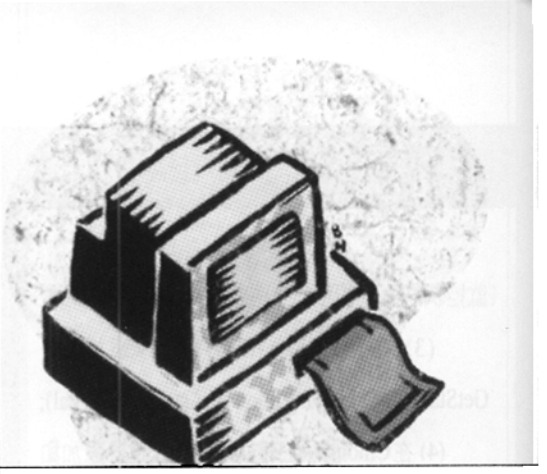
Semi-structure Data Model and the technique of Data Mining for Internet-oriented



陈一明 （广东茂名学院师范学院 525000）

基于XML数据模型及面向Web数据挖掘技术

摘要：本文主要对Web上的数据结构特征及其数据挖掘技术进行分析，并就把XML作为一种半结构化的数据模型实施查询与模型抽取，从而完成面向Web数据挖掘的方法，且结合SQL Server 2000的应用及实现智能查询应答的思想进行探讨。

关键词：XML 数据挖掘 数据模型 智能查询应答

数据挖掘已成为数据库研究、开发和应用最活跃的分支，也是一个多学科交叉的领域。但随数据量的不断增加，我们有一种将被数据淹没的感觉。人们已不满足仅对数据进行简单的查询，而是希望计算机能帮我们分析数据、理解数据和作出决策等。如何利用所面对的大量数据（特别是互联网上的数据），从中发现有用的知识是我们的新课题。

1 数据挖掘与半结构化数据模型

数据挖掘是从大型数据库或数据仓库中发现并提取隐藏在其中的信息或者知识的过程。其目的是帮助人们寻找数据之间的关联，发现被忽略的对于预测趋势和决策行为十分有用的要素。数据挖掘技术是人工智能领域中的知识发现技术和数据库技术共同发展的结果，其实质的内涵是在一个已知状态的数据集上，通过设定一定的学习算法，从数据集中获取所谓的知识。

传统意义的数据库、数据仓库和Web中的数据是我们所拥有的数据资源，也是进行数据挖掘的基础。传统数据库中的数据结构性很强，其中的数据为完全结构化的数据。数据仓库是由数据库导出的用于分析型的数据环境。如果把Web网站上的信息看作一个更大、更复杂的数据库，每一个站点就是一个数据源，每个数据源都是异构的，而每一站点之间的信息和组织都不一样，这就构成了一个巨大的异构数据库环境。

传统的数据库都有一定的数据模型，可以根据模型来具体描述特定的数据，同时可以很好地定义和解释相关的查询语言。而Web上的数据非常复杂，没有特定的模型描述，每一站点的数据都各自独立设计，并且数据本身具有自述性和动态可变性，其结构也不可琢磨，我们将它称为半结构化的数据。如我们的简历，其中有如性别、年龄等结构化的信息，也有如个人特长的描述等非结构化的信息。

如果想要利用Web上的数据进行数据挖掘，必须先要研究站点之间异构数据的集成问题，只有将这些站点的数据都集成起来，提供给用户一个统一的视图，才有可能从巨大的数据资源中获取所需的东西。其次，还要解决Web上的数据查询问题，因为如果所需的数据不能很有效地得到，对这些数据进行分析、集成、处理就无从谈起。针对Web上的数据半结构化的特点，寻找一个半结构化的数据模型则成为了解决上述问题的关键所在。此外，除了要定义这样一个半结构化数据模型外，还需要一项技术能够自动地从现有数据中将这个模型抽取出来，这就是所谓的模型抽取技术。因此半结构化数据模型及其抽取技术是面向Web的数据挖掘技术实施的前提。

2 XML与面向Web的数据挖掘

XML是一种可扩展的标记语言，也是一种半结构化的数据模型。XML的本身是自描述的，包含很多元数据，而且它同时可以扩展或增加元素的元数据；XML解决了HTML不能解决的Internet发展速度快而接入速度慢和可利用的信息多，但难以找到自己需要的那部分信息的问题。XML具有简单、开放、高效且可扩充和国际化等特点，这些特点决定了其卓越的性能表现。因此，XML将成为在Web上的数据查询和模式抽取的一个重要途径。

XML具有可升级的三层架构模型，数据的集成、发送、处理和显示是下面过程中的每一个

步骤如图1所示。

XML能够使不同来源的结构化的数据很容易地结合在一起。软件代理商可以在中间层的服务器上对从后端数据库和别处来的数据进行集成，然后，数据就能被发送到客户或其他服务器做进一步的集合、处理和分发。XML的扩展性和灵活性允许它描述从搜集的Web页到数据记录等不同种类的数据，从而可通过多种应用得到数据。同时，由于基于XML的数据是自我描述的，用户可以方便地进行本地计算和处理，XML格式的数据发送给客户后，客户可以用应用软件解析数据并对数据进行编辑和处理。XML文档对象模式(DOM)允许用脚本或其他编程语言处理数据，数据计算不需要回到服务器就能进行。XML可以被利用来分离使用者观看数据的界面，使用简单灵活开放的格式，可以给Web创建功能强大的应用软件。另外，数据发到桌面后，能够用多种方式显示。

XML可以通过以简单开放扩展的方式描述结构化的数据。由于数据显示与内容分开，XML定义的数据允许指定不同的显示方式，使数据更合理地表现出来。通过XML，数据可以粒状地更新，每当一部分数据变化后，不需要重发整个结构化的数据。变化的元素必须从服务器发送给客户，变化的数据不需要刷新整个使用者的界面就能够显示出来。XML也允许加进其他数据，加入的信息能够进入存在的页面，不需要浏览器重新发一个新的页面。XML是用户与不同数据源进行交互的标准语言，客户收到数据后可以进行处理，也可以在不同数据库间进行传递。XML解决了数据的统一接口问题，但与其他的数据传递标准不同的是，XML并没有定义数据文件中数据出现的具体规范，而是在数据中附加TAG来表达数据的逻辑结构和含义。

XML应用于将大量运算负荷分布在客户端，即客户可根据自己的需求选择和制作不同的应用程序以处理数据，而服务器只须发出同一个XML文件。应用XML将处理数据的主动权交给了客户，改变传统的"Client/Server"由客户向服务器发出不同的请求，服务器分别予以响应的工作方式：服务器所作的只是尽可能完善、准确地将数据封装进XML文件中。XML的自解释性使客户端在收到数据的同时也理解数据的逻辑结构与含义，从而使广泛、通用的分布式计算成为可能。XML还被应用于网络代理，以便对所取得的信息进行编辑、增减以适应个人用户的需要。

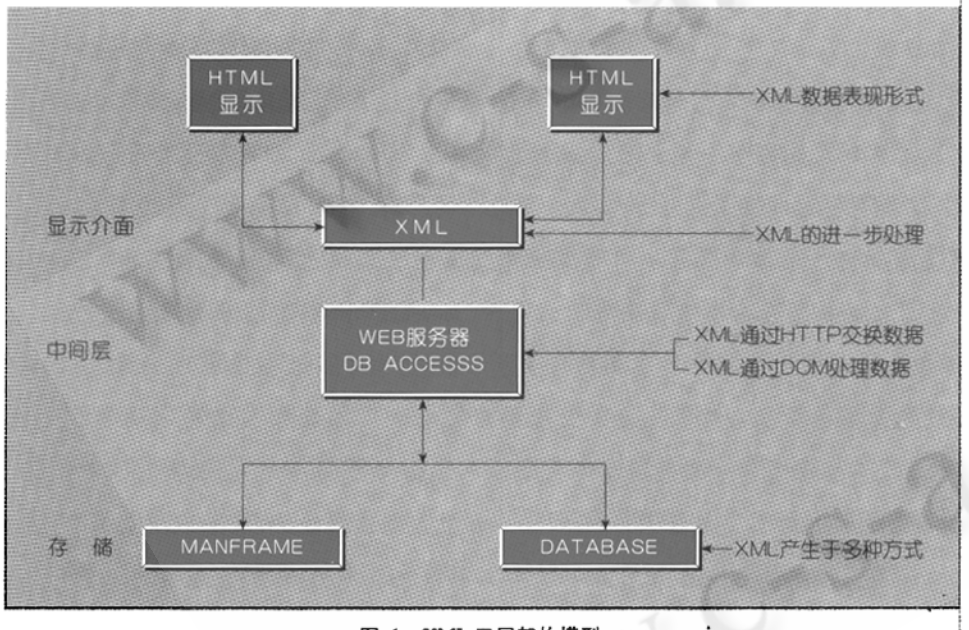


图1 XML三层架构模型

3 基于SQL Server 2000的XML的应用

支持XML的数据库能够在XML文档和数据库之间进行数据的传输。通常是设计成为能够存储和提取以数据为中心的文档，在把XML文档进行解析以后，存储到相应的表格中。当然，也可以存储以文档为中心的文档，就是把整个文档作为一个单一的表中的一个字段，然后通过文本检索机制进行查询。因为许多数据库现在能够把内容发布到网站上，基于XML的数据库和XML服务器之间的差别就变得很模糊。Microsoft SQL Server 2000的新特性之一就是支持XML。

3.1 使用HTTP访问SQL Server 2000数据库

可以通过HTTP方式直接访问具备XML功能的SQL Server2000数据库。访问SQL Server 2000的HTTP允许进行如下活动

(1) 在URL中直接使用SQL语句。例如：http://IISServer/nwind?sql= SELECT+*+FORM+Customers+FOR+XML+AUTO&root=root该语句中FOR XML子句表示返回的结果是一个XML文档，而不是标准的结果集。Root参数表明了单个的顶层元素。

(2) 直接在URL中指定模板。模板包含了一条或多条SQL语句的XML文档，它允许从一个有效的XML文档中把数据集中到一起来获取，这样就不需要直接在URL中指定查询语句。

(3) 在URL中指定模板文件。可以使用存储在文件中的模板，在URL中直接指定一个模板文件，避免在URL中写一段很长的SQL查询语句，并且还可以加强系统的安全。

(4) 在简化XML数据计划上使用Xpath查询。在映射计划上编写Xpath查询在概念上就如同使用CREATE VIEW语句创建了视图，然后编写SQL语句来查询他们一样。

(5) 在URL中直接指定数据库对象。数据库对象 (如表和视图) 可以指定为URL的一部分，Xpath也可以在数据库对象上指定。

3.2 读取XML数据

在SQL Server 2000中可以使用执行SQL语句返回XML数据，而不是标准的数据行。这样的查询可以直接执行，也可以通过存储过程执行。要直接获取结果集可以在SELECT语句中使用FOR XML子句，可以为该子句指定三种XML的模式：RAW、AUTO或EXPLICIT。例如：下面的SELECT语句获取了Northwind数据库中Customers表和Orders表中的信息，查询中为FOR XML子句指定了AUTO模式：

```
SELECT Customers.CustomerID,ContactName,
Company Name,
Orders.CustomerID,OrderDate
FROM Customers,Orders
WHERE Customers.CustomerID = Orders.
CustomerID
AND(Cusomers.CustomerID=N' ALFKT'
OR Customers.CustomerID = N' XYZAA')
ORDER BY Customers.CustomerID
FOR XML AUTO
```

3.3 编写对XML文档的查询语句

Transact-SQL关键字OPENXML为内存中的XML文档提供了数据行。OPENXML与表或视图是比较相似的，提供了对XML数据的访问途径。在数据行提供者 (表、视图或OPENROWSET) 可以作为数据源出现的地方，都可以在SELECT或SELECT INTO语句中使用OPENXML关键字。要使用OPENXML编写对文档的查询语句，必须首先调用sq-xml-preparedocument系统过程分解XML文档，并得到对分解后文档的处理。使用系统存储过程可以释放内存中的XML文档。

4 数据挖掘和智能查询应答的结合

我们知道，在数据挖掘过程的处理框架中，处理是由查询启动的，即由查询指定和任务相关的数据、要求挖掘的知识种类、关联限制、有趣的阈值等，并且此过程可以反复进行，虽然目前有如SQL Server 2000提供的MDX等高级形式语句，可以方便地处理数据仓库的查询问题，也有如基于XML的半结构化数据模型，可以进行对Web的数据挖掘，然而，在很多情况下，用户可能并不明确知道要挖掘什么东西或者数据库有什么限制，因此不能给出精确的查询。智能查询应答在这种情况下能帮助分析用户的目的，用智能的方式回答查询请求。数据挖掘和智能查询应答结合是数据挖掘发展的趋势。下面讨论其一种可能的通用框架。

总体来讲，我们可以把数据库系统中的查询归为两种类型：数据查询和知识查询。数据查询用来发现存储在数据库系统中的具体数据，它与数据库系统的一个基本的检索语句对应。知识查询用来发展规则、模式和数据库中的其他知识，它对应于数据库知识的查询，包括演译规则、完整性约束、概化规则、频繁模式以及其他的规则等。查询并没有明显地存储在数据库中的知识，通常要由一个数据挖掘的过程导出。

查询应答机制可以根据它们反应方式的不同分为如下两类：直接查询应答和智能查询应答。直接查询应答是指通过精确地返回所要的东西来回答查询，而智能查询应答包括两个阶段，先分析查询目的，然后返回概化的邻居或与查询相关的关联信息。如考虑顾客对有关某一本书籍的书名、作者、价格和出版社的查询，只需打印出该书这些属性的值。但是，返回有关查询的信息而又不用显式方式提出 (例如书的评价、销售统计、或买此书的顾客多半也要买一些书目)，则是对同一查询的智能应答。如：假设一个网上在线购物中心维护了几个商业数据库，这几个数据库可能包括在线目录库，在线事务历史库和Web日志库。数据查询是执行许多在线服务的例程，例如这样的查询："列出所有在买的自行车"，或者"找出某人在2000年4月购买的所有东西"，对这些查询的直接回答是列出有特定属性的项目列表，而智能回答提供给用户的是用于辅助决策的附加信息。

总之，智能查询应答采用数据挖掘技术来分析用户查询的意图，提供与查询相关的概化和关联信息。这扩展了查询处理系统的能力和可用性，能够在电子商务或其他应用中发挥作用。

5 结束语

面向Web的数据挖掘是一项复杂的技术，XML作为一种半结构化的数据模型，可以很容易地将XML的文档描述与关系数据库中的属性对应起来，实施精确地查询与模型抽取。微软的SQL Server 2000支持完全集成的XML环境，使面向Web的数据挖掘成为可能。同时，数据挖掘技术已经在许多系统中得到应用，但如果用户不能精确地告知要挖掘什么东西或者数据库有什么限制的情况下，计算机则显得力不从心。智能查询应答能在用户不能给出精确查询的情况下用智能的方式帮助用户达到目的。数据挖掘和智能查询应答结合的思想可能是数据挖掘技术的一个发展的趋势。

参考文献

1 罗运模，SQL Server 2000数据仓库应用与开发，人民邮电出版社，2001,7。

2 长城工作室数据组，SQL Server 2000高级应用，人民邮电出版社，2001,8。

3 Inmon,W.H.著，王志海等译，数据仓库，机械工业出版社，2000,5。

4 Lou Agosta著，潇湘工作室译，数据仓库技术指南，人民邮电出版社，2000,11。

5 Jiawei Han,Micheline Kamber著，范明等译，数据挖掘概念与技术，机械工作出版社，2001,8。