流数据分析与管理综述*

金澈清, 钱卫宁, 周傲英+

(复旦大学 计算机科学与工程学系,上海 200433)

Analysis and Management of Streaming Data: A Survey

JIN Che-Qing, QIAN Wei-Ning, ZHOU Ao-Ying+

(Department of Computer Science and Engineering, Fudan University, Shanghai 200433, China)

+ Corresponding author: Phn: +86-21-65643503, Fax: +86-21-65643503, E-mail: ayzhou@fudan.edu.cn, http://www.fudan.edu.cn

Received 2003-09-01; Accepted 2004-03-29

Jin CQ, Qian WN, Zhou AY. Analysis and management of streaming data: A survey. *Journal of Software*, 2004,15(8):1172~1181.

http://www.jos.org.cn/1000-9825/15/1172.htm

Abstract: The study on streaming data is one of the hot topics among the database circle all over the world recently. During the past three decades, conventional database technologies have been well developed and widely applied. Unfortunately, they could not be adopted to handle a new kind of data, named streaming data, which is generated from applications such as network routing, sensor networking, stock analysis, etc. Because of the rapid data arriving speed and huge size of data set in stream model, novel algorithms that only require seeing the whole data set once are devised to support aggregation queries on demand. In addition, this kind of algorithms usually owns a data structure far smaller than the size of the whole data set. The ways to devise such synopsis data structures are introduced. These different approaches are also compared by listing historical works upon two classical problems over stream.

Key words: streaming data; synopsis data structure; landmark model; sliding window model

摘 要: 有关流数据分析与管理的研究是目前国际数据库研究领域的一个热点.在过去30多年中,尽管传统数据库技术发展迅速且得到了广泛应用,但是它不能够处理在诸如网络路由、传感器网络、股票分析等应用中所生成的一种新型数据,即流数据.流数据的特点是数据持续到达,且速度快、规模宏大;其研究核心是设计高效的单遍数据集扫描算法,在一个远小于数据规模的内存空间里不断更新一个代表数据集的结构——概要数据结构,使得在任何时候都能够根据这个结构迅速获得近似查询结果.综述国际上关于流数据的概要数据结构生成与维护的研究成果,并通过列举解决流数据上两个重要问题的各种方案来比较各种算法的特点以及优劣.

关键词: 流数据;概要数据结构;界标模型;滑动窗口模型

* Supported by the National High-Tech Research and Development Plan of China under Grant No.2002AA413310 (国家高技术研究发展计划(863))

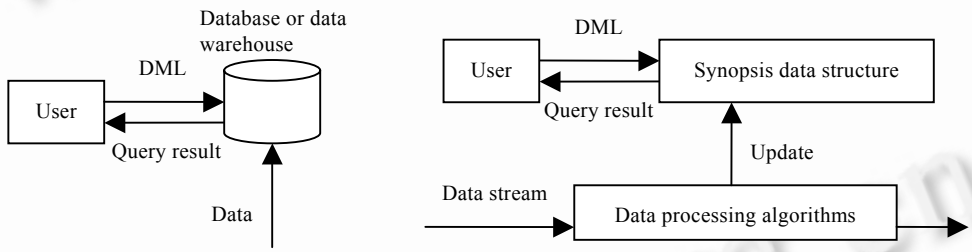
作者简介: **金澈清**(1977－),男,浙江文成人,博士生,主要研究领域为流数据管理,数据挖掘;**钱卫宁**(1976－),男,博士,讲师,主要研究领域为对等计算系统中的数据管理,流数据管理与挖掘,Web 数据管理;**周傲英**(1965－),男,博士,教授,博士生导师,主要研究领域为对等计算系统,流数据分析与处理,Web 数据管理与挖掘.

中图法分类号: TP311 文献标识码: A

30 多年来,数据库技术发展迅速且得到了广泛应用.一方面,数据建模形式多样,从层次数据库、网状数据库、关系数据库、对象数据库,直到关系对象数据库等等;另一方面,数据规模也越来越大.传统数据库技术的一个共同点是:数据存储在介质中,可以多次利用;用户提交数据操纵语言(data manipulation language,简称 DML)来获取查询结果.尽管传统数据库获得了巨大的成功,但是在 20 世纪末,一种新的应用模型却对它提出了有力的挑战.这种名为流数据(streaming data)[1]的应用模型广泛出现在众多领域,例如金融应用、网络监视、通信数据管理、Web 应用、传感器网络数据处理等等.

令 t 表示任一时间戳,a_t 表示在该时间戳到达的数据,流数据可以表示成$\{\ldots,a_{t-1},a_t,a_{t+1},\ldots\}$.区别于传统应用模型,流数据模型具有以下 4 点共性:(1) 数据实时到达;(2) 数据到达次序独立,不受应用系统所控制;(3) 数据规模宏大且不能预知其最大值;(4) 数据一经处理,除非特意保存,否则不能被再次取出处理,或者再次提取数据代价昂贵.利用传统技术处理这种模型,必须将数据全部存储到介质中,然后通过提交 DML 语句访问存储介质来获取查询结果.但是,由于数据规模宏大且到达速度很快,传统技术难以满足实时要求.

在很多实际应用中,例如决策支持系统、查询优化等,用户并不需要获得确切值,而仅需要一个近似值.因此,设计单遍扫描算法(one-pass algorithm),实时地给出近似查询结果就成为数据流模型下数据处理的目标.算法的关键在于设计一个远小于数据集规模的结构,从而可以在内存中处理数据.相对于数据流的规模而言,这种名为概要数据结构(synopsis data structure)的规模至多应该是次线性的.即如果流的长度为 N,则概要数据结构大小不超过 $O(\text{polylog}(N))$,并且处理流上每一组数据的时间不超过 $O(\text{polylog}(N))$[1].图 1 显示了传统数据处理技术和数据流处理技术的差异.



(a) Traditional data processing model (a) 传统数据处理模型 (b) Data stream processing model (b) 数据流处理模型

Fig.1 Comparison between the traditional data processing model and data stream processing model

图 1 传统数据处理模型和数据流处理模型的比较

从图 1 可以看出,传统的数据处理技术将所有数据存放到数据库或者数据仓库中;系统响应用户提交的 DML 语句,搜索数据存储媒介,返回查询结果.当数据规模很大时,数据往往以磁盘或者磁带为介质,因而执行查询操作需要大量的 I/O 交换,效率低下,不能适应实时系统的需求.相反,新的流数据处理技术并不保存整个数据集,仅维护一个远小于其规模的概要数据结构,从而能够常驻内存.流数据处理技术往往包含两部分算法,一部分监控流中的数据,更新概要数据结构;另一部分响应用户查询请求,返回近似查询结果.举例来说,假设一个电话公司想要获知某一天通话时长最大的地区以及通话的小时数.在传统方法下,首先所有通话记录被存放在外部介质中,然后进行分析,得到最终查询结果.由于数据规模宏大,需要大量的 I/O 交换,可能需要 10 分钟才能够得到结果:甲地区:43521.5 小时.而在数据流处理方法中,仅仅需要访问内存中的概要数据结构,因而可能仅仅需要 10 秒就能获得答案:甲地区:43500±50.0 小时.可以看出,数据流处理方法比传统的处理方法要快很多.尽管该方法得到的结果并不精确,但是往往并不影响用户的最终决策.

最近几年,流数据处理技术发展很快.一方面,出现了很多流数据模型下的管理系统,即数据流管理系统(data stream management system,简称 DSMS),包括斯坦福大学的 STREAM 项目[1]、施乐公司的 Tapestry 项目[2]、加州大学伯克力分校的 Telegraph 项目[3,4]、布朗大学和麻省理工学院合作的 Aurora 项目[5]等等,这些系统针对

具体行业背景,给出较全面的数据管理解决方案;另一方面,基于流数据模型的数据挖掘技术也得到了广泛的研究,包括做聚类分析[6]、决策树分析[7,8]、密度估计[9]等等.上述数据流研究的核心就是概要数据结构的设计.文献[1,10,11]从不同角度综述了流数据的研究进展.文献[1]侧重于如何构建 DSMS.文献[10]介绍了几种生成概要数据结构的方法,但是主要集中于作者自身的成果.文献[11]介绍了流数据模型下查询和挖掘的一些算法,但是未能涵盖最新的研究成果.本文试图克服以上文献的不足,较为全面地介绍构建高效的概要数据结构的各类算法.

数据流模型根据不同的时序范围可以划分成多种子模型,包括界标模型(landmark model)、滑动窗口模型(sliding window model)和快照模型(snapshot model).令 n 表示当前时间戳,s,e 分别是两个已知的时间戳.界标模型的查询范围从某一个已知的初始时间点到当前时间点为止,即$\{a_s,\ldots,a_n\}$.滑动窗口模型仅关心数据流中最新的 W(W 也称为滑动窗口大小)个数据,其查询范围是$\{a_{\max(n-W+1,0)},\ldots,a_n\}$,随着数据的不断到达,窗口中的数据也不断平移.快照模型则将操作限制在两个预定义的时间戳之间,表示为$\{a_s,\ldots,a_e\}$.界标模型和滑动窗口模型由于要不断处理新来的数据,更接近于真实应用,因而得到更加广泛的研究.

本文第 1 节和第 2 节分别介绍在界标模型和滑动窗口模型下生成概要数据结构的常用方法.第 3 节介绍两个数据流领域的重要问题——获取热门元素列表问题和求解中位点问题,列举不同的解决方案,并且比较这些方案的特点.第 4 节进行总结.

1 基于界标模型的方法

界标模型所要处理的数据范围从一个固定时间戳到当前时间戳.令初始时间戳为 s,当前时间戳为 n,则查询范围可以标记为$\{a_s,\ldots,a_n\}$.创建基于界标模型的概要数据结构,要求这个结构能够近似模拟这个数据集合的特征.直方图方法、抽样方法、小波方法、哈希方法等都是非常有效的手段.直方图方法能够有效地表示大数据集合轮廓,因此,构造各种直方图成为概要数据结构的一大选择.抽样方法从大数据集中选取部分数据,表征整个集合,方法简便.小波变换方法是一个重要的信号处理方法.它利用变换后生成的少数小波参数近似模拟原始信号.在数据流领域,对整个数据集进行小波变换,保存部分重要的小波参数,就能够近似模拟原先数据集合.哈希方法可以将大值域的数据集映射生成一个小值域内的目标数据集,研究所生成的目标数据集合的特性,能够推断出原始数据集的某些特征.本节将详细介绍如何用这 4 种方法构建概要数据结构.它们之间的比较,将在第 3 节结合两个经典的问题进行.

1.1 直方图(histogram)

直方图技术[12,13]就是将一个大数据集划分为很多个连续的桶(bucket),也就是小数据集,每个桶都由一个数字来代表其特征.直方图表示法直观、简洁,能够很好地表示大数据集的轮廓,因此在一些商业数据库中采用.直方图又可以划分成多种,例如等宽直方图(equi-width histogram)、压缩直方图(compressed histogram)、V-优化直方图(V-optimal histogram)等.

1.1.1 等宽直方图(equi-width histogram)

等宽直方图的目标是使各个桶的高度(即桶所含的数据量)比较平均.

文献[14]提出了一种单遍扫描算法.该算法包含两个重要的操作(拆分和合并)和两个门槛值(上限门槛和下限门槛).每次“看”到流中的元素,算法首先增加该元素所属的桶的高度.如果桶的高度超过上限门槛值,就调用拆分操作,将该桶等分为两个小桶;反之,如果桶的高度低于下限门槛值,就执行合并操作,将该桶和相邻的桶合并成一个新桶,使得新桶的高度介于上限门槛值和下限门槛值之间.

维护等宽直方图能够获得数据集的分位点[15].

1.1.2 压缩直方图(compressed histogram)

压缩直方图可以看成是等宽直方图的一个扩充.在等宽直方图中,各桶所包含的数据量比较接近.如果数据分布比较均匀,等宽直方图能够较好地模拟数据集.但是,一旦数据集中存在某些所占比例特别大的元素(这些元素也被称为热门元素),等宽直方图表示法就会产生较大的误差.文献[16]注意到了这个问题,提出了压缩直方

图表示法.压缩直方图法单独为那些热门元素创建桶,对其他元素仍然采用维护等宽直方图的方法,因而能够更真实地模拟数据集.

1.1.3 V-优化直方图(V-optimal histogram)

不同于以上两种直方图,V-优化直方图[17]的目标是使各桶的方差之和最小.假设数据集中各个元素的值为$v_1,v_2,\ldots,v_n$,将数据集划分成多桶之后,令 b_i 表示元素 v_i 所在桶的平均值.V-优化直方图的目标是使$\sum(v_i-b_i)^2$ 的值最小.文献[18]给出的基于动态规划的算法需要多次遍历数据集,且时间和空间复杂性均较大.文献[19]发展了文献[18]的算法,在数据集中的元素已经排过序的前提下,仅仅需要次线性的空间和时间复杂度就能够逼近最优解.文献[20]释放了文献[19]的前提,对于任意数据集,均可用次线性空间复杂度获得 V-优化直方图.V-优化直方图可以优化查询结果.

1.2 抽样方法(sampling)

抽样方法也是生成概要数据结构的常用手段.它从数据集中抽取小部分数据代表整个数据集,并根据该样本集合获得查询结果.抽样方法可以分成均匀抽样(uniform sampling)和偏倚抽样(biased sampling)两种.在均匀抽样方法中,数据集中各元素以相同的概率被选取到样本集合中;而在偏倚抽样方法中,不同元素的入选几率可能不同.水库抽样方法和精确抽样方法都属于均匀抽样方法,而计数抽样方法则属于偏倚抽样方法.下面介绍这几种抽样方法.

1.2.1 水库抽样(reservoir sampling)

水库抽样方法[21]单遍扫描数据集,生成均匀抽样集合.令样本集合的容量为 S,在任一时刻 n,数据流中的元素都以 S/n 的概率被选取到样本集合中去.如果样本集合大小超出 S,则从中随机去除一个样本.可以证明,各元素的入选几率相同.文献[21]同时推荐了一个技巧来提高算法效率:在原算法中,对于流中的每一个元素都需要“扔骰子”,判断该元素是否以 S/n 概率被选中;改进的算法转而判断一次需要略过多少个后续元素,从而大大减少了扔骰子的次数.

1.2.2 精确抽样(concise sampling)

在水库抽样方法中,样本集合中各个元素单独占据一个位置,即使它们具有相同的数值,因而表达的效率并不高.举例来说,假设元素 1 出现了 8 次,元素 2 出现了 1 次,则样本集合被表示为(1,1,1,1,1,1,1,1,2,…).精确抽样方法[22]改进了样本集合的表示方法.对于仅出现一次的元素,类似于水库抽样,仍然用元素代码表示;而对于多次出现的元素,则利用结构⟨*value*,*count*⟩表示,*value* 代表元素代码,*count* 表示样本集合中该元素的数目.这样,在精确抽样中,上面样本集合就表示为(⟨1,8⟩,2,…).很明显,精确抽样方法比水库抽样方法更节约空间.精确抽样算法维护一个初始值为 1 的概率参数 T,各元素以概率 $1/T$ 加入到样本集合中去.如果该元素已经存在于样本集合中,则相应的计数器加 1(对于仅出现一次的元素,需要改由结构⟨*value*,*count*⟩进行表示,且 *count* 值为 2);否则,将该元素添加到样本集合中去.一旦样本集合溢出,改变参数 T 到 T',$T'>T$.样本集合中的各个元素均以概率 T/T' 被删除,从而腾出空间以便存放新数据.精确抽样方法通过逐步提高参数 T 的值,实现数据流上的均匀抽样.

1.2.3 计数抽样(counting sampling)

计数抽样方法[22]是精确抽样方法的一个变种.二者的区别在于样本集合溢出时如何处理.在计数抽样算法中,当样本集合溢出时,首先将参数 T 提高到 T'.对于其中的任意一个元素,都是首先以概率 T/T',之后以概率 $1/T'$ 判断是否减去 1.一旦该计数器值已经降为 0,或者某一次随机判断之后计数器的值并没有减小,则结束对该元素的操作.计数抽样方法不是均匀抽样方法,但却能有效地获得数据集中的热门元素列表.

1.3 小波方法(wavelet)

小波分析方法是一种通用的数字信号处理技术.类似于傅立叶变换,小波分析根据输入的模拟量,变换成一系列的小波参数,并且少数几个小波参数就拥有大部分能量[23].根据这个特性,可以选择少数小波参数,近似还原原始信号.小波分析方法也被应用到数据库领域,例如对高维数据进行降维处理、生成直方图等.小波种类很多,最常见且最简单的是哈尔小波(Haar wavelet).从而可以估算任一元素的数值或者任一范围之和(range sum),即某一区间内所有元素之和.

文献[24]提出了一种基于哈尔小波技术,在数据流上生成直方图的算法.该算法将整个数据集变换成一系列的小波参数,并且有选择地保留有限个高能量参数,从而近似模拟原始数据集.文献[25]改进了该算法.新算法能够同时支持数据的插入操作和删除操作.然而,上述两种算法均不能在理论上保证误差的范围.在文献[26]中,作者观察到对于时序数据而言,只需要保存最多 $\log N$ 个计数器,就能够获得任一时刻的小波参数.这意味着,如果流中元素已经排好序,则仅需 $O(B+\log N)$的存储空间,就能够获得 B 个最大的小波参数.文献[27]提出的算法则能够以概率 $1-\delta$保证结果的误差在一个小范围之内,其中,δ是一个接近于 0 的用户定义参数.

1.4 哈希方法(Hash)

定义一组哈希函数,将数据从一个范围映射到另一个范围中去,是计算机领域的一个常用手段.本节介绍 3 种利用哈希函数生成概要数据结构的方法:Bloom Filter 方法、Sketch 方法和 FM 方法.

1.4.1 Bloom Filter 方法

Bloom Filter 方法[28]自从 1970 年被提出来之后,就广泛应用于网络[29]、数据库[30]、P2P 系统[31]等众多领域.该算法的最大特点是,仅使用一小块远小于数据集数据范围的内存空间表示数据集,并且各个数据仍然能被区分开来.假设所申请的内存大小为 m 比特位,该方法创建 h 个相互独立的哈希函数,能将数据集均匀映射到[1..m]中去.对任何元素,利用哈希函数进行计算,得到 h 个[1..m]之间的数,并将内存空间中这 h 个对应比特位都置为 1.这样,就可以通过检查一个元素经过 h 次哈希操作后,是否所有对应的比特位都被置 1 来判断该元素是否存在.这种判断方法可能会产生错误——虽然某元素并不存在,但是它所对应的 h 个比特位都已经被其他元素所设置了,从而导致被误认为存在.然而,这种错误发生的概率随着内存的增加可以非常小.文献[32]改进了传统的方法,在每一个位置上都用计数器代替比特位,从而不仅能够判断元素是否存在,而且能够估算元素的值.

1.4.2 Sketch 方法

Sketch 方法能够解决流上的很多问题,例如,估计数据集的二阶矩大小[33]、估计数据集自连接的大小[34]、获得数据集中热门元素的列表*[35]等.下面以估算数据集的二阶矩为例,详细说明算法的步骤.令 m_i 表示一个数据集中元素 i 的个数,则数据集合的二阶矩 $F_2=\sum m_i^2$.

假设 Z 是算法中一个初始值为 0 的计数器,ξ是一个能够将流中各元素均匀映射到{−1,1}的哈希函数.对于在时刻 t 流经的元素 a_t,均修改计数器 Z 值:$Z=Z+\xi(a_t)$.在不同时刻到达的相同元素对计数器的更新方法是一致的.算法可以在任一时刻 n 获得最终的估算值.令 m_i 表示其时元素 i 的累积频数,可以看出,$Z=\sum m_i\xi(i)$.$Z^2=\sum m_i^2+\sum m_i m_j\xi(i)\xi(j)$.如果哈希函数$\xi$对各元素独立,则 $E(Z^2)=\sum m_i^2$.试举例加以说明.假设数据流上经过的元素都是[1..5]之间的整数,某一个哈希函数能够将数从{1,2,3,4,5}分别映射为{1,−1,1,−1,−1};流上经过的 10 个数分别是{1,5,3,2,4,2,5,1,2,4},则各元素的个数{m_i}分别是{2,3,1,2,2}.最后,计数器 Z 的值为−4,Z^2=16;真实值 F_2=22.为了降低误差,设置了多组计数器以及多组相互独立的哈希函数,在所有估计值中取中值为估计值,从而在理论上保证误差不超过预定义值.

Indyk 充分利用了 p-stable 分布的特性,扩展了 Sketch 方法[36,37],可以估算出数据集的 p 阶矩大小,其中 $0<p<2$.

1.4.3 FM(Flajolet-Martin)方法

FM 方法[38]是求解数据集中不相同元素的个数(即 F_0)的有力手段.它所采用的哈希函数(least significant 1 bit,简称 LSB)将一个大小为 M 的数据集映射到范围[0..log$M$−1]中去,且映射到 i 的概率是 $1/2^{i+1}$.假设不相同元素的个数是 D,且哈希函数独立随机,则恰有 $D/2^{i+1}$ 个不同元素映射到 i.这个性质可以用于估计 D 的值.文献[39]扩展了 FM 方法,在对多个数据流做复杂的集合操作之后,能够得到结果集合上的不相同元素的个数.

2 基于滑动窗口模型的方法

假设窗口的大小是 W,在任一时间点 n,滑动窗口模型的查询范围是{$a_{\max(0,n-W+1)},\ldots,a_n$}.时间点

* 关于热门元素列表的详细定义参见第 3 节.

max(0,$n-W+1$)之前的数据全部忽略不计.在滑动窗口模型下构造概要数据结构比在界面模型下更具挑战性的原因在于,不仅新数据不断到达,而且旧数据会过期.因此,如何处理过期数据,使得查询结果一直可靠,就成为一大难题.目前的研究成果主要有指数直方图技术、基本窗口技术和链式抽样技术.

2.1 指数直方图(exponential histogram)

指数直方图[40]技术是最早用来生成基于滑动窗口模型的概要数据结构的方法.传统的直方图技术将数据集划分成多个桶,相邻桶的元素值连续.而指数直方图则是按照元素的到达次序构建桶.桶的容量按照不同级别呈指数递增,从小到大分别是 1,2,4,8,…各个级别桶的个数均不超过一个预定义的门槛值.每“看到”流中的一个元素,视应用需求就决定是否创建一个最低级别的桶.例如,文献[40]统计 1 的个数时,仅当元素值为 1 时才创建新桶;文献[41]在维护方差时,则对于每个数都创建新桶.桶中除了用来表征属于本桶的元素的数值之外,还包含一个时间戳,代表桶所包含的元素中最“旧”元素的时间戳.如果某级别桶数目超过预设门槛值,则合并本级别一对最早生成的桶,创建一个高级别桶.桶合并操作可能会导致更高级别桶的连锁合并操作.举例来说,假设各级别桶的个数最多是 5 个,在某一时间点,指数直方图已经有 12 个桶,第 1 级别和第 2 级别都已经达到最大值,第 3 级别的桶有两个,各桶的级别可以表示成{1,1,1,1,1,2,2,2,2,2,3,3}.此时,如果创建一个第 1 级别的桶,则第 1 级别桶的个数会溢出,需要合并两个最“旧”的该级别桶,生成一个第 2 级别的桶;这个操作又导致第 2 级别的桶的个数溢出,需进行合并以生成一个第 3 级别的桶.最后,所有桶的级别是{1,1,1,1,2,2,2,2,3,3,3}.指数直方图仅维护还未过期的桶,一旦最“旧”的桶所有元素都过期了,就删除该桶,释放其所占空间.

指数直方图能够解决滑动窗口模型下的很多问题,例如基本计数(basic counting)问题、求和问题[40]、方差问题[41]等.文献[42]提出的算法类似于指数直方图,并具有更高的效率.

2.2 基本窗口(basic window)

基本窗口技术[43]将大小为 W 的窗口按照时间次序划分成 k 个等宽的子窗口,称为基本窗口,每个基本窗口包含 W/k 个元素,且由一个小结构表示基本窗口的特征.如果窗口所包含的元素均已过期,则删除表征这个基本窗口的小结构.用户可以基于这些未过期的小结构得到查询结果.

文献[43]采用这种方法,快速地从众多股票中得到相关的几支股票.这种方法还可以用于获得数据集中的热门元素列表[44].

2.3 链式抽样(chain-sampling)

链式抽样方法[45]能够获得在滑动窗口上均匀抽样的样本集合.假设窗口大小是 W,则在任何时间点 n,流中的元素以概率 $1/\min(n,W)$被添加到样本集合中去.当元素被选择到样本集合中去时,必须同时决定一个备选元素,以便于当这个元素过期时,利用备选元素代替该元素.由于在数据流中不能够预测将来的数据,因此,实际上仅从[$n+1\ldots n+W$]中随机选取一个数作为备选元素的时间戳 t.当到达时间点 t 时,这个备选元素才最终被确定.备选元素以后也会过期,因此也需要为它选择一个备选元素,方法同上.可以看出,样本集合中的任一元素,均有一个备选元素的“链”,元素过期后,马上用“链”上的下一个元素取代它.

3 应用例子

前面两节介绍了在界标模型下和滑动窗口模型下生成概要数据结构的多种方法.在界标模型下,主要有直方图、抽样、小波、哈希等方法;在滑动窗口模型下,有指数直方图、基本窗口和链式抽样等方法.本节以两个数据流上的热门研究问题为例,列举解决这两个问题的不同方法,并且比较这些方法的优劣.

(1) 在流上挖掘热门元素.根据用户定义的门槛参数$s\in(0,1)$,输出在整个数据流中所占比重大于s的所有元素.很多应用需要用到这个信息.例如,网络路由器需要监控比例特别高的 IP 包,防止 DOS 攻击;对于决策支持系统而言,了解数据集中哪些元素关系密切也非常重要.

(2) 在流上挖掘分位点.令集合大小是 N,参数$\phi\in(0,1)$,分位点元素就是数据集排序之后第ϕN 位置的元素.分位点是数据集合的一个重要统计量.获得分位点有助于优化查询计划,提高数据库系统的性能.

表 1 和表 2 分别是这两个问题的研究进展.

Table 1 The research progress of mining frequent items over stream

表 1 在流上挖掘热门元素的研究进展

References	Year	Technique	Whether support deletion	Whether the error bounded	Randomized or deterministic	Space requirement
[22]	1998	Counting sampling	No	No	—	$O(k)$
[46]	2002	Sampling	No	Yes	Yes	$O(\varepsilon^{-1})$
[47]	2002	Sampling	No	Yes	Yes	$\Omega(\varepsilon^{-1}\log(\varepsilon n))$
[35]	2002	Hash (Sketch)	Yes	Yes	No	$\Omega(k/\varepsilon^2\log n)$
[48]	2003	Hash	Yes	Yes	No	$O(k(\log k+\log 1/\delta)\log M)$
[49]	2003	Hash	Yes	Yes	No	$O(\varepsilon^{-1}\log(-M/\log\rho))$
[44]	2003	Basic window	No	No	—	$O(N/b)$

Table 2 The research progress of mining quantile over stream

表 2 在流上挖掘分位点的研究进展

References	Year	Technique	Whether support deletion	Whether the error bounded	Randomized or deterministic	Space requirement
[50]	1998	Sampling	No	Yes	Yes	$O(\varepsilon^{-1}\log^2(\varepsilon N))$
[51]	1999	Sampling	No	Yes	No	$O(\varepsilon^{-1}\log^2(\varepsilon^{-1}\log^2\log\delta^{-1}))$
[15]	2001	Equi-Width histogram	No	Yes	Yes	$O(\varepsilon^{-1}\log(\varepsilon N))$
[52]	2002	Hash	Yes	Yes	No	$O(\log^2\|U\|\log(\log(\|U\|)/\delta)/\varepsilon^2)$

从表 1 和表 2 可以看出,解决这两个问题的方法很多,包括抽样、哈希、直方图等.不同方案具有不同的性能指标.下面分别从 4 个性能指标比较不同的算法,包括是否支持删除操作、是否保证误差范围、是否为确定性算法、空间复杂度如何等.

数据流模型上的某些应用同时具有插入和删除两种操作.例如,一种防止网络的 DOS 攻击的方法是,监控通过路由器的网络连接,查看从哪里发起的连接流量比较高.这就需要分别在连接建立和断开的时候插入新记录和删除原有记录.抽样方法仅仅保留数据集的部分元素,丢失了其他元素的信息.当要删除的元素不在样本集合中时,往往引入误差,而且这种误差随着时间的推移很可能累积.因此,抽样方法不适合这种应用.文献[46,47]的抽样算法在仅有插入操作的模型下性能良好,但是,修正算法在有删除操作的模型下性能下降很多[48].相反,基于哈希方法的方案[35,48,49,52]能够支持数据流的删除操作.在哈希方案中,往往存在一批计数器,当"看到"流中的元素时,就按照一定的规则,对相关计数器进行操作.插入操作和删除操作的区别在于,对计数器的操作是相反的.因此,作用于同一个元素上的一对插入和删除事务可以不引入任何误差,从而保证算法的性能.

虽然数据流算法只能返回近似查询结果,但是大部分算法都能将其误差限制在一个预定义的小范围之内.从表 1 和表 2 可以看出,抽样方法、直方图方法、哈希方法都能够做到这一点.小波方法虽然没有被用来解决这两个问题,但是文献[27]提出的算法同样能够保证查询结果的误差范围.

这些能够在理论上保证误差范围的算法又可以分为确定性算法和非确定性算法两种.确定性算法所得到的查询结果在任何情况下都是可信的;非确定性算法只能保证结论正确的概率很高,给定足够的空间,其概率值可以接近于 1.在哈希方案[35,48,49,52]中,由于所使用的哈希函数的参数是随机生成的,且往往取多个计数器的均值或者中值作为查询结果,因此,所设计出来的算法一般都是非确定性算法.均匀抽样方法由于要保证数据集中各元素以相同概率入选样本集合,并从样本集合得到最终结论,算法的查询结果往往是非确定性的.而偏倚抽样方法(例如文献[46,47])和直方图[15]方法都能设计成确定性的算法.设计基于小波方法的确定性算法是一个难题.文献[27]给出了一个非确定性的算法.

不同算法的空间复杂度区别也很大.从表 1 和表 2 可以看出,抽样方法和直方图方法所需要的空间比较小,对误差ε的变化也不是很敏感,大部分[15,46,47,50,51]都近似正比于ε^{-1}.哈希方法的空间需求则与具体方法密切相关文献[48,49]的空间需求较小,但是文献[35,52]所需用到的空间就比较大,大致正比于ε^{-2}.

在滑动窗口模型下有效解决这两个经典问题仍然是非常困难的.目前唯一的工作体现在文献[44]中.我们期待其他方案的出现.

4 总 结

传统数据库技术在 20 世纪得到了非常成功的发展.但是在一种名为数据流的模型中,数据持续到达,且速度快、规模宏大.传统技术由于时间、空间复杂度高,难以对这种应用模型进行有效处理,亟需新的研究方法来解决.针对这种新型应用模型,一种新的思路就是设计高效的单遍数据集扫描算法,在一个远小于数据规模的内存空间里不断更新一个代表数据集的结构——概要数据结构,从而实时、高效地获得近似查询结果.

本文回顾了最近几年来国际上在该领域的主要研究成果,综述了在数据流模型下(包括界标模型和滑动窗口模型)构造概要数据结构的各种方法.界标模型下的做法主要有直方图方法、抽样技术、小波技术和哈希技术等等.滑动窗口模型下的做法主要有指数直方图、基本窗口技术、链式抽样等方法.各种方法各具特点,各有优劣.我们同时以在流上挖掘热门元素和在流上挖掘分位点这两个非常重要的统计量为例子,对各种方法进行了比较.

References:

[1] Babcock B, Babu S, Datar M, Motwani R, Widom J. Models and issues in data streams. In: Popa L, ed. Proc. of the 21st ACM SIGACT-SIGMOD-SIGART Symp. on Principles of Database Systems. Madison: ACM Press, 2002. 1~16.

[2] Terry D, Goldberg D, Nichols D, Oki B. Continuous queries over append-only databases. SIGMOD Record, 1992,21(2):321~330.

[3] Avnur R, Hellerstein J. Eddies: Continuously adaptive query processing. In: Chen W, Naughton JF, Bernstein PA, eds. Proc. of the 2000 ACM SIGMOD Int'l Conf. on Management of Data. Dallas: ACM Press, 2000. 261~272.

[4] Hellerstein J, Franklin M, Chandrasekaran S, Deshpande A, Hildrum K, Madden S, Raman V, Shah MA. Adaptive query processing: Technology in evolution. IEEE Data Engineering Bulletin, 2000,23(2):7~18.

[5] Carney D, Cetinternel U, Cherniack M, Convey C, Lee S, Seidman G, Stonebraker M, Tatbul N, Zdonik S. Monitoring streams—A new class of DBMS applications. Technical Report, CS-02-01, Providence: Department of Computer Science, Brown University, 2002.

[6] Guha S, Mishra N, Motwani R, O'Callaghan L. Clustering data streams. In: Blum A, ed. The 41st Annual Symp. on Foundations of Computer Science, FOCS 2000. Redondo Beach: IEEE Computer Society, 2000. 359~366.

[7] Domingos P, Hulten G. Mining high-speed data streams. In: Ramakrishnan R, Stolfo S, Pregibon D, eds. Proc. of the 6th ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining. Boston: ACM Press, 2000. 71~80.

[8] Domingos P, Hulten G, Spencer L. Mining time-changing data streams. In: Provost F, Srikant R, eds. Proc. of the 7th ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining. San Francisco: ACM Press, 2001. 97~106.

[9] Zhou A, Cai Z, Wei L, Qian W. M-Kernel merging: Towards density estimation over data streams. In: Cha SK, Yoshikawa M, eds. The 8th Int'l Conf. on Database Systems for Advanced Applications (DASFAA 2003). Kyoto: IEEE Computer Society, 2003. 285~292.

[10] Gibbons PB, Matias Y. Synopsis data structures for massive data sets. In: Tarjan RE, Warnow T, eds. Proc. of the 10th Annual ACM-SIAM Symp. on Discrete Algorithms. Baltimore: ACM/SIAM, 1999. 909~910.

[11] Garofalakis M, Gehrke J, Rastogi R. Querying and mining data stream: you only get one look—A tutorial. In: Franklin MJ, Moon B, Ailamaki A, eds. Proc. of the 2002 ACM SIGMOD Int'l Conf. on Management of Data. Madison: ACM Press, 2002. 635.

[12] Kooi RP. The optimization of queries in relational databases [Ph.D. Thesis]. Cleveland: Case Western Reserve University, 1980.

[13] Piatetsky-Shapiro G, Connell C. Accurate estimation of the number of tuples satisfying a condition. SIGMOD Record, 1984,14(2): 256~276.

[14] Gibbons PB, Matias Y, Poosala V. Fast incremental maintenance of approximate histograms. In: Jarke M, Carey MJ, Dittrich KR, Lochovsky FH, Loucopoulos P, Jeusfeld MA, eds. VLDB'97, Proc. of the 23rd Int'l Conf. on Very Large Data Bases. Athens: Morgan Kaufmann, 1997. 466~475.

[15] Greenwald M, Khanna S. Space-Efficient online computation of quantile summaries. In: Proc. of the 2001 ACM SIGMOD Int'l Conf. on Management of Data. Santa Barbara: ACM Press, 2001. 58~66.

[16] Poosala V, Ioannidis Y, Haas P, Shekita E. Improved histograms for selectivity estimation of range predicates. SIGMOD Record, 1996,25(2):294~305.

[17] Ioannidis Y, Poosala V. Balancing histogram optimality and practicality for query result size estimation. SIGMOD Record, 1995, 24(2):233~244.

[18] Jagadish HV, Koudas N, Muthukrishnan S, Poosala V, Sevcik K, Suel T. Optimal histograms with quality guarantees. In: Gupta A, Shmueli O, Widom J, eds. VLDB'98, Proc. of the 24th Int'l Conf. on Very Large Data Bases. New York: Morgan Kaufmann, 1998. 275~286.

[19] Guha S, Koudas N, Shim K. Data-Streams and histograms. In: Yannakakis M, ed. Proc. of the 33rd Annual ACM Symp. on Theory of Computing. Heraklion: ACM Press, 2001. 471~475.

[20] Gilbert A, Guha S, Indyk P, Kotidis Y, Muthukrishnan S, Strauss M. Fast, small-space algorithms for approximate histogram maintenance. In: Reif JH, ed. Proc. of the 34th Annual ACM Symp. on Theory of Computing. Montréal: ACM Press, 2002. 389~398.

[21] Vitter JS. Random sampling with a reservoir. ACM Trans. on Mathematical Software, 1985,11(1):37~57.

[22] Gibbons PB, Matias Y. New sampling-based summary statistics for improving approximate query answers. In: Haas LM, Tiwary A, eds. SIGMOD 1998, Proc. of the ACM SIGMOD Int'l Conf. on Management of Data. Seattle: ACM Press, 1998. 331~342.

[23] Jawerth B, Sweldens W. An overview of wavelet based multiresolution analyses. SIAM Review, 1994,36(3):377~412.

[24] Matias Y, Vitter JS, Wang M. Wavelet-Based histograms for selectivity estimation. In: Haas LM, Tiwary A, eds. SIGMOD 1998, Proc. of the ACM SIGMOD Int'l Conf. on Management of Data. Seattle: ACM Press, 1998. 448~459.

[25] Matias Y, Vitter JS, Wang M. Dynamic maintenance of wavelet-based histograms. In: Abbadi AE, Brodie ML, Chakravarthy S, Dayal U, Kamel N, Schlageter G, Whang KY, eds. VLDB 2000, Proc. of the 26th Int'l Conf. on Very Large Data Bases. Cairo: Morgan Kaufmann, 2000. 101~110.

[26] Gilbert AC, Kotidis Y, Muthukrishnan S, Strauss MJ. Surfing wavelets on streams: One-Pass summaries for approximate aggregate queries. In: Apers PMG, Atzeni P, Ceri S, Paraboschi S, Ramamohanarao K, Snodgrass RT, eds. VLDB 2001, Proc. of the 27th Int'l Conf. on Very Large Data Bases. Roma: Morgan Kaufmann, 2001. 79~88.

[27] Garofalakis M, Gibbons PB. Wavelet synopses with error guarantees. In: Franklin MJ, Moon B, Ailamaki A, eds. Proc. of the 2002 ACM SIGMOD Int'l Conf. on Management of Data. Madison: ACM Press, 2002. 476~487.

[28] Bloom B. Space/time tradeoffs in hash coding with allowable errors. Communications of the ACM, 1970,13(7):422~426.

[29] Fan L, Cao P, Almeida J, Broder AZ. Summary cache: A scalable wide-area Web cache sharing protocol. IEEE/ACM Trans. on Networking, 2000,8(3):281~293.

[30] Mullin JK. Optimal semijoins for distributed database systems. IEEE Trans. on Software Engineering, 1990,16(5):558~560.

[31] Marais H, Bharat K. Supporting cooperative and personal surfing with a desktop assistant. In: UIST'97. Proc. of the 10th Annual ACM Symp. on User Interface Software and Technology. Banff: ACM Press, 1997. 129~138.

[32] Cohen S, Matias Y. Spectral bloom filters. In: Halevy AY, Ives ZG, Doan AH, eds. Proc. of the 2003 ACM SIGMOD Int'l Conf. on Management of Data. San Diego: ACM Press, 2003. 241~252.

[33] Alon N, Matias Y, Szegedy M. The space complexity of approximating the frequency moments. In: Miller G, ed. Proc. of the 28th Annual ACM Symp. on the Theory of Computing. Philadelphia: ACM Press, 1996. 20~29.

[34] Alon N, Gibbons PB, Matias Y, Szegedy M. Tracking join and self-join sizes in limited storage. In: Proc. of the 18th ACM SIGACT-SIGMOD-SIGART Symp. on Principles of Database Systems. Philadelphia: ACM Press, 1999. 10~20.

[35] Charikar M, Chen K, Farach-Colton M. Finding frequent items in data streams. Theoretical Computer Science, 2004,312(1):3~15.

[36] Indyk P. Stable distributions, pseudorandom generators, embeddings and data stream computation. In: Blum A, ed. The 41st Annual Symp. on Foundations of Computer Science, FOCS 2000. Redondo Beach: IEEE Computer Society, 2000. 189~197.

[37] Cormode G. Stable distributions for stream computations: It's as easy as 0, 1, 2. 2003. http://www.research.att.com/conf/mpds2003/schedule/cormode.ps

[38] Flajolet P, Martin GN. Probabilistic counting algorithms for data base applications. Journal of Computer and Systems Sciences, 1992,31(2):182~209.

[39] Ganguly S, Garofalakis M, Rastogi R. Processing set expressions over continuous update streams. In: Halevy AY, Ives ZG, Doan AH, eds. Proc. of the 2003 ACM SIGMOD Int'l Conf. on Management of Data. San Diego: ACM Press, 2003. 265~276.

[40] Datar M, Gionis A, Indyk P, Motwani R. Maintaining stream statistics over sliding windows. In: Eppstein D, ed. Proc. of the 13th Annual ACM-SIAM Symp. on Discrete Algorithms. San Francisco: ACM/SIAM, 2002. 635~644.

[41] Babcock B, Datar M, Motwani R, O'Callaghan L. Maintaining variance and k-Medians over data stream windows. In: Neven F, ed. Proc. of the 22nd ACM SIGACT-SIGMOD-SIGART Symp. on Principles of Database Systems. San Diego: ACM Press, 2003. 234~243.

[42] Gibbons PB, Tirthapura S. Distributed streams algorithms for sliding windows. In: SPAA 2002: Proc. of the 14th Annual ACM Symp. on Parallel Algorithms and Architectures. Winnipeg: ACM Press, 2002. 63~72.

[43] Zhu Y, Shasha D. StatStream: Statistical monitoring of thousands of data streams in real time. In: Bernstein P, Ioannidis Y, Ramakrishnan R, eds. Proc. of the 28th Int'l Conf. on Very Large Data Bases. Hong Kong: Morgan Kaufmann, 2002. 358~369.

[44] DeHaan D, Demaine ED, Golab L, Lopez-Ortiz L, Munro JI. Towards identifying frequent items in sliding windows. Technical Report, CS-2003-06, Waterloo: University of Waterloo, 2003.

[45] Babcock B, Datar M, Motwani R. Sampling from a moving window over streaming data. In: Eppstein D, ed. Proc. of the 13th Annual ACM-SIAM Symp. on Discrete Algorithms. San Francisco: ACM/SIAM, 2002. 633~634.

[46] Demaine E, L´opez-Ortiz A, Munro JI. Frequency estimation of Internet packet streams with limited space. In: Möhring RH, Raman R, eds. Algorithms—ESA 2002, Proc. of the 10th Annual European Symp. Rome: Springer-Verlag, 2002. 348~360.

[47] Manku GS, Motwani R. Approximate frequency counts over data streams. In: Bernstein P, Ioannidis Y, Ramakrishnan R, eds. Proc. of the 28th Int'l Conf. on Very Large Data Bases. Hong Kong: Morgan Kaufmann, 2002. 346~357.

[48] Cormode G, Muthukrishnan S. What's hot and what's not: Tracking most frequent items dynamically. In: Proc. of the 22nd ACM SIGACT-SIGMOD-SIGART Symp. on Principles of Database Systems. San Diego: ACM Press, 2003. 296~306.

[49] Jin C, Qian W, Sha C, Yu JX, Zhou A. Dynamically maintaining frequent items over a data stream. In: Carbonell J, ed. Proc. of the 2003 ACM CIKM Int'l Conf. on Information and Knowledge Management. New Orleans: ACM Press, 2003. 287~294.

[50] Manku GS, Rajagopalan S, Lindsay BG. Approximate medians and other quantiles in one pass and with limited memory. In: Haas LM, Tiwary A, eds. SIGMOD 1998, Proc. of the ACM SIGMOD Int'l Conf. on Management of Data. Seattle: ACM Press, 1998. 426~435.

[51] Manku GS, Rajagopalan S, Lindsay BG. Random sampling techniques for space efficient online computation of order statistics of large datasets. In: Delis A, Faloutsos C, Ghandeharizadeh S, eds. SIGMOD 1999, Proc. of the ACM SIGMOD Int'l Conf. on Management of Data. Philadelphia: ACM Press, 1999. 251~262.

[52] Gilbert AC, Kotidis Y, Muthukrishnan S, Strauss MJ. How to summarize the universe: Dynamic maintenance of quantiles. In: Bernstein P, Ioannidis Y, Ramakrishnan R, eds. Proc. of the 28th Int'l Conf. on Very Large Data Bases. Hong Kong: Morgan Kaufmann, 2002. 454~465.