基于部分函数依赖的结构匹配方法

李国徽[1)] 杜小坤[1)] 杜建强[2)]

[1)](华中科技大学计算机学院 武汉 430074)

[2)](江西中医学院计算学院 南昌 330006)

摘 要 模式匹配是模式集成、数据仓库、电子商务以及语义查询等领域中的一个难点.它主要利用元素自身信息(如元素名、数据类型等信息)、数据实例信息(模式中的数据)和结构信息(模式元素相互关联的关系)来挖掘元素语义以获得正确的映射关系.文中介绍了一种将数据实例信息与结构信息相结合来辅助匹配的新方法.此方法首先根据模式对应的数据实例信息来计算模式元素间的部分函数依赖度(模式结构信息),然后根据部分函数依赖关系建立模式元素间的依赖图,再根据元素依赖图计算元素间的结构相似度,最后得到模式元素间的映射关系.由于利用了更多的结构信息辅助匹配,所以文中方法在性能上要优于其它仅使用完全函数依赖结构信息进行匹配的方法.实验表明此方法在查准率、查全率以及全面性等各个指标上都优于已有的其它方法.

关键词 模式匹配;部分函数依赖;结构匹配

中图法分类号 TP311 **DOI 号**: 10.3724/SP.J.1016.2009.00240

A Structure Matching Method Based on Partial Functional Dependencies

LI Guo-Hui[1)] DU Xiao-Kun[1)] DU Jian-Qiang[2)]

[1)](*School of Computer Science & Technology, Huazhong University of Science & Technology, Wuhan* 430074)

[2)](*School of Computer Science & Technology, Jiangxi University of Traditional Chinese Medicine, Nanchang* 330006)

Abstract Schema matching is a difficulty in many database application domains, e.g., data integration, E-business, data warehousing and semantic query. We can get correct mapping by mining the semantics of elements from the elements' own information (e.g., elements' names and elements' data types), data instances for elements and structure information. In this paper, we introduce a new algorithm which integrates the data instance information and structure information to supply matching. At first, we calculated the degree of partial functional dependency according to the data instance information, and then we constructed the graph of partial functional dependency (Fig. 3) based on the degree of partial functional dependency, the degree of structure similarity was calculated according to the graph fo partial functional dependency, at last, according to the degree of structure similarity and semantic similarity, the mapping was generated . Because of the more structure information was used, the performance of this algorithm is better than the algorithm only use the complete functional dependency information. Extensive simulation experiments were conducted and the results(Fig. 8, Fig. 9) show that this algorithm is better than other related algorithms in various performance metrics such as precision, recall and overall.

Keywords schema matching; partial functional dependency; structure matching

---

收稿日期:2008-04-15;最终修改稿收到日期:2009-10-12.本课题得到国家"八六三"高技术研究发展计划项目基金(2007AA01Z309)、国家自然科学基金(60873030)、国防预研基金(9140A04010209JW0504、9140A15040208JW0501)及中央高校基本科研业务费专项资金资助.**李国徽**,男,博士,教授,博士生导师,主要研究领域包括主动、实时数据库、移动计算、并行(并发)程序的同步正确性和数据集成等.E-mail: guohuili@hust.edu.cn. **杜小坤**(通信作者),男,1980 年生,博士研究生,主要研究方向为数据集成及模式匹配.E-mail: hustdxkun@163.com. **杜建强**,男,1968 年生,教授,主要研究领域为医学图像处理.

1 引　言

模式匹配是模式间的一个二元操作,它以源模式和目标模式为输入,以两个模式中元素(在关系型数据库中对应于关系的属性)间的映射关系为输出.随着数据库应用的日趋广泛,模式匹配在越来越多的应用领域中发挥着重要作用,如模式集成、数据仓库、电子商务、语义 WEB 和 P2P 数据库等领域.目前的模式匹配工作大都是由操作人员手工进行,这就要求操作人员必须对数据库的模式结构以及每个模式元素的语义都很熟悉,这是一个枯燥、费时且容易出错的工作.随着数据库技术的不断发展,数据库模式逐渐增大.数据库中有数百个关系、数千个属性都是比较常见的,而且它们由不同的设计人员设计,这就使得全面了解数据库的模式结构变得愈加困难,甚至是一个不太可能完成的任务,因此需要一种自动的模式匹配方法来代替费力、费时且容易出错的手工匹配.目前,这方面的研究成果已经相当丰富[1-8],它们分别利用模式中不同类型的信息来挖掘模式元素的语义,然后进行元素匹配.目前利用的信息主要有如下 3 种类型:

(1)元素自身信息.元素自身信息(元素名、数据类型等)是模式中最基本的信息,是元素语义最直观的反映.早期对模式匹配的研究[2,6-7,9]大多是基于元素自身信息.

(2)数据实例信息.数据实例信息是模式描述的对象,所以也能够准确地反映元素语义,但是从大量的数据实例中提取准确的元素语义是一个很困难的过程.文献[10]是这方面的研究成果.

(3)结构信息.模式中元素间的关联关系构成了模式的结构信息,结构信息能够有效地辅助匹配,但缺点是模式中定义的结构信息不够丰富(例如在关系型数据库中只存在元素间的主、外键关系).目前这方面的研究成果主要有文献[11-12].

目前模式匹配的研究中利用的结构信息主要是模式元素间的主、外键关系,它们由设计者在模式设计阶段指定.但主、外键关系并不能全面地反映出模式中元素间的关联关系,因为设计者在设计模式结构时为了满足关系数据库严格的规范化定义,会省略某些关联关系或对其进行修正.如例 1 所示.

例 1.　表 1 是某公司进销存管理系统数据库对供应商信息进行管理的一个关系,它包括供应商编号、名称、地址、电话、联系人、备注等信息.

表 1　供应商信息表

ManufaID (PK)	CompanyName	Address	Telephone	LinkMan	SupType	Remark
A02001	南京通用电器有限公司	南京苜蓿园大街 128 号	025-84855496	黄甘	监控系统	210007
A02002	深圳市新安锦辉电子厂	深圳市宝安 44 区 4 号楼	0755-29961658	毛维金	电子器件	518101
A02003	深圳市宝安区新安金海牛电子厂	深圳市宝安 44 区 4 号楼	0755-27837528	梁鹭	电子器件	518101
A03001	慈溪市华威电子有限公司	慈溪市桥头镇工业区	0574-63550423	毛维金	电子器件	315317
A03002	桂林市兴华探测器有限公司	桂林市施家园路 31-2 号	0773-5825656	石伟	安检门	541004
…	…	…	…	…	…	…

从表 1 可以看出,关系以供应商编号(ManufaID)作为主键,因此属性 ManufaID 能够函数决定其它属性.除此之外,我们不能够发现其它的结构信息(元素间的关联),但通过与该关系的设计人员沟通,我们发现它的各个属性间还存在着如下一些关联关系:

(1)当某供应商不与其它供应商重名时,知道供应商名称就能够知道该供应商的其它信息(事实上某公司的供应商中名字相同的非常少,所以"供应商名称决定供应商的其它信息"对绝大多数供应商来说是适用的).

(2)事实上,供应商的联系电话(Telephone)对于不同的供应商来说是绝对不同的,所以知道了供应商的联系电话,就能够确定是哪个供应商以及该供应商的其它信息.

(3)原则上每个供应商都应该有唯一的联系地址和联系人,但可能存在某些供应商提供的地址不够详细、联系人重名或者同一业务员代理多家供应商产品的情况,所以"联系人姓名决定供应商的其它信息"和"联系地址决定供应商的其它信息"并不对所有供应商都适用.

这些关联关系在以往的研究中被称为部分函数依赖[13],与元素间的主、外键关系一样,这些部分函数依赖也能够有效地支持模式匹配.本文介绍的就是一种利用数据实例信息充分挖掘元素间的结构信息来辅助匹配的新方法,主要有如下创新点:

(1)介绍了一条根据数据实例信息得到结构信息来辅助匹配的新思路.

(2)给出了一种根据元素间部分函数依赖计算结构相似度的新方法.

(3)给出了一种调整结构相似度的新方法.

本文第 2 节介绍相关的研究工作;第 3 节介绍部分函数依赖的概念及如何利用其表示元素间的关联关系;第 4 节介绍基于部分函数依赖的结构匹配方法的具体步骤;第 5 节对本方法进行实验评价;第 6 节为总结与展望.

2 相关工作

模式匹配研究目前成果丰硕[1-8],它们分别利用了模式中不同类型的信息来进行匹配.元素自身信息是模式匹配中使用的最基本信息,早期的模式匹配方法[2,6-7,9]都重点利用了这一信息;模式中包含的数据实例信息也可以辅助匹配,文献[10]介绍的就是一种利用数据实例信息提高匹配准确度的方法;文献[11-12]则介绍了如何利用模式的结构信息来进行匹配;文献[8]中介绍了一种利用数据库查询日志中的查询语句来辅助匹配的方法.由于本文介绍的是利用结构信息(由数据实例计算得到的部分函数依赖关系)来辅助匹配的新方法,因此这里我们简要介绍几种与本文相关的利用数据实例和结构信息进行模式匹配的方法.

DUMAS[10]是一种利用数据实例信息来辅助匹配的方法,它首先利用重复数据检测算法检测出源模式和目标模式中重复(相似)的数据,然后根据重复数据中相同数据对应的元素相同的原理得到互相匹配的元素对,但该方法的难点在于如何准确定位重复的数据.

Cupid[12]方法利用元素自身信息和模式结构信息进行匹配.它基于层次结构的模式,将模式中内部相关联的元素互相连接构成模式树;然后利用元素名、数据类型等元素自身信息计算元素间的语义相似度,并根据得到的语义相似度对元素进行分类;再根据元素的结构信息(模式树中与该元素相连接和邻近的元素与其它元素的语义相似信息)计算元素的结构相似度;最后将每个元素对的元素相似度和结构相似度加权平均得到最终的相似度,并选取最终相似度值最高的元素作为最终的匹配结果.

SF[11]方法首先利用图结构来表示源模式和目标模式,然后利用名称匹配器对两个图结构中的每一对节点计算其语义相似度并根据语义相似度选出所有的候选匹配对,再对候选匹配对的相似度进行调整(由于两个相似节点的相邻节点也相似,所以候选匹配对的相似度受相邻候选匹配对的相似度的影响)得到最终的相似度.文中还给出了几种根据相似度选取匹配结果的不同策略.但该方法的图结构中包含了过多的节点信息,所以具有很高的时间复杂度.

除了文献[11-12]中利用的结构信息(主要是元素间的主、外键关系)外,本文还利用了模式元素间的部分函数依赖关系.本方法首先根据模式元素自身信息计算模式元素间的语义相似度并选取候选匹配对,根据模式的数据实例信息计算模式元素间的部分函数依赖度并选取元素的有效部分函数依赖集;然后建立函数依赖图,再计算候选匹配对的结构相似度并根据相邻节点的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相结合选取最终的匹配结果.由于本方法有效地利用了元素间的部分函数依赖关系,所以匹配效果明显优于其它未使用部分函数依赖关系的方法.

3 部分函数依赖

通常人们思考问题时都会对获取的信息加以一定程度地抽象.例如:人们通常会说"鸟会飞",没有人会对这个命题的正确性产生怀疑,因为通常所见的鸟类都是能够飞行的,当然也有一些特殊的情况:企鹅也属于鸟类,但企鹅却不会飞,但这样的特殊情况并不会导致我们对"鸟会飞"这个命题的正确性产生怀疑.这里我们把这样的一些特殊情况称为该命题的例外.数据库设计时我们也经常会碰到类似的情况,由于关系数据库严格的规范化定义,个别例外就会导致整个命题不正确,从而该命题所表示的信息就不能在关系数据库中反映出来.如例 1 中,对于命题"根据学生姓名就能够知道他的其它信息",由于会存在"学生重名"这样的个别例外情况,所以关系数据库无法表示元素 StuName 同其它元素间的关联关系.Berzal 和 Cubero 等对这种关系进行了研究[13],把元素间的这种关联关系称为部分函数依赖(partial functional dependency),当部分函数依赖中不存在例外时即为通常意义上的函数依赖.下面我们给出部分函数依赖的几个相关定义.

定义 1. r 为关系 R 中的数据实例集(记录构成的集合),$X,Y\subseteq R$ 为两个属性集,我们称 $r_e\subset r$ 为部分函数依赖 $X\mapsto Y$ 的例外元组集合,当且仅当 r_e 满足如下条件时:

(1)$(r-r_e)$ 中所有元组满足 $X\mapsto Y$.

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

$X \mapsto Y$.

(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).

我们称 r_e 中元组的数目为部分函数依赖例外数.

图 1 中,对于数据实例集 r 以及部分函数依赖 $Year \mapsto Course$,当 $Year$="1990"时,$Course$ 可取多个值,此时部分函数依赖 $Year \mapsto Course$ 产生冲突,r_{ae} 中列出了所有产生冲突的元组,为了满足条件(1),当 $Year$="1990"时,属性 $Course$ 只能取集合{4,3,2}中的一个值;为满足条件(2),当 $Course$ 选定某一值 A 后,r_e 中不应包含满足条件 $Year$="1990"和 $Course=A$ 的元组;为满足条件(3),在选择 $Course$ 的取值时,应选择对应元组最多的取值.这里取 $Course$="4",因为 $Course$ 取值为"4"时,对应的元组有 4 个,而取值"3"和"2"分别对应的元组数为 2 个和 1 个.最后 r_{ae} 中剩余的元组即为例外元组,如图 1 中 r_e 所示.

$r=$

ID	Year	Course
1	1991	3
2	1990	4
3	1990	4
4	1990	4
5	1990	4
6	1990	3
7	1990	3
8	1990	2

$r_{ae}=$

ID	Year	Course
2	1990	4
3	1990	4
4	1990	4
5	1990	4
6	1990	3
7	1990	3
8	1990	2

$r_e=$

ID	Year	Course
6	1990	3
7	1990	3
8	1990	2

图 1　部分函数依赖例外数

定义 2. r 为关系 R 的数据实例,$X,Y \subseteq R$ 为两个属性集,r_e 为部分函数依赖 $X \mapsto Y$ 的例外元组集合,则部分函数依赖 $X \mapsto Y$ 的部分函数依赖度 $\alpha = \#(r-r_e)/\#(r)$(后面我们将部分函数依赖记为 $X \overset{\alpha}{\mapsto} Y$,$Y$ 函数依赖于 X 的部分函数依赖度记为 $w(X,Y)$).

据部分函数依赖度定义,图 1 中部分函数依赖 $Year \mapsto Course$ 的部分函数依赖度 $\alpha = \#(r-r_e)/\#(r) = (8-3)/8 = 0.625$,记为 $Year \overset{0.625}{\mapsto} Course$.给出部分函数依赖的相关定义后,可以方便地根据关系对应的数据实例信息计算关系中任意两个元素 m,n 之间的部分函数依赖度 $w(m,n)$.因此,对例 1 中的关系 S,我们计算得到其部分函数依赖集 $PFD(S)=\{CompanyName \overset{1}{\mapsto} ManufaID, ManufaID \overset{1}{\mapsto} Address, Address \overset{0.8}{\mapsto} Telephone, Remark \overset{0.8}{\mapsto} LinkMan, Telephone \overset{1}{\mapsto} CompanyName, SupType \overset{0.6}{\mapsto} Address, \cdots\}$.

4　基于部分函数依赖的结构匹配方法的具体步骤

本文第 3 节中给出了部分函数依赖的定义及其计算方法,并利用与模式 S 对应的数据实例信息计算得到了模式 S 的部分函数依赖集 $PFD(S)$.本节将介绍如何利用部分函数依赖集 $PFD(S)$进行模式匹配.为了描述上的方便,表 2 给出了与源模式 S 对应的目标模式 T,模式 T 也是对进销存数据库中供应商信息的描述,在后面的介绍中我们以模式 S 为源模式,模式 T 为目标模式.

表 2　供应商信息表(T)

Company_ID (PK)	Name	Address	Phone	Fax	Postal Code	Contact Person
A02001	天长市长久电器有限公司	安徽省天长市	0550-7022139	0550-7038928	239300	徐承义
A02002	常州市新迈电子有限公司	常州市钟楼区劳动西路常宁公寓	0519-86862318	0519-86892370	213001	李仲南
A02003	杭州晶新电子有限公司	杭州市市心中路 398 号金城广场 B 座 1801 室	0571-82814526	0571-82814786	311200	陈斌
A03001	常熟市常新电子有限公司	江苏省常熟市常福路	0512-52611888	0512-52611888	215523	叶云兴
B01001	上海双腾电子电器有限公司	上海市崇明县崇明工业园区西引路 578 号	021-69625142	021-69625110	202150	黄晓东
…	…	…	…	…	…	…

4.1　方法准备

第 3 节的最后部分给出了模式 S 的部分函数依赖集,结合实际情况考查该集合时发现如下两个问题:

(1) 虽然 $Address \overset{0.8}{\mapsto} Telephone$ 与 $Remark \overset{0.8}{\mapsto} LinkMan$ 的函数依赖度相同,但事实上属性

Remark 与属性 *LinkMan* 之间并无任何关联关系，发生这种情况是由于计算函数依赖度时我们选取的数据实例的数量太少，从而在计算依赖度时产生了较大的随机误差，这里我们通过增加数据实例数量的方法来避免误差的产生（事实上模式匹配的应用环境中一般都存在大量的数据实例）.

（2）$SupType \overset{0.6}{\mapsto} Address$ 的函数依赖度太低，事实上 *SupType* 和 *Address* 之间并没有任何关联关系，这样的部分函数依赖会对匹配操作产生负面影响，应该去除掉，通常我们选取依赖度大于阈值 ∂ 的依赖关系（从本文 5.3 节实验部分可以看出，阈值 ∂ 选取 0.8 左右比较合适，这里我们选取 $\partial=0.8$）.

对部分函数依赖集的以上两个问题采取相应的处理措施后可得到模式的有效部分函数依赖集（EPFD）. 例 1 中关系 S 的有效部分函数依赖集为

$EPFD(S)=\{CompanyName \overset{0.98}{\mapsto} ManufaID,$
$ManufaID \overset{1}{\mapsto} Address, Address \overset{0.94}{\mapsto} Telephone,$
$Telephone \overset{1}{\mapsto} CompanyName, \cdots\}$.

在根据部分函数依赖计算模式元素间的结构相似度之前，我们首先对源模式和目标模式中的元素根据其自身信息计算它们之间的相似度，称之为语义相似度[9]，然后根据语义相似度对目标模式中的每个元素生成其候选匹配集. 在计算结构相似度时，以候选匹配集为基础，仅计算每个元素与其候选匹配集中所有元素的结构相似度，这样可有效地降低算法的时间复杂度. 候选匹配集一般有如下 3 种选取策略[6]：

（1）*MaxN*：选取相似度最高的 N 个匹配项为候选匹配集.

（2）*MaxDelta*：选取与相似度最大值间差值小于 d 或者最大值的 $\alpha\%$ 的匹配项为候选匹配集.

（3）*Threshold*：选取相似度大于固定阈值（Threshold）的匹配项为候选匹配集.

单一的选择标准都存在缺点，例如：*MaxN* 和 *MaxDelta* 返回的值可能相似度都很低，而 *Threshold* 返回的值可能非常少或者非常多（据 *Threshold* 的大小而定），因此，我们将多条标准结合考虑. 根据算法的特点，我们的算法将 *MaxDelta* 和 *Threshold* 这两种策略相结合，为目标模式中的每个元素 m 生成相应的候选匹配集 $CAND(m)$. 如表 2 中属性 *Company_ID*，计算其与表 1 中各个属性间的语义相似度为 {(*CompanyName*, 0.7), (*ManufaID*, 0.4), (*Address*, 0.14), …}，我们取 *MaxDelta* 策略的 α 值为 50%，*Threshold* 策略的阈值为 0.3，选取属性 *Company_ID* 的候选匹配集为 {*ManufaID*, *CompanyName*}.

4.2 依赖图的建立

图结构是事物间相互关系最直观有效的表现方式，所以这里用图结构来表示元素间的关联关系，图中的节点表示元素，节点间的连线表示元素间的部分函数依赖关系，边的权重表示元素间的部分函数依赖度. 根据模式 S 的有效部分函数依赖集 $EPFD(S)$ 可生成模式 S 的部分函数依赖图 $G(V,E)$，其中 V 是节点集合，每个节点表示模式中的一个元素，E 是有向边集合，每条有向边表示有效部分函数依赖集 $EPFD(S)$ 中的一个部分函数依赖关系. 例如，$CompanyName \overset{0.98}{\mapsto} ManufaID$ 在图结构中用一条从节点 *CompanyName* 到节点 *ManufaID* 的权重为 0.98 的有向边表示. 图 2 是模式 S 的完全函数依赖图（仅包含完全函数依赖关系），图 3 是模式 S 的部分函数依赖图.

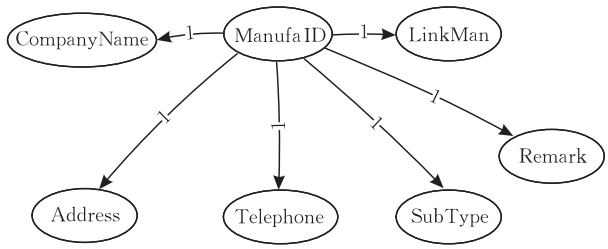


图 2 模式 S 的完全函数依赖图

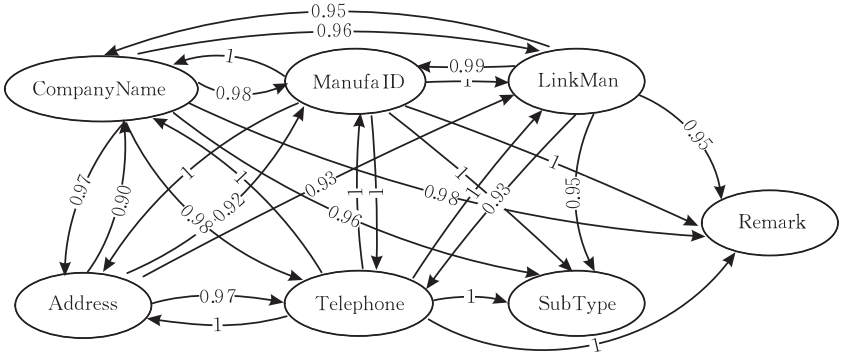


图 3 模式 S 的部分函数依赖图

图 2 根据模式 S 的完全函数依赖集(所有完全函数依赖组成的集合)构建而成,图 3 则根据模式 S 的有效部分函数依赖集 $EPFD(S)$ 构建而成.将两图对比,可明显看出部分函数依赖提供了更多的结构信息,能更有效地支持匹配操作.下面以图 3 为例介绍部分函数依赖图中的一些概念:图 3 中有一条从节点 *CompanyName* 指向节点 *Address* 的权重为 0.97 的有向边,据该边我们称节点 *CompanyName* 为节点 *Address* 的依赖节点,节点 *Address* 为节点 *CompanyName* 的决定节点,边的权重 0.97 称为元素 *Address* 对元素 *CompanyName* 的部分函数依赖度.

4.3　结构相似度计算

建立部分函数依赖图后,接下来我们根据部分函数依赖图计算元素间的结构相似度.图 4 为目标模式 T 的部分函数依赖图.

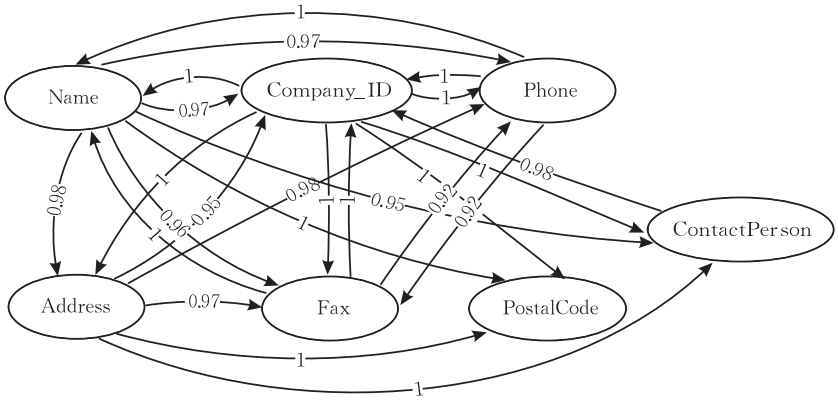


图 4　模式 T 的部分函数依赖图

与一个元素关联的其它元素构成了该元素的结构信息,在部分函数依赖图中表示为该元素的依赖节点和决定节点,其对应的集合分别称为依赖元素集和决定元素集.定义如下.

定义 3.　元素 e 的决定元素集为部分函数依赖图中以该元素为起点的所有有向边的终点所表示元素的集合,同时把以 e 为起点的有向边的权重及该边终点所表示的元素合称为 e 的一个决定结构信息,把所有决定结构信息组成的集合称为 e 的决定结构信息集.

如图 3,模式 S 中的元素 *CompanyName* 的决定元素集为{*ManufaID*、*Address*、*Telephone*、*LinkMan*、*SupType*、*Remark*},决定结构信息集为{(*ManufaID*,0.98),(*Address*,0.97),(*Telephone*,0.98),(*LinkMan*,0.96),(*SupType*,0.96),(*Remark*,0.98)}.

定义 4.　元素 e 的依赖元素集为部分函数依赖图中以该元素为终点的所有有向边的起点所表示元素的集合,同时把以 e 为终点的有向边的权重及该边起点所表示元素合称为 e 的依赖结构信息,把所有依赖结构信息组成的集合称为 e 的依赖结构信息集.

如图 3,模式 S 中的元素 *CompanyName* 的依赖元素集为{*ManufaID*、*Address*、*Telephone*、*LinkMan*},其依赖结构信息集为{(*ManufaID*,1),(*Address*,0.90),(*Telephone*,1),(*LinkMan*,0.95)}.

同理我们把元素间的结构相似度分为决定结构相似度和依赖结构相似度,分别根据决定结构信息集和依赖结构信息集来计算决定结构相似度和依赖结构相似度.由于二者的计算过程相似,下面我们以决定结构相似度的计算为例介绍它们的计算过程.

在引入部分函数依赖这个概念之前,两个元素间的结构相似度一般是指与它们分别关联的所有元素组成的集合中互为候选匹配的元素占所有元素的比率[12].引入部分函数依赖这个概念之后,虽然同为候选匹配对,但由于元素间的函数依赖关系有强弱(部分函数依赖度的高低)之分,因此对结构相似度的促进作用也有差异,所以前面计算结构相似度的方法在这里已经不再适用.为了方便后面的介绍,我们定义了如下概念.

定义 5.　模式 S 中存在部分函数依赖关系 $A \overset{\alpha}{\mapsto} C$,模式 T 中存在部分函数依赖关系 $B \overset{\beta}{\mapsto} D$,若 (A,B) 和 (C,D) 为两个候选匹配对,则称 (A,B) 和 (C,D) 互为促进匹配对,它们之间相互促进作用的大小称为相似度促进度.

下面我们通过如下 3 个问题来分析促进匹配对间的相似度促进度.

(1)(C,D)对(A,B)相似度促进度的大小与(C,D)的相似度之间有什么关系?

(2)(C,D)对(A,B)相似度促进度的大小与函数依赖度α,β之间有什么关系?

(3)(C,D)作为(A,B)的促进匹配对,对(A,B)的相似度促进度有多大?

对问题(1),显然(C,D)的相似度越高,对(A,B)的相似度促进度越大.对问题(2),当$\alpha=\beta$,即A、B分别以相同的函数依赖度决定C、D时,(C,D)对(A,B)的相似度促进度最大;$|\alpha-\beta|$越大,则(C,D)对(A,B)的相似度促进度越小.例如:模式S对应的部分函数依赖图(图3)中存在部分函数依赖关系:$S.ManufaID \overset{1}{\mapsto} S.Address$和$S.CompanyName \overset{0.97}{\mapsto} S.Address$,模式$T$对应的部分函数依赖图(图4)中存在部分函数依赖关系:$T.Name \overset{0.98}{\mapsto} T.Address$.从中我们可以看出$S.CompanyName \overset{0.97}{\mapsto} S.Address$与$T.Name \overset{0.98}{\mapsto} T.Address$的函数依赖度相接近,即两者差值为0.01,而$S.ManufaID \overset{1}{\mapsto} S.Address$与$T.Name \overset{0.98}{\mapsto} T.Address$的函数依赖度差值为0.02,据前面分析我们得出如下结论:因为$0.02>0.01$,所以$(S.Address,T.Address)$对$(S.ManufaID,T.Name)$的相似度促进度比对$(S.CompanyName,T.Name)$的相似度促进度小.问题(1)、(2)是对相似度促进度大小与函数依赖度、促进匹配对相似度之间关系的定性分析,问题(3)则是对它们之间关系的定量分析.表3列举了描述依赖度差值d、(C,D)间相似度m与相似度促进度ε之间定量关系的几种方法.从本文第5.2节我们可以看出,与第1种线性函数和第2种抛物线函数相比,第3种以自然对数为底的指数函数能够更准确地描述三者间的定量关系.

表3 3种相似度间差值d与促进度ε之间的定量关系

序号	名称	公式描述
1	线性函数	$\varepsilon=(1-d)\times m\times a$($a$为参数)
2	抛物线函数	$\varepsilon=1-m\times d^2$
3	指数函数	$\varepsilon=m\times e^{-d\times\mu}$($\mu$为参数)

通过对以上3个问题的分析,我们得到了促进度与函数依赖度、促进匹配对相似度之间的定性关系,并给出了有效的公式描述.下面我们利用该公式来计算结构相似度.算法如图5所示.需要说明的是:由于语义相似度的计算都采用启发式的方法,计算出的相似度数值并不具有实际意义,所以该算法中将所有候选匹配对间的语义相似度都看作1(可有效减少计算量,对结果无明显影响).

CalDcssim(x,y,X,Y)

输入:候选匹配对(x,y),x的决定元素集X,y的决定元素集Y

输出:x和y的函数决定结构相似度$dcssim(x,y)$

1. 从X中任取元素m,令$Q=Y\cap CAND(m)$,并从X中去除m;
2. 若$Q=\varnothing$,转入步4;
3. 在Q中选择使$|w(y,n)-w(x,m)|$最小的元素n,令
   $dcssim(x,y)=dcssim(x,y)+m\times e^{-|w(x,m)-w(y,n)|\times\mu}$;
4. 若$X\neq\varnothing$,则转入步1;
5. 从Y任取元素p,令$Q=X\cap CAND(p)$,并从Y中去除p;
6. 若$Q=\varnothing$,转入步8;
7. 在Q中选择使$|w(x,q)-w(y,p)|$最小的元素q,令
   $dcssim(x,y)=dcssim(x,y)+m\times e^{-|w(x,q)-w(y,p)|\times\mu}$;
8. 若$Y\neq\varnothing$,则转入步5;
9. 返回$dcssim(x,y)=dcssim(x,y)/(|X|+|Y|)$;

图5 决定结构相似度计算

图5中的算法以候选匹配对(x,y)以及x,y对应的决定元素集X,Y为输入,步1~4遍历X中所有元素,对每一个元素m,求m的候选匹配集$CAND(m)$与集合Y的交集Q,若$Q\neq\varnothing$,则在Q中选择这样一个元素n,使得$|w(y,n)-w(x,m)|$最小,即相似度促进度最大,然后得出(m,n)对(x,y)的相似度促进度为$e^{-|w(x,m)-w(y,n)|\times\mu}$,并将该值与其它促进匹配对的相似度促进度相加.处理完X中的元素后,步5~步8对Y中元素做相同处理,最后步9将所有相似度促进度之和对X,Y中元素数量求平均得到候选匹配对(x,y)的决定结构相似度$dcssim(x,y)$.

对于依赖结构相似度我们采用与决定结构相似度同样的计算方法,不同的是这里的X为元素x的依赖元素集,Y为元素y的依赖元素集,最后求得元素x和y的依赖结构相似度$dpssim(x,y)$.

4.4 结构相似度传递

4.3节中计算的候选匹配对的结构相似度是所有促进匹配对的语义相似度对其促进度的综合,但事实上促进匹配对的结构相似度对候选匹配对的结构相似度也具有促进作用,即互为促进匹配对的两个候选匹配对的结构相似度间也能够相互促进.据此,我们采用传递调整算法对结构相似度进行调整优化,使结构相似度能够更准确地反映元素间结构上的相似程度.

在介绍传递调整算法之前,我们首先介绍决定集结构相似度和依赖集结构相似度的概念.以x的决定元素集X和y的决定元素集Y为例,根据X,Y我们定义二部图$G(X,Y,E)$,$E=\langle(m,n)\mid m\in X\wedge n\in Y\wedge n\in CAND(m)\rangle$,$E$中每条边的权值为所关联的一对候选匹配对$(m,n)$对候选匹配对$(x,y)$的相似度促进度,这里为

$$dcssim(m,n)\times e^{-|w(x,m)-w(y,n)|\times\mu}.$$

我们用该二部图的最大流通量表示这两个元素的决定集结构相似度(式(1)),对式(1)的右边采用匈牙利算法[15]计算二部图最大流通量.

$$dcssim(X,Y)=\max\Big(\sum_{m\in X}^{m}\sum_{n\in Y}^{n}(dcssim(x,y)\times e^{-|w(x,m)-w(y,n)|\times\mu})\Big) \quad (1)$$

对于依赖集结构相似度,我们以 x 的依赖元素集 X 和 y 的依赖元素集 Y 为例,同理我们可以得到其依赖集结构相似度(式(2)),对式(2)的右边同样采用匈牙利算法[15]计算二部图最大流通量.

$$dpssim(X,Y)=\max\Big(\sum_{x\in X}^{x}\sum_{x\in Y}^{y}dpssim(m,n)\times e^{-|w(x,m)-w(y,n)|\times\mu}\Big) \quad (2)$$

然后根据以上定义,我们分别对候选匹配对的决定结构相似度和依赖结构相似度进行调整.

首先我们对决定结构相似度进行调整,对任意一对候选匹配(x,y),x 对应的决定节点集合为 X,y 对应的决定节点集合为 Y,则 x,y 间的决定结构相似度可利用如式(3)进行调整.

$$dcssim(x,y)=\alpha\times dcssim(x,y)+\beta\times dcssim(X,Y),\quad \alpha+\beta=1 \quad (3)$$

根据式(3)对所有候选匹配对的决定结构相似度都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决定结构相似度的变化都小于阈值 λ,说明调整已充分,调整过程结束;若多次调整仍未达到要求,则在第 N 个调整周期后结束调整.同理,对依赖结构相似度的调整与对决定结构相似度的调整类似.

经过上述调整我们可以得到比较真实反映元素间结构相似程度的决定结构相似度和依赖结构相似度,下面我们将介绍如何根据决定结构相似度、依赖结构相似度及原有的语义相似度生成模式元素间的映射.

4.5　映射生成

映射生成是映射关系确定的过程,它是模式匹配过程中的一个重要步骤,其确定的映射关系作为模式匹配的结果直接输出.文献[11]中介绍了一种解决映射问题的方法:稳定婚姻法.其核心思想是:选择满足如下两个条件的匹配对组成的集合作为模式映射结果.

(1)集合中所有匹配对的相似度之和最大.

(2)其中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.

本文也采用相同的方法来生成映射.前面我们得到了元素间的语义相似度、决定结构相似度和依赖结构相似度,但是在映射生成过程中,使用 3 种不同的相似度标准会使过程复杂、准确率降低,所以在生成映射之前我们先对这 3 种相似度采用加权平均法进行合并,生成候选匹配对总的相似度 $sim(x,y)$,如式(4):

$$sim(x,y)=\alpha\times lsim(x,y)+\beta\times dcssim(x,y)+\gamma\times dpssim(x,y),\ \alpha+\beta+\gamma=1 \quad (4)$$

得到候选匹配对总的相似度后,我们再根据稳定婚姻法选取最终的映射结果输出.第 5 节将对本方法得到结果的精确性进行实验评价.

5　算法实验评价

5.1　实验情况介绍

本方法利用数据实例信息挖掘元素间的结构信息,并以之辅助匹配.为验证本方法的有效性,我们将本方法与模式匹配领域中常用的一些方法进行实验对比,并以如下 3 个指标来描述对比结果.

(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所有正确匹配的比率;

$$Recall = T/R.$$

(3)全面性(Overall):通过匹配方法节省的工作量占总的匹配工作量的比率.

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

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[6,11-12].

对测试用例的选取我们分为两个步骤,首先选取模式结构,然后生成数据实例.这里的源模式和目标模式分别取自两家销售同类产品的公司的进销存管理系统,称为 DB1 和 DB2.表 4 列出了两个模式中关系和属性的基本情况.模式中的数据实例采用 DTM Data Generator① 生成.测试中我们将模式及

① http://www.sqledit.com/dg/download.html

其对应数据导入 MySql5.0 数据库中，使用 ODBC 连接数据库以获取各种模式信息. 主机硬件采用 Intel Pentium 4 2.0GHz，1GB 内存；操作系统为 Windows XP SP2.

表 4 DB1、DB2 的关系及属性数目

模式	关系数目	属性数目
DB1	25	224
DB2	27	252

5.2 相似度促进度与相似度及函数依赖度定量关系的实验证明

4.3 节中对相似度促进度与相似度及函数依赖度的关系进行了讨论，给出了 3 种描述它们之间关系的方法，这里我们通过实验来对这 3 种方法进行分析对比. 取数据实例集的大小为 3000，对比结果如图 6 所示.

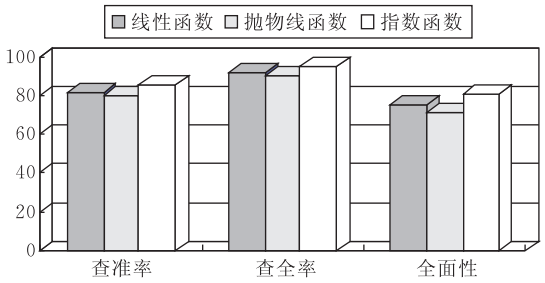


图 6 3 种不同的描述方法之间的指标对比图

通过图 6 我们可以看出，当用指数函数描述函数依赖度与相似度促进度之间的关系时，在查准率、查全率和全面性 3 个指标上都优于其它两种描述方法，所以本方法中采用指数函数来描述函数依赖度与相似度促进度间的关系.

5.3 参数 ∂ 不同取值的匹配结果对比

本文 4.1 节中，对所有的部分函数依赖关系，我们选取依赖度大于阈值 ∂ 的依赖关系作为有效部分函数依赖来辅助匹配. ∂ 值的选取对匹配结果有很大程度的影响，图 7 是阈值 ∂ 在区间[0.4,1]之间变化时，查准率、查全率、全面性 3 个指标的变化图. 由于函数依赖度低于 0.4 时无任何实际意义，在此我们对函数依赖度值小于 0.4 的依赖关系不予考虑.

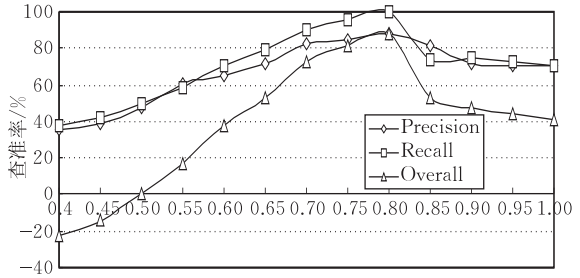


图 7 不同阈值 ∂ 对算法性能的影响

图 7 描述了算法各项性能指标随阈值 ∂ 变化的趋势图，如图所示，算法的查准率(Precision)从 36%提高到 88%再下降到 70%，这说明算法的查准率随着 ∂ 从 0.4～1 的变化而先升后降，并在某一中间值达到最高，即匹配结果中正确匹配占所有匹配结果的比率达到最高. 算法的查全率从 38%提高到 100%然后下降到 71%，说明算法的查全率也随着 ∂ 从 0.4～1 的变化而先升后降，并在某一中间值达到最高，即匹配结果中包含的正确匹配数目最多. 算法的全面性同样从－22.7%提高到 88%然后下降到 41.4%，也说明全面性随着 ∂ 从小到大的变化而先升后降，并在某一中间值达到最高，即匹配算法节省的工作量最多.

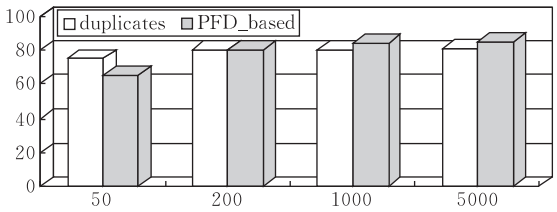
从图中我们发现：当 ∂ 取值在 0.8 左右时，算法的各项性能指标都达到最大. 所以为了使算法的性能最优，∂ 应在 0.8 左右取值. 当 ∂ 取值过小时，很多依赖度很小的部分依赖关系参与匹配，依赖度数值偏小意味着这个依赖关系不具有实际意义的可能性很大，这样的依赖关系参与匹配会使算法的准确度降低；当 ∂ 取值过大时，有些依赖度数值较大的依赖关系被过滤，依赖度数值较大意味着该依赖关系具有实际意义的可能性很大，这样的依赖关系不参与匹配过程会使算法的准确度降低，而当 $\partial=1$ 时，几乎所有的非完全函数依赖都不参与匹配过程，算法仅仅利用了模式中的完全函数依赖关系，准确度大幅下降.

5.4 同类方法对比

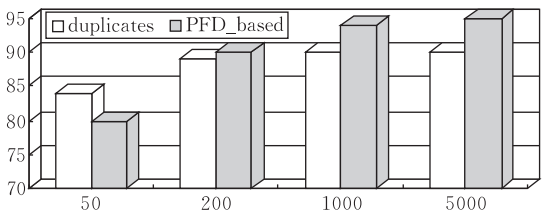
数据实例信息与结构信息是模式匹配中利用的两类重要信息，利用数据实例信息进行匹配的方法中具有代表性的是 duplicates[10] 方法，而 cupid[12] 方法和 SF[11] 方法则都是常用的利用结构信息来进行匹配的方法，因为本方法将这两种信息结合起来，利用数据实例信息计算得到的结构信息(部分函数依赖)辅助匹配. 所以为了验证本方法的有效性，下面我们首先将本方法与利用数据实例信息进行匹配的 duplicates 方法进行对比，由于数据集中数据量的大小对两种方法的实验结果都有影响，所以我们分别生成了四种不同数据量大小(50,200,1000,5000)的数据集进行对比，实验结果图 8 所示.

从图 8(a)两种方法查准率的对比图中我们可以看出：随着数据集中数据量的不断增大，duplicates 方法和 PFD_based 方法的查准率都在增大，但是相对于 duplicates 方法，PFD_based 方法受数据量大小的影响更大. 当数据量为 50 时，PFD_based 方法的查准率低于 duplicates 方法，当数据量增大到

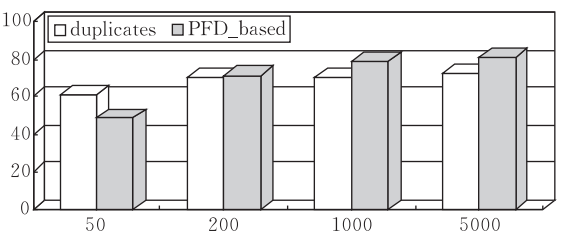
1000 时,PFD_based 方法的查准率就明显高于 duplicates 方法,同时我们还发现当数据集的数据量从 1000 增加到 5000 时,两种方法查准率的提高幅度都比较小.据此我们得到如下结论:数据集越大,越能够提高方法的查准率,但当其增大到一定程度后再继续增大时,则对查准率的提高幅度并不大,反而会提高算法的时间复杂度.据图 8(b)和图 8(c)分别对查全率和全面性的对比我们也可以发现和图 8(a)相同的规律.



(a) 两种方法针对不同数据量的数据集合的查准率对比图



(b) 两种方法针对不同数据量的数据集合的查全率对比图



(c) 两种方法针对不同数据量的数据集合的全面性对比图

图 8　两种方法针对不同数据量的数据集合的指标对比

图 8 中分别对查准率、查全率和全面性 3 个指标进行对比,我们可以得到如下结论:与 duplicates 方法相比,PFD_based 方法对数据量的要求更高,需要大量的数据实例,但当数据实例的数量大于 1000 时,算法的各项指标相比 duplicates 方法都有较大提高,大幅度提高了模式匹配的精确度.同时考虑到算法的时间复杂度,我们一般取数据量的大小位于区间[1000,5000].接下来我们再将本方法与利用结构信息的 Cupid 方法和 SF 方法进行对比,这里取数据集的大小为 3000.实验数据如图 9 所示.

从图 9 中 3 种方法在各个指标上的对比我们可以看出:在查准率指标上,PFD_based 方法高于其它两种方法,即 PFD_based 方法的匹配结果中正确匹配所占的比率最高;在查全率指标上,PFD_based 方法高于其它两种方法,即 PFD_based 方法的匹配结果中包含的正确匹配最多;在全面性指标上,PFD_based 方法也高于其它两种方法,即节省的工作量也比其它两种算法要多.

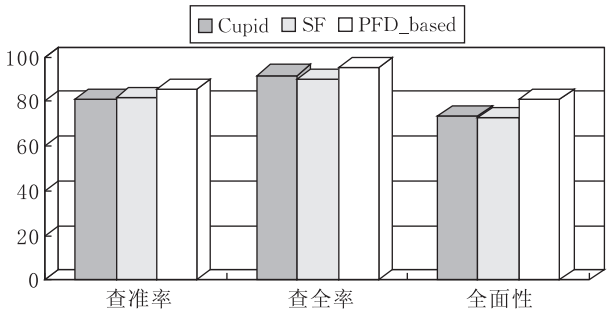


图 9　Cupid、SF 和 PFD_based 3 种方法的不同指标对比

通过上面的实验我们可以看出,PFD_based 方法将数据实例信息和结构信息结合起来进行匹配,在各项性能指标上都明显优于单独利用数据实例信息的 duplicates 方法,也明显优于单独利用结构信息的 SF 方法和 Cupid 方法.

6　总结与展望

本文提出了一种通过数据实例信息得到元素间的部分函数依赖关系,然后利用其辅助模式匹配的新方法,并从理论分析和实验结果两个方面论证了此方法能够在一定程度上提高模式匹配的精确度.目前的模式匹配方法大多以元素自身信息为主来进行匹配,其它种类的信息只起到了一些辅助的作用,然而同一描述对象由不同的模式设计者描述时,其自身信息可能有很大的差异,从而在模式匹配过程中产生误导作用,使得单独利用元素自身信息进行匹配的效果并不理想,事实也正是如此.数据实例信息能够最真实地反映元素的语义,且不会由于设计者的不同而产生差异,能够在模式匹配过程中发挥更重要的作用.未来我们可以利用数据实例信息来获取更真实的元素语义,并进行匹配,提高匹配的精确度.

参　考　文　献

[1] Zhao Hui-Min. Semantic matching across heterogeneous data sources. Communications of the ACM, 2007, 50(1): 45-50

[2] Li W, Clifton C. SemInt: A tool for identifying attribute correspondences in heterogeneous databases using neural network. Data & Knowledge Engineering, 2000, 33(1): 49-84

[3] Warren R H, Tompa F W. Multicolumn substring matching for database schema translation//Proceedings of the VLDB. Seoul, Korea, 2006: 331-342

[4] Bohannon P, Elnahrawy E, Fan Wen-Fei, Flaster M. Putting context into schema matching//Proceedings of the VLDB. Seoul, Korea, 2006: 307-318

[5] Madhavan J, Bernstein P A, Doan An-Hai, Halevy A. Corpus-based schema matching//Proceedings of the 21st International Conference on Data Engineering (ICDE). Berlin, Germany, 2005: 57-68

[6] Do Hong-Hai, Rahm E. COMA—A system for flexible combination of schema matching approaches//Proceedings of the VLDB. Hong Kong, China, 2002: 610-621

[7] Aumueller D, Do Hong-Hai, Massmann S, Rahm E. Schema and ontology matching with COMA++//Proceedings of the SIGMOD. Baltimore, Maryland, 2005: 906-908

[8] Elmeleegy H, Ouzzani M, Elmagarmid A. Usage-based schema matching//Proceeding of the 24st International Conference on Data Engineering (ICDE). Cancun, China, 2008: 20-29

[9] Rahm E, Bernstein P A. A survey of approaches to automatic schema matching. VLDB Journal, 2001, 10(4): 334-350

[10] Bilke A, Naumann F. Schema matching using duplicates//Proceeding of the 21st International Conference on Data Engineering (ICDE). Berlin, Germany, 2005: 69-80

[11] Sergey Melnik, Hector Garcia-Molina, Erhard Rahm. Similarity flooding: A versatile graph matching algorithm and its application to schema matching//Proceedings of the ICDE. San Jose, California, 2002: 117-128

[12] Madhavan J, Bernstein P A, Rahm E. Generic schema matching with cupid//Proceedings of the VLDB. San Francisco, CA, USA, 2001: 49-58

[13] Berzal F, Cubero J C, Cuenca F, Medina J M. Relational decomposition through partial dependencies. Data & Knowledge Engineering, 2002, 43(2): 207-234

[14] Gusfield D, Irving R. The Stable Marriage Problem: Structure and Algorithms. Cambridge, MA: MIT Press, 1989

[15] Lu Zheng-Nan, Zhang Huai-Sheng. Foundation of Operations Research. Hefei: University of Science and Technology of China Press, 2006(in Chinese)
(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)

LI Guo-Hui, born in 1973, Ph. D., professor, Ph. D. supervisor. His main research interests include active databases, real-time database, mobile computing, synchronization for parallel/concurrent programming and data integration.

DU Xiao-Kun, born in 1980, Ph. D. candidate. His main research interests include data integration and schema matching.

DU Jian-Qiang, born in 1968, professor. His main research interests are medical imaging.

Background

Along with the development of database in recent years, database application is becoming more extensive and stored information is becoming larger. People propose a further request on the database application, e. g. schema integration, data warehousing, E-business, semantic WEB, P2P database and so on. As a basic operation in the field of data integration, schema matching plays an important role in these application fields. At present, research about schema matching is very rich, from using of element's own information at the beginning to using of data instance related to elements and structure information between elements afterwards, these researches promote the development of schema matching greatly. The method in this paper is a new method which uses structure information to assist match. Different from the normal method of schema matching, the structure information used by author is the partial functional dependency which computed by the data instance of the schema. It fills the gaps of the lack of structure information in the method of schema matching, greatly enriched the structure information. In this paper, the author has given a whole solution about problems met in the process of schema matching according to the characteristic of partial functional dependency. At last, Extensive simulation experiments are conducted between the method in this paper and the other related methods, the results show that this algorithm is better than other related algorithms in various performance metrics such as precision, recall and overall with the help of a certain number of data instances. This operation is part of the data integration project in our lab. This project integrates several databases from different real estate companies and the auto schema matching operation saved much manpower in data integration. The work in this paper is supported by National High Technology Research and Development Program (863 Program) of China under grant No. 2007AA01Z309, the National Natural Science Foundation of China under grant No. 60873030, the Fund of Defense Preliminary Research under grant Nos. 9140A04010209JW0504 and 9140A15040208JW0501, the Special Funds of Central Colleges for Basic Scientific Research and Operating Expenses.