XML 数据关联挖掘技术*)

鞠时光 阎星娥 蔡 涛 辛 燕

(江苏大学计算机科学与通信工程学院 江苏 212013)

摘 要 XML 数据的灵活性、自描述性以及可扩展性使得越来越多的领域开始采用它作为主要的存储格式和传输媒介,因而产生了大量的 XML 数据,积累了丰富的信息。但是 XML 表述的数据特点比较复杂,这就为数据挖掘人员提出了新的挑战。文章从表述 XML 数据的模型开始介绍,按照模型对 XML 关联挖掘算法进行分类,介绍了主要的一些算法,并探讨了目前存在的问题和主要的发展方向。

关键词 原 XML 数据,数据挖掘,XML 数据库,半结构化模型

The Technology of Mining Association Rules in XML Data

JU Shi-Guang YAN Xing-E CAI Tao XIN Yan

(Dept. of Computer Science & Communication Engineering, Jiangsu University, Jiangsu 212013)

Abstract The flexibility, self-description and expansibility of XML data has made it develop dramatically and become a major standard for storing and exchanging information. The increasing amount of available XML data and complexity of characteristics of XML data pose new challenges to the data mining community. This paper first introduces the data model presenting XML data, then describes the main algorithm of Mining XML Association Rules classified data model, and finally explores the main problem now and develop direction future.

Keywords Native XML data, Data mining, XML database, Semi-structured model

XML 作为存储和交换信息的标准已成为主流[1,2],因而产生了大量的 XML 数据。为了更有效地利用这些数据,人们希望通过关联挖掘发现蕴含在数据中的有效知识。这就为关联挖掘提出了一个新的挑战。

目前的关联挖掘主要是在传统的关系数据库或者数据仓库之上进行的,而这两者都具有固定的数据模型,可以根据模型具体地描述特定的数据,并且数据和结构是分离的。而用 XML 表述的数据特点比较复杂,没有特定的模型可以描述,它通过将数据内容和能够表达这些内容的语义标签结合在一起,实现了数据层次的自描述,也就是说,在 XML 数据中,结构和数据的区分是模糊的。

因此,研究 XML 关联挖掘首要的是找到合适的数据模型,在模型基础上研究挖掘方法,进而发现有效信息。基于这一思路,本文首先简单介绍比较有代表性的半结构模型,然后根据模型分别介绍典型的 XML 挖掘方法,最后从方法本身出发,对当前的 XML 关联挖掘做了一个比较详细的综合描述。

1 半结构化模型与原 XML 数据规范

半结构化数据[3~6]它不象存储在传统数据中的数据拥有可以预先知道的固定的结构,而是只有数据出现的时候才能知道结构,也就是说结构和数据是融合在一起的。

目前,半结构化模型要数 Yannis 等人提出的 OEM 模型[7]最著名。OEM 最初是为解决异构数据源的信息集成而提出的。其基本思想是对每个需要交换的信息给定一个标记,这个标记表明此信息的含义。OEM 模型可以看作是一棵带标记的有方向的树。每个节点看作是一个对象。每个对象用一个四元组{*Label*, *Type*, *Value*, *Object-ID*}来表示。其中,Object-ID 是对象的唯一的可变长的标识;Label 是个可变长的字符串,描述对象含义,它出现在两个对象的连接边上;Type 表示对象值的类型;如 integer、string 等这些基本数据类型。Value 为对象值,可以是一个原子值,也可以是复合值。Value 若为复合值,表明该对象含有一系列子对象,它的值是一系列{*Label*, *Object-ID*}对的集合。也就是说,在 OEM 中,所有的中间节点均是复合值,叶子节点都是原子值。还可以看到,模式信息以 Label 形式与数据保存在一起,因而实现了半结构化数据的自描述。

OEM 模型不仅可以表示半结构化数据,还可以表示结构数据,目前已经有不少研究人员使用或者改进后使用这种模型。如斯坦福大学开发的专门用来管理半结构化数据的数据库管理系统 Lore[3,4]。

除了 OEM 模型外,还有其它一些半结构化模型,比如 Kohei Maruyama[6]等人提出的基于原型的模型(Prototype-based Model)。在这种模型中,类和实例以及方法和数据之间是没有区别的。每个新类通过拷贝原型来动态创建。新创建的类继承了原型的所有特征,还可以在本地根据新类自身特点去修改这些特征,甚至在运行过程中进行修改。它的这种动态拷贝以及灵活的环境正好适合于半结构化数据的自描述特征。

这些半结构化模型都能较好地描述半结构化数据,但是用来描述 XML 数据,还有些欠缺。XML 数据类似于半结构化数据,但有所不同。正如 Dan Suciu[8]指出的,它们都可以用一个带标记的图表示,都缺少模式信息,都是自描述的。另外,XML 是有序的,而半结构化数据是无序的,XML 可以混合文本和元素,XML 中还有属性、实体等许多其它成员,这也使得 XML 的处理更加复杂。

2 基于传统模型的 XML 关联挖掘

XML 很大的一个特点是其标签具有语义性,它的标签很大程度上类似于表结构中的属性字段,因此 D. Braga 等人将 XML 数据转换成关系表,在关系表上进行关联挖掘,最后将挖掘生成的规则再转换成对应的 XML 格式[10~12]。

*)本课题的研究得到国家 863 项目(2002AA412020)及江苏省自然科学基金(NO. BK200204)的资助。

算法整体上借助 Mine Rule 的思想[13]，扩展 Xquery[14]语法，以 Xpath[15]为技术支持，提出了一种叫做 XMINE[12]的工具。XMINE 语法如图 1 所示。

具体算法可分为如下三步。

步骤 1：预处理，XMINE 的表述语句被执行，将 XML 关联挖掘问题转换成对应的关系表形式。首先，根据 *ROOT* 行，利用 XPath 分析器选出被分析的事务，假定事务集合为 F_D，然后根据 LET 中的 *BODY* 和 *HEAD* 行选出有关项，假定项集合为 F_I，如果有 WHERE 语句，则要进行依据条件筛选集合 F_D 和 F_I；然后根据两个集合将挖掘问题转换为关系表，表的列对应于 F_I 中各个元素，表的行数对应于 F_D 的元素个数，如果 F_D 中某一项包含 F_I 中某一项，则两项在关系表中交汇的地方值为 1，否则为 0。

步骤 2：关联挖掘，在关系表上采用我们熟悉的任何一种传统的关联挖掘算法挖掘出关联规则。

步骤 3：后续处理，将从关系表中产生的关联规则转换为 XML 格式。这种转换与预处理中的转换一致，也就是说在预处理后，XML 中元素与关系表的对应关系被保存下来，根据这种对应关系做反向转换即可生成所需的 XML 关联规则。

```
XMINE RULE
IN document("www.atlantis.edu/research.xml")//IN 指定了数据源
FOR ROOT IN//People/*/Publications/*//ROOT 限定了被分析的一系列事务
LET BODY:=ROOT/Author // BODY 和 HEAD 分别限定了生成规则的 body 和 head
    HEAD:=ROOT/Author
WHERE ROOT/@year=2001
EXTRACTING RULES WITH //EXTRACTING 指明对规则的约束
SUPPORT =0.1 AND CONFIDENCE=0.2
RETURN//RETURN 表明最后生成规则的样式
〈RULE……〉
〈/RULE〉
```

图1　XMINE 语法

这种算法，实现简单，依赖现有的 Xpath 技术可以完成许多工作，同时在关系表上的关联挖掘可利用成熟的技术。但它只能解决一些简单的路径检索等问题。另外，可以用传统关系模型表示的 XML 文档必须是结构规则的，因此这就大大限制了这个方法的适用范围。

3　基于半结构模型的 XML 关联挖掘

大多数半结构模型将 XML 数据表示成一棵有序的、带标记的树，因此大多数在半结构模型上进行的 XML 关联挖掘算法首先在树上寻找频繁结构，然后在频繁结构基础上进一步与内容结合，通过构建结构层等方法挖掘关联规则。在结构层上挖掘关联规则等同于关系数据库中的关联挖掘，因此绝大多数 XML 挖掘算法主要研究频繁结构发现这一问题，也有人将这一问题称之为模式发现[6]。

3.1　典型对象(Representative Object)算法

最简单的模式发现是在一棵树中从单个对象出发，发现对象间的关联。S. Nestorov[16]等提出的典型对象算法就是以数据文档中的单个节点为处理对象，从此节点出发到子节点，同时记录遍历路经，不断迭代，直到找到有关此节点的所有的频繁模式。

这种方法提出了简单路径表示和数据路径两个基本概念：简单路径表示是由圆点分割的标记序列，数据路径则是由逗号分割的节点和标记交替形成的序列，其中，任两个相邻的节点之间，前一节点包含指向后一节点的一条路经，这条路径的标记为两个节点之间的标记。

典型对象算法的基础是 *Continuation*(O, *Path*)函数的集合。其中，O 代表一给定节点，Path 代表一段简单路径：$(l_1, l_2, \cdots, l_n)$，$n \geqslant 0$。

Continuation(O, *Path*)由以下两种标记组成：① 标记 l 属于集合，如果存在一条数据路径 p：$(o, l_1, o_1, \cdots, l_n, o_n, l, o_{n+1})$；② 空节点标记属于集合，如果数据路径 p：$(o, l_1, o_1, \cdots, l_n, o_n)$的 o_n 节点是一个原子节点。

也就是说，如果 *Path* 为空，那么任一个节点 o 的 *continuation* 就是 o 的所有连向子节点的边上的标记结合。如果 *Path* 的长度大于0，那么从节点 O 开始，在遍历 $n+1$步后，最后一步的边上的标记属于集合，并且如果第 n 步后，存在有节点是原子节点，那么空节点也属于集合。节点 o 的表达对象就是指 o 的 *continuation* 函数的实现。因此方法实际上就是通过求 o 的 *continuation* 函数来找到所有从 o 出发的路径上的频繁模式。

3.2　树表示(Tree-expression)法

典型对象算法虽然可以找到以某一个对象为主的关联模式，但实际中人们更关心多个对象间的关联。同时，K. Wang and H. Liu[17,18]等认为多个描述同类信息的多个文档的多个不规则结构之间包含着一些相同的结构信息，基于这一点他们提出了树表示法，在多个文档中寻找"典型"结构，也就是相似结构。

方法引入一种树表示方法表示 OEM 中的各个子树。设 te_i 是文档节点 o_i 的树表示，其基本思路如下：① 空模式是任何文档节点的树表示；② 对于一个文档节点 o，如果 $val(\&o) = \{l_1:\&o_1, \cdots, l_p:\&o_p\}$，并且$\{i_1, \cdots, i_k\}$是$\{1, \cdots, p\}$的子集，$k>0$，$\{l_{i_1}:te_{i_1}, \cdots, l_{i_k}:te_{i_k}\}$就是节点 o 的一个树表示。

为了表示文档节点的树表示，定义了路径表示(Path-expressions)，一个有 k 个叶节点的树表示是由 k 个路径表示 $p_1 \cdots p_k$ 的序列所表示的。

算法过程如下：

步骤1：计算频繁1-树表示。

步骤2：由频繁 $k-1$树表示生成频繁 k 树。在生成频繁 k 树的过程中，K. Wang 等采用了经典 Apriori[17]方式的一种技巧：任何频繁 k 树表示 $p_1 \cdots p_k$ 是由两个频繁 $k-1$树表示 $p_1 \cdots p_{k-2} p_{k-1}$和 $p_1, \cdots p_{k-2} p_k$ 组成的。整个算法过程以次定理为依据，从频繁1树表示开始，按从小到大的顺序生成频繁 k 树表示，并且一个频繁 k 树表示频繁当且仅当两个频繁 k−1 树表示频繁。此定理非常类似于经典 Apriori 算法中的性质：项集$\{i_1, \cdots, i_k\}$频繁当且仅当$\{i_1, \cdots, i_{k-2}, i_{k-1}\}$和$\{i_1, \cdots, i_{k-2}, i_k\}$都频繁。但实际上在处理过程中有很大的不同，首先路径表示序列之间不能用简单的包含与否确定是否是子集关系，其次，因为它是一个序列，所以不同的排列表示不同的子树。也正是因为这些不同导致了算法比 Apriori 算法要复杂许多。在树表示法中，由所有形如 $p_1 \cdots p_{k-2} p_{k-1}$和 $p_1 \cdots p_{k-2} p_k$ 的匹配路径组合生成 $p_1 \cdots p_k$ 时，假定两个匹配路径组合的叶子节点分别是 l 和 l'，采用通过 l' 扩展 l 的方法，相当于将 l 的兄弟节点作为它的子节点进行扩展。然后计算候选 k-树表示的支持度。

步骤3：剪枝。K. Wang 等提出的树表示法不仅可以适用于无环图，对有环图也适合。方法采用对相同标记添加不同上标的方法表示一个有环图中的相同节点，按遍历方向，先遍历的节点的上标小，后遍历的上标大。方法提出了"非自然"和"超非自然"两种树表示概念。如果一个树表示中所有非叶子节点的具有相同标记的 k 个分支上的上标，遵循从左到右从1到 k 的顺序，那么这个树表示就是自然的，否则就是非自然的。超非自然是指分支上的上标是无序的。在剪枝步骤将所有

非自然和超非自然的候选树表示也全部删除。这一点只要想到上标是为了区分有环中节点的先后遍历关系就很容易理解了。

步骤4：最大化。在介绍树表示法的开始，我们就提到它主要是用于发现多个描述同类信息的多个文档的相同的结构信息，因此在算法的最后步骤，它将所有可以包含在其它频繁树表示中的一些小的频繁树表示删除。其实这一步主要看实际需要，如果要保存所有的频繁树表示，这一步就可以略去。

树表示法还有一个突出的特点是引入了通配符。我们知道，XML 数据尽管包含一定的结构，但结构非常不规则，也不确定，因此能发现的完全一样的结构比较少，基于这种情况，方法引入通配符，其可以替代任何标记，以尽量多地发现典型结构。不过通配符的引入更导致了算法复杂程度的增加，并且 Gao Cong[20]等已经专门针对通配符的问题对其进行了改进。

3.3 增量算法 Freqt

可以看到，典型对象和树表示法都采用直接生成检验的策略，但是很多 XML 文档都比较大，因此这种策略的效率不是很好，考虑到这种情况，T. Asai[21]等提出了一种增量算法 Freqt。

算法的关键是最右扩展，一种只在树的最右分支上增加新节点生成新树的技巧，这种技巧保证了新树不破坏旧树的先序遍历，也就是说新增加的节点永远是最右叶子节点。并且由于方法是一个增量算法，因此另外一个关键就是通过频繁 $k-1$ 模式的出现次数计算候选 k 模式的出现次数，并且只要凭借频繁 $k-1$ 模式的最右叶子节点的出现频率，就可以得到候选 k 模式的出现次数，因此这种方法还有很大的存储优势，即只要保存频繁模式的最右出现频率就可以有效地实现增量频繁度的计算。

算法过程如下：

步骤1：计算频繁 l 模式，遍历树获得并保存每个模式的出现次数；

步骤2：在频繁 $k-1$ 模式的基础上进行所有可能的扩展，为了保证新树不破坏旧树的先序遍历，只进行最右扩展，即只在频繁 $k-1$ 模式的最右分支上增加新节点，生成候选 k 模式；

步骤3：计算候选 k 模式的支持度，产生频繁 k 模式。由于算法是增量算法，不需要重新计算候选 k 模式中所有节点的匹配情况，借助频繁 $k-1$ 模式的匹配记忆，只要考虑新增加的节点即可，因此一个关键就是如何保存频繁 $k-1$ 模式的匹配情况。借助最右扩展技巧，T. Asai 等证明并不需要保存树中所有节点的匹配情况，只要保存树的最右页节点的匹配情况即可。即可以根据频繁 $k-1$ 模式的最右页节点的出现次数计算候选 k 模式的最右页节点的出现次数，最后根据候选 k 模式的最右页节点的出现次数计算候选 k 模式的支持度，进而产生频繁 k 模式；

步骤4：继续上述步骤，直到生成所有的频繁模式。

3.4 TreeMiner 方法

同样采用最右扩展技巧的还有 Zaki[22]提出的 TreeMiner 方法。它是一种从一系列带标记的有序的树中发现频繁树的方法，即从森林中发现频繁树的方法。方法采用类似于 Freqt 中的最右扩展的枚举技巧，对有序树采用一种特殊字符串的表示方法，结合深度优先的搜索顺序以及一种垂直分解树的方法，实现了与数据量成线性比例的运算量。

方法为树中节点规定了一个范围：对于一棵以节点 n_l 为根节点的子树 T，n_r 是 T 的最右叶子节点，那么 n_l 的范围是一个 $[l,r]$ 的间隔。其中 l 是 n_l 的位置，而 r 是 n_r 的位置。为了有效的操作和计算，Zakj 还提出了一种字符串表示：它是树的标记的深度遍历，在深度遍历过程中，我们知道每个分支分别前向和后向遍历两次，为了区分两种方向，在每次后向遍历时，前面添加区分标志 -1。也就是说，这种字符串表示就是添加了后向遍历标志 -1 的树的标记的深度遍历。

方法假定树是以字符串表示形式存在，采用候选-剪枝的基本思路，由频繁 $k-1$ 树生成候选 k 树。但是候选树生成过程中，采用了前缀等价类的思想，即需要保证生成树与原先树共享相同的前缀，我们知道，新添加的元素必然是某个已有节点的子节点，但是并不是 $k-1$ 树中所有节点都可以作为新元素的父节点，只有在树的根节点到最右叶子节点的分支上的节点才可以，因为如果添加到不在这个分支上的节点下面，则必然会破坏共享相同前缀的规则。也就是说，生成候选树也就是在上一级的频繁树上增添一个节点，将新节点同原先的某个节点连接起来，因此有一个节点之间连接的规则，假设两个节点 (x,i) 和 (y,j) 相连接（其中，x,y 表示节点的 label，i,j 表示原先节点在深度遍历中的位置，$i=-1$ 表示单个节点）：

① 如果 $i=j$，那么在树只有一个节点的情况下，只将节点 $(y,j+1)$ 添加到树中，如果树不是只有一个节点，那么将节点 (y,j) 和 $(y,j+1)$ 都加到树中；

② 如果 $i>j$，那么将节点 (y,j) 添加到树中；如果 $i<j$，那么没有新的候选树生成。

为了快速地计算候选树的支持度，方法还引入范围列表的概念。在最开始，就求得每个树中每个节点的范围列表，每个节点的范围列表是由一系列三元组 (t,m,s) 表示的：t 表示候选树所在树的 id；m 表示前缀树中对应的节点的标记；s 是节点的范围。

每次计算某个候选树是否频繁的时候，利用节点的范围列表求候选树的范围列表，而判定一个候选树是否出现在某个树中，或者在一个树中出现了几次，只要计算 t 就可以了。

方法根据节点的范围列表求候选树的范围列表。我们知道，每个新加的节点，必然是原先某个节点的子节点或者兄弟节点，因此只要判断两个节点的关系就可以了，根据方法提出的节点的范围概念，判断两个节点的关系就很简单了。对于候选剪枝，也遵循经典 Apriori 算法的性质，频繁树的任何子树都必须是频繁的。

方法基本步骤为：

步骤1：计算频繁1树，也就是频繁节点；

步骤2：通过频繁节点间的连接生成频繁2树；

步骤3：对频繁2树中的每一项，添加新的元素生成候选3树，剪枝，生成频繁3树；

步骤4：以此类推，直到频繁2树中的所有项都被处理。

4 基于原 XML 数据的关联挖掘

笔者同事也曾经设计过一种适合于 XML 数据的数据模型：扩展关系模型。考虑到 XML 数据分为结构数据和非结构数据两部分的特点，将 XML 数据分两部分存储，对结构数据采用传统的关系数据库存储，对非结构化数据则采用对象型存储。

其实，采用哪种数据模型并不是绝对的。因为 XML 数据本身就是多种多样的，我们知道，XML 文档一般分为两类：以数据为中心和以文档为中心。以数据为中心的 XML 文档一般含有比较规则的结构，一般都是由数据库生成，而以文档为中心的文档一般是人为创建的，结构比较混乱，这在 Ronald Bourret[9]的文章中提得非常清楚。因此在实际处理当中，对以数据为中心的 XML 文档，传统模型、半结构模型以及其它

一些数据模型都可以表示。而对以文档为中心的 XML 文档，一般只能用改进后的半结构模型表示，并且有时可能还需要人为规定的限制条件。

扩展关系模型考虑到半结构化数据中一部分为结构数据，一部分为非结构化数据的特点，将整体的半结构化数据进行拆分，分解为结构数据和非结构数据分别进行保存。因此 Lisa Singh[23,24]等在这个模型上进行关联挖掘时，将结构数据保存在关系表中，对非结构数据进行预处理，从中抽取出一种扩展概念层次结构，进而联系结构数据和这种扩展概念层次结构，产生关联规则。扩展概念层次结构之所以被称为扩展，是因为原先的概念层次结构是一棵树，保证只有一个根节点，但这种方法中的结构可能是一个图。其中，每个概念保存着指向与它相关概念的指针以及指向包含此概念文档的指针，同其它概念有三种关系：Parent、Child 和 Sibling。

这种方法有两个关键点：扩展概念结构的生成和存储。扩展概念结构产生于类似纯文本的文档部分中，因此它的建立是一个非常复杂的问题，Lisa Singh 采用离线处理的方式，作为数据挖掘的预处理部分，在领域专家的帮助下手工建立。关于它的存储，由于它主要是用于规则的生成，因此有效的存储是关键，Lisa Singh 等采用动态哈希表技术，采用了一个线性哈希函数。存储在哈希表中的每个入口概念包含两类指针，一类是包含此概念的文档指针，另外一类是与此概念相关的概念指针，同时指出相关概念与入口概念的关系，Parent、Child 或 Sibling。

算法主要寻找结构属性和非结构属性之间的关系，因此，首先从结构数据中找出结构属性所在的文档集合 A，然后从扩展概念结构中找出非结构概念 c_1 所在的文档集合 B，求 A 和 B 的交集 C，从扩展概念结构中找出与 c_1 相关的概念 c_r 所在的文档集合 D（根据给定关系 P、C 或者 S 寻找），求 C 和 D 的交集 E，计算规则的置信度 $=\#E/\#C$，如果置信度大于最小置信，返回规则，继续处理所有未被处理的 c_r，直到结束。

对于这种方法，正如 Lisa Singh 等人自己提出的，扩展概念层次结构的生成是一个比较难的问题，期待不断发展的 Agent 等人工智能技术可以辅助解决这一问题。另外一个就是文中只提到了一个结构属性和多个非结构概念生成的关联规则，可以考虑多个结构属性和多个概念生成的规则。对于这一点，笔者认为不难，这类似于混合维关联规则，可以参见文[25]。

5 从挖掘内容看 XML 关联挖掘

在前面的文章中，我们根据依据的模型对算法进行了简单的分类。其实从算法处理的 XML 数据我们可以发现各个算法又有区别。我们知道，XML 作为一种半结构化数据，本身分为结构数据和非结构数据两部分。实际上，并不是所有的算法对两部分数据都可以进行挖掘。比如，D. Braga 等人提出的 Xmine 工具主要通过将 XML 数据转换到关系表中进行关联挖掘，显然它只能挖掘 XML 的结构数据。因此，根据各个算法所能处理的 XML 数据部分，可以将算法分为三类：挖掘结构数据、挖掘非结构数据及挖掘结构和非结构数据。

所有采用关系模型表示的 XML 关联挖掘算法，由于关系模型本身的特点，决定了它只能挖掘结构数据部分。而对于以半结构模型为表示方法的关联挖掘算法，一般分两步走，首先进行模式发现，在模式发现的基础上构建结构层，进而进行关联挖掘，而构建结构层的过程就是根据发现的频繁结构创建关联表，即标签对应属性，标签间的内容对应属性值。因此也属于挖掘结构数据。

挖掘非结构数据的算法在 XML 关联挖掘中比较少，这是因为在 XML 中挖掘非结构数据类似于挖掘纯文本数据，因此这一类算法在文本挖掘领域中比较多。

对于挖掘结构和非结构数据的算法，目前还比较少。Lisa Singh 等采用扩展关系结构模型属于这一类。除此之外，K. Taniguchi[26]所提出的路径表达法也属于这一类。算法采用半结构模型表示 XML 文档，对 XML 数据的挖掘从节点和纯文本两部分分别进行处理。从节点部分发现频繁节点序列，从纯文本部分发现一种称作词-关联模式（word-association pattern）[27]，而最终所产生的关联路径由两部分生成的模式组合而成。方法将一个 HTML 文档称作一页，页对应一棵有序的、带标记的树，这棵树可以表示成一系列路径的组合。方法规定：树由元素节点和文本节点组成，每个节点有名和值两个属性，元素节点的名就是元素名，值为空，文本节点的值就是文本本身，而所有文本的名都是保留字符 #Text，路径由节点序列组成，但是每条路径至多只能在最后包含一个文本节点。但是，一个中间节点的值则是所有子节点的值的组合。对于路径，名就是路径上所有节点的名的组合，值就是最后一个节点的值。而所谓的路径表达就是由节点名组成的一个序列。方法分两步处理的基本思想就是在树中的一系列路径中，寻找最频繁的节点序列，也就是频繁的路径表达，同时寻找最后一个节点的词-关联模式，节点序列和词-关联模式的频繁度的和超过一定界限的，就是最终的关联路径。

结论 尽管我们按照模型可以将算法分成几类，但实际中每个模型并不是孤立使用的，扩展关系结构本身就是关系模型和对象模型的结合。而采用半结构模型的算法，在挖掘到频繁结构后依然要和关系模型结合，因此在研究中，应针对实际需要，寻找最合适的模型，最有效的算法。

对 XML 关联挖掘已经有了很多算法，并各有特点，但是，它们还存在一些共性的问题：

① 从所挖掘 XML 的数据来看，这些算法大多只挖掘了 XML 数据的结构部分，而非结构数据这一相当大的部分被忽略掉了。

② 所有的算法都是从最简单的方面开始考虑，对于 XML 数据的属性、实体引用等等复杂部分均没有考虑。

③ 对基于一定约束的关联挖掘都没有涉及，对算法的效率考虑得很少。

在未来的研究中，我们还需要针对以上这些问题，在 XML 关联挖掘等方面做更深入的研究。比如将 XML 挖掘和文本挖掘结合，将结构数据和非结构数据结合，挖掘更多信息；对表示 XML 的模型进行改进，考虑属性和实体引用等数据；采用增量算法等提高挖掘效率，根据用户兴趣研究基于一定约束的关联挖掘等等。

参 考 文 献

1 World Wide Web Consortium. Extensible Markup Language (XML) Version 1.0 (W3C Recommendation). http://www.w3c.org/xml/, Feb. 1998

2 W3C, Extensibe Markup Language (XML) 1.0 (Second Edition), W3C Recommendation, Oct. 2000. http://www.w3.org/TR/REC-xml

3 Abiteboul S, et al. The lorel query language for semi-structured data: [Technical report]. Dept. of Computer Science, Stanford University, 1996. Available by anonymous ftp to db.stanford.edu

4 McHugh J, et al. Lore: A Database Management System for Semi-structured Data. http://citeseer.nj.nec.com/cache/papers/cs/553/http:zSzzSzwww-db.stanford.eduzSzpubzSzpaperszSzlore97.pdf/mchugh97lore.pdf

5 Goldman R, McHugh J, Widom J. From Semi-structured Data to XML: Migrating the Lore Data Model and Query Language. http://citeseer. nj. nec. com/cache/papers/cs/24625/http: zSzzSzxml. coverpages. orgzSzLore-WebDB99. pdf/goldman99from. pdf.

6 Maruyama K, Uehara K. Mining Association Rules from Semi-structured Data. www. ai. cs. scitec. kobe-u. ac. jp/report/maru-199912. pdf

7 Papakonstantinou Y, Garcia-Molina H, Widom J. Object exchange across heterogeneous information sources. In: Proc. of the Eleventh Intl. Conf. on Data Engineering, Taipei, Taiwan, Mar. 1995. 251～260

8 Prof. Dan Suciu. Managing XML and Semistructured Data: lecture 2 : XML. http://www. cs. washington. edu/homes/suciu/COURSES/590DS/02xmlsyntax. htm. Spring 2001

9 Bourret R. XML and Databases. http://www. rpbourret. com/xml/XMLAndDatabases. htm. July, 2003

10 Braga D, Campi A, Klemettinen M, Lanzi P L. Mining association rules from xml data. In: Proc. of the 41h Intl. Conf. on Data Warehousing and knowledge discovery(DaWak 2002)Sep. Aixen-Provence, France, 2002. accepted.

11 Braga D, et al. Discovering interesting information in xml data with association rules:[Technical Report 2002-15]. Dipartimento di Elettronica e Informazione-Politecnico di Milano, 2002

12 Bragal D, et al. A Tool for Extracting XML Association Rules. In: Proc. of the 14th IEEE Intl. Conf. on Tools with Artificial Intelligence(ICTAI'02)2002

13 Meo R, Psaila G, Ceri S. A new sql-like operator for mining association rules. In VLDB'96, Mumbai(Bombay), India, 1996. 122～133

14 World Wide Web Consortium. XQuery 1. 0: An XML Query Language(W3C Working Draft). http://www. w3c. org/TR/2001/WD-xquery-20011220, DEC. 200

15 World Wide Web Consortium. XML Path Language(XPath)Version 1. 0(W3C Recommendation). http://www. w3c. org/tr/xpath/, Nov. 1999

16 Nestorov S, Ullman J, Wiener J, Chawathe S. Representative Objects: Concise Representations of Semi-structured Hierarchal Data. In: Proc. of 13th Intl. Conf. on Data Engineering, 1997. 79～90

17 Wang K, Liu H. Schema discovery for semi-structured data. In: Intl. Conf. on Knowledge Discovery and Data Mining, Newport Beach, Aug. 1997. 271～274

18 Wang K, Liu H. Discovering Typical Structures of Documents: A Road Map Approach. In: Proc. of 21st Annual Intl. ACM SIGIR Conf. on Research and Development in Information, 1998. 146～154

19 Agrawal R, Imielinski T, Swami A. Mining association rules between sets of items in large databases. In: Proc. of the ACM SIGMOD Conf. on Management of data, 1993. 207～216

20 Cong G, Yi L, Liu B, Wang K. Discovering Frequent Substructures from Hierarchical Semi-structured Data. www. siam. org/meetings/sdm02/proceedings/sdm02-11. pdf.

21 Asai T, Abe K, Kawasoe S, Arimura H, Sakamoto H, Arikawa S. Efficient Substructure Discovery from Large Semi-structured Data. In: Proc. the 2nd SIAM Int'l Conf on data mining (SDM2002). 2002. 158～174

22 Zaki M J. Efficiently Mining Frequent Trees in a Forest, Computer Science Department, Rensselaer Ploytechnic Institute: [PRI-TR01-7-2001]. 2001. http://www. cs. rpi. edu/～ zaki/PS/TR01-7. ps. gz

23 Singh L, Scheuermann P, Chen B. Generating association rules from semi-structured documents using an extended concept hierarchy. In CIKM, 1997. 193～200

24 Singh L, Chen B, Haight R, Scheuermann P, Aoki K. A Robust System Architecture for Mining Semi-structured Data. In: Proc. of 13th Intl. Conf. on Data Engineering, 1997. 79～90

25 Xin Y, Ju S. Mining Conditional Hybrid-dimension Association Rule on the basis of Multi-dimension Transaction Database. In: The Second Intl. Conf. on Machine Learning and Cybernetics the IEEE Systems, Man and Cybernetics Technical Committee on Cybernetics, Xi-an, China, Aug. 2003

26 Taniguchi K, Sakamoto H, Arimura H, Shimozono S, Arikawa S. Mining Semi-Structured Data by Path Expressions. In: Proc. The 4th Int'l Conf. on Discovery Science, LNAI 2226, 2001. 387～388

27 Shimozono S, Arimura H, Arikawa S. Efficient discovery of optimal word-association patterns in large text databases. New Generation Computing, 2000, 18: 49～60

(上接第22页)

16 Pal P P, Webber F, Schantz R E, et al. Intrusion Tolerant Systems. IEEE Information Survivability Workshop(ISW-2000)

17 Wang Xunhua. Intrusion- Tolerant Password-Enabled PKI. http://middleware. internet2. edu/pki03/presentations/secondpki. pdf, 2003

18 Stavridou V, Dutertre B, Riemenschneider R A, et al. Intrusion Tolerant Software Architectures. In: Proc. of the DARPA Information survivability Conference and Exposition(DISCEXII'01)

19 Madan B B, Goseva-Popstojanova K, Vaidyanathan K, Trivedi K S. Modeling and quantification of security attributes of software systems. In: Proc. Int. Conf. DSN, (IPDS stream), volume 2, 2002. 505～514

20 Valdes A, Almgren M, Cheung S, et al. An Adaptive Intrusion-Tolerant Server Architecture. http://www. sdl. sri. com/users/valdes/DIT_arch. pdf, 2002

21 Cachin C, Poritz J A. Secure Intrusion-tolerant Replication on the Internet. In: Proc. of the Intl. Conf. on Dependable Systems and Networks(DSN'02)

22 Hiltunen M A, Schlichting R D, Ugarte C A. Building Survivable Services Using Redundancy and Adaptation. IEEE transactions on computers, 2003, 52(2): 181～194

23 Wang F, Gong F, Sargor G, et al. SITAR: A Scalable Intrusion Tolerance Architecture for Distributed Server. IEEE SMC Information Assurance Workshop' 01

24 Wolf A L, Heimbigner D, Bend J K. Don't Break: Using Reconfiguration to Achieve Survivability. IEEE Information Survivability Workshop(ISW-2000)

25 Valdes A, Almgren M, Cheung S, et al. Dependable Intrusion Tolerance: Technology Demo. In: Proc. of the DARPA Information Survivability Conference and Exposition(DISCEX'03)

26 Kihlstorm K P, Moser L E, Melliar-Smith P M. The SecureRing Group Communication System. ACM transactions on Information and System Security, 2001, 4(4): 371～406

27 Dutertre B, Sa ï di H, Stavridou V. Intrusion-Tolerant Group Management in Enclaves. DSN'01

28 Reiter M. A Secure Group Membership Protocol. In: Proc. of the IEEE Symposium on Research in Security and Privacy, 1994. 176～189

29 Ramasamy H V. Group Membership Protocol for an Intrusion Tolerant Group Communication System: [MS thsis]. University of Illinois at Urbana-Champaign, 2002

30 Ramasamy H V, Pandey P, et al. Quantifying the Cost of Providing Intrusion Tolerance in Group Communication System. In: Proc. of the Intl. Conf. on Dependable Systems and Networks (DSN'02)

31 Sanders W H, Cukier M, Webber F, Pal P, et al. Probabilistic Validation of Intrusion Tolerance. http://www. dist-systems. bbn. com/papers/2002/SAN/02SAN02. pdf, 2002

32 Goseva-Popstojanova K, Wang Feiyi, Wang Rong, et al. Characterizing Intrusion Tolerant Systems Using A State Transition Model. (DISCEXII'01). http://panda. ece. utk. edu/～fwang2/papers/darpa00. pdf, 2002

33 Madan B B, Trivedi K S. Security modeling and quantification of intrusion tolerant system. http://srel. ee. duke. edu/PAPERS/Madan-FA2002240. pdf, 2002