计算机研究与发展
Journal of Computer Research and Development
https://crad.ict.ac.cn
62(5): 1235−1247, 2025

弹性梯度集成的概念漂移适应

郭虎升[1,2] 张羽桐[1] 王文剑[1,2]

[1](山西大学计算机与信息技术学院 太原 030006)

[2](计算智能与中文信息处理教育部重点实验室(山西大学) 太原 030006)

(guohusheng@sxu.edu.cn)

Elastic Gradient Ensemble for Concept Drift Adaptation

Guo Husheng[1,2], Zhang Yutong[1], and Wang Wenjian[1,2]

[1](*School of Computer and Information Technology*, *Shanxi University*, *Taiyuan* 030006)

[2](*Key Laboratory of Computational Intelligence and Chinese Information Processing* (*Shanxi University*), *Ministry of Education*, *Taiyuan* 030006)

Abstract With the surge of streaming data, concept drift has become an important and challenging problem in streaming data mining. At present, most ensemble learning methods do not specifically identify the types of concept drift and do not adopt efficient ensemble adaptation strategies, resulting in uneven performance of models on different concept drift types. To address this, we propose an elastic gradient ensemble for concept drift adaptation (EGE_CD). Firstly, the gradient boosting residual is extracted and the flow residual ratio is calculated to detect the drift site, and then the residual volatility is calculated to identify the type of drift. Then, the drift learners are extracted by using the change of learner loss, and the corresponding learners are deleted by combining different drift types and residual distribution characteristics to realize elastic gradient pruning. Finally, the incremental learning method is combined with the sliding sampling method to optimize the fitting process of the learner by calculating the optimal fitting rate, and then the incremental gradient growth is realized according to the change of the residual of the learner. The experimental results show that the proposed method improves the stability and adaptability of the model to different concept drift types and achieves good generalization performance.

Key words concept drift; drift type; gradient boosting; drift detection; elastic gradient pruning; incremental gradient growth

摘 要 随着流数据的大量涌现,概念漂移已成为流数据挖掘中备受关注且具有挑战性的重要问题.目前,多数集成学习方法未针对性地识别概念漂移类型,并采取高效的集成适应策略,导致模型在不同漂移类型上的性能参差不齐.为此,提出了一种弹性梯度集成的概念漂移适应(elastic gradient ensemble for concept drift adaptation, EGE_CD)方法.该方法首先通过提取梯度提升残差,计算流动残差比检测漂移位点,之后计算残差波动率识别漂移类型;然后,利用学习器损失变化提取漂移学习器,结合不同漂移类型与残差分布特征删除对应学习器,实现弹性梯度剪枝;最后,将增量学习与滑动采样方法结合,通过计算最优拟合率优化学习器拟合过程,再根据残差变化实现增量梯度生长.实验结果表明,所提方法提高了模型对不同漂移类型的稳定性与适应性,取得了良好的泛化性能.

收稿日期:2024-05-31;**修回日期**:2024-09-10

基金项目:国家自然科学基金项目(62276157, U21A20513, 62476157, 62076154, 61503229);山西省重点研发计划项目(202202020101003)

This work was supported by the National Natural Science Foundation of China (62276157, U21A20513, 62476157, 62076154, 61503229) and the Key Research & Development Program of Shanxi Province (202202020101003).

通信作者:王文剑(wjwang@sxu.edu.cn)

关键词 概念漂移；漂移类型；梯度提升；漂移检测；弹性梯度剪枝；增量梯度生长

中图法分类号 TP18

DOI: 10.7544/issn1000-1239.202440407 **CSTR:** 32373.14.issn1000-1239.202440407

大数据时代，各种应用领域都不断涌现出动态数据，涵盖网购、气象、舆情、智能城市以及病毒传播等方面. 相对于静态数据，这些数据以高速、实时、多变、难以预测为特点，通常被称为流数据[1-3]. 流数据挖掘的目标是使学习模型更准确地预测数据分布变化，从而提高在线学习模型的性能.

概念漂移是流数据在现实世界中的一个重要特征，也是流数据分析挖掘不可避免的挑战之一，其典型特征是当环境发生变化时，数据分布也会相应改变[4-5]. 然而，由于流数据分布变化方式的多样性，不同类型的概念漂移[6-7](如突变型和渐变型)会以多种方式发生，例如变化时间、速度和方式等方面都可能存在差异. 不同类型概念漂移的发生会直接影响模型性能，并导致性能不同程度的下降[8-9].

现有的概念漂移处理方法大多基于集成学习，通过特定规则(如加权平均、组合投票等)组合训练好的分类器，以达到比单一学习器更出色的性能[10-11]. 然而，当前相关研究大多集中在单一类别概念漂移上，对于不同类型概念漂移的适应性建模相对欠缺[12]. 在概念漂移发生后，如何有效应用模型和合理调整学习器，以提高模型的稳定性和适应性，依然是一个亟待解决的问题.

为解决上述问题，本文提出了一种弹性梯度集成的概念漂移适应方法. 该方法采取先检测后适应的策略，首先计算流动残差比检测漂移位点，之后利用残差波动率识别漂移类型；随后，根据梯度损失特征提取漂移学习器，结合漂移类型与残差分布特征，采取弹性梯度剪枝策略删除漂移学习器以提高模型稳定性；最后，通过计算最优拟合率优化学习器拟合过程，并结合滑动采样方法，采取增量梯度生长策略增量更新学习器以提高模型适应性. 本文工作的主要贡献有3点：

1)通过探测流动残差比与残差波动率，判断残差稳定程度，为集成模型识别不同类型概念漂移提供了一种可行方案；

2)通过分析梯度损失特征提取漂移学习器，并对其进行弹性梯度剪枝以提高模型的稳定性；

3)通过滑动采样数据优化学习器拟合过程，增量更新学习器以提高模型的适应性.

1 相关工作

目前，已有较多文献对含概念漂移的流数据挖掘问题进行了研究，根据是否对概念漂移的发生进行检测，大致可分为基于主动检测的概念漂移处理方法与基于被动自适应的概念漂移处理方法.

基于主动检测的概念漂移处理方法将概念漂移检测融入模型中，通过对学习模型的性能指标进行监测来判断是否发生概念漂移，当检测到概念漂移发生时对模型做出相应调整. 常见的主动检测方法通常使用滑动窗口技术来实现，其通过将数据流划分为固定大小的窗口，利用窗口的滑动来判断是否有概念漂移发生. ADWIN[13]通过滑动窗口保存最新数据，利用2个子窗口内样本的均值差异来检测概念漂移. DDM(drift detection method)[14]通过监测窗口中分类误差的置信度和误差标准差来检测概念漂移. CD-TW[15]通过对比历史窗口与实时窗口的数据分布变化来检测漂移，并依靠时间步长识别概念漂移类型. EOCD(ensemble optimization for concept drift)[16]将遗忘机制与DDM方法结合，当检测到概念漂移时，通过删除较长的旧概念以减少性能损失. DAWIDD(dynamic adapting window independence drift detection)[17]利用滑动窗口进行样本的保存与删除，结合样本独立性测试检测概念漂移. OCDD(one-class drift detector)[18]方法使用带滑动窗口的单类学习器实现了无监督问题的概念漂移检测. SNDC(self-adaption neighborhood density clustering)[19]方法通过比较相邻数据块聚类的相似性检测概念漂移，随后调整聚类中心点适应概念漂移. 基于主动检测的概念漂移处理方法有助于学习模型及时发现漂移并触发模型更新，但当漂移检测出现误检或漏检时，模型容易错误更新，导致模型性能下降.

基于被动自适应的概念漂移处理方法不检测数据流中是否存在概念漂移，其通过不断调整学习器来适应数据分布的变化. 在被动自适应方法中，基于集成学习的处理方式较为常见，其根据学习单元大小分为基于数据块的集成和在线集成. 基于数据块的集成是一种对固定数量的输入实例进行处理的方法. SEA(streaming ensemble algorithm)[20]在连续的数

据块上构建分类器，再使用启发式替换策略对分类器进行集成. ACDWM(adaptive chunk-based dynamic weighted majority)[21]为每个数据块构建单独的分类器，根据其在当前数据块上的性能动态加权. SEOA(selective ensemble-based online adaptive deep neural network)[22]根据每个分类器的损失将神经网络不同层次的自适应密度单元作为集成的基分类器进行动态加权. EnHAT(ensemble combined with Hoeffding adaptive tree)[23]将霍夫丁自适应树算法与基于数据块生成的决策树集成相结合，以在更短时间内适应概念漂移. 在线集成方法是一种对样本进行逐一处理的增量学习方法. OAUE(online accuracy updated ensemble)[24]在增量更新的基础上提出一种具有成本效益的分量权重函数，以估计分类器的误差. AUE2(accuracy updated ensemble)[25]将基于精度的加权机制与霍夫丁树的增量特性结合，以改善集成对不同类型概念漂移的反应. OnlineBagging[26]将在线学习与 Bagging[27]方法结合，并通过权重分配机制调整不同学习器的权重. DSEL(dynamic ensemble selection)[28]基于 Bagging 方法将数据预处理与动态集成选择结合，提高了对模型复杂数据流的有效性. OneNet(online ensembling network)[29]将指数梯度下降方法与离线强化学习结合，动态调整模型权重，提高了概念漂移的适应速度. 基于被动自适应的概念漂移处理方法不会受到漂移检测过程中漏检或误检的影响，但由于需要根据模型性能逐步更新模型，故模型效率较低.

本文提出了一种弹性梯度集成的概念漂移适应方法. 与传统方法相比，该方法通过探测梯度提升残差变化，实现了对不同概念漂移类型的识别，根据不同漂移类型特征，采取弹性梯度剪枝与增量梯度生长，有针对性地更新模型，提高了模型在不同类型概念漂移中的泛化性能.

2 弹性梯度集成的概念漂移适应方法

本文提出了一种弹性梯度集成的概念漂移适应方法，模型总体结构如图 1 所示. 在漂移类型检测阶段，首先利用梯度提升残差变化特征计算流动残差比，检测概念漂移位点，再计算残差波动率判断残差变化幅度识别漂移类型. 在弹性梯度剪枝阶段，首先利用学习器损失变化特征提取漂移学习器，再结合不同漂移类型与残差分布特征删除合适的漂移学习器，提高模型的稳定性. 在增量梯度生长阶段，结合增量学习方法，利用滑动采样数据计算最优拟合率，以优化学习器拟合过程，同时根据残差变化动态增量更新学习器，提高模型的适应性.

2.1 漂移类型检测

数据序列可以用时间步长划分，因此，流数据可以被形式化为

$$S=\{(x_i,y_i)\}_{i=1}^{t}, \tag{1}$$

其中 x_i 是对应时刻的样本实例，y_i 是该样本实例对应的标签. 假设流数据中数据的联合概率分布表示为 $P(\boldsymbol{x},\boldsymbol{y})$，若在时刻 t，流数据的概率分布发生变化，即 $P_{t-1}(\boldsymbol{x},\boldsymbol{y})\neq P_t(\boldsymbol{x},\boldsymbol{y})$，则该时刻发生了概念漂移. 其中，$\boldsymbol{x}=(x_i)_{i=1}^{t}$.

为根据不同漂移类型调整模型，首先需要检测漂移位点并识别漂移类型，本文通过计算流动残差比检测漂移位点，其次通过计算残差波动率识别漂移类型. 梯度提升残差作为梯度提升模型[30]更新的重要指标，伴随模型的更新生成，梯度提升模型 F_m 象

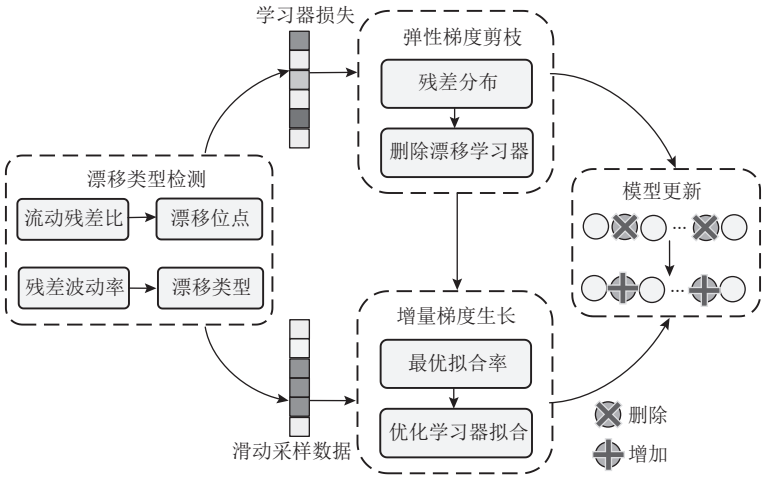


Fig. 1 Overall structure of EGE_CD model

图 1 EGE_CD 模型总体结构

征着m个学习器的集成,第m轮梯度提升模型的更新可以表示为

$$F_m(\boldsymbol{x})=F_{m-1}(\boldsymbol{x})+h_m(\boldsymbol{x}). \tag{2}$$

学习器$h_m(\boldsymbol{x})$正是对$F_{m-1}(\boldsymbol{x})$与真实标签$\boldsymbol{y}$之间差异的拟合结果,其主要依靠拟合第m轮样本x_i的梯度提升残差形成,残差的计算方法为:

$$r_m(x_i)=-\left[\frac{\partial l(y_i,F_{m-1}(x_i))}{\partial F_{m-1}(x_i)}\right]. \tag{3}$$

因本文损失函数采用平方损失计算,则模型损失$l(y_i,F_{m-1}(x_i))=\frac{1}{2}\left(y_i-\sum_{j=1}^{m-1}h_j(x_i)\right)^2$,此时残差可以化简为$r_m(x_i)=y_i-F_{m-1}(x_i)$.所得残差将作为新的样本标签,用于生成新的学习器.因此,在模型的预测过程中,可通过提取所有学习器的残差计算模型残差,其可反映数据分布变化对模型的影响.模型整体残差可表示为

$$R=\frac{1}{M}\sum_{m=1}^{M}r_m, \tag{4}$$

其中M为模型中学习器的总数.给定初始梯度提升模型$F_M(\boldsymbol{x})$,其训练过程中可得初始模型残差为$\widetilde{R}$,在s个连续的数据块$\{D_1,D_2,\cdots,D_s\}$上,可得对应模型残差为$\{R_1,R_2,\cdots,R_s\}$,则流动残差比可表示为

$$p_i=\frac{R_i}{\widetilde{R}}(i=1,2,\cdots,s). \tag{5}$$

若p_i满足$p_i>\tau$(τ为漂移位点探测参数),则将i作为概念漂移起始位点输出,反之继续进行检测,直到找到满足$p_i>\tau$的数据块时停止检测.

假设第b个数据块D_b处发生概念漂移,残差大幅度增加.若漂移类型不同,残差变化幅度也会存在明显差异.因此本文将数据块D_b后n个模型残差的标准差作为残差波动率来识别漂移类型,其计算方法如式(6)所示.

$$\sigma_R=\sqrt{\frac{1}{n}\sum_{i=1}^{n}(R_{b+i}-\overline{R})}, \tag{6}$$

其中$\overline{R}$为D_b后n个模型残差的平均值,σ_R可以反映不同数据块上残差的变化幅度.若$\sigma_R>\delta$(δ为漂移稳定性探测参数),则残差变化幅度过大,识别此次漂移为突变型,否则识别此次漂移为渐变型.漂移类型检测过程如图2所示.

2.2 弹性梯度剪枝

当出现不同类型漂移时,模型泛化性能也会存在差异,为减小不同类型漂移对模型的影响,本节提出了一种弹性梯度剪枝方法.该方法利用梯度提升模型中学习器的损失变化特征提取漂移学习器,结合漂移类型与残差分布特征,采取不同剪枝策略以删除合适的漂移学习器,提高模型稳定性.模型弹性梯度剪枝过程如图3所示.

在梯度提升模型中,伴随学习器的迭代更新,学习器损失应满足$l_{m+1}-l_m\leqslant 0$,但当发生概念漂移时,部分学习器受漂移影响,会使本应下降的损失增加.假设数据块D_b处发生概念漂移,若学习器h_m与h_{m-1}损失满足:

$$\left[E\left(l_m^{b-1}\right)\leqslant E\left(l_{m-1}^{b-1}\right)\right]\wedge\left[E\left(l_m^{b}\right)>E\left(l_{m-1}^{b}\right)\right], \tag{7}$$

则$h_m(\boldsymbol{x})$是漂移学习器.$E\left(l_m^b\right)$为学习器h_m在数据块D_b上损失的期望.当发生不同类型漂移时,漂移学习器的个数不同,因此为提高模型在不同漂移类型下的稳定性,需采取针对性的剪枝策略对模型进行调整.

当数据块D_b处发生突变型概念漂移时,数据分布迅速变化,新的数据分布持续很长时间,模型残差大幅度上升,学习器残差分布满足$\sum_{m=1}^{M}\mathbb{I}(r_m>R_{b-1})>\frac{M}{2}$.此时,漂移学习器数量较多,旧模型已不再适用,需删除所有学习器,并使用漂移数据D_b训练新的梯度提升模型.突变型漂移剪枝策略可表示为

$$F_{\text{pru}}^{\text{SU}}(\boldsymbol{x})=F_{M-k}(\boldsymbol{x})+F'_M(\boldsymbol{x})=\sum_{m=1}^{M}h'_m(\boldsymbol{x}), \tag{8}$$

其中$k=M$,此时在数据块D_{b+1}处的模型损失为

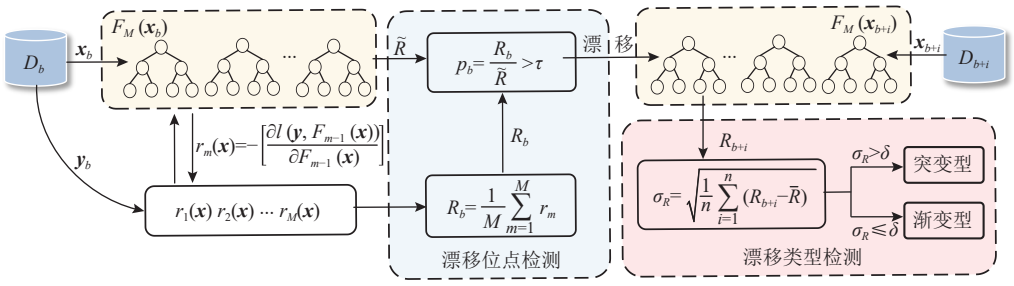


Fig. 2 Drift type detection process

图2 漂移类型检测过程

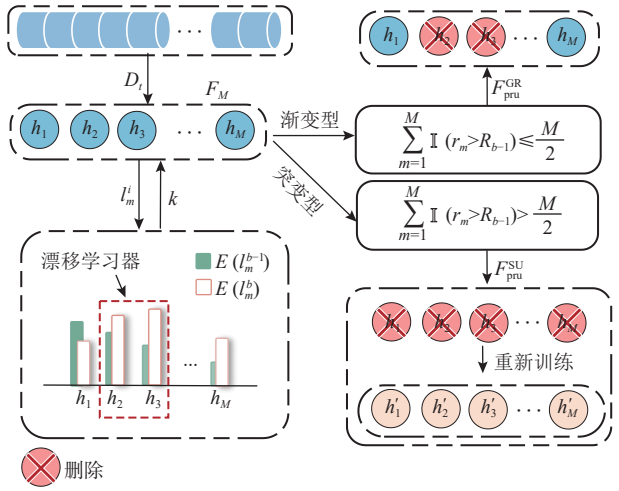


Fig. 3 Elastic gradient pruning process

图 3 弹性梯度剪枝过程

$$l_{b+1}^{\mathrm{SU}}(\boldsymbol{y},F_{\mathrm{pru}}^{\mathrm{SU}}(\boldsymbol{x}))=\frac{1}{2}\left(\boldsymbol{y}-\sum_{m=1}^{M}h'_m(\boldsymbol{x})\right)^2. \tag{9}$$

当数据块D_b处发生渐变型概念漂移时，数据分布缓慢变化，模型残差在一段时间内持续小幅度上升，学习器残差分布满足$\sum_{m=1}^{M}\mathbb{I}(r_m>R_{b-1})\leqslant\frac{M}{2}$. 此时，漂移学习器数量较少，由梯度提升模型学习器迭代更新的特性可知，累积损失最小学习器h_d后的学习器即为漂移学习器，因此为快速恢复模型稳定性，仅需删除h_d后合适的k个学习器. 渐变型漂移剪枝策略可表示为

$$F_{\mathrm{pru}}^{\mathrm{GR}}(\boldsymbol{x})=F_{M-k}(\boldsymbol{x})=\sum_{m=1}^{d}h_m(\boldsymbol{x})+\sum_{m=d+k+1}^{M}h_m(\boldsymbol{x}), \tag{10}$$

其中$k=\arg\min_k l_{b+1}(\boldsymbol{y},F_{M-k}(\boldsymbol{x}))(0\leqslant k<M)$，此时在数据块$D_{b+1}$处的模型损失可表示为

$$l_{b+1}^{\mathrm{GR}}(\boldsymbol{y},F_{\mathrm{pru}}^{\mathrm{GR}}(\boldsymbol{x}))=\frac{1}{2}\left(\boldsymbol{y}-\sum_{m=1}^{d}h_m(\boldsymbol{x})-\sum_{m=d+k+1}^{M}h_m(\boldsymbol{x})\right)^2. \tag{11}$$

2.3 增量梯度生长

在含概念漂移的流数据中，模型剪枝之后，其适应能力会出现一定程度的下降. 为使剪枝后的模型能够快速适应新的分布，本节在模型剪枝的基础上引入学习器的增量更新，利用滑动采样方法更新学习器训练数据，通过计算最优拟合率优化学习器拟合过程，再结合漂移类型变化调整增量学习器个数，实现模型的增量梯度生长，提高模型的适应性. 模型增量梯度生长过程如图 4 所示.

模型首先利用滑动窗口w_s抽取实时漂移数据作为滑动采样数据$D'_{w_s}=\{(x'_i,y'_i)\}_{i=1}^{w_s}$，之后使用剪枝后的模型$F_{\mathrm{pru}}(\boldsymbol{x})$计算最优拟合率$\theta$，并更新增量学习数据

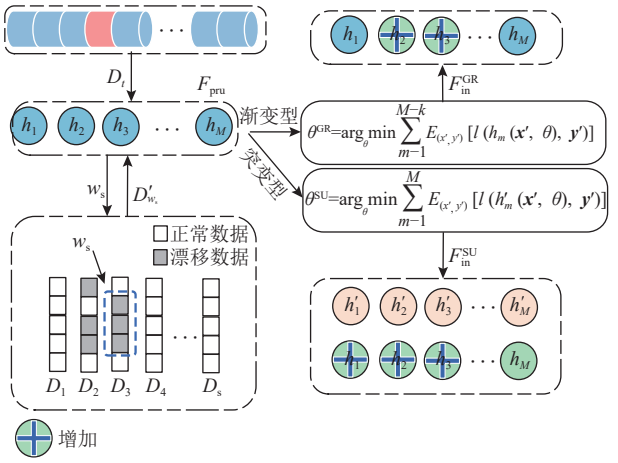


Fig. 4 Incremental gradient growth process

图 4 增量梯度生长过程

为$\{(x'_i,\theta r_c(x'_i))\}_{i=1}^{w_s}$，最优拟合率可表示为

$$\theta=\arg\min_{\theta}\sum_{m=1}^{M'}E_{(\boldsymbol{x}',\boldsymbol{y}')}[l(h_m(\boldsymbol{x}';\theta),\boldsymbol{y}')], \tag{12}$$

其中M'为剪枝后学习器的个数，$h_m(\boldsymbol{x}';\theta)$为剪枝后的学习器. 使用增量学习数据训练学习器，可得增量学习器为

$$\hat{h}_c(\boldsymbol{x}';\theta)=\sum_{j=1}^{J}\hat{a}_{cj}I=\sum_{j=1}^{J}[\arg\min_{a}\sum_{x'_i\in\hat{R}_{cj}}l(\theta r_c(x'_i),F_{c-1}(x'_i)+a)]I, \tag{13}$$

其中$\hat{a}_{cj}$为增量学习器的叶节点，J为叶节点的个数，I为固定的叶节点学习率，$\hat{R}_{cj}$为对应的叶节点区域. 然而，单个学习器难以适应不同漂移类型，因此需针对不同漂移类型制定合适的增量梯度生长策略.

当数据块D_b处发生突变型概念漂移时，由于此时残差变化幅度较大，新数据持续时间较长，剪枝后的模型仍不能快速拟合残差至正常水平，此时需在剪枝模型的基础上训练M个增量学习器以快速拟合残差. 突变型漂移增量梯度生长策略可表示为

$$F_{\mathrm{in}}^{\mathrm{SU}}(\boldsymbol{x})=F_{\mathrm{pru}}^{\mathrm{SU}}(\boldsymbol{x})+\hat{F}_M(\boldsymbol{x})=\sum_{m=1}^{M}h'_m(\boldsymbol{x})+\sum_{c=1}^{M}\hat{h}_c(\boldsymbol{x}), \tag{14}$$

此时在数据块D_{b+1}处的模型损失可表示为

$$l_{b+1}^{\mathrm{SU}}(\boldsymbol{y},F_{\mathrm{in}}^{\mathrm{SU}}(\boldsymbol{x}))=\frac{1}{2}\left(\boldsymbol{y}-\sum_{m=1}^{M}h'_m(\boldsymbol{x})-\sum_{c=1}^{M}\hat{h}_c(\boldsymbol{x})\right)^2. \tag{15}$$

当数据块D_b处发生渐变型概念漂移时，由于此时残差增长幅度小、上升时间长，与突变型漂移对比，残差容易拟合，此时仅需在剪枝模型的基础上将模型增量生长至M个学习器，即训练k个增量学习器以

持续拟合残差. 渐变型漂移增量梯度生长策略可表示为

$$F_{\text{in}}^{\text{GR}}(\boldsymbol{x}) = F_{\text{pru}}^{\text{GR}}(\boldsymbol{x}) + \hat{F}_k(\boldsymbol{x}) = \sum_{m=1}^{d} h_m(\boldsymbol{x}) + \sum_{m=d+k+1}^{M} h_m(\boldsymbol{x}) + \sum_{c=1}^{k} \hat{h}_c(\boldsymbol{x}). \quad (16)$$

此时在数据块D_{b+1}处的模型损失可表示为

$$l_{b+1}^{\text{GR}}(\boldsymbol{y}, F_{\text{in}}^{\text{GR}}(\boldsymbol{x})) = \frac{1}{2}\left(\boldsymbol{y} - \sum_{m=1}^{d} h_m(\boldsymbol{x}) + \sum_{m=d+k+1}^{M} h_m(\boldsymbol{x}) + \sum_{c=1}^{k} \hat{h}_c(\boldsymbol{x})\right)^2. \quad (17)$$

2.4 算法实施流程

本文提出一种弹性梯度集成的概念漂移适应方法, 该方法首先计算流动残差比检测漂移位点, 再计算残差波动率识别漂移类型. 之后在弹性梯度剪枝阶段, 利用学习器损失提取漂移学习器, 针对不同漂移类型, 删除对应的漂移学习器, 以提高模型稳定性; 在增量梯度生长阶段, 采用滑动窗口抽取漂移数据, 计算最优拟合率, 优化学习器拟合过程, 再针对不同漂移类型, 调整增量学习器个数, 以提高模型适应性.

算法 1. 弹性梯度集成的概念漂移适应算法.

初始化: 数据流$SD = \{D_1, D_2, \cdots, D_t, \cdots\}$, 初始梯度提升模型$F_M = \{h_1, h_2, \cdots, h_M\}$, 初始模型残差$\widetilde{R}$, 漂移稳定性探测参数$\delta$, 漂移位点探测参数$\tau$.

① while 流数据SD未结束
② 第t个位点对应数据块D_t进入;
③ 采用F_M中的每个基学习器h_i对当前样本预测, 根据式(4)计算模型残差R_t;
④ 根据式(5)计算流动残差比p_i;
⑤ if $p_i > \tau$ /*发生概念漂移*/
⑥ 根据式(6)计算模型残差波动率σ_R;
⑦ if $\sigma_R > \delta$ /*发生突变型概念漂移*/
⑧ 模型剪枝更新:
$F_{\text{pru}}^{\text{SU}}(\boldsymbol{x}) \leftarrow F_{M-k}(\boldsymbol{x}) + F'_M(\boldsymbol{x})$;
⑨ 根据式(12)计算最优拟合率θ;
⑩ 模型增量生长:
$F_{\text{in}}^{\text{SU}}(\boldsymbol{x}) \leftarrow F_{\text{pru}}^{\text{SU}}(\boldsymbol{x}) + \hat{F}_M(\boldsymbol{x})$;
⑪ else /*发生渐变型概念漂移*/
⑫ 查找累积损失最小学习器h_d;
⑬ 计算学习器剪枝个数:
$k = \arg\min_k l_{b+1}(\boldsymbol{y}, F_{M-k}(\boldsymbol{x}))$;
⑭ 模型剪枝更新:
$F_{\text{pru}}^{\text{GR}}(\boldsymbol{x}) \leftarrow F_{M-k}(\boldsymbol{x})$;
⑮ 根据式(12)计算最优拟合率θ;
⑯ 模型增量生长:
$F_{\text{in}}^{\text{GR}}(\boldsymbol{x}) \leftarrow F_{\text{pru}}^{\text{GR}}(\boldsymbol{x}) + \hat{F}_k(\boldsymbol{x})$;
⑰ end if
⑱ end if
⑲ 在D_{t+1}上进行测试, 得到实时精度与模型损失;
⑳ end while

2.5 模型复杂度分析

假设训练一个学习器的时间复杂度为$O(h) = O(en\log(n))$, 其中e是特征数量, n是训练集的大小. 由于梯度提升模型利用之前所有学习器的残差进行后续训练, 因此M个学习器的梯度提升模型的时间复杂度为$O(Mh)$. 因梯度提升残差在模型训练过程中获取, 故流动残差比与残差波动率伴随模型更新计算可得, 其时间复杂度均为$O(1)$, 因此漂移类型检测过程的时间复杂度为$O(1)$. 当发生突变型概念漂移时, 弹性梯度剪枝过程中删除M个学习器以及训练新的梯度提升模型的时间复杂度为$O(M + Mh)$, 增量梯度生长过程利用滑动采样数据块训练M个增量学习器的时间复杂度为$O(Mh')$, 其中$O(h') = O(en'\log(n'))$, n'为滑动采样数据块的大小. 当发生渐变型概念漂移时, 弹性梯度剪枝过程中删除k个学习器的时间复杂度为$O(k)$, 增量梯度生长过程训练k个增量学习器的时间复杂度为$O(kh')$. 由此可得, 模型的整体运行时间复杂度可表示为

$$O(Mh + \beta(M + Mh + Mh') + (1-\beta)(k + kh')) = O(M(h(1+\beta) + \beta h' + k(1-\beta)(1+h')), \quad (18)$$

其中$\beta \in \{0, 1\}$, 当发生突变型概念漂移时, $\beta = 1$, 发生渐变型概念漂移时, $\beta = 0$. 经分析可得, 与传统在线学习模型发生漂移便重新训练整体模型的概念漂移适应方法相比, 本文方法在时间复杂度上具有明显优势. 此外, 通过滑动采样技术与增量学习策略, 本文方法能够在面对大规模数据流时, 保证较高计算效率的同时有效利用数据信息, 使模型保持良好的泛化性能.

3 实验分析

为验证本文提出的 EGE_CD 方法的有效性, 本文在具有不同类型概念漂移的标准数据集和真实数据集上进行实验, 并从精度、鲁棒性以及收敛性这 3 个方面进行评价. 实验平台为 Windows10 操作系统, CPU 为酷睿 i7-3, 2 GHz 内核, 内存为 8 GB, 本方法采用 Python 3.7 的环境编写和运行.

3.1 实验数据

为了检验算法对不同类型概念漂移的处理能力，本文使用大规模在线分析(massive online analysis, MOA)平台[31]中的流数据生成器产生了6个具有突变式、渐进式以及增量式的概念漂移数据集，除此之外，本文还选取了3个真实数据集，具体的数据集信息如表1所示.

3.2 评价指标

为验证所提出的方法EGE_CD的性能，本节从模型的精度、收敛性及鲁棒性等方面进行了分析，具体指标有4个.

1)平均实时精度(average real accuracy, *Avgracc*). 表示模型在每个时间步的实时精度的平均值，该指标用于反映模型的整体分类性能. 其定义为：

$$Avgracc = \frac{1}{T}\sum_{t=1}^{T}\frac{n_t}{D_t}, \tag{19}$$

其中T表示总的时间步数，n_t表示时间步t内正确分类的样本数，D_t表示样本块大小.

2)累积精度(cumulative accuracy, *Cumacc*). 表示模型到当前时刻的累积预测正确样本数和总样本数的比值，该指标用于反映模型过去所有时刻的整体性能. 其定义为：

$$Cumacc = \frac{1}{T_t \times D_t}\sum_{i=1}^{T_t} n_i, \tag{20}$$

其中T_t表示当前累积的时间步数.

3)收敛速度(recovery speed under accuracy, *RSA*). 表示模型在概念漂移发生后实时精度恢复到稳定所需要的时间步数*step*与恢复过程中模型平均错误率*avge*的乘积. 其定义为：

$$RSA = step \times avge. \tag{21}$$

由于模型在不同概念漂移类型的数据集上性能波动变化不同，本文采用漂移位点后的K个参照点的精度差异来判断该点是否为收敛点. 若K个参考点的精度差异小于阈值ε，且K个参照点的前半部分和后半部分差异也小于ε，则该位点为收敛位点. 其定义为：

$$\left|acc_t - \frac{1}{K}\sum_{j=1}^{K} acc_{t+j}\right| < \varepsilon \text{ and}$$

$$\frac{2}{K}\left|\sum_{j=1}^{\frac{K}{2}} acc_{t+j} - \sum_{j=\frac{K}{2}+1}^{K} acc_{t+j}\right| < \varepsilon, \tag{22}$$

其中$acc_t = \frac{n_t}{D_t}$，漂移恢复率越小表明模型能越快恢复到漂移前的性能.

4)鲁棒性(robustness, *ROB*)[32]. 表示模型的稳定性与泛化性能的好坏. 本文在平均实时精度上分析了不同算法的鲁棒性，其定义为：

$$ROB_A(D) = \frac{Avgracc_A(D)}{\min Avgracc(D)}, \tag{23}$$

其中$Avgracc_A(D)$表示算法A在数据集D上的平均实时精度，$\min Avgracc(D)$表示在数据集D上所有算法中的最小平均实时精度. 鲁棒性值越大，说明模型越稳定.

3.3 参数设置

1)模型子树边界M. 实验中采用CART回归树作为基学习器来构建梯度提升模型. 由于本文在对模型更新过程中采用增量生长来提高模型适应性，M限制了模型增量生长的范围，因此，本文选取$M \in \{500, 1000, 1500, 2000, 2500\}$进行讨论，得到了在不同$M$下的模型性能差异.

2)数据块大小w. 过小的数据块上无法得到足够多数据的样本特征，训练的基学习器稳定性较差，而过大的数据块可能会包含概念漂移，影响模型的分

Table 1 Datasets Information

表1 数据集信息

数据集	实例数	维度	类别数量	漂移类型	漂移数量	漂移位点
RBFblips_su	100×10^3	10	2	突变型	3	25×10^3, 50×10^3, 75×10^3
RBFblips_gr	100×10^3	10	2	渐变型	3	25×10^3, 50×10^3, 75×10^3
Hyperplane_su	100×10^3	10	2	突变型	3	25×10^3, 50×10^3, 75×10^3
Hyperplane_gr	100×10^3	10	2	渐变型	3	25×10^3, 50×10^3, 75×10^3
Sea_su	100×10^3	3	2	突变型	3	25×10^3, 50×10^3, 75×10^3
Sea_gr	100×10^3	3	2	渐变型	3	25×10^3, 50×10^3, 75×10^3
Electricity	45.3×10^3	8	2			
Spam	9.3×10^3	500	2			
Weather	25.6×10^3	8	2			

类效果. 因本文采用的数据集实例数存在较大差异，所以将w统一设置为100.

3)漂移位点探测参数τ与漂移稳定性探测参数δ. 通过对实验中梯度提升残差的统计可以发现，不同漂移类型下残差波动率差异较为明显而流动残差比的最大值差异较小，因此结合实验所得残差数据，本文设置$\tau = 1.25$, $\delta = 0.01$.

3.4 实验结果与分析

为有效衡量EGE_CD方法的分类性能，本文选取OnlineBoosting[26], OneNet[29], DWM[33], Learn++.NSE[34], LeverageBag[35], OzaBaggingADWIN[36], OnlineRUSBoost[37]在精度、收敛性和鲁棒性3个方面进行对比实验和结果分析.

3.4.1 精度结果和分析

本节首先分析了在不同模型子树边界M下模型的表现性能. 表2展示了EGE_CD方法在不同参数下的平均实时精度. 可以看出，当模型子树边界达到1 000时，模型的平均实时精度相对较高. 当M继续增加时，模型在复杂度增加的同时，也因为较大集成模型的过拟合问题导致模型性能出现一定程度的下降. 而当模型子树边界降低时，模型复杂度减小，模型易出现欠拟合问题，进而无法更好地适应概念漂移数据. 因此，在后续的对比实验中，本文方法的模型子树边界M设置为1 000.

表3展示了不同方法在各个数据集上的平均实时精度以及平均排名情况. 可以看出，在所有数据集上EGE_CD方法均能取得最好的模型性能. 在标准数据集与Electricity数据集上，EGE_CD方法与其他方法的性能差异较为明显，这是因为该方法将梯度提升模型的残差拟合与学习器的增量更新结合，使

Table 2 Average Real-Time Accuracy Under Different M

表2 不同M下的平均实时精度

数据集	平均实时精度（排名）				
	M=500	M=1 000	M=1 500	M=2 000	M=2 500
RBFblips_su	0.838 2(3)	**0.838 9**(1)	0.838 6(2)	0.837 5(4)	0.836 9(5)
RBFblips_gr	0.781 3(2)	0.775 0(5)	0.779 1(4)	**0.783 9**(1)	0.780 4(3)
Hyperplane_su	0.916 4(3)	**0.917 4**(1)	0.915 7(4)	0.915 4(5)	0.916 7(2)
Hyperplane_gr	0.881 5(4)	0.882 5(2)	**0.883 3**(1)	0.882 2(3)	0.880 8(5)
Sea_su	0.839 6(3)	0.839 7(2)	0.839 2(4)	0.837 8(5)	**0.840 4**(1)
Sea_gr	0.824 9(4)	0.825 1(3)	0.824 9(5)	0.825 2(2)	**0.825 6**(1)
Electricity	0.885 1(5)	**0.888 3**(1)	0.887 4(3)	0.885 8(4)	0.887 5(2)
Spam	0.916 6(3)	0.916 9(2)	**0.917 2**(1)	0.915 7(4)	0.915 6(5)
Weather	0.831 5(3)	0.829 2(5)	**0.833 4**(1)	0.832 7(2)	0.829 7(4)
平均排名	3.33	2.44	2.78	3.33	3.11

注：黑体数值表示最高平均实时精度.

Table 3 Average Real-Time Accuracy of Different Methods on Each Dataset

表3 不同方法在各数据集上的平均实时精度

数据集	平均实时精度（排名）							
	OneNet	DWM	Learn++.NSE	LeverageBag	OnlineBoosting	OzaBaggingADWIN	OnlineRUSBoost	EGE_CD（本文）
RBFblips_su	0.803 5(3)	0.595 8(8)	0.671 6(7)	0.767 0(4)	0.726 6(6)	0.828 3(2)	0.755 7(5)	**0.838 9**(1)
RBFblips_gr	0.750 9(3)	0.574 9(8)	0.647 7(7)	0.719 3(5)	0.686 0(6)	0.756 7(2)	0.708 6(4)	**0.775 0**(1)
Hyperplane_su	0.876 4(3)	0.878 1(2)	0.806 8(7)	0.774 6(8)	0.812 6(6)	0.854 8(4)	0.814 7(5)	**0.917 5**(1)
Hyperplane_gr	0.836 0(2)	0.817 9(3)	0.754 8(7)	0.732 7(8)	0.762 3(6)	0.817 1(4)	0.767 3(5)	**0.882 5**(1)
Sea_su	0.833 2(2)	0.778 2(4)	0.761 9(7)	0.762 9(6)	0.761 7(8)	0.813 9(3)	0.767 3(5)	**0.839 7**(1)
Sea_gr	0.823 2(2)	0.772 2(4)	0.752 3(8)	0.754 5(6)	0.754 3(7)	0.808 6(3)	0.761 0(5)	**0.825 1**(1)
Electricity	0.855 2(2)	0.810 9(4)	0.815 8(3)	0.695 0(8)	0.708 0(7)	0.731 9(5)	0.711 3(6)	**0.888 3**(1)
Spam	0.957 8(3)	0.968 2(2)	0.924 1(4)	0.885 7(8)	0.885 8(7)	0.913 7(5)	0.890 7(6)	**0.969 9**(1)
Weather	0.819 3(2)	0.813 2(4)	0.787 9(5)	0.764 4(8)	0.766 9(7)	0.818 4(3)	0.778 2(6)	**0.829 2**(1)
平均排名	2.44	4.33	6.11	6.78	6.67	3.44	5.22	1

注：黑体数值表示最高平均实时精度.

模型在数据量较大时能保持较好的预测性能. 结合整体排名来看, EGE_CD 方法排名最高, 这说明该方法能够提高模型在流数据中的适应性, 具有较好处理不同类型概念漂移的能力.

图 5 展示了各个方法在所有数据集上的累积精度. 可以看出, 在所有数据集上, EGE_CD 方法的累积精度均高于其他方法. 这是由于该方法结合不同漂移类型对梯度提升模型进行适当剪枝与增量更新, 使模型具有较强的稳定性与适应性.

本文利用 Bonferroni-Dunn 测试[38] 计算了所有方法两者之间的显著性差异. 若 2 种方法的秩和平均差值大于临界差CD, 则这 2 种方法的性能存在显著差异.

$$CD = q_\alpha \sqrt{\frac{S(S+1)}{6N}}, \tag{24}$$

其中q_α为显著水平α下的临界值, S为方法个数, N为数据集个数, 经计算可得, 当$\alpha = 0.05$时, $CD = 3.106\ 1$. 不同方法的平均实时精度和最终累积精度的统计分析结果如图 6 所示. 其中, 平均序值在一个临近值域内的方法用黑线连接. 结果表明, 在统计意义上, 本文所提的 EGE_CD 方法排名最好且具有明显优势.

3.4.2 消融实验

为验证本文方法中弹性梯度剪枝与增量梯度生长结合的有效性, 实验比较了 EGE_CD 在模型子树边界$M \in \{1\ 000, 1\ 500, 2\ 500\}$时采用弹性梯度剪枝(EGP)、增量梯度生长(IGG)和结合 2 种方法(EGP+IGG)的实时精度与累积精度. 由图 7 可以看出, 在不同的模型子树边界下, 若单独采用弹性梯度剪枝, 模型在大部分数据集上表现最差, 这是由于在模型剪枝后未能及时增加学习器, 提高模型适应性, 若单独采用增量梯度生长, 模型表现虽有所提升但仍明显弱于 EGE_CD. 结果表明, 在不同的模型子树边界下, EGE_CD 均取得了良好的泛化性能, 有效验证了该方法的合理性.

3.4.3 模型收敛性分析

当流数据发生概念漂移后, 在线学习模型能否快速收敛到新的数据分布是衡量算法的重要指标.

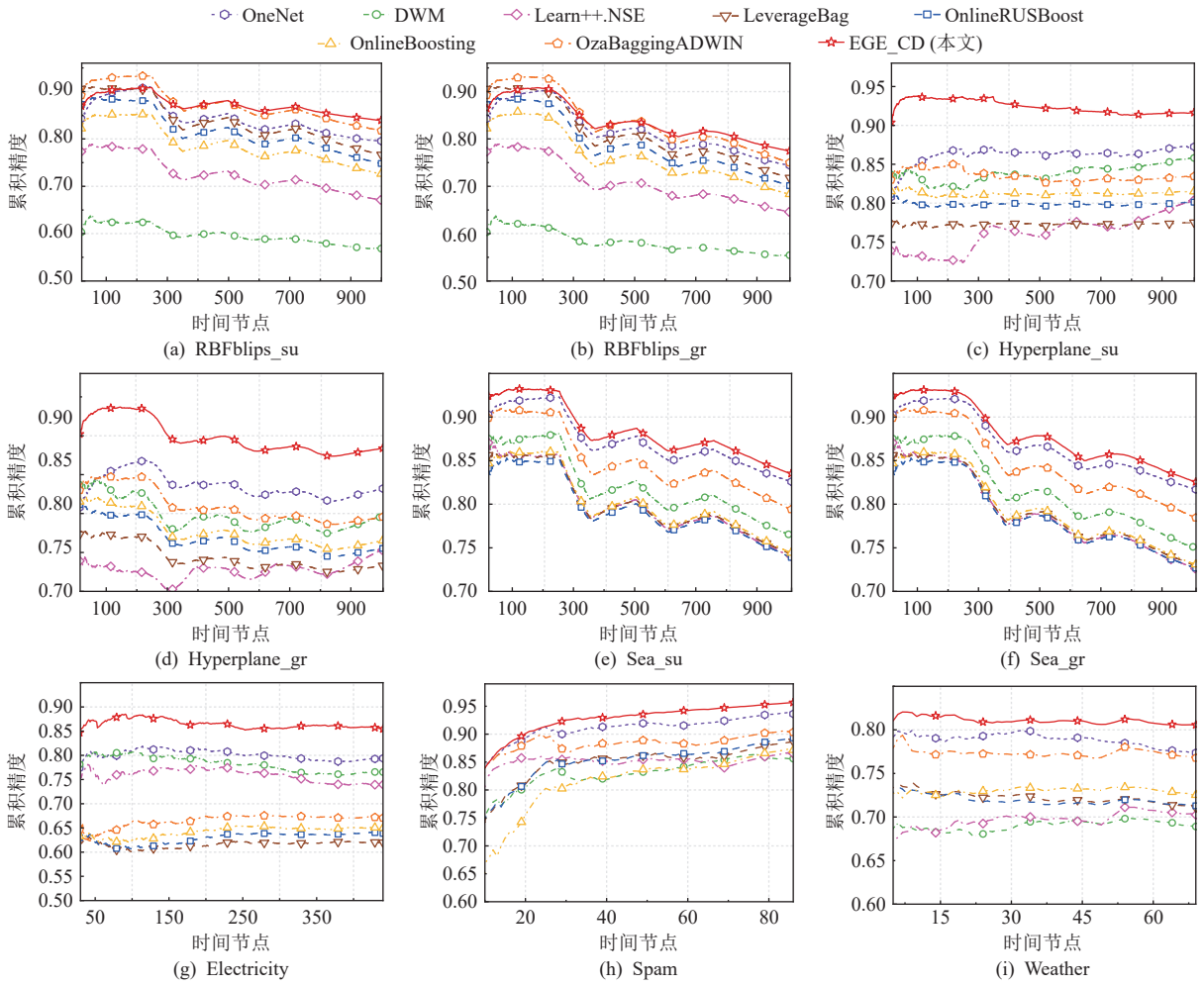
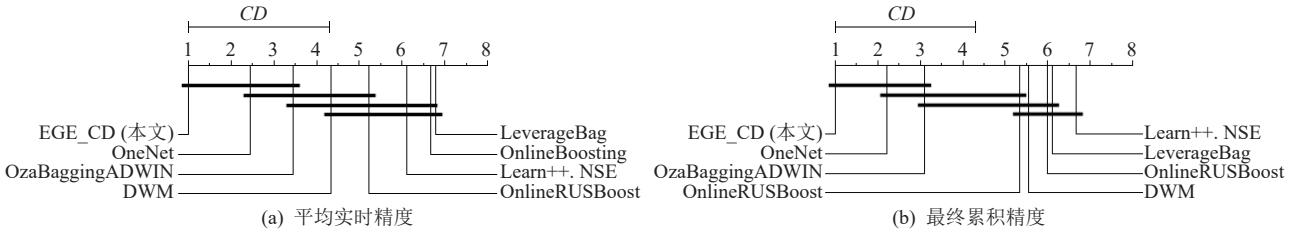


Fig. 5 Comparison of cumulative accuracy of different methods on each dataset

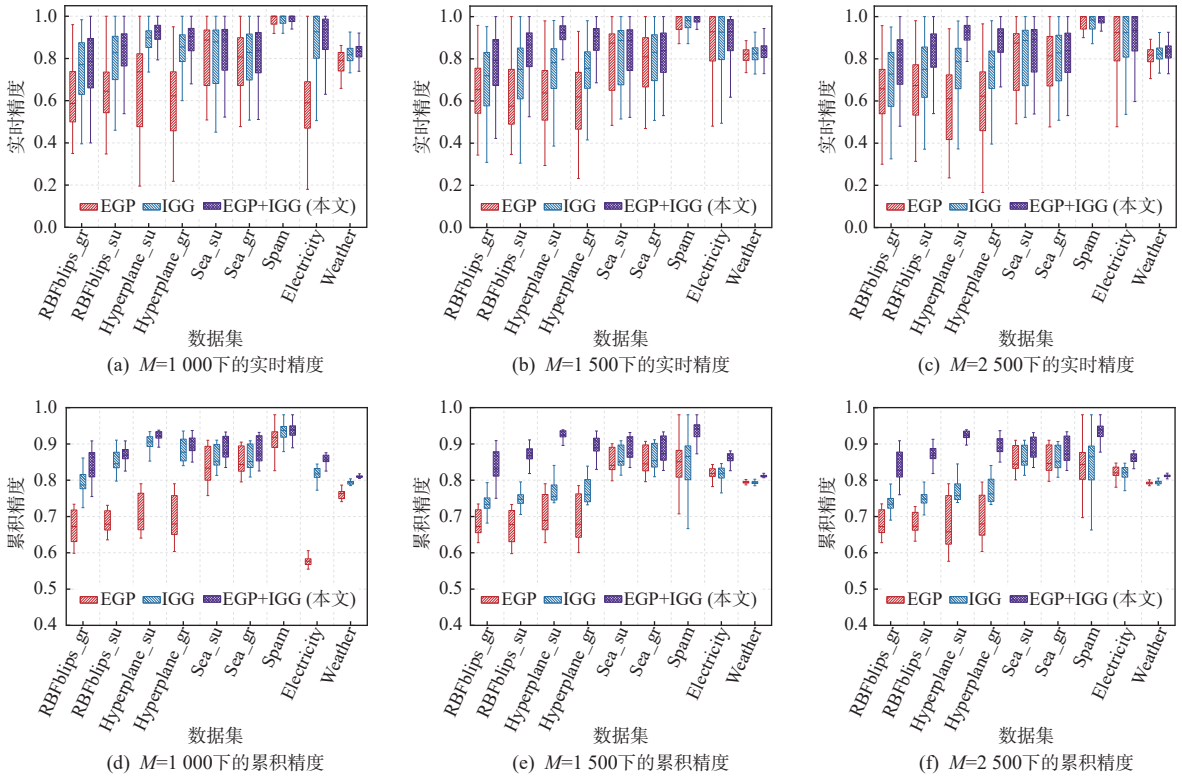
图 5 不同方法在各数据集上的累积精度比较



(a) 平均实时精度 (b) 最终累积精度

Fig. 6 Bonferroni-Dunn test results of average real-time accuracy and final cumulative accuracy of different methods

图 6 不同方法平均实时精度和最终累积精度的 Bonferroni-Dunn 检验结果



(a) M=1 000下的实时精度 (b) M=1 500下的实时精度 (c) M=2 500下的实时精度

(d) M=1 000下的累积精度 (e) M=1 500下的累积精度 (f) M=2 500下的累积精度

Fig. 7 Comparison of real-time accuracy and cumulative accuracy of three methods under different M

图 7 不同 M 下 3 种方法的实时精度与累积精度比较

表 4 展示了不同方法在已知漂移位点的 6 个数据集上的对应漂移位点处的收敛速度. 表 4 中每个方法对应数据集的 3 个值分别代表前、中、后 3 个位点的收敛速度. 从表 4 中可以看出, EGE_CD 方法在多数情况下都具有较快的收敛速度, 这是由于该方法在概念漂移发生后及时对模型进行弹性剪枝与增量生长, 使模型能够在漂移发生后拥有较好的适应能力. 在整体排名中 EGE_CD 方法处于第 1, 说明该方法具有较快的收敛速度, 收敛性能较好.

3.4.4 模型鲁棒性分析

为了衡量各个方法的算法稳定性, 本节计算每个方法在各个数据集上的鲁棒性. 图 8 展示了计算结果, 不同的小矩形高度代表的是算法在不同数据集上的鲁棒性值大小, 每一列上的数值代表算法在所有数据集上的鲁棒性值总和, 即该算法的整体鲁棒性. 可以看出, 在多数情况下, EGE_CD 方法的鲁棒性都能取得较好的排名, 且整体鲁棒性最高, 这说明该方法在含概念漂移的流数据中具有更强的鲁棒性, 能够提高模型整体的稳定性.

4 结束语

针对流数据中的不同类型概念漂移导致模型性能存在差异的问题, 本文提出一种弹性梯度集成的概念漂移适应方法. 该方法利用流动残差比与残差波动率来检测不同类型概念漂移的发生; 随后根据梯度损失变化提取漂移学习器, 再结合不同漂移类型与残差分布, 实现弹性梯度剪枝, 提高模型稳定性;

Table 4 Recovery Speed Under Accuracy of Different Methods on Each Dataset

表 4 不同方法在各数据集上的收敛速度

数据集	OneNet	DWM	Learn++.NSE	LeverageBag
RBFBlips_su	2.73/1.77/3.06	3.39/24.89/18.52	6.27/1.99/4.43	2.76/4.02/8.62
RBFBlips_gr	2.09/1.18/1.79	6.28/1.56/3.57	2.49/9.82/5.18	**1.08**/1.38/2.73
Hyperplane_su	3.70/3.08/2.48	3.29/3.36/2.00	2.40/2.75/2.56	4.61/3.58/3.53
Hyperplane_gr	**1.16**/1.83/2.76	18.26/17.88/7.33	2.05/4.60/3.33	3.82/4.17/2.11
Sea_su	2.46/3.07/2.94	4.42/7.46/4.12	2.13/4.18/5.76	2.70/7.99/2.76
Sea_gr	1.15/1.67/1.96	1.75/2.31/2.67	0.96/3.86/2.69	3.75/5.93/2.39
平均排名	3.33	6.33	5.28	5.56

数据集	OnlineBoosting	OzaBaggingADWIN	OnlineRUSBoost	EGE_CD（本文）
RBFBlips_su	1.84/1.80/6.57	2.23/3.40/5.08	**1.74**/2.19/1.84	7.56/**1.68**/**1.42**
RBFBlips_gr	1.88/3.57/2.18	1.67/1.83/2.24	2.36/3.42/7.62	2.41/**1.14**/**0.91**
Hyperplane_su	1.61/**1.49**/**1.79**	8.10/8.59/1.98	2.59/2.54/2.82	**0.96**/2.05/2.46
Hyperplane_gr	3.26/3.45/4.64	3.35/1.05/5.26	2.58/3.86/2.72	1.56/**0.51**/**0.34**
Sea_su	2.84/2.80/6.33	3.88/2.80/4.17	**1.43**/2.48/2.34	2.74/**2.46**/**0.93**
Sea_gr	3.02/1.86/3.56	1.52/**1.55**/1.81	1.25/2.45/2.05	**0.84**/2.10/**1.64**
平均排名	4.33	4.56	4	**2.33**

注：数据集均包含前、中、后 3 个漂移位点，对应 3 个收敛速度；黑体数值表示最高收敛速度.

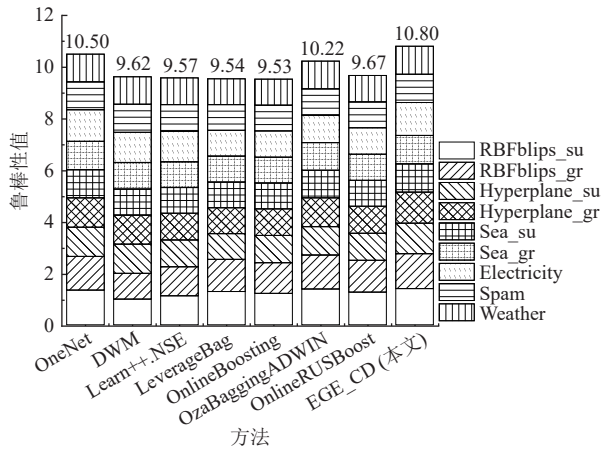


Fig. 8 Comparison of robustness of different methods

图 8 不同方法的鲁棒性比较

最后，结合增量学习方法，通过滑动采样数据计算最优拟合率，并根据残差变化实现增量梯度生长，提高模型适应性. 本文在 9 个具有不同漂移类型的数据集上进行了实验. 实验结果表明，与其他方法相比，本文方法在适应不同类型概念漂移方面具有显著优势，但可以发现，在发生剧烈的概念漂移时，本文方法的漂移恢复速度较慢，因此在未来工作中，将进一步研究提高在线学习模型收敛速度的方法，并探讨模型的可解释性，将其应用于存在多种概念漂移类型的实际问题中. 这将有利于解决概念漂移问题中在线学习模型收敛慢的问题，实现在线学习模型在更多漂移类型上的应用.

作者贡献声明：郭虎升提出思想，负责方法设计、论文写作及修改；张羽桐负责论文写作、代码实现、数据测试及论文修改；王文剑负责写作指导、论文的修改审定.

参 考 文 献

[1] Habeeb R A A, Nasaruddin F, Gani A, et al. Real-time big data processing for anomaly detection: A survey[J]. International Journal of Information Management, 2019, 45: 289−307

[2] Zhai Tingting, Gao Yang, Zhu Junwu. Survey of online learning algorithms for streaming data classification[J]. Journal of Software, 2020, 31(4): 912−931 (in Chinese)
(翟婷婷, 高阳, 朱俊武. 面向流数据分类的在线学习综述[J]. 软件学报, 2020, 31(4): 912−931)

[3] Du Hangyuan, Wang Wenjian, Bai Liang. A novel evolving data stream clustering method based on optimization model[J]. SCIENTIA SINICA Informationis, 2017, 47(11): 1464−1482 (in Chinese)
(杜航原, 王文剑, 白亮. 一种基于优化模型的演化数据流聚类方法[J]. 中国科学: 信息科学, 2017, 47(11): 1464−1482)

[4] Wen Yimin, Liu Shuai, Miao Yuqing, et al. Survey on semi-supervised classification of data streams with concept drifts[J]. Journal of Software, 2022, 33(4): 1287−1314 (in Chinese)
(文益民, 刘帅, 缪裕青, 等. 概念漂移数据流半监督分类综述[J]. 软件学报, 2022, 33(4): 1287−1314)

[5] Jothimurugesan E, Hsieh K, Wang J, et al. Federated learning under distributed concept drift[C]//Proc of the 26th Int Conf on Artificial Intelligence and Statistics. New York: PMLR, 2023: 5834−5853

[6] Liu Sanmin, Xue Shan, Wu Jia, et al. Online active learning for drifting data streams[J]. IEEE Transactions on Neural Networks and Learning Systems, 2021, 34(1): 186−200

[7] Lu Jie, Liu Anjin, Dong Fan, et al. Learning under concept drift: A review[J]. IEEE Transactions on Knowledge and Data Engineering, 2018, 31(12): 2346−2363

[8] Karimian M, Beigy H. Concept drift handling: A domain adaptation perspective[J]. Expert Systems with Applications, 2023, 224: 119946

[9] Chen Yingying, Yang Xiaowei, Dai Hongliang. Cost-sensitive continuous ensemble kernel learning for imbalanced data streams with concept drift[J]. Knowledge-Based Systems, 2024, 284: 111272

[10] Gomes H M, Barddal J P, Enembreck F, et al. A survey on ensemble learning for data stream classification[J]. ACM Computing Surveys, 2017, 50(2): 1−36

[11] Liu Weike, Zhang Hang, Ding Zhaoyun, et al. A comprehensive active learning method for multiclass imbalanced data streams with concept drift[J]. Knowledge-Based Systems, 2021, 215: 106778

[12] Celik B, Vanschoren J. Adaptation strategies for automated machine learning on evolving data[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2021, 43(9): 3067−3078

[13] Bifet A, Gavalda R. Learning from time-changing data with adaptive windowing[C]//Proc of the 7th SIAM Int Conf on Data Mining. Philadelphia, PA: SIAM, 2007: 443−448

[14] Gama J, Medas P, Castillo G, et al. Learning with drift detection [C]//Proc of the 17th Brazilian Symp on Artificial Intelligence. Berlin: Springer, 2004: 286−295

[15] Guo Husheng, Ren Qiaoyan, Wang Wenjian. Concept drift class detection based on time window[J]. Journal of Computer Research and Development, 2022, 59(1): 127−143 (in Chinese)
(郭虎升, 任巧燕, 王文剑. 基于时序窗口的概念漂移类别检测[J]. 计算机研究与发展, 2022, 59(1): 127−143)

[16] Neto A F, Canuto A M P. EOCD: An ensemble optimization approach for concept drift applications[J]. Information Sciences, 2021, 561: 81−100

[17] Hinder F, Artelt A, Hammer B. Towards non-parametric drift detection via dynamic adapting window independence drift detection (DAWIDD)[C]//Proc of the 37th Int Conf on Machine Learning. New York: PMLR, 2020: 4249−4259

[18] Gözüaçık Ö, Can F. Concept learning using one-class classifiers for implicit drift detection in evolving data streams[J]. Artificial Intelligence Review, 2021, 54(5): 3725−3747

[19] Xu Shuliang, Feng Lin, Liu Shenglan, et al. Self-adaption neighborhood density clustering method for mixed data stream with concept drift[J]. Engineering Applications of Artificial Intelligence, 2020, 89: 103451

[20] Street W N, Kim Y S. A streaming ensemble algorithm (SEA) for large-scale classification[C]//Proc of the 7th ACM SIGKDD Int Conf on Knowledge Discovery and Data Mining. New York: ACM, 2001: 377−382

[21] Lu Yang, Cheung Y N, Tang Yuanyan. Adaptive chunk-based dynamic weighted majority for imbalanced data streams with concept drift[J]. IEEE Transactions on Neural Networks and Learning Systems, 2019, 31(8): 2764−2778

[22] Guo Husheng, Zhang Shuai, Wang Wenjian. Selective ensemble-based online adaptive deep neural networks for streaming data with concept drift[J]. Neural Networks, 2021, 142: 437−456

[23] Weinberg A I, Last M. EnHAT—Synergy of a tree-based ensemble with Hoeffding adaptive tree for dynamic data streams mining[J]. Information Fusion, 2023, 89: 397−404

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

[25] Brzezinski D, Stefanowski J. Reacting to different types of concept drift: The accuracy updated ensemble algorithm[J]. IEEE Transactions on Neural Networks and Learning Systems, 2013, 25(1): 81−94

[26] Oza N C, Russell S J. Online bagging and boosting[C]//Proc of the 8th Int Workshop on Artificial Intelligence and Statistics. New York: PMLR, 2001: 229−236

[27] Breiman L. Bagging predictors[J]. Machine Learning, 1996, 24(2): 123−140

[28] Zyblewski P, Sabourin R, Woźniak M. Preprocessed dynamic classifier ensemble selection for highly imbalanced drifted data streams[J]. Information Fusion, 2021, 66: 138−154

[29] Zhang Yifan, Wen Qingsong, Wang Xue, et al. OneNet: Enhancing time series forecasting models under concept drift by online ensembling[C] //Proc of the 37th Int Conf on Neural Information Processing Systems. New York: ACM, 2023: 69949−69980

[30] Friedman J H. Greedy function approximation: A gradient boosting machine[J]. Annals of Statistics, 2001, 29(5): 1189−1232

[31] Bifet A, Holmes G, Pfahringer B, et al. Moa: Massive online analysis, a framework for stream classification and clustering[C]//Proc of the 1st Workshop on Applications of Pattern Analysis. New York: PMLR, 2010: 44−50

[32] Zhao Peng, Zhou Zhihua. Learning from distribution-changing data streams via decision tree model reuse[J]. SCIENTIA SINICA Informationis, 2021, 51(1): 1−12 (in Chinese)
(赵鹏, 周志华. 基于决策树模型重用的分布变化流数据学习[J]. 中国科学: 信息科学, 2021, 51(1): 1−12)

[33] Kolter J Z, Maloof M A. Dynamic weighted majority: An ensemble method for drifting concepts[J]. The Journal of Machine Learning Research, 2007, 8: 2755−2790

[34] Elwell R, Polikar R. Incremental learning of concept drift in nonstationary environments[J]. IEEE Transactions on Neural Networks, 2011, 22(10): 1517−1531

[35] Bifet A, Holmes G, Pfahringer B. Leveraging bagging for evolving data streams[C]//Proc of the Machine Learning and Knowledge Discovery in Databases: European Conf. Berlin: Springer, 2010: 135−150

[36] Bifet A, Holmes G, Pfahringer B, et al. New ensemble methods for evolving data streams[C]//Proc of the 15th ACM SIGKDD Int Conf

on Knowledge Discovery and Data Mining. New York: ACM, 2009: 139−148

[37] Wang B, Pineau J. Online bagging and boosting for imbalanced data streams[J]. IEEE Transactions on Knowledge and Data Engineering, 2016, 28(12): 3353−3366

[38] Demšar J. Statistical comparisons of classifiers over multiple data sets[J]. The Journal of Machine Learning Research, 2006, 7: 1−30



Guo Husheng, born in 1986. PhD, professor, PhD supervisor. Senior member of CCF. His main research interests include machine learning, data mining, and computational intelligence.

郭虎升, 1986 年生. 博士, 教授, 博士生导师. CCF 高级会员. 主要研究方向为机器学习、数据挖掘、计算智能.



Zhang Yutong, born in 1999. Master candidate. His main research interests include stream data mining and online machine learning.

张羽桐, 1999 年生. 硕士研究生. 主要研究方向为流数据挖掘、在线机器学习.



Wang Wenjian, born in 1968. PhD, professor, PhD supervisor. Distinguished member of CCF. Her main research interests include machine learning, data mining, and computational intelligence.

王文剑, 1968 年生. 博士, 教授, 博士生导师. CCF 杰出会员. 主要研究方向为机器学习、数据挖掘、计算智能.