软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software,2012,23(3):539−549 [doi: 10.3724/SP.J.1001.2012.04042]

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

一种非清洁数据库的数据模型*

王宏志+, 李建中, 高 宏

(哈尔滨工业大学 计算机科学与技术系,黑龙江 哈尔滨 150001)

Data Model for Dirty Databases

WANG Hong-Zhi+, LI Jian-Zhong, GAO Hong

(Department of Computer Science and Technology, Harbin Institute of Technology, Harbin 150001, China)

+ Corresponding author: E-mail: wangzh@hit.edu.cn

Wang HZ, Li JZ, Gao H. Data model for dirty databases. *Journal of Software*, 2012,23(3):539−549. http://www.jos.org.cn/1000-9825/4042.htm

Abstract: Dirty data brings new challenges for data management. Current methods of dirty data management are mainly data cleaning. Such methods have limitations when dealing with in applications. In some systems, dirty data has to be tolerated. Therefore, the management of databases with dirty data becomes an important issue. The crucial problem is to obtain query result with a clean degree satisfying clean requirement of applications from databases with dirty data. From the aspect of dirty data management, a data model for dirty databases is presented in this paper. This paper proposes the representation of dirty data, data operators for dirty data and the computation method of clean degree of tuples with support of data operation. The equivalent transformation rules for query expressions on dirty data and the preliminary implementation of the data model are also discussed in this paper.
Key words: data quality; dirty data; data model; query processing

摘 要: 非清洁数据为数据管理带来了新的挑战,当前,处理非清洁的数据清洗方法在实际应用中存在一定的局限性,因此需要在一定程度上容忍非清洁数据的存在.这样,研究管理包含非清洁数据的数据库管理技术就成为了重要的问题,其核心在于如何从包含非清洁数据的数据库中得到满足应用所要求的清洁度的查询结果.从非清洁数据处理角度出发,提出了一种非清洁数据库的数据模型.该模型提出了非清洁数据的表示方法,支持非清洁数据的数据操作,并且支持数据操作清洁度的计算.同时还讨论了查询表达式的等价转换规则和模型的初步实现.
关键词: 数据质量;非清洁数据;数据模型;查询处理
中图法分类号: TP301 **文献标识码**: A

在信息社会中,数据质量决定了数据的价值.然而,实际应用中的数据可能存在严重的质量问题,这些质量问题对应用产生了巨大的影响.由 SAS[1]和 Merrill Lynch[2]资助的报告指出,每年美国企业因为数据质量问题损失超过 6 000 亿美元,对于大多数企业来说,获取和清洗数据占用了信息集成投资的 50%~80%.中国在信息化过

* 基金项目: 国家自然科学基金(61003046, 60933001); 国家重点基础研究发展计划(973)(2012CB316200); 中国博士后科学基金(201003447); 教育部博士点基金(20102302120054); 哈尔滨工业大学优秀青年教师培养计划(HITQNJS.2009.052); 数据工程与知识工程教育部重点实验室(中国人民大学)开放课题(KF2011003)

收稿时间: 2010-05-21; 定稿时间: 2011-04-28

程中也存在同样的问题.数据质量问题是由非清洁数据造成的,非清洁数据是指具有不一致、不精确、错误、冗余、过时等问题的数据.非清洁数据有多种来源:一是数据本身来源的不清洁会导致非清洁数据,例如数据采集和录入的不精确;另一方面,数据模式的不清洁和信息集成中模式不匹配也可能造成数据的非清洁.此外,数据上查询本身的非清洁也可能造成非清洁的查询结果,例如表 1 中,模式为(PID,title,publication)的表 pub 上查询题目为“On View and XML”的文献,用户将查询误写成“select title, publcation from pub where title= “On View and XML”(Q_1),如果假定查询是清洁的,该查询得不到任何结果,而且会产生错误;但是如果定义了模式的相似度,则可查询相似属性 publication 的结果.

Table 1 Pub table

表 1 Pub 表

PID	Title	Publication	ε
DBLP: conf/pods/99	On View and XML	PODS	0.9
DBLP: conf/pods/99	On View and XML	PODS	0.9
DBLP: journals/sigmod/DongS00	Incremental maintenance of recursive views using relational calculus/SQL	SIGMOD record	0.9
DBLP: conf/pods/BenediktGLS00	Constraint databases: A tutorial introduction	PODS	0.9
DBLP: journals/sigmod/Halevy00	Theory of answering queries using views	SIGMOD record	0.9

虽然,非清洁数据的处理已经引起了学术界和工业界的关注,但是当前的方法却主要集中在数据清洗上.实际应用中,数据清洗具有局限性:第一,很多非清洁数据是难以被彻底清洗干净的;第二,对非清洁数据的清洗可能会造成信息的损失;第三,对于信息更新频繁的系统,频繁地执行非清洁数据的辨识和清洗将极大地减低系统的效率.

从上述分析可以看出,非清洁数据的辨识与清洗很难保证数据库不具有非清洁数据,且会降低信息管理系统的性能,不能快速有效地解决非清洁数据带来的问题.在很多情况下,需要在一定程度上容忍非清洁数据的存在.这样,研究管理包含非清洁数据的数据库管理技术成为了重要的问题,其核心在于如何从包含非清洁数据的数据库中得到满足应用所要求清洁度的查询结果.本文称包含非清洁数据的数据库为非清洁数据库.

当前,已经有部分工作开始了对非清洁数据库上查询处理技术的研究[3−6].文献[3,4]的基本思想是,对非清洁数据上的查询进行改写,从而使得到的查询能够从不一致的数据中检索出满足一致性的结果.文献[5]提出了利用概率方法对存在重复数据的非清洁数据库的查询结果进行清洗的策略.这种方法通过改写查询,使得每条查询结果中包含该结果清洁的概率.这种方法的局限性在于仅考虑重复数据这种非清洁因素.文献[6]研究了在不完备数据上 Skyline 查询的处理算法.这些工作仅针对某种特定的非清洁因素或者查询类型.非清洁数据的有效管理需要一种统一的模型对非清洁数据库及其操作加以描述.

在数据结构和操作方面,非清洁数据库的数据模型与传统的关系模型有两点不同:一是传统关系模型假设数据是清洁的,缺少描述数据清洁度的机制;另一方面,传统关系模型假设数据操作作用在清洁数据上,并且得到清洁的结果没有考虑数据操作对结果清洁度的影响,而且传统关系模型上缺少与结果清洁度相关的操作.

当前的概率数据模型[7]可以用来描述非精确数据,但这仅仅是非清洁数据的一种.从数据操作的角度来看,概率模型不能有效地描述数据操作对操作结果清洁度的影响,且缺少非清洁结果的过滤操作,不能够根据用户对查询结果的清洁度需求获取相应的结果.而且,概率数据模型不能描述模式非清洁的数据库,尽管有一些工作考虑到信息集成中引入的模式不确定性[8,9],但这些工作中仅考虑模式匹配过程中引入不确定性,而没有考虑到查询与数据模式不匹配等多种情况引入的非清洁因素.

针对当前非清洁数据库模型研究的不足,本文提出了非清洁数据库的数据模型,其特点是:(1) 以关系模型为基础,引入描述数据清洁度的机制;(2) 在非清洁数据上重新定义了传统关系操作,此定义不仅描述了操作的功能,也描述了操作结果的清洁度和操作数据清洁度之间的关系;(3) 提出支持非清洁数据上有结果清洁度要求查询的新操作;(4) 本模型既可以用来描述由于数据输入造成的数据非清洁性,又可以用来描述信息集成过程中由模式匹配引入的模式非清洁性.

第 1 节描述本文所用的数据用例.第 2 节提出本文提出模型的数据结构.第 3 节提出模型中的代数操作.第

4 节将讨论数据模型的实现.

1 数据用例

本文以一个描述论文和作者的关系数据库为例来说明数据模型和操作,该关系包含 3 张表,其模式分别为 Pub(PID(CHAR),title(CHAR)),Author(AID(INT),name(CHAR),organization(CHAR))和 Pub_Author(PID(CHAR), AID(INT)).假定在 PID,title,AID,name 和 organization 列上存在非清洁数据,其非清洁的因素可包括错误、不完整、不精确等.这 3 个表中的数据分别见表 1~表 3,其中每个表的最后一列表示清洁度,含义在第 2 节详细论述.

Table 2 Author table

表 2 Author 表

AID	Name	Organization	ε
1	Serge Abiteboul	INRIA	0.9
2	Guozhu Dong	Wright State University	0.9
3	Juanwen Su	U C Santa Barbara	0.9
4	Jan Van den Bussche	Universiteit Hasselt	0.9
5	Alon Halevy	University of Washington	0.9
6	Alon Havy	University Hashingt	0.8

Table 3 Pub_Author table

表 3 Pub_Author 表

PID	AID	ε
DBLP: conf/pods/99	1	0.9
DBLP: journals/sigmod/DongS0	2	0.9
DBLP: journals/sigmod/DongS00	3	0.9
DBLP: conf/pods/BenediktGLS00	4	0.9
DBLP: journals/sigmod/Halevy00	5	0.9

2 数据结构

非清洁数据中数据的非清洁因素可以归为两类,即数据本身非清洁性和数据间的非清洁性.前者包括数据的错误、不准确、不完全等,后者包括数据的重复、不一致等.本文提出模型的数据结构通过在元组上添加非清洁度来描述数据本身的非清洁度,在元组集合上定义元组相似性来描述元组间的非清洁度.数据库中元组属性值的取值范围记为 dom,dom∪dom×dom∪dom×dom×dom∪….记为 dom^*.

定义 2.1(元组). 元组是二元组 $t=(V_t,\varepsilon_t)$,其中,$V_t\in\mathrm{dom}^*$是元组的值,$\varepsilon_t\in[0,1]$是元组 t 的清洁度.

定义 2.2(元组集合). 元组集合是元组的集合.

定义 2.3(关系). 关系是二元组 $R=(T_R,S_R)$,其中 T_R 是关系中元组的集合,S_R 是关系的模式,$\forall t\in T_R$ 满足模式 S_R.

例如,表 1 中的每一行可以看作一个元组,每行中的 0.9 是该行的清洁度;在不考虑模式的情况下,表 1 可以看作是元组集合;这个元组集合和模式,Author(AID(INT),name(CHAR), organization(CHAR))共同构成了一个关系.

关系和元组集合的区别在于,关系存在模式而元组集合可能包括模式不同的元组.通过将元组集合 P 中每条元组匹配到给定的模式 S_R 上得到 T_R,P 可以表示成为关系 $R=(T_R,S_R)$.

定义 2.4(数据库). 数据库是二元组 $D=(R_D,sim)$,其中,相似性函数 $sim:\mathrm{dom}^*\times\mathrm{dom}^*\to[0,1]$,表示两个元组相似性,$M_D=\{S_R|R\in R_D\}$.

同一数据库中,不同类型属性或元组之间相似性的计算可以有不同的方法,该相似性函数可以根据需求定义.研究人员已经从不同角度提出了一些计算元组或者属性之间相似性的方法[10].单个属性相似性的定义包括编辑距离、Jaccard 相似性等,包含多个属性的元组之间相似性的计算法方法有基于二分图匹配的方法[11]等,再经过归一化,这些计算方法均适用于相似性函数 sim.例如在我们的用例中,在数据类型为字符串的属性值 s_1 和 s_2 的相似性函数可定义为

$$1-\frac{edit\_dist(s_1,s_2)}{|s_1|+|s_2|} \tag{1}$$

其中,$edit_dist(s_1,s_2)$是 s_1 和 s_2 之间的编辑距离.两个模式相同元组之间的相似性定义为对应列相似性的平均值.

数据库上的查询根据数据库的模式定义.

3 非清洁数据库上的操作

根据操作的性质,非清洁数据上的操作可以分为以下 4 类:

- 查询操作.包括选择、投影、笛卡尔积;
- 集合操作.包括集合的并、交、差;
- 分析操作.包括分组、聚集;
- 结果提取操作,用于在查询结果中提取需要的结果.

其中,除结果提取操作之外的 3 类和普通关系代数类似,增加的结果提取操作是为了从非清洁数据中提取出满足清洁度要求的结果.前 3 类操作的结果是由定义 2.3 定义的关系.接下来,第 3.1 节~第 3.3 节依次定义这些操作.因查询和具有特定模式的关系相关,本节中操作都定义在关系上,操作的结果也是具有模式的关系.

3.1 查询操作

非清洁数据上的查询操作和清洁数据上的查询操作有两个不同之处:一是清洁数据上的操作假定投影和选择相关的模式与关系的模式一致;而在非清洁数据上的操作,由于考虑到模式的非清洁性,投影和选择的相关模式可以与关系的模式不同.另一个是考虑到数据的非清洁性,查询操作的结果也需要清洁度的描述.

定义 3.1(投影). 关系 R 上的投影操作定义为 $Proj_S(R)=(R',S)$,其中,$R'=\{(match(t,S),\varepsilon_t\times sim(S_R,S))|t\in T_R\}$.

函数 match(t,S)将 t 转化成模式为 S 的数据,如果 S 是 S_R 的子集,则结果就是 t 在 S 上的投影,否则需要模式转换.在信息集成中,研究人员提出的一些关系数据库模式的匹配方法[12]适用于该模式匹配函数的定义.

在本文的例子中,类似字符串相似性的定义,模式中两个属性 a_1 和 a_2 的相似性根据其编辑距离定义为

$$sim(a_1,a_2)=1-\frac{edit\_dist(a_1,a_2)}{|a_1|+|a_2|}.$$

$match(t,S)$定义为元组 t'对于每个属性 $s_a\in S$,其属性 $t'[s_a]=t[\arg\min_{t_a\in S_t}(sim(t_a,s_a))]$ (S_t 是 t 的属性集合).其中,函数 $\arg\min_{t_a\in S_t}(sim(t_a,s_a))$ 表示模式 S_t 中与 s_a 名字距离最小的属性.模式 S_1 和 S_2 的相似性定义为 S_2 中每个属性到 S_1 中最相似属性相似性的平均值,即 $sim(S_1,S_2)=\frac{\sum_{t_2\in S_2}\min_{t_1\in S_1}(t_1,t_2)}{|S_2|}$.查询 Q_1 需要在 pub 表上根据模式 S=(title, publcation)进行投影 $Proj_S$(pub).根据上述定义,pub 模式中的 title 属性对应 S 中的 title 属性,相似性为 1;pub 模式中的 publication 属性对应 S 中的 publcation 属性,相似性为 0.952,则得到的中间结果 M_1 见表 4 所示,其中,每个元组的清洁度均为 0.9×0.952.

Table 4 Intermediate results of projection (M_1)

表 4 投影中间结果 M_1

Title	Publcation	ε
On View and XML	PODS	0.857
On View and XML	PODS	0.857
Incremental maintenance of recursive views using relational calculus/SQL	SIGMOD record	0.857
Constraint databases: a tutorial introduction	PODS	0.857
Theory of answering queries using views	SIGMOD record	0.857

定义 3.2(选择). 关系 R 上的选择操作定义为 $Sel_c(R)=(R',S_R)$,其中,$R'=\{(V_t,\varepsilon_t\times D(V_t,c))|t\in T_R\}$.

其中,函数 $D(V_t,c)$计算 V_t 对约束 c 的满足程度,其计算方法如下:

- 如果 c 形式为 $V_t=v_c$ 的原子约束(V_t 和 v_c 的定义域为 dom_v),$D(V_t,c)=sim(V_t,v_c)$;

- 如果 c 形式为 $V_t \neq v_c$ 的原子约束(V_t 和 v_c 的定义域为 dom_v),$D(V_t,c)=1-sim(V_t,v_c)$;
- 如果 c 形式为 $V_t > v_c$ 的原子约束(V_t 和 v_c 的定义域为 dom_v),$D(V_t,c)=\max\limits_{v\in \mathrm{dom}\wedge v>v_c}\{sim(V_t,v)\}$;
- 如果 c 形式为 $V_t < v_c$ 的原子约束(V_t 和 v_c 的定义域为 dom_v),$D(V_t,c)=\max\limits_{v\in \mathrm{dom}\wedge v<v_c}\{sim(V_t,v)\}$;
- 对于复合约束 $c=c_1 * c_2$, $D(V_t,\ c)=\begin{cases} D(V_t,\ c_1)\times D(V_t,\ c_2), * = \wedge \\ 1-(1-D(V_t,\ c_1))\times(1-D(V_t,\ c_2)), * = \vee \end{cases}$;
- 对于复合约束 $c=\neg c'$,$D(V_t,c)=1-D(V_t,c')$.

例如,查询 Q_1 需要在 title 属性上选择操作 $Sel_{title='\text{On Views and XML}'}(pub)$,该查询的结果见表 5,记为 M_2.其中,第 1 条记录的清洁度根据原清洁度 0.9 和公式(1)计算出其 title 属性的值“On View and XML”与约束“On Views and XML”的相似性是 0.89,则计算出改结果的清洁度是 0.9×0.89=0.801.

Table 5 Intermediate results of selection (M_2)

表 5 选择中间结果 M_2

PID	Title	Publication	ε
DBLP: conf/pods/99	On View and XML	PODS	0.801
DBLP: conf/pods/99	On View and XML	PODS	0.829
DBLP: journals/sigmod/DongS00	Incremental maintenance of recursive views using relational calculus/SQL	SIGMOD record	0.272
DBLP: conf/pods/BenediktGLS00	Constraint databases: A tutorial introduction	PODS	0.323
DBLP: journals/sigmod/Halevy00	Theory of answering queries using views	SIGMOD record	0.358

定义 3.3(笛卡尔积). 关系 R 上的笛卡尔积操作定义为 $Cartesian(R_1,R_2)=(R',(S_{R_1},S_{R_2}))$,其中,

$$R'=\{((V_{t_1},V_{t_2}),\varepsilon_{t_1}\times\varepsilon_{t_2})\mid t_1\in T_{R_1},t_2\in T_{R_2}\}.$$

与关系上的连接操作类似,关系 R 和 S 以 c 为约束的连接操作可以表示为 $Join(R_1,R_2,c)=Sel_c(R',(S_{R_1},S_{R_2}))$,其中,$R'=\{(V,\varepsilon_r\times D(V,c))\mid (V,\varepsilon_r)\in Cartesian(R_1,R_2)\}$.

我们用例子来说明连接操作,查询 select title from Pub,Author,Pub_Author where name=‘Guozhu Dong’and Author.AID=Pub_Author.AID and Pub_Author.PID=PUB.PID (Q_2)需要对 3 个表 Pub,Author,Pub_Author 进行连接.设先进行表 Pub 在模式(title,PID)上的投影,然后,表 Pub 和 Pub_Author 以 Pub_Author.PID = PUB.PID 为条件进行连接 Join(Pub,Pub_Author,“1.PID=2.PID”),结果见表 6.其中,为了便于说明结果来源,用 $\mathrm{PID}_{\mathrm{pub}}$ 和 $\mathrm{PID}_{\mathrm{author}}$ 两列分别表示表 Pub 和 Author 对应的记录编号.

Table 6 Results of join (M_3)

表 6 连接结果 M_3

Title	$\mathrm{PID}_{\mathrm{pub}}$	$\mathrm{PID}_{\mathrm{author}}$	ε
On View and XML	DBLP: conf/pods/99	DBLP: conf/pods/99	0.810000
On View and XML	DBLP: conf/pods/99	DBLP: journals/sigmod/DongS0	0.497045
On View and XML	DBLP: conf/pods/99	DBLP: journals/sigmod/DongS00	0.486000
On View and XML	DBLP: conf/pods/99	DBLP: conf/pods/BenediktGLS00	0.576000
On View and XML	DBLP: conf/pods/99	DBLP: journals/sigmod/Halevy00	0.475435
On View and XML	DBLP: conf/pods/99	DBLP: conf/pods/99	0.810000
On View and XML	DBLP: conf/pods/99	DBLP: journals/sigmod/DongS0	0.497045
On View and XML	DBLP: conf/pods/99	DBLP: journals/sigmod/DongS00	0.486000
On View and XML	DBLP: conf/pods/99	DBLP: conf/pods/BenediktGLS00	0.576000
On View and XML	DBLP: conf/pods/99	DBLP: journals/sigmod/Halevy00	0.475435
Incremental maintenance of recursive views using relational calculus/SQL	DBLP: journals/sigmod/DongS00	DBLP: conf/pods/99	0.486000
Incremental maintenance of recursive views using relational calculus/SQL	DBLP: journals/sigmod/DongS00	DBLP: journals/sigmod/DongS0	0.795273
Incremental maintenance of recursive views using relational calculus/SQL	DBLP: journals/sigmod/DongS00	DBLP: journals/sigmod/DongS00	0.810000
Incremental maintenance of recursive views using relational calculus/SQL	DBLP: journals/sigmod/DongS00	DBLP: conf/pods/BenediktGLS00	0.564107
Incremental maintenance of recursive views using relational calculus/SQL	DBLP: journals/sigmod/DongS00	DBLP: journals/sigmod/Halevy00	0.724737

Table 6 Results of join (M_3) (continue)

表 6 连接结果 M_3(续)

Title	PID_{pub}	PID_{author}	ε
Constraint databases: A tutorial introduction	DBLP: conf/pods/BenediktGLS00	DBLP: conf/pods/99	0.576000
Constraint databases: A tutorial introduction	DBLP: conf/pods/BenediktGLS00	DBLP: journals/sigmod/DongS0	0.544909
Constraint databases: A tutorial introduction	DBLP: conf/pods/BenediktGLS00	DBLP: journals/sigmod/DongS00	0.564107
Constraint databases: A tutorial introduction	DBLP: conf/pods/BenediktGLS00	DBLP: conf/pods/BenediktGLS00	0.810000
Constraint databases: A tutorial introduction	DBLP: conf/pods/BenediktGLS00	DBLP: journals/sigmod/Halevy00	0.540000
Theory of answering queries using views	DBLP: journals/sigmod/Halevy00	DBLP: conf/pods/99	0.475435
Theory of answering queries using views	DBLP: journals/sigmod/Halevy00	DBLP: journals/sigmod/DongS0	0.708750
Theory of answering queries using views	DBLP: journals/sigmod/Halevy00	DBLP: journals/sigmod/DongS00	0.724737
Theory of answering queries using views	DBLP: journals/sigmod/Halevy00	DBLP: conf/pods/BenediktGLS00	0.540000
Theory of answering queries using views	DBLP: journals/sigmod/Halevy00	DBLP: journals/sigmod/Halevy00	0.810000

3.2 集合操作

清洁数据上集合操作要求参与操作的集合模式相同.而考虑到模式的非清洁性,在非清洁数据上定义的集合操作不要求参与操作的集合模式相同.但是对模式不同的元组集合进行集合操作时,需要首先进行模式转换,再进行集合操作,作为结果的元组在模式转换的过程中可能损失清洁度.集合操作并、交和差的定义见定义 3.4~定义 3.6.

定义 3.4(并). 关系 R 和 S 的并定义为 $Union(R_1,R_2)=(T_{R_1}\cup proj_{S_{R_1}}(T_{R_2}),S_{R_1})$.

定义 3.5(交). 关系 R 和 S 的交定义为 $Intersection(R_1,R_2)=(T_{R_1}\cap proj_{S_{R_1}}(T_{R_2}),S_{R_1})$.

定义 3.6(差). 关系 R 和 S 的差定义为 $Diff(R_1,R_2)=(T_{R_1}-proj_{S_{R_1}}(T_{R_2}),S_{R_1})$.

3.3 聚集操作

分组是聚集操作的基础,因此我们首先定义分组的方法.与清洁数据不同,非清洁数据上的分组不可以简单地根据属性值的完全相等进行分组,而需要在分组中充分考虑数据的非清洁性.根据分组依据的不同,非清洁数据上的分组有两种定义方法:一种是基于元组之间的相似性,另一种是基于分组元组的值.

定义 3.7((γ,A)分组). 关系 R 的(γ,A)分组 $S_{(\gamma,A)}$是元组集合:

$$G_{(\gamma,A)}=\{t|t\in R\wedge\forall t'\in G_{(\gamma,A)},sim(\pi_A(V_t),V_{t'})\geqslant\gamma\wedge\forall r\in R-G_{(\gamma,A)},sim(V_t,V_{t'})<\gamma\}.$$

分组之间可能会有重叠,例如,对表 Author 中的数据根据 organization 属性以γ=0.8 进行(0.8, {organization})分组,由于 sim(“University of Washington”,“University Hashingt”)>0.8,sim(“Universiteit Hasselt”, “University Hashingt”)>0.8 而 sim(“University of Washington”,“Universiteit Hasselt”)<0.8,因而 University Hashingt 同时被分到有 University of Washington 和 Universiteit Hasselt 的两个分组中去,但 University of Washington 和 Universiteit Hasselt 不能被分到同一组.

定义 3.8(分组中心). 在一个(γ,A)分组 G 中,其中心 C_G 是元组 t,满足

$$\forall r\in G,\sum_{t'\in G}sim(V_t,V_{t'})\geqslant\sum_{t'\in G}sim(V_r,V_{t'}).$$

定义 3.9(γ聚集). 关系 R 以属性集合 A 进行分组,以聚集函数 fun 为聚集函数,在属性 m 上的γ聚集定义为 $agg(R,m,A,fun,\gamma)=(R',\{m\}\cup A)$,其中,$R'=\{((fun(\pi_{\{m\}}(G_{(\gamma,A)})),\pi_A(C_{G_{(\gamma,A)}})),\varepsilon_{G(\gamma,A)})|G_{(\gamma,A)}\subseteq T_R\}$,

$$\varepsilon_{G_{(\gamma,A)}}=\begin{cases}\dfrac{2\cdot\sum_{t_1\in G_{(\gamma,A)},t_2\in G_{(\gamma,A)}}sim(\pi_A(V_{t_1}),\pi_A(V_{t_2}))}{|G_{(\gamma,A)}|(|G_{(\gamma,A)}|-1)}, & |G_{(\gamma,A)}|>1\\ 1.0, & |G_{(\gamma,A)}|=1\end{cases}.$$

例如,对表 Author 中数据的γ聚集 agg(Author,name,{organization},count,0.8)的结果是{(INRIA,1,1.0),(Wright State University,1,1.0),(U C Santa Barbara,1,1.0),{Universiteit Hasselt,2,0.821},{University of Washington,2,0.86}}.

定义 3.10(概率分组). 关系 R 在属性集合 A 上对应值 v 的一个概率分组定义为

$$G_{v,A}=\{(V_r,\varepsilon)|r\in T_R\wedge\varepsilon=\varepsilon_r\times sim(v,\pi_A(V_r))\}.$$

$G_{v,A}$的一个可能分组 $G'\subseteq G_{v,A}$,其可能性 $P_{G'}$定义为

$$\prod_{t\in G'}\varepsilon_t\times sim(v,\pi_A(V_r))\prod_{t\in G_{v,A}-G'}(1-\varepsilon_t\times sim(v,\pi_A(V_r))).$$

例如,对关系 Author 根据 organization 进行概率分组,对应值“University of Washington”的分组 $G_{\text{“University of Washington”},\{organization\}}$中,可能分组{U C Santa Barbara,University of Washington}的可能性是 1×(0.9×*sim*(U C Santa Barbara,University of Washington))×(1−0.9×*sim*(INRIA,University of Washington))×(1−0.9×*sim*(Universiteit Hasselt,University of Washington))×(1−0.8×*sim*(University Hashingt,University of Washington))=0.021.

定义 3.11(概率聚集). 关系 R 的以属性集合 A 进行分组,以聚集函数 *fun* 为聚集函数,在属性 m 上的概率聚集定义为 $agg(R,m,S,fun)=\{R',\{m\}\cup A\}$,其中,$R'=\{(r,\pi_A(G_{v,A}),\varepsilon_r)\},|v\subseteq\pi_A(T_R)\wedge(r,\varepsilon_r)\in Pr_{fun}(Proj_{(m)}(G_{v,A}),v)\}$.

函数 $Pr_{fun}()$是根据数据清洁度进行的聚集操作,其输出是所有可能的聚集值及其清洁度.该清洁度用可能性来表示,定义为

$$Pr_{fun}(G,v)=\left\{\left(r_{G'}\sum_{g\in G'}P_g\right)\middle|G'\subseteq 2^G\wedge\forall G_1,G_2\in G',fun\left(T_{G_1}\right)=fun\left(T_{G_2}\right)\wedge r_{G'}=fun\left(T_{G_1}\right)\right\}.$$

例如,对于概率聚集 *agg*(Author,name,{organization},count),对应值“University of Washington”可能的聚集值分别是 1,2,3,4,5,其中,聚集值为 2 的清洁度对应着所有包含值“University of Washington”且元素个数为 2 的子集合概率的和.

请注意,概率聚集的定义仅对分布聚集函数和代数聚集函数有效,整体聚集函数的定义和计算将是我们进一步的研究工作.

3.4 数据提取操作

与清洁的关系数据库不同,非清洁的关系数据库上的查询结果有清洁度要求,而由操作产生的结果不一定能够满足查询中的清洁度要求.因此,需要有结果提取函数对查询操作生成的结果进行提取,得到满足要求的结果.本节提出用于提取结果的操作,根据不同的查询需求,我们提出了 3 种结果提取操作,分别是提取清洁度大于阈值结果的ε提取、提取清洁度最高的 k 个结果的 Top$K$ 提取和提取出包含信息量最大的 k 重要提取.

定义 3.12(ε提取). 对于关系 R,ε提取定义为 $Extraction_\varepsilon(R)=(R',S_R)$,其中,$R'=\{t|t\in R\wedge\varepsilon_t\geqslant\varepsilon\}$.

例如,对于查询 Q_1 以ε=0.5 提取结果,则在 M_2 中提取元组(On View and XML,PODS,0.801)和(On View and XML,PODS,0.829).

定义 3.13(Top$K$ 提取). 对于关系 R,Top$K$ 提取定义为 $Extraction_{TopK}(R)=(R_E,S_R)$,其中,$R_E$ 满足下列 3 个条件:(1) $R_E\subseteq R$;(2) $|R_E|=k$;(3) $\forall t\in R_E,\forall t'\in R_R-R_E,\varepsilon_t\geqslant\varepsilon_{t'}$.

例如,对于查询 Q_1 以 k=3 提取结果,则在 M_2 中提取元组(On View and XML,PODS,0.829),(On View and XML,PODS,0.801)和(Theory of Answering Queries Using Views,SIGMOD Record,0.358).

ε提取和 Top$K$ 提取可以提取出清洁度较高的查询结果,但是它们的不足之处在于清洁度较高的查询结果可能指代的同一对象,这样的结果对用户来说相当于重复结果.因此,我们提出了 k 重要提取,见定义 3.14,用以提取差别最大的 k 个结果.k 重要提取得到的结果集合,是在查询所有大小为 k 的结果子集中元组之间相似性最小的.在上述例子中,ε提取的结果和 Top$K$ 提取结果中的前两条记录是非常相似的结果.

定义 3.14(k 重要提取). 对于关系 R,k 重要提取定义为 $Extraction_{significant\text{-}k}(G)=(R_E,S_R)$,其中,$R_E$ 满足下列 3 个条件:(1) $R_E\subseteq R$;(2) $|R_E|=k$;(3) $\forall R'_E\subseteq R$ 满足$|R'_E|=k$, $\sum_{\forall t_1,t_2\in R_E}sim(v_{t_1},v_{t_2})\leqslant\sum_{\forall t_1,t_2\in R'_E}sim(v_{t_1},v_{t_2})$.

k 重要提取可以分别与ε提取和 Top$K$ 提取加以复合,分别用以提取清洁度大于ε的 k 重要结果和清洁度最高的 k_1 个结果中最重要的 k_2 个结果.例如,对上述 Top$K$ 结果可以进一步以 k=2 进行 k 重要提取,则结果为(On View and XML,PODS,0.829)和(Theory of Answering Queries Using Views,SIGMOD Record,0.358).

3.5 非清洁数据上查询表达式的等价转换规则

类似清洁数据上的关系操作,非清洁数据上的查询表达式等价转换规则可以为查询优化提供支持.本节定义非清洁数据上查询表达式的等价转换规则.

我们首先定义非清洁数据上代数表达式的等价性.

定义 3.15(等价). 如果两个非清洁数据上的查询代数表达式 E_1 和 E_2 在同一个非清洁数据库 D 上的查询结果相等,即 $E_1(D)=E_2(D)$,则 E_1 和 E_2 等价,记做 $E_1=E_2$.

请注意,这个等价的定义和清洁数据上查询表达式等价定义的不同之处在于,该定义中查询结果相等既包括结果集合的相等,也包括结果集合中对应元组之间清洁度的相等.

根据非清洁数据的性质,我们研究了清洁数据上关系操作的等价性规则在非清洁数据模型上的适用性,并分析了非清洁数据上的新的操作与其他操作的关系,得到如下关于查询操作等价转换规则,其中,q 是一个查询代数表达式:

(1) $Sel_{c_1}(Sel_{c_2}(q)) = Sel_{c_1 \wedge c_2}(q)$;

(2) $Cartesian(q_1,Cartesian(q_2,...,Cartesian(q_{k-1},q_k)...)=$
$Cartesian(Cartesian(q_1,Cartesian(q_2,...,Cartesian(q_{k-2},q_{k-1})...),q_k)$;

(3) $Cartesian(q_1,q_2)= Proj_{S_2,S_1}(Cartesian(q_2,q_1))$,其中,$S_1$ 是 q_1 的模式,S_2 是 q_2 的模式;

(4) $Cartesian(Sel_c(q),q')=Sel_c((Cartesian(q,q'))$;

(5) $Cartesian(Proj_S(q_1),q_2)= Proj_{S,S_{q_2}}(Cartesion(q_1,q_2))$,其中 S_{q_2} 是 q_2 所对应的模式.

以下查询等价规则对于特定的转换函数成立:

(6) $Sel_c(Proj_S(q))= Proj_S(Sel_{Map(S_q,c)}(q))$,其中,$Map(S_q,c)$是把约束 c 映射到 q 所对应的模式上的函数.

关于查询中的提取操作,有下列等价转换规则:

(7) $Extraction_{TopK}(Extraction_\varepsilon(q))=Extraction_\varepsilon(Extraction_{TopK}(q))$;

(8) $Extraction_\varepsilon(Proj_S(q))=Extraction_\varepsilon(Proj_S(Extraction_\varepsilon(q)))$;

(9) $Extraction_\varepsilon(Sel_C(q))=Extraction_\varepsilon(Sel_C(Extraction_\varepsilon(q)))$;

(10) $Extraction_\varepsilon(Cartesian(q_1,q_2))=Extraction_\varepsilon(Cartesian(Extraction_\varepsilon(q_1),Extraction_\varepsilon(q_2)))$;

(11) $Extraction_{TopK}(Proj_S(q))=Proj_S(Extraction_{TopK}(q))$;

(12) $Extraction_{TopK}(Cartesian(q_1,q_2))=Extraction_{TopK}(Cartesian(Extraction_{TopK}(q_1),Extraction_{TopK}(q_2)))$;

(13) $Extraction_{significant\text{-}k}(Sel_C(q))=Sel_C(Extraction_{significant\text{-}k}(q))$.

利用这些等价规则,可以对非清洁数据上的查询操作进行优化.例如,对于查询 Q_2 以ε=0.8 提取结果,对应的操作是 $Extract_{0.8}$(Join(Join(Proj$_{(title,PID)}$Pub,Pub_Author,“1.PID=2.PID”),Proj(AID)Author,“1.PID=2.PID”));如果按照从内到外的顺序执行操作(表 3),第 1 个连接操作(Join(Proj$_{(title,PID)}$Pub,Pub_Author,“1.PID=2.PID”)的结果有 25 条,而第 2 个连接操作的结果有 150 条,最后的 Extract 操作需要在此 150 条元组上进行.根据第 3.1 节中的讨论,Join 操作是 Cartesioin 和 Sel 操作的复合,根据规则(9)和规则(10),该操作可以等价地改写为 $Extract_{0.8}$(Join($Extract_{0.8}$(Join(Proj$_{(title,PID)}$Pub,Pub_Author,“1.PID=2.PID”)),Proj(AID)Author,“1.PID=2.PID”)). 尽管增加了一次在 25 条元组上进行的 Extract 操作,但是第 2 个 Join 操作将在一个有 5 条元组表(即表 M_3 中$\varepsilon\geqslant$0.8 的元组)和 Author 表上进行,结果仅有 30 条元组,最后一个 Extract 操作仅需在此 30 条元组上进行.这样的等价变换一方面减少了 Extract 操作的次数,另一方面减少了第 2 次 Join 的候选元组,加快了连接的速度.

4 模型的实现

本节讨论本文描述模型的初步实现策略,其中,第 4.1 节讨论清洁度的获取方法,第 4.2 节讨论操作的初步实现策略.

4.1 清洁度的获取

在本模型中,元组的清洁度是一个重要参数,该模型的应用需要有效的清洁度计算方法.本节将讨论可能的清洁度定义方法以及获取来源.当前,清洁度定义可以用两种方法描述:一种是概率的方法,即将元组清洁度定义为该元组准确的概率;另一种是相对误差的方法,即定义元组对于真实值的相对误差.尽管基于相对误差描述的清洁度值在[0,1]之间,但其表示的是数据相对于真实值的误差,与概率有本质的不同.由于本文提出模型的清洁度定义在[0,1]区间内,并以乘法作为清洁度改变的计算方法,因此适用于这两种清洁度的定义.

在实际应用中,元组的清洁度可以通过人工或者自动的方法获取,主要来源包括:

- 人工添加,最直接的方法是由用户根据领域知识和数据的来源添加数据的清洁度.这种方法的问题在于,当数据量很大时,需要大量的人力;
- 属性清洁度组合.由于一些属性的清洁度是可以预知的,比如在科学统计数据库中描述某仪器采回数据的关系,模式为(Time,Value),其中:Value 列来源于数据采集设备,该设备的相对误差是知道的;而 Time 列是在采样时由系统添加,可以看成清洁的列.因此,该关系中元组的清洁度定义为 Value 列的清洁度;
- 模式转换.在信息集成中,在数据之间进行模式转换会产生清洁度的损失,在一些信息集成方法中会给出模式转换时数据的损失,经过归一化以后,该损失可以用作表示数据清洁度;
- 信息提取.信息提取的过程中,一些技术会根据原始数据的特点或者机器学习的方法求得某数据属于某元组概率或者准确率,这个值可用作数据清洁度的描述;
- 实体识别.当前,有很多实体识别的方法将数据集合划分成为实体,通过描述同一实体的不同元组之间的不一致性来描述实体的清洁度[5].

以上多种方法在一些应用中是可以结合使用的,根据本模型的特点,在结合使用时,元组的清洁度应为多来源清洁度的乘积.一方面使得元组的清洁度满足取值在[0,1]之间;另一方面,这样的方法体现了多种清洁度的复合.如上文讨论,从两种清洁度定义来看,乘法都可以使其满足多种清洁度来源的复合.例如信息提取系统中,首先进行信息提取,然后进行实体识别,则最终结果的清洁度可以定义为信息提取步骤的清洁度与实体识别步骤清洁度的乘积.

4.2 操作的初步实现策略

通过在每个关系中添加描述清洁度的列,现有的关系数据库系统可以对本模型所描述的数据进行管理.然而,现有关系数据库中的管理机制不足以支持所有的操作.考虑到随着操作的进行数据清洁度是递减的,因此根据查询的需求,在操作执行的过程中某些环节需要过滤掉清洁度不可能满足查询要求的中间结果,这样可以减小中间结果的数据量,从而加速查询的处理.例如,对于如果查询要求结果中每条元组的清洁度在 0.5 以上,则在查询处理的过程中可以过滤掉清洁度小于 0.5 的中间结果,因为由这些中间结果生成最终结果的清洁度一定小于 0.5.

本节给出本文提出模型的初步实现策略,更加高效的方法是我们进一步研究的目标.根据模型中对数据结构的定义,非清洁数据可以存储成普通的关系数据.其中的相似性函数、约束判定函数和模式匹配可以通过自定义函数或者存储过程实现.

一些传统的关系数据库操作算法、现有的概率数据库操作算法、数据库上近似查询的算法可以用来实现部分操作.下面简要对不同操作对应方法的应用加以讨论:

- 投影.如果关系的模式与要求的模式相同投影操作,可以采用传统关系数据库上;否则,可以通过模式匹配的方法[12]实现;
- 选择.根据相似性定义方法的不同,选择操作采取不同的方法实现.文献[13–15]提出的方法适用于基于编辑距离、Jaccard 函数等相似性定义的选择操作;
- 连接.连接操作的算法也随编辑距离定义的方法不同,当前有一些针对编辑距离、Jaccard 距离等相似性定义的近似连接算法[16,17],这些方法适用于针对这类相似性定义的连接操作的实现;

- 集合操作.根据定义可知,非清洁数据上的集合操作是投影操作和传统关系上集合操作的复合;
- 聚集操作.当前有一些非清洁数据上的聚集方法提出,其中,文献[8]中提出的方法适用于概率聚集的实现,但适用于γ聚集的算法未见提出;
- 提取操作.非清洁数据上的提取操作根据提取方法不同而不同,其中,ε提取和 Top$K$ 提取可通过在清洁度属性上过滤后进行选择实现.k 重要提取问题是 NP 难问题,这个问题和最大边子图问题[18]等价,可以通过最大边子图问题的近似算法求解[19,20].

5 结 论

本文提出一种新的非清洁数据模型,该模型能够有效地表达数据的清洁度以及操作对清洁度的影响,该模型包括一个以关系操作为核心的操作代数,可以有效地支持非清洁数据的各种该应用.它引进了结果提取的操作,可以针对不同需求提供满足清洁度要求的结果.与其他模型相比,本文提出的非清洁数据模型是一个具有很强表达能力和完整关系操作的数据模型.

注意到,第 4.3 节介绍的操作虽然能够用来实现部分操作,然而并不完全适用非清洁数据的处理,在大规模非清洁数据的处理中效率也受到影响.因此,设计新的非清洁数据查询操作算法以及非清洁数据库上查询优化算法是我们的下一步工作.

References:

[1] Eckerson W. Data Quality and the Bottom Line: Achieving Business Success through a Commitment to High Quality Data, Vol.1. Seattle: The Data Warehousing Institute, 2002. 1−36.

[2] Shilakes CC, Tylman J. Enterprise information portals. RC#60232206, United States: Merrill Lynch, 1998. 1−64.

[3] Fuxman A, Miller R. First-Order query rewriting for inconsistent databases. In: Eiter T, Libkin L, eds. Proc. of the 10th Int'l Conf. on Database Theory. Edinburgh: Springer-Verlag, 2005. 337−351. [doi: 10.1016/j.jcss.2006.10.013]

[4] Fuxman A, Fazli E, Miller RJ. ConQuer, efficient management of inconsistent databases. In: Özcan F, ed. Proc. of the ACM SIGMOD Int'l Conf. on Management of Data. Baltimore: ACM Press, 2005. 155−166. [doi: 10.1145/1066157.1066176]

[5] Andritsos P, Fuxman A, Miller RJ. Clean answers over dirty databases: A probabilistic approach. In: Liu L, Reuter A, Whang KY, Zhang J, eds. Proc. of the 22nd Int'l Conf. on Data Engineering. Atlanta: IEEE Computer Society, 2006. 30. [doi: 10.1109/ICDE.2006.35]

[6] Khalefa ME, Mokbel MF, Levandoski JJ. Skyline query processing for incomplete data. In: Proc. of the 24th Int'l Conf. on Data Engineering. Cancún: IEEE Computer Society, 2008. 556−565. [doi: 10.1109/ICDE.2008.4497464]

[7] Koch C. On query algebras for probabilistic databases. SIGMOD Record, 2008,37(4):78−85. [doi: 10.1145/1519103.1519116]

[8] Gal A, Martinez MV, Simari GI, Subrahmanian VS. Aggregate query answering under uncertain schema mappings. In: Proc. of the 25th Int'l Conf. on Data Engineering. Shanghai: IEEE Computer Society, 2009. 940−951. [doi: 10.1109/ICDE.2009.55]

[9] Dong XL, Halevy A, Yu C. Data integration with uncertainty. In: Koch C, Gehrke J, Garofalakis MN, Srivastava D, Aberer K, Deshpande A, Florescu D, Chan CC, Ganti V, Kanne C, Klas W, Neuhold EJ, eds. Proc. of the 33rd Int'l Conf. on Very Large Data Bases. Vienna: ACM Press, 2007. 687−698. [doi: 10.1007/s00778-008-0119-9]

[10] Elmagarmid AK, Ipeirotis PG, Verykios VS. Duplicate record detection: A survey. IEEE Trans. on Knowledge and Data Engineering, 2007,19(1):1−16. [doi: 10.1109/TKDE.2007.250581]

[11] Li MH, Wang HZ, Li JZ, Gao H. Duplicate record detection method based on optimal bipartite graph matching. Journal of Computer Research and Development, 2009,46(Suppl.):339−345 (in Chinese with English abstract).

[12] Madhavan J, Bernstein PA, Doan AH, Halevy AL. Corpus-Based schema matching. In: Proc. of the 21st Int'l Conf. on Data Engineering. Tokyo: IEEE Computer Society, 2005. 57−68. [doi: 10.1109/ICDE.2005.39]

[13] Li C, Wang B, Yang XC. Vgram: Improving performance of approximate queries on string collections using variable-length grams. In: Koch C, Gehrke J, Garofalakis MN, Srivastava D, Aberer K, Deshpande A, Florescu D, Chan CC, Ganti V, Kanne C, Klas W, Neuhold EJ, eds. Proc. of the 33rd Int'l Conf. on Very Large Data Bases. Vienna: ACM Press, 2007. 303−314.

[14] Yang XC, Wang B, Li C. Cost-Based variable-length-gram selection for string collections to support approximate queries efficiently. In: Wang JT, ed. Proc. of the ACM SIGMOD Int'l Conf. on Management of Data. Vancouver: ACM Press, 2008. 353−364. [doi: 10.1145/1376616.1376655]

[15] Li C, Lu JH, Lu YM. Efficient merging and filtering algorithms for approximate string searches. In: Proc. of the 24th Int'l Conf. on Data Engineering. Cancún: IEEE Computer Society, 2008. 257−266. [doi: 10.1109/ICDE.2008.4497434]

[16] Lieberman M, Sankaranarayanan J, Samet H. A fast similarity join algorithm using graphics processing units. In: Proc. of the 24th Int'l Conf. on Data Engineering. Cancún: IEEE Computer Society, 2008. 1111−1120. [doi: 10.1109/ICDE.2008.4497520]

[17] Xiao C, Wang W, Lin XM, Yu JX. Efficient similarity joins for near duplicate detection. In: Huai JP, Chen R, Hon HW, Liu YH, Ma WY, Tomkins A, Zhang XD, eds. Proc. of the 17th Int'l Conf. on World Wide Web. Beijing: ACM Press, 2008. 131−140. [doi: 10.1145/1367497.1367516]

[18] Garey M, Johnson D. Computers and Intractability: A Guide to the Theory of NP-Completeness. New York: W.H. Freeman and Company, 1979.

[19] Feige U, Peleg D, Kortsarz G. The dense *k*-subgraph problem. Algorithmica, 2001,29(3):410−421. [doi: 10.1007/s004530010050]

[20] Arora S, Karger D, Karpinski M. Polynomial time approximation schemes for dense instances of NP-hard problems. In: Proc. of the 27th Annual ACM Symp. on Theory of Computing. Las Vegas: ACM Press, 1995. 284−293. [doi: 10.1145/225058.225140]

附中文参考文献:

[11] 李默涵,王宏志,李建中,高宏.一种基于二分图最优匹配的重复记录检测算法.计算机研究与发展,2009,46(增刊):339−345.

王宏志(1978－),男,辽宁丹东人,博士,副教授,CCF 高级会员,主要研究领域为 XML 数据管理,数据质量管理,海量数据管理.

李建中(1950－),男,博士,教授,博士生导师,CCF 高级会员,主要研究领域为数据库系统,无线传感器网络.

高宏(1966－),女,博士,教授,博士生导师,CCF 高级会员,主要研究领域为并行数据库系统,数据仓库,无线传感器网络.