



基于 Petri 网的数据清洗规则链自动组合与检测

何 俊¹, 张云飞¹, 张德海²

(1. 昆明学院 信息工程学院, 昆明 650214; 2. 云南大学 软件学院, 昆明 650206)

摘 要: 针对传统规则链顺序执行方法面向大规模数据清洗任务时存在的规则冗余和逻辑冲突问题, 提出一种规则链自动组合与检测方法。结合上下文信息设计通用、领域和自定义的三层规则库, 基于 Petri 网建立规则链组合模型(RCCM), 实现规则链自动生成、逻辑正确性与状态可达性检测以及规则链优选。以某地区扶贫领域的数据清洗应用为例, 通过 RCCM 模型实现的实验结果表明, 该方法能明显减少错误数据的产生, 提高数据清洗质量和效率。

关键词: 数据清洗; 分层规则库; 规则链; Petri 网; 逻辑冲突

开放科学(资源服务)标志码(OSID):



中文引用格式: 何俊, 张云飞, 张德海. 基于 Petri 网的数据清洗规则链自动组合与检测[J]. 计算机工程, 2020, 46(11): 124-131.

英文引用格式: HE Jun, ZHANG Yunfei, ZHANG Dehai. Automatic combination and detection of data cleaning rule chains based on Petri Net[J]. Computer Engineering, 2020, 46(11): 124-131.

Automatic Combination and Detection of Data Cleaning Rule Chains Based on Petri Net

HE Jun¹, ZHANG Yunfei¹, ZHANG Dehai²

(1. College of Information Engineering, Kunming University, Kunming 650214, China;

2. College of Software, Yunnan University, Kunming 650206, China)

[Abstract] To address the rule redundancy and logical conflicts of the sequential execution method of traditional rule chains applied to massive Data Cleaning(DC) task, this paper proposes an automatic combination and detection method for rule chains. A three-layer rule base including general, field-specific and customized layers is designed based on the context information. Then a Rule Chain Combination Model(RCCM) is established based on Petri Net(PN) to realize the automatic generation of rule chains, the detection of logical correctness and state accessibility, as well as the optimization of rule chains. The proposed method takes the DC application in the field of poverty alleviation in a certain area as an example. Experimental results on RCCM implementation show that the proposed method can significantly reduce generated error data and improves the quality and efficiency of DC.

[Key words] Data Cleaning(DC); hierarchical rule base; rule chain; Petri Net(PN); logical conflict

DOI:10.19678/j.issn.1000-3428.0056429

0 概述

随着信息技术的快速发展, 数据规模逐渐扩大, 劣质数据不断增加, 从而导致数据质量低下, 在一定程度上降低了数据可用性, 因此数据清洗(Data Cleaning, DC)技术应运而生^[1]。目前的数据清洗方法多数关注技术本身或者针对某个领域的语义和业务逻辑规则进行清洗, 面对复杂应用领域的大规模、

异构数据时表现出清洗效率低下、出错率高等问题。虽然清洗规则的孤立使用简化了问题的复杂度, 但由于没有严格的规则间逻辑校验机制, 致使规则冗余普遍存在, 逻辑冲突不易发现, 因此最终严重影响数据修复质量。

目前, 国内外学者在数据清洗规则库领域进行了大量研究。文献[2]提出一种基于动态可配置规则的数据清洗方法, 具有跨领域、可重用、可配置和

基金项目: 国家自然科学基金(61263043, 61864004); 云南省地方本科高校基础研究联合专项(2017FH001-05)。

作者简介: 何 俊(1977—), 男, 副教授、博士, 主研方向为数据分析; 张云飞, 讲师、硕士; 张德海, 副教授、博士。

收稿日期: 2019-10-28 **修回日期:** 2019-12-01 **E-mail:** 369885901@qq.com

可扩展等特点,提升了规则重用和清洗效率。文献[3]将数据质量问题分为单数据源模式层问题、单数据源实例层问题、多数据源模式层问题和多数据源实例层问题四大类,并给出了较清晰的规则分层思路。文献[4]针对数据噪声、缺失值和不一致数据等脏数据问题进行识别和修复。文献[5-6]围绕相似重复记录的识别与剔除方法展开研究,以召回率和准确率作为算法评价指标,对解决规则冗余问题具有一定的指导作用。文献[7]将数据清洗结合端到端质量执行机制进行上下文整体清洗。文献[8]对基于特征相似度、上下文和关系的规则推理方法进行研究,但没有给出具体模型和执行路径。文献[9-11]针对数据清洗中的逻辑不一致问题,利用规则推理方法进行降噪,具有一定的参考价值。文献[12]提出一种模仿专家手动操作的基于规则的数据清洗方法,但该方法未给出具体的实现步骤和算法。此外,文献[13-15]给出了大数据清洗规则的系统架构、具体方法和实现过程。文献[16-18]在大数据清洗系统中充分考虑了数据一致性问题,并有效地提升了数据质量。

目前,虽然在数据清洗、领域规则库和规则清洗等方面具有较多的研究,但是针对规则链组合和规则一致性问题的研究尚不多见。因此,本文提出一种分层的规则库,采用Petri网(Petri Net, PN)对其进行建模,并使用形式化方法对规则链流程的正确性和可达性进行推理与检测,同时对规则链进行优选。

1 分层规则库与规则链

数据清洗具有逻辑性强、上下文相关和不同领域重用难等特征^[19],可见,组合大量规则以批量执行数据清洗任务则较为复杂。因此,通过建立包含通用层、领域层和自定义层的三层规则库,将规则按可重用程度和规则间相关程度进行划分,重点关注同层内规则之间的逻辑关系,可为进一步实现规则批量执行提供基础。

定义1(规则) 将规则定义为一个三元组^[20],即 $R = (R^d, R^c, R^l)$,假设 R 为不可分割的最小逻辑单元,即原子规则,其中: R^d 表示规则唯一标识,由规则的层编码和顺序码组合而成; R^c 表示基于上下文的规则描述,定义为一个二元组 $R^c = (D^i, R^x)$, D^i 是待处理的目标数据项集合参数, R^x 是规则操作描述文档,采用Petri网标记语言(Petri Net Markup Language, PNML)进行描述^[21]; R^l 表示规则间的逻辑关系,定义为一个三元组 $R^l = (P^R, C^R, S^R)$, P^R 是前置规则集, C^R 是冲突规则集, S^R 是后续规则集。

定义2(规则层) 规则库是 R 的集合,包括通用规则层(General Rules Layer, GRL)、领域规则层(Field Rules Layer, FRL)和自定义规则层(Custom

Rules Layer, CRL),分别表示为 L^G 、 L^F 和 L^C 。每个规则层定义为一个三元组,以通用规则层为例, $L^G = (L^i, \{R\}, L^a)$,其中, L^i 表示层编码, $\{R\}$ 表示该层中所有规则的集合, L^a 表示规则层间的操作权限限制集。

定义3(规则选择集) 根据业务需求从规则库中选择或者自定义有限个 R 组成的集合称为规则选择集 $S = \{R_1, R_2, \dots, R_n\}$,称 $R_i^{d*} = \{R_1^d, R_2^d, \dots, R_n^d\}$ 为选择编码集, $R^{m*} = \{\cup R_k^m\}$, $k = 1, 2, \dots, n$ 为选择目标数据项集, $R^{l*} = \{\cup R_k^l\}$, $k = 1, 2, \dots, n$ 为选择规则间的逻辑关系集。

定义4(规则链) 假设在规则选择集 S 中包含 n 个规则且 $n \neq 0$,并设其中任意一个规则 R_k 为初始规则,则可根据业务需求建立选择集 S 中的规则链:

$$C = C_k(m) = \{(R_k^d, R_{k+1}^d, \dots, R_{k+m-1}^d) \cdot ((R_j^d \in R_{j+1}^l \cdot P^R) \vee (R_{j+1}^d \in R_j^l \cdot S^R)) \wedge ((R_j^d \notin R_{j+1}^l \cdot C^R) \vee (R_{j+1}^d \notin R_j^l \cdot C^R))\} \quad (1)$$

其中, $k \leq j \leq k+m-1$, m 是该规则链中的规则数量且满足 $0 \leq m \leq n$ 。规则链 C 是根据规则之间的前置规则集 P^R 、冲突规则集 C^R 和后续规则集 S^R 的逻辑和约束关系连接而成,只有满足以下条件,才能连接两个规则:

- 1) 前一个规则在后一个规则的前置规则集中或者后一个规则在前一个规则的后置规则集中。
- 2) 两个规则都不在对方的冲突规则集中。

由定义4可知,此处的规则链只定义了规则的执行顺序,并未考虑规则的并行、分支和循环等问题。规则选择集 S 中所有满足上述条件的规则链 C 组成的集合称为规则链生成集,记为 $C^* = \{\cup C_i\}$ 。

定义5(层间规则链组合) 假设 C_i 是通用规则层 L^G 中的规则链, C_j 是领域规则层 L^F 中的规则链,当且仅当满足以下条件时,两个规则链可以组合为层间规则链 C_{ij}^* 且 $C_{ij}^* = C_i \cup C_j$ 是一条新组成的层间规则链:

- 1) $C_i \not\subset \emptyset, C_j \not\subset \emptyset$ 。
- 2) $C_i \cdot R \notin L^G \cdot L^a, C_j \cdot R \notin L^F \cdot L^a$ 。

如果规则选择集 S 中有 n 个规则且所有规则之间都可以任意连接,则总共可生成 $(n-1)!$ 个规则链,但每一条规则链可能存在并行、选择和循环等多种组合关系。因此,需要进一步研究规则组合方法以及规则链的逻辑正确性和规则链优选等问题。

2 Petri网与规则链组合模型

2.1 Petri网

Petri网是一种状态变迁模型,用于描述系统异步和并发状态的变迁关系。

定义6(Petri网) 将Petri网定义为一个四元组 $P_N = (P, T, F, M)$ ^[22],当满足下列条件时,称 P_N 为Petri网:

- 1) $P \cup T \neq \emptyset, P \cap T = \emptyset$.
- 2) $F \subseteq \{(P \times T) \cup (T \times P)\}$.
- 3) $P \cup T = \{x \mid \exists y: (x, y) \in F\} \cup \{y \mid \exists x: (x, y) \in F\}$.
- 4) $M: P \rightarrow \mathbb{N}^+$ 是 P_N 的标识函数, M_0 为初始标识, \mathbb{N}^+ 为正整数集。

5) 触发规则, 如果 $\forall p \in {}^t t: M(p) \geq 1$, 则称变迁 t 是使能的, 表示为 $M[t >]$ 。如果状态标识 M 下 t 是使能的, 则称 t 可以触发, 且触发后得到的后继标识为 M' , 记为 $M[t > M']$, 并且:

$$M'(p) = \begin{cases} M(p) + 1, & p \in t' - {}^t t \\ M(p) - 1, & p \in {}^t t - t' \\ M(p), & \text{其他} \end{cases} \quad (2)$$

其中, P 为库所集合, T 为变迁集合, F 为基于 P 和 T 建立的有向弧集合。

定义 7 (输入集和输出集) 对于 $\forall x \in P \cup T$, 称 ${}^x = \{y \mid (y \in P \cup T) \wedge ((y, x) \in F)\}$ 为 x 的输入集, $x' = \{y \mid (y \in P \cup T) \wedge ((x, y) \in F)\}$ 为 x 的输出集^[22]。

定义 8 (可达标识集) 若 Petri 网中存在 $t \in T$ 使得 $M[t > M']$, 则称 M' 是从 M 可达的, 则 P_N 中从 M 可达的全部标识集合称为可达标识集^[22], 记为 $R(M)$, 且对 $\forall t \in T$, 推得 $\exists M \in R(M) \Rightarrow \exists M' \in R(M)$ 。

定义 9 (关联矩阵) 在 Petri 网中, 若 $P = \{p_1, p_2, \dots, p_n\}$, $T = \{t_1, t_2, \dots, t_m\}$, 则可表示为矩阵 $A = [a_{ij}]_{n \times m}$, 当且仅当 A 满足下列条件时, A 称为 P_N 的关联矩阵^[22-23]:

$$a_{ij} = a_{ij}^+ - a_{ij}^-, i \in \{1, 2, \dots, n\}, j \in \{1, 2, \dots, m\} \quad (3)$$

$$a_{ij}^+ = \begin{cases} 1, & (t_i, p_j) \in F \\ 0, & (t_i, p_j) \notin F \end{cases} \quad (4)$$

$$a_{ij}^- = \begin{cases} 1, & (p_j, t_i) \in F \\ 0, & (p_j, t_i) \notin F \end{cases} \quad (5)$$

定义 10 (迁移矩阵) 当且仅当矩阵 $K = A^- \text{diag}(t_1, t_2, \dots, t_n) A^+$ 满足下列条件时, 称 K 为 P_N 的迁移矩阵^[22-23]:

- 1) 当 $|t_i| = 1$ 时, 变迁触发, 其中, t_i 是 P_N 中的变迁, $i = 1, 2, \dots, n$ 。
- 2) 当 $|t_i| = 0$ 时, 变迁触发失效。

2.2 基于 Petri 网的规则链组合模型

在数据清洗操作开始前, 根据业务需求选择适合的规则选择集 S , 而从 S 中生成无冗余规则、逻辑正确和最优的规则链至关重要, 直接关系到规则链的自动执行和数据清洗质量。因此, 基于 Petri 网建立规则链组合模型 (Rule Chain Combination Model, RCCM), 在规则集执行前使用形式化方法对规则链的正确性和可达性进行检测。

定义 11 (规则链组合模型) 当且仅当满足下列条件时, 称四元组 $Q = (S, C^*, P_N, M)$ 为数据清洗规则链组合模型, 其中: S 表示包含 n 个原子规则 R 的规则选择集; P_N 表示包含有限库所集、有限变迁集和有向规则关系的 Petri 网; C^* 表示 S 的规则链生成集, P_N 的变迁集合 $T \subseteq S$; M 表示 P_N 中库所和变迁的标识符状态函数集。

对于规则链组合模型作如下说明:

1) RCCM 模型中 P_N 的所有库所集合 P 包含前置规则集 P^R (表示为 P_p)、后续规则集 S^R (表示为 P_s) 及冲突规则集 C^R (表示为 P_c), 并满足 $P = P_p \cup P_s \cup P_c$ 。为与 Petri 网特性保持一致, 定义两个特殊规则库所: 源规则库所和终止规则库所, 其中, 源规则库所对应规则链中的起始规则, 终止规则库所对应规则链中的终止规则。

2) RCCM 模型中 P_N 的变迁表示规则链中的规则 R , 在变迁集合 T 中, 对于 $\forall t \in T$, 在 ${}^t t$ 和 t' 中至少有一个前集和后续的元素相匹配且不在冲突集中。此时变迁使能, 即 $M[t > M']$ 并将规则不冲突作为变迁触发的前提条件:

$$M'(P) = \begin{cases} M(P) + 1, & P \in t' - {}^t t \text{ 且 } P \notin P_c \\ M(P) - 1, & P \in {}^t t - t' \text{ 且 } P \notin P_c \\ M(P), & \text{其他} \end{cases} \quad (6)$$

3) 此处的规则链已经由定义 4 中的规则顺序执行, 扩展到规则并行、分支和循环等逻辑结构。因此, RCCM 模型中基本逻辑结构包含顺序、并行、分支和循环 4 种, 如图 1 所示。

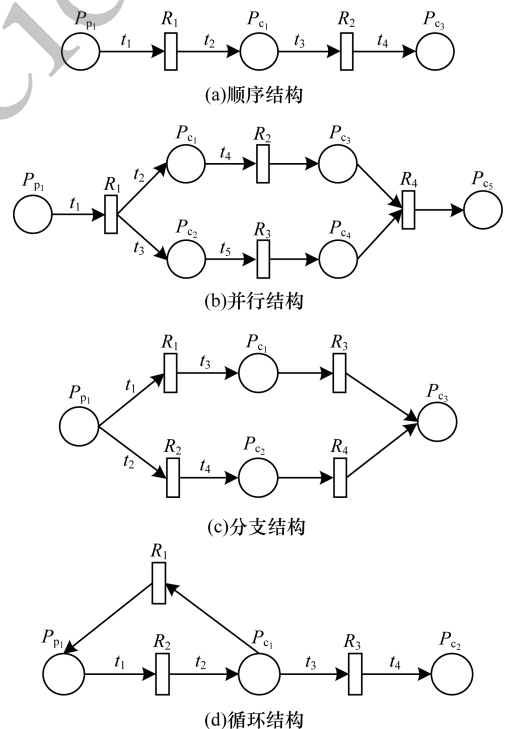


图 1 RCCM 模型中的基本逻辑结构

Fig. 1 Basic logical structure in RCCM model

3 RCCM 模型分析与检测

3.1 RCCM 模型相关问题

RCCM 模型是利用 Petri 网对并发和异步系统进行形式化表达和逻辑验证,构建可重用、可靠、高效的规则链组合和优选方法,提高数据清洗质量和效率。RCCM 模型形式化分析的前提条件为:

1) 规则语义规范性。为保证规则语义的一致性、清洗操作的协同性,采用 Petri 网描述语言对规则进行形式化描述,同时与模型语义保持一致。

2) 孤立规则。在给定的规则选择集 S 中,不能组成任何规则链的单个规则将被模型检测为孤立规则或冗余规则,尽管这些孤立库所不纳入模型重点考虑的范畴,但在实际应用中具有重要意义,必须作为单独的一类规则链参与数据清洗的执行过程。

3) 规则链优选指标。在保证规则链正确性、可达性和无死锁的前提下,需要在给定的规则链生成集 S^* 中判断最优规则链,当且仅当满足下列条件的规则链称为最优规则链 $C^*(m)$:

$$C^*(m) = \left\{ C_i \mid \left\{ \frac{m_i}{\sum_{j=0}^{m-1} l_j^i + 1} \right\} \right\} \quad (7)$$

其中, m_i 表示规则链 C_i 的规则数量, l_j^i 表示规则链 C_i 中第 j 个规则的重复计数。

4) 层间规则链组合。为简化模型且不失一般性,在组合层间规则链后直接进行数据清洗操作,不在模型中进行单独处理。

3.2 RCCM 模型分析

3.2.1 规则链生成与正确性检测

设规则链生成集 C^* 中的规则链 C 对应 Petri 网 P_N , 对于 $\forall k \in \mathbb{N}^+$, $R_k \in T$, 若 $\exists R_j \in T$, 使 $P_{c,k} \cap P_{p,j} \neq \emptyset$ 且 $R_k \notin P_{s,j}$, $R_j \notin P_{s,k}$, 则此时两个规则满足组合条件,且对应的标识符满足 $M(P_{c,k} \cap P_{p,j}) \subseteq M(R)$, 组合示意图如图 2 所示。

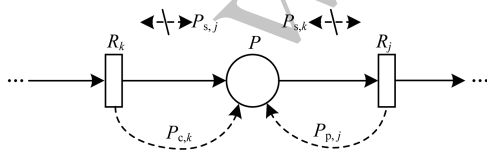


图2 规则 P_N 组合示意图

Fig.2 Schematic diagram of rule P_N combination

规则链中每组合一个规则需要完成一次回溯检测过程。当遍历所有 k 时,即可生成一个关于规则

链 C 的 P_N , 构建完成一个规则链 P_N 需要 $k(k-1)/2$ 次回溯检测,保证了规则链的逻辑正确性,且根据 P_N 的特点,规则链中可能存在顺序、并行、分支和循环 4 种结构。重复上述操作,可得到满足使能条件的所有关于规则链 C 的 P_N 集合,记为 P_x ,并将这些满足逻辑正确性的规则链数量记作 x 。

3.2.2 规则链可达性检测

在规则链 C 中, $\forall t_i \in T, R_j \in C$, 满足关联矩阵 A_{ij} , 即 C 中至少存在一条执行路径,满足迁移矩阵 K , 则规则链 C 满足可达性要求。此时, $\forall M' \in R(M)$, $\exists R_k \in C$, 使 $M_k[t_k, R_k > M'$ 。规则链可达性检测过程也是规则前置集和后续集的匹配验证过程,将这些满足可达性的规则链 C 的 P_N 集合记为 P_N^y , 数量记作 y 。

3.2.3 规则链优选

当 $x, y > 0$ 且 $P_N^x \cap P_N^y \neq \emptyset$ 时,可从 $P_N^x \cap P_N^y$ 中选择最优规则链,即从满足逻辑正确性和可达性的 P_N 中寻找最优规则链。根据式(7)中判断最优规则链的评价标准,通过计算找出最优规则链 $C^*(m)$ 并进行标识,同时将最优规则链 $C^*(m)$ 作为后续自动完成数据清洗操作的执行依据。

3.3 规则链生成与检测算法

根据上述规则链生成、规则链正确性和可达性检测,设计规则链生成与检测算法,具体如下:

算法1 规则链生成与检测算法

输入 规则选择集 S 、初始规则 R_0 、规则链最大长度 N

输出 规则链检测结果、最优规则链 $C^*(m)$

1. CheckModel(S, N)
2. For $i = 0$ to N when//读取选择集中的所有元素
3. $R[i] = S.R$;
4. $R[i].R_i = R.R_i \& R_c \& R_j$;
5. $C[0][0] = R_0$;//初始化规则链
6. For $i = 0$ to M when//生成与回溯检测规则链
7. For $j = 0$ to N when
8. If $(R_k^d, R_{k+1}^d, \dots, R_{k+m-1}^d) \mid (\forall R_j) ((R_j^d \in R_{j+1}^d \cdot P^R) \vee (R_{j+1}^d \in R_j^d \cdot S^R)) \wedge ((R_j^d \notin R_{j+1}^d \cdot C^R) \vee (R_{j+1}^d \notin R_j^d \cdot C^R))$
9. $C[i][j] = R_i$;
10. Check 1 is true.
11. For $i = 0$ to M when
12. If $(R_j \notin C[i])$
13. $M_k[t_k, R_k > M'$;//检测规则链可达性
14. Else If $([a_{ij}]_{j \times i}^+ \neq 0)$
15. $R_k = R_k + t_j$;
16. If $([a_{ij}]_{j \times i}^- = 1 \wedge R_j \notin C[i]) \parallel ([a_{ij}]_{j \times i}^+ = 1 \wedge R_j \notin C[i])$
17. Check 2 is true.
18. For $i = 0$ to M when//规则链优选

19. For $j = 0$ to N when
20. If $(C[i][j] = C[i][j+1])$
21. $l[i] = l[i] + 1$; //规则重复计数
22. $p = \max\{m_i / (l[i] + 1)\}$;
23. $C^*(m) = C[i]$;
24. Output Check 1 Check 2 and $C^*(m)$

以规则选择集和规则链最大长度作为算法输入,通过第 1 行~第 4 行读取选择集中的元素,第 5 行~第 7 行为初始化规则 R_0 并对每一个规则进行回溯遍历,第 8 行判断规则是否满足加入规则链的条件。第 11 行~第 17 行计算 Petri 网的状态可达图,测试 RCCM 库所及变迁是否正确,检测所生成的每一条规则链是否正确和可达。通过遍历可能生成多条规则链,因此第 18 行~第 24 行利用第 7 行的计算规则选择最优规则链并对其进行输出。假设规则选择集 S 中的规则数目为 n ,生成的规则链数目为 m ,每次循环都需要进行全部规则遍历,因此算法中规则链生成对应的时间复杂度为 $O(n^2)$,规则链检测对应的时间复杂度为 $O(m \times n)$,空间复杂度均为 $O(m \times n)$ 。算法 1 实现了规则链生成、规则链正确性和可达性检测以及规则链自动优选过程,从逻辑上保证了后续数据清洗操作执行的可靠性。

4 实验结果与分析

4.1 实验数据集设置

实验以某地区扶贫领域的数据清洗应用为背景,从实际数据清洗规则库中提取出部分规则作为选择集,建立 RCCM 模型。以该地区实际扶贫数据为实验数据,分别使用本文方法和传统规则链顺序执行方法^[24]进行对比实验。实验数据集设置如下:

1) 实验目标数据集 DataSet,主要包括贫困人口基础数据集和其他辅助清洗数据集。贫困人口基础数据集为{序号,户编号,人编号,姓名,证件类型,证件号码,与户主关系,民族,文化程度,在校状况,劳动力状况,务工时间,大病保险,脱贫属性,脱贫年份,户属性,房屋状况,人均纯收入,联系电话,识别时间,帮扶责任人编码},数据记录 350 000 条。其他辅助清洗数据集包含人口、卫健、教育、银行、交通、税务、工商、残联、民政等 9 个行业单位的异构数据记录 900 多万条^[25]。

2) 数据清洗分层规则和规则选择集 S 。根据数据清洗业务的目标要求,第 1 次先抽取 GRL 层中的 5 个规则、FRL 层中的 10 个规则及 CRL 层中的 5 个规则,共 20 个规则作为规则选择集(如表 1~

表 3 所示),并在此基础上再次增加规则数量。GRL 层主要包括通用清洗规则,通常作为进一步开展业务清洗的基础。FRL 层主要包括业务逻辑比对和逻辑错误数据清洗规则,通常需要符合业务实际情况。CRL 层包括根据用户扩展的规则。每次实验选取的规则将作为 RCCM 模型实现的规则选择集 S 。

表 1 GRL 层中的规则设置
Table 1 Rules setting of GRL layer

编号	含义
1-001	重复记录检测及处理
1-002	空数据项检测及处理
1-003	异常数据项检测及处理
1-004	身份证号码数据项规范性验证及处理
1-005	日期数据项规范性验证处理

表 2 FRL 层中的规则设置
Table 2 Rules setting of FRL layer

编号	含义
2-001	姓名和身份证号码联合比对及处理
2-002	贫困户有劳动力归属兜底一批检测
2-003	同户贫困人口人均收入不一致检测
2-004	脱贫户房屋不达标检测
2-005	贫困户有劳动力无务工时间不达标检测
2-006	义务教育时间儿童未在校校验及处理
2-007	脱贫人口人均收入未达标校验
2-008	贫困人口有超标机动车检测
2-009	残疾证与身份证号码不一致检测
2-010	贫困人口有企业股份检测

表 3 CRL 层中规则设置
Table 3 Rules setting of CRL layer

编号	含义
3-001	贫困人口手机号码非本地
3-002	贫困人口与帮扶责任人手机号码相同
3-003	帮扶责任人帮扶对象超过 5 户
3-004	帮扶责任人符合政策条件情况检查
3-005	帮扶责任人无帮扶对象

4.2 RCCM 模型实现

根据规则选择集 S 建立 RCCM 模型。本文首先需要根据清洗目标建立每一个规则的前置规则集 P^R 、冲突规则集 C^R 和后续规则集 S^R ,然后使用算法 1 的回溯遍历方法生成规则链,经过正确性和可达性检测后生成规则链 P_N 集,最后计算出最优规则链 $C^*(m)$ 执行数据清洗操作。扶贫领域的 RCCM 模型执行流程如图 3 所示。

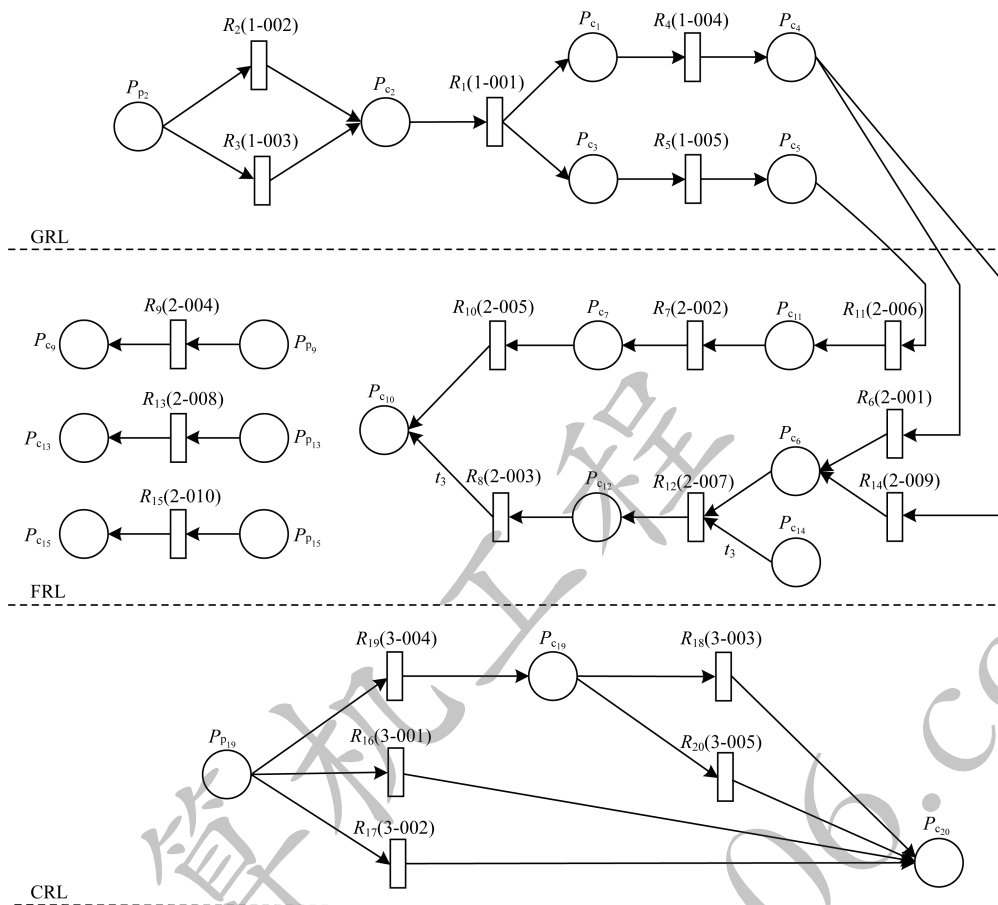


图 3 扶贫领域的 RCCM 模型执行流程

Fig. 3 Execution process of RCCM model in poverty alleviation field

从 RCCM 模型实现结果可以看出,RCCM 模型有效优化了规则之间的逻辑关系和执行顺序,例如规则 1-001 和 1-003,如果采用传统规则链顺序执行方法先执行 1-001 再执行 1-003,即先检测重复记录再检测和处理异常数据,则重复记录通常会严重影响异常数据检测和处理,而采用 RCCM 模型先执行 1-003 再执行 1-001,避免了上述问题。另外,3-003 和 3-005 有逻辑冲突,即帮扶责任人帮扶对象超过 5 户和帮扶责任人无帮扶对象两种情况不可能同时存在,无需同时执行两个规则,属于规则并行结构。因此,RCCM 模型通过检测逻辑冲突,选择最优规则链,从而提高数据清洗效率。

4.3 对比方法与结果分析

在扶贫领域数据清洗实际应用场景中,具有数据量大、异构数据源多和分级清洗等特点^[26],由于目前采用传统规则链顺序执行方法主要存在效率低、错误传递等问题,因此将通过逐步增加规则数量的方式,分别采用本文方法和传统规则链顺序执行方法对实验目标数据集 DataSet 进行数据清洗再比

较实验结果。

实验环境为包含 2 个 8 核 CPU 的服务器 1 台、Windows 10 Server 操作系统、SQL Server 2014 数据库,并采用 XML 的方式存储规则。实验待清洗目标数据为 356 123 条贫困人口基础数据,辅助数据为 9 325 642 条行业扶贫数据,分别采用本文方法和传统规则链顺序执行方法各自独立开展 4 次实验,规则数从第 1 次的 20 个分别增加至 50 个、100 个、200 个(由于规则编辑和配置工作量较大,因此本文中不再增加规则数量),其中分层规则数量采用各层规则等比增加的方式。同时,为避免引入特殊规则使实验结果失真,规则均在同一类型基础上进行增加。时间消耗以服务器记录时间为准,错误数据的评判标准为采用实验数据集与国家扶贫办基础数据库已校准的对应数据集进行比对,若发现不一致则再经过人工核对,最后确认为符合规则逻辑但被错误删除或修改的数据,如表 4 所示。从实验结果看,本文方法和传统规则链顺序执行方法都产生了错误数据,错误数据量和时间开销均随着规则数量的增多而增加。

表 4 本文方法与传统规则链顺序执行方法的实验结果对比

Table 4 Comparison of experimental results of the proposed method and traditional rule chain sequential execution method

方法	规则数量为 20		规则数量为 50		规则数量为 100		规则数量为 200	
	时间/s	错误数据量	时间/s	错误数据量	时间/s	错误数据量	时间/s	错误数据量
本文方法	193	3.0	512	9.0	1 043	21.0	1 988.0	28
传统规则链顺序 执行方法	207	12.0	877	44.0	2 471	172.0	4 567.0	422
平均值	200	7.5	699	26.5	1 757	96.5	3 277.5	225

本文分别从错误数据量和时间开销两方面对实验结果进行分析,如图 4 和图 5 所示。可以看出,随着规则数量逐步增多,传统规则链顺序执行方法的错误数据量急剧增加,而本文方法的错误数据量增加比较平稳,说明其可以有效减少错误数据的产生,并且所消耗的时间更少,具有更高的执行效率。

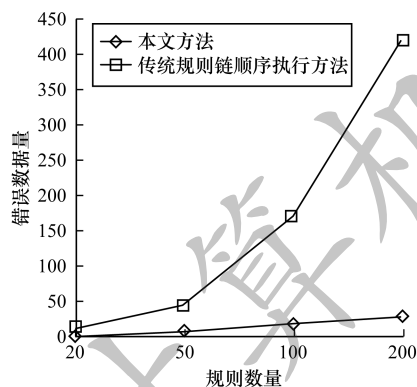


图 4 本文方法与传统规则链顺序执行方法的
错误数据量对比

Fig. 4 Comparison of the number of error data between
the proposed method and traditional rule chain
sequential execution method

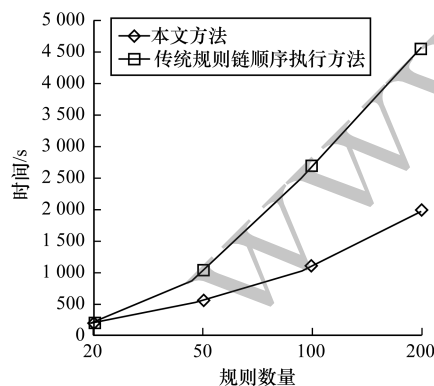


图 5 本文方法与传统规则链顺序执行方法的
时间消耗对比

Fig. 5 Comparison of time consumption between the
proposed method and traditional rule chain
sequential execution method

5 结束语

本文针对数据清洗规则链组合和规则一致性问题,提出一种分层的规则库,并采用 Petri 网建立数据清洗规则链组合模型,对规则链进行逻辑正确性和可达性检测,从而选择最优规则链执行数据清洗操作。实验结果表明,该方法能有效减少错误数据量,并具有更高的执行效率。后续将对规则链分层组合效率进行研究,进一步提高规则重复利用率和数据修复质量。

参考文献

- [1] WU Xindong, DONG Bingbing, DU Xinzhen, et al. Data governance technology [J]. Journal of Software, 2019, 30(9): 2830-2856. (in Chinese)
吴信东,董丙冰,堵新政,等.数据治理技术[J].软件学报,2019,30(9):2830-2856.
- [2] ZHU Huijuan, JIANG Tonghai, ZHOU Xi, et al. Data cleaning method based on dynamic configurable rules [J]. Computer Application, 2017, 37(4): 1014-1020. (in Chinese)
朱会娟,蒋同海,周喜,等.基于动态可配置规则的数据清洗方法[J].计算机应用,2017,37(4):1014-1020.
- [3] RAHM E, DO H. Data cleaning: problems and current approaches [J]. IEEE Data Engineering Bulletin, 2000, 23(4): 3-13.
- [4] TANG N. Big data cleaning [M]//SHENG Q Z, WANG G R, JENSEN C S. Web technologies and applications. Berlin, Germany: Springer, 2014: 13-24.
- [5] LEE M L, LING T W, LOW W L. IntelliClean: a knowledge based intelligent data cleaner [C]//Proceedings of the 6th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. New York, USA: ACM Press, 2000: 290-294.
- [6] MONGE A E. Matching algorithms within a duplicate detection system [J]. IEEE Data Engineering Bulletin, 2000, 23(4): 14-20.
- [7] PAPOTTI P, CHU X, ILYAS I F. Holistic data cleaning: putting violations into context [C]//Proceedings of the 29th International Conference on Data Engineering. Washington D. C., USA: IEEE Press, 2013: 458-469.

- [8] CAO Jianjun, DIAO Xingchun, WANG Ting, et al. Research on domain-independent data cleaning: a survey[J]. Computer Science, 2010, 37(5): 26-29. (in Chinese)
曹建军, 刁兴春, 汪挺, 等. 领域无关数据清洗研究综述[J]. 计算机科学, 2010, 37(5): 26-29.
- [9] SALEM R, ABDO A. Fixing rules for data cleaning based on conditional functional dependency[J]. Future Computing and Informatics Journal, 2016, 1(1/2): 10-26.
- [10] LEDERER C, ALTSTADT S, ANDRIAMONJE S. Web usage data cleaning: a rule-based approach for Weblog data cleaning[C]//Proceedings of the 20th International Conference on Big Data Analytics and Knowledge Discovery. Washington D. C., USA: IEEE Press, 2018: 1-10.
- [11] YANG Xue, TANG Luliang, ZHANG Xia, et al. A data cleaning method for big trace data using movement consistency[J]. Sensors, 2018, 18(3): 824-828.
- [12] SORRISO A, SORRENTINO P, RUCCO R, et al. An automated magnetoencephalographic data cleaning algorithm[J]. Computer Methods in Biomechanics and Biomedical Engineering, 2019, 22(14): 1116-1125.
- [13] DALLACHIESA M, EBAID A, ELDAWY A, et al. NADEEF: a commodity data cleaning system[C]//Proceedings of 2013 ACM SIGMOD International Conference on Management of Data. New York, USA: ACM Press, 2013: 541-552.
- [14] BATINI C, CAPPIELLO C, FRANCALANCI C, et al. Methodologies for data quality assessment and improvement[J]. ACM Computing Surveys, 2009, 41(3): 16-52.
- [15] FENG Fujun, YAO Junping, LI Xiaojun. Research on the technology of data cleaning in big data[EB/OL]. [2019-09-04]. https://www.researchgate.net/publication/328918985_Research_on_the_Technology_of_Data_Cleaning_in_Big_Data.
- [16] BESKALES G, ILYAS I F, GOLAB L, et al. On the relative trust between inconsistent data and inaccurate constraints[C]//Proceedings of the 29th International Conference on Data Engineering. Washington D. C., USA: IEEE Press, 2013: 541-552.
- [17] FAN Wenfei, LI Jianzhong, MA Shuai, et al. Interaction between record matching and data repairing[J]. Journal of Data and Information Quality, 2014, 4(4): 16-19.
- [18] FAN W F, GEERTS F, TANG N, et al. Inferring data currency and consistency for conflict resolution[C]//Proceedings of the 29th International Conference on Data Engineering. Washington D. C., USA: IEEE Press, 2013: 11-28.
- [19] PAPOTTI P, XU C, ILYAS I F. Holistic data cleaning: putting violations into context[C]//Proceedings of 2013 IEEE International Conference on Data Engineering. Washington D. C., USA: IEEE Press, 2013: 458-469.
- [20] PENG Nan, WANG Hongya, DING Wencheng, et al. Finding interesting cleaning rules from dirty data[C]//Proceedings of the 10th International Symposium on Computational Intelligence and Design. Washington D. C., USA: IEEE Press, 2017: 378-382.
- [21] RIAHI I, MOUSSA F. A formal approach for modeling context-aware human-computer system[J]. Computers and Electrical Engineering, 2015, 44: 241-261.
- [22] YUAN Chongyi. The application of Petri Net[M]. Beijing: Science Press, 2013. (in Chinese)
袁崇义. Petri网应用[M]. 北京: 科学出版社, 2013.
- [23] JIANG Wei, ZHOU Kaiqing, MO Liping. Parameter optimization strategy of fuzzy Petri Net utilizing hybrid GA-SFLA algorithm[M]. Berlin, Germany: Springer, 2019.
- [24] LU Xing. Data cleaning technology and application of big data[J]. Electronic Technology and Software Engineering, 2019(9): 157. (in Chinese)
卢星. 大数据的数据清洗技术及运用[J]. 电子技术与软件工程, 2019(9): 157.
- [25] HE Jun, ZHANG Dehai. Big data intelligent analysis model of targeted poverty alleviation in Yunnan minority areas[J]. Journal of Yunnan University for Nationalities (Natural Science Edition), 2018, 27(3): 249-254. (in Chinese)
何俊, 张德海. 云南少数民族地区精准扶贫大数据智能分析模型[J]. 云南民族大学学报(自然科学版), 2018, 27(3): 249-254.
- [26] NIE Yanmin. Using big data to improve poverty alleviation effect in deep poverty areas[J]. People's Forum, 2019(15): 50-51. (in Chinese)
聂燕敏. 用大数据提升深度贫困地区脱贫成效[J]. 人民论坛, 2019(15): 50-51.