XML 数据查询中值匹配查询代价估计算法*

曲卫民+, 孙 乐, 孙玉芳

(中国科学院 软件研究所 系统软件与中文信息中心,北京 100080)

A Result Size Estimation Algorithm for Value Predication in XML Query

QU Wei-Min+, SUN Le, SUN Yu-Fang

(Chinese Information Processing Center, Institute of Software, The Chinese Academy of Sciences, Beijing 100080, China)

+ Corresponding author: E-mail: quweimin@chinalife.com.cn, http://www.sonata.iscas.ac.cn

Received 2004-04-12; Accepted 2004-09-08

Qu WM, Sun L, Sun YF. A result size estimation algorithm for value predication in XML query. *Journal of Software*, 2005,16(4):561−569. DOI: 10.1360/jos160561

Abstract: Result size estimation of value predication in XML query is a multiple attributes dependent problem. It is different from the counterpart in relational database, for the multiple attributes in XML involve not only the value data, but also the structural information. To solve the problem, this paper proposes a wavelet-based histogram for the result size estimation of value predication in XML query. It also gives the way to identify the multi-dimensional dependent element set, to rewrite the value predication and value denotation of structural information. Experimental results show that the algorithm achieves on accurate result size estimation for value predication in XML query.

Key words: XML; value predicate; result size estimation

摘 要: XML 数据查询中值匹配查询条件的查询代价估计问题是一种典型的多元素查询条件代价估计问题.它与传统关系型数据库中的多元素查询条件不同,因为 XML 数据中的值信息分布不仅与其他值信息分布相关,还与 XML 数据中的结构信息相关,而且当 XML 数据结构比较复杂时,可能会形成高维元素相关.针对以上问题,提出了一种面向XML数据的基于小波的多维直方图查询代价估计算法,并提出了确定XML数据中以某值元素为主键的相互依赖元组的方法,将值匹配条件改写为多元素查询条件的方法以及结构信息的值化方法.实验结果证明,提出的方法取得了较准确的查询代价估计结果.

关键词: XML;值匹配条件;查询代价估计

中图法分类号: TP311 **文献标识码**: A

XML 已经成为 Internet 以及电子商务中进行数据表示和数据交换事实上的标准.由于其丰富的表达能力和自描述性、灵活性等特点,XML 已被广泛应用于电子商务、数字图书馆、智能 Internet 检索等领域.如何高效、

* Supported by the National Natural Science Foundation of China under Grant No.60203007 (国家自然科学基金); the National High-Tech Research and Development Plan of China under Grant No.2003AA1Z2110 (国家高技术研究发展计划(863)); the New Star Plan of Science & Technology of Beijing of China under Grant No.H020820790130 (北京市科技新星计划)

作者简介: 曲卫民(1977－),男,山西运城人,博士,主要研究领域为 XML 数据库,信息检索;孙乐(1971－),男,博士,副研究员,主要研究领域为信息检索,中文信息处理;孙玉芳(1947－2005),男,博士生导师,研究员,主要研究领域为系统软件,数据库理论.

准确地完成对 XML 数据的查询成为当前的一个研究热点.为了高效地完成对 XML 数据的查询,一个非常重要的环节就是对查询代价的估计.利用这种对查询代价的估计,我们可以完成很多重要的处理.首先,可以生成更优的查询执行计划,从而缩短整个查询执行的时间;其次,用户可以利用查询代价估计改进自己的查询,快速得到自己所提交的查询的部分结果及结果集的大小估计,并根据这些信息,按照需要改进自己提交的查询,这一点在 Internet 的环境中非常有用.

XML 查询中经常会出现一些值匹配条件,包括等值条件或值域条件,如“library/book[year≥1990]/title”等.文献[1−6]虽然都研究了 XML 数据中的查询代价估计问题,但都局限于对查询中的路径表达的代价估计,包括简单路径表达[1−3,5,6]和复杂路径表达[4],但还很少有文献专门研究过 XML 查询中值匹配查询条件的代价估计问题.

若要对这种值匹配条件的查询进行代价估计,最简单的方法就是为这类值信息建立直方图结构以保存其数据分布特征信息,当在查询时遇到此类值匹配条件时,便利用相应直方图信息进行代价估计,这种方法中一般为每种类型(标记)的值保存一个统一的直方图结构.但此方法实际上是不可行的,因为 XML 数据中的值信息并不是孤立存在的,它与 XML 数据中的结构(该值在 XML 数据中所处的位置)以及 XML 数据中的其他值信息(该值所在标记路径上的其他值元素)都有十分密切的关系.如果忽略这种相互依赖关系,将会给值匹配条件查询的代价估计结果带来巨大的偏差.这种相互信赖关系主要体现在以下两个方面:

① 标记路径与值信息之间的相互依赖关系.对 XML 数据统计模型中的某个节点 v 来说,若 v 为一个值信息节点,content(v)表示数据归纳节点 v 所对应的源数据节点集,则 content(v)中值信息的分布特征与到达源数据节点 v_s∈content(v)的后向标记路径有(或从 v 出发的前向标记路径)有很大关系.举例来说,对一个用来保存图书馆中的书目信息的 XML 数据,若既包含计算机类的图书,也包含儿童读物类的图书,则由标记路径“library/computer/book/price”和“library/Children/book/price”到达的表示价格的节点中所包含的价格信息明显会具有不同大小的值.如果我们仅对所有的 price 节点保存一个统一的直方图结构,可以想象它不可能对“library/computer/book[price>50]”这样的查询给出准确的代价估计.

② 值信息与值信息之间的相互依赖关系.对 XML 数据统计模型中的某个值信息节点 v,content(v)中值信息的分布特征与统计模型中其他值信息节点的值信息分布也可能有相当大的关系.举例来说,仍以用来保存图书馆中的书目信息的 XML 数据为例,一本书的价格信息与其出版的时间(年)也会有很大的关系,即 price 与 year 之间,对于查询“library/computer/book[year>1998]/price”与“lib/computer/book[year<1980]/price”其所到达的 price 节点的值信息分布特征会有很大的不同(新出版图书的价格可能比旧书的价格要高很多).因此这也证明我们不能只为 price 节点保存一个统一的一元直方图结构.

由以上分析可见,XML 数据中的值信息分布特征是一种典型的多因素关联分布.而且 XML 数据中的这种多因素关联值信息分布与传统的关系型数据中的关联数据分布(joint data distribution)有明显的不同,因为 XML 数据中不仅存在值信息之间的相互关联,还存在值信息与结构信息之间的相互关联,这也是 XML 数据的本质特点所造成的.因此,在对包含值匹配条件的 XML 数据查询进行代价估计时,我们不能单独考虑该值的独立分布特征,而应该综合考虑查询中的其他信息,包括路径表达信息和查询中的其他值匹配条件.为完成这种值匹配条件的查询代价估计,我们仍然使用直方图方法,但此时不能使用简单的一元直方图,而必须使用多维直方图,以适应 XML 数据中的多因素关联数据分布的要求.

1 代价估计方法的确定

在关系型数据库中,对于包含同一关系表中多个元素值匹配条件的查询,其查询结果集的大小依赖于多个值之间的关联分布特征,即多个元素所有可能值组合的出现频率.若采用元素间相互独立的假设会给查询结果集的大小估计造成很大的偏差.因此,人们开始研究多维直方图(multi-dimensional histogram)[7,8],试图捕获多个元素之间的相互关联关系.这些方法的实验结果证明,即使较简单的多元直方图也可以获得比多元素相互独立假设情况下好很多的查询代价估计结果.MuraliKrishna[7]等人提出了一种 Spatial Index Partitioning 方法以建立多元素等高直方图,这种方法的缺陷是其在分割时对于每一种元素(即一维)只考虑了一次,这种启发式策略有

点过于简单.Posala[8]等人提出了两种新的建立多维直方图的方法,第 1 种方法是将多元素联合分布(joint data distribution)分解成多个相互独立的部分(buckets),在每个部分(bucket)中假设其中的数据分布满足均匀分布.在这种方法中,利用 MHIST-2 算法建立的多维 MaxDiff(V,A)直方图是最准确的,而且可以获得比以前的各种多维直方图更优的查询代价估计结果.第 2 种方法则使用了强大的奇异值分解技术,利用少量的单个数据分布来模拟二维数据联合分布,但这种方法仅限于二维数据,对于多维数据联合分布则无能为力.

前面所述的这些用于建立多维直方图的方法都存在一个共同的缺陷,就是难以适应高维度的数据联合分布,即"curse of dimension"[9]的问题,如果相互关联的元素很多,则这些方法中使用的直方图大小会急剧增加,并在维度大于 6 或 7 时达到不可接受的程度.而在 XML 数据中,值信息分布就是一种高维度联合分布,即某个值信息不仅与其所在的标记路径相关,还可能与很多个其他的值信息相关,这种值信息的数目可能会非常多,尤其当 XML 数据结构比较复杂时,这种情况就很有可能发生.因此对于 XML 数据中的多维数据联合分布,我们需要一种能够有效适应这种高维数据联合分布的多维直方图建立方法,而基于小波的直方图[10]正是这样一种用于建立多维直方图的方法.

2 基于小波的直方图

小波变换是一种多层分解数据的数学工具.小波是一种用来模拟数据总体形状的函数,这种被模拟的对象可能是图像、曲线或曲面,它利用较小的空间表示这些对象在多层次上的特征的方法.基于小波的直方图的基本思想就是将传统直方图中的数据进行一定的变换,力图使用较小的空间存储传统直方图中保存的信息,并在使用时恢复这些信息,其具体的方法可以简单描述如下:

① 首先在预处理阶段,我们为要建立直方图的数据 X 计算出其扩展递增数据分布(extended cumulative data distribution)T^{c+},计算时可以使用全部原数据,也可以使用原数据的一个样本数据;

② 对第 1 步中计算出的扩展递增数据分布 T^{c+}进行小波变换,得到 N 个小波系数;

③ 从 N 个小波系数中挑选 m(m<<N)个小波系数,其余的全部设为 0.m 的大小根据存储空间的限制决定,而挑选 m 个小波系数则使用一定的过滤方法(thresholding method)进行.

通过上述的算法处理,我们得到了 m 个小波系数,这些值以及其所处的位置被存储在直方图中,利用这些数据我们可以在进行查询代价估计时恢复原有的递增数据分布.为估计形如"a≤x≤b"的查询条件的结果集大小,我们可以先利用基于小波的直方图中的信息计算出扩展递增数据分布中对应 a−1 和 b 的值,然后将这两个值相减便得到了最终的查询代价估计.实验证明,基于小波的直方图与传统的直方图以及基于实例的代价估计方法相比,在使用相同的存储空间时,可以获得更加准确的查询代价估计结果.

上面的算法中只给出了使用小波变换方法建立一维直方图的方法,但这种方法也可以很容易地扩展到建立多维直方图,而且由于小波变换本身的特点,这种基于小波的多维直方图可以方便地应用到高维度的情况中,而不会发生前面所述的维度爆炸的问题.文献[10]中的实验证明,这种基于小波的多维直方图与文献[9]中建立的多维直方图相比,可以获得更准确的查询代价估计结果.

2.1 一些基本概念

元素 x 的频率统计分布 $T=\{(v_1,f_1),(v_2,f_2),...,(v_n,f_n)\}$,其中,$v_1$ 表示元素 x 的一个值,f_i 表示 x 具有值 v_1 的记录个数.X 元素值 v_i 的递增数据频率 c_i 是指源数据 R 中满足 $t.x$≤v_i 的记录 $t\in R$ 的个数,即 $c_i=\sum_{j=1}^{i}f_j$.元素 x 的递增数据分布 $T^c=\{(v_1,c_1),(v_2,c_2),...,(v_n,c_n)\}$,其中 v_i 为 x 的值,c_i 则表示对应于 v_i 的递增数据频率.如果使用值域 D={0,1,2,…,N−1}表示元素 x 所有的可能值,而 $V=\{v_1,v_2,...,v_n\}$ 则代表元素 x 在源数据 R 中实际出现的所有值,其中可能存在 $v_i=v_j(i\neq j)$,并有 $V\subset D$.我们把 x 的递增数据分布由值域 V 扩展至值域 D 中,并令 D 中所有不属于 V 的值对应的出现频率为 0,便可能得到扩展递增数据分布 T^{c+} .由上述定义可知,在得到元素 x 在源数据 R 中的频率统计分布 T 后,假设 x 的值域 D={0,1,2,…,N−1},则可以得到 x 的扩展递增数据分布 $T^{c+}=\{(0,c_0),(1,c_1),...,(N-1,c_{N-1})\}$.

2.2 基于小波的多维直方图

基于小波的一维直方图的建立方法可以很容易地扩展到多维的情况.首先,我们来看如何定义建立多维数据的频率统计分布、递增数据分布以及扩展递增数据分布.设多维元素为$\{x_1,x_2,\ldots,x_d\}$,而$D_k=\{0,1,...,N_k-1\}$为元素x_k的值域,v_k表示元素x_k在源数据R中实际出现过的所有值的集合,$V_k=\{v_{k,1},v_{k,2},...,v_{k,n_k}\}$,其中$v_{k,1}<v_{k,2}<...<v_{k,n_k}$,$n_k$为$V_k$中值的个数,则对应元素$x_k$的频率统计分布为$T_k=\{(v_{k,1},f_{k,1}),(v_{k,2},f_{k,2}),...,(v_{k,n_k},f_{k,n_k})\}$,$f(i_1,i_2,...,i_d)$表示多维元素$\{x_1,x_2,...,x_d\}=(v_{1,i_1},v_{2,i_2},...,v_{d,i_d})$的联合频率,其含义为源数据中包含$x_k=v_{k,i_k}(1\le k\le d)$的记录个数.多维元素的联合数据频率统计分布$T_{1,...,d}$则由全部{值组合,该值组合对应的联合频率}这样的数据对组成,它可以用一个$n_1\times...\times n_d$的多维频率矩阵$F_{1,...,d}$表示,其第$[i_1,...,i_d]$项的值为$f(i_1,...,i_d)$,然后我们可以仿照一维的情况定义递增数据分布$T^c_{1,...,d}$和扩展递增数据分布$T^{c+}_{1,...,d}$,其中,对应多维元素值$\{x_1,x_2,...,x_d\}$的递增数据频率$F^{c+}_{1,...,d}$是一个$N_1\times...\times N_d$的矩阵.其定义如下:

$$p[x_1,x_2,...,x_d]=\sum_{i_1=0}^{x_1}\sum_{i_2=0}^{x_2}...\sum_{i_d=0}^{x_d}f(i_1,i_2,...,i_d).$$

小波变换的一个显著优点就是它可以方便地由一维元素推广至多维元素的处理中,因为无论是一维元素还是多维元素,在进行小波变换前都是由递增数据频率组成的一维向量,而不在于其中保存的递增数据频率具体表达的含义(是面向一维元素还是多维元素的),在利用小波系数重构递增数据分布时采取的方法也是一样的,因为要处理的都是一维数据,即在预处理阶段,我们首先计算出联合数据的多维频率矩阵$F_{1,...,d}$,然后利用其计算出递增联合数据分布$T^c_{1,...,d}$,再得到扩展递增联合数据分布$T^{c+}_{1,...,d}$.得到$T^{c+}_{1,...,d}$后,我们利用小波变换对其进行分解,得到小波系数,最后再进行一定的过滤,得到实际需要存储的m个小波系数.

在进行查询代价估计时,为处理如"$(a_1\le x_1\le b_1)\&\&...\&\&(a_d\le x_d\le b_d)$"的查询,我们需要利用小波系数重构$2^d$个递增频率$p[x_1,x_2,...,x_d]$,其中$a_j-1\le x_j\le b_j,1\le j\le d$,然后利用下面的方法得到上述查询的代价估计:

$$E=\sum_{a_j-1\le x_j\le b_j}\prod_{i=1}^{d}s(i)\times p[x_1,...,x_d].$$

其中,E表示对上述d维查询条件的代价估计结果,

$$s(j)=\begin{cases}1, & \text{如果}x_j=b_j\\ -1, & \text{如果}x_j=a_j-1\end{cases},1\le j\le d.$$

当$x_j=-1$时,我们令$p[x_1,x_2,...,x_d]=0$.

3 利用基于小波的直方图对 XML 查询中的值匹配条件进行代价估计

3.1 以某值元素为主键的多维相互依赖元组的定义

由第 1 节对 XML 查询中的值匹配条件查询的特征分析可得,XML 中的值匹配条件查询也是一种涉及多元素的查询代价估计问题.其中的多维数据包括查询中包含的所有值匹配条件中涉及到的值信息,以及每一个值匹配条件在查询中出现的标记路径信息.以查询"library/book[year>1999&&price<100]/author"为例,该查询的含义为查询图书馆中所有 1999 年后出版且价格在 100 元以下的图书作者.对于该查询中的值匹配条件查询,即price<100,其中涉及到的多维元素有 year,price,Path(year),Path(price)4 种,其中 Path(year)表示 year 在查询中的标记路径(此处为"library/book/year"),Path(price)则表示 price 在查询中的标记路径(此处为"library/book/price").但值得注意的是,XML 数据中多维元素与传统关系型数据中的多维元素有明显的不同,传统的关系型数据中的多维元素一般是指关系表中的各个字段,其对应的多维元素也指的是这些值同时出现在关系表中的某一记录中,而在 XML 数据中很难确定类似于记录这样的数据单元(我们需要这样的单元来统计多维元素的出现频率).

我们不可能简单地为所有相关的信息(元素)建立一个统一的多维直方图用来反映这些信息的相互依赖关系.本文采用了这样一种确定多元素间相互依赖关系的方法.首先,我们称 XML 数据中具有相互依赖关系的多

维元素为相互依赖元组.确定相互依赖元组的方法为:以某一值元素 V 作为相互依赖元组中的主键,并根据此值元素确定相互依赖元组中的其他元素.包括该值元素的标记路径 Path(V),所有与到达值元素 V 的节点路径中的任一节点直接相连的值元素 V_r 及其标记路径 Path(V_r).其中"直接相连"的含义是指在 XML 数据中两个节点之间存在一条边将二者相连.下面通过举例,说明本文中确定相互依赖元组的方法,如图 1 所示的源 XML 数据.

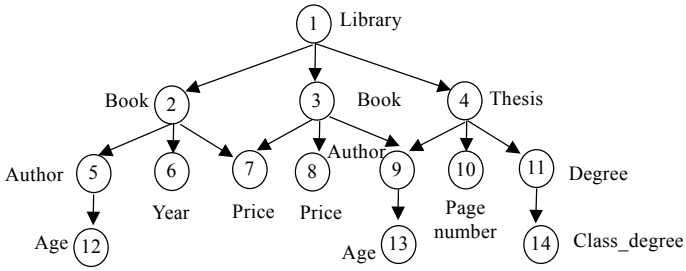

Fig.1 An example of XML data

图 1 XML 数据例子

对于图 1 中的源 XML 数据,其中标记 age,year,price,pagenumber 为值元素,其余的均为非值元素.若以 age 作为相互依赖元组的主键,则该相互依赖元组的成员包括 Path(age),year,price,pagenumber,Path(year),Path(price),Path(pagenumber).其中 Path(age)为值元素 age 的标记路径(后向);Year 被选定为相互依赖元素,因为节点 6 与到达节点 12(标记为 age)的节点路径 1/2/5/12(library/book/author/age)中的节点 2(author)直接相连;price 被选定为相互依赖元素,因为节点 7(price)也与到达节点 12(age)的节点路径 1/2/5/12(library/book/author/age)中的节点 2(author)直接相连;至于 pagenumber 则因为节点 11(pagenumber)与到达节点 13(age)的节点路径 1/4/9/13(library/thesis/author/age)中的节点 4(thesis)直接相连.Path(year),Path(price), Path(pagenumber)则分别为值元素 year(6),price(7),pagenumber(11)的后向标记路径.节点 8 的标记虽然亦为 price,但其与值元素 age 的任意节点路径都没有直接相连的关系,因此不应作为相互依赖元组中 price 元素的节点.

在确定了以某值元素 V 为主键的相互依赖元组后,我们便可以对任何查询中出现的包含该值元素的值匹配查询条件进行估计,并在估计的同时考虑该值匹配条件在查询中出现的位置,即该值元素在查询中的标记路径,以及查询中同时出现的其他与该标记路径直接相连的值匹配条件.实际上,在 XML 数据中,与某值元素 V 相互依赖的其他值元素并不限于前面所定义的相互依赖元素中确定的值元素,还包括其他一些值元素,如图 1 中的 XML 数据中的 age(节点 13)与 class_degree(节点 14,表示博士或硕士)之间也是具有相互依赖关系的.此外,值元素 V 与其前向标记路径也是具有相互依赖关系的,但本文中为简化处理,忽略了这些相互依赖关系.

3.2 标记路径的值化表示

前面所论述的直方图中保存的都是某种数值信息的数据分布,而在 XML 中,为反映值信息与结构信息之间的相互依赖关系,我们还必须在直方图中保存结构信息(标记路径),如何将这种结构信息转化为值信息以将其保存在直方图中便成为了首先需要解决的一个问题.本文中采用了以下的方法将标记路径利用一种特殊的值信息进行表示.

为源数据中的每种标记赋予一个正整数值,该正整数值的位数由源数据中不同的标记数决定,如若标记数小于 10,则选一位数即可,若标记数大于 9 但小于 100,则采用二位数表示标记,对于值为 1 的标记,则用 01 进行表示,依此类推.然后,我们就可以利用标记被赋予的值,将标记路径"library/book/author",若 lib=01,book=02,author=03,则该标记路径可以写为 010203.由于在建立直方图时,我们需要对值进行大小比较,因此还需要为标记路径的值表示法也建立大小比较关系.

设标记路径"$l_1/l_2/.../l_n$"可以用值 $a_1a_2...a_n$ 表示,其中 a_i 均为相同位数的正整数,且有 $l_i = a_i$(即标记l_i可用a_i表示).相似地,标记路径"$l'_1/l'_2/.../l'_m$"可用 $a'_1a'_2...a'_m$ 表示.

a. 依次比较 a_i 与 a'_i,$1 \le i \le \min\{n,m\}$直至找到 $a_i \ne a'_i$ 或 $i > \min\{n,m\}$;

b．若 $a_i \neq a'_i$,且有 $a_i > a'_i$,则有 $a_1a_2...a_n > a'_1a'_2...a'_m$,若有 $a_i < a'_m$,则有 $a_1a_2...a_n < a'_1a'_2...a'_m$;

c．若 $i > \min\{n,m\}$,且 $n=m$,则 $a_1a_2...a_n = a'_1a'_2...a'_m$;若 $n>m$,则有 $a_1a_2...a_n > a'_1a'_2...a'_m$,若 $n<m$,则有 $a_1a_2\ldots a_n < a'_1a'_2...a'_m$.

采用上述的标记路径赋值法,其优点是直观、易实现,还有就是方便对包含"//"的复杂路径表达的处理,这一优点我们将在后面提到.在建立直方图时,对每一个值元素我们都需要确定其值域 D 和值集 V,对于标记路径信息来说,其值集 V 就是源 XML 数据中所有出现过的标记路径.准确地说,应该是所有到达值元素节点的标记路径集,值域 D 本身的含义是标记路径可能出现的所有值的集合,即源数据中所有标记之间的任意顺序的任意组合,但实际上没有必要这样确定标记路径的值域,我们可取 $D=V$,并假设用户提交的查询不会出现源数据中不存在的路径,即用户事先知道了源 XML 数据的模式信息,或通过系统辅助得到了这样的信息.

3.3 基于小波的多维直方图的建立

在解决了对标记路径的赋值问题后,我们就可以开始为 XML 数据中的值信息建立基于小波的多维直方图.假设我们选取的主键值元素为 V_0,其相互依赖元组中包括 Path(V_0),记为 P_0,以及相关的其他值元素 V_i 及其标记路径 $P_i(1 \le i \le m)$,因此,这个相互依赖元组可表示为 $\{V_0,P_0,V_1,P_1,...,V_m,P_m\}$.我们采用以下步骤为这个相互依赖元组建立基于小波的多维直方图.

① 计算相互依赖元组的联合数据频率 $f(v_0,p_0,v_1,p_1,...,v_m,p_m)$,其中 v_i 为 V_i 所取的值, p_i 为 P_i 所取的值 $(0 \le i \le m)$, $f(v_0,p_0,v_1,p_1,...,v_m,p_m)$ 的计算可以通过对主键 V_0 中的所有值(对应于源 XML 数据中的每个值元素节点)逐个进行处理获得,假设当前处理的主键值元素节点为 e,则首先令 $v_0 = value(e)$,$value(e)$表示节点 e 的值,其次令 $p_0 = Path(e)$,$Path(e)$表示节点 e 的后向标记路径(再转化为值),再从 $NodePath(e)$(表示节点 e 的节点路径)上找出所有与任一 $NodePath(e)$上的节点直接相连的值元素节点 e_k,令 $v_k = value(e_k)$, $p_k = Path(e_k)$,对相互依赖元组中其他不满足上述关系的所有值元素 V'_k及P'_k,则令 $v'_k = NULL, p'_k = NULL$,$NULL$ 表示空值(对 XML 数据中的相互依赖元组中的值元素和所有标记路径元素,我们都必须在其值域中添加一个特殊值——空值,并假设空值比其他所有值都小),最后再对得到的值组合进行频率统计,即可得到联合数据频率 $f(v_0,p_0,v_1,p_1,...,v_m,p_m)$.

② 计算相互依赖元组的联合数据分布 $T_{v_0,p_0,v_1,p_1,...,v_m,p_m}$.

③ 根据 $T_{v_0,p_0,v_1,p_1,...,v_m,p_m}$ 计算相互依赖元组的递增联合数据分布 $T^c_{v_0,p_0,v_1,p_1,...,v_m,p_m}$.

④ 根据 $T^c_{v_0,p_0,v_1,p_1,...,v_m,p_m}$ 再计算相互依赖元组的扩展递增联合分布 $T^{c+}_{v_0,p_0,v_1,p_1,...,v_m,p_m}$.

⑤ 使用 Harr wavelet(或其他小波变换)对 $T^{c+}_{v_0,p_0,v_1,p_1,...,v_m,p_m}$ 进行小波变换,得到小波系数.

⑥ 对小波系数进行过滤得到 m 个需实际保存的小波系数.

3.4 查询代价的估计

3.4.1 将值匹配查询条件改写为多元素查询条件

在利用上述算法中建立的基于小波的多维直方图进行值匹配条件的查询代价估计时,我们必须首先保证该值匹配条件涉及的值元素是该多维直方图对应的相互依赖元素中的主键.而且在进行查询代价估计前,我们必须先将 XML 查询中的值匹配条件改写为多维直方图中的多元素查询条件,可通过以下步骤改写原查询:

① 先提取要进行查询代价估计的值匹配条件,并以其中的值元素作为主键;

② 提取该值元素在查询中所处的标记路径;

③ 依次提取与②中标记路径直接相连的其他值匹配条件及其值元素在查询中所处的标记路径;

④ 将以上提取的各种查询条件组合在一起便得到要进行代价估计的多元素查询条件.

这其中存在两个重要的问题需要解决:

a．对于相互依赖元组中,没有出现在查询中的值元素 V_j,我们则设该值元素为 NULL 值,该元素的标记路径亦为 NULL 值,即 $v_j = NULL, p_j = NULL$.

b．若查询中出现"//"等复杂路径表达,此时我们无法将标记路径直接改写为值形式,因为原有的标记路径赋值方法都是针对简单路径表达的.对于这类复杂路径表达,本文采取的方法是只取路径中第 1 个"//"之前的

标记路径部分(假设这部分标记路径对应的值为 $a_1a_2...a_m$),则可将该标记路径的值查询条件改写为 $a_1a_2...a_m < p < a_1a_2...(a_m+1)$,这种改写后的查询条件有效地包含了路径关系表达“//”所表达的查询含义,因为任何以“//”之前的标记路径作为开头部分的其他标记路径所对应的值都小于 $a_1a_2...(a_m+1)$,但却大于 $a_1a_2...a_m$,这种关系是由我们前面定义的标记路径赋值的大小比较关系所决定的,这也是本文中所采取的标记路径赋值法的重要优点之一.

下面通过一个例子来说明将 XML 查询中的值匹配条件改写为多元素查询条件的方法.

例:若采用图 1 中给出的源 XML 数据:

① 当查询为“library/book[year>1998&&price>50]”时,对其中的值匹配条件 price>50 进行代价估计,则以 price 为主键的相互依赖元组为{price,Path(price),year,Path(year)},其对应的多元素查询条件为 price>50&&Path(price)=“library/book”&&year>1998&&Path(year)=“library/book”, 若 library=01,book=02, 则有 Path(price)=“0102”,Path(year)=“0102”.

② 当查询为“library/book[year>1998]/author[age>30]”时,对其中的值匹配条件 age>30 进行代价估计,以 age 为主键的相互依赖元组为{age,Path(age),year,Path(year),price,Path(price),pagenumber,Path(pagenumber)},其对应的多元素查询条件为 age>30&&Path(age)=“library/book/author”&&year>1998&&Path(year)= “library/book”=“0102”&price=NULL&&Path(price)=NULL&&pagenumber=NULL&&Path(pagenumber)= NULL,若 author=03, 则 Path(age)=“010203”.

③ 当查询为“library//author[age>30]”时,为简单起见,假设以 age 为主键的相互依赖元组为{age,Path(age),year,Path(year)},则其对应的多元素查询条件为 age>30&&“01”<Path(age)<“02”&&year= NULL&&Path(year)=NULL.

3.4.2 利用基于小波的多维直方图完成查询代价估计

在得到 XML 查询中某个值匹配条件所对应的多元素查询条件之后,我们便可以采用与普通的基于小波的多维直方图中相似的方法对该多元素查询条件进行代价估计.

设由 XML 查询中的值匹配条件得到的多元素查询条件为

$$(a_0 \le v_0 \le b_0)\&\&(a_0' \le p_0 \le b_0')...\&\&(a_m \le v_m \le b_m)\&\&(a_m' \le p_m \le b_m').$$

首先,利用小波系数重构出计算查询代价估计所需的 2^{2m} 个多维递增频率值 $p[v_0,p_0,...,v_m,p_m]$,然后采用下面的方法,计算多元素查询条件的代价估计.为方便表示,我们令 $v_0=x_1,p_0=x_2,...,v_m=x_{2m+1},p_m=x_{2m+2}$,并相应地替换 $a_0,a_0',...,a_m,a_m'$,即 $a_0=c_1,a_0'=c_2,...,a_m=c_{2m+1},a_m'=c_{2m+2}$,以及 $b_0,b_0',...,b_m,b_m'$ 为 $d_1,...,d_{2m+2}$,则有

$$E=\sum_{x_i\in\{c_i-1,d_i\},1\le i\le 2m+2}\prod_{j=1}^{2m+2}s(j)\times p[x_1,x_2,...,x_{2m+2}],$$

其中,

$$s(j)=\begin{cases}1, & 当x_j=d_j;\\ -1, & 当x_j=c_j-1;\end{cases}$$

对于由标记路径转化而来的 c,若 $c=n_1n_2...n_k$,则 $c-1=n_1n_2...(n_k-1)$.

4 实验及结果

本实验的目的是:① 验证使用基于小波的多维直方图方法对 XML 查询中的值匹配条件进行查询代价估计的可行性和准确性;② 分析基于小波的多维直方图占用空间大小和代价估计性能之间的关系.

4.1 实验环境

本文中的所有实验都完成于一台具有 PⅣ1.7GHz 的 CPU,512Mb 的 SDRAM 内存,80G 硬盘的 PC 机上,采用的操作系统是 Windows XP 系统.

4.2 实验数据集

为确保实验的准确性,我们同时采用两种类型的实验数据集,一种是真实数据集 DBLP[11],另一种是合成数

据集 XMark[12].

① DBLP.是一种真实存在的XML数据集,包含了DBLP数据库中的参考文献等有关信息,其特点是数据结构相对简单,数据分布也较均匀;

② Xmark.是一种合成 XML 数据集,包含了有关商业拍卖网站的一些信息.在本文中,我们采用了 scale 为 0.1 做 benchmark 数据生成器产生了 10M 左右的数据集.与 DBLP 相比,其数据结构比较复杂(树型较宽,深度也较深),数据分布存在较大的倾斜.

本文选取的 DBLP 数据集大小为 30Mbyte,其中包含 1 399 765 个节点.选取的 XMark 数据集大小为 11Mbyte,其中包含 206 130 个节点.

4.3 实验查询

实验中我们综合使用了多种形式的查询,包括简单路径表达中的值匹配条件和复杂路径表达中的值匹配条件,以及多个标记的等值匹配条件查询和值域匹配条件的查询,使用多种形式的查询,其目的在于验证基于小波的多维直方图方法是否可以适应各种不同条件下的值匹配条件.由于本文实验中只实现了值匹配条件和标记路径之间的相互依赖关系,而没有考虑值匹配条件之间的相互依赖关系,因此实验查询中也都只包含一个值匹配条件,具体采用的实验查询见表 1、表 2.

Table 1 Experimental queries (XMark)

表 1 实验查询(XMark)

	Queries
Q1	Site/regions/asia/item[quantity≥2]/name
Q2	Site/open_auctions/open_auction[quantity=1]/type
Q3	Site/open_auctions/open_auction[1000≤initial≤10000]/type
Q4	Site/close_auctions/close_auction[1000≤initial≤10000]/seller
Q5	Site/regions//item[quantity≥2]/name

Table 2 Experimental queries (DBLP)

表 2 实验查询(DBLP)

	Queries
Q1	Dblp/article[year≥1998]/author
Q2	Dblp/article[year=1993]/title
Q3	Dblp/inproceedings[1993≤year≤1998]/booktitle
Q4	Dblp/proceedings[1600≤volume≤1800]/publisher
Q5	Dblp//inproceedings[1990≤year≤1993]/title

4.4 实验结果及分析

实验中我们分析了直方图占用空间从120byte~720byte时共6种情况下基于小波的多维直方图进行查询代价估计的结果,因为每种数据集中采用的实验查询都涉及到两种标记的值元素,因此需要为这两种值元素分别建立多维直方图,而我们所建立的直方图又为二维直方图(主键值元素和其路径),每个小波系数对应的项保存 3 个值信息,因此 120byte 的占用空间相当于为每种值元素的多维直方图保存了 20 个小波系数,即 m=20,720byte 的占用空间相当于为每种值元素的多维直方图保存了 120 个小波系数,即 m=120.图 2 给出了不同的占用空间大小时利用基于小波的多维直方图法进行查询代价估计的相对误差率.

由图 2 可以看出,当多维直方图占用的空间不断增大时,即我们在其中保存越来越多的小波系数时,利用基于小波的多维直方图法进行值匹配条件查询的代价估计的误差率在稳定地不断下降,对于任何一个查询都是如此.使用这种方法我们获得了相当准确的代价估计结果,由图 2 可得,在 XMark 数据集中,以查询 Q3 为例,当多维直方图大小为 180byte 时,查询代价估计误差率已下降到 15%以下,当多维直方图大小为 720byte 时,几乎所有查询的代价估计误差率都下降到了 10%以下.此外,我们还可以看出,对复杂路径表达查询中的值匹配条件,利用基于小波的多维直方图方法进行代价估计时,其误差率比简单路径表达查询中的值匹配条件要高,例如 XMark 中的查询 Q5 与 Q3,这是因为对于复杂路径表达查询,我们在给出相应的路径匹配条件时包含了比真实查询中更多的路径,因此也就造成了其代价估计结果比实际的肯定要高,最终产生的代价估计误差也就随之增大.而且相对于值域匹配条件,对 XML 查询中的等值条件的查询代价估计误差率要高一些,例如,XMark 中的查询 Q2 和 Q1,这是由基于小波的直方图方法本身的特点所决定的.此外还可以看出,我们从 DBLP 数据集中得到的查询代价估计误差率几乎总比 XMark 数据集中要小,这是因为 DBLP 数据集相对于 XMark 来说其数据分布(值信息分布)比较均匀.总之,对于在实验中采用的所有查询,利用基于小波的多维直方图方法进行代价估计时,当多维直方图大小为 700Kbyte 左右时,其误差率也都下降到了 15%以下.

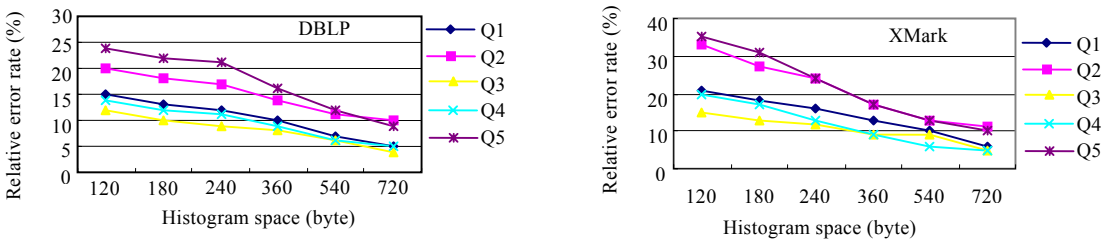

Fig.2 The relation between space and relative error rate

图 2 基于小波的多维直方图占用空间与相对误差率之间的关系

以上实验证明:基于小波的多维直方图是一种有效、准确的对 XML 查询中的值匹配条件进行查询代价估计的方法,而且随着其占用空间的增加,其代价估计误差率也逐渐降低.

5 结 论

XML 查询中的值匹配查询条件的查询代价估计问题是一种典型的多元素查询条件代价估计问题.而且,这种多元素查询条件与传统关系型数据库中的多元素查询条件不同,因为 XML 数据中的某一值信息分布不仅与 XML 数据中的其他值信息分布相关,还与 XML 数据中的结构信息相关,而且当 XML 数据结构比较复杂时,可能会形成高维元素相关.针对以上问题,本文提出了一种利用基于小波的多维直方图完成对 XML 查询中的值匹配查询条件的查询代价估计的方法,并提出了确定以某值元素为主键的多维元素相互依赖元组的方法,将值匹配条件改写为多元素查询条件的方法以及结构信息的值化方法等.实验证明,本文提出的方法取得了较准确的查询代价估计结果.

References:

[1] McHugh J, Widom J. Query optimization for XML. In: Proc. of the VLDB. 1999. 315−326. http://www.vldb.org/conf/1999/P32.pdf

[2] Chen Z, Jagadish HV, Korn F, Koudas N, Muthukrishnan S, Ng RT, Srivastava D. Counting twig matches in a tree. In: Proc. of the ICDE. 2001. 595−604. http://citeseer.ist.psu.edu/chen01counting.html

[3] Aboulnaga A, Alameldeen AR, Estimating JN. The selectivity of XML path expressions for Internet scale applications. In: Proc. of the VLDB. 2001. 591−600.

[4] Wu Y, Patel JM, Jagadish HV. Estimating answer sizes for xml queries. In: Proc. of the EDBT. 2002.

[5] Polyzotis N, Garofalakis M. Statistical synopses for graph structured XML databases. In: Proc. of the SIGMOD. 2002.

[6] Freire J, Haritsa JR, Ramanath M, Roy P, Sim_eon J. StatiX: Making XML count. In: Proc. of the 2002 ACM SIGMOD Int'l. Conf. on Management of Data. 2002.

[7] Muralikrishna M, Dewitt DJ. Equi-Depth histograms for estimating selectivity factors for multi-dimensional queries. In: Proc. of the ACM SIGMOD Conf. 1988. 28−36.

[8] Poosala V, Ioannidis Y. Selectivity estimation without the attribute value independence assumption. Technical Report, Bell Labs, 1997.

[9] Deshpande A, Garofalakis M, Rastogi R. Independence is good: Dependency-Based histogram synopses for high-dimensional data. In: Proc. of the ACM SIGMOD 2001. 2001. 199−210. http://www.bell-labs.com/user/ minos/Papers/sigmod01dbhist-cam.pdf

[10] Vitter JS, Wang M. Approximate computation of multidimensional aggregates of sparse data using wavelets. ACM SIGMOD, 1999.

[11] DBLP Computer Science Bibliography. http://www.informatik.uni-trier.de/~ley/db/

[12] Schmidt A, Waas F, Kersten M, Florescu D, Manolescu I, Carey M, Busse R. The XML benchmark project. Technical Report INSR0103, 2001.