基于一阶在线学习方法的特征演化学习.

本文第 1 节介绍在线学习方法和多样化特征空间的相关工作. 第 2 节给出演化特征空间的具体介绍和形式化表示. 第 3 节阐述本文所提出的特征演化的置信-加权学习算法 CWFE, 包括模型更新和整体流程. 第 4 节分析 CWFE 的实验结果, 验证了所提算法优于已有的基于一阶在线学习方法的特征演化学习算法. 第 5 节总结本文工作, 并针对本文的不足之处提出下一步工作展望.

1 相关工作

我们的工作与机器学习中的两个研究方向有关: 在线学习和多样化特征空间.

1.1 在线学习

在线学习(online learning)[1,9−11]是一种处理大规模流数据挖掘任务的机器学习技术, 可以实时快速地对模型进行增量调整和更新, 提高预测的准确率. 最早的在线学习算法是 20 世纪 50 年代 Rosenblatt 提出的感知器(perceptron)算法[12], 它是一种在线线性分类算法, 旨在解决线性可分的问题. 存在的在线线性分类算法可以分为一阶方法和二阶方法. 作为最简单且最流行的一阶方法 OGD[4], 根据当前样本计算一个梯度, 并对当前模型进行一次梯度下降更新. PA[8]算法是通过一个带约束的优化问题来进行模型更新, 使得新的分类器模型尽可能地接近当前的模型, 同时保证了当前样本到模型的最大间隔. Pegasos 算法[13]为 OGD 方法使用了一个更先进的步长调节方式来解决基于 ℓ_2 范数的 SVM 优化问题. 然而, 基于一阶信息的在线学习方法通常会忽略参数更新的方向. 针对这个问题, 基于二阶信息的置信-加权(confidence-weighted, CW)[14]方法假设预测模

型满足均值为 $\boldsymbol{\mu}\in\mathbb{R}^d$、协方差为 $\boldsymbol{\Sigma}\in\mathbb{R}^{d\times d}$ 的高斯分布, 并给出了错误边界的理论证明. 然而, 置信-加权方法是在数据可分离的假设下进行模型更新, 可能会导致对噪声数据的过拟合. 于是, 权重的自适应正则化(adaptive regularization of weights, AROW)[15]通过采用软边距的平方铰链损失和置信惩罚来放宽数据可分离假设. 作为另一种解决方案, 软置信度加权(soft confidence-weighted, SCW)[16]为不同实例分配自适应边距.

1.2 多样化特征空间

作为多样化特征空间的一种, 演化特征空间的特性是新特征空间出现的同时旧特征空间消失. 然而, 已学习到的旧模型将随着旧特征的消失而被忽略, 这是对已有模型的一种浪费. 为了充分利用旧特征空间中已学习的模型, 假设新旧特征空间存在一定的重叠阶段. 针对这类型的演化特征空间, Hou 等人提出了 FESL 模型[3,5], Liu 等人提出了 PAFE 模型[7]. 然而, 这两种模型在新旧特征重叠阶段假设每个样本的特征都存在. 在实际应用中, 特征演化往往是无法预知的, 例如, 在生态系统中同一时间段部署的传感器, 由于每个传感器周围的环境不同, 如腐蚀性较高的环境、高温环境等, 传感器由于短路、受损、老化等问题被随时更换, 则新的特征可能会随时出现而旧的特征随时消失. 因此, 具有不可预测特征演化的预测学习(prediction with unpredictable feature evolution, PUFE)[17]用频繁方向(frequent directions, FD)技术以流的方式来计算矩阵的行向量, 在重叠阶段通过旧特征空间中可观察的特征来补全已消失的特征. 同时, 数据分布变化经常发生在流数据中, 在重叠阶段学习的新旧特征空间之间的映射函数不再可靠, 特征和分布演化学习(feature and distribution evolving stream learning, FDESL)算法[18]设计了一种跨两个不同特征空间的变异度量技术来衡量旧特征空间中的数据与新特征空间中的数据之间的差异. 与演化特征空间最为相似的是, Hou 等人提出的增量和递减特征空间[19], 它的特性是随着流数据的到来, 部分特征(旧特征)消失, 另一部分特征依旧存在, 同时新的特征出现, 即新旧特征不存在重叠阶段, 但流数据间存在重叠特征. 基于此类演化空间, 一遍增量和递减学习算法(one-pass incremental and decremental learning, OPID)[19]设计了一个具有“压缩-扩展”风格的算法来利用不断演化的特征中的知识. Dong 等人针对增量和递减特征空间设计了一种新颖的在线演化度量学习(online evolving metric learning, EML)[20]算法, 该算法通过结合平滑的 Wasserstein 距离来处理演化的特征. 根据在线更新文本文档的原理, Zhang 等人提出了一种梯形特征空间(trapezoidal feature space)[21,22], 其特性是已有特征不会消失, 新特征不断出现, 即当前样本的特征至少包含前一个样本的所有特征. 基于流特征的在线学习(online learning with streaming features, OL_{SF})[22]结合在线被动-主动学习方法和流特征选择技术对具有无限样本和特征的梯形数据流进行学习. 然而, 不管是具有重叠阶段的特征演化空间, 还是具有重叠特征的增量和递减特征演化空间, 或是呈递增特性的梯形特征空间, 它们都遵循着明确而固定的规律, 因此将当前的演化特征进一步扩展到任意特征空间[23-25]上, 并且取得了不错的实验结果.

已有的在线学习方法大都研究的是具有相同且固定特征空间的流数据. 近几年研究工作者才开始对多样化特征空间进行研究学习, 已有的多样化特征空间学习方法均是采用较为简单的一阶在线学习更新模型.

2 演化特征空间

为方便理解本文所有问题的形式化, 我们先介绍本文使用的一些符号, $(\boldsymbol{x}_t^{S_i},y_t)$ 表示第 t 时刻接收到 S_i 特征空间下的数据实例, 其中, i=1,2, S_1 表示旧特征空间, S_2 表示新特征空间, 且 $\boldsymbol{x}_t^{S_i}\in\mathbb{R}^{d_i}$ 是具有 d_i 维的样本, $y_t\in\{-1,+1\}$是其对应的标签; T_1表示旧特征空间下的样本数; T_2表示新特征空间下的样本数; B表示新旧特征同时存在的重叠阶段. 本文主要关注和研究的是具有重叠阶段的演化特征空间[3], 图 1 给出了在两个特征空间下的演化流数据产生过程, 其中在 T_1时间段内, 旧特征空间 S_1是可访问的, 接收 S_1下的样本 $\boldsymbol{x}_t^{S_1}\in\mathbb{R}^{d_1}$, 并更新 S_1下的预测模型; 在重叠阶段 B时间段内, 旧特征空间 S_1依旧可访问, 同时获得了一些新特征空间 S_2下的样本 $\boldsymbol{x}_t^{S_2}\in\mathbb{R}^{d_2}$, 可学习新特征空间 S_2下的预测模型; 在 T_2时间段内, 原始特征空间 S_1消失, S_1下的预测模型失效, 只能访问新特征空间 S_2下的样本, 并对 S_2下的预测模型进行更新. 在实际应用中, 时间段 T_1和 T_2会比较大, 重叠时间段 B 相对较小, 则它包含的样本对于 S_2中的模型训练可以忽略不计, 因此, 在新特征空间 S_2

情况下, 我们只训练 T_2 时间段内的样本来更新模型.

值得注意的是, 在 T_2 时间段内, 原始特征空间 S_1 下的预测模型对于新特征空间 S_2 下的样本是不可用的, 对于已学习到的模型忽略不用无疑是一种浪费. 因此, 具有重叠阶段的演化特征空间预测问题可以分为 3 个子问题: 基模型更新、旧特征空间恢复以及集成新旧特征空间中的模型.

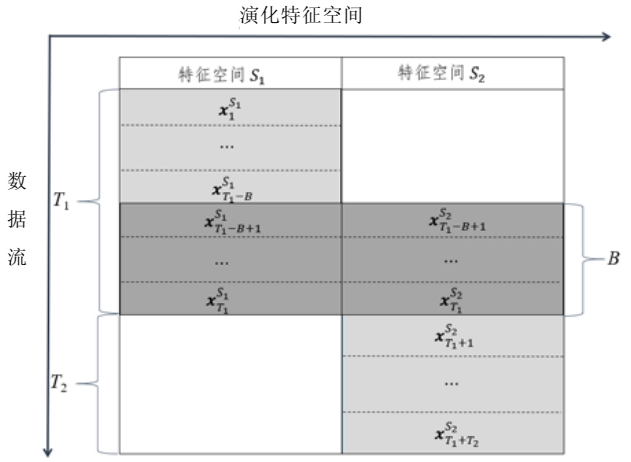


图 1 两个特征空间情况下的演化流数据图示说明

3 特征演化的置信-加权学习方法设计

在本节中, 我们首先运用二阶在线学习方法来进行模型更新; 其次, 学习重叠阶段的线性映射来恢复旧特征空间; 最后, 在只有新特征空间 S_2 的情况下, 继续更新旧模型和学习新模型, 并引入两种集成方法.

3.1 模型更新设计

应从 3 个方面来设计学习的目标函数[15]: 一是学到的新模型在当前训练实例上损失应尽可能的小; 二是学到的新模型在之前的模型上不应更新步长太大; 三是学到的模型对与当前训练实例相同或相似的未来实例的预测应能提高准确率. 具体来说, 在第 t 轮, 当接收到样本 $\boldsymbol{x}_t$ 的真实标签是 y_t 时, 我们将更新模型以确保它在第 t 个实例上遭受较小的损失, 且对预测具有较高的置信度. 其中, 损失函数为 $\ell(\boldsymbol{\mu}^{\mathrm{T}}\boldsymbol{x}_t, y_t)=\max(0, 1-y_t\boldsymbol{\mu}^{\mathrm{T}}\boldsymbol{x}_t)$. 则权重的自适应正则化 AROW[15]的目标函数可以化为如下形式:

$$\mathcal{L}(\boldsymbol{\mu},\boldsymbol{\Sigma})=D_{KL}(\mathcal{N}(\boldsymbol{\mu},\boldsymbol{\Sigma})\,\|\,\mathcal{N}(\boldsymbol{\mu}_{t-1},\boldsymbol{\Sigma}_{t-1}))+\lambda_1\ell_{h^2}(\boldsymbol{\mu}^{\mathrm{T}}\boldsymbol{x}_t,y_t)+\lambda_2\boldsymbol{x}_t^{\mathrm{T}}\boldsymbol{\Sigma}\boldsymbol{x}_t \tag{1}$$

其中, $D_{KL}(\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})\|\mathcal{N}(\boldsymbol{\mu}_{t-1}, \boldsymbol{\Sigma}_{t-1}))$是 KL 散度, 用来约束新旧模型分布差距; $\ell_{h^2}(\boldsymbol{\mu}^{\mathrm{T}}\boldsymbol{x}_t,y_t)=(\max\{0,1-y_t\boldsymbol{\mu}^{\mathrm{T}}\boldsymbol{x}_t\})^2$ 是平方铰链损失, λ_1, $\lambda_2\geqslant 0$ 是平衡超参数. 为了简单起见, 在接下来的论述中, 记 $f_t=\boldsymbol{\mu}^{\mathrm{T}}\boldsymbol{x}_t$ 是模型的预测值, 且 $\lambda_1=\lambda_2=\dfrac{1}{2\gamma}$, 其中, $\gamma>0$. 在公式(1)中, KL 散度的具体形式如下:

$$D_{KL}(\mathcal{N}(\boldsymbol{\mu},\boldsymbol{\Sigma})\,\|\,\mathcal{N}(\boldsymbol{\mu}_{t-1},\boldsymbol{\Sigma}_{t-1}))=\frac{1}{2}\log\left(\frac{\det\boldsymbol{\Sigma}_{t-1}}{\det\boldsymbol{\Sigma}}\right)+\frac{1}{2}Tr(\boldsymbol{\Sigma}_{t-1}^{-1}\boldsymbol{\Sigma})+\frac{1}{2}(\boldsymbol{\mu}_{t-1}-\boldsymbol{\mu})^{\mathrm{T}}\boldsymbol{\Sigma}_{t-1}^{-1}(\boldsymbol{\mu}_{t-1}-\boldsymbol{\mu})-\frac{1}{2}d \tag{2}$$

其中, d 是向量$\boldsymbol{\mu}$的维数.

当 $\ell(f_t, y_t)>0$ 时, 我们按以下两步优化更新模型.

- 更新均值$\boldsymbol{\mu}_t$:

$$\boldsymbol{\mu}_t=\boldsymbol{\mu}_{t-1}+\frac{\boldsymbol{\Sigma}_{t-1}y_t\boldsymbol{x}_t\ell(f_t,y_t)}{\boldsymbol{x}_t^{\mathrm{T}}\boldsymbol{\Sigma}_{t-1}\boldsymbol{x}_t+\gamma} \tag{3}$$

- 更新置信矩阵$\boldsymbol{\Sigma}_t$:

$$\boldsymbol{\Sigma}_t = \boldsymbol{\Sigma}_{t-1} - \frac{\boldsymbol{\Sigma}_{t-1}\boldsymbol{x}_t\boldsymbol{x}_t^{\mathrm{T}}\boldsymbol{\Sigma}_{t-1}}{\boldsymbol{x}_t^{\mathrm{T}}\boldsymbol{\Sigma}_{t-1}\boldsymbol{x}_t + \gamma} \tag{4}$$

3.2 重叠阶段的线性映射

如图1所示: 在T_2时间段内, 只能接收到新特征空间S_2的数据, 观察不到旧特征空间S_1中的历史数据, 从而无法使用旧特征空间已学到的模型. 为了不浪费已学到的模型, 继而用到新模型的预测中, 在重叠阶段 B 内学习了新特征和旧特征的线性映射关系 $\varphi: \mathbb{R}^{d_2} \to \mathbb{R}^{d_1}$, 即 $\varphi(\boldsymbol{x}_t^{S_2}) = \boldsymbol{M}^{\mathrm{T}}\boldsymbol{x}_t^{S_2}$, 进而用 S_2 中的数据来恢复 S_1 中的数据[3,5]. 因此, 求解线性映射的具体形式如下:

$$\arg\min \sum_{t=T_1-B+1}^{T_1} \| \boldsymbol{x}_t^{S_1} - \varphi(\boldsymbol{x}_t^{S_2}) \|^2 = \arg\min \sum_{t=T_1-B+1}^{T_1} \| \boldsymbol{x}_t^{S_1} - \boldsymbol{M}^{\mathrm{T}}\boldsymbol{x}_t^{S_2} \|^2 \tag{5}$$

其中, $\boldsymbol{M}$ 是线性映射对应的系数矩阵. 上述式子可以得到一个闭式解, 具体如下:

$$\boldsymbol{M}_* = \left(\sum_{t=T_1-B+1}^{T_1} \boldsymbol{x}_t^{S_2}(\boldsymbol{x}_t^{S_2})^T\right)^{-1}\left(\sum_{t=T_1-B+1}^{T_1} \boldsymbol{x}_t^{S_2}(\boldsymbol{x}_t^{S_1})^T\right) \tag{6}$$

因此, 在 T_1 时间段内, 学习特征空间 S_1 下的预测模型$\boldsymbol{\mu}_{1,t}$, 以及在重叠阶段 B 内, 学习新旧特征的线性映射的具体流程见算法 1.

算法 1. 特征空间 S_1 的模型 $(\boldsymbol{\mu}_{1,T_1}, \boldsymbol{\Sigma}_{1,T_1})$ 和线性映射φ.

① 初始化: $\boldsymbol{\mu}_{1,0}=\boldsymbol{0}$, $\boldsymbol{\Sigma}_{1,0}=\boldsymbol{I}$, γ=0.1;
② for t=1,…,T_1
③ 接收样本: $\boldsymbol{x}_t^{S_1} \in \mathbb{R}^{d_1}$;
④ 计算: $f_{1,t} = \boldsymbol{\mu}_{1,t-1}^T\boldsymbol{x}_t^{S_1}$,并揭示样本真实标签 $y_t \in \{-1, +1\}$;
⑤ 计算瞬时损失: $\ell(f_{1,t}, y_t) = \max(0, 1 - y_t\boldsymbol{\mu}_{1,t-1}^T\boldsymbol{x}_t^{S_1})$;
⑥ 用公式(3)更新均值$\boldsymbol{\mu}_{1,t}$;
⑦ 用公式(4)更新置信矩阵$\boldsymbol{\Sigma}_{1,t}$;
⑧ if $t>T_1-B$
⑨ 用公式(5)学习线性映射φ;
⑩ end
⑪ end

3.3 两种集成方法CWFE-c和CWFE-s

在 T_2 时间段内,用接收到新特征空间 S_2 的数据来训练新模型 $\boldsymbol{\mu}_{2,t}$, 同时利用线性映射来恢复旧特征空间的数据, 进而继续更新已学到的模型 $\boldsymbol{\mu}_{1,T_1}$, 同时引入两种集成方法:组合预测(CWFE-c)和当前最优预测(CWFE-s)[3,5]来提高两种基模型的预测性能.

(1) 组合预测 CWFE-c:在 T_2 时间段内, 对新旧模型的指数加权平均预测.它只取决于所有基模型过去的表现进行预测, 且分配给基模型的权重以简单增量的方式计算.令 $a_{1,t-1}$, $a_{2,t-1}$ 分别是新旧模型在第 t 时刻的权重, 则组合预测模型如下:

$$p_t = a_{1,t-1}f_{1,t} + a_{2,t-1}f_{2,t} \tag{7}$$

其中, $f_{1,t}$ 是旧模型预测值, $f_{2,t}$ 是新模型预测值.权重的更新如下所示:

$$a_{i,t} = \frac{v_{i,t}}{v_{1,t} + v_{2,t}},\ i = 1, 2 \tag{8}$$

其中, $v_{i,t} = a_{i,t}\mathrm{e}^{-\eta\ell(f_{i,t}, y_t)}$, 且$\eta = \sqrt{8(\ln 2)/T_2}$. 则关于组合预测 CWFE-c 的具体流程见算法 2.

算法 2. 特征空间 S_2 的组合预测模型 CWFE-c.

① 初始化: $a_{1,T_1} = a_{2,T_1} = 1/2$, $\eta = \sqrt{8(\ln 2)/T_2}$, $\boldsymbol{\mu}_{2,T_1} = \boldsymbol{0}$, $\boldsymbol{\Sigma}_{2,T_1} = \boldsymbol{I}$, $\gamma = 0.1$;
② 调用算法 1, 得到 $(\boldsymbol{\mu}_{1,T_1}, \boldsymbol{\Sigma}_{1,T_1})$ 和φ;

③ for $t=T_1+1, \ldots, T_1+T_2$

④ 接收样本: $\boldsymbol{x}_t^{S_2} \in \mathbb{R}^{d_2}$;

⑤ 计算新旧模型预测值: $f_{1,t}=\boldsymbol{\mu}_{1,t-1}^T\varphi(\boldsymbol{x}_t^{S_2})$, $f_{2,t}=\boldsymbol{\mu}_{2,t-1}^T\boldsymbol{x}_t^{S_2}$;

⑥ 用式(7)计算组合预测值 p_t, 并揭示样本真实标签 $y_t\in\{-1,+1\}$;

⑦ 计算瞬时损失: $\ell(p_t, y_t)$;

⑧ 用公式(8)更新权重 $a_{1,t}, a_{2,t}$;

⑨ 用公式(3)和公式(4), 分别更新$(\boldsymbol{\mu}_{1,t}, \boldsymbol{\Sigma}_{1,t})$和$(\boldsymbol{\mu}_{2,t}, \boldsymbol{\Sigma}_{2,t})$;

⑩ end

在 T_2 时间段内, 随着新特征空间 S_2 中数据的累积, 在重叠阶段 B 内学习的线性映射越来越不可靠, 则旧模型在 T_2 时间段内的表现会变得越来越差.因此, 引入第 2 个集成方法:当前最优预测.

(2) 当前最优预测 CWFE-s: 在 T_2 时间段内通过权重的分布来选择权重较大的基模型进行预测, 形式如下:

$$u_{i,t}=\frac{a_{i,t-1}}{a_{1,t-1}+a_{2,t-1}},\ i=1,2 \tag{9}$$

其中, 权重 $a_{i,t}$ 的更新如下所示:

$$a_{i,t}=\frac{\delta}{2}\Delta_t+(1-\delta)v_{i,t},\ i=1,2 \tag{10}$$

其中, $\delta=\dfrac{1}{T_2-1}$, $v_{i,t}=a_{i,t-1}\mathrm{e}^{-\eta\ell(f_{i,t},y_t)}$, $i=1,2$, $\Delta_t=v_{1,t}+v_{2,t}$, $H(x)=-x\ln x-(1-x)\ln(1-x)$ 是在 $x\in(0,1)$下的熵函数, 且 $\eta=\sqrt{8/T_2(2\ln 2+(T_2-1)H(1/(T_2-1)))}$. 则当前最优预测 CWFE-s 的具体流程见算法 3.

算法 3. 特征空间 S_2 的当前最优预测模型 CWFE-s.

① 初始化: $a_{1,T_1}=a_{2,T_1}=1/2$, $\eta=\sqrt{8/T_2(2\ln 2+(T_2-1)H(1/(T_2-1)))}$, $\boldsymbol{\mu}_{2,T_1}=\boldsymbol{0}$, $\boldsymbol{\Sigma}_{2,T_1}=\boldsymbol{I}$, $\gamma=0.1$;

② 调用算法 1, 得到 $(\boldsymbol{\mu}_{1,T_1},\boldsymbol{\Sigma}_{1,T_1})$ 和φ;

③ for $t=T_1+1, \ldots, T_1+T_2$

④ 接收样本: $\boldsymbol{x}_t^{S_2} \in \mathbb{R}^{d_2}$;

⑤ 计算新旧模型预测值: $f_{1,t}=\boldsymbol{\mu}_{1,t-1}^T\varphi(\boldsymbol{x}_t^{S_2})$, $f_{2,t}=\boldsymbol{\mu}_{2,t-1}^T\boldsymbol{x}_t^{S_2}$;

⑥ 根据公式(9)得到模型$(\boldsymbol{\mu}_{i,t}, \boldsymbol{\Sigma}_{i,t})$, 并预测 $p_t=f_{i,t}$;

⑦ 揭示样本真实标签 $y_t\in\{-1,+1\}$, 计算瞬时损失: $\ell(p_t, y_t)$;

⑧ 用公式(10)更新权重 $a_{1,t}, a_{2,t}$;

⑨ 用公式(3)和公式(4), 分别更新$(\boldsymbol{\mu}_{1,t}, \boldsymbol{\Sigma}_{1,t})$和$(\boldsymbol{\mu}_{2,t}, \boldsymbol{\Sigma}_{2,t})$;

⑩ end

4 实验设计和结果分析

实验中选用了 8 个合成数据集(http://www.lamda.nju.edu.cn/code_FESL.ashx)对所提算法进行对比研究, 其详细信息见表 1.

表 1 实验数据集说明

数据集	#样本	#特征空间 S_1	#特征空间 S_2
Australian	690	42	29
Credit-a	653	15	10
Credit-g	1 000	20	14
Diabetes	768	8	5
German	1 000	59	41
Kr-vs-kp	3 196	36	25
Splice	3 175	60	42
Svmguide3	1 284	22	15

4.1 对比算法和实验设置

本文将所提算法 CWFE-c、CWFE-s 和其过程中包含的 3 个算法 NCW、RCW-u、RCW-f 进行对比, 同时与基于一阶信息的特征演化学习算法 FESL[5]和 PAFE[7]进行对比. 其中, 对比的时间段为只可获得新特征空间数据的 T_2 时间段.

- NOGD[5]、NCW: 分别通过 OGD、AROW 对新特征空间 S_2 进行模型更新;
- NPA-d[7]: 通过 PA 对 S_2 进行模型更新, 不同于 NOGD、NCW 的是, 通过重叠阶段得到旧特征到新特征的映射, 进而用已学到的旧模型对新模型进行初始化;
- ROGD-u[5]、RPA-u[7]、RCW-u: 通过利用重叠阶段学习的线性映射, 用 S_2 中的数据来恢复旧特征空间 S_1 中的数据, 进而继续分别用 OGD、PA、AROW 来更新旧特征空间 S_1 已学到的模型;
- ROGD-f[5]、RPA-f[7]、RCW-f: 在新特征空间 S_2 下, 直接运用旧特征空间 S_1 已学到的旧模型来预测被恢复的旧特征空间中的数据, 旧模型不更新;
- FESL-c[5]、FESL-s[5]、PAFE-c[7]、PAFE-s[7]、CWFE-c、CWFE-s: 分别将新旧特征空间中的模型 NOGD 和 ROGD-u、NPA-d 和 RPA-u、NCW 和 RCW-u 进行组合预测-c 和选择当前最优预测-s.

为了验证算法的有效性, 文中采用 2 种评价标准: 分类精度(accuracy)和 *F*-measure, 其结果是通过 10 次独立运行的平均结果. 同时, 所有数据集的旧特征时间段 T_1 和新特征时间段 T_2 均设置为样本数的一半, 重叠阶段的大小设置为 10. 另外, 本文所提算法中的超参数为 $\gamma = 0.1$, 是用来平衡损失和数据置信度. 对比算法 FESL[5]中步长的设置和 PAFE[7]中超参数的设置和原文献中的参数设置保持一致.

4.2 结果分析

表 2 给出了所有对比算法在表 1 数据集上的 Accuracy 和 *F*-measure 结果.

表 2 所有对比算法在 8 个合成数据集上的平均测试性能对比

算法	Australian		Credit-a		Credit-g		Diabetes	
	Accuracy	*F*-measure	Accuracy	*F*-measure	Accuracy	*F*-measure	Accuracy	*F*-measure
NOGD	.767	.791	.811	.827	.659	.401	.650	.006
NPA-d	.771	.792	.797	.808	.620	.364	.597	.329
NCW (ours)	•.864	•.872	•.854	•.856	•.750	•.487	•.678	•.401
ROGD-u	.849	.863	.826	•.861	.733	•.444	.650	.067
RPA-u	.849	.861	.855	.859	.727	.336	.669	.341
RCW-u (ours)	•.855	•.867	•.856	.858	•.737	•.444	•.679	•.400
ROGD-f	.809	•.835	.784	.809	•.716	.302	.650	.023
RPA-f	•.811	.834	•.803	•.817	.704	.028	.652	.137
RCW-f (ours)	.784	.799	.716	.736	.712	•.410	•.677	•.383
FESL-c	.849	.863	.854	**•.860**	.733	.444	.649	.040
FESL-s	.849	.863	.854	**•.860**	.733	.444	.648	.066
PAFE-c	.849	.861	**•.855**	.858	.727	.336	.668	.343
PAFE-s	.849	.861	**•.855**	.858	.727	.336	.667	.343
CWFE-c (ours)	**•.868**	**•.876**	**•.855**	.857	.749	.477	**•.678**	.399
CWFE-s (ours)	.865	.873	.854	.856	**•.750**	**•.485**	**•.678**	**•.401**

算法	German		Kr-vs-kp		Splice		Svmguide3	
	Accuracy	*F*-measure	Accuracy	*F*-measure	Accuracy	*F*-measure	Accuracy	*F*-measure
NOGD	.684	.293	.612	.572	.568	.581	.680	.326
NPA-d	.653	.427	.660	.646	.573	.589	.648	.305
NCW (ours)	•.758	•.531	•.908	•.902	•.768	•.773	•.808	•.501
ROGD-u	.700	.000	.678	.574	.612	.631	.779	.345
RPA-u	.705	.236	.732	.707	.604	.624	.780	.373
RCW-u (ours)	•.711	•.316	•.743	•.722	•.616	•.632	•.792	•.448
ROGD-f	.700	.000	.564	.173	.567	.526	•.748	.114
RPA-f	•.705	.136	.654	•.624	•.569	.512	.741	.032
RCW-f (ours)	.697	•.311	•.670	.622	.561	•.547	.738	•.360
FESL-c	.700	.002	.678	.575	.612	.631	.779	.346
FESL-s	.703	.130	.666	.588	.612	.631	.778	.345
PAFE-c	.705	.236	.731	.707	.604	.624	.780	.373
PAFE-s	.705	.236	.731	.707	.604	.624	.780	.373
CWFE-c (ours)	**•.758**	.515	.907	**•.902**	**•.768**	**•.773**	.807	.493
CWFE-s (ours)	**•.758**	**•.522**	**•.908**	**•.902**	**•.768**	**•.773**	**•.808**	**•.501**

表 2 中, 用•标记每个网格中较好的结果, 用粗体标注集成方法中最好的结果.

在表 2 的最后一个网格中, 即集成方法的对比中, 本文所提算法 CWFE-c 和 CWFE-s 的 Accuracy 结果均优于其他对比算法, 尤其是在 Kr-vs-kp 数据集上, 所提算法 CWFE 分别比 FESL 和 PAFE 高出 0.23, 0.17 个精度点; 在 Splice 数据集上, 所提算法 CWFE 比 FESL 和 PAFE 的 Accuracy 高出 0.16 左右; 而在其他数据集上, 所提算法的 Accuracy 结果均高出 0.05 以内. 这个现象说明了在特征演化学习上, 基于二阶在线学习方法的模型更新得到的分类性能要优于基于一阶在线学习方法的模型预测. 同时, 作为集成方法的两个基模型, NCW 和 RCW-u 在表 2 的前两个网格内的 Accuracy 结果也均优于 NOGD, NPA-d 和 ROGD-u, RPA-u 的结果, 尤其是 NCW 在 Kr-vs-kp 的结果比 NOGD 和 NPA-d 分别高出 0.29, 0.24 个精度点, 在 Australian, Credit-g, German, Splice 和 Svmguide3 等 5 个数据集上, 均高出 0.07 以上的精度点. RCW-u 在 8 个数据集上的 Accuracy 结果均比 ROGD-u, RPA-u 高出 0.01 左右, 再次验证了基于二阶在线学习方法进行模型更新的特征演化学习的分类性能较一阶在线学习方法的更新算法有明显提升. 第 3 个网格内的对比结果显示了一个有趣的现象: 本文所提算法 RCW-f 在 8 个数据集上, 只有 2 个数据集上的结果高于 ROGD-f, RPA-f. 这一类算法的设定是旧特征学习的模型不进行更新, 直接用于预测用新特征空间恢复的旧特征空间数据. 这可能的原因可以归结为如下两点: (1) 随着新特征空间的数据越来越多, 在重叠阶段学习的新特征到旧特征的映射越来越不可靠; (2) 相比于一阶在线学习方法, 基于二阶的在线学习方法更新的模型挖掘了特征之间的关系. 另外, CWFE 的 *F*-measure 结果除了在 Credit-a 数据集上比 FESL 低了 0.003 个精度点, 在其他 7 个数据集上均高于 FESL 和 PAFE 算法. 这也再次验证了本文所提的基于二阶的特征演化学习方法比基于一阶的特征演化学习方法更为有效.

表 3 给出了所有对比算法中两种集成模型在 8 个数据集上的运行时间.从表 3 中可以看出, 基于一阶信息的特征演化学习方法 FESL 和 PAFE 的运行时间大致相同, 而本文所提的基于二阶信息的特征演化学习方法 CWFE 的运行时间明显高于 FESL 和 PAFE 的运行时间.这是因为基于二阶信息的特征演化学习方法需要计算特征的置信矩阵信息, 因此往往具有较高的时间复杂度.

表 3 所有对比算法在 8 个合成数据集上的运行时间 (ms)

算法	Australian	Credit-a	Credit-g	Diabetes	German	Kr-vs-kp	Splice	Svmguide3
FESL-c	3.4	2.8	5.3	3.4	5.3	15.1	16.3	5.5
FESL-s	7.1	6.1	11.4	7.8	10.7	31.8	32.8	12.1
PAFE-c	3.4	2.5	4.7	4.0	5.7	15.3	17.8	5.6
PAFE-s	7.7	6.5	10.2	8.4	12.2	33.7	34.3	12.2
CWFE-c (ours)	9.7	5.7	10.0	6.5	23.0	38.2	68.4	13.0
CWFE-s (ours)	13.4	8.9	15.2	10.2	28.1	54.2	84.3	19.8

为进一步验证表 2 中结果分析的合理性, 现给出所提算法在 8 个数据集上平均累计损失趋势图, 如图 2 所示. 在第 t 时刻, $\overline{\ell}_t$ 代表在 1, …, t 上的平均累积损失, 即 $\overline{\ell}_t=(1/t)\sum_{i=1}^{t}\ell_i$. 同时, 平均累积损失越小越好.

从图 2 的实验结果来看, 我们有以下观察结果.

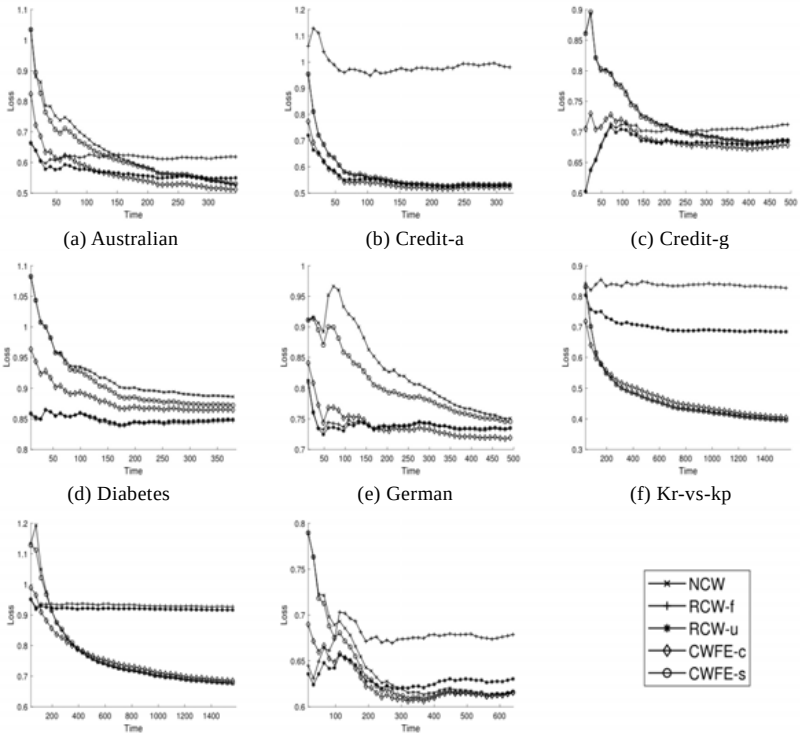
(1) NCW 是带叉号标记的曲线, 在所有数据集上, 随着到来样本数量的增加, 平均累积损失迅速下降. 这符合 T_1+1, …, T_1+T_2 轮的现象, 即: 随着样本越来越多, 分类性能越来越好;

(2) RCW-u 是带星号标记的曲线, 在大部分数据集上呈下降趋势; 但相对于 NCW, 下降趋势不是很明显. 是因为 RCW-u 在 1, …, T_1 轮在特征空间 S_1 上已趋于收敛, 所以在更多的恢复数据上再次更新不会带来太多的性能提升; 而 RCW-u 曲线在 Credit-g 数据集上先迅速上升而后呈下降趋势, 在 Diabetes 和 Splice 基本平稳不变. 这说明利用重叠阶段恢复的旧特征数据不一定是可靠的, 则随着恢复数据的增多, 不但不会带来性能的提升, 反而会降低性能;

(3) RCW-f 是带加号标记的曲线, 在 Credit-a 和 German 数据集上先呈下降趋势迅速进入平稳状态, 在 Diabetes, Kr-vs-kp 和 Splice 数据集上呈平稳状态, 在 Australian, Credit-g 和 Svmguide3 数据集上呈上升趋势. 这都是合理的, 因为 RCW-f 是旧特征已学到模型在新特征空间中固定不变, 如果恢复的

数据是不可靠的甚至是错误的, 随着更多的恢复数据, 它会表现得更差;

(4) 本文所提的方法 CWFE-c 带菱形标记的曲线和 CWFE-s 带圆圈标记的曲线是基于 NCW 和 RCW-u 的集成模型, 因此它们的平均累积损失也会减少.



(a) Australian (b) Credit-a (c) Credit-g

(d) Diabetes (e) German (f) Kr-vs-kp

(g) Splice (h) Svmguide3 (i) Legend

图 2 3 个基模型与所提算法在 8 个合成数据集上的平均累积损失趋势

5 总结与展望

本文首次将基于二阶信息的在线学习方法引入到演化特征空间中, 并提出了一种基于置信-加权的特征演化学习算法(CWFE). 置信-加权方法是经典的基于二阶信息的在线学习方法, 在学习过程中不仅考虑了一阶信息的模型更新, 同时挖掘了数据特征之间的关系, 进而提高了模型预测性能. 同时, 将旧特征空间已学到的模型和置信矩阵均用于恢复特征的模型更新中, 相对于一阶方法的特征演化学习, 性能得到了较为明显的提升. 尽管基于二阶信息的在线学习方法可以获得较高的预测性能, 且收敛速度更快, 但它的时空复杂度也较高. 在高维数据面前, 从平衡时空复杂度和性能两个角度来看, 二阶方法和一阶方法的对比并不占优势.因此, 在对预测性能不影响或者影响较小的情况下, 如何改进二阶方法的时空复杂度, 并应用到高维的特征演化学习中, 是我们下一步重点关注的问题.

References:

[1] Zhai TT, Gao Y, Zhu JW. Survey of online learning algorithms for streaming data classification. Ruan Jian Xue Bao/Journal of Software, 2020, 31(4): 912−931 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/5916.htm [doi: 10.13328/j.cnki.jos.005916]

[2] Guo HS, Zhang AJ, Wang WJ. Concept drift detection method based on online performance test. Ruan Jian Xue Bao/Journal of Software, 2020, 31(4): 932−947 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/5917.htm [doi: 10.13328/j.cnki.jos.005917]

[3] Hou B, Zhang L, Zhou Z. Learning with feature evolvable streams. In: Proc. of the 30th Annual Conf. on Neural Information Processing Systems. 2017. 1417−1427.

[4] Zinkevich M. Online convex programming and generalized infinitesimal gradient ascent. In: Proc. of the 20th Int'l Conf. on Machine Learning. 2003. 928−936.

[5] Hou B, Zhang L, Zhou Z. Learning with feature evolvable streams. IEEE Trans. on Knowledge and Data Engineering, 2021, 33(6): 2602−2615.

[6] Freund Y, Schapire RE. A decision-theoretic generalization of on-line learning and an application to boosting. Journal of Computer and System Sciences, 1997, 55(1): 119−139.

[7] Liu YF, Li WB, Gao Y. Passive-aggressive learning with feature evolvable streams. Journal of Computer Research and Development, 2021, 58(8): 1575−1585 (in Chinese with English abstract).

[8] Crammer K, Dekel O, Keshet J, *et al*. Online passive-aggressive algorithms. Journal of Machine Learning Research, 2006, 7(3): 551−585.

[9] Li ZJ, Li YX, Wang F, *et al*. Online learning algorithms for big data analytics: A survey. Journal of Computer Research and Development, 2015, 52(8): 1707−1721 (in Chinese with English abstract).

[10] Pan ZS, Tang SQ, Qiu JX, *et al*. Survey on online learning algorithms. Journal of Data Acquisition and Processing, 2016, 31(6): 1067−1082 (in Chinese with English abstract).

[11] Hoi S, Wang J, Zhao P. Libol: A library for online learning algorithms. Journal of Machine Learning Research, 2014, 15(1): 495−499.

[12] Rosenblatt F. The perceptron: A probabilistic model for information storage and organization in the brain. Psychological Review, 1958, 65(6): 386−408.

[13] Shalev-Shwartz S, Singer Y, Srebro N, *et al*. Pegasos: Primal estimated sub-gradient solver for SVM. Mathematical Programming, 2011, 127(1): 3−30.

[14] Crammer K, Dredze M, Pereira F. Exact convex confidence-weighted learning. In: Proc. of the 21st Annual Conf. on Neural Information Processing Systems. 2008. 345−352.

[15] Crammer K, Kulesza A, Dredze M. Adaptive regularization of weight vectors. In: Proc. of the 22nd Annual Conf. on Neural Information Processing Systems. 2009. 414−422.

[16] Hoi S, Wang J, Zhao P. Exact soft confidence-weighted learning. In: Proc. of the 29th Int'l Conf. on Machine Learning. 2012. 107−114.

[17] Hou B, Zhang L, Zhou Z. Prediction with unpredictable feature evolution. IEEE Trans. on Neural Networks and Learning Systems, 2021, 1−10.

[18] Zhang Z, Zhao P, Jiang Y, *et al*. Learning with feature and distribution evolvable streams. In: Proc. of the 37th Int'l Conf. on Machine Learning. 2020. 11317−11327.

[19] Hou C, Zhou Z. One-pass learning with incremental and decremental features. IEEE Trans. on Pattern Analysis and Machine Intelligence, 2018, 40(11): 2776−2792.

[20] Dong J, Cong Y, Sun G, *et al*. Evolving metric learning for incremental and decremental features. IEEE Trans. on Circuits and Systems for Video Technology, 2021, 14(8): 1−13.

[21] Zhang Q, Zhang P, Long G, *et al*. Towards mining trapezoidal data streams. In: Proc. of the 2015 IEEE Int'l Conf. on Data Mining. 2015. 1111−1116.

[22] Zhang Q, Zhang P, Long G, *et al*. Online learning from trapezoidal data streams. IEEE Trans. on Knowledge and Data Engineering, 2016, 28(10): 2709−2723.

[23] He Y, Wu B, Wu D, *et al*. Online learning from capricious data streams: A generative approach. In: Proc. of the 28th Int'l Joint Conf. on Artificial Intelligence. 2019. 2491−2497.

[24] He Y, Wu B, Wu D, *et al*. Toward mining capricious data streams: A generative approach. IEEE Trans. on Neural Networks and Learning Systems, 2021, 32(3): 1228−1240.

[25] Beyazit E, Alagurajah J, Wu X. Online learning from data streams with varying feature spaces. In: Proc. of the 33rd AAAI Conf. on Artificial Intelligence. 2019. 3232−3239.

附中文参考文献:

[1] 翟婷婷, 高阳, 朱俊武. 面向流数据分类的在线学习综述. 软件学报, 2020, 31(4): 912–931. http://www.jos.org.cn/1000-9825/5916.htm [doi: 10.13328/j.cnki.jos.005916]

[2] 郭虎升, 张爱娟, 王文剑. 基于在线性能测试的概念漂移检测方法. 软件学报, 2020, 31(4): 932–947. http://www.jos.org.cn/1000-9825/5917.htm [doi: 10.13328/j.cnki.jos.005917]

[7] 刘艳芳, 李文斌, 高阳. 基于被动-主动的特征演化流学习. 计算机研究与进展, 2021, 58(8): 1575–1585.

[9] 李志杰, 李元香, 王峰, 等. 面向大数据分析的在线学习算法综述. 计算机研究与发展, 2015, 52(8): 1707–1721.

[10] 潘志松, 唐斯琪, 邱俊洋, 等. 在线学习算法综述. 数据采集与处理, 2016, 31(6): 1067–1082.

刘艳芳(1987－), 女, 博士生, 讲师, CCF 专业会员, 主要研究领域为在线学习, 机器学习, 数据挖掘.

李文斌(1991－), 男, 博士, 助理研究员, CCF 专业会员, 主要研究领域为机器学习, 度量学习, 小样本学习, 计算机视觉.

高阳(1972－), 男, 博士, 教授, 博士生导师, CCF 杰出会员, 主要研究领域为强化学习, 多智能体系统, 计算机视觉, 大数据分析.