实时数据流的算法处理及其应用

张晓龙 著

科学出版社

实时数据流的算法处理及其应用

张晓龙　著

科学出版社
北京

内 容 简 介

本书介绍聚类方法及其优化过程以及实时数据流在企业生产过程中的实际应用。全书共五个部分：第一部分介绍实时数据流和聚类方法的背景，包括实时数据流的特点、进行数据分析的技术以及研究现状。第二部分详细介绍聚类方法中的简单聚类，包括基于衰减窗口与剪枝维度树的数据流聚类、实时数据流动态模式发现与跟踪方法以及相关实验证明等内容。第三部分详细阐述增量聚类技术，包括增量聚类、网格划分策略，以及两个特点不同的增量聚类算法等内容。第四部分介绍聚类算法的一个应用——边界技术检测。第五部分以实时数据流在某钢铁厂的实际应用为案例，剖析实时数据流在实际生产中的应用过程和方法，通过实时数据流分析企业生产过程，最后将聚类方法应用于该案例。

本书不仅可供对实时数据流挖掘领域或实时数据流聚类算法感兴趣的学生和老师阅读和参考，而且适合企业生产人员系统全面地了解和掌握聚类挖掘算法原理，帮助改进生产过程。

图书在版编目(CIP)数据

实时数据流的算法处理及其应用/张晓龙著. —北京：科学出版社，2018.6

ISBN 978-7-03-057906-5

Ⅰ.①实… Ⅱ.①张… Ⅲ.①数据采集 Ⅳ.①TP274

中国版本图书馆 CIP 数据核字(2018)第 127841 号

责任编辑：杜 权 / 责任校对：董艳辉

责任印制：彭 超 / 封面设计：苏 波

科学出版社 出版

北京东黄城根北街 16 号

邮政编码：100717

http://www.sciencep.com

武汉中科兴业印务有限公司印刷

科学出版社发行 各地新华书店经销

*

开本：B5(720×1000)

2018 年 6 月第 一 版 印张：9 1/4

2018 年 6 月第一次印刷 字数：200 000

定价：68.00 元

(如有印装质量问题，我社负责调换)

前　言

实时数据流作为一种新的数据形态和数据处理模型，具有许多传统的数据库或者数据仓库中的数据不具备新的特性。传统的关系型数据库以及数据仓库一般存储的是没有时间概念的、相对静止的数据，而实时数据流具有连续、近似无限、时变、有序及快速流动等特性，且实时数据流中数据点的出现顺序、速率、时刻均不可控制。具体来讲具有数据量大、时序性、快速变化、潜在无限等特点。

以上所描述的数据流的特点，决定了传统的数据挖掘技术无法直接应用于实时数据流，必须研究和开发出适合数据流模型的算法，适用于实时数据流的数据挖掘给数据挖掘带来了全新的研究内容，引起了数据挖掘和数据库领域学者极大的兴趣，他们在实时数据流查询、实时数据流挖掘等领域展开了广泛的研究。本书主要从实时数据流挖掘的角度分析和讨论数据流聚类分析、数据流分类、数据流频繁模式挖掘、数据流的关联规则分析以及实时数据流在企业生产中的应用等。

本书共八章，主要内容如下：

第 1 章介绍了实时数据流的算法处理及应用的研究背景以及最新进展，分析了目前实时数据流算法处理及应用方面的若干研究方向与实例。

第 2 章简要介绍了数据流聚类技术，对基于衰减窗口与剪枝密度维度树的数据流聚类进行了详细的介绍与分析。通过基于人工数据流和真实数据流的实验表明，PDStream 算法可以识别数据流中具有任意形状的聚类簇，在线处理速度快，系统消耗小，同时具有较好的计算精度和效率。

第 3 章介绍了实时数据流中的模式，提出了动态模式发现与跟踪方法，并对该方法进行了实验分析。针对现有模式演化分析算法无法精确定位和详细描述模式在实时数据流中的变化，动态模式发现与跟踪方法基于模拟数据集和真实数据集的实验表明，算法可以有效地发现数据流中的动态模式，并能过通过模式匹配对比分析跟踪模式的演化过程。

第 4 章简要介绍了增量式聚类方法与网格划分策略，对于提高数据流聚类具有较大的帮助。为后续第 5 章和第 6 章提出的算法做了必要的理论铺垫。

第 5 章提出了基于网格和密度维度树的增量聚类算法 IGDStream。对算法思想做了详细的描述，并给出了实验结果与分析。IGDStream 算法是基于 PDStream

算法，提出了一种增量式的聚类。由于是对新增数据对象批次进行处理，IGDStream 算法继承了 PDStream 算法能处理任意形状簇、对噪声数据的处理能力良好等优点，同时，通过逐步动态聚类，能实现对大型数据集和实时数据流的聚类分析，有很好的可扩展性和伸缩性。

第 6 章详细介绍了提出的基于密度维度树的增量式网格聚类算法 IGDDT。IGDDT 算法是在 PDStream 算法基础上提出了一种新的基于密度维度树的增量式网格聚类算法。该算法保持了传统网格聚类高效的优点，采用网格细分的方法对聚类边缘进行详细描述，以提高聚类质量。在真实数据集与仿真数据上，IGDDT 算法可以在数据空间中的数据点不断增加时，在已得到的聚类模式的基础上，增量地对其进行调整实现增量聚类，从整体上提高了聚类的有效性。

第 7 章研究了采用衰减窗口技术和基于网格的方法实现实时数据流的聚类及其边界检测算法 GDBOUND，该算法通过计算每个网格的密度以及网格之间的相似程度，决定其是否归属于某个聚类模式，并对聚类后的结果扫描，从中发现其边界，避免对整个数据空间重新进行处理，以提高系统的性能。

第 8 章研究了实时数据流及聚类方法在工业生产中的应用。以钢铁企业质量监控为例，介绍了实时数据流的处理平台，重点分析了实时数据流以及聚类技术在钢铁质量监控中的应用手段，并对应用结果进行了分析。

本书的出版得到国家自然科学基金的经费资助(60975031)，在此表示感谢。本书的编写和出版过程中还得到了科学出版社的支持和帮助，在此一并表示衷心的感谢。

限于著者水平，书中的疏漏之处在所难免，恳请广大读者和专家指正。

张晓龙

2018.4.18

目　录

第1章 实时数据流和聚类方法的背景

数据流(data stream)最初是通信领域使用的概念,代表传输中所使用的信息的数字编码信号序列。1998 年 Henzinger 提出数据流为“只能以事先规定好的顺序被读取一次的数据的一个序列”。传统的数据库系统(database system,DBS)、数据仓库(data warehouse)以及数据集市(data mart)通常用来存储没有时间概念的相对静止的数据,基于这种静态数据集合的数据挖掘与知识发现已被研究多年,相关的技术也日趋成熟[1]。

实时数据流是一种越来越重要的数据模型,针对实时数据流的数据挖掘和知识发现有很多限制条件。本章首先介绍了实时数据流的基本概念及其相对于传统静态数据所独有的特点,其次逐一介绍了在这种数据模型上进行聚类、分类、频繁模式挖掘、关联规则分析等知识发现技术的主要内容及现状,最后,简要地介绍了国外实时数据流挖掘技术的应用实例。

1.1 实时数据流

1.1.1 实时数据流的定义

近年来,随着数据采集技术和网络技术的快速发展,是移动通信设备和无线传感网络得到了深入研究和广泛应用,在许多应用领域中,信息以数据序列的形式出现[2]:电信公司每天产生的实时电话记录流、互联网上 Web 用户的点击流、网络监测中的数据包、工厂自动化控制中产生的控制信息流、各类传感器网络中的通信数

据流、金融领域的证券交易数据流、零售业务中的销售数据流、航天卫星向地面接收站发回的数据流等形成了一种与传统数据库中静态数据不同的数据形态，这种新的带有时间序列的数据形态被称为实时数据流(real time data stream)，广泛出现于电信、互联网、金融、航天、工业等领域。这些实时数据流一般具有数据量无穷、数据有严格的先后顺序、数据概念随时间快速变化、富含噪声等显著特点[3-4]。

实时数据流是一个有序的数据点序列：$\overline{X}_1,\overline{X}_2,\cdots,\overline{X}_k,\cdots$，整个数据流对应于一个由小到大的时间序列 $T_1,T_2,\cdots,T_k,\cdots$，表示数据点$\overline{X}_i$在 T_i 时刻到达，也就是规定了当 $T_i<T_j$ 时，对应的数据点$\overline{X}_i$比数据点$\overline{X}_j$先到达。每一个数据点$\overline{X}_i$都是一个 d 维向量，记作$\overline{X}_i=(x_i^1,x_i^2,\cdots,x_i^d)$，其中，$x_i^j$ $(1\leqslant j\leqslant d)$分别代表数据点$\overline{X}_i$的 d 个属性值。

实时数据流作为一种新的数据形态和数据处理模型，具有许多传统的数据库或者数据仓库中的数据不具备新的特性。传统的关系型数据库以及数据仓库存储的一般是没有时间概念的、相对静止的数据，而实时数据流具有连续、近似无限、时变、有序及快速流动等特性，且实时数据流中数据点的出现顺序、速率、时刻均不可控制[5]。具体来讲具有如下特点：

(1) 数据量巨大(massive)。实时数据流一般具有海量的数据，例如，我国2010年发射的“嫦娥二号”探月卫星在正常的绕月探测过程中，向地面接收站发送的月球图像等数据流为 6 Mbit/s，一年的数据量可达 28 TB。

(2) 时序性(temporally ordered)。在数据流模型提出之初，就规定了当$T_i<T_j$ 时，数据点$\overline{X}_i$比数据点$\overline{X}_j$先到达，数据点之间存在严格的先后顺序。

(3) 快速变化(vary rapidly)。由于数据流的单向流动性，当前时间段内的数据流所表达的概念与下一时间段的数据流所表达的概念可能会有很大变化，这种现象称为数据流的概念漂移(concept drift)，这样的数据流也称为进化数据流(evolving data stream)。

(4) 潜在无限(potentially infinite)。从理论上讲，数据流永远没有终止的时刻，具有无限性，所以说它是潜在无限的。

(5) 高维性(high dimensional)。现实世界中的真实数据流往往都具有高维的特性，且数据流一般同时具有连续属性和离散属性[6]。

(6) 存储限制(memory restrict)。数据流的数据量十分庞大，具有潜在无限的特性，不可能像传统的数据仓库一样将数据全部存储起来，再进行挖掘，挖掘算法在运行的空间上是受到限制的。

(7) 时间限制(time restrict)。多数实时数据流挖掘系统要求具备很短的响应

时间，能够随时(anytime)满足用户请求，所以数据流挖掘是一个连续在线的过程，要求处理的速度快。

(8) 单边扫描或者有限次扫描(once scan)。算法在对数据流进行处理时，只允许按照数据点流入的先后顺序逐个处理，回头去重新扫描前面的数据点是不允许的。

以上所描述的数据流的特点，决定了传统的数据挖掘技术无法直接应用于实时数据流，必须研究和开发出适合数据流模型的算法，适用于实时数据流的数据挖掘给数据挖掘带来了全新的研究内容，引起了数据挖掘和数据库领域学者极大的兴趣，他们在实时数据流查询、实时数据流挖掘等领域展开了广泛的研究。本书主要从实时数据流挖掘的角度分析和讨论数据流聚类分析、数据流分类、数据流频繁模式挖掘、数据流的关联规则分析等。

1.1.2　实时数据流的研究现状

实时数据流以上的这些新特性，为数据流相关领域的研究以及实际工程应用带来了困难。一方面，实际应用领域需要对海量实时数据流进行在线的、持续的、快速的处理，这些特点和要求已经远远超出了传统数据库系统的数据处理能力；另一方面，实时数据流中隐藏着丰富的知识和信息，但是这种新的数据形态给基于实时数据流的数据挖掘和知识发现带来了新的挑战，及时有效地从实时数据流中挖掘出有用的知识用于指导生产实践以及决策支持，具有重要的研究价值和实践意义。目前，基于实时数据流模型的数据挖掘技术的研究已成为研究热点，引起了国内外研究者的广泛关注。

基于实时数据流的数据挖掘的研究最早可以追溯至1999年，Henzinger等在其论文“*Computing on Data Streams*”中就已将数据流作为一种新的数据处理模型提出来[7]，此后，数据流开始作为一个新的研究方向出现在数据挖掘与数据库领域的几大顶级国际会议中，如VLDB、SIGMOD、SIGKDD、ICDE、ICDM等会议每年都有多篇关于数据流处理的文章，由于数据流这一概念刚出现不久，在这一时期对数据流的研究尚未得到广泛的关注。

在此后的一段时间内，实时数据流作为一种新的数据形态慢慢地引起了数据库和数据挖掘领域的研究人员极大的兴趣，同时也开始得到了广泛的关注。这一时期比较显著的阶段性研究成果就是有学者根据实时数据流的新特性提出了一种与传统的数据库管理系统(Data Base Management System，DBMS)相对应的数据

流管理系统(data stream management system,DSMS)[8]。如图 1.1 所示,在传统的 DBMS 中,数据库或数据仓库中存储的是有固定的关系结构、长时间静态并可随机访问的数据,由用户(程序)通过 DML(data manipulation language)语言提交查询请求,返回精确的查询结果;而在 DSMS 中,如图 1.2 所示,存储的是半结构化的、与时间有关并只能按顺序访问的海量数据流,数据流处理系统实时地维护一个数据流的摘要信息,当用户(程序)提出查询请求时,系统快速地返回一个近似结果。

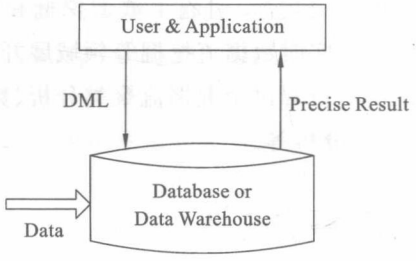

图 1.1 DBMS 体系结构

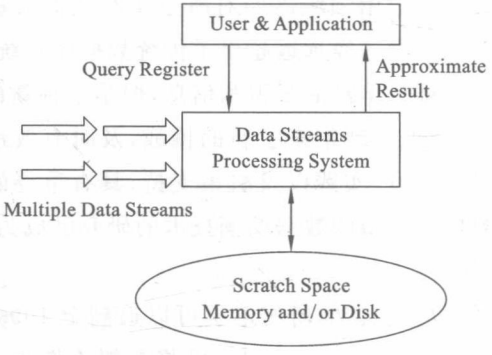

图 1.2 DSMS 体系结构

DSMS 研究的内容上大致可分为两个方面:①DSMS 的设计与构建。建立 DSMS 方面的研究主要集中在数据流查询。已有多个研究机构进行了关于 DSMS 的研究,并构建出一些原型系统,如 STREAM,TelegraphCQ[9] 及 Aurora[10] 等。②数据流的数据挖掘。数据流挖掘方面的研究主要包括多数据流挖掘和单数据流挖掘,单数据流与多数据流的区别在于挖掘的对象是一条单个的数据流或者同时处理多条数据流。目前已有学者提出了一些数据流挖掘算法,并设计和开发出数据流挖掘原型系统。如美国伊利诺伊大学厄巴纳-尚佩恩分校(University of

Illinois at Urbana-Champaign，UIUC）的 MAIDS（mining alarming incidents from data streams）[11]就是一个集查询、聚类、分类、频繁项集挖掘，以及处理结果可视化五大功能为一体的数据流挖掘系统。本章主要研究和探讨的是实时数据流挖掘。

在国外，实时数据流挖掘方面有两个比较有影响的研究小组，一个是斯坦福大学的 R. Motwani 教授领导的研究小组，他们主要侧重于数据流管理、数据流的连续查询以及数据流的聚类研究，提出了不同于 DBMS 的 DSMS 系统，其研究得到了美国国家自然科学基金的资助；另一个是 UIUC 的 C. Aggarwal 教授和韩家炜教授领导的研究小组，他们的研究主要侧重于数据流分析，在数据流的在线分析、聚类、分类、频繁项集挖掘以及数据流可视化等方面做了相关的研究工作，他们的研究也得到了美国军方和美国国家自然科学基金的资助。国内有关这一领域的研究起步比较晚，从 2004 年开始，有关数据流挖掘研究的文章才开始出现，其中又以华东师范大学的周傲英教授、中山大学的印鉴教授、哈尔滨工业大学的李建中教授及东南大学的孙志挥教授等各自领导的研究小组的研究比较活跃。

1.2　实时数据流聚类

聚类（Clustering）就是按照一定的要求和规律对事物进行区分和分类的过程。在这一过程中没有任何关于类别的先验知识，也没有教师的指导，仅靠事物间的相似性作为类属划分的准则，属于无监督学习的范畴。对聚类问题有如下定义：对于一个给定的数据点集合，把相似的数据点划分在一起，形成一个或多个组，相似性的度量采用某种距离函数，使得聚类簇内的数据点具有较高的相似度，簇间的数据点具有较低的相似度。

在数据挖掘领域中已经存在许多用于聚类静态数据的聚类算法，诸如基于划分的方法、基于层次的方法、基于密度的方法、基于网格的方法等，由于数据流具有与常规的静态数据不同的特点，数据流的聚类分析需要研究适合数据流模型的、内存受限的、处理时间有限的单遍数据流扫描的聚类算法。数据流对在其上进行聚类的算法提出了如下要求[12]：

（1）聚类簇数事先未知。算法不可能预先知道数据流将会被聚类成为几个聚类簇，不但如此，在数据流环境下，随着数据的不断流入，聚类簇的数目在动态地改变。

（2）聚类形状任意。这对数据流聚类来说至关重要，例如，在网络监控环境

中，连接的分布一般是不规则的，这也表明，传统的基于欧几里得空间距离的相似度准则多数产生球形的聚类结构，对数据流的任意形状聚类簇效果不好。

(3) 对孤立点的分析能力。由于数据流存在波动和进化(evolving)，在当前时刻被认为是孤立点的数据点，有可能在后续一系列数据点到达之后变成一个新的聚类簇，聚类算法必须能够实时地监控数据流的变化情况。

实时数据流聚类的形式化描述如下：设 $DS=\{\overline{X}_1,\overline{X}_2,\cdots,\overline{X}_k,\cdots\}$ 表示数据流，各个数据点到达的时间戳分别是 $T_1,T_2,\cdots,T_k,\cdots$，每个数据点 $\overline{X}_i=(x_i^1,x_i^2,\cdots,x_i^d)$，其中 x_i^j $(1\leqslant j\leqslant d)$ 是 $\overline{X}_i$ 在第 j 个特征上的赋值，数据流聚类分析就是把数据流 DS 中当前时间窗口(实时数据流挖掘中常用“时间窗口”这一概念，其实质就是某个预先指定长度的特定时间段) $T_i=[T_h,T_n]$ 内的 $n-h+1$ 个数据点，按照该数据点与已有聚类簇间的远近关系以及约束条件，逐个加入到由前面 $h-1$ 个数据点所形成的聚类结构 CF_{i-1} 中，形成新的聚类结构 CF_i。

在 CF_i 中，已经流入的 n 个数据点被划分成 k 个不相交的模式子集 $S=S_1,S_2,\cdots,S_k$，满足如下条件：

$$\begin{cases} S_1\cup S_2\cup\cdots\cup S_k=S \\ S_i\cap S_j=\varnothing \qquad\qquad (1\leqslant i,j\leqslant k,i\neq j) \end{cases} \tag{1.1}$$

数据点 $\overline{X}_j$ $(1\leqslant j\leqslant n)$ 对子集 S_i $(1\leqslant i\leqslant k)$ 的隶属关系可用隶属函数表示为[13]

$$\omega_{S_i}(\overline{X}_j)=\omega_{ij}=\begin{cases} 1, & \overline{X}_j\in S_i \\ 0, & \overline{X}_j\notin S_i \end{cases} \tag{1.2}$$

$$M_{pk}=\left\{\omega_{ik}\left|\omega_{ij}\in\{0,1\},\sum_{i=1}^{k}\omega_{ij}=1,\forall j;0<\sum_{j=1}^{n}\omega_{ij}<n,\forall i\right.\right\} \tag{1.3}$$

其中，隶属函数必须满足条件 $\omega_{ij}\in M_{pk}$，也就是要求每一个数据点能且只能隶属于某一个聚类簇，同时要求已经存在的聚类簇是非空的。对于后续的数据点，采用相同的方式增量地聚类到前面已经存在的聚类结构中。现有实时数据流聚类技术及相关算法的详细情况将在后面章节进行介绍和比较分析。

1.3 实时数据流分类

数据分类主要分为两个过程，即分类器或模型的构造(其中，模型是基于训练数据集标记了类别的元组构造)和分类器或模型的使用(其中，模型用于预测新数据集中元组的类标号)。在传统的环境下，训练数据一般驻留在一个相对静止的数

据库中，许多分类方法允许扫描训练数据很多次，所以说，分类器构造的第一步就是典型的脱机批处理过程。然而，在实时数据流环境下没有脱机阶段，数据流的特点也决定了存储和多次扫描是不可行的，由此可见，传统的分类方法完全不适合实时数据流环境。

以传统的决策树分类技术为例，它们一般都遵循自顶向下的递归策略，只是用于选择最佳分裂属性的统计度量不同。决策树由内部节点、分支节点及叶子节点组成，属性选择度量用来选择当前节点的分裂属性，该属性是按类最好的区分训练元组的属性，分类属性的每个可能值都产生一个分支，训练元组也据此来划分，并在每个分支递归地重复这一过程。但是在数据流环境下，既不可能收集完整的数据集，也不可能实现重复扫描数据，因此，该类方法需要重新考虑[14-16]，此外，前文提到的概念漂移也使得建立起来的分类器随时间而改变。

为了解决这些问题而实现数据流的分类，目前已有学者进行了相关的研究，并提出了一些可行的解决办法[17-23]。主要的算法分为如下几类：Hoeffding 树算法[17]、快速决策树[18](very fast decision tree，VFDT)、概念自适应快速决策树[19](concept-adapting very fast decision Tree，CVFDT)、分类器系综[20](classifier ensimble)(使用投票方法考虑多个分类器)。

1.3.1　Hoeffding 树算法

该算法的主要思想是“小样本通常足以选择最佳分裂属性”，其核心理论为概率论中的切尔诺夫不等式(Hoeffding 界)，假设变量 r 范围为 R，观察 n 个样本后，样本观测平均值为 $\bar{r}$，则样本真值以 $1-\delta$ 的概率保证至少为 $\bar{r}-\varepsilon$，其中，

$$\varepsilon=\sqrt{\frac{R^2\ln(1/\delta)}{2n}} \tag{1.4}$$

Hoeffding 树算法就是以这样的高概率确定在一个节点选择分裂属性时需要的样本的最小数量 n，该属性将与使用无限样本得到的属性一样，接下来的工作就与传统决策树的构建类似。Hoeffding 树算法一个非常重要的特性是它和样本分布是独立的，对于数据流来说，这正是所希望实现的，因为不太可能看到信息增益的概率分布，或者使用哪种其他的属性选择度量。此外，Hoeffding 树的另一个优点在于不会对同一数据进行多次扫描，这一点对于数据流环境是很重要的，因为数据流经常会变得太大，无法存储，而且，Hoeffding 树算法是增量的，即随着新的样例不断流入，可以增量地将新样例并入树中，即使树正在构造，也可以用它对数据分类，树会持续增长，并且随着更多训练数据流入变得更加准确。

但是，由于树型结构本身的复杂性，算法会在近似等价分裂质量的属性花费大量的时间，在内存的消耗上也没有明显的优势，而且，Hoeffding 树算法不能处理概念漂移，因为 Hoeffding 树中的节点一旦创建，就无法改变。

1.3.2 快速决策树

快速决策树算法是对 Hoeffding 树算法的一种改进与扩充，以提高算法处理速度和内存利用率。VFDT 算法在选择属性时更主动地打破平局，在训练完许多样例后计算评估函数，一旦内存消耗太大就解除最没有希望的叶子节点的活跃状态，删除分类效果不好的分裂属性。VFDT 算法对于数据流分类有较好的效果，并且在速度和准确率上也比传统的方法好很多，但是 VFDT 算法的一个问题是仍然不能处理实时数据流中的概念漂移问题。

1.3.3 概念自适应快速决策树

为了能够处理实时数据流中的概念漂移，需要一种新的途径来及时识别数据流中与当前概念不再一致的元素，滑动窗口就是常用的方法之一，当新的样例到来时，插入到窗口起始处，同时从窗口尾部删除相应数量的样例，在滑动窗口中重复使用传统的分类方法。但是窗口大小 ω 是一个敏感参数，因为如果 ω 太大，模型就不能准确地描述概念的漂移，如果 ω 太小，又会导致没有足够的样例来构建模型。此外，不断地在滑动窗口中构造新的分类器模型会使系统开销增大。

概念自适应快速决策树就是在 VFDT 算法的基础上做了改进，以适应数据流的概念漂移。CVFDT 算法就是基于滑动窗口的，CVFDT 算法通过增加与新样例相关联的计数，减少与旧样例相关联的计数来更新统计量，因此，如果数据流中存在概念漂移，一些节点可能不再满足 Hoeffding 界，这时将产生一棵替代的子树，子树的根具有新的最佳分裂属性，随着新的样例流入，替代的子树持续生长，但是暂时不用于分类。一旦替代的子树变得比已有的子树更准确，旧的子树将被取代。相关的实验研究表明，CVFDT 在随时间变化的数据流上能获得比 VFDT 更好的准确度。而且，VFDT 树中累积了太多过时的样例，这一点决定了 CVFDT 树要比 VFDT 树小很多。