

面向软件缺陷数据的负关联规则挖掘方法

万 琳, 范秋灵

(装甲兵工程学院 信息工程系, 北京 100072)

摘 要: 面向软件缺陷数据的关联规则挖掘的意义在于探寻软件缺陷数据之间的关联关系, 为软件开发、测试人员提供一定的指导作用. 负关联规则阐述的是不同项目之间的相互排斥关系, 具有非常重要的意义. 本文通过引入数据矩阵的概念及遗传算法的思想, 提出了一种新颖的负关联规则挖掘方法 GMNAR, 并通过实验验证了该方法的有效性和高效性.

关键词: 软件缺陷; 数据矩阵; 负关联规则; 遗传算法; 适应度函数

中图分类号: TN402

文献标识码: A

文章编号: 1000-7180(2015)04-0050-06

A Negative Association Rules Mining Method Oriented to Software Defect Data

WAN Lin, FAN Qiu-ling

(Academy of Armored Forces Engineering, Beijing 100072, China)

Abstract: Negative association rules describe the mutual exclusion of different items, which is also very important. In this thesis, a novel negative association rules mining method called GMNAR is proposed by introducing discernable matrixes concept and Genetic algorithm idea. And the experiment shows that it is effective and efficient.

Key words: software defect; data matrixes; genetic algorithm; negative association rules; fitness function

1 引言

近年来, 软件测试技术飞速发展, 各测评机构通过执行大量的测试任务, 已积累了海量高复杂性的软件缺陷数据. 挖掘软件缺陷数据的关联关系具有非常重要的意义. 例如, 软件测试人员可以根据软件缺陷数据之间的正关联关系, 判断哪些缺陷是同时发生的, 从而保证测试的完整性; 可以根据软件缺陷数据之间的负关联关系判断哪些相关的测试是可以忽略的, 从而缩减测试时间, 提高测试效率^[1].

传统的关联规则通常被称为正关联规则, 只能发现那些高频率、强关联的显式模式, 形如 $X \rightarrow Y$ 的蕴涵式. 然而, 实际的数据库中还有很多正关联规则方法无法发现的低频率、强关联的隐式模式, 即负关联规则^[2], 形如 $X \rightarrow \neg Y$, $\neg X \rightarrow Y$, $\neg X$

$\rightarrow \neg Y$ 的蕴涵式, 这种规则主要描述的是数据集中哪些项目不可能一起发生或者较少一起发生. 在软件测试过程中, 负关联规则有着非常重要的指导作用.

目前, 关于正关联规则的研究方法已经比较成熟, 但是, 关于负关联规则的研究^[3-7]却很少, 虽然给出了一些负关联规则的挖掘方法, 但是, 这些方法在进行频繁项集的搜索时, 还是采用传统 Apriori 算法, 存在频繁扫描数据库以及候选项集过多的问题. 本文基于频繁项集的思想, 引入遗传算法和矩阵方法, 提出了一种基于数据矩阵和遗传算法的负关联规则挖掘方法 (Negative Association Rules Mining Based on Discernable Matrixes and Genetic Algorithm, GMNAR), 仅需扫描一次数据库即可, 提高了频繁项集的发现效率, 并改进了遗传算法, 实现了有效规则的规模收敛.

2 基于数据矩阵和遗传算法的负关联规则挖掘方法

本文结合矩阵方法和遗传算法的优势,提出GMNAR方法,该方法的核心思想是首先对数据进行离散化抽象,然后扫描一次数据库,生成数据矩阵,并对矩阵进行简化,并根据简化矩阵生成染色体集合,采用改进的遗传算法进行操作,最终通过规则评价策略实现负关联规则的提取。

2.1 数据离散化

软件缺陷数据的类型是多种多样的,复杂多变的数据使得抽象知识的有效提取非常困难,很大程度上限制了数据的实用性。因此,在进行挖掘之前,首先要对数据进行离散化处理,转换为可以挖掘的数据类型。在此,引入粗糙集理论^[8]中的决策系统 S 表示待挖掘的事务集 T ,其数学表示形式为多元参数组合 $S = \langle U, A, V, f \rangle$ 。考虑到负关联规则描述的是事务数据库中不同项目 P 不可能同时发生或较少同时发生的情况,即规则前件 X 和规则后件 Y 之间的互斥关系,因此,根据其特点,将 S 重定义为: $S = \langle U, P, X, Y, V, f \rangle$,各参数含义如表1所示。

表1 S 各参数含义对照表

序号	参数名称	含义
1	U	待挖掘的事务
2	P	事务数据库中所包含的项目
3	X	关联规则的前件
4	Y	关联规则的后件
5	V	$V = \bigcup_{P \in A} V_P, V_P$ 为项目 P 的值域
6	f	$f: U \times (X \cup Y) \rightarrow V$

根据 V_P 的不同取值类型,可以采取不同的编码方式对事务集 T 中的数据进行离散化处理:

(1) 若 V_P 的取值是有限的、离散的,且 $0 < |V_P| < \infty$,则将这些取值分别编码为“1”,“2”,……;

(2) 若 V_P 的取值是连续的或者无限的,则采用等距离划分方法将这些取值划分为有限个区间,并分别编码为“1”,“2”,……;

(3) 若某项目取值为空或者某事务不包含该项目,则编码为“0”。

2.2 数据矩阵生成

矩阵方法打破了传统的逐层搜索思想,它将事务数据库映射为0-1矩阵,并通过对该矩阵的处理发现有价值的关联规则^[6],该方法只需要扫描一次

数据库即可完成关联规则的提取,减少了运算量。但是,0-1矩阵无法表示2.1节中所述的数据离散化取值,因此提出数据矩阵的构造方法如下。

设项集 $I = \{i_1, i_2, \dots, i_d\}$ 是 d 个项目的集合, $T = \{t_1, t_2, \dots, t_N\}$ 是 N 个事务的集合,则其对应的数据矩阵按照如下方式进行构造。

对矩阵 A 的每一个元素 $\{a_{ij}\}$ 进行如下定义:

$$a_{ij} = \begin{cases} p_{ij}, & i \neq m \wedge j \neq n \\ \text{num}(p_{ij} \neq 0), & i = m \vee j = n \\ 0, & i = m \wedge j = n \end{cases} \quad (1)$$

式中, m 为矩阵 A 的行数, $m = N + 1$, n 为矩阵 A 的列数, $n = d + 1$, $i = 1, 2, 3, \dots, m$, $j = 1, 2, 3, \dots, n$, p_{ij} 为离散化后的项目属性编码值,则每行表示一个事务,每列表示一个项目。 num 函数用于计算矩阵元素 a_{ij} 所在行或者所在列非零元素的个数。因此,矩阵的最后一行就表示了各个项目在数据库中出现的次数,最后一列就表示了每个事务包含的项目数,这一项是为后续的矩阵化简提供依据。

2.3 数据矩阵简化

为了减少系统运算量,提高挖掘效率,根据负关联规则的特点对数据矩阵的行、列分别进行化简。

(1) 列简化

频繁项集是指满足给定最小支持度阈值的项集,支持度是指项集在事务库中出现的频率。负关联规则挖掘的首要任务就是发现事务库中的频繁项集,然后从频繁项集中提取规则前件 X 与规则后件 Y 之间的负关系。

由频繁项集的性质“频繁项集的所有非空子集都必须是频繁的”可以得出以下推论:

推论1 若项集 X 是非频繁项集,对于任意的项集 $Z \supset X$,则项集 Z 也是非频繁项集。

根据推论1,可以通过删除掉非频繁项集对矩阵进行化简。

根据给定的支持度阈值 $\min \text{sup}$,计算频繁项集出现的最少次数 n ,公式为

$$n = \text{ceiling}(\min \text{sup} \times N) \quad (2)$$

式中, N 为事务总数, $\text{ceiling}(x)$ 函数用于计算不小于变量 x 的最小正整数。

将式(2)的计算结果 n 依次与矩阵 A 的最后一行元素 a_{mj} 进行比较,若 $a_{mj} < n$,则表示第 j 列项目的出现次数小于频繁项集的最少出现次数,因此其为非频繁项集,故删除矩阵的第 j 列。化简完成后删除用于计算项目出现次数的矩阵的最后一行。

(2) 行简化

负关联规则主要研究形如 $X \rightarrow Y, X \rightarrow \neg Y, X \rightarrow Y, X \rightarrow \neg Y$ 的关联关系, 根据其定义可知, 每一条负关联规则都必须包括规则前件 X 及规则后件 Y , 则规则的研究对象的个数至少为 2. 因此, 可以剔除仅包含一个项目的事务, 即删除矩阵最后一列中值为“1”的行. 化简完成后删除用于计算事务所含项目数的矩阵的最后一列.

2.4 遗传操作

本文利用遗传算法进行负关联规则的挖掘, 其搜索过程是首先产生一组初始规则, 然后利用各种进化准则对初始规则进行优化, 得到满足优化标准的规则. 由于遗传算法应用领域的不同, 本文根据负关联规则的特性重新设计了遗传算法的编码方式、初始种群生成、适应度函数及遗传算子.

2.4.1 染色体编码

由于前述建立的数据矩阵可以抽象表示缺陷数据的原始关联特点, 故遗传算法的初始种群即可由数据矩阵直接对应而来, 方法如下:

将矩阵的每个行向量编为一条染色体, 每个染色体代表一条规则, 每个基因对应一个项目属性, 0 表示该项目的属性空缺, “1, 2, ……”表示该项目属性的编码, 则种群规模为简化矩阵的行数 N .

2.4.2 遗传算子设计

遗传算法主要有三类遗传算子: 选择、交叉、变异, 根据缺陷数据特点及负关联规则提取要求, 设计算子的具体操作如下:

(1) 选择算子

选择算子的基本原则是将适应度高的优良个体遗传下一代, 淘汰适应度低的个体. 但是对于缺陷数据的关联规则而言, 两个适应度高的个体不一定会繁衍出优秀的个体, 甚至有可能会破坏原本适应性很高的个体, 因此, 在选择算子中, 采用最优生存法则, 即保留适应度值大于临界值的个体, 直接遗传到下一代, 从而避免最优个体被遗传操作破坏.

(2) 交叉算子

交叉算子可以避免陷入局部最优, 决定了算法的全局搜索能力. 本方法采用单点交叉算子, 具体执行过程为: 选取两个个体, 随机设定某一个基因后的位置为交叉点; 根据一定的交叉概率 p_c , 在各自的交叉点前或后互相交换双方的部分染色体, 从而产生两个新的个体.

以染色体 10050 和 02306 为例, 设定染色体的第 4 位为交叉点, 交换这两条染色体交叉点前的基因, 如图 1 所示.

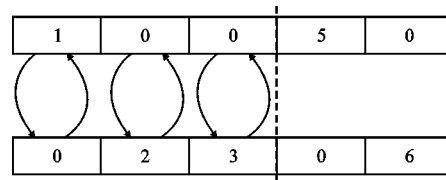


图 1 染色体前点交叉示意图

前点交叉操作后产生的新染色体为: 02350 和 10006.

(3) 变异算子

基因变异时, 若该基因所在位对应的取值只有 0 和 1, 那么变异策略一般为 0 变为 1, 1 变为 0. 由 2.1 节所述编码规则可知, 虽然本方法中的每个基因所对应的值不是只有 0 和 1, 但实际上其取值只可能为两种情况, 一种为 0, 一种为其对应的非 0 项目属性编码值, 因此设计变异算子为: “0”值变异为该基因位对应的项目属性编码值, 非 0 值变异为 0.

以染色体 10050 为例, 其变异过程如图 2 所示.

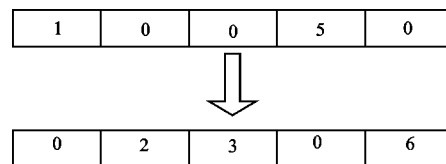


图 2 染色体变异示意图

2.4.3 适应度函数构造

适应度函数是遗传算法的关键因素, 其选取的好坏直接影响着最终的挖掘结果.

文献[9]中, Xiong 等人提出了相关度 φ 的概念, 定义如下:

$$\varphi = \frac{f_{11}f_{00} - f_{01}f_{10}}{\sqrt{f_{1+}f_{+1}f_{0+}f_{+0}}} \quad (3)$$

式中, f 表示项目出现在一个事务中的频度, 这一衡量标准可以有效反应两个项目之间的关联程度. 本方法中, 为了清晰表示负关联关系, 分别用 p_{XY} 、 $p_{\neg X \neg Y}$ 、 $p_{\neg XY}$ 、 $p_{X \neg Y}$ 代替 f_{11} 、 f_{00} 、 f_{01} 、 f_{10} , 其中, P 表示事务中项目发生的次数, N 为事务个数. 各参数含义如表 2 所示.

表 2 式(4) 参数含义表

	Y	$\neg Y$	Sum
X	p_{XY}	$p_{X \neg Y}$	p_{X+}
$\neg X$	$p_{\neg XY}$	$p_{\neg X \neg Y}$	$p_{\neg X+}$
Sum	p_{+Y}	$p_{+\neg Y}$	N

由于实际计算过程中, 规则前件 X 及规则后件 Y 发生的次数比不发生的次数更容易计算, 因此, 对

式(3)进行如下变形:

$$\begin{aligned}\varphi &= \frac{p_{XY}p_{\neg X\neg Y} - p_{\neg XY}p_{X\neg Y}}{\sqrt{p_{X+}p_{+Y}p_{\neg X+}p_{+-Y}}} = \frac{p_{XY}(N - p_{XY} - p_{\neg XY} - p_{X\neg Y}) - p_{\neg XY}p_{X\neg Y}}{\sqrt{p_{X+}p_{+Y}p_{\neg X+}p_{+-Y}}} \\ &= \frac{Np_{XY} - p_{XY}p_{XY} - p_{XY}p_{\neg XY} - p_{XY}p_{X\neg Y} - p_{\neg XY}p_{X\neg Y}}{\sqrt{p_{X+}p_{+Y}p_{\neg X+}p_{+-Y}}} = \frac{Np_{XY} - p_{XY}(p_{XY} + p_{X\neg Y}) - p_{\neg XY}(p_{XY} + p_{X\neg Y})}{\sqrt{p_{X+}p_{+Y}p_{\neg X+}p_{+-Y}}} \\ &= \frac{Np_{XY} - (p_{XY} + p_{X\neg Y})(p_{XY} + p_{\neg XY})}{\sqrt{p_{X+}p_{+Y}p_{\neg X+}p_{+-Y}}} = \frac{Np_{XY} - p_{X+}p_{+Y}}{\sqrt{p_{X+}p_{+Y}(N - p_{X+})(N - p_{+Y})}}\end{aligned}$$

则在本方法中,适应度函数定义如下:

$$\varphi_{XY} = \frac{Np_{XY} - p_{X+}p_{+Y}}{\sqrt{p_{X+}(N - p_{X+})p_{+Y}(N - p_{+Y})}} \quad (4)$$

Cohen 指出,对负关联规则而言,当 $\varphi_{XY} \leq -0.5$ 时,说明 X 和 Y 强相关;当 $-0.5 < \varphi_{XY} \leq -0.3$ 时,说明 X 和 Y 中等相关;当 $-0.3 < \varphi_{XY} \leq -0.1$ 时,说明 X 和 Y 弱相关;当 $\varphi_{XY} < -0.1$ 时,说明 X 和 Y 几乎不相关. 因此以 φ_{XY} 为评价遗传操作的适应度函数,选择 $\varphi_{XY} \leq -0.5$ 的优良个体进行子代生成.

例:对缺陷数据集

$$\begin{Bmatrix} 1 & 0 & 0 & 5 & 0 \\ 0 & 2 & 3 & 0 & 6 \\ 1 & 2 & 3 & 0 & 6 \\ 0 & 2 & 0 & 5 & 6 \\ 1 & 2 & 3 & 5 & 0 \end{Bmatrix}$$

中的染

色体 10050 来说, $X = \{1, 5\}$, $Y = \{6\}$, $P_{XY} = 0$, $P_{X+} = 2$, $P_{+Y} = 3$, 则其相关性

$$\varphi_{XY} = \frac{5 \times 0 - 2 \times 3}{\sqrt{2 \times (5 - 2) \times 3 \times (5 - 3)}} = -1,$$

可知该染色体负关联强相关,为优质染色体.

2.5 负关联规则提取

经过遗传算法多代搜索之后,获取了所有具有强关联关系的项目规则,但是对于负关联来说,有可能是 $X \rightarrow \neg Y$, $\neg X \rightarrow Y$, $\neg X \rightarrow \neg Y$ 的任何一种形式,而仅仅通过染色体无法分辨规则中哪些项目是“正”,哪些项目是“负”,因此,本方法的最后一部分工作就是对关联规则进行评价并提取出具体形式的负关联规则.

由负关联规则定义可知,当给定置信度阈值 minconf 和支持度阈值 minsup 时,若满足 $c(\neg X \rightarrow \neg Y) \geq \text{minconf}$ 且 $s(X \cup \neg Y) \geq \text{minsup}$,或 $c(X \rightarrow \neg Y) \geq \text{minconf}$,或 $c(\neg X \rightarrow Y) \geq \text{minconf}$,即可判定该规则为相应形式的负关联规则.

以染色体 10050 为例,设 $\text{minconf} = 0.5$, $\text{minsup} = 0.25$. 根据置信度式(5)、式(6)、式(7)和支持度式(8):

$$c(X \rightarrow \neg Y) = 1 - c(X \rightarrow Y) = 1 - s(X \cup$$

$$Y)/s(X) \quad (5)$$

$$c(\neg X \rightarrow Y) = \frac{s(Y) - s(X \cup Y)}{1 - s(X)} \quad (6)$$

$$c(\neg X \rightarrow \neg Y) = 1 - c(\neg X \rightarrow Y) \quad (7)$$

$$s(X) = \sigma(X)/N \quad (8)$$

式中, $\sigma(X)$ 为支持度计数,表示项目 X 在事务库中出现的次数; N 为事务总个数, $s(X)$ 为支持度, $c(X \rightarrow Y)$ 为置信度.

计算得:

$$\textcircled{1} c(15 \rightarrow 6) = 1 - (0/5)/(3/5) = 1 > \text{minconf} = 0.5, \text{则可提取负关联规则 } 15 \rightarrow \neg 6.$$

$$\textcircled{2} c(15 \rightarrow 6) = (3/5 - 0/5)/(1 - 2/5) = 1 > \text{minconf} = 0.5, \text{则可提取负关联规则 } \neg 15 \rightarrow 6.$$

$$\textcircled{3} c(15 \rightarrow \neg 6) = 1 - c(15 \rightarrow 6) = 1 - 1 = 0 < \text{minconf} = 0.5, \text{则不能提取负关联规则 } \neg 15 \rightarrow \neg 6.$$

3 应用实例

为了更清晰地阐述 GMNAR 方法,在此从 XX 装备信息管理软件测试过程中产生的软件缺陷数据中提取部分缺陷数据作为实验数据,采用本方法挖掘不同的测试项之间的关联关系,实验数据如表 3 所示.

表 3 软件缺陷与测试项关系表

缺陷标识	相关测试项
AZH-V1.0-WBG-001	测试项 1、测试项 4、测试项 5
AZH-V1.0-WBG-002	测试项 2、测试项 3、测试项 6
AZH-V1.0-WBG-003	测试项 1、测试项 2、测试项 3、测试项 6
AZH-V1.0-WBG-004	测试项 2、测试项 5、测试项 6
AZH-V1.0-WBG-005	测试项 1、测试项 2、测试项 3、测试项 5

对“测试项”这一项目属性进行编码,则 $V_p = \{\text{测试项 1, 测试项 2, 测试项 3, 测试项 4, 测试项 5, 测试项 6}\}$, 分别编码为 $\{1, 2, 3, 4, 5, 6\}$, 设:规则前件 $X = \{\text{测试项 1, 测试项 2, 测试项 3, 测试项 4, 测$

试项 5}, 规则后件 $Y = \{\text{测试项 6}\}$, 则离散化后的缺陷数据如表 4 所示.

表 4 离散化数据表

编号	X				Y	
	1	2	3	4	5	6
1	1	0	0	4	5	0
2	0	2	3	0	0	6
3	1	2	3	0	0	6
4	0	2	0	0	5	6
5	1	2	3	0	5	0

依据 2.2 节所述构造其数据矩阵为

$$A = \begin{pmatrix} 1 & 0 & 0 & 4 & 5 & 0 & 3 \\ 0 & 2 & 3 & 0 & 0 & 6 & 3 \\ 1 & 2 & 3 & 0 & 0 & 6 & 4 \\ 0 & 2 & 0 & 0 & 5 & 6 & 3 \\ 1 & 2 & 3 & 0 & 5 & 0 & 4 \\ 3 & 4 & 3 & 1 & 3 & 3 & 0 \end{pmatrix} \quad (9)$$

设支持度阈值 $\text{minsup} = 0.25$, 由式(2)可得 $n = \text{ceiling}(0.25 \times 5) = 2$, 由于 $a_{64} = 1 < 2$, 因此, 删除矩阵的第 4 列以及矩阵的最后一行和最后一列, 经过简化的矩阵为

$$B = \begin{pmatrix} 1 & 0 & 0 & 5 & 0 \\ 0 & 2 & 3 & 0 & 6 \\ 1 & 2 & 3 & 0 & 6 \\ 0 & 2 & 0 & 5 & 6 \\ 1 & 2 & 3 & 5 & 0 \end{pmatrix} \quad (10)$$

依据矩阵 B 得染色体集合 $\{10050, 02306, 12306, 02056, 12350\}$, 设置信度阈值 $\text{minconf} = 0.5$, 支持度阈值 $\text{minsup} = 0.25$, 经过遗传操作之后, 获得的规则集合为 $\{00050, 02306, 02006, 10000, 10050\}$.

最终依据 2.5 节所述方法提取出负关联规则集:

$$\text{NegativeAR} = \{15 \rightarrow \neg 6, \neg 15 \rightarrow 6, \neg 1 \rightarrow 6, 1 \rightarrow \neg 6, \neg 5 \rightarrow 6, 5 \rightarrow \neg 6\}$$

通过调整置信度阈值 minconf 和支持度阈值 minsup 可以确定所提取负关联规则所表示关联关系的发生概率, 如果设定 $\text{minconf} = 1$, 则提取出的负关联关系一定发生. 故此负关联规则集可说明以下结果:

(1) 由于 $c(15 \rightarrow \neg 6) = 1, c(1 \rightarrow \neg 6) = 2/3, c(5 \rightarrow \neg 6) = 2/3$, 所以可得: 如果测试项 1 和测试项 5 同时出现某类缺陷, 那么测试项 6 一定不会出

现该类缺陷. 如果测试项 1 或测试项 5 出现某类缺陷, 那么测试项 6 出现缺陷的可能性很大. 因此, 在进行软件测试时, 如果发现测试项 1 和测试项 5 同时出现了某类缺陷, 那么无需再进行测试项 6 的测试.

(2) 由于 $c(\neg 15 \rightarrow 6) = 1, c(\neg 1 \rightarrow 6) = 1, c(\neg 5 \rightarrow 6) = 1$, 所以可得: 当测试项 1 或者测试项 5 不出现某类缺陷时, 测试项 6 必定产生该类缺陷, 应加强执行测试项 6.

4 实验及结果分析

为了验证本文提出的 GMNAR 方法的有效性与可行性, 利用 Matlab2012a 模拟实现了文献[3]提出的 Apriori 负关联规则挖掘方法与本文提出的 GMNAR 方法, 并对其实验结果进行了比较.

为了对比实验结果的可信性, 本文仿真实验数据使用 IBM 公司的实验数据生成器 (www.ibm.com/us/quest/syndata.html/#assocSyndata) 生成, 该数据生成器是目前国际上比较通用的数据挖掘实验数据库. 为了对比 GMNAR 方法在不同规模数据库中的性能, 本实验选用 4 个基础实验数据库: T16, I5, D0, 5k, T130, I12, D6k, T1400, I130, D56k, T15000, I1400, D570k, 仿真实验结果如图 3、4 所示.

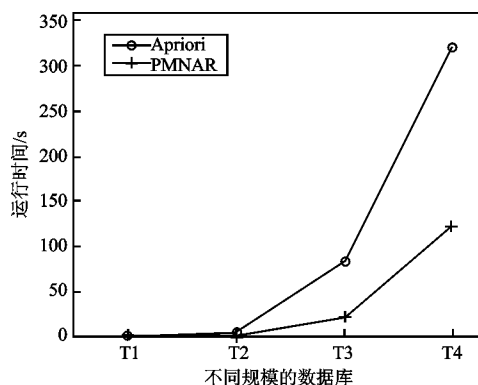


图 3 两种方法在不同规模数据库的执行时间比较

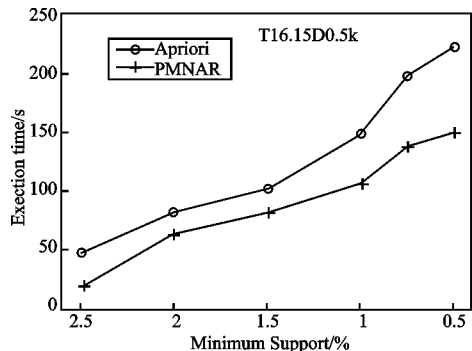


图 4 两种方法在数据库 T1 中挖掘的负关联规则数量比较

实验结果对比分析如下:

(1)图3表明,对于小规模数据库来说,两种方法的执行时间基本相同,而随着数据库规模的增大,GMNAR方法的执行时间明显少于Apriori方法的执行时间。

GMNAR方法仅需扫描一次数据库即可获得数据矩阵,一般情况下遗传算法发现关联规则所需的时间会远远少于获取数据矩阵所消耗的时间,因此,GMNAR方法的时间主要消耗在数据矩阵的构建过程中;而Apriori方法需要多次扫描数据库。当数据库规模比较小时,两者运行时间差距不明显;随着数据规模的增大,扫描数据库的时间远远多于生成数据矩阵的时间。因此,GMNAR方法比Apriori方法运行速率高一些。

(2)图4表明,GMNAR方法可以实现负关联规则的提取,且挖掘的负关联规则数量有所缩减,一定程度上解决了包含负项目的频繁项集爆炸问题。

通过引入相关度 φ_{xy} 这一评价依据,根据规则的支持度值和置信度值,实现了负关联规则的有效提取,并且通过调整置信度阈值的取值范围,可以剔除部分冗余信息,一定程度上缩小负关联规则的数量规模,尽快发现最有效的规则集。

5 结束语

本文提出了一种面向软件缺陷数据的负关联规则挖掘方法—GMNAR方法,并以XXX装备信息管理软件部分缺陷数据为例,进行了详细阐述^[10]。该方法引入了数据矩阵的思想以及改进的遗传算法,实现了负关联规则的提取,减少了数据库扫描次数,缩减了负关联规则数量规模,一定程度上提高了挖掘的效率。

参考文献:

[1] Agrawal R. Mining association rules between sets of i-

tems in large databases[C]// Proceeding of the 1993 ACM SIGMOD International Conference on Management of Data. Washington, ACM Press, 1993: 207-216.

[2] Brin S. Beyond market baskets: Generalizing association rules to correlations[C]// Proceeding of the 1997 ACM SIGMOD International Conference on Management of Data. USA: Arizona, ACM Press, 1997: 265-276.

[3] 董祥军,王淑静,宋瀚涛,等.负关联规则的研究[J].北京理工大学学报,2004,24(11):978-981.

[4] 朱红蕾,徐志刚,李明.负关联规则挖掘算法研究[J].微电子学与计算机,2010,27(5):167-169.

[5] 马占欣,陆玉昌.负关联规则挖掘中的频繁项集爆炸问题[J].清华大学学报:自然科学版,2007,47(7):1212-1215.

[6] 张卫国,高贤强,武晓宏.基于支持度-置信度框架的负关联规则研究[J].微电子学与计算机,2009,26(4):102-105.

[7] 朱玉全,陈耿,杨鹤标.正负关联规则挖掘算法研究[J].计算机科学,2006,3(3):188-190.

[8] 王国胤. Rough 集理论与知识获取[M]. 西安:西安交通大学出版社,2001.

[9] Xiong H. Exploiting a support-based upper bound of pearson's correlation coefficient for efficiently identifying strongly correlated pairs[C]// The 10th Intl. Conf. on Knowledge Discovery and Data Mining. Seattle, WA, 2004 8:334-343.

[10] Cohen J. Statistical power analysis for the behavioral sciences[M]. 2nd ed. New Jersey, Lawrence Erlbaum, 1988.

作者简介:

万琳女,(1974-),博士,副教授.研究方向为军事信息系统、软件工程、软件测试。

范秋灵(通讯作者)女,(1989-),硕士研究生.研究方向为军事信息系统、软件工程、软件测试。E-mail: fanqiuling1989@163.com.

(上接第49页)

作者简介:

刘杰男,(1988-),硕士研究生.研究方向为视频图像处理及应用。E-mail: 466914350@qq.com.

谢正祥男,(1938-),教授,博士生导师.研究方向为生物医

学信号和图像处理。

吕霞付男,(1966-),教授,研究生导师.研究方向为视频图像处理及应用。

陈学渊男,(1988-),硕士研究生.研究方向为视频图像处理及应用。