# 大规模数据集的增量式关联规则挖掘

张根香，陈海山
ZHANG Gen-xiang，CHEN Hai-shan

厦门大学 软件学院，福建 厦门 361005
Software School of Xiamen University，Xiamen，Fujian 361005，China
E-mail：zhanggenxiang2007@yahoo.cn

ZHANG Gen-xiang，CHEN Hai-shan.Incremental association rules mining for large data set.Computer Engineering and Applications，2009，45（29）：120-124.

**Abstract**：Business activity and engineering practice always accumulate large dataset with important information.But because of the dataset's largeness and frequent updating，if the Apriori based algorithm is applied to incremental association rules mining，it is not only inefficient，but also many redundant rules will be produced with low minimum support while some interesting rules will be lost with high minimum support，which leads to the algorithm's blunt perception to those rules.So，following genetic principle，and combining with natural immune involution theory and relative bionic mechanism，this paper proposes an IOGA（Immune Optimization based Genetic Algorithm）approach for incremental association rules mining.Experiment demonstrates the proposed method's effectiveness and presents its good performance in perceiving rules' subtle change，reducing redundant rules and finding interesting rules.
**Key words**：immune optimization；genetic algorithm；association rules；incremental mining

**摘 要**：商业活动和工程实践中通常会积累一些大规模的携带重要信息的数据，由于这种数据集经常有更新且数据量较大，在对它们进行增量式关联规则挖掘时，若采用基于传统的 Apriori 算法进行计算，一方面难以取得较好的效率；另一方面支持度设置过低会产生大量的冗余规则，设置过高则会把一些支持度不高但有用的规则过滤掉而导致算法对这些新规则感应迟钝。因此，借助遗传算法的相关机理，同时结合自然界的免疫进化理论及相关仿生机制，提出一种 IOGA（Immune Optimization based Genetic Algorithm，基于免疫优化的遗传算法）增量式关联规则挖掘方法。通过实验表明，该方法应用于大规模数据集的增量式关联规则挖掘时，可以及时地感知规则的变更并发现有用的规则，减少了冗余规则的产生，同时挖掘效率也有明显提高。
**关键词**：免疫优化；遗传算法；关联规则；增量式挖掘
**DOI**：10.3778/j.issn.1002-8331.2009.29.036 **文章编号**：1002-8331（2009）29-0120-05 **文献标识码**：A **中图分类号**：TP31

## 1 引言

随着社会信息化的不断加强，关联规则挖掘作为数据挖掘中的一个重要领域，已被数据库界广泛研究，并对于不同特点的数据集，在原有的基础上不断被优化、改进，使挖掘的效率不断提高，为企事业单位决策分析带来很高的参考价值，在商业、教育、科研等领域已经有许多成功的应用。

关联规则挖掘中已经有一些非常经典的算法，如 Agrawal 等人提出的基于频繁项集的 Apriori 算法[1]、J.Han 等提出的不产生候选项集的 FP-Growth 算法[2]。针对不同领域应用和出于其他方面因素的考虑，各种相应的改进算法亦不断出现，如应用于增量式关联规则挖掘的经典 FUP 算法[3]。另外，随着生物计算智能[4]的发展和模糊论[5-6]的应用，关联规则的人工智能挖掘方法也受到广泛关注和研究，如文献[7]所提出的动态数据集中关联规则挖掘的进化论方法、文献[8]提出的量化数据集中模糊关联规则挖掘、及文献[9]提出的多目标优化遗传算法的关联规则挖掘等。针对不同特点的数据集，这些算法各有所长。

对商业活动和工程实践中产生的大数据集，以及数据的动态更新特点，为高效地发现有趣的规则并感知规则的变化，提出基于免疫优化遗传算法（IOGA）的增量式关联规则挖掘方法，并将该算法和 FUP 算法应用于病毒文件关联规则挖掘，对两种实验结果进行对比分析。

## 2 关联规则的形式化定义

定义 1 进行关联规则挖掘的数据集记为 $D$，集合 $I=\{i_1,i_2,\cdots,i_m\}$ 是 $D$ 中全体项目组成的集合，$I$ 的任何子集 $X$ 称为 $D$ 中的项目集。当 $|X|=k$ 时，称集合 $X$ 为 $k$-项目集（$k$-Itemset）。设 $t_k$ 和 $X$ 分别为 $D$ 中的事务和项目集，如果 $X\subseteq t_k$，称事务 $t_k$ 包含项目集 $X$。

定义 2 进行关联规则挖掘的数据集 $D$ 表示为：$D=\{t_1,t_2,\cdots,t_n\}$，$t_k$（$k=1,2,\cdots,n$）称为事务或记录（Transactions），且 $t_k=\{i_{k1},i_{k2},\cdots,i_{kp}\}$，其中 $i_{km}$（$m=1,2,\cdots,p$）$\in I$，是第 $k$ 个事务或记录的第 $m$ 个属性值。

作者简介：张根香（1983-），女，硕士研究生，主研方向：数据库和数据挖掘；陈海山（1957-），男，教授，主研方向：数据库应用技术和信息化技术。
收稿日期：2009-04-07 修回日期：2009-06-25

**定义 3** 数据集 $D$ 中包含项目集 $X$ 的事务数称为项目集 $X$ 的支持数,记为 $count(X)$。项目集 $X$ 的支持度记为 $support(X)$:

$$support(X)=\frac{count(X)}{|D|}\times 100\% \tag{1}$$

其中$|D|$是数据集 $D$ 的事务数或记录数,若 $support(X)$ 不小于用户指定的最小支持度 $minsupport$,则称 $X$ 为频繁项目集,简称频繁集(或大项目集),否则称 $X$ 为非频繁项目集,简称非频繁集(或小项目集)。

**定义 4** 若 $X$、$Y$ 为项目集,且 $X\cap Y=\varnothing$,蕴涵式 $X=>Y$ 称为关联规则,$X$、$Y$ 分别称为关联规则 $X=>Y$ 的前提和结论。项目集 $X\cup Y$ 的支持度称为关联规则 $X=>Y$ 的支持度,记作 $support(X=>Y)$,则有:

$$support(X=>Y)=support(X\cup Y) \tag{2}$$

关联规则 $X=>Y$ 的置信度记作 $confidence(X=>Y)$,且有:

$$confidence(X=>Y)=\frac{support(X\cup Y)}{support(X)}\times 100\% \tag{3}$$

支持度和置信度是描述关联规则的两个重要概念,前者用于衡量关联规则在整个数据集中的统计重要性,后者用于衡量关联规则的可信程度。只有支持度和置信度均达到一定阈值的关联规则,才可能是用户感兴趣、有用的关联规则。通常,用户根据需要指定最小支持度阈值($minsupport$)和最小置信度阈值($minconfidence$)。

**定义 5** 若 $support(X=>Y)\geqslant minsupport$,且 $confidence(X=>Y)\geqslant minconfidence$,则称关联规则 $X=>Y$ 为强规则,否则称关联规则 $X=>Y$ 为弱规则。挖掘关联规则的任务就是要挖掘出 $D$ 中的所有强规则。

## 3 基于 Apriori 的 FUP 算法及分析

FUP 算法[10]是 D.W.Cheung 等人提出的增量式关联规则挖掘算法。其算法用到的标记符号如表 1 所示。

表 1 相关符号定义及说明

| 表示符号 | 含义说明 |
|---|---|
| $DB$ | 原始数据集 |
| $\|DB\|$ | $DB$ 中含有的记录数 |
| $db$ | 新增的数据集 |
| $\|db\|$ | $db$ 含有的记录数 |
| $L_{DB}^{k}$ | $DB$ 中的频繁 $k$ 项集 |
| $L_{db}^{k}$ | $db$ 中的频繁 $k$ 项集 |
| $C^{k}$ | $db\cup DB$ 中的候选 $k$ 项集 |
| $L^{k}$ | $db\cup DB$ 中的频繁 $k$ 项集 |
| $minsupport$ | 最小支持度阈值 |

### 3.1 FUP 算法思想

基于表 1 的符号定义,FUP 算法核心思想如下:

(1) 首先扫描 $db$,产生候选 1 项集 $C_{db}^{1}$,对此项集中的任意项 $X$:

①若 $X$ 在 $db$ 中为频繁一项集,在 $DB$ 中也为频繁一项集,则把 $X$ 放入 $L^{1}$;

②若 $X$ 在 $db$ 中为非频繁一项集,在 $DB$ 中为频繁一项集,计算 $X$ 的支持度

$$Support(X)=\frac{X.count^{DB}+X.count^{db}}{|DB|+|db|} \tag{4}$$

$X.count^{DB}$ 表示 $X$ 在 $DB$ 中出现的次数,$X.count^{db}$ 表示 $X$ 在 $db$ 中出现的次数。如果 $Support(X)\geqslant minsupport$,则把 $X$ 放入 $L^{1}$;

③若 $X$ 在 $db$ 中为频繁项集,但不在 $DB$ 的频繁项集中,扫描 $DB$ 得到 $X$ 在 $DB$ 中的支持数,再由式(4)计算 $X$ 的支持度 $Support(X)$。如果 $Support(X)\geqslant minsupport$,则把 $X$ 放入 $L^{1}$;

(2) 对于 $L_{DB}^{1}$ 与 $C_{db}^{1}$,令集合 $U=(L_{DB}^{1}-C_{db}^{1})$,若 $U$ 不为空,对于 $U$ 中的每个元素 $X$,按式(5)计算其支持度:

$$Support(X)=\frac{X.count^{DB}}{|DB|+|db|} \tag{5}$$

如果 $Support(X)\geqslant minsupport$,则把 $X$ 放入 $L^{1}$;

(3) 由得到的 $L^{k}$ 构造 $C^{k+1}$,对 $C^{k+1}$ 中的每个 $k+1$ 项集 $X$:

①若 $X$ 在 $DB$ 中是频繁项集,且在 $db$ 中也为频繁项集,把 $X$ 放入 $L^{k+1}$;

②若 $X$ 在 $DB$ 中是频繁项集,但在 $db$ 中为非频繁项集,则扫描 $db$ 计算 $X$ 在 $db$ 中的支持数,再根据式(4)计算出 $X$ 的支持度,如果 $Support(X)\geqslant minsupport$,把 $X$ 放入 $L^{k+1}$;

③若 $X$ 在 $DB$ 中为非频繁项集,但在 $db$ 中为频繁项集,扫描 $DB$ 计算 $X$ 在 $DB$ 中的支持数,再根据式(4)计算出 $X$ 的支持度,如果 $Support(X)\geqslant minsupport$,把 $X$ 放入 $L^{k+1}$;

④若 $X$ 在 $DB$ 中为非频繁项集,但在 $db$ 中为非频繁项集,则不把 $X$ 放入 $L^{k+1}$。

(4) 判断当前 $L^{k}$ 是否为空,是则停止,否则转入(3)。

### 3.2 FUP 算法分析

从 3.1 节可以看出,FUP 算法主要利用旧频繁集对在 $DB\cup db$ 中的候选项目集是否为频繁项集进行分析,从而在确定频繁项目集的过程中去掉不满足要求的候选项。因此,在新增加的数据集与原数据集相差不大的情况下,FUP 算法可以取得较高效率。但是对于一些大数据集而言,使用 FUP 算法对其进行增量式关联规则挖掘存在以下不足:

首先,FUP 算法仍然是基于频繁项集,对于大数据集,势必要耗费大量时间处理规模巨大的候选项目集,同时还要多次重扫描数据库 $DB$ 和 $db$ 来对候选集进行模式匹配。

其次,在现实生活中,人们往往希望能从增加的数据集中发现新的规则,这就意味着新增数据集与原数据集存在一定差异,那么此时使用 FUP 算法则难以取得较高的效率。

最后,FUP 算法主要基于支持度和置信度,支持度的设置对结果的影响很大,设置过低会产生大量冗余且没有意义的规则,但设置过高又可能会把一些有趣但支持度不高的规则过滤掉。日常生活中事物在不断变化,数据集在不断地更新,这就意味着某些规则可能在渐渐地改变。如果只是通过控制支持度和置信度很难及时感应到这种缓慢的规则变更,从而对某些重大的决策产生影响。

## 4 免疫优化遗传算法

遗传算法(Genetic Algorithms,GA),是模拟生物在自然环境中遗传和进化过程而形成的一种自适应全局优化概率搜索算法,其被用于解决各种复杂工程问题时,虽然已经取得了较为满意的结果,但也存在以下局限性[11]:

(1) 存在过早收敛现象,容易限入局部最优解;

(2) 算法的局部搜索能力较弱,往往与最优解失之交臂;

(3) 当涉及到大数据集时,算法收敛速度比较慢,需要花费较多的搜索时间;

(4) 算法的收敛性在一定程度上依赖于初始种群,当初始种群质量较差时,算法可能不收敛或收敛到非最优解。

免疫优化遗传算法在遗传算法的基础上引入动态免疫进化,借助于工程免疫计算(EIC)的仿生机理[4]:免疫识别、免疫记忆、免疫调节等来克服遗传算法的上述缺陷,增强寻优搜索能力。

### 4.1 基本概念

信息熵[4]:对于由 $N$ 个基因组成的 $M$ 个抗体,基因位 $j$ 的信息熵 $E_j(M)$ 用式(6)表示。

$$E_j(M)=\sum_{i=1}^{s}-p_{ij}\lg(p_{ij}) \tag{6}$$

其中 $s$ 为基因位上的字符集大小,若采用二进制编码则 $s=2$,$p_{ij}$ 为等位基因 $i$ 在基因座 $j$ 上出现的概率,即基因座 $j$ 上出现等位基因 $i$ 的次数与抗体个数 $M$ 之比。

平均信息熵[4]:$M$ 个抗体的平均信息熵 $E(M)$ 用式(7)计算。

$$E(M)=\frac{1}{N}\sum_{j=1}^{N}E_j(M) \tag{7}$$

亲和度[4]:指抗体和抗原之间的匹配程度及抗体之间的相似程度,前者定量地评估抗体与抗原之间的相互作用,后者则定量地评估抗体之间的相互作用。这里采用信息熵对亲和度进行度量,任意两个抗体 $\upsilon$ 和 $\omega$ 之间的亲和度 $A_{\upsilon\omega}$ 可用式(8)表示。

$$A_{\upsilon\omega}=\frac{1}{1+E(\omega)} \tag{8}$$

浓度[4]:用来度量抗体群体的多样性,对于任意抗体 $\upsilon$,其浓度 $C_\upsilon$ 可用式(9)表示。

$$C_\upsilon=\frac{N\upsilon}{M} \tag{9}$$

$N_\upsilon$ 表示与抗体 $\upsilon$ 的亲合度大于 $t$ 的抗体个数,$t$ 为最小亲合度阈值,取值为 $t\in[0.9,1]$,$M$ 为总的抗体数量。

属性影响度:数据集 $D$ 拥有 $m$ 个属性,这 $m$ 个属性组成属性集 $I=\{i_1,i_2,\cdots,i_m\}$。对于第 $k$ 个属性,其影响度记为 $E_k$。对于 $E_k$ 的计算,不同的编码可以灵活地采用不同的方式。若编码采用二进制,则 $E_k$ 定义为式(10)。

$$E_k=\frac{\sum_{i=1}^{|D|}T_{ik}}{|D|} \tag{10}$$

其中,$T_{ik}$ 为第 $i$ 条记录的第 $k$ 个属性,且 $T_{ik}\in\{0,1\}$,$|D|$为数据集中记录数。

数据集差异度:对于两个数据集 $D_1$ 与 $D_2$,其差异度 $dif(D_1,D_2)$ 定义为 $D_1$ 与 $D_2$ 每个属性影响度平方差的累积和,即如式(11)所示。

$$dif(D_1,D_2)=\sqrt{\frac{\sum_{i=1}^{m}(E_{1i}-E_{2i})^2}{m}} \tag{11}$$

其中 $E_{1i}$ 表示 $D_1$ 中第 $i$ 个属性影响度,$E_{2i}$ 表示 $D_2$ 中第 $i$ 个属性影响度,$m$ 为数据集中属性个数。因此,$dif(D_1,D_2)$ 越大则表明 $D_1$,$D_2$ 相差越大。

### 4.2 IOGA 基本思想

(1) 免疫系统基本相关原理[4]及应用

①免疫识别

免疫识别是免疫系统的核心功能,其本质是自我与非自我的区分,而这种区分主要根据抗体与抗原的亲和度实现。

②免疫记忆

免疫系统对第一次入侵的抗原产生抗体后对此抗原进行记忆,当下次该抗原或与该抗原结构上相似的抗原出现时,免疫系统的再次应答就表现得更有效、迅速。

③免疫调节

免疫调节主要指免疫系统可以通过本身的免疫平衡功能促进和抑制抗体的浓度,把免疫反应的强度控制在一定水平上。基于免疫调节的免疫选择利用抗体浓度、抗体与抗原的亲合度对系统进行调节,使亲合度高且浓度低的抗体得到促进并以较大的概率被选择进入下一代,同时抑制亲和度高且浓度较高的个体,从而在保留高亲和度个体的同时维持种群多样性,防止算法过早收敛以实现全局优化。

(2) 免疫优化遗传算法

免疫优化遗传算法分别利用免疫系统的以上基本原理,把种群中的个体作为抗体,将目标函数和约束条件作为抗原对待,采用信息熵来衡量个体之间的亲和度;通过个体适应度和浓度对个体生成进行加速或抑制,把优良个体转化为记忆细胞;个体的演化迭代则通过遗传算法中的交叉算子和变异算子来完成。

免疫优化遗传算法在遗传算法的算子上做了如下改进:

选择算子:

①初始种群 $P$ 的产生采取随机产生方式;

②基于适应度和浓度度量确定 $P$ 中 $n$ 个最佳个体集 $P_n$ 作为记忆细胞;

③对群体中的这 $n$ 个最佳个体进行复制,生成临时群体 $C$,其中复制的规模是适应度度量的单调递增函数。

变异算子:

变异对种群的多样性和局部搜索有影响,变异算子是一个保证搜索达到整个搜索域的次要措施。这里使用交换算子[4]进行变异,即在父代个体上随即产生两个交换点,把此两点的基因进行对换,同时变异概率反比于个体的适应度,从而生成一个成熟的群体 $C^*$;

交叉算子:

考虑到数据集的数据量问题,为了加快收敛速度,同时又防止过早局部收敛,这里采用 $n$ 点交叉方式,其中 $n$ 为正整数。

基因优化记忆:

在演化迭代过程中,对抗体中与决策相关的基因进行进一步的优化,即在计算个体适应度时计算出个体(抗体)非决策基因位与决策基因位相关度,如果相关度低于某一阈值,则下次迭代时该非决策基因位不参与计算。从而提高处理效率并减少冗余规则,提高规则兴趣度。

### 4.3 基于 IOGA 的增量式关联规则挖掘

数据集的更新会使关联规则发生变化,而关联规则的变化,可以体现在数据集的差异度上,即新增数据集与原始数据集差异度越大,新增数据集中含新规则的可能性越大[12]。因此,IOGA 在进行增量式关联规则挖掘时,先计算出原始数据集与新增数据集差异度;然后根据差异度大小,从原始数据集中取出相应比例的数据集,与新增数据集组成新的数据集;最后在新组成的数据集中使用免疫优化遗传算法计算。这样,既可以保留原支持度较高的强关联规则,又有利于发现新的关联规则。

基于 IOGA 的增量式关联规则挖掘主要包括下列步骤:

(1) 编码

基于处理的方便性,使用二进制编码。

(2) 适应度函数计算

对于种群中的个体 $X$,其适应度函数 $f(X)$ 由式(12)描述[11]。

$$f(X)=a\times\frac{sup(X)}{support}+b\times\frac{conf(X)}{confidence} \tag{12}$$

其中,$sup(X)$ 表示 $X$ 的支持度,$support$ 表示最小支持度阈值,$conf(X)$ 表示 $X$ 的置信度,$confidence$ 表示最小置信度阈值,$0<a<1$,$0<b<1$,且 $a+b=1$。

(3) 属性列减约

属性列减约是对于拥有较多属性的数据集,为加快处理速度,首先应该把非关键属性过滤掉,即:数据集 $D_d$ 中第 $k$ 个属性影响度为 $E_k$,若 $E_k<a\times support$(其中 $a$,$support$ 与式(12)中的相同),则此列不参与计算。

(4) 算法描述

①计算原始数据集 $D$ 中每个属性的影响度 $E_1,E_2,\cdots,E_m$(其中 $m$ 为属性减约后的属性个数);

②计算新增数据集 $d$ 中每个属性的影响度 $e_1,e_2,\cdots,e_m$;

③计算 $D$ 与 $d$ 的差异度 $dis1=dif(D,d)$;

④对于③中的 $dis1$:

当 $0<dis1<0.5$ 时,从 $D$ 中选择 $dis1\times|D|$ 条记录形成 $D'$;

当 $0.5\leqslant dis1<1$ 时,从 $D$ 中选择 $(1-dis1)\times|D|$ 条记录形成 $D'$;

⑤计算 $D'$ 与 $D$ 的差异度 $dis2=dif(D,D')$,如果 $dis2$ 大于指定的最小差异度阈值,转入④进行重新选择;

⑥由 $D'$ 与 $d$ 组成新的数据集 $D_d$;

⑦对 $D_d$ 中的属性列进行减约;

⑧从 $D_d$ 中随机选择个体产生初始种群 $G$;

⑨对于 $G$ 中的个体 $X$,按照式(12)计算其适应度 $f(X)$;按照式(9)计算其浓度;

⑩对于个体 $X$,

若适应度大于适应度阈值且浓度低于浓度阈值,则直接把其放入交配池;

若适应度大于适应度阈值且浓度高于浓度阈值,则只把此个体放入交配池,对其他使此个体浓度升高的个体进行标记,不把它们放入交配池,以此维持种群多样性;

⑪由交叉算子按照交叉概率对交配池中的个体进行 $n$ 点交叉操作;

⑫对剩下适应度低且非标记个体按变异概率进行变异;

⑬对个体基因进行优化记忆;

⑭判断有没有达到规定的进化代数,如果没有,则转入⑨;否则结束,输出关联规则及其支持度和置信度。

## 5 IOGA 算法与 FUP 算法在应用实例中的对比分析

算法所应用的实例是计算机病毒文件与其所调用的 API 函数之间的关联规则挖掘,即外部程序文件访问哪些 API 会导致计算机中病毒的可能性大。

### 5.1 数据说明

数据集相关描述如表 2 所示。

原始数据集中有 6 000 条记录,新增的数据集分别有 1 000,2 000,3 000,4 000 条记录。

使用 FUP 算法时,把每一条记录当一个事务进行处理。

使用 IOGA 算法时,把数据集转换为矩阵表示[3],矩阵每一行表示某文件所调用的 *API* 向量,用 *API* 整型向量来表示属性列,例如,对于第 $i$ 个文件,如果它调用了 71,260,89,55 这一 *API* 向量列表,则矩阵第 $i$ 行的 71,260,89,55 列值为 1,第 $i$ 行的其他列值为 0。数据集共有 *API* 列表向量 2 000 个,那么矩阵应含有 2 000 列(由于实际上每个文件所调用的 *API* 没有超过 100 个,因此为提高效率,计算前先对属性列进行简约)。

表 2 数据集属性说明

| 列名 | 功能 | 说明 |
|---|---|---|
| ID | 文件编号 | 顺序编号,含病毒和正常文件 |
| FileName | 文件名 | 样本的文件名 |
| FileSort | 文件类型 | 0=正常文件;1=病毒文件 |
| APISeq | 文件所调用的 API 列表(字符向量) | 形如"kernel32.dll,getsystemtime;user32.dll,invertrect;"这样的列表;由";"分割 APIs |
| IntVectorofAPI | 文件所调用的 API 列表(整形向量) | 形如"1,2,3"这样的列表;由","分割 APIs;每个 API 使用一个整型的 ID 来表示 |

### 5.2 实验结果及分析

通过与 FUP 算法的对比,从三个方面对 IOGA 算法进行分析。

(1) 算法的时间效率(支持度阈值为 0.26,置信度阈值为 0.80)

首先新增的 1 000 条数据与原始数据相差不大,即 $dif$=0.02,其他新增的数据与原始数据存在相同程度的差异,即新增数据为 2 000,3 000,4 000 时,其与原始数据的差异度均为:$dif$=0.23。

图 1 展示了 IOGA 与 FUP 在实现大数据集的增量式关联规则挖掘时的耗时。其中,横坐标表示增加的数据集的大小,纵坐标表示耗时。

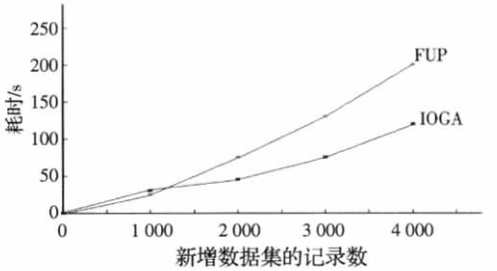


图 1 FUP 与 IOGA 算法应用于病毒文件的增量式关联规则挖掘耗时

从上图可以看出,当增加的数据为 1 000 时,由于其与原始数据相似度较高,因此采用 FUP 算法所耗费的时间比 IOGA 要少;但是与原始数据存在一定差异时,FUP 算法耗费时间比 IOGA 算法要多。可见,IOGA 算法在新增的数据与原始数据差异较大情况下进行增量式关联规则挖掘时,可以取得更好的效率。

(2) 所发现规则的兴趣度(支持度阈值为 0.26,置信度阈值为 0.80)

FUP 算法应用于原始数据集时产生的规则中有如下三条规则:

81,91,56→virus file(*support*:0.31,*confidence*:0.83)

81,91,79→virus file(*support*:0.30,*confidence*:0.84)

81,91,79,56→virus file(*support*:0.29,*confidence*:0.84)

而 IOGA 算法应用于原始数据集时产生的关联规则中包含前面三条规则中的第三条,前面两条均没有,即

81,91,79,56→virus file(*support*:0.28,*confidence*:0.83)

实际上这也是正确的,因为基于 IOGA 通过属性列简约及基因优化处理后,留下的规则中不会含有与决策属性列(即是否为病毒文件)无关的 *API* 向量。因此文件调用的 *AP* 向量

56、79 与此文件本身为病毒文件是关联的。而在 FUP 算法中的规则 81,91,56→virus file 及 81,91,79→virus file 是规则 81,91,79,56→virus file 的冗余,这也是由 FUP 算法本身的特性所决定的。

因此,由于数据集本身的特性(病毒文件所占的比例小),基于FUP 算法的支持度设置过低会导致挖掘效率下降,同时产生大量的冗余关联规则,从而难以判断到底哪些规则是用户真正感兴趣的规则;过高时又会把支持度过低但比较关键的关联规则过滤掉。但基于 IOGA 算法可以减少冗余规则的产生,让用户较容易地发现兴趣度高的规则。

(3) 对规则变更的感知能力(支持度阈值为 0.26,置信度阈值为 0.80)

IOGA 算法从原始数据集中挖掘出来的支持度,置信度较高的关联规则中包含如下三条,如表 3 所示。

表 3 原始关联规则中部分支持度,置信度较高规则

| 关联规则 | 支持度 | 置信度 |
|---|---|---|
| 91,81,79,56→virus file | 0.28 | 0.83 |
| 260,91,56→virus file | 0.29 | 0.86 |
| 264,174→virus file | 0.29 | 0.85 |

在新增的 1 000 条记录中有 420 个病毒文件主要调用了如下几类 *API* 向量(这几类 *API* 向量不含在原始关联规则中,且新增数据集与原始数据集差异度为 0.23):

第一类病毒文件:81,56,174,55

第二类病毒文件:79,63,54

第三类病毒文件:89,77

经增量式挖掘后,所产生的规则如表 4 所示。

表 4 增量式挖掘后,关联规则的支持度和置信度

| 关联规则 | 支持度 | 置信度 |
|---|---|---|
| 91,81,79,56→virus file | 0.24 | 0.79 |
| 260,91,56→virus file | 0.25 | 0.82 |
| 264,174→virus file | 0.27 | 0.81 |
| 81,56,174,55→virus file | 0.16 | 0.87 |
| 79,63,54→virus file | 0.19 | 0.85 |
| 89,77→virus file | 0.18 | 0.89 |

从上表可以看出,IOGA 在一定程度上既保留了原有的关联规则,又从增量数据集中挖掘出新的支持度不高但可反映问题的规则。这是一个感知变化,发现新规则,保留支持度较高的老规则的过程。而 FUP 算法由于单纯基于最小支持度阈值,最小置信度阈值而无法感知这种变化。

## 6 结束语

提出的 IOGA 算法,在遗传算法的基础上,引入了免疫系统相关动态仿生原理,如免疫识别、免疫记忆、免疫调节,以此加快遗传算法的处理速度和增强其全局寻优搜索能力,应用于增量式关联规则挖掘时能及时地挖掘出支持度不高但比较关键的新规则,得到的规则属性同时也具有较高的兴趣度。最后,把该算法应用于病毒文件的增量式关联规则挖掘,并与经典的 FUP 算法进行对比。结果表明,对于动态更新的大数据集中增量式关联规则挖掘,IOGA 算法在效率上和规则的质量上都可以取得较好的结果。

算法也存在需要改进的地方,如需要人为控制的参数过多,其中包括支持度阈值、置信度阈值、个体浓度阈值、种群大小、遗传代数、交叉概率、变异概率等,因此,算法在自适应性方面还有待加强。

## 参考文献:


[1] Agrawal R.Mining association rules between sets of items in large databases[C]//Proc of the ACM SIGMOD Intl Conf on Management of Data,Washington D C,United States,1993:207-216.

[2] Han J,Pei J,Yin Y.Mining frequent patterns without candidate generation[C]//Proc of the ACM SIGMOD Intl Conf on Management of Data,Dallas,Texas,United States,2000:53-87.

[3] 倪志伟,高雅卓,李伟东,等.基于矩阵的增量式关联规则挖掘算法[J].计算机工程与应用,2008,44(13):153-155.

[4] 肖人彬,曹鹏彬,刘勇.工程免疫计算[M].北京:科学出版社,2007.

[5] 张文宇,贾嵘.数据挖掘与粗糙集方法[M].西安:西安电子科技大学出版社,2007.

[6] 刘有才,刘增良.模糊专家系统原理与设计[M].北京:北京航空航天大学出版社,1995.

[7] Shenoy P D,Srinivasa K G,Venugopal K R.Evolutionary approach for mining association rules on dynamic database[C]//Proc of Pacific Asia Conf on Advances in Knowledge Discovery and Data Mining,Seoul,Korea,2003:325-336.

[8] de Graaf J M,Kosters W A,Witteman J J W.Interesting fuzzy association rules in quantitative databases[C]//Pro of the 5th European Conf on Principles of Data Mining and Knowledge Discovery,Darmstadt,Gremany,2001:140-151.

[9] Kaya M,Alhajj R.Muti-objective genetic algorithm based method for mining optimized fuzzy association rules[C]//Proc 5th Intl Conf on Intelligent Data Engineering and Automated Learing(IDEAI),Exeter,England,2004:758-764.

[10] Cheng D W,Han J,Ng V T.Maintenance of discovered association rules in large database:An incremental update technique[C]//Proc Intl Conf on Data Engineering,Washington D C,United States,1996.

[11] 莫宏伟.人工免疫系统原理与应用[M].哈尔滨:哈尔滨工业大学出版社,2002.

[12] Tiovonen H.Sampling large databases for association rules[C]//Proc 22nd Int Conf on Very Large Databases(VLDB),San Francisco,CA,USA,1996:134-145.


(上接 51 页)


[8] 李维诗,李江雄,柯映林.平面多边形方向及内外点判断的新方法[J].计算机辅助设计与图形学学报,2000,12(6):405-407.

[9] 陈正阳,王丽青,陈树强.基于单调链的判断点是否在多边形内的方法[J].计算机工程,2007,33(17):239-240.

[10] 黄杏元.地理信息系统概论[M].北京:高等教育出版社,1989:122-138.

[11] 王结臣.多边形拓扑关系构建的栅格算法[J].测绘学报,2002,31(3):249-254.

[12] de Berg M,van Kreveld M,Overmars M,et al.Computational geometry——Algorithms and applications[M].2ed.New York:Springer,1997:138-140.