软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software,2015,26(10):2596−2613 [doi: 10.13328/j.cnki.jos.004798]

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

基于信息元的模式匹配方法*

杜小坤[1], 李国徽[2], 王江晴[1], 帖 军[1], 李艳红[1]

[1](中南民族大学 计算机科学学院,湖北 武汉 430074)

[2](华中科技大学 计算机科学与技术学院,湖北 武汉 430074)

通讯作者: 杜小坤, E-mail: hustdxkun@163.com, http://www.scuec.edu.cn

摘 要: 结构信息是模式匹配的重要辅助信息,当模式中出现多个自身信息相似的元素时,结构信息是正确区分其匹配关系最有效的依据,这在匹配大型模式时显得尤为重要.已有的研究成果对结构信息的使用存在信息不够准确、缺少有效的描述形式、处理耗时等缺点,极大地阻碍了结构信息的使用.为了充分利用结构信息,提出一种基于信息元的模式匹配方法(IU_Based),该方法首先将模式元素按照描述实体的不同划分为不同的信息元,然后计算信息元间的相似度并获取其匹配关系,最后在相互匹配的信息元之间选择元素匹配关系.实验结果表明,IU_Based 方法能够有效地解决结构信息使用中的相关问题,提高匹配准确率.

关键词: 模式匹配;结构信息;结构优化;信息元匹配

中图法分类号: TP311 **文献标识码**: A

中文引用格式: 杜小坤,李国徽,王江晴,帖军,李艳红.基于信息元的模式匹配方法.软件学报,2015,26(10):2596−2613. http://www.jos.org.cn/1000-9825/4798.htm

英文引用格式: Du XK, Li GH, Wang JQ, Tie J, Li YH. Schema matching method based on information unit. Ruan Jian Xue Bao/Journal of Software, 2015,26(10):2596−2613 (in Chinese). http://www.jos.org.cn/1000-9825/4798.htm

Schema Matching Method Based on Information Unit

DU Xiao-Kun[1], LI Guo-Hui[2], WANG Jiang-Qing[1], TIE Jun[1], LI Yan-Hong[1]

[1](College of Computer Science, South-Central University for Nationalities, Wuhan 430074, China)

[2](School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan 430074, China)

Abstract: Structure information is one of the most important types of auxiliary information in schema matching. When a schema has multiple elements with same semantics, the structure information is the most effective information to get correct matching for these elements. This is especially important in big-scale schema. Existing schema matching methods have some weaknesses such as inaccurate structure information, lack of effective description form, and high time complexity in the utilization of structure information therefore greatly hindering the use of structure information. In order to fully use the structure information, a new schema matching method based on information unit(IU_Based) is proposed in this paper. In IU_Based, the elements are first grouped in different information units according to the entity described. Then, the similarity between information units is calculated and the matching relation between information units is obtained based on the similarity. Finally, the matching between elements is selected from the matched information units. Extensive simulation experiments are conducted and the results show that IU_Based method can solve the problems in the use of structure information and take full advantage of structure information to improve the accuracy of match result.

Key words: schema matching; structure information; structure optimization; information unit matching

模式匹配主要研究异构数据源间的数据转换问题,广泛应用于数据空间、数据集成、语义 Web 等热点研

* 基金项目: 国家自然科学基金(61173049, 61309002); 湖北省自然科学基金(2014CFB915)
收稿时间: 2014-04-24; 修改时间: 2014-08-12, 2014-10-09; 定稿时间: 2014-11-19; jos 在线出版时间: 2015-02-02
CNKI 网络优先出版: 2015-02-02 15:22, http://www.cnki.net/kcms/detail/11.2560.TP.20150202.1522.004.html

究领域,是信息广度扩展的重要手段.模式匹配经过十多年的研究,目前已取得丰富的研究成果[1,2],这些成果以模式元素为对象,利用模式中的各种信息(以元素自身信息为主,同时包含模式结构信息和数据实例信息等)挖掘元素语义并计算元素间的相似度,最后依据相似度选取元素匹配关系.这种以元素为基本操作对象的匹配策略对小型模式匹配任务具有很好的匹配效果,但对大型模式匹配任务完成的效果却并不理想,严重影响了模式匹配算法的实用性.随着信息化的不断深入,数据库模式规模逐渐扩大(现有的数据库中有上百个表,数千个属性都是比较常见的),该问题变得日益突出.如何提高模式匹配结果的准确率(特别是大型模式)成为相关研究领域中的一个重要问题.

结构信息反映元素所处的环境,是其语义的重要组成部分.同一元素在不同的环境中可能呈现出不同的语义,例如描述名称信息的元素“Name”可能出现在同一模式的多个不同关系中,其具体语义与所处环境密切相关,若要获取该元素的准确匹配关系,对结构信息的合理使用至关重要.已有研究提出了多种利用结构信息辅助提高匹配准确率的方法,这些方法利用元素的相互关联(函数依赖、属于同一关系等)描述结构信息,一定程度上缓解了上述问题,但也存在较为明显的缺点,下面以图 1 为例进行介绍.

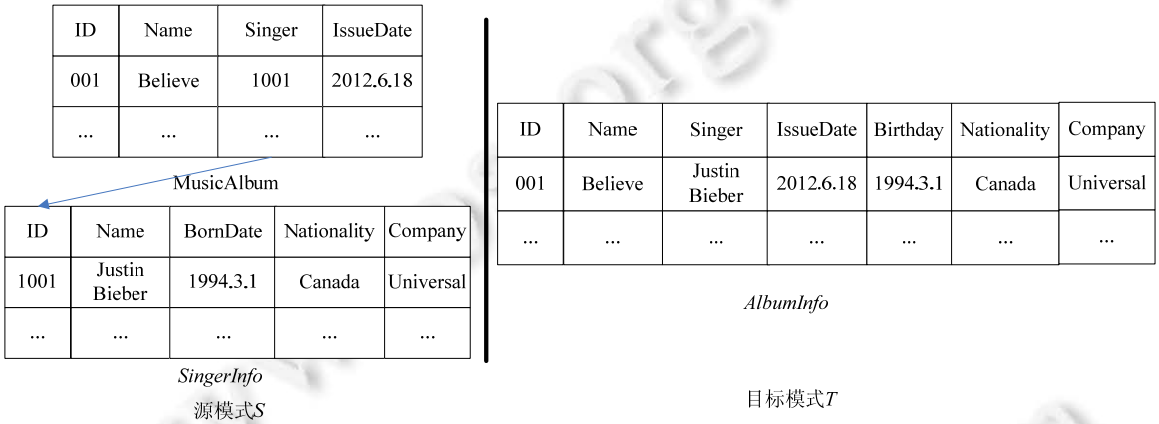


Fig.1 Source schema S and target schema T

图 1 源模式 S 和目标模式 T

(1) 实体是现实世界中可区别于其他对象的一个“事件”或“物体”,现实世界是由一组称作实体的基本对象以及对象间的联系构成.由于数据库模式是现实世界在信息世界中的反映,所以模式结构的实质是模式中实体间的相互关联,如图 1 中源模式 S 的结构是歌手信息和音乐专辑这两个实体的相互关联,已有方法通过元素关联(例如通过元素间的函数依赖关系集$\{ID\rightarrow Name, ID\rightarrow Singer,\ldots\}$)难以准确、直观地描述实体关联(结构信息).

(2) 元素间可能的关联数是元素个数的平方,不仅如此,与元素直接关联的多个元素间的间接关联也是结构信息处理需要考虑的因素,所以在进行大型模式匹配时,利用元素关联描述所进行的结构信息处理其时间复杂度较高(SF 算法的最大缺点即是时间效率低下).

(3) 虽然在规范化理论的指导下,针对相同对象建立的模式结构基本相同,但不同结构的优化目的和手段导致模式结构存在一定的差异.如图 1 中模式 S 和 T 都针对歌手信息和音乐专辑实体建立模式,但为了提高歌手和专辑信息相互查询的速度,模式 T 的设计人员将两个实体的属性放到同一关系中,这将导致直接从模式 T 中抽取的元素关联关系不够准确(无法根据关联关系识别出该关系包含两个实体).所以直接从模式中获取的结构信息往往不够准确.

实体是模式中信息组织的基本单元,模式元素则用来描述实体某一方面的信息.若能将模式元素按照描述实体的不同划分成不同的分块,然后利用实体间的相互关联来描述模式的结构信息,则能有效地解决上述问题.据此,本文提出一种基于信息元的模式匹配方法(一个实体对应的所有属性构成模式的一个基本信息单元,简称信息元).该方法首先通过预处理操作将模式元素依据描述实体的不同划分到不同的信息元中,然后计算信息元间的相似度并获取其匹配关系,最后在相互匹配的信息元中间选取元素匹配关系(由于本方法采取的策略是尽

量为每个信息元选取匹配,所以仅适用于相似数据模式的匹配,对互补数据源暂不支持).

本文的主要创新点在于:

(1) 提出了分块模式匹配的新思路,通过分块操作将元素划分到不同的信息元中,利用信息元间的关联能够更准确地描述结构信息,提高处理效率.同时,用户对信息元匹配结果进行干预的效率也更高.

(2) 提出一种合理的模式划分方法,模式设计时一般会将描述同一实体的元素放到同一关系中,但也存在较多的异常情况,合理地加以划分是分块匹配思路成功的前提.

(3) 提出一种高准确率的信息元匹配算法,信息元包含多个元素,如何提取信息元的语义并获取准确的匹配关系是分块匹配思路成功的关键.

本文第 1 节介绍相关的研究成果并引出新的方法.模式划分预处理算法和基于信息元的匹配算法分别在第 2 节、第 3 节加以介绍.第 4 节对本文提出的方法与已有方法进行实验对比.第 5 节是结论与展望.

1 相关工作

模式匹配的主要目标是获取高准确率的元素匹配关系,目前已有的研究成果基本都从如下 3 个方面着手:① 获取新的辅助信息;② 优化辅助信息的使用方式;③ 提供高效的用户干预手段.下面我们分别对其进行介绍.

1.1 获取新的辅助信息

在获取新的辅助信息方面,Rahm 等人在文献[1,2]中分别对最新的研究成果进行了总结,其中多数研究成果[3−6]以元素自身、模式结构、数据实例等信息为依据选取匹配关系.

典型的元素自身信息有元素名称、数据类型、说明信息等,对元素名称、说明信息等文本类信息首先进行一致性处理(消除其中缩写、简写,发现同义、近义词等),然后利用启发式方法计算文本间的相似度,并以此作为元素自身信息的相似度,对数据类型则利用各种数据类型间的兼容性表示其相似性.自身信息是元素最基本的信息,是选取元素匹配关系时最重要的辅助信息.当模式规模逐步增大时,模式内部元素自身信息相似的情况较为普遍,结构信息的作用日益重要.目前使用结构信息辅助模式匹配的方法主要有 Cupid 方法[3]、SF 方法[7]和 PFD_Based 方法[5]等.

(1) Cupid.该方法首先对元素名进行一致性处理,并根据其中的关键词对元素进行分类,对相互兼容的类别间的元素计算语义相似度(能够减少语义相似度计算的工作量);然后以语义相似度及元素间关联为基础计算元素的结构相似度并利用相关元素结构相似度相互影响的原理对结构相似度进行调整,最后将语义相似度和结构相似度,综合后选取相似度值大于某一阈值的元素对作为匹配结果.该方法利用树型结构描述模式元素间的关联,对结构信息进行了有效的利用,但元素间的结构关联仅限于同属于一个关系的各个元素,结构信息较为单一,未充分利用.

(2) SF.该方法以关系模式中的各种信息(包括模式元素、关系、数据类型等多种信息)及其关联为基础建立模式结构图,首先计算图中节点的语义相似度,然后采用相似度传递算法,使得关联节点的语义相似度相互影响,直至相似度趋于稳定(即传递前后相似度变化值小于给定阈值)或传递次数超过设定值 N,最后选取对应信息为元素的节点相似度,并以此为基础选取匹配关系.该方法深入挖掘模式中丰富的结构信息,但是,由于建立的图结构中包含模式的所有信息(包括元素、元素的数据类型等信息),这些信息在图中都以节点的形式出现并参与匹配,图结构复杂且包含较多的无效节点,算法的时间复杂度极高,不利于大型模式的匹配.

(3) PFD_Based.该方法利用元素间的函数依赖关系描述结构信息,并通过对元素数据进行分析发掘隐含的依赖关系以丰富模式的结构信息.首先以一定量的数据实例为基础计算元素间隐含的函数依赖关系,然后建立元素依赖图,并以语义相似度为基础统计关联元素的相似性以获取元素的结构相似度,最后将结构相似度与语义相似度相结合选取匹配关系.该方法利用函数依赖关系描述结构信息,并从数据实例信息中获取隐藏的依赖关系,丰富了结构信息的内容.但其仍然以元素间的关联表示结构信息,处理算法时间复杂度高,用户干预困难.

自身信息和结构信息都会随着设计习惯和设计目的的不同而发生变化,但数据实例信息是模式中较为稳

定的信息,不会随着设计人员、设计目的的不同而改变.若能发现模式间的相同数据,则可据此推导出对应元素的匹配关系,DUMAS 方法[6]即是利用该特征来搜索元素的匹配关系.但待匹配模式间可能不存在重复数据,所以数据实例信息的使用存在较大的局限性.

除了上述较为直观的信息外,研究人员还发现一些其他类型的信息可用于辅助匹配.申德荣等人提出的SKM 模型[8]利用模式结构信息以及同领域其他模式间的匹配信息获取元素匹配关系.Elmeleegy 等人[9]提出了一种利用查询日志辅助获取元素匹配的 Usage_based 方法,该方法以元素在查询语句中出现的位置信息为依据辅助匹配,Ding 等人[10]对此作了进一步的研究.Pinkel[11]提出一种利用本体来辅助获取模式元素间复杂匹配关系的方法.

1.2 优化信息使用方式

除了挖掘新的辅助信息外,还可以对已有信息的使用方式进行优化.为了综合多种信息提高匹配准确率,Li 等人提出一种利用神经网络综合各种信息的 SEMINT 方法[4],首先选取一个待匹配的模式对神经网络进行训练(每个元素的各种信息输入后,神经网络能够判断该元素与其自身匹配),然后从另一模式中任选一个元素,将其相关信息输入后,神经网络自动为该元素选取合适的匹配关系.该方法处理每个匹配任务都需要对神经网络进行训练,时间复杂度较高.Aumueller 等人[12]提出了一种可动态配置各种不同类型信息匹配器的 COMA++模型.如何对 COMA++模型中的各种匹配器进行配置对匹配结果至关重要,同时几乎所有的匹配器都需要由用户设定各种参数,参数设置的优劣会对匹配结果产生较大影响,Peukert 等人[13]阐述了上述问题,并给出了相应的自动配置(self-configuring)模型,取得了较好的效果.Rahm 等人[14]提出一种根据模式原始结构(辅以人工操作)将模式分块,首先获取块间的匹配关系,然后根据块匹配关系选取元素对应关系的方法.但根据原始结构得到的分块结果不够准确,从而影响了最终结果的准确率.

1.3 高效的用户干预手段

由于模式自身表达能力的缺陷,模式中的各种辅助信息并不能够完全准确地描述模式语义,所以自动模式匹配算法并不能保证获取完全正确的匹配结果,高效、方便的人工干预方式也是目前的一个研究重点.用户浏览所有的元素匹配关系,选择其中的错误匹配进行修改是用户干预的常规手段,这种方式不仅要求用户十分熟悉模式,而且效率低下.为了提高用户干预的效率,Bonifati 等人提出了 SPICY 方法[15],首先为每个元素获取多个候选匹配元素,然后从中任意选取多个可能的全局匹配方案进行实际数据转换,并自动地对转换结果进行评价,用户根据评价结果选择局部最优的匹配方案后再重复上述过程,直至选择出用户满意的结果.当元素数量及每个元素的候选匹配较多时,可能的全局方案数量非常大,需要用户参与的次数增多,算法耗时.Zhang[16]等人提出借助用户对现实世界的敏感直觉来辅助匹配的思路.针对匹配过程中的不确定因素生成一些简单问题,通过 CrowdSourcing 平台向用户提问,再根据用户的回答对映射结果作相应的修正并继续提问,直至获取满足要求的映射结果;但 CrowdSourcing 平台无法保证用户回答的正确性,并且问题的提取是一项较为困难的工作.黄少滨等人[17]提出的 PVMM 模型将专家的手工干预过程放到匹配算法之前,通过专家提供的少量准确的匹配关系大幅度提高全局匹配的准确率;但缺少用户对最终结果的检验,无法保证得到高准确率的匹配结果.

本文提出一种基于信息元的模式匹配方法,首先识别并还原源、目标模式中的结构优化以消除结构差异;然后将元素依据描述实体的不同划分到不同的信息元中,计算信息元间的相似度并获取其匹配关系,最后在相互匹配的信息元间选取元素匹配关系.通过消除结构差异的预处理操作,使结构信息更准确,同时利用信息元间的关联关系所描述的结构信息更直观、处理算法效率更高.由于已有的模式匹配方法对小型模式具有极高的匹配准确率,所以准确的信息元匹配关系意味着准确的元素匹配关系,并且用户的干预可提前至信息元匹配阶段,由于信息元数量相对元素来说较少,所以能够有效地降低用户干预的劳动强度.具体算法流程如图 2 所示.

本文中使用了较多的符号,表 1 给出一些主要符号的说明.

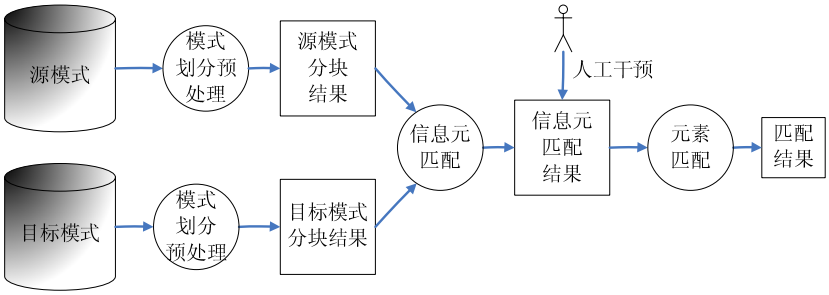


Fig.2 Flow diagram of IU_BASED method

图 2 基于信息元的模式匹配方法流程图

Table 1 Introduction of the main symbol in this paper

表 1 文中主要符号说明

符号	说明	符号	说明
DOM(*A*)	属性 *A* 的值域	*ISim*(IU_1,IU_2)	信息元 IU_1 和 IU_2 的内部相似度
RedundantSet(*B*)	属性 *B* 的冗余属性集	*OSim*(IU_1,IU_2)	信息元 IU_1 和 IU_2 的外部相似度
CAND(*e*)	元素 *e* 的候选匹配集	*SOSim*(*IUS*,*IUT*)	信息元集合 *IUS*,*IUT* 的外部相似度
CAND(*IU*)	信息元 *IU* 的候选匹配	*P*(IU_1,IU_2)	信息元 IU_1 和 IU_2 的相似概率

2 模式划分预处理

模式由实体及实体间的关联信息构成,在范式理论指导下,设计人员通常会将描述同一实体的元素放到同一关系中(满足 3NF),所以按照元素所属关系的不同进行划分具有一定的合理性(Cupid 方法即利用属于同一关系的元素间存在关联的方式来描述结构信息,取得了较好的效果).但结构优化也是模式设计的一个必要环节,不同的优化目的和优化方法对模式结构会产生不同的影响,所以完全依据原始结构进行划分是不准确的,需要综合考虑结构优化的影响.文献[18,19]中介绍了常见的结构优化策略及具体方法,主要措施有:① 增加冗余列;② 增加派生列;③ 合并表;④ 重复表;⑤ 分割表.这些优化措施主要用于使模式结构产生如下形式的变化:① 元素冗余;② 纵向合并;③ 横向分割.当然,除了上述常用的手段及相应的结构变化外,可能还有一些其他手段及相应的模式结构变化,本文仅以上述 3 种主要形式为例进行介绍.由于不同结构变化的识别及处理不存在冲突,所以对其他形式的结构变化可根据具体需求添加相应的识别及处理方法.下面首先介绍在结构变化识别中发挥重要作用的部分函数依赖概念,然后分别介绍不同形式结构变化及相应的识别和处理方法.

2.1 部分函数依赖

若要准确识别模式结构的变化,不仅要知道变化后的结构,还要获取变化前的结构信息.变化后的模式信息可从待匹配的模式中直接获取,如何获取变化前的结构信息呢?模式元素间的依赖关系是元素对应数据的固有性质,不会随着模式结构的变化而变化,这使得利用模式数据获取变化前的结构信息成为可能.Berzal 等人[20]对此进行了较为深入的研究,下面介绍相关的概念.

定义 1. 对关系 r 中任意两个属性集 X,Y,我们称满足如下条件的元组集合 $r_e \subset r$ 为关系 r 的函数依赖例外集:

(1) ($r-r_e$)中所有元组满足 $X \to Y$.

(2) $\forall t \in r_e$, $(r-r_e)\cup\{t\}$ 中的元组不都满足 $X \to Y$.

(3) 不存在 $r'_e \subset r$ 满足条件(1)和(2),并且 $\#(r'_e)<\#(r_e)$ ($\#(r)$ 表示关系 r 中的元组数).

同时,将 r_e 中的元组数称为关系 r 上部分函数依赖例外数,记为 $\exp_e(X \to Y)=|r_e|$;将 $|r-r_e|$ 与总元组数的比值称为部分函数依赖度,记为 $pfd_e(X \to Y)=|r-r_e|/|r|$.

根据上述定义,以数据实例为基础可计算元素间的部分函数依赖度.数据实例的规模对依赖度值的计算有较大的影响,我们前期的研究成果[5]针对数据集规模进行了相关实验,本文参照该实验结果选取数据实例规模

为 3 000 个元组,并选取大于给定阈值ω(ω值设定为 0.98)的部分函数依赖关系描述模式的结构信息.对图 1 中的模式 T,计算其中任意两个元素间的函数依赖度,见表 2,根据表中数据,选取阈值ω为 0.98 后得到描述其结构信息的函数依赖集:

$$F_T=\{ID\rightarrow(Name,Singer,IssueDate,Birthday,Nationality,Company),Name\rightarrow(ID,Singer,IssueDate,Birthday,Nationality,Company),Singer\rightarrow(Birthday,Nationality,Company)\}.$$

Table 2 The pfd between elements of schema T in Fig.1

表 2 图 1 中模式 T 的元素间部分函数依赖度

	ID	Name	Singer	IssueDate	Birthday	Nationality	Company
ID		1	1	1	1	1	1
Name	1		1	1	1	1	1
Singer	0.14	0.14		0.14	1	1	1
IssueDate	0.64	0.64	0.64		0.64	0.64	0.64
Birthday	0.13	0.13	0.98	0.14		0.01	0.01
Nationality	0.01	0.01	0.02	0.01	0.01		0.74
Company	0.01	0.01	0.02	0.01	0.01	0.11	

2.2 属性冗余

为了提高查询的执行速度,设计人员会将某些属性重复存储以避免费时的连接操作.如在图 3 所示的目标模式 T 中,为了提高根据专辑名查询歌手姓名的查询速度,设计人员在描述音乐专辑信息的关系 *MusicAlbum* 中添加了属性 *SName*,该属性是描述歌手信息的关系 *SingerInfo* 中属性 *Name* 的重复,这种情况称为模式中的属性冗余.模式匹配时,上述两个元素应匹配源模式 S 中的同一元素,但已有的模式匹配方法却未考虑该情况,为其分别选取匹配元素,这显然会造成错误的匹配.

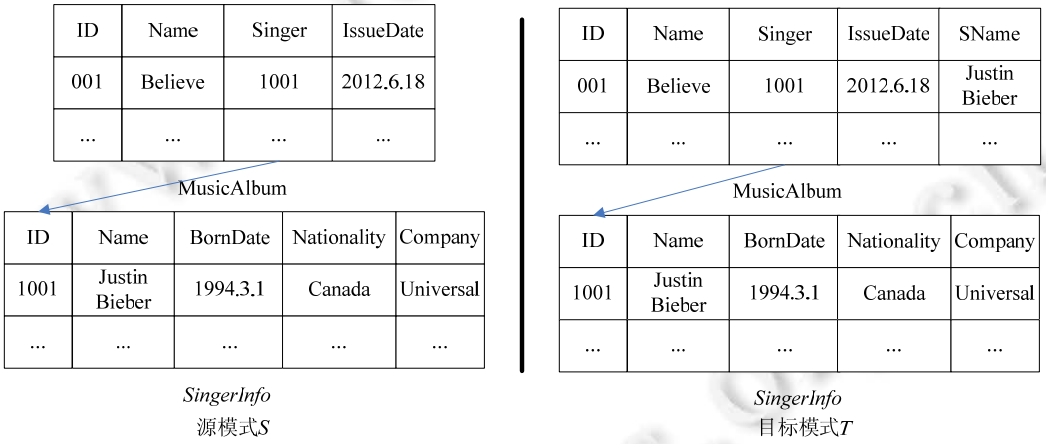


Fig.3 Redundant attributes of schema structure

图 3 模式结构的冗余属性

定义 2. 对同一模式中存在的函数依赖关系 $A\rightarrow B, A\rightarrow B'$, 若对 $DOM(A)$中任意值 x,属性 B 和 B'存在满足下列情况之一的对应值 y 和 y':

(1) 若 y=*null*,则 y'=*null*;

(2) 若 y!=*null*,则 y'=*null* 或 y'=y;

则称依赖关系 $A\rightarrow B, A\rightarrow B'$ 相同,且 B'是 B 的冗余属性,B 是 B'的源属性,源属性 B 可能存在多个冗余属性,我们把 B 的所有冗余属性的集合记为 *RedundantSet*(B).

根据冗余属性的定义,该方法选取模式中所有部分函数依赖度为 1 的依赖关系以检查其中的冗余属性.冗余属性的判断过程需要考虑如下两个问题:数据集规模和算法执行效率.对数据集规模而言,选取与计算部分函数依赖度相同大小的数据集.根据冗余属性与其对应源属性数据间的等值关系,仅需对个别元组进行判断即可

排除大量的属性.所以,在判断冗余属性时,首先仅做少量的数据连接,当发现属性间的值满足冗余属性的条件时才进行所有数据的连接判断,从而提高了冗余属性的识别效率.

冗余属性是源属性的重复,有着相同的匹配元素.对识别出的冗余属性,在模式匹配时可采取简单的删除策略,不参与模式匹配的过程,获取源属性的匹配关系后,为对应的冗余属性选取相同的匹配元素.

2.3 纵向合并

设计人员经常采用的另一种结构优化手段是将多个相互关联的实体合并到同一关系中,这样能够显著提高相互查询的响应速度,这种结构变化称为模式的纵向合并.图 1 中,模式 T 将音乐专辑和歌手信息存储到同一个关系中,虽然降低了添加和修改操作的效率,但显著提高了音乐专辑和歌手信息互相查询的速度.在模式匹配时,目标模式 T 中关系 *AlbumInfo* 的所有属性具有基本相同的结构信息,若其中存在自身信息相似的元素,则难以有效识别.本文利用文献[21]中介绍的改进合成法进行分解,以消除纵向合并优化操作,具体如算法 1 所示.

算法 1. Schema_Decompose(R).

输入:关系 $R(U,F)$,F 为关系 R 中元素满足的函数依赖关系集;

输出:关系 R 的分解 ρ.

{

1.函数依赖集F极小化处理(去除冗余依赖,获取最小覆盖).

2.将F中未出现的属性单独构成一个关系,并从U中删除.

3.如有F中存在函数依赖$X\rightarrow A\in F$且$XA=U$,则算法终止,否则进行第4步.

4.对F按相同左部原则分组,每组的全部属性为一个属性集U_i,若生成的某一属性集U_i被其他属性集U_j包含,则去掉U_i,否则令F_i为F在U_i上的投影,每对$\langle U_i,F_i\rangle$构成一个分解后的关系模式.

5.在第4步得到的模式分解结果$\rho=\{R_1(U_1,F_1),R_2(U_2,F_2),\ldots,R_n(U_n,F_n)\}$中判断是否存在某个$U_i$使得关系$R$的主码$X\in U_i$,若存在,则返回$\rho$为分解结果,否则,返回$\rho'=\rho\cup R\times\langle X,FX\rangle$.

}

以图 1 的模式 T 为例,第 1 步中对 T 中的函数依赖集 F_T 进行极小化处理,获取 F_T 的最小函数依赖集为

$$F_{TM}=\{ID\rightarrow Name,ID\rightarrow Singer,ID\rightarrow IssueDate,Name\rightarrow ID,Singer\rightarrow BornDate,$$
$$Singer\rightarrow Nationality,Singer\rightarrow Company\}.$$

由于 F_{TM} 不满足第 2 步和第 3 步提出的条件,直接进行第 4 步,将关系划分为

$$\rho=\{R_1(ID,Name,Singer(FK),IssueDate),R_2(Singer,BornDate,Nationality,Company)\}.$$

由于在上述划分中,主键 ID 已经包含在 R_1 中,所以ρ为模式分解的结果,该结果与图 1 所示模式 S 的结构基本一致.

2.4 横向分割

若一个关系中包含较多元组,设计人员可采用横向分割的方法进行优化.如图 4 所示,为提高对歌手信息的查询速度,设计人员将歌手根据其所属唱片公司的不同分割存储到不同关系中,我们把这样一组关系称为相似关系.若直接对其进行匹配,则匹配算法会对每个相似关系的每个元素寻找唯一的匹配,这会导致错误匹配.若能在匹配前正确识别目标模式中的相似关系并采取有效的措施,则能避免匹配结果中的类似问题,提高匹配准确率.

横向分割得到的多个关系间具有基本相同的属性集,相关定义如下.

定义 3. 对模式中任意两个关系 $R_1(e_1,e_2,\ldots,e_n)$和 $R_2(f_1,f_2,\ldots,f_m)$,若满足如下关系:

$$\frac{|\{e_i\,|\,(CAND(e_i)\cap R_2)\neq\varnothing\}|+|\{f_i\,|\,(CAND(f_i)\cap R_1)\neq\varnothing\}|}{m+n}\geqslant\mu,$$

则称 R_1 和 R_2 元素相似.

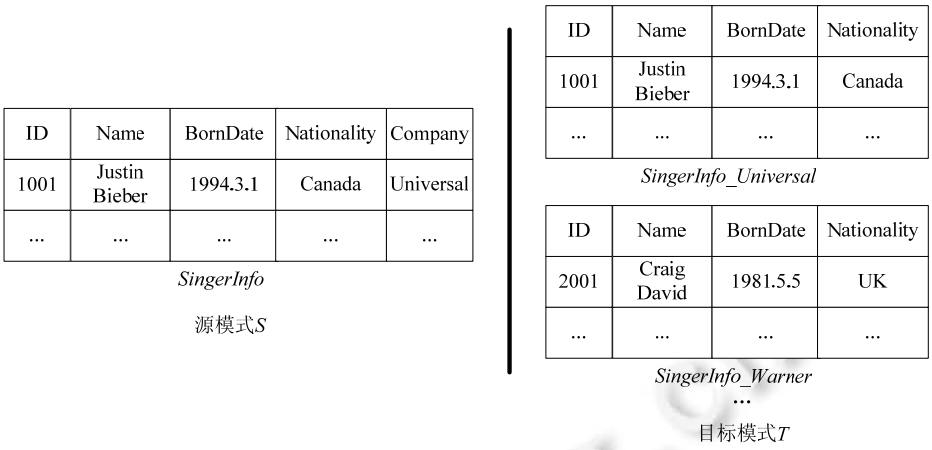


Fig.4 Horizontal partition of schema structure

图 4 模式结构的横向分割

定义 4. 对任意元素相似的关系 R_1 和 R_2,若 R_1 中存在外键 e_i,其对应关系 R'的主键,则 R_2 中一定也存在对应 R'主键(或 R'的相似关系)的外键 f_j,且 $f_j \in CAND(e_i)$,则称 R_1 和 R_2 为相似关系.

算法 2. 横向分割识别(R).

{

1.采用典型的 COMA++匹配方法,计算 R 中元素间自身信息相似度.

2.为 R 中每个元素 e 选取候选匹配集 $CAND(e)$.

3.根据元素相似性获取关系的元素相似度并选取元素相似的关系.

4.根据结构相似规则从元素相似的关系中选取相似关系.

}

算法 2 给出了相似关系的识别算法,由于相似关系具有传递性,可将具有相似性的关系划分为一组.例如图 4 模式 T 中的多个相似关系(*SingerInfo_Universal*,*SingerInfo_Warner*,...)可划分到同一组中.获取模式中的相似关系后,可依据相似关系间元素的对应关系进行关系合并操作,例如可将上述相似关系组合并后得到关系 *SingerInfo*{*ID*,*Name*,*BornDate*,*Nationality*,*Type*}.

本节分析了常见的结构优化对模式结构的影响,并给出了相应的识别算法和处理措施.不同结构变化的识别算法和处理措施之间相对独立,具有较强的可扩展性,在具体运行时只需按照一定顺序依次执行即可.

3 基于信息元的匹配算法

已有匹配方法以元素间关联描述结构信息,这使得结构信息使用困难、处理算法时间复杂度较高.由于预处理后的模式中每个关系对应一个实体,即每个关系中的所有元素构成一个信息元,所以本文以此为依据提出一种以信息元为基本操作单元的模式匹配方法(IU_Based).图 5 是 IU_Based 方法的流程图.

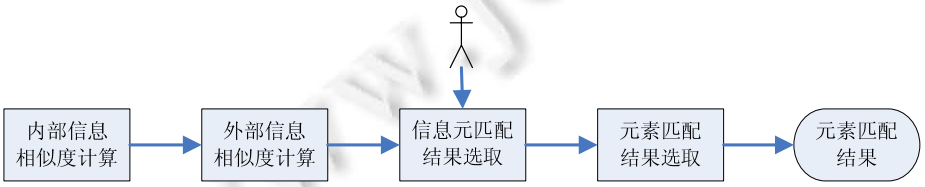


Fig.5 Follow chart of IU_Based method

图 5 IU_Based 方法流程图

该方法首先根据信息元的内部信息计算其内部相似度,然后依据信息元相互关联的结构信息计算其外部相似度,再将内部相似度和外部相似度综合得到信息元的相似概率,最后以相似概率为基础为每个信息元选取匹配关系,并在相互匹配的信息元间获取元素匹配关系.下面我们对 IU_Based 方法的各个步骤进行详细介绍.

3.1 信息元内部相似度计算

信息元是由描述一个实体的所有元素组成的集合,集合中元素的信息构成了信息元的内部信息.本文通过统计集合间元素的相似度以获取信息元的内部相似度.具体步骤为,首先计算信息元间各元素的相似度,然后为目标模式中每个元素选取候选匹配,最后对元素间的候选匹配关系进行统计,得到内部相似度.

对于元素相似度的计算,目前有许多较为成熟的研究成果,这些成果抽取元素的各种语义信息,并以此为依据计算元素相似度.该方法中仅利用元素相似度进行统计以得到内部相似度,且在后续操作中会对结构信息进行处理,所以计算元素相似度时仅考虑元素自身信息,具体采用文献[3]中介绍的语义相似度计算方法,并采用 MaxDelta 和 Threshold 两种策略相结合为每个元素选取候选匹配.

Table 3 The candidate matching of every element in schema *T* of Fig.1
表 3 图 1 模式 *T* 中每个元素的候选匹配

Element in schema *T*	Candidate matching element in schema *S*
$R_1.ID$	*MusicAlbum.ID*,*SingerInfo.ID*
$R_1.Name$	*MusicAlbum.Name*,*SingerInfo.Name*
$R_1.IssueDate$	*MusicAlbum.IssueDate*
$R_2.Singer$	*MusicAlbum.Singer*
$R_2.Birthday$	*SingerInfo.BornDate*
$R_2.Nationality$	*SingerInfo.Nationality*
$R_2.Company$	*SingerInfo.Company*

对图 1 中的待匹配模式 *S* 和 *T*,目标模式 *T* 经预处理后得到两个信息元 R_1(*ID*,*Name*,*Singer*,*IssueDate*), R_2(Singer,*BornDate*,*Nationality*,*Company*).由于 R_1 中属性 *Singer* 是外键,其匹配关系与其对应的主键 $R_2.Singer$ 相同.表 3 所示为图 1 中模式 *T* 与模式 *S* 之间元素的候选匹配关系.

定义 5. 元素的自身语义相似关系存在传递性,即与某个元素自身语义相似的所有其他元素间也存在自身语义相似关系,我们把待匹配模式中与元素 *e* 自身语义相似的所有元素组成的集合称为元素 *e* 的自身语义相似闭包,记为 e^+.

定义 6. 元素 *e* 的自身语义相似闭包中的元素个数称为元素 *e* 的通用度,记为 $generality(e)=|e^+|$.

例如表 3 中元素 $R_1.Name$,其语义相似闭包$\{R_1.Name\}^+=\{R_1.Name,MusicAlbum.Name,SingerInfo.Name\}$,元素 $R_1.Name$ 的通用度为 3.

元素通用度越高,即与该元素自身语义相似的元素越多,则当前环境下该元素对所在信息元的标识性越弱,对相似度的贡献越小;反之,若元素通用度越低,则对所在信息元的标识性越强,对相似度的贡献就越大.例如:信息元 R_1 中同时包含 *Name* 和 *IssueDate* 两个元素,前者的通用度为 3,后者的通用度为 2,这意味着元素 *IssueDate* 比 *Name* 能够更明显地标识其所在信息元的特征.在充分考虑元素通用度的前提下,我们通过如下公式(1)计算信息元 IU_1 和信息元 IU_2 间的内部相似度 $ISim(IU_1,IU_2)$.

$$ISim(IU_1,IU_2)=\frac{\sum_{e\in IU_1} contribute(e)}{|IU_1|} \tag{1}$$

公式(1)中$|IU_1|$表示信息元 IU_1 中元素个数,$contribute(e)$表示元素 *e* 对信息元相似度的贡献值,其值根据公式(2)来设置.

$$contribute(e)=\begin{cases}1/generality(e), & \text{if } cand(e)\cap element(IU_2)\neq\varnothing\\ 0, & \text{else}\end{cases} \tag{2}$$

公式(2)给出了信息元中每个元素的相似度贡献值,通过与公式(1)相结合,可方便地计算出信息元间的内部相似度.具体如算法 3 所示.

算法 3. *Calculate-In-Similarity*(*IU*(*S*),*IU*(*T*)){.

输入:模式 *S* 中所有信息元集合 *IU*(S)={IUS_1,IUS_2,…,IUS_m}

模式 *T* 中所有信息元集合 *IU*(*T*)={IUT_1,IUT_2,…,IUT_n}

模式 *T* 中任一元素 *x* 的候选匹配集 *CAND*(*x*);

输出:*S* 中信息元任一 IUS_i 与 *T* 中任一信息元的 IUT_j 内部相似度 *ISim*(IUS_i,IUT_j).

```
foreach information unit IUS_i in S{
    foreach information unit IUT_j in T{
        ISim(IUS_i,IUT_j)=0;
            foreach element x in IUT_j{
                if(any element y in CAND(x) is in IUS_i)
                    ISim(IUS_i,IUT_j)+=(1/NUM(x));} //NUM(x)为元素x的通用度
                        ISim(IUS_i,IUT_j)=ISim(IUS_i,IUT_j)/|IUS_i|;
}}}
```

该算法遍历源模式和目标模式中每个信息元,根据公式(1)和公式(2)计算每对信息元间的相似度.对图 1 所示的模式运行算法 3 后,信息元间的相似度如下:*ISim*(*MusicAlbum*,R_1)=7/24,*ISim*(*SingerInfo*,R_1)=2/15,*ISim*(*MusicAlbum*,R_2)=1/8,*ISim*(*SingerInfo*,R_2)=3/10.

3.2 信息元外部相似度计算

两个信息元是否相似不仅与其内部信息相关,还与其关联的其他信息元相关,将与该信息元关联的其他信息元称为该信息元的外部信息.本小节计算信息元外部信息的相似度,称为外部相似度.在计算外部相似度之前,使用 MaxDelta 策略根据内部相似度为模式中每个信息元 *IU* 选取候选匹配信息元,所有候选匹配信息元组成集合 *CAND*(*IU*).

定义 7. 假设信息元 IU_1 存在一个以信息元 IU_2 为参照的外键,我们称 IU_1 是 IU_2 的参照元,IU_2 是 IU_1 的被参照元,记为 *IUREF*(IU_1,IU_2).信息元 IU_1 的所有参照元的数目称为 IU_1 的参照元数,记为 *REFERNUM*(IU_1);信息元 IU_1 的所有被参照元的数目称为 IU_1 的被参照元数,记为 *REFERBYNUM*(IU_1).IU_1 的所有参照元和被参照元组成的集合称为 IU_1 的直接关联信息元集,记为 *DRIUSet*(IU_1).

定义 8. 互为候选匹配的信息元 IUS_1,IUT_1,若目标模式中存在参照关系 *IUREF*(IUT_2,IUT_1),同时源模式中也存在参照关系 *IUREF*(IUS_2,IUS_1),且 $IUT_2 \in CAND(IUS_2)$,则称 IUS_1 和 IUT_1 之间存在参照元相似关系.

定义 9. 互为候选匹配的信息元 IUS_1,IUT_1 若目标模式中存在参照关系 *IUREF*(IUT_1,IUT_2),同时源模式中也存在参照关系 *IUREF*(IUS_1,IUS_2),且 $IUT_2 \in CAND(IUS_2)$,则称 IUS_1 和 IUT_1 之间存在被参照元相似关系.

定义 10. 两个信息元间可能存在多个参照元相似关系,称为参照元相似数.同时,也可能存在多个被参照元相似关系,称为被参照元相似数;信息元间的参照元相似数和被参照元相似数之和称为其结构相似数,记为 *SSimNum*(IUS_i,IUT_j).

结构相似数是与信息元直接关联的其他信息元相似情况的统计,以此为基础,根据如下公式(3)可计算出信息元 IUS_i 和 IUT_j 间的外部相似度 *OSim*(IUS_i,IUT_j).

$$OSim(IUS_i, IUT_j) = \frac{SSimNum(IUS_i, IUT_j)}{REFERNUM(IUS_i) + REFERBYNUM(IUT_j)} \tag{3}$$

3.3 外部相似度调整

信息元的外部信息不仅包含与其直接关联的信息元,还包含间接关联的信息元.但第 3.2 节得到的外部相似度仅考虑到直接关联信息元的相似度,不能准确反映信息元外部信息的相似程度.本小节将利用传递调整算法对外部相似度进行调整,使其能够综合反映信息元所有外部信息的相似程度.

在介绍调整算法之前,首先介绍直接关联信息元集合的外部相似度概念.以信息元 IU_i,IU_j 的直接关联信息元集合为例,定义二部图 $G(DRIUSet(IU_i),DRIUSet(IU_j),E)$,$E=\langle(IU_m,IU_n) \mid IU_m\in DRIUSet(IU_i)\wedge IU_n\in DRIUSet(IU_j)\wedge IU_j\in CAND(IU_i)\rangle$,$E$ 中每条边的权值为所关联信息元的外部相似度,即 $OSim(IU_i,IU_j)$.我们使用该二部图的最大流通量表示信息元集合的外部相似度,如公式(4)所示.对公式(4)的右部采用匈牙利算法[22]计算二部图的最大流通量.

$$SOSim(DRIUSet(IU_i),DRIUSet(IU_j))=\max\left(\sum_{IU_m\in DRIUSet(IU_i)}\sum_{IU_n\in DRIUSet(IU_j)}OSim(IU_m,IU_n)\right) \tag{4}$$

得到信息元集合的外部相似度后,根据公式(5)对任意候选匹配对(IU_i,IU_j)的外部相似度进行调整.

$$OSim(IU_i,IU_j)=\alpha\times OSim(IU_i,IU_j)+\beta\times SOSim(DRIUSet(IU_i),DRIUSet(IU_j)),\ \alpha+\beta=1 \tag{5}$$

根据公式(5)对所有候选匹配对的外部相似度进行一次调整称为一个调整周期,若两个调整周期之间所有候选匹配对的外部相似度变化小于阈值λ,说明已调整充分,整个调整过程结束;若经过多次调整仍未达到要求,则调整至第 N 个周期后结束.

上述调整过程兼顾间接关联信息元的相似性,所以得到的外部相似度能够更准确地反映信息元所有外部信息的相似程度.

3.4 匹配关系获取

由于已有的算法针对小型(特别是像信息元这样仅包含一个实体的多个属性)模式具有非常高的匹配准确率,得到准确的信息元匹配关系后,利用已有算法即可快速获取高准确率的元素匹配关系,因此如何选取高准确率的信息元匹配关系对算法的性能至关重要.本小节给出了一种信息元匹配关系的选取算法.

文献[10]介绍了一种重要的匹配关系选取方法:稳定婚姻法.其核心思想是:选择这样一些匹配对,使相似度之和最大,但不存在这样的两个匹配对(x,y),(m,n),x 与 n 的相似度大于 x 与 y 的相似度,同时 y 与 m 的相似度大于 y 与 x 的相似度.但该策略存在一个较为明显的缺点:相似度值的计算都是基于启发式方法,计算出的相似度具体数值并不具有实际的意义,不同匹配对的相似度值并不能够直接进行比较,所以仅根据相似度值的大小进行匹配选取并不一定准确.例如:$Sim(A,B)=0.27$,$Sim(A,C)=0.2$,并不能据此说明 B 与 A 匹配的可能性比 C 与 A 更高,因为除 A 外还可能存在其他元素与 B 和 C 匹配,假设 B 还存在候选匹配 D 与其相似度为 $Sim(D,B)=0.4$,而 A 是 C 唯一的候选匹配,此时 C 与 A 匹配的可能性会更高.

据此,本文使用了相似概率这一概念.相似概率表示每个信息元与其候选匹配能够相互匹配的概率值,不同匹配对的相似概率值具有可比性,有利于稳定婚姻法进行对比选取.

对于目标模式中的任意信息元来说:所有候选匹配的相似度值总和越高,表明该信息元在源模式中存在实际匹配的概率越高;反之则越低.对同一个信息元的不同候选匹配来说,相似度值越高,匹配的概率也就越高;反之则越低.据此,对目标模式中任一信息元 IU 与其候选匹配集 $CAND(IU)$ 中的任一信息元 IU' 间的匹配概率可根据公式(6)计算.

$$P(IU,IU')=\frac{SIM(IU,IU')}{\sum_{IU_i\in CAND(IU)}SIM(IU,IU_i)+d} \tag{6}$$

公式(6)中,d 为参数,在相同情况下,d 越大,IU 存在匹配的概率越小,即算法越悲观;d 越小,IU 存在候选匹配的可能性越大,算法越乐观.分别用内部和外部相似度替换公式(6)中的 $SIM(IU,IU')$,可计算得到目标模式中每个信息元与其候选匹配信息元的内部相似概率(IP)和外部相似概率(OP).然后根据公式(7)将二者综合得到信息元的综合相似概率.

$$Sim(IU,IU')=\frac{\lambda\times OP(IU,IU')+\sigma\times IP(IU,IU')}{2},\ \lambda+\sigma=1 \tag{7}$$

由于不同匹配对的相似概率能够相互比较,以相似概率为基础,利用稳定婚姻法为源模式中每个信息元选取匹配信息元.与元素匹配算法相似,信息元匹配算法自动获取的匹配结果中会出现错误的匹配,人工干预操作

不可避免(用户手工对匹配结果进行直接干预),由于信息元的数量远少于元素数量,所以用户对信息元匹配结果干预的工作量大幅降低.获取信息元间的匹配关系后,由于信息元中包含较少的元素,已有的匹配方法即能获取极高准确率的元素匹配结果.

3.5 IU_Based方法时间性能分析

IU_Based 方法合理地使用模式结构信息辅助匹配,由于分块机制显著降低了匹配算法的问题规模,所以算法的时间复杂度较低.本小节对 IU_Based 方法与 Cupid 方法、PFD_Based 方法的时间性能进行了分析对比(SF 方法的时间性能明显低于后两种方法,所以不参与比较).根据本文第 1.1 节的介绍,3 种方法中共有的核心操作(最耗时的操作)主要有元素自身相似度计算和相似度调整.对元素自身相似度计算,由于 3 种匹配方法都采用相似的处理方法,所以该步骤操作的时间复杂度大致相同,不进行比较.对相似度调整操作,由于调整过程需要进行多遍,且总的调整遍数具有较大的随机性,所以选定一遍调整所需的时间为对象进行分析对比.假定模式中含有 n 个元素,m 个信息元,平均每个信息元有 a 个元素($m\times a\approx n$),信息元间的关联关系总数为 x 且关联关系均匀分布,每个信息元与 $\frac{2x}{m}$ 个信息元关联.

IU_Based 方法中,一遍调整会对其中每个信息元候选匹配对的外部相似度进行调整,具体操作为利用这两个信息元关联的信息元集合的相似度对其进行调整,集合的相似度采用匈牙利算法进行计算.由于每个信息元与 $\frac{2x}{m}$ 个信息元关联,且匈牙利算法的时间与节点数和边数成正比,所以对每个候选匹配对调整的时间复杂度为 $\frac{2x}{m}\times e_1$ (e_1 为二部图的边数且 $e_1\leqslant\left(\frac{2x}{m}\right)^2$).对每个候选匹配对进行一次调整的总时间复杂度为 $m\times\frac{2x}{m}\times e_1$,即 $2xe_1$.

PFD_Based 方法中,一遍调整是对其中每个元素候选匹配对的相似度进行调整.该方法中的元素关联不仅包含完全函数依赖,还包括部分函数依赖,并且其数量具有很大的随机性,所以分析中仅考虑其中的绝对函数依赖.若模式中的信息元关联数为 0,则其中元素间的绝对依赖关系数为 $m\times(a-1)$,增加一个信息元关联,元素关联数至少增加 a 个(增加一个信息元关联,则至少增加如下依赖关系:一个信息元的主键函数决定另一个信息元的所有元素),所以总的绝对依赖关系数为 $m\times(a-1)+xa$,其中每个元素的关联元素个数为 $(m\times(a-1)+xa)/n$.对每个候选匹配相似度的调整算法与 IU_Based 方法相似,其时间复杂度为 $\frac{m\times(a-1)+xa}{n}\times e_2$ (e_2 为二部图的边数且 $e_2\leqslant\left(\frac{m\times(a-1)+xa}{n}\right)^2$),总的时间复杂度为 $(m\times(a-1)+xa)\times e_2$. 假设 $e_1\approx e_2$ (事实上,e_1 总是大于 e_2),则 PFD_Based 方法一遍调整的时间远远多于 IU_Based 方法的调整时间.

Cupid 方法中对相似度调整的过程如下:首先建立模式结构树(可能有多棵),其中非叶子节点对应信息元,叶子节点对应属性,然后对结构树中的任意候选匹配对(s,t),根据公式(8)调整[3]其相似度.其中,$leaves(s)$为模式结构树中以 s 为根的子树的所有叶子节点,$stronglink(x,y)$表示 x,y 的相似度超过给定阈值.若模式中信息元关联数为 0,则模式结构树中的节点关联数有 $m\times a$ 个,每增加一个信息元关联,节点关联数至少增加 a 个(即其中一个信息元与另一个信息元的所有节点连接),所以总的关联数有 $a(m+x)$,平均每个非叶子节点的关联数有 $\frac{a(m+x)}{m}$, 对每对非叶子节点按照公式(8)进行调整的时间复杂度为 $\left(\frac{a(m+x)}{m}\right)^2$, 对所有非叶子节点的相似度进行一次调整的总时间复杂度为 $(a(m+x))^2$. 在一定规模的模式中,某个元素关联的信息元数量远小于整个模式中的信息元关联数,所以 IU_Based 方法关联信息元集合二部图的边数 e_1 小于信息元关联总数 x,即

$$2xe_1<2x^2<a^2(m+x)^2,$$

即 Cupid 方法一遍调整的时间远远多于 IU_Based 方法的调整时间.

$$ssim(s,t)=\frac{\left|\begin{matrix}\{x\mid x\in leaves(s)\wedge\exists y\in leaves(t),stronglink(x,y)\}\\ \cup\{x\in leaves(t)\wedge\exists y\in leaves(s),stronglink(y,x)\}\end{matrix}\right|}{\left|leaves(s)\cup leaves(t)\right|} \tag{8}$$

综上,在几种利用结构信息辅助匹配的方法中,IU_Based 方法一遍调整耗费的时间较少.

4 实验结果分析

为了验证 IU_Based 方法的性能,本节将其与相关方法进行了实验对比.第 4.1 节介绍实验的基本环境;第 4.2 节中对本文各种优化识别算法的准确率进行验证;为了验证预处理操作能否提高匹配结果的准确率,第 4.3 节中对添加预处理的 COMA++方法和原始方法的匹配结果进行对比;为了验证 IU_Based 方法能更准确、高效地利用结构信息,第 4.4 节中将该方法与几种典型的模式匹配方法在匹配准确率及时间性能方面进行对比;第 4.5 节对本文方法降低用户工作量的性能进行验证.

4.1 实验基础

为了验证本文方法对不同规模的模式的匹配效果,选取多个不同规模的匹配任务进行测试.具体见表 4.

Table 4 Basic information of the schema used in experiments

表 4 实验选取模式的基本情况

匹配任务	模式	关系数目	属性数目	备注
SM1	DBS1	12	73	两个小型论坛的后台数据库
	DBT1	14	75	
SM2	DBS2	43	307	两个销售同类产品公司的进销存数据库
	DBT2	38	294	
SM3	DBS3	126	905	两家生产同类型产品的中型制造企业的 ERP 数据库
	DBT3	133	938	

模式中的数据来源于实际的生产数据,若关系中元组数超过 3 000,则取前 3 000 个元组,若不足 3 000,则使用 DTM Data Generator[**]自动生成足够的数据.数据库管理系统为 MySql5.5,使用 ODBC 连接数据库获取各种信息.主键硬件采用 Intel Core i3 双核 2.27G 处理器,4G 内存;操作系统为 Windows 7.在此基础上,对匹配结果采用如下 3 个指标进行评价.

(1) 查准率(precision):匹配结果中正确匹配结果占所有匹配结果的比率.

$$Precision=T/P=T/(T+F).$$

(2) 查全率(recall):匹配结果中正确匹配结果占实际匹配结果的比率.

$$Recall=T/R.$$

(3) 全面性(overall):通过使用匹配算法所节省的工作量占总的匹配工作量的比率.

$$Overall=Precision\times\left(2-\frac{1}{Recall}\right)=\frac{T-F}{R}.$$

其中,T为匹配算法返回的正确匹配结果;P为匹配算法返回的所有匹配结果;F为匹配算法返回的错误匹配结果;R 为所有正确的匹配结果.查准率、查全率和全面性能够比较全面地反映匹配方法的性能,是模式匹配研究中最常用的 3 个评价指标[4].

4.2 优化识别准确率验证

为了验证第 2 节中各种识别算法的识别准确率,我们首先通过人工方式对 DBS3 和 DBT3 中的各种结构优化进行识别,然后利用第 2 节中的识别算法对结构优化进行自动识别,实验结果见表 5.

** http://www.sqledit.com/dg/download.html

Table 5 Identification of structural optimization in DBS3 and DBT3

表 5 DBS3 和 DBT3 中的结构优化识别

	DBS3			DBT3		
	人工识别	正确识别	错误识别	人工识别	正确识别	错误识别
属性冗余	64	62	4	57	54	2
纵向合并	24	23	2	26	26	2
横向分割	5	5	0	7	7	0

通过表 5 中的实验数据可知,本文第 2 节中提出的属性冗余、纵向合并、横向分割识别算法能够有效识别模式中相应的结构优化,具有较高的识别准确率.

4.3 预处理操作必要性验证

为了验证预处理操作对最终匹配结果的影响,设计如下实验:首先分别对匹配任务 SM3 进行如下 3 种不同方式的处理:① 不进行预处理(COMA++);② 自动预处理(PP+COMA++);③ 人工预处理(MANUAL+COMA++);然后将处理后的模式利用 COMA++算法进行匹配,匹配结果如图 6 所示.

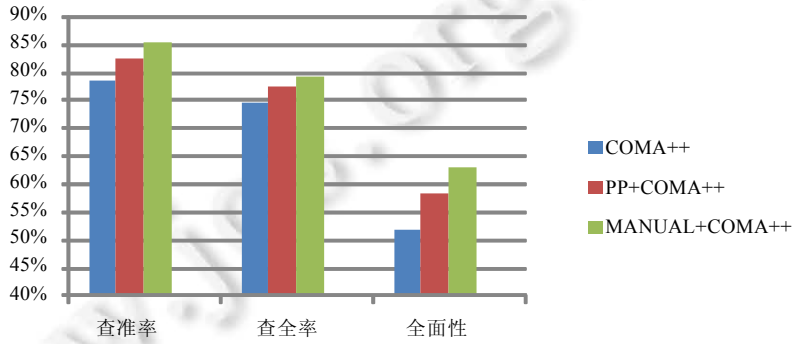


Fig.6 Comparison between COMA++ and COMA++ with preprocess

图 6 进行预处理与不进行预处理匹配结果比较

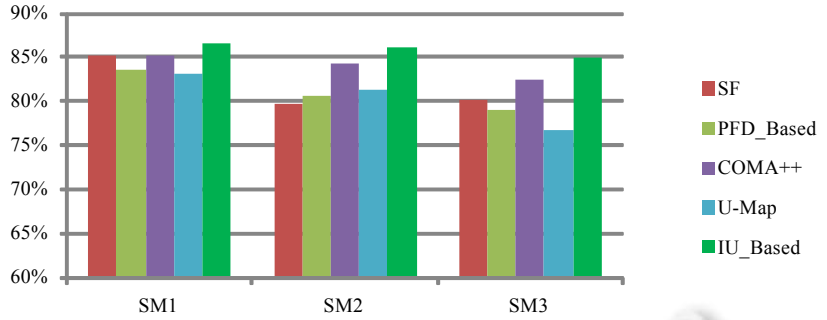
通过图 6 所示的数据可以发现,预处理操作为匹配算法提供了更准确的结构信息,使得匹配结果的各项指标均有较为显著的提高.虽然自动预处理算法中存在错误识别情况,使得其匹配结果准确率低于人工预处理后的准确率,但由于出错的几率较小(见第 4.2 节实验结果),其最终匹配准确率与不进行预处理时相比仍然有着显著的提高.

4.4 IU_Based方法与其他方法比较

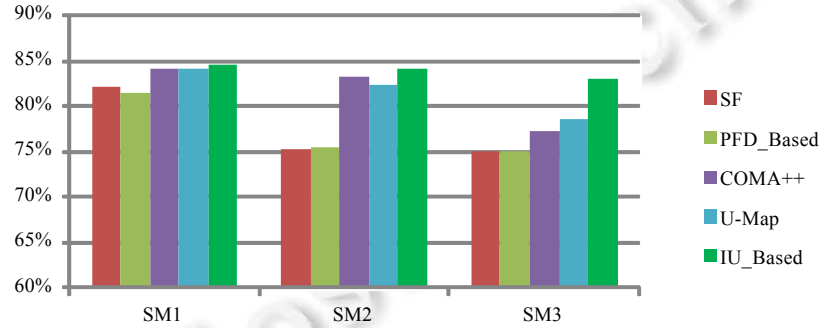
IU_Based 方法利用预处理及分块策略能够更准确、快速地使用结构信息.为了验证该方法在结构信息使用方面的优势,将其与利用结构信息辅助匹配的 SF,PFD_Based 方法在匹配准确率、时间性能方面进行对比.另外,为了验证 IU_Based 方法能够显著提高匹配结果的准确率,将其与典型的 COMA++方法(其中配置分块匹配策略)和 U-Map 方法的匹配准确率进行比较.这些方法分别完成匹配任务 SM1,SM2,SM3 后(其他方法前添加自动预处理操作),匹配结果准确率如图 7 所示.

从图 7 所示的实验结果可知,与其他典型的匹配方法相比,IU_Based 方法在各项指标上均有所提高,并且随着模式规模的逐步变大,IU_Based 方法的优势越来越明显.以查全率为例,对仅包含 10 多个关系的匹配任务 SM1,指标最低的 PFD_Based 方法为 81.6%,而最高的 IU_Based 方法也仅达到 84.6%;而对包含 100 多个关系的匹配任务 SM3,指标最低的 SF 方法为 75.1%,最高的 IU_Based 方法却达到 83.1%,具有明显的提高,所以 IU_Based 方法针对大型模式匹配任务具有更好的效果.其他指标也存在类似特点.

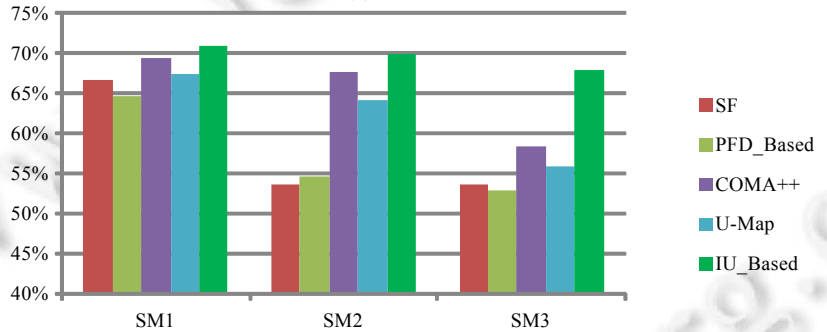
IU_Based 方法与利用结构信息的 Cupid,SF,PFD_Based 方法的运行时间对比如图 8 所示.



(a) 查准率对比



(b) 查全率对比



(c) 全面性对比

Fig.7 Comparison among IU_Based and related methods

图 7 IU_Based 方法与相关方法实验结果对比

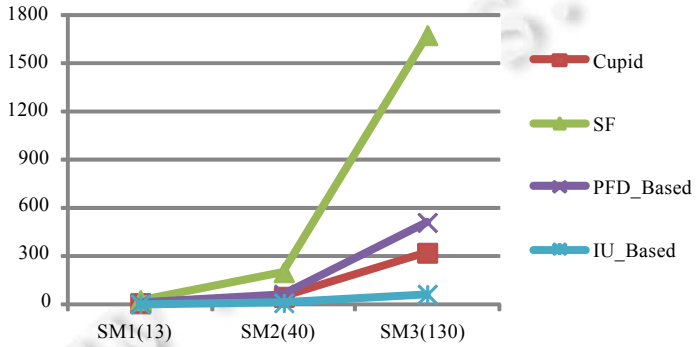


Fig.8 Comparison of time performance among IU_Based and related methods

图 8 IU_Based 方法与相关方法时间性能对比

图 8 中纵坐标为算法运行时间,横坐标为 3 个不同规模模式从小到大的排列.从图中数据可以发现,随着模式规模的扩大,算法执行时间迅速延长.其中最耗时的算法为 SF 方法,最省时的算法为 IU_Based 方法.当处理具有约 13 个信息元的匹配任务 SM1 时,各种方法的执行时间差别并不大;随着模式规模的扩大,处理具有约 130 个信息元的匹配任务 SM3 时,执行时间迅速延长,其中,IU_Based 方法的执行时间最短,约为 64s,与执行时间最长的 SF 算法相差 26 倍.据此可知,IU_Based 方法针对大型模式匹配任务有着较好的时间性能.

综上,在无人工干预的情况下,IU_Based 方法在匹配准确率指标上与其他方法相比具有较为明显的优势,同时在时间性能上也明显优于同类方法,对大型模式尤为如此.

4.5 节省工作量对比

在一些对匹配结果准确率有较高要求的应用环境中,自动匹配算法难以满足应用需求,人工干预不可避免.IU_Based 方法将人工干预提前至信息元匹配关系的选取过程中,能够较大程度地节省用户的工作量.Overall 指标用于衡量为得到完全准确的匹配结果,自动匹配方法节省的用户工作量.为了更准确地衡量 IU_Based 方法在节省工作人员工作量方面的效果,本节使用修正全面性指标 Modi-Overall 来表示节省的用户工作量.

$$Modi\text{-}Overall = \frac{T - F - UF}{R} \tag{8}$$

公式(8)给出了修正全面性指标,与一般全面性指标相比,增加了信息元匹配时用户干预次数(*UF*)的考量(由于操作人员发现信息元匹配错误并修改与发现元素匹配错误并修改在操作时间及难度上并无明显区别,所以这里将其同等对待),能够更准确地反映 IU_Based 方法匹配算法节省操作人员工作量的实际情况(其他匹配方法的 *UF* 值为 0,仍为原有的全面性指标).为了综合评价对不同规模模式的效果,分别对匹配任务 SM1,SM2,SM3 进行匹配,并且参与匹配的模式均经过自动预处理.实验结果如图 9 所示.

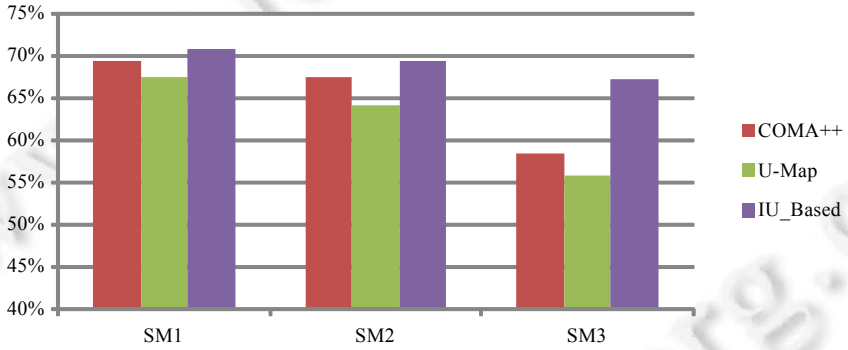


Fig.9 Comparison about modified overall

图 9 修正全面性指标对比

图 9 中给出了 COMA++,U-Map 和 IU_Based 方法分别对不同规模模式匹配结果的修正全面性指标对比情况.从图中数据可知,随着模式规模的扩大,前两种以元素为基本操作对象的匹配方法的全面性指标从 68%左右下降到约 57%,而以信息元为基本操作对象的 IU_Based 方法却仅从 71%下降到 67%,所以对大型模式匹配任务使用 IU_Based 方法效果更好.

5 总结与展望

即使对相同的现实世界对象,不同的设计习惯、设计目的会导致设计的模式存在巨大差异,主要表现为元素自身信息差异和模式结构差异,这些差异是模式匹配问题难以有效解决的重要原因.已有方法对元素自身差异进行了有效处理,但对模式结构差异目前还没有有效的解决方案,导致获取的结构信息不够准确、使用困难等一系列问题.本文分析了其结构差异产生的原因,并提出对模式进行结构优化识别及还原的预处理策略,实验

结果显示,该预处理策略能够有效处理模式结构差异问题,为模式匹配方法提供更准确的结构信息以提高匹配结果准确率.由于预处理策略能够将模式还原为基本满足 3NF 标准的模式,模式中每个关系对应现实世界中的一个实体,本文据此提出一种基于信息元的 IU_Based 匹配方法,首先将模式按照描述实体的不同划分为不同信息元(按照预处理后的关系进行划分),然后根据信息元内部(元素集合)和外部(信息元关联)相似度获取信息元匹配关系,并由用户对匹配结果进行手工干预以获取准确的信息元匹配关系,最后利用已有匹配方法在相互匹配的信息元间获取元素匹配关系(已有匹配方法对小型模式具有极高的匹配准确率).由于信息元间的关联能够更简洁地描述结构信息,所以能够使 IU_Based 方法具备良好的时间性能.同时,由于信息元包含多个元素,所以用户对信息元匹配关系的干预更高效,降低了用户的劳动强度.在后续的研究中,我们将重点关注更准确、高效的结构优化识别方法以获取更为准确的结构信息.同时,由于元素匹配的最终目标是数据映射,所以直接在对应的信息元进行准确、高效的数据映射也是下一步的研究重点.

References:

[1] Rahm E, Bernstein PA. A survey of approaches to automatic schema matching. VLDB Journal, 2001,10(4):334−350. [doi: 10.1007/s007780100057]

[2] Bernstein PA, Madhavan J, Rahm E. Generic schema matching, ten years later. Proc. of the VLDB Endowment, 2011,4(11): 695−701.

[3] Madhavan J, Bernstein PA, Rahm E. Generic schema matching with Cupid. In: Proc. of the VLDB. Roma: Morgan Kaufmann Publishers, 2001. 49−58.

[4] Li WS, Clifton C. SEMINT: A tool for identifying attribute correspondences in heterogeneous databases using neural networks. Data & Knowledge Engineering, 2000,33(1):49−84. [doi: 10.1016/S0169-023X(99)00044-0]

[5] Li GH, Du XK, Du JQ. A structure matching method based on partial functional dependencies. Chinese Journal of Computers, 2010,33(2):240−250 (in Chinese with English abstract). [doi: 10.3724/SP.J.1016.2010.00240]

[6] Bilke A, Naumann F. Schema matching using duplicates. In: Proc. of the 21st Int'l Conf. on Data Engineering. Tokyo: IEEE Computer Society, 2005. 69−80. [doi: 10.1109/ICDE.2005.126]

[7] Melnik S, Garcia-Molina H, Rahm E. Similarity flooding: A versatile graph matching algorithm and its application to schema matching. In: Proc. of the 18th Int'l Conf. on Data Engineering. San Jose: IEEE Computer Society, 2002. 117−128. [doi: 10.1109/ICDE.2002.994702]

[8] Shen DR, Yu EY, Zhang X, Kou Y, Nie TZ, Yu G. SKM: A schema matching model based on schema structure and known matching knowledge. Ruan Jian Xue Bao/Journal of Software, 2009,20(2):327−338 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/3203.htm [doi: 10.3724/SP.J.1001.2009.00327]

[9] Elmeleegy H, Elmagarmid A, Lee J. Leveraging query logs for schema mapping generation in U-MAP. In: Proc. of the 2011 ACM SIGMOD Int'l Conf. on Management of Data. Athens: ACM, 2011. 121−132. [doi: 10.1145/1989323.1989337]

[10] Ding G, Dong H, Wang G. Appearance-Order-Based schema matching. In: Database Systems for Advanced Applications. Busan: Springer-Verlag, 2012. 79−94. [doi: 10.1007/978-3-642-29038-1_8]

[11] Pinkel C. Interactive pay as you go relational-to-ontology mapping. In: Proc. of the Semantic Web—SWC. Sydney: Springer-Verlag, 2013. 456−464. [doi: 10.1007/978-3-642-41338-4_31]

[12] Aumueller D, Do HH, Massmann S, Rahm E. Schema and ontology matching with COMA++. In: Proc. of the 2005 ACM SIGMOD Int'l Conf. on Management of Data. Baltimore: ACM, 2005. 906−908. [doi: 10.1145/1066157.1066283]

[13] Peukert E, Eberius J, Rahm E. A self-configuring schema matching system. In: Proc. of the 28th Int'l Conf. on Data Engineering. Arlington: IEEE Computer Society, 2012. 306−317. [doi: 10.1109/ICDE.2012.21]

[14] Rahm E, Do HH, Maßmann S. Matching large XML schemas. ACM SIGMOD Record, 2004,33(4):26−31. [doi: 10.1145/1041410.1041415]

[15] Bonifati A, Mecca G, Pappalardo A, Raunich S, Summa G. Schema mapping verification: The spicy way. In: Proc. of the 11th Int'l Conf. on Extending Database Technology. Uppsala: Springer-Verlag, 2008. 85−96. [doi: 10.1145/1353343.1353358]

[16] Zhang JC, Tong YX, Chen L, Cao CC. Reducing uncertainty of schema matching via crowdsourcing. Proc. of the VLDB Endowment, 2013,6(9):757−768. [doi: 10.14778/2536360.2536374]

[17] Huang SB, Liu GF, Wan QS, Cheng Y, Shen LS. A schema matching model based on partial verified matching relations. ACTA Automatica Sinica, 2013,39(10):1642−1652 (in Chinese with English abstract). [doi: 10.3724/SP.J.1004.2013.01642]

[18] Dong H, Liu Hj. Research on how to improve the design of documental database. Journal of the China Society for Scientific and Technical Information (in Chinese with English abstract), 1999,18(2):43−49. [doi: 10.3969/j.issn.1000-0135.1999.01.007]

[19] Cui YS, Zhang Y, Zeng C, Feng JH, Xing CX. A survey of database physical structure optimization technology. Ruan Jian Xue Bao/Journal of Software, 2013,24(4):761−780 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/4355.htm [doi: 10.3724/SP.J.1001.2013.04355]

[20] Berzal F, Cubero JC, Cuenca F, Medina JM. Relational decomposition through partial functional dependencies. Data & Knowledge Engineering, 2002,43(2):207−234. [doi: 10.1016/S0169-023X(02)00056-3]

[21] Hao ZX, Lu ZW. The syntheses of the decomposition of relational schema: A further study. Journal of Computer Research and development, 1992,29(8):60−62 (in Chinese with English abstract).

[22] Lu ZN, Zhang HS. Foundation of Operations Research. Hefei: University of Science and Technology of China Press, 2006 (in Chinese).

附中文参考文献:

[5] 李国徽,杜小坤,杜建强.基于部分函数依赖的结构匹配方法.计算机学报,2010,33(2):240−250. [doi: 10.3724/SP.J.1016.2009.00240]

[8] 申德荣,余恩运,张旭,寇月,聂铁铮,于戈.SKM:一种基于模式结构和已有匹配知识的模式匹配模型.软件学报,2009,20(2):327−338. http://www.jos.org.cn/1000-9825/3203.htm [doi: 10.3724/SP.J.1001.2009.00327]

[17] 黄少滨,刘国峰,万庆生,程媛,申林山.一种基于部分已验证匹配关系的模式匹配模型.自动化学报,2013,39(10):1642−1652. [doi: 10.3724/ SP.J.1004.2013.01642]

[18] 董慧,刘厚嘉.文献数据库优化设计的探讨.情报学报,1999,18(1):43−49. [doi: 10.3969/j.issn.1000-0135.1999.01.007]

[19] 崔跃生,张勇,曾春,冯建华,邢春晓.数据库物理结构优化技术.软件学报,2013,24(4):761−780. http://www.jos.org.cn/1000-9825/4355.htm [doi: 10.3724/SP.J.1001.2013.04355]

[21] 郝忠孝,路正午.关系模式分解合成法的进一步研究.计算机研究与发展,1992,29(8):60−62.

[22] 路正南,张怀胜.运筹学基础教程.合肥:中国科学技术大学出版社,2006.

杜小坤(1980－),男,湖北钟祥人,博士,讲师,主要研究领域为模式映射,关键词查询.

帖军(1976－),男,博士,副教授,CCF 会员,主要研究领域为移动数据库,物联网.

李国徽(1973－),男,博士,教授,博士生导师,CCF 高级会员,主要研究领域为主动数据库,实时数据库,移动计算,并行/并发程序同步.

李艳红(1973－),女,博士,副教授,主要研究领域为移动时空数据库.

王江晴(1964－),女,博士,教授,CCF 高级会员,主要研究领域为智能算法,图像处理,模式识别.