流数据挖掘综述*)

孙玉芬　卢炎生
(华中科技大学计算机科学与技术学院　武汉 430074)

摘　要　作为一种新的数据形态，流数据对数据挖掘提出了诸多挑战。学者们已提出大量处理流数据的挖掘算法。本文对这些算法进行了综述。首先介绍了多个不同的数据流模型，这些模型对算法设计有着不同的要求。然后，总结了流数据挖掘算法的特点，并给出了算法中常用的技术。最后，分析了各个流数据挖掘任务中的代表性算法。

关键词　数据流，数据挖掘，时空复杂度，滑动窗口

An Overview of Stream Data Mining

SUN Yu-Fen　LU Yan-Sheng
(Computer Department of Huazhong University of Science and Technology, Wuhan 430074)

Abstract　Data streams pose great challenges to data mining. Many stream data mining algorithms have been proposed. In this paper, we give an overview of these algorithms. Firstly, the data stream models are introduced. Then the characters of stream data mining algorithms are summarized and several techniques that are used in these algorithms are introduced. At last, the representative algorithms of every mining task are analyzed.

Keywords　Data stream, Data mining, Time-space complexity, Sliding window

1　引言

通信领域中的电话记录数据流、Web 上的用户点击数据流、网络监测中的数据包流、各类传感器网络中的检测数据流、金融领域的证券数据流、卫星传回的图像数据流以及零售业务中的交易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。这些数据流产生的数据量在多个应用领域中快速增长，小型无线传感设备的广泛使用将进一步使数据流体积的增长速度提高几个数量级。而且，产生数据流的应用通常要求在线实时处理。如何及时有效地处理数据流，从中挖掘出有用的知识，将对多个应用领域产生重大意义。

Henzinger 等人于 1998 年在论文“Computing on Data Stream”中首次将数据流作为一种数据处理模型提出来[1]。从 2000 年开始，数据流作为一个热点研究方向出现在数据挖掘与数据库领域的几大顶级会议中，如 VLDB、SIGMOD、SIGKDD、ICDE、ICDM 等会议每年都有多篇有关数据流处理的文章。目前，数据流研究大致可分为两个方面：数据流管理系统(Data Stream Management Systems, DSMS)和流数据挖掘[2]。其中，建立数据流管理系统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 DSMS 的研究，并构建出一些系统，如 STREAM[3]，TelegraphCQ[4]，Aurora[5] 等。流数据挖掘方面的研究主要包括多数据流挖掘和单数据流挖掘。目前学者们已提出了大量流数据挖掘算法，并开发出流数据挖掘系统。如 UIUC 的 MAIDS(Mining Alarming Incidents from Data Streams)就是一个集查询、聚类、分类、频繁项挖掘，以及处理结果可视化五大功能为一体的流数据挖掘系统[6]。本文主要讨论流数据挖掘算法。

2　数据流模型

数据流是一个以一定速度连续到达的数据项序列 x_1，…，x_i，…，x_n，…，这个数据项序列只能按下标 i 的递增顺序读取一次[1]。数据流是现象驱动的，其速度与数据项到达的次序无法被控制。数据流通常具有潜在无限的体积，且数据可能的取值是无限的，处理数据流的系统无法保存整个数据流。而数据流的在线处理要求又使系统无法进行代价昂贵的磁盘存取。因此，数据流中的数据项在被读取一次之后，就被丢弃，以后不可能再读到。在实际应用中，某些超大型的静态数据集要求处理算法只能进行一次线性扫描以降低算法的处理代价。此时，算法的输入也可看作是一种数据流[1,7]。

目前，在数据流研究领域中存在多种数据流模型。不同的数据流模型具有不同的适用范围，需要设计不同的处理算法。可以分别按照数据流中数据描述现象的方式和算法处理数据流时所采用的时序范围对这些模型进行划分。

设数据流中的数据项 x_1，…，x_i，…，x_n 依次按下标顺序到达，它们描述了一个信号 A。按 x_i 描述信号 A 的方式，数据流模型可分为以下几类[8]：

(1)时序(Time Series)模型：$A[i]=x_i$。此时，数据流中的每个数据项都代表一个独立的信号。

(2)现金登记(Cash Register)模型：令 $x_i=(j,I_i)$，且 $I_i\geqslant 0$，则 $A_i[j]=A_{i-1}[j]+I_i$。此时，数据流中的多个数据项增量式地表达一个 $A[j]$。

(3)十字转门(Turnstile)模型：令 $x_i=(j,U_i)$，则 $A_i[j]=A_{i-1}[j]+U_i$。其中，U_i 可为正数，也可为负数。此时，数据流中的多个数据项表达一个 $A[j]$。$A[j]$随着数据的流入，可能会增加，也可能会减小。

*)本文得到湖北省自然科学基金项目“时空数据库的关键技术研究与实验”(ABA048)的资助。孙玉芬　博士生，研究方向为流数据挖掘和聚类分析；卢炎生　教授，博导，研究方向为特种数据库、数据挖掘和软件测试。

在这3种模型中，Turnstile是最具一般性的数据流模型，其适用范围最广，也最难处理。流数据分类与聚类通常使用的是时序模型，它们将数据流中的每个数据项看作一个独立的对象。若将$A[j]$记为信号j出现的次数，则流数据频繁模式挖掘通常使用的是Cash Register模型，只允许数据的插入。也有算法研究了同时存在数据插入和删除时的流数据频繁模式挖掘问题。此时，算法应用的是数据流的Turnstile模型。

由于数据流是一个长期、动态的过程，部分算法在处理数据流时并不是将所有的数据流数据作为处理对象，而是根据应用需求选取某个时间范围内的数据进行处理。按算法处理数据流时所选取的时序范围，数据流模型可分为以下几类[9]：

(1)快照模型(snapshot model)：处理数据的范围限制在两个预定义的时间戳之间。

(2)界标模型(landmark model)：处理数据的范围从某一个已知的初始时间点到当前时间点为止。

(3)滑动窗口模型(sliding window model)：处理数据的范围由某个固定大小的滑动窗口确定，此滑动窗口的终点永远为当前时刻。其中，滑动窗口的大小可以由一个时间区间定义，也可以由窗口所包含的数据项数目定义。

在这3种模型中，界标模型和滑动窗口模型是采用得比较多的模型。界标模型通常将数据流的起始点作为数据处理的初始时间点。此时，算法对数据流中所有数据进行处理，数据流上只存在插入操作。在滑动窗口模型中，窗口随着数据的流入向前滑动，窗口中存在数据的插入和删除。滑动窗口模型非常适用于只要求对最近时间段内的数据进行处理的应用。

3 流数据挖掘算法的特点

数据流实时、连续、有序、快速到达的特点以及在线分析的应用需求，对流数据挖掘算法提出了诸多挑战。数据流对挖掘算法的典型要求如下：

(1)单次线性扫描。即算法只能按数据的流入顺序依次读取数据一次。

(2)低的时间复杂度。流算法是在线算法，为了跟上数据流的流速，算法处理每个数据项的时间不能太长，最好能为常数时间。

(3)低的空间复杂度。流算法是主存算法，其可用的空间是有限的，算法的空间复杂度不能随数据量无限增长。

(4)能在理论上保证计算结果具有好的近似程度。

(5)能适应动态变化的数据与流速。产生数据的现象可能在不断变化，导致数据内容与流速的改变。

(6)能有效处理噪音与空值。这是一个具有健壮的算法所必须具有的能力。

(7)能作on demand的挖掘。即能响应用户在线提出的任意时间段内的挖掘请求。

(8)能作anytime的回答。即算法在任何时刻都能给出当前数据的挖掘结果。

(9)建立的概要数据结构具有通用性。算法所构建的概要数据结构不仅能支持算法当前的目标计算，而且能支持其他的计算。

在上述要求中，第1至3条是一个流数据挖掘算法所必须满足的。早期的流数据挖掘算法都是以这三项为目标设计的，如文[10，11]。对于算法的空间复杂度，理想的情况是它与数据流长度N无关。但是，目前大部分问题都无法找到这样的解。因此，这个要求就让步为找到空间复杂度为$O(poly(\log N))$的算法，即次线性算法。算法的时间复杂度通常以每个数据项到来时，更新概要数据结构或目标计算结果所需要的时间来衡量，理想的情况是算法处理每个数据项的时间为常数。其中，概要数据结构是算法为支持目标计算而在内存中保存的数据流数据的压缩信息。对于构建概要数据结构的算法，通常没有对在概要数据结构上计算目标函数所需要的时间做严格的要求。

近似性与自适应性是数据流算法的两大特点[3]。由于一次线性扫描以及时间与空间的限制，数据流算法往往只能得到对所处理的问题的近似计算结果。能在理论上保证其计算结果的近似程度，是算法应该考虑的一个问题。算法的自适应性是指当流数据内容或流速受各种因素的影响而发生改变时，算法能够根据这些改变自动调整计算策略与计算结果。

噪音与空值是一个健壮的算法所必须解决的问题。对于流数据挖掘算法，这个问题显得更为突出。这是因为在挖掘数据库中的静态数据集之前，通常会进行数据的预处理，消除数据中的噪音与空值。而在在线进行的流数据挖掘过程中，无法在挖掘前对数据进行预处理。而且，数据流中的数据在采集以及传输过程中，都可能出现错误，产生噪音或空值。数据流的动态变化性更进一步增加了噪音识别的困难。当产生数据流的现象发生改变时，新数据无法被现有数据模型所描述，可能被误认为是噪音。

在一些应用中，用户可能在数据流流入过程中提出对某个时间段内的数据进行挖掘的请求。能回答这种请求的算法被称为具有on demand回答能力的算法。算法通常采用多窗口技术来近似解决这类问题。能对挖掘请求给出anytime的回答，指算法在任何时刻都能给出对当前数据最精确的计算结果。这要求算法每读取一个数据项，就更新处理结果。

有些算法构建的概要数据结构只能用来支持算法的目标计算，如文[12]中为计算数据流对之间的滞后关联而保留的统计量。有的概要数据结构是对数据流数据一般性的压缩，还可用来支持其他计算，如文[13]中保留的多个基本窗口内数据的傅立叶变换系数。这样的概要数据结构显然比只能支持当前计算的概要数据结构更为有用。

4 相关技术

基于对多个流数据挖掘算法的分析，我们总结了算法常用的一些技术。这些技术主要包括概要数据结构、滑动窗口技术、多窗口技术、衰减因子、近似技术等。

4.1 概要数据结构

在流数据处理系统中，由于数据量远大于可用内存，系统无法在内存中保存所有扫描过的数据，而流数据查询与挖掘经常会要求读取这些数据。为了避免代价昂贵的磁盘存取，流数据处理系统必须在内存维持一个概要数据结构，以保留扫描过的信息。文[9]已对建立概要数据结构的方法作了综述，本文在这里只作一个简单的介绍。

目前，生成数据流概要数据结构的主要方法包括取样(sampling)、直方图(histogram)、小波变换、Sketching、Load shedding和哈希方法。其中，取样方法将数据流中的数据项以一定概率抽取到概要数据结构中。直方图按照数据项的取值或出现频率将数据项划分为桶，对每个桶压缩表示。小波方法对原有数据做小波变换，将保留原有数据主要信息的少

数几个小波参数作为概要数据保存。Sketching 对数据作垂直取样。Load shedding 在负载过大时直接丢弃一些数据项。哈希方法通过一组哈希函数,将大量数据映射到少量桶中。

4.2 滑动窗口技术

使用滑动窗口的需求来自于算法和应用。在算法方面,滑动窗口减少了算法需要处理的数据量,并对挖掘变化的数据流提供支持。在应用方面,有些应用只对最近的数据感兴趣,要求算法对以当前时间为终点的某个滑动窗口内的数据进行处理。

在滑动窗口上进行数据挖掘最大的困难在于过期数据的移除。随着数据的流入,滑动窗口中最早到达的数据将滑出窗口的范围,算法需要消除这些数据对滑动窗口上的目标计算所造成的影响。解决这个问题的最直接的做法是保存滑动窗口内的所有数据,当某个数据滑出窗口时,根据这个数据的值,将其从计算结果中消除。目前,多个采用滑动窗口模型的挖掘算法使用这种方法实现滑动窗口上的计算,如 CVFDT[14]。这种方法可以精确地对滑动窗口内的计算结果进行增量式地更新。但是,由于要保存窗口内的所有数据,对于其大小超过可用内存空间的滑动窗口,仍然需要进行磁盘存取。

为减少滑动窗口内数据所占用的空间,另一种方法以降低滑动窗口上计算的精度为代价,使用小于滑动窗口内数据体积的空间,支持滑动窗口上计算的增量式更新。这种方法将数据流划分为小的固定长度的段(bucket,或 basic window),对每个段,仅保存段内数据的概要信息,如 StaStream[13]。滑动窗口在这些段上滑动。当流入的数据积累成一段时,抽取这一段的概要信息,将其加入滑动窗口,并从滑动窗口中删除最早的段。这样,内存中就只需要保存滑动窗口中多个段的概要信息。此时,滑动窗口的增量式更新粒度由一个数据项增大为一个数据段。这种方法通常只支持大小为段大小的整数倍的滑动窗口上的计算[15]。文[13]通过保存每个段的数据的离散傅立叶变换系数,能支持任意窗口大小内的数据流关联系数计算。

4.3 多窗口技术

基于滑动窗口的方法一般都要求用户事先指定窗口大小,算法在运行过程中只能给出此滑动窗口上的计算结果。而在很多应用中,用户可能在线提出某个窗口上的挖掘请求,此窗口的大小没有事先确定,而且窗口的终点可能也不是当前时刻。为了支持这样的应用需求,学者们提出一种多窗口方法,支持用户的在线挖掘请求。

多窗口技术在内存或磁盘中保存数据流上多个窗口内数据的概要信息。在有些算法中,每个窗口的范围都是从数据流起始点到窗口建立的时刻点,窗口中的数据存在重叠,如 CluStream 所使用的 pyramidal 时间框架[16]。另一类多窗口技术将数据流划分为多个固定长度的段,每个段都形成一个窗口。当内存中的窗口数达到一定数目时,就将这多个窗口合并,形成概要层次更高的窗口。随着数据流的流入,概要层次不同的多个窗口形成一个层次结构。此时,每个窗口相当于对数据流上两个预定义的时间戳之间数据的一个快照。

4.4 衰减因子

除了滑动窗口技术,另一种可被用来消除历史数据对当前计算结果的影响的方法是使用衰减因子[17]。在这种方法中,每个数据项都被赋予一个随时间不断减小的衰减因子,数据项的值与衰减因子相乘后再参与计算。因此,数据项对计算结果的影响随时间的推移逐渐减小。这种方法的实现很简单,但是,与滑动窗口技术相比,其计算结果的意义不是非常明确。在使用滑动窗口的算法中,用户明确地知道他是在对哪些数据进行处理。而在使用衰减因子的方法中,每项数据都只是部分地参与了计算,用户无法确定计算结果到底由哪些数据得到。

4.5 近似技术

由于数据流处理严格的时间与空间限制,确定且精确的数据流算法比较少见。对于大多数算法,只能以降低计算结果的精度为代价,换取算法时空复杂度的降低。能在理论上保证近似程度的算法是比较理想的近似算法。

目前,有多种近似技术可用来降低算法的时空复杂度。例如,基于概要数据结构的算法都是近似算法。这是因为在构建概要数据结构时,不可避免地存在着信息的损失。概要数据结构只能近似还原原有数据。基于多窗口技术和衰减因子的算法也是近似算法。除了使用这些通用的压缩技术,也可针对具体的挖掘任务,设计相应的近似算法[10]。

4.6 自适应技术

由于数据流是动态变化的,处理数据流的算法必须能够根据数据分布的变化以及数据流流速的变化自动调节算法的处理策略。动态系统中的自适应技术根据系统的反馈自动调节系统参数。目前,在处理变化的数据流时,算法通常将分类器的分类精度作为反馈,在精度下降时重新建立分类模型。

5 流数据挖掘算法

流数据挖掘的对象可以是多条数据流,也可以是单条数据流。挖掘多条数据流的主要目的是分析多条并行到达的数据流之间的关联[2,12,13,15,18,19]。对单数据流的挖掘则涵盖了分类、频繁模式挖掘、聚类等多项传统数据挖掘中的主要任务。挖掘变化的数据流是一项特殊的任务,目前主要是以单数据流为对象进行研究的。

5.1 多数据流关联分析

现有的多数据流关联分析主要采用 3 种方法,即计算数据流对之间的关联系数[12,13]、计算多条数据流的主分量[2,15],以及计算多条数据流中存在的聚类[18,19]。

5.1.1 关联度计算

关联度计算指在多条数据流中,计算每对数据流之间的关联系数,从而发现具有高的正关联或负关联的数据流对。当数据流数目较大时,在线计算每对数据流之间的关联系数是不现实的。文[13]实现的 StaStream 系统通过使用离散傅立叶变换的保距特性与系数的对称特性,推导出数据流对傅立叶变换系数之间的距离与关联系数之间的关系。系统只对傅立叶变换系数之间距离满足一定条件的数据流对计算关联系数。StaStream 采用数据流的滑动窗口模型,并将每条数据流划分为小的固定长度为 b 的段(基本窗口),对每个段,保存段内数据的离散傅立叶变换系数。系统将滑动窗口内的段作为数据流的概要数据结构。这个概要数据结构还可为其他计算提供支持。

StaStream 没有对数据流对之间存在滞后关联的情况作太多讨论,但是这种情况在应用中比较常见。文[12]提出的 BRAID 方法,讨论了滞后关联(lag correlation)的计算。BRAID 采用界标模型,对数据流从起始到当前时刻所有的数据进行处理。其概要数据结构只能用来计算数据流对之间的关联系数。

5.1.2 主分量计算

文[15]对多条数据流组成的矩阵作奇异值分解(Singular Value Decomposition,SVD)分析,并使用得到的特征值和特征向量表达数据流之间的关联。文章采用数据流的十字转门(Turnstile)模型,给出了界标模型和滑动窗口模型下的算法。其中,采用滑动窗口模型的算法将滑动窗口内的数据划分为多个段,分段保存矩阵的组成数据,当某个段的时间戳滑出滑动窗口时,将整个段删去。

文[15]在数据流上进行 SVD 分析的计算代价过大。文[2]采用 PCA(principal component analysis)技术分析多数据流,将 n 条数据流用 k 个隐藏变量表示,其中 $k \ll n$。但是,它没有使用 SVD 计算主分量,而是基于自适应过滤技术(adaptive filtering techniques)实现了一个增量式的主分量获取算法。文中使用指数衰减因子来逐渐消除历史数据对计算结果的影响。

5.1.3 多数据流聚类

与定量计算数据流之间的关联统计量不同,另一种多数据流关联分析方法对多条数据流进行聚类分析[18,19],发现彼此间相似的数据流。文[18]主要讨论了如何减少每个时刻需要计算的数据流对之间的距离。文章采用数据流的界标模型。文[19]提出一个滑动窗口模型下的多数据流聚类方法。此方法基于一个层次概要数据结构支持任意大小滑动窗口内的多数据流聚类。

5.2 单数据流挖掘

为避免与已有文献的重复,对于单数据流挖掘,本文只分析了各项挖掘任务中最具有代表性的算法,更全面的算法介绍请参见文[20]。

5.2.1 分类

为了对数据流进行实时处理,要求算法在看到整个数据流之前就能处理数据并得到处理结果。而传统的判定树构造算法必须从一开始就能够读取整个数据集。文[21]提出一个针对数据流的增量式判定树构造算法。算法基于 Hoeffding 不等式,以一定概率保证其增量式生成的判定树与使用传统算法在整个数据集上生成的树相差不大。

设数据中包含 c 个类别,$G(X_i)$ 为构建判定树时为选择分裂属性所计算的各属性的信息增益。设在读取 n 个数据后,$\bar{G}(X_a)$ 最大,$\bar{G}(X_b)$ 次大。对于某个指定的 δ,若 $\Delta\bar{G}=\bar{G}(X_a)-\bar{G}(X_b)\geqslant\sqrt{\frac{(\log c)^2\ln(1/\delta)}{2n}}$,则 Hoeffding 不等式以概率 $1-\delta$ 保证选择 X_a 作为分裂属性是正确的。此时,算法只使用部分数据项就能以一定概率正确选择树节点的分裂属性,从而将只能在整个数据集进行的批处理式的判定树构造算法改进为一个增量式的判定树构造算法。

文[22]设计了一个基于聚类分析的数据流分类算法。还有大量文献研究了如何在变化的数据流上建立分类器[14,23~29],这些文献将在后面详细讨论。

5.2.2 频繁项挖掘

在动态数据集上挖掘频繁项是一项困难的任务[30]。数据流所要求的单次线性扫描进一步增加了这项任务的难度。随着数据的不断流入,频繁项可能会变得不频繁,非频繁项也可能成为频繁项。要精确地计算数据流中的频繁项,算法必须保存所有的历史数据。但是,对于流数据,这是一个无法达到的要求。因此,数据流上的频繁项挖掘算法只能得到近似计算结果。

文[10]提出一个近似的流数据频繁项挖掘算法。算法保存多个三元组记录:(e,f,Δ),其中,e 是数据流中的元素,f 为 e 的估计频率,Δ 是 e 的最大可能的错误,即若记 e 的真实频率为 f_e,则 $f_e\leqslant f+\Delta$。对于选定的参数 ε,每当算法遇到一个没有被记录的新元素 e' 时,就生成一个新元组 $(e',1,\lfloor\varepsilon N\rfloor)$;每当算法读取 $\left\lceil\frac{1}{\varepsilon}\right\rceil$ 个数据项时,就删除所有 $f+\Delta\leqslant\varepsilon N$ 的元组。其中,N 为目前已读取的总的数据项数目。通过这种处理,算法保证所有 $f_e>\lfloor\varepsilon N\rfloor$ 的元素都被记录,且 $f\leqslant f_e\leqslant f+\varepsilon N$。对于任意的频繁度阈值 $s>\varepsilon$,输出满足条件 $f\geqslant(s-\varepsilon)N$ 的项就保证所有 $f_e\geqslant sN$ 的项都被输出,且所有输出项都满足条件 $f_e\geqslant(s-\varepsilon)N$。在实际应用中,大部分数据的出现频率都较低,通过采用上述方法,算法不需要记录出现频率较低的数据,从而既节省了计算空间,同时又保证了输出的质量。

上述算法采用的是数据流的界标模型,在整个数据流上进行计算。文[31]将其扩展到数据流的时间窗口模型上,实现了多时间粒度的频繁项挖掘。

5.2.3 聚类

第一个以数据流为分析对象的聚类算法是由斯坦福大学的 Sudipto Guha 等人提出的[32]。这个算法采用分而制之(Divide-and-Conquer)的思想,将数据流划分为多个段,算法对每段分别聚类,得到第一层簇中心。当第一层簇中心达到一定数目后,对其进行聚类,得到第二层簇中心。这样的过程伴随数据的流入一直进行,在每个时刻,系统最多维持 m 个第 i 层中心点。此算法在整个数据流上进行计算。由于每次都要积累一定数目的数据后才进行处理,此算法只能看作是分批批处理算法。

算法 CluStream[16] 与 HPStream[17] 是流数据挖掘中的两个重要的聚类分析算法。这两个算法都在线计算 micro-cluster[33]。CluStream 使用金字塔时间结构(Pyramidal Time Frame)保存一系列 micro-clusters 的快照,从而能够以较小的误差计算任意时间段内的聚类,且能够分析随时间推进聚类的改变。HPStream 针对高维聚类问题,动态地选择使聚类体积最小的那些维与聚类关联,实现了一个子空间聚类算法。与 CluStream 在整个数据流上计算 micro-cluster 不同的是,HPStream 使用衰减因子随时间推进不断衰减历史数据,并在聚类数目过多时,删除最早加入的聚类。CluStream 与 HPStream 都是增量式的聚类算法,在每个数据项到来时都进行处理,因此它们都能作 anytime 的回答。

在流数据聚类研究中,还出现大量其他的挖掘方法。文[34]和文[35]将基于网格的聚类算法应用到数据流上。文[36]扩展了 K-划分算法和 CURE 算法。文[37]提出 K-means 算法的变体以聚类二元数据流。文[38]实现了一个滑动窗口模型上的 K-Medians 聚类算法。

5.3 挖掘变化的数据流

由于数据流是一类流速与数据内容都随时间动态变化的数据对象,挖掘变化的数据流成为流数据挖掘领域的一个特有的研究内容。目前,挖掘变化的数据流包括两方面的研究:模型跟踪[28]和变化挖掘[39]。其中,模型跟踪与机器学习中的概念跟踪[40~43]可以看作同一个概念。变化挖掘与统计中的改变点检测(change-point detection)[44]和时序数据上的分割(segmentation)[45,46]类似。

在变化的数据流上建立分类模型,是一个研究得比较多

的问题[14, 23~27, 29]。文[14]中提出的 CVFDT 基于文[21]中的 VFDT 算法，在一个动态调节大小的滑动窗口上，增量式地建立一个随数据变化动态变化的判定树。为消除历史数据对判定树计算的影响，算法必须保留整个滑动窗口内的所有数据。

另一类在变化的数据流上建立分类模型的方法通过建立多个分类模型来实现模型的跟踪[24~29]。文[27]第一个将机器学习中的集成方法(ensemble methods)用来对变化的数据流分类。算法在数据流上维持一个由固定数目的分类模型组成的 ensemble。每当算法读取一定量的数据后，就在此段数据上建立一个分类模型。若此分类模型能够提高 ensemble 的分类性能，则用其替换 ensemble 中性能最差的一个分类模型。Ensemble 使用无权重的 majority voting 投票规则对数据进行分类。文[24]与文[25]根据各分类模型在当前数据段上的预测错误期望，赋予它们适当的权重。文[28]使用 logistic 回归技术，通过最大化数据的似然给 ensemble 中的各个分类模型赋予最优的权重。文[26]在 ensemble 中的所有分类模型中找出其训练数据集与当前数据最相似的分类模型。为了实现这个分类模型选择策略，算法为每个分类模型保存相应的训练数据集。这种选择策略可有效减少 ensemble 中的冗余信息，尽可能扩大 ensemble 中分类模型的数据覆盖范围。在对当前数据进行分类时，文[24~28]都是使用多个分类模型分类结果的组合，文[29]在多个分类模型中选择对当前数据分类效果最好的分类模型进行分类。

文[47]按数据流中是否发生概念转移以及训练模型的数据是否充分这几种不同的情况，使用不同的数据集训练得到多个分类模型，然后选择分类正确度最高的模型作为当前最优模型。

文[48]研究了如何发现数据流中数据分布的改变并将其可视化。文章通过计算数据流入时数据空间中各数据点密度的改变情况，发现数据点的转移轨迹与趋势，并将这些转移以图形的形式表示出来。

比较两个数据集的数据分布是否相同，是统计上曾经研究过的一个问题[49]。通过在数据流上维持多个数据窗口，文[50]将检测数据流中的变化转化为比较两个数据集的分布是否相同的问题，并提出一个非参数方法来解决这个问题。文[51]更进一步，设计了一个新的数据结构来度量两个数据集之间的相似度。

结束语 本文介绍了现有的数据流模型，总结了流数据挖掘算法的九大特点，并讨论了算法中常用的几种技术。文中还针对不同的挖掘任务，分析了其代表性算法。这些内容对于深入了解流数据挖掘并提出新的挖掘算法，都有重要意义。

参 考 文 献

1 Henzinger M R, Raghavan P, Rajagopalan S. Computing on data streams. SRC Technical Note 1998-011. Digital systems research center: Palo Alto, California, 1998

2 Papadimitriou S, Sun J, Faloutsos C. Streaming Pattern Discovery in Multiple Time-Series. In: Proc of the 31st VLDB Conf, 2005. 697~708

3 Babcock B, et al. Models and issues in data stream systems. In: Proc of 21st ACM Symposium on Principles of Database Systems (PODS 2002), 2002. 1~16

4 Chandrasekaran S, et al. TelegraphCQ: Continuous Dataflow Processing for an Uncertain World. Proc of The Conf on Innovative Data Systems Research (CIDR), 2003

5 Abadi D J, et al. Aurora: A New Model and Architecture for Data Stream Management. The Intl Journal on Very Large Data Bases, 2003, 12(2): 120~139

6 Cai Y D, et al. MAIDS: Mining Alarming Incidents from Data Streams. In: Proc of the 2004 ACM SIGMOD Intl Conf on Management of data, 2004. 919~920

7 O'Callaghan L. Approximation algorithms for clustering streams and large data sets: [Ph D Thesis]. The Department of Computer Science, Stanford University, 2003

8 Muthukrishnan S. Data streams: Algorithms and applications. In: Proc of the fourteenth annual ACM-SIAM symposium on discrete algorithms, 2003. 413~413

9 金澈清，钱卫宁，周傲英. 流数据分析与管理综述. 软件学报，2004, 15(8): 1172~1181

10 Manku G S, Motwani R. Approximate frequency counts over data streams. In: Proc of the 28th VLDB Conf, 2002. 346~357

11 O'Callaghan L, et al. Streaming-Data Algorithms for High-Quality Clustering. In: Proc of the 18th Intl Conf on Data Engineering (ICDE'02), 2002. 685~694

12 Sakurai Y, Papadimitriou S, Faloutsos C. BRAID: Stream Mining through Group Lag Correlations. In: Proc of the 2005 ACM SIGMOD Intl Conf on Management of Data, 2005. 599~610

13 Zhu Y, Shasha D. StatStream: Statistical Monitoring of Thousands of Data Streams in Real Time. In: Proc of the 28th VLDB Conf, 2002. 358~369

14 Hulten G, Spencer L, Domingos P. Mining Time-Changing Data Streams. In: Proc of the 7th ACM SIGKDD Intl Conf on Knowledge Discovery and Data Mining, 2001. 97~106

15 Guha S, Gunopulos D, Koudas N. Correlating Synchronous and Asynchronous Data Streams. In: Proc of The 9th ACM SIGKDD Intl Conf on Knowledge Discovery and Data Mining, 2003. 529~534

16 Aggarwal C C, et al. A Framework for Clustering Evolving Data Streams. In: Proc of the 29th VLDB Conf, 2003. 81~92

17 Aggarwal C C, et al. A Framework for Projected Clustering of High Dimensional Data Streams. In: Proc of the 30th VLDB Conf, 2004. 852~863

18 Yang J. Dynamic Clustering of Evolving Streams with a Single Pass. In: Proc of the 19th IEEE Intl Conf on Data Engineering (ICDE'03), 2003. 695~697

19 Dai B-R, et al. Clustering on Demand for Multiple Data Streams. In: Proc of the Fourth IEEE Intl Conf on Data Mining (ICDM'04), 2004. 367~370

20 Gaber M M, Zaslavsky A, Krishnaswamy S. Mining data streams: A review. SIGMOD Record, 2005, 34(2): 18~26

21 Domingos P, Hulten G. Mining High-Speed Data Streams. In: Proc of the sixth ACM SIGKDD Intl Conf on Knowledge Discovery and Data Mining, 2000. 71~80

22 Aggarwal C C, et al. On Demand Classification of Data Streams. In: Proc of the 2004 ACM SIGKDD Intl Conf on Knowledge Discovery and Data Mining, 2004. 503~508

23 Yang Y, et al. Combining Proactive and Reactive Predictions for Data Streams. Proc of the 2005 ACM SIGKDD Intl Conf on Knowledge Discovery and Data Mining, 2005

24 Wang H, et al. Mining Concept-Drifting Data Streams using Ensemble Classifier. In: Proc of the 9th ACM SIGKDD Intl Conf on Knowledge Discovery and Data Mining, 2003. 226~235

25 Kolter J Z, Maloof M A. Dynamic Weighted Majority: A New Ensemble Method for Tracking Concept Drift. In: Proc of the Third IEEE Intl Conf on Data Mining, 2003. 123~130

26 Rushing J, et al. A Coverage Based Ensemble Algorithm (CEBA) for Streaming Data. In: Proc of the 16th IEEE Intl Conf on Tools with Artificial Intelligence (ICTAI'04), 2004. 106~112

27 Street W N, Kim Y. A Streaming Ensemble Algorithm (SEA) for Large-Scale Classification. In: Proc of the Seventh ACM SIGKDD Intl Conf on Knowledge Discovery and Data Mining, 2001. 377~382

28 Chu F, Wang Y, Zaniolo C. An Adaptive Learning Approach for Noisy Data Streams. In: Proc of the Fourth IEEE Intl Conf on Data Mining (ICDM'04), 2004. 351~354

29 Zhu X, Wu X, Yang Y. Dynamic Classifier Selection for Effective Mining from Noisy Data Streams. In: Proc of the Fourth IEEE Intl Conf on Data Mining, 2004. 305~312

30 Cormode G, Muthukrishnan S. What's hot and what's not: Tracking most frequent items dynamically. ACM Trans. on Database Systems, 2005, 30(1): 249~278

(下转第 11 页)

ble inversion: [Master's thesis]. Massachusetts Institute of Technology, available from http://theory.lcs.mit.edu/~cis/cis-theses.html, 2003
8 Kuwakado H, Tanaka H. Transitive signature scheme for directed trees. IEICE TransFundamental, 2003, E86-A(5): 1120~1126
9 Van Heijst E, Pedersen T P, Pfitzmann B. New constructions of fail-stop signatures and lower bounds. In: Crypto'92. Berlin: Springer-Verlag, 1993, LNCS 740: 15~30
10 Zhou S J. Transitive Signatures Based on Non-adaptive Standard Signatures. Cryptography ePrint Archive. Report 2004/044/
11 Zhu H. Model for undirected transitive signatures. IEE Proceedings Communications, 2004, 151(4): 312~315
12 Shahandashti S F, Salmasizadeh M, Mohajeri J. A Provably Secure Short Transitive Signature Scheme from Bilinear Group Pairs. In: SCN 2004. Berling: Springer-Verlag, 2005, LNCS 3352: 60~76
13 Bellare M, Neven G. Transitive signatures: New Schemes and Proofs. IEEE Transactions on Information Theory, 2005, 51(6): 2133~2151
14 黄振杰，郝艳华. 一个高效的有向传递签名方案. 电子学报, 2005, 33(8): 1497~1501
15 Ma C G, Wu P, Gu G C. A New Method for the Design of Stateless Transitive Signature Schemes. In: APWeb Workshops 2006. Berlin: Springer-Verlag, 2006, LNCS 3842: 897~904
16 Aho A V, Garey M R, Ullman J. The transitive reduction of a directed graph [J]. SIAM Journal of Computing, 1972, 1(2): 131~137
17 Goldwasser S, Micali S, Rivest R. A digital signature scheme secure against adaptive chosen-message attacks. SIAM Journal on Computing, 1988, 17(2): 281~308
18 Schnorr C P. Efficient identification and signatures for smartcards. In: Brassard G. editor. Advances in Cryptology - CRYPTO 1989. Berlin: Springer-Verlag, 1990, LNCS 435: 239~252
19 Bellare M, Namprempre C, Pointcheval D, et al. The one-more-RSA-inversion problems and the security of Chaum's blind signature scheme. Journal of Cryptology, 2003, 16(3): 185~215
20 Bellare M, Rogaway P. Random oracles are practical: A paradigm for designing efficient protocols. In: ACM. editor. Proceedings of the First Conference on Computer and Communications Security. Fairfax, 1993. 62~73
21 Canetti R, Goldreich O, Halevi S. The Random Oracle Methodology, Revisited. Journal of the ACM, 2004, 51(4): 557~594
22 Cramer R, Shoup V, Signature schemes based on the strong RSA assumption. ACM Transactions on Information and System Security (ACM TISSEC), 2000, 3(3):161~185
23 Fischlin M. The Cramer-Shoup Strong-RSA signature scheme revisited. In: Public Key Cryptography - PKC'03. Berlin: Springer, 2003, LNCS 2567: 116~129
24 Boldyreva A. Threshold signatures, multisignatures and blind signatures based on the gap-Diffie Hellman- group signature scheme. In: Desmedt Y. editor. Advances in Cryptology - Public-Key Cryptography 2003. Berlin: Springer-Verlag, 2003, LNCS 2567: 31~46
25 Boneh D, Franklin M. Identity Based Encryption from the Weil Pairing. SIAM Journal of Computing, 2001,32 (3) :586~615
26 Boneh D, Lynn B, Shacham H. Short signatures from the Weil pairing. In: Boyd C. editor. Advances in Cryptology - ASIACRYPT 2001. Berlin: Springer-Verlag, 2001, LNCS 2248: 514~532
27 Boneh D, Mironov I, Shoup V. A Secure Signature Scheme from Bilinear Maps. In: Proceedings of RSA-CT'03, Berlin: Springer-Verlag, 2003, LNCS 2612: 98~110
28 Goldreich O, Goldwasser S, Micali S. How to construct random functions. Journal of the ACM, 1986, 33(4): 792~807
29 Yi X, Tan C H, Okamoto E. Security of Kuwakado-Tanaka Transitive Signature Scheme for Directed Trees. IEICE Transactions on Fundamentals, 2004, E87-A(4): 955~957
30 马春光，杨义先. 可转移离线电子现金. 计算机学报, 2005, 28(3): 301~308
31 马春光，杨义先，胡正名，等. 可直接花费余额的电子支票系统. 电子学报, 2005, 33(9): 1562~1566

(上接第 5 页)

31 Giannella C, et al. Mining frequent patterns in data streams at multiple time granularities. In: Next Generation Data Mining, 2003. 191~212
32 Guha S, et al. Clustering Data Streams. In: Proc of the 41st Annual Symposium on Foundations of Computer Science, 2000. 359~366
33 Zhang T, Ramakrishnan R, Livny M. BIRCH: An Efficient Data Clustering Method for Very Large Databases. In: Proc of the 1996 ACM SIGMOD Intl Conf on Management of Data, 1996. 103~114
34 Park N H, Lee W S. Statistical Grid-Based Clustering over Data Streams. ACM SIGMOD Record, 2004, 33(1): 32~37
35 Lu Y, et al. A Grid-Based Clustering Algorithm for High-Dimensional Data Streams. In: Proc of the 1st Intl Conf on Advanced Data Mining and Applications (ADMA), 2005. 824~831
36 Wang Z, et al. Clustering Data Streams on the Two-Tier Structure. In: Advanced Web Technologies and Applications: 6th Asia-Pacific Web Conf (APWeb 2004), 2004. 416~425
37 Ordonez C. Clustering Binary Data Streams with K-Means. In: Proc of the 8th ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery, 2003. 12~19
38 Babcock B, et al. Maintaining Variance and k-Medians over Data Stream Windows. In: Proc of the twenty-second ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems, 2003. 234~243
39 Dong G, et al. Online Mining of Changes from Data Streams: Research Problems and Preliminary Results. Proc of the 2003 ACM SIGMOD Workshop on Management and Processing of Data Streams. 2003
40 Widmer G. Tracking Context Changes through Meta- Learning. Machine Learning, 1997, 27(3): 259~286
41 Widmer G, Kubat M. Learning in the Presence of Concept Drift and Hidden Contexts. Machine Learning, 1996, 23(1): 69~101
42 Bartlett P L, Ben-David S, Kulkarni S R. Learning Changing Concepts by Exploiting the Structure of Change. Machine Learning, 2000, 41(2): 153~174
43 Harries M, Horn K. Learning Stable Concepts in a Changing World. In: Selected Papers from the Workshop on Reasoning with Incomplete and Changing Information and on Inducing Complex Representations, 1996. 106~122
44 Gerencser L, Molnar-Saska G. Change detection of Hidden Markov Models. In: 43rd IEEE Conf on Decision and Control, 2004. 1754~1758
45 Keogh E, Kasetty S. On the Need for Time Series Data Mining Benchmarks: A Survey and Empirical Demonstration. Data Mining and Knowledge Discovery, 2003, 7(4): 349~371
46 Yamanishi K, Takeuchi J-I. A Unifying Framework for Detecting Outliers and Change Points from Non-Stationary Time Series Data. In: Proc of the 8th ACM SIGKDD Intl Conf on Knowledge Discovery and Data Mining, 2002. 676~681
47 Fan W. Systematic Data Selection to Mine Concept-Drifting Data Streams. In: Proc of the 10th ACM SIGKDD Intl Conf on Knowledge Discovery and Data Mining, 2004. 128~137
48 Aggarwal C C. A Framework for Diagnosing Changes in Evolving Data Streams. In: Proc of the 2003 ACM SIGMOD Intl Conf on Management of Data, 2003. 575~586
49 Batu T, et al. Testing That Distributions are Close. In: Proc of the 41st Annual Symposium on Foundations of Computer Science, 2000. 259~269
50 Kifer D, Ben-David S, Gehrke J. Detecting Change in Data Streams. In: Proc of the 30th VLDB Conf, 2004. 180~191
51 Wang H, Pei J. A Random Method for Quantifying Changing Distributions in Data Streams. The 9th European Conf on Principles and Practice of Knowledge Discovery in Databases (PKDD). 2005