计算机研究与发展
Journal of Computer Research and Development
DOI:10.7544/issn1000-1239.2020.20190570
57(9): 1939-1948, 2020

异构模式中关联数据的一致性规则发现方法

杜岳峰　李晓光　宋宝燕
(辽宁大学信息学院　沈阳　110136)
(duyuefeng@lnu.edu.cn)

Discovering Consistency Constraints for Associated Data on Heterogeneous Schemas

Du Yuefeng, Li Xiaoguang, and Song Baoyan
(*Information College*, *Liaoning University*, *Shenyang* 110136)

Abstract Data consistency is a central issue of data quality management. With capability of expressing data relationship abstractly and formally, constraints are a technique for data consistency management. However, the diversity on heterogeneous schemas from multi-source brings great challenges to data consistency management, especially for constraints fusion. Besides, for both data from single-sources and multi-sources, they are related. These relationships can be used to strengthen the expression of constraints for semantics, which helps to probe potential data error. In practice, CINDs (conditional inclusion dependencies) and CCFDs (content-related conditional functional dependencies) are two effective techniques respectively for attributes match under heterogeneous schemas and consistency maintenance on content-related data. Based on this, we study how to discover consistency constraints for associated data on heterogeneous schemas. We firstly investigate the three fundamental problems related to CCFDs discovery. And we also illustrate that the implication, satisfiability and validation problems are NP-complete, coNP-complete, PTIME. Aiming at searching for the CCFDs in the space entirely, we present 2-level lattice according to the division between the conditional attribute set and the variable attribute set of CCFDs. After that an incremental method of discovering the fusion constraints over CINDs and CCFDs is proposed, which combines CCFDs on heterogeneous schemas via CINDs. Finally, our method is experimentally verified effectively and scalablely by using two real-life data.

Key words heterogeneous schemas; associated data; CINDs (conditional inclusion dependencies); CCFDs (content-related conditional functional dependencies); constraints discovery

摘　要 数据一致性是数据质量管理的一项核心事务.规则约束作为一种抽象化、形式化的数据关系表达技术,可以有效地进行数据一致性管理.但是,在进行多源数据一致性管理的过程中,由于异源数据所属的关系模式不同,给一致性规则融合带来了挑战.另外,不论同源数据还是异源数据,数据之间是相互关联的,可以利用这种关系强化规则约束中语义含义的表达作用,发现数据中的潜在错误.具体地,条件包含依赖(conditional inclusion dependencies, CINDs)和内容相关的条件函数依赖(content-related

收稿日期:2019-08-19;**修回日期**:2020-03-04
基金项目:国家自然科学基金项目(U1811261);辽宁公共舆情与网络安全大数据系统工程实验室专项资金;辽宁省自然科学基金项目
This work was supported by the National Natural Science Foundation of China (U1811261), the Project of Liaoning Provincial Public Opinion and Network Security Big Data System Engineering Laboratory, and the Natural Science Foundation of Liaoning Province.
通信作者:李晓光(xgli@lnu.edu.cn)

conditional functional dependencies, CCFDs)可以分别用于异构模式的属性匹配和内容关联数据的一致性维护.基于此,对面向异构关系模式中关于关联数据的一致性规则发现问题进行研究.首先,针对使用 CINDs 进行异构模式中 CCFDs 规则发现的基本问题进行分析,对规则发现的可满足性、蕴含性和可验证性问题进行解释,它们分别满足 NP-complete,coNP-complete,PTIME 的复杂性判定问题.其次,为了对规则空间内的全部 CCFDs 进行发现,以 CCFDs 中的条件属性和变量属性为划分依据,提出了一种 2 级 lattice 的搜索结构.再次,设计了一种基于 CINDs 和 CCFDs 的异构关联数据一致性规则发现方法,使用 CINDs 对规则形式进行融合,而后通过增量发现方式查找一致性规则.最后,通过在 2 组真实数据进行实验,验证了方法的有效性和高效性.

关键词 异构关系模式;关联数据;条件包含依赖;内容相关的条件函数依赖;规则发现

中图法分类号 TP311

数据一致性是质量管理的一项核心事务[1],主要针对数据中存在的冲突情况进行处理.规则约束作为数据一致性管理的重要技术,可以从语义层面对数据实体关系进行形式化地抽象,有效地对冲突数据进行检测和修复,比如条件函数依赖(conditional functional dependencies, CFDs)[2].

随着信息产业的发展和大数据的普及,现实生活中的数据呈现出海量数据增长和多源多模态的分布特点,导致大数据环境下的一致性问题变得更加突出.具体表现为:一是数据中潜在的错误更容易产生;二是对异构模式数据的管理更加复杂.

图 1 描述了异构模式中的数据冲突情况.

T,S 是商品销售记录的 2 种关系模式.版权分类($Catg$)、类型($Type$)、名字($Name$)、价格区间($Price$)、出版商(Pub)是 T 上的 5 个属性.类型($Type$)、书名($Title$)、价格区间($Price$)、出版商(Pub)是 S 上的 4 个属性.I_T,I_S 是 T 和 S 上的关系实例.其中,黑体部分为错误数据,括号内为现实生活中对应的真值情况.

t_{id}	$Catg$	$Type$	$Name$	$Price$	Pub
t_1	literature	crime	Different Seasons (Hope Spring Eternal)	0-50	Shanghai Culture Publisher
t_2	file	crime	Shawshank Redemption	0-100	Warner Bros.

CFD ϕ_T: ($Catg$, $Name \rightarrow Price$), tp(tp_0("literature", _||_))

(a) I_T

s_{id}	$Type$	$Title$	$Price$	Pub
s_1	literature	Shawshank Redemption	**50-100** (0-50)	People's Publisher
s_2	literature	Shawshank Redemption	**50-100** (0-50)	Shanghai Culture Publisher

CFD ϕ_S: ($Type$, $Title \rightarrow Price$), tp(tp_0("literature", _||_))

(b) I_S

Fig. 1 Inconsistencies under heterogeneous schemas

图 1 异构模式中的数据冲突

文献[2]中,CFDs 是一种可以根据元组的特定条件属性判定数据一致性的规则约束方法.具体地,对于 CFD ϕ:($X \rightarrow Y$,tp),$X \rightarrow Y$ 是函数依赖(functional dependencies, FDs)的一般形式,tp 是特定的条件属性模板.对于任意元组 u,v,在 $u[X]=v[X]=tp[X]$ 的情况下,如果 $u[Y]=v[Y]$,则称 u,v 是一致的;否则,如果 $u[Y]\neq v[Y]$,则称 u,v 是不一致的.

ϕ_T,ϕ_S 是 T,S 上的 CFDs.以 ϕ_S 为例,ϕ_S 描述了在 S 上,对于所有的文学作品,它们的名字可以决定它们的价格区间.虽然 ϕ_T 和 ϕ_S 表达的意思相近,但是由于处于异构关系模式,所以对应的属性也不尽相同(如 T 中 $Catg$ 与 S 中的 $Type$ 对应).另外,对于相同的 $Type$ 属性,它们在 T 和 S 中表达的含义也不相同,会造成属性的二义性.这些都增加了规则融合的难度.

其次,尽管 I_T,I_S 中的记录都分别满足 ϕ_T,ϕ_S 的一致性要求,比如对于 s_1,s_2,在 $s_1[Type]=s_2[Type]=$"literature",$s_1[Title]=s_2[Title]=$"Shawshank Redemption"的情况下,有 $s_1[Price]=s_2[Price]$.文学作品"Shawshank Redemption"的价格区间都是 50~100 元,说明它们在 I_S 中是一致的.但是 $s_1[Price]$,$s_2[Price]$与真实值相不符,使得它们成为一种潜在的错误.

对于异构关系模式中的潜在错误数据,可以使用如图 2 所示的条件包含依赖(conditional inclusion dependencies, CINDs)[3]和内容相关的条件函数依赖(content-related conditional functional dependencies, CCFDs)[4]来进行处理.

η 可以根据特定的文学作品名字,将 T,S 模式中的 Pub、$Price$、类型($Catg$,$Type$)、名字($Name$,

Title)属性进行匹配.现实生活中的数据是相互关联的[5],可以利用这种关系将关联数据放在一起进行检测.现实生活中,"Shawshank Redemption"和"Different Seasons (Hope Spring Eternal)"指的是同一本文学作品,因此,可以把2类数据放在一起进行检测.进而,可以通过 η 将模式 T,S 联系到一起,然后使用 φ_T 检测出 s_1,s_2 中的潜在错误.

CIND η: $S(Pub, Price, Type=$"literature", $Title=$"Shawshank Redemption")$\subseteq T(Pub, Price, Catg=$"literature", $Name=$"Different Seasons(Hope Spring Eternal)")
CCFD φ_T: $(Catg, Name \mid \rightarrow Price), Sc\{Sc_0\{$"literature, Different Seasons (Hope Spring Eternal)", "literature, Shawshank Redemption"$\}\}$

Fig. 2 CINDs and CCFDs under heterogeneous schemas

图2 异构模式中的CINDs和CCFDs

但是,由于人们对领域和专业知识的局限,缺乏对数据关系的深层理解,使用人工方法进行规则发现有可能出现规则缺失、混淆和冲突的情况.在耗费大量人力资源的情况下还无法得到可靠的结果,这就需要一种可靠高效的自动规则发现方法.本文在进行异构模式一致性规则发现的研究过程中遇到的挑战有:

1) 异构模式中的属性关系复杂,存在不同属性之间的匹配问题,在规则中混合CINDs进行规则发现会对一致性规则的发现问题产生影响,可能造成属性对应关系的混乱.

2) 在使用CINDs对一致性规则进行融合和规则发现时,由于CINDs的属性之间存在包含问题,这会对融合后的一致性规则的条件范围产生影响.

针对上述问题,本文利用CINDs对CCFDs进行异构关系模式下的规则发现,具体贡献为:

1) 对使用CINDs进行异构关系模式下CCFDs规则发现问题进行分析,解释规则发现的可满足性、蕴含性和可验证性问题.

2) 提出了一种使用CINDs进行异构模式和实例的融合方法.

3) 设计了一种基于2级lattice结构的CCFDs规则发现方法,对所有规则空间进行搜索.

4) 通过在NBA数据和豆瓣数据上进行实验,验证本方法的有效性和高效性.

1 相关工作

数据质量管理是数据库领域研究的一个经典问题.本文主要研究异构模式关联数据中一致性规则的发现问题.目前与本文研究最为密切的相关工作包括3个方面:

1) 规则约束.规则约束是数据质量管理的一个重要技术,包含很多种类,如条件函数依赖(CFDs)[2]、扩展条件函数依赖(extended conditional functional dependencies, eCFDs)[6]、内容相关的条件函数依赖(CCFDs)[4]、条件包含依赖(CINDs)[3]等.特别地,文献[7]提出了一种图函数依赖(graph entity dependencies, GEDs),可以对图上的实体数据进行一致性分析.

通常,不同类型的规则可以混合使用,同时解决多类数据质量的混合问题.文献[3]使用CINDs对特定条件的异构模式数据进行匹配,并分析了CINDs与CFDs之间的相互作用关系.文献[8]提出了一种针对实体一致性的链接算法.

2) 规则发现.规则发现是对规则约束进行形式化语义抽象的自动发现方法.具体地,Huhtala等人在文献[9]中基于lattice结构提出了一种用于FDs规则发现的TANE方法.基于此,钟评等人在文献[10]提出了一种基于置信度的FDs规范发现方法.再者,Chiang等人在文献[11]中提出了一种CFDs规则方法,可以搜索特定条件下的函数依赖.此外,文献[12-13]提出了GEDs和OFDs(ontology functional dependencies)的规则发现方法.

3) 异构数据清洗.异构数据清洗主要针对异构模式的数据和语义2方面内容进行质量管理.具体地,马茜等人在文献[14]中提出了一种针对多源多模态数据的动态感知方法.Dallachiesa等人在文献[15]中提出了一种自动商业数据清洗系统,可以对多种类型的语义规则进行管理.文献[16-19]对质量管理的评价标准和可扩展的大规模数据质量管理等内容进行了相关研究.

此外,时序数据[20]、知识图谱、图数据[21]、实体数据质量管理的研究内容正在逐渐升温,成为数据质量管理的研究热点.其中,樊文飞等人通过长期对数据质量的研究,在文献[7,22]中着重描述了未来数据质量管理的研究和发展方向.Ortona等人在文献[23]中对使用知识图谱进行规则发现的相关内容进行了研究.

2 异构关联数据的一致性问题

本节针对异构模式关联数据的一致性规则发现问题,首先对CINDs和CCFDs进行介绍,然后给出

了规则发现问题的相关概念及分析.

本文使用 $R=(t_{id},A_1,A_2,\cdots,A_N)$ 形式的关系模式,其中 t_{id} 表示记录编号;$A_1,A_2,\cdots,A_N$ 表示 R 上的属性,$dom(A)$ 表示属性 A 的定义域,R 上的所有属性集合记作 $attr(A)$,$|attr(A)|$ 表示集合中的属性个数.

2.1 规则约束

定义 1. 内容相关的条件函数依赖(CCFDs)[4,24]. CCFDs 是一种可以同时对多个条件之进行一致性检测的完整性约束.关系 R 上的 CCFD 记作

$$\varphi:(C|Y\rightarrow A,Sc=\bigcup Sc_i), \tag{1}$$

其中,C 为条件属性集合,Y 为变量属性集合,C 和 Y 由"|"分隔,并且 $C,Y\subset attr(R)$,$C\cap Y=\varnothing$,C,Y 合称为规则左部,记作 $lhs(\varphi)$,单属性 A 称为规则右部,记作 $rhs(\varphi)$;$Y\rightarrow A$ 是一个标准函数依赖;Sc_i 是关于 C 的属性值集合,$Sc_i\subset dom(C)$,Sc 是 Sc_i 的集合,即 $Sc=\bigcup Sc_i$.并且,对于任意元组 t_i,t_j,如果 $t_i[C],t_j[C]\in Sc_i$,t_i 和 t_j 需要放在一起进行检测.图 2 中的 φ_T 就是模式 T 上的一条 CCFD 规则.

形式化语义:I 是 R 上的一个实例,I 是满足 φ 的当且仅当:对于 $\forall t_i,t_j\in I$,在 $t_i[C],t_j[C]\in Sc_i$ 的情况下,如果 $t_i[Y]=t_j[Y]$,则有 $t_i[A]=t_j[A]$,并且记作 $I\models\varphi$;否则,记作 $I\not\models\varphi$.Σ 是 CCFDs 规则集合,对于 $\forall\varphi\in\Sigma$,都有 $I\models\varphi$,则称 I 是满足 Σ 的,记作 $I\models\Sigma$.

定义 2. 条件包含依赖(CINDs)[3].CINDs 是一种可以对异构模式进行属性匹配的规则约束.关于模式 R_a,R_b 的 CIND 记作

$$\eta:(R_a[X;X_k]\subseteq R_b[Y;Y_k],T_k=\bigcup t_k), \tag{2}$$

其中,$X,X_k\subset attr(R_a)$,$X\cap X_k=\varnothing$,同理有 Y,$Y_k\subset attr(R_b)$,$Y\cap Y_k=\varnothing$;$R_a[X]\subseteq R_b[Y]$是一个标准包含依赖;T_k 是条件属性集合的模板,$T_k=\bigcup t_k$,t_k 是条件属性值的对应情况,$t_k=(t_k[X_k]\parallel t_k[Y_k])$,$t_k[X_k]\in dom(X_k)$,$t_k[Y_k]\in dom(Y_k)$,使用"$\parallel$"进行分割.图 2 中的 η 就是关于 T,S 的一条 CIND 规则.

形式化语义:I_a,I_b 是 R_a,R_b 上的实例,(I_a,I_b)是满足 η 的当且仅当:对于 $\forall t\in I_a$,$\forall s\in I_b$,$\exists t_k\in T_k$,如果 $t[X_k]=t_k[X_k]$,$s[Y_k]=t_k[Y_k]$,则有 $dom[X]\subseteq dom(Y)$,记作$(I_a,I_b)\models\eta$.否则,记作$(I_a,I_b)\not\models\eta$.Γ 是 CINDs 规则集合,对于 $\forall\eta\in\Gamma$,都有$(I_a,I_b)\models\eta$,则称 I_a,I_b 是满足 Γ 的,记作$(I_a,I_b)\models\Gamma$.

2.2 异构模式下 CCFDs 发现问题

本文主要针对异构关联数据中的一致性规则发现问题进行研究,使用 CINDs 对 CCFDs 进行异构模式下的规则发现,其问题描述如下:

定义 3. 异构模式下 CCFDs 发现问题.给定关系模式 R_a,R_b 下的实例 I_a,I_b 以及对应的 CINDs 规则集合 Γ,找到模式融合 $R_a\oplus_\Gamma R_b$ 上所有 CCFDs 的最小性规则集合 Σ.

1) 异构模式融合并不简单等同于模式和实例的合并,具体的异构融合过程 $R_a\oplus_\Gamma R_b$ 将在 3.1 节中进行介绍.

2) 对于属于相同模式$(C|Y\rightarrow A,Sc)$的候选合并条件值 c_1,c_2,c_3,假设 c_1 可以分别与 c_2,c_3 进行合并形成 $Sc_i=\{c_1,c_2\}$,$Sc_i'=\{c_1,c_3\}$,但是不能同时与 c_2,c_3 一起合并.并且,由于记录中会出现不能同时满足 Sc_i 和 Sc_i' 的情况,进而造成检测死锁.因此,本文不会对候选条件值进行重复合并,即对于 $\forall Sc_i,Sc_i'\in Sc$,满足 $Sc_i\cap Sc_i'=\varnothing$.

3) 对于发现得到的 CCFDs 规则集合需要满足 Armstrong 最小性和规则数量最小性[4]要求.

本文对 CCFDs 规则的可满足性、蕴含性和可验证性 3 个基本问题进行分析.

1) 可满足性是指对于给定的实例 I 以及 Σ(满足 $I\models\Sigma$),对于 $\forall\varphi\in\Sigma$,φ 是否能被验证,进而 Σ 是否能被验证.

2) 蕴含性是指给定 Σ 和任意 φ,Σ 的表示结果是否能够涵盖 φ.如果 Σ 蕴含 φ,记作 $\Sigma\models\varphi$.

3) 可验证性是指对于给定的被验证的完全实例 $\tilde{I}$($\tilde{I}$ 中不含任何错误并且包含所有的正确情况)和 Σ,Σ 是否能满足 $\tilde{I}$.可验证性可以检查规则集合 Σ 中的规则是否存在错误的情况.

定理 1. CCFDs 规则的可满足性、蕴含性和可验证性问题分别属于 NP-complete, coNP-complete, PTIME.

证明. CCFDs 是在 CFDs 的基础上,对关联数据的条件值进行合并得到的一种特殊情况,这里以 CCFDs 规则可满足性为例进行说明.Fan 等人在文献[25]中使用合取范式 $\gamma:X_1\vee X_2\vee\cdots\vee X_n$(其中,$X_i$ 都是由小项合取得到的)对 MAXGSAT(maximum generalized satisfiability)问题[26]进行规约,证明 CFDs ϕ 的可满足性问题是属于 NP-complete 的.具体地,MAXGSAT 是 MAXSAT(maximum satisfiability)的一般问题,描述了最大可满足性问题的一般形式,

被证明是满足 NP-complete 的.其中,$X_1,X_2,\cdots,X_n\in lhs(\phi)$,$X_i$ 满足 $l_1\wedge l_2\wedge l_3$ 的形式,l_1,l_2,l_3 是 X_i 在条件值 t_p 下的取值情况,X_i 的定义域为 $dom(X_{i,t_p})$,进而判定 $\gamma:X_1\vee X_2\vee\cdots\vee X_n$ 的真值情况.对于 CCFDs 而言,其证明过程满足 CFDs 证明的基本过程,区别在于 l_1,l_2,l_3 是 X_i 在条件值 Sc_i 下的取值情况,X_i 的定义域为 $dom(X_{i,Sc_i})$.这样可以在 $O(|Sc_i|)$时间内将 CCFDs 可满足性问题转化为 CFDs 可满足性问题,其中 $|Sc_i|$ 表示 Sc_i 中包含的条件值的个数.因此,可以把 CCFDs 可满足性问题看作是 CFDs 可满足性的等价问题.所以,CCFDs 规则的可满足性问题是属于 NP-complete 的.同理,可以将 CCFDs 规则的蕴含性问题和可验证问题等价转化成 CFDs 规则的蕴含性问题和可验证性问题,并且 CFDs 规则的这 2 个问题在文献[3]中被证明是分别属于 coNP-complete,PTIME 的.因此,CCFDs 规则的蕴含性问题和可验证问题也分别属于 coNP-complete,PTIME 的. 证毕.

需要说明的是,对于 CCFDs 和 CINDs 混合规则的可满足性、蕴含性、可验证性问题满足文献[3]描述的情况,均属于不确定性问题.

3 异构模式下的 CCFDs 规则发现方法

针对 2.2 节中提出的 CCFDs 发现问题,本节将从异构模式融合、搜索结构和规则发现方法的设计这 3 个方面对规则发现的实现过程进行描述.

3.1 异构模式融合

在异构关系模式中,由于关系模式的差别,一种关系模式上的规则集合很难在另一种关系模式上产生作用.这样一方面会降低规则的复用率和作用效果,另一方面也会影响规则对整体数据进行归纳抽象.因此,首先需要对异构数据进行融合.但是,异构融合不能等同于简单的模式和实例合并.针对这一问题,对于给定异构关系模式 R_a,R_b 下的实例 I_a,I_b 以及 CINDs 规则集合 Γ,本文提出了一种异构模式融合和异构实例融合的概念.

对于 $\eta\in\Gamma$,$\eta:(R_a[X;X_k]\subseteq R_b[Y;Y_k],T_k=\bigcup t_k)$,本文首先给出 R_a,R_b 关于 η 的模式融合形式:

$$R_a\oplus_\eta R_b=(X\oplus_\eta Y,X_k\oplus_\eta Y_k,\\ attr(R_a)-lhs(\eta),attr(R_b)-rhs(\eta)),\quad(3)$$

其中,$X\oplus_\eta Y$(或者 $X_k\oplus_\eta Y_k$)表示属性 X 和 Y(或者 X_k 和 Y_k)融合后组成的新的模式属性,$dom(X\oplus_\eta Y)=dom(X)\cup dom(Y)$,$dom(X_k\oplus_\eta Y_k)$同前.$lhs(\eta)$表示 η 的左部属性集合,即 $lhs(\eta)=X\cup X_k$.这样,原来异构模式中能够通过 η 进行匹配的属性将被放在一起,共同作为融合模式中的一个属性;不能被匹配的属性仍然单独保留在融合后的模式中.并且,对于融合后的属性值,其阈值为 2 个属性的并集.

定义 4. 异构模式融合 $R_a\oplus_\Gamma R_b$.$R_a\oplus_\Gamma R_b$ 表示整个异构模式空间中的模式融合形式,融合过程为

$$R_a\oplus_\Gamma R_b=(\bigcup_{\eta\in\Gamma}X\oplus_\eta Y,\bigcup_{\eta\in\Gamma}X_k\oplus_\eta Y_k,\bigcap_{\eta\in\Gamma}attr(R_a)-\\ lhs(\eta),\bigcap_{\eta\in\Gamma}attr(R_b)-rhs(\eta)),\quad(4)$$

其中,$\bigcap_{\eta\in\Gamma}attr(R_a)-lhs(\eta)$ 表示 R_a 中无法通过 Γ 进行匹配的属性集合.在 $R_a\oplus_\Gamma R_b$ 通过 Γ 将异构空间中的所有属性进行匹配,形成一个统一的融合模式.需要说明的是,模式匹配的过程可能形成 1 对 n 的匹配形式,比如 $R_a[X]$可以分别在 η_1 和 η_2 作用下与 $R_b[Y_1]$和 $R_b[Y_2]$分别进行匹配,那么融合的结果将分别保留为 $X\oplus_{\eta_1}Y_1$ 和 $X\oplus_{\eta_2}Y_2$.对于条件值不同,但是 $R_a[X;X_k]\subseteq R_b[Y;Y_k]$形式相同的 CINDs,也不会出现重复融合的情况.

规则发现问题通常是从实例中对语义关系进行抽象.模式融合会对关系属性进行扩展,这样也会对关系实例产生变化,进而对规则发现产生影响.接下来给出异构实例融合的概念.

定义 5. 异构实例融合 $I_a\oplus_\Gamma I_b$.$I_a\oplus_\Gamma I_b$ 表示在融合模式下合并得到的关系实例,构建过程为

$$I_a\oplus_\Gamma I_b=\begin{cases}I_a[x]+I_b[y],(x,y)\in\bigcup_{\eta\in\Gamma}X\oplus_\eta Y\\ \quad\text{or }(x,y)\in\bigcup_{\eta\in\Gamma}X_k\oplus_\eta Y_k,\\ I_a[x]+I_b[y]^*,(x,y)\in\\ \quad\bigcap_{\eta\in\Gamma}attr(R_a)-lhs(\eta),\\ I_a[x]^*+I_b[y],(x,y)\in\\ \quad\bigcap_{\eta\in\Gamma}attr(R_b)-rhs(\eta),\end{cases}\quad(5)$$

其中,$I_b[y]^*$ 表示对于 R_b 中原本不存在的属性,使用“*”值进行填充.需要说明的是,“*”值是一个特殊值,既不是空值,也不是实值,不能用“*”值来表示数据的真实情况,但是认为“*”值与任何数据都不冲突,即对于 $\forall t,s\in I_a\oplus_\Gamma I_b$,如果 $t[A]=$“*”或者 $s[A]=$“*”,则 $t[A]=s[A]$.这种情况在实际数据中是不允许的.但是,规则发现作为数据分析的一个中间过程,“*”值不会对规则发现造成影响,是可以接受的.表 1 给出了融合后的 $R_a\oplus_\Gamma R_b$ 及 $I_a\oplus_\Gamma I_b$.

Table 1 Fusion Instance $I_a \oplus_\Gamma I_b$ on Fusion Schema $R_a \oplus_\Gamma R_b$

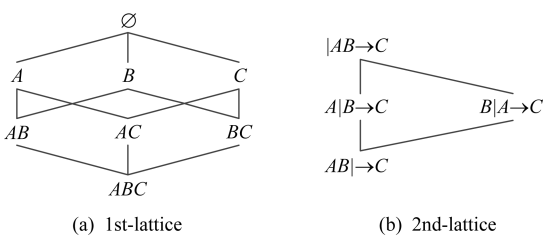
表 1 融合后 $R_a \oplus_\Gamma R_b$ 模式的 $I_a \oplus_\Gamma I_b$

$R_a \oplus_\Gamma R_b$	t_{id}	$Type \oplus Catg$	$Title \oplus Name$	$Pub \oplus Pub$	$Price \oplus Price$	$Type$
$I_a \oplus_\Gamma I_b$	t_1	literature	Different Seasons (Hope Spring Eternal)	Shanghai Culture Publisher	0—50	crime
$I_a \oplus_\Gamma I_b$	t_2	film	Shawshank Redemption	Warner Bros.	0—100	crime
$I_a \oplus_\Gamma I_b$	t_3	literature	Shawshank Redemption	People's Publisher	50—100	*
$I_a \oplus_\Gamma I_b$	t_4	literature	Shawshank Redemption	Shanghai Culture Publisher	50—100	*

3.2 2 级 lattice 结构

函数依赖 FDs、条件函数依赖 CFDs、扩展函数依赖 eCFDs 是进行数据一致性管理的重要技术.在关系模式 R 中,FDs 可以表示为 $\gamma: X \to A$,属性集合 $X, A \in R$.对于 R 上的实例 I,γ 表示 X 在 I 上的属性值可以唯一决定 A 的属性值,即对于 $\forall u, v \in I$,如果 u, v 是一致的,那么有 $u[X]=v[X], u[A]=v[A]$.文献[8]提出了一种用于 FDs 规则发现的 lattice 搜索结构,可以根据属性个数从左部属性和右部属性对规则空间进行划分,直到遍历所有的搜索空间,可以有效对 FDs 进行发现,图 3(a)是一个 lattice 结构实例,其中边(AB, ABC)就表示 FDγ: $AB \to C$ 的规则空间.

对于 CFDs 和 eCFDs,由于在 FDs 规则的基础上,在规则左部中对条件属性进行了划分,在进行规则发现的过程中就需要分别考虑条件属性和变量属性的情况.特别地,CCFDs 是 CFDs 和 eCFDs 的一种特殊情况,规则将 CFDs 和 eCFDs 中具有关联关系的规则进行了合并.基于 lattice 结构的基础原理,本文提出了一种异构模式的 2 级 lattice 结构,如图 3 所示.在进行规则发现时,从 $|AB \to C$ 开始,经过 $A|B \to C, B|A \to C$ 直到 $AB| \to C$ 结束.图 2 中的 ϕ_T 就是在 $Catg, Name| \to Price$ 层次中对条件值进行合并得到的 CCFD.



(a) 1st-lattice (b) 2nd-lattice

Fig. 3 2-level lattice

图 3 2 级 lattice 结构

3.3 CCFDs 规则发现方法

本文使用 $R_a \oplus_\Gamma R_b$ 对异构关系模式进行融合,然后使用 2 级 lattice 结构构建 CCFDs 的搜索空间,对于满足相同形式 $C|Y \to A$ 的条件值进行合并,得到对应的 CCFD.但是,并不是所有的条件值都能进行合并,有的条件值之间是相互冲突的.因此,本文提出了一种用于条件值冲突的判定方法,如定理 2 所示.

定理 2. 一条 CCFD$\varphi: (C|Y \to A, Sc = \bigcup Sc_i)$关于实例 $I_a \oplus_\Gamma I_b$ 是不冲突的当且仅当:

对于 $\forall Sc_i \in Sc$,有

$$\left| \bigcup_{c_i \in Sc_i} \sigma_{C=c_i} \pi_{C \cup Y}(I_a \oplus_\Gamma I_b) \right| = \left| \bigcup_{c_i \in Sc_i} \sigma_{C=c_i} \pi_{C \cup Y \cup A}(I_a \oplus_\Gamma I_b) \right|.$$

其中,$\sigma_{C=c_i} \pi_{C \cup Y}(I_a \oplus_\Gamma I_b)$表示在实例 $I_a \oplus_\Gamma I_b$ 上做条件值为 $C=c_i$ 的选择操作和关于属性 C, Y 的投影操作,$\left| \bigcup_{c_i \in Sc_i} \sigma_{C=c_i} \pi_{C \cup Y}(I_a \oplus_\Gamma I_b) \right|$表示 $\sigma_{C=c_i} \pi_{C \cup Y}(I_a \oplus_\Gamma I_b)$ 中不同值的个数.

证明.首先使用反证法证明定理的充分性.假设 $\left| \bigcup_{c_i \in Sc_i} \sigma_{C=c_i} \pi_{C \cup Y}(I_a \oplus_\Gamma I_b) \right| \neq \left| \bigcup_{c_i \in Sc_i} \sigma_{C=c_i} \pi_{C \cup Y \cup A}(I_a \oplus_\Gamma I_b) \right|$,那么至少存在一对条件值 $c_i, c_j \in Sc_i$ 及对应的元组 $t, s \in I_a \oplus_\Gamma I_b$,使得 $t[C]=c_i, s[C]=c_j$,并且 $t[Y]=s[Y]$,有 $t[A] \neq s[A]$.那么,说明 c_i 和 c_j 是冲突的,不应该放在一个 Sc_i 内,这与题设相悖,所以假设不成立,原命题成立.同理,定理的必要性也可以通过此过程用反证法进行证明.定理的充分性和必要性都是正确的,所以定理 2 成立. 证毕.

需要说明的是,定理 2 从条件值冲突的角度对规则合并进行分析,也可以使用定理 2 对规则是否成立进行判定,进而用于 CCFDs 的规则发现.

对于 2.2 节描述的规则最小性问题.本文研究的条件值合并是在满足 Armstrong 最小性的 CFDs 上进行的,因此合并后的 CCFDs 仍然满足 Armstrong 最小性.文献[4]中证明 CCFDs 规则数量最小性是最大子团的一个等价问题,属于 NP-complete.对于本文所描述的异构关系模式下的 CCFDs 规则发现问题,通过 CINDs 将 R_a, R_b 进行融合成 $R_a \oplus_\Gamma R_b$,其条件属性可以在 PTIME 内转化成 R_a 和 R_b 的条件属性,使得异构模式下的 CCFDs 规则发现的数量最小性问题仍然是一个 NP-complete 问题.

对于给定层次形式 $C|Y \to A$,需要合并候选条件值集合 C 及样本实例 $I_a \oplus_\Gamma I_b$,本文提出一种启发式的条件值合并方法,如算法 1 所示.

算法 1. CCFD 条件值合并方法 h-CCFD.

输入:规则模式 $\mathcal{C}|Y\rightarrow A$、候选条件值集合 $\mathcal{C}$、异构关系实例 $I_a \oplus_\Gamma I_b$;

输出:CCFDφ:$(\mathcal{C}|Y\rightarrow A, Sc=\bigcup Sc_i)$.

① $Sc=\varnothing$;

② while $\mathcal{C}\neq\varnothing$ do

③ $Sc_i=\mathcal{C}.pop()$;

④ for $c_i\in\mathcal{C}$ do

⑤ if $inconflict(Sc_i, c_i, I_a\oplus_\Gamma I_b)$ then

⑥ $Sc_i=Sc_i\bigcup c_i, \mathcal{C}=\mathcal{C}-c_i$;

⑦ end if

⑧ end for

⑨ $Sc=Sc.add(Sc_i)$;

⑩ end while

⑪ return $\varphi:(\mathcal{C}|Y\rightarrow A, Sc=\bigcup Sc_i)$.

其中,行④~⑧用于从 $\mathcal{C}$ 中查找能与 Sc_i 进行合并的条件值,$inconflict(Sc_i, c_i, I_a\oplus_\Gamma I_b)$ 使用定理 2 来验证在 $I_a\oplus_\Gamma I_b$ 上 Sc_i 和 c_i 是否冲突,如果 Sc_i 和 c_i 不冲突,则对 Sc_i 和 c_i 进行合并.算法 1 h-CCFD 的计算复杂度为 $O(|\mathcal{C}|^2)$.

4 实验分析

对于异构模式下关联数据的 CCFDs 规则发现问题,本文通过在 2 组真实数据进行实验来验证方法的有效性和高效性.

4.1 实验设置

硬件方面,本实验使用 Intel Core i5-7400 (3.00 GHz) CPU,搭载 8 GB RAM 主机进行实现,程序设计使用 Java 语言进行实现.

实验数据集,本文使用 NBA 数据和豆瓣数据进行融合,并使用融合后的数据进行规则发现.

1) NBA 数据.队员数据 Players 和赛季比赛统计信息 Stat 是从体育信息数据库[①]上抽取得到的真实数据.Players 数据包含 14 个属性、超过 11 000 的信息记录.Stat 数据包含 11 个属性、超过 200 000 的比赛信息.以球队作为条件设计使用 30 条 CINDs 对 Players 和 Stat 进行融合,例如 η:$Players(player, pos, season, min, drafted, pts, team=\text{“HOU”})\subseteq Stat(player, pos, lea, min, no, pts, tname=\text{“HOU”})$.

2) 豆瓣(Douban)数据.豆瓣电影 Movie 和豆瓣读书 Book 是从豆瓣网站[②]上抽取得到的部分真实数据.Movie 数据中包含 10 个属性,元组数为 50 000 条.Book 数据中包含 12 个属性,元组数为 47 000 条.并且,对 Movie 和 Book 融合后得到的数据包含 62 000 条元组和 14 个属性.

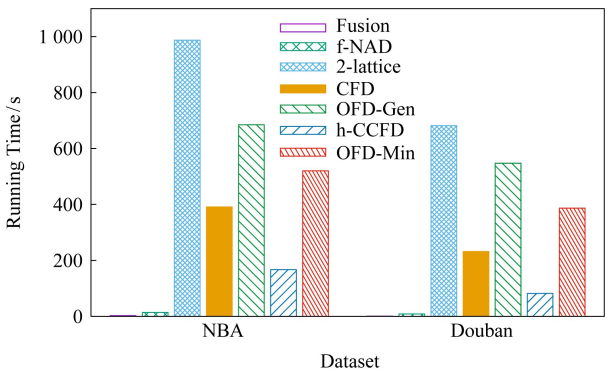
4.2 实验结果及分析

本文从规则发现方法的基本性能和扩展性 2 个方面进行验证和分析.

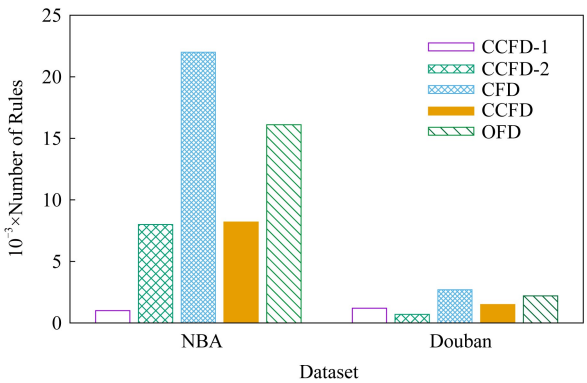
4.2.1 基本性能

首先从运行时间和规则数量对整体性能进行评价,然后给出规则发现的部分实例,其中,豆瓣数据使用合并元组 9 332 条,包含 3 112 条电影实体.

图 4(a)给出了 NBA 和 Douban 数据集中,规则发现过程融合阶段(fusion)、2 级 lattice 下 CFDs 发现阶段(2-lattice)、规则合并阶段(h-CCFD)的运行时间情况.



(a) Running time



(b) Number of rules

Fig. 4 Performance of rules discovery

图 4 规则发现的基本性能

其中,模式和实例的融合时间(fusion)最短,分别为 2.6 s 和 0.8 s,只占整个运行时间的极小比例.同样情况下,使用文献[15]中的规则融合方法(f-NAD)

① https://www.rotowire.com/basketball/

② https://www.douban.com/

需要 14.1 s 和 8.8 s,这是因为 f-NAD 需要对规则中的各个属性进行拆分,然后使用统一的形式进行表达.

CFDs 作为 CCFDs 合并的候选规则.在进行候选规则发现过程的实验中,如 2-lattice 所示,在 2 级 lattice 结构下对 CFDs 规则进行发现的时间最长,相比较文献[2]中使用的 freeset-closedset 方法,对应图 4(a)中的 CFD 方法,时间要超过 1 倍左右.但是,该方法不利于对规则进行 $C|Y\rightarrow A$ 形式划分并用于 CCFDs 合并,所以这里仍推荐使用 2 级 lattice 结构进行 CCFDs 规则发现.OFD-Gen 是在 fusion 的基础上,对每个实体使用 1 级 lattice 方法进行候选规则发现的过程,但是由于实体的数量较多,增加了整体的运行时间.对于 NBA 和 Douban 数据,由于融合的属性个数基本相同,所以构成的 2 级 lattice 空间的大小也基本相同.

在对规则进行合并的过程中,h-CCFD 在 2-lattice 结构发现 CFDs 的基础上,使用启发式方法对候选条件值进行合并,可以在较短时间内完成.OFD-Min 在进行规则发现的过程中首先对实体进行判定和识别,然后根据实体的内容、属性及相互关系进行规则发现,需要大量的运行时间,在 NBA 和 Douban 数据上分别需要 520s 和 387s,是 h-CCFD 方法的 3.8 和 6.2 倍.并且,OFD 方法(OFD-Gen 与 OFD-Min 之和)要高于 CCFD 方法(2-lattice 与 h-CCFD 之和)分别为 6%和 25%,具体会受到实际数据分布的影响.

图 4(b)给出了 NBA 和 Douban 数据中规则发现的数量情况.其中,CCFD-1 和 CCFD-2 指在 2 个数据集中分别在未融合情况下进行规则发现的情况,比如 NBA 中 CCFD-1 指单独在 Players 数据中发现 CCFD 的规则数量,CCFD-2 指单独在 Stat 数据中发现 CCFD 的规则数量.其中,NBA 数据集中,CCFD-1,CCFD-2、CCFD 合并得到的规则数量分别为 1 009,8 028,8 238 条,并且有 878 条来自 Players 和 Stat 的规则合并到了一起,合并的比率为 10.7%.在 Douban 数据集中,合并的数量是 425 条,比率为 28.3%,这是因为合并的情况会受到实际数据分布的影响.

从 CFD 对 CCFD 的规则数量情况来看,由于 NBA 中各个队伍的整体差别不是很明显,所以各个队伍可以合并在一起,并且平均每 2.7 条规则就可以合并成 1 条规则.特别地,现实生活中条件值的合并情况由数据的实际分布情况来决定.OFD 使用同义词进行实体识别,将同一指向的规则合并到一起,分别在 NBA 和 Douban 上得到 16 132 条和 2 237 条规则,但是合并的效果不如 CCFD 明显.

下面给出在实验过程中发现的部分规则实例.NBA 数据集中发现的规则实例分别为 CCFD ϕ_1:$(player\oplus player \mid team\oplus tname\rightarrow drafted\oplus no, Sc)\ Sc\{Sc_0\{\text{“Jordan”},\text{“MJ”},\text{“Air Jordan”}\}\}$ 和 OFDρ_1:$(player\oplus player \mid team\oplus tname\rightarrow_{\{\text{'Jordan'},\text{'MJ'}\}} drafted\oplus no)$.同样,*Douban* 数据集中发现的规则实例分别为 CCFD ϕ_2:$(Title\oplus Name \mid Type\oplus Catg\rightarrow Writer\oplus Writer, Sc)\ Sc\{Sc_0\{\text{“Shawshank Redemption”},\text{“Different Seasons”}\}\}$ 和 OFDρ_2:$(Title\oplus Name \mid Type\oplus Catg\rightarrow_{\{\text{'月黑高飞','肖申克的救赎'}\}} Writer\oplus Writer)$.

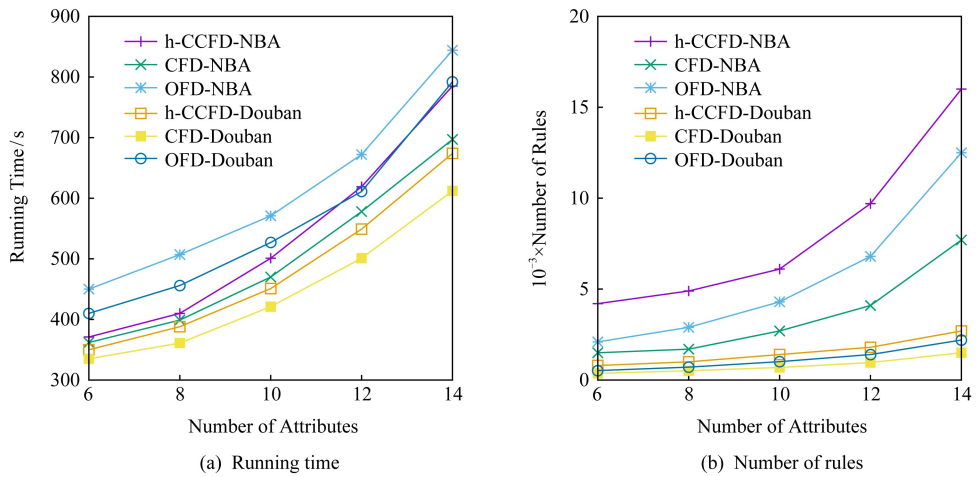
4.2.2 可扩展性

本节将从模式属性和元组数量 2 个方面考量规则发现方法的可扩展性.具体地,通过分别改变属性个数和元组个数,观察运行时间和规则数量的变化情况.

1) 属性数量变化.Douban 数据集使用 9 332 条合并元组,包含 3 112 条电影实体.图 5 给出了属性数量变化情况下规则发现过程的时间变化和规则数量的变化情况.CFDs 和 h-CCFD 的运行时间和规则数量都随属性数量呈指数增长趋势,这是因为 2 级 lattice 结构是一种随属性个数而呈指数变化的结构.OFD 中虽然使用的是 1 级 lattice 结构,但是需要对每一个实体都进行识别和规则发现,所以整体运行时间更长.

另外,图 5(b)所示的规则数量指数增长缓慢,是因为新增属性之间关系较弱,不易形成新的规则约束,这是属性结构和数据分布所决定的.特别地,规则数量呈现指数增长,这反映了规则发现的实际计算情况.但是,这种增长的速度过快,不是一个有利因素,尤其是对于大数据计算和实时计算而言,这种计算代价难以接受.因此,还需要针对这一问题设计剪枝方法、线性代价的近似方法,甚至亚线性方法.

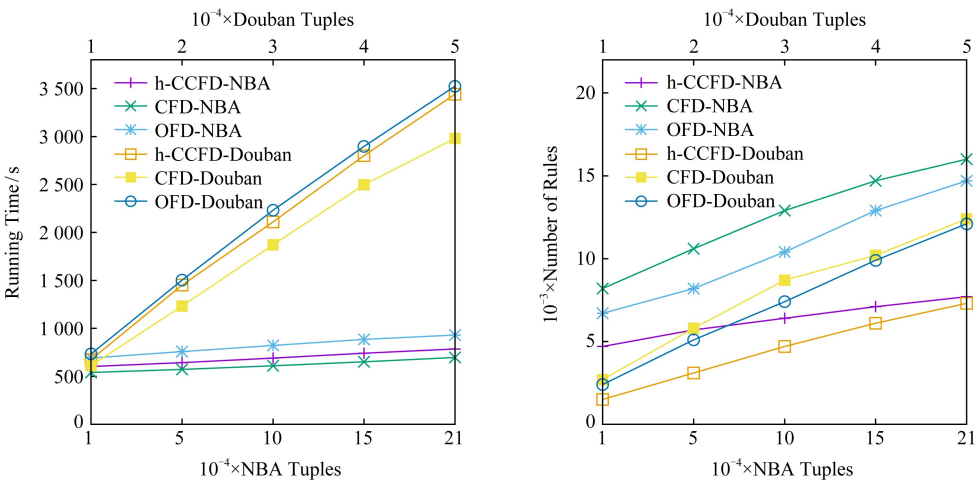
2) 元组数量变化.图 6 给出了元组数量变化情况下规则发现过程的时间变化和规则数量的变化情况.CFDs,h-CCFD,OFD 的运行时间和规则数量都随属性数量呈线性增长趋势,而且变化比较平缓,这是因为在随着元组数量增加的过程中,元组中包含的不同值也在均匀增加.当 NBA 数据大于 150 000,Douban 数据大于 40 000 时,随着元组数量的增加,新增元组所包含的不同值的数量却没有明显增加,所以规则数量增加的情况也会变得更加平缓.



(a) Running time (b) Number of rules

Fig. 5 Scalability on attributes

图 5 属性变化情况下的可扩展性



(a) Running time (b) Number of rules

Fig. 6 Scalability on tuples

图 6 元组变化情况下的可扩展性

通过上述实验,说明本文提出的方法可以有效对异构数据进行融合,并且可以通过数据关联关系快速、准确地对 CCFDs 进行规则发现.

5 结束语

本文通过对异构关联数据的一致性问题进行分析研究,提出一种异构模式下一致性规则约束的发现方法,并对规则发现的可满足性、蕴含性和可验证性问题进行了分析.具体地,本文使用 CINDs 对异构模式进行融合,提出一种基于 2 级 lattice 结构的 CCFDs 规则发现方法,最后通过实验验证了本文所提方法的有效性和高效性.

参 考 文 献

[1] Fan Wenfei, Geerts F. Foundations of Data Quality Management [M]. San Rafael, CA: Morgan & Claypool Publishers, 2012

[2] Fan Wenfei, Geerts F, Li Jianzhong, et al. Discovering conditional functional dependencies [J]. IEEE Transactions on Knowledge and Data Engineering, 2010, 23(5): 683-698

[3] Ma Shuai, Fan Wenfei, Bravo L. Extending inclusion dependencies with conditions [J]. Theoretical Computer Science, 2014, 515: 64-95

[4] Du Yuefeng, Shen Derong, Nie Tiezheng, et al. Discovering context-aware conditional functional dependencies [J]. Frontiers of Computer Science, 2017, 11(4): 688-701

[5] Ding Xiaoou, Wang Hongzhi, Zhang Xiaoying, et al. Association relationships study of multi-dimensional data quality [J]. Journal of Software, 2016, 27(7): 1626-1644 (in Chinese)
(丁小欧, 王宏志, 张笑影, 等. 数据质量多种性质的关联关系研究[J]. 软件学报, 2016, 27(7): 1626-1644)

[6] Ma Shuai, Duan Liang, Fan Wenfei, et al. Extending conditional dependencies with built-in predicates [J]. IEEE Transactions on Knowledge and Data Engineering, 2015, 27(12): 3274-3288

[7] Fan Wenfei, Lu Ping. Dependencies for graphs [C] //Proc of the 36th ACM SIGMOD-SIGACT-SIGAI Symp on Principles of Database Systems. New York: ACM, 2017: 403-416

[8] Liu Qiao, Zhong Yun, Liu Yao, et al. Consistent collective entity linking algorithm [J]. Journal of Computer Research and Development, 2016, 53(8): 1696-1708 (in Chinese)
(刘峤, 钟云, 刘瑶, 等. 基于语义一致性的集成实体链接算法[J]. 计算机研究与发展, 2016, 53(8): 1696-1708)

[9] Huhtala Y, Karkkainen J, Porkka P, et al. Efficient discovery of functional and approximate dependencies using partitions [C] //Proc of the 14th SIGMOD Int Conf on Data Engineering. Los Alamitos, CA: IEEE Computer Society, 1998: 392-401

[10] Zhong Ping, Li Zhanhuai, Chen Qun. A functional dependencies checking method in relational data [J]. Chinese Journal of Computers, 2017, 40(1): 207-222 (in Chinese)
(钟评, 李战怀, 陈群. 关系数据中函数依赖检测方法[J]. 计算机学报, 2017, 40(1): 207-222)

[11] Chiang F, Miller R J. Discovering data quality rules [J]. Proceedings of the VLDB Endowment, 2008, 1(1): 1166-1177

[12] Fan Wenfei. Discovering graph functional dependencies [C] //Proc of the 2018 Int Conf on Management of Data. New York: ACM, 2018: 427-439

[13] Baskaran S, Keller A, Chiang F, et al. Efficient discovery of ontology functional dependencies [C] //Proc of the 2017 ACM Conf on Information and Knowledge Management. New York: ACM, 2017: 1847-1856

[14] Ma Qian, Gu Yu, Zhang Tiancheng, et al. A heterogeneous multi-source multi-model sensory data acquisition method based on data quality [J]. Chinese Journal of Computers, 2013, 36(10): 2120-2131 (in Chinese)
(马茜, 谷峪, 张天成, 等. 一种基于数据质量的异构多源多模态感知数据获取方法[J]. 计算机学报, 2013, 36(10): 2120-2131)

[15] Dallachiesa M, Ebaid A, Eldawy A, et al. NADEEF: A commodity data cleaning system [C] //Proc of the 2013 ACM SIGMOD Int Conf on Management of Data. New York: ACM, 2013: 541-552

[16] Chu Xu, Ilyas I F, Krishnan S, et al. Data cleaning: Overview and emerging challenges [C] //Proc of the 2016 SIGMOD Int Conf on Management of Data. New York: ACM, 2016: 2201-2206

[17] Chung Y, Krishnan S, Kraska T. A data quality metric (DQM): How to estimate the number of undetected errors in data sets [J]. Proceedings of the VLDB Endowment, 2017, 10(10): 1094-1105

[18] Schelter S, Lange D, Schmidt P, et al. Automating large-scale data quality verification [J]. Proceedings of the VLDB Endowment, 2018, 11(12): 1781-1794

[19] Lehmberg O, Bizer C. Stitching Web tables for improving matching quality [J]. Proceedings of the VLDB Endowment, 2017, 10(11): 1502-1513

[20] Ma Shuai, Hu Renjun, Wang Luoshu, et al. An efficient approach to finding dense temporal subgraphs [J]. IEEE Transactions on Knowledge and Data Engineering, 2019, 30(1): 1-14

[21] Ma Shuai, Hu Renjun, Wang Luoshu, et al. Fast computation of dense temporal subgraphs [C] //Proc of the 33rd IEEE Int Conf on Data Engineering. Los Alamitos, CA: IEEE Computer Society, 2017: 361-372

[22] Rammelaere J, Geerts F. Explaining repaired data with CFDs [J]. Proceedings of the VLDB Endowment, 2018, 11(11): 1387-1399

[23] Ortona S, Meduri V V, Papotti P. RuDiK: Rule discovery in knowledge bases [J]. Proceedings of the VLDB Endowment, 2018, 11(12): 1946-1949

[24] Du Yuefeng, Shen Derong, Nie Tiezheng, et al. Discovering condition-combined functional dependency rules [C] //Proc of Asia-pacific Web Conf. Berlin: Springer, 2014: 245-257

[25] Fan Wenfei, Geerts F, Jia X, et al. Conditional functional dependencies for capturing data inconsistencies [J]. ACM Transactions on Database Systems, 2008, 33(2): 1-48

[26] Alimonti P. New local search approximation techniques for maximum generalized satisfiability problems [J]. Information Processing Letters, 1996, 57(3): 151-158

Du Yuefeng, born in 1986. PhD, lecturer. Member of CCF. His main research interests include data quality management, data semantics analysis, etc.

Li Xiaoguang, born in 1973. PhD, professor. Member of CCF. His main research interests include data mining, machine learning, graph-data analysis, etc.

Song Baoyan, born in 1965. PhD, professor, PhD supervisor. Senior member of CCF. Her main research interests include database theory, data-stream theory, big data management, etc.