计算机科学
COMPUTER SCIENCE

面向延迟标签场景下的可解释信用评估模型

辛博，丁志军

引用本文

辛博，丁志军. 面向延迟标签场景下的可解释信用评估模型[J]. 计算机科学，2024，51(8)：45-55.

XIN Bo, DING Zhijun. Interpretable Credit Evaluation Model for Delayed Label Scenarios[J]. Computer Science, 2024, 51(8): 45-55.

计算机科学 COMPUTER SCIENCE
http://www.jsjkx.com
DOI:10.11896/jsjkx.230900107

面向延迟标签场景下的可解释信用评估模型

辛 博 丁志军
嵌入式系统与服务计算教育部重点实验室(同济大学) 上海 201804
上海市网络金融安全协同创新中心(同济大学) 上海 201804
(2232941@tongji.edu.cn)

摘 要 随着社会经济的快速发展,信贷业务在金融领域中扮演着越来越重要的角色,利用机器学习算法进行信用评估成为了当前主流的方法。然而,目前仍存在一些问题亟待解决,如延迟标签带来的有标签数据不充分、模型滞后性的问题,以及动态信用评估模型缺乏可解释性的问题。针对这些问题,提出了一种面向延迟标签场景的可解释信用评估模型。该模型在动态模型树的基础上进行了加权改进,结合了延迟标签更新算法和自适应阈值的伪标签选择策略,将延迟标签数据看作反馈数据和伪标签数据两种状态分别进行处理,平衡了有标签数据不充分和模型滞后带来的影响,并实现了模型的可解释性。最后,在一些合成和真实的信用评估数据集上对模型进行了实验,与其他主流的算法相比,其更好地权衡了预测性能和可解释性。

关键词:信用评估;延迟标签;可解释性;动态模型树;伪标签选择
中图分类号 TP3-05

Interpretable Credit Evaluation Model for Delayed Label Scenarios

XIN Bo and DING Zhijun
Key Laboratory of Embedded System and Service Computing of Ministry of Education(Tongji University),Shanghai 201804,China
Shanghai Network Finance Security Collaborative Innovation Center(Tongji University),Shanghai 201804,China

Abstract With the rapid development of social economy,credit business plays an increasingly important role in the financial field,and using machine learning algorithms for credit evaluation has become the mainstream method. However,there are still some problems to be solved,such as the inadequacy of labeled data and model lag caused by delayed labels,and the lack of interpretability in dynamic credit evaluation models. To address these problems,this paper proposes an interpretable credit evaluation model for delayed label scenarios. Built upon the foundation of dynamic model trees,the model incorporates weighted enhancements. It combines delayed label update algorithms and a pseudo-label selection strategy with adaptive thresholds,treating delayed label data as both feedback data and pseudo-label data,effectively mitigating the impacts of insufficient labeled data and model lag. Moreover,the model achieves interpretability. It is finally tested on some synthetic and real credit evaluation datasets,demonstrating superior balance between predictive performance and interpretability compared to other mainstream algorithms.

Keywords Credit evaluation,Delayed label,Interpretability,Dynamic model tree,Pseudo-label selection

1 引言

信用评估是金融领域的一个重要问题,它涉及到信贷机构对借款人的信用风险进行量化和预测,从而决定贷款的条件以及是否发放贷款[1]。是否发放贷款的决定通常被建模为二元分类,旨在区分信用良好和信用不良的借款人。几乎所有最先进的有监督分类算法都已被应用于信用评估,包括逻辑回归、神经网络、集成模型、决策树、支持向量机等[2]。

然而,信用评估领域仍然面临着一些挑战和问题,如延迟标签问题[3]就是其中之一。延迟标签,指在信用评估过程中,借款人的真实还款情况需要经过一段时间才能观察到。在此之前,信贷机构需要根据借款人的其他信息来预测其还款概率,并做出贷款决策。在这种情况下,信贷数据的标签存在不同程度的延迟。传统的有监督模型只能通过有标签的数据进行训练,而不能利用延迟标签数据,这会导致数据不充分的问题。尽管随后数据的标签会到达,为模型提供相应的反馈,但由于时间跨度和反馈数据的分布可能已经发生变化,此时再对模型进行调整可能会导致预测结果出现偏差,使得模型具有滞后性。因此,信用评估首先需要一个合适的策略来补充当前时刻的有标签数据,尽可能准确地反映最新数据的分布。其次,由于借款人的行为具有不确定性,其还款时间可能在合同期内波动,存在提前还款、按期还款、违约、审查等情况[4],因此同一批数据的反馈标签可能延迟在各个时间步到达,而不同批次的反馈标签也可能会在同一时间步到达。因此,为了有效地进行信用评估,我们还需要设计一个合适的延迟标签处理算法,以确保对不同延迟程度的反馈数据进行

到稿日期:2023-09-19 返修日期:2023-12-13
通信作者:丁志军(dingzj@tongji.edu.cn)

准确反映，继而修正模型。

另外，根据 GDPR“解释权”和 ECOA 等现有法规，借款人有权了解贷款被拒绝的原因。同时，贷方也需要了解模型的预测，以确保做出正确的决定。因此，信用评估模型必须是可解释的，才能被金融机构采用。可解释性目前缺乏统一的定义，并且无法以标准化方式进行测量[5]，因此可解释性通常由启发式度量来表示，例如模型大小或复杂性，也就是说，模型越简单，就越容易解释。类似地，决策树的可解释性可以通过分裂节点的数量或树的深度来量化[6]。

目前，在延迟标签的背景下，动态模型的可解释性问题尚未充分研究和解决。一方面，现有的增量学习模型通常以提高预测性能为代价，牺牲了可解释性，例如神经网络[7]、集成学习[8]等模型虽然具有出色的非线性拟合能力，但其内部结构和运算过程难以理解和解释；另一方面，采用单一的简单模型虽然具有明确的逻辑规则，但难以很好地处理延迟标签问题。因此，如何设计一个既能处理延迟标签，又具有高可解释性的信用评估模型，是一个亟待解决的问题。

为了解决这个问题，本文提出了一种基于改进动态模型树的信用评估模型。在动态模型树的基础上做出改进，引入了加权的思想，提高了其处理不同状态延迟标签数据的能力，并针对性地设计了延迟标签更新算法和自适应阈值的伪标签选择策略，它们将为改进的动态模型树提供合适的权重数据。具体来说，在预更新阶段，通过延迟标签更新算法调整标签刚刚到达的反馈数据权重，同时纠正伪标签错误可能引发的影响。接着进行动态模型树的预更新。在正式更新阶段，模型首先基于最新时间步的特征向量进行分类预测，同时通过自适应阈值的伪标签选择策略筛选出高可靠性的伪标签数据，将其与有标签数据结合，进一步更新模型。最终完成面向延迟标签场景的可解释信用评估模型，模型的具体流程如图 1 所示。

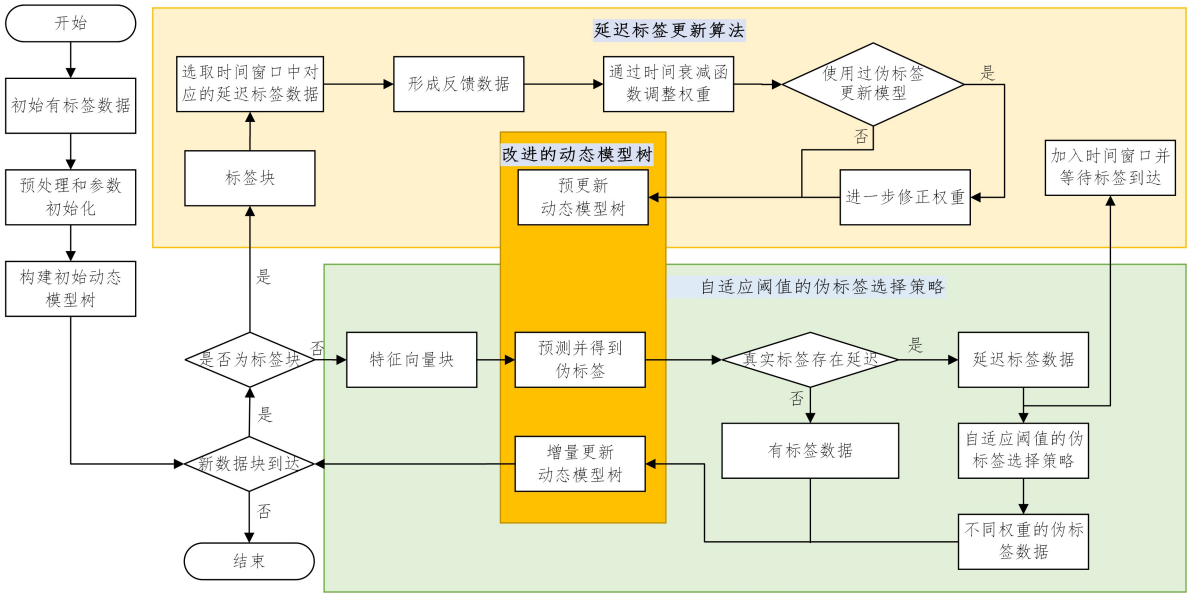


图 1 面向延迟标签场景下的可解释信用评估模型的总体框架流程

Fig. 1 Flow chart of interpretable credit evaluation model for delayed label scenario

本文的主要贡献可以归纳为以下 3 个方面。

1)提出了基于改进动态模型树的信用评估模型，该模型在保持可解释性的同时引入了加权机制，增强了其处理伪标签状态和反馈标签状态的延迟标签数据的能力，有助于更好地适应信用评估模型的需求。

2)提出了一种自适应阈值的伪标签选择策略。该策略的关键创新在于动态选择不同可信程度的伪标签数据，以解决当前时刻标签数据不足的问题。另外，通过采用移动平均策略，阈值允许在模型更新时自适应地调整，以适应数据分布的变化。

3)提出了一种延迟标签更新算法，使用时间衰减函数对反馈数据的权重进行调整。该算法同时修正了错误分类的伪标签，有助于确保模型在后续训练中朝着准确的方向发展。

2 相关工作

2.1 延迟标签问题

在研究中，通常假设给定一个数据流，每个实例在性能评估后都被赋予了正确的标签。然而，在某些实际应用中，这一假设是无效的。例如，在信贷行业，借款人是否能按时还款只能在经过几个周期后才能确定；在天气预测方面，明天是否会下雨的真实标签只有在后天才能获得。在以上情景中，数据和标签之间存在一定的延迟现象，被称为延迟标签问题。

延迟标签问题按照标签的使用方式主要被分为 3 类：前处理、后处理以及两阶段处理。前处理的方法主要是将延迟标签数据当作无标签或伪标签数据来使用。Dyer 等[9]提出了压缩对象样本提取框架，该框架利用初始的有标签数据构建每个类的数据分布中心，然后使用新的无标签数据来调整这些中心。但是这种方法考虑的是极端延迟场景，即标签永远不会到达，这与信用评估的情况并不匹配。Gao 等[10]建立了一个由有监督模型和半监督模型组成的双生网络，通过检测概念漂移、知识蒸馏和自适应权重的调整来解决延迟标签问题，比集成学习模型表现更好。前处理方法的优点在于保证了最新时刻的数据被充分利用，但无标签数据的训练容易受到分类器性能、概念漂移等因素的影响，导致预测结果不

准确,甚至对后续更新产生重大影响。后处理的方法是将延迟标签数据用作有监督模型的历史反馈数据。Kuncheva 等[11]首先研究了标签不会立即到达的情形,他们使用了两个在线最近邻分类器来研究延迟标签的 3 种事后处理策略。然而,这项研究未考虑数据随时间的演变。Gao 等[12]首次尝试了迁移学习,提出了一种基于 KNN 的改进 TrAdaBoost 模型,通过主动检测概念漂移,模型筛选了源域数据中未发生漂移的部分,并用这些数据构建新的分类器。实验证明,该方法可以很好地应对延迟标签场景下的突然性概念漂移问题。后处理方法尽管充分利用了标签反馈后的数据信息,但未考虑特征早已到达的事实。这可能导致两个问题:1)最新时刻的有标签训练数据不足;2)模型滞后,需要反馈信息才能充分完善模型。

最后,两阶段处理方法同时考虑了前处理和后处理的情况,目前被认为是处理延迟标签问题较为有效的方法。Pozzolo 等[13]提出了一个具有延迟反馈和主动学习的欺诈检测框架,将延迟标签数据视为反馈数据和无标签数据来进行两阶段处理。然而,这项研究仅考虑了恒定延迟的理想情况,并且模型只保留了最新时间窗口内的分类器,可能导致灾难性的遗忘问题。Das 等[14]提出了 SkipE-RNN 模型,它通过动态演化的跳跃连接机制为未标记的数据分配伪标签,当获得标签反馈后,生成的伪标签与真实标签进行比较,如果发现标签不匹配,模型将受到惩罚。尽管这种方法似乎提供了一种合理的途径来解决延迟标签问题,但由于采用了神经网络模型且具有较高的复杂性,其在可解释性方面存在挑战。

综上所述,一个优秀的信用评估模型在处理延迟标签时,需要满足以下几个要素:首先,在标签尚未到达的情况下,延迟标签数据需要能够有效地参与模型训练,以确保在数据稀缺的情况下也能进行准确预测;其次,一旦数据标签可用,模型应当能够以合理的方式进行修正,以纠正潜在的错误;最为重要的是,该模型必须具备高度的可解释性,以使模型的决策和预测能够被清晰地理解和解释。

2.2 节将介绍目前信用评估的可解释性的相关工作,并探讨这些方法在延迟标签场景下的动态信用评估环境中的应用潜力。

2.2　信用评估的可解释性

在学术界,高级信用评分模型的准确性和可解释性之间的权衡一直存在争议[15]。监管机构也致力于揭示机器学习技术带来的新风险,并强调在贷款领域提高透明度和可解释性的必要性。通常,可解释性分为事后解释和内在解释两种方式。前者主要是利用一些额外的解释方法,与信用评估中的"黑箱"模型一起使用,如 LIME[16]和 SHAP[17]等。这些方法的优点在于它们不依赖于特定模型的内部结构,同时专注于局部解释特定样本的模型预测结果。然而,在动态数据流的环境中,数据持续涌入,构建额外的解释方法可能会耗费巨大的资源,因为每个新数据点都可能需要重新计算解释,这在资源和实时性方面存在挑战。

相反,内在解释指构建本身就具备可解释性的预测模型。通常情况下,当模型的复杂性足够低时,人们更容易分析其中的逻辑和特征关系。Dong 等[18]提出了一个两阶段规则提取方法,将树集成模型转换为具有较少规则约束的简化模型,相比随机森林更好地权衡了可解释性与预测性能之间的关系。Alangari 等[19]提出了一种从本质上可解释 GMM 的方法,通过识别有区别的和重叠的特征来确定聚类的特征以及聚类权重,从而提供了全局视角。然而,这些方法只能为每个单独时刻提供解释,在动态模型更新过程中还需要解释模型复杂性变化的原因。

目前,决策树是数据流分类问题中最常用的内在可解释模型,通过分析决策树在每个时刻的分支策略或剪枝操作,可以回答增量更新中产生的变化。Domingos 等[20]首先提出了 Hoeffding 树,它是一种增量决策树归纳算法,通过 Hoeffding 不等式来决定是否要在树的节点上进行拆分。该算法可以保证在较少数据的情况下做出决策,同时保持较高置信度,但是在处理高维数据和概念漂移问题时表现不佳,树的构建和更新会变得十分复杂。Potts 等[21]提出了一种线性模型树,通过引入增量节点分裂规则,允许树的节点在数据不断增加时自动扩展,同时提出了停止树增长和修剪的增量方法。但是,这种方法仍然采用统计测试的方法来构建模型树。Haug 等[22]提出了动态模型树,根据节点处简单模型的损失变化来维护模型树,并保证了合理的一致性和最小性,有助于更直观和可解释地在线学习。

动态模型树的框架不依赖于 Hoeffding 不等式、启发式纯度测量或显式概念漂移检测机制,从而消除了现有增量决策树的一些主要弱点。然而,该框架仍专注于处理一般的有标签数据流,无法明确地区分延迟标签场景下不同状态的延迟标签数据。基于此,进行了相应的改进,以提高其在延迟标签场景中的应用能力(将在 3.2.1 节中详细介绍)。2.3 节将简要介绍动态模型树的特性和工作原理。

2.3　动态模型树

与传统的 Hoeffding 增量树不同,动态模型树在每个节点中维护一个简单模型,节点的生长和修剪也依赖于模型损失的变化。动态模型树的每一个节点都可以由一组时间索引 $S_t \subseteq \{1,\cdots,t\}$ 来表示,这组时间索引对应于截至到时间步 t 时到达该节点的特征向量。令 X_{S_t} 和 Y_{S_t} 分别代表特征向量和标签,θ_{S_t} 代表节点中简单模型的参数,动态模型树的目标是模型的整体损失尽可能地小,因此参数需要满足:

$$\theta_{S_t}^{*}=\underset{\theta_{S_t}}{\operatorname{argmin}} L(\theta_{S_t},Y_{S_t},X_{S_t})=\underset{\theta_{S_t}}{\operatorname{argmin}} \sum_{t\in S_t} L(\theta_t,Y_t,X_t) \tag{1}$$

假设每个时间步之间是独立的,那么在增量学习的过程中,可以用基于梯度的近似方法[23]独立更新每个时间步 θ_t,前一个时间步的参数可以作为后一个时间步的先验参数。

为了实现高可解释性的增量学习,动态模型树还定义了两个属性:一致性和最小性。

1)一致性。一致性保证了节点分裂时,子节点损失一定小于父节点损失。令 S_t 表示该节点特征向量对应的时间索引,$C_t \subseteq S_t$ 表示分裂后左子树的时间索引,$\bar{C}_t = S_t \backslash C_t$ 表示分裂后右子树的时间索引。那么分裂前后,树整体损失会发生相对应的变化,可以表示为:

$$G_{S_t,C_t}=L(\theta_{S_t},Y_{S_t},X_{S_t})-L(\theta_{C_t},Y_{C_t},X_{C_t})-L(\theta_{\bar{C}_t},Y_{\bar{C}_t},X_{\bar{C}_t}) \tag{2}$$

其中，G_{S_t,C_t} 表示损失变化量。动态模型树的目标是找到最佳的分裂决策，使得 G_{S_t,C_t} 最大化且 $G_{S_t,C_t}\geqslant 0$。

2)最小性。动态模型树使用新的候选分裂或叶节点替换现有的内部节点，从而确保模型的最小性。使用一组时间索引 I_t 表示一个内部节点对应的特征向量。同样地，这个内部节点的每个叶节点使用集合 J_t 来表示。那么对于一个可能的候选分裂，该节点处的损失的变化可以表示为：

$$G_{I_t,C_t}=\sum_{J_t\subseteq I_t}L(\theta_{J_t},Y_{J_t},X_{J_t})-L(\theta_{C_t},Y_{C_t},X_{C_t})-L(\theta_{\bar{C}_t},Y_{\bar{C}_t},X_{\bar{C}_t}) \tag{3}$$

其中，$C_t\subseteq I_t$ 表示候选分裂中左子树的特征向量对应的时间索引，$\bar{C}_t=I_t\backslash C_t$ 表示右子树的特征向量对应的时间索引。如果 $G_{I_t,C_t}>0$，则使用一个新的内部节点进行替换，同时添加两个新的叶节点。或者，也可以将子树使用一个叶节点来替换，为此需要比较当前内部节点的损失和它的子树的损失，相应的损失变化为：

$$G_{I_t}=\sum_{J_t\subseteq I_t}L(\theta_{J_t},Y_{J_t},X_{J_t})-L(\theta_{I_t},Y_{I_t},X_{I_t}) \tag{4}$$

如果 $G_{I_t}>0$ 且 $G_{I_t}\geqslant G_{I_t,C_t}$，则直接将子树替换为叶节点，以获得更小的树。式(3)和式(4)同时保证了模型最小化。

最后，动态模型树更新时需要重新计算候选分裂损失 $L(\theta_{C_t},Y_{C_t},X_{C_t})$ 和 $L(\theta_{\bar{C}_t},Y_{\bar{C}_t},X_{\bar{C}_t})$。在当前节点 θ_{S_t} 的基础上执行一次梯度更新，可以用热启动的方式优化候选分裂的参数 θ_{C_t}，从而得到候选分裂损失的近似值：

$$L(\theta_{C_t},Y_{C_t},X_{C_t})\approx L(\theta_{S_t},Y_{C_t},X_{C_t})-\frac{\omega}{|C_t|}\|\nabla_{\theta_{S_t}}L(\theta_{S_t},Y_{C_t},X_{C_t})\|_2^2 \tag{5}$$

通过式(5)可以近似不同候选分裂的损失，而无需维护相应的简单模型。此外，利用父节点优化过程中计算的梯度可以进一步提高效率。

3 面向延迟标签场景下的信用评估模型

3.1 延迟标签定义

首先对延迟标签场景下的数据进行符号化定义。假设数据流 $D=\{D_1,D_2,\cdots,D_t,\cdots\}$，$t$ 表示时间步，根据量纲的不同，它可以表示天、周或者月。时间步 t 的数据块表示为 $D_t=(Z_t,\Theta_t)$。其中，集合 $Z_t=\{x_t^i\}_{i=1}^{N_t}$ 表示时间步 t 时到达的 N_t 个数据的特征向量，Θ_t 定义为截至到时间步 t 到达的特征向量的延迟标签。集合 Θ_t 既可以是空的，也可以是非空的。当 Θ_t 非空时，它可以表示为：

$$\Theta_t=\{y_t^{(k,i)}\}_{(k,i)\in L_t} \tag{6}$$

其中，L_t 由元组(k,i) 构成，k 表示标签所对应的特征向量到达的时间，i 表示特征向量位于原数据块中的位置，那么 $y_t^{(k,i)}$ 则表示该特征向量的标签在时间步 t 到达。需要注意的是，标签一定是不早于特征向量到达，因此 $t\geqslant k$，那么延迟可以表示为 $\tau=t-k$。在本文的场景中，仅考虑有限的延迟标签问题，因此设定 $\tau\in[0,\Delta t_{\max}]$，$\Delta t_{\max}$ 为允许的最大延迟时间，0 表示没有延迟，即标签可以立即获得。

3.2 模型总体架构

模型主要由 3 个关键的模块组成，分别是改进的动态模型树、延迟标签更新算法和自适应阈值的伪标签选择策略。改进的动态模型树是最基础的模块，通过引入加权的思想，它允许不同状态的数据增量地更新模型。延迟标签更新算法和自适应阈值的伪标签选择策略是专门为解决延迟标签问题而设计的，它们将为改进的动态模型树提供动态的权重数据。模型的总体算法如算法 1 所示。

算法 1 延迟标签场景下的可解释信用评估模型

输入：初始有标签数据 D_0，不断到达的特征向量块 Z_t 和标签块 Θ_t，时间窗口 Ω_t

输出：每个时间步的最新信用评估模型

1. 对初始有标签数据 D_0 进行预处理，并对模型参数进行初始化定义，构建初始改进的动态模型树
2. t←1
3. while 特征向量块 Z_t 和标签块 Θ_t then
4. 通过标签块 Θ_t 和时间窗口 Ω_t 选取延迟标签数据，并构造成反馈数据 X_t^h
5. 使用延迟标签更新算法调整所有反馈数据权重 W_t^h
6. if 反馈数据使用过伪标签训练模型 then
7. 使用延迟标签更新算法，以进一步修正权重 W_t^h
8. end if
9. 使用加权的反馈数据 X_t^h 更新改进的动态模型树
10. 预测特征向量块 Z_t 并得到伪标签 $\tilde{Y}_t$
11. 筛选 Z_t 中有标签数据 X^t 和延迟标签数据 X_t^u
12. If 延迟标签数据 X_t^u 不为空 then
13. 使用 X_t^u 更新时间窗口 Ω_t
14. 使用伪标签选择策略评定 X_t^u 伪标签质量
15. 对不同质量的伪标签执行相应的权重调整，得到伪标签权重 W_t^u
16. end if
17. 使用 X_t^l 与 X_t^u 一同加权更新改进的动态模型树
18. t←t+1
19. end while

3.2.1 改进的动态模型树

在本文的延迟标签场景下，信贷数据主要分为最新有标签数据和延迟标签数据。而延迟标签数据又根据标签状态，被分为反馈数据和伪标签数据。由于这些数据在分布和质量上存在差异，最新有标签数据相对更为可信和有价值，因此应该赋予最高的权重。而伪标签数据和反馈数据受到诸如预测准确性和概念漂移等因素的影响，应该被赋予适度较低的权重。为了使动态模型树能够区分并处理它们，引入加权思想，首先将动态模型树的总体损失函数定义为：

$$L(\theta_{S_t},Y_{S_t},X_{S_t})=L(\theta_t,Y_t,X_t)+\lambda L(\theta_{S_{t-1}},Y_{S_{t-1}},X_{S_{t-1}}) \tag{7}$$

其中，$L(\theta_{S_{t-1}},Y_{S_{t-1}},X_{S_{t-1}})$ 表示截至到上一时间步 $t-1$ 的动态模型树的总体损失，λ 表示模型的遗忘因子。通过设置遗忘因子保证了模型中已训练的数据随时间变化时权重会逐渐降低，使得每个时间步的最新数据将拥有更高权重。

其次，不同于单纯的有标签数据流挖掘问题，本文同时存在 3 种不同标签状态的数据，因此当前时间步的特征向量 $\boldsymbol{X}_t=\{X_t^l,X_t^u,X_t^h\}$，标签 $Y_t=\{Y_t^l,Y_t^u,Y_t^h\}$。其中 $X_t^l\subseteq Z_t$，$Y_t^l\subseteq\Theta_t$，表示当前时间步的有标签数据和对应的真实标签；$X_t^u\subseteq$

Z_t 表示最新延迟标签数据，也是即将使用的伪标签数据，其伪标签 Y_t^u 由分类器预测得到；X_t^h 表示反馈数据，取自于时间窗口中的延迟标签数据，$Y_t^h \subseteq \Theta_t$ 表示刚刚到达的反馈标签。3 种形式的数据权重各不相同，因此对当前时间步的损失进行进一步定义：

$$L(\theta_t, Y_t, X_t) = L(\theta_t, Y_t^l, X_t^l, W_t^l) + L(\theta_t, Y_t^u, X_t^u, W_t^u) + L(\theta_t, Y_t^h, X_t^h, W_t^h) \tag{8}$$

其中，伪标签数据的权重 W_t^u 和反馈数据的权重 W_t^h 都是向量形式，它们分别由自适应阈值的伪标签选择策略和延迟标签更新算法计算获得，而有标签数据的权重 W_t^l 在均衡的数据流中由向量 1 表示。对于不均衡数据流，也可以按照类别适当地对权重再次进行调整。另外，为了方便表示时间步 t 的权重，定义 $W_t = \{W_t^l, W_t^u, W_t^h\}$。

最后，需要确定动态模型树中具体的损失函数。损失函数的选择依赖于简单模型，而简单模型可以是任意的。本文选用逻辑回归模型作为动态模型树中的简单模型，用于计算损失。同时，逻辑回归也可以通过提取不同子组的特征权重为模型进行局部解释。3 种形式的数据在简单模型中的处理方式相同。令 $f_\theta(x)$ 表示逻辑回归模型预测的概率，则逻辑回归的加权损失函数表示为：

$$L(\theta, Y, X, W) = -\sum_{i=1}^{n} w_i (y_i \log(f_\theta(x_i)) + (1-y_i)\log(1-f_\theta(x_i))) \tag{9}$$

除了损失函数层面的加权处理外，动态模型树的更新和维护也需要相应的改进。算法 2 给出了一个节点在时间步 t 的加权更新策略。

算法 2　节点在时间步 t 时的加权更新过程

输入：时间步 t 的特征向量 $\mathbf{X}_t$，标签 Y_t，权重 W_t；逻辑回归模型 f_{θ_t}；损失函数 L，梯度 ∇L；加权计数 n，遗忘因子 λ

输出：加权更新后的叶节点

*** 更新损失、梯度和加权计数 ***

1. 通过式(8)、式(9)计算当前时间步 t 的加权损失 $L(\theta_t, Y_t, X_t)$，并计算梯度 $\nabla_{\theta_t} L(\theta_t, Y_t, X_t)$
2. $n_t \leftarrow \sum W_t$
3. $L(\theta_{S_t}, Y_{S_t}, X_{S_t}) \leftarrow \lambda L(\theta_{S_{t-1}}, Y_{S_{t-1}}, X_{S_{t-1}}) + L(\theta_t, Y_t, X_t)$
4. $\nabla_{\theta_{S_t}} L(\theta_{S_t}, Y_{S_t}, X_{S_t}) \leftarrow \lambda \nabla_{\theta_{S_{t-1}}} L(\theta_{S_{t-1}}, Y_{S_{t-1}}, X_{S_{t-1}}) + \nabla_{\theta_t} L(\theta_t, Y_t, X_t)$
5. $n_{S_t} \leftarrow \lambda n_{S_{t-1}} + n_t$

*** 维护候选分裂损失并计算损失变化，以左节点为例 ***

6. $G_{max} \leftarrow 0.0$
7. $C_{best} \leftarrow$ None
8. for all 候选分裂集合 C do
9. $Y_t^C \subseteq Y_t$；$X_t^C \subseteq X_t$；$W_t^C \subseteq W_t$
10. 通过式(8)、式(9)计算划分给左节点的数据损失 $L(\theta_t, Y_t^C, X_t^C)$，并计算梯度 $\nabla_{\theta_t} L(\theta_t, Y_t^C, X_t^C)$
11. $n_t^C \leftarrow \sum W_t^C$
12. $L(\theta_{S_t}, Y_{C_t}, X_{C_t}) \leftarrow \lambda L(\theta_{S_{t-1}}, Y_{C_{t-1}}, X_{C_{t-1}}) + L(\theta_t, Y_t^C, X_t^C)$
13. $\nabla_{\theta_{S_t}} L(\theta_{S_t}, Y_{C_t}, X_{C_t}) \leftarrow \lambda \nabla_{\theta_{S_{t-1}}} L(\theta_{S_{t-1}}, Y_{C_{t-1}}, X_{C_{t-1}}) + \nabla_{\theta_t} L(\theta_t, Y_t^C, X_t^C)$
14. $n_{C_t} \leftarrow \lambda n_{C_{t-1}} + n_t^C$
15. $L(\theta_{C_t}, Y_{C_t}, X_{C_t}) \leftarrow$ 式(5)
16. $G_{S_t, C_t} \leftarrow$ 式(2)
17. if $G_{S_t, C_t} > G_{max}$ then
18. $G_{max} \leftarrow G_{S_t, C_t}$
19. $C_{best} \leftarrow C$
20. end if
21. end for

*** 判断分裂条件是否满足 ***

23. if $G_{max} \geqslant 0$ then
24. 按照最佳分裂 C_{best} 分裂
25. end if

3.2.2　延迟标签更新算法

在本文的场景下，一些延迟标签数据会在获得反馈标签后，加入模型进行补充训练，因此首先需要构建一个时间长度为 T 的时间窗口，用来保留过去 T 个时间步标签尚未到达的数据。时间步 t 得到的延迟标签数据可以表示为：

$$X_t^u = Z_t \backslash Z_t \otimes \Theta_t \tag{10}$$

其中，$a \otimes b$ 被定义为查找运算，即在特征向量块 a 中寻找与标签块 b 中标签对应的特征向量，那么 $Z_t \otimes \Theta_t$ 可以表达为标签和特征向量同时到达的数据，即有标签数据 X_t^l。去除有标签数据部分，得到的便是延迟标签数据。时间窗口可以构建为 $\Omega_t = \{X_{t-T}^u, \cdots, X_{t-2}^u, X_{t-1}^u\}$，当时间窗口已满且下一时间步数据到达时，删除最早进入时间窗口的特征向量块 X_{t-T}^u，加入新的延迟标签数据块 X_t^u。

在使用延迟标签数据 X_t^u 更新时间窗口 Ω_t 前，需要先取出时间窗口中与标签块对应的延迟标签数据并构造成反馈数据，进而调整数据的权重，并增量更新模型。该过程可以表示为：

$$X_t^h = \bigcup_{i=t-T}^{t-1} X_i^u \otimes \Theta_t \tag{11}$$

X_t^h 中包含了时间步 $[t-T, t-1]$ 内标签刚刚到达的反馈数据的特征向量，这些数据由于到达的时刻不同，往往呈现不同的分布。一般情况下，越早到达的数据，与最新时刻数据发生偏移的程度越大。基于此，需要设计一个时间衰减函数来计算反馈数据的权重，随着时间的推移，时间窗口内的特征向量将自适应地降低权重。时间衰减函数可以表示为：

$$w_t^{h(i)} = e^{-\varphi(t-k)} = e^{-\varphi\tau} \tag{12}$$

其中，$w_t^{h(i)}$ 表示单个特征向量的历史权重，φ 是一个正的衰减常数，k 表示特征向量到达的时间，t 表示标签到达的时间，$\tau = t-k$ 表示延迟。延迟 τ 越大，表明反馈数据到达越早；$w_t^{h(i)}$ 越小，即反馈数据的权重越低。这样可以反映出反馈数据的概念漂移，使得模型更关注近期数据。一般来说，可以根据数据特征和时间步量纲来选择合适的 φ 值。

然而，同一批数据由于延迟时间不同，它们可能在不同的时刻作为反馈数据加入模型训练。出于数据分布一致性的原则，它们应该设置相同的权重，但时间衰减的权重函数会让延迟越大的数据表现出越小的权重。为此，一个简单且有效的修正方法是对模型的遗忘因子进行进一步定义，即：

$$\lambda = e^{-\varphi} \tag{13}$$

该方法的出发点在于权重的作用更多地表现在模型的损失函数及其后续的更新上，因此模型中数据的损失应该与反馈数据呈现相同的衰减速度，以确保反馈数据参与训练时在损失方面的贡献与同一时刻更早到达的数据保持一致。

最后，为了减轻数据不充分引发的问题，我们曾使用一部分延迟标签数据进行伪标签训练。因此，在标签可用后，需要将反馈数据的真实标签与伪标签进行对比，重新评估并调整它们的权重，以尽可能地纠正伪标签错误导致的模型偏差。

按照伪标签的状态，将它们分为 3 类。第一类是未经过伪标签训练的数据，对于这类数据，直接使用反馈后的信息加入模型训练；第二类是进行过伪标签训练且伪标签正确的数据，这部分数据事实上已经正确地被模型学习，因此不必再次训练；第三类是进行过伪标签训练但是伪标签错误的数据，这表明当时的模型不能很好地区分这类数据，因此需要反馈后的数据信息纠正。这里采用了一个简单的思路，通过给予这类数据一个更大的权重进入模型再训练，从而尽可能地覆盖掉之前错误的影响。令 $y_t^{h(i)}$ 表示延迟到达的真实标签，$\tilde{y}_t^{h(i)}$ 表示伪标签，那么 3 类数据经过再调整后的权重表示为：

$$w_t^{h(i)}=\begin{cases}e^{-\varphi\tau}, & \text{if } \tilde{y}_t^{h(i)} \text{ is None}\\ 0, & \text{if } \tilde{y}_t^{h(i)}=y_t^{h(i)}\\ \kappa e^{-\varphi\tau}, & \text{if } \tilde{y}_t^{h(i)}\neq y_t^{h(i)}\end{cases} \tag{14}$$

其中，κ 是一个大于 1 的调节因子，可以根据错误伪标签带来的影响程度进行调整，以增大它在损失函数中的影响力。延迟标签更新算法通过从时间窗口中构造反馈数据并进行两次权重调整，实现了对新旧数据的平衡处理，它的具体流程如算法 3 所示。

算法 3 延迟标签更新算法

输入：当前时间步 t 的特征向量块 Z_t，标签块 Θ_t，时间窗口 Ω_t，调节因子 κ

输出：反馈数据更新后的改进的动态模型树

1. 根据式(11)构造反馈数据 X_t^h，标签为 Y_t^h
2. 根据式(10)找出 Z_t 中的延迟标签数据，更新 Ω_t
3. 计算反馈数据 X_t^h 的延迟 τ，采用式(12)计算权重
4. for all $x_t^{h(i)}\in X_t^h$ do
5. 查询 $x_t^{h(i)}$ 的伪标签 $\tilde{y}_t^{h(i)}$ 和真实标签 $y_t^{h(i)}$
6. if $\tilde{y}_t^{h(i)}$ is None then
7. $w_t^{h(i)}\leftarrow e^{-\varphi\tau}$
8. else if $\tilde{y}_t^{h(i)}=y_t^{h(i)}$ then
9. $w_t^{h(i)}\leftarrow 0$
10. else
11. $w_t^{h(i)}\leftarrow \kappa e^{-\varphi\tau}$
12. end for
13. 所有样本 $x_t^{h(i)}\in X_t^h$ 的权重构成集合 W_t^h，使用 W_t^h，X_t^h 和 Y_t^h 更新改进的动态模型树

3.2.3 自适应阈值的伪标签选择策略

常见的扩充有标签数据的方法是，使用无标签数据预测得到的伪标签，与有标签数据一起更新模型。一个比较主流的伪标签选择策略是设定置信度阈值。但是，这种方式往往需要设定很高的阈值才能保证数据的质量，使得伪标签的利用率低。另外，固定阈值的方式不能适应模型的动态变化，无法反映数据的分布情况，在随时间变化的模型中不够灵活和准确。针对上述问题，本文提出了一种自适应阈值的伪标签选择策略，在保证高于阈值的伪标签数据被充分利用的同时，允许低于阈值的部分伪标签以较低的权重进行训练，从而增加伪标签数据的利用率。另外，为了适应数据分布的变化，使用移动平均的方法根据最新数据自适应地调整阈值，可以更好地应对数据概念漂移带来的影响。

根据中心极限定理，当模型预测的数据量足够大时，总体数据预测的均值将逐渐接近高斯分布。因此，这里给出一个假设：在时间步 t 时，一个优秀的分类器 f_t 的预测概率服从均值为 μ_t、方差为 σ_t^2 的高斯分布。定义 μ_t 为阈值，那么高于 μ_t 的数据被认为是高质量的伪标签数据，给予其与有标签数据相同的权重进行训练。另外，低于阈值 μ_t 的数据中可能包含新的特征信息，因此筛选并利用这部分数据是有必要的。这里引入 Z-score 的方法，它的计算式为：

$$z_t^{(i)}=\frac{\hat{y}_t^{u(i)}-\mu_t}{\sigma_t} \tag{15}$$

其中，$\hat{y}_t^{u(i)}$ 表示单个数据的伪标签预测概率；$z_t^{(i)}$ 是衡量该数据的预测概率距离均值的标准差个数，一般情况下，68% 数据的 $z_t^{(i)}\in[-1,1]$，95% 数据的 $z_t^{(i)}\in[-2,2]$，99% 数据的 $z_t^{(i)}\in[-3,3]$。通常认为置信度高于 95% 的数据是较为可靠的，由于是向下选择，因此筛选 $z_t^{(i)}\in[-2,0]$ 的数据作为伪标签数据的进一步补充。这部分数据的质量并不高，因此需要使用较低的权重来训练它们。伪标签数据权重的定义为：

$$w_t^{u(i)}=\begin{cases}1, & \hat{y}_t^{u(i)}>\mu_t\\ e^{\rho\frac{\hat{y}_t^{u(i)}-\mu_t}{2\sigma_t}}, & \mu_t-2\sigma_t\leqslant\hat{y}_t^{u(i)}\leqslant\mu_t\\ 0, & \hat{y}_t^{u(i)}<\mu_t-2\sigma_t\end{cases} \tag{16}$$

其中，大于阈值 μ_t 的数据拥有最高权重；Z-score 位于$[-2,0]$区间的数据通过指数函数的形式给予一个衰减的权重，ρ 用于调节权重的衰减速率；其他的伪标签数据被认为是低质量数据，预测错误率较高，不参与伪标签训练。

随着时间的推移，数据的分布发生改变，因此分类器也需要做出调整来适应变化。使用移动平均的方法实时调整阈值，在上一时间步均值为 μ_{t-1}、方差为 σ_{t-1}^2 的高斯分布的基础上，更新当前时间步的数据预测概率分布的变化。均值的更新可以表示为：

$$\mu_t=m\mu_{t-1}+(1-m)\bar{y} \tag{17}$$

其中，$m\in(0,1)$ 表示阈值移动的速度，代表上一时间步的衰减程度，根据时间步的量纲和数据漂移程度可以设置不同的值；$\bar{y}$ 表示当前时间步预测数据概率的均值。方差的更新方式类似于式(17)，表示为：

$$\sigma_t^2=m\sigma_{t-1}^2+(1-m)\sigma^2 \tag{18}$$

其中，σ^2 是当前时间步预测数据概率的方差。通过对当前时间步的数据预测，得到数据预测概率的近似高斯分布，如果数据和模型均处于稳定状态，那么 $\mu_{t-1}\approx\bar{y}$，μ_t 不会发生剧烈变化；反之，模型将朝着最新数据分布的状态进行移动。自适应阈值的选择策略如算法 4 所示。

算法 4 自适应阈值的伪标签选择策略

输入：当前时间步 t 的特征向量块 Z_t，标签块 Θ_t，动态模型树分类器 f_t，阈值移动速度 m

输出：更新后的改进的动态模型树

1. 使用分类器 f_t 预测 Z_t，得到伪标签 $\tilde{\mathbf{Y}}_t$ 和概率 $\hat{Y}_t$

2. 计算 $\hat{Y}_t$ 的均值 $\bar{y}$ 以及方差 σ^2

3. $\mu_t \leftarrow m\mu_{t-1}+(1-m)\bar{y}$

4. $\sigma_t^2 \leftarrow m\sigma_{t-1}^2+(1-m)\sigma^2$

5. 根据式(10)找出 Z_t 中的延迟特征向量块 X_t^h，其标签 $\tilde{Y}_t^h \subseteq \tilde{Y}_t$，概率 $\hat{Y}_t^h \subseteq \hat{Y}_t$

6. $W_t^h \leftarrow \varnothing$

7. for all $x_t^{u(i)} \in X_t^h$ do

8. $\quad \hat{y}_t^{u(i)} \in \hat{Y}_t^h$

9. $\quad w_t^{u(i)} \leftarrow$ 式(16)

10. $\quad W_t^h \leftarrow W_t^h \cup w_t^{u(i)}$

11. end for

12. 使用 W_t^h，X_t^h 和 $\tilde{Y}_t^h$ 连同有标签数据更新动态模型树

4　实验研究与结果分析

4.1　实验环境

所有模型和实验均以 Python(3.9.16)实现，并运行在 CPU 为 Intel(R) Core(TM) i5-13500HX(2.50 GHz)，RAM 为 32 GB 的机器上，操作系统是 Windows11。

对于实验结果，本文采用了先测试后训练的评估策略[24]，这是一种在数据流学习中常用的评估方法。由于数据流具有时间相关性，并且随着时间不断地产生，因此无法对数据流进行随机划分和采样，也无法保证数据集的不同排列具有相同或相似的分布和特征。因此，与静态的批量评估相比，数据流评估缺乏统计意义，多次交叉实验取平均的方法并不适用。作为替代，本文报告了模型在整个生存空间的平均性能及其标准差。

在模型的参数方面，本文给出了一些建议。首先是学习率设置，由于训练初期模型不稳定，因此建议设定一个较高的初始学习率，一般情况下为 0.05，随后进行自适应的调整。模型遗忘因子 λ 与延迟标签衰减常数 φ 相关，根据时间步量纲不同和数据漂移程度不同，φ 的取值也不尽相同。实验中采用网格搜索的方法进行选择，搜索范围为{0.01，0.05，0.1，…，0.5}。超参数 κ 用于尽可能覆盖之前错误标签的影响，本文的实验中尝试在{2，3，5，10，…}内取最好的效果。伪标签权重衰减速率 ρ 默认为 1；阈值移动更新速度 m 同样需要根据数据漂移程度进行不同的取值，建议的搜索范围为{0.5，…，0.9，0.95，0.99}。

4.2　主要数据集描述

为了评估本文算法的有效性，分别在合成数据集和真实的信用评估数据集上进行了实验，所有的数据集都是二分类。由于合成数据集没有明确定义延迟标签的行为，因此需要构造延迟标签时间，设定最大延迟时间 $\Delta t_{max}=10$，并假设每个样本的延迟标签时间在$[0,\Delta t_{max}]$内随机生成。数据集的基本信息如表 1 所列，具体描述如下。

1)SEA 数据集[25]：具有 3 个连续属性的合成数据集，其中只有前两个属性 α_1，α_2 与分类有关。类别的决策边界由方程 $\alpha_1+\alpha_2=\beta$ 给出，可以通过添加噪音影响 β 值的变化模拟数据流的概念漂移情况。

2)HyperPlane 数据集[26]：基于超平面生成器生成的增量概念漂移数据集，超平面将空间分为两个不相连的部分，通过改变权重 w_i 的相对大小来改变超平面的方向和位置。该数据集一直处于增量概念漂移的状态。

3)Agrawal 数据集[27]：具有 3 个分类属性和 6 个连续属性的合成数据集，它被用于模拟贷款申请。该数据集使用 10 个不同的函数将实例映射到两个不同的类，因此可以通过改变函数来模拟概念漂移。实验中为该数据集设定了 3 段增量漂移，其他时刻则保持稳定。

4)LC 数据集：P2P 借贷平台 Lending Club 的贷款数据，包括贷款者的详细信息与借贷情况。本实验选取了 2012 年到 2017 年间连续 60 个月的贷款数据，并将每个月的数据视作一个数据块。贷款最大期限为 3 年，因此最大延迟时间设定为 $\Delta t_{max}=36$。

5)Bank Loan 数据集：该数据集来自某信贷平台的贷款记录，记录了 2012－2018 年贷款者申请贷款的详细信息，实验选取其中连续 70 个月的贷款数据。贷款申请期限分为 3 年和 5 年两种情况，因此分别设定最大延迟时间为 $\Delta t_{max}=36$ 和 $\Delta t_{max}=60$。

6)ppdai 数据集：该数据集为拍拍贷提供的真实业务数据，包含 2015－2017 年的贷款数据。由于其中存在大量未完成的贷款信息，经过清洗后选择了其中连续 20 个月的数据进行实验。允许的最大贷款期限为 12 个月，因此最大延迟时间设定为 $\Delta t_{max}=12$。

表 1　数据集的基本信息

Table 1　Statistics of datasets

Data Set	# Samples	# Features	Chunk Size
SEA	1 000 000	3	10 000
Agrawal	1 000 000	9	10 000
HyperPlane	500 000	50	5 000
LC	1 517 239	145	4 752～41 140
Bank Loan	713 680	47	3 433～26 980
ppdai	118 767	37	2 104～8 848

4.3　评价指标

4.3.1　预测性能

针对二分类问题，可以根据分类器的预测标签和数据的真实标签将它们分为 4 种情况，构成的混淆矩阵如图 2 所示。

Real \ Predict	TURE	FALSE
TURE	TP	FN
FALSE	FP	TN

图 2　混淆矩阵

Fig. 2　Confusion matrix

本文采用 F1-score 和 AUC 作为模型性能的评价指标。对于合成数据集，采用 F1-score 进行评估，它是精确率和召回率的调和平均数。

精确率可以表示为：

$$Precision=\frac{TP}{TP+FP} \tag{19}$$

召回率可以表示为：

$$Recall=\frac{TP}{TP+FN} \tag{20}$$

因此，F1-score 可以写成：

$$F1=\frac{2*Precision*Recall}{Precision+Recall}$$

$$=\frac{2TP}{2TP+FP+FN} \tag{21}$$

对于真实的信用评估数据集，由于存在严重的数据不均衡问题，使用 AUC 作为评价指标能够更好地反映模型性能。其具体表达式可以表示为：

$$AUC=\frac{\sum_{positives} r-\frac{n_{pos}(n_{pos}+1)}{2}}{n_{pos}n_{neg}} \tag{22}$$

其中，r 为数据集中预测为正样本的概率，n_{pos} 为数据集中正样本的数量，n_{neg} 为数据集中负样本的数量。

4.3.2 可解释性

正如之前提到的，动态模型树通过定义一致性的分裂策略和最小性的剪枝策略，保证了最终的模型具有更浅的树形结构。因此，与原论文中的实验一样，将分裂数和模型参数量作为可解释性指标。分裂数是增量决策树可解释性的更可靠指示，它将树的每个内部节点算作一次分裂。而采用多数类预测的叶节点不进行统计。相反，采用分类器的叶节点贡献一次分裂。与只计算节点总数相比，分裂数说明了增量树的不同叶子类型。模型参数量是描述模型整体复杂性的一种度量，计算时将多数类预测的叶节点算作 1 个附加参数，分类器模型的叶节点算作 d 个参数，其中 d 表示用于计算的特征数量，最后加和得到模型整体的参数量。然而，模型整体参数量并不总是能准确地反映模型的可解释性，因为相比之下，分裂数更能体现模型的全局可解释性，而在局部所关注的往往是单个模型的表达能力。因此，模型参数量是一种比较保守的可解释性评估方法。

4.4 相关算法

为了检验模型的效果，将其与一些相关的算法进行了比较。首先，选择了 4 种经典的增量树模型作为对比。

1）FIMT-DD[28]：典型的模型树框架之一。它使用 Hoeffding 边界作为分裂准则，通过 Page Hinkley 测试来检测和适应概念漂移。FIMT-DD 在每个叶节点建立一个线性模型，用于预测或分类。

2）HT-MC：Hoeffding 树的基础版本，使用多数类别作为叶节点的预测策略，即预测出现次数最多的类别标签。

3）HT-NBA：Hoeffding 树的一个变体，使用朴素贝叶斯作为叶节点的预测策略，即基于贝叶斯定理和属性条件独立性假设来计算后验概率。

4）EFDT[29]：使用任意时间分裂策略进行树的分裂决策，它在每个节点上尽快地选择并部署一个分裂，然后不断地重新评估该分裂，如果发现有更好的分裂，则替换掉原来的分裂方式。

另外，为了进一步比较模型的性能，还与两个集成模型和一个双生网络模型进行了对比。

1）ARF[30]：随机森林在数据流学习中的一个变体，通过增量的方式构建多棵决策树。它使用概念漂移检测器来监测和适应数据流中的概念变化，如果检测到漂移，则重置或替换相应的决策树。

2）ILFDL[12]：一种针对延迟标签数据的集成增量学习的算法，通过概念漂移检测维护最新的数据特征并构建分类器，然后将分类器保存到时间窗口中构成集成分类器，最后通过加权投票的方法输出结果。

3）NMFDL[10]：一种针对延迟标签数据的双生网络算法，通过知识蒸馏的方法增量地更新一个有监督模型和一个半监督模型，最后通过加权平均的方法输出结果。

在这些对比算法中，ILFDL 和 NMFDL 算法是专门的延迟标签处理算法，实验中不做额外的调整。对于 4 种增量树模型和 ARF 算法，表 2 列出了模型相关参数的取值，未列出的采用默认值。同时，实验采用了一种最常见的延迟标签处理方法，为它们维护一个时间窗口用于保存延迟标签数据，待标签到达后再进行反馈训练。

表 2 模型参数设置

Table 2 Model parameter settings

参数	含义	设置
T	时间窗口大小	Δt_{max}
split_confidence	分裂决策阈值	0.01
binary_split	是否为二叉树	True
learning_rate	FIMT-DD 中模型学习率	0.01
leaf_prediction	HT-NBA 叶子预测方式	朴素贝叶斯
min_reevaluate	EFDT 再评估数目下限	100
n_estimators	ARF 中基分类器个数	3

4.5 实验结果分析

4.5.1 预测性能评估

首先针对模型的预测性能进行了实验。表 3 列出了 SEA，Agrawal 和 Hyperplane 3 个数据集在整个数据流生存空间内的平均 F1-score。表 4 列出了模型在 LC，Bank Loan 和 ppdai 3 个不均衡真实数据集上的平均 AUC 指标。另外，图 3 给出了模型在整个数据流过程中每个时间步的表现，能够更清晰直观地分析数据分布变化带来的影响。

表 3 平均 F1-score

Table 3 Average value of F1-score

算法	数据集		
	SEA	Agrawal	HyperPlane
Our Method	0.88±0.03	**0.80±0.11**	**0.81±0.04**
FIMT-DD	0.78±0.10	0.55±0.08	0.74±0.05
HT-MC	0.85±0.03	0.73±0.13	0.64±0.02
HT-NBA	0.85±0.04	0.76±0.12	0.72±0.02
EFDT	0.85±0.04	**0.80±0.10**	0.68±0.03
ARF	**0.89±0.03**	0.77±0.08	0.64±0.02
ILFDL	0.84±0.04	0.78±0.13	0.70±0.05
NMFDL	0.87±0.05	0.79±0.12	0.78±0.06

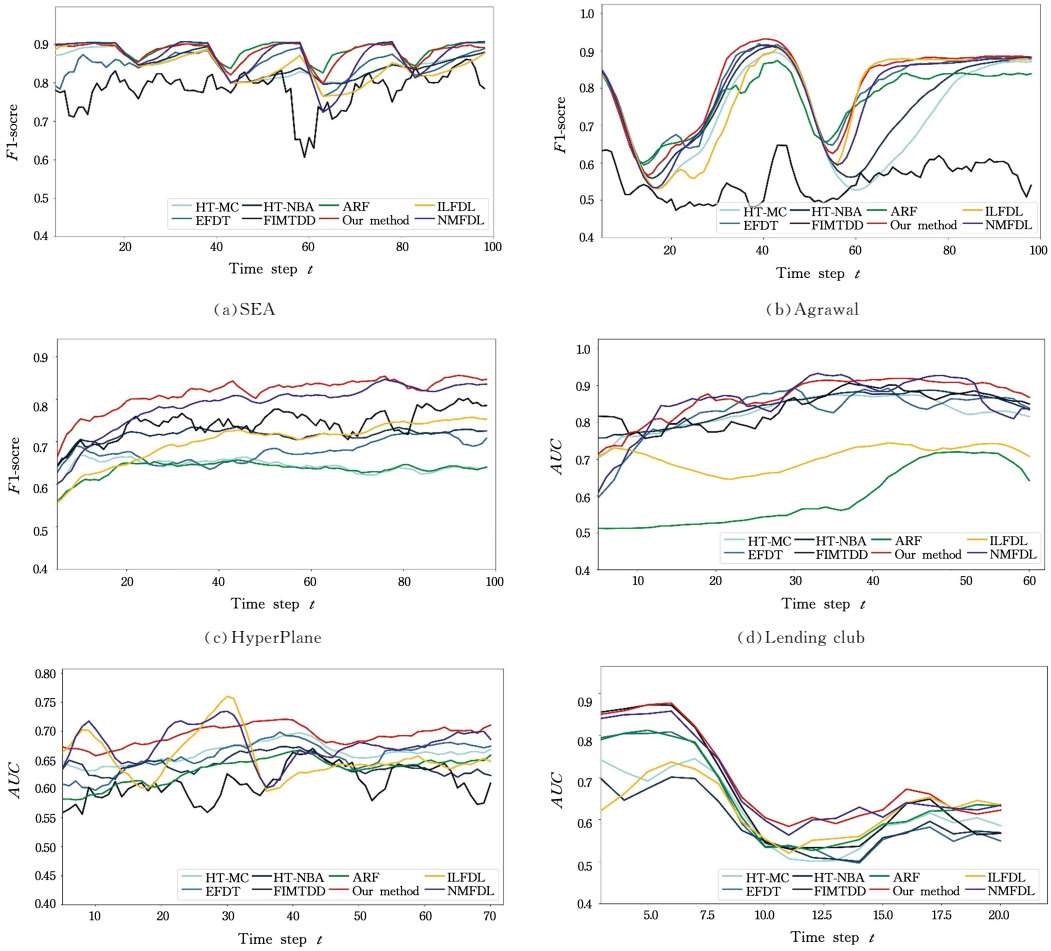
表 4 平均 AUC

Table 4 Average value of AUC

算法	数据集		
	LC	Bank Loan	ppdai
Our Method	**0.86±0.03**	**0.70±0.02**	**0.70±0.12**
FIMT-DD	0.84±0.06	0.61±0.06	0.67±0.14
HT-MC	0.82±0.05	0.66±0.02	0.63±0.10
HT-NBA	0.84±0.04	0.64±0.02	0.60±0.08
EFDT	0.82±0.09	0.66±0.03	0.64±0.13
ARF	0.59±0.08	0.63±0.03	0.65±0.11
ILFDL	0.71±0.03	0.65±0.04	0.62±0.08
NMFDL	0.85±0.09	0.69±0.04	0.69±0.11

从表 3 和表 4 中可以看出,本文提出的模型在所有的数据集上均有良好的平均表现,尤其是在特征数量较多的数据集上,性能明显高于其他基于增量树的模型,即使是与双生网络的方法进行对比,所提模型也取得了 1% 的优势。

从图 3 中可以看出,本文模型在整个数据流过程中保持了较高的稳定性和准确性,能够及时地适应数据分布的变化和标签的变化。相比之下,其他模型方法表现出较差的预测性能和非概念漂移带来的突然波动,表明它们在处理延迟标签和数据动态变化问题上的不足。



(a)SEA　(b)Agrawal

(c)HyperPlane　(d)Lending club

(e)Bank loan　(f)ppdai

图 3　不同时间步的模型表现

Fig. 3 Model performance at different time steps

4.5.2　可解释性评估

由于集成学习和双生网络是黑盒模型,因此仅在单棵增量树模型的算法上进行了可解释方面的对比实验。表 5 列出了每个模型在数据集中的平均分裂数,表 6 则列出了模型在整个数据流生存空间内的平均参数量。

从表 5 可以看出,模型树可以保持比 Hoeffding 树更浅的树形结构,这可以归因于简单模型的灵活性。与 FIMT-DD 模型树相比,本文提出的模型会进行更多次的分裂,以此来换取模型性能的提升。通常情况下,额外分裂的次数仅在 3 次以内,这对可解释性的影响微乎其微。但在特殊情况下,FIMT-DD 算法会盲目追求更浅的模型结构而忽略性能的表现。例如,在 Agrawal 数据集中,FIMT-DD 算法的预测性能极差,而本文模型在尝试更多分裂的情况下,达到了几乎最优的性能,相比同样性能的 EFDT 算法而言,分裂数要减少一半。因此,可以认为改进的动态模型树在保证性能的同时,也能够达到较高的可解释性,而不是牺牲其中一方来满足另一方。

表 5　平均分裂数

Table 5　Average number of splits

算法	数据集					
	SEA	Agrawal	HyperPlane	LC	Bank Loan	ppdai
Our Method	3.4±1.6	40.0 ±26.0	2.1±1.3	**2.0±1.0**	1.4±1.2	3.3±1.9
FIMT-DD	**1.0±0.0**	**1.8±1.2**	**1.1±0.4**	2.1±1.0	**1.0±0.0**	**1.0±0.0**
HT-MC	543.1±331.8	577.2±365.5	254.7±158.1	201.5±156.9	200.2±165.3	49.1±38.1
HT-NBA	1087.1±663.7	1155.4±731.0	510.3±316.1	404.0±313.8	401.5±330.6	99.2±76.2
EFDT	101.5±68.4	95.0±72.8	33.7±18.2	36.7±18.6	11.3±5.9	4.6±2.5

表 6 模型参数量

Table 6 Model parameters

算法	数据集					
	SEA	Agrawal	HyperPlane	LC	Bank Loan	ppdai
Our Method	7.8±3.3	204.2±130.1	78.3±32.6	103.7±34.5	43.4±23.1	39.9±18.3
FIMT-DD	**3.0±0.0**	**12.94±6.17**	**52.1±10.0**	104.8±34.4	36.0±0.0	18.0±0.0
HT-MC	1087.1±663.7	1155.4±731.0	510.3±316.1	404.0±313.8	401.5±330.6	99.2±76.2
HT-NBA	2175.3±1327.3	5780.8±3655.1	13036.0±8061.2	13544.1±10901.8	7445.0±6115.9	950.9±723.5
EFDT	204.0±136.9	190.9±145.6	68.4±36.3	**74.4±37.2**	**23.5±11.8**	**10.1±5.0**

另外，从表 6 可以看出 EFDT 和 FIMT-DD 使用了更少的参数，本文模型排名第三。正如之前所述，评估模型参数量是一个较为保守的统计，因为随着时间的推移，模型树将保持更少的分裂数，这从表 5 中可以观察得到；而其他增量树将生成越来越大的树，分裂数和模型参数量将会有明显增加，模型树的优势将显现出来。一般来说，本文模型所采用的基于损失的测量方法，相比基于纯度的启发式测量方法而言，更易于阐释模型的可解释性，因为它直接考虑了模型的预测效果和输入输出之间的关系。

4.5.3 消融实验

为了进一步验证模型的有效性，本文选取了部分数据集进行消融实验，分别给出模型在 F1-score 和 AUC 上的性能表现。

表 7 列出了消融实验的具体情况。

表 7 消融实验

Table 7 Ablation experiment

Ablation		F1-score		AUC	
延迟标签更新算法	自适应阈值伪标签选择策略	Agrawal	HyperPlane	Bank Loan	ppdai
√	√	**0.80±0.11**	**0.81±0.04**	**0.70±0.02**	**0.70±0.12**
×	×	0.66±0.11	0.79±0.05	0.63±0.07	0.55±0.12
×	√	0.69±0.07	0.75±0.04	0.67±0.01	0.59±0.12
√	×	0.78±0.10	**0.81±0.05**	0.68±0.03	0.68±0.12
$\varphi=0$	√	0.78±0.11	0.80±0.05	0.69± 0.02	0.69±0.11
$\kappa=0$	√	0.79±0.11	0.80±0.04	0.68±0.03	0.69±0.11

实验消除了模型两个关键组成部分的贡献：延迟标签更新算法和自适应阈值的伪标签选择策略。从表 7 可以观察得知，延迟标签更新算法和自适应阈值的伪标签选择策略共同作用，可以达到最优的预测性能。而原始的动态模型树不做任何数据处理，最终得到了一个较差的效果。其次，本文比较了延迟标签更新算法和自适应阈值的伪标签选择策略的单独作用，实验结果表明两种方法均能在一定程度上提升模型效果。最后，将模型与两个退化版本进行了对比：1)反馈数据不进行权重衰减；2)不修正伪标签数据的错误。通过实验，这两种方法的性能有所下滑，进一步证明了算法的有效性。

通过消融实验可以发现，延迟标签更新算法有效地提升了模型的性能表现，它允许反馈数据以一定的权重对模型进行补充，同时还尽可能地减少了伪标签错误和数据概念漂移带来的影响。但是，使用伪标签训练模型是把双刃剑，它允许更多的数据丰富模型，同时也会引入噪音数据导致模型出现偏差，因此单纯使用伪标签的方法在模型提升方面收效甚微，甚至出现反作用。而将两种算法结合，即使用伪标签训练模型，然后经过延迟标签更新算法再调整，可以有效地解决上述问题，达到了最好的性能表现。

结束语 本文构建了一个面向延迟标签场景下的可解释性信用评估模型，通过改进动态模型树算法，使得它在保留可解释性的基础上，能够处理不同权重的数据，然后再结合延迟标签更新算法和自适应阈值的伪标签选择策略有效地应对了延迟标签数据所面临的滞后性和数据不充分的问题。实验表明，在合成数据集和真实的信用评估数据集上，本文算法拥有不错的预测性能和可解释性，能够较好地处理延迟标签问题。但是，正如之前所述，伪标签训练是把双刃剑，会引入噪音数据而影响模型性能。延迟标签更新算法尽管可以在一定程度上覆盖错误标签的影响，但无法完全消除。未来的研究将尝试引入遗忘学习的方法，删除或替换模型中错误的标签数据信息，从而重新引导树的生长方向，这可能是一个有趣的方向。

参 考 文 献

[1] BASTANI K, ASGARI E, NAMAVARI H. Wide and deep learning for peer-to-peer lending[J]. Expert Systems with Applications, 2019, 134: 209-224.

[2] LESSMANN S, BAESENS B, SEOW H V, et al. Benchmarking state-of-the-art classification algorithms for credit scoring: An update of research[J]. European Journal of Operational Research, 2015, 247(1): 124-136.

[3] GOMES H M, GRZENDA M, MELLO R F D, et al. A Survey on Semi-supervised Learning for Delayed Partially Labelled Data Streams[J]. ACM Computing Surveys, 2022, 55(4): 1-42.

[4] TAN F, HOU X, ZHANG J, et al. A Deep Learning Approach to Competing Risks Representation in Peer-to-Peer Lending[J]. IEEE transactions on neural networks and learning systems, 2018, 30(5): 1565-1574.

[5] DU M, LIU N, HU X. Techniques for interpretable machine learning[J]. Communications of the ACM, 2019, 63(1): 68-77.

[6] JIAO L, YANG H, LIU Z G, et al. Interpretable fuzzy clustering using unsupervised fuzzy decision trees[J]. Information Sciences, 2022, 611: 540-563.

[7] LIU H, ZHOU Y, LIU B, et al. Incremental learning with neural

networks for computer vision:a survey[J]. Artificial Intelligence Review,2023,56(5):4557-4589.

[8] YU Z,WANG D,ZHAO Z,et al. Hybrid Incremental Ensemble Learning for Noisy Real-World Data Classification[J]. IEEE transactions on cybernetics,2017,49(2):403-416.

[9] DYER K B,CAPO R,POLIKAR R. COMPOSE:A Semisupervised Learning Framework for Initially Labeled Nonstationary Streaming Data[J]. IEEE transactions on neural networks and learning systems,2013,25(1):12-26.

[10] GAO H,DING Z. A Novel Machine Learning Method for Delayed Labels [C]// 2022 IEEE International Conference on Networking,Sensing and Control(ICNSC). IEEE,2022:1-6.

[11] KUNCHEVA L I,SáNCHEZ J S. Nearest Neighbour Classifiers for Streaming Data with Delayed Labelling[C]// 2008 Eighth IEEE International Conference on Data Mining. IEEE,2008:869-874.

[12] GAO H,DING Z,PAN M. Incremental Learning Method for Data with Delayed Labels[J]. Computing and Informatics,2022,41(5):1260-1283.

[13] POZZOLO A D,BORACCHI G,CAELEN O,et al. Credit Card Fraud Detection:A Realistic Modeling and a Novel Learning Strategy[J]. IEEE transactions on neural networks and learning systems,2017,29(8):3784-3797.

[14] DAS M,PRATAMA M,ZHANG J,et al. A Skip-Connected Evolving Recurrent Neural Network for Data Stream Classification under Label Latency Scenario[C]// Proceedings of the AAAI Conference on Artificial Intelligence. AAAI,2020:3717-3724.

[15] GUNNARSSON B R,BROUCKE S V,BAESENS B,et al. Deep learning for credit scoring:Do or don't? [J]. European Journal of Operational Research,2021,295(1):292-305.

[16] RIBEIRO M T,SINGH S,GUESTRIN C. "Why Should I Trust You?"[C]// Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM,2016:1135-1144.

[17] LUNDBERG S M,ERION G G,CHEN H,et al. From local explanations to global understanding with explainable AI for trees [J]. Nature Machine Intelligence,2020,2(1):56-67.

[18] DONG L A,YE X,YANG G. Two-stage rule extraction method based on tree ensemble model for interpretable loan evaluation [J]. Information Sciences,2021,573:46-64.

[19] ALANGARI N,MENAI M E,MATHKOUR H,et al. Intrinsically Interpretable Gaussian Mixture Model[J]. Information,2023,14(3):164.

[20] DOMINGOS P,HULTEN G. Mining high-speed data streams [C]// Proceedings of the sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM,2000:71-80.

[21] POTTS D,SAMMUT C. Incremental Learning of Linear Model Trees[J]. Machine Learning,2005,61(1/2/3):5-48.

[22] HAUG J,BROELEMANN K,KASNECI G. Dynamic Model Tree for Interpretable Data Stream Learning[C]// 2022 IEEE 38th International Conference on Data Engineering (ICDE). IEEE,2022:2562-2574.

[23] BROELEMANN K,KASNECI G. A Gradient-Based Split Criterion for Highly Accurate and Transparent Model Trees[C]// Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence. IJCAI,2019:2030-7.

[24] GRZENDA M,GOMES H M,BIFET A. Delayed labelling evaluation for data streams[J]. Data Mining and Knowledge Discovery,2020,34(5):1237-1266.

[25] STREET W N,KIM Y. A streaming ensemble algorithm(SEA) for large-scale classification[C]// Proceedings of the seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM,2001:377-382.

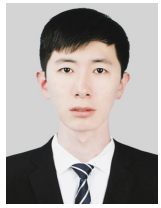
[26] HULTEN G,SPENCER L,DOMINGOS P. Mining time-changing data streams[C]// Proceedings of the seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM,2001:97-106.

[27] AGRAWAL R,IMIELINSKI T,SWAMI A N. Database Mining:A Performance Perspective [J]. IEEE Transactions on Knowledge and Data Engineering,1993,5(6):914-925.

[28] IKONOMOVSKA E,GAMA J,DZEROSKI S. Learning model trees from evolving data streams[J]. Data Mining and Knowledge Discovery,2011,23:128-168.

[29] MANAPRAGADA C,WEBB G I,SALEHI M. Extremely Fast Decision Tree [C]// Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. ACM,2018:1953-1962.

[30] GOMES H M,BIFET A,READ J,et al. Adaptive random forests for evolving data stream classification[J]. Machine Learning,2017,106:1469-1495.



XIN Bo, born in 2000, postgraduate. His main research interests include credit evaluation and machine learning.

DING Zhijun, born in 1974, Ph.D, Professor, Ph.D supervisor, is a senior member of CCF(No. 14797S). His main research interests include intelligent software engineering, cloud computing and services, big data credit reporting and financial risk control.

(责任编辑:柯颖)