软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software,2020,31(4):912–931 [doi: 10.13328/j.cnki.jos.005916]

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

面向流数据分类的在线学习综述*

翟婷婷[1,2], 高 阳[2], 朱俊武[1]

[1](扬州大学 信息工程学院,江苏 扬州 225127)

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

通讯作者: 翟婷婷, E-mail: zhtt.go@gmail.com

摘 要: 流数据分类旨在从连续不断到达的流式数据中增量学习一个从输入变量到类标变量的映射函数,以便对随时到达的测试数据进行准确分类.在线学习范式作为一种增量式的机器学习技术,是流数据分类的有效工具.主要从在线学习的角度对流数据分类算法的研究现状进行综述.具体地,首先介绍在线学习的基本框架和性能评估方法,然后着重介绍在线学习算法在一般流数据上的工作现状,在高维流数据上解决"维度诅咒"问题的工作现状,以及在演化流数据上处理"概念漂移"问题的工作现状,最后讨论高维和演化流数据分类未来仍然存在的挑战和亟待研究的方向.

关键词: 在线学习;流数据分类;维度诅咒;概念漂移;稀疏在线学习;演化流分类

中图法分类号: TP181

中文引用格式: 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

英文引用格式: 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). http://www.jos.org.cn/1000-9825/5916.htm

Survey of Online Learning Algorithms for Streaming Data Classification

ZHAI Ting-Ting[1,2], GAO Yang[2], ZHU Jun-Wu[1]

[1](School of Information Engineering, Yangzhou University, Yangzhou 225127, China)

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

Abstract: The objective of streaming data classification is to learn incrementally a decision function that maps input variables to a label variable, from continuously arriving streaming data, so as to accurately classify the test data that may arrive anytime. The online learning paradigm, as an incremental machine learning technology, is an effective tool for classification of streaming data. This paper mainly summarizes, from the perspective of online learning, the recent development of algorithms for streaming data classification. Specifically, the basic framework and the performance evaluation methodology of online learning are first introduced. Then, the latest development of online learning algorithms for general streaming data, for alleviating the "curse of dimensionality" problem in high-dimensional streaming data, and for resolving the "concept drifting" problem in evolving streaming data are reviewed respectively. Finally, future challenges and promising research directions for classification of high-dimensional and evolving streaming data are also discussed.

Key words: online learning; streaming data classification; curse of dimensionality; concept drifting; sparse online learning; evolving data stream classification

* 基金项目: 国家重点研发计划(2017YFB0702600, 2017YFB0702601); 国家自然科学基金(61906165, 61432008, 61872313); 江苏省高等学校自然科学研究项目(19KJB520064)

Foundation item: National Key Research and Development Program of China (2017YFB0702600, 2017YFB0702601); National Natural Science Foundation of China (61906165, 61432008, 61872313); Natural Science Foundation of the Jiangsu Higher Education Institutions of China (19KJB520064)

本文由"非经典条件下的机器学习方法"专题特约编辑高新波教授、黎铭教授、李天瑞教授推荐.

收稿时间: 2019-02-22; 修改时间: 2019-07-11; 采用时间: 2019-09-20; jos 在线出版时间: 2020-01-10

CNKI 在线出版时间: 2020-01-14 09:53:08, http://kns.cnki.net/kcms/detail/11.2560.TP.20200114.0952.001.html

21 世纪以来,随着互联网、电子商务、移动通信和物联网等技术的飞速发展,人们可搜集到的数据呈现爆炸性增长,以“大数据”驱动的数据科学应运而生,且受到世界各国政府、知名企业和科研机构的高度重视和密切关注.流数据是大数据的一个重要来源[1,2],源自日常生活中的许多简单操作,例如刷信用卡、网上购物以及网络社交等.从流式数据中学习展现出前所未有的挑战[3].

(1) 无限的数据量:数据以流的形式连续不断地来到,因此存储所有的数据进行多遍扫描是不现实的.为了能够处理无限量的数据,学习算法需要能够工作在资源受限的环境中,算法存储学习模型所需的空间代价应独立于所处理的样本数.

(2) 样本产生速度快:这个特点要求算法必须具备实时处理和分析的能力.

流数据分类是流数据挖掘中一项非常重要的研究任务,该任务旨在从流式数据中增量学习一个从输入变量到类标变量的映射函数,以便对随时到达的新样本进行准确分类.传统的统计机器学习算法,又称为“批量学习”算法,例如,决策树算法 ID3、C4.5 和 CART 以及基于序列最小优化的 SVM 等,在学习过程中需要多遍扫描所有可利用的数据,且一旦训练结束,当有新的训练数据到来时,不能在旧模型的基础上进行增量更新,只能重新训练一个新模型,因此很难处理流数据带来的样本无限量和实时分析的挑战.相比之下,在线机器学习(简称在线学习)作为一种增量式的机器学习技术,能够对模型进行实时增量更新,学习过程中随时可以使用当前学到的模型对未知样本进行预测,且一旦对样本处理完毕,经常不需要对其进行存储和再访问,因此在线学习非常适用于处理大规模流式数据.

除了在数据流应用上具有优势以外,在学习理论方面,在线学习与统计机器学习相比也具有一定的优势.统计机器学习假设给定一个训练数据集,该集合中的样本是独立同分布的,且服从某个未知分布 P.基于该假设,统计机器学习旨在从假设空间中寻找一个期望风险最小的模型,使用该模型对测试数据集进行预测,其中,模型的期望风险定义为使用该模型对采样于分布 P 中的样本进行预测所产生的期望损失[4].在线学习假设数据按照时间顺序逐个到达,特别地,算法在 t_1 时刻只能观察到从 t=1 到 t=t_1 时刻的数据,而不能观察到 t>t_1 时刻的数据,在每个时刻 t,算法都基于当前观察到的数据序列构建一个在线预测器,以便对 t+1 时刻到达的数据进行在线预测.在线学习不需要假设数据采样于某个固定分布,它旨在找到一个在线预测器序列,使得悔恨最小化[5−7],其中悔恨定义为算法使用其生成的预测器序列进行在线预测所产生的累积损失与使用假设空间中某个固定最优的预测器进行在线预测所产生的累积损失之差.对比发现,统计机器学习的定义依赖于样本独立同分布假设,而在线学习则不依赖于此假设.在线学习无需对数据分布作任何假设的特点使其也适用于数据分布随着时间发生变化的演化流数据上.事实上,严格意义上讲,在线学习比统计机器学习更难,因为在线学习算法可被用来求解统计学习问题.具体地,Cesa-Bianchi 等人[8]证明,通过在线到批量转换技术(online-to-batch conversion),可以将一个运行在独立同分布的数据集合上的具有“低悔恨”的在线学习算法转换为一种“低风险”的批量学习算法.

由于在线学习在理论和应用方面的优势,以及近年来大数据应用需求的显著增长,在线学习已成为热门的研究方向.尤其是在 2007 年,Shai 等人首次利用随机次梯度下降设计出一种高效的在线 SVM 求解算法——Primal Estimated Sub-gradient Solver for SVM,简称 Pegasos[9],该算法具有良好的收敛保证,且在大规模的文本分类数据集上,与先进的批量 SVM 算法相比,Pegasos 在保持良好的泛化性能的同时,学习效率提高了一个数量级,由此掀起了在线学习在机器学习领域的研究和应用高潮,并在近 5 年持续成为 ICML[10−13]、NIPS[14−18]等顶级国际会议和 JMLR[19−23]、TKDE[24−26]等顶级国际期刊的热点研究问题.

本文主要从在线学习算法适用的流数据的特点对算法进行综述,具体地,分别调查了适用于一般流数据、高维流数据和演化流数据的在线分类算法研究现状.尽管 Hoi 等人[27]近来对在线学习算法给出一个颇为全面的综述,根据在线反馈的类型和任务监督的类型对算法进行分类,但本文主要根据在线算法适用的数据特点对算法进行分类,而且 Hoi 等人的工作中缺乏对演化流数据在线算法的调查,因此本文的工作与 Hoi 等人的工作互相补充.在国内,李志杰等人[28]和潘志松等人[29]分别在 2015 年和 2016 年对在线学习算法进行了综述,两篇综述均从在线算法的学习特点对算法进行分类,既没有考虑在线算法所适用的流数据特点,也没有区分在线学习的两种不同的学习设置——完全信息设置和部分信息设置,从这个角度来看,本文有望弥补上述不足.

本文第 1 节介绍在线学习的基本定义和性能评价方法.第 2 节~第 4 节分别对面向一般流数据、高维流数据以及演化流数据的在线分类算法进行详细的对比分析.第 5 节讨论高维和演化流数据上的分类任务仍然存在的挑战以及具有一定前途的研究方向.最后,第 6 节对全文进行总结.

1 在线学习

在线学习提供了一种可扩展性良好且灵活的、方法建模广泛的预测问题,例如分类、回归和排名问题等[6,7].感知机[30]或许是第一个也是最简单的在线学习算法,它被设计以回答一系列的 Yes 或 No 的问题,其中每个问题被表示为一个特征向量,也称为实例,问题的答案用类标表示,回答问题所用的预测器被表示为向量空间中的一个超平面,又称为权重向量.感知机算法以“可加”的方式来更新它的权重向量,即在每次更新时将权重向量加上或减去当前的输入实例.随着时间的发展,更多复杂的在线学习算法相继被提了出来.本节接下来对在线学习的一般定义和性能评价方法进行介绍.

1.1 在线学习定义

一个在线学习过程可以被建模为一个重复的预测游戏[6],如图 1 所示:在第 t 轮游戏中,环境从某个问题域 X 中取出一个新问题 $\boldsymbol{x}_t$,学习算法被要求从某个已知的决策空间 D 中选择一个决策模型 $\boldsymbol{w}_t$ 来回答该问题,一旦算法提交了它的回答 $\hat{y}_t \in Y$,环境就揭示该问题的正确答案 $y_t \in Y$,然后算法受到损失 $l(\boldsymbol{w}_t;(x_t,y_t))$,该损失称为瞬时损失,衡量使用 $\boldsymbol{w}_t$ 预测 $\boldsymbol{x}_t$ 的类标 y_t 的不准确程度,根据该损失信息,算法可以改进其决策模型,以便在下一轮中做出更好的预测.

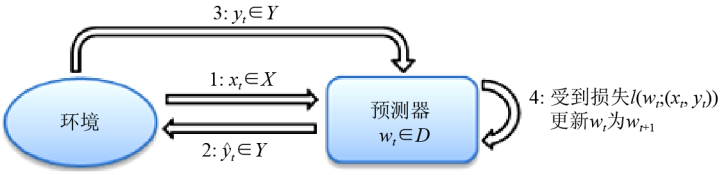

Fig.1 The process of online learning

图 1 在线学习过程

在上述定义中,当 $X=\mathbb{R}^d$, $Y=\{+1,-1\}$时,问题就是在线二分类问题,当 $Y=\{1,2,...,c\}$时,问题就是多分类问题,而当 $Y \in \mathbb{R}$ 时,则是在线回归问题.针对不同的问题,可以使用不同的损失函数,例如二分类问题中经常使用 0-1 损失、铰链损失、逻辑损失[31]等,而回归问题中常使用最小二乘损失和ε不敏感损失等.不同的损失函数对预测的不准确程度的惩罚不一样,常见的损失函数的表达式见表 1.损失函数的数学性质,像凸性、强凸性、利普希茨连续性以及强光滑性等,会影响算法的收敛性以及收敛速度.

Table 1 Expressions of common loss functions

表 1 常见的损失函数表达式

损失函数	表达式
0-1 损失	如果 $\mathrm{sgn}(\boldsymbol{w}^\top \boldsymbol{x}) \neq y$, 则 $l(\boldsymbol{w};(\boldsymbol{x},y))=1$; 否则, $l(\boldsymbol{w};(\boldsymbol{x},y))=0$
逻辑损失	$l(\boldsymbol{w};(\boldsymbol{x},y))=\log(1+\mathrm{e}^{-y\boldsymbol{w}^\top x})$
铰链损失	$l(\boldsymbol{w};(\boldsymbol{x},y))=\max\{0,1-y\boldsymbol{w}^\top \boldsymbol{x}\}$
平方铰链损失	$l(\boldsymbol{w};(\boldsymbol{x},y))=\max\{0,1-y\boldsymbol{w}^\top \boldsymbol{x}\}^2$
多分类铰链损失	$l(\boldsymbol{w};(\boldsymbol{x},y))=\max\left\{0,1+\max_{c\neq y} \boldsymbol{w}_c{}^\top \boldsymbol{x}-\boldsymbol{w}_y{}^\top \boldsymbol{x}\right\}$, 其中,$\boldsymbol{w}_c$ 是第 c 类的决策模型
最小二乘损失	$l(\boldsymbol{w};(\boldsymbol{x},y))=(\boldsymbol{w}^\top \boldsymbol{x}-y)^2$
ε不敏感损失	$l(\boldsymbol{w};(\boldsymbol{x},y))=\max\{0,\lvert \boldsymbol{w}^\top \boldsymbol{x}-y\rvert -\varepsilon\}$

当决策域 D 是凸集且损失函数 $l(\boldsymbol{w},(x,y))$关于 $\boldsymbol{w}$ 是凸函数时,在线学习问题就变成了在线凸优化问题[32].在线凸优化在在线学习中占有举足轻重的地位,它借助凸分析工具,可以推导出高效的在线学习算法[7,33].

1.2 性能评价方法

在线学习使用悔恨(regret)来衡量算法的性能.记 $\boldsymbol{w}^*$是假设空间 D 中的某个固定的最优决策模型:

$$\boldsymbol{w}^* = \arg\min_{\boldsymbol{w}\in D}\sum_{t=1}^{T} l(\boldsymbol{w};(\boldsymbol{x}_t,y_t)),$$

其中,T 是预测游戏的轮数.悔恨度量算法对没有采用事后看来的最优模型 $\boldsymbol{w}^*$ 的后悔程度,形式上,悔恨定义为

$$R_T = \sum_{t=1}^{T} l(\boldsymbol{w}_t;(\boldsymbol{x}_t,y_t)) - \sum_{t=1}^{T} l(\boldsymbol{w}^*;(\boldsymbol{x}_t,y_t)).$$

在线学习算法的目标就是最小化在 T 轮预测游戏中的悔恨.特别地,低悔恨定义为悔恨随着学习次数的增加呈现次线性增长,即 $\lim_{T\to\infty} R_T / T = 0$, 这表明,算法和最优模型 $\boldsymbol{w}^*$之间的平均累积损失之差随着学习次数 T 的增加逐渐趋近于 0,因此意味着算法最终学到的模型可以收敛到 $\boldsymbol{w}^*$.通常我们追求低悔恨的算法,且主要关心算法在最坏情况下的悔恨上界[7].悔恨是确定性在线学习算法的性能评价方法,对于随机型算法,即算法选择决策模型时采用某种随机化策略,可以使用期望悔恨来评价性能,此时希望算法至少取得次线性的期望悔恨上界,且该界以高概率成立.

2 一般流数据分类研究现状

一般流数据是最简单的流数据特点,正因为如此,该研究领域得到快速发展,涌现出丰富的算法,对所有算法进行描述是困难的,这里仅介绍一些代表性的方法.图 2 给出本节的算法分类框架,根据环境在每次学习过程中所提供的问题和答案的完整性,在线学习可以分为完全信息和部分信息下的学习.在完全信息设置情况下,算法知道完整的问题和答案的信息,而在部分信息设置情况下,算法只知道部分的答案信息(Bandit 反馈)、或者只能收到部分问题上的答案(在线主动学习)、或者只能观察到问题表示的部分特征向量(部分属性).接下来,我们分别对两种设置下的学习算法进行介绍.

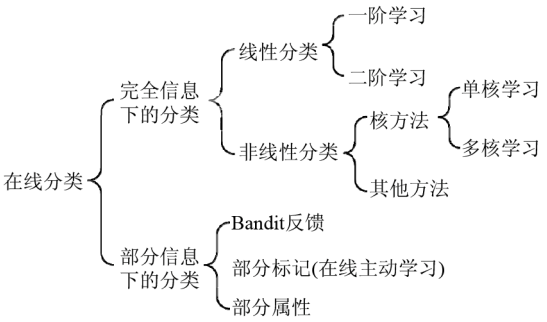

Fig.2 Taxonomy of online learning algorithms for general streaming data

图 2 一般流数据上的在线学习算法分类

2.1 完全信息下的在线线性分类

在线线性学习旨在学习一个线性预测器.线性方法尽管简单,但在文档类型的数据上却可以提供极具竞争力的结果[34].存在的线性分类算法可以分为一阶和二阶方法:一阶方法在学习的过程中仅使用瞬时损失函数的函数值和次梯度信息;二阶方法除了使用一阶信息外,还使用或近似使用瞬时损失函数的二阶导数信息,即 Hessian 矩阵信息.与一阶方法相比,二阶方法的时空复杂度较高,但收敛速度更快,因此在保持好的收敛速度的前提下,改进二阶算法的时空复杂度可以有不错的研究前景.

(1) 一阶方法

在线梯度下降(online gradient descent,简称 OGD)[32]或许是最简单也是最流行的一阶方法,它在学习的每一轮,总是沿着瞬时损失函数的负梯度方向修正模型,然后再将修正后的模型投影到可行域内.Zinkevich 等人证明,使用学习步长 $\eta_t=\Theta(1/\sqrt{t})$,对于任意可微且梯度的范数有界的凸瞬时损失函数序列,在 T 轮学习中,OGD 可以取得 $O(\sqrt{T})$ 的悔恨,且这个界是紧的,不能再改进.

被动主动算法(passive aggressive learning,简称 PA)[35]在学习过程中考虑预测的置信水平,即样本到当前决策边界的间距,使用该信息来辅助模型更新.具体地,算法在每轮学习中在决策空间中挑选一个新模型 $\boldsymbol{w}_{t+1}$,使得 $\boldsymbol{w}_{t+1}$ 尽可能接近当前模型 $\boldsymbol{w}_t$,且对当前样本 $\boldsymbol{x}_t$ 分类能取得合适的置信水平.理论上,PA 算法具有与感知机算法相当的误差界,但从实验结果上看,PA 算法经常优于感知机.本质上,PA 与 OGD 都是沿着瞬时损失函数的负梯度方向更新模型,只是 PA 采用更复杂的学习步长:每个样本上的学习步长与该样本被分类的置信水平相关.

Pegasos 算法[31]使用 OGD 方法来求解 ℓ_2 范数正则化的 SVM 规划问题,并赋予 OGD 方法更先进的步长调度方式 $\eta_t=1/(\lambda t)$,其中,λ 是正则化系数.由于 ℓ_2 范数正则化的铰链损失函数是强凸的,在新步长调度方式下,Pegasos 可以取得 $O(\ln T)$的悔恨界,这个界相比于 $O(\sqrt{T})$ 是一个大的提升.

不同于 OGD 和 Pegasos 使用预定义的步长调度,多步长梯度算法(multiple eta gradient,简称 MetaGrad)[16]同时考虑多个学习步长,每个步长产生一个相应的子模型,算法在每次学习过程中,对子模型进行倾斜指数加权平均得到主模型,子模型的权重正比于其在数据上的经验性能.MetaGrad 的优势在于适用于丰富的损失函数类,包括指数凹函数、强凸函数、一般的凸函数以及梯度满足 Bernstein 条件的随机函数,在这些函数类上,MetaGrad 至少取得 $O(\sqrt{T})$ 的悔恨界,很多情况下可以取得 $O(\ln T)$的悔恨界.

(2) 二阶方法

在线牛顿方法(online Newton step,简称 ONS)[7,36]是最具代表性的二阶方法,它并不直接使用瞬时损失的 Hessian 信息,而是利用瞬时损失的梯度信息近似表示 Hessian 矩阵,这种近似表示对于指数凹函数是可行的.具体地,ONS 在每轮学习中,朝着 Hessian 近似矩阵的逆乘以梯度的方向来调整当前模型,然后根据 Hessian 近似矩阵所定义的范数将调整后的模型投影到可行域中.从只利用梯度信息这方面来考虑,ONS 实际上是一阶方法.对于指数凹的损失函数,ONS 可以取得 $O(\ln T)$的悔恨界.

尽管 ONS 收敛速度快,但在每轮学习中需要花费 $O(d^2)$的时间和空间代价,其中,d 是模型的维度.为了解决 ONS 平方的时空代价问题,概要牛顿方法(sketched online Newton,简称 SON)[15]被提了出来,该方法使用少量仔细选择的方向(称为概要)来逼近二阶信息.通过这种方法,SON 每轮学习需要的时间代价约减为 $O(d)$,空间代价正比于概要的大小.

置信度加权学习(confidence-weighted learning,简称 CW)[37−39]是另一种二阶算法,它是 PA 算法的拓展.它假设线性分类模型服从均值向量为$\boldsymbol{\mu}$、协方差矩阵为Σ的高斯分布 $\boldsymbol{N}(\boldsymbol{\mu},\Sigma)$,算法使用$\boldsymbol{\mu}$来预测样本类标.具体地,在每轮学习中,CW 寻找一个新分布 $\boldsymbol{N}(\boldsymbol{\mu}_{t+1},\Sigma_{t+1})$,使得对当前样本正确分类的概率大于预定义的阈值$\theta\in(0.5,1]$,在满足该约束的前提下算法尽量保持新分布与当前分布 $\boldsymbol{N}(\boldsymbol{\mu}_t,\Sigma_t)$之间的 KL 散度最小.

算法 CW 总是强迫当前样本被正确分类,这个约束使得它极容易受到噪声数据的影响而产生过拟合,并且以概率形式表示的约束条件使得 CW 较难拓展到其他类型的学习任务中.于是,自适应权重正则化(adaptive regularization of weights,简称 AROW)[40]方法被提出来以解决上述两个问题,它将 CW 算法中的约束放松,然后转变为正则化项加到 KL 散度函数中,使得每次学习中算法只需求解一个无约束优化问题,这样做不仅提高了对噪声的鲁棒性,也容易泛化到其他任务中.

软置信度加权学习(soft confidence-weighted learning,简称 SCW)[41]算法也被提出来以解决 CW 算法的问题,但是与 AROW 算法不同,SCW 算法利用软间距 SVM 的思想,自适应地为不同的样本指派合适的置信度(间距),实验结果表明,多数情况下,该方法比 AROW 准确率更高且更高效.

2.2 完全信息下的在线非线性分类

非线性学习旨在学习一个非线性预测器来解决线性不可分问题.非线性分类方法可以分为核方法和非核的其他方法.核方法主要是基于 SVM 的方法,而非核方法中绝大多数是基于决策树的方法.两类方法不同的学习机制决定它们在不同特点的数据上各自具有优势,因此,当前对两类方法的研究呈现出平行发展的趋势.

(1) 核方法

核方法可以分为单核和多核方法.单核方法[31,42]使用一个预定义的核函数,通过该核函数导出的非线性映射,将原始空间的样本映射到一个新的更高维的特征空间中,然后在新特征空间中学习一个线性预测器.通常由于核函数导出的非线性映射不能显示表示,学习模型不能被表示成新特征空间中的向量,只能通过存储一些原始样本(称为支持向量)以及它们的贡献系数来间接表示.单核方法会遭遇“核诅咒”问题[42,43],即存储模型所需的支持向量的数目会随着处理数据量的增加呈现无限增长的趋势,从而导致耗尽可利用的内存资源,对测试样本的响应时间也无限延长.为提高核方法的可扩展性,发展出基于预算维护的方法[43−47]、基于核函数近似表示的方法[19]和基于核心集表示的方法[21].

基于预算维护的方法在学习过程中总是保持支持向量的数目上界于一个预定义的预算常数,每当支持向量的数目超过这个预算,预算维护步骤就被触发.常见的预算维护策略包括:删除、投影和合并.删除策略每次寻找一个支持向量进行删除,目前的删除策略包括删除最老的支持向量[44]、随机删除[45,47]以及删除对当前模型影响最小的支持向量[43]等.删除策略虽然简单、高效,但是往往不能取得令人满意的性能.投影策略旨在寻找一个支持向量,使得该支持向量用剩余的支持向量线性表示的投影误差最小.投影策略主要包括两步:(1) 确定要投影的支持向量;(2) 选择投影基.对第 1 步的处理包括精细搜索和简化搜索.精细搜索考虑对所有支持向量进行投影,从中选取投影误差最小的那个投影[46],这种策略显然太耗时,而简化搜索则选择一个对模型影响最小的支持向量,只对该支持向量进行投影[43],以此减少比较耗时的投影运算的次数.第 2 步投影基的选择包括使用投影支持向量的 k 近邻集[46],使用除投影支持向量外的所有支持向量[43],以及使用与投影支持向量同类的支持向量[48]等.合并策略[43,49]的主要思想是将两个支持向量合并为一个新创建的支持向量,使得合并误差最小化.投影和合并的时间代价比删除策略要大,但却能取得更好的性能.

基于核函数近似表示的方法[19]旨在找到一个可以近似表示给定核函数 $K(\cdot)$的显示特征映射$\varphi(\cdot)$,使得$\varphi(\boldsymbol{x}_1)^{\top}\varphi(\boldsymbol{x}_2)\approx K(\boldsymbol{x}_1,\boldsymbol{x}_2)$,在学习过程中,通过显示映射$\varphi(\cdot)$将原始空间中的样本映射到一个新的特征空间,在新空间中学习的线性模型可以直接通过向量来表示,无需保存原空间中的支持向量.具体地,Hoi 等人[19]提出两种近似方法,第 1 种方法适用于旋转不变核,在学习开始前,算法通过采样得到给定核函数的一组随机傅里叶变换组件,接下来,在每次学习过程中,当新样本到来时,通过傅里叶变换组件计算得到原始样本的新的特征表示,然后使用 OGD 来学习线性分类器;第 2 种是使用 Nyström 方法来近似核矩阵,根据近似核矩阵,构造出样本的新的特征表示方法,再使用 OGD 进行线性学习.相比于预算维护的方法,核函数近似表示法本质上是在特征映射后的新空间中进行线性学习,因此学习效率得到显著提升.

基于核心集表示的方法[21]首先构造出能够覆盖整个输入数据空间的δ覆盖核心集,然后,每当一个新的样本到来时,通过伯努利随机采样决定是将当前样本用核心集中的样本近似表示,还是直接将其加入到支持向量集中.因此,模型的大小就等于核心集的大小加上支持向量集的大小.根据伯努利采样方法是否统计上独立或相关于当前的预测模型 $\boldsymbol{w}_t$ 和样本(x_t,y_t),算法可取得不同的悔恨性能.特别地,当伯努利采样方法独立于 $\boldsymbol{w}_t$ 和(x_t,y_t)时,通过适当地控制采样参数可以证明算法所保存的模型大小是有界的.尽管该方法能在线地构造输入数据空间的核心集,但是对覆盖参数δ的恰当估计需要提前遍历所有输入数据,因此,对数据的总遍历次数超过 1 次,不是严格意义上的在线算法.

在线多核学习[50−52]以在线的方式从一个预定义的核函数集合中同时学习多个核分类器及其最优线性组合.Jin 等人[50]在 2010 年首次从理论层面探索在线多核分类问题,提出一组多核学习算法,这些算法均使用感知机算法为每个核函数学习一个核分类器,并利用专家学习算法 Hedge 来学习核分类器的组合系数,算法的差别在于更新核分类器的组合系数时利用不同的信息,例如误分类次数或误分类程度,且每轮学习中选择不同集合

的核分类器进行更新,例如选择所有核分类器或者随机选择一部分核分类器.2012 年,Hoi 等人[51]探索了高效的多核学习方法,为降低计算代价,提出预测时以及对核分类器进行更新时,都只考虑当前核分类器的一个子集.像单核方法一样,多核方法在学习每个核分类器时,同样会遭遇“核诅咒”问题.鉴于此,Lu 等人[53]在 2015 年考虑将预算维护和多核学习结合起来,提出预算多核学习方法,以提高算法的可扩展性.相比于单核学习,多核学习的研究工作还不够丰富,研究方法也很局限,未来有待继续探索.

(2) 其他方法

其他非线性的方法主要是基于决策树的算法[54−63],对决策树算法的研究一直是流数据分类领域中比较活跃的方向.在线决策树归纳算法设计的关键是,如何确定选择划分属性时所需的最少样本数,使得用少量样本所选出的划分属性以极高的概率等于使用整个流数据的样本所选出的划分属性,从而保证在线学习所学出的决策树渐进地收敛于一个批量学习算法所输出的决策树.为了达到这个目标,大量的决策树算法已被提出,其中,一种流行的增量决策树算法是 Domingos 等人提出的 Hoeffding 树[54],又称为快速决策树(very fast decision tree,简称 VFDT).Hoeffding 树算法初始时建立一棵仅包含一个叶子节点(即树根)的树,然后递归地将叶子节点替换为决策节点来生长树,每个决策节点联系着一个属性测试,每个叶子节点存储相关属性值的统计信息,用以评估某个属性划分的好坏;每当新样本到来时,对样本的属性值进行一系列的属性测试,将样本遍历到一个叶子节点,然后更新该叶子节点上的统计信息,最后考虑在该叶子节点上所有可能的属性划分,如果有足够的统计信息支持某个属性划分,就将该叶子节点替换为决策节点,根据所选的属性对原叶子节点处的样本进行划分.Hoeffding 树最大的创新在于使用 Hoeffding 界确定选择划分属性时所需的样本数.

原始的 Hoeffding 树只能处理离散属性,对测试数据遍历到叶子节点后使用多数表决的方式进行分类,Gama 等人[56]在 2003 年对 Hoeffding 树进行了拓展,增加了其处理连续属性的能力,同时在叶子节点处考虑使用朴素贝叶斯分类器进行分类.同年,Jin 等人[55]提出一种数值区间裁剪方法以高效地处理连续属性,并充分利用信息熵或基尼指数函数的特性来减少选择划分属性时所需的样本数.Holmes 等人[57]在 2005 年又将在叶子节点处的预测改进为自适应地在多数表决和使用朴素贝叶斯分类之间进行选择,以提高 Hoeffding 树处理噪声和复杂问题的能力.Hoeffding 树总是选择当前信息增益最大的属性进行划分,为了进一步改进其性能,Pfahringer 等人[58]在 2007 年探索了 Hoeffding 选项树,允许树中存在选项节点,在选项节点上可以选择多个属性分别进行划分,因此每个样本从树根节点到叶子节点存在多条路径,对样本的分类采用加权多数投票.Liang 等人[63]在 2015 年又将 Hoeffding 树的学习拓展到处理不确定流数据的应用中.上述算法都是单分类器算法,鉴于 Hoeffding 树属于不稳定算法,即训练数据的微小变化就会导致所生成的树存在极大的差异,很多工作开始考虑 Hoeffding 树集成算法[64−66]以进一步提高算法的泛化性能,这些工作通过不同的方法促进多样性,例如在线 bagging、随机投影以及控制树的大小等.

早期增量决策树算法研究主要集中于改进 Hoeffding 树的泛化性能或对 Hoeffding 树算法进行拓展.近 5 年来,Rutkowski 等人[60−62,67,68]探索并设计了一些与 Hoeffding 树算法具有显著差别的新算法.在 2013 年,Rutkowski 等人[60]认为直接将 Hoeffding 界应用到属性划分度量函数(如信息增益或基尼指数)上的做法有问题,违反了 Hoeffding 界使用的前提条件,因此提出 McDiarmid 树,用 McDiarmid 界代替 Hoeffding 界来确定选择划分属性时所需的样本数.使用 McDiarmid 界,算法做出属性划分决策往往需要较多的样本,导致树生长得较慢,影响树的性能,因此,Rutkowski 等人[61]在 2014 年提出使用多元德尔塔方法,结合泰勒定理和高斯分布的特性重新分析该问题,并推导出高斯决策树(Gaussian decision tree,简称 GDT)算法,该算法可以极大地约简选择属性划分所需的样本数.GDT 只适用于二分类问题,Rutkowski 等人[62]又将 GDT 的思想拓展到 CART 决策树算法,使得可以增量学习 CART 决策树,并处理多分类问题.2015 年,Rutkowski 等人[67]探索出一种新的属性划分不纯度度量函数;2018 年,他们又提出几个新的属性划分标准[68].虽然在线决策树算法的研究工作比较丰富,但目前缺乏对现有方法的综合性调查和全面对比分析的工作.

2.3 部分信息下的在线分类算法

完全信息设置中经常假设所有数据类标的完整信息都可以获取,且表示数据的特征向量的获取代价较小,

然而,在很多实际应用中,这样的假设并不成立,因此一些工作开始研究部分信息或弱信息下的在线分类算法.相比于完全信息的设置,部分信息设置下,算法可利用的信息受限,学习也更具有挑战性.

(1) bandit 反馈的在线多分类

在具有 bandit 反馈的多分类任务中[69−74],算法在每轮学习中被要求从 k 个类标中选出一个作为当前实例的预测类标,在它提交了预测后,环境并不揭示实例真正的类标,而只是告诉学习算法它的预测是否正确.这种仅能获取部分反馈信息的分类任务自然地起源于很多真实的互联网应用,例如在在线推荐应用中,系统仅能获取所推荐商品的反馈信息,即顾客买还是没买所推荐的商品,而无法得到未推荐商品上的反馈信息.

2008 年,Kakade 等人[69]首先对上述问题进行研究,提出 Banditron 算法,该算法可以看作是全反馈的感知机算法在 bandit 反馈下的变种.具体地,为处理部分反馈的挑战,Banditron 算法每次预测时,以概率 $\gamma\in(0,0.5)$ 随机选择一个类标(探索步骤),以概率 $1-\gamma$ 选择当前分类器的预测类标(利用步骤),其中,参数 γ 用于控制探索和利用的权衡.Banditron 算法的更新公式在期望上等于感知机算法的更新公式.相比于最优的离线线性分类器的累积铰链损失,Banditron 算法期望的错误预测数是 $O(T^{2/3})$.Banditron 算法对权衡参数 γ 比较敏感,因此,Valizadegan 等人[70]在 2011 年针对该问题提出 3 个利用算法的在线性能来辅助调节参数 γ 的自适应策略,实验结果表明了他们所提策略的有效性.

2009 年,Chen 等人[71]从不同的角度来考虑 bandit 反馈的多分类问题,所提算法在预测错时,并不对当前样本的类标进行探索(即不包括探索步骤),但却充分利用“当前样本不属于哪一类”的部分反馈信息来更新模型.具体地,首先利用基于边界的一对多(one-vs-rest)约简方法[75]将一个多分类问题转换为多个二分类问题,再使用 PA 算法[35]作为二分类的学习算法来求解每个二分类问题.通过这种一对多的约简方法,部分反馈信息通过更新相应的二分类器得到充分利用.Chen 等人为所提算法提供了相对误差界分析,并通过实验验证该算法比 Banditron 表现得更好.

2011 年,Hazan 等人[72]考虑使用耦合 softmax 预测的对数损失函数来研究该问题,提出 Newtron 算法.该算法在预测时,根据预测模型计算当前实例属于每类的概率,然后以概率 $1-\gamma$ 利用该概率分布选择一个类标,以概率 γ 均匀且随机地选择一个类标,以此得到算法的输出类标;算法基于追随近似领袖法(follow the approximated leader,简称 FTAL)[36]的思想进行模型更新,每次迭代中,首先构造对数损失的近似损失函数,然后利用追随正则化领袖法获得下一轮的预测模型.算法的对数损失函数有一个可以控制函数的指数凹性质的参数,根据该参数的不同取值,Newtron 算法的悔恨上界可以从 $O(\log T)$ 变化到 $O(T^{2/3})$.Newtron 算法的悔恨界仅适用于对数损失函数,一般的损失函数并不满足这个理论界.

2013 年,Crammer 等人[73]在假设类标的生成满足某个线性概率模型的前提下,进一步探索上述问题,提出基于置信度的 bandit(confidence based bandit,简称 Confidit)算法,该算法基于二阶感知机,利用上置信界(upper confidence bound,简称 UCB)[76]算法的思想将探索和利用步骤结合在一起,每次预测时,既考虑每一类分类器的预测值,又考虑预测的不确定性,选择预测值与预测的不确定性之和最大的那个类作为算法的预测类标.相比于最优的离线线性分类器的累积 0/1 损失,Confidit 算法的期望错误预测数是 $O(\sqrt{T}\log T)$.尽管 Confidit 与 Banditron 相比,其期望误差界得到极大的改进,但它的类标生成假设却限制了其适用范围.

2017 年,Beygelzimer 等人[74]考虑为一组更一般的损失函数设计高效的算法,该组损失函数是 0/1 损失的紧上界函数,由参数 $\eta\in[0,1]$ 控制,$\eta=0$ 对应于多分类铰链损失函数,$\eta=1$ 则对应于平方多分类铰链损失函数.Beygelzimer 等人为该组损失函数设计一种二阶 Banditron 算法(second order Banditron algorithm,简称 SOBA),算法预测时采用与 Banditron 算法[69]相同的类标探索和利用策略,模型更新则是基于二阶感知机算法.

理论分析的结果表明,针对这组损失函数,SOBA 算法都能近似取得 $O(\sqrt{T}/\eta)$ 的悔恨上界.

(2) 部分标记下的在线主动学习

在垃圾邮件在线分类系统[77]等应用中,可靠类标信息的获取需要花费较大的代价,为了减少标注的代价,应尽量减少对实例在线标记的次数.在这种设置下,算法在每次学习过程中,在对当前实例进行分类后,可以选择是否查询该实例的可靠类标信息,算法的目标是在受限的标记代价下尽可能地构建准确的预测模型,这称为在

线主动学习[78,79].典型地,要实现该目标,一种在线主动学习算法需要解决两个问题:第一,决定查询哪些实例的类标;第二,如何有效地利用已获取的类标信息来更新模型.

2004 年,Cesa-Bianchi[80]等人首先对该问题展开研究,分别将一阶和二阶感知机算法拓展到主动学习的场景中.具体地,他们的算法在每次做查询决定之前,首先从参数为$\delta/(\delta+m_i)$的伯努利分布中提取一个随机量 $Z_t\in\{0,1\}$,其中,δ是一个控制标记查询比例的预定义参数,$m_t=\left|\boldsymbol{w}_t^{\top}\boldsymbol{x}_t\right|$是当前实例 $\boldsymbol{x}_t$ 到当前决策模型 $\boldsymbol{w}_t$ 的距离,衡量预测的置信度.如果 Z_t=1,算法就查询 $\boldsymbol{x}_t$ 的类标,然后根据得到的类标,采用标准的一阶或二阶感知机的更新规则来更新当前的决策模型 $\boldsymbol{w}_t$;如果 Z_t=0,算法不作查询,也不更新模型.Cesa-Bianchi[80]等人证明,即使利用更少的类标,他们的算法在期望上仍然能够取得与标准的感知机算法相同的误差界.随后,采用与上述方法相同的类标查询机制,Cesa-Bianchi 等人[81]又将 Winnow 算法[82]拓展到主动学习的设置中,Zhao 等人[83]和 Lu 等人[84]也对 PA 算法[35]进行拓展.2016 年,Hao 等人[26,85]提出一种更为复杂的二阶主动学习算法,该算法与基于感知机的主动学习算法[80]的差异在于,其类标查询决策不仅考虑预测边界信息,也考虑二阶的置信度信息,且它采用类似于 AROW 算法[40]的更新规则来更新模型.除了上述基于统计和机器学习的工作以外,还存在一些基于演化软计算的工作[79],关于在线主动学习的更多工作可以参考 Lughofer 近来给出的综合性调查报告[86].

(3) 部分属性下的在线学习

在一些典型的流数据应用中,获取实例属性值的代价较大[87,88],为减少属性值采集的代价,可以对每个实例上可采集的属性值数目施加一个约束,算法在每次学习中仅能访问那些已采集到的属性值,但允许算法主动选择在下一个实例上需要采集哪些属性上的值.在这种部分属性值缺失的设置下,如何有效地进行在线预测是一个值得深入研究的问题,要解决该问题,典型地需要两步:第 1 步,决定观察当前实例上的哪些属性值.第 2 步,如何利用已获得的有限属性信息来有效地更新模型.目前对该问题的研究首先从回归领域展开.

2010 年,Cesa-Bianchi 等人[87]首先对部分属性下的线性预测问题进行调查研究,提出一种高效的线性回归算法,该算法是 Pegasos 回归算法在部分属性设置下的变种,其主要思想是利用可以获得的部分属性信息,构造平方损失函数的梯度 $\boldsymbol{g}$ 的无偏估计 $\hat{\boldsymbol{g}}$,然后在模型更新过程中,用 $\hat{\boldsymbol{g}}$ 来代替 $\boldsymbol{g}$.2012 年,Hazan 和 Koren[89]又将最常用的回归算法,包括岭回归、Lasso 和支持向量回归算法,分别扩展到部分属性设置下.具体地,对岭回归和 Lasso 的拓展也利用基于梯度无偏估计的思想,而对支持向量回归算法的拓展,由于其损失函数是非光滑的,利用一个具有解析性质的损失函数来近似表示原损失函数,然后再使用梯度无偏估计的方法进行学习.理论分析的结果表明,Hazan 和 Koren 所提出的算法相比于 Cesa-Bianchi 等人的算法,其达到指定精度所需要的样本数,即样本复杂度,得到极大的改进.

上述工作致力于解决回归问题,如何将其高效地拓展到分类问题常用的损失函数,仍然是一个开放的问题,但研究已经表明,梯度无偏估计的方法不能轻易地用在非光滑的损失函数上,例如铰链损失[89,90],因此,对于更一般的损失函数,新的高效的学习方法仍有待进一步探索.

3 完全信息下的高维流数据分类研究现状

在很多应用领域中,表示数据的特征向量的维度很高,非稀疏学习算法在这种情况下可能会遭遇“维度诅咒”问题而产生过拟合.稀疏模型通过大量约减数据维度有助于构建可靠的分类模型,是维度诅咒问题典型的解决方案[91−96],因此,对高维流数据分类算法的研究集中在稀疏在线学习领域.由于部分信息设置下算法可利用的数据信息受限会进一步增加稀疏学习的难度,所以目前的研究主要是在完全信息设置下展开的.

在稀疏在线学习问题中,一个线性预测器 $\boldsymbol{w}$ 的稀疏性被定义为 $\boldsymbol{w}$ 中非零元素的个数,也称为 $\boldsymbol{w}$ 的 ℓ_0 伪范数,记为 $\|\boldsymbol{w}\|_0$. 但是,从所有满足给定的 ℓ_0 伪范数约束的线性预测器中找到一个经验风险最小的预测器是 NP 难问题[97].为避免这一难题,已有两类方法被提了出来.第 1 类方法用凸的 ℓ_1 范数约束或 ℓ_1 范数正则化来代替非凸的 ℓ_0 伪范数约束;第 2 类方法基于 ℓ_0 截断方法,典型地在每次学习中分为两步:第 1 步,求解一个无稀疏约束的凸优化问题.第 2 步,在第 1 步得到的解附近寻找一个满足给定的 ℓ_0 伪范数约束的新解.表 2 列出近来这些方法重要的研究工作.这些方法在模型更新策略方面,都是从简单的一阶更新向着更先进的二阶更新发展.在稀疏策略方

面,基于 ℓ_1 范数约束/正则化的方法通过精细地调整与稀疏性相关的参数来促进模型的稀疏性,不能事先量化所获得模型的稀疏性,也不能保证在在线学习的每次迭代中模型的 ℓ_0 伪范数上界于一个固定常数,而基于 ℓ_0 截断的方法则可以满足此要求.

Table 2 Recent work in sparse online learning

表 2 稀疏在线学习最近的工作

稀疏策略	参考文献/方法
ℓ_1 范数约束	[98,99]
ℓ_1 范数正则化	FOBOS[100], TrunGrad[101], ℓ_1-RDA[102], CMD[103], ℓ_1-ARDA[104], Ada-FOBOS[104], SOL[105]
ℓ_0 截断	OFS[24], SALC[106], SOFS[107], BARDA[108], BAMD[108]

3.1 基于ℓ_1范数约束/正则化的方法

基于 ℓ_1 范数约束的方法使用高效的投影算法将梯度下降更新后得到的解投影在一个给定半径的 ℓ_1 球上获得稀疏性[98,99].基于 ℓ_1 范数正则化的方法旨在通过最小化 ℓ_1 正则化的损失函数来获得稀疏模型.

Duchi 等人[100]在 2009 年提出前向后向分割(forward backward splitting,简称 FOBOS)算法,该算法在每次迭代过程中首先执行一个梯度下降更新,得到一个中间解,然后在中间解附近找到一个低 ℓ_1 范数复杂度的新解.得到新解的过程只需通过对中间解中低于某个阈值的系数进行截断.对于 ℓ_1 正则化的凸损失函数,FOBOS 算法可以取得 $O(\sqrt{T})$ 的 ℓ_1 正则化悔恨界.

FOBOS 算法每次迭代中直接对中间解进行截断的操作过于激进,会阻碍模型在稀疏属性上的有效更新.相比之下,Langford 等人[101]提出一种更温和的截断梯度(truncated gradient,简称 TrunGrad)方法,该方法每隔 k 次迭代才对模型进行稀疏化操作,将模型的系数以温和的方式逐渐地缩小为 0.理论上,TrunGrad 可以取得与 FOBOS 相当的正则化悔恨界.

2010 年,Xiao[102]将对偶平均(dual averaging,简称 DA)算法拓展到正则化损失函数的情形中,提出正则化对偶平均(regularized DA,简称 RDA)算法;同年,Duchi 等人[103]也对在线镜像下降(mirror descent,简称 MD)算法进行拓展,提出组合目标镜像下降(composite-objective MD,简称 CMD)算法.RDA 和 CMD 在每轮的优化过程中,均未对正则化项加以线性化,而是直接将其加到优化目标中,以帮助取得稀疏解.对于 ℓ_1 范数正则化的凸损失函数序列,经过 T 轮学习,RDA 和 CMD 算法都取得 $O(\sqrt{T})$ 的 ℓ_1 正则化悔恨界.FOBOS 被证明是 CMD 算法的特例.

RDA 和 CMD 算法都探索了如何利用损失函数的正则化结构来提高模型的稀疏性,但它们都未充分利用学习早期迭代中的次梯度信息,该信息有助于识别有辨别力但出现次数少的特征.因此,Duchi 等人[104]在 2011 年分别对 RDA 和 CMD 进行改进,提出自适应的 RDA(adaptive RDA,简称 ARDA)和自适应的 CMD(adaptive CMD,简称 ACMD),在学习过程中将早期迭代中的次梯度信息保存在一个矩阵 $\boldsymbol{G}$ 中,根据 $\boldsymbol{G}$ 执行更有信息量的基于梯度的学习.两种算法在模型更新的过程中需要计算矩阵 $\boldsymbol{G}$ 的逆和平方根,这个操作需要花费较大的时空代价,Duchi 等人因此分别提供一种简化版的算法,只需维护 $\boldsymbol{G}$ 的对角矩阵,从而极大地约减了算法的时空复杂度.

对于完全矩阵和对角矩阵的算法,Duchi 等人都证明算法可以取得与 $O(\sqrt{T})$ 相当的正则化悔恨界.多个实验调查结果表明,由 ARDA 和 ACMD 分别导出的 ℓ_1-ARDA 和 Ada-FOBOS 算法比非自适应的 ℓ_1-RDA 和 FOBOS 算法能够取得更好的测试性能,且解的稀疏水平也更高.

2014 年,Wang 人[105]提出一个泛化的稀疏在线分类框架,由该框架可以导出一些存在的一阶稀疏在线学习算法和一种新的二阶稀疏学习算法.然而,王等人并没有为该框架提供详细的理论分析,只是通过实验验证所导出算法的有效性.

3.2 基于ℓ_0截断的方法

Jin 等人[24]在 2014 年提出基于截断的在线特征选择(online feature selection,简称 OFS)算法.算法在每次迭代中,首先将梯度下降更新后得到的解投影到一个 ℓ_2 范数球,使得解中绝大部分元素集中在最大的值,然后截断最小的元素,仅保留 B 个最大的元素,其中,B 是一个预定义常数.Jin 等人提供 OFS 的误差界来证明其有效性.

Zhai 等人[106]在 2018 年也基于梯度下降技术提出一种稀疏近似线性分类(sparse approximated linear classifier,简称 SALC)算法.算法在每次迭代过程中,在梯度下降更新后得到的解 $\boldsymbol{z}_{t+1}$ 附近寻找一个最稀疏(ℓ_0 伪范数最小)的向量 $\boldsymbol{w}_{t+1}$,这一步可以通过对 $\boldsymbol{z}_{t+1}$ 的元素按照绝对值从小到大排列后截断若干最小的元素来实现.SALC 有一个稀疏参数ε可以控制 $\boldsymbol{w}_{t+1}$ 与 $\boldsymbol{z}_{t+1}$ 之间的 ℓ_2 范数距离,从而决定 $\boldsymbol{w}_{t+1}$ 的 ℓ_0 稀疏性.Zhai 等人证明通过设置合适的学习步长,SALC 能够取得 $O(\sqrt{T})$ 的悔恨界.

OFS 和 SALC 均是基于一阶算法 OGD 设计的截断方法,没有利用二阶信息来辅助学习.Wu 等人[107]在 2017 年基于二阶算法 AROW 探索了截断方法,提出二阶在线特征选择(second-order OFS,简称 SOFS)算法.为了降低 AROW 的空间代价,SOFS 使用对角版本的 AROW 算法,并在每次迭代中截断模型中具有最小置信度的元素.然而,Wu 等人并没有为 SOFS 提供任何理论保证.

Zhai 等人[108]在 2018 年又基于二阶的自适应次梯度算法设计了新颖的在线特征算法,提出预算的 ARDA (budgeted ARDA,简称 BARDA)和预算的自适应 MD(budgeted adaptive MD,简称 BAMD).两种算法在评估特征的相关性或重要性时,综合考虑特征在当前预测器中的权重和特征在历史数据流中出现的频率.Zhai 等人为两种算法提供详细的悔恨分析,并通过实验证明所提算法相比于 OFS 和 SOFS 的优越性.

4 完全信息下的演化流数据分类研究现状

当前演化流数据分类的工作仍然主要集中在完全信息设置下.演化流数据是指流中数据不再满足独立同分布假设,数据分布会随着时间发生演化,即存在“概念漂移”现象.形式上,概念漂移指的是输入变量到类标变量之间的函数关系会随着时间发生难以预料的变化[109,110].概念漂移影响决策制定,为处理好它,学习算法应具备遗忘能力,能够随着时间逐步忘记旧的概念,使得所学模型与最新的概念保持一致.演化流数据分类旨在捕获流数据中输入变量和类标变量之间最新的函数关系.

存在的演化流数据分类算法可以分为单模型算法和多模型算法[111].单模型算法主要是传统批量学习算法的增量版本,且配备了特定的概念漂移处理机制.基于决策树[112,113]、k 近邻法[114]及 SVM[43]等单模型算法都已经被探索过.在单模型算法中,为将稳态流数据上比较流行的 VFDT 树算法应用于演化流数据,概念自适应的快速决策树(concept-adapting VFDT,简称 CVFDT)算法[115]被提了出来,它维护一个决策树总是与流中最近的 k 个样本上的概念保持一致.相比于多模型算法,单模型算法的时空开销小,但泛化性能较差.

多模型算法也称为自适应集成算法,在内存中维护多个增量模型做出联合预测.多模型方法可以删除过期的模型并创建最新的模型,借此来遗忘旧概念并适应最新概念,因此能够更灵活地处理概念漂移.目前已有的算法可以分为两类:第 1 类是基于数据块的集成[116−120],适用于流数据分批次到达的情形;第 2 类是在线集成,每次仅处理一个样本,样本处理完后无需对其进行存储和再访问.表 3 从以下几个方面总结了最近的在线集成分类算法:(1) 使用的组件分类器,即基分类器;(2) 组件多样性策略;(3) 组件评估策略;(4) 组件聚合策略;(5) 是否使用漂移检测机制;(6) 是否调整集成的组件成员结构,即创建新组件或删除过时的组件.

Table 3 A summary of online ensemble algorithms

表 3 在线集成算法总结

算法	组件分类器	多样性策略	组件评估策略	聚合策略	漂移检测	结构调整
OzaBag[121]	任意在线分类器	重采样	–	多数投票	–	×
DWM[122]	任意在线分类器	周期性创建新组件	指数递减	加权多数投票	×	√
ASHTBag[64]	Hoeffding 树	重采样+控制树大小	指数加权移动平均	加权多数投票	×	×
ADWINBag[64]	任意在线分类器	重采样	ADWIN[123]	多数投票	√	√
LevBag[124]	任意在线分类器	重采样	ADWIN	多数投票	√	√
DDD[125]	任意集成	不同的重采样参数	序列准确率	选择加权多数投票	√	×
OAUE[126]	任意在线分类器	周期性创建新组件	均方差	加权多数投票	×	√
EBPegasos[110]	BPegasos[43]	控制预测参数大小	ADWIN	加权多数投票	√	√

2005 年,Oza[121]提出一种在线版本的 bagging 算法,简称为 OzaBag.每当一个新实例来到,OzaBag 分别为每

个组件分类器从参数为 1 的泊松分布中采样得到一个自然数 k,然后让组件分类器学习当前实例 k 次,由于不同的组件分类器采样得到的 k 不同,因此每个样本被每个组件分类器学习的次数也不同,就好像每个组件分类器是运行在不同的实例集合上.这样做是为了模拟离线的 bootstrap 自助采样,因为在离线情况下,每个实例被选中的概率接近于参数为 1 的泊松分布.OzaBag 使用简单的多数投票进行分类.然而,由于缺乏结构调整,OzaBag 较难适应突然发生的概念漂移.

2007 年,Kolter 等人[122]提出 DWM(dynamic weighted majority)算法来处理概念漂移,该算法维护一个动态加权的组件池,其中的每个组件都有一个权重,每当组件做出错误的预测时,其权重就折扣$\beta\in(0,1)$倍,算法每处理完 p 个实例,就将权重小于给定阈值的组件删除,并根据集成算法的加权投票预测结果来决定是否创建新的组件.由于组件权重是呈指数递减的,DWM 容易受到噪声的影响而误删除有用的组件.

2009 年,Bifet 等人[64]为 Hoeffding 树专门设计了一种集成算法 ASHTBag(adaptive-size Hoeffding tree bagging),该算法使用不同大小的 Hoeffding 树作为组件分类器,同时利用 OzaBag 的重采样方法使得每棵树运行在不同的实例集合上,它也用指数加权移动平均来评估每棵树的误差,当加权投票预测时,每棵树的投票权重反比于它的平方误差.类似于 OzaBag,ASHTBag 也缺乏结构调整,较难处理突然发生的概念漂移.

Bifet 等人[64]又考虑在 OzaBag 算法中融入 ADWIN 漂移检测算法[123]来检测漂移和评价组件分类器的性能,提出 ADWINBag,该算法在检测到漂移后,会丢掉过时的组件,创建新的组件.

2010 年,Bifet 等人[124]又改进了 ADWINBag,提出 LevBag(leveraging bagging)算法,不再采用参数为 1 的泊松分布进行重采样,而采用参数为 6 的泊松分布.这增加了样本被学习的次数,同时也使得算法容易受到噪声数据的影响,因为噪声实例也倾向于多次被学习.

2012 年,Minku 等人[125]受到多样性对概念漂移影响的启发[127],提出 DDD(diversity for dealing with drifts)算法,它维护 4 个具有不同程度多样性的集成算法,每个集成使用不同参数的泊松分布进行重采样.漂移检测技术被用来监视流数据的状态,以便在不同的时期选择不同的集成来预测.每个集成的性能使用从上次检测到漂移开始再到当前为止的序列准确率[109,128]来评估.由于 DDD 是集成算法的再次集成,时间和空间代价较大.

2014 年,Brzezinski 等人[126]模拟基于数据块的集成算法的思想,提出 OAUE(online accuracy updating ensemble)算法,该算法周期性地创建和删除组件,使用组件分类器在最近的 d 个样本上预测的均方误差来评价其性能.但该算法很难选择合适的周期来执行组件创建和删除:一方面,大的周期有助于产生稳定的组件分类器,但会造成对概念漂移的反应较慢;另一方面,小的周期有助于对概念漂移做出快速反应,但产生的组件分类器的性能不够稳定,这个困境又称为稳定性和可塑性的困境.

2017 年,Zhai 等人[110]专门为高维演化流数据算法设计了一种集成算法 EBPegasos(ensemble BPegasos),该算法使用在线核 SVM 算法——BPegasos[43]作为组件分类器,通过控制 BPegasos 的预算参数来构建多样化的组件,并为每个组件配备 ADWIN 漂移检测器来监视和评估组件在最新数据上的性能,一旦检测到漂移,算法首先衡量漂移对当前组件分类性能造成的影响,对于影响严重的漂移,算法就逐渐删除旧的组件,创建新的组件,对于影响不大的漂移,算法则保持增量更新旧的组件,以这样的方式,算法试图在遗忘和利用旧概念之间取得好的平衡.

上面我们分析了几种典型的在线集成算法,对基于数据块的集成算法的对比分析可以参考许冠英等人[129]近来的综述.根据上述分析,为了能够及时地处理各种类型的概念漂移,近几年的算法倾向于在学习过程中采用某种漂移检测技术监视流数据的状态,使得这些算法的性能受制于漂移检测算法的性能.为了进一步改善演化流数据上分类算法的性能,未来可以向着开发出漏检率和误报率更低的鲁棒的漂移检测算法发展[130−133].

5 未来研究方向

大数据应用需求的显著增长推动了在线学习领域的快速发展,已涌现出更为丰富的方法.表 4 从流数据特点和在线学习特点两个维度上,总结了近年来流数据分类在不同设置下研究工作的丰富程度(用三角的个数来表示).

Table 4 The degree of richness of research on streaming data classification in different settings

表 4 不同场景设置下流数据分类工作的丰富程度

流数据特点	在线学习	
	完全信息	部分信息
一般	ΔΔΔΔΔ	ΔΔΔ
高维度	ΔΔΔ	Δ
演化(概念漂移)	ΔΔΔ	Δ

从表 4 中可以看到,目前面向流数据分类的在线学习研究主要集中于完全信息设置,部分信息设置下的研究工作则相对较少或缺乏.完全信息设置经常假设所有数据的类标(或反馈)的信息都可以及时获取,且表示数据的特征向量的获取代价较小,这样的假设无疑对流数据采集之后的特征表示和标记工作要求太高,在很多真实的流数据应用中往往很难满足.相比而言,部分信息设置可以降低对大量标记和大量数据特征的依赖和需求,缓解标记和表示流数据的压力,因而更符合真实的应用场景.因此我们认为,未来的研究应该转向更具挑战性的部分信息设置的在线学习.特别地,结合流数据的特点,图 3 展示了几个有前途的研究方向.

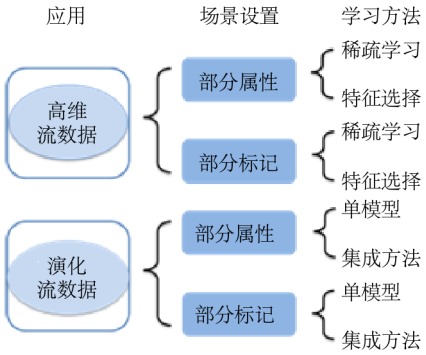

Fig.3 Future research directions for classification of high-dimensional or/and evolving streaming data

图 3 高维和演化流数据分类有前途的研究方向

(1) 高维流数据上部分属性的稀疏在线学习/在线特征选择

在该设置下,为了使算法收敛到一个好的稀疏模型,当算法在每次迭代中选择要观察的属性值集合时,一方面需要充分利用当前已有的信息来衡量哪些属性是最相关的,另一方面也要探索新的潜在重要的属性,算法需要在探索和利用之间进行合理的权衡,并根据获得的部分属性信息有效地更新学习模型,还要确保学到的模型满足给定的稀疏性约束.目前,Wang 等人[24]和 Foster 等人[134]已对该问题进行了一些初步探索,然而,为设计出更有效的算法,可以借鉴更先进的探索和利用的权衡策略,例如 UCB 算法[76]的思想等,并采用更先进的模型更新规则.

(2) 高维流数据上部分标记的稀疏在线学习/在线特征选择

在该设置下,为在受限的标记代价下构建尽可能准确的稀疏预测模型,算法需要谨慎挑选要标注的样本,同时充分利用已获得标记的样本信息来构建稀疏预测模型.尽管在线主动学习近年来得到越来越多的关注[26,79,84−86],但是面对高维流数据,如何将主动学习与稀疏学习或特征选择有效地结合起来,仍有待进一步探索.

(3) 演化流数据上部分属性的在线单模型/集成学习

在该设置下,算法选择要观察的属性时,应选择与当前概念最相关的属性集.这个问题的挑战在于,概念会发生漂移,导致与分类性能最相关的属性集也会随之发生变化.目前鲜有对该问题的研究,亟待填补空白.

(4) 演化流数据上部分标记的在线单模型/集成学习

在该设置下,当算法选择要标注的样本时,要保证所选样本能够尽可能地反映最新的数据分布情况.这个问题的挑战在于,概念漂移会导致数据的分布状况发生变化,从而要求算法能尽早感知到漂移发生并对样本选择策略做出调整.目前主动学习的研究集中在稳态流数据上,只有少量的工作可以处理有“概念漂移”的演化流数

据[135−137],因此对该方向有待深入调查研究.

6 总 结

互联网、电子商务、移动通信及物联网等应用领域中的数据往往以流的形式源源不断地生成,数据呈现出大规模、高速到达的特点,需要对数据进行实时处理和分析,而传统的批量处理的机器学习方法很难满足上述需求,相比之下,在线学习由于其在线更新的计算模式非常契合流数据的特点,有望成为流数据领域中的主流方法.本文从两个维度——流数据特点(即一般、高维、演化)和在线学习特点(即完全信息、部分信息),对现有的流数据上的在线分类算法进行了分类对比分析.在综述中发现,现有的在线分类方法主要集中在完全信息设置下,而在部分信息设置下的研究工作则较少,尤其是在高维和演化流数据上.导致该现象的其中一个原因是部分信息设置下,算法可利用的数据信息非常受限,例如部分实例的标记信息难以获取、部分实例的属性信息缺失等,从而导致学习难度的大幅度提高.考虑到部分信息设置比较符合真实的流数据应用场景,同时结合当前的流数据分类研究现状,给出未来在流数据分类领域中在线学习的几个有前途的研究方向.

References:

[1] Aggarwal CC. A survey of stream classification algorithms. In: Data Classification: Algorithms and Applications. CRC Press, 2014. 245–274.

[2] Krempl G, Zliobaite I, Brzezinski D, Hüllermeier E, Last M, Lemaire V, Noack T, Shaker A, Sievi S, Spiliopoulou M, Stefanowski J. Open challenges for data stream mining research. SIGKDD Explorations, 2014,16(1):1–10.

[3] Zhai TT. Online learning algorithms for classification of streaming data [Ph.D. Thesis]. Nanjing: Nanjing University, 2018 (in Chinese with English abstract).

[4] Vapnik V. An overview of statistical learning theory. IEEE Trans. on Neural Networks, 1999,10(5):988–999.

[5] Shalev-Shwartz S, Singer Y. Online learning: Theory, algorithms, and applications [Ph.D. Thesis]. Jerusalem: Hebrew University, 2007.

[6] Shalev-Shwartz S. Online learning and online convex optimization. Foundations and Trends in Machine Learning, 2012,4(2): 107–194.

[7] Hazan E. Introduction to online convex optimization. Foundations and Trends in Optimization, 2016,2(3/4):157–325.

[8] Cesa-Bianchi N, Conconi A, Gentile C. On the generalization ability of online learning algorithms. IEEE Trans. on Information Theory, 2004,50(9):2050–2057.

[9] Shalev-Shwartz S, Singer Y, Srebro N. Pegasos: Primal estimated sub-gradient solver for SVM. In: Proc. of the Int'l Conf. on Machine Learning (ICML 2007). 2007. 807–814.

[10] Zhang L, Yi J, Jin R, Lin M, He X. Online kernel learning with a near optimal sparsity bound. In: Proc. of the Int'l Conf. on Machine Learning (ICML 2013). 2013. 621–629.

[11] Daniely A, Gonen A, Shalev-Shwartz S. Strongly adaptive online learning. In: Proc. of the Int'l Conf. on Machine Learning (ICML 2015). 2015. 1405–1411.

[12] György A, Szepesvári C. Shifting regret, mirror descent, and matrices. In: Proc. of the Int'l Conf. on Machine Learning (ICML 2016). 2016. 2943–2951.

[13] Shamir O, Szlak L. Online learning with local permutations and delayed feedback. In: Proc. of the 34th Int'l Conf. on Machine Learning (ICML 2017). 2017. 3086–3094.

[14] Quanrud K, Khashabi D. Online learning with adversarial delays. In: Proc. of the Advances in Neural Information Processing Systems (NIPS 2015). 2015. 1270–1278.

[15] Luo H, Agarwal A, Cesa-Bianchi N, Langford J. Efficient second order online learning by sketching. In: Proc. of the Advances in Neural Information Processing Systems (NIPS 2016). 2016. 902–910.

[16] Erven T, Koolen WM. MetaGrad: Multiple learning rates in online learning. In: Proc. of the Advances in Neural Information Processing Systems (NIPS 2016). 2016. 3666–3674.

[17] Luo H, Wei CY, Zheng K. Efficient online portfolio with logarithmic regret. In: Proc. of the Advances in Neural Information Processing Systems (NIPS 2018). 2018. 8245–8255.

[18] Gillen S, Jung C, Kearns MJ, Roth A. Online learning with an unknown fairness metric. In: Proc. of the Advances in Neural Information Processing Systems (NIPS 2018). 2018. 2605–2614.

[19] Lu J, Hoi SCH, Wang J, Zhao P, Liu Z. Large scale online kernel learning. Journal of Machine Learning Research, 2016,17:47:1–47:43.

[20] Shi T, Zhu J. Online Bayesian passive-aggressive learning. Journal of Machine Learning Research, 2017,18:33:1–33:39.

[21] Le T, Nguyen TD, Nguyen V, Phung DQ. Approximation vector machines for large-scale online learning. Journal of Machine Learning Research, 2017,18:111:1–111:55.

[22] Lei Y, Shi L, Guo Z. Convergence of unregularized online learning algorithms. Journal of Machine Learning Research, 2017,18:171:1–171:33.

[23] Chaudhuri S, Tewari A. Online learning to rank with top-*k* feedback. Journal of Machine Learning Research, 2017,18:103:1–103:50.

[24] Wang J, Zhao P, Hoi SCH, Jin R. Online feature selection and its applications. IEEE Trans. on Knowledge and Data Engineering, 2014,26(3):698–710.

[25] Wang J, Wang M, Li P, Liu L, Zhao Z, Hu X, Wu X. Online feature selection with group structure analysis. IEEE Trans. on Knowledge and Data Engineering, 2015,27(11):3029–3041.

[26] Hao S, Lu J, Zhao P, Zhang C, Hoi SCH, Miao C. Second-order online active learning and its applications. IEEE Trans. on Knowledge and Data Engineering, 2018,30(7):1338–1351.

[27] Hoi SCH, Sahoo D, Lu J, Zhao P. Online learning: A comprehensive survey. CoRR, 2018, abs/1802.02871. http://arxiv.org/abs/1802.02871

[28] Li ZJ, Li YX, Wang F, He GL, Kuang L. Online learning algorithms for big data analytics: A survey. Journal of Comuter Research and Develoment, 2015,52(8):1707–1721 (in Chinese with English abstract).

[29] Pan ZS, Tang SQ, Qiu JY, Hu GY. Survey on online learning algorithms. Journal of Data Acquisition & Processing, 2016,31(6):1067–1082 (in Chinese with English abstract).

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

[31] Shalev-Shwartz S, Singer Y, Srebro N, Cotter A. Pegasos: Primal estimated sub-gradient solver for SVM. Mathematical Programming, 2011,127(1):3–30.

[32] Zinkevich M. Online convex programming and generalized infinitesimal gradient ascent. In: Proc. of the Int'l Conf. on Machine Learning (ICML). 2003. 928–936.

[33] Cesa-Bianchi N, Lugosi G. Prediction, Learning, and Games. New York: Cambridge University Press, 2006. 40–66.

[34] Yuan GX, Ho CH, Lin CJ. Recent advances of large-scale linear classification. Proc. of the IEEE, 2012,100(9):2584–2603.

[35] Crammer K, Dekel O, Keshet J, Shalev-Shwartz S, Singer Y. Online passive-aggressive algorithms. Journal of Machine Learning Research, 2006,7:551–585.

[36] Hazan E, Agarwal A, Kale S. Logarithmic regret algorithms for online convex optimization. Machine Learning, 2007,69(2/3):169–192.

[37] Dredze M, Crammer K, Pereira F. Confidence-weighted linear classification. In: Proc. of the Int'l Conf. on Machine Learning (ICML). 2008. 264–271.

[38] Crammer K, Dredze M, Pereira F. Exact convex confidence-weighted learning. In: Proc. of the Advances in Neural Information Processing Systems (NIPS 2008). 2008. 345–352.

[39] Crammer K, Dredze M, Pereira F. Confidence-weighted linear classification for text categorization. Journal of Machine Learning Research, 2012,13:1891–1926.

[40] Crammer K, Kulesza A, Dredze M. Adaptive regularization of weight vectors. Machine Learning, 2013,91(2):155–187.

[41] Wang J, Zhao P, Hoi SCH. Exact soft confidence-weighted learning. In: Proc. of the Int'l Conf. on Machine Learning (ICML). 2012.

[42] Kivinen J, Smola AJ, Williamson RC. Online learning with kernels. IEEE Trans. on Signal Processing, 2004,52(8):2165–2176.

[43] Wang Z, Crammer K, Vucetic S. Breaking the curse of kernelization: Budgeted stochastic gradient descent for large-scale SVM training. Journal of Machine Learning Research, 2012,13(1):3103−3131.

[44] Dekel O, Shalev-Shwartz S, Singer Y. The forgetron: A kernel-based perceptron on a fixed budget. In: Proc. of the Advances in Neural Information Processing Systems (NIPS 2005). 2005. 259−266.

[45] Cavallanti G, Cesa-Bianchi N, Gentile C. Tracking the best hyperplane with a simple budget perceptron. Machine Learning, 2007, 69(2/3):143−167.

[46] Wang Z, Vucetic S. Online passive-aggressive algorithms on a budget. In: Proc. of the 13th Int'l Conf. on Artificial Intelligence and Statistics (AISTATS 2010). 2010. 908−915.

[47] Zhao P, Wang J, Wu P, Jin R, Hoi SCH. Fast bounded online gradient descent algorithms for scalable kernel-based online learning. In: Proc. of the Int'l Conf. on Machine Learning (ICML). Edinburgh, 2012.

[48] Orabona F, Keshet J, Caputo B. Bounded kernel-based online learning. Journal of Machine Learning Research, 2009,10:2643−2666.

[49] Wang Z, Vucetic S. Twin vector machines for online learning on a budget. In: Proc. of the SIAM Int'l Conf. on Data Mining (SDM 2009). 2009. 906−917.

[50] Jin R, Hoi SCH, Yang T. Online multiple kernel learning: Algorithms and mistake bounds. In: Proc. of the 21st Int'l Conf. on Algorithmic Learning Theory (ALT 2010). 2010. 390−404.

[51] Hoi SCH, Jin R, Zhao P, Yang T. Online multiple kernel classification. Machine Learning, 2013,90(2):289−316.

[52] Diethe T, Girolami MA. Online learning with (multiple) kernels: A review. Neural Computation, 2013,25(3):567−625.

[53] Lu J, Hoi SCH, Sahoo D, Zhao P. Budget online multiple kernel learning. CoRR, 2015, abs/1511.04813.

[54] Domingos P, Hulten G. Mining high-speed data streams. In: Proc. of the 6th ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining (KDD 2000). 2000. 71−80.

[55] Jin R, Agrawal G. Efficient decision tree construction on streaming data. In: Proc. of the 9th ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining (KDD 2003). 2003. 571−576.

[56] Gama J, Rocha R, Medas P. Accurate decision trees for mining high-speed data streams. In: Proc. of the 9th ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining (KDD 2003). 2003. 523−528.

[57] Holmes G, Kirkby R, Pfahringer B. Stress-testing hoeffding trees. In: Proc. of the 9th European Conf. on Principles and Practice of Knowledge Discovery in Databases (PKDD 2005). 2005. 495−502.

[58] Pfahringer B, Holmes G, Kirkby R. New options for Hoeffding trees. In: Proc. of the 20th Australian Joint Conf. on Artificial Intelligence. 2007. 90−99.

[59] Hashemi S, Yang Y, Mirzamomen Z, Kangavari MR. Adapted one-versus-all decision trees for data stream classification. IEEE Trans. on Knowledge and Data Engineering, 2009,21(5):624−637.

[60] Rutkowski L, Pietruczuk L, Duda P, Jaworski M. Decision trees for mining data streams based on the McDiarmid's bound. IEEE Trans. on Knowledge and Data Engineering, 2013,25(6):1272−1279.

[61] Rutkowski L, Jaworski M, Pietruczuk L, Duda P. Decision trees for mining data streams based on the Gaussian approximation. IEEE Trans. on Knowledge and Data Engineering, 2014,26(1):108−119.

[62] Rutkowski L, Jaworski M, Pietruczuk L, Duda P. The CART decision tree for mining data streams. Information Sciences, 2014,266:1−15.

[63] Liang C, Zhang Y, Shi P, Hu Z. Learning accurate very fast decision trees from uncertain data streams. Int'l Journal of Systems Science, 2015,46(16):3032−3050.

[64] Bifet A, Holmes G, Pfahringer B, Kirkby R, Gavaldà R. New ensemble methods for evolving data streams. In: Proc. of the 15th ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining (KDD 2009). 2009. 139−148.

[65] Bifet A, Frank E, Holmes G, Pfahringer B. Accurate ensembles for data streams: Combining restricted Hoeffding trees using stacking. In: Proc. of the 2nd Asian Conf. on Machine Learning (ACML 2010). 2010. 225−240.

[66] Pham XC, Dang MT, Dinh SV, Hoang S, Nguyen TT, Liew AWC. Learning from data stream based on random projection and Hoeffding tree classifier. In: Proc. of the Int'l Conf. on Digital Image Computing: Techniques and Applications (DICTA 2017). 2017. 1−8.

[67] Rutkowski L, Jaworski M, Pietruczuk L, Duda P. A new method for data stream mining based on the misclassification error. IEEE Trans. on Neural Networks and Learning Systems, 2015,26(5):1048–1059.

[68] Jaworski M, Duda P, Rutkowski L. New splitting criteria for decision trees in stationary data streams. IEEE Trans. on Neural Networks and Learning Systems, 2018,29(6):2516–2529.

[69] Kakade SM, Shalev-Shwartz S, Tewari A. Efficient bandit algorithms for online multiclass prediction. In: Proc. of the Int'l Conf. on Machine Learning (ICML 2008). 2008. 440–447.

[70] Valizadegan H, Jin R, Wang S. Learning to trade off between exploration and exploitation in multiclass bandit prediction. In: Proc. of the 17th ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining (KDD 2011). 2011. 204–212.

[71] Chen G, Chen G, Zhang J, Chen S, Zhang C. Beyond banditron: A conservative and efficient reduction for online multiclass prediction with bandit setting model. In: Proc. of the IEEE Int'l Conf. on Data Mining (ICDM 2009). 2009. 71–80.

[72] Hazan E, Kale S. Newtron: An efficient bandit algorithm for online multiclass prediction. In: Proc. of the Advances in Neural Information Processing Systems (NIPS 2011). 2011. 891–899.

[73] Crammer K, Gentile C. Multiclass classification with bandit feedback using adaptive regularization. Machine Learning, 2013,90(3): 347–383.

[74] Beygelzimer A, Orabona F, Zhang C. Efficient online bandit multiclass learning with $\tilde{O}(\sqrt{T})$ regret. In: Proc. of the 34th Int'l Conf. on Machine Learning (ICML 2017). 2017. 488–497.

[75] Allwein EL, Schapire RE, Singer Y. Reducing multiclass to binary: A unifying approach for margin classifiers. Journal of Machine Learning Research, 2000,1:113–141.

[76] Auer P, Cesa-Bianchi N, Fischer P. Finite-time analysis of the multiarmed bandit problem. Machine Learning, 2002,47(2/3): 235–256.

[77] Sculley D. Online active learning methods for fast label-efficient spam filtering. In: Proc. of the 4th Conf. on Email and Anti-Spam (CEAS 2007). 2007.

[78] Chu W, Zinkevich M, Li L, Thomas A, Tseng BL. Unbiased online active learning in data streams. In: Proc. of the 17th ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining (KDD 2011). 2011. 195–203.

[79] Lughofer E, Pratama M. Online active learning in data stream regression using uncertainty sampling based on evolving generalized fuzzy models. IEEE Trans. on Fuzzy Systems, 2018,26(1):292–309.

[80] Cesa-Bianchi N, Gentile C, Zaniboni L. Worst-case analysis of selective sampling for linear-threshold algorithms. In: Proc. of the Advances in Neural Information Processing Systems (NIPS 2004). 2004. 241–248.

[81] Cesa-Bianchi N, Gentile C, Zaniboni L. Worst-case analysis of selective sampling for linear classification. Journal of Machine Learning Research, 2006,7:1205–1230.

[82] Littlestone N. Learning quickly when irrelevant attributes abound: A new linear-threshold algorithm. Machine Learning, 1988,2(4): 285–318.

[83] Zhao P, Hoi SCH. Cost-sensitive online active learning with application to malicious URL detection. In: Proc. of the 19th ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining (KDD 2013). 2013. 919–927.

[84] Lu J, Zhao P, Hoi SCH. Online passive-aggressive active learning. Machine Learning, 2016,103(2):141–183.

[85] Hao S, Zhao P, Lu J, Hoi SCH, Miao C, Zhang C. SOAL: Second-order online active learning. In: Proc. of 16th IEEE Int'l Conf. on Data Mining (ICDM 2016). 2016. 931–936.

[86] Lughofer E. Online active learning: A new paradigm to improve practical useability of data stream modeling methods. Information Sciences, 2017,415:356–376.

[87] Cesa-Bianchi N, Shalev-Shwartz S, Shamir O. Efficient learning with partially observed attributes. Journal of Machine Learning Research, 2011,12:2857–2878.

[88] Zolghadr N, Bartók G, Greiner R, György A, Szepesvári C. Online learning with costly features and labels. In: Proc. of the Advances in Neural Information Processing Systems (NIPS 2013). 2013. 1241–1249.

[89] Hazan E, Koren T. Linear regression with limited observation. In: Proc. of the 29th Int'l Conf. on Machine Learning (ICML 2012). 2012.

[90] Cesa-Bianchi N, Shalev-Shwartz S, Shamir O. Online learning of noisy data. IEEE Trans. on Information Theory, 2011,57(12): 7907–7931.

[91] Figueiredo MAT. Adaptive sparseness for supervised learning. IEEE Trans. on Pattern Analysis and Machine Intelligence, 2003,25(9):1150–1159.

[92] Guyon I, Elisseeff A. An introduction to variable and feature selection. Journal of Machine Learning Research, 2003,3:1157–1182.

[93] Yu L, Liu H. Efficient feature selection via analysis of relevance and redundancy. Journal of Machine Learning Research, 2004,5: 1205–1224.

[94] Brown G, Pocock AC, Zhao MJ, Luján M. Conditional likelihood maximisation: A unifying framework for information theoretic feature selection. Journal of Machine Learning Research, 2012,13:27–66.

[95] Tan M, Tsang IW, Wang L. Towards ultrahigh dimensional feature selection for big data. Journal of Machine Learning Research, 2014,15(1):1371–1429.

[96] Rao NS, Nowak RD, Cox CR, Rogers TT. Classification with the sparse group lasso. IEEE Trans. on Signal Processing, 2016,64(2): 448–463.

[97] Shalev-Shwartz S, Srebro N, Zhang T. Trading accuracy for sparsity in optimization problems with sparsity constraints. SIAM Journal on Optimization, 2010,20(6):2807–2832.

[98] Duchi JC, Shalev-Shwartz S, Singer Y, Chandra T. Efficient projections onto the ℓ_1 ball for learning in high dimensions. In: Proc. of the Int'l Conf. on Machine Learning (ICML 2008). 2008. 272–279.

[99] Condat L. Fast projection onto the simplex and the ℓ_1 ball. Mathematical Programming, 2016,158(1/2):575–585.

[100] Duchi JC, Singer Y. Efficient online and batch learning using forward backward splitting. Journal of Machine Learning Research, 2009,10:2899–2934.

[101] Langford J, Li L, Zhang T. Sparse online learning via truncated gradient. Journal of Machine Learning Research, 2009,10:777–801.

[102] Xiao L. Dual averaging methods for regularized stochastic learning and online optimization. Journal of Machine Learning Research, 2010,11:2543–2596.

[103] Duchi JC, Shalev-Shwartz S, Singer Y, Tewari A. Composite objective mirror descent. In: Proc. of the 23rd Conf. on Learning Theory (COLT 2010). 2010. 14–26.

[104] Duchi JC, Hazan E, Singer Y. Adaptive subgradient methods for online learning and stochastic optimization. Journal of Machine Learning Research, 2011,12:2121–2159.

[105] Wang D, Wu P, Zhao P, Wu Y, Miao C, Hoi SCH. High-dimensional data stream classification via sparse online learning. In: Proc. of the IEEE Int'l Conf. on Data Mining (ICDM 2014). 2014. 1007–1012.

[106] Zhai T, Koriche F, Wang H, Gao Y. Tracking sparse linear classifiers. IEEE Trans. on Neural Networks and Learning Systems, 2019,30(7):2079–2092. [doi: 10.1109/TNNLS.2018.2877433]

[107] Wu Y, Hoi SCH, Mei T, Yu N. Large-scale online feature selection for ultra-high dimensional sparse data. ACM Trans. on Knowledge Discovery from Data, 2017,11(4):48:1–48:22.

[108] Zhai T, Wang H, Koriche F, Gao Y. Online feature selection by adaptive sub-gradient methods. In: Proc. of the European Conf. on Machine Learning and Knowledge Discovery in Databases (ECML/PKDD 2018), Part II. 2018. 430–446. [doi: 10.1007/978-3-030-10928-8_26.

[109] Gama J, Zliobaite I, Bifet A, Pechenizkiy M, Bouchachia A. A survey on concept drift adaptation. ACM Computing Surveys, 2014, 46(4):44:1–44:37.

[110] Zhai T, Gao Y, Wang H, Cao L. Classification of high-dimensional evolving data streams via a resource-efficient online ensemble. Data Mining and Knowledge Discovery, 2017,31(5):1242–1265.

[111] Hosseini MJ, Gholipour A, Beigy H. An ensemble of cluster-based classifiers for semi-supervised classification of non-stationary data streams. Knowledge and Information Systems, 2016,46(3):567–597.

[112] Gama J, Fernandes R, Rocha R. Decision trees for mining data streams. Intelligent Data Analysis, 2006,10(1):23–45.

[113] Bifet A, Holmes G, Pfahringer B, Frank E. Fast perceptron decision tree learning from evolving data streams. In: Proc. of the Pacific-Asia Conference on Knowledge Discovery and Data Mining. 2010. 299–310.

[114] Bifet A, Pfahringer B, Read J, Holmes G. Efficient data stream classification via probabilistic adaptive windows. In: Proc. of the 28th Annual ACM Symp. on Applied Computing (SAC 2013). 2013. 801–806.

[115] Hulten G, Spencer L, Domingos P. Mining time-changing data streams. In: Proc. of the 7th ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining (KDD 2001). 2001. 97–106.

[116] Elwell R, Polikar R. Incremental learning of concept drift in nonstationary environments. IEEE Trans. on Neural Networks, 2011, 22(10):1517–1531.

[117] Brzezinski D, Stefanowski J. Accuracy updated ensemble for data streams with concept drift. In: Proc. of the Int'l Conf. on Hybrid Artificial Intelligence Systems. 2011. 155–163.

[118] Brzezinski D, Stefanowski J. Reacting to different types of concept drift: The accuracy updated ensemble algorithm. IEEE Trans. on Neural Networks and Learning Systems, 2014,25(1):81–94.

[119] Bonab HR, Can F. GOOWE: Geometrically optimum and online-weighted ensemble classifier for evolving data streams. ACM Trans. on Knowledge Discovery from Data, 2018,12(2):25:1–25:33.

[120] Zhao QL, Jiang YH, Lu YT. Ensemble model and algorithm with recalling and forgetting mechanisms for data stream mining. Ruan Jian Xue Bao/Journal of Software, 2015,26(10):2567–2580 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/4747.htm [doi: 10.13328/j.cnki.jos.004747]

[121] Oza NC. Online bagging and boosting. In: Proc. of the IEEE Int'l Conf. on Systems, Man and Cybernetics (SMC 2005). 2005. 2340–2345.

[122] Kolter JZ, Maloof MA. Dynamic weighted majority: An ensemble method for drifting concepts. Journal of Machine Learning Research, 2007,8:2755–2790.

[123] Bifet A, Gavalda R. Learning from time-changing data with adaptive windowing. In: Proc. of the 7th SIAM Int'l Conf. on Data Mining (SDM 2007). 2007. 443–448.

[124] Bifet A, Holmes G, Pfahringer B. Leveraging bagging for evolving data streams. In: Proc. of the Joint European Conf. on Machine Learning and Knowledge Discovery in Databases. 2010. 135–150.

[125] Minku LL, Yao X. DDD: A new ensemble approach for dealing with concept drift. IEEE Trans. on Knowledge and Data Engineering, 2012,24(4):619–633.

[126] Brzezinski D, Stefanowski J. Combining block-based and online methods in learning ensembles from concept drifting data streams. Information Sciences, 2014,265:50–67.

[127] Minku LL, White AP, Yao X. The impact of diversity on online ensemble learning in the presence of concept drift. IEEE Trans. on Knowledge and Data Engineering, 2010,22(5):730–742.

[128] Gama J, Sebastiao R, Rodrigues PP. On evaluating stream learning algorithms. Machine Learning, 2013,90(3):317–346.

[129] Xu GY, Han M, Wang SF, Jia T. Summarization of data stream ensemble classification algorithm. Application Research of Computers, 2020,37(1) (in Chinese with English abstract). http://www.arocmag.com/article/01-2020-01-001.html [doi: 10.19734/j.issn.1001-3695.2018.09.0510]

[130] Cabral DRL, Barro RSM. Concept drift detection based on Fisher's exact test. Information Sciences, 2018,442/443:220–234.

[131] Pesaranghader A, Viktor H, Paquet E. Reservoir of diverse adaptive learners and stacking fast Hoeffding drift detection methods for evolving data streams. Machine Learning, 2018,107(11):1711–1743.

[132] Liu A, Lu J, Liu F, Zhang G. Accumulating regional density dissimilarity for concept drift detection in data streams. Pattern Recognition, 2018,76:256–272.

[133] Pan WB, Cheng G, Guo XJ, Huang SX. An adaptive classification approach based on information entropy for network traffic in presence of concept drift. Chinese Journal of Computers, 2017,40(7):1556–1571 (in Chinese with English abstract).

[134] Foster D, Kale S, Karloff H. Online sparse linear regression. In: Proc. of the 29th Annual Conf. on Learning Theory (COLT 2016). 2016,49:960–970.

[135] Zliobaite I, Bifet A, Pfahringer B, Holmes G. Active learning with drifting streaming data. IEEE Trans. on Neural Networks and Learning Systems, 2014,25(1):27–39.

[136] Mohamad S, Mouchaweh MS, Bouchachia A. Active learning for data streams under concept drift and concept evolution. In: Proc. of the Workshop on Large-scale Learning from Data Streams in Evolving Environments (STREAMEVOLV 2016) co-located with the 2016 European Conf. on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML/PKDD 2016). 2016.

[137] Park CH, Kang Y. An active learning method for data streams with concept drift. In: Proc. of the IEEE Int'l Conf. on Big Data (BigData 2016). 2016. 746–752.

附中文参考文献:

[3] 翟婷婷.面向流数据分类的在线学习算法研究[博士学位论文].南京:南京大学,2018.

[28] 李志杰,李元香,王峰,何国良,匡立.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.

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

[120] 赵强利,蒋艳凰,卢宇彤.具有回忆和遗忘机制的数据流挖掘模型与算法.软件学报,2015,26(10):2567–2580. http://www.jos.org.cn/1000-9825/4747.htm [doi: 10.13328/j.cnki.jos.004747]

[129] 许冠英,韩萌,王少峰,贾涛.数据流集成分类算法综述.计算机应用研究,2020,37(1). http://www.arocmag.com/article/01-2020-01-001.html [doi: 10.19734/j.issn.1001-3695.2018.09.0510]

[133] 潘吴斌,程光,郭晓军,黄顺翔.基于信息熵的自适应网络流概念漂移分类方法.计算机学报,2017,40(7):1556–1571.

翟婷婷(1988－),女,河南济源人,博士,讲师,主要研究领域为机器学习,模式识别.

朱俊武(1972－),男,博士,教授,博士生导师,CCF 高级会员,主要研究领域为知识工程,本体论,机制设计,云计算.

高阳(1972－),男,博士,教授,博士生导师,CCF 高级会员,主要研究领域为大数据分析,机器学习,多智能体系统,视频/图像处理.