基于属性实例集合语义相似的模式匹配

李蓉蓉 王 晖 陈 冉
(武汉大学计算机学院 武汉 430072)

摘 要 近年来,模式匹配作为Web信息集成管理与应用中的重要问题,得到了广泛关注和研究。已有模式匹配方法大多是基于模式信息的,对数据实例信息利用则较少。针对数据集成环境下模式信息不全或存在冲突的模式信息导致模式匹配结果不正确的问题,给出了计算属性间语义相似性的方法以提高模式匹配的性能,分析了模式内语义相近多属性间的语义差别,进一步给出了基于带权二分图最大化算法的模式匹配方法。通过实验,说明基于实例集合语义相似的模式匹配方法能在模式信息不全面或存在冲突的情况下,得到更完整、更准确的模式匹配。

关键词 模式匹配,数据实例,语义相似
中图法分类号 TP311 **文献标识码** A

Schema Mapping Based on Semantic Similarity of Data Instances Sets

LI Rong-rong WANG Hui CHEN Ran
(School of Computer Science, Wuhan University, Wuhan 430072, China)

Abstract Schema mapping, as the important issue in Web integrated information management and application, has been widely concerned and researched in recent years. Most schema mapping methods have been implemented based on schema information, and schema mapping methods based on data instances are rare. There are many cases in which schema information is insufficient or conflict, and incorrect schema mapping method may be got. In order to improve the performanceof schema mapping, a measure was proposed to evaluate semantic similarity between attributes, also the semantic difference between two semantic similarattributes in the same schema was analyzed. Further, a mapping approach based on weighted bipartite was proposed. The mapping method proposed was evaluated with experiments and was verified to be efficient and effective to obtain the more complete and accurate schema mapping in situations where schema information is incomplete or conflicting.

Keywords Schema mapping, Data instances, Semantic similarity

1 引言

Web数据集成是用户从异构数据源获取规整数据的有效途径,其中对异构数据源的模式进行匹配是重要任务之一。模式匹配的目标是发现两模式的元素(或属性)之间可能存在的语义映射关系,匹配结果一般表示为匹配属性对集合。

已有模式匹配方法多数基于数据源的模式信息,主要通过分析数据源查询接口和结果页面包含的模式信息(如模式名及其属性名,模式的树状结构(DOM树))等分析属性间的语义相似性。最终得到模式中元素(或属性)结点间的匹配关系。Cupid[1]结合属性命名的语法和语义特征、属性结构信息及约束信息进行模式匹配。COMA[2]提出了一种基于匹配器组合方法的弹性模式匹配机制。Clio[3]提出了半自动发现、使用和管理语义的模式匹配方法。为解决异构数据交换问题,一些基于XML模式匹配的方法被提出,文献[4]提出了一种基于最长公共子序列、计算上下文路径的XML模式匹配方法。也有少量模式匹配方法基于数据实例,如文献[5]提出了一种特定领域内基于实例的模式匹配方法,事先设计向多数据源提交的查询样例,将查询关键字交叉赋给各数据源接口的查询元素,根据查询结果建立某数据源的接口模式与另一数据源的结果模式中属性的映射关系,结合各数据源自身的接口模式和结果模式间的映射关系,进行交叉验证,最终获得数据源间的模式匹配;DUMAS[6]是一种利用数据实例来计算模式匹配结果的方法,即假定两个数据源具有相同或相似属性值的重复数据,利用Soft-TFIDF相似性计算方法对重复数据的字段逐个进行比较,得到属性相似性矩阵以进行模式匹配。基于实例的方法大多假定数据源间存在重复实例,另一些方法则试图通过发现实例属性间的各种不同的依赖关系来进行模式匹配。文献[8]提出了挖掘数据实例中的结构信息,如:属性之间的部分函数依赖关系,进行模式匹配的方法。

数据实例是模式信息的具体表现,在已知模式信息不全

到稿日期:2011-01-22 返修日期:2011-04-05 本文受国家自然科学基金面上项目(61070011),国家重点基础研究发展计划(973)"验证示范-服务大众的按需回答原型系统"(2007CB310806)资助。
李蓉蓉(1973—),女,博士生,主要研究方向为数据集成与融合,E-mail:rrli@whu.edu.cn;王 晖(1983—),男,博士生,主要研究方向为Web数据集成;陈 冉(1987—),男,硕士生,主要研究方向为数据抽取。

或冲突时，可从中分析挖掘隐含的模式信息作为模式信息的补充。但若数据实例无重复或重复较少时，模式匹配则比较困难。比如，两超市 A，B 由于认知差异各自建立的销售信息模式不同，ID 分别被两超市用来表示顾客号和商品号，这种情况下仅根据已知的模式信息进行匹配很难得到正确结果；但分析数据实例的取值特征仍可能实现正确匹配。而 Deep Web 上获取的不同来源的数据显然具有类似的异构性。

各数据实例能表现有个体差异的实体，它们的关键字取值不同，但具有一些共同的语义特征。基于这一直觉，本文提出了一种基于实例的模式匹配方法，旨在判定两类实体（即两模式）的属性实例集合间是否具有相似的语义特征。通过属性实例分析计算属性间的语义相似度，并对同一模式中语义近似属性之间的关系进行分析；基于属性语义相似度提出了一种全局最优的模式匹配方法，最后通过实验证明了这一模式匹配方法的有效性。

本文第 1 节介绍了一些模式匹配的研究方法与现状；第 2 节重点介绍了分析属性实例的分布特征、语义相似性以确定属性间语义映射的方法；第 3 节对模式内语义相似属性的关系进行分析；第 4 节介绍了生成模式匹配对的方法；第 5 节则通过实验检验基于实例的方法的有效性；最后对研究方法进行总结，并给出未来进一步可进行的研究工作。

2 属性实例集合的语义相似度

假设模式匹配问题中的属性匹配关系都是 1∶1。在模式信息不完整、模糊或缺乏时，属性名仅作为一个符号，并不能清晰、全面地反映语义。

基于属性实例集合的匹配问题的关键是如何计算集合间的语义相似度。问题的形式化描述为：

根据属性 $S_1.a_i, S_2.b_j$ 的实例集合为 $A=\{v_1^{a_i}, v_2^{a_i}, \cdots, v_l^{a_i}\}$ 和 $B=\{v_1^{b_j}, v_2^{b_j}, \cdots, v_k^{b_j}\}$，计算 A 和 B 间的语义相似度。而这一计算需要领域知识作为支持。

2.1 属性实例集合的语义相似性

两个属性实例集合中元素的语义信息相似性越大，说明两属性对应相同或相似语义概念的可能性就越大。

2.1.1 实例间和实例集合之间的语义关系

直觉上，数据源属性实例集合内部的元素语义紧密程度一般比它们与其它实例集合的元素间的高，通过查询得到数据视图，其中实例与查询是语义相关。

两数据的语义相似性因它们所属语义粒度的不同而不同，如："食品"与"洗发液"的相似度比"洗发液"与"护发液"间的相似度低，不同粒度的地理区域之间的关系也属于类似情况。两属性实例集合可能对应着同一粒度级别的语义不同的一些子区间，或者不同粒度级别的一些子区间，实例集合都具有一定的语义相似性。当两属性实例集合对应着一些相同的最小语义粒度的子区间时，两者的语义相似性较高。

2.1.2 基于实例进行属性间的语义映射

一些属性的实例集合内的元素具有相同数据模式（data pattern），如：学生学号、商品号等编号属性中不同位置的编号位分别代表不同含义，可先根据模式特征计算相应属性的语义相似性。再对实例无模式的属性对分析两实例集合内具体元素之间的语义相似性。

如：从 Web 上不同数据源查询得到产品信息，其中无重复记录，但缺少记录的模式信息。

根据图 1 中已知的不完整模式信息（如属性名）不能得到正确的属性匹配关系。但通过分析两数据源的实例集合，可发现其中都包含了两个日期属性、商品名称、商品代号，从而得到正确的属性匹配。

商品代号	商品名	购买日期	顾客代号
wyyp-0605	Shampoo	2011-6-20	2013-6-19
…	…	…	…

spdh	spm	gmrq	gkdh
sxsp	Milk	2011-6-20	2011-6-27
…	…	…	…

图 1 模式表达的语义信息不完整

2.2 属性语义相似性的计算过程

下面对两实例集合 A 和 B 间的并集 $U=A\cup B$ 的元素进行聚析，通过分析两集合的元素在各聚类中的分布来计算语义相似性。假定集合 A 和 B 分别包含 10 和 11 个数据。对集合 A 和 B 的并集进行聚类后，得到的聚类结果如图 2 所示，一个虚线椭圆代表一个聚类，三角形和矩形代表其中元素分别来自集合 A 和集合 B。图 2 中集合 A、B 的元素取值分布比较相似时，表明语义距离普遍比较小。

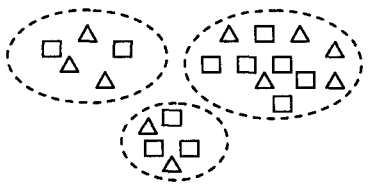

图 2 两集合元素在各聚类中的分布

计算两实例集合语义相似性的方法 MIC（Matching based on Instances Clustering）分为 3 步：

Step1 将两实例集合合并，并集记为 U；

Step2 对 U 中的元素进行聚类；

Step3 根据聚类结果计算两集合的语义相似性。

2.2.1 聚类过程（Clustering）

根据相似性大小对两实例集合的并集进行聚类时，要求在聚类结果 $C_1, C_2, \cdots, C_k$ 和它们的中心点 $O_1, O_2, \cdots, O_k$ 之间使式(1)中的 E 取最大值。

$$E=\sum_{i=1}^{k}\sum_{j=1}^{|c_k|}\mathrm{sim}(O_i, v_{ij}) \tag{1}$$

式中，聚类算法的效果取决于计算样本点对的相似性函数 $\mathrm{sim}(v_{ij}, v_{tk})$。K-medoid 聚类算法的基本过程如下：

输入：聚类数目 k，n 个样本点；

输出：使 E 最大化的 k 个聚类。

(1)随机选择 k 个样本点作为 k 个聚类的初始中心点；

(2)计算剩余样本点与各中心点 $O_i(i=1,\cdots,k)$ 的相似性将它们依次加入相应聚类中；

(3)若 E 能减小，则更新每个簇的中心点；

(4)反复执行第(2)、(3)步，直到中心点的集合不再发生变化。

相似性函数与具体领域、数据类型有关。对文本型数据而言，可用编辑距离、基于 n-gram 的相似性方法，基于 WordNet 计算文本型数据间的相似度，也可用基于 Web 搜索结果的规范化互信息方法，如：NGD（Normalized Google Dis-

tance)。

2.2.2 量化集合间的语义相似性

两实例集合 A,B 的语义相似性可表现为两者在聚类结果中分布的相似性，可以规范化条件熵[7]表示：将 U 看作随机变量，先计算 U 在给定聚类结果 T 时的条件熵，再用 U 的信息熵对条件熵进行规范化。根据两集合在聚类结果中的分布(即样本点分布)，按式(2)计算基于聚类的条件熵：

$$H(U|C)=-\sum_{c\in C}\sum_{u\in U}p(u,c)\log p(u|c) \tag{2}$$

集合的语义相似度用式(3)计算。

$$Coupling(A,B)=\frac{H(U|C)}{H(U)} \tag{3}$$

根据图2所示的聚类结果进行计算，得到两集合 A 和 B，具有相同信息熵 $H(U)=0.9984$。根据聚类结果分布，得到 $H(U|C)=0.983$。说明两集合的相似度为 0.9847。

2.2.3 聚类结果不均匀的情形

实例集合间的元素对的相似性比集合内部某些元素之间的大时，聚类结果中分属于两集合的元素的分布将比较均匀，说明两实例集合内元素语义相似，那么集合间的语义相似度则较高。但在某些特殊情况下，当单个实例集合内元素相似性大，即它们在较小语义粒度上相似时，两实例集合间的实例对在相对较大的语义粒度上相似。

生成的聚类中来自不同集合的实例数量悬殊，甚至出现某些聚类是 A 或 B 的子集，则以两集合间实例对的相似度来衡量实例集合 A 和 B 的语义相似度，本文仅选取具有代表性的实例来计算集合的语义距离，而最具有代表性的是各聚类的中心点。此时，计算这些特殊聚类 C_i 的中心点 O_i 与其它有 A、B 元素共存聚类的中心点 O_j 的相似度 $\mathrm{sim}(O_i,O_j)$。另一个极端情况是每个生成的聚类均为 A 或 B 的子集的情形。设 $l+m=k$ 个聚类的中心点分别为 $O_1^A,O_2^A,\cdots,O_l^A,O_1^B,O_2^B,\cdots,O_m^B$，两集合的语义相似度仍用式(4)的聚类中心点间的相似性表示。

集合间的相似度 $\mathrm{Sim\_}O(A,B)$ 取其中最小的相似度值：

$$\mathrm{Sim\_}O(A,B)=\min(\{\mathrm{sim}(O_i,O_j)\mid \mathrm{sim}(O_i,O_j)>\theta\})$$
$$i=1,\cdots,l;j=1,\cdots,m \tag{4}$$

最后，属性实例集 A 和 B 的语义相似度用式(5)表示：

$$\mathrm{Sim}(A,B)=\max(Coupling(A,B),\mathrm{Sim\_}O(A,B)) \tag{5}$$

3 语义近似属性间的关系

当两个模式分别包含多个具有相似概念但语义不同的属性时，为了让模式 S_1 中语义近似的多个属性与模式 S_2 中语义近似的多个属性正确匹配，必须区分模式内语义相似的多属性间的微小语义差别及其关系。

3.1 区分语义近似的属性关系

语义近似属性之间的关系可以分为上下或前后关系和相等关系。

(a) 次序关系

值域重叠的多属性有一定的语义联系，从联系的语义内涵来看，可以分为几种情况：1)若被描述的联系是一种动作，那么这些属性则分别表示动作的主体和受体；2)整体与部分的关系；3)大小或顺序关系。这些联系一般都可以看作属性间的不同次序关系。一般可以把实例集合中语义相近的属性间的次序关系表示成层次结构，其中对于最上层结点(设它们是某属性 A 的部分实例)，不存在其它属性 B 的实例与之形成前后次序关系，这时称属性 A 是 B 的上级，B 是 A 的下级，而层次结构最底层的结点是下级属性 B 的实例。

(b)非次序关系

如果同一模式中出现了值域相似度特别大的不易分辨语义差别的多个属性，如：表示“装配件”与“零件”属性的后面各有一个属性分别说明零、配件的规格型号。此时，根据邻近原则，某属性的近邻属性有补充说明作用，有助于分析语义差别。如：航班时刻表中通过始发时间和到达时间的先后能确定始发地点和到达地点的先后次序。因此，可通过邻属性之间的语义关系来确定无次序关系的语义近似属性间的关系。

一般地，若 $S_1.a_s,S_2.a_t$ 语义近似，$S_1.a_{s+1},S_2.a_{t+1}$(不失一般性，考虑后相邻)属性语义也近似，如果能通过上述分析确定 $S_1.a_s,S_1.a_t$ 的上下级关系，那么认为它们的相邻属性也是层次关系。

3.2 判定语义近似属性的关系

下面就分别以不同的属性数据类型来说明同一模式 S_i 中语义近似多属性间的语义关系的判定。

日期型属性和数值型属性的实例间具有可比性，一般可以挖掘出这些类型的属性实例之间的大小关系来确定属性之间的次序。

定义 1 关系 R 有两个日期型或数值型属性 D_1,D_2，在 R 的某个实例 r 中全部或大部分元组 t 在这两个属性上的取值都满足 $t.D_1\leqslant t.D_2$，则称属性 D_1 是属性 D_2 的上级属性，属性 D_2 是属性 D_1 的下级属性，记作 $D_1\xrightarrow{precede}D_2$。

文本型数据的情形则较复杂，文本型数据通常是描述性内容，一般表示人名、事物名称或仅为单纯的描述文本信息，因为纯粹的描述文本信息内容变化较大，表达的语义比较丰富。为简单起见，这里仅讨论同一模式中出现的描述实体名称的多个文本型属性，分别对应着实体之间的主动与被动的关系、整体与部分的关系(is-a 或 part-of)。如：两个表示人名的文本型属性，这两个属性可能说明属性代表的实体联系是人与人之间指导与被指导的上下级关系。如：一个导师可以指导多个研究生，一个研究生可以指导本科生。

定义 2 从元组的两属性实例集合形成的多层次结构中，顶层结点若为属性 D_1 的取值，从顶层向下总能找到子树 T，T 中的最后一层结点即叶子结点个数大于 1，则称属性 D_1 是属性 D_2 的上级属性，属性 D_2 是属性 D_1 的下级属性，记作 $D_1\xrightarrow{precede}D_2$。

3.3 近似语义关系判定算法

设关系 R 有两个文本型属性 D_1,D_2，在 R 的某个实例 r 中存在一系列不相交的元组分组 $t\_Set_i\in\{t\_Set_1,t\_Set_2,\cdots,t\_Set_h\}$，这些元组两个属性上的取值形成深度为 h 的层次结构。顶层结点 $t\_Set_1.D_1$(即一组元组 t_Set_1 的属性 D_1 取值集合)没有更上层结点，且顶层每个结点都与不同的下层结点 $t\_Set_1.D_2$ 构成 t_Set_1 元组；除了顶层结点 $t\_Set_1.D_1$ 外，其他各层中间结点既可以作为一组元组 t_Set_i 的属性 D_1 的取值，又可以作为另一元组 t_Set_{i+1} 的属性 D_2 的取值，即 $t\_Set_i.D_2=t\_Set_{i+1}.D_1(i=1,\cdots,h-2)$。

要确定两个文本型属性之间的上下级(层)关系,需要在关系实例中能找到某些元组,它们的某一属性 D_1 的取值不作为其它元组的另一属性 D_2 的取值,即它具有顶层实例结点的特征,向上看它没有更上层结点,往下看分析它的下层结点个数 t_Set_2 是否大于 1,若大于 1,则可以判定 $D_1 \xrightarrow{precede} D_2$。如图 3 中粗线所示,两个元组中的属性 D_1 值都为 $g1$,属性 D_2 为{$g3$,$g4$},称为属性 D_1 值 $g1$ 确定属性 D_2 值的集合{$g3$,$g4$},记为 $g1 \rightarrow \{g3, g4\}$;否则,反复取得下层元组 t_Set_3,直到找到某个元组集合$|t_Set_i|>1$。

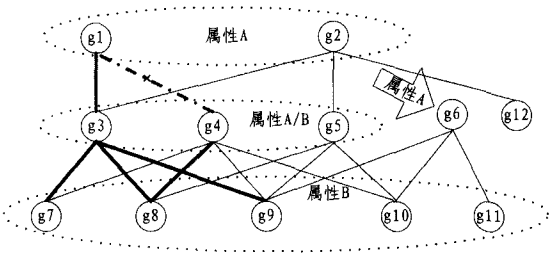

图 3　同一模式中相似属性的实例层次

```
intislayers(r,D1,D2)
//判断 r 中 D1 的一个取值总能对应多个 D2 的取值?
  intisprecede=0;
  float support;
    A<-select D1 from r;
    B<-select D2 from r;
  C=A∩B;A=A-C;B=B-C;
  //A、B 为两个值域范围相同的属性不同取值部分
    a=GetFirstElement(A);
while(a! =NULL)
  D2_V<-select D2 where D1==a from r;
  if(| D2_V |>1)//D1 属性值为 a 的元组数多于 1
    isprecede++;
  else if(| D2_V |==1)
      b=GetFirstElement(D2_V);
    if (b∉B)
        D1_V<-select D1 where D2==b from r;
        if(| D1_V |==1)
    // a->b 的取值个数比为 1∶1,由 b 向其它层深入分析
          a =b;
          while(a! =NULL)
            D2_V<- select D2 where D1==a from r;
            if(|D2_V|>1) isprecede++;
            else if((a=GetElement(D2_V)==NULL)
//已到达某一路经末端,不同元组内的属性取值为空
              break;
  a=GetNextElement(A);
  support=isprecede/|A|;//所有 A 中包含顶层结点的比例
```

若需要判断 D_2 的一个取值对应多少个 D_1 的取值,可以将算法 islayer 再调用一次,参数 D_1,D_2 对调位置。

算法仅需分析属性 D_1 的实例不同于 D_2 的实例集合中的部分实例在层次结构中作为顶层结点的情况,计算出其中 D_1 的实例作为顶层结点的比例。算法中,判断 D_1 取值为 a 时,D_2 是否可以对应多个不同的取值 b,若是,则说明可以将其作为顶层结点;否则需要深入更下层实例结点,以发现 1∶n 的实例映射关系,计算该实例的个数 *isprecede* 及其比例。

图 3 是一个复杂层次结构,可以将它看作零配件组装关系图。所有结点都表示大大小小的零配件,连接上下层结点的每条边表示下层零配件可以作为上层零配件的一部分被组装。其中,$g1$,$g2$,$g6$ 为层次关系中的最顶层结点,这 3 个实例值表示的零配件(即最终配件)仅在属性 A 中出现。中间层结点表示的零配件名称则均可作为 A、B 两属性的值,最底层则表示最基本的、不可再拆分的零件,对应着属性 B 的取值。若关系实例中没有 D_1 为 $g1$、D_2 为 $g4$ 的元组,则要找一个 3 层结点的子树才能确定次序关系,如图 3 中虚粗线所示。

4　模式匹配的生成

若两属性之间是匹配关系,那么它们的相似性也会比较大,但相似性大并不意味着两个属性是匹配关系。

一般来说,在两数据源 $S_1=\{a_1, a_2, \cdots, a_m\}$ 和 $S_2=\{b_1, b_2, \cdots, b_n\}$ 中,虽然属性对(a_i, b_j)的实例集合语义相似性最大,但不能简单地判定属性对(a_i, b_j)是正确的属性匹配关系。假定属性对(a_i, b_j)一旦被错误判定为属性映射关系,那么正确的属性匹配(a_i, b_k),(a_g, b_j)中的 b_k,a_g 将会与其他属性被错误判定是映射关系。模式匹配可以看作是两模式内所有属性对的语义相似性之和整体最大化时的属性对集合,即模式匹配是全局优化问题。两个模式 S_1,S_2 的匹配结果如图 4 所示,其中,双向虚线边表示基于实例计算得到的每个属性对之间的语义相似性。

$$M=\{(a_i, b_j) \mid \mathrm{Max}\sum_{i,j}\mathrm{Se\_Sim}(a_i, b_j)\} \tag{6}$$

式中,属性实集合 A,B 的语义相似度为 $\mathrm{Sim}(a_i, b_j)$。在匹配结果中每个属性最多只出现一次。

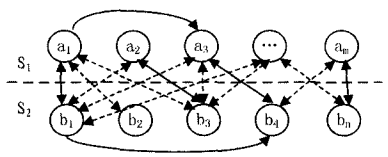

图 4　基于二部图的模式匹配

若把两模式中包含的属性看作两顶点集合,模式间属性对的相似性就是跨越两顶点集的边,由此构成了一个带权二部图。确定两模式之间匹配关系的问题就转换为在带权二部图上求最大匹配的问题。通过一些处理,将模式内具有约束关系的多个属性看作特殊顶点(图 4 中的 a_1,a_3);去掉权值很小的边,规模较大的二部图可分解为多个子图,再对子图分别计算其最大匹配。

Kuhn-Munkres 算法是计算带权二部图的最大匹配经典算法,利用这一算法生成模式匹配,且能使所有属性对相似度之和最大,时间复杂度为 $O((\max(m,n))^4)$,m 和 n 分别为两属性的属性个数。

5　实验

实验环境:处理器为 Intel(R) Core(TM)2 Duo CPU P8700,主频 2.53HZ,内存 2GB,32bit 操作系统。用 MatLab R2010a 在几种不同数据集上进行试验。

表 1 给出了两数据源 Books 基于聚类的语义相似性(每个数据源规模为 2000)。实验中,聚类个数 K 的确定很重要,K 太小,不足以体现实例分布的差异;K 太大,聚类时间较长。经反复试验,K=50 时较合适。实验结果表明,单个属性实例集合的聚类时间为 1.2s～1.5s,对 Books 进行模式匹配

时间为5s～7s。确定模式内语义近似属性间的关系所花费的时间较少，大约10ms。由于Books为出版物记录，在Web上可检索到book的相关信息，用GND来衡量实例相似性。当基于聚类的相似度θ小于0.3时，则需要再计算两聚类中心点间的距离作为语义相似性。表格中"—"表示无需计算或者不可计算。

表1 Books数据模式匹配

	书号	书名	作者	出版社	…
sh	0.2042,0.632	0.0001,0.031	—,0.014	0.0091,0	…
sm	0.0003,0.0	0.5904,—	0.8550,—	0.0162,0	…
zz	—,0.0301	0.2147,0.004	0.3000,0.7466	0.2809,0	…
cbs	—,0.0143	0.0904,0.031	0.3035,0	0.3072,0.8	…
…	…	…	…	…	…

另外，在商品信息集合和机械产品集合上也进行了实验。将本文所提出的方法与基于实例的DUMAS方法[6]在重复记录数量不同时的性能进行了比较，在重复记录比较少时，所提出的匹配方法MIC性能较好(见图5)。

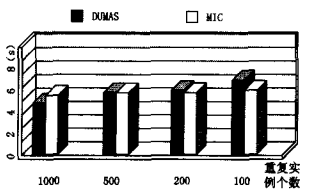

图5 重复实例个数与匹配时间的关系

将方法MIC与Cupid方法相结合，与基于实例提取部分函数依赖的PFD方法[8]进行比较，本文方法在查准率、查全率、全面性方面(见图6、图7)有所提高。

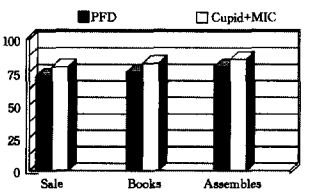

图6 PFD与Cupid+MIC查准率对比

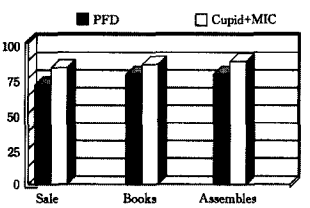

图7 PFD与Cupid+MIC查全率对比

通过设定合理的阈值来提高属性取值匹配度计算方法的效率，实验证明该方法能得到较全面的、正确的属性匹配关系。经验证，在缺少已知模式信息时，基于实例聚类分析的匹配方法是有效的。

结束语 模式信息包括各属性名称及其类型等属性的信息、模式的结构信息、数据实例。在进行匹配时，可以使用语义说明相互效果的多种模式信息，来提高匹配的精确度。本文所提出的模式匹配方法可以作为基于模式信息的一个辅助手段、有力补充，检验并纠正错误匹配，在缺少模式信息的场合，如果数据实例较全面体现了模式语义信息，借助领域知识，可利用本文所提方法来解决多数据源的模式匹配问题。通过把确定模式中属性对匹配的问题看作一个整体，从而避免将具有大量相同数据的属性实例集合对应的两个语义有差异的不匹配属性错误判定为匹配的。这一方法的优点在于即使数据实例中无重复描述的情形，但根据属性类型、实例的语义特征及部分属性之间的关系能分析出属性的语义相似性，可得到正确的模式匹配结果。

未来研究工作中将考虑模式匹配中存在的复杂匹配关系的发现，在模式的基础上解决实体识别的问题；并将就动态数据集成环境中数据源模式动态变化时的模式匹配方法进行研究。

参考文献

[1] Madhavan J, Bernstein P, Rahm E. Generic Schema Matching with Cupid[C]// VLDB 2001. 2001:49-58

[2] Hong Hai-do, Rahm E. COMA-A system for flexible combination of schema matching approaches[C]// VLDB 2002. 2002: 610-621

[3] Hernández M A, Popa L, Ho H, et al. Clio: A Schema Mapping Tool for Information Integration[C]// The8th International Symposium on Parallel Architectures. Algorithms, and Networks, ISPAN. 2005:11

[4] Zerdazi A, Lamolle M. Computing Path Similarity Relevant to XML Schema Mapping[C]// OTM Workshops. 2008:66-75

[5] Wang Ji-ying, Wen Ji-rong, Lochovsky F, et al. Instance-based Schema Mapping for Web Databases by Domain-specific Query Probing[C]// Proc. 30th VLDB. Conf. 2004:408-419

[6] Bilke A, Naumann F. Schema Mapping using Duplicates [C]// ICDE 2005. 2005:69-80

[7] Arndt C. Information Measures: Information and its description in Science and Engineering[M]. Berlin: Springer, 2001:370-373

[8] Li Guo-hui, Du Xian-kun, Hu Fang-xiao. A schema matching method based on Partial functional dependencies[C]// Frontier of Computer Science and Technology (FCST2008). 2008:131-138

(上接第127页)

[2] Zhang Qing-hua, Reeves D S, et al. Analyzing Network Traffic to Detect Self-decrypting Exploit Code[C]// Proceedings of the ACM Symposium on Information, Computer and Communications Security(ASIACCS). 2007

[3] Polychronakis M, Anagnostakis K G, Markatos E P. Emulation-based Detection of Non-self-contained Polymorphic Shellcode [C]// Proceedings of the Third Conference on Detection of Intrusions and Malware and Vulnerability Assesment(DIMVA). 2006

[4] Snort[EB/OL]. http://www.snort.org/

[5] Bro[EB/OL]. http://www.bro-ids.org/

[6] Pin[EB/OL]. http://www.pintool.org/

[7] Metasploit[EB/OL]. http://www.metasploit.com