文章编号:1002-0411(2005)02-0249-04

知识发现中可继承性问题的研究

冯兴杰[1,2], 黄亚楼[2]

(1. 中国民用航空学院计算机科学与技术学院, 天津 300300; 2. 南开大学软件科学学院, 天津 300071)

摘 要:提出知识发现中的可继承性问题,通过对知识发现过程和挖掘算法形式化描述和分析,抽象出各个阶段的形式联系及其约束条件,在此基础上提出初等知识的概念.在引入初等知识后,对传统的挖掘算法、增量式挖掘算法、可继承性挖掘算法进行形式化描述和比较,得出如下结论:可继承性挖掘算法能够有效的提高数据集变化、参数变化情况下的数据挖掘效率.*

关键词:知识发现; 初等知识; 增量式挖掘算法; 可继承性挖掘算法

中图分类号: TP301 文献标识码: A

On the Inheritability Problem in Knowledge Discovery in Database

FENG Xing-jie[1,2], HUANG Ya-lou[2]

(1. *School of Computer Science and Technology, Civil Aviation University of China, Tianjin 300300, China;*
2. *School of Software, Nankai University, Tianjin 300071, China*)

Abstract: The inheritability problem in the KDD(knowledge discovery in database) process is presented. The KDD process, the DM(data mining) algorithms, and the relation between them are formally described and deeply analyzed. The basic concept named PK(primary knowledge) is put forward. Then the traditional DM algorithm, incremental DM algorithm and the inheritable DM algorithm are formally described and compared. Finally, it is concluded that the inheritable DM algorithm can improve the efficiency of data mining with the variable data set and parameters.

Keywords: KDD; primary knowledge; incremental DM algorithm; inheritable DM algorithm

1 引言 (Introduction)

迭代和交互是知识发现过程中固有的特性[1],当面对“海量”数据时,现有的数据挖掘算法的有效性和可伸缩性是当前该领域面临的一个“难题”,也是数据挖掘领域的一个长期研究热点.另一方面,我们必须面对一个事实:虽然计算资源正遵循着著名的 Moore规律,以每 18个月翻一番的速度增长,但是待分析的数据集也正在以同样的或更高的速度增长.因此,数据挖掘所面对的是一个动态变化的环境,主要表现在:1)数据挖掘过程中固有迭代和交互;2)待分析的数据集高速变化.

一般的挖掘算法对迭代和交互的处理仅仅是“在新的参数约束下重新执行一次”,与上一次的数据挖掘过程和结果没有任何联系,这势必造成大量计算资源的浪费.特别的,当每次数据挖掘过程要运行较长时间时,漫长的响应时间是不可容忍的.“基于约束”的挖掘算法在解决交互性方面会有所帮助,通过使用约束条件可以有效地缩减搜索空间、优化挖掘计算[2],但是对于多次迭代却无能为力.另一方面,数据集的不断变化也是一个事实,如何在变化了的数据集上开展数据挖掘过程也是研究者们不可回避的问题.“增量式”挖掘算法是解决可变数据集挖掘的一条途径[3,4],但是参数变化情况下却无能为力.

可继承性数据挖掘方法正是在这种背景下提出的.它主要研究在数据集变化、参数变化情况下,如何在下一次挖掘过程中有效地利用上次挖掘结果,以提高挖掘效率.

2 知识发现中的可继承性问题分析 (The analysis of the inheritability problem in KDD)

数据库中的知识发现是一个迭代的、交互的过

* 收稿日期:2004-04-29
基金项目:教育部科学技术研究重点资助项目(02038)

程. Fayyad等在文献[1]中提出的用于数据库中的知识发现的 KDP(knowledge discovery process,知识发现过程)模型,由多个步骤组成,并且任何两个步骤都有可能被重复执行.它主要包括:挖掘需求制订、相关数据选择、数据清洗、数据变换、数据挖掘、知识表示和知识的解释或应用等阶段.数据库中的知识发现的问题可以形式化地描述为如下六元组:

$$(T, D, C, P, M, K)$$

其中:

T:表示知识发现的目标和任务;

D:表示与 T相关的数据;

C:表示一组有助于发现特定知识的基本概念或背景知识;

P:表示数据预处理操作的集合,又可以进一步表示为如下形式:

$$P::= P_{\text{select}}[*\ P_{\text{clean}}][*\ P_{\text{trans}}]$$

其中:[]表示可选操作.

M:表示数据挖掘算法.

K:表示发现的知识.

下面分别就 P_{select}、P_{clean}、P_{trans}、M讨论其约束特性:

1) P_{select}:表示相关数据选择算子,根据知识发现的目标 T选择相关的数据.数据库中虽然存储了大量的数据,但这些数据不一定全和挖掘目标相关.通常情况下,对于一个特定的挖掘目标只需分析数据库的一个子集,所以用户就需要有选择地抽取相关数据,然后在其上实施挖掘算法.因此 P_{select}受知识发现的目标 T的约束,记为:$P_{\text{select}}|_T: D_{\text{origin}} \rightarrow D_{\text{select}}$.

2) P_{clean}:表示数据清洗算子,其目的是使数据相对于知识发现目标 T保持语义完整性.因为大多数挖掘算法没有考虑脏数据、丢失数据的问题,所以该步骤显得更为重要.数据在存储到数据库之前,都需要经过数据库模式中的各类约束条件的相容性验证,但是因为允许存储空值,使数据丢失成为一种普遍存在的现象.因此,P_{clean}受知识发现的目标 T和特定约束条件 C的约束,记为:$P_{\text{clean}}|_{T\wedge C}: D_{\text{select}} \rightarrow D_{\text{clean}}$.

3) P_{trans}:表示数据变换算子.由于一些算法对数据的表现形式有特殊的要求,所以在挖掘之前需要对数据进行适当的变换.因此 P_{trans}要受数据挖掘算法 M的约束,记为:$P_{\text{trans}}|_M: D_{\text{clean}} \rightarrow D_{\text{trans}}$.

4) M:表示数据挖掘算法,在经过选择、清洗、变换的数据集上执行挖掘算法.算法的选择依赖于要发现的知识 K,记为:$M|_{K\wedge C}: D_{\text{p}} \rightarrow K$,其中 D_{p}表示经过预处理后的待挖掘数据集.

通过分析知识发现过程中各个阶段可以发现有如下关系:数据选择 $P_{\text{select}}|_T$受限于挖掘目标 T,而数据变换 $P_{\text{trans}}|_M$则受限于挖掘算法.从这个层次上而言,数据选择应通过对原始数据集的增量更新方法实现(该问题属于预处理阶段,在本文中不做更深入的研究).数据变换受限于挖掘算法,通过进一步分析算法,可以发现挖掘算法 M所涉及的计算又分为两个阶段:参数无关阶段 $M|_{\{\emptyset\}}$和参数相关阶段 $M|_{\{\delta\}}$,即:

$$M::= M|_{\{\emptyset\}} \cdot M|_{\{\delta\}}$$

$M|_{\{\emptyset\}}: D_{\text{p}} \rightarrow D_{\text{PK}}$,其中 D_{PK}表示数据到知识过渡的中间表示形式,称为:初等知识(primary knowledge, PK).

$$M|_{\{\delta\}}: D_{\text{PK}} \rightarrow K$$

因此可继承性应该集中在与参数无关阶段 $P_{\text{trans}}|_M$和 $M|_{\{\emptyset\}}$中研究.一个更为通用的方法是:使参数无关阶段 $P_{\text{trans}}|_M$和 $M|_{\{\emptyset\}}$具有"同类算法独立性",也就是说通过引入适当的数据结构,可以长期地保存 $P_{\text{trans}}|_M$和 $M|_{\{\emptyset\}}$输出的初等知识(PK) D_{PK}.参数相关阶段 $M|_{\{\delta\}}$则在初等知识 D_{PK}上工作.在此基础上,能够比较自然地将"增量式"、"基于约束"的思想引入到整个算法设计的框架内.

3 三种类型挖掘算法的比较 (The comparison of three kinds of data mining algorithms)

从直观意义上讲,直接从原始数据到目标知识($D_{\text{origin}} \rightarrow K$)的变化显得过于突兀,在两者之间引入初等知识,有利于使知识发现过程变得平缓($D_{\text{origin}} \rightarrow D_{\text{PK}} \rightarrow K$).从数据组织的角度看,引入初等知识不仅能为从原始数据到目标知识的过渡提供一层缓冲,而且为有效地保存中间结果,改善每个阶段算法的起点提供了机会,即:不必每次都从原始数据的扫描开始.这是一种典型的"以空间换时间"的做法.初等知识是和知识类型有关的一个概念,很难给出统一的定义,但是我们可以从它在知识发现中所处的位置、它的内容及结构进行描述.

定义 1 初等知识是从数据抽象到知识过程中的一种中间表示;它的内容通常包括对数据集的统计分析、基本聚集以及在挖掘阶段非常有用的其它操作的结果;通过设计合适的数据结构,初等知识可以在计算机中长期存储.

3.1 三种类型挖掘算法的比较

在知识发现和数据挖掘领域,研究者们提出了很多挖掘算法.根据对可变数据集的处理策略划分,可以分为两类:1)非增量式挖掘算法(本文中称为传统挖掘算法),在变化后的数据集上重新运行挖掘算法;2)增量式挖掘算法,只对增量数据部分进行挖掘,并将挖掘结果同之前的结果合并.但是当算法参数变化时,增量式挖掘算法必须在变化后的数据集上重新运行.

我们提出一类新的挖掘算法,称为:可继承性挖掘算法.在数据集变化和参数变化情况下,该类算法能够利用初等知识进行数据挖掘,而不用重新扫描原始数据集,从而能够有效提高挖掘效率.下面对传统挖掘算法、增量式挖掘算法、可继承性挖掘算法进行形式化描述和比较.

记:D_i表示 t_i时刻的数据集,$i=1,2,3,\cdots$,在初始阶段数据集为 D_0,D_0从 t_0 到 t_k时刻的变化可以表示为:$D_0 \xrightarrow{\delta D_1} D_1 \xrightarrow{\delta D_2} \cdots \xrightarrow{\delta D_k} D_k$,其中 $\delta D_i = D_i - D_{i-1}$表示数据库在第 i时刻的增量部分 $(1 \leqslant i \leqslant k)$.

记:$F_p^T = P_{\text{select}}[\,*\,P_{\text{clean}}][\,*\,P_{\text{trans}}]$表示数据预处理工作,$F_m^T$表示挖掘操作.

如果用户想在 t_b和 t_c $(1 \leqslant b \leqslant c \leqslant k)$时刻进行挖掘,则对传统挖掘算法需要进行如下操作:

$$F_p^T(D_b) \rightarrow D_{p_b}^T \mapsto F_m^T(D_{p_b}^T) \rightarrow K_b \tag{1}$$

对 D_c需要类似的操作,即:

$$F_p^T(D_c) \rightarrow D_{p_c}^T \mapsto F_m^T(D_{p_c}^T) \rightarrow K_c \tag{2}$$

对增量挖掘算法而言,因为它利用了(1)所做的工作 $(|\delta D_i| < |D_i|)$,使(2)式更为有效,即:

$$F_p^I(\delta D_c) \rightarrow \delta d_p^I \mapsto F_m^I(\delta d_p^I, K_b) \rightarrow K_c \tag{3}$$

其中:F_m^I表示增量挖掘算子.

在引入初等知识后,对数据集的每个增加部分 δD_i,定义如下初等知识抽取功能:

$$F_{\text{PK}}(\delta D_i) \xrightarrow{t_i} (R_i, A_i, \delta d_i) \quad i = 1,2,\cdots,k \tag{4}$$

其中:F_{PK}表示对数据集执行的统计分析、基本聚集以及在挖掘阶段非常有用的其它操作的集合.一般而言,应满足如下要求:

$$M|_{\{\varnothing\}} \subseteq F_{\text{PK}} \tag{5}$$

(5)式表示:数据挖掘算法中和参数无关的操作应该提取出来,归并到功能 F_{PK}中.

F_{PK}的输出是一个三元组 $(R_i, A_i, \delta d_i)$,其含义如下:

- R_i表示初等知识抽取注册表项,在 δD_i和 $(A_i, \delta d_i)$之间起到"接口"的作用,它至少应该包含如下信息:

(Source_Name, Window_No, Ini_Record_Num, Cur_Record_Num, PK_Name, …),其中 Source_Name表示源数据集名称,Window_No表示源数据集窗口编号,Ini_Record_Num表示数据窗口编号为 Window_No的初始记录数目,Cur_Record_Num表示数据窗口编号为 Window_No的当前记录数目,PK_Name表示初等知识名称;

- A_i表示 δD_i上的聚集结果;
- δd_i代表 δD_i的浓缩表示.

称三元组 $(R_i, A_i, \delta d_i)$为初等知识,记为:PK_i,这样经过自动的、定期的执行,当数据库增长到 D_k时,便获得了一系列初等知识,记为:

$(R_1, A_1, \delta D_1)$,$(R_2, A_2, \delta D_2)$,…,$(R_k, A_k, \delta D_k)$或 $(PK_1, PK_2, \cdots, PK_k)$.

引入初等知识后,因为充分利用了(4)所做的工作,将使参数变化情况下的数据挖掘更为有效,即:

$$\begin{aligned} M|_{\{\varnothing\}} \cdot M|_{\{\delta\}}(D_c) &= M|_{\{\delta\}}(M|_{\{\varnothing\}}(D_c)) \\ &= M|_{\{\delta\}}(PK_1, PK_2, \cdots, PK_k) \end{aligned} \tag{6}$$

其中:$M|_{\{\varnothing\}}$:表示挖掘算子的参数无关阶段,$M|_{\{\delta\}}$:表示挖掘算子的参数相关阶段.从(6)式可以看出,初等知识 $(PK_1, PK_2, \cdots, PK_k)$只需计算一次,使得在数据集变化或者参数改变时不需要重新扫描原始数据库,因而能够提高挖掘效率.

综合(4)和(6)式,可以看出因为利用了初等知识,可继承性挖掘算法能够有效地改善数据集变化、参数变化情况下的数据挖掘算法效率.

3.2　可继承性聚类挖掘算法示例

下面以聚类挖掘为例,应用如上描述的基本思路构建可继承性聚类挖掘算法.整个设计过程分为三步:1)设计初等知识结构;2)设计参数无关挖掘算子 $M|_{\{\varnothing\}}$;3)设计参数相关挖掘算子 $M|_{\{\delta\}}$.

(1)设计初等知识结构

采用 BIRCH算法中使用的 CF(cluster feature,聚类特征)树作为初等知识的结构,记为:$PK[(R, A, \delta D)]$.第一部分 R表示初等知识抽取注册表;第二部分 A用于保存聚集操作的结果,通过 CF树的非叶结点与之对应;第三部分 δD用于保存浓缩的数据库,通过 CF树的叶结点与之对应.

(2)设计参数无关挖掘算子 $M|_{\{\varnothing\}}$

参数无关挖掘算子 $M|_{\{\varnothing\}}$的主要功能是按照初等知识的结构,从原始数据中生成初等知识.在此以 CF树作为初等知识结构,建立 CF树的过程是一个动态构建的过程,类似于 B+树,下面给出算法伪代

码:

功能:在 CFTree中插入节点.

参数: $CFTree$—指向 CFTree根结点的指针;

d—数据对象;

B—分支数目;

L—叶结点数据对象数目;

T—距离阈值.

1) 根据选定的距离度量从根结点开始递归地寻找距离最近的叶结点,记为: L_i;

2) 如果满足 $Dist(d, L_i) \leqslant T \wedge Count(L_i) < L$,则数据对象 d归并到 L_i,更新叶结点 L_i的 CF;

3) 否则在叶结点中添加一个新的实体;

4) 如果有容纳数据对象 d的空间,则插入数据对象 d;

5) 否则分裂叶结点 (选择距离最远的一对实体作为种子进行分裂,并按距离最近原则分配剩余结点);

6) 更新路径上的所有非叶结点的 CF;.

7) 算法结束.

(3)设计参数相关挖掘算子 $M|_{\{\delta\}}$

参数相关挖掘算子 $M|_{\{\delta\}}$的主要功能是在初等知识之上进行数据挖掘.在此我们采用凝聚的层次聚类可继承数据挖掘方法 (inheritable data mining - agglomerative hierarchical CLUstering, IDM-AHCLU),可以在初等知识之上进行聚类.下面给出算法伪代码:

功能:在初等知识之上进行数据挖掘.

参数: D—初等知识;

k—聚类个数;

δ—距离阈值.

1) 记 $NumOfClus$为 D中叶结点的个数;

2) 置 $MinDist$的初值为 ∞;

3) 计算 D中距离最小的叶结点,记为: CF_i和 CF_j;

4) 如果 $NumOfClus > k$且 $MinDist \leqslant \delta$则

{

5) 将 CF_i和 CF_j合并到 CF_i中;

6) 删除 CF_j;

7) 将 $NumOfClus$减 1;

8) 转移到 3)步;

}

9) 否则,算法结束.

4 结论 (Conclusion)

知识发现中的可继承性问题是数据挖掘应用中的重要课题,通过对知识发现过程和挖掘算法的形式化描述和分析,抽象出了各个阶段的形式联系及其约束条件,在此基础上提出初等知识的概念.通过对传统挖掘算法、增量式挖掘算法、可继承性挖掘算法的比较,发现可继承性挖掘算法能够有效地提高数据集变化、参数变化情况下的数据挖掘效率.将来的工作重点是根据初等知识的概念和可继承性挖掘算法的特性要求设计具体的挖掘算法,比如:针对关联规则、聚类等设计相应的可继承性挖掘算法.

参 考 文 献 (References)

[1] Fayyad U, Piatetsky-Shapiro G, Smyth P. The KDD process for extracting useful knowledge from volumes of data [J]. Communications of the Association for Computer Machinery, 1996, 39(11): 27~34.

[2] Han J W, Kamber M. Data Mining Concepts and Techniques [M]. 北京:机械工业出版社, 2002.

[3] Ester M, Kriegel H P, Sander J, *et al*. Incremental clustering for mining in a data warehousing environment [A]. Proceedings of the 24th International Conference on Very Large Databases [C]. New York: Morgan Kaufmann Publishers, 1998. 323~333.

[4] 欧阳为民,蔡庆生. 基于时间窗口的增量式关联规则更新技术 [J]. 软件学报, 1999, 10(4): 426~429.

作者简介

冯兴杰 (1969-),男,博士,副教授.研究领域为数据库及数据仓库,智能信息处理理论与技术.

黄亚楼 (1964-),男,教授,博士生导师.研究领域为智能机器人系统,智能信息处理理论与技术,数据挖掘.