软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software, 2022,33(4):1287−1314 [doi: 10.13328/j.cnki.jos.006476]

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

概念漂移数据流半监督分类综述*

文益民[1,2], 刘 帅[2], 缪裕青[2], 易新河[2], 刘长杰[2]

[1](广西图像图形与智能处理重点实验室(桂林电子科技大学), 广西 桂林 541004)

[2](桂林电子科技大学 计算机与信息安全学院, 广西 桂林 541004)

通信作者: 文益民, E-mail: ymwen@guet.edu.cn

摘 要: 在开放环境下, 数据流具有数据高速生成、数据量无限和概念漂移等特性. 在数据流分类任务中, 利用人工标注产生大量训练数据的方式昂贵且不切实际. 包含少量有标记样本和大量无标记样本且还带概念漂移的数据流给机器学习带来了极大挑战. 然而, 现有研究主要关注有监督的数据流分类, 针对带概念漂移的数据流的半监督分类的研究尚未引起足够的重视. 因此, 在全面收集数据流半监督分类研究工作的基础上, 对现有带概念漂移的数据流的半监督分类算法进行了多角度划分; 并以算法采用的分类器类型为线索, 对已有的多个算法进行了介绍与总结, 包括现有数据流半监督分类采用的概念漂移检测方法; 在一些被广泛使用的真实数据集和人工数据集上, 对部分代表性数据流半监督分类算法进行了多方面的比较与分析; 最后, 提出了当前概念漂移数据流半监督分类中一些值得进一步深入探讨的问题. 实验结果表明: 数据流半监督分类算法的分类准确率与众多因素有关, 但与数据分布的变化关系最大. 本综述将有助于感兴趣的研究者快速进入数据流半监督分类问题领域.

关键词: 数据挖掘; 概念漂移; 数据流; 集成学习; 半监督分类

中图法分类号: TP181

中文引用格式: 文益民, 刘帅, 缪裕青, 易新河, 刘长杰. 概念漂移数据流半监督分类综述. 软件学报, 2022, 33(4): 1287–1314. http://www.jos.org.cn/1000-9825/6476.htm

英文引用格式: Wen YM, Liu S, Miao YQ, Yi XH, Liu CJ. Survey on Semi-supervised Classification of Data Streams with Concept Drifts. Ruan Jian Xue Bao/Journal of Software, 2022, 33(4): 1287−1314 (in Chinese). http://www.jos.org.cn/1000-9825/6476.htm

Survey on Semi-supervised Classification of Data Streams with Concept Drifts

WEN Yi-Min[1,2], LIU Shuai[2], MIAO Yu-Qing[2], YI Xin-He[2], LIU Chang-Jie[2]

[1](Guangxi Key Laboratory of Image and Graphic Intelligent Processing (Guilin University of Electronic Technology), Guilin 541004, China)

[2](School of Computer Science and Information Security, Guilin University of Electronic Technology, Guilin 541004, China)

Abstract: In the open environment, data streams have the characteristics of high-speed data generation, unlimited data volume, and concept drift. In the task of data stream classification, it is expensive and impractical to generate a large amount of training data by manual annotation. A data stream with a small number of samples labeled and a large number of samples unlabeled and with concept drifts presents a great challenge to machine learning. However, the existing research mainly focuses on supervised classification of data streams, while semi-supervised classification of data streams with concept drifts has not yet attracted attention enough. Therefore, based on the comprehensive collection of the work of semi-supervised classification of data streams, this study sorts the existing semi-supervised data stream classification algorithms into several types from several aspects, describes and summarizes many existing algorithms based on the types of classifiers used in the algorithms and the concept drift detection methods utilized. On some widely employed real and synthetic datasets, several representative semi-supervised classification algorithms for data streams are chosen to be compared and analyzed in

* 基金项目: 广西自然科学基金(2018GXNSFDA138006); 国家自然科学基金(61866007); 教育部人文社会科学研究项目(17JDGC022); 广西图像图形与智能处理重点实验室项目(GIIP2005, GIIP201505, GIIP201706)

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: 2021-05-30; 修改时间: 2021-07-16; 采用时间: 2021-08-27; jos 在线出版时间: 2021-10-26

many aspects. Finally, this study proposes some issues that are worthy to be further discussed in future for semi-supervised classification of data streams with concept drifts. The experimental results show that the classification accuracy of the algorithms for semi-supervised data stream classification is related to many factors, but it has the greatest relationship with the changes of data distribution. This review will help the interested researchers quickly enter into the field of semi-supervised classification of data streams.

Key words: data mining; concept drift; data stream; ensemble learning; semi-supervised classification

传统的机器学习算法得在静态封闭环境[1]下运行, 且通常假定算法测试时的数据分布与其训练集的数据分布相同[2−7]. 但在实际中, 数据的分布、类别和属性常受环境等因素的影响而不断发生难以预知的变化[8,9]. 在国际人工智能大会(AAAI)的主席报告中, Dietterich 教授指出: 未来的人工智能需要应对各种未知的情况, 并称其为“Unknown Unknowns”. 周志华教授在新智元 AI World 2018 世界人工智能峰会上的“关于机器学习的一点思考”中, 将该问题定义为一种开放环境下的机器学习问题. Zhang 等人将该问题定义为机器学习的鲁棒性问题[10]. Wang 等人提及的域泛化[11]也与该问题相关. 随着数据存储和处理技术的飞速发展, 开放环境下的高风险应用给传统机器学习带来了巨大挑战, 比如自动驾驶汽车、自主武器、远程辅助外科手术等. 这一类应用无一例外都是一旦出现了问题, 就会造成巨大损失.

数据流分类[12−14]在大规模实时数据处理应用中起着至关重要的作用. 与传统的静态数据挖掘相比, 数据流挖掘包含以下特性[15−19]: (1) 巨大的数据量和有限的计算资源使得存储所有数据以便分析和处理的方式变得不切实际; (2) 高速产生的数据流意味着算法应该对它们进行实时处理, 因此需要对整个数据集进行多次遍历以建立分类器的算法会变得低效; (3) 环境的变化、设备的损耗等因素使得数据流往往包含概念漂移, 即数据的分布随着时间而发生不同程度的变化, 独立同分布假设在包含概念漂移的数据流中不成立.

在实际数据流分类场景中, 受标注成本、数据产生速度和数据量巨大等因素的影响, 可用于学习的标记样通常比较少, 从而大部分样本无标记. 比如在网络入侵检测系统中, 网络连接数据必须被有效地处理以维护安全的网络环境. 由于网络连接的数量巨大、速度非常快, 对所有连接数据进行人工标注昂贵且不现实, 系统必须在半监督条件下准确且高效地完成检测和模型更新. 仅利用少量标记样本学习得到的模型的泛化能力会比较差, 而且会造成大量无标记样本的浪费. 近年来, 数据流半监督分类[20]开始得到学术界的关注, 并已经被应用于入侵检测[21]、故障诊断[22]、垃圾邮件处理[23]、推荐系统[24]和平面铣削[25]等相关领域.

在数据流半监督分类情形下, 通常假定有少量样本在被分类后其标记可以获得, 而大量样本未被标记. 因此, 与监督数据流分类相比, 数据流的半监督分类存在一些额外挑战: 首先, 用少量标记样本训练的分类模型的泛化能力会比较差, 如何利用大量无标记样本的内在结构和分布来辅助训练? 其次, 基于准确率的概念漂移检测方法不能很好地适应半监督条件的概念漂移检测, 如何同时检查 $P(X)$和 $P(Y|X)$的变化? 最后, 分类模型更新时, 标记样本和无标记样本应被同时考虑以完成分类模型的更新, 从而最大程度利用已有样本来提高分类模型的泛化能力.

自 Schlimmer 等人[26]于 1986 年首次提出“概念漂移”以来, 国内外众多研究人员从不同角度对概念漂移数据流分类算法进行了详细综述. Kuncheva 等人[27]于 2004 年对变化环境中的在线学习和集成学习进行了综述和讨论. Tsymbal 等人[28]于 2004 年分别从样本选择、样本加权和集成学习角度对概念漂移的相关工作进行了详细介绍. 王涛等人[29]于 2007 年从平稳分布和带概念漂移的数据流分类两个方面分别综述了已有的数据流分类算法. Žliobaitė[30]于 2010 年对概念漂移检测方法进行了归类和总结, 并分别介绍了与概念漂移相关的 4 种应用类型: 异常行为检测控制、推荐系统中的个人协助、信誉评估中的决策和环境交互中的人工智能应用. Gama[31]于 2012 年从数据流分类、聚类和频繁项挖掘等方面分别介绍了数据流挖掘领域中的工作. Hoens 等人[32]于 2012 年对现有概念漂移检测方法进行了回顾和总结, 并对现有概念漂移数据流中的类别不平衡问题也进行了归纳总结. 文益民等人[33]于 2013 年从单模型和集成模型、带概念漂移的数据流学习理论、概念漂移检测等方面对概念漂移数据流分类进行综述, 并对涉及概念漂移数据流半监督分类与主动学习的部分算法也做了简单总结. Gama 等人[34]于 2014 年从算法的内存使用情况、概念漂移检测、模型更新方式以及模型评估指标等 4 个方面对概念漂移自适应学习算法进行综述. Krempl 等人[35]于 2014 年讨论了数据流挖掘的 8 个开放

挑战,探讨了出当前研究与有意义应用之间的差距, 并提出未解决的问题. Goncalves 等人[36]于 2014 年对已有的 8 种概念漂移检测(ddm, eddm, pht, stepd, dof, adwin, Paired Learners, and ecdd)方法进行了实验, 以分析不同方法对不同概念漂移类型的检测能力. Ditzler 等人[37]于 2015 年从主动检测和被动检测两个方面对现有概念漂移检测方法进行了综述, 作者将主动检测划分为 Hypothesis Tests, Change Point Methods, Sequential Hypothesis Tests 和 Change Detection Tests 这 4 种方法, 将被动检测划分为单模型和集成模型两种方法, 分别介绍了相关研究工作. 作者还提供了 MOA, Online Nonstationary Boosting, Learn++.NSE 等开源软件和部分数据集的链接. Nguyen 等人[38]于 2015 年分别以不同的聚类模型(包括层次、网格、密度)和不同的分类模型(包括决策树、贝叶斯、神经网络、集成模型)为线索, 对现有的数据流聚类和数据流分类算法进行了综述, 并给出了传统的聚类、分类模型和数据流聚类、分类模型之间的对应关系. 丁剑等人[39]于 2016 年对涉及决策树分类和集成分类的数据流分类算法及常见的 Clustream, DGClust 和 D-Stream 等数据流聚类算法以及频繁模式挖掘和高效用模式挖掘等内容进行了综述. Webb 等人[40]于 2016 年提出了用于概念漂移定量分析的第一个综合框架, 主要涉及漂移目标、漂移频率、漂移过渡方式、漂移重现和漂移量的大小等多种类型, 并给出了不同概念漂移类型的正式定义. Krawczyk 等人[18]于 2017 年对数据流分类和回归任务中的集成学习进行了综述, 并对数据流半监督集成学习相关算法进行了总结和归类. Iwashita 等人[41]于 2018 年对现有概念漂移检测方法进行了综述, 并给出了现有算法在有无概念漂移检测方法、不同类型的概念漂移处理方法以及分类模型等视角的归类结果, 介绍了部分数据流半监督分类算法, 还介绍了用于实验的大量真实数据集和人工数据集. Khamassi 等人[42]于 2018 年从数据处理方式、分类器数量、自适应等方面对检测概念漂移的方法进行了总结. Lu 等人[43]于 2019 年从概念漂移检测的方法、概念漂移的理解(何时产生、程度如何、延续时间多长)、评测数据与评测方法以及概念漂移数据流分类中的类别不平衡、大数据挖掘、主动学习与半监督学习、数据驱动的决策支持系统等方面进行了全面的总结, 尤其注意总结了 2015 年以来概念漂移数据流分类的新研究.

尽管已经存在许多关于数据流半监督分类的理论和应用研究工作, 但是在上述综述中, 数据流半监督分类算法仅在部分综述中被提及[19,33,37,41], 且这些综述对数据流半监督分类的相关工作总结得不够全面, 该类算法的共同特点、面临的问题与挑战等内容尚没有得到总结与归纳. 对于具体应用而言, 什么样的场景选择什么样的算法也不得而知. 因此, 针对数据流半监督分类算法的综述性文章仍然缺乏. 本文主要对半监督环境下的概念漂移数据流分类算法和概念漂移检测方法进行了归类与介绍, 并对部分数据流半监督分类算法进行了比较与分析, 还提出了当前数据流半监督分类领域中一些值得进一步深入探讨的问题. 这能使相关研究人员能够快速了解该领域的研究现状.

据作者所知, 本文应该是带概念漂移的数据流半监督分类的第一篇综述. 本文的贡献主要包括 4 个方面.

(1) 对数据流半监督分类算法进行了多角度划分, 便于根据研究与应用的要求选择算法;
(2) 以分类模型为主线, 对现有的多个针对带概念漂移的数据流的半监督分类算法进行了总结与介绍, 有利于感兴趣者快速了解各算法特点;
(3) 对数据流半监督分类问题的概念漂移检测方法进行了归纳与总结;
(4) 基于真实数据集和人工数据集, 对部分有代表性的数据流半监督分类算法进行了多方面的比较与分析.

本文第 1 节介绍概念漂移的定义和类型. 第 2 节介绍数据流半监督分类的研究概况, 对数据流半监督分类算法进行了多角度划分. 第 3 节以分类模型为线索, 介绍数据流半监督分类的研究进展. 第 4 节介绍现有数据流半监督分类算法涉及的概念漂移检测方法. 第 5 节对部分数据流半监督分类算法进行了多方面的比较与分析. 第 6 节提出当前数据流半监督分类中一些值得深入探讨的问题. 最后是总结.

1 问题描述

在很多数据流分类的应用场景中, 只能获得少量有标记的样本. 根据被标记样本在整个数据流中的位置, 可以划分为 3 种情形.

(1) 部分样本被随机标记. 这种情形更加符合实际应用场景;

(2) 有标记样本仅在模型初始化时给出, 后续样本全部为无标记样本. 这种情形被称为 Initially Labeled Streaming Environment (ILSE)[44,45], 或者叫 Extreme Verification Latency (EVL)[46]. 此种情形下的数据流半监督分类通常假设数据的分布逐渐发生变化;

(3) 在分块处理数据流中, 部分数据块上样本的都有标记[47], 而部分数据块中的样本都无标记.

1.1 概念漂移的定义

根据数据源产生的数据是否服从独立同分布假设, 可以将数据流分为两类: 静态数据流[48]和包含概念漂移的动态数据流[49]. Žliobaitė 等人[50]认为, 动态数据流中的概念漂移产生于数据分布的意外性变化. 现有工作[13,40]通常从概率角度定义该变化. 假设时间点 t 到来的样本都是由具有联合概率分布 $P^t(\boldsymbol{x},y)$的数据源产生的. 在静态数据流中, 任意两个不同时间点的样本服从同一数据分布 $P^t(\boldsymbol{x},y)=P^{t+\lambda}(\boldsymbol{x},y)$. 在包含概念漂移的动态数据流中, 对于两个不同时间点的样本, 如果 $P^t(\boldsymbol{x},y)\neq P^{t+\lambda}(\boldsymbol{x},y)$, 就表示发生了概念漂移.

根据贝叶斯决策理论, 一个分类器可以被描述为: 给定样本 $\boldsymbol{x}$, 使其后验概率 $P(y_i|\boldsymbol{x})$最大的类别 y_i 为该样本的预测类别, 其定义如下:

$$P(y_i \mid \boldsymbol{x}) = \frac{P(y_i) * P(\boldsymbol{x} \mid y_i)}{P(\boldsymbol{x})}.$$

其中, $P(y_i)$是类的先验概率, $P(\boldsymbol{x}|y_i)$是样本 $\boldsymbol{x}$ 相对于类别 y_i 的类条件概率, $P(\boldsymbol{x}) = \sum_{i=1}^{c} P(y_i)P(\boldsymbol{x} \mid y_i)$ 是用于归一化的证据因子, c 表示类的数量. 根据贝叶斯决策理论, Kelly 等人[51]认为, 产生概念漂移的原因主要有 3 种.

(1) 类的先验概率 $P(y_i)$发生变化;

(2) 类的条件概率 $P(\boldsymbol{x}|y_i)$发生变化;

(3) 后验概率 $P(y_i|\boldsymbol{x})$发生变化.

1.2 概念漂移的类型

根据概念漂移的定义, 可以将概念漂移分为两种类型: 真实漂移和虚漂移[19,34]. 真实漂移定义为由 $P(y_i|\boldsymbol{x})$的变化而导致的概念漂移; 虚漂移定义为在不影响 $P(y_i|\boldsymbol{x})$的情况下, 由数据分布 $P(\boldsymbol{x})$的变化而导致的概念漂移. 然而, 开放环境下产生的数据流可能包含了以上两种概念漂移类型的混合.

根据数据分布随时间变化的形式, 可以将概念漂移分为突变、渐变、增量和重现这 4 种形式[15,30,33,34,41]. 图 1 给出了由于数据均值不同程度的变化而导致的 4 种概念漂移的抽象化描述.

类型 I(突变概念漂移(abrupt)): 一个概念在某一时刻突然切换到另一个概念;

类型 II(渐变概念漂移(gradual)): 数据在一个过渡阶段内, 通过从两个不同分布中动态采样而产生. 在该阶段内, 随着时间的推移, 从一个分布中采样的概率减少, 从另一个分布中采样的概率增加;

类型 III(增量概念漂移(incremental)): 数据分布随时间变化, 这些分布之间存在细微的差异. 只有考虑更长的观察时间时, 才会表现出明显的漂移现象;

类型 IV(重现概念漂移(reoccurring)): 相同的数据分布在未来一段时间内再次出现.

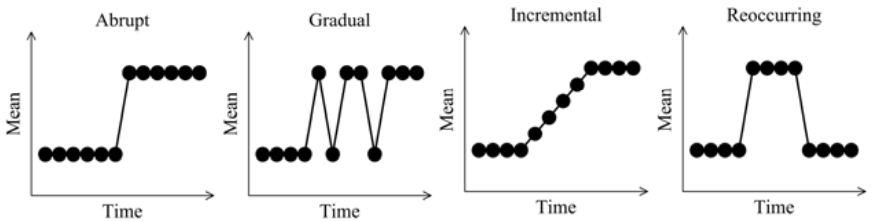

图 1 概念漂移的 4 种类型

在数据流分类领域, 除了前面所述的概念漂移, 其实还包括样本特征的动态变化、样本所属类别的数量的动态变化. 样本类别数量的变化在文献中通常被称为概念进化[33]. 需要说明的是: 本文题目中虽然标有关

键字“概念漂移”, 其实在综述时还介绍了概念进化问题. 这是因为概念进化问题在数据流分类领域很常见.

2 数据流半监督分类研究的概况

半监督情形下的数据流分类降低了数据的标注成本, 适应了数据的高速产生, 更加适合众多实际应用场景的需求. 因此, 该领域相关的研究工作正在逐年增加. 本节根据分类模型对现有数据流半监督分类算法进行了归类, 给出了如表 1 所示的划分. 表 1 还从模型更新方式、模型数量和概念漂移检测方式等角度对数据流半监督分类算法进行了归类.

表 1 算法的不同划分

分类模型	参考文献	年份	算法名字	模型更新		模型数量		主动检测
				在线	分块	单模型	多模型	
聚类	55	2008	SmSCluster	–	√	–	√	–
	57	2009	LabelStream	–	√	–	√	–
	56	2012	ReaSC	–	√	–	√	–
	16	2012	SEClass	–	√	–	√	–
	15	2016	SPASC	–	√	–	√	√
	58	2018	SSCLCR	–	√	–	√	√
	59	2018	DISSFCM	–	√	√	–	–
	60	2020	SCBELS	–	√	–	√	√
	61	2020	ORSSL-MC	√	–	√	–	–
	62	2021	ESCR	–	√	–	√	√
决策树	63	2010	REDLLA	√	–	√	–	√
	65	2011	CSL-Stream	√	–	√	–	–
	64	2012	SUN	√	–	√	–	√
	67	2017	TriTDS	–	√	–	√	–
	17	2018	SCo-Forest	–	√	–	√	√
	68	2020	CAPLRD	–	√	–	√	√
	69	2020	SSCADP	–	–	–	–	–
图	70	2012	KAOGINCSSL	–	√	√	–	–
SVM	72	2009	RK-TS^{3}VM	–	√	√	–	–
	71	2009	SSL-E	–	√	√	–	–
	22	2017	4SFD	–	√	√	–	√
神经网络	75	2019	ISLSD	–	√	√	–	–
	76	2019	SSOE-FP-ELM	–	√	√	–	–
其他	47	2010	ECU	–	√	–	√	–
	52	2016	IS^{3}RS	–	√	√	–	–
	44 45	2012 2013	COMPOSE	–	√	√	–	–
	53	2015	SluiceBox	√	–	–	√	–
	46	2019	AMANDA	–	√	√	–	–
	54	2021	SSE-PBS	–	√	–	√	–

根据算法使用的分类模型, 可以将现有算法划分为基于聚类、决策树、图、支持向量机(support vector machine, SVM)、神经网络和其他等 6 种. 基于聚类的算法主要通过半监督聚类算法得到聚簇集合, 这些簇保存了用于分类的统计信息(比如簇的质心、簇中类的分布等). 基于图的算法维持着一个图结构, 监督信息存储在部分顶点上用于分类, 顶点之间通过具有权值的边连接, 通过图中新旧顶点的替换和权值的更新, 使模型适应动态的数据流环境. 基于决策树的算法通常利用半监督学习算法标记更多样本进而完成模型更新, 比如: REDLLA 和 SUN 算法分别在决策树的叶子节点中使用 *K*-means 和 *K*-modes 算法来标记样本; SCo-Forest 算法对随机森林中的多棵随机树运用 Co-training 算法的思想来扩充标记样本. 基于 SVM 的算法利用支持向量或改进的半监督 SVM 来实现分类任务, 比如 RK-TS^{3}VM 算法使用改进后的迁移半监督的支持向量机 TS^{3}VM 作为分类器用于分类. 在基于其他分类模型的算法中, 分类模型可以是 *K*NN[52]、Bayes[53]、标记传播[46]等常用算法, 还可以是一些特殊的算法, 比如推荐系统中的 extBRISMF 算法[24], 或不限定于分类模型[54]. 由于涉及这些模型的算法比较少, 因此本文将这些模型归类为基于其他模型. 从表 1 中可以发现, 基于聚类的模型占据

了最大的比重. 这是因为聚类算法能够更好地捕获数据的分布形成基于簇的分类器, 更加适应于数据流半监督分类. 基于决策树的模型得益于快速决策树的高效性, 使其更加适应数据流的高速特性. 因此, 该类模型也比较常见.

算法使用的模型更新方式包括两种: 在线和分块. 在线更新是分类模型逐个学习新样本, 当条件满足时对分类模型进行更新; 分块更新是指分类模型逐块学习新样本, 利用新样本块对分类模型进行更新. 相对于在线模型更新方式, 分块更新模型具有可用于更新模型和检测概念漂移的样本更多、学习得到的模型更强、更加适用于集成模型等优势. 因此, 分块更新模型占据了主要部分.

算法分类模型中使用的分类器的数量包括单模型和多模型. 在单模型的算法中, 单个分类器在数据流的整个处理阶段被维持, 通过分类器的替换或更新等方式来使模型适应新的数据流环境. 集成模型通常维持着一个固定大小、由多个单分类器组成的分类器池, 池中每个基分类器可以是相同模型(比如基于簇的分类模型)或不同模型(比如基于决策树和聚簇的分类模型). 通过分类器池中单个分类器的替换、更新或各个分类的权值调整等方式来使模型适应新的数据流环境. 表 1 中的数据显示, 单模型占据的比例与多模型占据的比例大致相当.

算法对概念漂移的适应分两种情形: 一种是不检测, 分类模型被动持续更新, 被动适应新概念; 第 2 种是主动检测概念漂移, 主动适应更新模型适应新概念. 不检测指没有显式的概念漂移检测过程, 该类算法主要通过单模型的在线更新/分块更新、多模型中基分类器的替换和更新、基分类器的权值调整等方式来使模型适应新的概念. 在主动检测方法中, 算法周期性地检测数据流中是否出现新概念或重现概念. 该方法作为一个独立部分, 将在第 4 节被详细的介绍. 相比于不检测, 主动检测需要额外的参数, 而参数的确定比较困难,往往取决于不同的应用环境, 而且需要特殊的背景知识. 因此, 不检测的方法在参数数量上具有明显优势, 但是不能很好地处理包含重现概念漂移的数据.

3 数据流半监督分类的研究进展

本节以算法的分类模型为线索, 对现有的数据流半监督分类的研究进展进行了综述. 为体现数据流半监督分类研究综述的完整性, 在该领域中, 与主动学习相关的一些研究工作也将被简单介绍.

3.1 基于聚类的算法

该类算法的分类模型包括单模型和集成模型. 该类算法利用各种聚类算法获得聚簇集合(有时这些聚簇会被进一步划分为微簇), 每个聚簇中保存了簇中样本的统计信息(比如质心、类分布或半径), 这些信息可被用于分类. 聚类能较好地体现数据分布的局部和全局特征, 也很容易被更新, 很适合处理无标记, 因此聚类常被用于数据流的半监督分类, 其中, 半监督聚类比监督聚类更常见. 但是, 聚类并不能保证所有的簇包含有标记样本. 对于不包含标记样本的簇, 算法需要额外的一些操作(比如标记传播)为其指派一个合适的标记, 这会影响算法的效率. 该类算法需要预先指定聚簇数量, 且在整个数据流处理过程中保持聚簇数量不变. 考虑到有概念漂移的数据流环境中新旧概念簇的出现或消失, 簇的数量对算法性能会有很大影响.

Masud 等人分别于 2008 年和 2012 年提出了基于半监督聚类的数据流集成分类算法 SmSCluster[55]和 ReaSC[56]. Woolam 等人于 2009 年提出了基于半监督聚类的数据流集成分类算法 LabelStream[57]. 在这些算法中, 样本以数据块的方式到来, 数据块中只有部分样本被标记. 这些算法每采集到一个数据块就利用半监督聚类算法(MCI-Kmeans)获得簇集合, 从而得到一个基于聚簇的分类器.

SmSCluster 算法分别在前 L 个数据块上训练 L 个基于簇的基分类器, 并添加到分类器池中用于初始化分类模型. 在分类阶段, SmSCluster 分别计算每个基分类器中距离待分类样本距离最近的 Q 个簇, 分类器池中 L 个基分类器对应的 $L\times Q$ 个簇集合中累计归一化频率(CNFrq)最高的类别即为预测类别. 在模型更新阶段, 首先在当前数据块 D^t 上新建一个基于簇的分类器, 若该分类器的簇集合中包含有其他基分类器簇集合中所不含的新类别, 算法以概率ρ将该分类器中包含新类的簇添加到其他基分类器的簇集合中. 然后, 新建分类器被加入分类器池中, 计算池中的所有基分类器在 D^t 中有标记样本上的分类准确率, 准确率最差的基分类器被淘汰.

与 SmSCluster 算法不同, LabelStream 和 ReaSC 算法对聚类得到的每个簇进一步划分得到更纯的微簇, 并利用标记传播算法给无标记的微簇分配一个伪标记. 每个微簇存储了该微簇中样本的统计信息. 这些微簇的集合构成一个基于簇的分类器而存储在大小为 L 的分类器池中. 在分类阶段, LabelStream 和 ReaSC 中每个基分类器使用各簇对样本 $\boldsymbol{x}$ 进行带权集成分类, 得到该样本属于各个类别的概率向量 $\hat{y}$. 在模型更新阶段, 同样在当前数据块 D^t 上新建一个基于簇的分类器, 但是当有新的类别到来时, LabelStream 和 ReaSC 没有概率 ρ 的限制, 直接将 D^t 上训练的簇模型中包含新类的微簇添加到分类器池中每个基分类器的微簇集合中.

徐文华等人[16]于 2012 年提出了一种基于半监督学习的数据流集成分类算法 SEClass. 样本以块的方式到来, 数据块中的部分样本被标记. SEClass 采用半监督聚类算法 SW-Kmeans 对当前数据块 D^t 聚类, 得到簇的集合, 将它作为基分类器添加到分类器池中, 池中每个基分类器对应一个权值. SW-Kmeans 的特点是在 K-means 的基础上引入了簇的同质性和样本属性的重要度. 在分类阶段, 在每个基分类器的簇集合中, 将距离样本 $\boldsymbol{x}$ 最近的簇中的多数类作为相应基分类器给出的分类结果. 所有基分类器的分类结果的带权和中最大值对应的类别即为最终分类. 对分类器池更新时, SEClass 计算各基分类器对 D^t 中标记样本的分类准确率, 而更新各分类器的权值.

Hosseini 等人[15]于 2016 年提出了一种处理有重现概念漂移的数据流的半监督集成分类算法 SPASC. 样本以块的方式到来, 数据块中的部分样本被标记. SPASC 假设聚类得到的每个簇服从高斯分布, 使用 EM 方法求解隐变量(每个样本所属的簇)来获得簇集合. 将获得的簇集合作为一个基分类器保存于分类器池中, 并初始化权重 $\boldsymbol{w}$ 为 1. 当数据块 D^t 到来时, 针对 D^t 中的每个样本, 选择分类器池中权值最大的基分类器, 采用 K 近邻算法对样本分类. 每个样本被分类后, 若其标记立即获得, 则根据对该样本是否被分类正确立即调整各基分类器的权值. SPASC 采用半监督贝叶斯方法计算每个基分类器与 D^t 间的相似性, 以判断是否发生概念漂移. 若出现重现概念或分类器池已满, 则用 D^t 增量更新最大相似性对应的基分类器; 否则, 算法在 D^t 上新建一个基于簇的分类器并添加到分类器池中. 最后, 计算每个基分类器在 D^t 上的错误率, 以更新各基分类器的权值.

Qin 等人[58]于 2018 年在 SPASC 的基础上提出了一种基于局部成分替换的概念漂移数据流半监督分类算法 SSCLSR, 它和 SPASC 的主要区别在于: 当检测到重现概念时, 才将对应的分类器进行增量更新; 否则, 在 D^t 上训练一个新的基于簇的分类器. 依次遍历池中各个基分类器中的每个簇, 计算该簇参与对 D^t 中有标记样本分类的总次数中分类正确次数所占的比率. 若该比率小于指定阈值, 则该簇被替换为新建分类器中距离该簇最近的簇.

Casalino 等人[59]于 2018 年提出了一种基于增量自适应模糊聚类的数据流半监督分类算法 DISSFCM. 样本以分块的方式到来, DISSFCM 利用半监督模糊 C 均值聚类算法对数据块进行聚类. 在分类阶段, 对于数据块 D^t 中的每个样本, 距离其最近的簇对应的类别即为预测类别. 在模型更新阶段, 将已获得聚类原型与 D^t 合并, 使用 SSFCM 算法聚类, 得到新的基于簇的分类模型 f_{new}, 并计算 f_{new} 的每个簇的重构误差值, 根据重构误差值确定哪些簇要再进行聚类, 以将其划分成两个簇. 模型更新机制能够动态适应簇的数量, 能更好地捕获概念漂移.

Wen 等人[60]于 2020 年在 SPASC 的基础上提出了基于 BIRCH 集成和局部结构映射的数据流半监督分类算法 SCBELS, 该算法和 SPASC 的主要区别在于: 分类器池中的每个基分类器是一个 BIRCH 集成分类器, 每个 BIRCH 集成分类器由多棵聚类特征树组成; 半监督贝叶斯方法和迁移学习中的局部结构映射策略被结合来计算每个样本和各分类器之间的局部相似性, 从而实现概念漂移检测, 使得对概念漂移的检测比 SPASC 更准. 在分类阶段, t 时刻新建的 BIRCH 集成分类器或被更新的 BIRCH 集成分类器用于对数据块 D^{t+1} 中样本的分类. 在模型更新阶段, 若从 D^t 上检测到重现概念, 则用 D^t 更新分类器池中的对应分类器; 否则, 利用 D^t 再训练一个分类器加入分类器池.

Din 等人[61]于 2020 年提出了一种针对概念漂移数据流分类的在线可靠半监督学习算法 ORSSL-MC(本文作者命名). 算法使用 K-means 对一组用于分类器初始化的有标记样本进行聚类, 将得到的簇集合作为分类器. 当样本 $\boldsymbol{x}$ 到来时, 计算分类器的簇集合中跟 $\boldsymbol{x}$ 邻近的簇, 使用 KNN 方法对 $\boldsymbol{x}$ 进行集成分类. 如果能获得 $\boldsymbol{x}$ 的

标记, 则利用 $\boldsymbol{x}$ 对簇集合进行调整. 具体是: 首先, 调整参与对 $\boldsymbol{x}$ 分类的簇的重要性, 重要性太低的簇会被删除; 然后计算离 $\boldsymbol{x}$ 最近的簇及其距离: 若距离小于该簇的半径, 则将 $\boldsymbol{x}$ 加入该簇并更新该簇的聚类特征; 若距离大于该簇的半径或者是 $\boldsymbol{x}$ 的标记与簇的标记不一致, 则以 $\boldsymbol{x}$ 为成员建立一个新簇; 然后, 若簇集合的规模小于最大容量, 则将新簇加入簇集合; 否则, 先将已有簇合并后, 再将新簇加入.

Zheng 等人[62]于 2021 年提出了一种能处理有重现概念漂移和类别增加的数据流的半监督分类算法 ESCR. 算法先将前 p 个用于初始化的标记样本划分成若干数据块, 然后利用 K-means 将各数据块进行聚类得到 k 个簇, 各簇只存储其聚类特征. 将在各数据块上得到的簇集合作为集成分类器 $\boldsymbol{M}$ 的基分类器. 后续样本被随机地打上标记, ESCR 只对未标记样本进行分类. 当新样本 $\boldsymbol{x}$ 到达时, 利用分类器 $\boldsymbol{M}$ 来判断 $\boldsymbol{x}$ 是否是一个“例外”.

- 若是“例外”, 则将其存入缓存 B 中, 当 B 满时, 则检测 B 中样本是否是新类: 若是新类, 则利用 B 中样本更新 $\boldsymbol{M}$; 若不是新类, 则由 $\boldsymbol{M}$ 对其分类并给出分类置信度;
- 若不是“例外”且 $\boldsymbol{x}$ 是标记样本, 则 $\boldsymbol{x}$ 用于训练新分类器, 用于更新 $\boldsymbol{M}$;
- 若不是“例外”且 $\boldsymbol{x}$ 是未标记样本, 则由 $\boldsymbol{M}$ 对 $\boldsymbol{x}$ 给出分类结果和分类置信度, 将 $\boldsymbol{x}$ 和其分类置信度分别存入分数窗口和样本窗口, 再基于 Jensen-Shannon 散度的变化检测概念漂移. 若没检测到重现概念只检测到概念漂移, 则从样本窗口和 B 中不是“例外”的样本中挑选样本, 训练新的分类器用于更新 $\boldsymbol{M}$.

总结起来: 从算法流程来看, SmSCluster[55], ReaSC[56], LabelStream[57]和 SEClass[16]是同一类算法, 它们的共同点在于采用集成模型, 在当前数据块上训练的分类器不断加入集成模型, 以确保对当前概念(包括新类)的适应. 由于使用了集成分类机制, 分类器池的大小会影响分类模型对新概念的适应能力. SPASC[15], SSCLSR[58]和 SCBELS[60]是同一类算法, 它们共同点在于实施显式概念漂移检测, 根据检测结果更新分类器池中的分类器. DISSFCM[59]和 ORSSL-MC[61]是都是单模型的算法, 它们的共同特点在于, 在初始化数据上建立的模型被后续样本不断更新. 为了既能实现对概念漂移的适应还能实现对新类别的适应, ESCR[62]则使用了更复杂的机制. 从 2008 年提出的 SmSCluster 到 2021 年提出的 ESCR, 分类方法基本没变, 但在聚簇的更新方法上, 基于前人的经验, 后面算法对聚簇的更新处理更加精细, 以更准确地刻画数据的分布. 因此可以想见, 未来可能会有更多的对聚簇更精细处理的方法出现.

3.2 基于决策树的算法

该类算法以决策树作为分类模型或者是集成模型的基分类器, 主要通过对树结构的调整或者是集成模型的更新来显式或者隐式地适应带概念漂移的数据流环境. 因为数据流中只有部分样本有标记, 不少算法结合半监督学习策略来扩展有标记样本的数量, 以提高分类模型的泛化能力或使得概念漂移的检测更准确.

李培培等人[63,64]分别于 2010 年和 2012 年提出了基于快速决策树(VFDT)的半监督数据流分类算法 REDLLA 和 SUN. 样本以在线方式到达, 将样本落入决策树的叶子结点, 并根据样本标记更新各节点的相关统计信息. 当决策树接收样本的数量达到一个检测周期时, 使用聚类算法对每个叶子节点进行聚类而形成概念簇. 在每个概念簇中实施标记扩散以增加标记样本的数量, 并更新各叶子节点的相关统计信息. 当叶子节点累积样本的数量达到设定的最小分割阈值时, 根据 Hoeffding 条件, 选择最优属性分裂该叶子节点. 算法计算叶子节点中新簇与历史簇间的距离和各自半径间的关系. 在 SUN 中, 若某个节点的所有叶子结点都检测到发生概念漂移, 则自底向上对决策树进行剪枝. 在 REDLLA 中, 若检测到概念漂移, 同样对决策树进行剪枝, 但叶子节点会保留历史概念簇集信息, 以实现对重现概念的检测.

Nguye 等人[65]于 2011 年提出了一种基于动态树结构的数据流半监督分类算法 CSL-Stream. 样本以在线方式到达. 随着样本的到来, 算法建立一棵在线增量更新的动态决策树. 叶子节点中, 每个训练样本都设置有权重, 算法根据节点的权重值将节点划分为密集节点、过渡节点和稀疏节点. 执行完建树过程, 会判断是否达到 t_p 周期: 若达到, 则对密集节点进行聚类操作. 自第 2 个 t_p 周期起, 每隔 t_p 周期会对树剪枝, 并对树增量更新. 对样本进行分类时, 即找到最近的簇, 且它们之间的距离小于阈值, 则给出标记; 否则, 将该样本视为噪声. 该算法被 Anastasovski 等人[66]用于检测 Web 中的恶意流量.

胡学钢等人[67]于 2017 年提出了一种基于 Tri-training 的数据流半监督集成分类算法 TriTDS. 样本以块的方式到来. 算法在前 k 个数据块上分别训练出 k 个基于 Tri-training 的基分类器来初始化集成分类模型 $\boldsymbol{M}$. 数据块 D^{i+1} 到来后, 利用 $\boldsymbol{M}$ 对 D^{i+1} 中的样本分类. 当获得 D^{i+1} 中部分样本的标记后, 利用 D^{i+1} 依次测试 $\boldsymbol{M}$ 中的分类器, 若某一分类器 $\boldsymbol{T}_r$ 的分类准确率小于阈值, 则将 $\boldsymbol{T}_r$ 在 D^{i+1} 上重新训练. 为保持基分类器的多样性, 用于更新剩下的基分类器 $\boldsymbol{T}_{r+1},\ldots,\boldsymbol{T}_k$ 采用的数据块为 D^{i+2}, 先利用 $\boldsymbol{M}$ 对 D^{i+2} 分类, 再依次测试剩下的基分类器在 D^{i+2} 上的错误率. 如此循环, 直至更新到最后一个分类器.

Wang 等人[17]于 2018 年提出了一种基于 Co-Forest 的数据流半监督分类算法 SCo-Forest. 样本以块的方式到来, 每个数据块中包含标记样本和无标记样本. 算法初始化规模为 m 的集成分类器 $\boldsymbol{M}$ 为空. 每采集到一个数据块, 先使用 ADWIN2 方法检测当前数据块对 $\boldsymbol{M}$ 是否存在概念漂移: 若存在概念漂移, 则将 $\boldsymbol{M}$ 中的某个基分类器移除; 若无概念漂移, 则利用当前数据块中的标记样本训练一棵树, 这样直到 m 个决策树都已训练好. 当后续数据块 D^t 到达后, 使用已经训练好的 m 个决策树在 D^t 上利用 Co-Forest 训练, 得到的集成分类器再用于分类.

Wen 等人[68]于 2020 年提出一种利用迁移学习处理带重现概念漂移的数据流的半监督分类算法 CAPLRD. 样本以块的方式到来. 对于第 1 个数据块, 首先利用标记样本建立一棵 VFDT 树, 并存入分类器池中. 接着, 每采集到一个新的数据块 D^t, 先使用激活分类器对 D^t 进行分类. 当获得 D^t 中部分样本的标记后, 使用基于迁移学习的半监督学习算法 SSLNE 给 D^t 中的无标记样本打上伪标记, 以扩充其有标记样本的数量. 然后利用扩充后的标记样本实施重现概念漂移检测: 若检测到概念漂移, 则利用扩充标记样本后的 D^t 训练一棵 VFDT 树加入分类器池; 若检测到重现概念, 则利用扩充标记样本后的 D^t 更新对应的分类器.

Liu 等人[69]于 2020 年提出一种基于自适应峰值聚类的数据流半监督分类算法 SSCADP. 样本以块的方式到来, 每个数据块中包含标记样本和无标记样本. 算法使用第 1 个数据块中的样本在线训练一棵决策树 $\boldsymbol{T}$. 在线训练过程中, 样本落入叶子节点. 根据样本是否有标记更新相关的统计量. 当落入树中样本的数量满足一个周期时, 自底向上对所有叶子节点中的样本实施半监督学习, 扩充叶子节点中标记样本的数量, 然后基于自适应峰值聚类方法实施概念漂移检测. 算法根据概念漂移检测的结果对树进行剪枝. 若某个叶子节点中样本的数量满足分裂条件, 则对该叶子节点实施分裂. 当后续数据块 D^t 到来时, 用 $\boldsymbol{T}$ 对 D^t 分类, 当 D^t 中部分样本被标记后, 用 D^t 对 $\boldsymbol{T}$ 实施增量训练.

总结起来: REDLLA[63], SUN[64], SSCADP[69]和 CSL-Stream[65]是同一类算法, 这些算法的共同特点是利用树便于实施增量学习的特性, 从而容易使得树能调整以适应新概念. 但当标记样本过少时, 训练得到的树的泛化能力可能较差. TriTDS[67], CAPLRD[68]和 SCo-Forest[17]则都是利用集成学习, 通过集成模型中基分类器的调整, 以适应新概念. 从 2010 年提出的 REDLLA 到 2020 年提出的和 SSCADP, 因为要实施树的增量更新, 分类模型中一般都是使用 VFDT 方法. 在概念漂移的检测方法上, 基于标记样本的检测方法和基于分布结构的检测方法都被使用. 但是, Liu 等人[69]尝试了更精细的概念漂移检测方法.

3.3 基于图的算法

基于图的算法在整个数据流处理过程中维持着一个图结构. 一个节点表示一个样本, 有边的样本之间很相似. 图的更新主要通过添加/删除顶点和边以及更新边的权值来完成, 以适应数据流的不断变化.

Bertini 等人[70]于 2012 年提出了一种基于 K-associated 最优图的数据流半监督分类算法 KAOGINCSS. 样本以块的方式到来, 其中既有标记样本也有未标记样本. 为使用数据块中包含的未标记样本, 算法将 K-associated 最优图算法推广成半监督 K-associated 最优图算法. 在第一个数据块上学习得到一个 K-associated 最优图. 将在新数据块上得到的 K-associated 最优图与前一最优图合并后, 淘汰图中不频繁用于分类的组件, 从而实现对新数据块的学习. 对样本分类时, 计算待分类样本在图中的近邻. 若满足一定条件, 将其与图中的近邻连接, 然后根据贝叶斯理论计算该样本属于其近邻所属组件的概率, 根据最大的概率给出分类结果.

用图来表示样本及样本的相似关系, 可以比较好地发现样本分布的局部特征, 也便于更新. 因此, 基于图的方法与基于聚类的方法有着很多类似之处, 可以相互借鉴而产生新的算法. 该类算法也面临两难——当顶

点较多时, 图操作比较费时; 但顶点过少时, 图结构又很难准确地体现样本的分布. 从目前来看, 此类算法较少见.

3.4 基于SVM的算法

基于 SVM 的分类模型都是以分块方式来处理数据流. SSL-E[71]基于平滑假设, 可求出分类函数的闭式解. RK-TS^{3}VM[72]和 4SFD[22]虽然使用了支持向量机, 但并没有利用支持向量机的一些优点, 比如支持向量来进行概念漂移检测或分类模型更新. 因此从本质上来讲, 这里的支持向量机可以替换成其他分类器.

针对演化数据流分类, Jia 等人[71]于 2009 年提出一种基于平滑演化假设的数据流半监督分类算法 SSL-E. 样本以块的方式到来, 其中未标记样本的类别被标记为 0. 根据平滑演化假设, 算法将在当前数据块 D^t 上要学习的分类器与在前一时间点已学习到的分类器在再生核希尔伯特空间中的距离作为正则项, 从而使得历史上学到的分类器可用于帮助对当前数据块的学习. 算法最终得到闭式解, 用于对未标记样本的分类.

Zhang 等人[72]于 2009 年提出了一种结合关系 K-means(RK)[73]学习模型和迁移半监督支持向量机(TS^{3}VM)的数据流半监督分类算法 RK-TS^{3}VM. 样本以块的方式到来, 其中既有标记样本也有未标记样本. 考虑到与下一个数据块 D^{t+1} 之间的关系, 算法假定当前数据块 D^t 包含有属于两种概念的样本, 而将其划分成 4 种不同类型的样本子集. 学习过程分为 4 步: 第 1 步使用 RK 算法学习类型 I 和类型 IV 中的样本, 获得 k 个聚类中心 $\boldsymbol{\mu}=\{\mu_1,\ldots,\mu_k\}$; 第 2 步计算类型 I–类型 III 中样本与 k 个聚类中心的距离; 第 3 步将这 k 个距离作为新的属性加入样本原来的属性中, 扩充类型 I–类型 III 中样本的属性; 第 4 步采用 TS^{3}VM 学习加入了属性被扩充后的类型 I–类型 III 中样本, 得到分类模型 $\boldsymbol{F}$. μ和 $\boldsymbol{F}$ 组合成分类器对 D^{t+1} 中的样本分类. 当 D^{t+1} 中部分样本的标记获得之后, 又开始新一轮学习, 旧模型被丢弃.

Hu 等人[22]于 2017 年提出了一种基于自适应故障检测的数据流半监督分类算法 4SFD. 样本以块的方式到来. 算法首先使用初始有标记样本训练 SVM 分类器并用于对后续样本分类, 同时维持着一个用于训练该分类器的样本集合 X^t. 接收到数据块 D^t 后, 就计算集合 D^t 和集合 $D^t\cup X^t$ 之间的α型表面体积之比[74], 以识别渐变概念漂移和突变概念漂移. 当发生渐变概念漂移时, 算法使用类似于 COMPOSE[44,45]中的样本提取方法生成新的样本集合训练新的 SVM 分类器. 当检测到突变概念漂移时, 在不同特征组合形成的标记样本集合上训练不同的分类器. 计算这些分类器在 D^t 上的 3 个分类性能指标: 有标记样本上的分类精度、无标记样本上的分类置信度以及无标记样本上分类结果中各类的紧凑和分离程度. 计算每种特征组合的得分以选择得分最高者, 该特征组合形成的标记样本被用于训练新的 SVM 分类器. 同时, 历史分类器被丢弃.

3.5 基于神经网络的算法

基于神经网络的算法主要通过调整网络参数来更新模型以适应新的概念. 由于神经网络的训练需要大量的样本, 因此, 基于神经网络的方法对于渐变或者增量的概念漂移以及类别增量会有比较好的效果. 据作者所知, 目前还未见到有将神经网络用于突变概念漂移的数据流分类.

Li 等人[75]于 2019 年提出了一种基于深度神经网络的数据流半监督分类算法 ISLSD. 样本以块的方式到来, 其中既有标记样本也有未标记样本. 分类模型由生成网络、判别结构和桥接器组堆叠而成. 在训练第 1 个数据块时, 样本(包括有标记样本和无标记样本)分别被送至生成网络和基于半监督哈希的判别结构中. 生成网络实现特征提取, 判别式结构将有标记和无标记样本原始特征空间映射到海明空间, 以计算样本之间的相似性. 最后, 用桥接器来加强生成网络参数与判别式参数之间的相关性. 最后建立两者的混合目标函数对其进行优化. 模型训练完成后, 当下一个数据块 D^t 到来时, 将样本输入到生成网络中, 输出其特征表示, 再根据 softmax 函数进行分类. 然后, 利用 D^t 更新分类模型.

Silva等人[76]于 2019 年提出了使用带遗忘参数的在线半监督弹性极限学习机来处理概念漂移数据流分类问题的算法 SSOE-FP-ELM. 样本以块的方式到来, 其中既有标记样本也有未标记样本. 在初始化阶段, 该算法使用数据块中的样本计算特征图 H_0. 数据块 D^t 被极限机分类后, 获知 D^t 中部分样本的标记, 算法采用遗忘程度因子减弱已学习过的样本对后续学习的影响, 同时加入新的学习样本, 以实现对新样本学习. 同时, 遗忘

部分已学习的知识, 从而更快地适应 D^t 中包含的概念. 遗忘程度因子的计算, 采用了监督和无监督相结合的办法.

3.6 其他算法

本节介绍基于其他分类模型的算法, 这些分类模型包括 KNN[52]、标记传播[46]等, 或者是两种方法的混合[47,53], 以及不局限于具体分类器的框架性方法[54]. 这里面还包括针对 EVL 环境的算法[44−46]. 另外, 带概念漂移的数据流半监督分类算法还被用于文本过滤[77]和推荐[24].

Dyer 等人[44,45]于 2012 年和 2014 年提出了一种基于计算几何的数据流半监督分类算法 COMPOSE. 该方法基于一个假设: 持续改变的数据分布在两个相邻时段会有重叠. 为适应 EVL 环境, 算法首先获取一个所有样本都有标记的数据块 L 作为初始化训练集; 随后, 每当一个未标记的数据块 U^t 到来, 通过调用现有的半监督学习算法对 D^t 中的样本分类; 然后, 基于 $D^t \cup L$ 为每一个类别创建一个 α 形状边界(轮廓包络); 接着, 让每个类的轮廓持续不断收缩直至预设比率, 从而获得每个类分布的核心支持区域; 然后, 将从每个类的核心支持区域提取的样本替换 L, 作为下一个阶段的标记样本集.

Ferreira 等人[46]于 2019 年针对 EVL 环境提出了基于密度自适应的数据流半监督分类算法 AMANDA. 与 COMPOSE 类似, 该算法由开始、学习、分类、加权和过滤这 5 部分组成. 算法维持着一个有标记的训练样本集合. 在开始阶段, 算法使用包含各类有标记的样本来初始化带标记的训练样本集合. 在学习阶段, 一个分类器在带标记的训练样本集合上被训练并被用于分类. 当前数据块 D^t 到来后, 在加权阶段, 基于核密度估计算法计算 D^t 和带标记的训练样本组成的集合中每个样本的密度. 在过滤阶段, 算法过滤掉部分低密度非核心区域的样本. 剩下的高密度核心区域中的样本作为新的带标记训练样本, 被用于训练新的分类器, 以分类即将到来的数据块 D^{t+1} 中的样本.

Zhang 等人[47]于 2010 年提出了一种基于分类模型和聚簇模型集成的概念漂移数据流半监督分类算法 ECU. 样本以块的方式到来, 无标记的块和有标记的块随机到来. 对无标记的块使用 K-means 算法构建聚簇模型; 对有标记的块使用决策树训练分类模型. 在集成池中有标记簇的集合和聚簇模型中无标记簇的集合中, 以簇质心为顶点构成一个图, 利用标记传播算法为无标记簇分配伪标记. 在分类阶段, 基于簇的基分类器通过距离样本 $\boldsymbol{x}$ 最近的簇对应的标记来实现对 $\boldsymbol{x}$ 的分类. 池中所有基分类器的分类结果的加权为最终分类结果. 在集成池更新阶段, 如果当前数据块 D^t 分类结束后全部样本被标记, 则在 D^t 上新建一个分类模型和一个聚簇集模型; 否则, 只新建一个聚簇模型. 新建的分类/聚簇模型替换掉池中最旧的模型. 算法再根据池中每个模型和 D^t 上新建模型的一致性关系, 为每个模型重新计算权值.

Parker 等人[53]于 2015 年提出了一种监测和跟踪新类的数据流半监督分类算法 SluiceBox. 在初始化阶段, 算法使用指定数量的标记样本分别训练一个分类模型和一个聚类模型. 样本以在线的方式到来. 算法每接收一个样本 $\boldsymbol{x}^t$, 若该样本有标记, 则分别更新聚类模型和分类模型; 否则, 使用分类模型和聚类模型分别对 $\boldsymbol{x}^t$ 分类, 并根据距离该样本最近簇的标记来判断该样本是否为新类. 不是新类的样本通过到最近簇的距离和簇半径之间的关系来判断是否为异常样本, 异常样本被添加到缓存集合中. 缓存集合中的每个异常样本被分类并被用于更新聚簇模型.

Feng 等人[52]于 2016 年提出了一种基于自表示选择策略的数据流半监督分类算法 IS^{3}RS. 算法由决策、自表示选择和 Co-training 这 3 部分组成. 样本以块的方式到来. 算法维持着一个有标记样本集合. 在决策部分, 算法在有标记样本集合上应用 K 近邻分类器对当前数据块 D^t 中的样本进行分类. 在自表示选择阶段, 数据块 D^t 通过自表示学习获得一个可以有效描述整个数据块的样本子集 X^t. 在 Co-training 阶段, X^t 被划分成多个子集用于训练多个分类器. 算法在这些分类器上利用 Co-training 对 X^t 中的样本进行分类, 给出分类置信度最高的分类结果, 获得伪标记的样本被添加到标记样本集合中用于扩充该集合.

Khezri 等人[54]于 2021 年提出了一种不局限于具体分类器的框架性方法 SSE-PBS. 算法维护一个分类器池, 当数据块 D^t 到来时, 由分类器池中的分类器对其实施带权集成投票. 分类模型的更新包括两个步骤.

- 第 1 步, 当获得 D^t 中部分样本的标记后, 利用这些标记样本计算分类器池中各分类器的权值. 在 D^t

上训练分类器 h_{new}, 计算其权值并加入池. 若池满, 则用 h_{new} 替换池中权值最小的分类器;

- 第 2 步, 利用池中的分类器计算 D^t 中样本的伪标记和分类置信度. 采用基于性能的标准挑选伪标记样本, 再增量更新分类器池中的各分类器.

3.7 基于主动学习的数据流半监督分类

在基于主动学习的数据流半监督分类算法中, 需要查询标记的样本通过选择性一些低置信度[78−80]、难分类[81]、不可靠[82,83]、少数类样本[84]或使集成分类最不一致[85]的样本而得到. 实施主动学习的关键在于要确保被查询标记的样本能体现数据本来的分布. 由于主动学习是一个独立的研究领域, 需要较大的篇幅才能对现有的基于主动学习的数据流半监督分类算法进行详细地归类和总结. 因此, 本节仅介绍部分算法中满足什么样条件的样本的标记可被查询.

Haque 等人提出的 SAND[78], ECHO[79]算法和李南等人提出的 CASD[81]算法都是主动学习领域中基于聚簇的数据流半监督分类算法. 在 SAND, ECHO 算法中, 关联度和纯度被用于计算模型对样本 $\boldsymbol{x}$ 分类的置信度. 当检测到概念漂移后, 采取主动学习策略, 有选择地查询那些分类置信度小于指定阈值的样本的标记. 大于阈值的样本则使用预测标记作为样本标记, 将获得标记的样本集合用于训练新的分类器. 关联度通过计算样本 $\boldsymbol{x}$ 到最近的簇的距离和簇的半径之间的差值得到, 纯度则表示距离样本 $\boldsymbol{x}$ 最近的簇中多数类的比例. 在 CASD 算法中, 样本 $\boldsymbol{x}$ 被分类后, 根据样本 $\boldsymbol{x}$ 到任意基分类器对应簇的集合中最近的簇的距离和簇的半径之间的关系来判断该样本是否被覆盖, 没有被覆盖的样本和落在簇边界的样本被查询标记.

Sriwatanasakdi 等人[80]提出的 A2INGD 算法基于样本到图中最近邻顶点的距离与阈值之间的关系检测新类, 将当前数据块 D^t 划分为新类和已有类两个集合: 对于已有类集合, 该集合中每个样本的分类置信度通过计算距离该样本最近和次近顶点之间的类分布差异得到, 置信度小于指定阈值的样本的标记被查询; 新类集合中所有样本的标记被查询.

Masud 等人[82]提出的算法对两类样本进行标记查询: 第 1 类是落在集成分类器的决策边界外的样本, 第 2 类是多数投票的比例小于指定阈值的样本. Žliobaitė 等人[83]基于不确定性、样本标记的难度以及搜索空间的随机性, 提出了 3 种选择查询样本标记的策略.

Arabmakki 等人[84]提出的 RLS-SOM 中, 算法在每个时间段都维持着一个在当前数据块 D^t 中少数类样本对应的集合 *MinS* 和一个支持向量集合 *SVs*. D^t 中距离 *MinS* 和 *SVs* 中的样本最近的样本被查询标记, 形成一个危险集合 CS_A. 若 CS_A 中仅包含多数类的样本, 则 SOM 算法被用于产生该数据块的拓扑结构. 拓扑结构中, 每个节点周围的样本被打上标记, 并将被打上标记的少数类样本添加到集合 *Temp* 中. 最后, 在 CS_A 和 *Temp* 形成的训练样本集上训练新的基于决策树的分类器, 并将其添加到分类器池中.

Zhu 等人提出的 MV[85]算法维持着最近数据块上训练得到的 k–1 个分类器. 当前数据块 D^t 中每个通过集成分类得到标记的无标记样本 I_x 都会与所有有标记样本构成一个评价集合 Λ_x. 集成分类器中的每个基分类器基于对评价集合 Λ_x 中的每个样本的预测结果会产生对整个评价集合 Λ_x 的一个方差值, 进而计算每个分类器的权值和方差值而得到集成模型在每个样本上的方差值. 最大方差值对应的样本被查询标记并被添加到有标记样本集合中.

4 主动概念漂移检测方法

在开放环境下, 数据的分布常因环境变化、设备损耗等多种因素的影响而随时间发生变化(概念漂移现象), 有效地检测出这种变化进而及时调整模型, 将使模型更好地适应新的数据流环境. 显式的概念检测使得学习过的历史概念得到保留, 还能使得分类模型更新更有针对性, 从而避免一些无意义的模型更新, 提高了数据的管理效率. 针对带概念漂移的数据流的半监督分类中的概念漂移检测, 学界已经提出了两类方法. 一类办法是利用有标记样本使用监督情形下的检测方法来检测概念漂移, 但这种方法忽略了数量众多的无标记样本. 同时, 在半监督条件下, 由于标注样本不多, 使得使用分类准确率来检测概念漂移这一通常做法也变得不太可行; 第 2 类办法是将标记样本和无标记样本都利用起来, 因此, 数据流半监督分类时大多都是采用第 2

类检测方法.

在 Wang 等人[17]提出 SCo-Forest 中, ADWIN2[86]算法被用于判断两个子窗口 W_0 和 W_1 之间的样本均值 $|u_{W_0}-u_{W_1}|$ 差异是否大于指定阈值 ε_{cut} 来检测概念漂移. 其中, $W=W_0\cup W_1$, $\varepsilon_{cut}=\sqrt{1/2m*\ln(4|W|/\delta)}$, $m=\dfrac{1}{1/|W_0|+1/|W_1|}$ 表示两个窗口的调和平均数, δ表示用户定义的置信度参数.

在 Hu 等人[22]提出的 4SFD 中, 算法维持着一个用于训练分类器的样本集合 X^t. 每接收一个数据块 D^t, 就基于α型表面重构方法[74]来计算集合 D^t 和集合 $D^t\cup X^t$ 之间的α型表面体积之比 $R=V'/V$. 该比值 R 和两个阈值 Th_n 和 Th_g 之间的大小关系分别被用于识别渐变概念漂移和突变概念漂移.

李培培等人[63,64]提出的 REDLLA 和 SUN 每隔一个检测周期采用自底向上搜索策略定位每个叶子节点, 并对叶子节点中的样本进行聚类. 其中: SUN 采用 *K*-modes, REDLLA 则采用 *K*-means. 计算上一个检测周期形成的概念簇集 M_{hist}的半径 r_{hist}和当前检测周期形成的概念簇集 M_{new}的半径 r_{new}. 两个概念簇集的距离为 $dist$. 当 $dist\geqslant r_{hist}+r_{new}$ 时, 则判断该叶子节点发生突变概念漂移. 另外, 考虑到样本类别分布 $P(Y|X)$变化引起的概念漂移, 算法还利用了概念簇中样本的标记信息. 若新概念簇集 M_{new} 中的类别分布与历史概念簇集 M_{hist} 中的类别分布完全相反, 或数据类别分布的变化率达 95%以上, 则也判断该叶子节点发生概念突变. 在 REDLLA 算法中, 每个节点会保存所有之前检测周期中形成的历史概念簇集合, 采用与 SUN 相同的检测机制对上个检测周期 M_{last}和当前检测周期 M_{new}的概念簇集进行概念漂移检测. 若检测到发生突变概念漂移, 则进一步判断当前概念是否为重现概念. Liu 等人[69]在此基础上提出了基于自适应峰值聚类的概念漂移检测方法, 其特点在于更进一步地考虑聚簇数量和分布密度的变化.

王海燕等人[87]提出的 SSC 通过判断当前数据块 D^t 和用于构建集成模型的每个分类器对应的数据块 $D^i(i<t)$之间的基于簇的语义距离 $dist(G^t,G^i)=\left(1/\sum_{h=1}^{r}|g_h^t|\right)\sum_{p=1}^{r}(g_p^t/sim(g_p^t,G^i))$ 是否大于给定阈值来检测概念漂移. G^t或$G^i=\{g_1^t,g_2^t,...,g_p^t,...,g_r^t\}$或$\{g_1^i,g_2^i,...,g_q^i,...,g_r^i\}$ 表示在 D^t 或 D^i 上聚类得到的簇. g_p^t/g_q^i, $sim(g_p^t,G^i)=\max\limits_{g_q^i\in G^i}|g_q^i|/\left(\sum_{h=1}^{r}|g_h^t|*sim(g_p^t,g_q^i)\right)$. 其中, $sim(g_p^t,g_q^i)$ 表示簇间的相似性, 即两个簇的半径和与簇质心之间距离的比值.

Hosseini 等人[15]提出的 SPASC 算法假设分类器池中每个基分类器描述一个概念, 用半监督贝叶斯方法计算每个基分类器 C_j 与当前数据块 D^t 之间的相似性: $\sum_{i=1}^{n}\log_2 P(y_i^t|x_i^t,C_j)+\sum_{i=1}^{m}\log_2 P(C_j|x_i^t)$. 其中, 第 1 项考虑了有标记样本和基分类器 C_j 之间的相似性, 第 2 项则考虑了所有样本和 C_j 之间的相似性. n 和 m 分别表示 D^t 中有标记样本和全部样本的数量, $P(y_i^t|x_i^t,C_j)$ 表示 C_j 对样本 x_i^t 的预测类别属于其真实类别 y_i^t 的概率, $P(C_j|x_i^t)$ 表示样本 x_i^t 来自 C_j 对应概念的概率. 相似性超过阈值就意味着 D^t 带来了重现概念, 否则是产生了新概念. 但是在 SPASC 中, 数据块 D^t 被压缩成 D^t 中所有样本的均值, 忽略了 D^t 中样本分布的细节特征. 因此, Wen 等人[60]提出使用 Gao 等人[88]提出的局部结构映射策略来计算概率 $P(C_j|x_i^t)$, 使得概念漂移检测更准确.

Kim 等人[89]提出了一种检测概念漂移算法. 样本 $\boldsymbol{x}_i$ 在线到达.该算法定义两个滑动窗口: 参考窗口 W_{ref} 和检测窗口 W_{det}. 针对样本 $\boldsymbol{x}_i$, 定义变量 $X(\boldsymbol{x}_i)$等于分类器对 $\boldsymbol{x}_i$ 的预测类别向量和分类置信度向量之间的距离.算法通过判断 W_{det} 中 $X(\boldsymbol{x}_i)$的均值 m_{det} 是否超过 W_{ref} 中 $X(\boldsymbol{x}_i)$的均值 m_{ref} 的置信区间上界来检测概念漂移. 即 $m_{\text{det}}\geqslant m_{\text{ref}}+(z_\alpha\times s_{\text{ref}})/\sqrt{n}$, 其中: z_α表示参数, s_{ref}表示 W_{ref}上 $X(x_i)$的标准方差, n 表示参考窗口中的样本数量. 若检测到概念漂移, 丢弃现有分类模型, 并在包含少量有标记样本的检测窗口上训练一个新的分类器. 作者根据参考窗口的情况提出了 3 种概念漂移检测算法: 固定参考窗口(FRW)、移动参考窗口(MRW)和集成参考窗口(ERW).

Tan 等人[90]提出的 DensityEst 维持着一个静态评估器 *s_clf* 和一个增量评估器 *i_clf*. 当数据块 D^t 到来后, 算法首先从 D^t 的标记样本和未标记样本中提取可靠标记样本. 用可靠标记样本训练静态评估器和增量评估器. 在静态评估器得到的后验概率分布上, 使用核密度估计方法来预测增量评估器得到的后验概率分布的密

度ρ, 该密度通过误差率函数 $erf(p)=\frac{1}{\sqrt{\pi}}\int_0^{\rho}e^{-t^2}dt$ 得到相应的扩散值ε. 若该值小于指定阈值φ, 则表示从 D^t 中检测到了概念漂移.

总结起来: 第 2 类办法中, 有的利用空间距离[17,22,63,64,87], 有的利用概率[15,60], 有的则利用统计量[89,90]. 由于标记样本少, 大多都是基于滑动窗口来检测. 因此, 窗口大小的设置会影响检测的准确率. 越准确的检测计算会越复杂, 概念漂移的检测会越慢, 这显然又不符合数据流分类的速度需求.

5 实 验

第 3 节以分类器类型为线索对数据流半监督分类的研究进展进行了详细的归类、分析和总结, 概括了各类算法的特点. 为了对概念漂移数据流半监督分类算法的其他归类(在线和分块、有无概念漂移以及模型数量)的特点有更加全面的认识和理解. 本节给出了具体的实验方案, 依据实验方案生成包含不同概念类型的数据集, 并对多个算法在这些数据集上的实验结果进行全面分析.

5.1 方案设计

由于数据流半监督分类领域涉及到的算法比较多, 本文无法对所有算法进行对比实验. 因此, 本节将在一些被广泛使用的真实和人工数据集上对部分算法: SPASC, ReaSC, SmSCluster, SEClass, SUN 和 REDLLA 进行多方面的实验和分析. 这些算法涵盖了在线和分块处理数据流的算法、主动概念漂移检测(包括重现概念漂移检测)和不进行概念漂移检测的算法、基于单模型和多模型的算法.

不同应用场景下产生的数据流中, 数据的分布随时间变化的方式不同. 为分析算法对各种概念漂移数据流分类场景的适应能力, 本次实验分别使用 4 个被广泛使用的真实数据集和 8 个人工数据集. 这些数据集已经在综述文献[39,41]中被详细地介绍和归类. 人工数据集包括在 MOA[91]上利用 4 种类型的 Streams Generators (Sea, Agrawal, Sine 和 Hyperplane)生成的 7 个人工数据集和 1 个用于模拟高斯分布随时间变化的数据集. 这些数据集涵盖了突变、渐变、增量、重现等多种概念漂移类型. 真实数据集则用于表示更加复杂的实际应用场景.

在本次实验中, 为了模拟半监督环境, 假定部分样本被分类结束后, 其真实标记可以被算法获得. 有标记样本的比例对算法的性能具有一定影响, 因此, 本文设置了不同的样本标记比例[5%,10%,20%,30%].

在基于多模型的算法中, 分类器池中基分类器的数量对样本分类所做的贡献因算法不同而存在差异. 分类器池越大, 包含不同概念的基分类器可能越多. 采用集成投票表决进行分类的 ReaSC 和 SmSCluster 算法可能会受到非当前概念的分类器的影响而表现出较差的分类性能. 采用加权集成投票表决的 SEClass 算法利用权值削弱了非当前概念的分类器的影响, 可能呈现较好的性能. 采用最大权值进行分类的 SAPSC 算法在权值调整正确的情况下可能会完全避免非当前概念的分类器的影响. 因此, 本文对分类器池的不同大小进行了实验, 用于分析其产生的影响.

不同的算法具有不同的参数. SPASC, ReaSC, SmSCluster 和 SEClass 等算法都属于基于聚簇的数据流半监督分类算法, 且都属于多模型算法. 该类算法的主要参数是分类器池的大小和簇的数量. 簇参数的设置因数据集的不同而不同. 在样本的标记比例默认值设置为 20%条件下, 为了公平比较, 对于每个数据集, 在[1–100]取值区间内, 以 5 为步长分别设置簇的参数为[5,10,…,95,100], 在所有簇参数下获得的最高准确率对应的簇参数值作为这 4 个算法在该数据集上各自的簇参数设置. 为了尽可能使分类器池能够容下每一个概念对应的分类器, 对于 Agrawal/Sea/Sine-(abr/gra)数据集, 分类器池的大小默认设置为概念数量的 2 倍, 算法在其他数据集上的分类器池的大小默认设置为 10. SUN 和 REDLLA 算法属于基于决策树的数据流半监督分类算法, 且都属于单模型算法, 该类算法的性能主要受有标记样本的比例的影响. 其他参数(SEClass 中的同质性参数 e, SmSCluster 算法的注入概率ρ)对这些算法的影响较小, 因此这些参数的设置均遵循原文中的设置.参数设置的详细信息见表 2.

表 2 数据集和参数设置

	数据集	数量	维度	类别	块的大小	簇的数量	池的大小
真实	Weather	18 159	8	2	360	95	10
	Electricity	45 312	8	2	1 000	75	10
	Forest Cover type	581 012	54	7	1 000	95	10
	Poker Hand	829 201	10	10	1 000	100	10
人工	Sea-abr/gra	80 000 / 115 000	3	2	1 000	50	8
	Sine-abr/gra	80 000 / 115 000	4	2	1 000	100	8
	Agrawal-abr/gra	80 000 / 115 000	9	2	1 000	45	8
	Hyperplane	80 000	10	2	1 000	85	10
	Gaussian	6 000	2	4	300	100	10

5.2 数据集

在真实数据集中,

- Nebraska Weather 数据集来源于 Bellevue, Nebraska 的 Offutt 空军基地, 由于其具有 50 年(1949–1999)的时间跨度和多样化的天气模式, 使其成为长期降水分类/预测漂移问题, 目标是预测是否会在某一天下雨. 由于时间跨度大, 肯定存在概念漂移;
- Electricity 是一个被广泛使用的真实数据集, 是从澳大利亚新南威尔士州电力市场收集而来. 类标记表示这一天相对于过去 24 小时移动平均值的价格变化(UP 或 DOWN);
- Forest Cover type 数据集包含位于科罗拉多州北部罗斯福国家森林的 4 个较宽区域内的森林覆盖类型信息, 用于描述 7 种可能的森林覆盖类型之一, 是数据流分类领域中使用最广泛的数据之一;
- Poker Hand 数据集包含 829 201 个样本, 每个样本(hand)由从 52 张标准牌组中抽出的 5 张扑克牌组成, 每张扑克牌由两个属性描述(花色和牌的大小)描述. 该数据集包含 10 个类别(0–9), 分别对应 Nothing in hand, One pair, Two pairs, Three of a kind, Straight, Flush, Full house, Four of a kind, Straight flush 和 Royal flush.

在人工数据集中,

- Sea 数据集包含 3 个属性, 其中只有前两个属性相关, 这 3 个属性的值都在 0–10 之间. 数据集被划分为 4 个具有不同概念的块, 在每个块中, 分类函数为 $f_1+f_2\leqslant\theta$, 其中, f_1 和 f_2 表示前两个属性, θ表示阈值, 阈值设置为 8, 9, 7, 9.5, 分别对应于 4 个概念, 表示为 C_1, C_2, C_3, C_4. 生成的数据流中概念变化顺序为 C_1-C_2-C_3-C_4-C_1-C_2-C_3-C_4;
- Sine 数据集在二维空间中的决策面为 $f(x)=x_2-a\sin(bx_1+\theta)+c=0$. 该数据集可以生成 4 个概念.
  - 当 $a=b=1$, $\theta=c=0$ 时, 概念 1: $f(x)<0$ 表示正类; 概念 2: $f(x)\geqslant 0$ 表示正类;
  - 当 $a=0.3$, $b=3\pi$, $\theta=0$, $c=0.5$ 时, 概念 3: $f(x)<0$ 表示正类; 概念 4: $f(x)\geqslant 0$ 表示正类.

  生成的数据流中, 概念变化顺序为: C_1-C_2-C_3-C_4-C_1-C_2-C_3-C_4;
- Agrawal 生成器生成一个包含 9 个属性的数据流, 6 个数值属性和 3 个分类属性的不同组合产生了 10 个不同类型的分类函数, 这些函数决定了贷款是否应该获得批准. 10 个函数对应于 10 个概念(C_1,C_2,…,C_{10}). 生成的数据流中概念变化顺序为 C_1-C_2-C_5-C_6-C_1-C_2-C_5-C_6. 上述生成器中, 每个生成器分别生成包含突变(XX-abr)和渐变(XX-gra)概念漂移类型的两种数据流, 其中, 每个概念包含 10 000 个样本. 在渐变(XX-gra)概念漂移数据流中, 相邻的两个概念之间有 5 000 个样本用于模拟一个概念逐渐过渡为另一个概念的变化过程;
- Hyperplane 数据集用于模拟增量概念漂移. d=10 维空间中的超平面是满足 $\sum_{i=1}^{d} w_i x_i = \frac{1}{2}\sum_{i=1}^{d} w_i$ 的点 $\boldsymbol{x}$ 的集合, 其中: x_i 表示样本 $\boldsymbol{x}$ 的第 i 维属性值, 满足 $\sum_{i=1}^{d} w_i x_i \geqslant w_0$ 的样本为正类, 否则为负类. 通过改变属性权重的大小 $w_i=w_i+d\sigma$来改变超平面的位置和方向, 其中, σ表示变化方向被反转的概率 10%, d 表示变化的幅度 0.1;

- Gaussian Drift 数据集是一个包含 4 个高斯分布的人工数据集, 这 4 个高斯分布的均值和方差根据参数方程不断变化[92], 该数据集被用于模拟增量概念漂移数据流分类场景. 在本文中, 该数据集被直接当作一个包含 4 个类别样本的数据集, 而不考虑类别数量的变化.

算法随机标注一些样本的标记来模拟半监督环境, 为了缓解这种随机性对算法的分类结果产生特殊性和不稳定性等影响, 本次实验使用 MOA 生成的每个人工数据集都包含相同概念顺序的 10 份数据, 在真实数据集上则随机改变有标记样本的位置 10 次. 各取 10 次结果的平均值作为各自的最终分类结果.

5.3 结果与分析

为了对算法进行全面地分析和比较, 本节从分块准确率、累计准确率、标记比例和分类器池大小的影响等 4 个方面对各算法进行实验, 并给出了详细的对比分析.

5.3.1 累积准确率

为了比较各个算法的性能, 基于每个数据集上的 10 次随机实验, 本文统计了 10 次随机实验中算法累计准确率的标准差(Std), 以比较各算法的稳定性, 并对各算法的累计准确率(Acc)进行了 95%置信水平的成对 t 检验, 以比较各算法在累计准确率上的高低. 如表 3 所示, 表中的每个算法均与其右侧所有算法分别在 12 个数据集上进行比较, 以实现 6 个算法的两两比较. 每个算法对应一个赢/输比较的标识符(◆/◇, ●/○, ★/☆, ■/□, ▲/△). 以◆/◇为例, 表中每个累计准确率后面的实心图形◆表示 ReaSC 算法在相应行对应的数据集上的累计准确率 *Acc* 显著优于相应列对应的对比算法, 即 ReaSC 获胜(win); 空心图形◇与实心图形◆表示的含义相反, 即 Reasc 失败(loss); 若 ReaSC 算法右侧所在列的实验结果中没有◆/◇, 则表示 ReaSC 算法和相应列对应的对比算法在相应行对应的数据集上的累计准确率 *Acc* 之间没有显著差异, 即平局(tie). 两两比较使得每个算法可与其他 5 个算法在 12 个数据集上产生 60 次输、赢或平局的比较结果. Win-tie-loss (Acc)对应累计准确率的输、赢或平局的统计结果. 算法还以相同的成对比较方式在标准差上进行两两比较, 在比较过程中, 标准差小的算法较稳定, 即在性能方面胜出(win). Win-loss (Std)对应每个算法和其他 5 个算法在 12 个数据集上进行 60 次标准差比较的输/赢的统计结果.

从表 3 中的实验结果可以看出: 对于基于聚簇的算法, SEClass 算法整体性能表现最佳, 其次是 ReaSC 和 SmSCluster, 再次是 SPASC; 对于基于决策树的算法, REDLLA 比 SUN 略好; 相比于基于聚簇的算法, 基于决策树的算法在累计准确率和标准差上的表现稍微差一些. 可能存在多种原因导致此现象, 比如分类模型的差异、数据在线和分块处理之间的差异、单模型和集成模型之间的差异以及概念漂移检测的方法不同等.

从表 3 中的实验结果还可以看到: 尽管 SPASC, REDLLA 和 SUN 都主动地检测概念漂移, 但相比于 SEClass 和 ReaSC, 它们的累计准确率都要差一些. 这是因为主动概念漂移检测的优势主要在于节省对新概念的学习时间. 当检测到重现概念之后, 从分类器池或者存储的结构中提取出适合对重现概念分类的部分以实现对新概念的分类, 而不主动进行概念漂移检测的算法通常是对新概念重新学习. 两者相比较, 由于分类器池中的分类器都不会被增量更新, 所以主动概念漂移检测本身并不一定能带来累计分类准确率的提升. 因此, 未来基于重现概念漂移检测的算法主要要解决两个方面的问题: 第一是如何更准确地识别概念; 第二是如何利用属于同一概念的样本实现对属于同一概念的分类器的增量学习.

另外, 从 Hyperplane 数据集的实验结果中可以发现: ReaSC, SmSCluster, REDLLA 和 SUN 算法的分类准确率较高, SEClass 和 SPASC 算法分类准确率较差. 这种现象的原因在于: 在增量概念漂移数据流上, 由于数据分布的变化不显著, SEClass 和 SPASC 算法根据检测到的概念挑选分类器的做法并不奏效; 而对于 Gaussian 数据集情况相反, SEClass 和 SPASC 算法则表现出较好的分类准确率. 这主要是由于数据块的分布变化比较大, 适合 SEClass 和 SPASC 算法挑选分类器.

对于真实数据集, 从表3中发现, 不同算法在不同数据集上表现出的分类准确率不同. 由于我们并不清楚这些数据集中数据分布的具体变化情况, 很难分析出各个算法应对各种类型的概念漂移的能力. 但从整体性能来看, REDLLA 和 SUN 算法在 Electricity 和 Poker Hand 数据集上表现最佳; ReaSC 和 SmSCluster 算法在

Weather 和 Forest Covertype 数据集上的性能较好.

表 3 累计准确率、标准差和统计监测结果

	ReaSC (◆/◇)		SmSCluster (●/○)		SEClass (★/☆)	
Agrawal-abr	60.11±0.45		60.76±0.3	◇	57.72±0.31	◆●
Agrawal-gra	**62.24±0.32**		62.02±0.29	◆	57.28±0.27	◆●
Sea-abr	87.58±0.45		87.83±0.34	◇	**88.37±0.35**	◇○
Sea-gra	88.13±0.26		88.14±0.3	–	**88.44±0.16**	◇○
Sine-abr	58.71±0.28		58.88±0.29	–	**80.48±0.52**	◇○
Sine-gra	67.19±0.58		67.34±0.28	–	**75.42±0.39**	◇○
Hyperplane	**79.56±2.89**		79.13±2.83	◆	70.22±1.11	◆●
Gaussian	37.95±1.5		38.96±1.34	–	**69.87±1.06**	◇○
Poker hand	52.78±0.2		52.92±0.21	–	58.53±0.16	◇○
Weather	**74.1±0.81**		74.07±0.5	–	69.27±0.7	◆●
Electricity	66.06±0.41		66.3±0.56	–	63.84±0.44	◆●
Forest cover type	77.01±0.26		**77.3±0.33**	◇	58.58±0.11	◆●
Win-tie-loss (*Acc*)	32-10-18		32-11-17		32-9-19	
Win-loss (*Std*)	43-17		48-12		53-7	
	SPASC (■/□)		REDLLA (▲/△)		SUN	
Agrawal-abr	**62.44±0.6**	◇○☆	56.71±2.1	◆●■	56.19±3.61	◆●■
Agrawal-gra	61.94±0.62	☆	58.73±2.03	◆●■	59.0±2.26	◆●☆■
Sea-abr	79.96±1.83	◆●★	82.88±0.6	◆●★□	81.33±0.57	◆●★▲
Sea-gra	79.41±1.62	◆●★	84.01±0.82	◆●★□	82.38±1.25	◆●★□▲
Sine-abr	75.87±0.94	◇○★	48.98±1.15	◆●★■	52.97±2.0	◆●★■△
Sine-gra	69.62±0.73	◇○★	50.6±2.6	◆●★■	55.13±2.06	◆●★■△
Hyperplane	62.48±4.6	◆●★	76.79±4.74	◆●★□	77.48±4.12	◆●☆□
Gaussian	60.37±1.76	◇○★	44.73±2.1	◇○★■	38.01±3.44	★■▲
Poker hand	54.78±1.29	◇○★	**60.68±4.63**	◇○□	60.6±4.47	◇○□
Weather	67.25±1.96	◆●★	68.94±1.02	◆●□	68.52±1.34	◆●
Electricity	57.68±1.24	◆●★	**70.33±3.33**	◇○☆□	68.91±3.86	◇☆□
Forest cover type	58.51±1.27	◆●	57.42±1.98	◆●	55.08±4.78	◆●★
Win-tie-loss (*Acc*)	22-13-25		12-13-35		16-12-32	
Win-loss (*Std*)	17-43		11-49		8-52	

注: 粗体表示在对应数据集上 6 个算法中分类准确率的最大者

5.3.2 分块准确率

为了分析各算法在各种数据流分类场景下更细致的差异, 本文给出了所有算法的准确率随数据块变化的趋势图. 数据包括 3 个具有突变概念漂移的数据集、3 个具有渐变概念漂移的数据集、2 个具有增量概念漂移的数据集以及 4 个真实数据集. 实验结果如图 2 所示.

从包含突变或渐变概念漂移的数据流对应的准确率变化趋势图中可以发现: 除 SPASC 外, 几乎所有算法都能跟踪数据流中概念的变化, 在数据块上的准确率有明显的下降、恢复和增长过程. 对于 SPASC, 主要的原因在于其概念漂移检测方法在部分数据集上还不够准确, 以及当分类器池满时, 强行更新最大相似性对应的分类器. 对于有增量概念漂移的数据, 各个算法的分类准确率都在持续的变化, 没有明显的下降、恢复和增长过程. 对于实际数据, 概念漂移的情况比较复杂, 因此各算法的准确率变化比较频繁, 但也能大致看出, 数据流中概念变化导致了分类准确率的变化.

为了更清楚地分析各算法准确率的变化情况, 本文跟踪了突变概念漂移情形下, 各算法在新概念第 1 个和第 2 个数据块上的分类准确率. 实验结果见表 4 和表 5.

从表 4 中的实验结果可以看出: 对于 Agrawal 数据和 Sine 数据, 在遇到新概念的第一数据块, 也就是发生概念漂移时, SPASC 的分类准确率下降的程度最小, 其次是 REDLLA; 对于 Sea 数据, 各算法分类准确率下降的程度相差不大. 这种现象产生的原因主要在于: SPASC 采取了在分类时动态调整权值的方法, 也就是 SPASC 假设了对样本在线分类后, 部分样本的类别可立即获得, 正确分类的分类器权值会增大. 相比于其他分类算法, 当数据比较难分的时候, SPASC 的这种动态调整权值的方法有一定效果.

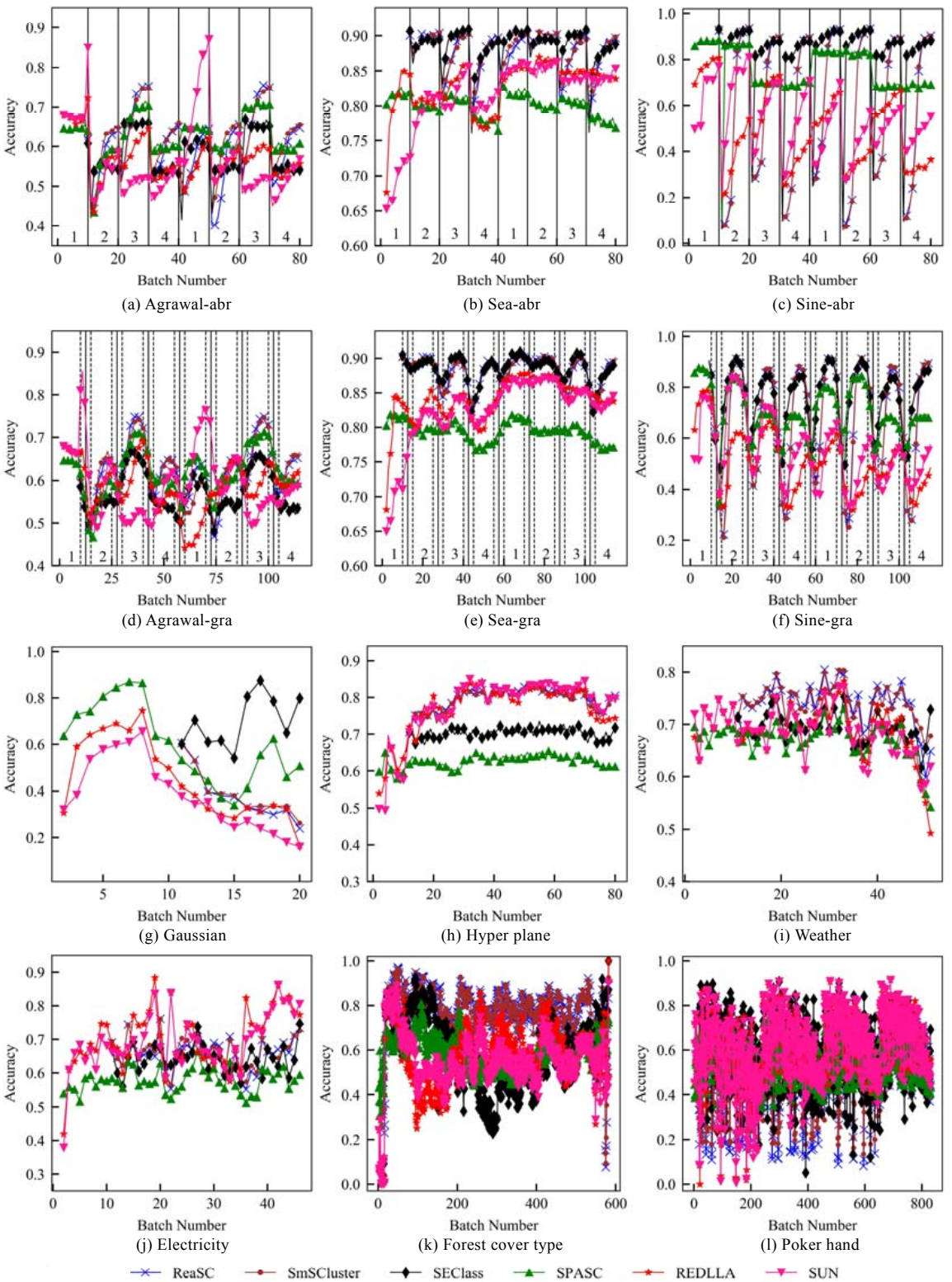

图 2　准确率随数据块的变化趋势图

表 4 新概念的第 1 个数据块上各算法的准确率

Dataset	*i*-th data batch	Algorithm					
		ReaSC	SmSCluster	SEClass	SPASC	REDLLA	SUN
Agrawal-abr	11	0.424 6	0.420 4	0.426 4	0.435 5	0.440 8	0.469
	21	0.521	0.510 8	0.536 9	**0.573 5**	0.546 4	0.474 4
	31	0.509 3	0.523 7	0.526 2	**0.587 2**	0.517 2	0.463 9
	41	0.490 6	0.478	0.415 5	**0.636 1**	0.513 3	0.548 9
	51	0.4	0.431 6	0.431 2	**0.580 8**	0.526 6	0.522
	61	0.526 6	0.512 8	0.557	**0.682 8**	0.575 5	0.483
	71	0.498 4	0.528 3	0.518 1	**0.600 5**	0.535 6	0.45
Sea-abr	11	**0.871 9**	0.874 4	0.861 2	0.797 8	0.799 2	0.741 7
	21	0.812 7	0.812 5	**0.829 6**	0.805 5	0.816 2	0.795 3
	31	0.773 8	0.787 1	0.761 3	0.779 1	0.771 1	**0.785 3**
	41	0.822 7	0.837 3	0.846 3	0.826 8	0.833 5	**0.850 2**
	51	**0.882 4**	0.878 8	0.869 6	0.801 7	0.845 8	0.846 9
	61	0.806 5	0.819 4	0.825 5	0.811 4	**0.841 5**	0.837 2
	71	0.773 2	0.791 9	0.762 2	0.771 4	**0.841 9**	0.826 8
Sine-abr	11	0.066 2	0.061 4	0.067 8	0.105 4	0.193 2	**0.219**
	21	0.267 3	0.266 1	0.272 4	**0.694 9**	0.469 6	0.339 1
	31	0.099 5	0.097 2	0.107 4	**0.662 6**	0.248 7	0.259 4
	41	0.261 5	0.248 5	0.250 8	**0.841 6**	0.507 7	0.336 9
	51	0.071 1	0.057 9	0.072 5	**0.815 1**	0.299 5	0.292 9
	61	0.272 9	0.271 1	0.278 5	**0.685**	0.548 5	0.425 6
	71	0.098 6	0.095 7	0.115 5	**0.677 1**	0.311 4	0.414 5
win-loss (*Acc*)		33-72	35-70	46-59	74-31	71-34	56-49

注: 粗体表示在对应数据块上 6 个算法中分类准确率的最大者

表 5 概念漂移发生后各算法准确率的恢复

Dataset	*i*-th data batch	Algorithm					
		ReaSC	SmSCluster	SEClass	SPASC	REDLLA	SUN
Agrawal-abr	12	0.460 1	0.449	**0.538**	0.434 8	0.434 8	0.462 5
	22	0.554 5	0.547 7	**0.658 2**	0.626 9	0.550 3	0.483
	32	0.517 7	0.543 1	0.534 4	**0.589 3**	0.517 1	0.475
	42	0.487 9	0.485 2	0.611 4	**0.649 5**	0.493 1	0.557 4
	52	0.401 6	0.472 5	0.540 5	**0.597 7**	0.515 9	0.511 6
	62	0.568 4	0.559 1	0.667 9	**0.700 3**	0.568 2	0.491 7
	72	0.511 4	0.544 5	0.542 4	**0.589 8**	0.530 8	0.465 3
Sea-abr	12	**0.885 1**	0.884 9	0.883 6	0.800 3	0.803 2	0.771 8
	22	0.830 4	0.837 5	**0.867**	0.814 5	0.81	0.797 8
	32	0.807 7	0.819 3	**0.839 8**	0.784 8	0.780 4	0.793 6
	42	0.842 5	0.860 7	**0.890 8**	0.827 2	0.837 8	0.842 8
	52	**0.892 3**	0.890 6	0.888 4	0.805 4	0.849 6	0.848 7
	62	0.824 7	0.832 7	**0.878 9**	0.811	0.847 4	0.835 5
	72	0.811 2	0.821	0.845 2	0.782 3	**0.850 2**	0.837 7
Sine-abr	12	0.083 4	0.078 9	**0.860 5**	0.860 3	0.217 3	0.428 6
	22	0.283	0.291 5	**0.815 2**	0.700 1	0.467 6	0.393 2
	32	0.113 6	0.116 3	**0.810 4**	0.680 5	0.255 3	0.297 8
	42	0.297 8	0.281 2	**0.859 9**	0.837 1	0.519 6	0.338 8
	52	0.088 6	0.073 6	**0.857 8**	0.838	0.291 2	0.276 3
	62	0.297	0.294 7	**0.818 4**	0.685 1	0.558	0.425 4
	72	0.120 5	0.111 2	**0.816 8**	0.676	0.308 8	0.401 8
win-loss (*Acc*)		38-67	39-66	93-12	61-44	46-59	38-67

注: 粗体表示在对应数据块上 6 个算法中分类准确率的最大者

从表 5 中的实验结果可以看出: 对于 Agrawal 数据, SPASC 的分类准确率恢复最快; 而对于 Sea 数据和 Sine 数据, SEClass 的分类准确率恢复最快. 这里的原因主要在于各算法对概念漂移适应的准确性. 对于 SPASC 而言, 在 Agrawal 数据上算法检测到概念漂移的准确性比在 Sea 数据和 Sine 数据上要好. 这是因为 Sea 数据是通过分类面平移产生概念漂移, Sine 数据会通过概念反转产生概念突变, 这两种情形下, $p(\boldsymbol{x})$不发生变化, 因此, 此种情形下, SPASC 的概念漂移检测方法精度不高; 而对于 SEClass, 它是通过计算在 D^t 上的分类准确率来调整分类器权值的, 因此它调整很快.

5.3.3 标记比例的影响

为了分析标记比例的大小(即有标记样本的数量)对算法分类准确率产生的影响, 我们将标记比例的大小分别设置为 0.05, 0.1, 0.2 和 0.3. 实验结果见表 6–表 8.

从表 6–表 8 中的实验结果可以看到: 随着有标记样本比例的增加, 所有算法的分类准确率都在增加. 这显然是因为标记的样本越多, 各算法用于训练和更新分类模型的知识也更多, 这就使得模型的训练和更新更加准确. 还可以看到: 当有标记样本的比率过小时(0.05), 各算法的累计分类准确率会比较差. 所以在实际应用中, 受限于条件与成本, 样本标记比率不能太高的情形下, 随机选择样本标记类别的做法可能不合适, 而主动标记样本可能更好. 另外还可以看到: 当样本被标记的比率成倍增加时, 算法的分类准确率并不能得到成倍的提升. 这与数据本身的可分性有关. 因此实际应用中, 可根据数据本身的可分性和成本等因素, 综合确定样本的标记比率.

表 6 在包含突变或者渐变概念漂移的数据集上, 不同标记比例下各算法的累计分类准确率

Dataset	Ratio	SPASC	ReaSC	SmSCluster	SEClass	REDLLA	SUN
		XX-abr					
Agrawal	0.05	0.582 5	0.594 3	0.596 3	0.558 8	0.536 2	0.528 5
	0.1	0.589 1	0.597 3	0.601 1	0.565 9	0.564 8	0.535 9
	0.2	0.624 4	0.601 1	0.607 7	0.577 2	0.567 1	0.561 9
	0.3	0.631 2	0.6023	0.609 6	0.586 6	0.572 5	0.588 8
Sea	0.05	0.799 9	0.872 1	0.870 3	0.839	0.811 7	0.776 1
	0.1	0.794 4	0.875 7	0.874 9	0.866 1	0.828 2	0.796
	0.2	0.799 6	0.875 8	0.878 3	0.883 7	0.828 8	0.813 3
	0.3	0.795 8	0.878 8	0.879 2	0.885 9	0.836 3	0.819 3
Sine	0.05	0.494	0.580 9	0.579 6	0.748 7	0.489 3	0.521 2
	0.1	0.497 8	0.585 6	0.583 2	0.778 7	0.497 8	0.520 4
	0.2	0.758 7	0.587 1	0.588 8	0.804 8	0.489 9	0.529 7
	0.3	0.765 2	0.587 3	0.589 6	0.815 1	0.508 2	0.54
		XX-gra					
Agrawal	0.05	0.589 1	0.611 3	0.607 8	0.552 8	0.565 6	0.533 3
	0.1	0.594 1	0.616 9	0.614 3	0.560 1	0.576 1	0.569 6
	0.2	0.619 4	0.622 3	0.620 2	0.572 8	0.587 2	0.59
	0.3	0.626 3	0.625 1	0.626 4	0.582	0.579 8	0.594 3
Sea	0.05	0.792 4	0.873 6	0.872 2	0.830 8	0.826 9	0.725 8
	0.1	0.789 7	0.878 1	0.876 9	0.871 4	0.834	0.816 3
	0.2	0.794 1	0.881 3	0.881 4	0.884 5	0.840 1	0.823 8
	0.3	0.785 4	0.885 2	0.884 4	0.886 8	0.841 3	0.829 2
Sine	0.05	0.492 8	0.665 8	0.664 5	0.704 7	0.49	0.520 1
	0.1	0.495 9	0.674 2	0.669 9	0.728 8	0.498 7	0.531 8
	0.2	0.696 2	0.671 9	0.673 5	0.754 2	0.505 9	0.551 3
	0.3	0.694 5	0.672 3	0.676 4	0.767 6	0.512 9	0.556 9

表 7 在包含增量概念漂移的数据集上, 不同标记比例下各算法的累计分类准确率

Ratio	Incremental					
	SPASC	ReaSC	SmSCluster	SEClass	REDLLA	SUN
	Hyperplane					
0.05	0.528 3	0.783 1	0.767 5	0.646 6	0.683 1	0.688 9
0.1	0.537 1	0.794 1	0.779 3	0.671 9	0.731	0.759 8
0.2	0.624 8	0.795 6	0.791 3	0.702 1	0.767 9	0.774 8
0.3	0.680 7	0.799 2	0.800 8	0.728 7	0.763 2	0.783 3
	Gaussian					
0.05	0.485 6	0.371	0.383 3	0.662	0.404 2	0.314 1
0.1	0.508 6	0.369 3	0.380 4	0.680 9	0.425 7	0.354 1
0.2	0.603 7	0.379 5	0.389 6	0.698 7	0.447 3	0.380 1
0.3	0.661	0.376 6	0.387 1	0.701 1	0.438 8	0.385 4

表 8 在真实数据集上, 不同标记比例下各算法的累计分类准确率

Ratio	SPASC	ReaSC	SmSCluster	SEClass	REDLLA	SUN
	Poker hand					
0.05	0.257	0.507 9	0.504 4	0.580 4	0.568 9	0.571 6
0.1	0.251	0.521 9	0.516 1	0.583 2	0.575 3	0.563 6
0.2	0.547 8	0.527 8	0.529 2	0.585 3	0.606 8	0.606
0.3	0.547 5	0.529 6	0.538 4	0.585	0.585 3	0.594
	Electricity					
0.05	0.550 9	0.640 1	0.646 6	0.605 1	0.557 1	0.546 1
0.1	0.573 1	0.655 4	0.658 7	0.625 5	0.618 5	0.568 2
0.2	0.576 8	0.660 6	0.663	0.638 5	0.703 3	0.689 1
0.3	0.617 6	0.665 4	0.665	0.645 1	0.715 6	0.720 7
	Weather					
0.05	0.655 2	0.715 2	0.719 7	0.648 7	0.677 5	0.673 5
0.1	0.651 4	0.728 7	0.731 5	0.672	0.678 5	0.679 3
0.2	0.672 5	0.741	0.740 7	0.692 8	0.689 4	0.685 2
0.3	0.689 7	0.743 1	0.747 2	0.701 9	0.686 4	0.684 8
	Forest cover type					
0.05	0.387 6	0.747 6	0.750 8	0.581 1	0.545 5	0.537 9
0.1	0.363 4	0.764 6	0.764 7	0.584 6	0.572 9	0.555 4
0.2	0.585 1	0.770 1	0.773	0.585 8	0.574 2	0.550 8
0.3	0.604 3	0.770 1	0.775 3	0.586 5	0.576 8	0.545 6

5.3.4 分类器池大小的影响

对于多模型算法, 为了分析分类器池大小(即池中最多能维持的分类器数量)对算法分类准确率产生的影响, 本文将分类器池的大小分别设置为 1, 3, 5 和 7, 标记比例的大小设置为 0.2. 实验结果见表 9–表 11.

表 9 在包含突变或者渐变概念漂移的数据集上, 分类器池的大小对分类准确率的影响

Dataset	Concept Num	Size	XX-abr				XX-gra			
			SPASC	ReaSC	SmSCluster	SEClass	SPASC	ReaSC	SmSCluster	SEClass
Agrawal	4	1	0.591	0.608 2	0.605 7	0.579 2	0.597 6	0.599 6	0.596 4	0.574 2
	4	3	0.612 8	0.622 2	0.621	0.579	0.612 2	0.621 4	0.619 7	0.574
	4	5	0.617 5	0.619 6	0.619 3	0.578 4	0.614 2	0.627 5	0.623 5	0.573 5
	4	7	0.623 6	0.610 3	0.613 4	0.577 6	0.615 9	0.626 4	0.622 9	0.573
Sea	4	1	0.784 2	0.816 2	0.808	0.846 2	0.781 3	0.810 3	0.805 1	0.845 3
	4	3	0.797 7	0.870 2	0.867 3	0.873 6	0.794 1	0.869 1	0.866 5	0.874 3
	4	5	0.799 6	0.877 9	0.877 3	0.879 8	0.794 1	0.879 1	0.878 4	0.881 8
	4	7	0.799 6	0.877	0.878 3	0.882 6	0.794 1	0.881 7	0.881 1	0.884 1
Sine	4	1	0.499 9	0.791 9	0.790 9	0.783 4	0.497 9	0.733 6	0.734 8	0.740 1
	4	3	0.750 7	0.762 9	0.763 8	0.802 4	0.653 7	0.755 3	0.758 1	0.751 1
	4	5	0.755	0.701 8	0.701 7	0.807 4	0.683 5	0.734 5	0.735 1	0.755 7
	4	7	0.758 1	0.628 7	0.629 2	0.806 3	0.693 5	0.697 3	0.697 1	0.755 2

表 10 在包含增量概念漂移的数据集上, 分类器池的大小对分类准确率的影响

Size	Incremental							
	Hyperplane				Gaussian			
	SPASC	ReaSC	SmSCluster	SEClass	SPASC	ReaSC	SmSCluster	SEClass
1	0.546 9	0.690 3	0.680 8	0.696 3	0.521 8	0.766 9	0.775 5	0.762 9
3	0.618 9	0.749 7	0.741 1	0.699 1	0.560 5	0.706 8	0.712 8	0.762 1
5	0.621 2	0.776 1	0.767	0.700 2	0.592 2	0.634 7	0.642 5	0.748 4
7	0.623 8	0.788 8	0.781 7	0.700 8	0.602 8	0.541 8	0.550 2	0.721

表 11 在真实数据集上, 分类器池的大小对分类准确率的影响

Size	Poker hand				Weather			
	SPASC	ReaSC	SmSCluster	SEClass	SPASC	ReaSC	SmSCluster	SEClass
1	0.247 6	0.525 6	0.521 8	0.585 5	0.672 5	0.687 8	0.678 8	0.696 6
3	0.517 4	0.542 8	0.535 6	0.585 8	0.672 5	0.720 5	0.722 5	0.696 9
5	0.528 7	0.536 4	0.530 7	0.585 6	0.672 5	0.731 7	0.734 8	0.696 2
7	0.537 8	0.530 4	0.528 7	0.585 5	0.672 5	0.737	0.738 6	0.695 7
	Electricity				Forest cover type			
1	0.568 4	0.635 3	0.631 8	0.628 9	0.412 1	0.835 8	0.837 6	0.581
3	0.576 7	0.651 2	0.651	0.631 1	0.467 3	0.819 6	0.821 3	0.582 2
5	0.576 8	0.653 6	0.655 6	0.636 4	0.512 3	0.803	0.805	0.583 1
7	0.576 8	0.655 7	0.654 3	0.635 7	0.545 4	0.788 2	0.79	0.584 1

从表 9–表 11 中分析相同算法在同一数据集上的实验结果可以发现: 分类器池大小为 7 时, 算法的准确率普遍大于分类器池大小为 1 时的准确率. 该现象符合集成模型在大多数情况下优于单模型的事实.

从表 9–表 11 中的实验结果还可以看到: 对于 SPASC 和 SEClass, 其分类准确率随分类器池的增大而提高, 但是提高幅度会逐渐变小; 对于 ReaSC 和 SmSCluster, 它们的分类准确率不会总是随分类器池的增大而提高. 其中的原因可分析如下.

对于 SPASC, 分类器池中的基分类器通常对应着不同概念. 随着分类器池的扩大, 算法可保存的已经学到的概念数量也随之增大, 使得概念漂移的检测能更准确. 因此理论上, 准确率应随着分类器池大小的增加而增大, 这一点在大多数数据集的实验结果上得到了验证. 而对于 ReaSC 和 SmSCluster 算法, 它们的分类器池更新规则是: 在当前数据块 D^t 上新建一个分类器并添加到分类器池中, 同时将分类器池中在当前数据块 D^t 上分类效果最差的分类器删除. 每次只替换掉一个分类器, 这会导致在新概念出现时, 分类器池容量越大, 历史概念对应的分类器被新概念对应的分类器完全替换的速度越慢. 由于分类是带权多数投票, 因此对于概念之间差异较小或数据分布变化不明显的数据流(Agrawal-gra, Sea-abr/gra, Hyperplane, Weather 和 Electricity), 历史概念对应的分类器在对新概念中的数据进行多数投票分类时起到了积极作用, 准确率会随着分类器池的扩大而增大. 而该类算法在其他数据流上则会出现相反的结果. SEClass 算法为分类器池中的每个分类器分配一个权值, 最大权值对应的分类器被用于分类. 因此,该算法有效地降低了历史概念对应的分类器对属于新概念的数据进行分类时产生的负影响.

6 未来的研究方向

由于开放环境下数据的产生受到众多因素的影响, 数据分布的变化更加复杂. 这将导致更加复杂的机器学习问题. 通过对现有带概念漂移的数据流的半监督分类算法的归类、总结以及对部分算法的实验结果的详细分析, 数据流半监督分类算法有以下几个方面的问题值得去做进一步的探讨.

(1) 概念漂移检测问题. 根据概念漂移产生的本质, 一个好的概念漂移检测方法应该同时检测分布 $P(\boldsymbol{X})$ 和 $P(\boldsymbol{Y}|\boldsymbol{X})$的变化. 在半监督环境下, 基于有标记样本的准确率的概念漂移检测方法会由于标记样本数量少而导致检测不准确. 基于无标记样本的分布变化来检测概念漂移检测方法显然也不完全可靠. 从综述中看到, 很少有同时考虑有标记和无标记样本来检测概念漂移的方法. 概念漂移的检测准确性对于数据流分类至关重要: 概念漂移检测准了, 分类器的更新也会准, 算法的分类准确率就能不断提高. 本文作者尝试借用迁移学习的方法来提高概念漂移检测的准确性, 取得了较好的分类效果[60]. 但是, 要想把概念漂移检测得更准, 可能需要更多的计算. 因此, 如何快速实施概念漂移检测, 是数据流半监督分类算法进入实际应用的关键;

(2) 重现概念学习问题. 在开放环境下, 数据的分布可能呈现周期性的变化. 当历史概念重复出现时, 相应的历史模型应被重新调用来对即将到来的样本实施分类和增量学习, 以避免重新学习. 在半监督环境下, 在一个数据块上重新学得的模型由于可学习的有标记样本量少而导致模型泛化能力比较差, 这又会导致对后续概念漂移检测的不准. 因此, 当检测到重现概念时, 如何利用当前获得的数据更新已学习的模型, 使得已学习的分类器的泛化能力不断提高? 然而, 大部分数据流半监督算法没有考虑重现概念的增量学习问题. 更进一步, 如何能在概念漂移检测不太准确的条件下, 使得增量学习能不断提高对重现概念的检测能力和泛化能力?

(3) 类别不平衡问题. 类别不平衡在监督环境下的数据流分类领域是一个具有挑战性的问题, 已有数据流分类综述中对该问题进行了总结和展望. 然而在半监督环境下, 类别不平衡现象将会使问题变得更加严重, 极端情况下可能导致某一类的样本不存在以至于无法学习该类样本中的知识, 概念漂移检测更会受到影响;

(4) 噪声数据问题. 受环境的变化、设备的损耗和人工错误地标注等因素的影响, 数据中往往存在噪声. 在半监督环境下, 有标记样本数量比较少, 模型会对噪声数据的适应性比较差, 可能因学习噪声数

据而导致局部无标记样本的错误利用. 比如: 在局部聚簇中, 通过打标记的方式来利用无标记样本进行模型训练的方法中, 如果该聚簇中有标记的噪声数据占多数, 那么该簇中的无标记样本将被打上错误标记, 将放大噪声数据带来的负面影响;

(5) 标记稀缺问题. 在数据流半监督分类的实际应用中, 从标注成本来看, 样本被标记的比率越低越好. 但是从本文的实验中也看出: 标记比率过低, 算法的分类准确率不会太好. 因此, 在标记稀缺条件下, 如何利用半监督分类、迁移学习、增量学习、主动学习的方法来提高算法的分类准确率, 也是非常紧迫的问题.

7 总 结

开放环境下的数据流半监督分类给机器学习带来了极大挑战. 本文对现有带概念漂移的数据流的半监督分类算法进行了介绍, 按照各算法采用的分类器类型, 对已有的多个算法逐个进行了介绍与总结; 归纳并总结了现有数据流半监督分类中采用的概念漂移检测方法; 在一些被广泛使用的真实数据集和人工数据集上, 对部分数据流半监督分类算法进行了多方面的比较与分析. 实验结果表明: 数据流半监督分类算法的分类准确率与分类策略、概念漂移检测方法、分类模型更新策略等众多因素有关, 但与数据分布本身关系最大. 在具体的研究与应用中, 最好能根据数据分布的特点选择数据流半监督分类算法以及样本标注比率与标注方式. 最后, 本文还提出了带概念漂移的数据流的半监督分类一些值得进一步深入探讨的问题.

尽管带概念漂移的数据流的半监督分类已经取得了很多研究成果, 并被应用于实际场景中. 但是该方面的研究仍然处于发展初期, 远还没有成熟, 期待学界进一步的研究和突破.

References:

[1] Lian D, Xie X, Chen E. Discrete matrix factorization and extension for fast item recommendation. IEEE Trans. on Knowledge and Data Engineering (TKDE), 2021, 33(5): 1919−1933. [doi: 10.1109/TKDE.2019.2951386]

[2] Wang H, Chen EH, Liu Q, *et al*. A united approach to learning sparse attributed network embedding. In: Proc. of the IEEE Int'l Conf. on Data Mining. Piscataway: IEEE, 2018. 557−566. [doi: 10.1109/ICDM.2018.00071]

[3] Zhang ML, Li YK, Liu XY, *et al*. Binary relevance for multi-label learning: An overview. Frontiers of Computer Science, 2018, 12(2): 191−202. [doi: 10.1007/s11704-017-7031-7]

[4] Guo GD, Chen LF, Ye YF, *et al*. Cluster validation method for determining the number of clusters in categorical sequences. IEEE Trans. on Neural Networks and Learning Systems, 2016, 28(12): 2936−2948. [doi: 10.1109/TNNLS.2016.2608354]

[5] Zhao XW, Liang JY. An attribute weighted clustering algorithm for mixed data based on information entropy. Journal of Computer Research and Development, 2016, 53(5): 1018−1028 (in Chinese with English abstract). [doi: 10.7544/issn1000-1239.2016.20150131]

[6] Huai BX, Chen EH, Zhu HS, *et al*. Toward personalized context recognition for mobile users: A semi-supervised Bayesian HMM approach. ACM Trans. on Knowledge Discovery from Data, 2014, 9(2): 1−29. [doi: 10.1145/2629504]

[7] Zhang ML, Zhou ZH. Exploiting unlabeled data to enhance ensemble diversity. Data Mining and Knowledge Discovery, 2013, 26(1): 98−129. [doi: 10.1007/s10618-011-0243-9]

[8] Guo GD, Li N, Chen LF. Concept drift detection for data streams based on mixture model. Journal of Computer Research and Development, 2014, 51(4): 731−742 (in Chinese with English abstract). [doi: 10.7544/issn1000-1239.2014.20120582]

[9] Bai L, Cheng XQ, Liang JY, *et al*. An optimization model for clustering categorical data streams with drifting concepts. IEEE Trans. on Knowledge and Data Engineering, 2016, 28(11): 2871−2883. [doi: 10.1109/TKDE.2016.2594068]

[10] Zhang X, Liu C, Suen C. Towards robust pattern recognition: A review. Proc. of the IEEE, 2020, 108(6): 894−922. [doi: 10.1109/JPROC.2020.2989782]

[11] Wang J, Lan C, Liu C, *et al*. Generalizing to unseen domains: A survey on domain generalization. In: Proc. of the 30th Int'l Joint Conf. on Artificial Intelligence. New York: ACM, 2021. 4627−4635. [doi: 10.24963/ijcai.2021/628]

[12] Sayed-Mouchaweh M, Lughofer E. Learning in Non-stationary Environments: Methods and Applications. Germany: Springer, 2013. [doi: 10.1007/978-1-4419-8020-5]

[13] Gama J, Ganguly A, Omitaomu O, *et al*. Knowledge discovery from data streams. Intelligent Data Analysis, 2009, 13(3): 403−404. [doi: 10.3233/IDA-2009-0372]

[14] Zhao P, Zhou ZH. Learning from distribution-changing data streams via decision tree model reuse. Scientia Sinica Informationis, 2021, 51(1): 1−12. (in Chinese with English abstract). [doi: 10.1360/SSI-2020-0170]

[15] 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. [doi: 10.1007/s10115-015-0837-4]

[16] Xu WH, Qin Z, Chang Y. Semi-supervised learning based ensemble classifier for stream data. Pattern Recognition and Artificial Intelligence, 2012, 25(2): 292−299 (in Chinese with English abstract). [doi: 10.16451/j.cnki.issn1003-6059.2012.02.010]

[17] Wang Y, Li T. Improving semi-supervised co-forest algorithm in evolving data streams. Applied Intelligence, 2018, 48(10): 3248−3262. [doi: 10.1007/s10489-018-1149-7]

[18] Ahmadi Z, Beigy H. Semi-supervised ensemble learning of data streams in the presence of concept drift. In: Proc. of the 7th Int'l Conf. on Hybrid Artificial Intelligent Systems. Berlin: Springer, 2012. 526−537. [doi: 10.1007/978-3-642-28931-6_50]

[19] Krawczyk B, Minku LL, Gama J, *et al*. Ensemble learning for data stream analysis: A survey. Information Fusion, 2017, 37: 132−156. [doi: 10.1016/j.inffus.2017.02.004]

[20] Zhu Y, Li Y. Semi-supervised streaming learning with emerging new labels. In: Proc. of the 34th AAAI Conf. on Artificial Intelligence. Menlo Park: AAAI, 2020. 7015−7022. [doi: 10.1609/aaai.v34i04.6186]

[21] Noorbehbahani F, Fanian A, Mousavi R, *et al*. An incremental intrusion detection system using a new semi-supervised stream classification method. Int'l Journal of Communication Systems, 2015, 30(4): 1−26. [doi: 10.1002/dac.3002]

[22] Hu Y, Baraldi P, Maio F, *et al*. A systematic semi-supervised self-adaptable fault diagnostics approach in an evolving environment. Mechanical Systems and Signal Processing, 2017, 88: 413−427. [doi: 10.1016/j.ymssp.2016.11.004]

[23] Sedhai S, Sun A. Semi-supervised spam detection in Twitter stream. IEEE Trans. on Computational Social Systems, 2017, 5(1): 169−175. [doi: 10.1109/TCSS.2017.2773581]

[24] Matuszyk P, Spiliopoulou M. Stream-based semi-supervised learning for recommender systems. Machine Learning, 2017, 106(6): 771−798. [doi: 10.1007/s10994-016-5614-4]

[25] Grzenda M, Bustillo A. Semi-supervised roughness prediction with partly unlabeled vibration data streams. Journal of Intelligent Manufacturing, 2019, 30(2): 933−945. [doi: 10.1007/s10845-018-1413-z]

[26] Schlimmer JC, Granger RH. Incremental learning from noisy data. Machine Learning, 1986, 1(3): 317−354. [doi: 10.1007/BF00116895]

[27] Kuncheva LI. Classifier ensembles for changing environments. In: Proc. of the Int'l Workshop on Multiple Classifier Systems. Berlin: Springer, 2004. 1−15. [doi: 10.1007/978-3-540-25966-4_1]

[28] Tsymbal A. The problem of concept drift: Definitions and related work. Technical Report, Department of Computer Science Trinity College Dublin, 2004.

[29] Wang T, Li ZJ, Yan YJ, *et al*. A survey of classification of data stream. Journal of Computer Research and Development, 2007, 44(11): 1809−1815 (in Chinese with English abstract). [doi: 10.1038/onc.2012.370]

[30] Žliobaitė I. Learning under Concept Drift: An Overview. Computer Science, 2010.

[31] Gama J. A survey on learning from data streams: Current and future trends. Progress in Artificial Intelligence, 2012, 1(1): 45−55. [doi: 10.1007/s13748-011-0002-6]

[32] Hoens TR, Polikar R, Chawla NV. Learning from streaming data with concept drift and imbalance: An overview. Progress in Artificial Intelligence, 2012, 1(1): 89−101. [doi: 10.1007/s13748-011-0008-0]

[33] Wen YM, Qiang BH, Fan ZG. A survey of the classification of data streams with concept drift. CAAI Trans. on Intelligent Systems, 2013, 8(2): 95−104 (in Chinese with English abstract). [doi: 10.3969/j.issn.1673-4785.201208012]

[34] Gama J, Žliobaitė I, Bifet A, *et al*. A survey on concept drift adaptation. ACM Computing Surveys, 2014, 46(4): 1−37. [doi: 10.1145/2523813]

[35] Krempl G, Žliobaite I, Brzeziński D, *et al*. Open challenges for data stream mining research. ACM SIGKDD Explorations Newsletter, 2014, 16(1): 1−10. [doi: 10.1145/2674026.2674028]

[36] Gonçalves JPM, de Carvalho Santos SGT, de Barros RSM, *et al*. A comparative study on concept drift detectors. Expert Systems with Applications, 2014, 41(18): 8144−8156. [doi: 10.1016/j.eswa.2014.07.019]

[37] Ditzler G, Roveri M, Alippi C, *et al*. Learning in nonstationary environments: A survey. IEEE Computational Intelligence Magazine, 2015, 10(4): 12−25. [doi: 10.1109/MCI.2015.2471196]

[38] Nguyen HL, Woon YK, Ng WK. A survey on data stream clustering and classification. Knowledge and Information Systems, 2015, 45(3): 535−569. [doi: 10.1007/s10115-014-0808-1]

[39] Ding J, Han M, Li J. Review of concept drift data streams mining techniques. Computer Science, 2016, 43(12): 24−29,62 (in Chinese with English abstract). [doi: 10.11896/j.issn.1002-137x.2016.12.004]

[40] Webb GI, Hyde R, Cao H, *et al*. Characterizing concept drift. Data Mining and Knowledge Discovery, 2016, 30(4): 964−994. [doi: 10.1007/s10618-015-0448-4]

[41] Iwashita AS, Papa JP. An overview on concept drift learning. IEEE Access, 2018, 7: 1532−1547. [doi: 10.1109/ACCESS.2018.2886026]

[42] Khamassi I, Sayed-Mouchaweh M, Hammami M, *et al*. Discussion and review on evolving data streams and concept drift adapting. Evolving Systems, 2018, 9(1): 1−23. [doi: 10.1007/s12530-016-9168-2]

[43] Lu J, Liu A, Dong F, *et al*. Learning under concept drift: A review. IEEE Trans. on Knowledge and Data Engineering, 2019, 31(12): 2346−2363. [doi: 10.1109/TKDE.2018.2876857]

[44] Dyer KB, Polikar R. Semi-supervised learning in initially labeled non-stationary environments with gradual drift. In: Proc. of the 2012 Int'l Joint Conf. on Neural Networks. Piscataway: IEEE, 2012. 1−9. [doi: 10.1109/IJCNN.2012.6252541]

[45] Dyer KB, Capo R, Polikar R. Compose: A semi-supervised learning framework for initially labeled nonstationary streaming data. IEEE Trans. on Neural Networks and Learning Systems, 2013, 25(1): 12−26. [doi: 10.1109/TNNLS.2013.2277712]

[46] Ferreira RS, Zimbrão G, Alvim LGM. AMANDA: Semi-supervised density-based adaptive model for non-stationary data with extreme verification latency. Information Sciences, 2019, 488: 219−237. [doi: 10.1016/j.ins.2019.03.025]

[47] Zhang P, Zhu XQ, Tan JL, *et al*. Classifier and cluster ensembles for mining concept drifting data streams. In: Proc. of the 2010 IEEE Int'l Conf. on Data Mining. Piscataway: IEEE, 2010. 1175−1180. [doi: 10.1109/ICDM.2010.125]

[48] 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. New York: ACM, 2000. 71−80. [doi: 10.1145/347090.347107]

[49] 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. New York: ACM, 2001. 97−106. [doi: 10.1145/502512.502529]

[50] Žliobaitė I. Adaptive training set formation [Ph.D. Thesis]. Vilniaus: Vilnius University, 2010.

[51] Kelly MG, Hand DJ, Adams NM. The impact of changing populations on classifier performance. In: Proc. of the 5th ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining. New York: ACM, 1999. 367−371. [doi: 10.1145/312129.312285]

[52] Feng ZX, Wang M, Yang SY, *et al*. Incremental semi-supervised classification of data streams via self-representative selection. Applied Soft Computing, 2016, 47: 389−394. [doi: 10.1016/j.asoc.2016.02.023]

[53] Parker BS, Khan L. Detecting and tracking concept class drift and emergence in non-stationary fast data streams. In: Proc. of the 29th AAAI Conf. on Artificial Intelligence. Menlo Park: AAAI, 2015. 2908−2913. [doi: 10.5555/2888116.2888121]

[54] Khezri S, Tanha J, Ahmadi A, *et al*. A novel semi-supervised ensemble algorithm using a performance-based selection metric to non-stationary data streams. Neurocomputing, 2021, 442(6): 125−145. [doi: 10.1016/j.neucom.2021.02.031]

[55] Masud MM, Gao J, Khan L, *et al*. A practical approach to classify evolving data streams: Training with limited amount of labeled data. In: Proc. of the 8th IEEE Int'l Conf. on Data Mining. Piscataway: IEEE, 2008. 929−934. [doi: 10.1109/ICDM.2008.152]

[56] Masud MM, Woolam C, Gao J, *et al*. Facing the reality of data stream classification: Coping with scarcity of labeled data. Knowledge and Information Systems, 2012, 33(1): 213−244. [doi: 10.1007/s10115-011-0447-8]

[57] Woolam C, Masud MM, Khan L. Lacking labels in the stream: Classifying evolving stream data with few labels. In: Proc. of the Int'l Symp. on Methodologies for Intelligent Systems. Berlin: Springer, 2009. 552−562. [doi: 10.1007/978-3-642-04125-9_58]

[58] Qin KK, Wen YM. Semi-supervised classification of concept drift data stream based on local component replacement. In: Proc. of the Int'l CCF Conf. on Artificial Intelligence. Singapore: Springer, 2018. 98−112. [doi: 10.1007/978-981-13-2122-1_8]

[59] Casalino G, Castellano G, Mencar C. Incremental adaptive semi-supervised fuzzy clustering for data stream classification. In: Proc. of the 2018 IEEE Conf. on Evolving and Adaptive Intelligent Systems. Piscataway: IEEE, 2018. 1–7. [doi: 10.1109/EAIS.2018.8397172]

[60] Wen YM, Liu S. Semi-supervised classification of data streams by BIRCH ensemble and local structure mapping. Journal of Computer Science and Technology, 2020, 35(2): 295–304. [doi: 10.1007/s11390-020-9999-y]

[61] Din SU, Shao J, Kumar J, *et al*. Online reliable semi-supervised learning on evolving data streams. Information Sciences, 2020, 525(7): 153–171. [doi: 10.1016/j.ins.2020.03.052]

[62] Zheng X, Li P, Hu X, *et al*. Semi-supervised classification on data streams with recurring concept drift and concept evolution. Knowledge-Based Systems, 2021, 215(3): 106749. [doi: 10.1016/j.knosys.2021.106749]

[63] Li PP, Wu XD, Hu XG. Mining recurring concept drifts with limited labeled streaming data. ACM Trans. on Intelligent Systems and Technology, 2012, 13(2): 241–252. [doi: 10.1145/2089094.2089105]

[64] Wu XD, Li PP, Hu XG. Learning from concept drifting data streams with unlabeled data. Neurocomputing, 2012, 92: 145–155. [doi: 10.1016/j.neucom.2011.08.041]

[65] Nguyen HL, Ng WK, Woon YK, *et al*. Concurrent semi-supervised learning of data streams. In: Proc. of the 13th Int'l Conf. on Data Warehousing and Knowledge Discovery. Berlin, Heidelberg: Springer, 2011. 445–459. [doi: 10.1007/978-3-642-23544-3_34]

[66] Anastasovski G, Popstojanova KG. Classification of partially labeled malicious web traffic in the presence of concept drift. In: Proc. of the 8th Int'l Conf. on Software Security and Reliability-Companion. Piscataway: IEEE, 2014. 130–139. [doi: 10.1109/SERE-C.2014.31]

[67] Hu XG, Ma LW, Li PP. Data stream ensemble classification algorithm based on tri-training. Journal on Data Acquisition and Processing, 2017, 32(5): 853–860 (in Chinese with English abstract). [doi: 10.16337/j.1004-9037.2017.05.001]

[68] Wen YM, Zhou Q, Xue Y, *et al*. Transfer learning for semi-supervised classification of non-stationary data streams. In: Proc. of the 27th Int'l Conf. on Neural Information Processing. Berlin: Springer, 2020. 468–477. [doi: 10.13140/2.1.3891.2644]

[69] Liu C, Wen Y, Xue Y. Semi-supervised classification of data streams based on adaptive density peak clustering. In: Proc. of the 27th Int'l Conf. on Neural Information Processing. Cham: Springer, 2020. 639–650. [doi: 10.1007/978-3-030-63833-7_54]

[70] Jr JRB, Lopes A, Liang Z. Partially labeled data stream classification with the semi-supervised *k*-associated graph. Journal of the Brazilian Computer Society, 2012, 18(4): 299–310. [doi: 10.1007/s13173-012-0072-8]

[71] Jia Y, Yan S, Zhang C. Semi-supervised classification on evolutionary data. In: Proc. of the 21st Int'l Joint Conf. on Artificial Intelligence. New York: ACM, 2009. 1083–1088.

[72] Zhang P, Zhu XQ, Guo L. Mining data streams with labeled and unlabeled training examples. In: Proc. of the 9th IEEE Int'l Conf. on Data Mining. Piscataway: IEEE, 2009. 627–636. [doi: 10.1109/ICDM.2009.76]

[73] Zhu X, Jin R. Multiple information sources cooperative learning. In: Proc. of the 21sth Int'l Joint Conf. on Artificial Intelligence. New York: ACM, 2009. 1369–1375.

[74] Guo B, Menon J, Willette B. Surface reconstruction using alpha shapes. Computer Graphics Forum, 2010, 16(4): 177–190. [doi: 10.1111/1467-8659.00178]

[75] Li Y, Wang Y, Liu Q, *et al*. Incremental semi-supervised learning on streaming data. Pattern Recognition, 2019, 88: 383–396. [doi: 10.1016/j.patcog.2018.11.006]

[76] Silva CASD, Krohling RA. Semi-supervised online elastic extreme learning machine with forgetting parameter to deal with concept drift in data streams. In: Proc. of the 2019 Int'l Joint Conf. on Neural Networks. Piscataway: IEEE, 2019. 1–8. [doi: 10.1109/IJCNN.2019.8852361]

[77] Widyantoro DH, Yen J. Relevant data expansion for learning concept drift from sparsely labeled data. IEEE Trans. on Knowledge and Data Engineering, 2005, 17(3): 401–412. [doi: 10.1109/TKDE.2005.48]

[78] Haque A, Khan L, Baron M. Sand: Semi-supervised adaptive novel class detection and classification over data stream. In: Proc. of the 30th AAAI Conf. on Artificial Intelligence. New York: ACM, 2016. 1652–1658.

[79] Haque A, Khan L, Baron M, *et al*. Efficient handling of concept drift and concept evolution over stream data. In: Proc. of the IEEE 32nd Int'l Conf. on Data Engineering. Piscataway: IEEE, 2016. 481–492. [doi: 10.1109/ICDE.2016. 7498264]

[80] Sriwatanasakdi N, Numao M, Fukui K. Concept drift detection for graph-structured classifiers under scarcity of true labels. In: Proc. of the IEEE 29th Int'l Conf. on Tools with Artificial Intelligence. Piscataway: IEEE, 2017. 461−468. [doi: 10.1109/ICTAI.2017.00077]

[81] Li N. Clustering assumption based classification algorithm for stream data. Pattern Recognition and Artificial Intelligence, 2017, 30(1): 1−10 (in Chinese with English abstract). [doi: 10.16451/j.cnki.issn1003-6059.201701001]

[82] Masud MM, Gao J, Khan L, *et al*. Classification and novel class detection in data streams with active mining. In: Proc. of the Pacific-Asia Conf. on Advances in Knowledge Discovery and Data Mining. Berlin: Springer, 2010. 311−324. [doi: 10.1007/978-3-642-13672-6_31]

[83] Žliobaitė I, Bifet A, Pfahringer B, *et al*. Active learning with drifting streaming data. IEEE Trans. on Neural Networks and Learning Systems, 2014, 25(1): 27−39. [doi: 10.1109/TNNLS.2012.2236570]

[84] Arabmakki E, Kantardzic M. SOM-based partial labeling of imbalanced data stream. Neurocomputing, 2017, 262: 120−133. [doi: 10.1016/j.neucom.2016.11.088]

[85] Zhu X, Zhang P, Lin X, *et al*. Active learning from stream data using optimal weight classifier ensemble. IEEE Trans. on Systems Man and Cybernetics Part B, 2010, 40(6): 1607−1628. [doi: 10.1109/TSMCB.2010.2042445]

[86] Bifet A, Gavaldà R. Learning from time-changing data with adaptive windowing. In: Proc. of the 2007 SIAM Int'l Conf. on Data Mining. Philadelphia: SIAM, 2007. 443−448. [doi: 10.1137/1.9781611972771.42]

[87] Wang HY, Hu XG, Li PP. Semi-supervised short text stream classification based on vector representation and label propagation. Pattern Recognition and Artificial Intelligence, 2018, 31(7): 634−642 (in Chinese with English abstract). [doi: 10.16451/j.cnki.issn1003-6059.201807006]

[88] Gao J, Fan W, Jiang J, *et al*. Knowledge transfer via multiple model local structure mapping. In: Proc. of the 14th ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining. New York: ACM, 2008. 283−291. [doi: 10.1145/1401890.1401928]

[89] Kim Y, Park CH. An efficient concept drift detection method for streaming data under limited labeling. IEICE Trans. on Information and Systems, 2017, 100(10): 2537−2546. [doi: 10.1587/transinf.2017EDP7091]

[90] Tan CH, Lee V, Salehi M. Online semi-supervised concept drift detection with density estimation. arXiv: 1909.11251, 2019.

[91] Bifet A, Holmes G, Kirkby R, *et al*. Moa: Massive online analysis. Journal of Machine Learning Research, 2010, 11(5): 1601−1604.

[92] Elwell R, Polikar R. Incremental learning of concept drift in nonstationary environments. IEEE Trans. on Neural Networks, 2011, 22(10): 1517−1531. [doi: 10.1109/TNN.2011.2160459]

附中文参考文献:

[5] 赵兴旺, 梁吉业. 一种基于信息熵的混合数据属性加权聚类算法. 计算机研究与发展, 2016, 53(5): 1018−1028. [doi: 10.7544/issn1000-1239.2016.20150131]

[8] 郭躬德, 李南, 陈黎飞. 一种基于混合模型的数据流概念漂移检测算法. 计算机研究与发展, 2014, 51(4): 731−742. [doi: 10.7544/issn1000-1239.2014.20120582]

[14] 赵鹏, 周志华. 基于决策树模型重用的分布变化流数据学习. 中国科学: 信息科学, 2021, 51(1): 1−12. [doi: 10.1360/SSI-2020-0170]

[16] 徐文华, 覃征, 常扬. 基于半监督学习的数据流集成分类算法. 模式识别与人工智能, 2012, 25(2): 292−299. [doi: 10.16451/j.cnki.issn1003-6059.2012.02.010]

[29] 王涛, 李舟军, 颜跃进, 等. 数据流挖掘分类技术综述. 计算机研究与发展, 2007, 44(11): 1809−1815. [doi: 10.1038/onc.2012.370]

[33] 文益民, 强保华, 范志刚. 概念漂移数据流分类研究综述. 智能系统学报, 2013, 8(2): 95−104. [doi: 10.3969/j.issn.1673-4785.201208012]

[39] 丁剑, 韩萌, 李娟. 概念漂移数据流挖掘算法综述. 计算机科学, 2016, 43(12): 24−29,62. [doi: 10.11896/j.issn.1002-137x.2016.12.004]

[67] 胡学钢, 马利伟, 李培培. 一种基于 Tri-training 的数据流集成分类算法. 数据采集与处理, 2017, 32(5): 853−860. [doi: 10.16337/j.1004-9037.2017.05.001]

[81] 李南. 基于聚类假设的数据流分类算法. 模式识别与人工智能, 2017, 30(1): 1−10. [doi: 10.16451/j.cnki.issn1003-6059.201701001]

[87] 王海燕, 胡学钢, 李培培. 基于向量表示和标签传播的半监督短文本数据流分类算法. 模式识别与人工智能, 2018, 31(7): 634−642. [doi: 10.16451/j.cnki.issn1003-6059.201807006]

文益民(1969－), 男, 博士, 教授, 博士生导师, CCF 杰出会员, 主要研究领域为机器学习, 数据流分类, 媒体分析与数据挖掘.

刘帅(1994－), 男, 硕士, 主要研究领域为机器学习, 数据挖掘, 推荐系统.

缪裕青(1966－), 女, 博士, 教授, CCF 专业会员, 主要研究领域为机器学习, 数据挖掘, 情感分析.

易新河(1969－), 女, 助理研究员,主要研究领域为媒体数据挖掘, 教育数据分析.

刘长杰(1994－), 男, 硕士, 主要研究领域为机器学习, 半监督学习, 数据挖掘.