软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software,2020,31(2):406−420 [doi: 10.13328/j.cnki.jos.005876]

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

基于符号语义的不完整数据聚集查询处理算法*

张安珍[1,2], 李建中[1], 高 宏[1]

[1](哈尔滨工业大学 计算机科学与技术学院,黑龙江 哈尔滨 150001)

[2](沈阳航空航天大学 计算机学院,辽宁 沈阳 110136)

通讯作者: 张安珍, E-mail: azzhang@hit.edu.cn

摘 要: 研究了基于符号语义的不完整数据聚集查询处理问题.不完整数据又称为缺失数据,缺失值包括可填充的和不可填充的两种类型.现有的缺失值填充算法不能保证填充后查询结果的准确度,为此,给出了不完整数据聚集查询结果的区间估计.在符号语义中扩展了传统关系数据库模型,提出了一种通用不完整数据库模型.该模型可以处理可填充的和不可填充的两种类型缺失值.在该模型下,提出一种新的不完整数据聚集查询结果语义:可靠结果.可靠结果是真实查询结果的区间估计,可以保证真实查询结果有很大概率在该估计区间范围内.给出了线性时间求解SUM、COUNT和AVG查询可靠结果的方法.真实数据集和合成数据集上的扩展实验验证了所提方法的有效性.

关键词: 不完整数据;近似查询处理;数据修复;结果估计;数据可用性

中图法分类号: TP311

中文引用格式: 张安珍,李建中,高宏.基于符号语义的不完整数据聚集查询处理算法.软件学报,2020,31(2):406−420. http://www.jos.org.cn/1000-9825/5876.htm

英文引用格式: Zhang AZ, Li JZ, Gao H. Aggregate query processing algorithm on incomplete data based on denotational semantics. Ruan Jian Xue Bao/Journal of Software, 2020,31(2):406−420 (in Chinese). http://www.jos.org.cn/1000-9825/5876.htm

Aggregate Query Processing Algorithm on Incomplete Data Based on Denotational Semantics

ZHANG An-Zhen[1,2], LI Jian-Zhong[1], GAO Hong[1]

[1](School of Computer Science and Technology, Harbin Institute of Technology, Harbin 150001, China)

[2](School of Computer Science, Shenyang Aerospace University, Shenyang 110136, China)

Abstract: This work studies the problem of aggregate query processing over incomplete data based on denotational semantics. Incomplete data is also known as missing values and can be classified into two categories: applicable nulls and inapplicable nulls. Existing imputation algorithms cannot guarantee the accuracy of the query result after imputation. The interval estimation of the aggregate query result is given. This study extends the relational model under the denotational semantic, which can cover all types of incomplete data. A new semantic of aggregate query answers over incomplete data is defined. Reliable answers are interval estimations of the ground-truth query results, which can cover the ground-truth results with high probability. For SUM, COUNT, and AVG queries, linear approximate evaluation algorithms are proposed to compute reliable answers. The extended experiments on the real datasets and synthetic datasets verify the effectiveness of the method proposed in this study.

Key words: incomplete data; approximate query processing; data reparation; result estimation; data usability

不完整数据(或缺失值)问题困扰数据库已久.有很多种原因可以导致数据缺失,主要分为两种:一种是可填充的缺失值,这种缺失值有对应的真实值;另一种是不可填充的缺失值,这种缺失值没有对应的真实值.不完整

* 基金项目: 国家自然科学基金(61702344)
Foundation item: National Natural Science Foundation of China (61702344)
收稿时间: 2018-12-11; 修改时间: 2019-04-25; 采用时间: 2019-07-02; jos 在线出版时间: 2019-08-09
CNKI 网络优先出版: 2019-08-12 12:08:00, http://kns.cnki.net/kcms/detail/11.2560.TP.20190812.1207.003.html

数据给查询处理带来很大的挑战,例如查询效率低下、数据分析困难、查询结果有误差等[1].不完整数据上的查询处理主要有两种解决方法:一种是缺失值填充,填充所有不完整的数据并在得到的完整数据集上进行查询处理;另一种方法是直接在不完整数据上进行查询处理.

缺失值填充方法是数据库界常采用的方法.研究人员提出了大量缺失值填充算法,如基于统计的、机器学习的[2].填充值也从内部填充扩展到外部知识填充.缺失值填充方法的优点是通过一次性修复保证后续查询结果的一致性,缺点是修复所有不完整数据的时间开销非常大,并且修复的结果往往很难达到准确度要求.没有任何一种缺失值填充算法能够保证填充结果的准确性,甚至填充后看似完整的数据与原始数据相差甚远,在填充后的数据集上执行查询,得到的查询结果也没有任何准确度保证.

在直接在不完整数据上查询处理方面,传统的关系数据库采用 NULL 来代替缺失值,并通过三值逻辑计算 NULL 在代数和布尔计算中的传递.然而在计算两个集合差值的时候,三值逻辑经常返回错误的结果.为此,研究人员提出确定结果(certain answers)作为不完整数据查询结果的语义.确定结果能够解决三值逻辑的缺陷,成为理论界广泛认可的标准.确定结果基于开放世界假设(open world assumption,简称 OWA)和封闭世界假设(closed world assumption,简称 CWA).封闭世界假设认为:数据库包含了现实世界的全部实体,只是其中一些实体的属性值缺失.开放世界假设认为:除了属性缺失外,现实世界中的实体也可能缺失.在给定假设下,确定结果通过对查询结果取交集找到满足所有可能世界的查询结果.然而,计算确定结果造成大量的不完整数据丢失,严重降低了查询结果的质量,并且计算确定结果的时间代价是指数级的,当数据规模较大时无法适用.

下面通过一个示例说明现有解决方法存在的问题.图 1 中的不完整充电记录表 *records* 记录了真实电动汽车充电信息.*records* 表有很严重的缺失值问题.造成充电记录缺失的原因有很多,例如,地址 *address* 属性缺失是由于记录人员操作马虎导致;故障类型 *error* 缺失是由于数据库设计模式不好,对于没发生故障的充电桩,*error* 值自然为空.电力公司经常在该表上执行查询语句 Q_1:SELECT *SUM*(*elect*) FROM *records* GROUP BY *area*.Q_1 旨在计算每个地区的总充电量,帮助电力公司管理人员分析哪个区域需要增援更多的充电桩,制定下一季度的调配方案.然而,由于 t_2 和 t_7 元组的 *area* 属性值缺失,无法判断其所在区域分组.若利用缺失值填充算法对缺失值进行填充,填充值很可能与真实值完全不同.若对 Q_1 求确定结果,确定结果中不包含 t_2 和 t_7,从而确定结果丢失了部分有效信息,则可靠性没有保证.

id	pile	money	elect	area	addr	lon	lat	error
t_1	3 018	27.15	22.64	CY	SCHOOL	—	—	—
t_2	—	—	—	—	—	—	—	NET-FAIL
t_3	1 007	64.66	53.88	SY	—	—	—	—
t_4	3 018	28.71	23.93	CY	SUBWAY	—	—	—
t_5	1 007	—	—	SY	—	—	—	PILE-FAIL
t_6	1 007	23.71	19.76	SY	—	—	—	—
t_7	1 007	10.51	15.03	—	—	—	—	—
t_8	3 018	24.71	20.6	CY	—	—	—	—

Fig.1 Incomplete charging records

图 1 不完整充电记录

为此,本文研究不完整数据查询处理问题,给出一种封闭世界假设下不完整数据聚集查询区间估计方法.在符号语义中扩展了传统的关系数据库模型,提出一种可以适用于不同缺失值的通用不完整数据库模型,并提出一种新的不完整数据上聚集查询结果语义:可靠结果.可靠结果是真实聚集查询结果的区间估计,通过估计不完整数据库所有可能世界的查询结果的上下界,保证真实查询结果一定在该区间范围内.一种直观的求解方法是:枚举不完整数据库所有的可能世界,在每个可能世界中计算聚集查询结果,最终得到查询结果的上下界.然而,这种方法的时间开销随缺失值个数的增加呈指数增长,在现实应用中很难使用.为此,本文给出一种线性时间求解方法,并通过理论分析证明其有效性.

本文给出的估计方法可以适用于任何缺失机制中.通常有 3 种缺失机制导致数据缺失:完全随机缺失(missing completely at random,简称 MCAR)、随机缺失(missing at random,简称 MAR)和不随机缺失(not missing at random,简称 NMAR).几乎所有的缺失值填充方法都假设数据缺失服从随机缺失(MAR).然而,现实中存在大量缺失不随机(NMAR)的情况.例如在问卷调查中,收入高的人群往往不愿意透露自己的薪资水平,从而导致薪资这一项缺失.在这种情况下,已有的缺失值填充技术不再适用,而本文给出的估计方法可以很好地应对缺失不随机的情况.本文给出的估计方法不假设数据分布,主要针对 SUM、COUNT 和 AVG 这 3 种聚集操作,其他聚集操作(VAR、GEOMEAN、PRODUCT 等)可以通过一定的扩展求得.

本文的主要贡献如下:

(1) 在符号语义理论下扩展关系数据库模型及聚集查询;
(2) 形式化定义了不完整数据聚集查询可靠结果问题,给出了可靠结果估计方法并通过理论分析证明其可靠性;
(3) 给出了 $O(N)$时间求 SUM 和 COUNT 查询可靠结果的算法,给出了 $O(N+m\log m)$时间求 AVG 查询可靠结果的算法,其中,N 是数据集大小,m 是数据中元组的个数;
(4) 通过真实数据集和合成数据集上的扩展实验,验证本文所提算法的有效性.

本文第 1 节回顾相关工作.第 2 节正式定义不完整数据聚集查询可靠结果问题.第 3 节给出基于符号语义的关系数据库模型及查询扩展.第 4 节给出求解 SUM、COUNT、AVG 聚集查询可靠结果的算法.第 5 节通过实验验证本文所提算法的准确性和可扩展性.最后,在第 6 节总结全文.

1 相关工作

- 缺失值填充

文献[3]研究缺失值填充的基础理论,分析了 3 种不完整数据修复方法:基于确定答案和表现系统的方法、基于逻辑的方法和基于相对值排序的方法,为缺失值填充建立了理论基础.按照缺失值填充所使用的数据来源,缺失值填充方法可以分为内部数据填充和外部数据填充:内部数据填充[4−8]是指利用正在修复的数据集合中的值来填充缺失值,而外部数据填充[9−11]是指利用正在修复的数据集合之外的数据来填充缺失值.缺失值填充按照填充方案可以分为两类:单值填充和多值填充.在单值填充算法中[12,13],一个缺失值只有 1 个候选的填充值;在多值填充算法中[14−16],每个缺失值通常有多个候选值,这些值按照可能性大小排序.通常情况下,多值填充能够得到更高的准确率,但是时间开销也较大.缺失值填充算法按照填充方法可以分为基于数据的填充算法、基于模型的填充算法和基于机器学习的填充算法:基于数据的填充算法利用已有的完整数据填充缺失数据,常见的方法有平均值填充、条件平均值填充、hot-dect 填充以及替换填充[17];基于模型的填充算法利用特定的模型计算缺失值,这类算法假设数据是基于某个特定模型生成的,主流的方法有回归模型[18]、似然函数模型[19]以及线性判别分析模型[17];基于机器学习的填充算法利用所有完整的数据和特定的机器学习算法计算缺失值,例如 C4.5 算法[20]、CLIP4 算法[21]、Naive-Bayes 算法[22].

- 不完整数据查询处理

早在 20 世纪 70 年代,Codd 就提出了三值逻辑用于解决 SQL 查询中缺失值问题.他提出用 NULL 来填充缺失值,并扩展算数和布尔计算使之适用于三值逻辑.然而很多情况下,三值逻辑不能给出正确的结果[23−27].为此,文献[28]提出了确定结果语义解决三值逻辑的缺陷,成为理论界广泛认可的标准.确定结果基于开放世界和封闭世界假设,在给定语义下,找到满足所有可能世界的确定结果集,该结果集中的元组和缺失的元组没有关联,无论缺失元组的真实取值是什么,它们都在确定结果中.然而,计算确定结果需要枚举所有可能世界,代价是指数级的,因此在现实应用中很少采用[29,30].不完整数据上其他查询包括聚集查询[31−33]、排序查询[34,35]、skyline 查询[36,37]、相似性查询[38]等.

2 问题定义

本文主要关注这类形式的聚集查询:SELECT AGG(attr) FROM table WHERE predicate.聚集属性是指出现在聚集操作(AGG)中的属性,AGG 可以是 SUM、COUNT、AVG 等.条件属性是指包含在选择条件语句(predicate)中的属性.给定不完整关系表 $R=\{D_1,D_2,\ldots,D_n\}$,聚集查询 AQ,本文旨在给出真实查询结果 $AQ(R^*)$的区间估计.

定义 1(查询结果偏差). 给定不完整关系表 R、聚集查询 AQ,查询结果偏差Δ为 R 上的聚集查询结果 $AQ(R)$和在真实完整关系表 R^*上查询结果的差值:$\Delta=AQ(R^*)-AQ(R)$.

根据定义 1 可知,真实查询结果可以利用查询结果偏差来估计,即 $\widehat{AQ}(R^*)=AQ(R)+\hat{\Delta}$.在实际应用的数据库中,很难预先知道缺失值的分布,通常缺失值的分布没有规律可循,因此很难给出Δ的无偏估计.有时数据库中的缺失值是由多种原因引起的,例如设备故障、人员操作失误、关系模式设计缺陷等等,没有单一的统计分析工具可以应对这种复杂的缺失情况.为此,本文从估计查询结果真实值的角度入手,通过估计Δ的最大值和最小值给出真实值所在的区间范围.

定义 2(可靠结果,reliable answer). 给定不完整关系表 R、聚集查询 AQ,可靠结果 $RA(R)$为 $RA(R)=[AQ(R)+\Delta_{\min},AQ(R)+\Delta_{\max}]$,其中,$\Delta_{\min}$ 和$\Delta_{\max}$ 分别是Δ的下界和上界.

不完整数据聚集查询可靠结果问题可形式化定义如下.

输入:不完整关系表 $R=\{D_1,D_2,\ldots,D_n\}$,聚集查询 AQ.

输出:可靠结果 $RA(R)$.

在现实应用中,数据分析人员很难相信不完整数据上的查询结果.即使经过复杂的数据清洗过程后,看似完整的数据和真实的数据之间的差距仍然未知.可靠结果可以帮助数据分析人员了解查询结果的概貌,从而帮助他们做出判断.

3 符号语义下关系数据库模型及查询扩展

本节先给出符号语义下关系数据模型的扩展,使得可填充缺失值和不可填充缺失值在该模型下得到很好的应用,然后给出该模型下的查询处理方法.

3.1 关系数据库模型扩展

令 R 表示有 n 个属性的关系表,$D_1,D_2,\ldots,D_n$ 是属性的值域.关系 R 可以表示为 $R:D_1\times D_2\times\ldots\times D_n\to T$,其中,$T$={true,false}.若元组 t 在关系 R 中,则 $R(t)$=true;否则,$R(t)$=false.在符号语义中,⊤和⊥两个符号被加入到属性值域中,其中,⊤表示不可填充的缺失值,⊥表示可填充的缺失值.此时,属性值域 D_i 扩展为 $D_i^0=D_i\cup\{\top,\bot\}$.例如,图 2 中的充电记录表 $record^0$ 是 $record$ 表在符号语义中的扩展,其中,$error$ 属性中的缺失值为不可填充的,用⊤表示;其余缺失的属性值是可填充的,用⊥表示.在二值逻辑中,值域 T={true,false},将可填充缺失值和不可填充值考虑进来后,T 扩展为 T^0={$unknown$,$inconsistent$}.此时,关系 R 可表示为 $R^0:D_1^0\times D_2^0\times\ldots\times D_n^0\to T^0$.

id	pile	money	elect	area	addr*	lon	lat	error
t_1	3018	27.15	22.64	CY	SCHOOL	⊥	⊥	⊤
t_2	⊥	⊥	⊥	⊥	⊥	⊥	⊥	NET-FAIL
t_3	1007	64.66	53.88	SY	⊥	⊥	⊥	⊤
t_4	3018	28.71	23.93	CY	SUBWAY	⊥	⊥	⊤
t_5	1007	⊥	⊥	SY	⊥	⊥	⊥	PILE-FAIL
t_6	1007	23.71	19.76	SY	⊥	⊥	⊥	⊤
t_7	1007	10.51	15.03	⊥	⊥	⊥	⊥	⊤
t_8	3018	24.71	20.6	CY	⊥	⊥	⊥	⊤

方法

Fig.2 Extended charging records in denotational semantic

图 2 符号语义下扩展的充电记录

3.2 查询处理扩展

给定查询 $Q:D_1\times D_2\times\ldots\times D_n\to T$,其对应了选择条件运算,可写为 $Q=F(p_1,p_2,\ldots,p_n)$.每个原始项有如下形式:$p_i=(D\ op\ V)$,其中,D 是属性值,op 是运算符(=,≠,<,≤,>,≥),V 是常数值(数值或字符串).每个原始项 p_i 可根据运算符 op 写成函数形式:$op:D\times V\to T$.当考虑缺失值时,需要扩展运算符 op,即 $op^0:D^0\times V^0\to T^0$.其中 $D^0=D\cup\{\top,\bot\}$,$T^0=T\cup\{unknown,inconsistent\}$.给定运算函数 op^0、属性值 d、常量 v,函数 $op^0(d,v)$的运算规则如下.

1. $op^0(d,v)$=true.若 $d\in D$,$op(d,v)$=true,或 $d=\bot$,对所有 $e\in D$,都有 $op(e,v)$=true.
2. $op^0(d,v)$=false.若 $d\in D$,$op(d,v)$=false,或 $d=\bot$,对所有 $e\in D$,都有 $op(e,v)$=false.
3. $op^0(d,v)=inconsistent$.若 $d=\top$.
4. $op^0(d,v)=unknown$.其他情况.

例 1:查询 Q_2:SELECT*FROM *records* WHERE (*elect*>50) OR (*elect*≤50),Q_2 中只有一个原始项 p_1=(*elect*>50),Q_2 可写成 $p_1\vee\neg p_1$.p_1 可写成 *greater than* 函数,对应的扩展为 $p_1^0=greater^0(elect^0,50)$.对 $record^0$ 中的 t_1,$t_1[elect^0]$=22.64,故 $greater^0(t_1[elect^0],50)$=false.对 t_2 来说,$t_2[elect^0]=\bot$,$t_2[elect^0]$的真实值可能是 *elect* 值域范围内任意值,无法判断 $greater^0(t_2[elect^0],50)$是 true 还是 false,因此,$greater^0(t_2[elect^0],50)=unknown$.其他元组的判别与之类似.

至此,我们已经介绍了如何判断每个原始项 p_i 的真假,接下来需要判断整个选择条件表达式 $Q=F(p_1,p_2,\ldots,p_n)$的真假.Q 是由各个原始项经过逻辑运算符∧(AND),∨(OR),¬(NOT)构成的,因此需要在符号语义下扩展这 3 个运算符,如图 3 所示.若直接利用图 3 中的真值表判断选择条件 $Q=F(p_1,p_2,\ldots,p_n)$的真假,可能会得到错误的结果,因为这是一个非真值函数系统.

AND^0	true	false	unknown	inconsistent
true	true	false	unknown	inconsistent
false	false	false	false	inconsistent
unknown	unknown	false	unknown	inconsistent
inconsistent	inconsistent	inconsistent	inconsistent	inconsistent

OR^0	true	false	unknown	inconsistent
true	true	true	true	inconsistent
false	true	false	unknown	inconsistent
unknown	true	unknown	unknown	inconsistent
inconsistent	inconsistent	inconsistent	inconsistent	inconsistent

NOT^0	true	false	unknown	inconsistent
	false	true	unknown	inconsistent

Fig.3 Extended AND, OR, NOT operation

图 3 扩展的 AND、OR、NOT 运算

定义 3(真值函数系统). 给定表达式 $F(p_1,p_2,\ldots,p_n)$,真值表 V,若 $V(F(p_1,p_2,\ldots,p_n))=F(V(p_1),V(p_2),\ldots,V(p_n))$成立,则成该系统是真值函数系统.

例 2:仍以 Q_2 为例,$Q_2^0=p_1^0\vee\neg p_1^0$.显然,$p_1^0\vee\neg p_1^0$ 是恒为真的命题.然而,若分别判断每个原始项,再根据图 3 求得整体的运算结果,则可能得到 *unknown* 结果.考虑 t_2,$p_1^0=(t_2[elect^0]>50)=unknown$ 且 $\neg p_1^0=(t_2[elect^0]\leq 50)=unknown$,根据图 3 可知,$p_1^0\vee\neg p_1^0=unknown$,违背了真值函数系统的条件.

为此,本文提出将查询选择条件转化为对应的主析取范式(principal disjunctive normal form,简称 PDNF)和主合取范式(principal conjunctive normal form,简称 PCNF).根据 PDNF 和 PCNF 制定新的查询处理规则,使得新规则下不再有非真值函数系统问题.为了方便阅读,后文中将符号语义下的扩展标志,每个变量上标的 0 去掉.先回顾一下 PDNF 和 PCNF 的定义.

定义 4(最小项). 最小项是原始项 p_i 或$\neg p_i$ 之积,每个最小项中不可能同时包含同一个原始项 p_i 和$\neg p_i$.n 个

原始项共有 2^n 个最小项,第 i 个最小项记为 m_i.

定义 5(析取范式 PDNF). 给定查询 $Q=F(p_1,p_2,\ldots,p_n)$, $PDNF(Q)=\sum_{i=0}^{2^n-1}m_i$.

定义 6(最大项). 最大项是原始项 p_i 或$\neg p_i$之和,每个最大项中不可能同时包含同一个原始项 p_i 和$\neg p_i$.n 个原始项共有 2^n 个最大项,第 i 个最大项记为 M_i.

定义 7(合取范式 PCNF). 给定查询 $Q=F(p_1,p_2,\ldots,p_n)$, $PCNF(Q)=\prod_{0}^{2^n-1}M_i$.

文献[39]中给出了将查询 Q 转换为对应的 $PDNF(Q)$和 $PCNF(Q)$的经典算法,本文直接使用该算法.接下来,给出基于 $PCNF(Q)$和 $PDNF(Q)$的查询处理规则.给定元组,查询 Q,$Q(t)$的取值有 4 种可能.

1. $Q(t)$=*inconsistent*. if $\exists i,1\leqslant i\leqslant n,d_i=\top$.
2. $Q(t)$=false. if $\forall m_j\in PDNF(Q),m_j(t)$=false.
3. $Q(t)$=true. if $\forall M_j\in PCNF(Q),M_j(t)$=true.
4. $Q(t)$=*unknown*.其他情况.

由上述规则可知,当元组中没有缺失值时,查询处理的方法和传统的方法一致.当元组中出现不可填充的缺失值$\top$时,$Q(t)$=*inconsistent*.为了更好地理解上述规则,给出下面这个示例.

例 3:考虑查询 Q_3:SELECT*FROM *records* WHERE ((*elect*>50) AND (*area*=SY)) OR ((*elect*$\leqslant$50) AND (*error*=PILE-FAIL)),有 3 个原始项:p_1:(*elect*>50),p_2:(*area*=SY),p_3:(*error*=PILE-FAIL),因此 $Q_3=(p_1\wedge p_2)\vee(\neg p_1\wedge p_3)$. $PCNF(Q_3)=M_0\wedge M_2\wedge M_4\wedge M_5=(p_1\vee p_2\vee p_3)\wedge(p_1\vee\neg p_2\vee p_3)\wedge(\neg p_1\vee p_2\vee p_3)\wedge(\neg p_1\vee p_2\vee\neg p_3)$,$PDNF(Q_3)=m_1\vee m_3\vee m_6\vee m_7=(p_1\wedge p_2\wedge p_3)\vee(p_1\wedge p_2\wedge\neg p_3)\vee(\neg p_1\wedge p_2\wedge p_3)\vee(\neg p_1\wedge\neg p_2\wedge p_3)$.以 *record* 中的元组 t_5 为例.

- 首先,t_5 的属性值里没有$\top$,因此第 1 条规则不满足.
- 接下来检查第 2 条规则,在 $PDNF(Q_3)$中,$m_1=(p_1\wedge p_2\wedge p_3)$,$p_1(\perp)$=*unknown*,$p_2$(SY)=true,$p_3$(PILE-FAIL)=true,因此 $m_1(t_5)$的结果不是 false,所以继续检查第 3 条规则.
- 在 $PCNF(Q_3)$中,对每个最大项 M_j,至少有 1 项 p_i 为真,因此,$Q_3(t_5)$=true.其他元组的判定规则与之类似.

给定元组 t,若 $Q(t)$=false,则 t 不满足选择条件,可以直接将其过滤掉,不参与下一步聚集计算过程;若 $Q(t)$=true,则将 t 加入到真值集 TR 中;若 $Q(t)$=*unknown*,则无法判定其是否满足选择条件,将其加入到可能集 MR 中;若 $Q(t)$=*inconsistent*,则将其直接过滤掉,因为对不可填充的值进行条件选择没有任何意义.至此,本文给出了不完整数据上扩展的查询处理规则,将关系表中的元组按照查询结果分到两个集合中——真值集 TR 和可能集 MR.接下来,本文给出聚集查询在符号语义下的扩展.

给定扩展的关系表 R:$D_1\times D_2\times\ldots\times D_n\rightarrow T$,聚集查询 AQ:$D_1\times D_2\times\ldots\times D_n\rightarrow T$.假设 AQ 中的聚集属性为 D_1.将真值集 TR 中 D_1 属性值缺失的元组加入到 $TR_\perp$中,其余元组加入到 TR_c 中.同样地,将可能集 MR 中 D_1 属性值缺失的元组加入到集合 $MR_\perp$中,将其余元组加入到 MR_c 中.聚集查询结果 $AQ(R)$=*aggregate*(TR_c),即聚集计算只在真值集 TR 中聚集属性完整的元组上进行.若$\exists t_i\in TR_c$,且 $t_i[D_1]=\top$,则直接从聚集计算中将该元组删除.

4 可靠结果求解算法

我们在前文中定义了可靠结果为 $RA(R)=[AQ(R)+\Delta_{\min},AQ(R)+\Delta_{\max}]$.上一节给出了 $AQ(R)$的求解方法,本节分别给出求解 SUM、COUNT 和 AVG 查询的$\Delta_{\min}$和$\Delta_{\max}$的方法.

4.1 SUM和COUNT查询

首先,考虑一种简单的情况.假设聚集属性 D_1 中没有缺失值,其他属性不做任何假设.根据查询结果偏差Δ的定义可知,Δ只与 MR 集合中的元组有关.当 MR 中的元组对应的真实值都满足查询条件时,Δ达到最大,反之最小.

引理 1. 若$\perp\notin D_1$,则$\Delta_{\max}$=*aggregate*(MR),$\Delta_{\min}$=0.

证明:若 D_1 没有缺失值,则聚集结果 $AQ(R)$=*aggregate*(TR).在一种极端的情况下,MR 中的元组都满足 AQ 的选择条件,此时真实值 $AQ(R^*)$=*aggregate*($TR\cup MR$),偏差$\Delta=AQ(R^*)-AQ(R)$=*aggregate*(MR).在另外一个极端中,MR 中的元组都不满足选择条件,此时,$AQ(R^*)$=*aggregate*(TR),Δ=0. □

接下来,考虑更一般的情况.D_1 中有缺失值,此时,TR 和 MR 集合中的元组在 D_1 属性上均有可能缺失,此时 $AQ(R)=aggregate(TR_c)$,求Δ需要同时考虑 MR 和 $TR_\perp$中的元组.

定理 1. 对 COUNT 查询,$\Delta_{max}=|MR|+|TR_\perp|,\Delta_{min}=|TR_\perp|$.

证明:根据引理 1 可知,若 D_1 中没有缺失值,则$\Delta_{max}=COUNT(MR),\Delta_{min}=0$.当 $TR_\perp\neq\varnothing$时,$\forall t_i\in TR\cup MR$, $COUNT(t_i[D_1])=1$.这是因为无论缺失值对应的真实值是多少,其计数值都为 1.因此,求Δ_{min}和Δ_{max}时,将 $TR_\perp$中所有的元组加入到 COUNT 计算中.$\Delta_{max}=COUNT(MR\cup TR_\perp),\Delta_{min}=COUNT(TR_\perp)$. □

接下来,给出 SUM 查询中Δ的上下界.与 COUNT 查询不同,D_1 属性中缺失值的真实值未知,从而 SUM 值未知.若 D_1 的值域$[\min(D_1),\max(D_1)]$已知,则可以根据如下定理求Δ_{min}和Δ_{max}.

定理 2. 对 SUM 查询,$\Delta_{max}=SUM(MR\cup TR_\perp)$,其中,对 D_1 属性的所有缺失值,$SUM(\perp)=\max(D_1),\Delta_{min}=SUM(TR_\perp)$,其中,$SUM(\perp)=\min(D_1)$.

证明:SUM 查询的Δ_{min}和Δ_{max}的求解和 COUNT 查询类似,只是⊥的真实值未知,将⊥的真实值记为$\perp^*$.$\Delta_{max}=SUM(MR\cup TR_\perp),\Delta_{min}=SUM(TR_\perp)$.其中,$\Delta_{max}$可以写成 $SUM(MR_c)+SUM(MR_\perp\cup TR_\perp)$,$SUM(MR_c)$是常量:

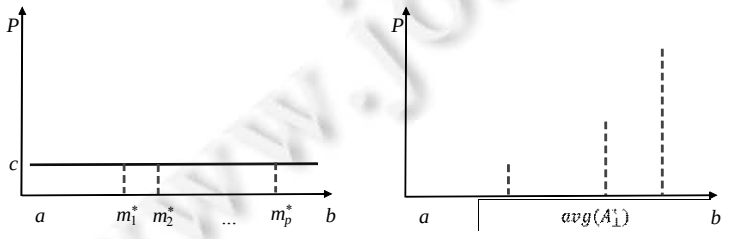
$$SUM(MR_\perp\cup TR_\perp)=\sum\perp^*.$$

$\perp^*$可以是 D_1 值域范围内任意值,当$\perp^*=\max(D_1)$时,Δ_{max}最大;同样地,当$\perp^*=\min(D_1)$时,Δ_{min}最小. □

接下来给出在聚集属性值域未知的情况下,估计 SUM 查询Δ_{min}和Δ_{max}的方法.本文详细阐述Δ_{max}的估计方法,Δ_{min}的估计方法与之类似.在 SUM 查询中,偏差$\Delta=SUM(MR_c)+SUM(MR_\perp\cup TR_\perp)$.由于 SUM 的计算仅与聚集属性取值有关,因此其他非聚集属性可以忽略.将 TR 和 MR 集合中的元组的聚集属性值加入到集合 A 中,A 中缺失值部分记为 $A_\perp$,完整部分记为 A_c,$A_\perp$对应的真实值集合为 $A_\perp^*$,$A_\perp^*$ 的平均值记为 $avg(A_\perp^*)$,SUM 查询的偏差可以改写为 $\Delta=SUM(MR_c)+SUM(A_\perp^*)$.

$SUM(MR_c)$是常量,因此只需要估计 $SUM(A_\perp^*)$ 的上下界,进而可得Δ的上下界. $SUM(A_\perp^*)$ 实际上是 $TR\cup MR$ 集合中聚集属性缺失的元组的真实 SUM 值.假设$|A_\perp|=p$,将$|A_\perp|$中的值 $t_i[D_1]$记为 m_i,则 $A_\perp=\{m_1,m_2,\ldots,m_p\}$.缺失值 m_i 的真实值记为 m_i^*,则 $A_\perp$对应的真实值集合为 $A_\perp^*=\{m_1^*,m_2^*,...,m_p^*\}$.不失一般性,假设 $A_\perp^*$ 是按升序排列的. m_i^* 缺失的概率记为 $P(m_i^*)$.

通常有 3 种缺失机制导致数据缺失:完全随机缺失(MCAR)、随机缺失(MAR)和不随机缺失(NMAR).在 MCAR 模式中,m_i^* 缺失的概率与自身及其他属性值无关,概率分布如图 4(a)所示.此时,可以将 $A_\perp^*$ 看做聚集属性 D_1^* 上的一次简单随机抽样,每个属性值都以相等的概率被抽到.因此, $avg(A_\perp^*)$ 的期望值等于 D_1^* 的均值 $avg(D_1^*)$.在 NMAR 模式中,m_i^* 缺失的概率与自身取值有关.图 4(b)给出一种可能的概率分布,在该分布中,值越大的越容易缺失,$avg(A_\perp^*)$ 的期值大于 $avg(D_1^*)$,这种情况下,称缺失概率与属性值正相关;反之,则为负相关.第 3 种缺失值模式MAR介于MCAR和NMAR之间,m_i^* 缺失的概率与完整的属性值有关,与自身无关.此时,$avg(A_\perp^*)$ 的值小于图 4(b)中的 $avg(A_\perp^*)$.总的来说,在 NMAR 模式中,缺失概率与属性值正相关时,$avg(A_\perp^*)$ 的值最大.本文将利用 $avg(A_\perp^*)$ 估计 $SUM(A_\perp^*)$.



(a) MCAR 模式 (b) NMAR 模式

Fig.4 Probability distribution of missing values

图 4 缺失值概率分布

引理 2. 若缺失概率与属性值正相关,则 D_1 属性值中至少存在 1 个大于 $avg(A_\perp^*)$ 的值是完整的,概率为

$$1-P(\max(D_1))^{p-k+1}.$$

证明:将 m_i^* 中大于 $avg(A_\perp^*)$ 的放到集合 S_1 中,设 $S_1=\{m_k^*,m_{k+1}^*,...,m_p^*\},2\leqslant k\leqslant p$.同时,将 D_1 中大于 $avg(A_\perp^*)$ 的放到集合 S_2 中,即 S_2 包含了所有($avg(A_\perp^*)$,$\max(D_1)$]之间的值.S_2 中所有值都缺失的概率为 $P(S_2)=\prod P(v_i)$, 其中,$v_i\in S_2$.由于 $v_i\leqslant\max(D_1)$,根据缺失概率与值正相关可知,$P(v_i)\leqslant P(\max(D_1))$.由此, $P(S_2)\leqslant P(\max(D_1))^{|S_2|}$,S_2 中至少有 1 个值完整的概率为 $1-P(S_2)\geqslant 1-P(\max(D_1))^{|S_2|}$.又由于 $S_1\subseteq S_2$,因此,

$$|S_2|\geqslant p-k+1, 1-P(S_2)\geqslant 1-P(\max(D_1))^{p-k+1}.$$ □

由引理 2 可知,D_1 中非缺失的最大值有很大概率大于 $avg(A_\perp^*)$.将 D_1 中非缺失的部分记为 D_1^c ,其中最大的值为 $\max(D_1^c)$.

定理 3. $\Delta_{\max}<SUM(MR_c)+\max(D_1^c)\cdot p$, 概率为 $1-P(\max(D_1))^{p-k+1}$.

证明:SUM 查询的偏差 $\Delta=SUM(MR_c)+SUM(A_\perp^*)$,其中,$SUM(A_\perp^*)=avg(A_\perp^*)\cdot p$.根据引理 2 可知,$\max(D_1^c)>avg(A_\perp^*)$ 的概率为 $1-P(\max(D_1))^{p-k+1}$,因此, $SUM(A_\perp^*)<\max(D_1^c)\cdot p$,$\Delta_{\max}<SUM(MR_c)+\max(D_1^c)\cdot p$. □

接下来给出 SUM 查询$\Delta_{\min}$ 的估计,证明与$\Delta_{\max}$ 类似,不再赘述.

定理 4. $\Delta_{\min}\geqslant\min(D_1^c)\cdot|TR_\perp|$,概率为 $1-P(\max(D_1))^{k-1}$,k 是 $TR_\perp^*$ 中比 $avg(TR_\perp)$大的第 1 个值的下标,$TR_\perp^*$ 升序排列.

本文给出的上下界在实际应用中非常可靠,以 SUM 查询的上界为例,概率值 $1-P(\max(D_1))^{p-k+1}$ 在真实数据集中非常接近于 1.平均来说,$p-k+1=p/2$,设聚集属性值集合大小为 1MB,即$|D_1|$=1MB,其中只有 0.1%的缺失值,则 $p/2$=524.设缺失的概率 $P(\max(D_1))$=0.99,则 $1-P(\max(D_1))^{p-k+1}\approx$0.99;并且,这个概率值会随着聚集属性缺失个数的上升而增大,随 $P(\max(D_1))$减小而增大.在实际应用中,数据集的大小往往远大于 1MB,且 $P(\max(D_1))$远远小于 0.99,因而概率值更接近于 1.

SUM 和 COUNT 可靠结果的求解算法如算法 1 所示.给定聚集查询 SUM 或 COUNT,不完整关系表 R,该算法返回可靠查询结果 $RA(R)$.假设 $R=\{D_1,D_2,\ldots,D_n\}$中有 m 条元组,R 的大小记为 N,则 $N=m\cdot n$,可靠结果可以在 $O(N)$时间内求得.首先,将查询 AQ 转换为 $PDNF(AQ)$和 $PCNF(AQ)$的时间是 $O(q^2)$的,其中,q 为查询中选择条件中的属性个数,通常很小,因此这部分时间可以忽略不计.接下来,算法对数据进行一遍扫描,构建真值集和可能集,并在其上计算 $RA(R)$,需要的时间为 $O(N)$.另外,算法 1 在运行过程中需要临时占用的存储空间与 R 的大小无关,空间复杂度为 $O(1)$.

算法 1. SUM 和 COUNT 查询可靠结果求解算法.

输入:聚集查询 AQ, 不完整关系表 R.

输出: 可靠结果 $RA(R)$.

1. TR,MR 初始化为空集
2. 将 AQ 转为对应的 $PDNF(AQ)$和 $PCNF(AQ)$
3. **for** R 中每个 t_i **do**
4. **if** $\forall M_j\in PCNF(Q), M_j(t)$=true **then**
5. $TR=TR\cup\{t_i\}$
6. **else**
7. $MR=MR\cup\{t_i\}$
8. **if** AQ 是 COUNT 查询 **then**
9. $RA(R)=[|TR|,|TR|+|MR|]$
10. **else**
11. $RA(R)=[SUM(TR_c)+\min(D_1^c)\cdot|TR_\perp|, SUM(TR_c)+SUM(MR_c)+\max(D_1^c)\cdot p]$

4.2 AVG查询

本节给出AVG查询的可靠结果求解方法,和SUM/COUNT查询可靠结果的求解方法有所不同.我们从一个简答的情况开始说明,假设聚集属性 D_1 中没有缺失值,其他属性不做任何假设.在 SUM 和 COUNT 查询中,根据引理 1 可知,$AQ(R^*)$的最大值为 $aggregate(TR\cup MR)$,最小值为 $aggregate(TR)$.对 AVG 查询求可靠结果,不能简单地将 MR 中的元组都加入到聚集计算中,或都不加入到聚集计算中.实际上,为了得到 $AVG(R^*)$的最大值,应该将 MR 中所有大于当前 TR 平均值的元组加入到 TR 中.同理,为了求最小的 $AVG(R^*)$值,应将 MR 中所有小于当前 TR 均值的元组加入到 TR 中.本文先给出求最大 $AVG(R^*)$的方法.将 MR 中所有大于当前 TR 平均值的元组加入到 TR 中有两种添加顺序:一种是从小到大,另一种从大到小.我们分别给出这两种添加顺序得到的平均值,并比较两者大小.假设 $MR=\{t_1,t_2,\ldots,t_{|MR|}\}$,$MR$ 按聚集属性 D_1 的值按升序排列,t_k 是第 1 个比 $AVG(TR)$大的值,即 $\forall i\geqslant k$, $t_i\geqslant AVG(TR)$.

引理 3. 将 MR 中的元组按照从 t_k 到 $t_{|MR|}$的顺序加入到 TR 中,新的 TR 集合记为 TR',则 $AVG(TR')$会逐渐增大,直至最后一个元组加入,此时, $AVG_1=\left(SUM(TR)+\sum_{i=k}^{m}t_i\right)\Big/(|TR|+|MR|-k+1)$.

证明:初始时,$AVG(TR)=SUM(TR)/|TR|$,t_k 加入到 TR 后,$TR'=TR\cup\{t_k\}$,此时,TR'的平均值为

$$AVG(TR')=(SUM(TR)+t_k)/|TR|+1.$$

由于 $t_k\geqslant AVG(TR)$,因此 $AVG(TR')\geqslant AVG(TR)$.同理,将剩余元组加入时 $AVG(TR')$会逐渐增大,最后一个元组加入后, $AVG_1=\left(SUM(TR)+\sum_{i=k}^{m}t_i\right)\Big/(|TR|+|MR|-k+1)$. □

若按从大到小的顺序将 MR 中的元组添加到 TR 中,可能会遇到一个边界值,加入该值后,平均值反而下降.

定义 8(边界值). 在将 MR 中的值加入到 TR 的过程中,边界值是比当前 $AVG(TR')$小的第 1 个值,或比 $avg(TR')$大的第 1 个值,记为 M^*.

引理 4. 若将 MR 中的元组按照从 $t_{|MR|}$到 t_k 的顺序加入到 TR 中,则 $AVG(TR')$会逐渐增大,直到遇到边界值 $M^*=t_x,k\leqslant x<|MR|,t_x<AVG(TR')$,此时达到最大值 $AVG_2=\left(SUM(TR)+\sum_{i=x+1}^{|MR|}t_i\right)\Big/(|TR|+|MR|-x)$.

证明:与引理 3 的证明相似,若将 t_x 加入到 TR'中,则由于 $t_x<AVG(TR')$,$avg(TR')$会减小. □

引理 3 给出了按由小到大的添加顺序得到的 AVG 最大值 AVG_1,引理 4 给出了由大到小的顺序下最大值 AVG_2.接下来的定理证明 AVG_2 是 $AVG(R^*)$的上界.

定理 5. $AVG_2\geqslant AVG_1$,$AVG(R^*)$的最大值是 AVG_2.

证明:若按从大到小的添加过程中没有遇到边界值,即将 $t_{|MR|}$到 t_k 都加入到 TR 中,则 $AVG_1=AVG_2$;否则,

$$AVG_2-AVG_1>\left(t_x(x-k+1)-\sum_{i=k}^{x}t_i\right)\Big/(|TR|+|MR|-k+1).$$

又由于 $\forall i\in[k,x),t_x>t_i,(x-k+1)\cdot t_x>\sum_{i=k}^{x}t_i$, 因此 $AVG_2-AVG_1>0$. □

$AVG(R^*)$的最小值可以通过类似的方法得到,将 MR 中所有小于 $AVG(TR)$的值按照从小到大的顺序加入到 TR 中,即从 t_1 到 t_{k1}.若遇到边界值大于当前的平均值 $AVG(TR')$,则停止添加,此时得到的 $AVG(TR')$就是最小的 $AVG(R^*)$.证明过程与最大 $AVG(R^*)$类似,不再赘述.

定理 6. 若将 MR 中的元组按从 t_1 到 t_{k1} 的顺序加入到 TR 中,则 $AVG(TR')$会一直增大,直到遇到边界值 $M^*=t_w,1<w\leqslant k-1,t_w>AVG(TR')$,此时达到 $AVG(R^*)$的最小值 $\left(SUM(TR)+\sum_{i=1}^{i=w-1}t_i\right)\Big/(|TR|+w-1)$.

当 D_1 中有缺失值时,扩展的聚集查询结果为 $AVG(R)=AVG(TR_c)$.为了求 $AVG(R^*)$的最大值,所有 $TR_\perp$中的值应该取 D_1 的最大值,即 $\max(D_1)$.此外,还需要考虑 MR 中哪些元组应该加入到 AVG 计算中,使得 $AVG(R^*)$最大.根据引理 3 和引理 4 可得到如下定理.

定理 7. 当 D_1 中有缺失值时,将 TR 和 MR 集合中聚集属性缺失的值替换为 $\max(D_1)$,将新得到的 MR 集合按照从大到小的顺序加入到 TR 中,直到遇到小于当前 TR'平均值的边界值为止,此时得到 $AVG(R^*)$最大值;反之,将 TR 和 MR 集合中聚集属性缺失的值替换为 $\min(D_1)$,将新得到的 MR 集合按照从小到大的顺序加入到 TR 中,直到遇到大于当前 TR'平均值的边界值为止,此时得到 $AVG(R^*)$最小值.

接下来,我们给出当聚集属性值域未知时求解 AVG 可靠结果的估计方法.与 SUM 查询类似,用 D_1 中非缺失属性值的最小值 $\min(D_1^c)$ 和最大值 $\max(D_1^c)$ 分别替定理 7 中的 $\min(D_1)$和 $\max(D_1)$.下面证明通过这种方法可以很大概率得到 $AVG(R^*)$的估计.

定理 8. 若将 *TR* 和 *MR* 中的缺失值替换为 $\max(D_1^c)$ 并且将 *MR* 中的值按照从大到小的顺序加入到 *TR* 中,直至遇到比当前 $avg(TR')$小的边界值 M^*,此时得到的平均值有 $1-P(\max(D_1))^{|S|-k+1}$ 的概率是最大的 $AVG(R^*)$;若将 *TR* 和 *MR* 中的缺失值替换为 $\min(D_1^c)$ 并且将 *MR* 中的值按照从小大的顺序加入到 *TR* 中,直至遇到比当前 $avg(TR')$大的边界值 M^*为止,此时得到的平均值有 $1-P(\max(D_1))^{k+1}$ 的概率是最小的 $AVG(R^*)$.

证明:令 *S* 集合包含 *TR* 和 *MR* 中大于 M^*的所有值.根据引理 4 可知,最大的均值为$(SUM(TR_c)+SUM(S))/(|TR_c|+|S|)$.假设将 *S* 按照升序排列,第 1 个比 $AVG(S)$大的值得标为 *k*,根据引理 2 可知,D_1 中至少存在 1 个大于 $AVG(S)$的值,概率为 $1-P(\max(D_1))^{|S|-k+1}$.最小 $AVG(R^*)$的证明与之类似. □

AVG 的可靠结果求解算法如算法 2 所示.给定聚集查询 *AQ*,不完整关系表 *R*,该算法返回可靠结果 *RA*(*R*).构建 *TR* 和 *MR* 时间复杂度为 $O(N)$.采用堆排序 *MA* 需要 $O(|MA|\log|MA|)$的时间,遍历 *MR* 需要 $O(|MR|)$的时间,$|MR|\leqslant m$.算法总的时间开销为 $O(N+m\log m)$.另外,算法第 2 行构建 *TR* 和 *MR* 的空间复杂度为 $O(1)$,算法第 4 行堆排序 MA 的空间复杂度为 $O(1)$,算法第 5 行~第 14 行与 *R* 的大小无关,空间复杂度为 $O(1)$.因此,总的空间复杂度为 $O(1)$.

算法 2. AVG 查询可靠结果求解算法.

输入:聚集查询 *AQ*,不完整关系表 *R*.

输出:可靠结果 *RA*(*R*).

1. $i=|MR|, j=1$
2. 和算法 1 一样构建 *TR* 和 *MR*
3. 将 *TR* 和 *MR* 中的缺失值替换为 $\max(D_1^c)$
4. 升序排列 *MR*,$MR=\{t_1,t_2,\ldots,t_{|MR|}\}$
5. **while** $t_i>avg(TR')$ **do**
6. $TR'=TR\cup\{t_i\}$
7. i--
8. **end**
9. 将 *TR* 和 *MR* 中的缺失值替换为 $\max(D_1^c)$
10. **while** $t_i>AVG(TR')$ **do**
11. $TR'=TR\cup\{t_j\}$
12. j++
13. **end**
14. **return** $RA(R)=\left[\left(SUM(TR)+\sum_1^{j-1}t_j\right)\Big/(|TR|+j-1),\left(SUM(TR)+\sum_{i+1}^{MR}t_i\right)\Big/(|TR|+|MR|-i)\right]$

5 实验分析

本文在真实数据集和模拟数据集上设计若干实验,验证所提算法的有效性及查询效率.具体来说,设计如下实验.

(1) 将本文算法与已有的 3 种缺失值填充方法进行准确率比较:可靠结果记为 *RA*(*R*),扩展的聚集查询结果记为 *AQ*(*R*),真实值记为 $AQ(R^*)$,贝叶斯填充算法填充后得到的查询结果记为 Bayesian,均值填充算法得到的查询结果记为 Mean,Hot-dect 填充算法得到的结果记为 Hot-dect.

(2) 将本文算法与 3 种缺失值填充算法的运行时间进行比较,测试算法的时间效率.

5.1 实验设置

数据集:采用真实数据集和合成数据集进行实验.

- Reviews 是亚马逊网站上真实电影评价数据集,有来自 889 176 用户的 7 911 684 条评价数据.创建 *reviews* 表,包含 4 个属性:*user*,*movie*,*rating* 和 *time*.
- TPC-H 数据集,其中,*lineitem* 表包含 6 001 199 条记录;*lineitem* 表模拟工业订单,包含 12 个属性——*quantity*,*returnflag*,*linestatus* 等.

令 r 表示数据集的总体缺失率,$r\in(0,1)$.将 *reviews* 表中的元组随机删除属性值,注入 $100r\%$的可填充的缺失值.将 *lineitem* 表中的 *linestatus* 属性标记为不可填充的缺失属性,注入 $10r\%$不可填充的缺失值.另外,对剩余其他属性值随机注入 $90r\%$的可填充缺失值.假设有两种缺失机制——MAR 和 NMAR:对 MAR,将 *reviews* 和 *lineitem* 表中的元组随机删除属性值;对 NMAR,将 *reviews* 表中 *rating* 属性值和 *lineitem* 表中 *quality* 属性值按照取值大小删除,值越大,删除的概率就越大.本文测试以下 3 个聚集查询.

- Q1: SELECT *COUNT*(*) FROM *reviews* WHERE (*rating*>3) OR (*rating*≤3);
- Q2: SELECT *AVG*(*rating*) FROM *reviews* WHERE *movie*=m_1;
- Q3: SELECT *COUNT*(*quantity*) FROM *lineitem* WHERE *quantity*>20.

Q1 查询 *reviews* 中电影评分大于 3 分的或者小于等于 3 分的评价数量,Q2 查询电影 m_1 的平均打分,Q3 查询 *lineitem* 表总量大于 20 的订单数量.其中,Q2,Q3 的聚集属性 *rating* 和 *quantity* 的值域都未知.

5.2 准确性分析

首先测试可靠结果 *RA*(*R*)对恒真选择条件的准确性.在 *reviews* 表上执行 Q1 查询,在 MAR 模式中,控制 r 从 0.1 变化到 0.5,实验结果如图 5 所示.从实验结果可以看出,无论缺失率 r 为多少,可靠结果 *RA*(*R*)和贝叶斯方法 Bayesian 总能得到真实值,而扩展的聚集查询结果 *AQ*(*R*)随着 r 的增大而呈线性减小趋势.这是由于我们在 *rating* 属性上随机注入缺失值,*AQ*(*R*)对缺失值的处理是直接舍弃,即 *COUNT* 值为 0,因此随着缺失值的增多,*COUNT* 值减少.而本文提出的可靠结果 *RA*(*R*)通过将查询选择语句转换为主析取范式和主合取范式,保证了恒为真的选择条件的评价结果一定为真.另外,贝叶斯填充方法无论将确实的 *rating* 值填充为多少,都符合选择条件,故也能得到真实值.

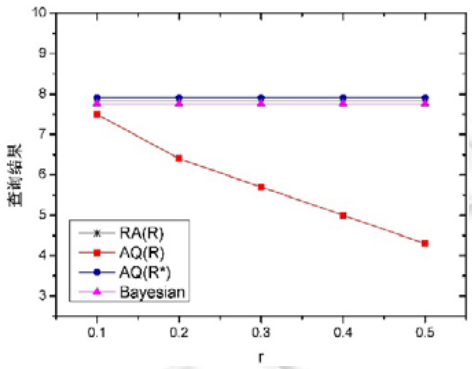
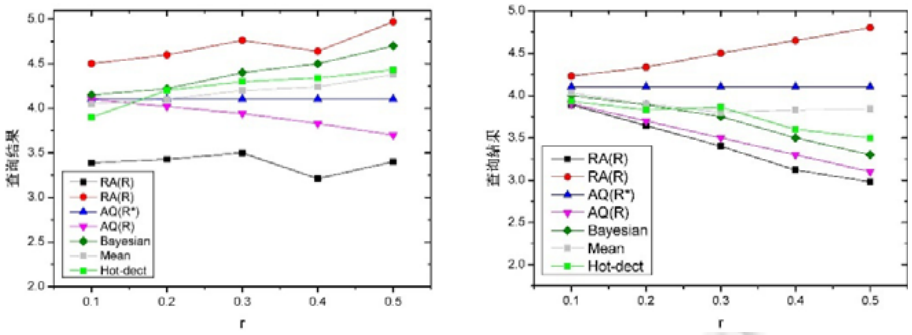


Fig.5 Query result of Q1

图 5 Q1 查询结果

接下来测试总体缺失率 r 对查询结果准确性的影响.在 *reviews* 表上执行 Q2 查询,在 MAR 模式和 NMAR 模式下,分别控制 r 从 0.1 变化到 0.5,实验结果如图 6(a)和图 6(b)所示.从实验结果来看,总体上真实值在所有情况下都在可靠结果 *RA*(*R*)的区间内,而扩展的聚集查询结果 *AQ*(*R*)和 3 种缺失值填充方法随着缺失率 r 的增大,越来越偏离真实值.



(a) MAR 模式下,Q2 的查询结果 (b) NMAR 模式下,Q2 的查询结果

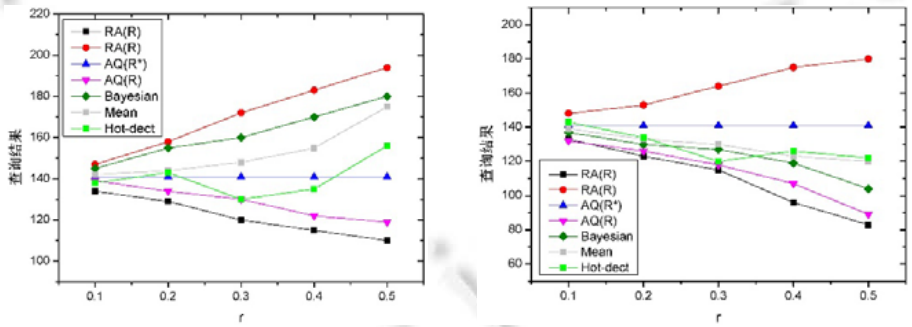
Fig.6 Impact of total missing ratio on the query result of Q2

图 6 总体缺失率对 Q2 查询结果的影响

接下来对每个情况分别进行分析.在图 6(a)中,查询为 Q2,Q2 查询电影的平均得分,缺失模式为 MAR.扩展的聚集查询结果 *AQ*(*R*)小于真实值,且随着 *r* 的增大呈线性衰减趋势.这是由于在 *reviews* 表中,*rating* 值频率最高的是 4 分和 5 分,当缺失比例增大时,整体的平均值就会降低.而 Bayesian 的结果总是大于真实值,这是由于在贝叶斯填充算法中,会利用概率最大的值来填充缺失值,在 *reviews* 表中概率最大的是 4 分和 5 分,因此所有小于 4 分的值一旦缺失,就会被填充为 4 分或 5 分,导致平均打分偏高.同样,均值填充算法(mean)将缺失的属性值填充为当前完整属性值的平均值,得到的查询结果也大于真实值.Hot-dect 方法利用最相近的元组的属性值填充缺失值,当 *rating* 缺失比例增大时,该方法得到的查询结果也将大于真实值.

在图 6(b)中,查询为 Q2,缺失模式为 NMAR.扩展的聚集查询结果 *AQ*(*R*)随着 *r* 的增大逐渐减小.这是由于注入缺失值时,*rating* 值大的缺失的概率大,因此,整体的电影评分均值下降.在这种情况下,Bayesian 的结果也小于真实值.在贝叶斯填充算法中,会用出现频率较高的值来填充缺失值,现在 *reviews* 表中 *rating* 出现频率高的都是小于 4 分的,因此,填充后的平均分低于原始平均分.Mean 和 Hot-dect 的情况与 Bayesian 类似,不再赘述.

在 lineitem 上执行 Q3 查询,同样在 MAR 模式和 NMAR 模式下,分别控制 *r* 从 0.1 变化到 0.5,实验结果如图 7(a)和图 7(b)所示.



(a) MAR 模式下,Q3 的查询结果 (b) NMAR 模式下,Q3 的查询结果

Fig.7 Impact of total missing ratio on the query result of Q3

图 7 总体缺失率对 Q3 查询结果的影响

在图 7(a)中,查询为 Q3,Q3 查询 *lineitem* 表总量大于 20 的订单数量,缺失模式为 MAR.扩展的聚集查询结果 *AQ*(*R*)随着 *r* 的增大逐渐减小,这是由于在 *reviews* 表中,*quantity* 取值中大于 20 的较多,当缺失比例增大时,*quantity* 大于 20 的订单数量会降低.而 Bayesian 的结果总是大于真实值,与 *reviews* 表中的情况类似,贝叶斯填

充算法会将缺失值填充为出现频率较大的值,即大于 20 的值,因此满足查询条件的订单数增多.与 Bayesian 类似,Mean 方法得到的结果大于真实值,而 Hot-dect 填充方法得到的结果没有规律,可能大于真实值,也可能小于真实值.

在图 7(b)中,查询为 Q3,缺失模式为 NMAR.扩展的聚集查询结果 $AQ(R)$随着 r 的增大逐渐减小.这是由于注入缺失值时,*quantity* 值大的缺失的概率大,因此,满足选择条件的订单随着缺失率的增大而减少.Bayesian 的结果也随着 r 的增大而减少.由于 *quantity* 值大的容易缺失,未缺失的 *quantity* 值较小的居多,因此,在填充后的 *quantity* 大于 20 的订单总量减小.Mean 和 Hot-dect 的结果分析与之类似,不再赘述.

最后,测试可靠结果方法对查询选择性的鲁棒性,即对不同选择率的查询,真实结果是否都在可靠结果的估计范围内.设置总体缺失率 r=0.1,从众多电影中选择 3 个有代表性的电影:频率最大的(m_1)、频率最小的(m_2)以及频率位于中位数的(m_3).缺失模式为 MAR,在 *reviews* 表上执行 Q2,选择条件分别为 $movie=m_1$,$movie=m_2$,$movie=m_3$.3 个电影的查询结果如图 8 所示.从实验结果上可以看出,对 3 种选择率差异较大的电影,可靠结果方法都能覆盖住真实值.

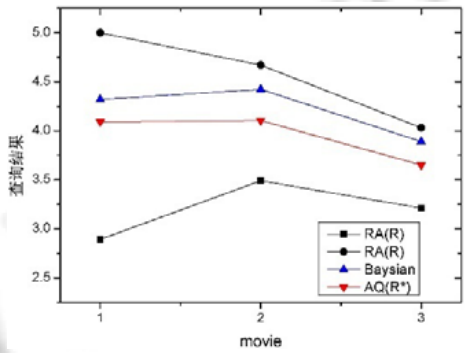


Fig.8 Effect of selectivity on query results

图 8 查询选择性对查询结果的影响

5.3 可扩展性分析

通过准确性比较可以看出,可靠结果在所有情况下都能保证覆盖真实值,而缺失值填充方法填充后的查询结果往往与真实值相差较远.接下来对比可靠结果与 3 种缺失值填充算法的效率.在 *lineitem* 表上执行查询 Q3,数据大小为 100MB~1GB,实验结果如图 9 所示.

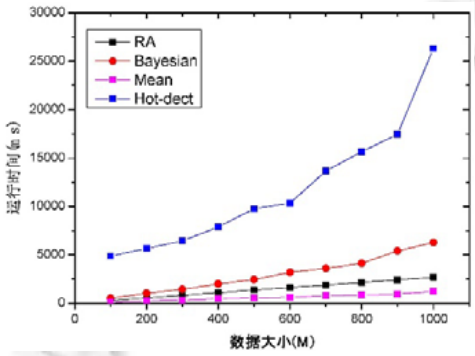


Fig.9 Comparison of runtime

图 9 运行时间比较

可以看出,随着数据量的增大,可靠结果的运行时间呈线性增长.在数据量为 900M 时,贝叶斯方法的运行时

间是可靠结果的 2 倍;随着数据量的增大,两者的差距也越来越大;Hot-dect 的运行时间最慢,均值填充算法的运行时间最快.

6 结 论

本文在符号语义中扩展了传统的关系数据库模型,提出了一种可以适用于不同缺失值的通用不完整数据库模型.在通用不完整数据库模型中,本文提出一种新的不完整数据上聚集查询结果语义:可靠结果.可靠结果与确定结果不同,它试图给出聚集查询真实结果的区间估计.本文给出求解可靠结果的线性时间算法,通过理论分析和扩展实验,验证了所提算法的有效性.

References:

[1] Gong XQ, Jin CQ, Wang XL, Zhang R, Zhou AY. Data-intensive science and engineering: requirements and challenges. Chinese Journal of Computers, 2012,35(8):1−16 (in Chinese with English abstract).

[2] Li JZ, Liu XM. An important aspect of big data: Data usability. Journal of Computer Research and Development, 2013,50(6): 1147−1162 (in Chinese with English abstract).

[3] Tian J, Yu B, Yu D, *et al*. Missing data analyses: A hybrid multiple imputation algorithm using gray system theory and entropy based on clustering. Application Intelligence, 2014,40(2):376−388.

[4] Zhang S. Shell-neighbor method and its application in missing data imputation. Application Intelligence, 2011,35(1):123−133.

[5] Zhang S, Jin Z, Zhu X. Missing data imputation by utilizing information within incomplete instances. Journal of Systems and Software, 2011,84(3):452−459.

[6] Zhu X, Zhang S, Jin Z, *et al*. Missing value estimation for mixed-attribute data sets. IEEE Trans. on Knowledge Data Engineering, 2011,23(1):110−121.

[7] Song S, Zhang A, Chen L, *et al*. Enriching data imputation with extensive similarity neighbors. Proc. of the VLDB Endowment, 2015,8(11):1286−1297.

[8] Wu S, Feng X, Han Y, *et al*. Missing categorical data imputation approach based on similarity. In: Proc. of the IEEE Int'l Conf. on Systems. 2012. 2827−2832.

[9] Li Z, Qin L, Cheng H, *et al*. TRIP: An interactive retrieving-inferring data imputation approach. In: Proc. of the IEEE Int'l Conf. on Data Engineering. 2016. 1462−1463.

[10] Ye C, Wang H, Li J, *et al*. Crowdsourcing-enhanced missing values imputation based on bayesian network. In: Proc. of the Database Systems for Advanced Applications. 2016. 67−81.

[11] Beskales G, Ilyas IF, Golab L, *et al*. Sampling from repairs of conditional functional dependency violations. Proc. of the VLDB Endowment, 2014,23(1):103−128.

[12] Rahm E, Do HH. Data cleaning: Problems and current approaches. IEEE Data Engineering Bulletin, 2000,23(4):3−13.

[13] Feelders AJ. Handling missing data in trees: Surrogate splits or statistical imputation. In: Proc. of the Principles of Data Mining and Knowledge Discovery. 1999. 329−334.

[14] Liu H, Zhang S. Noisy data elimination using mutual *k*-nearest neighbor for classification mining. Journal of Systems and Software, 2012,85(5):1067−1074.

[15] van Buuren S, van Mulligen EV, Brand JPL. Routine multiple imputation in statistical databases. In: Proc. of the Int'l Working Conf. on Scientific and Statistical Database Management. 1994. 74−78.

[16] Booth DE. Analysis of incomplete multivariate data. Technometrics, 2000,42(2):213−214.

[17] Lakshminarayan K, Harp SA, Samad T. Imputation of missing data in industrial databases. Application Intelligence, 1999,11(3): 259−275.

[18] Cios KJ, Pedrycz W, Swiniarski RW. Data mining methods for knowledge discovery. IEEE Trans. on Neural Networks, 1998,9(6): 1533−1534.

[19] Lazar NA. Statistical analysis with missing data. Technometrics, 2003,45(4):364−365.

[20] Quinlan JR. C4.5: Programs for Machine Learning. Morgan Kaufmann Publishers, 1993.

[21] Cios KJ, Kurgan LA. CLIP4: Hybrid inductive machine learning algorithm that generates inequality rules. Information Science, 2004,163(1-3):37−83.

[22] Farhangfar A, Kurgan LA, Pedrycz W. A novel framework for imputation of missing values in databases. IEEE Trans. on Systems Man and Cybernetics, 2007,37(5):692−709.

[23] Abiteboul S, Hull R, Vianu V. Foundations of Databases. Addison-Wesley, 1995.

[24] van Meyden R. Logical approaches to incomplete information: A survey. In: Proc. of the Logics for Databases and Information Systems. 1998. 307−356.

[25] Grahne G. The Problem of Incomplete Information in Relational Databases. Springer-Verlag, 1991.

[26] Imielinski T, Jr Lipski L. Incomplete information in relational databases. Journal of the ACM, 1984,31(4):761−791.

[27] Codd EF. Extending the database relational model to capture more meaning. ACM Trans. on Database System, 1979,4(4):397−434.

[28] Date CJ. Database in Depth Relational Theory for Practitioners. O'Reilly, 2005.

[29] Date CJ. A critique of Claude Rubinson's paper nulls, three—Valued logic, and ambiguity in SQL: Critiquing date's critique. ACM SIGMOD Record, 2008,37(3):2−22.

[30] Date CJ, Darwen H. A Guide to SQL Standard. 4th ed., Addison-Wesley, 1997.

[31] Chung Y, Mortensen ML, Binnig C, *et al*. Estimating the impact of unknown unknowns on aggregate query results. In: Proc. of the 2016 Int'l Conf. on Management of Data. 2016. 861−876.

[32] Wolf G, Khatri H, Chokshi B, *et al*. Query processing over incomplete autonomous databases. In: Proc. of the 33rd Int'l Conf. on Very Large Data Bases. 2007. 651−662.

[33] Raghunathan R, De S, Kambhampati S. Bayesian networks for supporting query processing over incomplete autonomous databases. Journal of Intelligence Information System, 2014,42(3):595−618.

[34] Haghani P, Michel S, Aberer K. Evaluating top-*k* queries over incomplete data streams. In: Proc. of the 18th ACM Conf. on Information and Knowledge Management. 2009. 877−886.

[35] Soliman MA, Ilyas IF, Ben-David S. Supporting ranking queries on uncertain and incomplete data. Proc. of the VLDB Endowment, 2010,19(4):477−501.

[36] Khalefa ME, Mokbel MF, Levandoski JJ. Skyline query processing for incomplete data. In: Proc. of the 24th Int'l Conf. on Data Engineering. 2008. 556−565.

[37] Gao Y, Miao X, Cui H, *et al*. Processing *k*-skyband, constrained skyline, and group-by skyline queries on incomplete data. Expert System Application, 2014,41(10):4959−4974.

[38] Cheng W, Jin X, Sun J, *et al*. Searching dimension incomplete databases. IEEE Trans. on Knowledge on Data Engineering, 2014, 26(3):725−738.

[39] Yannis B. Null values in database management a denotational semantic approach. In: Proc. of the 1979 ACM SIGMOD Int'l Conf. on Management of data. ACM, 1979. 162−169.

附中文参考文献:

[1] 宫学庆,金澈清,王晓玲,张蓉,周傲英.数据密集型科学与工程:需求和挑战.计算机学报,2012,35(8):1−16.

[2] 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.

张安珍(1990－),女,山东临沂人,博士,讲师,主要研究领域为数据质量,弱可用数据计算,查询处理.

李建中(1950－),男,博士,教授,博士生导师,CCF 会士,主要研究领域为数据库技术,并行计算,传感网.

高宏(1966－),女,博士,教授,博士生导师,CCF 杰出会员,主要研究领域为数据库管理,无线传感网,图计算.