一种基于双层窗口的概念漂移数据流分类算法

朱 群 [1] 张玉红 [1] 胡学钢 [1] 李培培 [1]

摘 要 数据流中概念漂移问题的研究已成为近年来流数据挖掘领域的研究热点之一. 已有的研究工作多依据单窗口中错误率的变化来检测概念漂移, 难以适应不同类型的漂移. 为此, 本文提出一种新的基于双层窗口机制的数据流分类算法 (Double-windows-based classification algorithm for concept drifting data streams, DWCDS), 该算法采用随机决策树模型构建集成分类器, 利用双层窗口机制周期性地检测滑动窗口中流数据分布的变化, 并动态地更新模型以适应概念漂移. 分析与实验结果表明: 该算法可以快速有效地跟踪检测含噪数据流中的概念漂移, 且抗噪性能与分类精度显著提高.

关键词 数据流, 概念漂移, 分类, 随机决策树, 滑动窗口

DOI 10.3724/SP.J.1004.2011.01077

A Double-window-based Classification Algorithm for Concept Drifting Data Streams

ZHU Qun[1] ZHANG Yu-Hong[1] HU Xue-Gang[1] LI Pei-Pei[1]

Abstract Tracking concept drifts in data streams has recently become a hot topic in data mining. Most of the existing work is built on a single-window-based mechanism to detect concept drifts. Due to the inherent limitation of the single-window-based mechanism, it is a challenge to handle different types of drifts. Motivated by this, a new classification algorithm based on a double-window mechanism for handling various concept drifting data streams (DWCDS) is proposed in this paper. In terms of an ensemble classifier in random decision trees, a double-window-based mechanism is presented to detect concept drifts periodically, and the model is updated dynamically to adapt to concept drifts. Extensive studies on both synthetic and real-word data demonstrate that DWCDS could quickly and efficiently detect concept drifts from streaming data, and the performance on the robustness to noise and the accuracy of classification is also improved significantly.

Key words Data stream, concept drift, classification, random decision tree, sliding widow

数据流已广泛出现在如网络安全、股票分析等实际应用领域[1−2], 这些数据具有快速性、连续性、多变化和无限性等特点[3], 且概念漂移[4] 现象常常出现. 这使得建立在原始数据集上的模型不再适应, 从而给传统分类问题提出极大的挑战.

目前, 针对数据流中的概念漂移问题, 已提出了一些新的分类算法. 其中, 包括基于单棵决策树模型的算法[5]、基于集成分类器的算法[6−8] 等. 然而, 上述算法多采用单滑动窗口机制在分类器局部结构中检测数据是否发生漂移, 存在单窗口机制固有的弱点, 即 "窗口值较大有利于低漂移率的数据流处理, 却不适应新的目标函数; 而小窗口能较快地适应概念漂移 (突变式或渐近式), 却常常由于事例不足导致学习不充分[9]".

因此, 本文提出一种新的依据窗口中原始数据分布变化检测概念漂移的数据流分类算法 DWCDS (Double-window-based classification algorithm for concept drifting data streams), 该算法采用双层窗口机制跟踪概念漂移, 并动态地调整窗口大小以增强算法的适应性, 克服了单窗口机制检测概念漂移的不足. 实验表明: 与单层窗口相比, 双层窗口机制有较优的漂移检测能力; 此外, DWCDS 算法具有较优的概念漂移处理能力, 且与基于单分类器模型的数据流概念漂移检测算法 CVFDT (Concept-adapting very fast decision tree learner)[5]、HT-DDM (A single Hoeffding tree with a drift detection method)[10] 和 HT-EDDM (A single Hoeffding tree with an early drift detection method)[11], 及基于集成分类模型的 MSRT (Multiple semi-random decision trees for concept-drifting data streams)[7]、CDRDT (A streaming data algorithm

收稿日期 2010-09-29 录用日期 2011-03-18
Manuscript received September 29, 2010; accepted March 18, 2011
国家重点基础研究发展计划 (973 计划) (2009CB326203), 国家自然科学基金 (60975034), 安徽省自然科学基金 (090412044), 合肥工业大学数据挖掘与智能计算研究中心 "千人计划" 团队人才培养专项基金 (2010HGXJ0715) 资助
Supported by National Basic Research Program of China (973 Program) (2009CB326203), National Natural Science Foundation of China (60975034), Natural Science Foundation of Anhui Province (090412044), and the Special Funds of Thousand Talents Program (2010HGXJ0715)
1. 合肥工业大学计算机与信息学院 合肥 230009
1. School of Computer and Information, Hefei University of Technology, Hefei 230009

for concept drifts in random decision tree)[8] 算法相比, DWCDS 大大提高了抗噪性能与分类精度.

本文的组织结构如下: 第 1 节介绍数据流中相关工作; 第 2 节介绍 DWCDS 的设计思想及其特点; 第 3 节对算法进行了实验比较和分析; 最后进行了总结与展望.

1 相关工作

近年来, 许多学者针对概念漂移数据流的分类问题提出了大量的算法与模型. Hulten 等于 2001 年提出基于单棵决策树模型的 CVFDT 算法, 采用滑动窗口机制为每个树结点创建替换子树, 周期性检测结点所在的概念是否发生漂移, 一旦发现替换子树的分类正确率优于旧子树, 则用此子树替换旧子树以确保模型反应最新概念. 然而, 树中每个结点仅包含检测周期中的部分数据, 未考察整个检测窗口中的数据分布变化. Gama 等于 2004 年提出一种新的概念漂移检测方法—HT-DDM 方法, 该方法实时监控模型的分类错误率, 并采用伯努利分布设置阈值区分概念漂移与噪音. 然而, 该方法不适于处理渐进式概念漂移, 基于此, Baena–Garcia 等于 2006 年提出改进的概念漂移检测方法—HT-EDDM 方法, 该方法同时依据模型的错误率变化与错误率之间的距离 (事例数) 判断概念漂移. Nishida 等于 2008 年提出一种突变式概念漂移检测方法—LID (A leaky integrate-and-detect) 方法[12], 依据当前模型的信息度和分类精度来检测概念漂移.

此外, 研究发现构造一系列简单的分类器比建立一个复杂的单分类器更加简单可行[13−15], 因此, 许多学者开始尝试采用构建集成分类器的方法处理数据流, 主要工作如下: Kolrer 等 2003 年提出的 DWM 算法[13]、2005 年提出 AddExp 算法[14], 以及孙岳等于 2008 年提出的 M_ID4 算法[16] 均采用不断更替模型中的基分类器的机制来适应概念漂移数据流, 却未真正检测概念漂移何时发生. Fan 等于 2004 年提出基于随机决策树的数据流概念漂移方法[15, 17], 利用新旧数据对应模型的精确度检测概念是否发生漂移, 并采用交叉验证机制提高分类精确度. Gama 等于 2005 年提出基于决策森林的 UFFT (An ultra fast forest tree system) 算法[6], 将多类问题划分成两类问题, 利用线性判别方法与贝叶斯分类器检测漂移, 而该算法仅可处理连续属性. Li 等于 2008 年提出基于半随机决策树模型的 MSRT 算法, 采用训练与测试双窗口机制, 利用 Hoeffding 边界不等式设定区分概念漂移与噪音的阈值. 在此基础上, Li 等于 2010 年提出了基于随机决策树模型的 CDRDT 算法. Wu 等于 2009 年设计的 AEC (An adaptive ensemble classifier for concept drifting stream) 算法[18] 基于在线 Bagging 策略构建集成分类器, 并使用新到达的事例不断地更新模型. 以上涉及的几种算法与 CVFDT 相似, 均利用决策树某个分支中保存的检测窗口中部分数据的分布变化检测概念漂移, 却忽视了整个检测窗口中数据分布的变化.

不同于以上提及的算法, 本文提出的 DWCDS 算法具有以下两方面的特点: 1) 基于随机决策树构建集成分类器, 有效提高了算法的抗噪性与分类精度; 2) 采用一种新的基于双层窗口检测概念漂移的机制, 并依据窗口中数据分布的变化判断概念漂移.

2 DWCDS: 基于双层窗口的概念漂移数据流分类算法

2.1 算法描述

本文基于双层窗口机制提出概念漂移数据流分类算法 DWCDS, 该算法构建 N 个基分类器, 每个基分类器由 K 棵随机决策树组成, 从而构成双层集成分类器. 利用不断到来的训练数据增量式地更新决策树模型, 同时周期性地检测滑动窗口中流数据分布的变化判断概念漂移, 并动态地调整窗口的大小. 算法基本思想描述如下:

算法 1. DWCDS

Input. 训练数据集 $DSTR$; 测试数据集 $DSTE$; 属性集合 A; 集成分类器 CT; 树的最大高度 h_0; 基分类器个数 N; 每个基分类器的容量 K; 滑动窗口 SW; 最小窗口阈值 $MinSW$; 最大窗口阈值 $MaxSW$; 基本窗口 w; 结点分割需要的最小事例数 $n_{\min}$; 漂移警告系数 τ_1; 漂移系数 τ_2.

Output. 分类错误率.

1. For (i=1; i<N; i++);
2. 采用 SW 中数据块增量式的构建 K 棵随机决策树作为基分类器 CT_i;
3. While (新数据到来)
4. If (滑动窗口 SW 满时)
5. 利用新到达的数据增量式地更新双层集成分类器 CT, 并依据滑动窗口与基本窗口中数据分布的变化判断当前流数据是否发生了概念漂移;
6. If (发生了漂移)
7. 从集成分类器 CT 中删除一个分类效果最差的基分类器 CT_i;
8. 构建一个新的基分类器加入 CT 中; 调整滑动窗口的大小;
9. For (每个测试事例)
10. 每个基分类器中的 K 棵随机决策树采用投票机制预测分类结果;
11. 投票选择 N 个基分类器的预测结果.

2.2 算法处理机制

2.2.1 双层集成分类器的构建

由于随机决策树具有较强的抗噪性, 且时空性能较优[19−20], 因此, DWCDS 算法增量式地创建 N

个包含 K 棵随机决策树的双层集成分类器, 其中, 在构建随机决策树时, 决策路径上属性的选择以及连续属性的离散化方法类似于 MSRT 算法.

为了提高算法的精度, 我们采用了构建双层集成分类器的方法. 对于属性维数较少的数据集, 为了避免重复构建随机决策树以及提高算法的时空性能, 每个基分类器中的随机决策树的个数 K 定义为$\min\{C_{|Attrs|}^{\lceil |Attrs|/2 \rceil}, 10\}$, 其中, $\lceil \cdots \rceil$ 为向上取整, $|Attrs|$ 为属性个数, $\lceil |Attrs| \rceil$[21−22] 为决策树的最大高度 h_0.

本文采用了最大类和 Naive Bayes 两种分类预测方法. 最大类方法适宜任何数据库, 但分类准确性和抗噪性不如 Naive Bayes; 而 Naive Bayes 却受属性间独立性的限制. 因此, 实验时依据数据集的属性特征分别采用了不同的分类预测方法.

2.2.2　双层窗口的概念漂移检测机制

本文提出一种新的双层窗口机制, 通过周期性检测窗口中流数据分布的变化来判断概念漂移. 双层窗口指的是由多个基本窗口 w_i 组成的滑动窗口 (见图 1), 其形式化表示为 $SW = \{w_1, w_2, \cdots, w_{n-1}, w_n\}$.

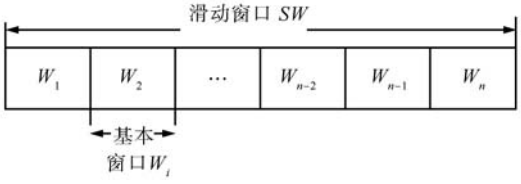


图 1　双层窗口结构图

Fig. 1　The structure diagram of a double-window

漂移检测时, 算法依据模型在滑动窗口中的分类错误率变化来判断原始数据的分布变化情况. 如果数据分布发生了变化, 现有模型不再适应当前的数据分布, 分类错误率则会上升; 如果数据分布未发生变化, 随着事例数目的增加, 依据统计结果表明模型的分类错误率将会下降[10−11, 13−15, 17, 19−23]. 此外, 为避免噪音数据的影响, 本文采用文献 [6] 中伯努利分布的方法, 设置不同的阈值区分概念漂移与噪音, 即对于一系列训练数据来说, 错误率是一个满足伯努利分布的随机变量, 对应的错误率和标准差分别为 $p_i \equiv error_i/i$, $s_i \equiv \sqrt{p_i(1-p_i)/i}$, 其中, i 为总事例个数, $error_i$ 为误分类的事例个数. 具体的漂移检测过程如下:

首先, 计算出当前滑动窗口 SW_i 中每个基本窗口 w_i 对应的错误率 p_i 与标准差 s_i; 然后, 以 w_n 中的数据作为最新的概念开始寻找漂移点 (如图 2 和 3). 考虑到概念漂移可能发生在滑动窗口的内部, 也可能发生在滑动窗口的最始端, 因此, 算法 DWCDS 分别考察这两种情况:

1) 概念漂移发生在滑动窗口的内部, 如图 2 所示. 算法从 w_n 开始依次往前查找可疑的漂移点, 若检测到 w_j 时, 发现满足不等式

$$p_j + \tau_1 \cdot s_j \le p_n \tag{1}$$

则认为从 w_j 开始进入漂移警报, 此时, 合并 $w_1 \sim w_j$ 与 $w_{j+1} \sim w_n$, 分别计算出新的均值 P、P' 和标准差 S, 若满足不等式

$$P + \tau_2 \cdot S \le P' \tag{2}$$

则认为发生了概念漂移; 反之, 认为是噪音影响 (其中, τ_1、τ_2 为常系数, 且 $\tau_1 < \tau_2$).

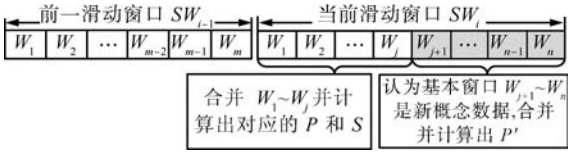


图 2　漂移点在滑动窗口的内部

Fig. 2　Drift within the sliding window

2) 概念漂移发生在滑动窗口的最始端, 如图 3 所示. 算法检测到 W_1 时仍未发现各基本窗口与 W_n 间有差异, 为了排除概念漂移发生在窗口最始端的可能性, 合并 SW_i 计算出其错误率 P' 并与 SW_{i-1} 的错误率 P、标准差 S 使用不等式 (2) 进行判定, 若满足条件, 则认为发生了概念漂移; 反之, 没有发生概念漂移.

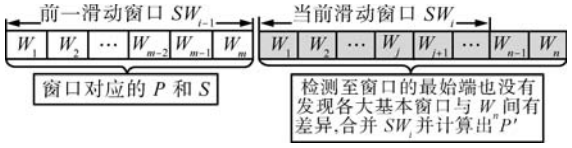


图 3　漂移点在滑动窗口的最始端

Fig. 3　Drift at the beginning of sliding window

算法假设每个滑动窗口中仅发生一次概念漂移, 因此, 漂移检测时采用从前往后和从后往前的查找策略效果是相同的, 而本文采用了后者. 与单窗口的检测机制相比, 双层窗口具有以下特点: 1) 抗噪性: 当判断出漂移预警时, 对各基本窗口合并, 相当于增大了窗口尺寸, 可以避免样本过少, 学习不充分或噪音造成的错误率偏差, 提高方法的抗噪性能; 2) 灵活性: 双层窗口检测过程中进行了划分和合并操作, 可根据具体的检测情况确定窗口尺寸. 其中,

划分操作解决了窗口尺寸较大时, 不能适应新目标函数的问题; 对各基本窗口的合并操作克服了窗口尺寸较小时, 由于实例不足引起的学习不充分问题.

另外, 为了使分类器适应概念漂移, 采取用新数据构建新的基分类器来替换原集成分类器最差的一个基分类器的方法. 此外, 为适应不同特征的概念漂移 (渐近式和突变式[24]), 算法根据检测结果对窗口大小进行动态调整, 并设置了最大最小界 ($MaxSW$ 与 $MinSW$). 若检测到概念漂移, SW 减少一个基本窗口的大小, 如果此时滑动窗口的大小已达到最小阈值 $MinSW$, 则保持原值不变; 若没有检测到概念漂移, SW 则增加一个基本窗口的大小, 直至达到最大阈值 $MaxSW$ 时不再增加.

2.3 算法分析

时间复杂度分析: DWCDS 算法的时间开销主要包括增量式建树、漂移检测和分类预测三部分. 与其他几种算法的性能比较见表 1. 其中, $|con|$、$|dis|$ 分别表示连续型和离散型结点的个数. 对于离散型结点所需的处理时间仅为 O(1), 而连续型结点需储存信息并进行离散化, 因此, 处理时间记为 O(t). n 表示滑动窗口中所含基本窗口 w_i 的个数; $|m_j^A|$ 表示属性 A_j 的取值个数; O(S_i) 表示数据块 i 中的事例个数; $|classes|$ 表示数据集的类别个数; $|n_{leaf}|$ 表示叶子节点的事例个数, $|Attrs|$ 表示数据集的属性个数.

由表 1 可知, 在增量式建树阶段, 算法 DWCDS 的时间复杂度同 CDRDT 算法, 低于其他两种方法. 而针对漂移检测步骤, 算法 DWCDS 仅与双层窗口中基本窗口个数相关, 所需时间远远小于其他几种算法, 因此, 漂移检测过程本身耗时非常少. 在分类预测阶段, 各算法的时间消耗均相同.

空间复杂度分析: 算法 DWCDS 的空间消耗主要在于分类器的构建与滑动窗口的存储. 其中, 分类器的空间消耗为 O($|dn| \cdot |v| \cdot |classes| + |ln| \cdot |Attrs| \cdot |v| \cdot |classes|$), $|dn|$、$|ln|$ 分别表示决策结点和叶子结点的个数, $|v|$ 表示各属性维取值的最大个数. 而滑动窗口的空间消耗为 O($k \cdot |w_i| \cdot |e|$), k 表示所含基本窗口 w_i 的个数, $|w_i|$ 表示基本窗口的大小, O(e) 表示每个事例的存储空间. 算法 CDRDT 的空间消耗同 DWCDS 的分类器空间消耗; 算法 CVFDT 和 MSRT 的空间消耗分别为 O($|dn| \cdot |Attrs| \cdot |v| \cdot |classes| + |ln|$) 和 O($|t| + |window| \cdot |e|$).

3 实验与性能分析

为验证 DWCDS 算法概念漂移检测机制的有效性, 本文选择了若干基准漂移数据集对 DWCDS 算法以及与该算法分类模型相同的单窗口机制漂移检测过程 (SWCDS) 进行了实验对比. 同时, 实验还对比了该算法与 MSRT 算法、CDRDT 算法、CVFDT 算法、MOA[25] (大型在线分析工具) 中提供的 HT-DDM 算法 (基于单棵 Hoeffding 树的 "Learning with drift detection" 概念漂移方法) 和 HT-EDDM 算法 (基于单棵 Hoeffding 树的 "Early drift detection method" 概念漂移方法) 在抗噪性及分类精度等方面的性能表现.

DWCDS 与 SWCDS 实验参数均设置为 $h_0 = \lceil |Attrs|/2 \rceil$, N=10, $k = \min\{C_{|Attrs|}^{\lceil |Attrs|/2 \rceil}, 10\}$, $n_{\min} = 200$, $\tau_1 = 1.5$, $\tau_2 = 3$. DWCDS 滑动窗口初始值 SW = 1 K, $MinSW$ = 400, $MaxSW$ = 2000, 基本窗口大小 w = 200, SWCDS 窗口大小 sw = 1000. 其中, h_0, N 与 $n_{\min}$ 的确定依据文献 [17–18, 25] 中的实验结论; 各类窗口值的确定均依据大量实验结果得出; τ_1, τ_2 的选择依据文献 [6] 中参数的取值以及大量实验的结论. 测试环境是基于 P4 2.5 GHz, 2 G 内存的 PC 机, 算法实现环境为 Windows XP Professional Visual C++.

3.1 实验数据

实验数据选择了仿真数据流 SEA、HyperPlane、真实数据库 KDDCup99[26]、Yahoo shopping data[27] 以及 UCI 的 LED 数据集. 其中, 仿真数据库类型涵盖了突变式和渐近式两种概念漂移, 在相关数据流文献中被广泛地使用[5, 7, 13−14]. KDDCup99 数据库是入侵检测竞赛数据, 模拟成数据流处理更真实地反映了数据流算法对网络数据的实时动态处理情况, 该数据库在文献 [28] 中作为模拟

表 1 时间复杂度分析

Table 1 Time complexity

算法	增量式建树	漂移检测	分类预测	
			Max	Bayes
DWCDS	O($\|con\| \cdot \|t\| + \|dis\|$)	O(n)		
MSRT	O($\|con\| \cdot \|t\| \cdot \|m_j^A\| + \|dis\|$)	O($\|t\| \cdot \sum_{j=1}^{\|Attrs\|} \|m_j^A\|$)		
CDRDT	O($\|con\| \cdot \|t\| + \|dis\|$)	O($\|S_i\| \cdot \|t\|$)	O($\|classes\|$)	O($\|n_{leaf}\| \cdot \|classes\| \cdot \|Attrs\|$)
CVFDT	O(($\|con\| + \|dis\|$) $\cdot \|t\| \cdot \sum_{j=1}^{\|Attrs\|} \|m_j^A\|$)	O($\|t\| \cdot \|m_j^A\|$)		

流数据抽样变化的数据源. Yahoo shopping data 是通过雅虎 Web 服务接口采集的相关产品的提供者和商家的雅虎购物数据库, 可以更有效地验证 DWCDS 算法较其他几种算法的可行性. 而选择 LED 数据集的目的在于其可以模拟各种噪音环境, 有利于发现算法在不同噪音下的抗噪性能. 另外, 由于 HyperPlane 数据集属性相关性较强, 因此该数据集采用最大类分类方法, 而其他数据集均使用朴素贝叶斯分类方法.

3.2　性能分析

首先对以下实验列表中用到的符号说明如下: DWCDS-b-f/f-b 表示算法 DWCDS 采用从后往前/从前往后的漂移检测方法; 数据集名称表示为: 数据库名称 + 训练数据库大小 + 测试数据库大小 + 数据库类型 (C: 连续型, D: 离散型, CD: 混合型) + 属性维数 + 噪音率.

3.2.1　概念漂移检测

表 2 描述了 DWCDS 算法在数据集 SEA、HyperPlane 和 KDDCup99 上从前往后和从后往前两种漂移检测方法的统计信息. 由实验结果可知, 除在数据集 KDDCup99 上, 两种漂移检测方法的实验效果相当, 而本文采用了后者. 在数据集 KDDCup99 上, 两种检测方法在误报率上存在较明显差异, 其原因分析可能在于数据集类分布的严重不平衡, 有大量类别标签的事例个数小于 200 (此时, 我们认为是噪音), 而噪音数据过多将会影响模型的精度, 从而引起误报.

表 3 描述了 DWCDS 算法在训练数据为 100 k, 噪音率为 10 % 的突变式概念漂移 SEA 数据集, 训练数据为 200 k, 噪音率为 5 % 的渐进式概念漂移 HyperPlane 数据集以及混合属性数据集 KDDCup99 训练数据为 490 k 时的漂移检测结果. 由统计信息可知: 在 SEA 数据集上, DWCDS 共检测 47 次, 数据集中的 3 次概念漂移均被正确检测出, HyperPlane 数据集上, 共检测 99 次, 正确检测出 15 次, 漏检 4 次, 存在少量的漏检与误报, 主要原因是: 为使 DWCDS 算法适于不同类型的概念漂移, 实验选择了平均检测效果较优的检测阈值, 导致出现漏检与误报. 在 KDDCup99 数据集上, DWCDS 共检测 251 次, 正确检测出概念漂移 23 次, 误报 9 次, 分析其出现多次漏检和误报的原因可能在于该数据集类分布极度倾斜, 检测时可能由于窗口中有效信息的不足将导致漏检和误报.

另外, 为说明本文提出的双层窗口机制在概念漂移检测中的有效性, 采用 DWCDS 算法中相同分类器实现了单窗口下的伯努利漂移检测方法 (SWCDS), 并选择了较优的窗口值与 DWCDS 进行了比较, 见表 3. 由表 3 可见, 基于单窗口机制的概念漂移检测方法 SWCDS 的漂移检测能力较差. 以 SEA 数据集为例, SWCDS 算法误报次数为 11 次, 而 DWCDS 算法仅为 6 次. 可见, 采用双层窗口机制进行漂移检测的 DWCDS 算法, 能有效地区分概念漂移与噪音, 降低误报的同时也减少了漂移检测次数 (如: 平均漂移检测次数由 90 次减少为 47 次), 从而使算法更快速有效地适应流数据.

表 2　从前往后与从后往前漂移检测统计信息

Table 2　Statistics of drifting detection from two directions

数据集		检测次数	误报次数	漏报次数
SEA	DWCDS-b-f	47	6	0
	DWCDS-f-b	48	6	0
HyperPlane	DWCDS-b-f	99	6	4
	DWCDS-f-b	100	5	4
KDDCup99	DWCDS-b-f	251	9	13
	DWCDS-f-b	255	17	12

表 3　漂移检测统计信息

Table 3　Statistics of drifting detection

数据集	算法	检测次数	误报次数	漏报次数
SEA-10K-10 % with 3 concept drifts	DWCDS	47	6	0
	SWCDS	90	11	0
HyperPlane-200K-5 % with 19 concept drifts	DWCDS	99	6	4
	SWCDS	190	14	4
KDDCup99-490K with 36 concept drifts	DWCDS	251	9	13
	SWCDS	484	19	13

由以上实验结果可知, 本文提出的基于双层窗口的概念漂移检测机制可以克服一定的噪音数据影响, 快速适应流数据中的概念漂移, 并有效地检测出漂移点. 由于基于单窗口的 CVFDT 以及基于双窗口的 MSRT 算法未给出具体的漂移检测过程, 因此, 本文将在分类错误率和抗噪性方面与其进行比较.

3.2.2 分类错误率

在分类精度方面, 将 DWCDS 算法与其他概念漂移数据流分类算法 MSRT、CDRDT、CVFDT、HT-DDM 和 HT-EDDM 在多数据集上进行了实验对比. 由实验结果可知 (见表 4): 在 LED、HyperPlane、KDDCup99 与 Yahoo shopping data 数据集上, DWCDS 算法效果最优, 分类精度较其他 5 种算法分别提高了 4.3% ~ 51.1%、8.4% ~ 15.3%、0.1% ~ 35.7% 与 2.8% ~ 71.8%. 在 SEA 数据集上, DWCDS 算法的分类精度优于 MSRT 与 CDRDT, 却略低于其他 3 种算法, 这是由数据集的特性所决定的.

3.2.3 抗噪性

本节实验选择了含噪率 5%~20% 的 LED、HyperPlane 与 SEA 数据集 (由于空间问题, 此处测试集仅选用了 SEA-1, 其他类似), 分别考察了 DWCDS 算法在不同噪音环境下的抗噪能力. 由实验结果可知: 在 LED 数据集上, 算法 DWCDS 比算法 MSRT、CDRDT、CVFDT、HT-DDM 和 HT-EDDM 的分类精度分别提高了 3.5% ~ 6.9%、3.3% ~ 10.7%、19.6% ~ 51.1%、24.4% ~ 50.1% 与 19.7% ~ 43.2% (见图 4). 在 HyperPlane 数据集上, 算法 DWCDS 的分类正确率也显著优于其他 5 种算法, 分别提高了 6.9% ~ 10.7%、3.8% ~ 7.7%、12.0% ~ 16.9%、7.9% ~ 13.0% 与 7.8% ~ 12.8% (见图 5). 在 SEA-1 数据集上, 算法 DWCDS 分类正确率明显优于算法 MSRT 与 CDRDT, 分别提高了 0.8% ~ 7.3%、1.0% ~ 7.3%; 而与 CVFDT、HT-DDM 和 HT-EDDM 相比, DWCDS 分类效果略差 (见图 6). 主要原因在于: SEA 数据集仅含三维属性, 因而 DWCDS、MSRT 与 CDRDT 中树高仅为 2, 树高过低导致分类精度下降.

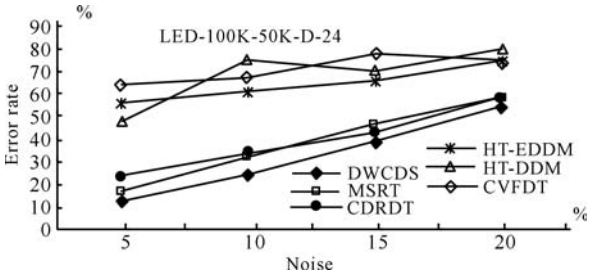


图 4 LED 数据集

Fig. 4 LED database

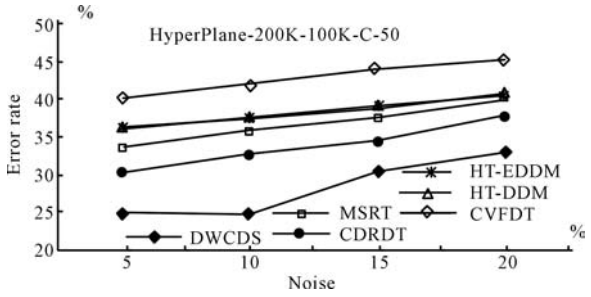


图 5 HyperPlane 数据集

Fig. 5 HyperPlane database

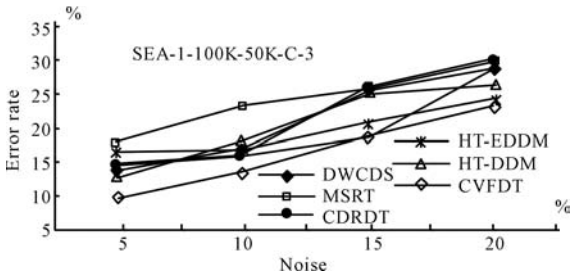


图 6 SEA-1 数据集

Fig. 6 SEA database

表 4 分类错误率比较

Table 4 Error rates of classification (%)

数据集		DWCDS	MSRT	CDRDT	CVFDT	HT-DDM	HTEDDM
LED-100K-50K-D-21-5%		12.51	16.76	23.24	63.59	47.66	55.70
HyperPlane-200K-100K-C-50-5%		24.78	33.21	30.24	40.03	36.13	35.98
SEA-100K-50K-C-3-10%	SEA-1	15.86	23.11	16.30	13.62	17.78	17.12
	SEA-2	14.40	27.70	14.74	19.74	14.26	13.64
	SEA-3	19.78	29.34	21.9	13.72	23.88	23.22
	SEA-4	17.02	27.98	17.88	23.24	14.76	14.34
KDDCup99-490K-310K-CD-41%		9.03	12.69	9.13	25.92	45.57	45.48
Yahoo shopping data-84K-28K-CD-22		1.65	50.31	4.45	15.52	21.85	73.47

表 5 时空性能比较

Table 5 Complexities of time and space

数据集	DWCDS		MSRT		CDRDT		CVFDT	
	时间 $(T+C)$/s	空间 (M)	时间 $(T+C)$/s	空间 (M)	时间 $(T+C)$/s	空间 (M)	时间 $(T+C)$/s	空间 (M)
LED	11+26	120	70+11	8	1+2	8	4+1	6
HyperPlane	17+77	26	164+43	20	3+7	5	83+7	94
SEA-1	1+4	2	10+6	4	1+2	1	9+0	54
KDDCup99	17+625	6	92+504	5	4+12	3	51+13	23
Yahoo shopping data	12+65	34	15+20	30	<1	<1	7+1	29

由此可见, 算法 DWCDS 的抗噪能力显著优于其他 5 种算法.

3.2.4 时空性能

在时空方面, 算法 DWCDS 与其他算法的比较结果见表 5 (T: 训练时间; C: 测试时间). 由实验结果可知: 算法 DWCDS 的训练时间与算法 CDRDT 的训练时间相当, 相差最大不超过 15 s. 均低于算法 MSRT 与 CVFDT 的训练时间, 而测试时间却略高于其他几种算法. 分析原因主要是: 算法 DWCDS 与 CDRDT 采用的是随机决策树模型, 分割属性和划分阈值的选择都是完全随机的, 因此, 每个结点所耗费的时间比较少; 而算法 MSRT 与 CVFDT 在计算划分阈值时采用的是信息增益的方法, 这就增加了每个结点的处理时间. 同时, 算法又对结点建立了多棵替代子树, 这就大大增加了训练模型的时间. 在对测试集进行测试时, 算法 DWCDS 需进行两次投票, 因此, 所需时间略大于其他几种算法.

在空间方面, 对于连续型数据集 HyperPlane、SEA 和混合型数据集 KDDCup99、Yahoo shopping data, 算法 DWCDS 空间消耗与算法 MSRT、CDRDT 相当, 要优于 CVFDT 算法. 在离散型数据集 LED 上, 算法 DWCDS 的效果略差于其他几种算法. 这是由所建决策树的特点决定的. 在建树时, 树高 h_0 会达到 $[n/2]$, 则每棵决策树规模较大, 从而空间消耗较大.

4 结束语

本文基于随机决策树模型提出了一种新的双层窗口机制检测概念漂移的数据流分类算法 DWCDS, 该算法利用滑动窗口与基本窗口中数据分布的变化检测潜在的概念漂移. 实验分析表明, 与单窗口机制检测概念漂移的方法相比, DWCDS 算法具有较优的概念漂移处理能力; 而与已有的概念漂移分类算法相比, 其在分类正确率、抗噪性等方面表现优越. 然而, 该算法需要构造多棵决策树, 在单机环境中存在空间上的劣势. 因此, 它比较适合多 PC 机并行运行环境. 如何在单机状态下提高空间性能, 如何确定分类阶段多决策树的交互投票机制, 以及如何更准确地从噪音中发现概念漂移是下一步的研究目标和方向.

References

1 Golab L, Ozsu M T. Issues in data stream management. *ACM SIGMOD Record*, 2003, **32**(2): 5−14

2 Zhu Y Y, Shasha D. Efficient elastic burst detection in data streams. In: Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data mining. New York, USA: ACM, 2003. 336−345

3 Babcock B, Babu S, Datar M, Motwani R, Widom J. Models and issues in data stream systems. In: Proceedings of the 21st ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems. New York, USA: ACM, 2002. 1−16

4 Castillo G, GamaJ, Medas P. Adaptation to drifting concepts. In: Proceedings of the 11th Portuguese Conference on Artificial Intelligence. Beja, Portugal: Springer, 2003. 279−293

5 Hulten G, Spencer L, Domingos P. Mining time-changing data streams. In: Proceedings of the 7th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. New York, USA: ACM, 2001. 97−106

6 Bifet A, Holmes G, Pfahringer B, Kirkby R, Gavalda R. New ensemble methods for evolving data streams. In: Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. New York, USA: ACM, 2009. 139−148

7 Li P P, Hu X Q, Wu X D. Mining concept-drifting data streams with multiple semi-random decision trees. In: Proceedings of the 4th International Conference on Advanced Data Mining and Applications. Berlin, Germany: Springer-Verlag, 2008. 733−740

8 Li P P, Wu X D, Hu X G, Liang Q H, Gao Y J. A random decision tree ensemble for mining concept drifts from noisy data streams. *Applied Artificial Intelligence*, 2010, **24**(7): 680−710

9 Widyantoro D H. Concept Drift Learning and Its Application to Adaptive Information Filtering [Ph. D. dissertation], Texas A and M University, USA, 2003

10 Gama J, Medas P, Castillo G, Rodrigues P P. Learning with drift detection. In: Proceedings of the 17th Brazilian Symposium on Artificial Intelligence. Maranhao, Brazil: Springer, 2004. 286−295

11 Baena-Garcia M, Campo-Avila J D, Fidalgo R, Bifet A, Gavalda R, Morales-Bueno R. Early drift detection method. In: Proceedings of the 4th International Workshop on Knowledge Discovery from Data Streams. Berlin, Germany: Citeseer, 2006. 77−86

12 Nishida K, Shimada S, Ishikawa S, Yamauchi K. Detecting sudden concept drift with knowledge of human behavior. In: Proceedings of the IEEE International Conference on Systems, Man and Cybernetics. Washington D. C., USA: IEEE, 2008. 3261−3267

13 Kolrer J Z, Marcus M A. Dynamic weighted majority: a new ensemble method for tracking concept drift. In: Proceedings of the 3rd IEEE Conference on Data Mining. Washington D. C., USA: IEEE, 2003. 123−130

14 Kolter J Z, Maloof M A. Using additive expert ensembles to cope with concept drift. In: Proceedings of the 22nd International Conference on Machine Learning. New York, USA: ACM, 2005. 449−456

15 Fan W. Systematic data selection to mine concept-drifting data streams. In: Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. New York, USA: ACM, 2004. 128−137

16 Sun Yue, Mao Guo-Jun, Liu Xu, Liu Chun-Nian. Mining concept drift from data streams based on multi-classifiers. *Acta Automatica Sinica*, 2008, **34**(1): 93−97
(孙岳, 毛国君, 刘旭, 刘椿年. 基于多分类器的数据流中的概念漂移挖掘. 自动化学报, 2008, **34**(1): 93−97)

17 Fan W. StreamMiner: a classifier ensemble-based engine to mine concept-drifting data streams. In: Proceedings of the 30th International Conference on Very Large Data Bases. Toronto, Canada: Morgan Kaufmann, 2004. 1257−1260

18 Wu D Y, Liu Y, Gao G, Mao Z D, Ma W S, He T. An adaptive ensemble classifier for concept drifting stream. In: Proceedings of the IEEE Symposium on Computational Intelligence and Data Mining. Washington D. C., USA: IEEE, 2009. 69−75

19 Fan W, Wang H, Yu P S, Ma S. Is random model better? On its accuracy and efficiency. In: Proceedings of the 3rd IEEE International Conference on Data Mining. Washington D. C., USA: IEEE, 2003. 51−58

20 Hu X G, Li P P, Wu X D, Wu G Q. A semi-random multiple decision-tree algorithm for mining data streams. *Journal of Computer Science and Technology*, 2007, **22**(5): 711−724

21 Fan W. On the optimality of probability estimation by random decision trees. In: Proceedings of the 19th National Conference on Artificial Intelligence. San Jose, USA: AAAI, 2004. 336−341

22 Li P P, Liang Q H, Wu X D, Hu X G. Parameter estimation in semi-random decision tree ensembling on streaming data. In: Proceedings of the 13th Pacific-Asia Conference on Knowledge Discovery and Data Mining. Berlin, Germany: Springer, 2009. 376−388

23 Widmer G, Kubat M. Learning in the presence of concept drift and hidden contexts. *Machine Learning*, 1996, **23**(1): 69−101

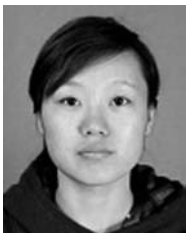
24 Widmer G, Kubat M. Learning flexible concepts from streams of examples: FLORA2. In: Proceedings of the 10th European Conference on Artificial Intelligence. New York, USA: John Wiley and Sons, 1992. 463−467

25 Holmes G, Kirkby R, Pfahringer B. MOA: massive online analysis [Online], available: http://sourceforge.net/projects/moa-datastream/, April 10, 2010

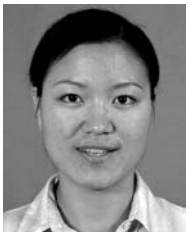
26 KDD cup 1999 data [Online], available: http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html, April 10, 2010

27 Yahoo! Shopping web services [Online], available: http://developer.yahoo.com/everything.html, April 10, 2010

28 Yang Y, Zhu X Q, Wu X D. Combining proactive and reactive predictions for data streams. In: Proceedings of the 11th ACM SIGKDD International Conference on Knowledge Discovery in Data Mining. New York, USA: ACM, 2005. 710−715



朱 群 合肥工业大学计算机与信息学院硕士研究生. 主要研究方向为人工智能与数据挖掘.
E-mail: qunzhu.hfut@gmail.com
(**ZHU Qun** Master student at the School of Computer and Information, Hefei University of Technology. Her research interest covers artificial intelligence and data mining.)



张玉红 合肥工业大学计算机与信息学院讲师. 主要研究方向为人工智能与数据挖掘. 本文通信作者.
E-mail: yuhong.hfut@gmail.com
(**ZHANG Yu-Hong** Lecturer at the School of Computer and Information, Hefei University of Technology. Her research interest covers artificial intelligence and data mining. Corresponding author of this paper.)

胡学钢 合肥工业大学计算机与信息学院教授. 主要研究方向为人工智能与数据挖掘. E-mail: jsjxhuxg@gmail.com
(**HU Xue-Gang** Professor at the School of Computer and Information, Hefei University of Technology. His research interest covers artificial intelligence and data mining.)

李培培 合肥工业大学计算机与信息学院博士研究生. 主要研究方向为人工智能与数据挖掘.
E-mail: peipeili.hfut@gmail.com
(**LI Pei-Pei** Ph. D. candidate at the School of Computer and Information, Hefei University of Technology. Her research interest covers artificial intelligence and data mining.)