第16卷 第12期
2006年12月

计算机技术与发展
COMPUTER TECHNOLOGY AND DEVELOPMENT

Vol.16 No.12
Dec. 2006

基于数据挖掘的课程相关性研究与分析

贾 文,臧明相,周 鸿
(西安电子科技大学 计算机学院,陕西 西安 710071)

摘 要:关联规则是数据库中某些特定事件同时发生的概率的简单陈述。关联规则挖掘就是利用特定方法发掘数据库中潜藏的关联规则的过程。文中主要运用数据挖掘中的关联规则和AprioriTid算法,以考务数据库为挖掘对象,并在挖掘过程中充分运用数据库技术适时地对挖掘数据进行筛选,有效地提高了挖掘效率。最后,对课程间的相关性进行了分析和研究,得到了有效性的结论。

关键词:数据挖掘;关联规则;AprioriTid算法
中图分类号:TP301.6;G434 **文献标识码**:A **文章编号**:1673-629X(2006)12-0178-03

Research of Curriculum Relevance Based on Data Mining

JIA Wen, ZANG Ming-xiang, ZHOU Hong
(Institute of Computer Science, Xidian University, Xi'an 710071, China)

Abstract: Association rule is a simple statement about the cooccurrence probability of certain specific events in the database. Association rule mining is the process which aims to dig the association rule hiding in the database by specific methods. AprioriTtid algorithm and association rule in the data mining are addressed. The examination database are employed as the data mining object. The efficiency of data mining are raised by fully used database technique and filtering original data. The relationship among the curriculums are researched, with conclusion satisfied.

Key words: data mining; association rule; AprioriTid algorithm

1 课程相关性问题的提出

随着科技事业日新月异的进步,信息技术伴随着互联网技术飞速发展。以信息技术为主的高等院校为适应这一发展,不断追新地调整着所开设专业的课程,这种调整不可避免地存在着盲动性和不合理性。教务管理人员以及任课教师却很难直接根据学生的成绩数据分布找出诸如前期课程与后继课程的关系,并据此进行专业课程的合理设置。因此借助于相应的数据挖掘工具,发现数据中隐藏的课程相关规律或模式,并直观地表示相关规律或模式,为决策提供支持,是非常必要的。

关联规则是当前数据挖掘研究的主要方式之一。基于教学考务数据库,运用关联规则挖掘出满足给定支持度和可信度阈值的不同专业、不同课程之间存在的模式或规律,以指导教务管理人员科学地设置专业课程,同时也可指导教师合理地安排教学进度及教学内容。例如,在教学考务数据库中,可以挖掘出"《高级语言编程》成绩优秀的计算机科学专业的学生,《面向对象程序设计》成绩也是优秀的可能性是63%",而电子机械专业学生的这种可能性是30%。这样,可以指导教师及学生加强《高级语言编程》课的教学及学习,以此来提高《面向对象程序设计》课的教学及学习效果,甚至是撤消微电子专业的《面向对象程序设计》课程的设置。

2 课程相关的关联规则的形式定义

以考务数据库中的学生成绩为例来形式化地描述课程相关的关联规则:

设 $I=\{i_1,i_2,\cdots,i_m\}$ 是由 m 个不同的课程组成的集合,$D=\{t_1,t_2,\cdots,t_n,\cdots\}$ 为学生的成绩库记录的集合,其中 $t_i\in I(1\leqslant i\leqslant n)$ 表明 D 中的每一条记录所包含的课程必在 I 中。设 X,Y 是一个 I 中项的集合;T 是 D 中的一个项集(即满足某种条件的一个集合),并且 $T\subseteq I$,如果 $X\subseteq T,Y\subseteq T$[1]。

一个关联规则是形如 $X\Rightarrow Y$ 的蕴涵式,这里 $X\subset I$,$Y\subset I$,并且 $X\cap Y=\varnothing$。规则 $X\Rightarrow Y$ 在成绩数据库 D 中的支持度(support)是成绩记录集中包含 X 和 Y 的记录数与所有记录数之比,记为 support($X\Rightarrow Y$),即

$$\text{support}(X\Rightarrow Y)=|\{T:X\cup Y\subseteq T,T\in D\}|/|D|$$

规则 $X\Rightarrow Y$ 在成绩数据库 D 中的置信度(confidence)是指包含 X 和 Y 的记录数与包含 X 的记录数之比,记为 confidence($X\Rightarrow Y$),即

$$\text{confidence}(X\Rightarrow Y)=|\{T:X\cup Y\subseteq T,T\in D\}|/$$

收稿日期:2006-03-22
作者简介:贾 文(1967-),男,陕西渭南人,工程师,研究方向为软件开发、计算机网络。

$|\{T: X \subseteq T, T \in D\}|$

给定一个学生成绩考务数据库 D,关联规则挖掘问题就是产生支持度和置信度分别大于用户给定的最小支持度(minsup)和最小置信度(minconf)的关联规则。例如,在 1 000 条学生成绩记录中有 600 条显示《高级语言编程》成绩优秀,而这 600 条记录中又有 360 条显示《面向对象程序设计》成绩优秀,则规则《高级语言编程》成绩优秀就会《面向对象程序设计》成绩优秀的可信度 C=360/600=0.6,支持度 S=360/1000=0.36。

从语义的角度来分析,规则的可信度表示这条规则的正确程度;支持度表示用这条规则可以推出百分之几的目标,即这一规则对于整体数据的重要程度。用户可以定义二个阈值,要求数据挖掘系统所生成的规则的支持度和可信度都不小于给定的阈值。

这样,就用蕴含式、支持度和可信度唯一标识了每一个挖掘出来的关联规则。例如,可以这样表示上面提到的例子:

support(《高级语言编程》成绩优秀⇒《面向对象程序设计》成绩优秀)=0.36;

confidence(《高级语言编程》成绩优秀⇒《面向对象设计》成绩优秀)=0.6。

3　数据挖掘过程

数据挖掘(Data Mining)是数据库知识发现 KDD(Knowledge Discovery in Database)的核心,是指从数据库中提取潜在的、有用的、最终可理解的知识的过程。关联规则算法则是数据挖掘的一个重要研究方向,其侧重于确定数据库中不同领域间的联系,找出满足给定支持度和可信度的多个域之间的相互关系[2],最具代表性的是经典的 Apriori 算法,其关联规则的挖掘过程可以分解成以下两个子问题:

(1)找出成绩数据库 D 中所有满足条件的具有用户指定最小支持度的每一个频繁项集 I,产生所有的 I 的非空子集;

(2)对于 I 的每一个非空子集 s,

若 support_count(I)/support_count(s)≥minconf,则输出关联规则"$s \Rightarrow (I - s)$"[3]。

由于第 2 个子问题较为容易和直观,目前大量的研究工作主要都集中在第 1 个子问题上。

表 1 为事务数据库 D 及其结构。

3.1　挖掘算法

挖掘算法采用经典的 AprioriTid 算法,考虑到采用纵向数据结构,在算法上作了一些调整[4]。

算法如下:

输入:事务数据库 D,最小支持度阈值 minsup。

输出:D 中的频繁项集 L。

处理流程:

(1) $\bar{C}_1$ = database D;

表 1　事务数据库 D 及其结构

学号(xh)	姓名(xm)	性别(xb)	课程编号(kcbh)	课程名称(km)	成绩(kscj)	专业名称(zymc)
0301001	习梦丽	女	030701	高数	87	计算机科学
0301002	刘海涛	男	030701	高数	73	计算机科学
0301003	宋文凭	男	030701	高数	81	计算机科学
…	…					
0301001	习梦丽	女	030509	面向对象	92	计算机科学
0301002	刘海涛	男	030509	面向对象	84	计算机科学
0301003	宋文凭	男	030509	面向对象	69	计算机科学
…	…					
0501001	史晓晖	男	050401	电路基础	81	电子机械

(2) $L_1 = \{C \in \bar{C}_1 \mid C.\text{count}/|D| \geqslant \text{minsup}\}$; $k = 2$;

(3) while($L_{k-1} \neq \emptyset$) do begin //直到不能生成频繁项目集为止

(4) C_K = apriori - gen(L_{k-1}); //生成含 k 个元素的候选项目集

(5) $\bar{C}_k = \emptyset$;

(6) For all $T \in \bar{C}_{k-1}$ do begin

//确定包含在事务中的标志为 t.TID 的 C_K 的候选项目集

(7) $C_t = \{C \in C_K \mid (C - C[k]) \in \text{t.set - of - itemsets} \wedge (C - C[k-1]) \in \text{t.set - of - itemsets}\}$;

(8) 计算集合 C_t 中增加的所有候选者;

(9) if($C_t \neq \emptyset$)then $\bar{C}_k$ += < t.TID, C_t >;

(10) end;

(11) $L_k = \{C \in C_K \mid C.\text{count}/|D| \geqslant \text{minsup}\}$;

(12) end;

(13) Answer = $U_k L_k$;

procedure apriori - gen(L_{k-1})

(1) for each $p \in L_{k-1}$

(2) for each $q \in L_{k-1}$

(3) if (p.item$_1$ = q.item$_1$) ∧ … ∧ (p.item$_{k-2}$ = q.item$_{k-2}$) ∧ (p.item$_{k-1}$ < q.item$_{k-1}$)

(4) $c = p \oplus q$; //连接两个项集

(5) if has - infrequent - itemset(c, L_{k-1})

(6) delete c; //除去不可能产生频繁项集的候选集

(7) else $C_k = C_k \cup \{c\}$

(8)

(9) return C_k

procedure has - infrequent - itemset(c, L_{k-1})

(1) for each(k - 1)subset s of c

(2) if $s \notin L_{k-1}$return TRUE; else return FALSE

其中,D 表示数据库;minsup表示给定的最小支持度;Answer 表示所有频繁项目集。

AprioriTid 方法在由候选频繁项目集确定频繁项目集时只需扫描一遍数据库,但由于数据库采用纵向结构,每

个事务的数据分布在许多条记录中，故笔者改进算法，为候选频繁项目集的每一个项目，逐遍扫描事务库 C_1，以得到所需数据。为此从数据库实现技术上着手，通过 SQL 中数据查询的集合特性、查询的可嵌套性，在循环中设计了一条 SQL 语句[5]：

select count(*) from t_bk where(km = kcbh1 and (xh in(select xh from t_bk where km = kcbh2 and (xh in(select xh from t_bk where km = kcbh3 and … kscj > 80)) and kscj > 80)) and kscj > 80)；

语句中 kcbh1，kcbh2，…为不同课程。

该语句可以一次求出几门功课同时为优秀的学生人数，有效地减少了程序的循环量和信息搜索量，从而大大地提高程序的运行效率。

3.2 挖掘结果

在事务数据库 D 有 15 000 条记录的情况下，生成成绩优秀的频繁 1 项集，并使用 SQL 语句：

select kcbh, km, xkrs, count(*) as rs, zymc , sup from Fr_tmp into Fr_D where (zymc = 计算机科学 or zymc = 电子机械) and (kcbh = 060307 or kcbh = 030509) group by zymc

把此频繁项集按记录所在的专业及课程筛选并放在 Fr_D 数据库中(见表 2)。

表 2 Fr_D 数据库

课程编号 (kcbh)	课程名称 (km)	选课人数 (xkrs)	成绩优秀人数(rs)	专业名称 (zymc)	支持度 (sup)
060307	高级语言编程	1016	803	计算机科学	0.79
060307	高级语言编程	451	339	电子机械	0.75
030509	面向对象	1016	751	计算机科学	0.74
030509	面向对象	451	203	电子机械	0.45

由频繁 1 项集生成频繁 2 项集见表 3。

表 3 由频繁 1 项集生成频繁 2 项集

课程名称 1 (km1)	课程名称 2 (km1)	选课人数 (xkrs)	成绩优秀人数(rs)	专业名称 (zymc)	支持度 (sup)
高级语言编程	面向对象	1016	651	计算机科学	0.63
高级语言编程	面向对象	451	126	电子机械	0.28

4 结 论

通过以上的分析和研究，在笔者设定 minsup 为 0.1 的情况下，使用 AprioriTid 算法进行课程相关性分析，并在挖掘过程中充分运用数据库的相关技术，适时地对挖掘的中间结果进行有针对性的处理，有效地提高了实验挖掘效率。也可得出在电子机械专业开设《面向对象程序设计》课程可能存在不合理性的结论，同时也说明了关联规则在课程相关性分析上的有效性和实用性。

参考文献：

[1] Hand D, Mannila H, Smyth P. Principles of Data Mining[R]. USA: Massachusetts Institute of Technology, 2001.

[2] Fayyad U, Piatetsky - Shapiro G, Smyth P. Knowledge Discovery and Data Mining. Towards a Unifying Framework[C] // Proceedings of the second International Conference on Knowledge Discovery and Data Mining(KDD'96). Portland, Oregon: AAAI Press, 1996.

[3] 毛国君. 数据挖掘原理与算法[M]. 北京：清华大学出版社，2005.

[4] 李德仁，王树良，李德毅，等. 论空间数据挖掘和知识发现的理论与方法[J]. 武汉大学学报：信息科学版，2002，27(6)：221 - 233.

[5] Li D, Di K, Li D, et al. Mining Association rules with Linguistic Cloud Models[J]. Journal of Software, 2000, 11(2): 143 - 158.

(上接第 123 页)

5 仿真结果

导出 VHDL 模型块到 ISE 中进行仿真[5]，得到如图 6 所示的仿真波形。

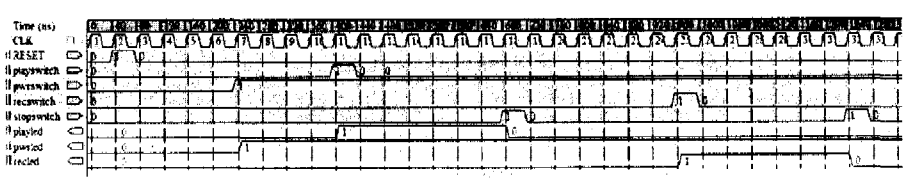

图 6 仿真波形

6 结束语

综上所述，使用 STATECAD 设计状态机的过程简单、快捷、有效，实现状态机设计的自动化，大大缩短了电子产品的研发周期。

参考文献：

[1] 朱明程. Xilinx 数字系统现场集成技术[M]. 南京：东南大学出版社，2001.

[2] 王 诚，薛小刚，钟信潮. FPGA/CPLD 设计工具—Xilinx ISE 使用详解[M]. 北京：人民邮电出版社，2005.

[3] 唐亚平，王学梅. 基于 VHDL 的有限状态机的优化问题[J]. 电子元器件应用，2005，7(8)：59 - 61.

[4] 林 敏. VHDL 数字系统设计与高层次综合[M]. 北京：电子工业出版社，2002.

[5] 石俊斌，林 辉. 在 PLD 开发中提高 VHDL 的综合质量[J]. 单片机与嵌入式系统应用，2003(4)：54 - 57.