文章编号:1001 - 9081(2008)09 - 2318 - 03

基于本体的 XML 关联规则挖掘方法

刘造新
(江西交通职业技术学院 信息工程系,南昌 330013)
(lzx0607@126.com)

摘 要:针对传统的数据挖掘技术不能直接应用到半结构化的 XML 数据挖掘问题,给出了基于本体的 XML 关联规则挖掘方法。该方法引入领域本体和哈希技术来改进产生频繁项目集的操作和生成关联规则的操作,并且使用哈希表存储相关的领域本体,从而将数据库操作转化为对 XML 内存树的操作,通过理论分析和实验验证了方法的挖掘效果,论述了该方法的优点。

关键词:数据挖掘;关联规则;本体;XML
中图分类号: TP311.138 **文献标志码**:A

Association rules mining method from XML based on Ontology

LIU Zao-xin
(*Department of Information Engineering, Jiangxi Vocational and Technical College of Communication, Nanchang Jiangxi* 330013, *China*)

Abstract: Due to the fact that the traditional data mining technology can not be directly applied to the semi-structured XML data mining, Ontology-based XML mining association rules mining method was put forward in this paper. In this method by introducing domain ontology and Hash technology, the operation of frequent item sets and generating association rules has been improved, and at the same time it is easy to transform the operation of the database into memory tree based on XML, and at last through theoretical and experimental analysis it has been verified the mining effects of the method, so the advantages of the method are shown.

Key words: data mining; association rules; Ontology; XML

0 引言

数据挖掘又被称为“数据库中的知识发现”(Knowledge Discovery in Database,KDD)[1],已被越来越多的领域所采用,并取得了较好的效果。目前,数据挖掘已经成为一个国际前沿的研究领域,是数据库研究、开发和应用最活跃的分支之一。

Web 技术自 20 世纪 90 年代出现以来,极大地改变了人们发布、获取和使用信息的方式,尤其是近年来,以 XML[2] 为基础的新一代 Web 环境的出现,很好地兼容了原有的 Web 应用,而且可以更好地实现 Web 中的信息共享与交换。其基于文本的方便性和半结构化特征使得 XML 在信息管理、电子商务、个性化出版、移动通信、网络教育、电子文档交换等诸多领域得到了广泛应用,而且其应用范围还在不断扩展。XML 已经开始成为 Internet 上数据描述和交换的事实标准。对于这些越来越多的采用 XML 文档格式进行存储、交换和表现的数据,除了已有的信息抽取、Web 搜索等信息处理方法之外,人们越来越需要获取更进一步的、深层次的知识,这就需要对其进行数据挖掘。

但是,正由于 XML 是一类半结构化的文本数据,具有文本文档和半结构化数据的诸多弱点,如解析文档时必须采用顺序读取的方式,访问效率不高;对信息的组织不规则,或者其结构可能经常变化,甚至可能不完整等。而传统的数据挖掘技术主要面对的是以结构化数据为主的关系数据库、事务数据库和数据仓库,这样,就不能直接将传统的基于关系数据库的挖掘方法,如 Apriori,应用到半结构化数据挖掘中。因此,开发出有效的针对 XML 的数据挖掘方法成为数据挖掘领域和 XML 技术领域的一项重要课题。

1 本体

本体(Ontology)是知识的形式化表示方法,它为人们观察问题和处理事务时所采用的术语和方法提供清晰一致的表示,提供领域的公共词汇表,以不同的形式化级别定义术语的涵义和术语间的关系。故以分类学方式组织,并包含典型建模原语的本体能够提供对某领域公共而一致的理解,克服通信内容的语义失配问题。

1.1 本体的构造方法

本体的构造分为以下五个阶段[3]。

1)确定本体应用的目的和范围:确立所研究的领域或任务,建立相应的领域本体或过程本体。

2)本体分析:定义本体所有术语的意义及其之间的关系。

3)本体表示:根据系统需要选用一种恰当的本体表示方法。

4)本体检验:主要检验本体的清晰性、一致性、完整性、可扩展性。

5)本体的建立:对所建本体按以上标准进行检验,符合要求的以文件形式存放,否则转 2)。

1.2 本体的表示

为了描述并表示本体,近几年来出现了多种本体描述语言。在目前 XML 已成为数据交换格式描述标准的环境下,这些语言都以 XML 为基础来构建。本文采用本体描述语言中其中的一种——OWL(Web Ontology Language)。OWL 由万维网联盟的 Web Ontology 工作组设计,是 DAML + OIL 的修订本。它的语法与 DAML + OIL 非常相似,因此可以很容易地

收稿日期:2008 - 03 - 13;**修回日期**:2008 - 05 - 26。 **基金项目**:国家自然科学基金资助项目(90412010)。
作者简介:刘造新(1973 -),男,江西安福人,副教授,硕士,主要研究方向:计算机网络、数据挖掘。

被转换为后者。OWL 能够用来清晰地表达词汇表中词条的含义以及这些词条之间的关系。而这种对词条和它们之间的关系的表达就称作 Ontology。OWL 相对 XML、RDF 和 RDFSchema 拥有更多的机制来表达语义,从而超越了 XML、RDF 和 RDFSchema 仅仅能够表达网上机器可读的文档内容的能力。

OWL 对于客观世界的描述主要从概念和属性两个方面进行,与其相应的描述手段是面向对象域的方式和面向数据类型域的方式。面向对象域的描述方式采用 RDFS 和 OWL 自身的语法进行,用于描述概念间分类化、层次化的继承关系以及相互间的关联关系。在进行面向数据类型域的描述时,OWL 支持 XML Schema 的所有数据类型进行概念属性的定义与表达。因此,OWL 通过对概念、概念属性及其相互间关系的描述,构成概念的复杂关系网络[4]。OWL 中的概念由类来表示,它可以是名字(如 URI)或表达式,而且提供大量的构造子来建立表达式,OWL 强大的表达能力正是由它所支持的概念构造子、性质构造子,以及各种公理所决定的。

2　在 XML 关联规则挖掘中应用本体

2.1　Apriori 算法的移植及优化

本文提出的挖掘算法是在 Apriori 算法的基础上改进得到的,Apriori 虽然已经经过了一定的优化,但在实际应用中还是不尽如人意。而且,几乎所有的优化 Apriori 算法都是基于关系数据库、数据仓库挖掘环境的,并不适合基于 XML 的数据挖掘环境,所以对 Apriori 算法进行了移植和优化,使之能对 XML 格式的数据集进行挖掘。

本文提出的移植和优化方案引入领域本体和哈希技术(Hash)来改进产生频繁项目集的操作和生成关联规则的操作,并且使用哈希表存储相关的领域本体,这样,就将经典 Apriori 算法的数据库操作转化为对 XML 内存树的操作。其优点是较好地发挥了 XML 的特长,可以脱离关系数据库操作,而且在查找本体、生成候选项目集和频繁项目集的时候都直接访问内存哈希表,减少了磁盘 I/O 操作,大大提高了算法的运行效率。当然这一方案也有一定的局限性,例如占用内存较大,空间复杂度较高等。方案如图 1 所示。

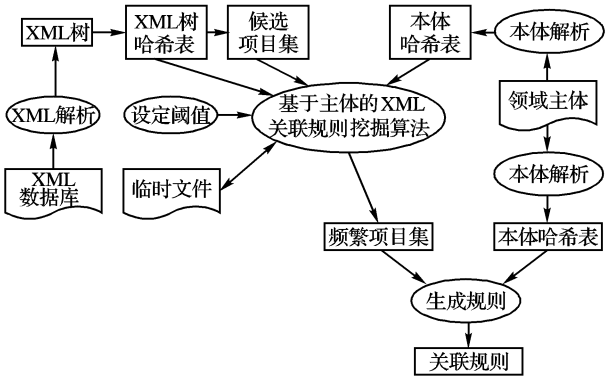

图 1　算法执行过程

在图 1 所示的改进方案中,首先用 Apriori 算法的思想从事务记录中寻找频繁项目集,找出频繁项目集后,继续寻找所有满足最小支持度和最小置信度的强关联规则,并计算其置信度。这里的 Apriori 算法是作了一定的改进的 Apriori 算法,其中引入了领域本体。方案的实现是用 Java 将 XML 数据源解析为 XML 树然后进行计算。这些候选项目集和频繁项目集均存储在本地磁盘或者内存中。当候选项目集过于庞大时,为了节省内存,将候选项目集存储为一个本地临时文件,产生完频繁项目集则删除,不再占据存储空间。内存中的 XML 树 XML_TREE_SUPPORT 用来存放频繁项目集。由于在计算置信度时要用到对应事务的置信度,为了加快查询速度,为 XML_TREE_SUPPORT 建立了索引,对每条事务的存放应用 Hash 技术。

2.2　基于本体的 XML 关联规则挖掘方法

Apriori 是最早的关联规则挖掘的算法之一,它共分为四个步骤(对于给定的数据库 D,最小支持度 *min-sup* 和最小信任度 *min-conf*):

1) 扫描数据库 D,找出大于或等于最小支持度之项目集合,称此项目集合为 L_1;

2) 利用长度为 $k-1$ 的大项目集合(L_{k-1}) 来产生候选项目集合(C_k);

3) 计算所有候选项目集合的支持度,判断是否大于或等于最小支持度,将符合条件之候选项目集合挖掘出来,成为长度为 k 的频繁集(L_k);

4) 重复以上步骤,直到无法产生新的候选项目集合,停止。

形式化描述如下:

C_k :Candidate itemset of size k,K-项候选集
L_k :frequent itemset of size k,K-项频集
L_1 = {frequent items};
for($k = 1; L_k != 0; k++$) {
  C_{k+1} = New candidates generated from L_k ;
  for each transaction $t \in D$ {
    increment the count of all candidates in C_{k+1} that are contained in t
  }
  L_{k+1} = candidates in C_{k+1} with min_support
}
return $U_k L_k$;

本文以这个算法为基础进行改进。

传统 Apriori 算法的挖掘对象是面向事务数据库的,而基于本体的 XML 关联规则挖掘算法的挖掘对象是面向 XML 数据源的,这里的 XML 数据源在数据预处理时存储到哈希表中,以加快遍历整个 XML 树的速度。

同时,算法对传统的 Apriori 算法进行以下三方面的改进:

优化 1:在把 XML 数据源存储到哈希表时,将每条事务中的每一个数据项的所有父类也加入到这条事务中,父类也和普通数据项一样加入到候选项目集中,在遍历 XML 树时,对父类也同样计数;

优化 2:根据预先存入哈希表的领域本体的层次关系,可以得到项集中的每一项父类,删除不出现在任何候选集中的父类;

优化 3:修剪同时包含项 x 及其祖先项的项集。

2.3　算法描述

在图 1 所示的方案中,将该 XML 文档解析为 XML 树,并且将本体解析为本体哈希表之后,执行如下的步骤来寻找频繁项目集。

1)基于 Apriori 的思想,首先生成候选 1-项目集 C_1。算法简单地遍历 XML 树,对每一个项目节点计数。设定最小事务支持度(即 MinSupport),一般由用户来指定。确定频繁 1-项目集的集合 L_1,L_1 中的任何一个业务的频率计数都大于或等于最小支持度。

2) 产生候选 2- 项目集。算法使用 $L_1 \infty L_1$ 产生候选项目集 C_2。C_2 中的每一个项目集是对两个属于 L_1 的频繁项目集作一

个连接来产生的,查询本体哈希表,删除同时包含项 x 及其祖先项的项集,并且删除不出现在任何候选集中的父类。扫描事务数据 XML 树,计算 C_2 中每个候选项目集的支持数,用一个临时 XML 文件存放 C_2。

3)选出事务中每个业务的频率计数不小于最小支持度的事务,从而确定频繁 2-项目集的集合 L_2,并计算其事务对应的哈希函数与项目数 2 组合成 XML_TREE_NODE_ID 加入到频繁项目集 XML 树中。

4)产生 3-项候选集的集合 C_3,利用 Apriori 算法的特点,采用剪枝技术,删除所有其子集不是频繁项目集的候选项目集,从而大大缩减了 3- 项候选集的大小,为生成频繁项目集 L_3 提高效率,查询本体哈希表,删除不出现在任何候选集中的父类。生成频繁 3- 项目集 L_3,计算每条事务的支持度。

5) 如此循环下去,不断生成候选集,再由此生成频繁项目集,直到候选项目集为空。

3 性能分析与实验

3.1 性能分析

文献[5]采用 XQuery 来实现传统的挖掘算法,在挖掘时需要反复读入 XML 数据库,因此效率较低,虽然为了减少读取次数,加入了一些更新操作,但由于 XML 是半结构化数据,更新起来非常困难。此外,使用 XQuery 也有一定的限制性,对于一些结构更为复杂、不规则的文档来说,用该方法进行挖掘就较难实现。文献[6]要求 XML 文档是经过校验的,因此,在数据源的选择上有一定的限制,不具有普遍应用意义,而且挖掘出的规则存在大量冗余。同时,由于其使用的是 DOM 接口,在挖掘大数据量的 XML 文档时效率将会很低。文献[7]通过构建 Pruning Tree 来提取频繁子结构,然后通过频繁子结构来得出关联规则。在 Pruning Tree 中,低于指定阈值的子树将被剪除,这在很大程度上减小了最终规则的冗余,但是,由于 XML 文档标记的开放性的特点,使用 Pruning Tree 可能无法判断哪些标记路径是频繁的,并不适用于挖掘表示事务数据库的 XML 文档的情况。根据上面的分析可以看出,传统的挖掘算法存在两方面的问题:

1)挖掘出太多的规则,这些规则中很多是无用的;

2)挖掘的规则往往过于具体,不能给出整体性的把握。

为了解决以上问题,应该在关联规则挖掘的过程中引入专家的领域知识和背景知识。本体是知识表示的一种形式,可以很好地将领域知识和数据挖掘算法结合起来,从而优化现有算法。基于本体的关联规则挖掘的优点还在于,它可以在多层次上进行数据挖掘,产生多层次的规则,所以基于本体的关联规则挖掘是多层关联规则挖掘的有效工具。

另外在实际应用中那些跨越层次的关联规则也是有意义的。例如在作商品推荐的时候,通常推荐的是一类商品,而不是很多种商品。

3.2 实验

为了验证文中提出的挖掘方法的有效性,用 Java 实现了该算法,并进行了算法实验。XML 文档使用微软的解析器 MSXML 来快速解析,以便生成文档树。本次实验使用的领域本体是由英国谢菲尔德大学的自然语言处理小组发布的供实验研究使用的 Animals 本体[8],它是描述在英国儿童读物中出现的动物种类的领域本体,它的目的是作为初级实验的标准供研究时使用。由于目前尚没有专门用于挖掘的 XML 文档,实验中采用的测试数据来源于作者模拟的图书数据,数据集包含 2000 多条事务,改造后的 XML 文档大小约为 5.7 MB。测试环境为:P4 3.0 CPU,512 MB 内存,Windows XP 系统。

由于提出的方法中采用了哈希表来存放 XML 文档树,因此可以在较短时间内遍历整个数据集,从而无需重复扫描 XML 文档,为挖掘过程节省了大量的开销。实验证明,该方法通过引入领域本体对 XML 文档中频繁出现的数据进行概化,能够有效地压缩 XML 文档的大小,挖掘出的关联规则更容易理解。挖掘的结果中频繁集数目随着最小支持度阈值增加的变化过程如图 2 所示。

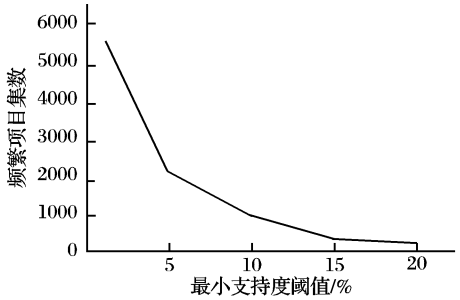

图 2 挖掘频繁集数目随阈值增加的变化

算法的缺点是占用内存较大,空间复杂度较高,另外由于引入了每一项父类,算法在遍历 XML 树时需要计数的项也相应增多,会对性能产生一定的负面影响。

4 结语

以往对 XML 文档的挖掘算法都是针对具体的数据项进行挖掘,本文将域本体引入到挖掘过程中,允许用户在更高的层次上进行挖掘,产生多层的挖掘结果,帮助用户更好地进行决策。同时阐述了本体的构建方法和本体的表示,介绍了基于本体的 XML 关联规则挖掘算法,在理论上分析了几种现有 XML 关联规则挖掘方法的不足,论述了本文算法的优点,并且对算法进行了实验,验证了算法的有效性。

参考文献:

[1] HAN JIA-WEI, KAMBER M. 数据挖掘: 概念与技术[M]. 范明, 译. 北京: 机械工业出版社, 2001.

[2] World Wide Web Consortium. Extensible markup language (XML) version 1.0 W3C recommendation[S/OL]. [2008 - 02 - 01]. http://www.w3.org/XML.

[3] 杨秋芬, 陈跃新. Ontology 方法学综述[J]. 计算机应用研究, 2002, 19(4): 5 - 7.

[4] HORROCKS I, PATEL - SCHNEIDER P F, van HARMELEN F. From SHIQ and RDF to OWL: The making of a Web Ontology language[EB/OL]. [2008 - 01 - 01]. http://www.w3.org/, 2004. 7: 348 - 356.

[5] WAN J W W, DOBBLE G. Mining association rules from XML data using XQuery[C]// Proceedings of the second workshop on Australasiam information security, Data Mining and Web Intelligence, and Software Internationlisation. New Zealand: Australian Computer Society, 2004: 169 - 174.

[6] DING QIN, REORDS K, LUMPKIN J. Deriving general association rules from XML data [C]// Proceedings of the ACIS Fourth International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel /Distributed Computing(SNPD'03). Germany: [s. n.], 2003: 348 - 352.

[7] MEISELS A, ORLOW M, MAOR T. Discovering associations in XML data[C]// Proceedings of the Third International Conference on Web Information System Engineering(Workshops). Washington, DC: IEEE Computer Society, 2002: 178 - 183.

[8] Automating Ontology learning for the semantic Web[EB/OL]. [2008 - 02 - 01]. http://nlp.shef.ac.uk/abraxas/resources.html.