Journal of Computer Applications
计算机应用,2012,32(2):299 -303
ISSN 1001-9081
CODEN JYIIDU
2012-02-01
http://www.joca.cn

文章编号:1001 -9081(2012)02 -0299 -05
doi:10.3724/SP.J.1087.2012.00299

数据流变化检测研究综述

宋擒豹,杜　磊*
(西安交通大学 电子与信息工程学院,西安 710049)
(*通信作者电子邮箱 dulei.323@stu.xjtu.edu.cn)

摘　要:数据流是一种动态数据,它在某种因素的驱动下可能会随时间发生变化,而这种变化往往隐含着现实世界的某种事件。如何及时、准确地发现数据流中的变化已成为数据流挖掘的一个研究热点,并且在实际中有非常广泛的应用。描述了数据流变化及变化检测的核心任务,归纳了变化检测的通用框架,分析评价了目前已知的数据流变化检测方法及其技术特点,最后展望了数据流变化检测技术的发展方向。

关键词:数据挖掘;数据流;变化检测;变化点
中图分类号:TP311.13　　**文献标志码**:A

Survey on change detection over data stream

SONG Qin-bao, DU Lei*
(*School of Electronic and Information Engineering, Xi'an Jiaotong University, Xi'an Shaanxi* 710049, *China*)

Abstract: Data stream is a type of dynamic data, which is driven by some hidden contexts and may change with time. Generally speaking, the change implies some event in the real world. To detect change over data stream timely and accurately has a quite wide range of practical applications, and has been one of the hot topics in data stream mining. In this paper, a comprehensive survey on data stream change detection was presented, and the key task of change detection was introduced. An inductive unifying framework of change detection process was also given. Besides, a variety of already existing change detection approaches and their features were reviewed and evaluated in detail. Finally, the research outlook of data stream change detection was discussed.

Key words: data mining; data stream; change detection; change point

0　引言

传统的数据挖掘技术基于静态的数据库,也就是说数据集是静态的、可多次访问的。然而,在实际生活和应用中,数据却常常是连续产生并不断增加的,而且对数据无法任意地访问。我们将这种随时间连续产生的数据序列称为数据流。早在20世纪中期,研究人员已经开始了对序列数据(可认为是数据流的早期模型)的研究[1],但直到1998年,数据流才作为一种数据模型被提出来[2]。此后,数据流便得到了广泛的关注,并在过去20年得到了快速的发展。

在数据流环境中,人们感兴趣的往往是当前数据集的模式以及这种模式跟以往的模式的区别,而对很久以前存储的数据不再重视。比如在一个实时控制系统中,管理员通常只关心当前的系统数据及其是否发生变化,以便对系统所处的状态有准确的把握;在证券交易市场,人们往往更关心股票价格的变化,对很久以前的股票价格甚至当前的股票价格则不太关心。这类实时应用迫切需要能及时准确追踪数据流的方法,即能快速准确发现数据流变化的方法。

由于数据流随时间不断产生,并且其特征随时间的流逝不断变化,因此变化(change)或变化点(change point)是数据流不可避免的内在属性。从直觉上来说,只要数据流发生了比较明显的变化,人们一定会从视觉上观察到这种变化并及时做出反映[3]。然而,事实往往并不如此。实际情况是,人们通常对变化是"视而不见(change blindness)"[4]的。即使能从直觉上察觉这种变化,人们也没有精力时刻注视着数据流来观察它是否发生了变化。

由于高速实时系统对数据流变化检测算法的需求,以及人工观察法的困境,借助计算机来检测数据流中的变化或变化点有着非常重要的现实意义。目前,变化检测已成为数据流挖掘的一个热点问题。

由于数据流通常来自不同的场景,如文本数据流、图像数据流、音频数据流等;而且在不同的应用中,人们关心的对象往往不同,因此,数据流上的变化有多种具体的形式。根据抽象层次的不同,数据流变化可以分为:1)数据流中数据分布的变化,如均值(mean/average)、方差(variance)等统计特征的变化;2)概念迁移,即从数据流原始数据中抽象出来的概念的变化,是高层次的变化。通常涉及概念迁移的数据为包含名词型属性的分类型数据。显然,这两种变化代表着数据流不同层次的变化。本文对数据流上数据分布的变化进行分析,评述了数据流变化检测的研究工作,并指出了未来的研究方向。本文不涉及概念迁移的研究,有关其详细的介绍可参考文献[5]。

1　基本概念和术语

1.1　数据流

1.1.1　数据流的定义

顾名思义,数据流就是随时间连续不断产生,并有着严格先后次序,通常趋于无界的数据集合[6-7]。在某一时刻,我们无法访问当前时刻之后的数据元素,也无法任意访问很久以前的历史数据。总之,数据流中的元素无法一次获得,这使得

收稿日期:2011-07-25;**修回日期**:2011-09-01。　**基金项目**:国家自然科学基金资助项目(61070006)。
作者简介:宋擒豹(1966 -),男,陕西华县人,教授,博士,主要研究方向:数据挖掘、软件缺陷预测、软件成本预测;　杜磊(1985 -),男,陕西高陵人,博士研究生,主要研究方向:数据流挖掘、时间序列分析。

数据流明显不同于传统的静态数据集。

为了便于叙述,我们将数据流表示为 $DS = \{x_1, x_2, \cdots, x_i, \cdots\}$,其中 x_i 为时刻 i 对应的数据元素,i 随时间递增且 $i \to \infty$,其中 x_i 为 1 维或者多维数值型数据。

1.1.2 数据流的特征

与传统的静态数据相比,数据流的特征如下:

1)动态性。数据流中的元素实时地、有序地、连续地到达,整个数据集呈现出动态特征,意味着所有的数据元素不是一次到达,且数据流是潜在变化的。

2)无限性或近似无限性。动态性导致数据流常常是无界的或者近似无界的($i \to \infty$)。随着时间的流逝,数据元素不断产生直至数据流终结。

3)单次扫描。由于数据流的动态性,使得对数据流的处理无法像对传统数据库那样多次访问多次处理,通常仅能访问一次或者很少的几次。

1.2 滑动窗口

滑动窗口是数据流挖掘算法常采用的技术,它是数据流的一段快照,有固定大小和可变大小之分。滑动窗口可表示为

$$SW = \{w_1, w_2, \cdots, w_N\}$$

其中:$w_1, w_2, \cdots, w_N$ 为窗口内部元素;N 为滑动窗口的大小,通常根据经验确定。

滑动窗口有时也称为子序列(Subsequence 或 Substream)。根据其大小可变与否,可以分为两类:

1)界标窗口。SW 的左边界为某一个固定点并保持不变,右边界随着数据元素的到来不断移动。该窗口的大小是单调增加的。

2)滑动窗口。SW 的左右边界均沿着时间轴滑动,窗口大小不变。SW 每滑动一步,删除其中最早的元素,并插入最新的数据流元素。

1.3 数据流变化

1.3.1 数据流变化的定义

前文提到,数据流变化的定义有多种具体形式,因此根据数据流来源的不同,变化的定义也有所不同。

文献[8]在单个滑动窗口上对数据流变化作了定义:给定数据流 DS 及其上一个滑动窗口 $SW = \{w_1, w_2, \cdots, w_i, w_{i+1}, \cdots, w_N\}$;$SW$ 每滑动一步,就被切分为两个子窗口 $sw_1 = \{w_1, w_2, \cdots, w_i\}$ 和 $sw_2 = \{w_{i+1}, w_{i+2}, \cdots, w_N\}$,并计算子窗口内数据元素两两之间距离的平均值,即 $D(sw_1, sw_2) = \dfrac{\sum_{j=1}^{N_2}\sum_{k=1}^{N_2} \| w_j - w_k \|}{N_1 \times N_2}$,并将 $D(sw_1, sw_2)$ 的最大值作为当前窗口的异常值;以该异常值是否超过某个阈值来判断数据流是否发生变化。如果该异常值小于阈值,则认为数据流没有发生变化;否则,数据流发生变化,并将该最大值对应的切分点称为变化点。

文献[9]对变化的定义与文献[8]相似,但该定义基于可变大小的滑动窗口。区别在于:文献[9]对 $D(sw_1, sw_2)$ 的计算是基于两个子窗口的均值,即 $D(sw_1, sw_2) = \| \mu_{sw_1} - \mu_{sw_2} \|$,因此不必计算数据元素两两之间的距离。

文献[10]采用两个等长滑动窗口来定义数据流的变化。定义如下:设 SW_1 和 SW_2 为 DS 上两个大小相同的滑动窗口,P_1 和 P_2 分别为它们对应的分布函数;当 $D(P_1, P_2)$(P_1 和 P_2 的差异或距离)大于某个阈值时,认为数据流发生了变化。

文献[11]虽然也是用两个滑动窗口来定义数据流变化,但却与文献[10]的定义明显不同。文献[11]的定义要求滑动窗口 SW_1 保持不变,而 SW_2 为界标窗口,最后根据 $D(P_1, P_2)$ 的差异判断数据流是否发生了变化。

文献[12]则在多个滑动窗口对数据流变化作了定义:首先为数据流设定了一个聚合函数 f(如求和 sum)以及多个滑动窗口 $SW_i (i = 1, 2, \cdots, m)$,并为每一个滑动窗口预先设定了对应的阈值 $f(SW_i)$。当聚合函数的值大于对应阈值时,认为变化发生。

文献[13]定义了网络流量的变化,当两个滑动窗口中的 IP 包数量的差异达到某个阈值时,认为是一种变化。

文献[14]则是从数据流速率的角度来定义变化的。如果相邻两个数据元素到来的时间间隔(数据流速率)的密度大于某个阈值时,认为数据流发生了变化。

另外还有基于金融时间序列变化的定义[15]等,在此不再一一列举。

1.3.2 数据流变化的分类

显而易见,数据流的变化可以分为两类:1)数据流突变(sudden change/abrupt change/burst),这种变化持续时间短,变化明显,数据流在很短的时间段内从一种状态变化到另一种状态;2)数据流渐变(gradual change/slow change),这种变化需要经过很长时间的积累,短期内几乎无法察觉这种变化,数据流状态变化不明显。

1.3.3 变化与位移的区别

在数据流变化中,常常将“变化”与“位移(motion)”混淆,因为二者都表示某种特征的改变。

通常,变化表示数据流某种特性的改变,即这种改变是结构上的,表示数据流本身从一种状态转化到了另一种状态。而位移则通常表示空间位置的变化,并不涉及结构上的变化。

2 数据流变化检测方法

2.1 数据流变化检测的处理流程

尽管数据流的定义不尽相同,但仍然可以用一个一般性的框架来描述数据流检测的过程。要检测数据流中的变化,需要经过以下步骤:

1)定义一个目标函数 F 用来描述数据流某种特征的变化,如均值、方差或求和函数等。

2)设定一个时间区间来衡量 F 变化的粒度,即滑动窗口的大小。窗口越小,对变化越敏感。

3)选择度量目标函数差异的函数,我们称为差异度量函数 D,如度量距离的欧氏距离,度量分布差异的 KL 距离(Kullback-Leibler divergence)[16]等。

4)选择判定策略,即根据差异函数 D 的评估值判断变化是否发生。通常有阈值比较法和统计假设检验法。

根据上面的步骤,我们将数据流变化检测抽象为:当 $D(F(SW_1), F(SW_2)) > \xi$ 时,认为数据流发生了变化,其中 ξ 为阈值,由经验知识获得,或者从历史数据中得到;SW_1 和 SW_2 为两个滑动窗口或同一个滑动窗口的两个子窗口;$D(F(SW_1), F(SW_2))$ 表示 SW_1 和 SW_2 的差异,可以是原始数据元素之间的差异,也可以是分布函数的差异。显而易见,该差异值越大,说明变化发生的可能性越大;反之则说明变化发生的可能性越小。

通过选择不同的目标函数 F 便可以检测数据流不同特征的变化。设 F 函数表示求均值函数,那么该变化则定义为数据流在滑动窗口 SW_1 和 SW_2 上的均值变化;如果 F 为求和函数,那么此时数据流变化指数据流在滑动窗口上 SW_1 和 SW_2 的和

的变化;如果 F 为频繁项集,则变化表示数据流中频繁项集的变化。即使在没有明确知道 F 的情况下,也可以通过计算 SW_1 和 SW_2 内元素两两之间的距离(文献[8]的定义)来表示数据流的变化。

目前已经有了许多检测数据流变化的方法,根据它们使用的滑动窗口的个数,我们将变化检测方法分为基于单一滑动窗口的检测方法、基于两个滑动窗口的检测方法和其他检测方法。

2.2　基于单一的滑动窗口的检测方法

早在 1954 年,Page[1] 就提出了 CUSUM 算法来检测工业过程控制产生的时间序列是否发生了突变。算法流程如下:在数据流上维持一个滑动窗口 SW;然后求出 SW 的平均值并计算每个数据元素与该平均值的差;最后累加求和这些差异值;当该差异值超过了给定的阈值,认为数据流发生了变化。CUSUM 仅能检测均值上升的变化,无法处理均值下降的变化。Barnard[17] 改进了 CUSUM 算法并提出了 V-mask 方法,解决了时间序列中均值上升和下降的变化。

文献[8,18]受算法 CUSUM 启发,基于欧氏距离来度量窗口内部的突变,并提出了两种节省时间开销的算法 MB-GT 和 MB-CUSUM。算法流程为:在数据流上维持单个滑动窗口 SW;每当 SW 滑动一步,就将 SW 切分为两个子窗口 SW_1 和 SW_2,接着计算 $D(SW_1,SW_2)$,以此来判断是否发生了变化。在 MB-GT 和 MB-CUSUM 中,设 SW 的大小为 N,则每滑动一步都要将 SW 切分 $N-1$ 次。显然,这里充斥着大量的重复计算,因此 MB-GT 和 MB-CUSUM 采用了一个矩阵数据结构来存储 SW 中元素之间的欧氏距离。这样在构建好该矩阵后,切分过程只需要进行相应的查找运算就可以求出 $D(SW_1,SW_2)$,节省了时间开销。然而,MB-GT 和 MB-CUSUM 只能处理包含一个突变的数据流,无法处理含有多个变化的数据流。

在文献[9]中,Bifet 基于单一可变大小的滑动窗口提出了数据流变化检测算法 ADWIN 和 ADWIN2,其中 ADWIN2 是在 ADWIN 基础上的改进,二者原理一致。算法流程如下:首先维持一个滑动窗口 SW,其大小根据检测结果不断变化;SW 每滑动一步,便将其切分为两个子窗口 sw_1 和 sw_2,并计算 sw_1 和 sw_2 均值的差异。当该差异大于某个阈值时,则认为数据流发生了变化,并缩小 SW 为 sw_2;如果该差异值小于阈值,则增大 SW,即只添加数据元素到 SW,不删除元素。ADWIN 和 ADWIN2 不但能发现数据流均值的突变,而且能检测到均值的渐变;但是无法检测到方差的变化。

文献[19]的方法在数据流上维持一个滑动窗口 SW;对历史数据流元素进行学习并提取模型(线性或者非线性);接着对该学习到的模型做属性选择(feature selection),以关键属性及其值来代表该模型;同时在 SW 上进行同样的过程;最后对二者的关键属性做差异比较,以判断变化是否发生。

文献[20]采用了多项式拟合的方法来检测数据流的变化。该方法首先在数据流上拟合一个多项式,并基于此多项式进行预测;当预测累积误差超过某个阈值时,则表明该多项式不再适用,即数据流发生了变化。该方法的不足之处是需要事先给出多项式的阶数。

2.3　基于两个滑动窗口的检测方法

Kifer 等[10] 使用两滑动窗口来检测数据流的变化,并提出了 FIND_CHANGE 方法。算法流程如下:在数据流上维持两个窗口 SW_1 和 SW_2,其中 SW_1 固定,SW_2 沿着数据流单步滑动。每当 SW_2 滑动一步,FIND_CHANGE 就计算 P_1 和 P_2 的差异以检测变化是否发生:如果 P_1 和 P_2 的差异大于某个阈值,则变化发生,并将 SW_2 中的第一个元素报告为变化点,同时,清空 SW_1 和 SW_2,重复算法;如果差异小于阈值,则 SW_1 维持不变,SW_2 向前滑动一步。在该方法中,数据分布 P_1 和 P_2 分别从滑动窗口中的数据元素得到。在进行阈值比较时,FIND_CHANGE 使用了假设检验方法,为此,文中提出了两种能很好描述数据流的统计量 φ_A 和 Ξ_A。该方法既能发现数据流均值的变化,也能发现数据流方差的变化,但是却无法区分二者。

文献[21]与文献[22]中的方法类似,它们都基于两个滑动窗口来检测数据流的变化。它们的检测流程为:在数据流上维持两个滑动窗口,SW_1 和 SW_2 之间有一个数据元素 x_i;每当 SW_1 和 SW_2 滑动一步,算法用 SW_1 中的元素预测 x_i(前向预测),同时也用 SW_2 中的元素预测 x_i(后向预测);当前向预测和后向预测的差异达到某个阈值时,认为数据流发生了变化,并将 x_i 对应的时刻称为变化点。其中文献[21]使用最大似然法来进行预测,文献[22]使用多项式拟合来进行预测。虽然二者都能发现数据流中的变化点,但是却只能发现数据流均值的变化,不能检测数据流方差的变化。

文献[11]使用大小不同的两个滑动窗口来检测数据流的变化。SW_1 大小固定,而且不沿着数据流滑动,SW_2 为界标窗口;该方法认为如果数据流的数据分布在 t 时刻从 P_1 变化到 P_2,那么在接下来的一段时间 $(t\sim t+n)$ 内,数据流元素必定与 t 时刻之前的元素明显不同。基于该思想,该算法增量式地更新 P_2,最后使用统计假设检验的方法来判断 P_2 是否明显不同于 P_1,以此判断数据流是否发生变化。该方法需要一定的先验知识,即分布函数 P_1。该方法既能检测均值的变化也能检测方差的变化,但无法识别变化是由均值还是方差所引起。

文献[23]的方法检测了数据流中频繁项集的变化。该方法首先在数据流上维持两个滑动窗口;然后分别计算其中的频繁项集,同时基于该频繁项集定义了信息熵,并以此信息熵来计算两个窗口的差异。如果该差异大于某个阈值,认为数据流发生了变化。

为了能检测到数据流上不同粒度的变化,还有同时在多个滑动窗口上检测数据流变化的方法[10,12]。这一类方法同时在数据流上维持着不同大小的滑动窗口对,即 $\{(SW_1,\overline{SW_1}),(SW_2,\overline{SW_2}),\cdots\}$,其中每对窗口的大小相同,窗口对间的大小则不同;最后通过比较这些滑动窗口对内两个窗口之间的差异,确定数据流不同粒度的变化。这种方法通常称为弹性(elastic)方法。容易看到,这类方法等价于同时运行了 m(滑动窗口对的数目)个算法。因此,本质上也属于基于两个滑动窗口的方法。

2.4　其他检测方法

文献[24]提出了检测数据流均值变化的 SDEM 和 SDAR 算法。该算法首先从数据流历史元素中根据有限混合模型(Finite Mixture Model, FMM)或者自回归(Auto Regress, AR)模型学习一个概率密度函数 p,然后随着数据元素的到来增量式地更新 p;在每次更新前,SD 方法基于前一个 p_{i-1} 对当前数据流元素进行打分得到 $score_i$,这样便可以得到一个 $score$ 数据流;然后在 $score$ 数据流上重复上面的步骤,当 $score_i$ 超过某个阈值时,认为数据流发生了突变。SDEM 和 SDAR 算法通过两阶段(two-stage)策略来发现数据流中的变化,它的一个优势是不但能检测数据流中的变化点,而且能发现数据流中的离群点,并使得二者能够在统一的框架下被检测。

CF 方法[25] 在 SD 方法上进行了扩展,使得不但可以检测

数据流均值的变化，而且能够发现方差的变化（方差由小变大的变化）。CF 算法首先采用 KL 距离对数据流的变化做了形式化描述，然后基于自回归模型来发现数据流中的变化点和离群点。

文献[13]对检测网络流量异常模式进行了研究。首先以固定单位时长对 IP 包进行计数，生成一个关于 IP 包计数的数据流；然后采用分组测试（group testing）的方法来辨别某个计数值是否发生了变化。

Kleinberg[14] 首先提出了在文本流上进行突变检测的工作，主要用来检测电子邮件和出版文献等的突发（burst）。该方法利用现有的文本挖掘技术，如话题检测与追踪（topic detection and tracking）技术和文本分类（text classification）技术，把需要进行突发检测的文本进行归类；接着针对某一类别的文本，按照时间顺序转换为文本流。利用两个文本之间的时间间隔的变化来检测文本流上升的突变，即时间间隔越短，说明单位时间内与该文本对应的事件大量涌现，以此断定出现了异常。

文献[26]则首先建立原始数据流的概要信息，然后在概要信息的基础上进行预测，最后根据预测的误差率来判断数据流的变化。

2.5 检测方法的比较

上述三类方法虽然采取的策略不同，但都致力于以最小的内存占用来及时准确地检测数据流变化。为了清晰地展示几种检测方法的性能，采用几种常用的评价指标来评价上面几类检测方法。

检测率（Detection Rate，DR），表示算法检测变化的能力。检测率越高，意味着算法越优秀。

检测率 = 准确检测的变化点 / 数据流中所有的变化点

误报率（False Alarm Rate，FAR），表示算法的鲁棒性。误报率越低，说明算法对噪声越不敏感，算法性能越好。

误报率 = 误报的个数 / 检测到的变化点个数

检测延迟（Detection Delay，DD），表示算法实时性。检测延迟越小，说明算法实时性越好；反之则越差。

检测延迟 = $T_{\text{准确检测的变化}} - T_{\text{真实变化}}$

其中 T 表示对应的时刻。

为了便于叙述，我们称基于单一滑动窗口的方法为第一类方法；称基于两个滑动窗口的方法为第二类方法。由于文献[1,13-14,17,26]与具体领域相关密切，缺乏一般性，在此不作对比。

第一类方法[8-9,18] 和第二类方法[10-11,21-22] 都是通过计算集合间（数据流上的两个窗口）的差异来检测变化。由于集合之间的差异需要一定时间的积累才能显现出来。因此，在变化发生的时刻及其后非常小的一段时间内，这两种方法无法检测出这种变化。这使得它们不可避免地有较大的检测延迟，其中第一类方法的延迟为第二个子窗口的大小；第二类方法的延迟为第二个滑动窗口的大小。其他检测方法[24-25] 是基于预测的策略，因此实时性好，检测延迟小。

由于第一类和第二类方法是比较集合之间的差异，因此受离群点的和噪声的影响较小，所以误报率通常比较低。同时，第一类方法对窗口进行多次切分，所以检测率较高。而第二类方法的检测率则完全依赖窗口的大小。窗口越大，检测率越低；窗口越小，检测率越高。其他检测方法则是比较数据元素的差异（为数据点打分），所以较易受离群点和噪声的影响，产生较多的误报。但由于该类算法对变化很敏感，因此检测率比较高。

最后，第一类和第二类方法适用范围广泛，不仅能处理一维的数据流，也能处理多维的数据流，并能很容易扩展到概念迁移[9] 等其他领域。相反，其他检测方法的适用性较低，通常只能处理时间序列，而且无法处理多维的数据流。

综上，三类检测方法在每个指标上的表现如表 1 所示。

表 1 几种检测方法的比较

方法	检测率	误报率	检测延迟	适用性
第一类方法	较高	较低	大	一维、多维数据流
第二类方法	较低	较低	大	一维、多维数据流
其他检测方法	高	高	小	一维时间序列

3 数据流变化检测的应用

由于数据流来源的多样性，使得数据流上的变化检测有着十分广泛的应用。

3.1 对其他挖掘任务的支撑

数据流变化检测虽然可以作为一项独立的工作，但是通常也作为数据流挖掘的一个支撑，如对分类的支持、对聚类的支持等。在文献[9,27]中，通过变化检测，使得分类任务能够实时地对数据流的状态有所感知，并及时更新或者重新训练分类器，以达到更高的分类效率。文献[28]则通过变化检测来提高数据流聚类效率。

3.2 网络流量分析

数据流挖掘的一大应用场景便是网络流量分析，文献[13,29-30]等已经对此做了大量的研究。在一个大型公司或者研究机构的主干网络（backbone）中，每秒钟都有着大量的数据传输，如何及时准确地发现该实时环境中的异常，是管理人员非常关注的一个问题。本文涉及的数据流变化检测可以从计数的角度来发现数据流中与流量有关的异常，比如拒绝服务攻击（Denial of Service，DoS）等。

3.3 金融数据分析

金融数据通常是以数据流的方式产生的，如证券交易数据、超市的购物数据等。通常，在这种数据中隐含着人们的某种行为模式。因此，对金融数据做变化检测能及时识别出这种模式，以便制定更适合的策略来提高交易的效益，如文献[24]对证券交易指数的分析。

4 数据流变化检测发展方向

4.1 变化的定义和表示

由 1.3.1 节的讨论可知，数据流的多样性导致数据流变化的定义也多种多样。因此每种具体的应用都有各自不同的检测算法，算法的泛化能力差，如针对文本流的变化检测算法无法处理过程控制的数据流变化。截至目前，还没有统一的数据流变化定义及检测算法。总的来说，数据流变化检测还处于技术领先于理论的阶段。因此，需要形式化的定义数据流变化，并从理论上对变化检测进行指导。这是目前数据流变化检测的难点之一。

4.2 及时性与准确性

一方面，数据流变化检测需要及时定位数据流变化发生的位置，以便快速采取应对措施，这要求检测算法必须有小的检测延迟；另一方面，数据流变化检测也应当准确地找出这种变化，尽量减少误报带来的风险。然而，检测的及时性与准确性通常是此消彼长，不可兼得的。因此，在设计数据流变化检测算法时，需要在及时性和准确性之间进行取舍。总的来说，应该根据具体应用和需求，使算法侧重于相应的指标。比如在实时性要求较高的系统中，算法的性能应该侧重于检测的

及时性;而在实时性要求不太严格的系统中,应当以检测的准确性为重。算法的及时性和准确性是数据流变化检测的另一个研究热点。

4.3 滑动窗口大小

由于数据流变化检测算法通常基于滑动窗口,因此算法性能受滑动窗口影响较大。滑动窗口选取得当,则检测效率非常高;但是一旦选择了不合适的滑动窗口,则会导致算法性能急剧恶化。虽然文献[9]中的方法能够根据检测的结果自行调整窗口的大小,但是该文中的方法仅是对均值的变化进行检测,无法适应其他特征变化的应用。

5 结语

数据流变化检测是数据流挖掘中一个重要的研究课题。然而,由于数据流的不确定性和来源的多样性,使得数据流变化检测尚无统一的形式化描述,目前还处于技术领先于理论的阶段。

本文通过对数据流变化研究现状的深入分析,总结了数据流变化检测的一般流程;对当前主流的变化检测算法做了详尽的说明,并对比了几类方法的技术特点;最后展望了在该研究领域中存在的一些问题以及未来的研究方向。

参考文献:

[1] PAGE E S. Continuous inspection schemes [J]. Biometrika, 1954, 41(1/2): 100 – 115.

[2] HENZINGER M R, RAGHAVAN P, RAJAGOPALAN S. Computing on data streams [EB/OL]. [2011 – 02 – 15]. http://wenku.baidu.com/view/bb5cdd748e9951e79b892737.html.

[3] LEVIN D T, MOMEN N, DRIVDAHL S B. Change blindness: The metacognitive error of overestimating change-detection ability [J]. Visual Cognition, 2000, 7(1): 397 – 412.

[4] RENSINK R A, O'REGAN J K, CLARK J J. To see or not to see: The need for attention to perceive changes in scenes [J]. Psychological Science, 1997, 8(5): 368.

[5] TSYMBAL A. The problem of concept drift: Definitions and related work [R]. Dublin: Trinity College Dublin, Department of Computer Science, 2004.

[6] AGGARWAL C C. Data streams: Models and algorithms [M]. Berlin: Springer-Verlag, 2007.

[7] MUTHUKRISHNAN S. Data streams: Algorithms and applications [J]. Foundations and Trends in Theoretical Computer Science, 2005, 1(2): 117 – 236.

[8] NIKOVSKI D, JAIN A. Fast adaptive algorithms for abrupt change detection [J]. Machine Learning, 2010, 79(3): 283 – 306.

[9] BIFET A, GAVALDA R. Learning from time-changing data with adaptive windowing [EB/OL]. [2011 – 02 – 20]. http://www.lsi.upc.edu/~abifet/TimevaryingE.pdf.

[10] KIFER D, BEN-DAVID S, GEHRKE J. Detecting change in data streams [C]// Proceedings of the 30th International Conference on Very Large Databases. San Francisco: Morgan Kaufmann, 2004: 180 – 191.

[11] MUTHUKRISHNAN S, BERG E V D, WU Y. Sequential change detection on data streams [C]// Proceedings of the 7th IEEE International Conference on Data Mining Workshops. Washington, DC: IEEE Computer Society, 2007: 551 – 550.

[12] ZHU Y, SHASHA D. Efficient elastic burst detection in data streams [C]// Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. New York: ACM, 2003: 336 – 346.

[13] CORMODE G, MUTHUKRISHNAN S. What's new: Finding significant differences in network data streams [C]// Proceedings of the 23rd Annual Joint Conference of the IEEE Computer and Communications Societies. Washington, DC: IEEE Computer Society, 2004: 1534 – 1545.

[14] KLEINBERG J. Bursty and hierarchical structure in streams [J]. Data Mining and Knowledge Discovery, 2003, 7(4): 373 – 397.

[15] CHEN JIE, GUPTA A K. Testing and locating variance change-points with application to stock prices [J]. Journal of the American Statistical Association, 1997, 92(438): 739 – 747.

[16] GUHA S, MCGREGOR A, VENKATASUBRAMANIAN S. Streaming and sublinear approximation of entropy and information distances [C]// Proceedings of the 17th Annual ACM-SIAM Symposium on Discrete Algorithm. New York: ACM, 2006: 733 – 742.

[17] BARNARD G A. Control charts and stochastic processes [J]. Journal of the Royal Statistical Society, 1959, 21(2): 239 – 271.

[18] NIKOVSKI D, JAIN A. Memory-based algorithms for abrupt change detection in sensor data streams [C]// Proceedings of the 5th IEEE International Conference on Industrial Informatics. Piscataway: IEEE, 2007: 547 – 552.

[19] HUANG W, OMIECINSKI E, MARK L, *et al*. History guided low-cost change detection in streams [C]// Proceedings of the 11th International Conference on Data Warehousing and Knowledge Discovery. Berlin: Springer-Verlag, 2009: 75 – 86.

[20] GURALNIK V, SRIVASTAVA J. Event detection from time series data [C]// Proceedings of the 5th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. New York: ACM, 1999: 33 – 42.

[21] LI Z, MA H, ZHOU Y. A unifying method for outlier and change detection from data streams [C]// Proceedings of 2006 International Conference on Computational Intelligence and Security. Berlin: Springer-Verlag, 2006: 580 – 585.

[22] LI Z, MA H, ZHOU Y. A unifying method for outlier and change detection from data streams based on local polynomial fitting [C]// Proceedings of the 11th Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining. Berlin: Springer-Verlag, 2007: 150 – 171.

[23] 刘学军, 徐宏炳, 董逸生, 等. 基于最大频繁项集信息熵的数据流变化检测[J]. 应用科学学报, 2006, 24(5): 129 – 40.

[24] YAMANISHI K, TAKEUCHI J. A unifying framework for detecting outliers and change points from non-stationary time series data [C]// Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. New York: ACM, 2002: 676 – 681.

[25] TAKEUCHI J, YAMANISHI K. A unifying framework for detecting outliers and change points from time series [J]. IEEE Transactions on Knowledge and Data Engineering, 2006, 18(4): 482 – 492.

[26] KRISHNAMURTHY B, SEN S, ZHANG Y, *et al*. Sketch-based change detection: methods, evaluation, and applications [C]// Proceedings of the 3rd ACM SIGCOMM Conference on Internet Measurement. New York: ACM, 2003: 234 – 247.

[27] TSYMBAL A, PECHENIZKIY M, CUNNINGHAM P, *et al*. Dynamic integration of classifiers for handling concept [J]. Information Fusion, 2008, 9(1): 56 – 68.

[28] ZHOU A, CAO F, QIAN W, *et al*. Tracking clusters in evolving data streams over sliding windows [J]. Knowledge and Information Systems, 2007, 15(2): 181 – 214.

[29] DU P, ABE S, JI Y, *et al*. Detecting and tracing traffic volume anomalies in SINET3 backbone network [C]// Proceedings of 2008 IEEE International Conference on Communications. Piscataway: IEEE, 2008: 5833 – 5837.

[30] WANG H, ZHANG D, SHIN K G. Change-point monitoring for the detection of DoS attacks [J]. IEEE Transactions on Dependable and Secure Computing, 2004, 1(4): 193 – 208.