第35卷 第20期 计 算 机 工 程 2009年10月
Vol.35 No.20 Computer Engineering October 2009
·软件技术与数据库· 文章编号：1000—3428(2009)20—0086—02 文献标识码：A 中图分类号：TP181

基于粒计算的关联规则挖掘算法

张月琴，晏清微
(太原理工大学计算机与软件学院，太原 030024)

摘 要：讨论粒计算在关联规则挖掘中的应用，通过对基本信息粒的划分、对粒子对象集合的映射，减少扫描项集所在的对象集合，提高算法的运行效率，从而更好地处理海量数据的规则发现，更适用于支持度较小、复杂度较高的数据集。仿真试验证明该算法有较低的求解复杂度及较高的求解效率。

关键词：粒计算；关联规则；频繁项集

Association Rules Mining Algorithm Based on Granular Computing

ZHANG Yue-qin, YAN Qing-wei
(College of Computer and Software, Taiyuan University of Technology, Taiyuan 030024)

【Abstract】 This paper discusses the application of granular computing in association rules mining. Through partition of information granules and map of granule object sets, the algorithm reduces the object sets required when scanning datasets, and improves the running efficiency, so that it can deal with the association rules discovery of massive dataset better, and it is more suitable for the dataset with small support and high complexity. Experimental results show that the algorithm has low computing complexity and high efficiency.

【Key words】 granular computing; association rules; frequent itemsets

1 概述

关联规则是数据挖掘中的研究热点，Apriori[1]关联规则挖掘算法是其中的经典算法，但它存在2个致命的性能瓶颈：(1)需要多次扫描数据库；(2)产生庞大的候选集。针对Apriori算法的不足，目前的关联规则挖掘算法主要从以下4个方面进行改进：(1)减少扫描数据库次数以减少I/O时间。(2)尽快计算候选项集支持度。(3)对候选项集进行最大程度的剪枝。(4)并行产生候选项目集。改进方案有FP-Growth算法[2]、DHP算法[3]等。但数据库很大时，构造基于内存的FP-Tree是不现实的，而DHP算法构造HASH表的开销比较大。

粒计算为复杂问题的求解提供了新的思路。从粒度的角度看关联规则挖掘，其超集频繁模式可以看作为“粗粒”，子频繁模式可以看作是“细粒”，那么频繁模式的父子关系可以看作是粒空间中的一种偏序关系。因此，关联模式中频繁模式下的父子关系满足商空间理论[4]中的保真原理和保假原理。文献[5]将商空间理论与粒之间的逻辑运算规则用于关联规则挖掘，通过调整粒的空间范围来提高算法的执行效率。粒计算的另一个重要模型粗糙集理论在关联规则挖掘及属性约简方面也有很多的研究[6]。可见粒计算在关联规则中的应用是有效且值得研究的。本文从另一个角度将粒计算思想用于关联规则的挖掘。通过对海量数据进行合适的粒化，并对粒子进行合成及分解，来降低问题的求解复杂度，提高问题求解效率，仿真试验证明了其可行性。

2 相关知识

定义 1 设 $S=\left(U,At=(C\cup D),L,\{Va|a\in At\},\{Ia|a\in At\}\right)$ 是一个信息系统，U 是所研究的对象集，At 是这些对象所具有的属性集，包括条件属性 C 和决策属性 D，L 是决策逻辑语言，Va 为属性值集，$Ia:U\to Va$ 为信息函数。取任意的 $a\in At$，$a(x)=v\in Va$，其中，$x\in U$，Va 是属性值集，称 $a(x)=v$，也写成 a_v 或 (a,v) 为 S 上的描述或Rough逻辑中的原子公式。原子公式可以通过逻辑与、或和非进行运算。对于 φ 以及解释区域 $m_s(\varphi)$，两者组成的有序对 $(\varphi,m_s(\varphi))$ 被称作基本粒(granule)。

定义 2 设 $gs(Gr)$ 为粒子元素的集合，表示从粒子对象集合的映射函数。对于任意粒子 $Gr=(\varphi,f^{-1}(\varphi))$，有 $gs(Gr)=f^{-1}(\varphi)$。

定义 3 设 $Gr1=(\varphi,f^{-1}(\varphi))$ 和 $Gr2=(\psi,f^{-1}(\psi))$ 是信息系统中的2个粒子，按照经典逻辑中的5个连接词，粒子之间的运算规则递归地定义如下：

$$(1)\ \neg Gr1\Leftrightarrow(\varphi,f^{-1}(\varphi))\Leftrightarrow(\neg\varphi,f^{-1}(\neg\varphi))$$

$$(2)\ Gr1\wedge Gr2\Leftrightarrow(\varphi,f^{-1}(\varphi))\wedge(\psi,f^{-1}(\psi))\Leftrightarrow(\varphi\wedge\psi,f^{-1}(\varphi)\cap f^{-1}(\psi))$$

$$(3)\ Gr1\vee Gr2\Leftrightarrow(\varphi,f^{-1}(\varphi))\vee(\psi,f^{-1}(\psi))\Leftrightarrow(\varphi\vee\psi,f^{-1}(\varphi)\cup f^{-1}(\psi))$$

$$(4)\ Gr1\to Gr2\Leftrightarrow(\varphi,f^{-1}(\varphi))\to(\psi,f^{-1}(\psi))\Leftrightarrow(\varphi\to\psi,f^{-1}(\varphi)\subseteq f^{-1}(\psi))$$

$$(5)\ Gr1\leftrightarrow Gr2\Leftrightarrow(\varphi,f^{-1}(\varphi))\leftrightarrow(\psi,f^{-1}(\psi))\Leftrightarrow(\varphi\leftrightarrow\psi,f^{-1}(\varphi)=f^{-1}(\psi))$$

基金项目：山西省自然科学基金资助项目(2008011028-1)
作者简介：张月琴(1964－)，女，副教授、硕士，主研方向：智能信息处理，粒计算；晏清微，硕士研究生
收稿日期：2009-01-30 **E-mail：**yanqw22468@163.com

其中，逻辑连接词→不表示经典逻辑下的蕴含关系，而表示2个粒子元素集合的包含关系。

定义 4 粒 $(\varphi, f^{-1}(\varphi))$ 的粒度定义为：$G(\varphi)=\left|f^{-1}(\varphi)\right|$，它指一个粒的大小。

3 基于粒计算的关联规则挖掘

基于粒计算的关联规则挖掘算法 GRA_Ass 的基本思想是：首先对信息系统进行粒子分解，设信息系统 S 分解后的粒子集合为 GrS，对于任意 $Gr \in GrS$ 是 S 上的一个描述 (a,v)，满足 $|Gr| \geq minSup$，同时，对于任意 Gr，要记录其粒子元素的集合，即 $gs(Gr)$，此粒子元素集合构成信息系统的一个粒子空间。然后对基本粒子按照要求进行逻辑运算，得到所有的频繁项集。粒子元素集合的获取算法过程如下：

输入 信息系统 $S=\{U, C\cup D, V, f\}$ 及 $minSup$

输出 信息系统的粒子空间 GrS 及所有粒子元素的集合 $gs(Gri)$

(1)令 $Grs=\varnothing$，$m=|Attr|$，$n=|U|$。

(2)对各个属性依次按等价关系进行划分 $U/IND(Attr_i)$，$\forall g_{ij} \in U/IND(Attr_i)$ 是第 i 个属性的第 j 个等价类。

(3)扫描数据库，求每个 g_{ij} 的支持度及其粒子元素的集合 $gs(g_{ij})$。

(4)对于任意粒子元素 g_{ij}，如果其支持度大于最小支持度 $minSup$，则将其加入粒子集合 GrS 中。

该算法在计算每个信息粒支持度的同时，记录了粒子元素的集合，使得以不同项开头的项集可以针对不同的对象集求解。然后对粒子空间的粒子进行逻辑运算，得到所有的频繁项集。

以下算法是对粒子集合中的粒进行逻辑运算，得到所有的频繁项集，为了减少扫描的对象集，对粒子集合中的粒逐一求以其开头的所有频繁项集。

输入 粒子空间 GrS

输出 所有的频繁项集 FrS

(1)令 $Frs=\varnothing$。

(2)粒子空间 GrS 中共有 $|Grs|$ 个信息粒，对各个粒元素 Gri 逐一求以 Gri 开头的项集 $ItemSets(Gri)$，同时在其粒子元素集合 $gs(Gri)$ 范围内求项集的支持度，如果某项曾经作为子集被确定不是频繁的，则其合成项必然也不是频繁的，无须再计算其支持度。

(3)对于 $ItemSets(Gri)$ 中的任意一项，如果其支持度大于 $minSup$，则是频繁的，将此项加入频繁项集合 FrS 中。其所有子集也是频繁的，因此，将所有子集也加入 FrS 中。

每次只考虑以某一个频繁 1 项集开头的所有频繁项集，并在计算支持度时只在此项集的粒子对象集合中进行，以缩小要考虑的问题空间，降低对内存的要求，同时为并行算法提供思路。

4 算法分析

本算法主要从以下 2 个方面进行改进：

(1)不管采用哪种算法，对数据库进行扫描的主要原因是要求得每个候选项集的支持度，以确定其是否为频繁项集，假设一个数据集最大项集为 K 项集，则求所有候选项集支持度至少需要循环 $\sum_{i=1}^{K}|C_i|\times num$ 次，num 为数据库记录数(或事务数)。而如果考虑以某一个项 Gr 开头的所有频繁项集，在扫描时只需要考虑 Gr 所在的数据库记录(事务)，因此，在求粒子集合时记录了粒子元素的集合，使得之后的项集在计算支持度时只需考虑相关的对象集合，那么对于最大项集为 K 项集的数据集，计算所有候选项集支持度总共需要循环 $\sum_{i=1}^{K}|C_i|\times num\times\alpha$ 次，其中，$0\leq\alpha\leq1$。

(2)在求以某一个项 $Gr1$ 开头的 k-1 项集产生 k 项集时，对项集之间的合并加以约束，首先要求 2 个 k-1 项集的前 k-2 项必须相同，其次要求 2 个 k-1 项集的最后一项属于不同属性列的等价类，这样尽量减少了候选项的产生。在求得以 $Gr1$ 开头的频繁项集后，根据“频繁项集的子集必是频繁的”，将减去 $Gr1$ 项后的所有子集也归入频繁项集合中，这样，再求另一个以 $Gr2$ 开头的频繁项集时，如果其合成项之前已加入到频繁项集合中，就不需要再做考虑，这也使问题空间的范围得以缩小。

5 仿真测试及结果分析

试验用计算机的配置如下：CPU 为 Pentium4 2.66 GHz，内存为 512 MB，运行平台是 Windows XP Professional SP2 下的 WEKA+eclipse。WEKA 是一个公开的数据挖掘工作平台，集合了大量数据挖掘算法，本试验是在 WEKA 下将 Apriori 算法对频繁项集的求解与本文算法做比较，数据集采用 5 组 arff 格式的 UCI 数据集，如表 1 所示。表 2 给出 2 种算法对数据集在相同支持度下获取频繁项集所消耗的时间。表 3 给出 2 种算法对数据集在不同支持度下获取频繁项集所消耗的时间。

表 1 5 个 UCI 数据集

数据集	样本个数	属性个数
Sobean	683	36
Kropt	28 056	7
Connect4	67 557	6
Adult	32 561	9
Ticdata_categ47	5 822	47

表 2 相同支持度时算法运行时间对比

数据集	支持度	Apriori 耗时/ms	GRA_Ass 耗时/ms
sobean	546	47	48
kropt	2 806	2 031	921
Connect	6 757	3 531	2 062

表 3 不同支持度时算法运行时间对比

数据集	支持度	Apriori 耗时/ms	GRA_Ass 耗时/ms
adult	13 024	843	836
	6 512	1 860	1 423
	4 884	2 605	1 813
Ticdata_categ47	2 329	3 687	2 548
	1 747	5 672	3 565
	1 164	16 594	8 875

可以看出，本文算法对海量数据集频繁项集求解的耗时比 Apriori 算法小；另外，对同一数据集，支持度越小，本文算法所耗费的时间代价越小，这是因为支持度越小，所需扫描的对象总数大大减少。

6 结束语

随着网络技术、数据库技术的飞速发展，需要处理的数据规模越来越大，因此，如何有效地对海量数据进行数据挖掘是一个亟待解决的问题。日趋成熟的粒计算研究为数据挖掘算法提供了新的方法和思路。本文从粒的角度考虑关联规则挖掘中的频繁项集，通过对粒子对象集合的映射、减少扫描项集所在的对象集合来提高算法的效率，试验证明本算法对数据量较大、复杂度较高的数据集是有效的。

(下转第 90 页)