

面向卫星遥测数据流的最小稀有模式挖掘方法

周忠玉 皮德常

(南京航空航天大学计算机科学与技术学院 南京 211106)

摘 要 模式挖掘是应用于卫星智能监控服务中的一项重要技术,当前频繁模式挖掘的使用率要远远高于稀有模式挖掘,然而对于卫星遥测数据流来说,频繁模式挖掘在安全监测和故障预防等方面所取得的成效不如稀有模式挖掘.因为频繁模式挖掘无法从卫星的遥测数据中揭示卫星可能存在的潜在故障.卫星遥测是持续不断进行的,所以其数据流存在数据量大、传输速度快和数据重复性高的特点.如果采用一般的稀有模式挖掘方法来挖掘卫星数据流,尽管其速度比频繁模式挖掘快,但总体上仍然较慢,不能满足卫星实时监测的需要.针对上述问题,本文提出一种可快速找出卫星遥测数据流中隐藏信息的最小稀有模式挖掘方法,它具有如下优点:(1)无需卫星领域知识;(2)引用滑动窗口技术并将主观参数(窗口尺寸)客观化,使得算法能够实时地处理数据流;(3)通过仅挖掘最小稀有模式方式来提高算法的挖掘效率;(4)该算法使用双向遍历技术提高算法的运行速度.从某在轨卫星的遥测数据流中选取10个关键特征参数进行算法验证.实验结果表明,本文所提算法能有效地从卫星遥测数据流中挖掘出全部的最小稀有模式,并且其挖掘速度比现有的方法快.

关键词 最小稀有模式;卫星;遥测数据流;自顶向下;自底向上;双向遍历;模式挖掘;数据挖掘

中图法分类号 TP311 **DOI号** 10.11897/SP.J.1016.2019.01351

Minimal Rare Pattern Mining Method For Satellite Telemetry Data Streams

ZHOU Zhong-Yu PI De-Chang

(College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing 211106)

Abstract The pattern mining is an important technology applied to satellite intelligent monitoring services. At present, the usage ration of frequent pattern mining is much higher than that of the rare pattern mining. However, as far as the research whose target object is satellite telemetry data stream, the results of the frequent pattern mining is not as effective as those of the rare pattern mining in terms of the security monitoring, fault prevention and so on. This is due to the fact that the frequent pattern mining can not reveal the potential faults and hidden dangers that may exist in the components of satellites from satellite telemetry data stream, and it cannot also provide preventive measures. Since telemetry for satellites is continuously carried out, the data stream on satellite telemetry has the following characteristics: in the first place, it includes a large amount of data; in another, its transmission speed is fast, and what is more, its data repetition rate is high. If the common rare pattern mining methods are used to mine satellite telemetry data stream, although its mining speed is faster than the mining speed of the frequent pattern mining, it is still slow on the whole and cannot meet the needs of real-time monitoring for satellite. What is more, the general rare pattern mining mainly takes static data as the object of the research. In order to find out all the rare patterns, the strategy of generating all candidates and then pruning is usually adopted, but which may be leading to the problem of combinatorial explosion and the

speed of algorithm is slower. With the development of the big data and the technologies of cloud platform, traditional storage devices have been unable to keep up with the speed of data transmission. Therefore, if static data is still used as the object of the research, the storage device may easily lose data when storing the data. In order to solve the above problems, this paper proposes a minimal rare pattern mining method that can quickly discover the hidden information in the satellite telemetry data stream. It has the following advantages: First, the algorithm proposed in this paper, that is, the minimal rare pattern mining algorithm, does not require knowledge in the satellite field. Second, this paper references the sliding window technology and objectifies the subjective parameters (that is, window size) so that the algorithm proposed in this paper can process satellite telemetry data stream in real time. Third, the algorithm improves the mining efficiency by only mining the minimal rare pattern, instead of all the rare patterns. Fourth, the algorithm uses bidirectional traversal techniques to improve the speed of the algorithm. Ten key features are selected from the satellite telemetry data stream of some orbiting satellite to verify the proposed method in this paper. The experimental results indicate that the algorithm proposed in this paper can effectively mine all the minimal rare patterns from satellite telemetry data stream, and the speed of the proposed algorithm is faster than that of the existing methods.

Keywords minimal rare pattern; satellite; telemetry data stream; top-down; bottom-up; bidirectional traversal; pattern mining; data mining

1 引 言

中国自主研发的北斗卫星系统是一款服务于全球的卫星导航和定位系统,在全球范围内提供定位精确度为 10 m,测速精确度为 0.2 m/s,授时精度为 10 ns 的高精度服务.它主要由 35 颗卫星组成,目前广泛应用于电信、交通、救灾、测绘等诸多领域^[1].它在全球范围内持续地传输大量的实时监测数据,但由于太空环境复杂多变,卫星的健康安全时刻面临着威胁挑战^[2-3].因此,为了确保北斗卫星的正常运行,必须提供一种智能服务来自动监控卫星的运行状态,并实时地处理卫星遥测数据.

卫星遥测数据是以数据流的形式呈现的.所谓数据流是指只能以预先规定的顺序被读取一次的数据序列^[4].起初,卫星产生的海量数据流大多用二三级存储设备进行存储,但随着内存技术和处理器技术的进步,使得二三级存储设备的数据处理速度越来越跟不上数据传输的速度^[5].如果数据处理的速度比数据传输的速度慢,就会导致大量的信息流失,这对于卫星的健康检测来说是不允许的.因此,有必要对卫星遥测数据流进行实时处理.但是遥测数据流的处理是十分困难的,原因如下:

(1) 卫星遥测数据流的传输速度非常快,需要

在短时间内处理大量的数据^[6].因此除了对硬件的性能要求外,处理算法必须是高效的、健壮的.

(2) 遥测数据流在卫星生命周期内是无限传输的,无法进行一次性处理,也无法得到最终的处理结果^[7].所以必须对数据流进行实时分段处理,从而得出部分处理结果.

(3) 卫星遥测数据流的数据维度较大^[6],导致其数据容量远超二三级存储设备的容量.所以为了保存完整的未经处理的数据信息,避免数据丢失,需要进行实时的数据处理,以便及时地得到部分的处理结果.在必要时,可以放弃已经处理过的无用数据,以便接纳新的遥测数据.

现如今,在轨卫星主要采用自主运行为主,地面遥控为辅的方式运行,卫星发送的遥测数据是地面人员监测卫星设备状态的唯一依据^[7].卫星由多个分系统组成,各个分系统根据需要在一些关键部件上设置遥测点,智能监控系统按一定周期对各个遥测点上的数值进行采集,包括角度、电流值、电压值、压力值、温度值、状态字、标识符、同步字等.由于采集到的卫星遥测数据是一种含有噪声和缺失值的高维大数据流,所以传统的人工服务和一般的数据分析工具无法满足卫星遥测数据的分析要求.因此,卫星监测系统需要一种智能服务来代替人工服务.它不仅能够实时地监控在轨卫星的参数状态,而且还

能提供一种高效的遥测数据分析方法,用于实时地挖掘卫星大数据流,而模式挖掘方法恰能满足这种要求^[8]. 遥测数据的模式挖掘对于卫星的自动异常检测和故障诊断都非常重要,是一种具有广阔应用前景的智能监控手段^[7].

目前智能服务的研究热点主要是关于移动互联网、物联网和传感器等背景下的研究^[9-12]. 在卫星传感器和数据挖掘的背景下,针对卫星智能监控方面的研究绝大多数都致力于研究频繁模式挖掘,而稀有模式挖掘研究甚少^[7,13-15]. 然而在卫星态势安全预防等领域,稀有模式挖掘更有研究意义与实用价值,因为它可以从少部分信息中揭示出频繁模式挖掘无法发现的隐藏信息^[16],而这些信息通常应用于卫星安全监测. 一种做法是将挖掘出的最小稀有模式作为异常发现方法的输入数据,之后通过计算异常因子来识别卫星存在的潜在异常. 例如:2015 年, Hemalatha 等人^[17]在乳腺癌异常检测的背景下,提出了一种通过挖掘稀有模式的方式来检测异常值的方法,并且通过实验证明其效率和准确率都高于基于频繁模式的异常检测方法. 通过挖掘稀有模式应用于卫星异常检测的基本思想是,那些构成最小稀有模式生成源的观察结果可能是异常值,因为它与所考虑的其它观察结果相比具有“区别特征值”^[17]. 因此,稀有模式挖掘作为一种重要的模式挖掘方法,可以智能地发现卫星遥测数据中的稀有模式,进而服务于在轨卫星智能监控. 尽管稀有模式挖掘方法还不如频繁模式成熟,但仍然存在一些稀有模式挖掘算法,它们的应用意义及效率要远高于现有的频繁模式挖掘算法,例如 Hyper-Linked RPM^[18]算法.

稀有模式一般是指支持度小于指定最小支持度阈值的模式,但随着算法要求的变化,也有不同的定义. 例如, Bhatt 和 Patel^[19]提出的以最大约束为限制条件的稀有模式,当且仅当该模式的支持度小于指定的最小频繁支持度阈值并且大于或者等于指定的最小稀有支持度阈值时,才可以称为稀有模式. 与大多数以静态数据为研究对象的模式挖掘相比,以动态数据流为对象的模式挖掘要困难许多,这是因为数据流包含一系列连续到达的、无限的和无序的数字序列^[6],并且这些数字序列通常只能读取一次^[4]. 为了解决这个难题, Chang 和 Lee^[20]提出了采用滑动窗口机制挖掘数据流的方法,得到了不错的实验结果. 其后的许多研究者都采用滑动窗口技术来处理数据流,例如 Lei^[16]和 Zhang 等人^[21]. 通过滑动窗口可以对数据流进行实时地处理,从而克服

处理数据流对象的困难. 卫星遥测数据流面临的问题和挑战总结如下:

(1) 在缺乏专业领域知识的前提下,如何从遥测数据流中挖掘出最小稀有模式. 已有的数据流处理方法通常会涉及很多专业领域知识^[22-23],从而增加了卫星遥测数据流挖掘的难度,也阻碍了研究成果的应用.

(2) 如何发现少部分数据中的隐藏信息. 稀有模式挖掘不如频繁模式挖掘成熟,我们可以借鉴的算法很少. 目前,大多数算法都是从大部分数据中挖掘出频繁模式,找出相应的规律,并将少部分数据直接舍弃. 然而,卫星在绝大多数情况下都是正常运行的,那些少部分数据中可能隐藏着卫星潜在的不正常运行的重要信息.

(3) 如何实时处理数据流. 由于数据流的诸多特性,导致以动态数据流为对象的算法比以静态数据为对象的算法要困难许多.

(4) 如何设定滑动窗口的尺寸. 滑动窗口尺寸通常是一个主观变量,对用户敏感,不具客观性.

(5) 如何提高挖掘算法的效率. 传统的稀有模式挖掘算法在处理卫星遥测数据流时速度较慢,因为它们首先需要生成全部的候选集,然后再从候选集中选出符合要求的模式.

针对上述困难和挑战,本文做出的主要贡献如下:

(1) 针对卫星遥测数据流,本文提出了一种无需专业领域知识的挖掘算法,该算法只需要输入一定格式的遥测数据就可以挖掘出最小稀有模式. 考虑到用户缺乏卫星专业领域知识,因此在设计算法时,仅以遥测数据为研究对象,不借助偏差、帆板旋转角等任何卫星领域知识. 这样可以使得卫星的智能监控服务系统在挖掘模块和知识库上的耦合度较低. 否则,一旦专业知识库出错或缺乏,就会导致挖掘模块无法运行.

(2) 本文提出了一种最小稀有模式挖掘算法 MRP-TB(Minimal Rare Pattern based on Top-down and Bottom-up). 该算法通过挖掘最小稀有模式来揭示隐藏在少部分数据中的规律^[16]. 与频繁模式挖掘不同,该算法以少部分卫星遥测数据流为研究对象,目的是找出在少部分数据中隐藏的有用信息.

(3) 本文首先使用连续小波去除卫星遥测数据的噪声,然后通过小波方差模拟出遥测数据流的第一主周期. 为了使滑动窗口尺寸只与数据流本身相关,而不受人为主观因素的影响,我们以数据流的第

一主周期为参照,设定滑动窗口的尺寸.然后再将滑动窗口技术应用于卫星遥测数据流.该技术主要对卫星遥测数据流进行分段,使得 MRP-TB 算法能够实时地处理海量的卫星遥测数据流.

(4)MRP-TB 算法无需生成全部候选集,也无需挖掘出全部稀有模式.最小稀有模式能够表征所有稀有模式,但其生成数量却远小于稀有模式.该算法通过挖掘最小稀有模式来提高算法效率.此外,为了进一步提高算法的运行速度,算法同时使用自顶向下和自底向上的双向遍历技术来生成树的候选集,寻找最小稀有模式,提高算法效率.

通过真实的卫星遥测数据对所提算法进行验证,大量实验表明本文所提的算法是有效的,并且其运行速度要高于传统算法.

本文第 2 节介绍稀有模式挖掘领域的相关工作;第 3 节引入模式挖掘的预备知识,给出最小稀有模式挖掘算法和示例,分析算法的理论复杂度;第 4 节进行实验并分析实验对比结果;第 5 节总结全文.

2 相关工作

目前,模式挖掘以频繁模式挖掘居多,而稀有模式挖掘相对较少^[16].从现有的关于稀有模式挖掘的研究来看,大多数关于稀有模式挖掘的研究都是基于静态数据的.但是,随着大数据和云计算技术的飞速发展,基于动态数据的稀有模式挖掘方法也许更为实用.本文将现有的稀有模式挖掘方法分为 2 类(如表 1 所示),即基于静态数据的稀有模式挖掘和基于动态数据流的稀有模式挖掘.

表 1 稀有模式挖掘分类			
类型 1: 基于静态数据			
Hyper-Linked RPM	Apriori-Inverse	AfRIM	RP-Tree
类型 2: 基于动态数据流			
SRP-Tree	MIP-DS	MRSP-SW	SRAM

2.1 基于静态数据的稀有模式挖掘

当从一般的密集型数据中挖掘稀有模式时,模式增长法有较大的优势.但在处理稀疏数据集和短格式数据集时,该方法存在性能上的缺陷.为此,Borah 和 Nath^[18]提出了 Hyper-Linked RPM 算法.该算法采用基于队列的超链接数据结构,使得算法在挖掘稀有模式的同时能够自动调整链接,从而减少树结构的内存开销.由于该算法主要处理稀疏数

据集和短格式数据,所以在挖掘密集型数据的稀有模式时,其性能不如模式增长类的算法.

Yun 和 Rountree^[24]根据 Apriori 算法提出了一种稀有模式挖掘算法 Apriori-Inverse.该算法可以发现“完美的零星规则”,所谓“完美的零星规则”是指在发现的规则中,模式本身以及模式中包含的所有单个项都是稀有的.在算法结构上,该算法采用与 Apriori 算法相似的自下而上的搜索方式.在阈值设定上,Apriori-Inverse 算法设定了 2 个阈值,即最小支持度阈值和最大支持度阈值.只有支持度在最小支持度阈值和最大支持度阈值之间的模式才认为是稀有的.然而,该算法只能挖掘“完美的零星规则”,对于那些“不完美的零星规则”(即在规则中,模式本身是稀有的,而单个项是频繁的)则无法挖掘.

与 Apriori-Inverse 算法不同,Adda 等人^[25]提出的 AfRIM 算法没有采用自底向上的搜索方式,而是采用了自顶向下的搜索方法.该算法通过将 Apriori 算法实例化来解决稀有模式挖掘的问题.由于 AfRIM 算法没有考虑舍弃那些支持度为 0(支持度为 0 的项通常不具有实际意义)的项,所以其运行效率不高.

由于上述算法大多数是基于 Apriori 算法,所以通常会生成大量的候选集,并且需要繁琐的剪枝策略.为此,这类算法需要大量的时间来执行这个过程,从而导致算法的运行效率不高.为了解决这个问题,许多研究者们开始采用基于 FP-Growth 算法的树形结构来代替类 Apriori 算法的剪枝策略,从而避免在候选集生成和剪枝步骤上浪费大量的时间.例如,Tsang 等人^[26]提出的基于 FP-Tree 结构的 RP-Tree 算法.该算法只需要对数据库进行一次扫描就可以挖掘出符合条件的稀有模式,其运行时间比使用 Apriori 策略的算法少很多.但是该算法仍有一些缺点,它只会生成特定类型的稀有模式,存在像 Apriori-Inverse 算法那样无法挖掘出“不完美的零星规则”的问题.

2.2 基于动态数据流的稀有模式挖掘

2012 年,Huang 等人^[27]提出了一种使用树形数据结构来搜索数据流的稀有模式方法,称之为流式稀有模式树 SRP-Tree.为了与数据流的传输速度同步,SRP-Tree 算法采用了滑动窗口技术,并且该算法可以仅对数据流进行一次扫描就可以找出所需结果.此外,该算法既可以挖掘动态数据流,也可以挖掘静态数据.为了适应不断变化的数据分布,该算法设置了不同的滑动窗口大小来进行稀有模式挖

掘,但是人为的预定义滑动窗口大小,始终无法赶上流数据分布的变化.该算法可以改为动态调整滑动窗口和阈值的方式,以适应不断变化的流数据环境.

2015 年 Hemalatha 等人^[17]提出了一种最小稀有模式挖掘算法 MIP-DS.该算法首先将数据流转换为二进制矩阵的形式,以此矩阵为处理对象,采用滑动窗口以跟上数据流分布的变化.此外,他们还提出了一种扩展算法 MIFPOD,该算法针对 MIP-DS 算法挖掘出的最小稀有模式,计算出基于异常值的最小稀有模式因子 MIFPOF,然后根据 MIFPOF 判断数据流中的异常值.但该算法存在 2 个缺点:(1)将数据流转换为二进制矩阵的形式,可能会丢失部分信息;(2)该算法采用了固定的最小支持度阈值,没有考虑概念漂移,不能适应数据流的变化.

传统的模式挖掘有 2 种局限性:(1)在静态的数据集中进行挖掘;(2)结果仅是频繁模式.为了突破这种局限性,Ouyang^[8]在 2017 年提出了 MRSP-SW 算法.该算法能够在动态的数据流中挖掘稀有模式.该算法与 MIP-DS 算法类似,采用滑动窗口技术来与数据流的变化同步.从其实验结果可知,算法是高效的.但是算法用的是合成数据,不能证明该算法在真实数据上是正确的、有效的.

为了确保网络服务质量,Liu 等人^[28]于 2017 年提出了一种针对网络性能的挖掘方法,称之为基于 Spark 的稀有关联规则挖掘算法 SRAM.该算法采用 FP-Growth 数据结构,并结合 Spark 等大数据技术,可以快速地动态数据流中挖掘出符合条件的稀有关联规则.由于该算法的目的是确保网络服务质量,所以将重心放在 NPC 和 KQI 的异常值检测上.据实验分析可知,该算法的准确度为 85%,仍然有较大的晋升空间.

3 高效的最小稀有模式挖掘算法

本节给出基于双向遍历的最小稀有模式挖掘算法.在 3.1 节介绍一些最小稀有模式挖掘的预备知识;在 3.2 节给出对应的算法和解释算法的简单示例;3.3 节给出算法的效率分析;3.4 节分析算法的复杂度.

3.1 预备知识

定义项集 $I = \{i_1, i_2, \dots, i_m\}$,其中 $i_k (1 \leq k \leq m)$ 为一个项,每一个项表示事务的一条信息.假设事务数据流 $DS = \{t_1, t_2, \dots, t_n\}$ 表示一个 n 事务的集合,

其中 $t_i (1 \leq i \leq n)$ 是一组项的集合,当 $n \rightarrow \infty$ 时表示该数据流有无穷多个事务.如果一个项的集合 $I' = \{i_u, i_v, \dots, i_w\} \subseteq I (u, v, w \in [1, m])$ 是非空的,那么称这个集合 I' 为项集或者模式.如果项集 I' 包含 k 个项,则项集 I' 的长度 $|I'| = k$,并且称该项集为 k -项集.一个数据流事务 t_i 通常用一个形如 $\langle T_{id}, I_{id} \rangle$ 的二元组来表示,其中 T_{id} 表示事务标识符, I_{id} 表示事务 T_{id} 对应的模式.

定义 1. 滑动窗口是一种流量控制技术,在同一时段内仅允许处理窗口内部的数据,用 SW (Sliding Window) 表示.滑动窗口 SW 使得算法仅允许处理数据流 DS 中最近接收的事务.滑动窗口的大小用符号 $|SW|$ 表示,意思是指数据流 DS 中最近接收的事务的数量.

定义 2. 项集或者模式 I' 的支持度定义为滑动窗口 SW 中包含模式 I' 的事务数量占滑动窗口 SW 中事务总数量的比例,用式(1)表示.其中自定义函数 $Count_{sw}(I')$ 表示在滑动窗口 SW 中包含模式 I' 的事务数量.

$$Support_{sw}^i(I') = \frac{Count_{sw}(i)}{|SW|}, i \in I' \quad (1)$$

定义 3. 一个模式 I' 被称为是稀有模式^[29],当且仅当它在滑动窗口 SW 中的支持度小于用户指定的最小支持度阈值 δ ,用式(2)表示.实际中通常使用其变形公式(3),这便于判断模式的类型.

$$Support_{sw}^i(I') < \delta, i \in I' \quad (2)$$

$$Count_{sw}(i) < \delta \times |SW|, i \in I' \quad (3)$$

定义 4. 一个模式 I' 被称为是频繁模式^[29],当且仅当它在滑动窗口 SW 中的支持度大于或者等于用户指定的最小支持度阈值 δ ,用式(4)表示.实际中会使用其变形公式(5),便于判断模式的类型.

$$Support_{sw}^i(I') \geq \delta, i \in I' \quad (4)$$

$$Count_{sw}(i) \geq \delta \times |SW|, i \in I' \quad (5)$$

定义 5. 如果一个稀有模式 I' 的所有子集都是频繁的,则称这个模式 I' 为最小稀有模式,记为 MP,用式(6)表示.

$$MP = \{i | Support_{sw}^i(I') < \delta \wedge Support_{sw}^j(I') \geq \delta\}, \forall j \subset i \quad (6)$$

3.2 简单的算法示例和 MRP-TB 算法

本节分为两个部分,第 1 部分介绍算法思想,第 2 部分举例说明算法步骤,然后给出相应算法的描述.

3.2.1 MRP-TB 算法思想

MRP-TB 算法是一种快速的最小稀有模式挖掘算法.该算法基于数据本身,而无需卫星领域知

识. 采用周期作为滑动窗口尺寸,使得算法可以实时地处理数据流并减少主观因素的影响. 通过双向遍历法(BT)生成中间项集,寻找最小稀有模式. 期间使用向下闭合性(反向单调性)来减少中间项集的生成数量,以进一步提高算法效率. 算法保证遍历方向始终向最小稀有模式进行,即始终保持顶部为稀有模式,底部为频繁模式.

假设滑动窗口中包含一个 5-事务的数据流 $DS=\{t_1, t_2, t_3, t_4, t_5\}$, DS 中的事务特征域为 (a, b, c, d, e) . 每条事务的详细特征值见表 2, 其中 $\{T_1, T_2, T_3, T_4, T_5\}$ 是事务标识符, (a, b, c, d, e) 表示特征列表.

表 2 事务数据流

T_{id}	I_{id}
T_1	$\{a, b, c, d\}$
T_2	$\{a, b, c\}$
T_3	$\{a, b, c, e\}$
T_4	$\{b, c, d\}$
T_5	$\{a, b, d\}$

设最小支持度阈值 $\delta=60\%$, 根据式(3)可知, 稀有模式最多只能出现 2 次; 根据式(5)可知, 频繁模式至少出现 3 次. 以双向遍历找出最小稀有模式: 首先, 考虑 1-项集, 根据表 2 可知只有 e 的支持度小于 δ , 所以 e 是长度为 1 的最小稀有模式, 因为 1-项集不存在非空子集. 其次, 考虑项集长度大于 1 的情况, 由于稀有模式的超集一定是稀有模式, 所以包含 e 的模式都是稀有模式. 从逆向考虑, 由于这些包含 e 的超集的子集中含有稀有模式 e , 所以在之后的步骤中不需要再考虑 e 的超集, 因为它们不可能是最小稀有模式. 而对于频繁模式来说, 其超集有可能是稀有模式, 也有可能是频繁模式, 所以在之后的步骤中仍需考虑 a, b, c, d 这些 1-频繁项集. 将频繁模式按其支持度升序排序, 支持度越小越有可能是稀有模式. 此外, 根据表 2 可知 5-项集 $abcde$ 的支持度为 0, 所以 $abcde$ 是稀有模式, 尽管它没有实际意义. 由于稀有模式的子集可能是稀有模式, 也有可能是频繁模式, 所以保留 $abcde$. 以频繁模式 (a, b, c, d) 作为底部(I_{bottom}), 以稀有模式 $abcde$ 为顶部(I_{top}), 生成中间模式 I_{mid} . 该中间模式 I_{mid} 必须满足式(7)和式(8), 其中 $bottom$ 是底部的长度, top 是顶部的长度, mid 是生成的中间项集的长度.

$$mid=\left\lceil\frac{bottom+top}{2}\right\rceil$$

(7)

$$I_{bottom}\subset I_{mid}\subset I_{top}$$

(8)

根据式(1)计算 I_{mid} 的支持度; 根据定义 3 和定

义 4 判断 I_{mid} 中的项集是频繁的还是稀有的. 若是稀有的, 则以 I_{mid} 作为新的顶部, 以 I_{bottom} 作为底部继续寻找 I_{mid} ; 若是频繁的, 则以 I_{mid} 作为新的底部, 以 I_{top} 作为顶部继续寻找 I_{mid} . 如此循环直至无法生成新的中间项集, 即顶部和底部的长度相差 1. 这样确保顶部始终为稀有模式, 底部始终为频繁模式, 遍历方向始终向最小稀有模式进行. 最后, 生成顶部的所有子集并计算其支持度, 若所有子集都是频繁的, 则该顶部为最小稀有模式; 反之, 则不是.

3. 2. 2 MRP-TB 算法过程及举例

真实数据流的维度较高并且具有无限性, 因而无法用真实的数据流进行举例. 为此, 需要将数据流有限化. 此外, 由于无法提前预知数据流中的数据, 所以必须对数据流进行实时处理. 我们设一个长度为 5 的滑动窗口 SW , 并确保处理的都是最近传来的数据, 即 $|SW|=5$. 为简化说明, 我们仍以 3. 2. 1 节中表 2 的事务为例, 阈值仍设为 60% .

以 $abcde$ 为顶部, a 为底部, 并通过表 2 对应的格空间示意图(如图 1 所示)来对最小稀有模式的挖掘过程进行简单说明. 图中粗体实线椭圆表示支持度为非 0 的稀有模式, 虚线椭圆表示最小稀有模式 (e, ad, cd) , 右上角数字表示对应模式的支持数. 根据式(7)可知, 由 $abcde$ 和 a 生成的中间项集长度为 3, 符合式(8)并且不是 e 的超集(根据上文可知无需考虑 e 的超集)的模式只有 abc, abd 和 acd . 计算支持度, 可知 abc 为频繁模式, abd 和 acd 为稀有模式. 这里仅以 abd 为例, 将 abd 作为新的顶部, 而 a 作为底部, 生成的中间项集为 2-项集 ad 和 ab . 计算后可知 ad 是稀有的, 所以将其作为新的顶部, a 仍为底部, 此时无法生成中间项集. 由于顶部 ad 的子集 a 和 d 均为频繁项集, 所以找到一个最小稀有模式 ad . 另一个项集 ab 是频繁的, 所以将其作为新的底部, abd 仍为顶部, 此时无法生成中间项集. 又由于顶部 abd 的子集 ad 是稀有的, 所以 abd 不是最小稀有模式. 过程如图 1 红色标注所示, 圆圈数字表示步骤, 外围有向箭头表示执行方向, 无向连线表示与之相连的项集的状态, I_{bottom} 表示状态为底部, I_{top} 表示状态为顶部, I_{mid} 表示中间状态, MRP 表示最小稀有模式, FP 表示频繁模式, “ $\rightarrow$ ”表示状态转换符. 不断重复上述步骤, 最终从表 2 的窗口数据中挖掘出 3 个最小稀有模式: e, ad 和 cd . 此外, 由于支持度为 0 的模式不具有实际意义, 所以如果最终的挖掘结果中含有支持度为 0 的模式, 则将其舍弃.

将上述过程用算法表示, 可以得出一个高效的

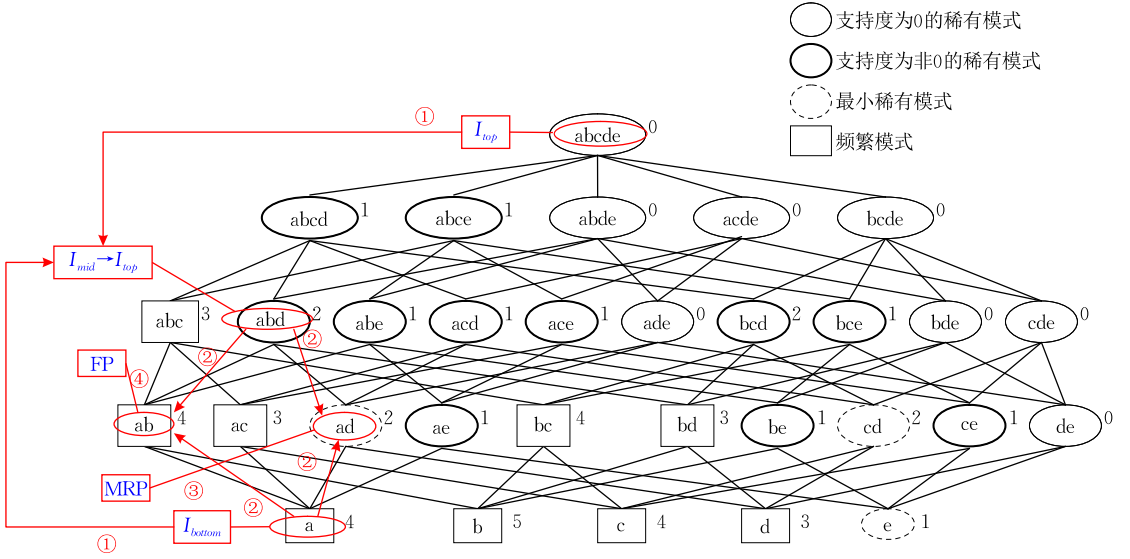


图 1 最小稀有模式挖掘过程演示

挖掘算法,即支持双向遍历的最小稀有模式挖掘算法,用算法 1 和函数 1 表示。

算法 1. MRP-TB.

输入: DS : 事务数据流

$|SW|$: 滑动窗口尺寸

δ : 最小支持度阈值

输出: MRP: 最小稀有模式

1. $C_1 \leftarrow \{1\text{-itemset}\}$
2. $Support_{SW}^i(C_1) = \frac{Count_{SW}(i)}{|SW|}, i \in C_1$
3. $MRP_1 \leftarrow \{i \in C_1 \mid Support_{SW}^i < \delta\}$
4. $FP_1 \leftarrow \{i \in C_1 \mid Support_{SW}^i \geq \delta\}$
5. $FP_1 \leftarrow \text{sort}\{Support_{SW}^j(FP_1)\}, j \in FP_1$
6. $C_{len} \leftarrow \{len\text{-itemset}\}$
7. $Support_{SW}^k(C_{len}) = \frac{Count_{SW}(k)}{|SW|}, k \in C_{len}$
8. $RP_{len} \leftarrow \{k \in C_{len} \mid Support_{SW}^k < \delta\}$
9. $FP_{len} \leftarrow \{k \in C_{len} \mid Support_{SW}^k \geq \delta\}$
10. $RP_{len} \leftarrow \text{sort}\{Support_{SW}^r(RP_{len})\}, r \in RP_{len}$
11. FOR all S_i in RP_{len} {
12. FOR all S_j in FP_1 {
13. $MRP_{next} \leftarrow BT(S_j, S_i)$
14. }
15. }
16. $MRP \leftarrow \text{Filtering}(MRP_1 \cup MRP_{next})$

函数 1. BT.

输入: S_{top} : 顶部项集, 属于集合 RP_{len}

S_{bottom} : 底部项集, 属于集合 FP_1

δ : 最小支持度阈值

输出: MRP_{next} : 长度 > 1 的最小稀有模式

1. $BT(S_{bottom}, S_{top}) \{$

2. $mid = \left\lceil \frac{bottom + top}{2} \right\rceil$
3. IF $(top - bottom \leq 1) \{$
4. IF $(S_{bottom} \text{ is rare}) \{S_{top} \text{ does not belong to } MRP_{next}\}$
5. ELSE IF (all the subitems of S_{top} are frequent)
6. $\{MRP_{next} \leftarrow S_{top}\}$
7. }
8. ELSE {
9. $S_{mid} \leftarrow \text{generate middle items between } S_{bottom} \text{ and } S_{top}$
10. FOR all S_i in S_{mid} {
11. IF $(S_i \text{ is frequent})$
12. $BT(S_i, S_{top})$
13. ELSE
14. $BT(S_{bottom}, S_i)$
15. }
16. }
17. RETURN MRP_{next}
18. }

MRP-TB 算法采用 3 种参数作为输入: 事务数据流 DS , 滑动窗口尺寸 $|SW|$ 和最小支持度阈值 δ . 算法结果是最小稀有模式 MRP. 首先, 考虑所有长度为 1 的模式, 将其放入集合 C_1 . 根据式(1)计算出 C_1 中所有模式的支持度, 根据式(2)找出所有稀有模式. 由于 1-项集不存在非空子集, 所以这些稀有的 1-项集就是最小稀有模式, 并将它们放入集合 MRP_1 ; 然后将剩余的频繁 1-项集放入 FP_1 , 并按其支持度升序排序. 其次, 找出所有长度为 len (len 是所有不同项的个数) 的模式, 并将其放入集合 C_{len} , 然后根据式(1)计算它们的支持度, 根据式(2)和式(4)区分稀有模式和频繁模式, 并将稀有的 len -项集放

入集合 RP_{len} , 将频繁的 len -项集放入集合 FP_{len} . 最后, 以集合 RP_{len} 为顶部, 集合 FP_1 为底部, 调用双向遍历函数 BT (Bidirectional Traversal) 找出所有长度大于 1 的最小稀有模式, 结合集合 MRP_1 输出最终结果 MRP. BT 函数采用算法 1 的处理结果集 RP_{len} 和 FP_1 作为输入, 输出结果是长度大于 1 的最小稀有模式 MRP_{next} . 虽然该函数与折半查找形似, 但却存在许多改进之处. 例如, 折半查找每次 (非最后一次) 迭代都存在唯一确定的中间项, 而 BT 函数每次迭代生成的中间项集的值和数量都无法确定, 需要根据限制条件进行确定. 因为折半查找法在 BT 函数中只能确定每次迭代生成的中间项集的长度, 而无法确定生成中间项集的具体值和数量. 此外, BT 函数在算法出口处并没有直接退出, 而是新增了一个关键性的限制条件, 用来判别某一模式是否为最小稀有模式, 如 BT 函数的第 4~6 行所示.

3.3 MRP-TB 算法效率分析

从 3.2.2 节图 1 示例的分析过程可以看出, 双向遍历算法 MRP-TB 只需 3 步就可以找到最小稀有模式 ad, 在图 1 中用 MRP 表示. 如果不采用双向遍历算法, 而使用单向遍历算法, 那么算法 MRP-TB 的效率会很低. 仍以图 1 寻找最小稀有模式 ad 为例, 下面对采用单向遍历算法的过程进行分析:

(1) 以自顶向下遍历为例. 第 1 步, 单向算法首先计算 abcde 的支持度, 并判断其是否是稀有的; 第 2 步, 以 abcde 为种子, 生成候选集. 通过计算支持度得到 2 个稀有模式, 即 abcd 和 abce; 第 3 步, 以 abcd 和 abce 为种子生成新的候选集. 在算法中, 由于上述生成过程是并行执行的, 所以在效率分析中以单个稀有模式为种子即可 (下同). 计算由 abcd 生成的候选集的支持度, 可以得到 3 个稀有模式, 即 abd、acd 和 bcd; 第 4 步, 以 abd 为种子, 生成候选集 ab 和 ad. 计算其支持度, 可以得到稀有模式 ad; 第 5 步, 以 ad 为种子, 生成候选集 a 和 d. 计算它们的支持度后可以确定 a 和 d 均为频繁的; 第 6 步, 从所有稀有模式中识别出最小稀有模式 ad.

(2) 以自底向上遍历为例. 第 1 步, 从所有的 1-项集中找出所有的稀有模式. 从图 1 中可以知道, 仅 e 为稀有的, 其它 4 个 1-项集均为频繁的; 第 2 步, 以频繁项 a 为种子, 生成候选集 ab、ac、ad 和 ae. 其中 ad 和 ae 为稀有的, ab 和 ac 为频繁的; 第 3 步, 以 ab 为种子生成新的候选集 abc、abd 和 abe. 其中 abd 和 abe 是稀有的, abc 是频繁的; 第 4 步, 以 abc 为种子生成候选集 abcd 和 abce. 计算它们的支持度可以确

定它们均为稀有模式; 第 5 步, 从所有稀有模式中识别出最小稀有模式 ad.

从上述分析过程并结合图 1 可知, 双向遍历算法仅需 3 步即可发现最小稀有模式 ad, 而单向的自顶向下方法需要 6 步, 单向的自底向上方法需要 5 步, 显然双向遍历 MRP-TB 算法的效率明显优于单向遍历.

3.4 MRP-TB 算法复杂度分析

首先, 分析 MRP-TB 算法的时间复杂度.

(1) 算法 MRP-TB 的第 1~4 行用于区分稀有 1-项集和频繁 1-项集, 需要计算每个项的支持度. 因此, 算法 MRP-TB 的第 1~4 行的时间复杂度为 $O(\omega N)$. 其中 ω 为事务的平均宽度, N 为事务数.

(2) 算法 MRP-TB 的第 5 行为频繁 1-项集的堆排序过程. 因此, 算法 MRP-TB 的第 5 行的时间复杂度为 $O(m \log_2 m)$. 其中 m 为频繁 1-项集的数量.

(3) 算法 MRP-TB 的第 6~9 行用于区分稀有 len -项集和频繁 len -项集 (len 是所有不同项的个数, 图 1 中 $len=5$, 即有 a、b、c、d、e 这 5 个不同的项). 其中, 最好的情况是 C_{len} 中只有一个支持度不为 0 的稀有项集, 因此只需要计算 1 次; 最坏的情况是长度为 len 的项的支持度为 0 或为频繁项, 则需要用长度 $len-1$ 并且支持度不为 0 的稀有项的项来代替, 时间开销为 $O(\omega N)$. 因此, 算法 MRP-TB 的第 6~9 行的最坏时间复杂度为 $O(\omega N)$.

(4) 算法 MRP-TB 的第 10 行为稀有 len -项集的堆排序过程. 因此, 算法 MRP-TB 的第 10 行的时间复杂度为 $O(n \log_2 n)$. 其中 n 为稀有 len -项集的数量.

(5) 算法 MRP-TB 的第 11~15 行为生成最小稀有模式的过程. 其中, 第 13 行为双向遍历过程, 其时间复杂度为 $O(\log_2 N)$, 进而推出算法 11~15 的时间复杂度为 $O(mn \log_2 N)$. 算法第 16 行为过滤函数, 其时间复杂度为 $O(1)$.

因此, 算法 MRP-TB 所需的总时间开销为 $O(2\omega N + m \log_2 m + n \log_2 n + mn \log_2 N + 1)$.

之后, 分析 MRP-TB 算法的空间复杂度.

(1) 算法 MRP-TB 的第 1 行和第 6 行为构造初始集合的过程, 第 3~4 行和第 8~9 行是构造稀有项集和频繁项集的过程. 因此空间复杂度与事务数量呈线性关系, 即为 $O(N)$.

(2) 算法 MRP-TB 的第 5 行和第 10 行是堆排序过程, 所需辅助空间的复杂度为 $O(1)$.

(3) 从函数 BT 第 10~15 行可见, 双向遍历方法

是一种递归过程,并且每次递归都会创建变量 mid . 由于 BT 函数的递归深度为 $O(\log 2N)$,因此函数 BT 的空间复杂度为 $O(\log 2N)$. 进而推到算法 MRP-TB 的第 11~15 行的空间复杂度为 $O(mn\log 2N)$.

因此,算法 MRP-TB 所需的总辅助空间为 $O(N+1+mn\log 2N)$.

4 实验结果和分析

为了评估本文提出的基于双向遍历的最小稀有模式挖掘算法 MRP-TB 的有效性,使用 2015 年某卫星的真实遥测数据流进行验证. 整个实验分为 3 部分,即第 1 部分是数据预处理;第 2 部分选取卫星数据流的关键特征验证所提算法的有效性,即证明算法可以找出最小稀有模式 MRP;第 3 部分,通过对比实验验证所提算法的效率.

4.1 数据预处理

实验数据是某卫星 2015 年的遥测数据,总共有 80 个特征参数. 我们截取 2015 年 3 月 1 日~2015 年 5 月 30 日这一个时间段的遥测数据作为本次实验的初始数据,约 1600 万条记录. 由于卫星遥测数据中存在许多冗余数据,所以我们将这些数据记录以 1min 为单位进行压缩,然后利用一维小波去除噪声,从而可以将原始记录压缩成约 13 万条数据记录^[30]. 但这种压缩方式具有特殊性. 因为卫星遥测数据中存在这样的特殊数据,它们在某一时间段内的数值是相同的或相近的. 因此,经过离散化处理后,这些数据大都属于同一区间,会映射到同一整数上. 假设在某一长度为 $|SW|$ 的窗口中,有 m 个数据值映射到整数 1 上, n 个数据值映射到整数 2 上,并且 $m+n$ 的个数无限接近 $|SW|$,那么 $Support(1) = m/|SW|$. 假设压缩率为 $p\%$,那么对项目 1 和 2 进行压缩后,项目 1 的支持度 $Support(1) = (m \times p\%) / [(m+n) \times p\% + (|SW| - m - n)]$. 由于 $m+n$ 无限接近 $|SW|$,即 $|SW| - m - n$ 趋于 0,所以 $(|SW| - m - n) \times p\%$ 仍然趋于 0. 因此, $Support(1) = (m \times p\%) / [(m+n) \times p\% + (|SW| - m - n) \times p\%] = m/|SW|$. 所以在这种特殊的数据上使用这种压缩方式不会改变项目的支持度,也不会影响稀有模式的判定. 再结合专家经验和文献^[30],最终选择以 1min 为单位对数据进行特殊的压缩处理. 此外,由于实验数据的维度较高,包含 80 个特征参数,因此需要对原始数据进行降维处理. 其原因如下:一方面,算法的性能受到特征参数数量的影响. 对于同一

个算法来说,增加目标数据的特征参数容易造成算法的计算量增大,从而导致算法的性能下降,更为重要的是,不必要的特征参数会干扰算法的挖掘效果. 因此需要通过降维来减少算法的计算. 另一方面,卫星遥测数据存在许多冗余参数. 所谓冗余参数是指那些变化趋势相近、存在明显函数关系的参数,由于这些参数关联性较大,符合相似的变化规律,因此不需要进行重复计算. 所以我们通过灰关联分析从 80 个特征参数中选取 10 个相对独立的关键特征,实现参数降维,以减少数据集中的冗余参数. 本次实验选取的特征如表 3 所示.

表 3 关键参数特征描述

ID	Feature ID
1	1026
2	1030
3	1031
4	1049
5	1053
6	1054
7	1058
8	1076
9	13395
10	13396

ID 表示序号,Feature ID 表示特征参数的遥测代号. 针对数值型的遥测参数,需要在挖掘之前对其进行离散化处理. 由于卫星遥测数据中每个参数的取值较为集中,因此等宽度划分与等深度划分都无法满足离散化要求,而基于聚类的划分方法可以直观地反应原始数据的分布情况,符合离散化要求^[31]. 因此,在考虑数据分布的情况下,本文对每个参数单独执行聚类,然后根据各自的聚类结果将它们的域划分为多个长度不等的不相交区间,从而得到 10 个参数的离散化结果.

4.2 最小稀有模式挖掘分析

由于实验设备受限,无法使用真实的卫星进行实验. 我们将卫星遥测数据模拟成数据流作为算法的输入. 整个算法代码是在集成开发环境 Eclipse 上完成的,编程语言是 JAVA,编译器采用 JDK 1.8. 算法在一台服务器上运行,环境配置如下: Intel(R) Xeon(R) CPU E5-2620、2.40GHz CPU、64GB RAM、Windows Server 2008 HPC Edition 64-bit OS.

表 4 给出了本文所提算法 MRP-TB 在真实的卫星数据(简称 Real Data)和 UCI 公开数据集的运行情况. 其中,公开数据集包括: Breast Cancer Wisconsin(简称 BCW)、Contraceptive Method Choice(简称 CMC)和 Mushroom. 通过对比数据集 BCW

和 CMC 的运行结果,可以看出维度相同时,所提算法在 BCW 上的时间开销小于在 CMC 上的时间开销.这是因为 BCW 的数据记录数小于 CMC 的数据记录数.而对比数据 Mushroom 和真实数据(Real Data)的运行结果,可以看出尽管 Mushroom 的记录条数远小于真实数据的记录数,但所提算法在 Mushroom 上的时间开销却大于在真实数据上的时间开销.这是因为 Mushroom 有较高的维度.因此,从表 4 可知,所提算法的时间开销与数据集记录条数和数据维度有关,并且记录条数越多、维度越高,算法消耗的时间越多.

表 4 MRP-TB 算法在多个数据集上的运行结果

数据集	记录条数	维数	最小稀有模式数量	时间开销/ms	内存开销/MB
BCW	699	9	12	10	6.406
CMC	1473	9	43	15	7.406
Mushroom	8124	22	54	73	9.938
Real data	132480	10	7	58	10.129

本节将所提算法与现有的 3 类稀有模式挖掘算法对比:Apriori-Inverse 算法,该算法用于挖掘完全

稀有模式;CORI 算法,该算法用于挖掘稀有关联项集;MIP-DS 算法,用于挖掘最小稀有模式.图 2 展示了 5 个连续窗口中挖掘出的稀有模式的数量分布.图 3(a)~(e)分别描述了 5 个连续窗口中稀有模式的长度分布情况.其中, L_i 表示长度为 i 的模式.从图 2 可以看出,本文所提算法挖掘的稀有模式数量要少于 Apriori-Inverse 算法和 CORI 算法.因此,可以确定最小稀有模式挖掘算法的效率要高于非最小稀有模式挖掘算法.此外,结合图 2 和图 3 可以知道,在稀有模式的数量和长度分布上,所提算法的挖掘结果与最小稀有模式挖掘算法 MIP-DS 相符.因此,可以确定本文所提 MRP-TB 算法具有较高的准确性.

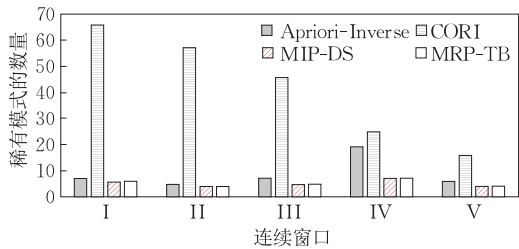


图 2 稀有模式数量分布

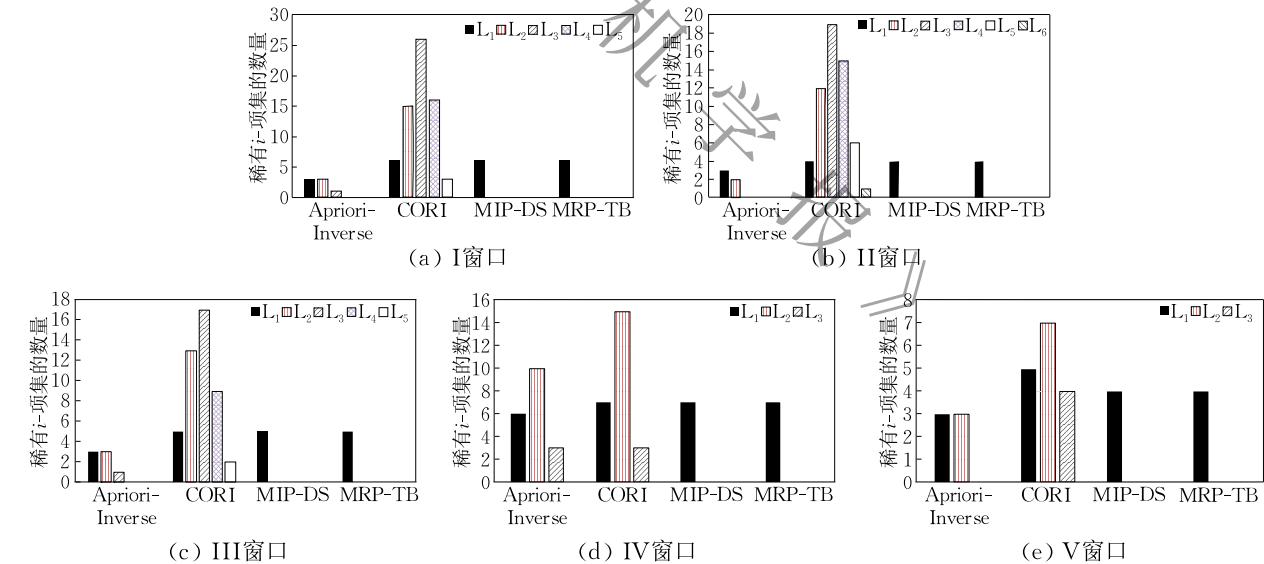


图 3 稀有模式长度分布

表 5 给出了 MRP-TB 算法在 5 种不同情况下的部分挖掘结果。 $|SW|$ 表示滑动窗口的尺寸, $step$ 表示滑动窗口的步长, $minimum\ support$ 表示在滑动窗口 SW 中的最小支持数,它是最小支持度阈值 δ 的另一种表示形式,是根据式(3)或式(5)的右半部分变换而来,即 $\delta \times |SW|$. $No. x\ window\ (x \in N+)$ 表示第 x 个滑动窗口(滑动窗口编号从 1 开始). $rare\ count$ 表示在滑动窗口中发现的最小稀有模式

的数量. MMU (Maximum Memory Usage) 表示在滑动窗口中算法的最大内存开销的平均值,单位为 MB. T 表示在滑动窗口中算法的平均执行时间,单位为 ms. MRP 表示在滑动窗口中发现的最小稀有模式. $Support\ Count$ 表示在滑动窗口中包含模式 MRP 的记录数,其值可以根据式(1)的自定义函数 $Count$ 计算.表 5 挖掘结果中的数字 1~10 与卫星属性名称相对应,它是卫星特征参数代号的简单表

表 5 MRP-TB 算法在 5 种不同情况下的部分挖掘结果

SW =1440 minimum support=144 step=10											
No.764 window rare count=7 MMU=13.60 MB T=22 ms			No.795 window rare count=7 MMU=15.59 MB T=24 ms			No.817 window rare count=7 MMU=15.14 MB T=24 ms			No.1680 window rare count=7 MMU=23.32 MB T=22 ms		
No.1882 window rare count=7 MMU=12.81 MB T=26 ms			No.2026 window rare count=7 MMU=19.36 MB T=27 ms								
MRP	Support Count	MRP	Support Count	MRP	Support Count	MRP	Support Count	MRP	Support Count	MRP	Support Count
{3}	71	{6}	2	{6}	9	{6}	4	{1,4}	1	{2,4}	71
{6}	9	{1,4}	2	{8}	73	{1,5}	55	{2,4}	60	{3,4}	64
{1,4}	129	{1,8}	2	{1,4}	129	{2,5}	55	{3,4}	53	{4,7}	126
{1,8}	73	{8,9}	139	{4,5}	98	{3,5}	55	{4,7}	109	{4,9}	98
{2,9}	98	{8,10}	90	{4,9}	91	{5,7}	55	{4,9}	78	{4,10}	90
{2,10}	93	{3,9,10}	110	{4,10}	80	{5,9}	24	{4,10}	79	{5,6}	7
{8,9,10}	125	{4,9,10}	80	{3,5,9,10}	129	{5,10}	25	{5,9,10}	134	{5,9,10}	95
SW =2880 minimum support=288 step=80											
No.76 window rare count=8 MMU=33.21 MB T=25 ms			No.1031 window rare count=8 MMU=18.14 MB T=25 ms			No.1085 window rare count=8 MMU=16.82 MB T=23 ms			No.1220 window rare count=8 MMU=8.60 MB T=24 ms		
No.1614 window rare count=8 MMU=18.10 MB T=22 ms			No.1618 window rare count=8 MMU=7.04 MB T=22 ms								
MRP	Support Count	MRP	Support Count	MRP	Support Count	MRP	Support Count	MRP	Support Count	MRP	Support Count
{1,4}	129	{6}	3	{1,3}	269	{6}	5	{5}	3	{2}	251
{1,6}	9	{1,4}	161	{1,9}	162	{1,4}	118	{1,4}	140	{3}	149
{1,8}	73	{2,5}	75	{1,10}	172	{2,9,10}	213	{1,9}	254	{5}	182
{1,10}	286	{3,9,10}	285	{3,9}	175	{3,5,10}	257	{1,10}	242	{1,4}	140
{4,6}	7	{4,5,9}	237	{3,10}	191	{3,9,10}	173	{4,8}	123	{4,8}	123
{5,6}	9	{4,5,10}	260	{2,9,10}	219	{4,8,9}	184	{8,9}	246	{1,9,10}	193
{4,9,10}	263	{4,9,10}	264	{5,9,10}	284	{4,8,10}	145	{8,10}	237	{4,9,10}	224
{6,8,10}	286	{1,5,9,10}	172	{6,8,9,10}	189	{4,9,10}	213	{4,9,10}	224	{8,9,10}	195
SW =1440 minimum support=288 step=300											
No.155 window rare count=8 MMU=38.84 MB T=24 ms			No.185 window rare count=8 MMU=56.32 MB T=23 ms			No.213 window rare count=8 MMU=63.40 MB T=21 ms			No.263 window rare count=8 MMU=9.40 MB T=24 ms		
No.265 window rare count=8 MMU=42.71 MB T=22 ms			No.289 window rare count=8 MMU=17.41 MB T=22 ms								
MRP	Support Count	MRP	Support Count	MRP	Support Count	MRP	Support Count	MRP	Support Count	MRP	Support Count
{4}	244	{2,5,6}	211	{1,9}	234	{5}	180	{1}	246	{8}	47
{5,9}	186	{2,6,9}	206	{1,10}	203	{1,8}	154	{3}	143	{1,6,9}	243
{5,10}	174	{2,6,10}	210	{3,8}	270	{3,8}	78	{5,6}	2	{1,6,10}	238
{1,9,10}	286	{2,9,10}	234	{3,9}	183	{1,9,10}	216	{2,8,9}	271	{1,9,10}	176
{6,8,9}	248	{6,9,10}	239	{3,10}	154	{2,8,9}	271	{2,8,10}	261	{3,6,9}	256
{6,8,10}	240	{5,6,8,9}	217	{8,9}	215	{2,8,10}	261	{2,9,10}	162	{3,6,10}	257
{6,9,10}	260	{5,6,8,10}	218	{8,10}	183	{3,9,10}	188	{5,9,10}	190	{3,9,10}	284
{8,9,10}	202	{5,8,9,10}	267	{2,9,10}	184	{8,9,10}	202	{6,9,10}	183	{2,6,9,10}	258
SW =2880 minimum support=576 step=960											
No.50 window rare count=7 MMU=97.95 MB T=28 ms			No.75 window rare count=7 MMU=15.92 MB T=26 ms			No.107 window rare count=7 MMU=76.56 MB T=25 ms			No.118 window rare count=7 MMU=28.79 MB T=24 ms		
No.121 window rare count=7 MMU=79.62 MB T=28 ms			No.132 window rare count=7 MMU=37.23 MB T=25 ms								
MRP	Support Count	MRP	Support Count	MRP	Support Count	MRP	Support Count	MRP	Support Count	MRP	Support Count
{4}	229	{3}	568	{5}	14	{1}	111	{4}	44	{3,9}	532
{5,9}	383	{6}	184	{6}	490	{5}	484	{2,8}	349	{3,10}	526
{5,10}	378	{1,9}	450	{1,9}	441	{4,6}	479	{3,8}	252	{1,5,9}	520
{6,8,9}	357	{1,10}	456	{1,10}	403	{4,9}	533	{8,9}	454	{1,5,10}	485
{6,8,10}	324	{2,9}	512	{3,9}	373	{4,10}	493	{8,10}	427	{1,9,10}	346
{6,9,10}	274	{2,10}	536	{3,10}	344	{8,10}	552	{2,5,9,10}	552	{2,9,10}	407
{8,9,10}	360	{8,9,10}	372	{2,9,10}	327	{6,8,9}	319	{3,5,9,10}	525	{5,9,10}	396
SW =4320 minimum support=1296 step=1440											
No.16 window rare count=7 MMU=45.22 MB T=27 ms			No.23 window rare count=7 MMU=158.05 MB T=29 ms			No.42 window rare count=7 MMU=223.66 MB T=32 ms			No.77 window rare count=7 MMU=89.90 MB T=26 ms		
No.83 window rare count=7 MMU=198.37 MB T=25 ms			No.84 window rare count=7 MMU=213.16 MB T=25 ms								
MRP	Support Count	MRP	Support Count	MRP	Support Count	MRP	Support Count	MRP	Support Count	MRP	Support Count
{1,9}	15	{4}	594	{4}	85	{1}	1240	{1}	500	{1}	486
{2,9}	15	{5}	1	{5}	749	{3}	1075	{2}	977	{4}	1184
{3,9}	15	{9,10}	718	{1,9,10}	1103	{5}	692	{3}	508	{5}	637
{5,9}	15	{1,6,9}	831	{6,8,9}	1022	{2,9}	812	{4}	146	{8}	1027
{7,9}	16	{1,6,10}	793	{6,8,10}	1028	{2,10}	840	{5}	880	{6,9}	959
{8,9}	16	{6,8,9}	636	{6,9,10}	871	{6,9}	1111	{6}	856	{6,10}	913
{9,10}	8	{6,8,10}	607	{8,9,10}	863	{6,10}	942	{9,10}	1023	{9,10}	1185

示形式. 以表 5 中的第 764 窗口存在的最小稀有模式 $\{2, 10\}$ 为例, 由于滑动窗口尺寸是卫星周期, 因此该模式对应到卫星数据中的实际意义如下: 在卫星运行周期内, 发现遥测代号 1030 表示的卫星部件和遥测代号 13396 表示的卫星部件构成了最小稀有模式 $\{1030, 13396\}$, 因此该记录内的这两个卫星遥测部件很有可能会发生异常, 需要提前防备.

本次实验引入滑动窗口和滑动步长的概念来实时处理卫星遥测数据流. 滑动窗口的尺寸是处理遥测数据流的一个重要参数, 它直接影响挖掘对象的基数. 因此, 当滑动步长相同时, 滑动窗口尺寸越大, 算法所需的时间开销就越大, 效率也越低. 以卫星数据流的第一主周期为参照, 设置滑动窗口尺寸. 通过小波分析可以得出, 遥测参数大多数特征的第一主周期都集中在 1440 左右, 所以本次实验滑动窗口的尺寸以 1440 为参照进行设置. 而滑动步长则决定了窗口对数据的划分粒度. 在窗口尺寸相同时, 如果步长较小, 数据划分的粒度较细, 那么挖掘算法执行的次数较多, 消耗的时间较大, 此时的算法效率较低, 但挖掘结果的准确性较高. 相反, 如果滑动步长较大, 划分粒度较粗, 那么算法的效率较高, 但挖掘结果的准确性会降低. 根据上述分析, 滑动窗口尺寸和滑动步长对算法的效率和准确率有一定的影响, 我们设置如下 5 种不同情况进行具体的对比分析:

- (1) 滑动窗口尺寸为 1 倍周期, 最小支持度阈值为 0.1, 滑动步长为 10;
- (2) 滑动窗口尺寸为 2 倍周期, 最小支持度阈值不变, 滑动步长增加为 80;
- (3) 滑动窗口尺寸为 1 倍周期, 最小支持度阈值为原来的 2 倍, 滑动步长增加到 300;
- (4) 滑动窗口尺寸和最小支持度阈值均为原来的 2 倍, 滑动步长增加到 960;
- (5) 滑动窗口尺寸和最小支持度阈值均为原来的 3 倍, 滑动步长增加到 1440.

为了保证实验结果具有一般性, 分别对上述 5 种情况下的算法都执行 1001 次, 然后对算法的内存开销和执行时间计算平均值. 由于每次都是基于同一组数据流对象, 所以挖掘的最小稀有模式都相同, 记录一组结果即可. 分别从每种情况中随机选取 6 组结果不重复的窗口进行分析, 结果见表 5.

对表 5 进行横向分析, 当滑动窗口尺寸为 1 倍周期, 最小支持度阈值为 0.1, 滑动步长为 10 时, 从随机选择的 6 个窗口中可以看出, 每一个窗口均可以挖掘

出最小稀有模式 MRP, 每个模式对应的支持数见 Support Count. 此外, 对比 No.817 窗口、No.1882 窗口和 No.2026 窗口, 可以发现同一模式在不同的窗口中的支持数是不同的, 它们之间互不关联. 例如, 最小稀有模式 $\{4, 10\}$ 在上述窗口中的支持数分别为 80, 79 和 90. 在这种情况下, 所有窗口的平均运行时间在 24 ms 左右, 平均最大内存开销在 17 MB 左右.

当滑动窗口尺寸为 2 倍周期, 最小支持度阈值不变, 滑动步长增加到 80 时, 从对应窗口的挖掘结果中可以看出, MRP-TB 算法仍可以挖掘出最小稀有模式. 尽管在 No.76 窗口, No.1031 窗口和 No.1220 窗口中出现了相同的模式 $\{1, 4\}$, 但它们的支持度是互不相同的. 因为它们之间是相互独立的, 不是重复结果. 在这种情况下, 所有窗口的平均运行时间在 24 ms 左右, 平均最大内存开销在 17 MB 左右.

当滑动窗口尺寸为 1 倍周期, 最小支持度阈值为原来的 2 倍, 滑动步长增加到 300 时, MRP-TB 算法可以从不同窗口中挖掘出对应的最小稀有模式. 尽管 No.155 窗口和 No.289 窗口存在相同的最小稀有模式 $\{1, 9, 10\}$, 但该模式在 155 窗口的支持数为 286, 在 289 窗口的支持数为 176. 因此, $\{1, 9, 10\}$ 不是重复结果. 在这种情况下, 所有窗口的平均运行时间在 23 ms 左右, 而平均最大内存开销并不集中, 存在较大差异.

当滑动窗口尺寸和最小支持度阈值均为原来的 2 倍, 滑动步长增加到 960 时, MRP-TB 算法可以从每个窗口中挖掘出对应的最小稀有模式 MRP. No.75 窗口和 No.107 窗口存在相同的模式 $\{1, 9\}$ 和 $\{1, 10\}$, 但由于它们的支持数不同, 所以不是重复结果. 在这种情况下, 所有窗口的平均运行时间在 26 ms 左右, 但平均最大内存开销存在较大差异.

当滑动窗口尺寸和最小支持度阈值均为原来的 3 倍, 滑动步长增加到 1440 时, MRP-TB 算法仍然可以挖掘出最小稀有模式. No.23 窗口和 No.42 窗口仍然存在不重复的相同模式 $\{4\}$. 在这种情况下, 所有窗口的平均运行时间在 27 ms 左右, 但平均最大内存开销存在较大差异.

对表 5 进行纵向分析, 在上述 5 种情况下, 算法的平均运行时间都在 25 ms 左右, 相差不大, 这说明在挖掘遥测数据流时, MRP-TB 算法的运行时间主要受事务数量的影响, 而受其它参数(窗口尺寸、阈值、步长)的影响相对较小, 因为本实验是在不同的参数设置下, 采用同一个数据源进行的. 从挖掘结果

上看,30 个不同的窗口中都可以发现相同的最小稀有模式,这说明 MRP-TB 算法对卫星遥测数据流挖掘是有效的、可信的。

图 4(a)~(e)描述了所提算法在上述 5 种情况下的全部内存消耗情况.从图 4(a)和(b)可以看出,由于算法刚开始执行时,需要申请、调度和释放内存,因此造成算法程序在前期窗口中的内存消耗陡然升高和降低,如图 4(a)的 487 号窗口之前,图 4(b)的 367 号窗口之前.除此之外,所提算法在绝大多数窗口中的最大内存开销均较为稳定,分别集中

在[100 MB,150 MB]和[50 MB,100 MB]之间.从图 4(c)~(e)可以看出,所提算法在这 3 种情况下的内存消耗非常不稳定,并且包含许多消耗 150 MB 以上内存的窗口.因此,在算法的平均最大内存开销方面,前 2 种情况的内存开销相对较少,并且随着算法的持续执行不会有明显的变化;而其它 3 种情况下的内存开销较大,并且稳定性较差,这说明 MRP-TB 算法的空间开销与窗口尺寸、阈值大小和步长这 3 个参数密切相关.因为在不同的参数组合下,所提算法的空间开销情况有较大的差异。

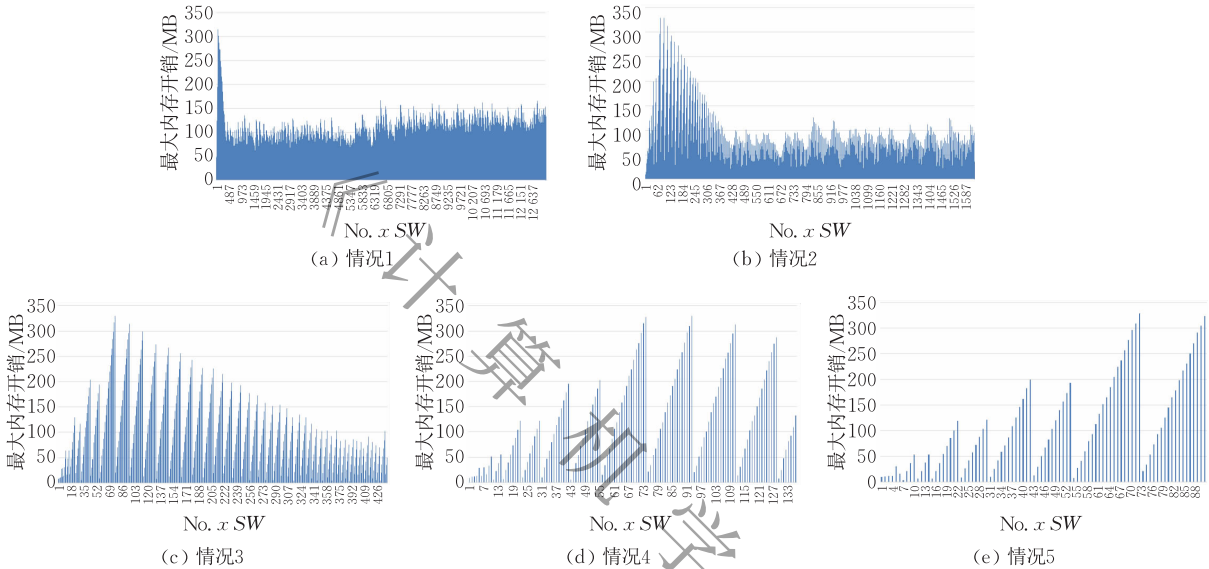


图 4 5 种情况下的全部内存消耗情况

4.3 算法运行时间对比

将本文提出的 MRP-TB 算法与 MRSP-SW 算法(Ouyang^[8],2017),MIP-DS 算法(Hemalatha,Vaidehi,Lakshmi^[17],2015)和 MCRP-Tree 算法(Bhatt,Patel^[19],2015)进行对比.实验采用可重叠性滑动窗口,设置滑动步长 $step=1$.图 5(a)~(c)分别描述了所提算法与对比算法在不同最小支持度阈值、不同滑动窗口尺寸和不同数量事务流下的运行时间。

从图 5(a)可以看出,随着支持度阈值的增加,MCRP-Tree 算法和 MRSP-SW 算法的运行时间呈上升趋势,MIP-DS 和 MRP-TB 算法呈下降趋势,并且 MIP-DS 算法的运行时间在 200~300 s 之间,而 MRP-TB 算法的运行时间在 200 s 以下.从图 5(b)可以看出,随着窗口尺寸的增加,4 种算法的挖掘时间均呈现上升趋势.但是 MCRP-Tree 算法和 MRSP-SW 算法的上升较为明显,而 MIP-DS 算法和 MRP-TB 算法的上升较为缓慢,并且 MRP-TB

算法的运行时间始终低于对比算法.从图 5(c)可知,当事务流的数量为 3×10^4 左右时,MRP-TB 算法和 MIP-DS 算法的运行时间相差不大,MCRP-Tree 算法和 MRSP-SW 算法的运行时间几乎相同.但是随着事务数量的增加,4 种算法均呈现明显的上升趋势.其中 MRP-TB 算法的上升趋势最低,并且其运行时间始终低于其它 3 种算法.目前,关于卫星数据挖掘的研究现状表明,当事务数量达到 3×10^4 左右时,单个挖掘算法的时间开销接近 300 s,并且整体呈上升趋势^[32].因此,根据研究现状和专家经验可知,本文所提算法的挖掘速度满足实际需求,并且在此基础上得到了进一步的提高。

对比算法的时间复杂度可知,本文所提算法与 MIP-DS 算法相比,其主要优势在使用双向遍历法寻找最小稀有模式,时间复杂度仅为 $O(\log_2 N)$,而 MIP-DS 算法需要时间开销为 $O(N)$ 才可以找到最小稀有模式.但是在算法开始处,本文所提算法和 MIP-DS 算法都需要为每个项目计算支持度,时间开

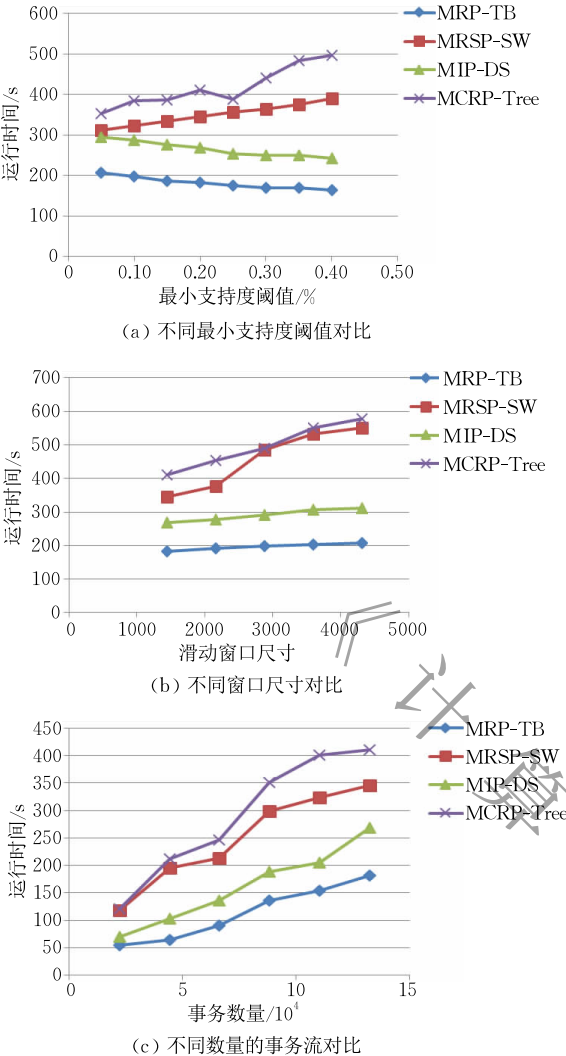


图 5 4 种算法的运行时间对比

销为 $O(N)$. 因此, 综合来看, 本文所提算法的时间复杂度均仍为 $O(N)$. 从图 5 可以看到, 和其它两种算法相比, 本文所提算法和 MIP-DS 算法的时间开销较为接近, 并且要小于 MIP-DS 算法的时间开销. 原因是本文所提算法采用了双向遍历法来寻找最小稀有模式, 其性能优于 MIP-DS 算法. 此外, MRSP-SW 算法的时间开销主要来自于循环嵌套, 其时间复杂度为 $O(N^2)$. 算法 MCRP-Tree 主要采用 FP 树的结构来挖掘稀有模式, 其中构造 FP 树的时间复杂度为 $O(\log_2 N)$. 在此基础上, 该算法又为每个项目构造一个状态树, 其时间复杂度为 $O(N \log_2 N)$. 因此, MCRP-Tree 算法的时间开销为 $O(N(\log_2 N)^2)$. 此外, 从图 5 也可以看出, 本文所提算法的时间消耗要远小于 MRSP-SW 算法和 MCRP-Tree 算法.

综上所述, 双向遍历最小稀有模式挖掘算法 MRP-TB 的运行时间最少, 速度最快, 内存开销低, 综合性能均优于目前的对比算法.

5 总 结

模式挖掘是卫星智能服务中的一项常用技术, 然而大多数模式挖掘算法都是关于频繁模式的算法, 很少涉及稀有模式挖掘. 现存的几种稀有模式挖掘方法无法对卫星遥测数据流进行实时处理, 它们或者采用自顶向下的遍历方法, 或者使用自底向上的遍历方法, 挖掘速度较慢, 效率低. 此外, 现有的稀有模式挖掘算法通常在生成全部候选集后再进行剪枝, 直至找出所有的稀有模式. 这种策略不仅浪费了大量的时间, 而且有可能面临组合爆炸. 针对上述问题, 本文提出一种无需专业领域知识的最小稀有模式挖掘算法. 首先, 将主观参数滑动窗口尺寸客观化, 在此基础上对卫星遥测数据流进行分段; 然后计算模式的支持度, 找出“顶部”和“底部”; 最后把自顶向下和自底向上的遍历技术相结合设计了双向遍历方法, 提出了一种高效的最小稀有模式挖掘算法. 通过卫星遥测数据流进行的大量实验表明, 该算法可以从遥测数据流中挖掘出卫星运行的最小稀有模式. 大量对比实验结果表明, 本文所提算法在效率上均优于对比算法.

在实验过程中, 我们发现如果挖掘对象是有限的静态数据, 而不是无限的数据流, 那么同比之下, 不设置滑动窗口需要几十毫秒的运行时间, 设置不重叠的滑动窗口需要几百毫秒的运行时间; 而设置步长为 1 的可重叠滑动窗口, 则需要几百秒的时间才可以得出结果. 所以滑动窗口的尺寸和步长设置非常重要, 一定要结合实际研究对象. 在下一阶段的任务中, 我们将研究参数优化方法, 结合卫星遥测数据流实际情况, 研究参数自主优化方法, 寻找步长和尺寸的最优解, 从而改进算法以获得更高效的结果.

参 考 文 献

[1] Wen Jing, Wang Da-Ming, Meng Yue-Yue, et al. Application of Beidou navigation satellite system to geological survey. Journal of Geomechanics, 2012, 18(3): 213-223(in Chinese) (温静, 汪大明, 孟月玥等. 北斗卫星导航系统在地质调查领域应用综述. 地质力学学报, 2012, 18(3): 213-223)

[2] Gu Sheng, Wei Jiao-Long, Pi De-Chang. Particle swarm optimization-fuzzy support vector machine based prediction of spacecraft parameters. Journal of Astronautics, 2014, 35(11): 1270-1276(in Chinese) (顾胜, 魏蛟龙, 皮德常. 一种粒子群模糊支持向量机的航天器参量预测方法. 宇航学报, 2014, 35(11): 1270-1276)

- [3] Dai Cheng-Long, Pi De-Chang, Fang Zhen, et al. A long-term lifetime prediction method for hemispherical resonator gyroscope. *Journal of Astronautics*, 2015, 36(1): 109-116 (in Chinese)
(代成龙, 皮德常, 方针等. 半球谐振陀螺仪寿命的一种长周期预测方法. *宇航学报*, 2015, 36(1): 109-116)
- [4] Henzinger M R, Raghavan P, Rajagopalan S. Computing on data streams//James M, Vitter J S, et al, eds. *External Memory Algorithms*. Boston, USA: American Mathematical Society, 1998: 107-118
- [5] Hu Jun-Zhe, Zhou Li-Zhu, Shi Jing. Research of random I/O schedule of tertiary device in massive storage management. *Computer Engineering & Applications*, 2002, 38(12): 70-73 (in Chinese)
(胡俊哲, 周立柱, 石晶. 三级存储设备随机调度研究. *计算机工程与应用*, 2002, 38(12): 70-73)
- [6] Han Meng, Wang Zhi-Hai, Ding Jian. Efficient decision tree for evolving data streams based on frequent patterns. *Chinese Journal of Computers*, 2016, 39(8): 1541-1554(in Chinese)
(韩萌, 王志海, 丁剑. 一种频繁模式决策树处理可变数据流. *计算机学报*, 2016, 39(8): 1541-1554)
- [7] Pang Jingyue, Liu Datong, Peng Yu, et al. Intelligent pattern analysis and anomaly detection of satellite telemetry series with improved time series representation. *Journal of Intelligent & Fuzzy Systems*, 2018, 34(6): 3785-3798
- [8] Ouyang W. Mining rare sequential patterns in data streams with a sliding Window//*Proceedings of the International Conference on Systems and Informatics*. Hangzhou, China, 2017: 1023-1027
- [9] Han Yanbo, Liu Chen, Su Shen, et al. A proactive service model facilitating stream data fusion and correlation. *International Journal of Web Services Research*, 2017, 14(3): 1-16
- [10] Han Yanbo, Zhu Meiling, Liu Chen. A service-oriented approach to modeling and reusing event correlations//*Proceedings of the 2018 IEEE 42nd Annual Computer Software and Applications Conference*. Tokyo, Japan, 2018: 498-507
- [11] Deng Shuiguang, Huang Longtao, Hu Daning, et al. Mobility-enabled service selection for composite services. *IEEE Transactions on Services Computing*, 2016, 9(3): 394-407
- [12] Zomaya A, Deng Shuiguang, Huang Longtao, et al. Toward mobile service computing: Opportunities and challenges. *IEEE Cloud Computing*, 2016, 3(4): 32-41
- [13] Han Meng, Wang Zhi-Hai, Yuan Ji-Dong. Efficient method for mining closed frequent patterns from data streams based on time decay model. *Journal of Astronautics*, 2015, 38(7): 1473-1483(in Chinese)
(韩萌, 王志海, 原继东. 一种基于时间衰减模型的数据流闭合模式挖掘方法. *计算机学报*, 2015, 38(7): 1473-1483)
- [14] Wu You-Xi, Zhou Kun, Liu Jing-Yu, et al. Mining sequential patterns with periodic general gap constraints. *Journal of Astronautics*, 2017, 40(6): 1338-1352(in Chinese)
(武优西, 周坤, 刘靖宇等. 周期性一般间隙约束的序列模式挖掘. *计算机学报*, 2017, 40(6): 1338-1352)
- [15] Kang Xu, Pi Dechang. A data-driven method of health monitoring for spacecraft. *Aircraft Engineering and Aerospace Technology*, 2018, 90(2): 435-451
- [16] Lei Yu, Li Man, Hu Wei-Song, et al. Efficient methods for rare sequential pattern mining. *Journal of Frontiers of Computer Science and Technology*, 2015, 9(4): 429-437(in Chinese)
(雷雨, 李曼, 胡卫松等. 高效的稀有序列模式挖掘方法. *计算机科学与探索*, 2015, 9(4): 429-437)
- [17] Hemalatha C S, Vaidehi V, Lakshmi R. Minimal infrequent pattern based approach for mining outliers in data streams. *Expert Systems with Applications*, 2015, 42(4): 1998-2012
- [18] Borah A, Nath B. Mining rare patterns using hyper-linked data structure//*Proceedings of the International Conference on Pattern Recognition and Machine Intelligence*. Cham, Switzerland, 2017: 467-472
- [19] Bhatt U Y, Patel P A. An effective approach to mine rare items using maximum constraint//*Proceedings of the 2015 IEEE 9th International Conference on Intelligent Systems and Control*. Coimbatore, India, 2015: 1-6
- [20] Chang J H, Lee W S. A sliding window method for finding recently frequent itemsets over online data stream. *Journal of Information Science & Engineering*, 2004, 20(4): 753-762
- [21] Zhang Lu-Min, Jia Yan, Zhou Bin, et al. Online bursty events detection based on emoticons. *Chinese Journal of Computers*, 2013, 36(8): 1659-1667(in Chinese)
(张鲁民, 贾焰, 周斌等. 一种基于情感符号的在线突发事件检测方法. *计算机学报*, 2013, 36(8): 1659-1667)
- [22] Chen Jing. Similarity Measure of Time Series for Satellite Telemetry Data [M. S. dissertation]. Harbin Institute of Technology, Harbin, 2015(in Chinese)
(陈静. 卫星遥测数据的时间序列相似性度量方法研究[硕士学位论文]. 哈尔滨工业大学, 哈尔滨, 2015)
- [23] Li Wei-Zheng, Meng Qiao. Fault detection for in-orbit satellites using an adaptive prediction model. *Chinese Journal of Space Science*, 2014, 34(2): 201-207(in Chinese)
(李维铮, 孟桥. 基于遥测数据动态特征的卫星异常检测方法. *空间科学学报*, 2014, 34(2): 201-207)
- [24] Yun S K, Rountree N. Finding sporadic rules using Apriori-Inverse//Ho T B, Cheung D, Liu H, eds. *Advances in Knowledge Discovery and Data Mining*. Berlin, Germany: Springer-Verlag, 2005: 97-106
- [25] Adda M, Wu L, Feng Y. Rare itemset mining//*Proceedings of the International Conference on Machine Learning and Applications*. Oregon, USA, 2007: 73-80
- [26] Tsang S, Yun S K, Dobbie G. RP-Tree: Rare pattern tree mining//*Proceedings of the International Conference on Data Warehousing and Knowledge Discovery*. Toulouse, France, 2011: 277-288
- [27] Huang D, Yun S K, Dobbie G. Rare pattern mining on data streams//*Proceedings of the International Conference on Data Warehousing and Knowledge Discovery*. Berlin, Germany, 2012: 303-314

[28] Liu R, Yang K, Sun Y, et al. Spark-based rare association rule mining for big datasets//Proceedings of the IEEE International Conference on Big Data. Boston, USA, 2017: 2734-2739

[29] Li Guo-Hui, Chen Hui. Mining the frequent patterns in an arbitrary sliding window over online data streams. Journal of Software, 2008, 19(10): 2585-2596(in Chinese)
(李国徽, 陈辉. 挖掘数据流任意滑动时间窗口内频繁模式. 软件学报, 2008, 19(10): 2585-2596)

[30] Kang Xu, Pi De-Chang, Tian Hua-Dong. An anomaly detection method based on angle deviation for satellite subsystem. Journal of Astronautics, 2017, 38(6): 638-646(in Chinese)
(康旭, 皮德常, 田华东. 一种基于角度偏离的卫星分系统异常检测方法. 宇航学报, 2017, 38(6): 638-646)

[31] Dong Xin, Pi Dechang. An effective method for mining quantitative association rules with clustering partition in satellite telemetry data//Proceedings of the 2014 2nd International Conference on Advanced Cloud and Big Data. Huangshan, China, 2014: 26-33

[32] Bao Jun-Peng, Yang Ke, Zhou Jing. Node level parallel and optimization of satellite data mining. Journal of Beijing University of Aeronautics and Astronautics, 2018. <https://doi.org/10.13700/j.bh.1001-5965.2018.0334>(in Chinese)
(鲍军鹏, 杨科, 周静. 卫星数据挖掘节点级并行与优化方法. 北京航空航天大学学报, 2018. <https://doi.org/10.13700/j.bh.1001-5965.2018.0334>)



ZHOU Zhong-Yu, M. S. candidate. His research interests include data mining and machine learning.

PI De-Chang, Ph. D., professor, Ph. D. supervisor. His research interests include data mining and big data management and analysis.

Background

The research topic of this paper mainly solves the problem of pattern mining in the field of data mining. Pattern mining is an important technology in satellite intelligent service system, mainly including frequent pattern mining and rare pattern mining. So far, most of the researches on pattern mining in the world are about frequent pattern mining. However, there are relatively few researches on rare pattern mining. Therefore, the development history and algorithm maturity of rare pattern mining are far less than that of frequent pattern mining.

The traditional rare pattern mining mainly aims at static data objects. In order to discover all the rare patterns, the strategy of generating all candidates is usually adopted, but which may be leading to combinatorial explosion and the speed of algorithm is slower. With the development of big data and cloud platform technologies, traditional storage devices have not been able to keep up with the speed of data transmission. Therefore, if static data is still used as research object, the storage device may easily lose data when storing data. In order to overcome these challenges, we propose a method to mine minimal rare pattern from satellite data streams. This method can perform segmented mining on dynamic data streams, so that we can obtain real-time results

feedback without causing data loss. We replace rare pattern with the minimal rare pattern that can characterize all rare patterns, so that the proposed method does not need to generate all candidates, and does not also need to mine all rare patterns, so that combinatorial explosion can be avoided and the operation efficiency of the algorithm can be improved. In the structure of the algorithm, the proposed algorithm in this paper uses bidirectional traversal technology which combines top-down and bottom-up method to search for candidates, which further speeds up the algorithm. We have successfully used the above method to mine the minimal rare patterns from satellite data streams. However, the specified value of the size of the sliding window and the step size may not be optimal solutions, and it needs further study. In this research, the hidden information of satellite data streams can be mined from a small part of the data that has been ignored, so the health of satellites can be detected in real time and the life span of satellites can also be increased.

This research is supported by the National Natural Science Foundation of China under Grant No. U1433116, and the Fundamental Research Funds for the Central Universities under Grant No. NP2017208.