ISSN 1000-9825, CODEN RUXUEW
*Journal of Software*, Vol.19, No.2, February 2008, pp.179−193
DOI: 10.3724/SP.J.1001.2008.00179

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel/Fax: +86-10-62562563

# 一种基于图模型的Web数据库采样方法*

刘 伟, 孟小峰+, 凌妍妍

(中国人民大学 信息学院,北京 100872)

## A Graph-Based Approach for Web Database Sampling

LIU Wei, MENG Xiao-Feng+, LING Yan-Yan

(School of Information, Renmin University of China, Beijing 100872, China)

+ Corresponding author: Phn: +86-10-62519453, E-mail: xfmeng@ruc.edu.cn, http://idke.ruc.edu.cn/xfmeng/

**Liu W, Meng XF, Ling YY. A graph-based approach for Web database sampling. *Journal of Software*, 2008, 19(2):179−193.** http://www.jos.org.cn/1000-9825/19/179.htm

**Abstract**: A flood of information is hidden behind the Web-based query interfaces with specific query capabilities, which makes it difficult to capture the characteristics of the Web database, such as the topic and the frequency of updates. This poses a great challenge for Deep Web data integration. To address this problem, a graph-based approach WDB-Sampler for Web database sampling is proposed in this paper, which can incrementally obtain sample records from a Web database through its query interface. That is, a number of samples are obtained for the current query, and one of them is transformed into the next query. The important characteristic of this approach is it can adapt to different kinds of attributes on the query interfaces. The extensive experiments on the local simulation Web databases and the real Web databases prove that the approach can achieve high-quality samples from a Web database at a lower cost.

**Key words**: deep Web; Web database; database sampling

**摘 要**: Web 数据库中,海量的信息隐藏在具有特定查询能力的查询接口后面,使人无法了解一个 Web 数据库内容的特征,比如主题的分布、更新的频率等,这就为 Deep Web 数据集成带来了巨大的挑战.为了解决这个问题,提出了一种基于图模型的 Web 数据库采样方法,可以通过查询接口从 Web 数据库中以增量的方式获取近似随机的样本,即每次查询获取一定数量的样本记录,并且利用已经保存在本地的样本记录生成下一次的查询.该方法的一个重要特点是不受查询接口中属性表现形式的局限,因此是一种一般的 Web 数据库采样方法.在本地的模拟实验和真实 Web 数据库上的大量实验表明,该方法可以在较小代价下获得高质量的样本.

**关键词**: deep Web;Web 数据库;数据库采样

**中图法分类号**: TP311 **文献标识码**: A

* Supported by the National Natural Science Foundation of China under Grant No.60573091 (国家自然科学基金); the National High-Tech Research and Development Plan of China under Grant No.2007AA01Z155 (国家高技术研究发展计划(863)); the Beijing Natural Science Foundation of China under Grant No.4073035 (北京市自然科学基金); the Program for New Century Excellent Talents in University of China (新世纪优秀人才支持计划)

Received 2007-09-03; Accepted 2007-10-19

Web 的迅速发展使其成为一个巨大的信息源.按照信息蕴含的深度,整个 Web 可分为 Surface Web 和 Deep Web 两大部分.Surface Web 是指 Web 中能够被传统搜索引擎(如 google,yahoo 等)索引到的内容,而 Deep Web 则是指不能被传统搜索引擎索引到的内容,这些内容主要存储在 Web 中大量可在线访问的 Web 数据库中.根据文献[1]在 2004 年的调查,目前整个 Web 中至少有 450 000 个可访问的 Web 数据库,并且数量仍在迅速增长,其中存储的信息覆盖了现实世界的各个领域,如商业、医学、体育等.Deep Web 中的信息量是 Surface Web 的 550 倍之多[2],这使得 Deep Web 成为人们获取信息的一个重要途径.用户对 Web 数据库的访问主要是通过其在 Web 页面中提供的具有特定查询能力的接口来获取所需要的结果.比如,购书网站当当(http://home.dangdang.com/)就是一个典型的商业 Web 数据库,图 1(a)所示为该网站提供的图书查询接口.如果用户想要购买数据库方面的图书,只需在属性"书名"上填写关键词"数据库"并提交,当当网就会动态生成包含符合该查询条件的查询结果的网页(如图 1(b)所示)返回给用户.

(a) Query interface page

(a) 查询接口页面

(b) Query result page

(b) 查询结果页面

Fig.1 Examples of Web database

图 1 Web 数据库示例

由于对 Web 数据库特有的访问方式,使得传统的搜索引擎无法有效地索引到其中的内容.为了帮助用户能够有效地利用 Deep Web 中的海量信息,研究者们展开了对 Deep Web 数据集成的研究,即建立一个 Deep Web 数据集成系统.该系统可以为用户提供一个集成查询接口,并把各个 Web 数据库返回的结果合并到一个统一的模式下.至今,在该研究领域已经取得了若干成果,比如查询接口集成[3,4]、Web 数据库的分类[5,6]、Web 数据的抽取[7,8]等.

由于 Deep Web 的规模巨大,使得 Deep Web 数据集成系统[9]中会集成上百甚至上千个 Web 数据库,极地地超过了传统数据集成系统中数据源的数量.然而,由于对 Web 数据库的访问只能通过其提供的具有特定查询能力的查询接口,给我们对 Web 数据库的了解带来了困难.这就需要对 Web 数据库进行采样,通过这个样本,我们可以了解该 Web 数据库的主题分布、更新频率以及大小等有用的特征.举个例子,在 Deep Web 数据集成系统中,对于一个给定的用户查询,事实上:(1) 有些 Web 数据库并不满足该查询,无须对其查询;(2) 有些 Web 数据库之间存在着较大的冗余,只选择其中 1 个或几个查询.如果用户查询被集成系统不加选择地直接分发到每个 Web 数据库中,则不但查询代价高,而且会返回大量冗余的结果,造成系统不必要的负担和用户等待时间过长.因此,要为一个给定的查询选择合适的 Web 数据库.一个可行的解决方案是:从每个 Web 数据库中获得一份样本并保存在本地,对用户的查询首先用样本代替对应的 Web 数据库进行考察,从而选择出合适的 Web 数据库进行真正的查询.

数据库采样是一个从数据库中随机选取记录的过程,可以获得数据库的有用的统计信息.对于不受限制的访问方式,过去已经提出了许多方法可以有效地从数据库中随机地采样[10−13].然而,由于对 Web 数据库的访问只能通过其提供的具有约束的查询接口,无法自由地从 Web 数据库中获取记录,因而传统的方法不利于实现对

Web 数据库的采样,给我们带来了巨大的挑战.

HIDDEN-DB-SAMPLER[14]是目前唯一的一项针对 Web 数据库采样而提出的工作,它存在一个明显的局限性就是只能处理数字和分类属性,即属性的取值是可数的或有限的(对范围属性的处理是把取值范围划分为若干小的离散的范围,比如年龄属性可划分为 0~10,11~20,…,91~100).而对于可以任意取值的关键词属性并没有给出相应的处理办法,比如图书领域中的书名、作者、出版社等属性.事实上,很多领域存在着大量的关键词属性,而且其中一些关键词属性通常被查询接口的设计者规定为必填的,比如图书领域的书名和工作领域的职位.表 1 列出了一些常见领域中存在的可任意取值的属性.当查询接口中存在这样的属性时,文献[14]所提出的方法则根本不能实现对该 Web 数据库的采样.

**Table 1** Examples for the attributes with random values

**表 1** 可任意取值的属性举例

| Domain | Attributes |
|---|---|
| Book | Title, Author, Publisher |
| Movie | Title, Directors, Actors |
| Job | Position, Company |

本文提出了一种一般的 Web 数据库采样方法,不受查询接口中属性表达形式的任何限制:即使查询接口中存在可任意取值的必填属性,我们的方法也同样能够实现对 Web 数据库的近似随机采样.由于 Web 数据库的特殊性,只能通过其所提供的查询接口获取记录,因此受查询接口的查询能力的限制,无法从 Web 数据库中自由地获取记录,这使得传统的随机采样方法无法应用.基于这种考虑,我们提出一种增量式的 Web 数据库采样方法 WDB-Sampler,其基本思想主要分为 4 步(如图 2 所示):第 1 步,从一个任意的有效查询(有返回结果)开始查询;第 2 步,从返回的查询结果页面中抽取记录;第 3 步,将这些记录放入本地样本库;第 4 步,从样本库中选取一个记录,将它转化为对 Web 数据库的下一次查询,转第 2 步.第 2 步的实现是属于从 Web 页面中抽取结构化数据的问题,至今已有许多工作[7,8],我们利用这些工作,因此对这一部分就不再加以介绍.

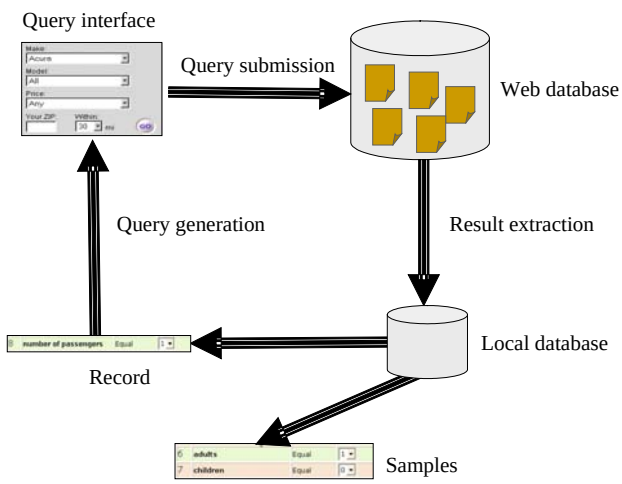



Fig.2 WDB-Sampler: Web database sampling process

图 2 WDB-Sampler:Web 数据库采样流程

对 Web 数据库的采样主要需要面对两个挑战:一是所得样本的偏差,即使样本的数据分布与 Web 数据库保持一致;二是获取样本的代价,即以尽可能少的查询次数来获得这些样本.针对这两个挑战,WDB-Sampler 在理论上需要考虑的问题有如下 3 个:样本的选取,即每次的查询结果中选取哪些记录作为样本;查询的选择,即每次的查询结果中选取哪一个记录作为下一次查询;采样的终止,即采样过程的终止条件是什么.

上面 3 个问题的解决将直接影响样本的客观性和获取样本的效率.为了有效地解决这些问题,我们提出以

图游历的方式指导对图 2 所示的 Web 数据库的采样过程.这里的“游历”与通常所说的“遍历”是有区别的,即每次只访问 1 个顶点的部分邻接顶点.我们后面还会对所提出的图模型以及图游历的过程作更正式、更详细的定义和描述.

总之,本文主要是在理论层面上给出这些问题的解决方案.本文的贡献主要总结为如下 3 个方面:

- 提出了一种一般的 Web 数据库采样方法 WDB-Sampler.该方法采取增量采样的方式,不受查询接口中对属性表现形式的影响.
- 提出一种新的 Web 数据库的图模型,并通过图游历的方式实现对 Web 数据库增量式的采样.
- 通过在本地模拟 Web 数据库和真实 Web 数据库上广泛的实验,其实验结果表明,我们提出的 Web 数据库采样方法可以同时保证样本的质量和采样的效率.

## 1 预备知识

### 1.1 相关符号定义

在现实中,Web 数据库大都是用当前流行的关系数据库实现的,比如 Oracle,My SQL 等.因此在本文中,我们把 Web 数据库看作一个关系模式的数据表.为使描述下为简洁,我们首先对本文用到的 Web 数据库相关的概念进行符号化定义,见表 2.

**Table 2** Symbols and their remarks

**表 2** 相关符号及其注释

| Symbol | Remarks | Symbol | Remarks |
|---|---|---|---|
| *WDB* | Web database | $A(R)$ | Attribute set of $R$ |
| $I$ | The query interface of WDB | $A(IR)$ | $A(I)\cap A(R)$ |
| $R$ | The records in WDB | $A(Q)$ | The attribute set of $Q$ |
| $A$ | Attribute set, denoted to be $\{a_1,a_2,\ldots\}$ | $R(Q)$ | The records of WDB satisfying query $Q$ |
| $Q$ | An available query on $I$ | $S$ | A sample set of WDB, $S\in WDB$ |
| $A(WDB)$ | The attribute set of WDB | $R_S(Q)$ | The records of $S$ satisfying query $Q$ |
| $A(I)$ | The attribute set of $I$ | $Cost(S)$ | The cost of obtaining $S$ from *WDB* |

需要指出的是,在现实中,并非查询接口中的所有属性都出现在查询结果的记录中,即 $A(I)\neq A(R)$.一般情况下,我们认为 $A(WDB)=A(I)\cup A(R)$,因为既不在查询接口中也不在查询结果记录中出现的属性对于 Web 数据库的使用者来讲是没有任何意义的.如果不加特殊说明,$A(Q)$则是指在查询接口和查询结果的记录中都出现的属性,即 $A(Q)=A(I)\cap A(R)$,因为我们无法对只在查询接口中出现的属性获取它在 Web 数据库中的取值.

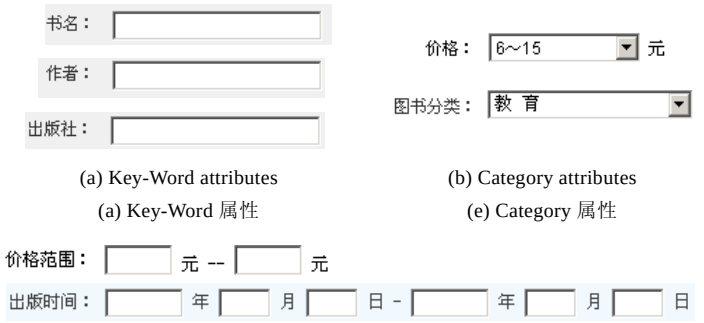
### 1.2 属性分类及查询表达

根据我们的观察,查询接口中不同的属性有着不同的表达形式,对查询结果的影响也并不相同.按照填写方式以及对查询结果的影响,通常可以把这些属性分为关键词属性、范围属性和分类属性 3 类.

- 关键词属性:用户可以在该属性上随意填写的文本,表示为 $a_q$ like $v_q$,其中 $v_q$ 为由 1 个或多个关键词组成的文本.
- 范围属性:用户可以在该属性上自由填写两个值(比如数字或日期),表示一个范围,表示为 $v_{q1}<a_q<v_{q2}$,其中 $v_{q1}$ 和 $v_{q2}$ 为两个数字.
- 分类属性:用户可以从该属性上有限个互斥的值中选取 1 个或多个,表示为 $a_q=v_q$,其中 $v_q$ 是分类属性中的一个值.

图 3 是我们从真实的图书领域的 *WDB* 查询接口中获得的 3 类属性的典型示例.需要注意的是,一个属性的语义并不能决定它属于哪种类型,同一语义的属性在不同 *WDB* 的查询接口经常会表示成不同的类型,这是由 *WDB* 的设计者决定的.比如,图 3(b)中的第 1 个属性看起来似乎是一个范围属性,但用户并不能自由填写范围,只能从有限的几个选择中选取,因此它是一个分类属性.图 3(c)中的第 1 个属性与图 3(b)中的第 1 个属性尽管都表示价格,但它却是典型的范围类型.因此,我们的方法是与属性的语义无关的,所关心的只是它在查询接口上

的表现形式.




(a) Key-Word attributes (b) Category attributes

(a) Key-Word 属性 (e) Category 属性

(c) Range attributes

(c) Range 属性

Fig.3 Examples of attribute category

图 3 属性分类示例

通过对大量真实查询接口的观察,我们给出一个合理的假设:在查询接口中,(1) 属性之间是“与”关系,即返回的查询结果必须同时满足用户在查询接口中给出的所有属性值的约束;(2) 在关键词属性上,关键词之间是“与”关系.根据我们对大量实际 *WDB*,特别是对电子商务 *WDB* 的观察,比如大家熟悉的国内的当当、淘宝,以及国外的亚马逊等,该假设是符合现实情况的.其原因很简单:*WDB* 更加注重返回结果的质量,而不是使用户淹没在大量无关的结果中.

基于上述的符号表达,对 *WDB* 的每一次查询 *Q*,可以用 SQL 的语法描述如下:

SELECT $a_{r1},a_{r2},\ldots,a_{rn}$

FROM *WDB*

WHERE … and $\langle a_{q1}$ like $v_{qi}\rangle,\ldots$, and $\langle v_{qj}<a_{qj}<v_{qj}\rangle,\ldots$, and $\langle a_{qk}=v_{qk}\rangle,\ldots$

其中,$a_{ri}(1<i<m)$表示在结果中出现的属性;$a_{qi},a_{qj},a_{qk}$ 分别表示在查询接口中出现的关键词属性、范围属性和分类属性.由于 SELECT 子句和 FROM 子句的内容对于 *WDB* 而言是确定的、不可改变的,因此,*Q* 可以进一步简化表示为$\{\ldots,\langle a_i$ like $v_i\rangle,\ldots,\langle v_{j1}<a_j<v_{j2}\rangle,\ldots,\langle a_k=v_k\rangle,\ldots\}$.

如果一个记录 *R* 满足给定一个查询 *Q*,由于 *WDB* 返回的结果集中所有记录必然会满足它,因此,对于 $\forall a_i\in S(Q)$,对它在查询结果中的值 $v_{rj}$,则满足下面 3 种情况:

- 如果是 $a_i$ 关键词属性,则 $v_{rj}\cap v_i\neq\varnothing$;
- 如果是 $a_i$ 范围属性,则 $v_{j1}<v_{rj}<v_{j2}$;
- 如果是 $a_i$ 分类属性,则 $v_{rj}=v_i$.

### 1.3 样本偏差评价方法

由于只能通过具有特定查询能力的查询接口来获取 *WDB* 中的记录,因此在理论上是无法保证 *S* 真正的客观性的.直觉上,如果 *S* 具有真正的随机性,则对于任何一个查询 $Q_i$,必然满足下面的关系:

$$\frac{|R(Q_i)|}{|R_S(Q_i)|}\approx\frac{|WDB|}{|S|}.$$

为了能够客观地评价 *S* 的质量,我们需要一组随机的查询 $RandomQ\{Q_1,Q_2,\ldots,Q_n\}$,分别在 *WDB* 和 *S* 中进行查询,通过比较同一查询下各自查询结果的数量,得到对样本偏差的一个评价,我们给出形式化的公式:

$$Quality(S)=\frac{\sqrt{\sum_{i=1}^{n}\left(\frac{|R(Q_i)S|}{|R_S(Q_i)WDB|}-1\right)^2}}{n} \tag{1}$$

*S* 为空无意义,因此,$|S|>0$;另外,如果 $R_S(Q_i)$为空,则为其赋一个足够小的值来代替.

通过公式(1)我们可以看出,对于一个 *WDB* 的样本集合 *S*,*Quality*(*S*)越趋近于 0,表示该样本记录集合的偏差越小;反之,*Quality*(*S*)越大,表示样本记录集合偏差越大,客观性越低.

不可否认的是,通过公式(1)对 *S* 的评价结果与 *RandomQ* 有直接的关系.极端情况下,如果对于每个查询 $Q_i$, *WDB* 只返回一个记录,那么样本记录集合将返回 0 个或 1 个记录,这会使偶然性增大.因此,在实验部分我们将介绍如何产生随机的查询.

### 1.4 采样代价

除了保证样本集合 *S* 的质量以外,另一个关键问题是采样的代价 *Cost*(*S*).在本文中,我们简单地把它定义为获得 *S* 通过查询接口向 *WDB* 提交查询的次数.这是因为一般情况下,本地的执行时间远小于 Web 中的网络数据传输时间,由于 *WDB* 的自主性,无法用一个常量来表示每次查询的执行时间,所以我们以向 *WDB* 提交查询的次数来代替.文献[14,15]也都采用相同的代价估算方法.

需要注意的是,在现实中,*WDB* 经常因返回查询结果的记录数量较多而分多次提供给用户,即在结果页面中每次只返回 *k* 个记录,如果用户想得到后续的记录,则必须通过翻页得到,这就相当于发送一次查询请求.比如,从 Amazon 查询有关“Java”的图书,共得到 46 979 个记录,每次只返回用户 12 个记录,用户如果要看到更多的结果,就要通过不断地翻页来实现,而每次翻页就相当于向 Amazon 发送了一次查询请求.

实际上,样本质量和采样代价是矛盾的:如果为了提高样本质量,必然要提交更多的查询来获得更多的记录,就会使采样代价提高.正如前面已经提到过的,针对这个矛盾,本文的目的是如何以尽可能小的代价来获得尽可能高质量的样本.

## 2 一种Web数据库图模型

本节我们提出一种新的 Web 数据库图模型,通过该模型我们以图游历方式达到对 Web 数据库采样的目的.我们首先提出若干相关定义、性质和定理,然后对这种 Web 数据库图模型进行深入讨论.

**定义 1(强查询)**. 对于两个查询 $Q_1$ 和 $Q_2$,如果 $A(Q_1)\supseteq A(Q_2)$,且对于$\forall a_i\in A(Q_2)$满足下面 3 个条件,那么,我们称 $Q_1$ 是 $Q_2$ 的一个强查询:

- 如果是 $a_i$ 关键词属性,则 $Q_1$ 在 $a_i$ 上的值是 $Q_2$ 在 $a_i$ 上值的超集;
- 如果是 $a_i$ 范围属性,则 $Q_1$ 在 $a_i$ 上的取值范围是 $Q_2$ 在 $a_i$ 上的子范围;
- 如果是 $a_i$ 分类属性,则 $Q_1$ 在 $a_i$ 上的值等于 $Q_2$ 在 $a_i$ 上的值.

例:表 3 给出 3 个关于图书的查询示例,其中涉及了文本(书名)、数字(价格)和分类(类别)3 类属性.$Q_1$ 分别是 $Q_2$ 和 $Q_3$ 的强查询.

**Table 3** Examples of strong query

**表 3** 强查询示例

| | Title | Price | Category |
|---|---|---|---|
| $Q_1$ | Thinking in Java | 〈30,40〉 | Computer |
| $Q_2$ | Thinking in Java | Null | Null |
| $Q_3$ | Java | 〈20,50〉 | Computer |

根据定义 1,我们可以得到下面两个性质:

**性质 1(包含性)**. 如果 $Q_1$ 是 $Q_2$ 的强查询,那么 $R(Q_1)\subset R(Q_2)$.

证明:$\forall$记录 $R_i\in R_{Q_1}$,对于属性 $a_j\in A(R_i)\cap A(Q_2)$的值 $v_j$ 分别作如下考虑:

- 如果 $a_j$ 是关键词属性,则由于 $v_j$ 是 $Q_1$ 在 $a_i$ 上的值的超集,而 $Q_1$ 在 $a_i$ 上的值是 $Q_2$ 在 $a_i$ 上值的超集,所以 $v_j$ 与 $Q_2$ 在 $a_i$ 上的值的交集不为空;
- 如果 $a_j$ 是范围属性,则 $v_j$ 必然在 $Q_1$ 在 $a_i$ 上的取值范围内,而 $Q_1$ 在 $a_i$ 上的取值范围是 $Q_2$ 在 $a_i$ 上的子范围,$v_j$ 在 $Q_2$ 在 $a_i$ 上的取值范围内;
- 如果 $a_j$ 是分类属性,则 $v_j$ 等于 $Q_1$ 在 $a_i$ 上的值,而 $Q_1$ 在 $a_i$ 上的值等于 $Q_2$ 在 $a_i$ 上的值,所以 $v_j$ 等于 $Q_1$ 在

$a_i$ 上的值.

从以上 3 个方面可以得出,$R_i$ 必然满足 $Q_2$,即 $R(Q_1)\subset R(Q_2)$. □

**性质 2(传递性)**. 如果 $Q_1$ 是 $Q_2$ 的强查询,而 $Q_2$ 是 $Q_3$ 的强查询,那么 $Q_1$ 必然是 $Q_3$ 的强查询.

证明:根据定义 1,$A(Q_1)\subseteq A(Q_2)\subseteq A(Q_3)$.对于属性 $a_j\in S(Q_1)$分别作如下考虑:

- 如果 $a_j$ 是关键词属性,$Q_1$ 在 $a_i$ 上的值是 $Q_2$ 在 $a_i$ 上值的超集,而 $Q_2$ 在 $a_i$ 上的值是 $Q_3$ 在 $a_i$ 上值的子集,则 $Q_1$ 在 $a_i$ 上的值是 $Q_3$ 在 $a_i$ 上值的超集;
- 如果 $a_j$ 是范围属性,$Q_1$ 在 $a_i$ 上的取值范围是 $Q_2$ 在 $a_i$ 上的子范围,而 $Q_2$ 在 $a_i$ 上的取值范围是 $Q_3$ 在 $a_i$ 上的子范围,则 $Q_1$ 在 $a_i$ 上的取值范围是 $Q_3$ 在 $a_i$ 上的子范围;
- 如果 $a_j$ 是分类属性,$Q_1$ 在 $a_i$ 上的值等于 $Q_2$ 在 $a_i$ 上的值,而 $Q_2$ 在 $a_i$ 上的值等于 $Q_3$ 在 $a_i$ 上的值,则 $Q_1$ 在 $a_i$ 上的值等于 $Q_3$ 在 $a_i$ 上的值.

从以上 3 个方面可以得出,$Q_1$ 必然是 $Q_3$ 的强查询. □

**定义 2(弱查询)**. 如果 $Q_1$ 是 $Q_2$ 的一个强查询,那么 $Q_2$ 是 $Q_1$ 的一个弱查询.

根据定义 2,我们同样可以得到与上面两个性质类似的性质,这里不再赘述和证明.

**定义 3(查询相关记录)**. 给定一个记录集合$\{R_1,R_2,\ldots,R_n\}$,如果通过提交某一个查询 $Q$,使得它们可以同时出现在同一个查询结果中,则称它们彼此是关于查询 $Q$ 相关的;反之,则称它们是非查询相关的.

例:设一个 *WDB* 的查询接口可以根据图书的名称、作者和类别 3 个属性进行查询.表 4 给出了 3 个记录,其中,$R_1$ 和 $R_2$ 可以通过作者属性同时查询出来,而 $R_3$ 则不能通过任何查询与 $R_1$ 或 $R_2$ 同时出现在同一查询结果集中,因此,我们说 $R_1$ 和 $R_2$ 是查询相关的,而 $R_3$ 与 $R_1$ 或 $R_2$ 是查询不相关的.通过这个例子也可以看出,两个记录是否查询相关,与查询接口的表达能力是密切相关的,如果查询接口中有出版社的属性,而 $R_3$ 与 $R_1$ 又恰好是同一出版社出版的,那么它们就是查询相关的了.

**Table 4** Examples of query-related records

**表 4** 查询相关记录示例

| | Title | Author | Catrgory |
|---|---|---|---|
| $R_1$ | Thinking in Java | Bruce Eckel | Computer |
| $R_2$ | Thinking in C++ | Bruce Eckel | Computer |
| $R_3$ | Harry potter | J.K. Rowing | Novel |

**定理 1**. 给定一个查询相关的记录集合 $R\{R_1,R_2,\ldots,R_n\}$,设 $Q\{Q_1,Q_2,\ldots,Q_m\}$为 $R$ 所有相关查询的集合.如果 $Q$ 非空,那么必然存在一个查询 $Q_i(1\le i\le m)$是这个集合中其他任意一个查询 $Q_j(1\le j\le m,j\ne i)$的强查询.我们把 $Q_i$ 称作记录集合 $R$ 的最强查询.

证明:证明分为两步:首先构造一个特定的查询 $Q_i(1\le i\le m)$,然后证明该查询为 $Q$ 中的其他任意一个查询 $Q_j(1\le j\le m,j\ne i)$的强查询.

第 1 步,构造查询 $Q_i$:需要确定 $Q_i$ 中有哪些属性以及每个属性上的取值.

对于$\forall a_k\in A(IR)$,考虑如下:

- 如果 $a_k$ 是关键词属性,并且其中各个记录在 $a_k$ 上的取值有非空的交集,则 $Q_i$ 包含 $a_k$,并且在 $a_k$ 上的取值为各个记录在 $a_k$ 上的取值的交集;如果交集为空,则 $Q_i$ 不包含 $a_k$.
- 如果 $a_k$ 是范围属性,则 $Q_i$ 包含 $a_k$,并且 $Q_i$ 在 $a_k$ 上的取值范围介于 $R$ 中记录在 $a_k$ 上取值的最大值和最小值之间.
- 如果 $a_k$ 是分类属性,如果 $R$ 中各个记录在 $a_k$ 上的取值相同,则 $Q_i$ 包含 $a_k$,并且 $Q_1$ 在 $a_k$ 上也取该值;否则,$Q_i$ 不包含 $a_k$.

第 2 步,证明 $Q_i(1\le i\le m)$是这个集合中其他任意一个查询 $Q_j(1\le j\le m,j\ne i)$的强查询.

对于$\forall a_k\in A(Q_j)$作如下考虑:

- 如果 $a_k$ 是关键词属性,则 $Q_j$ 在 $a_k$ 上的所有关键词必然出现在 $R$ 中每一个记录在 $a_k$ 上的值中,所以也必然出现在 $Q_i$ 在 $a_k$ 上的关键词集合中;

- 如果 $a_k$ 是范围属性,则 $R$ 中每一个记录在 $a_k$ 上的值必然在 $Q_j$ 于 $a_k$ 上的取值范围内,所以,$Q_i$ 在 $a_k$ 上的取值范围是 $Q_j$ 在 $a_k$ 上的子范围;
- 如果 $a_k$ 是分类属性,则 $R$ 中每一个记录在 $a_k$ 上的值必然等于 $Q_j$ 在 $a_k$ 上的取值,所以,$Q_i$ 在 $a_k$ 上的取值等于 $Q_j$ 在 $a_k$ 上的取值.

因此,根据定义 1,$Q_i$ 是 $Q_j$ 的强查询.进而定理得证. □

**定义 4(Web 数据库图模型(Web database graph,简称 WG)**. 对于一个给定的 *WDB*,它的图模型表示为 *WG*(*V*,*E*).其中,*V* 是顶点的集合,每个顶点 $v_i$ 都与 *WDB* 的记录 $R_i$ 一一对应,即|*V*|=|*WDB*|.*E* 是无向边的集合,如果两个记录是查询相关的,那么,它们对应的顶点之间存在一条相连的边.对于每一个顶点,附加对应记录的最强查询;对于每一条边,附加的查询为该边所连接的两个顶点对应记录的最强查询.

在该图模型中,每个顶点和每条边都附加了一个唯一的查询,查询的生成方法在定理 1 中已经给出.对于每个顶点,相当于记录集合 *R* 中只有顶点对应的那个记录;对于每条边,相当于记录集合 *R* 中只有该边所连接的两个顶点对应的记录.

需要指出的是,WG 是与 *WDB* 所提供的查询接口的查询能力密切相关的,因为前面我们已经指出,本文中涉及的查询都是查询接口中可表达的.因此,两个记录在 WG 中是否存在一条边,取决于是否存在一个查询接口可表达的查询使得这两个记录同时满足该查询.进一步来说,顶点和边上附加的查询也同样由查询接口的表达能力所决定.因此,对于两个具有相同内容的 *WDB*,如果它们的查询接口在查询能力上不同,那么所产生的 WG 也是不同的.实际上,已经有一些工作通过把一个数据库转化成图的形式去解决特定的问题,我们将在相关工作中进行简要的介绍.

## 3 基于WG的Web数据库采样方法

利用图模型 WG 可以把任意一个 *WDB* 在记录层次上转化为图的表示.我们可以从图中揭示出记录之间在查询上的关联关系.前面我们已经简要介绍了对 *WDB* 增量式的采样方法(如图 2 所示),并给出了需要解决的 3 个问题:样本的选取、查询的选择和终止条件.这一节,我们将详细讨论如何在 WG 中采用图游历的思想来实现对 *WDB* 增量式的采样.

由于我们无法越过查询接口直接得到 *WDB* 中的所有记录,因此也无法为一个给定的 *WDB* 建立真正的 WG.实际上,我们也并非有意为要采样的 *WDB* 建立一个完整的 WG,而是从当前 WG 中一个随机的点开始进行游历,实现对 *WDB* 的采样.基于 WG,采样过程的基本思想重新描述如下:

- 第 1 步,从任意一个查询 $Q_0$ 开始,并提交给 *WDB*;
- 第 2 步,把查询结果中的记录保存在本地中 $R_L$,对当前已经保存下来的记录 $R_L$ 建立 $WG_L$;
- 第 3 步,判断是否达到终止条件:如果是,则终止,否则进入下一步;
- 第 4 步,通过对当前 $WG_L$ 的分析,从 $R_L$ 中选取一个合适的记录形成下一次的查询,转第 1 步.

在第 1 步中,我们使用任意的查询 $Q_0$ 作为采样的开始,虽然 $Q_0$ 是通过人工产生,不可避免地带有主观性,但人工选择时只要保证 $Q_0$ 的查询结果足够多,比如大于 100,就保证了 $Q_0$ 是 *WG* 中度较大的一个点,从而可以避免不同初始查询带来的采样差异.

在第 2 步中,随着 $R_L$ 中保存记录数量的不断增多,$WG_L$ 也在不断地扩大(加入新的顶点和边),因此,我们对 $WG_L$ 采取增量式的维护方式,即把每次查询返回的记录添加到当前的 $WG_L$ 中,而不是每次对当前的 $R_L$ 重新构建 $WG_L$.显然,$WG_L$ 是 *WG* 的一个子图.对 $WG_L$ 的扩展,根据定义 4 很容易实现,这里就不再过多地介绍.

采样过程中最关键的步骤是第 3 步和第 4 步,共包括 3 个问题(在本文开始部分提到的 3 个问题):

(1) 如何根据 $WG_L$ 从当前的样本记录集合中选取一个合适的记录?

(2) 被选中的记录应该生成什么样的查询?

(3) 这个采样过程在什么情况下结束?

下面,我们首先根据基本思想形式化地给出 WDB-Sampler 的算法描述,然后针对上面的问题提出解决

策略.

### 3.1 WDB-Sampler算法

WDB-Sampler 算法是对采样过程形式化的描述(如图 4 所示),其思想前面已经给出,这里不再重复.其中有一点需要说明的是:对每次的查询结果,我们只获取第 1 页中的记录.这是因为我们基于如下两点考虑:第一,前面已经提到,要获取此次查询结果中更多的记录需要不断翻页,而翻页操作实际上相当于一次查询;第二,所有查询结果都是满足此次查询的,因此根据公式(1),会导致偏差急剧增大.

**Algorithm WDB-Sampler**
**Input**
$R_0$: A record in *WDB*.
**Output**
$S\{R_0,R_1,\ldots,R_n\}$: A set of records, which are samples from *WDB*.
**Begin**
initialize $WG_L$; // the initial $WG_L$ is empty
initialize $R_L$; // the initial $R_L$ is empty, and it is used to store the records obtained with each query
add $R_0$ to $R_L$;
$WG_L$=*buildGraph*($R_0$); //build $WG_L$ with only one node $R_0$
while *StopCriteria* is not reached do
$v_c$=*recordSelector*($WG_L$); //select an appropriate node (record) $v_c$ from $WG_L$ to generate the next query
$Q_c$=*queryGenerator*($v_c$); //generate the query $Q_c$ with the selected record $v_c$
$R_c$=*queryWDB*($Q_c$); //query *WDB* with $Q_c$ and then obtain the top-k records from the first result page
Add $R_c$ to $R_L$; //input $R_c$ into $R$ and remove the duplicated records
$WG_L$=*graphExpanding*($WG_L$); //expand $WG_L$ according to the fresh records
end
$S$=*amendDeviation*($R_L$);
return $S$;
**End**

Fig.4 WDB-Sampler: Web database sampling algorithm
图 4 WDB-Sampler:Web 数据库采样算法

算法中 recordSelector,queryGenerator 和 StopCriteria 分别对应着本节开始所提出的 3 个问题,它们也是该算法的关键.本节的剩余部分将对它们逐一进行讨论.

### 3.2 记录的选择(recordSelector)

为了产生下一次的查询,我们需要从当前保存在本地的记录集合中选取一个合适的记录作为查询,这就是 recordSelector 所要完成的功能.我们选择记录的目的是为了能让其产生的查询得到更多新的记录.从 $WG$ 的角度来看,就是从当前 $WG_L$ 中选取一个顶点 $v$,使得通过 $v$ 可以找到更多的不在 $WG_L$ 中的顶点.因此,我们的选择方案是:把 $WG_L$ 中的顶点按照它们的度从低到高排序,选取度最小的顶点.$WG_L$ 中顶点的度小,这有两种可能:在 $WG$ 中有很多与其邻接的顶点(新的记录)还没有访问到,或者该顶点在 $WG$ 中的度也小.因此,如果该顶点生成的查询获得记录数量小于 $k$,则丢弃该顶点并重新选择剩余顶点中度最小的.

### 3.3 查询的生成(queryGenerator)

当选定 $WG_L$ 中的一个顶点后,即从中选定了一个记录 $R_c$,下一步就是 queryGenerator 如何利用这个记录生成下一次的查询.显然,一个记录可以生成若干可能的查询.为了得到更多新的记录,对于每一个 $a_i \in A(IR)$,我们根据 $R_L$ 中的记录为其建立如下的统计信息:

- 如果 $a_k$ 是关键词属性,则统计 $R_L$ 中的记录所有出现的关键词及它们各自的出现频率;
- 如果 $a_k$ 是范围属性,则为其建立一个数轴,将 $R_L$ 中的记录在该属性上的值映射到这个数轴上;
- 如果 $a_k$ 是分类属性,则统计 $R_L$ 中的记录在该属性上各个分类取值的频率.

基于这些统计信息,由当前选定记录 $R_c$ 生成查询 $Q_c$ 的规则是:

- 对于 $Q_c$ 中的关键词属性,我们首先统计:(1) $R_c$ 在该属性上的关键词;(2) 在 $WG_L$ 中与 $R_c$ 相邻记录在该

属性上的关键词及频率.如果 $R_c$ 中关键词不出现在它的相邻记录中,那么从这些关键词中选择在 $R_L$ 中出现频率最低的那一个;否则,选择在它的相邻记录中出现频率最低的那一个.

- 对于 $Q_c$ 中的范围属性,预定义一个区间长度$\delta$,选择这样一个取值范围:长度为$\delta$,且使得在这个范围中,在 $WG_L$ 中与 $R_c$ 相邻记录中出现得最少.
- 对于 $Q_c$ 中的分类属性,选择在 $WG_L$ 中与 $R_c$ 相邻记录在属性上出现频率最低的那个取值.

### 3.4 采样过程的终止(StopCriteria)

如果没有终止条件,采样过程可以一直进行下去,在理论上可以得到 *WDB* 中所有的记录,但这并非我们的目的.我们的目的是得到足够可以作为样本的记录.直觉上,如果连续多次使得每次查询结果中总有一定比例的记录与 $R_L$ 中的记录重复,那么我们可以认为已经游历到了 *WG* 中的大部分空间中.因此,我们设定两个常量 $n_q$ 和$\sigma$,$n_q$ 是一个大于 1 的自然数,$\sigma$是一个 0~1 的百分比,其意义表示,如果连续 $n_q$ 次每次查询结果中总有超过$\sigma$比例的重复记录,那么采样过程就会终止.通常,$n_q$ 设为 5~10 之间,$\sigma$设为 5%~15%.

### 3.5 样本偏差的修正

事实上,如果我们把当前获得的 $R_L$ 作为样本,一般会有较大的偏差(公式(1)).因此,我们需要采取措施来修正偏差.根据我们的观察,通常 *WDB* 在结果页面中会给出一个统计数字来表示满足当前查询的记录数量.因此,我们保存采样过程中的所有 $Q\{Q_1,Q_2,\ldots,Q_m\}$,同时为每个查询时记录查询结果的数量.

$$Deviation(Q_i)=\frac{|R_L \| R(Q_i)|}{|R_L(Q_i)|}-\frac{\sum_{j=1}^{m}\frac{|R_L \| R(Q_j)|}{|R_L(Q_j)|}}{m} \tag{2}$$

我们进行样本偏差修正的基本思想是:通过对 $R_L$ 逐步地删减,使得 $Q$ 如果作为随机查询集合,则利用公式(1)得到的偏差尽可能地小.修正的过程描述如下:

第 1 步,我们利用公式(2)对每一个查询 $Q_i$ 作偏差估计,得到偏差的最大的查询 $Q_{\max}$;

第 2 步,如果 $Deviation(Q_{\max})$小于设定的值$\varepsilon$,终止,当前的为可用样本;否则,用 $Q_{\max}$ 查询 $R_L$,得到所有满足的记录;

第 3 步,对这些记录统计它们各自在 $Q$ 中可满足查询的数量(不包括 $Q_d$),设 $R_{\min}$ 为可满足查询的数量最少的记录;

第 4 步,从 $R_L$ 中将 $R_{\min}$ 移除,同时从 $WG_L$ 中将相应顶点移除,转第 1 步.

从上面的描述可知,该过程是离线完成的(即不需要与 *WDB* 有任何的交互),可以从 WDB-Sampler 算法中分离出来,选择一个合适的时机完成.因此,即使因 $R_L$ 数量大,造成修正代价较高,也不会对采样的代价造成影响.

## 4 实 验

为了客观评估 Web 数据库采样方法,我们根据 WDB-Sampler 算法实现了一个原型,并在本地模拟的 Web 数据库以及真实的 Web 数据库上进行了验证.下面首先介绍实验中使用的数据集,然后给出实验结果及相应的分析.

### 4.1 数据集

在本实验部分,我们使用两种数据集:一个本地模拟 Web 数据库和两个真实 Web 数据库.下面对它们分别进行简要介绍.

#### 4.1.1 本地模拟Web数据库

工作通(JobTong):招聘信息集成数据库,它集成了从当前国内流行的招聘网站(智联招聘、中华人才网、前程无忧网、易才等)爬取下来的招聘信息.尽管我们把它看作模拟实验数据,其中的数据全部是从现实的 *WDB* 获取的真实信息.我们使用了它在 2007 年 7 月 15 日之前的备份数据库,共存储了 982 951 条记录.我们手工为其设计了一个虚拟的查询接口:职位名称(关键词属性)、公司名称(关键词属性)、薪水要求(范围属性)、学历要

求(分类属性).

4.1.2 真实Web数据库

当当网(http://www.dangdang.com/):大型电子商务网站,尽管它提供各种商品的在线销售,我们只关注其中主要的图书记录.它不但提供了图书的高级查询接口(http://search.dangdang.com/book_search.html),可以在书名(关键词属性)、译作者(关键词属性)、出版社(关键词属性)、价格(范围属性)、出版时间(范围属性)上进行查询,而且还提供了图书的分类链接(http://www.dangdang.com/zhuanti2006/book/2001.shtml).利用图书的分类链接,我们可以获得每个分类的数量,从估算出到全部图书的近似总数为 62 万.由于许多图书同时属于多个分类,因此,这一数字并不十分准确,仅作为参考.

中国图书网(http://www.bookschina.com/):大型电子商务网站,只出售图书.它提供了高级查询接口(http://www.bookschina.com/book_find/advancedFind.asp),可以在书名、作者、出版社、出版时间上进行查询.在其主页中宣称共有图书 58 万种,我们用它作为该网站图书记录总数.

### 4.2 随机查询生成

前面我们已经给出评估样本质量的标准(公式(1)),但需要一组查询分别在样本记录集合 $S$ 和 Web 数据库上执行.查询的随机性对样本偏差的计算有直接的影响,而且查询的数量也要足够大,这样才能保证样本偏差与实际情况相符.基于这种考虑,我们提出一种简单的查询自动生成方法来代替人工生成查询.这种方法利用已经存储在本地的样本记录来生成:首先从样本记录中随机选取一个,然后再从该记录中随机选取一个属性值作为查询.从生成方法可以看出,这与前面所提出的方法存在根本的不同:前者要求返回结果数量尽可能地多,这样可以避免偶然性的发生.

另外,为了进一步保证偏差评估的客观性,在随即查询生成的过程中,我们丢弃了重复的查询和在采样过程中用过的查询.

### 4.3 实验结果及分析

下面分别给出在本地模拟 Web 数据库和真实 Web 数据库上的实验结果,并同时对结果进行分析讨论.

4.3.1 本地模拟Web数据库

本节所做实验在本地工作通数据库进行,相应的参数设置为 $n_q$=7,$\sigma$=15%.为了模拟的真实性,我们在采样过程中每次获取查询结果的 top-$k$ 个记录,同时为了验证 $k$(一个 Web 数据库每页最多可显示的记录数量)与样本质量的关系,分别赋予 $k$ 不同的值在此前提下进行采样得到下面的统计数据.

(1) 样本质量分析

我们分别随机生成 100,200 和 500 个查询对表 5 中展示的 4 次采样(最终样本)作偏差分析,得到图 5 所示的结果.

**Table 5** Statistics on local simulating Web databases

**表 5** 在本地模拟 Web 数据库上的统计数据

| Total record number | $k$ | Query number | Obtained record number | Distinct record number | Sampling ratio (%) | Sample number |
|---|---|---|---|---|---|---|
| 982 951 | 10 | 4 086 | 40 860 | 38 275 | 4.16 | 36 204 |
| | 15 | 2 912 | 43 680 | 39 942 | 4.44 | 36 811 |
| | 20 | 2 273 | 45 460 | 41 059 | 4.62 | 37 339 |
| | 30 | 1 642 | 49 260 | 41 731 | 5.01 | 36 974 |

从图 5 中我们可以得到 3 个结论:第一,样本偏差总体来看较小,平均在 10%左右;第二,在其他条件不变的情况下,样本偏差是与 $k$ 成反比的;第三,用于检测样本偏差的随机查询的数量较少时(100)可能会使测量的偏差与实际情况背离较大,数量较大时(200 和 500)可以认为测量值与实际值相符.

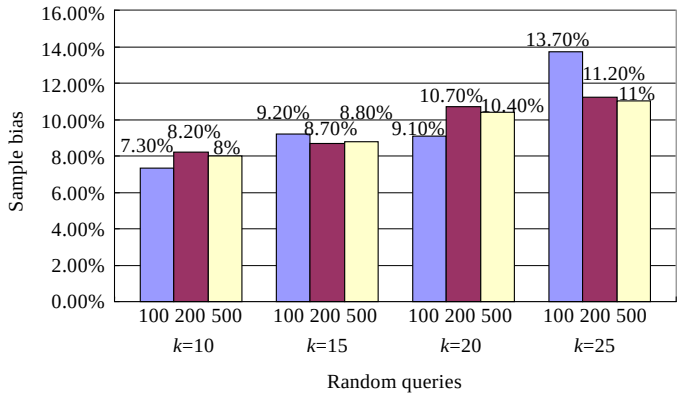



Fig.5 Local WDBs sample bias related simulation experiments

图 5 本地模拟 WDB 样本偏差相关实验

(2) 采样代价分析

从图 6 可以得出两个结论:一是随着 Web 数据库每页显示的结果数量的增加,采样代价会逐渐减小;二是在 $k$ 值大于 30 以后,采样代价的下降趋于平缓.这首先反映出样本偏差和查询代价是成反比的;另外,我们并不能通过增长 $k$ 来无限制地降低采样代价.在现实中,一个 Web 数据库的 $k$ 的大小如果是可改变的,我们就可以根据实际需要在样本偏差和查询代价之间作出权衡.

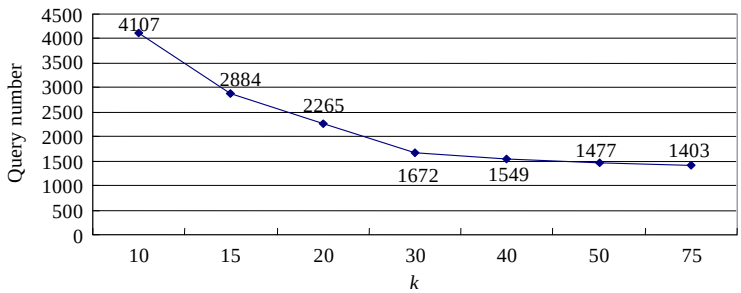



Fig.6 Relationship between $k$ and query cost

图 6 $k$ 与查询代价的关系

4.3.2 真实Web数据库

本节所做实验在两个真实的 Web 数据库进行,分别是当当网和中国图书网,因为对真实的 Web 数据库进行采样需要用 Wrapper 程序从结果页面中抽取记录,复杂性要远高于本地数据库上的实验,因此,我们把相应的参数设置为 $n_q$=5,$\sigma$=5%,略小于前面的设置.其 $k$ 值的大小分别为 20 和 10,所获得的实验数据见表 6.

**Table 6** Statistics on real Web databases

**表 6** 在真实 Web 数据库上的统计数据

| WDB | Record number | $k$ | Query number | Obtained record number | Distinct record number | Sampling ratio (%) | Sample number |
|---|---|---|---|---|---|---|---|
| Dangdang | 620 000 | 10 | 1 327 | 13 270 | 12 446 | 2.01 | 11 962 |
| BooksChina | 580 000 | 20 | 852 | 17 040 | 15 184 | 2.94 | 14 325 |

(1) 样本质量分析

我们分别随机生成 100 和 200 个查询对表 6 中展示的对两个 Web 数据库的采样(最终样本)作偏差分析,得到图 7 所示的结果.

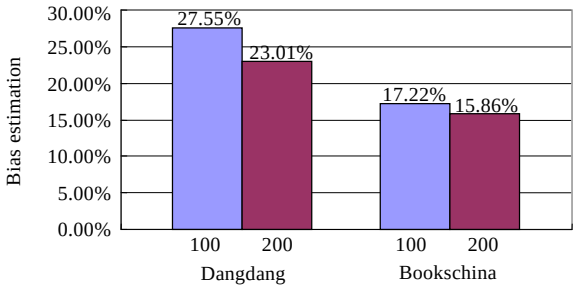



Fig.7 Real WDBs sample bias related simulation experiments

图 7 真实 WDB 样本偏差相关实验

从图 7 中我们可以得到 3 个结论:第一,样本偏差总体平均在 20%左右,整体要高于本地的实验结果;第二,随着随即查询数量的增多,样本偏差同样会逐渐减小;第三,中国图书网的样本偏差要明显小于当当网的样本偏差.产生第 1 个结论的原因,一个是我们把 $n_q$ 和$\sigma$设得较小,另一个是由于我们无法准确知道这两个 Web 数据库的大小.第 2 个结论十分直接,这里不再作解释.第 3 个结论是由于当当网的记录总量是我们根据每个分类累加所得,而事实上分类之间存在一定的重复,因而会大于实际值;而从中国图书网可以看出,其提供的数字虽然不是确切数字,但与事实数字应该十分接近.这也说明了我们的方法需要 Web 数据库大小的一个准确数字,而对 Web 数据库大小的估计也是我们目前开展的工作之一.

(2) 采样代价分析

与在模拟 Web 数据库上的实验结论类似,采样代价仍然与 $k$ 值成反比,这里不再赘述.

## 5 相关工作

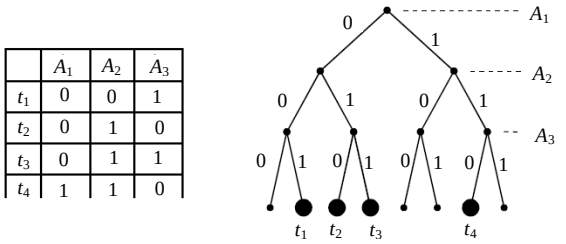
随机采样技术在数据库领域已经得到了大量深入的研究[10−13],传统的数据库采样技术被用来降低从数据库获取数据的代价,其应用包括直方图的估计方法和近似处理等.然而,对 Web 数据库采样的研究至今还未得到太多的关注.随着 Web 的飞速发展以及 Web 数据库数量的急剧增长,Web 数据库采样无论在理论上还是在应用中都成为 Web 数据库集成领域中迫切需要解决的问题.

真正针对 Web 数据库采样提出的工作有 HIDDEN-DB-SAMPLER[14].它首先把 Web 数据库简单化为一个布尔数据库,即每个属性上只能取为 1 或为 0 的布尔值,如图 8(a)所示,进而把这个布尔数据库构建为一棵树模型,如图 8(b)所示.树的每一层对应着一个属性,某一层上非叶节点的分支对应着该属性的一个取值,叶节点对应着 Web 数据库中的一个记录.这样,对 Web 数据库的一个查询可以表示为从根节点开始的一条路径,比如,通过(0,0,1)就可以到达 $t_1$ 所在的节点.其采样的基本思想是:从根节点开始,由两个分支中随机选取一个向下走,直至到达一个叶节点,如果该叶节点上存在一个记录,就把该记录作为样本保存.随后,又把简单的布尔类型扩展到一般的数字和分类属性上.但它的局限性也是明显的,在本文开始部分我们已经讨论了它的局限性,这里不再赘述.

我们需要关注的其他工作是 Web 中对搜索引擎或文档数据库(text database)的采样.对搜索引擎采样的研究已经有许多工作,这里我们对近年来具有代表性的一些工作进行介绍.文献[16]提出了利用搜索引擎返回的 top-$k$ 个结果在文档集合中随机漫步(random walk)的思想,为每个样本文档附加权值来修正其偏差,从而得到近似均匀分布的样本集合.随机漫步的思想类似于本文所提出的图游历的思想,但两者的主要区别在于:搜索引擎中的文档是由一组关键词构成;而 Web 数据库中的记录则是结构化的,存在非文本类型的属性.搜索引擎的采样方法也是通过向公共接口提交关键词查询来获得样本文档,查询生成的方法或者是手工产生[17],或者是利用用户的查询日志产生[18].而我们的方法则是从一个随机的查询开始,利用返回结果中的记录来产生下一次的查询.

目前,已有一些工作通过把一个数据库转化成图的形式去解决特定的问题.文献[15]为实现对 Web 数据库的爬取,把数据库中的每个属性值看作图中的一个顶点,根据如下规则建立顶点之间的关联:(1) 属于同一个记

录;(2) 属于同一属性且值相同.该方法可以有效地实现从 Web 数据库中获取大量的记录,虽然也可以作为一种增量式的采样方法,但它的目标是为了找到更多的新记录,难以保证样本的质量.文献[19]为实现在关系数据库上基于关键词的搜索,把数据库中的每个记录看作图中的一个顶点,按照记录之间外键关系建立顶点之间的关联.虽然该方法提出的图模型也是记录层次的,但由于 Web 数据库是自治的,因此我们无法获得 Web 数据库的模式信息,也无法根据主外键为 Web 数据库建立图模型.




| | $A_1$ | $A_2$ | $A_3$ |
|---|---|---|---|
| $t_1$ | 0 | 0 | 1 |
| $t_2$ | 0 | 1 | 0 |
| $t_3$ | 0 | 1 | 1 |
| $t_4$ | 1 | 1 | 0 |

(a) (b)

Fig.8 Illustration of HIDDEN-DB-SAMPLER

图 8 HIDDEN-DB-SAMPLER 示例

## 6 结论和未来工作

随着 Deep Web 的迅速发展,Deep Web 数据集成逐渐成为数据集成领域的一个热点研究问题.由于 Web 数据库的数量巨大而且只能通过具有特定查询能力的查询接口进行访问,因此需要通过对 Web 数据库的采样了解它们的内容特征.本文提出了一种增量式的 Web 数据库采样方法 WDB-Sampler,通过把一个 Web 数据库转化成图来表示,达到对其进行增量采样的目的.该方法不受查询接口中属性表达形式的限制,本地模拟实验和真实 Web 数据库上的实验表明,该方法可以在较小代价下获取高质量的样本.

不可否认,该工作有一些地方在未来仍然需要进一步的完善和探讨:第一,我们在采样过程中设置了若干参数,这些参数的取值是我们在实现过程中根据经验得到的,还需要在理论上进行分析;第二,对采样代价的评估目前只是通过对 Web 数据库的访问次数来衡量,还需要进一步给出更合理的评估方法;第三,我们下一步将在更多的 Web 数据库上开展实验,发现和改进需要完善之处,进一步降低样本的偏差.

**References**:

[1] Chang KCC, He B, Li CK, Patel M, Zhang Z. Structured databases on the Web: Observations and implications. SIGMOD Record, 2004,33(3):61−70.

[2] BrightPlanet.com. The deep Web: Surfacing hidden value. 2000. http://brightplanet.com

[3] He H, Meng WY, Yu C, Wu ZH. WISE-Integrator: An automatic integrator of Web search interfaces for e-commerce. In: Proc. of the 29th Int'l Conf. on Very Large Data Bases. San Fransisco: Morgan Kaufmann Publishers, 2003. 357−368.

[4] Wu WS, Yu C, Doan AH, Meng WY. An interactive clustering-based approach to integrating source query interfaces on the deep Web. In: Proc. of the 24th ACM SIGMOD Int'l Conf. on Management of Data. Paris: ACM Press, 2004. 95−106.

[5] Peng Q, Meng WY, He H, Yu C. WISE-Cluster: Clustering e-commerce search engines automatically. In: Proc. of the 6th ACM Int'l Workshop on Web Information and Data Management. Washington: ACM Press, 2004. 104−111.

[6] He B, Tao T, Chang KCC. Clustering structured Web sources: A schema-based, model-differentiation approach. In: Proc. of the 9th Int'l Conf. on Extending Database Technology. Heraklion: Springer-Verlag, 2004. 536−546.

[7] Zhao HK, Meng WY, Wu ZH, Raghavan V, Yu C. Fully automatic wrapper generation for search engines. In: Proc. of the 14th Int'l World Wide Web Conf. Chiba: ACM Press, 2005. 66−75.

[8] Zhai YH, Liu B. Web data extraction based on partial tree alignment. In: Proc. of the 14th Int'l World Wide Web Conf. Chiba: ACM Press, 2005. 76−85.

[9] Chang KCC, He B, Zhang Z. Toward large scale integration: Building a MetaQuerier over databases on the Web. In: Proc. of the 2nd Int'l Conf. on Innovative Data Systems Research. Asilomar, 2005. 44−55.

[10] Chaudhuri S, Das G, Srivastava U. Effective use of block-level sampling in statistics estimation. In: Proc. of the 24th ACM SIGMOD Int'l Conf. on Management of Data. Paris: ACM Press, 2004. 287−298.

[11] Haas PJ, Koenig CA. Bi-Level bernoulli scheme for database sampling. In: Proc. of the 24th ACM SIGMOD Int'l Conf. on Management of Data. Paris: ACM Press, 2004. 275−286.

[12] Olken F. Random sampling from databases [Ph.D. Thesis]. Berkeley: University of California, 1993.

[13] Piatetsky-Shapiro G, Connell C. Accurate estimation of the number of tuples satisfying a condition. In: Proc. of the 4th ACM SIGMOD Int'l Conf. on Management of Data. Boston: ACM Press, 1984. 256−276.

[14] Dasgupta A, Das G, Mannila H. A random walk approach to sampling hidden databases. In: Proc. of the 27th ACM SIGMOD Int'l Conf. on Management of Data. Beijing: ACM Press, 2007. 629−640.

[15] Wu P, Wen JR, Liu H, Ma WY. Query selection techniques for efficient crawling of structured Web sources. In: Proc. of the 22nd Int'l Conf. on Data Engineering. Atlanta, 2006. 47−56.

[16] Ziv B, Gurevich M. Random sampling from a search engine's index. In: Proc. of the 15th Int'l Conf. on World Wide Web. ACM Press, 2006. 367−376.

[17] Bradlow E, Schmittlein D. The little engines that could: Modeling the performance of World Wide Web search engines. Marketing Science, 2000,19(1):43−62.

[18] Lawrence S, Giles C. Searching the World Wide Web. Science, 1998,5360(280):98.

[19] Bhalotia G, Hulgeri A, Nakhe C, Chakrabarti S, Sudarshan S. Keyword searching and browsing in databases using BANKS. In: Proc. of the 18th Int'l Conf. on Data Engineering. San Jose: IEEE Computer Society, 2002. 431−440.

**刘伟**(1976－),男,山东聊城人,博士生,主要研究领域为 Deep Web 数据集成,Web 数据抽取.

**孟小峰**(1964－),男,博士,教授,博士生导师,CCF 高级会员,主要研究领域为 Web 数据管理,XML 数据管理,移动数据管理.

**凌妍妍**(1985－),女,硕士生,主要研究领域为 Deep Web 数据集成.