基于数据抽取器的知识发现处理模型

郑宏珍[1] 刘 扬[2] 战德臣[3]
(哈尔滨工业大学计算机科学与技术学院 威海 264209)[1]
(哈尔滨工业大学计算机科学与技术学院 哈尔滨 150001)[2]

摘 要 提出了基于数据抽取器的知识发现模型。在模型中,将知识发现过程分成数据预处理、数据抽取、数据挖掘和结果分析四个阶段。该模型利用标准的SQL语言构造数据抽取器,为不同的学习算法准备数据,减少数据挖掘算法对数据库直接调用的次数,避免了直接对大型数据库的数据进行调用,使得对大型数据库进行快速数据挖掘成为可能。可以加快知识发现过程,提高数据挖掘效率,实现对于大型数据库的知识发现。最后设计了SQL-C4.5算法,该算法实现了利用数据抽取器为决策树算法C4.5抽取必要的统计数据,实现了C4.5决策树的构建。

关键词 KDD,分类,决策树

Knowledge Discovery Processing Model Based on Data Extractor

ZHENG Hong-Zhen[1] LIU Yang ZHAN De-Chen
(Department of Computer Science & Technology, Harbin Institute of Technology, Weihai 264209)[1]
(Department of Computer Science & Technology, Harbin Institute of Technology, Harbin 150001)[2]

Abstract In this paper, a knowledge discovery model based on data extractor is proposed. According to this model, the process of knowledge discovery is divided into four stages: data preprocessing, data extraction, data mining and result analysis. This model, standard SQL is used to construct data extractor to prepare data for different learning algorithms, to reduce the number of times to invoke the database for the data mining algorithm, to avoid direct access to the data in large database and to make it possible to do rapid data mining to large database. By using this model, data discovery process has been sped up, efficiency of data mining has been promoted, and knowledge discovery for large database has been realized. Finally, we design the SQL-C4.5 algorithm, which realizes extracting necessary statistical data for decision-tree algorithm C4.5 and realizes construction of decision tree of algorithm C4.5.

Keywords Knowledge discovery, Data mining, Decision trees, Classification

1 引言

知识发现过程中非常重要的一个步骤是为数据挖掘算法准备合适的数据,也就是要为数据挖掘算法提供挖掘算法能够理解的数据[1]。许多大的公司和组织都将积累的数据以关系数据库的形式存储在数据库中,有几千万条纪录。因此,对大型的可靠的商业数据库进行有效的知识发现和数据挖掘至少需要解决两个问题[2]:(a)现存的知识发现算法和流行的数据库管理系统(如Oracle)的集成问题;(b)计算速度提高的能力,如数据处理的平行算法。挖掘算法通常是以能直接对数据文件进行操作的程序的形式来表现的,大多数流行的数据挖掘算法是连续化的,并假定数据库是驻留在内存中。而事实上,现在我们面临的是非常大的数据库,尽管CPU的速度和内存已经有了很大的提高,可待挖掘的数据却变得越来越多,因此,将这巨大的数据库调入内存让数据挖掘算法对其直接进行操作是非常不现实的。

另外,让数据挖掘算法访问整个数据库的内容不仅是非常麻烦的,也是不必要的。许多的数据挖掘算法是基于对特定数据库的相对简单的统计数据值上进行分析的,在这种情况下,包含在数据库中的全部信息就会出现过剩。我们设计这样一个框架体系,在该框架中的数据挖掘算法可以通过数据挖掘算子从DBMS中简单直接地得到所需要的统计值。数据挖掘算法可以借助一个数据交互接口指导DBMS处理大量的原数据来得到所需的统计值,而不须直接对数据库进行大量操作。这样可以大大减少数据挖掘的时间。

2 基于数据抽取器的知识发现处理模型

在挖掘中,数据准备、预处理是整个挖掘过程中最消耗时间的。为了节约挖掘时间,我们将数据准备和数据预处理阶段合并,以避免重复的数据处理工作。对于大的数据集,让数据挖掘算法访问整个数据库的内容不仅是非常麻烦的,也是不必要的。因此,我们提出数据抽取器的概念,数据挖掘算法可以通过数据抽取器从DBMS中简单地得到所需要的统计值,而不须直接对数据库进行操作。这样可以大大减少数据挖掘的时间。数据抽取器可以针对不同的算法抽取特定的数据,避免不同算法对数据集的重复操作,提高挖掘效率。

为了适用一般算法的学习目标,我们为数据抽取器定义了统一的接口。任何一个使用该接口的学习算法都可以用不同的数据抽取器进行数据抽取,不同的学习算法也可用同一数据抽取器进行知识提取,从而实现多目标学习。最终用户

郑宏珍 博士生,副教授,硕士导师,主要研究方向:数据挖掘、数据库技术、神经网络和人工智能;**战德臣** 博士,教授,博士生导师,主要研究方向:数据库技术、CIMS、人工智能、算法研究等。

可以通过定义数据抽取器的方法将学习任务所涉及到的可能的数据范围设定，这样数据集中就包含了最终用户的一些背景知识和经验。这样一方面最大程度地利用最终用户的经验知识；另一方面，由于给出了数据范围，学习算法处理的数据量减少，有利于提高学习速度。基于上述考虑，本文提出了如图 1 所示的基于数据抽取器的 KDD 模型。

该处理模型的特点是：

(1) 将数据挖掘过程压缩到只有 4 个阶段，将注意力放在数据挖掘的关键问题上。

(2) 通过标准的数据挖掘抽取器，抽取数据，节约了挖掘时间。

(3) 针对不同的算法可以抽取特定的数据，避免算法对数据集的重复操作，提高挖掘效率。

该模型将知识发现过程分为 4 个阶段：

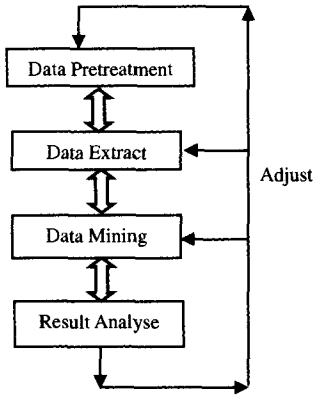

图 1　基于数据抽取器的知识发现过程模型

基于数据抽取器的知识发现处理模型充分体现了数据挖掘需要多次反复这一特点，符合循环渐进、螺旋上升的方法论。实际应用表明，这是一种有效的方法。

基于数据抽取器可以重复利用数据抽取结果，既可以利用同一算法的数据抽取结果，也可以利用不同算法之间的数据抽取结果。既然数据库有不变的数据结构，我们就可以利用重复的查询多次进行试验。因此，该数据抽取器应能够具有一定的通用性，以适应不同的数据挖掘算法对数据库管理系统的要求。

3　数据抽取器的设计

数据抽取器，即数据挖掘算法和 DBMS 的接口，数据挖掘算法通过其从 DBMS 中对数据库进行操作获得所需的合适的数据。这里给出数据抽取器的一般定义：

Extractor1 GROUP1(A,δ,DB) AS
SELECT A, f(*)
FROM DB
WHERE δ
GROUP BY A

这里，用 GROUP(A,δ,DB) 来表示一个 count by group 查询。A 为属性列表。δ 为 WHERE 中的条件子句，DB 表示一个数据库。用 f(*) 表示统计函数，包括 SUM—数字列的总计；AVG—数字列的平均；COUNT—数字列数；COUNT(*)—SELECT 查询出的行数；MAX—表达式的最大值，MIN—表达式的最小值等操作。SELECT 子句指定取出的列；FROM 子句指定从哪些数据库中取出；WHERE 子句则指定要从表中取出那些行；GROUP BY 子句将一个表分成若干组，可按列名分组，或按使用数值类型的计算列的结果分组。

Extractor2 GROUP2(A,a,δ,DB) AS
SELECT A, f(*)
FROM DB
WHERE δ
GROUP BY A
GROUP BY a

其中，GROUP2 (A, a, δ, DB) 表示一个 ordered-count-by-group 的查询，a 是 A 中的属性，a 为排序属性。通过定义这些基本的数据挖掘算法抽取器，可以提高数据挖掘算法数据挖掘的效率。

4　SQL-C4.5 决策树数据挖掘算法

根据 C4.5 决策树的构造原理，构造一棵 C4.5 决策树的基本过程定义为：

```
Input: A data set D
Output: A decision tree: T
Function GrowTree(D)
{
Initialize T and put all attribute in the root
If stopping criteria(D)=TRUE
  return a leaf;
Else
    For each attribute A
    chooseBestTest(A)
For each partition of examples based on chosen attribute values
  generate a subtree Ti=GrowTree (Di)
    Return T;
    / * containing a decision node based on chosen attributeand de-
       scendants Tree * /
    postprune()
}
```

函数 stopping ( ), chooseBestTest( ) 和 postprune( ) 它们是对数据库进行操作的函数。每个函数进行一种启发式搜索。最好的属性测试集最精确地反映了数据的结构，也利用决策树作为数据的挖掘算法，首先要考虑的是树的分枝准则，也就是选择那个分枝以得到最小且最有效率的决策树。其次要考虑分枝的停止准则，就是要判断什么时候停止分枝。几乎所有的树的分枝算法都是涉及到类变量和属性发生的概率，也就是说是变量计数值的函数。

因此，利用定义的 SQL 抽取器，完全能得出决策树算法所需要的概率。其中最著名且被广泛应用的一种决策树演绎方法是 C4.5[4]。Qulin 的 C4.5 决策树算法采用了来自于信息理论的启发式方法来选择属性测试集，主要依赖于后剪枝算法，在大型数据库的数据挖掘中使用效果很好。下面给出利用数据抽取器进行数据挖掘的 SQL- C4. 5 决策树算法。

```
ALgorithm SQL- C4.5
Input: A data set D
Output: A decision tree: T
Function GrowTree(D)
{
  Initialize T and put all attribute in the root For every node
  GROUP1(C,δ)
  If stopping criteria(D)=TRUE
    return a leaf;
  Else
    For each attribute X
    GROUP1({X,C},δ,DB)
    }
  Function Grx(count( * ))
{
  ChooseBestTest=max(GRX)
  For each partition of examples based on chosen attribute values
  generate a subtree Ti=GrowTree (Di)
  Return Tree
  / * containing a decision node based on chosen attributeand descend-
     ants Tree * /
  For each node of subtree
  GROUP1(C,δ)
```

```
postprune ()
}
```

结论 针对目前的知识发现过程模型在实际应用中存在挖掘周期长、对大型数据库的知识发现支持不够的问题，提出了基于数据抽取器的知识发现模型。该模型利用标准的SQL语言构造数据抽取器，为不同的学习算法准备数据，数据挖掘算法可以通过数据抽取器从DBMS中简单地得到所需要的统计值，数据抽取器可以针对不同的算法抽取特定的数据，减少数据挖掘算法对数据库直接调用的次数，避免了直接对大型数据库的数据进行调用，使得对大型数据库进行快速数据挖掘成为可能。可以加快知识发现过程，提高数据挖掘效率，实现对于大型数据库的知识发现。

为数据抽取器定义了统一的接口。任何一个使用该接口的学习算法都可以用不同的数据抽取器进行数据抽取，不同的学习算法也可用同一数据抽取器进行知识提取，从而实现多目标学习。最终用户可以通过定义数据抽取器的方法将学习任务所涉及到的可能的数据范围设定，这样数据集中就包含了最终用户的一些背景知识和经验。这样一方面最大程度地利用最终用户的经验知识；另一方面，由于给出了数据范围，学习算法处理的数据量减少，有利于提高学习速度。

最后设计了SQL- C4.5算法，该算法实现了利用数据抽取器为决策树算法C4. 5抽取必要的统计数据，实现了C4.5决策树的构建。

参考文献

1 Quinlan J R. Expert systems Discovering rules by induction from large collections ofin the Microelectronic Age. Edinburgh University Press,1979
2 Berson A,Smith S J. Data warehousing, Data Mining,&OLAP. McGraw-Hill Co.,1997. 113～150,333～514
3 Scotney B,McClean S. Efficient knowledge discovery through the integration ofheterogeneous data. Information and Software Technology,1999,41:569～578
4 Quinlan J R. C4. 5: Programs for Machine Learning. Morgan Kaufmann, 1993
5 Pei J,Han J,Mortazavi-Asl B,Zhu H. Mining Access Pattern efficiently fromWeb logs. In: Proc. 2000 Pacific Asia Conf. on Knowledge Discovery and Data Mining (PAKDD'00), Kyoto, Japan, April 2000
6 William J. Long, Medical informatics: reasoning methods. Artificial Intelli- gencein Medicine,2001,23:71～87
7 Ziarko W,Yao Y Y,Rough Sets and Current Trends in Computing (RSCTC 2000). Berlin: Springer-verlag, 2001
8 Wang Jue,Wang Ju. Reduction algorithms based on discernbility matrix: the ordered attributes method. Journal of Computer Science & Technology, 2001, 16(6):489～504
9 Wang Jue,Wang Ju. Reduction algorithms based on discernbility matrix: the ordered attributes method. Journal of Computer Science & Technology, 2001, 16(6):489～504
10 Lin T. Y,Yao Y. Y,Zadeh L A. Data Mining, Rough Sets and Granular Computing. New York: Physica-Verlag, 2002
11 Ziarko W,Yao Y Y. Rough Sets and Current Trends in Computing (RSCTC 2000). Berlin : Springer-verlag, 2001
12 Skowron A. Rough sets and Boolean reasoning. In: W. Pedryczed, ed. Granular Computing: An Emerging Paradigm. New York: Physical Verlag,2001. 95～124
13 Polkowski L,Tsumoto S,Lin T Y. Rough Set Methods and Applications: New Developments in Knowledge Discovery in Information Systems. New York: Physica -Verlag, 2000

(上接第109页)

而获取该记录所属SEE的权威度(authorityWeight)。

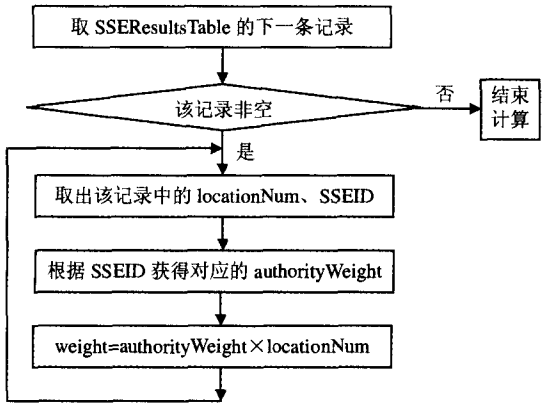

图3 计算最终排序权值的流程图

4 系统实例

随着网上教学信息资源的膨胀发展，一般资源搜索引擎在教学资源方面的查全率和查准率很难满足广大教学资源检索者的要求。要想获得一个比较全面、准确的结果，教学资源检索者就必须反复调用多个搜索引擎，然后在各个搜索引擎的结果中综合出最适合自己的内容。为了减轻教学资源检索者学习检索技巧、选择资源搜索引擎以及综合多个搜索结果的负担，我们以计算机学科领域资源为例，建立了一个面向计算机教学资源的元搜索引擎原型系统。

该试验在Windows 2000 Professional、Internet Information Server 5.0服务器、Access数据库、VB. NET软件开发环境下试验成功，试验结果如图4所示。本原型系统能够较好地满足用户进行计算机学科领域资源检索的要求。

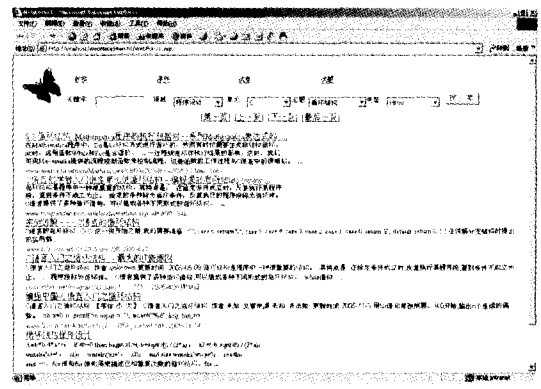

图4 搜索结果示例

结束语 由于元搜索引擎扩大了搜索的覆盖面，解决了搜索的可扩展性，方便获取来自多个搜索引擎的信息，而且提高了信息检索的查全率和查准率。把元搜索引擎技术引入到领域资源中，进而建立领域资源本体库[7]，并结合现代语义网技术，进一步提高本搜索引擎的智能化水平，对于推动领域资源的发展和共享必将起到十分积极的作用。

参考文献

1 Selberg E,Etzioni O. Multi-Engine Search And Comparison Using The MetaCrawler. In:Proceedings of the Fouth World Wide Web Conference '95, Boston USA, Dec. 1995
2 李广建,等. 元搜索引擎及其主要技术. 情报科学 2002(2)
3 张卫丰,徐宝文. Web搜索引擎框架研究. 计算机研究与发展,2000,37(3):376～378
4 张弓强. 一种元搜索引擎的查询结果处理模型. 北京工业大学
5 董荣胜,古天龙. 计算机科学与技术方法论. 北京:人民邮电出版社
6 Heaton J. 网络机器人Java编程指南. 童兆丰,李纯,刘润杰,译. 北京:电子工业出版社
7 蔡铭. 面向网络化制造基于语义的资源获取、智能检索和服务匹配原理与技术研究