不一致数据上查询结果的一致性估计

刘雪莉 李建中
(哈尔滨工业大学计算机科学与技术学院 哈尔滨 150001)

摘 要 主键约束是描述关系数据一致性的常用方法,基于主键约束的数据一致性修复返回一个极大子集,子集中不同数据的主键不同.对于合取查询 Q,一致性合取查询返回一个答案集合,答案集合是 Q 在数据集合 I 的每一个修复下查询结果的交集.文中将 Q 在 I 中的查询结果满足一致性的个数占总的结果个数的比例定义为查询结果的一致性程度.若 Q 不可一阶表达且不能在多项式时间内得到其一致性解,则当 Q 答案个数超过 30 时,使用抽样的方法给答案集合一致性程度的一个(ε,δ)-估计.由于布尔合取查询的一致性判定问题是 coNP-完全问题,因此在估计过程中,使用攻击图,通过攻击图对布尔查询 q 进行改写近似判断 q 近似一致性回答.实验表明了估计算法和近似判定算法具有较高的效率和准确率.

关键词 主键约束;一致性查询;合取查询;近似一致性
中图法分类号 TP311 **DOI号** 10.11897/SP.J.1016.2015.01727

Consistent Estimation of Query Result in Inconsistent Data

LIU Xue-Li LI Jian-Zhong
(*School of Computer Science and Technology, University of Harbin Institute of Technology, Harbin* 150001)

Abstract Primary key constraint is a natural mean for modeling inconsistency in relation data. A repair of a data set is a maximal subset of the data set without two distinct tuples sharing the same primary key. For conjunctive query Q, the consistent query answering problem returns answers that each tuple in answer satisfies every repair of data set. This paper defines consistent degree as the fraction of consistent query answers in query answers. When the number of the answers is not less than 30, sampling method gives a (ε,δ)-estimation of consistent degree. Because of intractability of consistent deciding problem, this paper defines τ-approximate consistency, and using attack graph rewrites query to approximate deciding the consistency of a tuple in query answers. Finally, experiments verifies the efficientness and effectiveness of the estimation algorithm and approximate deciding algorithm.

Keywords primary key; consistent query; conjunctive query; approximate consistency

1 引 言

近年来,数据质量问题引起了人们的高度重视.数据质量是数据分析结论有效性和准确性的基础,也是最重要的前提和保障.造成数据质量问题的因素有很多,数据的不一致性是其中的一个重要方面.在现实世界中,由于网络的普及,应用可以从多个数据源集成数据,使得不一致数据的传播愈演愈烈.不一致数据给社会经济造成了重大的损失.例如,在美

收稿日期:2014-05-18;最终修改稿收到日期:2014-10-27.本课题得到国家"九七三"重点基础研究发展规划项目基金(2012CB316200)、国家自然科学基金(61003046,6111113089)、国家教育部博士点基金(20102302120054)资助.**刘雪莉**,女,1987年生,博士研究生,主要研究方向为数据质量、海量数据计算.E-mail:shally78952286@163.com.**李建中**,男,1950年生,博士,教授,博士生导师,中国计算机学会(CCF)会员,主要研究领域为海量数据计算、无线传感器网络和CPS等.

国银行业,由于信息的不一致性导致的失察信用卡欺诈在 2006 年大约造成了 48 亿美元的损失;而在数据仓库项目的开发过程中,数据清洗通常需要花费 30%～80%的开发时间和开发预算①.

数据的一致性是指数据集合中不包含语义错误或相互矛盾的数据[1].例如,数据(国码="86",区号="10",城市="上海")含有语义错误,因为 10 是北京区号而非上海区号.目前关于数据不一致性的研究主要从两个方面着手:(1)不一致数据的检测和修复;(2)不一致数据上的一致性查询.不一致数据的检测方面,基本方法是建立一组一致性质量规则,若数据集合中存在着不一致信息,则不一致信息将会违背相应的规则从而被检测出来.数据修复是通过尽可能少的修改数据,使得数据集合满足一致性规则集合.然而,数据修复存在着一些问题:首先,删除不一致数据可能会造成有用信息的缺失;其次,并不能保证修复之后的数据一定是正确的,存在着将正确数据修复为错误数据的可能性.基于此,文献[2]提出了一致性查询问题.一致性查询处理是指回答用户查询时仅使用不含错误的信息.文献[2]中使用数据修复来定义一致性查询结果,对于一个主键约束的数据集,它的一个修复可以通过从数据集中挑选出任意两个主键都不相同的极大子集得到.显然,一个不一致的数据集合可能有多种修复.一致性查询结果指的是查询结果中的每一条记录都出现在对所有的修复进行查询的结果中.然而,数据的所有修复具有指数级可能空间,即使只考虑主键约束,一致性查询也是 coNP-完全问题.更重要的是,不一致的数据中也可能包含着用户需求的信息,只返回一致性的查询结果丢失了查询相关的有价值数据,返回的结果可能并不能满足用户的需求,但是返回全部结果又使用户对查询的准确度没有了解,从而造成错误的认知,甚至做出错误的决策.因此,本文考虑,返回所有的查询结果,并给定查询结果的一个一致性估计,使得用户获得查询结果的一致性程度,了解查询结果的总体特征.本文只考虑主键约束下不一致数据集查询结果的一致性估计.

例 1. 关系表 nba(Player,Season,Age,Tm,Lg,PTs)描述了球员参加 NBA 比赛的基本信息,如表 1 所示.其中 Player 表示球员的名字,Season,Age,Tm,Lg,PTs 分别描述了 NBA 赛季,年龄,所属球队,联赛,总的得分信息.Player,Season 下的下划线表示 Player,Season 是 nba 表的主键.假设查询 $q(x,t)=\mathrm{nba}(\underline{x,y},z,t,l,g)$,由一致性查询的定义,若用户只返回一致性的查询答案,则查询的一致性答案集合为{(Quincy Acy,TOR),(Jeff Adrien,CHA)}.

表 1 关系表 nba

Player	Season	Age	Tm	Lg	PTs
Quincy Acy	2013-14	23	TOR	NBA	9
Jeff Adrien	2013-14	27	CHA	NBA	209
Jeff Adrien	2013-14	26	CHA	NBA	57
Cole Aldrich	2013-14	25	NYK	NBA	2
Cole Aldrich	2013-14	24	TOT	NBA	100

显然,答案集合漏掉了 Cole Aldrich 的信息,这可能是用户想要获得的.为了尽可能全面回答用户的查询,本文返回所有的查询答案,同时给出查询答案的一致性准确程度给用户以指导,使得用户对查询的可靠性有一个大致的了解.在此次查询中,(Cole Aldrich,NYK)和(Cole Aldrich,TOT)是不一致的.一致性回答个数为 2,总的回答个数为 4,若将一致性准确率定义为一致性回答的个数占总回答个数的比例,则一致性准确率从为 2/4=1/2.用户可以根据一致性准确率信息判断查询信息的可用性.

从例 1 中可以看出,为了估计查询结果的一致性程度,需要确定答案集合中的每条元组的一致性.此问题可以形式化定义为 $CERTAINTY(q)$ 问题,即

$CERTAINTY(q)=\{I|$对于数据集 I 的每一个修复,存在一个元组 t,使得 $q(t)$ 为真$\}$.

之前的研究[3]指出此问题的复杂度上界为 coNP-完全,当 q 是一阶可表达时,复杂度降为一阶查询的复杂度 AC^0. q 是一阶可表达的是指 q 可以经过一阶改写得到一个一阶查询 q',q' 返回所有修复下满足 q 的答案集合,即 q 的一致性答案集合. q' 可以改写成 SQL 语句直接对关系数据集进行查询,也就是说,可以在多项式时间内得到查询的一致性结果,这使得一阶表达在一致性查询中有着重要的作用.然而,目前并没有一个可一阶表达与一致性查询判定易解的一个充要条件.因此,针对不可一阶表达的查询,本文引入了近似一致性的概念.本文的主要贡献如下:

(1)定义了查询结果的一致性程度估计,当查询不可一阶改写时,使用抽样的方法给出了一致性程度的一个(ε,δ)-估计.

① Woolsey B, Schulz M. Credit card statistics, industry facts, debt statistics. 2011. http://www.creditcards.com

(2) 对于不可一阶改写的查询，定义了新的查询改写 q_{relax}，设计判定算法使用 q_{relax} 的中间结果近似判定查询回答中元组 t 的一致性.

(3) 当合取查询中出现环以及自连接时，定义了查询改写 q_{acylic} 和 q_{sjf}，用来判定其回答的近似一致性.

(4) 实验验证了估计和近似判定算法的效率以及准确率.

本文第 1 节描述问题的背景和意义；第 2 节介绍相关工作；第 3 节给出背景知识介绍；第 4 节提出不可一阶改写查询答案一致性程度的(ε,δ)-估计算法；第 5 节介绍不可一阶改写查询的近似一致性判定方法；第 6 节使用实验验证估计算法的效率和准确性以及近似判定算法的效率和准确率；第 7 节总结全文以及提出未来工作.

2　相关工作

针对数据不一致问题，目前的处理方法主要有两种. 一种是建立描述一致性的相关规则，一旦数据集合中数据集违反了规则，则对数据进行修复使其处于一致的状态. 近年来随着数据质量越来越引起人们的重视，越来越多的规则被提出用来描述数据的一致性. 主要有以下几种：否定约束[4]、包含依赖[5]、外键约束和函数依赖[6]、聚集约束[7]、元组生成和等值生成依赖[3]. 同时有多种修复方式，例如：对称差分修复[3]、子集修复[8]、基数修复[9]、基于更新的修复[10]、投影连接修复[11]等.

即使检测出不一致数据，也可能没有确定的方法在保证数据真实准确的条件下完全修复不一致. 基于此，文献[12]第一次提出了一致性查询问题，一致性查询问题是指在存在不一致性数据的数据集合中，不完全修复数据情况下，查询得到一致性的查询结果，而查询结果的一致性由不一致数据的可能修复定义. 文献[3]定义了修复语义下的一致性查询，即查询需要满足所有的修复. 文献[13]是数据修复和一致性查询回答的一个综述. 国内也对一致性查询处理问题有了综述相关的研究[14]. 文献[15]研究了基于空值的一致性查询处理，将一致性约束转换为和查询相关的约束，查询过程中将不一致属性作为空值进而返回一致的查询结果，此方法得到的查询结果可能会丢失有用数据.

目前关于主键约束，子集修复的一致性查询已经有较多的研究成果. 文献[16]提出了 EQUIP 系统解决合取查询的一致性查询问题，其将一致性查询的补问题规约到 01 整形规划问题，通过求解规划方程去掉不满足一致性的解，最终得到一致性的回答. 此外，基于析取逻辑程序以及稳定语义模型可以解决任意合取查询的一致性回答问题[17]，且一致性限制并不局限于主键约束. 然而其复杂度为 Π_2^p. 文献[18-19]首次提出并研究了 $CERTAINTY(q)$ 问题，即布尔查询的一致性判定问题.

研究一致性查询的一个重要成果是一阶查询改写，一阶查询改写指将初始查询进行一阶改写，使得执行改写之后的查询可以得到原查询在数据集中的一致性查询结果. 文献[18-19]定义了一个可以一阶改写的不带自连接的合取查询类 C_{forest}，该类中的所有查询均可改写为一阶查询，从而在多项式时间内返回查询的一致性回答. 同时，他们指出对于 $q=\exists x\,\exists y(R(\underline{x},y)\wedge R(\underline{y},z))$ 不在 C_{forest} 中，然而其是一阶可表达的. 文献[20]提出了一个包含一阶可表达的查询的更大的类，指出不在该类中的涉及到两个不同关系表进行连接操作的查询一定是不可一阶改写的(关系表的个数大于 2 时没有此结果). 后来又给出了无环无自连接的合取查询一阶可表达的充分条件，其通过建立攻击图，当攻击图无环时则可得到查询时是一阶可表达的. 此外，文献[21]研究了满足函数依赖的情况下基于主键约束的一致性查询问题，其首先认为数据库是部分一致的(满足函数依赖)，基于此研究了 $CERTAINTY(q,\Sigma)$ 问题. 即在满足函数依赖集合 Σ 的情况下，q 是否是一阶可表达的，此问题限制 q 不带自连接. 文献[22-23]研究了 $CERTAINTY(q)$ 的变种：计数的复杂性问题，记为 $\sharp CERTAINTY(q)$. 给定一个数据集 db，$\sharp CERTAINTY(q)$ 返回满足 q 的 db 修复的个数，同时指出了对于不带自连接的合取查询 q，$\sharp CERTAINTY(q)$ 是 P 问题或是 $\sharp$P-完全问题.

本文的工作和查询一阶改写及 $\sharp CERTAINTY(q)$ 问题相关. 然而，不同于之前的工作，当查询不可进行一阶改写时，本文并不确切地计算 $\sharp CERTAINTY(q)$ 的值，因为此问题可能是难解的. 相应的，给定一个元组 t，计算 $\sharp CERTAINTY(q(t))$ 的一个下界，当其满足给定的阈值时，近似认为 t 属于一致性回答.

3　背景知识

首先，我们在 3.1 节介绍一致性查询的模型，3.2 节给出查询一阶可表达的判定方法.

3.1 一致性查询模型

定义 1. 不一致性. 对于描述一致性的规则(约束)集合 Σ,数据集合 I 称为一致的,当且仅当满足 $I \models \Sigma$,否则,I 成为不一致数据集.

定义 2. 修复 r 是不一致数据集 I 关于规则集合 Σ 的一个修复当且仅当:(1) $r \models \Sigma$,(2) 不存在 r' 满足 $r' \models \Sigma$ 且 r' 的修复代价小于 r.

定义 3. 一致性查询. 给定规则集合 Σ 以及不一致数据集 I,对 I 的任意查询 q,其一致性查询结果定义为 $CQA_{\Sigma}^{q}(I)=\bigcap CQA_{\Sigma}^{q}(r)$,其中 r 为 I 的所有可能修复.

本文研究的不一致查询问题的一致性约束规则为主键约束,修复类型为子集,查询类型为合取查询.

若关系数据集 I 满足主键为 x,关系模式为 $R(\underline{x},y,z)$ 的主键约束,则对于 $t_1,t_2 \in I$,若 $t_1.x=t_2.x$,则 $t_1.y=t_2.y$ 且 $t_1.z=t_2.z$. 将数据集中具有相同主键的元组结合称为一个块,假设主键值为 a,记 $\#block(a)$ 为主键值为 a 的元组集合的个数.

数据集 I 关于规则集合 Σ 的修复 r 是子集修复当且仅当:(1) $r \models \Sigma$,(2) 不存在 $|r'|<|r|$ 且 r' 是 I 的一个修复.

合取查询是包含选择、投影和连接操作的查询,其 datalog 表示形式为 $q(u_0):-R_1(u_1),R_2(u_2),\cdots,R_n(u_n)$,其中 $R_1,\cdots,R_n$ 表示关系名,$u_0,u_1,\cdots,u_n$ 是相应关系表中的自由元组,由常量和变量组成,$R_i(u_i)$ 称为 q 中的原子公式. 当 q 中所有表名只出现一次时,称查询 q 为不带自连接的查询. 当 u_0 为空时,q 为布尔合取查询.

3.2 查询一阶可表达的判定

主键约束下基于子集修复的合取查询的一致性查询是 coNP-完全问题. 然而,对于合取查询的一些子查询,可以在多项式时间内计算其一致性查询结果. 此类子查询具有的一个重要性质即一阶可改写.

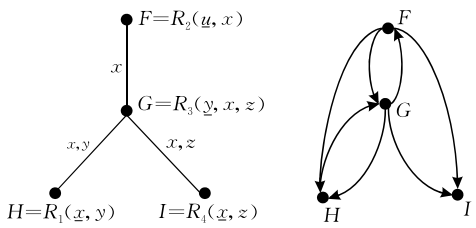
文献[24]中给出了无循环无自连接的一致性合取布尔查询一阶可改写的必要条件(见定理 1),其通过查询 q 的 join 树[25],建立查询 q 的攻击图,根据攻击图给出查询的一阶改写查询 q',使用 q' 对数据进行查询操作,查询的结果等价于 q 的一致性查询结果.

定理 1. 假设 q 为非循环无连接布尔连接查询,若 q 对应的攻击图 G_q 无环,则 $CERTAINTY(q)$ 是一阶可表达的.

攻击图的建立需要 join 树,查询 q 的 join 树是一个无向连通树,树中的每个节点表示查询中的一个原子公式且满足如下条件:若变量 x 出现在两个不同的原子公式 R_i 和 R_j 中,则 x 出现在连接 R_i 和 R_j 的唯一路径上的所有原子公式中. 若一个查询存在 join 树,则称该查询是无环的.

查询 q 的攻击图为一个有向图,攻击图中的顶点由 q 的原子公式构成. 原子公式具有主键约束,主键约束使得原子公式所代表的关系表中存在着属性值间的函数依赖. 例如,在原子公式 $F:R(\underline{x},y,z)$ 中,X 是 R 的主键,则存在着函数依赖 $\{x\}\rightarrow\{x,y,z\}$,可以简写为 $x\rightarrow xyz$(等价于 $x\rightarrow yz$). 此函数依赖不同于一般意义上的函数依赖,显然,此函数依赖的变量与查询相关. 记集合 $\kappa(q)$ 为查询 q 中的函数依赖集合. 集合 $F^{+,q}$ 为 F 的主键集合 $KVars(F)$ 关于原子 $q\backslash\{F\}$ 中的函数依赖集合的属性闭包. 其中,属性闭包的概念来源于关系数据库理论,如果 Σ 是属性集 U 上的函数依赖,$X\subseteq U$,则 X 关于 Σ 的属性闭包是集合 $\{A\in U \mid \Sigma \models X\rightarrow A\}$. 攻击图中有一条从 F 到原子公式 G 的边当且仅当 $F^{+,q}$ 不包含 join 树中 F 到 G 的唯一路径上出现的所有变量.

例 2. 查询 $q:-R_1(\underline{x},y),R_2(\underline{u},x),R_3(\underline{y},x,z),R_4(\underline{x},z)$ 的 join 树和攻击图为图 1[21] 所示.



(a) join树 (b) 攻击图

图 1 join 树和攻击图

对于无环查询 q 的攻击图 G_q,若图中顶点 $F=R(\underline{x},y)$ 的入度为 0,则称 F 不被任何顶点攻击,设 $q'=q\backslash\{F\}$,$\varphi(x,y)$ 为 $q'(x,y)$ 的一阶改写,则 q 可一阶改写为

$$\text{Rewrite}(F,q)=\exists x\exists y(R(\underline{x},y)\wedge\forall y(R(\underline{x},y)\rightarrow\varphi(x,y))).$$

4 问题定义及估计算法

4.1 问题定义

一致性查询估计的目标是估计查询结果的一致性程度,一致性程度是指查询结果中是一致性查询回答占总的查询结果的比例. 可以用一个概率值来描述,记为一致性程度 p_c,此估计问题形式化定义如下:

问题：一致性查询估计

输入：合取查询 Q，D，主键约束集合 Σ

输出：$Q(D)$的一致性程度 p_c

已知当查询 Q 一阶可表达时，可以在多项式时间内得到 Q 的一致性回答，因此，估计算法需要首先判断 Q 是否是一阶可表达的. 当查询可一阶表达时，则可以直接执行其一阶改写后的查询，得到一致性答案集合，使用一致性答案集合的元组个数除以 $Q(D)$中的元组个数，计算一致性程度估计 p_c. 当查询不可一阶表达时且不能在多项式时间内计算其一致性查询结果时，需要近似的估计查询结果的一致性程度.

下一节(4.2 节)说明当查询不可一阶可表达时，如何估计查询结果的一致性程度.

4.2　一阶不可表达的查询结果一致性估计

当 Q 不可一阶表达时，且 Q 的一致性回答不能在多项式时间内计算时，需要 $Q(D)$中的每条元组，将其元组 t 的属性值赋给 Q 后形成布尔查询 $q=Q(t)$，判断 q 的一致性回答是否为“是”，当 q 的一致性回答为“是”时，t 是 Q 的一致性回答. 然而，当查询结果集很大时，若是对每一条元组都判断其是否是一致的回答将会非常耗时. 因此，当查询的个数大于 30 时，本文将采用抽样的方法给出查询结果 p_c 的一个(ε,δ)-估计 $\hat{p}_c$，(ε,δ)-估计的定义见定义 4. 当查询结果小于等于 30 时，对答案集中的每一条元组，判定其一致性，返回一致性的元组在答案集合中的比例.

定义 4.　(ε,δ)-估计. 对于任意$\varepsilon(\varepsilon\geq 0)$和$\delta(0\leq\delta\leq 1)$，$\hat{I}$ 是估计量 I 的估计值，如果$Pr(|(\hat{I}-I)/I|\geq\varepsilon)\leq\delta$. 其中 $Pr(X)$随机事件 X 发生的概率，称 $\hat{I}$ 是 I 的(ε,δ)-估计.

将 $Q(D)$做为总体，可以用随机变量 X_i 表示 $Q(D)$中第 i 个元组 t_i的一致性. 其中，$X_i=1$ 表示 t_i是一致的答案，即对于数据所有的修复，t_i都出现在查询的结果中. $X_i=0$ 表示 t_i是不一致的答案，即存在一个修复，在此修复下，t_i不出现在查询的结果中. 基于此，抽样算法如算法 1 所示.

算法 1.　抽样算法.

1. 根据 ε,δ 确定抽样的次数；
2. 对 $Q(D)$进行重复抽样，判定抽出的样本的一致性；
3. 计算 $Q(D)$的一致性程度估计 $\hat{p}_c$.

算法第 1 步需要确定抽样次数，抽样次数由 ε,δ 决定. 由定理 2 可知，抽样的次数和查询回答的个数无关.

定理 2.　如果抽样次数 n 满足：$n\geq\frac{\phi^2_{\delta/2}}{4\varepsilon^2}+1$，则 $\hat{p}_c$是 p_c的(ε,δ)-估计，其中 $\phi_{\delta/2}$是标准正态分布的 $\delta/2$ 分位点.

证明.　$\hat{p}_c$的方差

$$var(\hat{p}_c)=\frac{s^2}{n}=\frac{1}{n}\cdot\frac{n}{n-1}\cdot\hat{p}_c(1-\hat{p}_c)\leq\frac{1}{n-1}\cdot\frac{1}{4}$$

由 $n\geq\frac{\phi^2_{\delta/2}}{4\varepsilon^2}+1$ 知：$n-1\geq\frac{\phi^2_{\delta/2}}{4\varepsilon^2}$，即$\frac{1}{4(n-1)}\leq\frac{\varepsilon^2}{\phi^2_{\delta/2}}$

可得 $var(\hat{p}_c)\leq\frac{\varepsilon^2}{\phi^2_{\delta/2}}$

又由 $0\leq E(p_c)=p_c\leq 1$，$var(\hat{p}_c)\leq\frac{\varepsilon^2}{\phi^2_{\delta/2}}p_c^2$

可得 $\phi_{\delta/2}\times\sqrt{var(\hat{p}_c)}\leq\varepsilon p_c$

则由中心极限定理可得

$$Pr(|\hat{p}_c-p_c|\geq\phi_{\delta/2}\times\sqrt{var(\hat{p}_c)})\leq\delta$$

则 $Pr(|\hat{p}_c-p_c|\geq\varepsilon p_c)\leq\delta$，

即 $Pr(|(\hat{p}_c-p_c)/p_c|\geq\varepsilon)\leq\delta$.　　证毕.

算法第 2 步需要对抽出的样本进行一致性判定，即判定 $t\in CERTAINTY(q)$？已知此问题的上界是 coNP-完全问题，因此在 NP≠P 的假设下没有确定的多项式时间算法判定查询是否有一致性答案. 基于此，对于复杂度为 coNP-完全问题的一致性查询，定义了近似一致性判定. 第 5 节介绍近似一致性判定算法.

对于不可一阶表达，但是一致性回答可以在多项式时间内获得的查询. 例如，查询 $Q(z,w)$：$-R_1(\underline{x},y,z)$，$R_2(\underline{y},x,w)$，其不可一阶改写，然而，可以在多项式时间查询得到其一致性解. 此类查询的一个重要特征是连接属性全部出现在查询的主键变量集合中. 对于此类查询，可以首先对其攻击图进行去环查询改写(见定义 7)，在去环查询得到的结果集中在多项式时间内过滤掉不满足一致性的解，从而精确的计算一致性程度.

5　近似一致性判定算法

由于当查询无环且不存在自连接时，定理 1 成立，因此本文分两种情况考虑布尔查询 q 的一致性判定. 5.1 节针对无环无自连接的查询，5.2 节简单地说明了查询有环或是自连接时的处理方法.

5.1　无环无自连接的查询近似一致性判定算法

当 G_q有环时，使用攻击图无法在多项式时间内判定 $CERTAINTY(q)$，因此，本节提出了近似一致性判定的概念. 在介绍近似一致性判定之前，我们首

先介绍一致性准确率的概念.

定义 5. 一致性准确率. 假设数据实例 I 有 n 种修复,元组 $t\in q(I)$ 在 n' 中修复下被返回,则 t 的一致性准确率为 n'/n.

在近似一致性判定中,使用一致性准确率来表示一致性的近似程度.

定义 6. 近似一致性回答(τ-$CERTAINTY(q)$):给定近似阈值 $\tau(0<\tau<1)$,对于 $t\in q(I)$,如果 t 的准确率 $p_t\geq\tau$,则 t 为 q 的 τ-近似一致性回答.

当 $\tau=1$ 时,t 为 q 的一致性查询回答,若 $CERTAINTY(q)$ 不是一阶可表达的,则 $CERTAINTY(q)$ 可能是 co-NP 完全问题. 因此,设 $\tau<1$,计算 t 的 τ-近似一致性回答判定答案.

通过观察我们发现:对于 q,若 q 的一致性回答为"是",则去掉 G_q 的环,对其进行改写得到 q_{relax},则 q_{relax} 的一致性回答为"是",即所有提供 q 一致性回答为"是"的中间结果是所有提供 q_{relax} 一致性回答为"是"的中间结果的子集. 因此,定义了查询的去环操作,如定义 7 所示.

定义 7. 去环查询改写. 对于无环无自连接的合取查询 $q(u_0)=R_1(u_1),R_2(u_2),\cdots,R_n(u_n),n\geq1$,若 q 的攻击图存在环 $R_i,R_j,\cdots,R_i$,则存在一个查询 $q_{\text{relax}}(u_0)=R_1(u_1'),R_2(u_2'),\cdots,R_n(u_n'),n\geq1$,$u_i'$ 为 R_i 中修改的变量集合. 若 q_{relax} 的攻击图无环,且不存在 u'',使得 $u''\subseteq u'$,则称 q_{relax} 是 q 的一个去环查询改写.

例 3. 设 $q_1=R_1(\underline{x},y),R_2(\underline{z},y)$,则改写后的查询 $q_{\text{relax}}=R_1(\underline{x},y),R_2(\underline{z},y')$.

去环操作放松了查询条件,减少了连接属性的个数. 因此,去环之后的查询 q_{relax} 和原来的查询 q 在查询结果上有定理 3 所描述的关系. 此外,提供 q_{relax} 一致性回答为"是"的中间结果包含执行 q 一致性查询的中间结果. 基于此,本文使用 q_{relax} 的中间结果近似判定 q 的一致性回答是否为"是". 定义 8 形式化定义了近似判定 q 的一致性回答需要用到的 q_{relax} 的中间结果.

定理 3. 查询 q 的改写 q_{relax} 的一致性查询结果为否,则 q 的查询结果为否.

显然,由于 q_{relax} 实际上放松了 q 的查询条件,定理 3 成立.

定义 8. 见证元组集合. 设 q_{relax} 是 q 的一个去环查询改写,若 q_{relax} 的一致性回答为是,则见证元组集合 $witness(U_{\text{key}},U)$ 为使得 q_{relax} 回答为是的中间结果集在 U_{key},U 属性集上的投影. 其中 U_{key},U 由常量和变量组成,U_{key} 表示 q 改写生成 q_{relax} 时改变的原子的主键值,U 表示改变后的原子变量和被改变的原子变量组成的变量对的值.

例 4. 查询 q_1 的见证元组集合 $witness(\underline{x},\underline{z},(y,y'))$ 为 R_1,R_2 的笛卡尔集.

若 q_{relax} 的一致性回答为"否",则由定理 3,q 的一致性回答为"否". 假设执行 q_{relax} 一致性回答为"是",执行过程中生成的 $witness(U_{\text{key}},U)$ 中有元组 t 使得 $t.U$ 中每对值相等,则可以根据 t 计算满足 q 的修复比例,即一致性准确率,近似判断 q 的一致性回答. 满足 q 的修复由两部分组成:(1) $t'.U_{\text{key}}=t.U_{\text{key}},t'.U\neq t.U$ 且 t' 出现在 $witness$ 中的元组加上 t 所代表的修复个数之和,记满足条件的元组个数为 l. 此部分元组满足修复的比例为 $frac\_part1=l/\sharp t$,其中 $\sharp t=\prod_{1\leq i\leq k}\sharp block(t.x_i)$. 比如表 3 中 $t,t_1,t_2,t_3,t_4,l=5$,$frac\_part1=5/\sharp t$;(2) $t'.U_{\text{key}}=t.U_{\text{key}}$ 且不出现在 $witness$ 中的元组集合所代表的修复个数. 比如元组 $(a,\cdots,c,(0,1),\cdots(1,1))$. (1)和(2)组成的元组共有 $\sharp t$ 个,需要为 $\sharp t-l$ 个元组计算其所能代表的 q 满足的修复个数. 查看表,如果存在 t_5,t_6 类型的元组,即 $t'.U_{\text{key}}\neq t.U_{\text{key}}$ 使 q 为"是",则说明在取 t_5,t_6 所在的元组为数据修复的一部分时,x_k 所在的关系表中主键为 c 值的元组可以任取组成完整的修复. t_5,t_6 的不同在于 $t_5.x_1=t.x_1$,t_5 提供的修复比例为 $(\sharp t-l)\cdot\sharp block(x_1)/(\sharp t\cdot\sharp t_5)$,$t_6$ 提供的修复比例为 $(\sharp t-l)/(\sharp t\cdot\sharp t_5)$. 这是因为对(2)中满足条件的每个元组而言,$t_5$ 类型的元组满足部分修复为形成 t_5 的 x_k 所在的关系表中的元组,主键为 d 值中的一条,以及形成 t_5 的 x_1 所在的关系表中主键值为 a 的一条,以及其他关系表的任意修复. 因为 a 已经出现在 t 中,因此,其代表的修复比例为 $\sharp block(x_1)/(\sharp t\cdot\sharp t_5)$. 然而,$t_5$ 和 t_6 提供的修复中有重复的情况,需要减去交集,即交集占的修复比例为 $1/(\sharp t_5\cdot\sharp t_6)$. 因此,假设有 m' 条类似于 t_5,t_6 这样的元组 $t_1',\cdots,t_m'$,则其提供的修复比例为 $p'=(\sharp t-l)\,p(frac\_t_1'\cup\cdots\cup frac\_t_{m'}')$,$frac\_t_i'$ 为元组 t_i' 提供的修复比例,其中 $frac\_t_i'=\left\{\prod_{1\leq i\leq k,\,t_i'.x_i=t.x_i}\sharp block(t.x_i)\right\}\Big/(\sharp t_i'\cdot\sharp t)$,$p(\bigcap frac\_t_i')=\prod 1/\sharp t_i'$,$p'$ 可使用相容事件发生的概率公式计算. 然而,当 m' 较大时,计算为指数时间. 由于 $p'\geq(\sharp t-l)\,p(frac\_t_i)$,因此我们可以首先计算单个元组提供的修复比例,将 $p=p+p'$ 和给定的近似性阈值比较,以尽早终止算法. 显然,当处理完所有的(1)(2)类型的元组时,$p<1$,则 q 的一致性答案返回"否".

表 2　表 *witness*

t_0	x_1	…	x_k	(y_1,y_1')	…	(y_m,y_m')
t_1	a	…	c	0	…	1
t_2	a	…	c	0	…	0
t_3	a	…	c	1	…	1
t_4	a	…	c	1	…	0
t_5	a	…	d	1		0
t_6	b	…	f	1		1
…	…	…	…	…	…	…

具体判定算法如算法 2 所示.

算法 2.　近似一致性判定.

输入：查询 q 的去环查询改写 q_{relax}，近似性阈值 τ

输出：q 的近似一致性回答“是”(1)或者“否”(0)

1. 一阶改写 q_{relax} 为 q_{fo}，执行 q_{fo}，记录见证元组集合 $witness(U_{\text{key}},U)$，设 $U_{\text{key}}=\{x_1,\cdots,x_k\}$，$U=\{(y_1,y_1'),\cdots,(y_m,y_m')\}$，若 q_{fo} 结果为空，则返回“0”；

2. 否则，删除 $witness(U_{\text{key}},U)$ 中 U 中 $y_i\neq y_i'$ 的元组. 若删除后有元组 t，对于 $1\leq i\leq k$，$\#block(t.x_i)=1$，则返回“1”；

3. 否则，寻找元组 t，使得 $\prod\limits_{1\leq i\leq k}1/\#block(t.x_i)$ 最小，由过程 1 和过程 2 计算返回结果.

过程 1.

1. 寻找 $witness$ 中 $t'.U_{\text{key}}=t.U_{\text{key}}$，$t'.U\neq t.U$ 记满足条件的元组个数为 l；

2. $p=(l+1)/\#t$.

过程 2.

1. 寻找 $witness$ 中 $t'.U_{\text{key}}\neq t.U_{\text{key}}$ 的元组集合 $t_1',\cdots,t_m'$. $j=1$；

2. 计算 $p'=(\#t-l)p(\bigcup\limits_{1\leq i\leq j}frac\_t_1)$；

3. 若 $p+p'\geq\tau$ 且 $j<m'$，则返回“1”；若 $j=m'$，$p+p'<1$，则返回“0”；若 $p+p'<\tau$ 且 $j<m'$，则 $j=j+1$，返回步 2.

近似一致性判定算法第 1 步需要将 q_{relax} 改写为一阶查询，之后可以将一阶查询修改为 SQL 语句，在现有的支持 SQL 操作的数据库上如 PostSql，可以直接执行. 若执行查询后查询结果为空，由定理 3，直接返回 q 的一致性回答为“否”，否则，算法在第 2 步中判断是否由一致性的数据可以使 q 的回答为“是”. 若有，则说明不一致数据集无论怎么修复，都存在元组使得 q 的回答为“是”，即一致性判定结果为“是”. 若不存在这样的一致性元组，则算法执行第 3 步. 分之前介绍的两种情况计算满足 q 的回答为“是”的修复比例. 在计算过程中，如果计算的修复比例 $p\geq\tau$ 时，算法终止，返回“是”.

显然，若 t 不是 q 的一致性查询回答，算法提前结束，然而 t 可能为 τ-$CERTAINTY(q)$. 在估计查询答案集合的一致程度时，若将 t 作为满足 τ-$CERTAINTY(q)$ 的答案返回，则会过高的估计查询的一致程度，因此，只有当 t 在当前计算中不能断定其是不一致时，才考虑近似一致性，近似认为 t 是一致的.

算法的正确性分析.

对于 $witness$ 中某个固定的 t，算法考虑了 t 所涉及到的关系表的主键在所有可能下的修复情况，因此，当 $\tau=1$ 时，算法能够确定的计算出所有满足 q 的修复比例. 当计算的修复比例为 1 时，确定的说明了 q 的一致性回答为“是”，否则，则说明 q 的一致性回答为否. $\tau<1$ 时，当 $p>\tau$ 时，近似返回了 q 的一致性回答为“是”；当 $p<\tau$ 且算法终止时，q 的一致性回答可确定为“否”，即对于“否”的答案，算法返回的结果都是确定的. 对于“是”的答案，算法返回的结果可能是近似一致的.

算法的时间复杂度分析.

对于 $q(u_0)=R_1(u_1),R_2(u_2),\cdots,R_m(u_m)$，$m\geq1$，攻击图建立的时间为 $O(m^3)$，无循环改写和一阶改写为常数时间. 设 R_i 的元组个数 $\leq n$，算法第 2 步的时间为 $O(n^m)$，算法第 2 步最坏情况下为 m 个表的笛卡尔积，即 $O(n^m)$. 第 3 步最坏情况下需要调用过程 X，假设 $witness$ 最坏情况下为 $O(n^m)$，则计算 p' 需要花费的时间为 $n^m+C_{n^m}^2+\cdots+C_{n^m}^{n^m}=2^{n^m}$，因此总的时间复杂度为 $O(2^{n^m})$. 而在最坏情况下 $\tau=1$ 时计算 q 答案为“是”时的修复个数为 $\#p$-完全问题[23]. 然而，在现实数据库中，由于多表连接结果远小于 $O(n^m)$，且给定的 $\tau<1$，数据集的不一致性很小，造成且生成的 $witness$ 表也不会太大. 因此，算法在执行过程中的时间复杂度要远远低于指数级. 实验很好地说明了这一点.

5.2　循环和自连接的查询 τ-近似一致性回答判定

对于无循环的合取查询，可利用 join 树建立权重攻击图，而有循环的合取查询，不存在 join 树. 文献[24]中指出有环的查询可以使用 intersection 树建立攻击图. 然而，intersection 树建立的攻击图不唯一，因此存在误判的情况，即攻击图无环时，查询不一定可一阶改写. 考虑到此情况，本文将在建立攻击图之前改写查询 q，将导致环出现的查询变量使用新的变量代替 q_{acylic}，然后为新的查询建立攻击图. 此时，当攻击图中无环时，由定理 1 可得，q_{acylic} 是一阶可表达的. 显然，q_{acylic} 的回答为“是”并不意味着 q 的一致性答案为是. 此时，我们需要计算 q_{acylic} 计算过程中生成“是”的中间结果改写后变量的赋值是否

和原始变量值相等，可以使用 5.1 节算法近似判断 q 的一致性. 当攻击图有环时，需要对 q_{acylic} 进行查询改写，此时 $q_{\text{acylic}_{\text{relax}}}$ 生成“是”的证据 *witness* 应该包括 q_{acylic} 中的变量以及新的改写新加的变量集合，之后的过程同 5.1 节相同.

对于存在自连接的查询，假设查询 q 中存在自连接 $R(\underline{x},y)\wedge R(\underline{y},z)$，则引入新的原子公式 R'，使得 $R(\underline{x},y)\wedge R(\underline{y},z)$ 改写为 $R(\underline{x},y)\wedge R'(\underline{y},z)$，形成新的查询 q_{sjf}，然后将 q_{sjf} 作为无自连接的查询判断其回答的一致性. 无论 q_{sjf} 是否可一阶表达，都需要在 *witness* 中判断是否 $(x=y)\wedge(y=z)$.

6 实 验

当 $q(u)$ 为合取查询时，在主键约束下解决 $CERTAINTY(q(u))$ 问题主要有 Conquer[18] 和 EQUIP[16] 两个系统，其中 Conquer 系统适用于查询可进行一阶改写的情况. 此系统将改写过后的一阶查询再次改写为 SQL 语句，可以在任意不一致的数据集中得到一致的查询结果. 显然，此系统不依赖于数据集，仅依赖于查询语句. EQUIP 系统则适用于所有的合取查询，该系统将数据集和查询规约成一个 0-1 整数规划(BIP)问题. 通过解方程组得到不满足一致性查询的解，将不一致解全部去掉之后得到最终一致性的查询结果. 较之 Conquer 系统，此系统依赖于数据集，当数据集改变时，整数规划方程需要重新建立. 由于系统性能受多种因素的影响，例如，数据集的不一致程度，查询解空间的大小等. 目前，并没有一个基准的测试用于比较其各自的性能[21]. 不过，大量的实验表明，如果查询是一阶可表达的，则将一阶查询变换为 SQL 语句的执行效率要高于 EQUIP 系统的效率. 这是因为 BIP 问题是 NP-完全问题，而一阶查询的问题复杂性为 AC^0.

不同于 Conquer 系统和 EQUIP 系统，本文不关注于求解查询的一致性答案，而是评估查询 q 在不一致数据集中的查询结果的一致性. 显然，对于不涉及到数据集中不一致数据的查询结果，其一定是一致的. 对于涉及到不一致数据的查询结果，则需要评估其一致性，因此，实验关注于评估的效率和评估的准确度以及不一致程度对评估准确度和效率的影响.

6.1 实验配置和数据集

本文的实验在 PC 机上运行，其内存 4 GB，硬盘 500 GB，CPU 为 Pentium@2.93 GHz，操作系统为 Windows 7，所有算法用 C++实现. 其中，查询实验使用 PostgreSQL，版本为 9.3.4-3.

(1) 基准查询. 对于不同的查询，一致性查询问题的复杂度从多项式时间到 coNP-完全不等. 因此，我们设计多组查询针对不同的情况对查询的一致性进行估计.

查询从时间复杂度上可分为 3 组：① 可进行一阶改写的查询；② 多项式时间，但是不可进行一阶改写；③ coNP-完全. 对于多项式时间但是不可进行一阶改写的查询，可以对其进行 q_{relax} 改写之后在多项式时间过滤掉不满足 q 一致性回答的 q_{relax} 回答. 3 组查询设计来源于文献[16]，如表 3 所示.

表 3 基准查询

可一阶改写查询	$Q_1(z):-R_1(\underline{x},y,z),R_2(\underline{y},v,w)$
	$Q_2(z):-R_1(\underline{x},y,z),R_2(\underline{y},x,w),R_3(\underline{x},y,d)$
多项式时间但不可一阶改写查询	$Q_3(z,w):-R_1(\underline{x},y,z),R_2(\underline{y},x,w)$
	$Q_4(z):-R_1(\underline{x},y,z),R_2(\underline{y},x,w),R_3(\underline{y},u,d)$
coNP-Hard 查询	$Q_5(z):-R_1(\underline{x},y,z),R_2(\underline{x}',y,w)$
	$Q_6(z):-R_1(\underline{x},y,z),R_2(\underline{x}',y,y),R_3(\underline{y},u,d)$

(2) 真实数据集. 真实数据集是 NBA 球员数据集①. 该数据集来源于篮球信息的一个网站以及维基百科，由 3 个表组成：球队信息 team，由 68 条记录构成，球员球队关系表 play_team，由 9413 条记录构成；以及教练球队关系表 team_coach，由 639 条记录构成. 由于数据的记录时间不同，若选择合适的主键，则存在着不一致的数据. 例如，在球员球队关系表中，假设以球员作为主键，则球员可能会在不同的时期属于不同的球队. 受数据集的限制，我们设计两类真实数据集上的查询：一类为多项式时间可解的，另一类的计算复杂度为 coNP-完全，查询设计如下：

$$q_1(y,z)=\text{team}(\underline{x}),\text{play\_team}(\underline{y},x),\text{team\_coach}(\underline{x},z),$$

$$q_2(y,z)=\text{team}(\underline{x}),\text{play\_team}(\underline{y},x),\text{team\_coach}(\underline{z},x),$$

其中 x 表示球队，y 表示球员，z 表示教练. q_1 的一致性回答在多项式时间内可以，q_2 则不可以.

(3) 合成数据集. 由于 TPC-H 的数据集大多不能表示不可一阶改写的查询的语义特征，因此，本文设计数据来验证基准查询的性能和估计的准确性. 数据集的生成由两部分组成：① 根据查询生成一致

① http://www.basketball-reference.com 和 wikipedia

的数据集.为了方便查询的执行,数据集中的属性值类型均为数字类型,数据集中关系表的主键根据选择的基准查询来确定,关系表的规模相同,在生成关系表时,使得任意两表连接的结果大概占原始表的20%;②在一致的数据集中引入不一致性.采用的方法为在一致的数据集中加入新的元组,新元组的主键值来源于一致数据集中的主键值.此过程由两个参数控制:① $\sharp conflicts$,即加入元组的个数;② key_size,即主键相同的元组的个数,默认为 2.

(4)参数设置.设计近似一致性的近似阈值为 0.5;抽样算法中 ε 为 0.1,δ 为 0.2.

6.2　实验结果

实验的目标是验证估计算法的效率和估计的准确率,以及判定算法的效率和准确率.我们首先在真实数据集上验证估计和判定算法的效率和准确率.之后在合成数据集上更深一步地挖掘算法所具有的特征.

数据集中,不违背主键约束的数据是一致的,从这部分数据中查询所得到的结果也是一致的.因此,每次实验我们将查询结果集 ans 分为两个部分:ans_c 和 ans_n.其中 ans_c 代表从一致的数据中查询得到的答案,ans_n 表示从剩下的数据中查询得到的结果和 ans_c 的差集.显然,在验证估计算法时,ans_c 集合中答案的一致性判定为 1,只需对 ans_n 中的答案进行近似一致性判定时.实验过程中抽样算法执行五次,其平均值为算法的效率和评估的准确度,同时对抽出的样本计算其判定的平均准确率.

6.2.1　真实数据集上算法的效率和准确性

由于 q_1 在多项式时间内可以查询到其一致的解.因此,其一致性为 q_1 经过一阶改写的查询得的结果集的个数占 q_1 查询的结果个数的比例,估计的准确度为 1.此时,不需要进行近似一致性判定.对于 q_2,首先要对其进行 q_{relax} 改写,在 q_{relax} 查询的结果集中进行抽样,判断样本的近似一致性.两个查询的效率和准确率如表 4 所示.

表 4　真实数据上的一致性查询效率及有效性

查询	效率/ms	评估准确率	近似判定准确率
q_1	2.020	1.00	
q_2	2.310	0.85	
cons(q_1)	11.511		1.0
cons(q_2)	100.000		0.7

由表 4 可以看出,一致性查询的时间要远远大于查询时间,其中,q_2 的一致性查询我们采用 EQUIP 中的方法得到查询结果.q_1 的一致性估计的准确度为 100%,一致性判定准确率也为 100%,这是因为我们可以在多项式时间内求得 q_1 的一致性回答.q_2 的一致性判定准确率为 85%,其一致性估计的准确度由两个因素决定:抽样算法给定的 δ 以及近似一致性估计所带来的误差.实验中查询生成 112724 条结果记录.其中 ans_n 的个数为 103165:抽样算法抽出 42 条记录判断一致性.从图中可以看出,其总的误差为 15%.准确率没有超过 90%的原因是由于近似一致性估计中错误的将不一致的答案判成了一致性答案.近似一致性判定的平均准确率为 70%.

6.2.2　合成数据集上算法的效率和准确度.

为了进一步研究算法的性能,我们在合成数据集上进行了一系列的实验.首先验证了评估算法的时间效率.

图 2 比较了基准的 6 个查询直接查询得到查询结果的时间效率以及查询一致性结果的效率.从图中可以看出,Q2 的一致性查询时间较长,这是因为 Q2 的一阶改写较为复杂.Q5,Q6 的一致性判定为 coNP-完全问题.在实际判定时,由于连接结果可能很小,可迅速的判断出其一致性,即近似估计算法在实际计算中效率不会很差.其次,实验比较了评估算法的准确性.评估的准确性由两部分确定:(1)抽样算法的相对误差造成的准确性降低;(2)判定算法中判定查询结果是一致的时候,使用了近似的方法,使得最终结果可能会出现将一致的数据误判为不一致的数据,进而造成评估值小于实际值.图 3 中由于近似阈值设置在 0.5,即当 50%的修复被满足时认为元组在一致性结果集中,造成了 coNP-Hard 查询的准确率最低在 50%左右.显然,能够一阶改写的查询的评估准确率为 100%,不能被一阶改写然而可以在多项式时间内可解的查询可以使用本文介绍的 q_{relax} 改写方法,得到精确的一致性解.

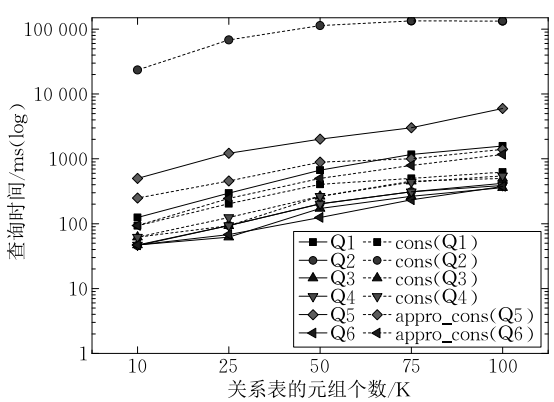


图 2　合成数据集上一致性查询的效率

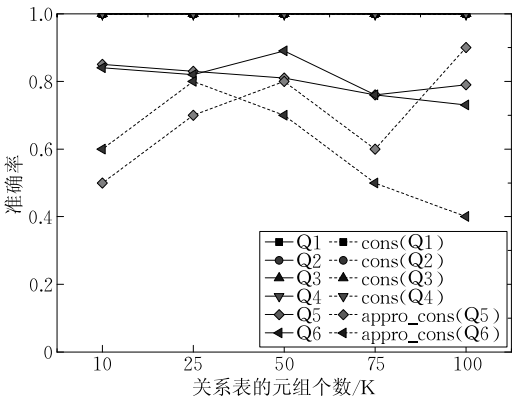


图 3 合成数据集上一致性查询的准确度

6.2.3 参数对算法性能的影响.

首先我们考虑抽样算法关于(ε,δ)的相对误差以及样本容量对相对误差的影响. 实验取 ε 为 0.05,0.1,0.15,0.2,δ 的值取 0.1,即 10%. 实验结果如图 4、图 5 所示. 从图 4 中我们可以看出,抽样算法的相对误差控制在 δ 附近,当 ε 取 0.05 时,相对误差的值为 8%,ε 越大,相对误差越小. 图 5 表明了随着样本容量的增加,相对误差逐渐减小. 当抽样的样本量为总体的 20%时,相对误差低于 3%. 这是因为当样本容量增加时,更能够反映样本的总体特征,从而使评估值更接近实际.

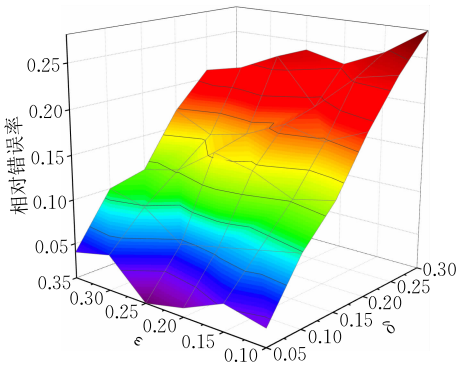


图 4 抽样算法关于(ε,δ)的相对误差

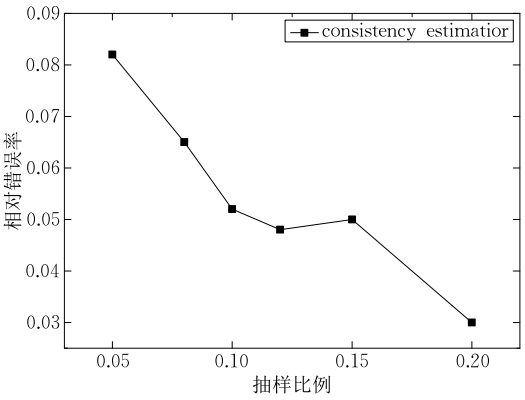


图 5 样本容量对相对误差的影响

其次,考虑了在近似一致性判定算法中,近似阈值 τ 对近似一致性判定准确率的影响. 此组实验取 Q5 查询 5 次测试的平均结果,结果如图 6 所示. 可以看出,随着近似阈值的增大,判定的准确度随之增加. 当近似阈值为 0.7 时,一致性判定的准确率在 90%以上. 显然,当近似阈值较大时,若一个元组是不一致回答,则其被判定为不一致的机会也大大增加,使得最后误判的结果数量减小,从而提高准确率.

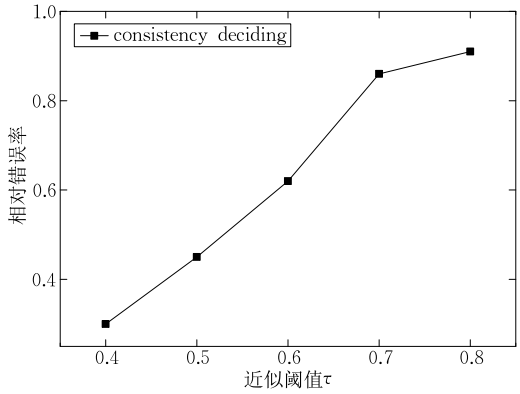


图 6 τ 对一致性判定准确率的影响

此外,我们还分析了数据集不一致程度以及查询结果集的大小对评估算法效率的影响,实验结果如图 7、图 8 所示. 当不一致程度增加时,满足一致性的查询回答减少,对于 coNP-Hard 一致性查询来说,其判定时间也会减少,见图 7 中 Q5,Q6 的评估时间,也随之减少,对于多项式时间及一阶可改写查询来说,其变化较少,只是因为不一致程度的增加不会改变数据集的扫描次数,对时间效率上不会有太大的影响. 而查询结果集的大小对一致性查询和近似一致性查询的影响很小,查询时间紧随着结果集的增大有着小幅增长. 需要注意的是对于 coNP-Hard 一致性查询来说,由于结果集在抽样之后可以一次性的判定其一致性,因此,结果集的增加不会使时间变化过大.

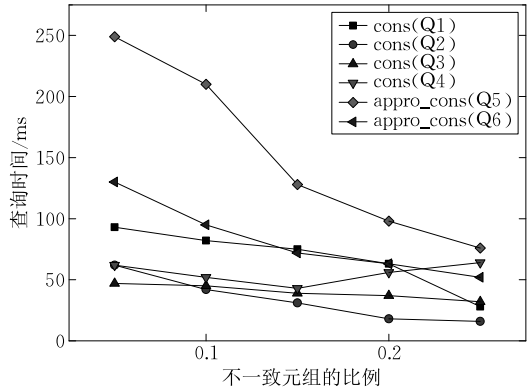


图 7 不一致程度对评估效率的影响

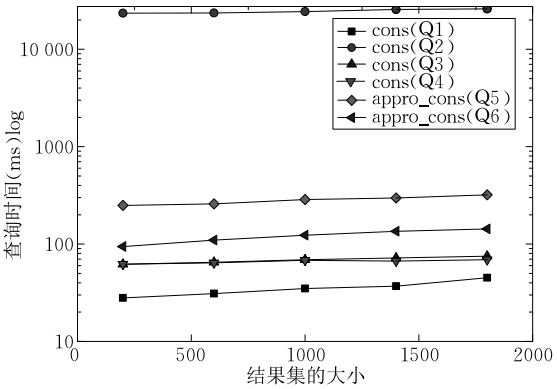


图 8　结果集的大小对一致性查询的影

7　结论和未来工作

本文研究了违背主键约束的不一致数据集中的查询结果一致性估计问题. 当查询不可一阶改写且不能在多项式时间计算时，采用简单抽样的方法给出了合取查询结果集关于一致性准确率的(ε,δ)-估计. 抽样算法均匀随机的抽取查询答案集合，判定样本的一致性. 由于判定答案是否是一致性回答的问题可能是难解的，本文定义了近似一致性，以大于给定的近似阈值来判定答案的一致性. 下一步工作将考虑一致性依赖是其他形式，比如函数依赖时查询结果的一致性估计问题，以及不一致数据程度对查询结果一致性的影响.

参　考　文　献

[1] Li Jian-Zhong, Liu Xian-Min. An important aspect of big data: Data usability. Journal of Computer Research and Development, 2013, 50(6): 1147-1162(in Chinese)
(李建中，刘显敏. 大数据的一个重要方面：数据可用性. 计算机研究与发展，2013，50(6)：1147-1162)

[2] Bertossi L, Chomicki J. Query answering in inconsistent databases//Chomicki J, van der Meyden R, Saake G eds. Logics for Emerging Applications of Databases. Berlin Heidelberg: Springer, 2004: 43-83

[3] Arenas M, Bertossi L, Chomicki J. Consistent query answers in inconsistent databases//Proceedings of the 18th ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems. Philadelphia, USA, 1999: 68-79

[4] Bertossi L, et al. The complexity and approximation of fixing numerical attributes in databases under integrity constraints. Information Systems, 2008, 33(4): 407-434

[5] Bravo L, Bertossi L. Consistent query answering under inclusion dependencies//Proceedings of the 2004 Conference of the Centre for Advanced Studies on Collaborative Research. Markham, Canada, 2004: 202-216

[6] Molinaro C, Greco S. Polynomial time queries over inconsistent databases with functional dependencies and foreign keys. Data & Knowledge Engineering, 2010, 69(7): 709-722

[7] Flesca S, Furfaro F, Parisi F. Querying and repairing inconsistent numerical databases. ACM Transactions on Database Systems, 2010, 35(2): 11-14

[8] Chomicki J, Marcinkowski J. Minimal-change integrity maintenance using tuple deletions. Information and Computation, 2005, 197(1): 90-121

[9] Lopatenko A, Bertossi L. Complexity of consistent query answering in databases under cardinality-based and incremental repair semantics//Proceedings of the 11th International Conference on Database Theory. Barcelona, Spain, 2006: 179-193

[10] Wijsen J. Database repairing using updates. ACM Transactions on Database Systems, 2005, 30(3): 722-768

[11] Wijsen J. Project-join-repair: An approach to consistent query answering under functional dependencies//Larsen H L, Pasi G, Ortiz-Arroyo D, et al, eds. Flexible Query Answering Systems. Milan, Italy: Springer, 2006: 1-12

[12] Bry F. Query answering in information systems with integrity constraints//Jajodia S, List W, McGregor G, Strous L eds. Integrity and Internal Control in Information Systems. USA: Springer, 1997: 113-130

[13] Bertossi L. Database repairing and consistent query answering. Synthesis Lectures on Data Management, 2011, 3(5): 1-121

[14] Xie Dong L Y. Study on consistent query answering in inconsistent databases. Frontiers of Computer Science in China, 2007, 1(4): 493-501

[15] Huang Fei, Liu Jie, Ye Dan. Consistent query answering based on virtual repairs with nulls. Application Research of Computers, 2009, 26(11): 4146-4150(in Chinese)
(黄飞，刘杰，叶丹. 基于空值修复的数据库一致性查询方法. 计算机应用研究，2009，26(11)：4146-4150)

[16] Kolaitis P G, Pema E, Tan W-C. Efficient querying of inconsistent databases with binary integer programming. Proceedings of the VLDB Endowment, 2013, 6(6): 397-408

[17] Barceló P, Bertossi L. Logic programs for querying inconsistent databases//Dahl V, Wadler P eds. Practical Aspects of Declarative Languages. New Orleans, USA: Springer, 2003: 208-222

[18] Fuxman A, Fazli E, Miller R J. Conquer: Efficient management of inconsistent databases//Proceedings of the 2005 ACM SIGMOD International Conference on Management of Data. Baltimore, USA, 2005: 155-166

[19] Fuxman A, Miller R J. First-order query rewriting for inconsistent databases. Journal of Computer and System Sciences, 2007, 73(4): 610-635

[20] Wijsen J. Consistent query answering under primary keys: A characterization of tractable queries//Proceedings of the 12th International Conference on Database Theory. Saint-Petersburg, Russia, 2009: 42-52

[21] Greco S, Pijcke F, Wijsen J. Certain query answering in partially consistent databases. Proceedings of the VLDB Endowment, 2014, 7(5): 353-364

[22] Maslowski D, Wijsen J. Counting database repairs that satisfy conjunctive queries with self-joins//Proceedings of the 17th International Conference on Database Theory. Athens, Greece, 2014: 155-164

[23] Maslowski D, Wijsen J. On counting database repairs//Proceedings of the 4th International Workshop on Logic in Databases. Uppsala, Sweden, 2011: 15-22

[24] Wijsen J. Certain conjunctive query answering in first-order logic. ACM Transactions on Database Systems, 2012, 37(2): 9

[25] Beeri C, et al. On the desirability of acyclic database schemes. Journal of the ACM, 1983, 30(3): 479-513

LIU Xue-Li, born in 1987, M. S. candidate. Her research interests is data quality, big data.

LI Jian-Zhong, born in 1950, professor, Ph. D. supervisor. His current research interests include data-intensive computing, wireless sensor networks and CPS, etc.

Background

This paper study consistent query estimation in inconsistent data based on primary key. It belongs to the data quality area. There are some related works concerning on consistent query deciding and consistent query answering in inconsistent data based on primary key problem. While, as far as we know, there are no works about consistent query estimation. We propose a (ε,δ)-estimation based on sampling method to estimate the consistency degree of query answers. And propose a concept of τ-approximation consistent deciding, also, we give an algorithm based on filter to approximately deciding the consistency of a query answer.

This study is fund by the National Basic Research Program (973 Program) of China under Grant No. 2010CB316200. This project aims to study the basic theory and key technology of massive information availability, hoping to achieve long-term development of social informatization in our country and the twelfth five-year strategic goal.

Our study group has been focusing on error detection, data repair and approximate computing over inconsistent data, incomplete data, outdated data and inaccuracy data. Many novel techniques and theory have been proposed to handle the error data management. Also, Many outstanding works have been published in worldwide conferences and transactions, such as SIGMOD, VLDB, ICDE, KDD, INFOCOME, TKDE, VLDB Journal et al.

This paper proposes a novel way on evaluation of consistency of query result over inconsistent data. It belongs to approximate computing theory and technique research.