doi: 10.3969/j.issn.1005-3026.2016.12.003

基于内容相关的条件函数依赖的一致性清洗方法

杜岳峰[1], 申德荣[1], 张 亮[2], 于 戈[1]
(1. 东北大学 信息科学与工程学院, 辽宁 沈阳 110819; 2. 中国人民解放军 65154 部队, 辽宁 凌源 122513)

摘 要: 基于条件函数依赖提出了一种内容相关的条件函数依赖,并给出基于内容相关的条件函数依赖的一致性清洗方法. 通过分析条件函数依赖之间的关系,将相关联的条件函数依赖合并组成内容相关的条件函数依赖. 内容相关的条件函数依赖可以检测多条件值下的数据一致性问题并提供可用于一致性修复的参考值. 同时,提出了一种一致性修复的代价模型. 模型参考内容相关的条件函数依赖对应元组的实际情况进行修复,实现代价最优,同时保证数据一致性. 通过在两组真实数据集上进行试验测试,证明提出的基于内容相关的条件函数依赖的一致性清洗方法能够准确地检测数据的一致性问题并加以修复.

关 键 词: 数据清洗;条件函数依赖;内容相关;数据一致性;修复代价模型
中图分类号: TP 311.13 **文献标志码**: A **文章编号**: 1005-3026(2016)12-1683-05

A Consistency Cleaning Method Based on Content-related Conditional Functional Dependencies

DU Yue-feng[1], *SHEN De-rong*[1], *ZHANG Liang*[2], *YU Ge*[1]
(1. School of Information Science & Engineering, Northeastern University, Shenyang 110819, China; 2. PLA 65154 Troops, Lingyuan 122513, China. Corresponding author: DU Yue-feng, E-mail: dr.duyuefeng@gmail.com)

Abstract: Based on conditional functional dependencies, content-related conditional functional dependencies (CCFDs) and the consistency cleaning method were presented based on CCFDs. By analyzing the relationship of the conditional functional dependencies, the related conditional functional dependencies were combined into CCFDs. The CCFDs can not only detect the consistencies under multi-conditional values, but also provide reference values for the consistency repairing. A consistency repairing-cost model was presented. Then the data was corrected to be consistent with the minimal repairing cost according to the actual data. And the repaired results are approved accuracy for both the inconsistency detection and the inconsistency repairing via the experimental evaluation on two real-life datasets.

Key words: data cleaning; conditional functional dependency; content relativity; data consistency; repairing-cost model

美国商业调查显示美国每年因数据质量造成的损失高达6000亿美元[1]. 数据一致性[2-3]是数据质量管理的一项重要内容. 不一致数据会使数据产生歧义进而对数据分析造成影响,所以必须加以更正.

随着对数据质量的研究愈加深入,关于数据一致性的管理技术也在不断成熟. 近年来,对数据一致性的研究主要包括:不一致数据的检测,不一致数据的修复,以及相关的质量管理系统. 文献[4-5]提出了一种条件函数依赖,通过对函数依赖进行扩展[6],可以更准确地对数据进行一致性检测. 文献[7]证明了一致性修复是一个NP完全问题,进而提出了一种启发式的修复方法. 文献[8]提出了一种质量管理系统,可以将一致性检测和修复融合在一起,对数据进行清洗. 数据内容之间是存在关联关系的,以上方法并没有加以考虑,因此,本文提出了一种基于内容相关的条件函数依赖,并以此对数据进行清洗.

收稿日期: 2015-07-31
基金项目: 国家重点基础研究发展计划项目(2012CB316201); 国家自然科学基金资助项目(61033007).
作者简介: 杜岳峰(1986-),男,辽宁沈阳人,东北大学博士研究生; 申德荣(1964-),女,辽宁铁岭人,东北大学教授,博士生导师; 于 戈(1962-),男,辽宁大连人,东北大学教授,博士生导师.

内容相关的条件函数依赖将相关联的条件函数依赖进行合并,可以检测多条件值下的数据一致性问题,并提供可用于一致性修复的参考值.结合内容相关的条件函数依赖,还提出了一种一致性的修复代价模型,首先计算修复代价,然后选择代价最低的修复策略,最终得到准确的修复结果.

1 修复规则定义和问题的提出

对于一个关系 R,R 上所有的属性集合记作 attr(R),R 中的元组数记作 n.

1.1 内容相关的条件函数依赖

定义 1 内容相关的条件函数依赖 CCFD (content-related conditional functional dependency):ψ:(C|Y→A,S_c).其中,C 是条件属性集合,Y 是变量属性集合,C 和 Y 由"|"分隔,并且 $C, Y \subset$ attr(R),$C \cap Y = \varnothing$,C,Y 合在一起称为规则左部,属性 A 称为规则右部;Y→A 是一个标准函数依赖;S_c 是合并后的条件值集合.

表 1 是 1994 年美国人口普查信息,下划线部分为错误数据,括号内为其真实值.同时,本文使用图 1 给出的条件函数依赖.一方面,条件函数依赖虽然可以检测出 t_1,t_2 之间存在的不一致问题,但是无法给出可用于修复的参考值;另一方面,条件函数依赖虽然可以保证 t_3 的一致性,但是无法检测出 t_3 中存在的错误信息.

表 1 1994 年美国人口普查信息

Table 1 1994 US adult census data

t_{id}	Country	Workclass	SalaryLevel
t_0	Brazil	employee	30 ~ 50 k
t_1	China	employee	30 ~ 50 k
t_2	China	employee	70 ~ 90 k(30 ~ 50 k)
t_3	India	employee	20 ~ 30 k(30 ~ 50 k)

通过分析表 1 中的数据关系,巴西、中国和印度的雇员工资水平是相互联系的.一方面,对于 t_1,t_2 之间的不一致问题,可以由 t_0 提供 SalaryLevel 值进行修复;另一方面,单独的 t_3 是满足一致性需求的,如果将 t_0 和 t_3 放在一起进行检测,就可以发现 t_3 中存在的错误,进而加以改正.

例 1 关于表 1 的条件函数依赖:

ψ:(Country,Workclass→SalaryLevel,tp). tp = (tp$_0$(Brazil,_ ‖ _), tp$_1$(China,_ ‖ _), tp$_2$(India,_ ‖ _))

本文得到的内容相关的条件函数依赖为

ψ:(Country|Workclass→SalaryLevel,S_c).

S_c = {Brazil,China,India}.

如果一条内容相关的条件函数依赖 ψ 关于 R 是成立的,当且仅当对于 $\forall u, v \in R$,在 $u[C]$,$v[C] \in S_c$ 的条件下,如果 $u[Y] = v[Y]$,那么 $u[A] = v[A]$.

定理 1 如果一条内容相关的条件函数依赖 ψ 关于 R 是成立的,当且仅当 $|\bigcup_{C_i \in S_c} \sigma_Y \pi_{C=C_i}(R)| = |\bigcup_{C_i \in S_c} \sigma_{Y \cup A} \pi_{C=C_i}(R)|$.其中,$\sigma_Y \pi_{C=C_i}(R)$ 表示在 R 上选择 $C = C_i$ 的元组,然后进行属性 Y 的投影操作.$|\bigcup_{C_i \in S_c} \sigma_Y \pi_{C=C_i}(R)|$ 表示 S_c 中所有情况下 $\sigma_Y \pi_{C=C_i}(R)$ 得到的结果总和中的不同值的个数.

证明:反证法证明充分性.设 $|\bigcup_{C_i \in S_c} \sigma_Y \pi_{C=C_i}(R)| \neq |\bigcup_{C_i \in S_c} \sigma_{Y \cup A} \pi_{C=C_i}(R)|$,那么至少存在一个非空集合 $S_c' \subset S_c$,使得在 $u[C]$,$v[C] \in S_c'$ 的情况下,对于 $u[Y] = v[Y]$,$u[A] \neq v[A]$.这与命题的题设相违背,所以假设不成立,原命题充分性成立.必要性的证明方法与充分性相同.定理 1 成立.

对于关系 R 上的一个实例 I,如果 I 满足内容相关的条件函数依赖 ψ,记作 $I \vDash \psi$.Σ 是内容相关的条件函数依赖的集合,如果 I 满足 Σ,记作 $I \vDash \Sigma$.

需要说明的是,本文只考虑合并具有相同 C,Y,A 属性的条件函数依赖.在此基础上,通过数据专家的分析将相关的条件函数依赖进行合并.

在现实生活中,内容相关的条件函数依赖表示相似的事物遵循相同的规则.在多条件值的记录中,如果某一条件值的记录出现数据一致性问题,可以通过其他条件值的记录进行检测和修复.

1.2 数据修复的代价模型

对于使用内容相关的条件函数依赖 ψ 检测出的 R 中存在的不一致元组集合 $E = \{t_k, \cdots, t_m\}$,本文设计了一种修复代价模型来修复不一致数据,首先给出一些与模型有关的定义.

定义 2 修复权重(repairing weight):$\omega_i = n_i/n$.其中,对于 $\forall C_i \in S_c$,n_i 表示 R 中 $C = C_i$ 条件下的元组数.权重越高表示修复 $C = C_i$ 的元组的代价也越大.

定义 3 修复目标值集合(repairing target set):

$$\mathrm{RT}(t_k) = \bigcup_{C_i \in S_c} \sigma_A \pi_{C=C_i, Y=t_k[Y]}(R). \quad (1)$$

其中,$t_k[Y]$ 是检测出的不一致元组 t_k 关于 Y 的属性值.针对出现的不一致元组 t_k,从包含在 S_c 的元组中,找出所有 $Y = t_k[Y]$ 的元组,选取这些元组关于 A 的属性值作为修复 t_k 的参考值.

定理 2 对于 $t_k, t_l \in E (k \neq l)$,如果 $t_l[Y] =$

$t_k[Y]$，那么 $\mathrm{RT}(t_l) = \mathrm{RT}(t_k)$.

对于包含相同 Y 属性值的不一致元组，它们拥有相同的修复目标值集合. 进而，本文提出下面的修复代价模型.

定义 4　修复代价模型(repairing-cost model)：

$$\mathrm{cost}(\mathrm{rt}(t_k)) = \sum_{i=0}^{n} \omega_i \cdot \mathrm{Isrepair}_i(\mathrm{rt}(t_k)). \quad (2)$$

其中：$\mathrm{rt}(t_k)$ 是需要修复的目标值，$\mathrm{rt}(t_k) \in \mathrm{RT}(t_k)$；$\mathrm{Isrepair}_i(\mathrm{rt}(t_k))$ 是修复判定函数，

$$\mathrm{Isrepair}_i(\mathrm{rt}(t_k)) = \begin{cases} 1, \text{if } t_i[Y] = t_k[Y], t_i[A] \neq \mathrm{rt}(t_k); \\ 0, \text{else}. \end{cases} \quad (3)$$

如果元组的 Y 属性值与修复的目标值相同，那么不进行修改；否则，考虑将 $t_i[A]$ 修改为 $\mathrm{rt}[t_k]$. 修复代价模型 $\mathrm{cost}(\mathrm{rt}(t_k))$ 用于计算将所有满足 $t_i[Y] = t_k[Y]$ 条件的不一致数据的 A 属性值修改为 $\mathrm{rt}[t_k]$ 的代价.

例 2 使用例 1 中给出的内容相关的条件函数依赖以及表 1 中的数据，首先得到修复权重 $\omega_0 = \omega_3 = 0.25$，$\omega_1 = \omega_2 = 0.5$，修复目标值集合 $\mathrm{RT}(t_0) = \{$“20 ~ 30 k”，“30 ~ 50 k”，“70 ~ 90 k”$\}$. 接下来，将“Brazil, China, India”3 个国家“employee”的“SalaryLevel”修复为“30 ~ 50 k”，其 $\mathrm{cost}(30 \sim 50\ \mathrm{k}) = 0.5 + 0.25 = 0.75$.

对于使用 ψ 检测出的不一致数据，为每一类包含相同 Y 属性值的不一致元组集合 E'（其中 $E' \subset E$，并且对于 $\forall E'_i, E'_j \in E'$，有 $E'_i[Y] = E'_j[Y]$），选择相同的修复目标值 $\mathrm{rt}[t_i]$，那么使用 ψ 进行修复的代价记作 $\mathrm{cost}(\psi)$. 对于使用内容相关的条件函数依赖的集合 Σ 进行修复的代价记作 $\mathrm{cost}(\Sigma)$.

1.3　问题的提出

本文要解决的问题是：给定关系 R 上的一个实例 I 以及关于 R 的内容相关的条件函数依赖的集合 Σ，使用 Σ 检测出 I 中的不一致数据并进行修复，使修复代价 $\mathrm{cost}(\Sigma)$ 最小并且满足 $I \models \Sigma$.

2　内容相关的条件函数依赖的清洗方法

2.1　数据一致性清洗方法

给定关系 R 上实例 I 以及内容相关的条件函数依赖集合 Σ，结合修复代价模型，本文提出的数据一致性清洗方法包括一致性检测和一致性修复两个过程，清洗方法如下：

算法 1 的第 5 行中，CCFDdetect( ) 用 ψ 来检测 I 中的数据是否一致，并将不一致的数据存在 E 中；第 7 行中，CCFDrepair( ) 对不一致进行修复，并将修复后的结果存在 I' 中. 其中，CCFDdetect( ) 和 CCFDrepair( ) 会在 2.2 节和 2.3 节中介绍. 对于算法 1，只要发现数据中存在不一致，就会使用 Σ 进行检测和修复，直到所有的数据一致为止.

算法 1：CCFD cleaning method

```
Input: Instance I, CCFDs Σ
Output: Consistent Instance I'
1: Initialize I' = I, E = null, tag = true;
2: while (tag) do
3:     tag = false;
4:     for each ψ in Σ do
5:         E = CCFDdetect(I', ψ);
6:         if (E is not empty) then
7:             I' = CCFDrepair( ), tag = true;
8: return I';
```

需要说明的是文中内容相关的条件函数依赖是不相互蕴含的[9]. 规则的蕴含关系的证明是一个 NP 完全问题，这里不进行详细讨论. 使用不相互蕴含的依赖进行修复一定会达到终止状态.

2.2　数据一致性检测方法

给定实例 I' 及内容相关的条件函数依赖 ψ，本文提出下面的数据一致性检测方法：

算法 2 的第 1 ~ 2 行使用定理 1 来判断 I' 中的数据是否一致，如果 I' 中的元组关于属性 C 的值包含在 ψ 中，那么使用 $\mathrm{select}(I', \psi)$ 返回这类元组的集合 T；如果数据存在不一致问题，通过第 4 ~ 7 行返回不一致的数据集合 E.

算法 2：CCFDdetect

```
Input: Instance I', CCFD ψ
Output: Error dataset E
1: T = select(I', ψ);
2: if (|T[Y]| = |T[Y∪A]|) then
3:     E = null;
4: else for i = 0 to |T| do
5:     for j = i + 1 to |T| do
6:         if (T_i[Y] = T_j[Y] && T_i[A] ≠ T_j[A]) then
7:             E = E ∪ T_i ∪ T_j;
8: return E;
```

2.3　数据一致性修复方法

给定不一致的数据集合 E 及内容相关的条件函数依赖 ψ，本文提出数据一致性修复方法.

算法 3 的第 1 行中，对于 E 中的所有错误数据，将所有包含相同 Y 属性值的错误归为一类，

使用 select$Y(E,\psi)$划分出所有错误类别存在集合 S 中;第 2 ~ 3 行,使用 repairtargets(E,S_i)找出每一类错误的修复目标集合 RT(S_i);第 4 ~ 7 行找出修复代价最小的目标值 rt,进而使用 repair(I',ψ,S_i,rt)将 I' 中处于 ψ 规则下满足 S_i 错误的元组统一修改为 rt,并返回修改后的结果 I'.

算法 3: CCFDrepair

Input: Instance I', Error dataset E, CCFD ψ
Output: Consistent Instance I'
1: $S = \text{select}Y(E,\psi)$;
2: for $i = 0$ to $|S|$ do
3: $\quad \text{RT}(S_i) = \text{repairtargets}(E,S_i)$, $\text{rt} = \text{rt}_0(S_i)$;
4: $\quad$ for $j = 1$ to $|\text{RT}(S_i)|$ do
5: $\quad\quad$ if $(\text{cost}(\text{rt}_j(S_i)) < \text{cost}(\text{rt}))$ then
6: $\quad\quad\quad \text{rt} = \text{rt}_j$;
7: $\quad I' = \text{repair}(I',\psi,S_i,\text{rt})$;
8: return I';

3 实验评估

本文使用两组真实数据进行实验,通过可扩展性实验和准确性实验验证清洗方法的效果.

3.1 实验设置

本实验使用的两组真实数据(Adults 数据集和 Census – Income 数据集)可以从 UCI 机器学习数据库中下载. 规则集合方面,本实验通过单独使用条件函数依赖和内容相关的条件函数依赖进行对比. 其中,Adults 数据集使用 1 172 个条件函数依赖,并将其合并成为 462 个内容相关的条件函数依赖;Census – Income 数据集使用的规则分别为 3 772 和 1 012. 进行一致性修复时,使用本文提出的代价模型同传统的 Voting 方法[10] 进行对比. Voting 使用投票的方法选取修改次数最少的修改策略进行一致性修复. 本实验的硬件环境为 Intel i7 – 2600(3.4 GHz)处理器及 8 GB 内存,使用 Java 语言实现.

3.2 可扩展性实验

本实验通过改变数据集中元组数量,观察一致性检测和修复的运行时间.

图 2 和图 3 描述了清洗方法在 Adults 及 Census – Income 数据集上的运行时间. 对于检测过程,由于内容相关的条件函数依赖将规则进行了合并,减少了检测的次数,所以花费的运行时间较短. 对于修复过程,由于内容相关的条件函数依赖检测出的错误更多,另外在修复时需要考虑更多的参考值,所以花费的运行时间稍长. 当元组数量 >25 000 时,运行时间开始趋于平缓.

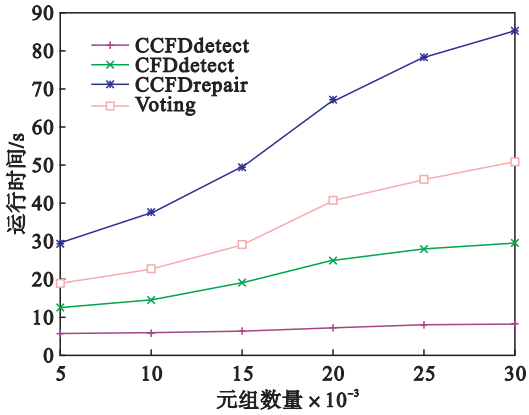

图 2 Adults 数据集上关于元组数量的运行时间
Fig. 2 Running time *w. r. t. n* over Adults

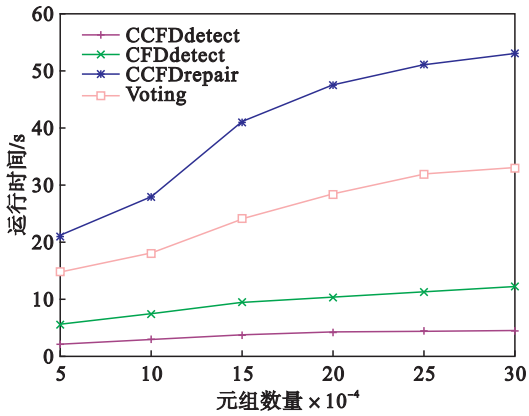

图 3 Census – Income 数据集上关于元组数量的运行时间
Fig. 3 Running time *w. r. t. n* over Census-Income

3.3 准确性实验

本实验通过改变数据集中错误数据的比例(noi),观察一致性检测和修复的结果. 其中,本文使用错误检测率(D-precision) $= \dfrac{\text{检测错误数}}{\text{检测次数}}$来描述检测的结果,使用错误修复准确率($R$-precision) $= \dfrac{\text{正确修复数}}{\text{实际错误数}}$来描述修复的结果.

图 4 表明内容条件函数依赖和函数依赖在 Adults 及 Census – Income 数据集上的错误检测率基本贴合数据集中的实际错误情况. 总体上 CFD 和 CCFD 保持了较高的错误检测率. 内容条件函数依赖在进行检测时需要借助其他条件下的数据进行检测,所以错误检测率稍高.

图 5 表明内容相关的条件函数在进行修复时比 Voting 方法的修复准确率更高. 其原因在于两个方面:1) 内容相关的条件函数依赖参考其他条件下的数据,检测出的错误更多;2) 内容相关的

条件函数依赖修复时参考其他的数据，修复更准确. 另外，随着错误数据比例的上升，内容条件函数依赖的修复准确率的变化更为平缓.

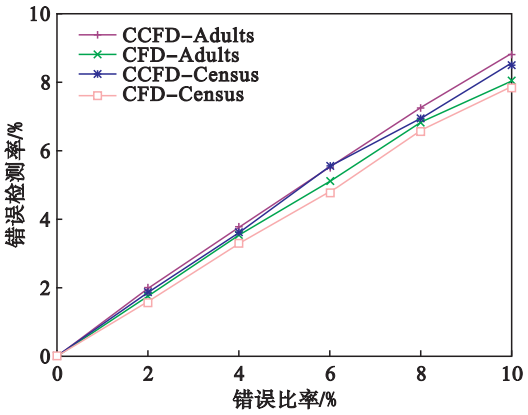

图 4　关于错误数据比例的错误检测率

Fig. 4　*D*-precision *w. r. t. noi*

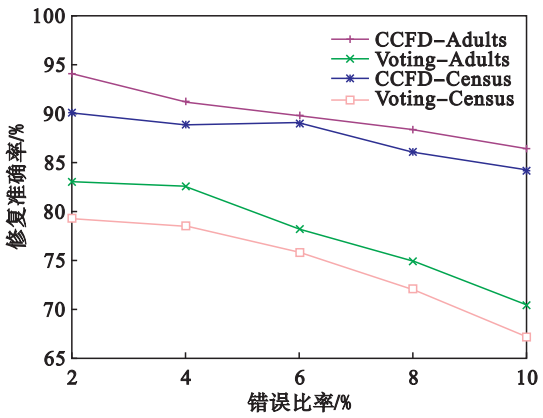

图 5　关于错误数据比例的错误修复准确率

Fig. 5　*R*-precision *w. r. t. noi*

此外，针对 Adults 数据集的原始数据，表 2 给出了检测和修复的实际结果.

表 2　Adults 数据集上检测和修复的实测数据

Table 2　Cleaning results and repair results on Adults

方法	检测错误数	实际错误数	检测次数	正确修复数	
				Voting	CCFDrepair
CFD	42 619	46 322	127 410	40 351	—
CCFD	43 541	46 322	127 410	—	42 881

从表 2 的结果可以看出，不论一致性检测还是一致性修复，考虑数据关联关系的方法都比传统方法更为准确. 此外，准确的检测是进行修复的基础，CCFD 方法的高准确性检测也为修复提供了良好的基础，得到的修复结果也更为准确.

4　结　　论

在条件函数依赖的基础之上，通过分析条件函数依赖的关系，本文提出了一种内容相关的条件函数依赖. 同时，本文提出了一种修复代价模型. 使用内容相关的条件函数依赖和修复代价模型进行数据一致性检测和修复，通过将关联的数据放在一起进行分析，可以更为准确地检测数据中存在的不一致问题并进行修复.

参考文献：

[1] Fan W. Data quality: theory and practice[C]// Proceedings of International Conference on Web-Age Information Management. Berlin: Springer-Verlag, 2012: 1 – 16.

[2] Fan W, Geerts F. Foundations of data quality management [M]. San Rafael: Morgan & Claypool, 2012: 1 – 201.

[3] Du Y F, Shen D R, Nie T, et al. Discovering condition-combined functional dependency rules[C]// Proceedings of the 16th Asia-Pacific Web Conference. Berlin: Springer-Verlag, 2014: 247 – 257.

[4] Fan W, Geerts F, Jia X, et al. Conditional functional dependencies for capturing data inconsistencies [J]. *ACM Transactions on Database Systems Tods Homepage*, 2008, 33 (2): 1 – 44.

[5] Fan W, Geerts F, Li J, et al. Discovering conditional functional dependencies [J]. *IEEE Transactions on Knowledge & Data Engineering*, 2011, 23(5): 683 – 698.

[6] Flesca S, Furfaro F, Parisi F. Consistency checking and querying in probabilistic databases under integrity constraints [J]. *Journal of Computer & System Sciences*, 2014, 80(7): 1448 – 1489.

[7] Fan W, Ma S, Tang N, et al. Interaction between record matching and data repairing [J]. *Journal of Data and Information Quality*, 2014, 4(4): 1 – 16.

[8] Dallachiesa M, Ebaid A, Eldawy A, et al. NADEEF: a commodity data cleaning system [C]// Proceedings of the 2013 ACM SIGMOD International Conference on Management of Data. New York: ACM, 2013: 541 – 552.

[9] Fan W. Dependencies revisited for improving data quality [C]// Proceedings of the 27th ACM SIGMOD – SIGACT – SIGART Symposium on Principles of Database Systems. Vancouver, 2008: 159 – 170.

[10] Bohannon P, Fan W, Flaster M, et al. A cost-based model and effective heuristic for repairing constraints by value modification[C]// Proceedings of the 2005 ACM SIGMOD International Conference on Management of Data. New York: ACM, 2005: 143 – 154.