数据流复杂查询处理的研究

魏定国[1,2] 吴时霖[2]

(复旦大学计算机科学与工程系 上海200433)[1] (广东商学院 广州510320)[2]

摘 要 在连续的数据流上提供查询的应答对很多应用环境来说是一个极为重要的需求。本文主要探索了如何使用有限的内存在数据流上进行聚集SQL查询,以获得近似的结果。使用随机草图技术,计算非常小的数据流草图,以获得聚集查询的近似结果,并保证误差能在一定的范围之内。并讨论了在草图方法中如何利用已有的直方图统计信息来提高应答的质量。其关键的思想就是对属性域进行智能化的划分,分解草图化问题,确保所获得查询的结果具有合适的近似精度。不论从理论还是实验上都可以证明草图提供的聚集查询结果比传统的直方图更有效、更精确。

关键词 聚集查询,数据流,草图

Research on Process of Complex Queries over Data Stream

WEI Ding-Guo[1,2] WU Shi-Ling[1]

(Department of Computer Science, Fudan University, Shanghai 200433)[1] (Guangdong Commercial College, Guangzhou 510320)[2]

Abstract Providing answers to queries over continuous data streams is a very important requirement for many application environments. In this paper, we explore primarily how to obtain approximate results of aggregate SQL queries over data streams with limited memory. By utilizing randomizing techniques to compute very small sketch synopses of the streams, approximate answers can be provided to aggregate queries with provable guarantees on the approximation error. We also discuss how existing statistical information based on histograms can be used in the sketch method to improve the quality of the answers. The key idea is to intelligently partition the domain of the attributes, decompose the sketching problem and obtain the results of the queries with reasonable guarantees on the quality of approximation. In theory as well as experiment, it has proved that sketches provide significantly more accurate and effective answers of aggregate queries compared to traditional histograms.

Keywords Aggregate query, Data stream, Sketch

1 引言

传统的数据库管理系统(DBMS)是建立在不变的数据集上,将数据可靠地存放在稳定的存储器之中,在数据的整个生命期中更新和查询的次数都是非常有限的。然而现在很多的应用中,数据的达到和需要的处理都是每天24小时连续不断的,很难用静态的、不变的数据集的处理方法来处理。例如大型电信和IP网络装置,来自网络的不同部分的各种数据需要不断收集和分析;在零售链锁、银行ATM和信用卡的业务处理以及股票行情的自动收集和万维服务器的日志记录等事务处理中都会产生快速、连续的大规模的数据流。在大多数的应用中,数据流实际上是累加、存放在数据库管理系统之中,对这些数据进行存取、分析,在线做出决策、推断出相关的结论(如欺骗的识别)都是非常重要的,但开销巨大。正因为如此,最近几年处理连续数据流的算法成为了一个研究热点。连续数据流上查询处理的两个关键参数是:(1)查询算法需要的内存;(2)查询处理需要的时间。前者构成了数据流处理算法设计的一个重要的约束,因为在通常的数据流环境下,都只有有限的内存资源可以用来运行查询处理算法。因而要求算法必须以简明方式将数据流概括成草图,并不需要非常精确。草图能存放在较小的内存之中,通常能够给用户查询提供近似的答案,并能保证合适的近似质量。这种近似的、在线查询的应答方式特别适合大多数的数据流处理,如趋势分析、犯罪和异常检测,其目的是要识别各种可能的问题,而不需要提供精确的结果。

本文基本解决了使用有限的内存在连续数据流上的复杂聚集查询的近似处理所涉及到的棘手的技术问题。

2 数据流和随机草图

2.1 数据流处理模型简介

图1为基于草图的数据流查询处理架构。Q 为关系 R_1,…,R_n 关系集上的任意一个SQL查询,$|R_i|$ 为 R_i 中的元组的个数。与传统的DBMS查询处理器不同的是:数据流查询处理器引擎对于 R_1,…,R_n 关系中的数据元组只可能看到一次;各种数据源中的数据流都是按固定的顺序流入的(由到达的数据流模式决定);在数据流上回溯或显式地存取过去的元组都是不可能的;每一个关系 R_i 的元组的到达顺序是任意的,而且重复的元组可以发生在 R_i 流的期间的任何时刻。

该数据流查询处理引擎只需要一定数量的内存空间,通常比整个数据集要小得多。该内存用来保存和维护表示每一个数据流 R_i 的简洁而精确的草图 $S(R_i)$。对每一个草图 $S(R_i)$ 都要求:

魏定国 副教授,博士研究生,主要研究方向为计算机网络协议与分布式软件、数据库系统。**吴时霖** 教授,博士生导师,主要研究方向为计算机网络协议与分布式软件、Petri网理论及应用。

(1)大大小于数据流 R_i 中的元组的数量(例如它的大小是$|R_i|$的对数或多重对数);

(2)R_i 中数据元组无论按什么顺序达到都能计算一遍。

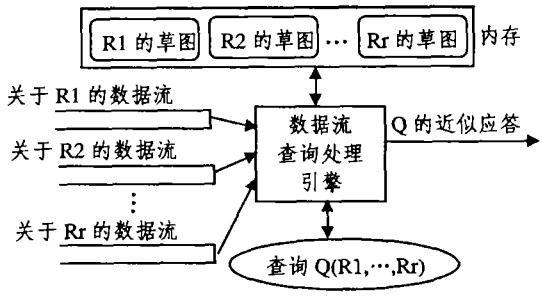

图1 数据流查询处理的体系结构

在任何时刻,查询处理算法能组合已保存了的草图 $S(R_1),\cdots,S(R_n)$,给查询 Q 产生一个近似的应答。

2.2 随机草图技术

随机草图的基本技术就是自连接大小跟踪技术。现在来考虑一个简单数据流处理的情况,当关系 R 的元组流入时,其目标是要估算关系的一个属性 $R.A$ 上的自连接的大小,寻求查询 $Q=\mathrm{COUNT}(R \bowtie_A R)$ 的近似结果。$\mathrm{dom}(A)$表示连接属性的域,$|\mathrm{dom}(A)|$表示域的大小,$f(i)$为 $R.A$ 中属性值 i 的频率,对自连接的大小 $SJ(A)=\sum_{i\in dom(A)} f(i)^2$[3]进行近似估算。在 Alon、Matias 和 Szegedy 的论文中证明了任何确定性算法产生一个与 SJ(A)紧密近似的估算至少需要 $\Omega(|\mathrm{dom}(A)|)$比特的存储空间,很显然为数据流实施这样的解决方案是不切实际的。但他们所建议的一个随机化技术能够在 SJ(A)近似的结果的质量上能提供高概率的保证,而且只需用 $O(\log|(\mathrm{dom}(A)|))$的空间。简单地说,他们的方案的基本思想是在 $R.A$ 的数据流上定义一个比较容易计算出来的随机变量 X[4,9],并保证:

(1)对 $SJ(A)$来说 X 是没有偏见的估算值(即没有按期望去校正),$E[X]=SJ(A)$;

(2)X 具有足够小的偏差 $\mathrm{Var}(X)$,使估算质量具有高概率的保证。

3 用数据流草图来近似复杂查询的答案

在这一节中,主要讨论基于草图的技术,在多重数据流关系 $R_1,\cdots,R_r$ 上的复杂、多连接的聚集 SQL 查询,并保证一定的近似质量。我们将重点考虑的那些具有一般形式的查询:"SELECT AGG FROM $R_1,R_2,\cdots,R_r$ WHERE E ",这里的 AGG 是一个任意的聚集操作符(如 COUNT、SUM 或 AVERAGE),E 表示 n 个形如 $R_i.A_j=R_k.A_l$ 等值连接约束的连接($R_i.A_j$ 表示关系 R_i 的第 j 个属性)。

首先来说明草图是怎样给 COUNT 聚集提供具有高概率质量保证的近似应答,然后将结论推广到其它聚集操作符如 SUM。为了导出关于估算错误的概率保证,需要确保属于关系的每一个属性最多在连接条件 E 中只出现一次。当然,这不是一个苛刻的要求,任何一个连接条件只要作适当的变换,就能满足这一要求:对任意一个在 E 中出现 $m>1$次的属性 $R_i.A_j$,可以追加 $m-1$个新的"属性"到 R_i 中,替换 $R_i.A_j$ 的 $m-1$发生在 E 中的属性,每一个用不同的新属性来替代。这 $m-1$个新的属性是 $R_i.A_j$ 的精确复制品,所以它们在 R_i 的每个元组中都可以采用与 $R_i.A_j$ 相同的值。例如,如果 $E=((R_1.A_1=R_2.A_1)\mathrm{AND}(R_1.A_1=R_3.A_1))$,可以通过追加一个新属性 A_2到 R_1中(A_2是 A_1的复制品),并替换 $R_1.A_1$的一个发生,就能修改它满足单一属性发生约束,那么 $E=((R_1.A_1=R_2.A_1)\mathrm{AND}(R_1.A_2=R_3.A_1)$。很显然,这些"属性"的增加只能在概念级水平上进行,作为草图计算逻辑的一部分。以后,我们都假设 E 满足单一属性的约束。

3.1 用草图回答 COUNT 查询

计数查询 Q_{COUNT}的输出就是在数据流关系 $R_1,\cdots,R_r$ 的交积中那些在连接属性上满足 E 的等值约束的元组个数。假设在 E 中将$2n$ 个连接属性重新命名为 $A_1,A_2,\cdots,A_{2n}$,在 E 中的每一个等值约束就具有 $A_j=A_{n+j}$的形式,其中$1\leqslant j\leqslant n$。设 $\mathrm{dom}(A_i)=\{1,\cdots,|\mathrm{dom}(A_i)|\}$是属性 A_i 的域,且 $D=\mathrm{dom}(A_1)\times\cdots\times\mathrm{dom}(A_{2n})$。同时也用 S_k 表示出现在 E 中的关系 R_k 的属性(已经重新命名了的)的子集,$D_k=\mathrm{dom}(A_{k_1})\times\cdots\times\mathrm{dom}(A_{k_{|S_k|}})$这里 $A_{k_1}\cdots A_{k_{|S_k|}}$是在 S_k 中的属性。从它们各自的域中指定一些值给连接的属性,记作 Z。如果 $Z\in D$,那么每一个连接的属性 A_j 都被 Z 指定一个值 $Z[j]$。另一方面,如果 $Z\in D_k$,那么 Z 只指定一个值 $Z[j]$给 $j\in S_k$,(为了方便,当上下文中差别是明显时,可以简单地使用 j 代表属性 A_j)用 $Z[S_k]$表示 Z 在 S_k 中属性上的投影,很显然 $Z[S_k]\in D_k$。由于 $Z\in D_k$,还可以用 $f_k(Z)$表示在 R_k 中元组的个数,其所有 $j\in S_k$ 的属性 j 的值都等于 $Z[j]$。本文中使用的符号都基于此定义。

查询 COUNT 的结果现在可以表示为 $Q_{\mathrm{COUNT}}=\sum_{Z\in D,\forall j,Z[j]=Z[n+j]}\prod_{k=1}^{r} f_k(Z[S_k])$。这实际上是对那些满足等值连接约束 E 的所有指定值 $Z\in D$,与 Z 相匹配的每一个关系中元组个数的乘积的累加之和。可以构造一个随机变量 X,它对 Q_{COUNT}没有任可偏见(也就是 $E[X]=Q_{\mathrm{COUNT}}$),然后使用标准的平均和中值选择方法[3]进行求解,这样就可以大大地提高 X 的估算精度和可信度,保证较小的相对误差能有较高的概率。

现在来说明如何构造 X 这样的随机变量。对于在 E 中每一对连接属性 j 和 $j+n$,建立一个独立随机变量族$\{\xi_{j,l}:l=1,\cdots,|\mathrm{dom}(A_j)|\}$,这里每一个 $\xi_{j,l}\in\{-1,+1\}$。每一个等值连接属性对 j 和 $n+j$ 必须共享同一个 ξ 族,因此对所有 $l\in\mathrm{dom}(A_j)$,都有 $\xi_{j,l}=\xi_{n+j,l}$;然而,可以为 n 个不同的等值连接对使用互相独立的随机源生成相应的 ξ 族,从而定义了 n 个不同的 ξ 族。由于随机变量属于为不同的属性对定义的族,因此它们是彼此完全独立的。正如前面提到的那样,由于属性对 $j,j+n$ 的族只要用 $O(\log|\mathrm{dom}(A_j)|)$的内存空间就可以有效在线构造出来,因此 n 个随机变量族总共所需要的空间应为 $\sum_{j=1}^{n} O(\log|\mathrm{dom}(A_j)|)$。

可以为每个关系 R_k 定义原子草图 X_k,$X_k=\sum_{Z\in D_k}(f_k(Z)\prod_{j\in S_k}\xi_{j,Z[j]})$,COUNT 的随机变量的估算值应为 $X=\prod_{k=1}^{r} X_k$(也就是原子关系草图 X_k 的积)。很显然,当 R_k 的元组流入时每一个原子草图 X_k 可以有效地计算出来;在特殊情况,可以将 X_k 初始化为0,对 R_k 流中的每一个元组 t,只需要将数量 $\prod_{j\in S_k}\xi_{j,t[j]}$加到 X_k 中就可以了,这里 $t[j]$表示 j 在元组中的属性值。很显然会有下面的引理:

引理3.1 随机变量 $X=\prod_{k=1}^{r} X_k=\prod_{k=1}^{r}\sum_{Z\in D}(f_k(Z)\prod_{j\in S_k}\xi_{j,Z[j]})$是 Q_{COUNT}查询的一个无偏见的估算值,也就是 $E[X]=Q_{\mathrm{COUNT}}$。

在传统的查询处理中，输入查询 Q_{COUNT} 的连接图可以定义为一个无向图，每一个关系 $R_i, i=1,\cdots,r$ 为图的一个结点，包含属性 j 和 $j+n$ 的关系结点之间每一个连接属性对 j，$j+n$ 就构成一个边。X 的偏差的上限和下限计算都依赖于 Q_{COUNT} 的连接图是非周期性的假设。因此，该技术只有在非周期多连接查询中才有良好的效果。幸运的是，在数据库实践中所遇到的 SQL 连接查询大多都是非周期性。在这样一个非周期性的假设的前提下，下面的引理3.2将给出 Q_{COUNT} 的无偏见估算 X 的偏差范围。为了简化说明，用 $SJ_k(S_k)=\sum_{Z\in D_k} f_k(Z)^2$ 表示关系 R_k 在 S_k 中所有属性上的自连接的大小。

引理3.2 假设 Q_{COUNT} 的连接图是非周期性的，那么对随机变量 $X=\prod_{k=1}^{r}X_k$ 有：

$$\prod_{k=1}^{r}SJ_k(S_k)-\sum_{Z\in D,Z[j]=Z[n+j]}\prod_{k=1}^{r}f_k(Z[S_k])^2\leqslant \mathrm{Var}(X)\leqslant ((2^n-1)^2+1)(\prod_{k=1}^{r}SJ_k(S_k)-\sum_{Z\in D,Z[j]=Z[n+j]}\prod_{k=1}^{r}f_k(Z[S_k]))^2$$

Q_{COUNT} 的最终估算值 Y 是 s_2 个随机变量 $Y_1,\cdots,Y_{s_2}$ 的中值，每一个 Y_i 是 s_1 个独立均匀分布的随机变量 X_{ij} 的平均值，其中 $1\leqslant j\leqslant s_1$，每个 X_{ij} 是按上面 X 相同的构造的方法在线构造出来的。因此，Q_{COUNT} 的草图的大小就是 $O(S_1\cdot S_2\cdot\sum_{j=1}^{n}\log|\mathrm{dom}(A_j)|)$。为了达到一定的精度并具有较高的可能性，$s_1$ 和 s_2 的值可以用下面定理3.1来推导：

定理3.1 设 Q_{COUNT} 是非循环的、多连接的在关系 $R_1,\cdots,R_r$ 上的 COUNT 查询，$Q_{COUNT}\geqslant L$ 和 $SJ_k(S_k)\leqslant U_k$。那么可以用大小为 $O(\frac{2^{2n}(\prod_{k=1}^{r}U_k)\log(1/\delta)}{L^2\epsilon^2}\sum_{j=1}^{n}\log|\mathrm{dom}(A_j)|)$ 的草图作为 Q_{COUNT} 的近似结果，其估算结果的相对误差最多为 ϵ，其概率至少为 $1-\delta$。

因限于篇幅，定理的证明就省略了。

3.2 用草图来回答 SUM 查询

复杂 COUNT 聚集查询的基于草图的近似计算方法也可以用来计算其它聚集函数的复杂查询，如在关系流上 SUM。SUM 查询可能有形如 SELECT SUM($R_i.A_j$)FROM $R_1,R_2,\cdots,R_r$ WHERE E。不失一般性，和前面一样，设 $A_1,\cdots,A_{2n}$ 是 E 中 $2n$ 个重新命名了的连接属性，$R_i=R_1$，A_{2n+1} 表示 R_1 中的属性，其值将加到连接的结果之中。并且对所有在 R_1 中连接属性都指定一个值 $Z(Z\in D_1)$，$SUM(Z)=\sum_{t\in R_1,\forall j\in S_1,t[j]=Z[j]}t[A_{2n+1}]$；因此，SUM(Z)基本上是由在关系 R_1 所有元组 t 中的属性 A_{2n+1} 与在连接属性集 S_1 中和 Z 相匹配的取值之和。因此 Q_{SUM} 可以表示为：

$$\sum_{Z\in D,\forall j,Z[j]=Z[n+j]}SUM(Z[S_1])\cdot\prod_{k=2}^{r}f_k(Z[S_k])$$

与 COUNT 情况相似，为了在一个数据流上获得查询的近似值 Q_{SUM}，使用独立随机变量族为每一个关系建立相对应的原子草图 X_k，对每一个连接属性对使用彼此不同、相互独立的 ξ 族。原子草图 $X_k(k=2,\cdots,r)$ 也按前面 COUNT 查询描述的那样定义，也就是 $X_k=\sum_{Z\in D_k}(f_k(Z)\prod_{j\in S_k}\xi_{j,Z[j]})$。然而，对于包含 SUM 属性的关系 R_1，X_1 可以用稍微不同的方式来定义：$X_1=\sum_{Z\in D_1}(SUM(Z)\prod_{j\in S_1}\xi_{j,Z[j]})$。很显然，对每一个流入 R_1 的元组 t 只需要简单地追加数量 $t[A_{2n+1}]\cdot\prod_{j\in S_1}\xi_{j,t[j]}$ 就可以完成 X_1 在 R_1 的元组流上的维护。可以使用与引理3.1和引理3.2相同的结论，随机变量 $X=\prod_{k=1}^{r}X_k$ 能得到 Q_{SUM} 的一个期望的值，偏差（假设是一个非循环的连接图）也能限制在引理3.2所规定的范围之内。这些结果可以用于为 Q_{SUM} 建立草图，其精度和概率与定理3.1所说明的相同。

4 用草图划分技术提高应答的质量

在定理3.1的证明中，为了确保对 Q_{COUNT} 的估算的相对误差 ϵ 的上限值具有较高的可能性，就必须对每一个 i，都保证 $\mathrm{Var}(Y_i)\leqslant\frac{\epsilon^2L^2}{8}$；可以通过定义每一个 Y_i 作为 s_1 个原子草图估算值 X 的独立均匀分布实例的平均值来达到这一目标，因此 $\mathrm{Var}(Y_i)=\frac{\mathrm{Var}(X)}{s_1}$。那么由引理3.2就有 $\mathrm{Var}(X)\leqslant 2^{2n}\cdot\prod_{k=1}^{r}SJ_k(S_k)$，在 $s_1(s_1\geqslant\frac{2^{2n+3}\cdot\prod_{k=1}^{r}SJ_k(S_k)}{\epsilon^2L^2})$ 份 X 的独立均匀分布的复制品求平均，能确保 Y_i 的偏差在要求的上限值范围之内。对于多连接来说，所关心问题是偏差上限值对 $\mathrm{Var}(X)$ 的限制（从引理3.2证明的），因此要确保给定的精度水平，所需的 X 实例数量 s_1 就会随在查询中连接数量 n 爆炸式地增长。

为了解决这个问题，下面提出一种全新的草图划分技术，利用数据流上的近似统计表来分解草图化问题，确保估算的精度。其基本思想是在查询中将连接属性的域进行智能化划分，在每一个划分的部分上分别估算 Q_{COUNT} 的部分值，既能最大限度减少在给定精度水平所需要的内存容量，又能在给定数量的草图空间取得最大可能的估算精度。该方法能够很自然地推广到其它聚集操作（例如：SUM，AVERAGE），是一种通用的技术。

从定理3.1可以得知，给定一个要求的精度水平，所需要的 X 独立均匀分布的实例数与在连接属性上的关系 $R_1,\cdots,R_r$ 上自连接的大小的乘积是成正比的。而且，实际上连接属性域经常是偏斜的，对不同的属性偏斜经常集中在不同地方。正因为如此，可以利用数据分布某些规律，对连接属性（一些子集）的域进行智能化划分，以获得组合属性空间的每一个部分 p，限制在 p 中关系的自连接大小的积要比整个（未划分）的属性空间的同样积（也就是 $\prod_{k=1}^{r}SJ_k(S_k)$）要小得多。因此，设 X_p 表示与属性空间的部分 p 相对应 Q_{COUNT} 一个部分的原子草图估算值，能期望偏差 $\mathrm{Var}(X_p)$ 比 $\mathrm{Var}(X)$ 小很多。

现在来考虑一个方案，在原子草图 X_p 的 s_p 个独立均匀分布的实例上求平均，并用 Y_i 表示在所有部分 p 上的平均值之和。然后可以得到 $E[Y_i]=Q_{COUNT}$，$\mathrm{Var}(Y_i)=\sum_p\frac{\mathrm{Var}(X_p)}{s_p}$。很显然，由于属性空间划分使自连接和偏差 $\mathrm{Var}(X_p)$ 都会减小，这意味着使需要确保 $\mathrm{Var}(Y_i)\leqslant\frac{\epsilon^2L^2}{8}$ 的独立均匀分布草图实例总数 $\sum_p s_p$ 也会减小；反过来，会使在指定 Y_i 估算值的精度水平下所需要的存储空间减小。下面将这些思相进行形式化，为单连接和多连接查询寻求草图划分的结果和算法。

还是以 Q_{COUNT} 聚集查询为例，草图划分技术通常是将每个连接属性 A_j 的域划分成 $m_j\geqslant 1$ 个不相交的子集，表示成 $P_{j,1},\cdots,P_{j,m_j}$。而且将连接属性对 A_j 和 A_{n+j} 的域匀均划分（也就是 $\mathrm{dom}(A_j)=\mathrm{dom}(A_{n+j})$）。从单个属性的划分就可导出组合（多维）连接属性空间的划分，组合空间的划分记为 P，因

此，$P=\{(P_{1,l_1},\cdots,P_{n,l_n}):1\leqslant l_j\leqslant m_j\}$。对每一个元素 $p\in P$ 标识整个属性空间的一个唯一划分部分，用 D_p 表示整个属性空间的 D 到 p 的一个约束；换句话说也就是：$D_p=\{Z\in D:Z[j],Z[n+j]\in p[j],\forall j\}$，这里的 $p[j]$ 表示属性 j 在 p 中的部分。同样 $D_{k,p}$ 是表示 D_p 在关系 R_k 的连接属性上的一个投影。

对于每一个部分 $p\in P$，可以构造随机变量 X_p 用来估算在域空间 D_p 上的 Q_{COUNT}，构造方法与前面的原子草图 X 相同。因此，对每一个部分 p 和连接属性对 j 和 $n+j$，就有一个独立随机变量族 $\{\xi_{j,l,p}:l\in p[j]\}$；对每一个（关系，部分）对 (R_k,p)，就定义一个随机变量 $X_{k,p}=\sum_{z\in D_{k,p}}(f_k(Z)\prod_{j\in S_k}\xi_{j,z[j],p})$。变量 X_p 可以从所有关系上的 $X_{k,p}$ 的乘积来获得，即 $X_p=\prod_{k=1}^{r}X_{k,p}$。很容易验证，对于部分 p 来说 $E[X_p]$ 就等于连接结果中元组的个数，由于期望值是线性的，所以 $E[\sum_p X_p]=\sum_p E[X_p]=Q_{COUNT}$。

由于各部分之间是相互独立的，所以有 $\mathrm{Var}(\sum_p X_p)=\sum_p \mathrm{Var}(X_p)$。为了减小各部分的估算偏差，对每一个 X_p 都构造独立均匀分布的实例。然而 $\mathrm{Var}(X_p)$ 在各部分上可能有很大的不同，通过保存和维护大量具有较高偏差的实例，能够使总的偏差大大减少。设 s_p 表示部分 p 的草图 X_p 的独立均匀分布的实例的份数，并设 $Y_{i,p}$ 表示这 s_p 份实例的平均值。然后计算 $\sum_p Y_{i,p}$ 作为 Y_i 的值（在独立均匀分布的实例上求平均不会改变期望值），从而保证 $E[Y_i]=Q_{COUNT}$。

草图划分方法的成功取决于当数据元组流入时是否能够为每一个（关系，部分）对有效计算 s_p 个 $X_{k,p}$ 的个独立均匀分布实例。对每一个部分 p，要为每一个属性对 $j,n+j$ 保存变量的 s_p 个独立族 $\xi_{j,p}$，每一个族都是用一个独立随机源生成的。而且对数据流中的每一个元组 $t\in R_k$ 和每一个部分 p（t 在 p 中，即 $t\in D_{k,p}$），将数值 $\prod_{j\in S_k}\xi_{j,t[j],p}$ 加到 $X_{k,p}$ 中。草图划分技术对一个元组识别相关部分的处理是非常有效，通过使用近似数据流统计表将在每个属性 A_j 的域中的值的连续区域分组成较小数量的粗略的散列表元（例如直方图统计表就能很容易给出这种散列表元）。属性 A_j 的 m_j 个部分也就构造成这样散列表元的子集，每个散列表元存储它相应部分的标识。由于这些散列表元的数量通常很少，给定一个输入的元组，包含 $t[j]$ 的散列表元（属于 A_j 的相关部分）就能很快确定（例如用二元或线性搜索）。这就能够为流入的数据元组非常有效地确定与之相关的部分。

在所有部分上的原子草图所需要的总存储空间为 $O(\sum_p s_p\sum_{j=1}^{n}\log|\mathrm{dom}(A_j)|)$，用来计算 Y_i。为了简化，这里对部分 p 的每一个 $\xi_{j,p}$ 族使用常量 $O(\sum_{j=1}^{n}\log|\mathrm{dom}(A_j)|)$ 代替精确的 $O(\sum_{j=1}^{n}\log|p[j]|)$，以近似存储开销。如果要有效地使用草图划分方法，就必须注意两个重要的问题：

（1）要选择好的划分集 P；

（2）确定为每一个部分构造 X_p 的独立均匀分布实例的数量 s_p。

很显然，既要保证每一个 Y_i 都具有一定的精度 ε，又要使分配给草图的总的存储空间达到最小的最终目标，有效地解决这两个问题是至关重要的。也就是要计算划分 P，给每个部分 p 分配存储空间 s_p（存放 s_p 份草图），使 $\mathrm{Var}(Y_i)\leqslant\frac{\varepsilon^2L^2}{8}$ 成立，使 $\sum_{p\in P}S_p$ 最小。

由于各部分之间是相互独立的，各原子草图又具有独立均匀分布的特性，因此有 $\mathrm{Var}(Y_i)=\sum_p\frac{\mathrm{Var}(X_p)}{s_p}$。给定一个属性空间划分 P，并保证 $\mathrm{Var}(Y_i)$ 在上限范围内，为确保整个草图空间最小化，就必须找到合适的 s_p。如何选择 s_p 的问题可以作为具体的最优化问题来处理。下面的定理可以说明如何计算这样一个最优分配问题。

定理4.1 P 是一个连接属性域的划分，那么可以分配空间 $s_p=\frac{8\sqrt{\mathrm{Var}(X_p)}\sum_p\sqrt{\mathrm{Var}(X_p)}}{\varepsilon^2L^2}$ 给每一个 $p\in P$，保证 $\mathrm{Var}(Y_i)\leqslant\frac{\varepsilon^2L^2}{8}$ 和 $\sum_p s_p$ 最小。

从上面的定理可以知道，给定划分 P，在一个给定精度水平下最佳的空间分配所需要的整个草图空间是：$\sum_p s_p=\frac{8(\sum_p\sqrt{\mathrm{Var}(X_p)})^2}{\varepsilon^2L^2}$。很显然，使草图总空间需求最小化的最优划分 P 就是要保证累加和 $\sum_p\sqrt{\mathrm{Var}(X_p)}$ 最小化。因此，我们只需要寻求计算这种最优划分 P 的技术，一旦找到了 P，就可以得用定理4.1为了每一个划分部分计算最优的空间分配。

有了这些思想和方法，就不难找到各种聚集查询的连接属性域的最优划分具体算法了。限于篇幅，在本文中就不作讨论了。

5 实验研究

我们在数据流环境下进行了基于草图技术查询处理的实验研究，比较了基于草图的方法和基于传统的直方图方法在数据流上处理复杂查询的近似结果；并检验了草图划分数量对查询结果质量的影响。在实验中，我们使用了人工编制的和实际的两种数据集，对各种类型的查询进行了大量的实验。其实验结果如图2所示。

从实验结果也可得出这样的结论：

·查询结果的质量 当估算复杂聚集查询时，基于草图的算法是相当精确的。甚至只有很少几 k 字节的内存，最终结果的相对误差是经常小10%。基于草图方法的应答精度比基于直方图的方法要高得多，精度可以提高三分之一到一个数量级。

·草图划分的作用 实验表明使用合适的算法对属性域进行划分，并仔细分配可用的内存空间给各部分相对应的草图，能够极大限度地提高返回结果的质量（大约从一个数量级到三个数量级）。

·近似属性统计表的影响 实验还说明草图划分技术非常有效和健壮，即使非常粗糙、近似的属性统计表，也仍然可以使用。

因此，实验能够有效地验证了本文中的论点：草图技术是切实可行性的，草图是数据流上复杂聚集查询处理的有力工具；在整个草图划分过程中，仔细分配可用的内存空间是非常重要的。

结束语 本文讨论了如何使用有限的内存近似解答聚集 SQL 查询问题。使用数据流草图技术，能够很方便地获得数据流草图，给聚集查询提供近似的解答，并保证误差在一定范围之内。由于随机化的草图具有较大的偏差，使得近似质量大为降低，影响了它的实际应用，因此我们开发了一个全新的草

图划分技术。利用数据流已有的统计信息将属性域智能化地划分，分解草图问题，保证结果能达到要求的近似质量。不论理论还是实践都可以证明本文中的草图技术所提供结果的质量比直方图技术提供的要好得多(大约从一个数量级到三个数量级的提高)，即使是基于粗糙的统计数据，草图划分也可以大大地提高估算的精度。因此草图技术是数据流查询处理的一种非常有效、实用的方法。

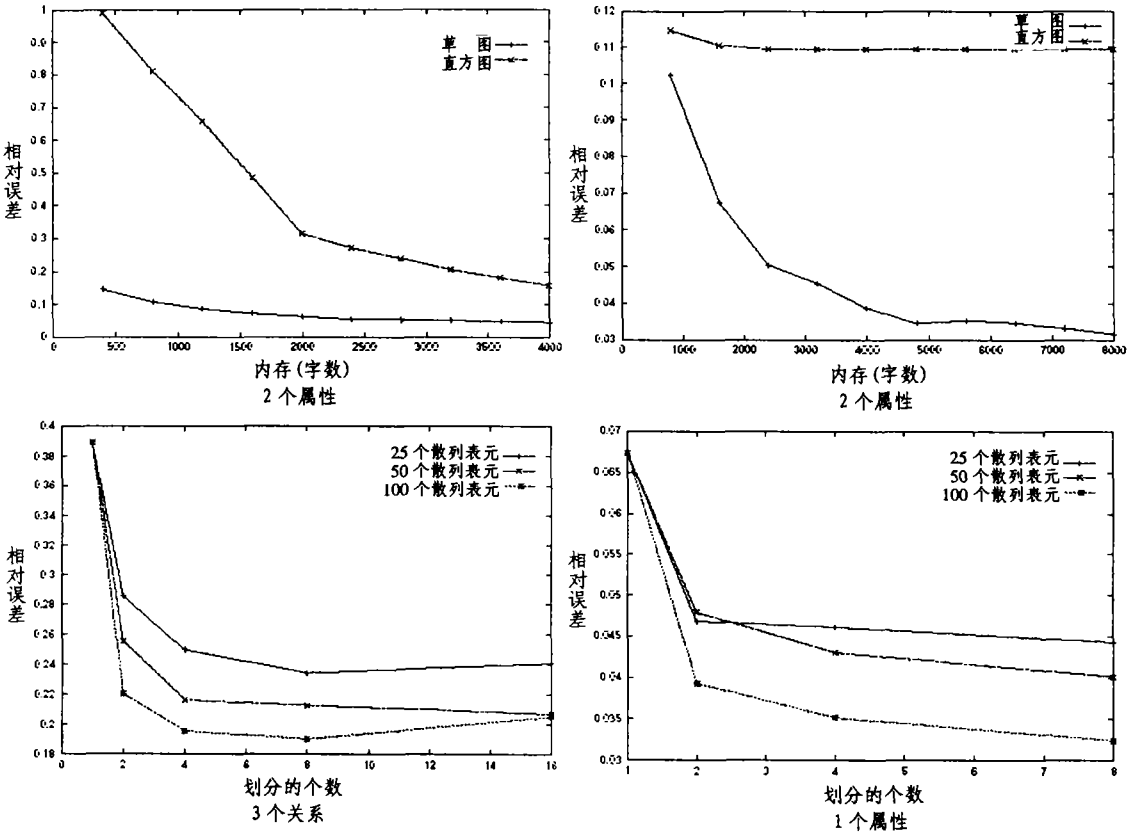

图2 实验研究的部分结果

参 考 文 献

1 Arasu A, et al. Characterizing memory requirements for queries over continuous data streams. In: Proc. 21st ACM SIGACT-SIGMOD-SIGART Symp on Principles of Database Systems, Madison, Wisconsin, May 2002. 221～232

2 Gilbert A, et al. Fast, small-space algorithms for approximate histogram maintenance. In: Proc. of the 2002 Annual ACM Symp. on Theory of Computing, 2002

3 Alon N, Matias Y, Szegedy M. The Space Complexity of Approximating the Frequency Moments. In: Proc. of the 28th Annual ACM Symp. on the Theory of Computing, May 1996

4 Alon N, Gibbons P B, Matias Y, Szegedy M. Tracking Join and Self-Join Sizes in Limited Storage. In: Proc. of the Eighteenth ACM SIGACT-SIGMOD-SIGART Symp. on Principles of Database Systems, May 1999

5 Alon N, Matias Y, Szegedy M. The Space Complexity of Approximating the Frequency Moments. In: Proc. of the 28thAnnual ACM Symp. on the Theory of Computing, May 1996

6 Babcock B, et al. Models and issues in data stream systems. In: Proc. 21st ACM SIGACT-SIGMOD-SIGART Symp. on Pinciples of Database Systems, Madison, Wisconsin, May 2002. 1～16

7 Motwani R, Widom J, et al. Query processing, approximation, and resource management in a data stream management system. In: Proc. First Biennial Conf. on Innovative Data Systems Research (CIDR), Jan. 2003

8 Guha S, Koudas N. Approximating a data stream for querying and estimation: Algorithms and performance evaluation. In: Proc. of the 2002 Intl. Conf. on Data Engineering, 2002. 567～576

9 Dobra A, Garofalakis M. Processing Complex Aggregate Queries over Data Streams. ACM SIGMOD June 2002

10 Chandrasekaran S, Franklin M. Streaming queries over streaming data. In: Proc. 28th Intl. Conf. on Very Large Data Bases, Aug. 2002