

模式匹配方法研究^{*}

郑文怡, 鞠时光

(江苏大学 计算机科学与通信工程学院, 江苏 镇江 212013)

摘 要: 从模式匹配的定义开始介绍, 对已有的模式匹配方法进行分类, 探讨了这些方法适用的领域和所能发掘的信息, 区分了实例级和模式级、元素级和结构级以及基于自然语言和基于约束的匹配程序, 以期在比较不同模式匹配方法或研究新匹配算法以及实现模式匹配组件时有所帮助。

关键词: 模式匹配; 数据集成; 匹配程序; 数据库

中图法分类号: TP311 文献标识码: A 文章编号: 1001-3695(2006)02-0060-04

Research on Schema Matching Approaches

ZHENG Wen-yi, JU Shi-guang

(School of Computer Science & Communication Engineering, Jiangsu University, Zhenjiang Jiangsu 212013, China)

Abstract: First introduced the definition of schema matching, then classified the existing approaches on schema matching, and studied what kind of information they can exploit, distinguished between schema-level and instance-level, element-level and structure-level, and language-based and constraint-based matcher. It is expected to be useful when comparing different approaches of schema matching, when researching a new match algorithm, and when implementing a schema matching component.

Key words: Schema Matching; Data Integration; Matcher; Database

在大量的数据库应用中, 如面向 Web 的数据集成、电子商务、模式改进或迁入、应用改进、数据仓库、数据库设计、站点的建立与管理、基于组件的开发等, 都需要用到模式信息。而操作模式信息的一个最基本的操作就是匹配, 即将两个模式作为输入, 产生两个模式中互相对应的元素的映射。

目前, 模式匹配基本上都是通过一个图形工具手工实现的, 很明显, 手工指定模式匹配非常烦琐、耗时、错误多, 因此是一个很昂贵的过程。而且, 相对于执行匹配的数量而言, 与所做工作的层次是成正比的, 随着网络数据库以及电子商务集成的迅速增长, 问题也越来越多。因此迫切需要一种快速低成本的集成方法, 这就要求支持模式匹配的自动化。为了提供对其自动化支持, 我们想要一种通用的、用户可定制的匹配实现方法, 同时能跨越不同的应用领域, 它的实现也将便于建立包含自动模式匹配在内的专用应用工具。幸运的是, 在模式内容转换和集成、机器学习以及信息搜索等方面已经有许多关于模式匹配的研究工作。

本文的主要目标是考察已有的方法, 对其进行分类以概括其共有的特征。首先定义了模式匹配问题并简单概括了其应用, 然后对不同方法进行分类, 并详细描述了它们所能应用的结构, 最后对当前和将来的模式匹配工作作出总结。

1 模式匹配定义

一个模式由关联元素集, 如表、列、类或 XML 元素或属性等组成。匹配操作的结果就是一个映射, 我们将映射定义为一

个映射元素集, 每个映射元素说明了模式 S1 中的某些(个)元素映射为模式 S2 中的某些(个)元素^[1]。表 1 表示了两个表示进货单的模式 PO 和 PurchaseOrder 的部分信息, PO 和 PurchaseOrder 的一个映射可能包含一个映射元素 m, 它关联了元素 Lines. Item. Line 和元素 Items. Item. ItemNumber。

表 1 两个进货单模式

PO	Purchase Order
Lines	Items
Item	Item
Line	ItemNumber
Qty	Quantity
Uom	UnitOfMeasure

一般来说, 一个映射元素会用一个关联表达式来说明其语义(称为值对应^[2]), 即用来说明模式元素是如何关联的。映射表达式也许是有向的, 如某一函数通过映射元素从 s1 元素指向 s2 元素; 也可能是无向的, 即 s1 和 s2 元素相结合的一种关系。如 m 的映射表达式可能是 Lines. Item. Line = Items. Item. ItemNumber。

这样, 我们可以定义匹配为一个函数^[3], 以两个模式 S1 和 S2 作为输入, 返回的这两个模式之间的映射作为输出。匹配结果的每个映射元素指明了模式 S1 中某些元素逻辑上对应(匹配)于某些模式元素 S2, 这里对应的语义用映射元素的映射表达式来表示。而模式匹配的目标^[1, 4]是: 给出两个模式 S1 和 S2, 对 S1 中的每个元素 s1, 利用所有可用信息(包括模式信息、数据实例以及用户反馈信息等)在 S2 中找到与 s1 在语义上最匹配的元素。

本文主要集中在返回不包含映射表达式的映射匹配算法上, 所以我们在 S1 和 S2 的幂集上经常将映射表示为一个相似

关系。如表 1 中的模式匹配的结果可能是 Lines. Item. Line Items. Item. ItemNumber。映射结果的完整规范同样包含了每个元素的映射表达式, 即 Lines. Item. Line Items. Item. ItemNumber。在本文其余部分, 我们仅用到 来表示映射关系, 而省略映射表达式。

2 模式匹配方法分类

图 1 显示了具有实例的模式匹配方法分类。实现匹配时可能需要用到匹配算法或者匹配程序, 则需要依赖于应用领域或者模式类型来选择匹配程序。如果我们想要用到多个匹配程序, 主要基于以下两点假设: 已经实现了独立的匹配程序, 它们在各自单一的匹配标准上均能计算出映射; 存在组合匹配程序, 要么是在一个混合匹配程序中使用了多个匹配标准(如名称和类型相同), 或者是结合不同匹配算法产生的多个匹配结果。对于单一匹配程序, 我们考虑了以下几种正交的分类标准:

- (1) 实例 vs. 模式。匹配方法能够考虑实例数据(如数据内容)或仅考虑模式级信息。
 - (2) 元素 vs. 结构匹配。可对单个模式元素(如属性)或对混合元素(如复杂模式结构)进行匹配。
 - (3) 语言 vs. 约束。匹配程序可以使用基于自然语言的方法(如基于对模式元素的名称和文本描述)或者基于约束的方法(如基于键和关系)。
 - (4) 匹配基数。映射的每个元素可能将一个模式中的一个或多个元素与另一个模式的一个或多个元素匹配, 共有四种情况: 1: 1, 1: n, n: 1, n: m。另外, 在实例级可能有不同的匹配基数。
 - (5) 辅助信息。大部分匹配程序不仅依赖输入模式 S1 和 S2, 同时也需要辅助信息, 如数据字典、全局模式、预先匹配决策和用户的输入等。
- 值得注意的是该分类方法不是用于区分模式的不同类型(如关系、XML、面向对象等)或是它们的内在表示, 因为算法主要是依靠它们所能发掘的信息类型, 而非信息的表示。

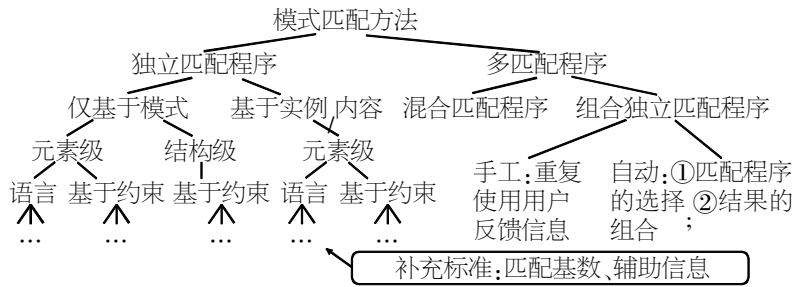


图 1 模式匹配方法分类

3 模式级匹配

模式级匹配程序仅考虑模式信息而非实例信息。可获取的信息包括模式元素常用的属性, 如名称、描述、数据类型、关系的不同类型(如 part-of, is-a 等)、约束以及模式结构。一般来说, 一个匹配程序会发现多个候选匹配, 对于每个候选匹配, 通常是通过在 0 到 1 范围内的标准化的数值来估计相似度, 以寻找出最优的匹配结果。

本文首先讨论元素 vs. 结构级匹配的主要情况, 然后讨论基于语言和约束的匹配程序, 最后概括出基于重用的辅助数

据, 如预定义的模式和匹配结果的方法。

3.1 元素 vs. 结构级

元素级匹配指的是对于第一个模式的每个元素, 该匹配决定第二个输入模式中匹配的元素。最简单的情况是仅考虑粒度最细级的元素, 我们称为原子级, 如 XML 模式中的属性或关系模式中的列。对于表 2 中显示的部分模式信息, 一个原子级匹配的例子就是 POShipToAddress. ZIP DeliverToAddress. PostalCode。元素级匹配并没有限制在原子级, 但是也可以应用到更高级元素, 如文件、记录、实体、类、关系表和 XML 元素。

元素级匹配可以通过类似于关系连接过程的算法加以实现。依赖匹配程序的类型, 在如名称、描述或模式元素的数据类型等性质的基础上得出匹配比较。对于模式 S1 中的每个元素, 根据匹配属性必须识别出另一模式 S2 中有相同或相似值的所有元素。一个类似于嵌套循环的连接过程的实现将 S1 中的每个元素和 S2 中的每个元素进行比较, 每对得出一个相似度矩阵。只有相似度值高于某一个临界值才考虑为匹配候选者。

另一方面, 结构级匹配指的是匹配在一个结构中同时出现的多个元素, 这类情况可能是依赖于要求的结构匹配的完整性和精确性。在理想情况下, 两个模式中结构的所有组件都完全匹配, 或者仅需要匹配组件中的某一部分(即部分结构匹配)。这两类情况如表 2 所示。

表 2 完全和部分结构匹配(例)

PO 模式元素	Purchase Order 模式元素	
POShipToAddress	DeliverToAddress	完全结构匹配 POShipToAddress 和 DeliverToAddress
Street	Street	
City	City	
State	USState	
Zip	PostalCode	
Contact	Customer	部分结构匹配 Contact 和 Customer
Name	CName	
Email	CEmail	
Telephone	CPhone	
Tax		

3.2 基于自然语言的方法

基于语言的匹配程序使用的是名称和文本(即单词或语句)来发现语义相似的模式元素, 这里主要讨论名称匹配方法。

基于名称的匹配是匹配具有相同或相似名称的模式元素。可以用多种方法来定义并计算名称的相似度, 包括以下几方面:

- (1) 名称的相等。一个重要的子情况就是从相同的 XML 命名空间获得名称等价性, 因为这样确保相同的名称确实具有相同的语义。
- (2) 预处理后规范名称表示的相等性。对于处理一些特殊的前缀/后缀符号尤为重要, 如 Cname- > customer name, EmpNO- > employee number。
- (3) 同义词的相等, 如 car automobile, make brand。
- (4) 上义词(若 Y 是 X 的一个类, 则称 X 是 Y 的一个上义词, 如“橡胶树”的上义词包括“树”和“植物”)的相等, 如 book 是 publication 的一个类, article 也是 publication 的一个类, 则表示 book publication, article publication 以及 book article。
- (5) 基于通用子串、发音 Soundex 代码(基于名称是如何发

音而不是如何拼写的一种代码) 等名称的相似性, 如 representedBy representative, ShipTo Ship2。

(6) 用户提供的匹配名称, 如 reportsTo manager, issue bug。

可以借助元素级匹配来驱动名称匹配, 如第 3.1 节中介绍的。在同义词和上义词的情况下, 类似连接的过程需要用到字典 D 作为进一步的输入。如果我们考虑以下表述的关系:

```
S1( name, ... ) // 每个 S1 模式元素的第一行
S2( name, ... ) // 每个 S2 模式元素的第一行
D( name1, name2, similarity)
/* name1-name2 的相似度, 介于 0 和 1 之间* /

用以下三种连接操作产生了一个所有匹配候选者列表:

select S2. name, S1. name, D. similarity
from S1, S2, D
where ( S1. name = D. name) and( D. name2 = S2. name) and ( D. si-
milarity > threshold)
```

这里假设 D 包含了相似名称上所有相关对的传递闭包。例如, 若 A-B-0.9, B-C-0.8, 则我们希望 D 也能包含 B-A-0.9, C-B-0.8, 以及 A-C- σ , C-A- σ 。直观上, 我们希望相似值 σ 是 $0.9 \times 0.8 = 0.72$, 但是这依赖于相似性的类型、同名异物词的用法以及其他一些可能的因素。例如, 也许 deliver-ship-0.9, ship-boat-0.9, 但是对于任何相似度值 σ , 都不存在 deliver-ship- σ 。

3.3 基于约束方法

模式经常会包含一些定义数据类型以及值域、唯一性、可选性、关系类型和基数等约束。如果输入的模式均包含了这样的信息, 匹配程序就能够利用这样的信息来判定模式元素间的相似性^[5]。例如, 相似性可以基于数据类型和域的等价、键的特征(如唯一性、主键、外键)的等价、关系基数等价(如1:1关系)或者派生关系。

通常可以用第 3.1 节介绍的用类似连接的元素级匹配方法来实现, 而这里则是用数据类型、结构、约束进行比较。数据类型和约束名称的等价(如 string varchar, primary key unique) 可以用一个专用的同义词表来提供。

表 3 给出的例子中, 根据类型和键的信息说明 Born 与 Birthday 匹配, Pno 匹配 EmpNo 或 DeptNo; 剩余的 Personel 元素 Pname 和 PDept 均为字符串, 因此有可能与 EmpName 与 DeptName 匹配。

表 3 基于约束匹配(例)

EMP 元素	PERSONNEL 元素
Employee	Personnel
EmpNo int, primary key	Pno int, unique
EmpName varchar(50)	Pname string
DeptNo int references Department	PDept string
Salary dec(15, 2)	Born date
Birthday date	
Department	
DeptNo int, primary key	
DeptName varchar(50)	

如表 3 所示, 单独使用约束信息经常会导致不完全的 $n:m$ 映射, 因为一个模式中可能有多个具有可比约束的元素。然而, 这种方法能够加以限制匹配候选者的数目, 而且可与其他匹配程序相结合(如名称匹配程序)。

某些结构信息可以解释为约束, 如内模式参照关系(如外

键) 以及邻接关系信息(如部分关系), 这类信息告诉我们在多级结构中哪些元素属于同一更高级模式元素。这类约束可以解释为结构而由此可以借助结构匹配的方法来寻求匹配。这样的映射不仅要考虑结构的拓扑问题, 还要考虑不同元素的类型(如属性、表/元素或域) 以及结构连接可能的不同类型(如部分或使用关系)。

3.4 匹配基数

一个 S1(或 S2) 元素可能与 S2(或 S1) 中的一个或多个元素匹配。这样, 在两个输入模式的元素匹配时有几个常用的关系基数, 即1:1以及面向集合的情况1:n, n:1和 n:m。

表 4 给出了以上四种情况的例子。第一行是假设没有其他 S1 元素匹配 Amount 以及没有其他 S2 元素匹配 Price 的1:1匹配。以前的工作主要都是集中研究1:1匹配, 因为很难自动判定其他情况的映射表达式。若要同时匹配多个 S1(或 S2) 中的元素, 需要用到表达式来说明这些元素是如何相关联的, 如第四行用到了一个 SQL 语句来连接两个表中的属性。在属性级对应的是一个 $n:m$ 关系(四个 S1 属性匹配两个 S2 属性), 而在结构级对应的是一个 $n:1$ 关系(S1 中的两个表 B 和 P 匹配 S2 中的一个表 A)。

表 4 匹配基数(例)

	匹配基数	S1 元素	S2 元素	匹配表达式
1	1:1 元素级	Price	Amount	Age = Year
2	$n:1$ 元素级	Price, Tax	Cost	Cost = Price $\times$ Tax
3	1: n 元素级	Name	FirstName LastName	FirstName, LastName = Extract(Name, ...)
4	$n:m$ 结构级	B. Title B. PuNo P. PuNo P. Name	A. BOOK A. Publisher	A. BOOK, A. Publisher = Select B. Title P. Name From B, P Where B. PuNo = P. PuNo

值得注意的是除了在模式级有匹配基数之外, 在实例级可能存在一些不同的匹配基数。如表 4 中的前三个例子, 一个 S1 实例与一个 S2 实例匹配(1:1, 实例级匹配)。第四行的例子对应一个 $n:1$ 实例, 即两个实例 B 和 P 对应一个实例 A。实例级 $n:m$ 匹配的例子就是将 S1 的单个 Sale 实例与 S2 中的不同 Sale 实例结合(如每个 Month, Quarter 等)。

3.5 重用模式和映射信息

我们前面已经讨论了除了输入模式之外的辅助信息的使用, 如数据字典、知识库、用户提供的匹配或不匹配信息。另一种利用辅助信息来提高匹配效率的方法是支持和发现通用模式组件的重用和预定义的映射。不仅重用全局定义的名称, 而且是整个模式段落, 包括如数据类型、键以及约束这类特征。对于经常用到的一些实体, 如地址、客户、员工、进货单以及配货单等, 这样做尤其有效, 可以把这类实体在一个模式库中定义并加以维护。但是整个世界不可能都符合这样一个模式, 它们应该由企业及其贸易伙伴、相关标准正文或相似组织加以定义以减少其可变度。用这种方法重用的元素应该包括其原始库的 ID, 如通过 XML 命名空间, 这样执行匹配时就能容易地判定并匹配来自同一库的模式段落和名称。

更通用的方法就是重用存在的映射。当将新的数据源集成到一个数据仓库或知识库时, 需要将不同但相似的模式匹配为同样的目标模式, 这时重用整个结构将非常有帮助。例如, 当一个模式 S1 必须映射成另一模式 S2, 而另一个模式 S 已经

映射成模式 S2, 这时重用整个结构就很有用。如果比起 S2, S1 与 S 更为相似, 尽管匹配有时不具传递性, 需要进一步考虑, 但是重用已存在的映射结果 Match(S, S2) 将可以简化匹配候选者的自动化产生过程, 尤其是允许重用手工指定的匹配。当然, 这些都是基于能探测出新的模式与预匹配的模式是相似的前提——本身就存在一个匹配问题。

4 实例级方法

实例级数据给出了模式元素真实内容的精确描述, 可以更进一步分析模式元素的内容和意义, 尤其当只有有限的可用模式信息时, 通常这类情况都是半结构化数据。在这种特殊情况下, 没有给出模式, 而只给出可由实例数据手工(如数据指南或从 XML 文档中构建的近似模式图) 构建的模式; 甚至当可获取充裕的模式信息时, 实例级匹配对于发现错误的模式信息的解释也是非常有效的。例如在看似同样合乎道理的模式级匹配中, 通过选择匹配实例更为相似的元素而避免产生歧义。

实例级匹配均可以应用到前面所讨论的大部分模式级匹配的方法, 尤其可适用于如下方面:

(1) 对于文本元素, 基于信息检索技术的语言特征化就是最适合的方法, 如在单词以及单词组合的相关使用频率等基础上抽取关键字和主题。例如在表 3 中, 从名称实例中我们也许可以断定 DeptName 比 EmpName 更匹配 PDept。

(2) 对更结构化些的数据, 如数值和字符串元素, 我们可以基于约束加以应用, 如值域和平均值或特征模式, 也许可以帮助识别电话数字、邮政编码、地形名称、地址, 日期表或与金钱有关的表(如基于币值特征)。在表 3 中, 如基于相似的值域, 实例信息也许能帮助将 EmpNo 优先映射为 Pno, 而 DeptNo 的值域是完全相反的。

实例级匹配也可以通过辅助信息加以执行, 如从匹配不同模式时获取的预映射。通过对单个关键字提供映射候选者, 这种方法对于匹配文本元素尤为有效。例如, 一个预分析可能会发现模式元素 CompanyName 和 Manufacturer 等经常会含有关键字 Microsoft, 对一个新的匹配, 如果模式 S2 的一个元素 X 经常包含术语 Microsoft, 甚至这个术语不经常出现在 S1 的实例中, 通过这一点也可以判定将 S1 中的 CompanyName 作为 X 的匹配候选者。

以上实例级匹配方法主要是为了发现元素级匹配, 若要找出模式元素集或结构集匹配需要对这些集合加以描述。显然, 主要的问题就是大量模式元素可能的组合数。

5 组合不同匹配程序

我们回顾了多种类型的匹配程序以及许多不同的变化, 它们各有优缺点。所以, 若一个匹配程序仅用一种方法似乎是不可能获得与结合多种方法同样理想的匹配候选者。而结合多种方法可以有两种方式: 混合匹配程序和组合独立执行的匹配程序。

混合匹配程序基于多种标准或信息资源(如利用结合了数据类型兼容性的命名空间和知识库的名称匹配) 直接组合多种匹配方法来决定匹配候选者, 这样比起多种匹配程序的独立执行能够提供更好的匹配候选者和更好的性能。效率之

所以能够得以提高是因为那些只符合许多标准之一的匹配候选者早期就能被过滤, 而且还可以解决那些需要考虑连接多个标准的复杂匹配(如表 3 中利用了键、数据类型和名称信息)。

另一方面, 包括混合匹配程序在内, 也可以组合独立执行的多个匹配程序的匹配结果。这就要求允许选择需要应用的匹配程序, 并且在应用领域和输入模式(如对于结构化和半结构化模式采用不同的方法) 的基础上寻求一种灵活的方式来结合它们。在最简单的情况下, 一个工具或用户一次仅调用一种专用匹配程序。用户对已发现的匹配候选者进行意见反馈(接受部分、拒绝剩余、手工添加匹配程序), 然后再调用另一个匹配程序来提高匹配结果。随后调用的匹配程序必须考虑到已经验证过的匹配以减少评估开销, 并且要集中在模式中尚未匹配的部分。

匹配实现本身可以选择和执行多个匹配程序并自动结合它们的结果。这里, 每个匹配程序产生的侯选匹配者的分级列表将被结合成为一个匹配结果列表。仅当某个候选匹配者的结合相似度高于某一个临界值时, 它才算是一个匹配结果。这个自动化方法减少了用户交互的数量, 但另一方面, 对于不同应用领域而言, 它因此也很难获得一个通用的解决办法。

6 结论

在许多数据库应用领域中, 如异构数据库集成、电子商务、数据仓库和语义查询等, 模式匹配是一个最基本的问题。本文详细研究了过去对模式匹配的研究工作并加以分类, 特别是区分了模式和实例级、元素和结构级、基于语言和约束的匹配程序, 并讨论其所能发掘的信息, 最后讨论了多匹配程序的结合, 希望对需要实现一个匹配算法的程序员以及寻找更有效和更合理的模式匹配算法的研究者们有所帮助。

过去的模式匹配工作主要集中在对某一特定领域研究, 既然它具有较强的广泛性, 在以后工作中我们将依旧会把模式匹配问题作为独立的问题来对待, 同时对不同方法的相关性能和精度进行定量分析, 以帮助我们了解已有方法的不足之处并加以改进。

参考文献:

[1] adhavan J, P A Bernstein, E Rahm. Generic Schema Matching Using Cupid[R] . MSR Tech. Report, MSR-TR-2001-58, 2001.

[2] Miller R, L Haas, M A Hernandez. Schema Mapping as Query Discovery[C] . VLDB, 2000. 77-88.

[3] Rahm E, P A Bernstein. On Matching Schemas Automatically[R] . SR Tech. Report, MSR-TR-2001-17, 2001.

[4] 周武, 金远平. XML 数据语义映射方法研究[J] . 计算机工程与应用, 2003, 39(28) : 88-91.

[5] Larson J A, S B Navathe, R ElMasri. A Theory of Attribute Equivalence in Databases with Application to Schema Integration[J] . IEEE Trans. Software Eng. , 1989, 15(4) : 449-463.

作者简介:

郑文怡(1981-), 女, 江西九江人, 硕士研究生, 主要研究方向为数据库集成; 鞠时光(1955-), 男, 江苏南通人, 教授, 主要研究方向为数据库、隐通道。