

doi: 10.7690/bgzd.2016.06.011

## 装备故障的时空共现模式挖掘

杨 乐, 包 磊

(海军工程大学电子工程学院指挥信息系统系, 武汉 430033)

**摘要:** 针对装备故障数据的时空特性, 给出一种基于 apriori 算法的快速挖掘算法。从装备故障全局的角度出发, 定义装备故障时空共现模式, 模式中引入故障实例空间参与率、候选同位模式参与指数、故障类型时域参与度、故障类型时域参与指数等描述指标, 并通过仿真分析比较快速挖掘算法与朴素挖掘算法的执行效率。仿真结果表明: 当装备故障时空数据量较大且含较高噪声时, 所提出的快速挖掘算法有更高的执行效率。

**关键词:** 装备故障; 空间同位模式; 时空共现模式; 时空数据挖掘; apriori 算法

**中图分类号:** TP311 **文献标志码:** A

## Mining Spatiotemporal Co-occurrence Pattern in Equipment Failure

Yang Le, Bao Lei

(Department of Command & Control, College of Electronic Engineering, Naval University of Engineering, Wuhan 430033, China)

**Abstract:** For spatiotemporal characteristics of the equipment failure data, a fast miner algorithm is proposed based on apriori algorithm. From a global perspective of equipment failure, equipment failure spatiotemporal co-occurrence pattern is defined. The pattern is described with several parameters, which are failure-instances spatial participation ratio (SPR), candidate spatial co-location pattern participation index (SPI), failure-types temporal participation degree (TPD) and failure-types temporal participation index (TPI). By simulating, analyzing and comparing with execution efficiency between the fast miner algorithm and naive miner algorithm, we find that the fast miner algorithm has higher execution efficiency when the failure equipment data is large and contains much more noise.

**Keywords:** equipment failure data; spatial co-location pattern; spatiotemporal co-occurrence pattern; spatiotemporal data mining; apriori algorithm

### 0 引言

随着科学技术的飞速发展, 武器装备得以不断更新, 多样化、复杂化、体系化、智能化成为武器装备的新特征。装备故障是武器装备使用过程中不可避免的现象, 鉴此, 对其进行高效快速的故障诊断, 以保持良好的战斗状态, 具有极其重要的意义。然而, 对于海量的武器故障数据仅靠维修人员使用传统的方法进行故障诊断与排除具有较大的局限性。加之, 装备体系化建设下, 各型装备间千丝万缕的关系致使装备故障呈现关联特征, 传统的故障诊断方法根本无法彻底排除关联故障。

装备故障的发生不仅与外界因素有关联(例如地理环境、天文环境、人为因素等), 故障与故障之间在时间和空间上也呈某种共现性, 某种装备故障的发生往往伴随着另一种或者多种装备故障的发生。例如, 美国海军曾经出现侦察飞机与水面舰艇距离接近某一阈值时, 由于电子干扰导致通信设备与导航设备同时出现故障。将数据挖掘技术引入装备故障诊断领域, 为装备故障诊断提供了一种新的

方法。据笔者所知, 围绕装备故障数据挖掘, 前人的工作主要集中于: 针对单一装备系统内的装备故障进行关联分析, 如舰艇故障<sup>[1]</sup>、导航装备故障<sup>[2]</sup>、通信装备故障<sup>[3]</sup>等。此类研究未能从装备故障全局角度出发, 摒弃单一装备系统对于装备故障数据的限制, 进行装备故障数据挖掘。同时, 随着武器装备的发展, 装备故障越来越具有时空特性, 大量的装备故障数据实属时空数据的范畴。针对上述问题, 笔者从装备故障全局的角度出发, 结合装备故障时空特性, 将时空数据挖掘领域的时空共现模式挖掘<sup>[4]</sup>引入装备故障领域, 给出一种基于经典 apriori 算法的快速挖掘算法, 并设计了仿真数据集。仿真结果表明: 该算法能高效的挖掘装备故障时空共现模式, 可从海量的装备故障实例中找出有价值的共现模式。

### 1 装备故障时空共现模式

装备故障时空共现模式, 是指 2 种或者以上装备故障, 在时间和空间上处于邻近。装备故障时空共现模式挖掘是从海量的战场装备故障时空数据

收稿日期: 2016-02-04; 修回日期: 2016-03-24

作者简介: 杨 乐(1991—), 男, 陕西人, 在读硕士, 从事领域为时空数据挖掘研究。

中,挖掘时间和空间上处于邻近的装备故障模式集,继而结合相关先验知识快速诊断导致装备故障发生的原因。通过挖掘和分析装备故障时空共现模式,有助于发现某些装备故障发生的自身潜在原因,多种装备故障发生的潜在关系,造成装备故障时空共现的共同外界因素等,从而提高装备保障的效率,对于装备故障时空共现模式的挖掘与研究,具有重要的价值与意义。

为方便后文模型建立与算法介绍时更加直观,笔者首先给出装备故障时空数据示例集。在图 1 中,A、B、C 和 D 分别代表一种装备故障类型;A1、A2、B2 等分别代表某种故障类型的实例;用实线相连的实例表示空间上满足邻近距离  $R$ ;数据集由 5 个时间槽组成。

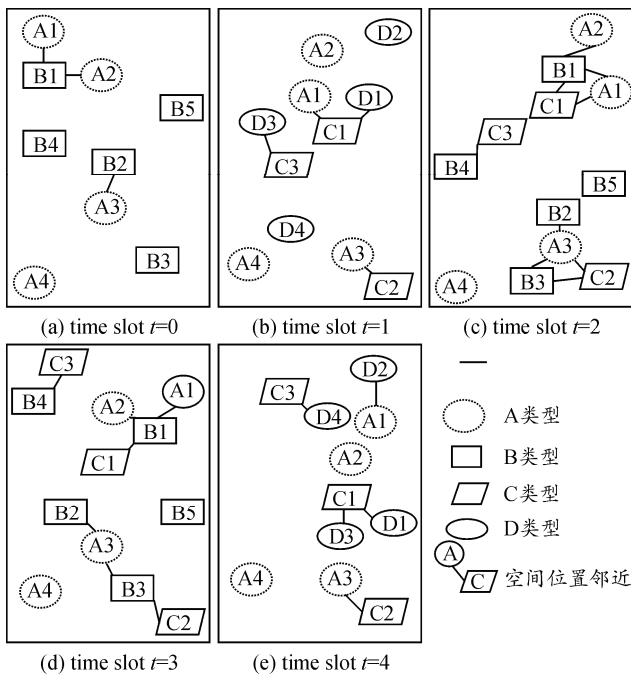


图 1 装备故障时空数据示例集

### 1.1 装备故障空间同位模式

定义 1 故障实例空间参与率 (failure-instances spatial participation ratio, SPR):

$$SPR(C_k, f_i) = \frac{C_k \text{ 中 } f_i \text{ 不同故障实例的数量}}{f_i \text{ 故障实例总数量}} \quad (1)$$

其中  $C_k = \{f_0, \dots, f_{k-1}\}$  为  $k$  元装备故障候选同位模式,  $f_i \in E$ ,  $E$  为包含  $n$  个空间装备故障类型的集合。以图 1 时间槽  $t=0$  中装备故障数据集为例,空间故障实例参与率计算结果见表 1。

表中,  $SPI\{A,B\}=2/5$ 。由于  $k$  元候选装备故障同位模式  $C_k$  中包含  $k$  种故障类型;因此,产生  $k$  个空间故障实例参与率 SPR。Shashi Shekhar 等将  $k$

个 SPR 中的最小值作为候选同位模式的参与指数 (candidate spatial co-location pattern participation index, SPI)<sup>[5]</sup>。

表 1 空间实例参与率

同位实例集	$\{A_1, B_1\}$ $\{A_2, B_1\}$ $\{A_3, B_2\}$
SPR( $C_2, A$ )	3/4
SPR( $C_2, B$ )	2/5

定义 2 候选同位模式参与指数:

$$SPI(C_k) = \min_{f_i \in C} \{SPR(C_k, f_i)\} \quad (2)$$

如果满足公式 (3), 则该候选装备故障同位模式即为同位模式, 公式中  $\theta_s$  表示空间阈值。假设  $\theta_s=0.5$ ,  $2/5$  小于  $0.5$ , 因此  $\{A,B\}$  并非同位模式。

$$SPI(C) \geq \theta_s \quad (3)$$

### 1.2 装备故障时空共现模式

装备故障时空共现模式是在空间同位模式的基础上引入时间维信息而建立的。时空数据集就是将多个空间数据集以某种时间顺序结合在一起, 用时间槽概念来表示, 一个时间槽表示某一时刻的空间数据集。后文时空共现模式的相关概念也是在空间同位模式定义基础上定义的。

定义 3 故障类型时域参与度 (failure-types temporal participation degree, TPD): 在时域数据集中, 某故障类型出现的时间槽数就是该故障类型的时域参与度。

定义 4 故障类型时域参与指数 (failure-types temporal participation index, TPI):

$$TPI(C_k) = \frac{C_k \text{ 出现的时间槽数量}}{\max \sum_{i=0}^{k-1} \{\text{Lifetime}(C_k, f_i)\}} \quad (4)$$

若满足下式, 则  $C_k$  为  $k$  元时空共现模式:

$$TPI_{m \in TF}(C_k) \geq \theta_t \quad (5)$$

其中:  $\theta_t$  为时间阈值;  $TF=[T_0, \dots, T_{n-1}]$  表示  $n$  个时间槽。以图 1 时空数据集为例, 设定  $\theta_s=\theta_t=0.5$ , 模式  $\{B,C\}$  统计结果如表 2 所示。

表 2 同位模式出现时间槽统计

时间槽	$\{B,C\}$	时间槽	$\{B,C\}$
TF=0	0	TF=3	1
TF=1	0	TF=4	0
TF=2	1	TPI	2/4

注: “1” 表示模式  $\{B,C\}$  在该时间槽中属于同位模式; “0” 表示模式  $\{B,C\}$  在该时间槽中不属于同位模式。

由此可见, 二元装备故障候选时空共现模式  $\{B,C\}$  为装备故障时空共现模式。

## 2 装备故障时空共现模式挖掘算法

本节介绍 2 种装备故障时空共现模式挖掘算法: 朴素挖掘算法和快速挖掘算法。快速挖掘算法

改善了朴素挖掘算法在大数据、高噪声条件下挖掘复杂度高的问题，有效地缩短了挖掘执行时间，提高了挖掘效率。

### 2.1 朴素挖掘算法

对于装备故障时空共现模式的挖掘，朴素挖掘

算法分 2 步进行：第 1 步，挖掘所有时间槽中各元装备故障同位模式；第 2 步，剪枝不满足时间阈值的模式，产生装备故障时空共现模式。以图 1 装备故障时空数据集为例数据，展示朴素挖掘算法的挖掘过程，如图 2 所示。

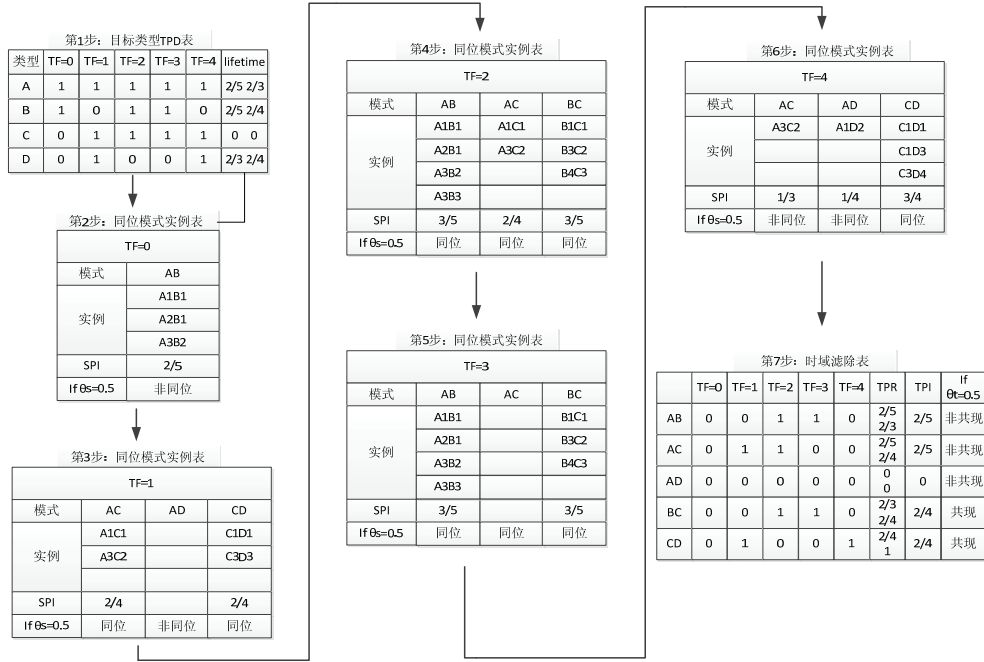


图 2 朴素算法挖掘过程

算法 1: Naive Miner Algorithm。

算法的输入有：类型集  $E$ ，时空数据集  $ST \langle \text{类型号, 实例号, } x \text{ 坐标, } y \text{ 坐标, 时间槽号、区域号} \rangle$ ，实例之间的邻近距离  $R$ ，时间框架  $TF = \{T_0, T_1, \dots, T_{n-1}\}$ ，区域框架  $AF = \{I, II, \dots, N\}$ ，空间阈值  $\theta_s$ ，时间阈值  $\theta_t$ 。

输出为各个区域满足空间邻近距离  $R$  和时空频繁阈值的正确并完整的部分时空共现模式集。

变量的符号及含义：共现模式长度  $k$ ；时间槽  $t \in \{T_0, T_1, \dots, T_{n-1}\}$ ；区域框架  $i \in \{I, II, \dots, N\}$ ； $T_k$  为  $k$  元共现模式实例集； $C_k$  为  $k$  元候选共现模式类型集； $SC_k$  为  $k$  元空间同位模式集； $PAP_t$  为  $t$  时间槽中空间同位模式集； $PAP$  为  $k$  元时空共现模式集。

伪代码：

- 1) **divide** area
- 2) **for each** area range  $i$  in  $\{I, II, \dots, N\}$
- 3) {
- 4) **initialization**:  $k=1, C_k=E, T_k=ST, PAP_k(0)=ST$
- 5) **for each** time\_slot  $t$  in  $\{T_0, T_1, \dots, T_{n-1}\}$
- 6)  $C_{k+1} = \text{gen\_candidate\_co\_occ}(C_k)$
- 7)  $T_{k+1} = \text{gen\_instance}(C_{k+1}, T_k, R)$
- 8) **While** (not empty  $C_{k+1}$ )

- 9) {
- 10)  $SC_{k+1} = \text{discover\_spatial\_index}(C_{k+1}, T_{k+1}, \theta_s)$
- 11)  $PAP_t = [PAP_t; SC_{k+1}]$
- 12)  $C_{k+2} = \text{gen\_candidate\_co\_occ}(SC_{k+1})$
- 13)  $T_{k+2} = \text{pro\_instance\_co\_occ}(C_{k+2}, T_{k+1})$
- 14) }
- 15)  $PAP = \text{union}\{PAP_2, \dots, PAP_t\}$
- 16)  $PAP = \text{discover\_temporal\_pre\_pattern}(PAP, \theta_t)$
- 17) }

第 6 步：gen\_candidate\_co\_occ 函数，产生  $k+1$  元候选同位模式集。该函数在所有  $k$  元同位模式集的基础上，使用 apriori 算法产生  $k+1$  元候选同位模式集  $C_{k+1}$ 。

第 7 步：gen\_instances\_co\_occ 函数，在每个时间槽内产生满足临近关系  $R$  的 2 元实例集。

第 10 步：discover\_spatial\_index 函数，通过比较候选空间同位模式参与指数与空间阈值  $\theta_s$  的大小，滤除不满足空间阈值条件的候选空间同位模式，获得  $k+1$  元空间同位模式。

第 12 步：gen\_candidate\_co\_occ 函数，产生  $k+2$  元候选同位模式集。

第 13 步：pro\_instance\_co\_occ 函数，产生  $k+2$

元同位实例集。

第 15 步：产生候选共现模式集。

第 16 步：discover\_temporal\_prev\_pattern 函数，滤除不满足时间阈值  $\theta_t$  的候选共现模式。

### 2.2 快速挖掘算法

真实环境中装备故障时空数据量大且往往伴有较高噪音，较低的算法复杂度显得尤为重要。朴素挖掘算法计算复杂度大，挖掘效率低的缺点随数据

量增大及噪音增加逐渐显现。快速挖掘算法的提出很好地降低了挖掘的复杂度，提高算法挖掘效率。

快速挖掘算法较朴素挖掘算法的优点在于，通过提前进行时域滤除，尽早剪枝必然不满足时间阈值的候选共现模式，从而大大减少了后续时间槽中要挖掘的候选共现模式集数量，降低了挖掘复杂度，提高了算法挖掘效率。以图 1 装备故障时空数据集为例，详细挖掘过程如图 3 所示。

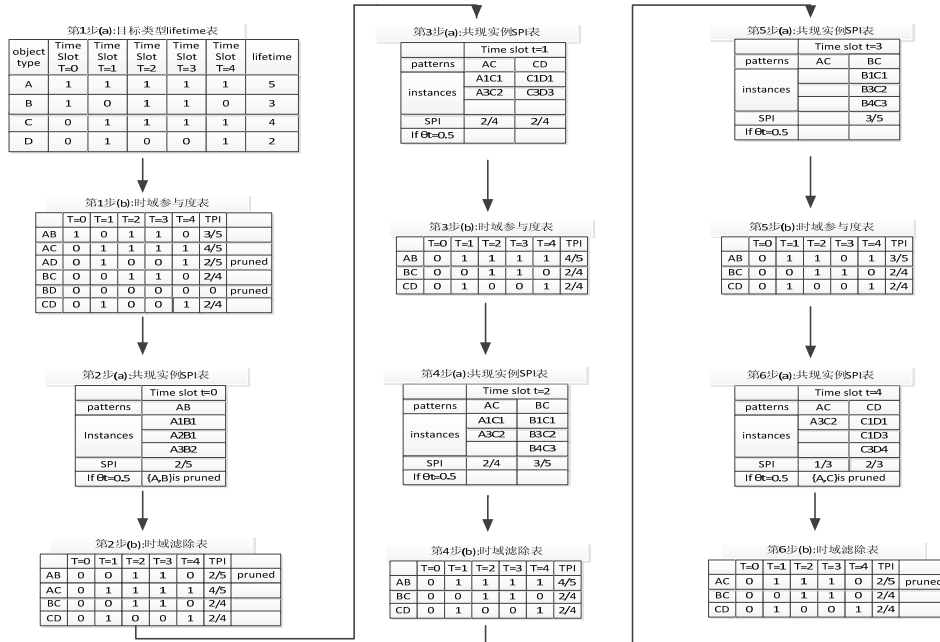


图 3 快速算法挖掘过程图

算法 2: Fast Miner Algorithm.

算法的输入有：类型集  $E$ ，时空数据集  $ST \langle$  类型号，实例号， $x$  坐标， $y$  坐标，时间槽号，区域号  $\rangle$ ，实例之间的邻近距离  $R$ ，时间框架  $TF = \{T_0, T_1, \dots, T_{n-1}\}$ ，区域框架  $AF = \{I, II, \dots, N\}$ ，空间阈值  $\theta_s$ ，时间阈值  $\theta_t$ 。

输出为各个区域满足空间邻近距离  $R$  和时空阈值的正确并完整的部分时空共现模式集。

变量的符号及含义：共现模式长度  $k$ ；时间槽  $t \in \{T_0, T_1, \dots, T_{n-1}\}$ ；区域框架  $i \in \{I, II, \dots, N\}$ ； $k$  元共现模式实例集  $T_k$ ； $k$  元候选共现模式类型集  $C_k$ ； $k$  元空间同位模式集  $SP_k$ ； $k$  元时间共现模式集  $TC_k$ ； $k$  元时空共现模式集  $PAP_k$ 。

伪代码：

- 1) **divide** area
- 2) **for each** area range  $i$  in  $\{I, II, \dots, N\}$
- 3) {
- 4) **initialization**:  $k=1, C_k=E, T_k=ST, PAP_k(0)=ST$
- 5)  $C_{k+1} = \text{gen\_candidate\_co\_occ}(E)$

- 6)  $T_{k+1} = \text{gen\_instances\_co\_occ}(C_{k+1}(t), T_k(t), R)$
- 7) **While**(not empty  $C_{k+1}$ )
- 8) {
- 9)  $TC_{k+1} = \text{calculate\_temporal\_prev\_index}(C_{k+1})$
- 10)  $C_{k+1} = \text{discover\_temporal\_prev\_pattern}(TC_{k+1}, \theta_t)$
- 11) **for each** time\_slot  $t$  in  $\{T_0, T_1, \dots, T_{n-1}\}$
- 12)  $SP_{k+1}(t) = \text{discover\_spatial\_prev\_patterns}(T_{k+1}(t), C_{k+1}(t), \theta_s)$
- 13)  $TC_{k+1} = \text{calculate\_temporal\_prev\_index}(SP_{k+1}(t))$
- 14)  $PAP_{k+1} = \text{discover\_PACOP\_prev\_patterns}(TC_{k+1}(t), \theta_t)$
- 15)  $T_{k+2} = \text{gen\_candidate\_co\_occ}(PAP_{k+1}, T_{k+1})$
- 16)  $C_{k+2}(t) = \text{update\_candidates}(PAP_{k+1}(t))$
- 17) }
- 18) **return** union  $X_i = \{PAP_2, \dots, PAP_{k+1}\}$
- 19) }

第 5 步: gen\_candidate\_co\_occ 函数，产生  $k+1$  元候选共现模式集。该函数在所有  $k$  元共现模式集的基础上，使用 apriori 算法产生  $k+1$  元候选共现模式集  $C_{k+1}$ 。

第 6 步: gen\_instances\_co\_occ 函数，在每个时

间槽中产生满足临近关系  $R$  的 2 元实例集。

第 9、13 步: calculate\_temporal\_prev\_index 函数, 计算各候选共现模式的 lifetime 值。

第 10 步: discover\_temporal\_prev\_pattern 函数, 滤除不满足时间阈值  $\theta_t$  的候选共现模式, 更新候选共现模式集  $C_{k+1}$ 。

第 12 步: discover\_spatial\_prev\_patterns 函数, 通过比较类型实例空间参与率与空间阈值  $\theta_s$  的大小, 滤除不满足空间阈值条件的候选空间同位模式, 获得空间同位模式。

第 14 步: discover\_PACOP\_prev\_patterns 函数, 运用时域阈值滤除不满足时间阈值条件的候选共现模式, 产生  $k+1$  元时空共现模式集。

第 15 步: gen\_candidate\_co\_occ 函数, 运用 apriori 算法在  $k+1$  元共现实例集  $T_{k+1}$  的基础上连接、剪枝产生  $k+2$  元共现实例集  $T_{k+2}$ 。

第 16 步: update\_candidates 函数, 运用 apriori 算法产生  $k+2$  元候选共现模式集  $C_{k+2}$ 。

### 3 仿真与分析

#### 3.1 仿真数据的产生

在 Huang 等<sup>[6]</sup>提出的空间数据产生器的基础上, 笔者构造了装备故障时空数据集, 用于评价算法的优劣性。装备故障时空数据集的产生基于尺寸为  $D \times D$  的空间框架。出于简化的考虑, 数据集所对应的每个时间槽被分成规则的网格, 网格的边表示邻近距离  $R$ 。时空数据集构建的顺序: 首先人为设定装备故障类型数、各故障类型子集数、故障类型的 lifetime; 然后产生故障实例集; 最后随机产生一定量的噪音数据。分别对有无噪音的装备故障时空数据进行挖掘, 比较算法优劣性。

仿真数据参数设定及数据集统计结果见表 3。

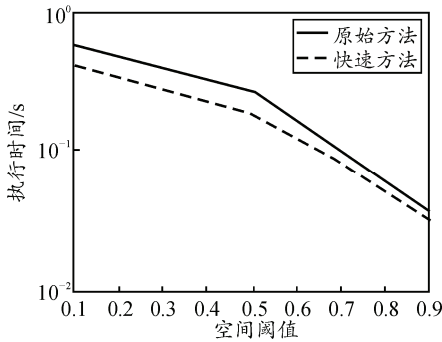
表 3 仿真数据参数设定及数据集统计结果

数据类别	邻近距离/R	目标类型数/E	时间槽数/TF	TPD	实例数				
					T=0	T=1	T=2	T=3	T=4
非噪音	10	10	2	3~5	470	390	360	150	100
噪音	10	10	2	3~5	120	130	50	120	0

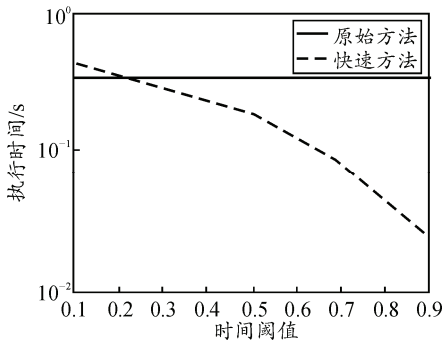
#### 3.2 仿真结果

1) 不加入噪声数据集。

基于不加入噪声数据集的仿真结果如图 4。



(a) 空间阈值  $\theta_s$  对执行时间影响趋势

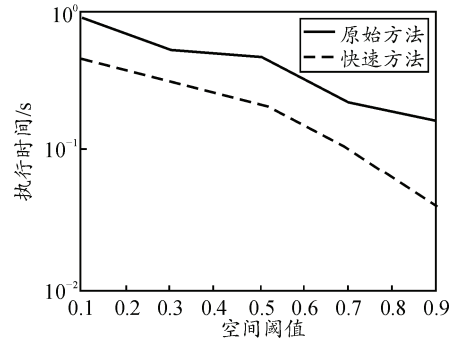


(b) 时间阈值  $\theta_t$  对执行时间影响趋势

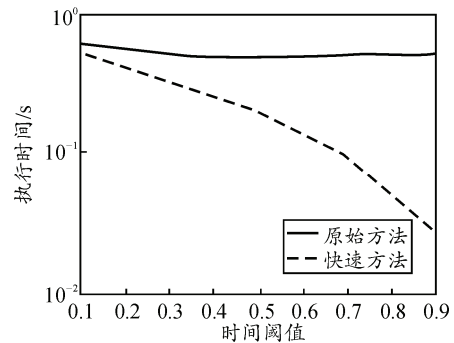
图 4 不加入噪声数据集

图 4(a)的参数设定: 时间阈值  $\theta_t=0.5$ , 时间槽

数 TF=5, 特征类型数  $E=10$ 。图 4(b)的参数设定: 空间阈值  $\theta_s=0.5$ , 时间槽数 TF=5, 特征类型数  $E=10$ 。



(a) 空间阈值  $\theta_s$  对执行时间影响趋势



(b) 时间阈值  $\theta_t$  对执行时间影响趋势

图 5 加入噪声数据集

2) 加入噪声数据集。

基于加入噪声数据集的仿真结果如图 5。



图 5(a)的参数设定：时间阈值  $\theta_t=0.5$ ，时间槽数  $TF=5$ ，特征类型数  $E=10$ 。图 5(b)的参数设定：空间阈值  $\theta_s=0.5$ ，时间槽数  $TF=5$ ，特征类型数  $E=10$ 。

图 4、图 5 反映了以下变化趋势：给定仿真数据，当空间阈值为定值时，随着时间阈值的增加，算法执行时间呈下降趋势；当时间阈值为定值时，随着空间阈值的增加，朴素算法执行时间几乎不变，而快速算法执行时间呈下降趋势。

增加数据量、加入噪音后，快速算法较朴素算法表现的更加优异，具体表现在：1) 时间阈值一定时，随空间阈值的增加，快速算法执行时间下降的趋势较朴素算法更加明显；2) 空间阈值一定时，快速算法的执行时间从开始就低于朴素算法，且随时间阈值的增加，快速算法的执行时间不断下降。综上所述：快速算法的算法复杂度低于朴素算法，并且随数据量及噪音的增加，这种优势越来越明显。

#### 4 结束语

笔者定义了装备故障时空共现模式，讨论了装备故障时空共现模式存在的潜在价值，给出了一种快速挖掘算法，详细分析了 2 种算法的挖掘过程及优劣性。最后构造了模拟仿真数据，用实验数据证

\*\*\*\*\*  
(上接第 45 页)

运行管理软件采用分布式控制系统思想，将风洞试验中的数据采集、数据处理和数据分析作为独立的子系统。一是简化了整个软件设计的难度；二是便于开发出比较完善的子系统软件，使各子系统运行更加稳定可靠；三是在试验运行中，个别子系统的故障不会造成整个测控系统的瘫痪，增强了系统的稳定性。

#### 3) DataSocket 编程技术的应用。

旋转天平试验局域网采用 NI 公司提供的 DataSocket 网络技术，这是一个基于 TCP/IP 工业标准的编程技术，对底层进行了高度封装，所提供的参数简单友好，简化了不同计算机应用程序的数据交换过程<sup>[6]</sup>。相对于以往的网络编程，该技术减少了重新编写局域网服务器程序的工作量，可以通过库函数直接调用 NI 开发环境提供的 DataSocket 服务器。考虑到 3 个子系统都要通过服务器进行指令传输和数据交换，故在运行管理和数采软件启动的同时都在本机上打开服务器，在软件关闭之前，服务器一直处于开启的状态，其他客户端可以随时联入。这种设计既不影响子系统独立操作，又兼顾

明了前文的分析结果。以后的工作将从 2 方面展开：1) 通过对时空数据集做区域划分处理，引入非装备故障因素的外界条件，寻找导致某些装备故障时空共现的潜在共同外界条件，从而改善装备保障质量与速度，提高武器装备战斗力；2) 研究开发更加高效、实用的挖掘算法。

#### 参考文献：

- [1] 董豆豆, 李登峰, 陈玉文. 基于关联规则的舰艇故障诊断数据挖掘系统结构框架[J]. 船舶工程, 2001(4): 61-64.
- [2] 刘勇, 胡柏青. 基于故障树的导航装备故障诊断专家系统研究[J]. 海军工程大学学报, 2002, 1(14): 41-44.
- [3] 郭玉鹏, 时和平, 巢蕾. 通信装备故障诊断贝叶斯网络[J]. 兵工自动化, 2011, 30(7): 79-81.
- [4] Celik M. Partial spatial-temporal co-occurrence patterns mining[J]. Information Systems and Communication Service, 2014, 44(1): 27-49.
- [5] Shashi Shekhar, Yan Huang. Discovering Spatial Co-location Patterns: A Summary of Results[M]. SSTD, 2001: 236-256.
- [6] Huang Y, Shekhar S, Xiong H. Discovering co-location patterns from spatial datasets: a general approach[J]. IEEE Trans Knowl Data Eng (TKDE), 2004, 16(12): 1472-1485.

了联网的情况，如果需要联网，其他系统只要通过面板控件与服务器进行连接即可。

#### 6 结束语

该测控系统由 3 个子系统组成，整个试验流程由一名用户通过中心调度方式即可完成，具有操作简便、稳定可靠的优点。试验应用情况表明：该系统抗干扰能力强，采集数据准确，有效地完成了测试任务，并提高了试验效率。

#### 参考文献：

- [1] 祝长江, 吕波, 虞择斌, 等.  $\phi 5\text{ m}$  立式风洞结构设计[J]. 实验流体力学, 2010, 24(2): 93-96.
- [2] 马军, 姜裕标, 梁鉴, 等.  $\phi 5\text{ m}$  立式风洞旋转天平试验装置研制[J]. 实验流体力学, 2012, 26(2): 77-80.
- [3] 程琴, 卢博友. 基于 DataSocket 的风机远程测试系统设计[J]. 微计算机信息, 2008, 24(9): 110-112.
- [4] 熊建军, 马军. MM440 变频器在弧形轨旋转控制系统中的应用[J]. 电工技术, 2011, 32(7): 39-41.
- [5] 吴侃, 于明礼, 李明. 基于 Matlab CUI 的超声电机控制系统串口通信研究[J]. 机电工程, 2015, 32(11): 1471-1476.
- [6] 蒋薇, 张晓波, 赖青贵. 基于 Labview 的仪器通信技术研究[J]. 计算机测量与控制, 2013, 21(4): 1030-1032.