换的思路是:在元组 t 的属性 A 的候选属性值的修复代价为负值时,找到最小的修复代价,记为 min$RCost(t,A,f)$,其数学表示为公式(6).

$$\min RCost(t,A,f)=\min\{RCost(t,A,v,v',f)|v'\in Vs(t,A)\} \tag{6}$$

将该候选属性记录中所有候选属性值的修复代价加上最小的修复代价的绝对值,这样就可以保证候选属性值的修复代价概率量化均为 0~1 之间的取值.

如果 A 与 B 相同,那么对于 $Vs(t,A)$中某候选属性值 v'的修复代价是指:为保证 t 满足 f,需要将 I 中所有 t'的 A 属性值由 v 改为 v'的编辑距离值之和,其中,t'是指 X 取值为 $t[X]$的元组,我们使用编辑距离作为修改操作数目.这里未使用更新操作前后冲突元组数目的变化量,是由于修改操作数目量化在某些情况下与直观感觉相符合.如图 1 所示,直观上,t_2 的订单状态(OO)候选属性值{P,F}的概率都应该是 0.5,确实无法分辨.按照修改操作数目度量 t_2 的候选属性值 P 和 F 的修复代价相同,可得与直观相符合的概率值.而使用冲突元组数目变化量则会使得 t_2 的候选属性值 P 和 F 的修复代价不相同,因为若更新为 P,更新前与 f 冲突的元组数目为 2,更新后还是 2,所以冲突变化量为 0.若更新为 F,更新前与 f 冲突的元组数目为 2,更新后变为 0,所以冲突变化量为−2.在经过坐标变化后,得到的概率值是不相同的,与直观不符.

由于 A 属性值的修改会影响到所有与属性 A 有关系的函数依赖集 F_A,所以需要对 F_A 中每个函数依赖计算修复代价,然后累加求和后,使用指数函数 $y=e^{-RCost}$ 转换为[0,1]的概率值.

4 不一致性修复算法

本节设计并实现了一种贪心修复算法,该算法以冲突元组候选属性记录和属性值的构建阶段和候选属性值的概率量化阶段得到的候选属性记录集合 CRs 为输入,输出是满足 F 的数据集 I_R.已有研究[2]证明:计算满足函数依赖集,且修复代价最小的数据修复问题是一个 NP 难问题.假如只考虑基于修复代价的概率值,不考虑基于相关性的概率值,然后使用指数函数的逆运算——对数函数,可将修复代价的概率值变换为修复代价,那么我们的问题就可以转化为文献所述的 NP 难问题.这说明,寻找满足函数依赖集并且概率值较高的可能世界实例也是一个 NP 难问题.这表明了算法的启发式特点.本节的贪心搜索算法是使用贪心技术的启发式算法.本节还证明了算法的可终止性,并分析了算法的复杂度.

该贪心算法一方面避免了计算量惊人的可能世界实例枚举操作,另一方面能够在满足函数依赖集 F 的所有可能世界实例中快速地找到概率值之和较高的实例,也就是修复解决方案.该算法的处理思路是:

首先,将候选属性记录由原来的三元组扩展为六元组,该六元组的形式为$\langle t_i,V_j,Vs,RPs,CPs,Ps\rangle$.其中,$RPs$ 用来存放修复代价的概率量化值,CPs 用来存放基于相关性的概率量化值,Ps 用来存放 RPs 和 CPs 融合后的概率值以表示该候选属性值正确的可能性.这 3 个概率值均与 Vs 建立一一对应关系.接着,依照候选属性记录集合 CRs 中每个候选属性记录$\langle t_i,V_j,Vs,Ps\rangle$将数据集 I 中元组 t_i 的 A_j 属性的属性值设置为 uncertain,表示未确定,此时,数据集记为 I_U.然后,从所有的候选属性值中抽取出满足函数依赖 F 的所有候选属性值,并选择概率值最高的候选属性值,将原来设置为 uncertain 的位置更新为该候选属性值.此时,数据集由原来的 I_U 变为 I'_U .不断迭代这个过程,直到所有的 uncertain 都标记为被更新过.假如该贪心算法可以找到满足函数依赖且概率值较高的可能世界实例,那么 uncertain 标记位的属性值就组成了该可能世界实例.此时,数据集由原来的 I_U 变为 I_R.假如执行若干次迭代后数据集变为了 I'_U ,找不到满足函数依赖 F 的候选属性值,但还存在 uncertain 标记的位置,那么将 uncertain 标记的元组 t 属性值必定与 I'_U 中某个元组 t'一起对某函数依赖 f 构成了冲突,那么将元组 t 的属性值改为与元组 t'相同,使得函数依赖不冲突.当所有的 uncertain 都处理完毕后,该贪心算法就找到了一个满足函数依赖且概率值较高的可能世界实例.

该贪心算法的输入是数据集 I、函数依赖集 F 和冲突元组的候选属性记录集合 CRs,其元素 $cr\in CRs$ 是冲突元组的那些具有候选属性值的属性,具体形式是一个六元组$\langle t_i,V_j,Vs,RPs,CPs,Ps\rangle$.输出满足 F 的数据集 I_R.其伪代码如算法 1 所示.L2~L10 是将候选属性集 CRs 中每个候选属性值的修复代价概率量化 RP 和基于相关性的概率量化 CP 中概率值融合成 P.L8 是将所有的候选属性值添加到 $sortedAllAVs$ 中.L9 将数据集 I 中元组 $cr.tid$

的 *cr.AN* 属性的属性值设置为 uncertain.L11~L28 是贪心搜索满足函数依赖且概率值较高的可能世界实例的过程.L14 是将所有的属性值按照其概率值降序排列.L16 是判断当前概率值最高的属性值 *sortedAllAVs.get*[*j*]是否满足函数依赖:如果满足,那么将 *isFind* 设置为 true,并将该属性值所属的候选属性记录的元组唯一标识、属性标识等信息存放到 *selectAVInfo*,跳出本次循环.L23 是当找到满足函数依赖并且概率值最高的值后,使用 *selectAVInfo* 的信息更新 I_U 为 I'_U,并将候选属性的候选属性值从 *sortedAllAVs* 中移除.L26 就是处理还存在 uncertain 标记的位置,但是未找到满足函数依赖 *F* 的候选属性值情况.L29 是当所有的不确定标记都被覆盖后,得到的数据集 I_U 就是修复方案 I_R.

算法 1. 搜索满足函数依赖集且概率值较高的可能世界实例贪心算法,记为 PWM 算法.

输入:数据集 *I*,函数依赖集 *F*,候选属性记录集合 *CRs*;

输出:满足函数依赖集 *F* 的数据集 I_R.

```
1.  BEGIN
2.    FOR EACH cr∈CRs DO
3.        FOR i←0 TO CRs.Avs.length DO
4.            P←(cr.getRP(i)+cr.getCP(i))*0.5
5.            add P into Ps
6.        END FOR
7.        cr.Ps←Ps
8.        sortedAllAVs.addAll(cr.getVs())
9.        setUncertain(I,cr.tid,cr.an)
10.   END FOR
11.   I_U←I
12.   FOR i←0 TO |CRs| DO
13.       isFind←false
14.       Sort sortedAllAVs by av.p in descending order
15.       FOR j←0 TO |sortedAllAVs| DO
16.           isFind←isConsistent(I_U,F,sortedAllAVs.get[j])
17.           IF (isFind)
18.              selectAVInfo←getCrAndAV(sortedAllAVs.get[j])
19.              BREAK
20.           END IF
21.       END FOR
22.       IF (isFind)
23.          I'_U ← update(I_U,selectAVInfo)
24.          I_U ← I'_U
25.       IF (!isFind)
26.          I'_U ← processExceptionCase(I_U,selectAVInfo)
27.          I_U ← I'_U
28.   END FOR
29.   Return I_R=I_U
30. END
```

定理 1. 给定任意数据集 *I* 和函数依赖集合 *F*,算法 PWM 总能终止,并且返回一个修复方案 I_R,使得 $I_R \models F$.

证明:候选属性元组集的数目为 *N*,也就是数据集 I_U 中不确定性标记的属性值数目,每一步都能处理掉一个

不确定性标记,得到 I'_U,且 I'_U 满足函数依赖集合 F.所以,算法 PWM 是可终止的,并且返回一个修复方案 I_R. □

- 复杂性分析

该算法的关键操作是找到一个能够更新不确定性标记的候选属性值.由伪代码可知,该算法的复杂度是 $O(|CRs|\times|sortedAllAVs|)$.假设数据集 I 中元组的数目是 N,属性名的数目为 M,每个属性名的候选属性值的数目为 L,那么最坏情况下,$|CRs|$的取值为 $N\times M$,而$|sortedAllAVs|$的取值为 $N\times M\times L$,所以最坏情况下,算法复杂度为 $O(N^2\times M^2\times L)$.然而,实际情况没有这么差,因为冲突元组的数目远远小于 N,经过过滤操作后候选属性值的数目 L 会减少很多,而且每当确定一个候选属性值,那么下一次循环中 *sortedAllAVs* 集合中与该属性值具有相同元组标识和属性名的属性值就会被移除,使得 *sortedAllAVs* 中元素数目很快减少.

5 实 验

本节在模拟数据集上评估并描述基于可能世界模型的不一致性修复方法的实验结果.实验运行环境、数据集以及数据质量的评估标准,如类似信息检索研究领域的查全率、查准率和 *F*-measure,会在第 5.1 节详细介绍.第 5.2 节则在算法的有效性方面进行详细分析.

5.1 实验配置

所有实验的运行环境配置为 Intel(R) Core(TM) i7-4710MQ 2.50GHz 处理器,8GB 内存,Windows 8.1 中文版 64 位操作系统.数据集使用 MySQL Workbench 5.2.47 CE 存储,编程语言是 Java.

实验使用的数据集有 TPCH 数据集——它是事务处理性能委员会提供的 TPC-H 基准测试集.其中,TPCH 数据集是模拟数据集.为了便于评价,假设源数据集是正确的,我们采用的插错策略是随机地选择数据集的一个子集,然后以概率 *right*%向某函数依赖 $f:X\rightarrow A$ 的右边属性 A 插入错误,该错误是从属性 A 的值域 $DOM(A)$中选择与 A 的原属性值不同的值 a'替换掉 A 的属性值 a;反之,则以概率 1–*right*%在函数依赖的左边属性名 $B\in X$ 插入错误,最终保证插入的错误率达到 *noi*%.其中,*noi*%是指错误的属性值数目与整个数据集大小的比值,且所有含错误的元组均是冲突元组,即,不满足函数依赖.本实验使用了 22 个函数依赖.

- 评价基准

评价基准扩展信息检索中常用的查准率、查全率和 *F*-measure.查准率是指修复算法正确更新的属性值数目与更新的属性值数目的比值,记为 Precision.查全率是指修复算法更新的属性值数目与数据集中错误的属性值数目的比值,记为 Recall.*F*-measure 由查全率和查准率计算得到,定义如下:

$$F\text{-}measure=2\times\frac{Precision\times Recall}{Precision+Recall}.$$

5.2 实验结果分析

本节在算法的有效性方面详细分析实验效果,其中,本文的方法用标签 PWM 标记,文献[2]中基于修复代价的修复方法用标签 ECR 标记.

图 4 是固定错误率 *noi*%为 0.06 和 *right*%为 0.8,数据集的元组数目从 200 增长到 1 600 时,查全率、查准率和 *F*-measure 的变化情况.

由图 4 可知,该方法在查全率、查准率和 *F*-measure 的度量上均高于基于修复代价的修复方法.

图 5 是固定 *right*%为 0.8,数据集的元组数目为 1 000,错误率 *noi*%从 0.02 变化到 0.1 时,查全率、查准率和 *F*-measure 的变化情况.

由图 5 可知,基于修复代价的修复方法和基于可能世界模型的修复方法随着错误率的增长,修复质量变化不大,但我们的方法在查全率、查准率和 *F*-measure 的度量上依然高于基于修复代价的修复方法.

图 6 是固定 *right*%为 0.8,数据集的元组数目为 1 800,错误率 *noi*%从 0.01 变化到 0.1 时,查全率和查准率的变化情况.

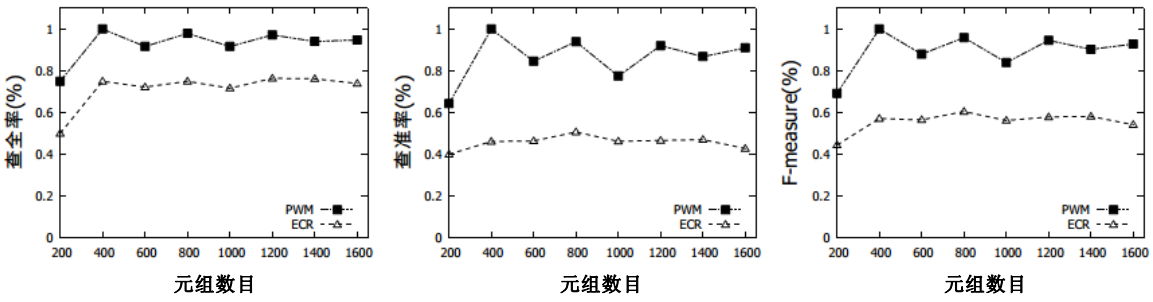


Fig.4 Recall, precision and *F*-measure on tuples number

图 4 数据集的元组数目变化情况下查全率、查准率和 *F*-measure 的对比

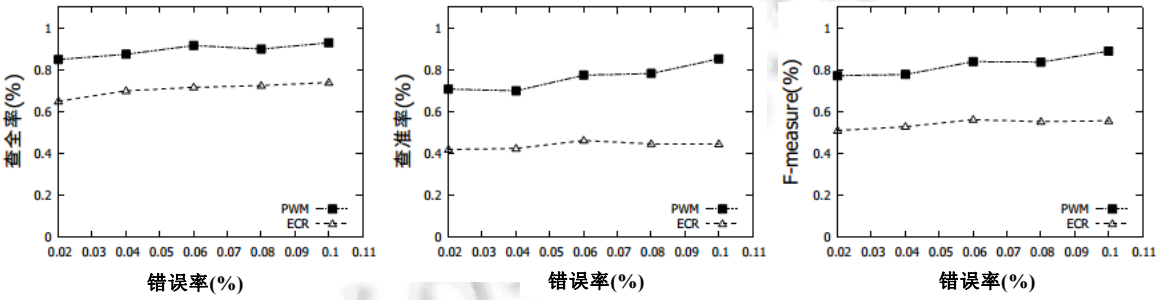


Fig.5 Recall, precision and *F*-measure on error rates

图 5 错误率变化情况下查全率、查准率和 *F*-measure 的对比

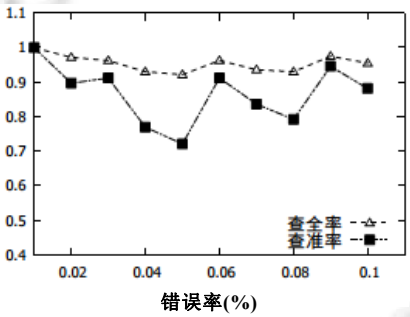


Fig.6 Recall and precision of PWM on error rates

图 6 不同错误率对 PWM 方法的影响

由图 6 可知,查全率随着插入错误的增多,变化幅度较小,而查准率则会有较大的波动,这主要是因为错误出现形式多样,特别是在函数依赖的左边出现错误,且左边的属性值分布比较混乱、程度比较高时,基于相关性的概率和修复代价的概率量化均无法有效分析真值的概率值,使得原有错误未改正确,而新的错误却被引入.

图 7 是对比了在固定 *right*%为 0.8,*noi*%为 0.06,数据集的元组数目从 800 变化到 2 000 的情况下,使用不同概率量化方法在查全率、查准率和 *F-measure* 的变化情况.其中,候选属性值的概率值仅使用基于相关性的概率量化,即 Only Correlation 所标记,仅仅使用修复代价的概率量化,即 Only Repair Cost 所标记,以及同时使用两者的概率量化,即 Both 所标记.

由图 7 可知,在仅使用修复代价的概率下,查准率和 *F*-measure 有比较大的波动,表现不够稳定,而使用基于相关性的概率量化则修复效果比较稳定.

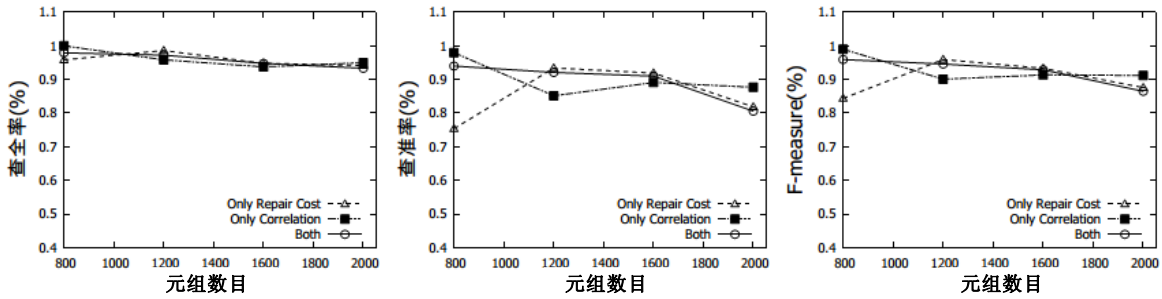


Fig.7 Recall, precision and *F*-measure of PWM on different probability metrics

图 7 不同概率值组合对 PWM 方法的影响

6 总结与展望

本文提出了一种基于可能世界模型的不一致性修复框架,设计并实现了一种融合了修复代价、属性值相关性的概率量化的不一致性修复算法,并在模拟数据上验证了算法的有效性.

在数据修复领域还有很多公开问题,比如:

- 在不一致性修复问题上,比较多的研究是关于文本数据的修复,而数值型数据的相关研究比较少;
- 大多数修复是假定数据依赖是存在的,然后以此为数据依赖提出近似算法,若数据依赖不存在,那么如何进行修复的研究相对较少.

今后,我们将对上述问题进行探索性研究.

References:

[1] Yakout M, Berti-Equille L, Elmagarmid AK. Don't be scared: Use scalable automatic repairing with maximal likelihood and bounded changes. In: Proc. of the ACM SIGMOD Int'l Conf. on Management of Data (SIGMOD 2013). New York: ACM Press, 2013. 553−564. [doi: 10.1145/2463676.2463706]

[2] Bohannon P, Flaster M, Fan WF, Rastogi R. A cost-based model and effective heuristic for repairing constraints by value modification. In: Proc. of the ACM SIGMOD Int'l Conf. on Management of Data. Baltimore: ACM Press, 2005. 143−154. [doi: 10.1145/1066157.1066175]

[3] Kolahi S, Lakshmanan LVS. On approximating optimum repairs for functional dependency violations. In: Proc. of the 12th Int'l Conf. on Database Theory (ICDT 2009). St. Petersburg: ACM Press, 2009. 53−62. [doi: 10.1145/1514894.1514901]

[4] Zhou AY, Jin CQ, Wang GR, Li JZ. A survey on the management of uncertain data. Chinese Journal of Computers, 2009,32(1): 1−16 (in Chinese with English abstract). [doi: 10.3724/SP.J.1016.2009.00001]

[5] Galhardas H, Florescu D, Shasha D, Simon E, Saita CA. Declarative data cleaning: Language, model, and algorithms. In: Apers PMG, Atzeni P, Ceri S, Paraboschi S, Ramamohanarao K, Snodgrass RT, eds. Proc. of the 27th Int'l Conf. on Very Large Data Bases (VLDB 2001). Roma: Morgan Kaufmann Publishers, 2001. 371−380.

[6] Raman V, Hellerstein JM. Potter's wheel: An interactive data cleaning system. In: Apers PMG, Atzeni P, Ceri S, Paraboschi S, Ramamohanarao K, Snodgrass RT, eds. Proc. of the 27th Int'l Conf. on Very Large Data Bases (VLDB 2001). Roma: Morgan Kaufmann Publishers, 2001. 381−390.

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

[8] Lian X, Lin YC, Chen L. Cost-Efficient repair in inconsistent probabilistic databases. In: Proc. of the 20th ACM Conf. on Information and Knowledge Management (CIKM 2011). Glasgow: ACM Press, 2011. 1731−1736. [doi: 10.1145/2063576.2063826]

[9] Mayfield C, Neville J, Prabhakar S. ERACER: A database approach for statistical inference and data cleaning. In: Proc. of the ACM SIGMOD Int'l Conf. on Management of Data (SIGMOD 2010). Indianapolis: ACM Press, 2010. 75−86. [doi: 10.1145/1807167.1807178]

[10] Hu YH, De S, Chen Y, Kambhampati S. Bayesian data cleaning for Web data. arXiv: 1204.3677, 2012.

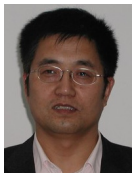
[11] Yakout M, Elmagarmid AK, Neville J, Ouzzani M, Ilyas IF. Guided data repair. PVLDB, 2011,4(5):279−289. [doi: 10.14778/1952376.1952378]

附中文参考文献:

[4] 周傲英,金澈清,王国仁,李建中.不确定性数据管理技术研究综述.计算机学报,2009,32(1):1−16. [doi: 10.3724/SP.J.1016.2009.00001]

徐耀丽(1987－),女,河南安阳人,硕士,CCF 学生会员,主要研究领域为数据质量.

陈群(1976－),男,博士,教授,博士生导师,CCF 高级会员,主要研究领域为大数据管理,物联网信息管理.



李战怀(1961－),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理论与技术.

钟评(1985－),男,硕士,CCF 学生会员,主要研究领域为数据质量.