一种基于人工免疫的新的频繁项挖掘算法*)

王 评 陈国龙

(福州大学 数学与计算机学院 福州 350002)

摘 要 以往算法的研究主要围绕着减少候选项目集进而减少事务数据库的扫描次数的角度,先求出候选项集,再计算候选项集的支持度求得频繁项集。本文改变过去求频繁项集的角度,从新的角度来看频繁项目集的定义,同时结合人工免疫的特点,设计一个基于人工免疫的新频繁项集挖掘算法。本文详细介绍了算法设计等。新算法的复杂度与支持度,数据库总容量有关。验证实验的结果与其他算法相比较证明了该算法的可行性、有效性和完备性。

关键词 免疫算法,频繁项集,支持度,关联规则

Information Feature Extraction Based on Immune Aglorithm

WANG Ping CHEN Guo-Long

(School of Math & Computer Scient, FuZhou University, Fuzhou 350002)

Abstract Tradition algorithm attempted to improve the mining efficiency reducing the number of database passes to control the I/O cost, which at first derives candidate itemsets from tuples in database, and count support of candidate itemsets to get frequent itemsets. Changing the former way of discovering frequent itemsets, we can understand the definition of frequent itemsets from another standpoint. Making use of atifical immune strategy , this paper present a new algorithm. In the paper, the algorithm is introduced in detail. The cost of new algorithm is related to the support and the total number of database. Finally, simulation shows the correctness and validity compared with other algorithm.

Keywords Immune algorithm, Support, Associate rules , Frequent itemsets

1 引言

经典频繁项挖掘算法 Apriori 算法[1],形成 k-频繁项集后,在搜索 k+1-候选项集时,由两个 k-频繁项集连接产生。一些不符合条件的项目集因为不需要进一步处理被剪除。剩下的项目集就是这一步的候选项目集,第二步计算候选项目集的支持度。有时候每一个频繁项集都是下一步的候选项集,都要仔细考虑。候选项目集的支持度计算需要扫描数据库。因此频繁项集挖掘过程的主要花费包括扫描数据库(I/O 时间)和新的候选项集产生的时间(CPU)。重复的扫描数据库占了大部分处理时间。候选项目集越大,扫描数据库的次数越多。频繁项集挖掘的处理过程花费也就越大于传统的算法,所以传统算法经常遇到"项集生成瓶颈"的问题,即由于生成的候选项集太多而造成算法效率的急剧降低。许多人或者用 bottom-up 方法,或者 top-down 方法,或者两者结合的方法减少候选项集。我们可以改变以往求频繁项集的角度,从另一角度来看频繁项目集的定义,频繁项集满足支持度也就是该项集在数据库中有等于或大于支持度计数条的事务包含这样的项目子集。如从包含 100 条事务的数据库中,寻找支持度为 0.5%的频繁项集,那么必然有 5 条以上的事务中,都包括这样的项目子集。从这个角度的理解,同时结合生物免疫机制给予的启发,本文设计了一个新的基于免疫最大频繁项集挖掘算法 FDIA(Frequent Itemset Discovering Based Immue Agorithm)。

2 基本原理和定义

定义 1 设 $I=\{i_1,i_2,\cdots,i_n\}$是项的集合,其中的元素称为项(item)。记 D 为事务 T 的集合,这里 $T\subseteq I$ 。对应每一个事务有唯一的标识,记作 TID。设 X 是一个 I 中项的集合,如果 $X\subseteq T$,那么称事务 T 包含 X。

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 $X\cap Y=\Phi$。规则 $X\Rightarrow Y$ 在事务数据库 D 中的支持度(support)是 D 中包含 X 和 Y 的事务数与所有事务数之比,记为 $support(X\Rightarrow Y)$,即:

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

规则 $X\Rightarrow Y$ 在交易集中的可信度(confidence)是指包含 X 和 Y 的交易数与包含 X 的交易数之比,记为 $confidence(X\Rightarrow Y)$,即:

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

给定一个交易集 D,挖掘关联规则问题就是产生支持度和可信度分别大于用户给定的最小支持度(min_supp)和最小可信度(min_conf)的频繁项目集[2]。

定义 2 事务一项目集信息表。给定项目集 $I=\{i_1,\cdots,i_m\}$,事务数据库 $D=\{U,A\}$其中 $U=\{x_1,\cdots,x_m\}$,$A=T\cup C$,$C=\{C_1,\cdots,C_n\}$,项目集合 $C_i\subseteq I$,$T=\{t_1,\cdots,t_n\}$,是事务标识符集合,则每个对象 $x_i\in U$ 有惟一个事务标识 Tid 和一个项目集相对应。为精确描述这种事务和项目间的对应关系,我们称 $S=(U,A,V,f)$为事务项目相关信息表,简称信息表。这里,$V=\{0,1\}$,$f:TXC\rightarrow V$[5]。

以下简述免疫应答原理。生物的免疫系统由淋巴细胞(包括 B 细胞和 T 细胞)和噬菌细胞等组成。B 细胞分泌抗体结合识别抗原。T 细胞促进和抑制 B 细胞的产生和分化。抗原首次入侵机体时,少量的 B 细胞被激活,分化的部分细胞演变为浆细胞并作为基于细胞存于体内,其余细胞繁殖一定数目的克隆细胞并经由亲和突变产生识别抗原能力更强的细胞且分泌抗体,此时抗原被中立;经突变后,识别能力弱或产生自反应的克隆被清除,同时免疫系统不断产生新细胞维持几天的自身平衡,这种应答属于初次应答;当同一抗原或相似抗原再次入侵时,体内基于细胞对抗原作出快速反应,此时抗原被清除较快,这次应答属于再次应答。在应答中,应答能力强的 B 细胞被激活参与进化。T 细胞在 B 细胞克隆到一

*)该课题得到福建省自然科学基金(编号:A0410010)和福州大学科技发展基金(编号 2003-xq-23)资助。王 评 硕士,研究方向为智能算法与信息处理。

定程度时抑制 B 细胞，防止无限克隆。T 细胞对 B 细胞起调节作用。抗体是特异的，因为一种抗体只强烈地结合几个类似的抗原决定基结构或模式[3,4]。

免疫系统由许多分布的具有一定功能的个体相互作用组成一个功能强大、复杂的大系统。免疫系统具备的学习，记忆，自适应等特性，借助生物免疫机制和免疫应答过程可设计基于免疫算法的最大频繁项挖掘算法(FDIA)。

3 FDIA 算法

3.1 FDIA 算法运行机制

FDIA 算法由 7 种机理构成：克隆选择，细胞克隆，亲和突变，群体组合，免疫选择，募集新成员，抗原聚类以及记忆细胞演化，其运行机制如图 1 所示。

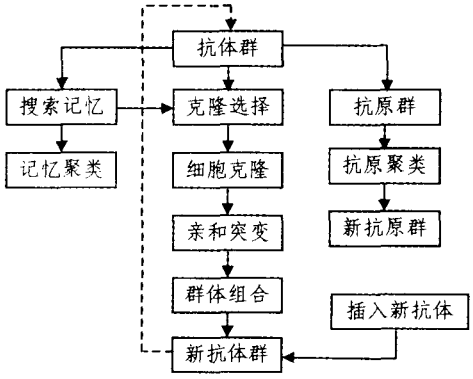

图 1 FDIA 算法运行机制

对图 1 部分操作给予定义：

(1)*抗体编码* 免疫算法的抗体编码一般是二进制或者十进制。此抗体由信息表中 n_0 行组成，即抗体每一个单元对应于事务数据库的一个事务的项目集。抗原为候选项目集。抗体的长度与支持度和记录总数相关。按如下公式获得：抗体长度 n_0 = 记录数 × 支持度。如数据库总容量 1000，支持度 1%，则 n_0 = 10，抗体可以编码如下：

T103	T21	T256	T34	T2	T7	T943	T339	T24	T462

这里 T_i 指信息表中第 i 行，$i \leq$ 数据库总容量。

(2)*亲和度函数* 分抗原与抗体的亲和度和抗体之间的亲和度。抗原与抗体亲和度体现抗原识别能力，实际代表的是问题的解和实际问题的接近程度。抗体之间的亲和度体现抗体在群体中与其他抗体的相似程度。

抗体 Ab 与抗原的亲和度：

$$f(u)=Ab_{i1} \cap Ab_{i2} \cap \cdots \cap Ab_{in_0}$$

其中 $Ab=Ab_{i1}Ab_{i2}\cdots Ab_{in_0}$，$n_0$ 为抗体长度。由 Ab 的组成和 $f(u)$ 的函数可知，$f(u)$ 的值中 1 的个数越多，亲和力越强，1 的个数越少，亲和力越弱。

3.2 抗原聚类算法

每次群体更新，搜索频繁项集，不一定是最大频繁项集，抗原群中包含了冗余个体，因而引入聚类算法处理群体中的过剩的个体。

给定抗原群体 $C=\{x_1,\cdots,x_m\}$，对新抗原 $C'=\{x_{m+1},\cdots,x_{m+k}\}$ 即将加入到 C 中，做如下过程：

step1 置 $i=1$

step2 对 C' 中一个个体 x_{m+i}，判断是否 $x_{m+i} \subseteq x_t, x_t \in C$，如果不是则 x_{m+i} 加入到 C 中。

step3 若 $i<k$，则置 $i=i+1$，返回第二步，否则算法结束。

3.3 FDIA 算法

在算法实现中，设事务数据库总容量 N，满足支持度计数值为 M。我们基于 FDIA 算法的框架描述，将抗原对应于当前状态下所要寻找的最大频繁项集，而每个抗体对应于事务数据库是 M 条项目集。为根据模拟图的机制，FDIA 算法设计如下：

step1 初始群体。在 C 中随机选取 n_0 个项目集组成一个抗体，其中 $C \in$ 事务数据 D，则可以随机产生 N 个抗体形成初始抗体群 A_0，这里 N 指抗体群的规模。初始抗原群和初始记忆细胞群均为空。

Step2 抗原及记忆细胞获取。计算抗体群 A_n 中的抗体亲和度，复制 A_n 中亲和度最强的个体，作为记忆细胞加入记忆细胞群，获 M_{n+1}；把 A_n 计算亲和度后得到的项目集视为抗原与抗原群组合，并用抗原聚类算法获 Ag_{n+1}。

Step3 克隆选择。按上一步计算 A_n 的亲和度结果，在选择率 α 下，选择 A_n 中 $s=Int(\alpha|A_n|)$ 个较高亲和度的抗体构成 B_n，设 $B_n=\{Ab_1,\cdots,Ab_s\}$，其中 $0<\alpha<1$。

Setp4 细胞克隆。B_n 中 Ab_i 经过细胞克隆算子获克隆细胞子群 C_i^0，$0<i<s$。

Step5 亲和突变。分两种突变方式超突变和均匀突变。细胞子群 C_i^0 与 C_j^0 之间作用，按超突变算子进行超突变，超突变算子的设置应该使其保留个体中长项目集单元。细胞子群 C_i^0 内部按均匀突变算子进行均匀突变。均匀突变算子的设置应该使其保留个体中长项目集单元。以上过程获群体 W。

Step6 群体组合和抑制。计算 W 中抗体与抗体的亲和度，并按 $\| Ab_i - Ab_j \| < \delta$，其中 $Ab_i, Ab_j \in W$。选择剩余抗体构成群体 R。

Step7 在 C 中随机组成 h_0 个抗体插入到群体 R 中获得新的抗体群 A_{n+1}，其中 $C \in$ 事务数据 D，$h_0=Int(\eta|A_n|)$。

Step7 判断循环条件，若条件不终止，返回第二步，否则终止循环。

Step8 从抗原群中提取最大频繁项集。

该算法中参数 α、η、δ 可根据 A_n 大小调节设定。

4 算法分析和实验结果

频繁项目的挖掘主要花费在扫描数据库时间(I/O 时间)。FDIA 算法不需要计算候选项目集的支持度，主要花费分两部分：(1)在群体更新时从数据库中提取新抗体部分单元的项目集。(2)抗体内的逻辑交运算次数 = 支持度计数 × 群体规模 × 迭代次数。

第一部分的花费整个过程大概估计相当于扫描一次数据库，大大减少了数据扫描量，节省了磁盘 I/O 时间。第二部分的花费相对于如今迅速发展的 CPU 速度，是可行的。

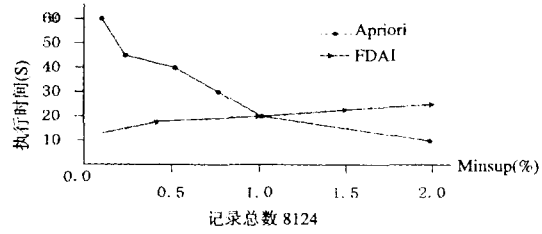

图 2

为验证 FDIA 算法的有效性、可扩展性与先进性，在 PII-II/256M 的计算机上，我们分别实现了 Apriori 和 FDIA 算法。根据数据库中事务数以及不同的最小支持度运行算法进行比较实验，得出了下面的结果。其中，数据库中的数据来自于 JeffSchlimmer 提供的 mushroom 数据，具体参数如下：

数据库文件大小为 2 28 MB，库务总数为 8124，项目数 119，事务的平均长度 $|T|=13$。图 2 是在数据库 $|D|=8124$ 条件下，最小支持度从 2% 减少到 0.1% 两种算法运行时间的

比较。从图2可以看出，随着最小支持度的减少，FDIA算法的运行时间在不断减少，Apriori算法的时间在增加。

而且FDIA算法的斜率要比Apriori算法的斜率小很多。因为支持度越少，候选项目集越多，Apriori算法扫描数据库的次数也越多。而FDIA算法不需要扫描数据库，支持度变化对它的影响不大。这正是FDIA算法的优势所在。

总结 本文提出的FDIA频繁项挖掘方法，利用生物免疫机制的特点，结合免疫算法，从新的角度挖掘频繁项集。与人工免疫的结合，为频繁模式挖掘开辟了新的途径，新的方法。我们认为这是一个有前途的研究思路。由于该算法是研究上第一次把人工免疫与频繁模式结合，只是一个基本算法，很多地方需要改进。如把群体更新时，如何产生新抗体，使新的抗体尽量挖掘到最大的频繁项集。以及在基础上如何根据数据库总容量调节群体的规模。如何把事务项目集的一些约束联系起来等都是今后非常有用的研究方向。

参考文献

1 Agrawal R, Imielinski T, Swami A. Mining association rules between sets of items in large database. In: Peter Buneman, Sushil Ajodia eds. Proc of ACM SIGMOD Conf on Management of Data, New York: ACM Press, 1993. 207～216
2 Han Jiawei, Kamber Micheline. Data Mining: Concepts and Techniques. Morgan Kaufmann Publishers, 2001
3 S de Castro L N, Von Zuben F J. Artificial Immune Systems: Part IBasic Theory and Applications[R]: [TR-DCA01/99]. 1999
4 Chun J S, Kim M K, et al. Shape Optimization of Electromagnetic Devices Using Immune Algorithm. IEEE Trans on Magnetics, 1997, 33(2): 1876～1897
5 王晓峰，王天然，等. 一种自顶向下挖掘长频繁项的有效方法. 计算机研究与发展，2004，41(1)：148～155
6 Lin Dao-I, Kedem Z M. Princer-Search: A New Algorithm for Discovering the Maximum Frequent Set. In: Bertram Ludascher, Wolfgang May, eds. Proc of the 6th European Conf. on Extending database technology, Proc. Lecture Note in Computer 1377. Berlin: Springer 1998, 1998. 105～119

(上接第148页)

create)。涉及Java安全许可，type属性包含许可类型的类的名字(如：Java. net.)跟随着相关的许可(如open打开)。如果该行为需要额外的信息定义操作的对象，则在action元素中包含Target子元素。

result元素被用来定义操作的结果。该元素包含有status属性暗示行为是否成功。属性type定义了result标记所附加的数据类型。如果没有结果数据，那么result元素为空，type属性被设置成none。

3.4 审计点

在每个系统调用处(前面所述的Aglet事件)设置审计点，Aglet的源代码开放性使得我们可以修改其代码来捕获安全相关的事件，记录完整的信息关于Aglet的身份、请求的操作，请求的参数、操作的目标、请求的结果。安全相关操作的捕获可以通过增强Aglet系统的原语来达到。而且，Aglet系统所实现的安全管理器也可以被扩展来记录成功和失败的操作的参数[5]。

此外，Aglet系统所实现的安全管理器被修改来记录有关系统级别的操作的详细信息。例如，文件操作(open、write等)、socket操作(accept等)，系统属性操作(如修改Java. home属性等)。

为了记录这些审计事件，有关Aglet的信息必须从相应的Aglet的线程中获取。因此我们给AgletThread类增加新的方法getAuditInfo，该方法返回Aglet的AgletInfo结构，包含有Aglet身份的各种信息。

3.5 审计文件分析器

审计日志的分析是审计的一项重要功能。由于审计文件格式是XML格式的，我们采用采用JDOM技术来分析审计文件。JDOM的处理方式有些类似于DOM，但它主要是用SAX实现的[6]。因此它既有DOM方式处理的方便性，同时又不必担心处理速度和内存的问题，这在日志文件比较大的时候就显得更为重要。

分析器可以自由选择要分析的日志文件，也可以根据日期、审计事件、审计级别等进行查询。能够根据日志文件生成各种需求的结果。

3.6 异常处理

当审计系统出现错误或者审计数据填满时候，必须根据审计配置文件中所设定的审计连续性策略来进行异常处理。

(1)*审计数据填满* 审计数据可能会增长得很快，必须要限制审计文件所占用的文件系统空间，审计管理员可以配置文件系统的预留空间(如8%)，在写磁盘时候进行检查，如果低于预留值，则视为异常情况，需要紧急处理。目前系统支持两种简单的处理：(1)关闭审计子系统。系统继续运行，但是发出警告提示：审计日志被填满。(2)关闭整个系统。

(2)*审计系统出现错误* 审计系统运行过程中出错的可能会有很多，如文件创建或者读写操作失败等，这些错误往往很难得到满意的恢复，从而导致审计子系统不可用。当出现这种异常情况的时候，需要紧急处理。目前系统支持两种简单的处理：(1)关闭审计子系统。系统继续运行，但是发出警告提示：系统审计子系统不可用。(2)关闭整个系统。

4 审计子系统的安全保护

审计是移动Agent系统的重要组成部分，它监督着系统各种安全特性的实施。如果审计系统自身的安全性不够，审计数据的可靠性无法得到保障，那么就无法提供准确的事后分析和追踪。在审计子系统的设计与实现中，我们考虑了Aglet系统的安全特性，保证了审计的自身安全。

(1)程序和代码的安全。审计数据的采集记录部分与Aglet平台相关，审计进程的执行不受外界影响，审计管理员配置程序和审计数据分析程序严格遵守强访问控制MAC和最小特权控制PAC，保证只有审计管理员才能有权利使用这些程序。

(2)审计配置文件和审计文件的安全。审计配置文件和审计文件都必须施加MAC控制，确保只有审计员才能查看和修改审计配置。此外，对审计文件进行加密，保证泄漏出去别人也无法看到。

结束语 本文针对Aglet系统设计并实现了一个审计子系统，审计采用XML格式的审计记录，能够灵活方便地完成审计功能，并通过审计文件分析器能够对审计日志进行有效的事后分析与追踪。这为以后的进一步统计分析和入侵检测等的开发打下了良好的基础。通过实验比较，与原Aglet系统相比效率下降少于5%。目前的移动Agent系统众多，如何将审计子系统扩展到其它的移动Agent系统是一项十分有意义的工作，也是我们下一步要进行的工作。

参考文献

1 Jansen W, Karygiannis T. NIST Special Piblications 800-19: Mobile Agent Security. National Institute of Standards and Technology: [Tech Rep: MD208999]. 1999
2 Jansen W. Countermeasures for mobile agent Security. Computer Communications, 2000, 23(10): 1667～1677
3 Karjoth G, Lange D B, Oshima M. A Security Model For Aglets. IEEE Internet Computing, Aug. 1997. 68～77
4 Wold Widre Web Consortium(W3C) Technical Reports and Publication. http://www.w3.org
5 Lange D, Oshima M. Programming and deploying java mobile agents with aglets. Massachusetts, USA: Addison-Wesley press, 1998
6 http://jdom.org/downloads/index.html