

基于 RFID 的围术期术后康复行为检测

魏大顺¹, 张德林², 董瑞国²

(1. 徐州医科大学医学信息学院, 江苏 徐州 221000;

2. 徐州医科大学附属医院, 江苏 徐州 221000)

摘要: 针对术后患者康复结果是否达到出院标准的评判具有主观性, 为此引入 RFID 技术, 提出基于 RFID 的术后康复行为检测方法; 由于患者在围术期过程中由 RFID 获得的活动路径信息具有高冗余、高维度、高无用等特性, 因此在挖掘之前, 首先计算事物数据库中各属性的条件信息熵进行属性约减; 为提高挖掘效率, 减少无用挖掘过程, 在传统的 FP-growth 基础上进行验证式挖掘, 并根据挖掘结果检测患者术后的活动行为是否达到标准。实验结果表明该算法的运行效率与挖掘质量较传统的 FP-Growth 算法有着较为显著的提高, 且挖掘结果与临床指标相符, 具有一定的临床可行性和有效性。

关键词: RFID; 行为检测; 数据挖掘

Perioperative Period Postoperative Rehabilitation Behavior Detection Based on RFID

Wei Dashun¹, Zhang Delin², Dong Ruiguo²

(1. School of Medical Information, Xuzhou Medical University, Xuzhou 221000, China;

2. Affiliated Hospital of Xuzhou Medical University, Xuzhou 221000, China)

Abstract: It is subjective to judge whether the rehabilitation results of postoperative patients meet the discharge standards. To this end, RFID technology is introduced, and a postoperative rehabilitation behavior detection method based on RFID is proposed. Because the patient's active path information obtained by RFID during the perioperative period has characteristics such as high redundancy, high dimensions, and high uselessness, before mining, first calculate the condition information entropy of each attribute in the transaction database to reduce the attributes; Improve the mining efficiency and reduce the useless mining process. Based on the traditional FP-growth, perform verification mining, and check whether the patient's post-operative activities meet the standard based on the mining results. The experimental results show that the operation efficiency and mining quality of the algorithm are significantly improved compared with the traditional FP-Growth algorithm, and the mining results are consistent with clinical indicators, which has certain clinical feasibility and effectiveness.

Keywords: RFID; behavior detection; data mining

0 引言

由于医疗报销等问题, 在围术期过程中医生要控制、缩短平均住院日, 术后如果没有较严重并发症, 一般要求病人尽快出院; 但是患者又担心康复不彻底; 而国家又没有相关的出院技术标准, 因此在患者各项生理指标正常的情况下, 我们提出一种以病人术后活动指标作为一种检测是否可以出院的方法。通过患者腕带标签, 局域网内的接收器检测这些标签, 并向位置软体传递即时位置状态等资料, 具体如图 1 所示。FP-growth 作为一种有效的关联规则挖掘算法已经被用于许多领域, 许多学者也针对不同的应用场景或者算法本身进行了许多改进。如基于节点表的

FP-growth 算法改进^[1]、基于 FP-tree 的快速构建算法^[2]和无项头的 FP-growth^[3]算法。但在医疗应用场景下这些算法都很难发挥他们的优势, 特别是在处理 RFID 所产生的事务数据, 这主要是由于其高维度性、高无用性、高冗余性等特性导致^[4]。例如基于节点表的 FP-growth 算法通过新增的节点表和频繁模式树两者中共通的项, 从而进行频繁项集挖掘, 然而当数据维度越高, 两者中共通的项则越多, 挖掘效率因此也就越低下; 基于 FP-growth 的快速构建算法通过动态调整各项的顺序从而减少数据库访问频度, 但是医疗数据具有高无用性以及高冗余性, 因此若利用此方法处理医疗数据将会消耗大量的内存; 无项头的 FP-growth 算法虽然可以精简生成的关联规则, 但是生成的关联规则中依然还会存在大量的冗余规则。在实际应用场景中, 医疗人员更多的是希望在他们已有的专业判断下, 进一步从某些关键因素进行确认患者是否达到出院标准, 比如患者在术后的行动时间与行动距离是否存在关联性、活动范围与术后时间是否存在关联性。若使用原始 FP-growth 或者目前已有的改进算法去解决这一需求, 不仅无

收稿日期: 2020-03-11; 修回日期: 2020-03-27。

基金项目: 江苏省卫生厅信息化项目(X201405)。

作者简介: 魏大顺(1993-), 男, 江苏徐州人, 研究生在读, 主要从事数据挖掘、围术期质量管理方向的研究。

通讯作者: 董瑞国(1962-), 男, 江苏徐州人, 硕士生导师, 教授, 主要从事围术期质量管理方向的研究。

法针对性的进行关联挖掘，而且会找出许多无用的关联规则^[5]，而这必然会严重影响挖掘效率，降低挖掘质量。本文针对此种应用场景，以胸外科手术数据为作为实验数据，提出了一种改进的 FP-growth 算法。该算法首先通过计算数据集中各属性的条件信息熵^[6]进行约减；创建 FP-Tree，根据判断分支中是否含有决策项以及项重要度大小进行相应的剪枝。该算法从整体上缩减了事务数据库的规模，对 FP-tree 进行了横向和纵向剪枝压缩，从而简化了最终频繁项集的挖掘，提高了挖掘效率，降低了算法运行所需内存与 CPU。实验结果表明，本文改进方法在可接受的精度范围内，挖掘效率高于已有主要挖掘算法。

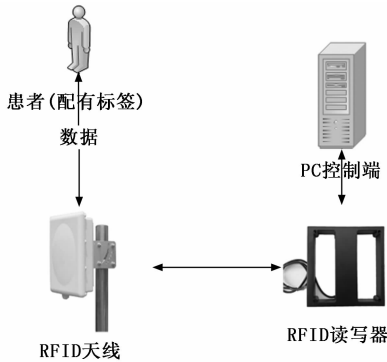


图 1 患者活动 RFID 射频识别

1 基础知识

本节介绍本文所需的基本知识，主要是 RFID 以及 FP-growth 算法。

1.1 基本定义

定义 1: RFID, 无线射频识别即射频识别技术 (radio frequency identification, RFID), 是自动识别技术的一种, 通过无线射频方式进行非接触双向数据通信, 利用无线射频方式对记录媒体 (电子标签或射频卡) 进行读写, 从而达到识别目标和数据交换的目的。

定义 2: FP-growth, FP-Growth 算法采取如下分治策略: 将提供频繁项集的数据库压缩到一棵频繁模式树 (FP-tree), 但仍保留项集关联信息。

定义 3: 支持度, 确定规则可以用于给定数据集的频繁程度, 公式如下:

$$s(X \rightarrow Y) = \frac{\sigma(X \cup Y)}{N} \quad (1)$$

其中: N 是事务的总数, X, Y 为对应的属性。

1.2 FP-growth 算法流程

输入: 事务数据库 (transaction database, TD); 最小支持度 (minimum support, MinSup)^[2]。

输出: 频繁模式的完全集。

1) 对 TD 中的事务进行逐条遍历, 统计每项支持度, 并且以集合 F 表示, 并且按照支持度大小重新排列, 最终得到频繁项表 L 。

2) 再一次对数据库 TD 进行扫描, 按照 L 中的顺序对

TD 进行重新排序, 得到 TD' ;

3) 创建频繁挖掘树 FP-tree, 其中根节点为 root, 默认为空, 将步骤 2) 得到的 TD 中的数据按照顺序插入 FP-tree 中, 若数据节点第一次在 FP-tree 中出现, 则新建节点, 并计数 1, 若待插入的节点已经存在于 FP-tree 中, 则原计数加 1, 同时令项头表为步骤 1) 中得到的 L , 且链头为项头, 连接内容相同的项;

4) 对项集 L 中的项逆序挖掘频繁项集。

2 对 FP-growth 算法的改进

FP-growth 算法在处理事务库时无法对无用属性的筛选以及对特定关联的挖掘, 从而导致挖掘速度缓慢以及影响挖掘质量。因此, 本文提出了一种改进的验证式挖掘算法—IEFP-growth (Information Entropy FP-growth) 算法。通过对 FP-growth 挖掘前的属性约减和 FP-tree 创建过程中的剪枝来进行优化。该算法首先通过计算数据集中各属性的条件信息熵^[7]进行约减; 在原算法扫描数据库生成频繁一项集的基础上, 创建 FP-Tree, 根据判断分支中是否含有决策项以及项重要度 IT (Item importance) 大小进行相应的剪枝, 若不含决策项则进行剪枝, 若含决策项且所在分支中含有项重要度大于重要度阈值的项则保留, 否则进行剪枝。该算法首先从整体上缩减了事务数据库的规模, 其次对 FP-tree 进行了横向压缩, 进而提高了整体挖掘速度与质量。

剪枝依据性质:

- 1) 支持度大的项并不一定和决策项具有关联;
- 2) 支持度大于 MinSup 且重要度大于或等于 IT 的项一定和决策项具有一定程度的关联。

2.1 IEFP-growth 的算法流程

输入: TD ; $MinSup$; IT 。

输出: 含决策属性的频繁模式集

1) 对 TD —事务数据库进行属性约减得到 TD' ;

(1) 计算 TD 中除决策属性^[8]外每个属性相对于决策属性的条件信息熵, 例如 D 相对于 C 的:

$$H(D | C) = \sum_{i=1}^{|U/C|} (p(X_i) \sum_{j=1}^{|U/C|} p(Y_j | X_i) * \log(p(| X_i))) \quad (2)$$

其中: D 是决策属性, C 是条件属性, U 为论域, X_i, Y_j 分别是某一属性的属性集合。

(2) 计算条件属性集 C 中相对于决策属性集 D 的主属性 C_r , 并令 $P=C_r, B=C-C_r$;

(3) 计算条件信息熵 $H(D | P)$, 转步骤 (6);

(4) 对 $i=1 \dots n, b_i \in B$ 中的每个属性计算条件熵 $H(D | P \cup b_i)$, 求:

$$SGF(b_i, P, D) = H(D | P) - H(D | P \cup b_i) \quad (3)$$

得到属性 b_i 相对于决策属性的重要度 SGF ;

(5) 找出属性 b_i , 该 b_i 可以让 SGF 取得最大值, 然后在 P 的后面追加 b_i , 并且去掉使 SGF 为 0 的 b_i ;

(6) 如果 $H(D|P) = H(D|C)$, 则转步骤 (7), 否则转步骤 (4);

(7) 依次判断 p 中每个属性是否可以去掉, 判断顺序是从 p 尾开始, p 头结束。假如 a 在 $core_c(D)$ 中, 则从 a 往前都不可约, 算法结束; 否则 a 是可约减的, 于是把 a 从 A 中删除。

(8) 得到处理之后的 TD'

2) 第一次扫描 TD' , 除去小于 $MinSup$ 的项, 得到频繁一项集 L ;

3) 再次遍历 TD' , 重新排列 TD' , 排序规则以 L 中各项的支持度为度量进行降序排序, 得到 TD'' ;

4) 建立 FP-tree 根节点, 值为 null, 将 TD'' 中的事务依次插入, 若分支中不含决策项则进行剪枝, 若含决策项且所在分支中含有项重要度 - SGF 大于重要度阈值 IT 的项则保留, 否则进行剪枝; 若节点不存在, 则新建并计数并置 1, 若节点存在, 则计数加 1;

5) 对创建好的 FP-tree 进行频繁项集的挖掘。

(1) 获取头指针表中每一个元素的条件模式基;

(2) 按照步骤 1) 至步骤 4) 的建树步骤, 对上一步骤获得条件模式基建立条件 FP 树;

(3) 重复以上步骤, 获取条件模式基并构建条件 FP 树, 直至无法构建条件 FP 树为止;

2.2 算法详解

(1) 以表 1 简易患者活动事务数据库 - TD 为例。

表 1 患者活动事务数据库 TD

活动时间/h	活动距离/m	心跳	术后时间/天
0.5	5	正常	5
0.5	5	慢	5
0.5	5	慢	5
1	5	正常	10
1	10	快	10
1	5	快	5
1	10	慢	5
1	10	正常	10

为使得 RFID 数据信息满足挖掘要求, 将活动时间、活动距离等数据以范围表示。

计算各属性重要度 SGF :

$$H(D|C) = 0.5$$

$$H(D|P) = 1$$

$$SGF(\text{活动时间}, P, D) = H(D|) - H(D|\text{半小时},) = 0.049$$

$$SGF(\text{心跳}, P, D) = H(D|) - H(D|\text{快},) = 0$$

$$SGF(\text{活动距离}, P, D) = H(D|) - H(D|\text{5 米以内},) = 0.311\ 278$$

根据属性重要度 SGF 进行属性约减得到表 2。

(2) 再次遍历 TD' , 重新排列 TD' , 排序规则以 L 中各项的支持度为度量进行降序排序, 得到 TD'' ;

(3) 在 TD'' 的基础上生成 FP-tree, 并且对根节点初始化, 其中根节点名称为 root, 默认值为空。按照频繁树生

表 2 约减后数据库 TD'

活动时间/h	活动距离/m	术后时间/天
0.5	5	5
0.5	5	5
0.5	5	5
1	5	10
1	10	10
1	5	5
1	10	5
1	5	10

成规则, 将 TD'' 中事务逐条压缩到频繁树中。若需验证活动时间与活动距离是否存在关联规则, 则待插入的分支中不含有“活动时间”属性, 则舍弃该分支, 若待插入分支中含有“活动时间”属性且所在分支中含有项重要度大于重要度阈值 IT 的项则保留, 否则进行剪枝; 若节点不存在, 则新建并计数并置 1, 若节点已经在 FP-tree 中出现, 则相应的计数加 1。

(4) 利用项头表和频繁模式树递归挖掘频繁项集。

3 实验与结果

本节通过将 IEFP-growth 算法与当前效果较好的基于节点表的 FP-growth 算法以及原始 FP-growth 算法进行实验对比, 对比分析本文所提出的 IEFP 算法的挖掘效果; 最后将挖掘结果与其对应的生理指标进行对比, 证明所提方法具有一定的临床可行性。

3.1 实验环境

软件环境: pycharm 编辑器; 硬件环境: Intel (R) Core (TM) i5 - 7300HQ CPU @ 2.50 GHz, 内存为 8 G, 64 位 Window10 操作系统;

3.2 实验一

随机选取 10 000 条实验数据, 事务数据事务中项的频繁项集支持度大多在 0.04% 到 0.18% 之间, 分别测试 FP-growth、基于节点表优化的 FP-growth 和 IEFP-growth 算法在不同支持度下的消耗时间以及经过差分隐私后的 IEFP-tree 的挖掘时间对比。详细数据内容见表 3。

表 3 不同算法运行时间对比图

支持度 (%)	FP-growth/ms	Node table FP-growth/ms	IEFP-growth/s
0.18	23	25	12
0.16	28	32	17
0.14	35	37	21
0.12	47	45	32
0.10	55	59	37
0.08	71	62	43
0.06	83	75	51
0.04	99	86	56

由表 3 可知, 第一列是最小支持度, 第二列、第三列

和第四列分别是 FP-growth 算法、基于节点表优化的 FP-growth 算法和 IEFP-growth 算法执行时间，并且 IEFP-growth 在各支持度下的消耗时间都低于 FP-growth 原始算法和基于节点表的 FP-growth 算法所需要的时间。

图 2 绘制了 FP-growth、基于节点表的 FP-growth 算法和 IEFP-growth 在最小支持度变化下各自挖掘时间的变化及对比情况。

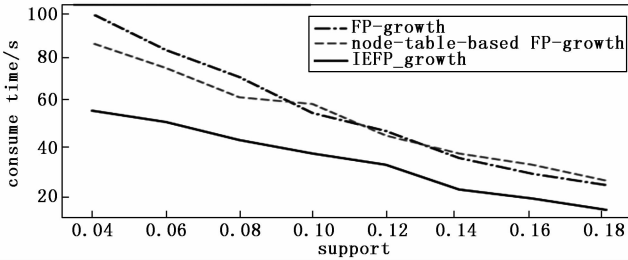


图 2 不同算法在不同支持度下的执行时间对比图

由图 2 可以看出，随着支持度的缩小，IEFP-growth 算法相比于 FP-growth 算法以及基于节点表的 FP-growth 算法在相同支持度下所需时间更少，且随着支持度的减小，所需时间增长速率缓慢，且支持度范围在 0.04~0.08 之间，IEFP-growth 消耗时间的增长速率明显低于其他两种算法，这表明支持度越小，IEFP-growth 比原始 FP-Growth、基于节点表的 FP-growth 挖掘能力越显著。

3.3 实验二

在不同事务量的情况下分别测试 FP-growth、基于节点表优化的 node table-FP-growth 和 IEFP-growth 算法，统计在不同事务量下的内存消耗情况^[9]。详细数据见表 4。

表 4 不同算法所需内存对比

事务量	FP-growth/MB	node table-FP-growth/MB	IEFP-growth/MB
5 000	63	51	34
10 000	162	147	57
15 000	326	286	119
20 000	437	372	181

由表 4 可知，第一列是事务量，第二列、第三列和第四列分别是 FP-growth 算法、基于节点表的 FP-growth 算法以及 IEFP-growth 算法在不同事务量情况下内存利用率占比，并且 IEFP-growth 算法在各事务量情况下，特别是事务量越大的情况下，所需要的内存相对于其他两种算法都相对较少。

图 3 绘制了 FP-growth 算法、基于节点表的 FP-growth 算法、以及经过差分隐私保护的 IEFP-tree 算法在不同事务量变化下各自内存利用率的柱状对比图。

从图 3 可以看出，在相同事务量下，IEFP-growth 算法所需内存比 FP-growth 算法和基于节点表的 FP-growth 算法所需内存少，且事务量越大，所需内存差距越明显。

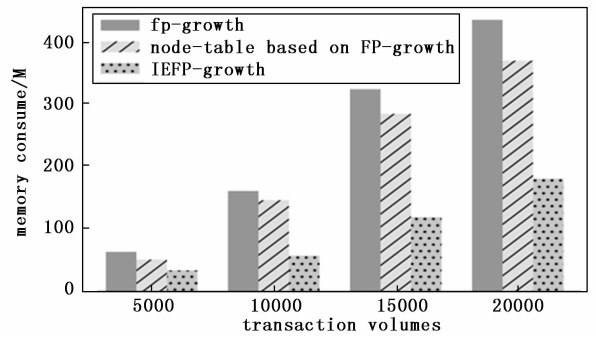


图 3 不同算法在不同事务量下所需内存图

3.4 实验三

在不同事务量的情况下进行 FP-growth、基于节点表优化的 FP-growth、IEFP-growth 挖掘，统计在不同事务量情况下的 CPU 使用情况^[10]。详细数据见表 5。

表 5 不同算法所需 CPU 对比

事务量	FP-growth /%	node table-FP-growth/%	IEFP-growth/%
5 000	34	21	12
10 000	42	52	18
15 000	73	61	27
20 000	81	77	37

如表 5 所示，第一列是事务量，第二列、第三列和第四列分别是 FP-growth、基于节点表的 FP-growth 和 IEFP-growth 算法 CPU 利用率占比，并且 IEFP-growth 算法在各事务量情况下，特别是事务量越大的情况下，所需要的 CPU 相对于其他两种算法都相对较少。

图 4 绘制了 FP-growth、IEFP-growth、基于节点表优化的 FP-growth 以及经过差分隐私保护的 IEFP-tree 算法在不同事务量变化下各自 CPU 利用率的变化。

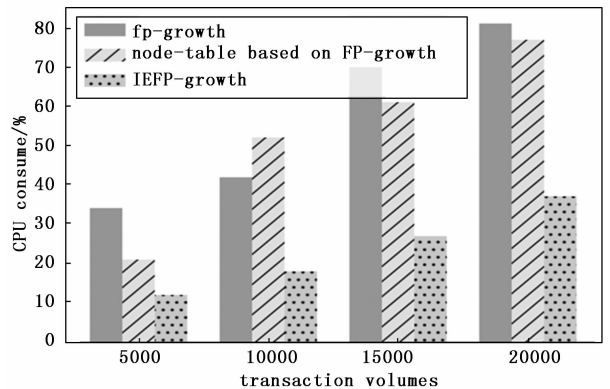


图 4 不同算法 CPU 变化图

从图 4 可以看出，IEFP-growth 算法所需 CPU 比 FP-growth 算法和基于节点表的 FP-growth 算法所需 CPU 少，且事务量越大，所需 CPU 差距越明显。

3.5 实验四

实验选取 10 000 条事务数据，分别利用 FP-growth 算

法、基于节点表的 FP-growth 算法以及 IEFP-growth 算法进行关联规则挖掘, 对挖掘得到的频繁模式集分析其中含有决策属性—活动时间的构成比^[11], 如图 5 所示。

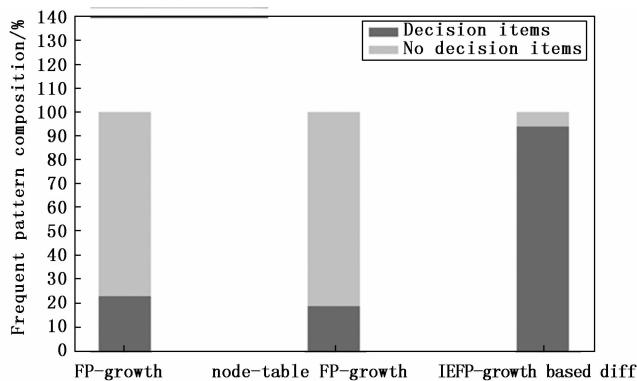


图 5 不同算法挖掘结果中频繁项集构成比图

从图 5 可以发现, IEFP-growth 挖掘得到的频繁项集中决策项占据了绝大部分, 其中少部分不是决策属性主要是在剪枝过程中由于重要度以及支持度等因素被减去, 也就是说假设的关联关系中存在少部分非关联规则, 而由 FP-growth 和基于节点表的 FP-growth 算法挖掘得到的频繁项集中与决策属性有关的频繁项只占了很小的一部分。因此 IEFP-growth 算法可以很好地进行针对性的挖掘, 对临床具有一定的价值。

3.6 实验五

为了证明本文所提方法具有一定的临床指导意义, 从电子病历系统中提取 100 位患者阑尾炎患者术后不同时间内白细胞数量平均值, 如图 6 所示, 以及术后的活动距离平均值, 如图 7 所示。

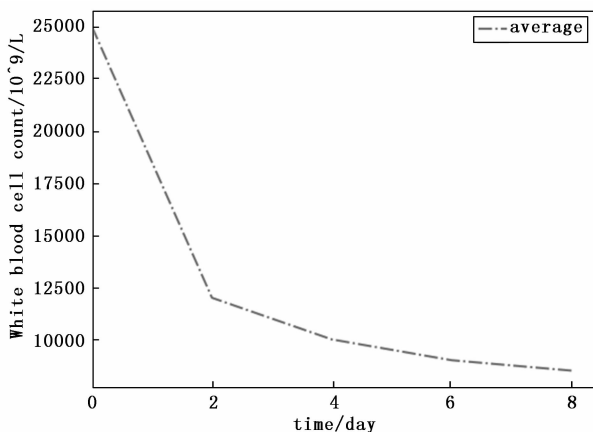


图 6 术后白细胞变化

从图 6 我们可以看出在天数为 0 时即未进行手术情况下, 白细胞的数量达到了 25 000/L 以上, 此时病人阑尾已经发生坏死、穿孔等症状, 但术后随着时间的推移, 白细胞数量逐渐下降, 在第 6 天之后白细胞数量已经趋近于正常水平, 且在第八天之后患者办理了出院, 因此没有了数据; 从图 7 我们可以看出, 患者的活动距离随着术后时间

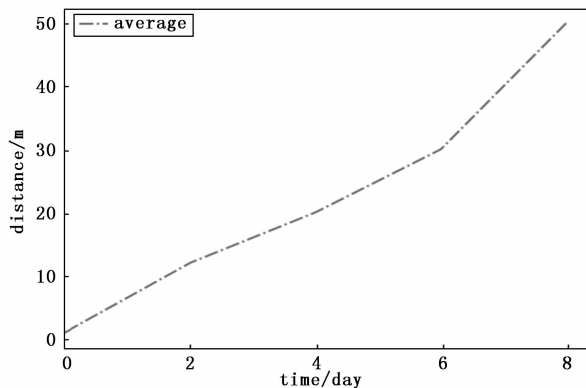


图 7 术后患者活动距离平均值

的推移逐步增加, 结合图 6 与图 7 我们可以发现患者的活动距离与白细胞数量在术后一定时间内具有一定的关联性, 而这也符合挖掘结果, 因此利用 RFID 获取的患者活动距离来挖掘数据间的关系并判断是否达康复行为具有一定的临床指导意义。

4 结语

本文通过挖掘 RFID 采集的患者活动信息数据, 为医疗人员提供患者术后出院参考。

针对 FP-growth 算法在进行频繁模式挖掘过程中, RFID 采集的患者数据无关属性过多以及无法进行目的的挖掘而引起挖掘效率和质量低下的问题, 提出一种验证式挖掘算法——IEFP-growth 算法。在对数据库属性约减的基础上, 根据判断 FP-tree 分支中是否含有决策项以及项重要度大小进行相应的横向和纵向剪枝压缩, 简化最终频繁项集的挖掘, 提高了挖掘效率和质量, 降低了算法运行所需内存和 CPU。实验结果表明了 IEFP-growth 算法在时间效率、内存与 CPU 利用率以及挖掘质量上较传统 FP-growth 算法都有所改进; 并且通过对比分析患者术后白细胞与活动距离的变化趋势, 发现白细胞变化与活动距离在一定时间内具有一定的关联, 证明该算法在临床检测上具有可行性与实用性^[12-18]。

参考文献:

- [1] Juan Pan, Jiajun Ding, Xiongxiang He. Research and application of pca-bp-bagging model in medical assistant diagnosis [A]. 2017 36th Chinese Control Conference (CCC) [C]. 2017.
- [2] 王建国, 袁伟. 基于节点表的 FP-Growth 算法改进 [J]. 计算机工程与设计, 2018 (1): 140-145.
- [3] Liu S, Wu Z H. Fast construction algorithm of out-of-core based scene acceleration data structure [J]. Journal of Sichuan University, 2016.
- [4] 余彪, 刘守全. 基于 FP-growth 算法的改进关联规则挖掘算法 [J]. 计算机与网络, 2017 (14): 69-72.
- [5] 凌绪雄, 王社国, 李洋. 无项头表的 FP-Growth 算法 [J]. 计算机应用, 2011, 31 (5): 1391-1394.

[6] 陈 媛, 杨 栋. 基于信息熵的属性约简算法及应用 [J]. 重庆理工大学学报, 2013, 27 (1): 42 - 46.

[7] 黄国顺, 曾凡智, 文 翰. 代数约简的条件信息熵表示及其高效约简算法 [J]. 计算机科学, 2014, 41 (7): 236 - 241.

[8] 王 林, 胡雪婷. 基于多决策属性的社会网络核心节点识别 [J]. 计算机工程, 2018, 44 (1): 206 - 210.

[9] 刘典型. 多虚拟机下基于内存缓存的动态块迁移算法 [J]. 计算机应用与软件, 2015 (3): 11 - 15.

[10] 王余伟, 等. 实时操作系统 CPU 使用率监测的软件容错研究 [J]. 计算机工程与科学, 2018, 284 (8): 1337 - 1343.

[11] 李 铁, 赵晓伟, 田 坚. 老年 2 型糖尿病患者并发胃肠神经病变的影响因素 [J]. 中国老年学杂志, 2013, 33 (4): 762 - 764.

[12] 张 岳, 王洪国, 邵增珍, 等. 基于先验位运算的频繁项集挖掘 [J]. 计算机应用研究, 2013, 30 (9): 2610 - 2612.

[13] 尹治华, 张大鹏, 谭 明, 等. 一种改进的基于 FP-Tree 的高效挖掘最大频繁项目集算法 [J]. 济南大学学报 (自然科学版), 2017 (2): 111 - 117.

[14] 宋 薇, 张晓民, 郭东恩. 基于前缀路径图的频繁闭项集挖掘算法 [J]. 计算机与数字工程, 2017, 45 (11): 2290 - 2294.

[15] 陈静锐, 王美君, 陈选菊, 等. 一种新的递归函数的递与归的分析方法 [J]. 保山学院学报, 2014, 33 (2): 62 - 67.

[16] 程 凡. 基于排序学习的信息检索模型研究 [D]. 合肥: 中国科学技术大学, 2012.

[17] 崔一辉, 宋 伟, 彭智勇. 基于差分隐私的多源数据关联规则挖掘方法 [J]. 计算机科学, 45 (6): 42 - 46.

[18] 曾 雷. 关联规则挖掘中 Apriori 算法的研究 [D]. 重庆: 重庆交通大学, 2016.

.....
(上接第 64 页)

$$\delta_{R200} = \frac{3 \times 0.94}{200 \times 7.508} \times 100\% = 0.188\%$$

从试验数据中计算出加载均值和卸载均值的最大差值为 2 mV, 以此得出推力架在 200 mV 范围内的迟滞性误差 δ_{H200} 为:

$$\delta_{H200} = \frac{2}{200 \times 7.508} \times 100\% = 0.133\%$$

在系统的精度计算时, 总的误差应包括系统误差和随机误差。推力架的系统误差可通过计算加载和卸载均值与标定直线间的最大差值来表示, 由此计算出系统误差为 1.84 mV。随机误差可取 3 倍的平均标准偏差 σ 。因此, 系统在 200 mN 的量程范围内的精度为:

$$A = \frac{1.84 + 3 \times 0.94}{200 \times 7.508} \times 100\% = 0.31\%$$

通过以上分析, 本文研究的推力架在 mN 级推力标定过程中具有较好的性能指标, 系统的非线性、重复性和迟滞性均在 0.2% 以内, 且系统的总精度也能够达到 0.3%, 说明系统标定获得的直线方程能较好地反映出系统输出与推力之间的关系, 能够对 mN 级微推力进行准确的测量, 满足工程中对测量的需求。

5 结束语

本文针对小型航天器的 mN 级微推力测量需求, 研制了性能稳定、功能完善的数字化试验样机。微推力架采用扭摆式测量原理, 由激光位移传感器和电磁力平衡组件组成测量和控制环节, 在 LabVIEW 软件编程中利用 PID 算法实现闭环控制。并采用一组精密砝码作为等效推力, 通过加载和卸载过程, 对微推力测量系统进行了标定和性能验证, 最终实现量程为 10~200 mN 范围的测量。经分析, 系统在量程范围内的标定精度为 0.331%, 且非线性、重复性和迟滞性等指标都满足 mN 级微推力测量的要求, 能够为未来的工程应用提供可靠地技术指导。

参考文献:

[1] 王 宇, 尤 政, 王广宇, 等. 一种多脉冲微推力的测量方法 [J]. 航空学报, 2009, 30 (12): 2257 - 2262.

[2] Youngner D W, Lu S T, Choueiri E, et al. MEMS Megapixel Micro - Thruster Arrays for Small Satellite Stationkeeping [R]. AIAA SSC00-X-2, 2000.

[3] 岑继文, 徐进良. 一种微推力测量的简化处理方法 [J]. 航空学报, 2008, 29 (2): 297 - 303.

[4] 洪延姬, 周伟静, 王广宇. 微推力测量方法及关键问题分析 [J]. 航空学报, 2013, 34 (10): 2287 - 2299.

[5] Merkwowitz S M, Ahmad A, Hyde T T, et al. LISA Propulsion Module Separation Study [J]. Classical and Quantum Gravity, 2005, 22 (10): S413 - S419.

[6] 王大鹏, 金 星, 周伟静. 一种用于微推力测量的电磁力校准装置设计与实现 [J]. 兵工自动化, 2017, 36 (9): 68 - 72.

[7] 刘向阳, 范宁军, 李科杰. 微型推进器推力测试的现状与发展趋势 [J]. 测控技术, 2004, 23 (5): 12 - 16.

[8] 高 航. 固体微推力器及其测试台的研究与改进 [D]. 北京: 北京理工大学, 2016.

[9] 刘万龙, 朱昊伟, 孙树江, 等. 国内微推力测试技术发展现状 [J]. 火箭推进, 2015, 41 (5): 7 - 11.

[10] 汤海滨, 刘 畅, 向 民, 等. 微推力全弹性测量装置 [J]. 推进技术, 2007, 28 (6): 703 - 706.

[11] 钱 婧, 汪希平, 田 丰, 等. 功率放大器 OPA544 在主动磁悬浮控制系统中的应用 [J]. 电子技术应用, 2011, 37 (6): 68 - 70.

[12] 杨 智, 朱海峰, 黄以华. PID 控制器设计与参数整定方法综述 [J]. 化工自动化及仪表, 2005, 32 (5): 1 - 7.

[13] 宋 铭. LabVIEW 编程详解 [M]. 北京: 电子工业出版社, 2017.

[14] 王海霞, 颜桂定, 李宝辉, 等. 直线电机运动控制系统的软件设计与实现 [J]. 电子测量与仪器学报, 2013, 27 (3): 264 - 269.