文章编号: 1001-0920(2018)04-0591-09 DOI: 10.13195/j.kzyjc.2017.0212

基于频繁项集树的时态关联规则挖掘算法

王 玲†, 李树林, 徐培培, 孟建瑶, 彭开香

(1. 北京科技大学 自动化学院，北京 100083；2. 北京科技大学
工业过程知识自动化教育部重点实验室，北京 100083)

摘 要: 针对目前时态关联规则研究中存在的挖掘效率不高、规则可解释性低、未考虑项集时间关联关系等问题, 在原有相关研究的基础上,提出一种新的基于频繁项集树的时态关联规则挖掘算法. 通过对时间序列数据进行降维离散化处理,采用向量运算生成频繁项集,提高频繁项集挖掘效率. 考虑到项集之间的时态关系以及树结构的优势,提出一种新的频繁项集树结构挖掘时态关联规则,其挖掘频繁项集与树结构构建同时进行,无需产生候选项集,提高了规则挖掘效率. 实验表明,对比于其他算法,所提出算法在挖掘效率和规则解释性方面效果更好,具有较好的应用前景.

关键词: 向量运算；时态关系；频繁项集树；时态关联规则

中图分类号: TP311 **文献标志码:** A

Temporal association rules mining algorithm based on frequent item sets tree

WANG Ling†, LI Shu-lin, XU Pei-pei, MENG Jian-yao, PENG Kai-xiang

(1. School of Automation & Electrical Engineering, University of Science and Technology Beijing，Beijing 100083，China；2. Key Laboratory of Knowledge Automation for Industrial Processes of Ministry of Education, University of Science and Technology Beijing，Beijing 100083，China)

Abstract: In order to improve the efficiency and enhance the interpretability in mining the temporal association rules, a new mining algorithm of temporal association rules based on frequent iemsets tree is proposed. The time series data is discretized after the dimension reduction, on this basis, vector operations are adopted to generate frequent itemsets to improve the efficiency. In view of the advantage of the structure of the tree and the temporal interval relation between the items, a frequent itemsets tree is constructed in parallel with mining frequent itemsets to improve the efficiency of rule mining without generating candidate itemsets. Experimental results show that the proposed algorithm can provide better efficiency and interpretability in mining temporal rules in comparison with other algorithms and has good application prospects.

Keywords: vector operation; temporal relationship; frequent itemsets tree; temporal association rules

0 引 言

在现实世界中,大多数信息都包含时间属性,如股票的每日波动、超市的交易、含时间的实验数据等,它们也被称为时态数据,即随时间变化且在不同时间状态上是相互关联的数据. 事实上,时态数据挖掘已经有了许多研究成果,其中时态关联规则是加了时间约束的关联规则,目的是找出时态事务集中同一维属性与时间之间的关联,以及基于时域的不同维属性之间的关系等,更好地挖掘隐藏在数据中的与时间关联的知识.

目前,时态关联规则的算法已有一些研究成果,文献[1]提出了一种基于时间区间延展与归并技术的时态关联规则发现算法,文献[2]利用具有时间约束的频繁项集挖掘关联规则. 然而,这些算法在挖掘频繁项集时均利用了Apriori算法,会产生大量的候选项集而降低运行效率. 为了提高挖掘的效率,Schlüter

收稿日期: 2017-03-04；**修回日期:** 2017-09-22.

基金项目: 国家自然科学基金项目(61572073)；中央高校基本科研业务费专项基金项目(FRF-UM-15-052)；北京科技大学研究生教育发展基金项目(230201506400060).

责任编委: 赵珺.

作者简介: 王玲(1974−), 女, 副教授, 博士, 从事数据挖掘、机器学习等研究；李树林(1990−), 男, 硕士生, 从事数据挖掘的研究.

†通讯作者. E-mail: lingwang@ustb.edu.cn

等[3]提出了两种树结构挖掘方法来发现多种时态关联规则. Yin等[4]设计了一种同时考虑全局和局部频繁模式的策略,不用借助先验知识便能够自动生成所有时态区间实现挖掘时态关联规则. 文献[5]考虑项集不同有效时间和FP-树的优势,提出了一种基于FP-树的时态关联规则分区挖掘方法. 文献[6]引入事务中项的权重因子,利用FP-树结构挖掘时态关联规则. 文献[7]引入压缩树结构来挖掘规则,利用模式生长方法生成所有相关时态模式,只需扫描一遍数据集. 然而,上述方法没有考虑多个项之间的时态关系,不能对时态数据中的隐藏信息进行充分挖掘,得到的规则可解释性较差.

文献[8]提出了一种基于压缩FP树且分而治之的多时间序列跨事务关联规则挖掘算法,利用规则前后件项集的时间差进行预测,规则可解释性较好,但仅适用于离散时态数据,且结果易受参数影响. 文献[9]结合模糊理论,实现连续时态数据的模糊离散化,根据项集及其生命周期发现具有时态约束的模糊关联规则,相比其他方法更具适用性,但结果受模糊参数影响较大. 文献[10]提出了基于两个项之间的时态关系挖掘时态关联规则的方法,但不能获得多个项之间的时态关系. 针对多个项之间的时态关系,Chen等[11]提出了基于区间数据发现闭合时态模式的CEMiner算法,并利用优化技术有效地减少了搜索空间. Ruan等[12]提出了能够在大规模区间型时态数据上并行、定量挖掘序列模式的框架. 这两种方法只涉及了时态模式挖掘,没有获得时态关联规则. 文献[13]中,时态模式表示为基于它们与集合中其他事件的关系而链接的事件集合,所得到的频繁模式用于生成时态关联规则,但由此产生的规则前件和后件的时态约束是相同的时间间隔,没有考虑时间间隔不同步的问题.

针对上述算法中存在的问题,本文提出一种基于频繁项集树的时态关联规则挖掘算法(Temporal association rule mining algorithm based on frequent item sets tree, T-FS-tree),通过对时间序列数据进行降维离散化处理,采用向量运算生成频繁项集,避免对数据库的重复扫描. 为了挖掘多元时间序列之间的时态区间关联规则,结合项集之间的多种时态关系,提出一种新的频繁项集树结构,使得频繁项集挖掘与树结构的构建同时进行,避免了候选项集的产生且提高了时态关联规则生成的效率. 实验表明,对比于其他算法,所提出算法在挖掘效率和规则解释性方面效果更好,具有较好的应用前景.

1 问题描述及相关定义

假定多元时间序列的集合$S=\{s_1,s_2,\cdots,s_i,\cdots,s_n\}(1\leqslant i\leqslant n)$,其中$s_i$为多元时间序列集合中第$i$个属性的时间序列. d_j为时刻j对S的观察值集合,$d_j=\{s_1(j),s_2(j),\cdots,s_i(j),\cdots,s_n(j)\}(1\leqslant j\leqslant m)$. 多元时间序列事务集$D$定义为$D=\{d_1,d_2,\cdots,d_j,\cdots,d_m\}$. 时态关联规则挖掘的关键步骤是获取模式序列的频繁模式. 为了将相同形态不同斜率和波动幅度的属性模式归为同一类,需要对时间序列进行离散化聚类处理.

定义1 (项集的离散化表示) 时间序列s_i的模式序列为$M_i=\{I_{i1},I_{i2},\cdots,I_{iq_i}\}$,元模式$I_{iq_i}=(\bar{k}_{iq_i},\Delta\bar{y}_{iq_i},\Delta t_{iq_i})$为模式序列$M_i$的项,是时间序列$s_i$中第$q_i$个聚类的离散化项表示,$\bar{k}_{iq_i}$、$\Delta\bar{y}_{iq_i}$和$\Delta t_{iq_i}$分别为时间序列$s_i$中第$q_i$个聚类所含所有子时间序列段的斜率均值、幅值变化均值和时间均值.

定义2 (项集的生命周期) 假定在时间区间$[T_1,T_2]$中,通过时间序列离散化聚类处理,形成离散事务相关的离散时态事务数据集为

$$D_z^{[T_1,T_2]}=\{\mathrm{TID}_1,\mathrm{TID}_2,\cdots,\mathrm{TID}_j,\cdots,\mathrm{TID}_N\},\quad 1\leqslant j\leqslant N,$$

离散事务数也可记作$|D_z^{[T_1,T_2]}|$. 实际上,离散时态事务数据集被划分为N个时间分区,分区$j(1\leqslant j\leqslant N)$对应编号为$\mathrm{TID}_j$的离散事务,记$\mathrm{TID}_j=\langle I_{\mathrm{TID}_j},[T_b^{\mathrm{TID}_j},T_e^{\mathrm{TID}_j}]\rangle$是一个二元组,$I_{\mathrm{TID}_j}$表示第$j$条离散时态事务$\mathrm{TID}_j$所含项集,对应于多元时间序列数据集$S$的模式序列项向量集合为$M_Z=\{I_{11},I_{12},\cdots,I_{1q_1},\cdots,I_{j1},I_{j2},\cdots,I_{jq_j},\cdots,I_{n1},I_{n2},\cdots,I_{nq_n}\}$,$I_{\mathrm{TID}_j}$仅包含了某些时间序列的特定聚类离散化项. $I_{\mathrm{TID}_j}\subseteq M_z$且都有一个时间戳$[T_b^{\mathrm{TID}_j},T_e^{\mathrm{TID}_j}](T_1\leqslant T_b^{\mathrm{TID}_j}\leqslant T_e^{\mathrm{TID}_j}\leqslant T_2)$,表示离散时态事务$\mathrm{TID}_j$的有效时间区间.

1) 假设在离散时态事务集$D_z^{[T_1,T_2]}$中,项集$X=\{I_{11},I_{22},\cdots,I_{jj},\cdots,I_{nn}\}(j\leqslant q_j,n\leqslant q_n)$且$X\subset I_{\mathrm{TID}_i}(1\leqslant i\leqslant l)$,同时存在于$l(1\leqslant l\leqslant N)$条离散时态事务中,则项集$X$的生命周期为所有离散时态事务有效时间区间的并集,即

$$\mathrm{lc}(X)[T_b^{\mathrm{TID}_1},T_e^{\mathrm{TID}_1}]\bigcup\cdots\bigcup[T_b^{\mathrm{TID}_i},T_e^{\mathrm{TID}_i}]\bigcup\cdots\bigcup[T_b^{|D_z^{[T_1,T_2]}|},T_e^{|D_z^{[T_1,T_2]}|}].$$

2) 假定项$I_{jj}\subset X$,同时存在于$l(1\leqslant l\leqslant N)$条离散时态事务中,则项$I_{jj}$的有效时间为

$$\Delta t_{jj}=\min(|T_e^{\mathrm{TID}_1}-T_b^{\mathrm{TID}_1}|,|T_e^{\mathrm{TID}_2}-T_b^{\mathrm{TID}_2}|,\cdots,|T_e^{\mathrm{TID}_l}-T_b^{\mathrm{TID}_l}|).$$

记项集X的有效时间集合为

$$\mathrm{lt}(X)=\{\Delta t_{11},\Delta t_{22},\cdots,\Delta t_{nn}\}.$$

3) 假定项I_{jj}在离散事务数据集$D_z^{[T_1,T_2]}$首次出现在时间区间$[T_b^{\mathrm{TID}_1},T_e^{\mathrm{TID}_1}]$内,最后一次出现在时间区间$[T_b^{\mathrm{TID}_l},T_e^{\mathrm{TID}_l}]$内,则项$I_{jj}$的生命周期为从$[T_b^{\mathrm{TID}_1},T_e^{\mathrm{TID}_1}]$到$[T_b^{\mathrm{TID}_l},T_e^{\mathrm{TID}_l}]$的总的时间区间,记为$\mathrm{lc}(I_{jj}),|\mathrm{lc}(I_{jj})|=|T_e^{\mathrm{TID}_l}-T_b^{\mathrm{TID}_1}|$表示项$I_{jj}$的生命周期长度.

定义3 (布尔离散时态矩阵)　假定离散时态事务集$D_z^{[T_1,T_2]}=\{\mathrm{TID}_1,\mathrm{TID}_2,\cdots,\mathrm{TID}_j,\cdots,\mathrm{TID}_N\}$ $(1\leqslant j\leqslant N)$,对应的模式序列项集$M_Z=\{I_{11},I_{12},\cdots,I_{1q_1},\cdots,I_{i1},I_{i2},\cdots,I_{iq_i},I_{n1},I_{n2},\cdots,I_{nq_n}\}(1\leqslant i\leqslant n)$,所对应的布尔离散时态矩阵为

$$X_B^{[T_1,T_2]}=\begin{bmatrix} b_1^{11} & b_1^{12} & \cdots & b_1^{nq_n}\\ \vdots & \vdots & \ddots & \vdots\\ b_l^{11} & b_l^{12} & \ddots & b_l^{nq_n}\\ \vdots & \vdots & \ddots & \vdots\\ b_N^{11} & b_N^{12} & \cdots & b_N^{nq_n}\end{bmatrix}. \tag{1}$$

其中: $b_l^{iq_i}$表示列向量$\mathrm{CV}_{iq_i}=(b_1^{iq_i},\cdots,b_l^{iq_i},\cdots,b_N^{iq_i})^{\mathrm{T}}$ $(1\leqslant l\leqslant N)$中第$l$个元素,且

$$b_l^{iq_i}=\begin{cases}1,\ I_{iq_i}\in M_Z;\\ 0,\ I_{iq_i}\notin M_Z;\end{cases}$$

i表示时间序列s_i; q_i表示时间序列s_i中第q_i个离散化状态; $\sum_{l=1}^{N}b_l^{iq_i}$表示列向量CV_{iq_i}中所有元素的和.

为了挖掘意义更广泛的时态关联规则,在实际应用中,不仅需要知道具有相同时间区间的不同时间序列对象发生的频繁程度,而且需要知道某个时间序列对象某一状态发生后紧接着另一个时间序列对象某个状态发生得是否频繁,发现不同的变化趋势. 下面给出具有时态约束的规则挖掘的定义.

定义4 (支持度和置信度)　设s_i、s_j和s_k是多元时间序列数据集合S中3个不同时间序列对象元素, M_i、M_j和M_k是对应于3个不同时间序列的非空模式序列集合,且$M_i\bigcap M_j\bigcap M_k=\varnothing$; I_{i3}、I_{j5}和I_{k7}是对应于3个模式序列的离散化项,这里考虑在时间段$[T_1,T_2]$内对离散事务数据集$D_z^{[T_1,T_2]}$的规则挖掘.

1) $N_{[T_1,T_2]}(I_{i3}\bigwedge I_{j5})$表示离散化项$I_{i3}$和$I_{j5}$在时间区间$\Delta t_{i3}$和$\Delta t_{j5}$内$(\Delta t_{i3}=\Delta t_{j5})$重复同时出现的事务个数;规则$I_{i3}\overset{[T_1,T_2]}{\longrightarrow}I_{j5}$表示在时间段$[T_1,T_2]$中项$I_{i3}$导致$I_{j5}$同时发生,其时态支持度是$D_z^{[T_1,T_2]}$中包含离散项$I_{i3}$和$I_{j5}$的事务数与$D_z^{[T_1,T_2]}$中事务数之比,即

$$\mathrm{support}(I_{i3}\overset{[T_1,T_2]}{\longrightarrow}I_{j5})=\frac{N_{[T_1,T_2]}(I_{i3}\bigwedge I_{j5})}{|D_z^{[T_1,T_2]}|}. \tag{2}$$

其时态置信度是$D_z^{[T_1,T_2]}$中包含离散项I_{i3}和I_{j5}的事务数与$D_z^{[T_1,T_2]}$中包含离散项I_{i3}的事务数之比,即

$$\mathrm{confidence}(I_{i3}\overset{[T_1,T_2]}{\longrightarrow}I_{j5})=\frac{N_{[T_1,T_2]}(I_{i3}\bigwedge I_{j5})}{N_{[T_1,T_2]}(I_{i3})}. \tag{3}$$

2) $N_{[T_1,T_2]}(I_{i1}\bigwedge(I_{k7})_1)$表示离散化项$I_{i1}$和$(I_{k7})_1$在时间区间$\Delta t_{i1}$内,项$I_{i1}$与紧后时间区间$\Delta t_{k7}$内(即$\Delta t_{i1}$与$\Delta t_{k7}$是相邻的时间区间且$\Delta t_{i1}<\Delta t_{k7}$)的项$I_{k7}$重复相邻出现的事务个数; $(I_{k7})_1$表示在时间段$[T_1,T_2]$内,某项存在项I_{k7}在一个紧后相邻的时间区间内发生. 规则$I_{i1}\overset{[T_1,T_2]}{\longrightarrow}(I_{k7})_1$表示在时间段$[T_1,T_2]$内项$I_{i1}$导致其紧后项$I_{k7}$发生,其时态支持度是$D_z^{[T_1,T_2]}$中包含离散项$I_{i1}$和$(I_{k7})_1$的事务数与$D_z^{[T_1,T_2]}$中事务数之比,即

$$\mathrm{support}(I_{i1}\overset{[T_1,T_2]}{\longrightarrow}(I_{k7})_1)=\frac{N_{[T_1,T_2]}(I_{i1}\bigwedge(I_{k7})_1)}{|D_z^{[T_1,T_2]}|}. \tag{4}$$

其时态置信度是$D_z^{[T_1,T_2]}$中包含离散项I_{i1}和$(I_{k7})_1$的事务数与$D_z^{[T_1,T_2]}$中包含离散项I_{i1}的事务数之比,即

$$\mathrm{confidence}(I_{i1}\overset{[T_1,T_2]}{\longrightarrow}(I_{k7})_1)=\frac{N_{[T_1,T_2]}(I_{i1}\bigwedge(I_{k7})_1)}{N_{[T_1,T_2]}(I_{i1})}, \tag{5}$$

3) $N_{[T_1,T_2]}(I_{i3}\bigwedge I_{j5}\bigwedge(I_{k7})_1)$表示离散化项$I_{i3}$、$I_{j5}$和$I_{k7}$在时间区间$\Delta t_{i3}$和$\Delta t_{j5}$ $(\Delta t_{i3}=\Delta t_{j5})$内,项$I_{i3}$、$I_{j5}$和紧后时间区间$\Delta t_{k7}$内(即$\Delta t_{j5}$与$\Delta t_{k7}$是相邻的时间区间且$\Delta t_{j5}<\Delta t_{k7}$)的项$I_{k7}$重复相邻出现的事务个数;规则$I_{i3}\bigwedge I_{j5}\overset{[T_1,T_2]}{\longrightarrow}(I_{k7})_1$表示在时间段$[T_1,T_2]$内,项$I_{i1}$、$I_{j5}$导致其紧后项$I_{k7}$发生,其时态支持度是$D_z^{[T_1,T_2]}$中包含离散项$I_{i1}$、$I_{j5}$和$(I_{k7})_1$的事务数与$D_z^{[T_1,T_2]}$中事务数之比,即

$$\mathrm{support}(I_{i3}\bigwedge I_{j5}\overset{[T_1,T_2]}{\longrightarrow}(I_{k7})_1)=\frac{N_{[T_1,T_2]}(I_{i3}\bigwedge I_{j5}\bigwedge(I_{k7})_1)}{|D_z^{[T_1,T_2]}|}. \tag{6}$$

其时态置信度是$D_z^{[T_1,T_2]}$中包含离散项I_{i1}、I_{j5}和$(I_{k7})_1$的事务数与$D_z^{[T_1,T_2]}$中包含离散项I_{i1}和I_{j5}的事务数之比,即

$$\mathrm{confidence}(I_{i3}\bigwedge I_{j5}\overset{[T_1,T_2]}{\longrightarrow}(I_{k7})_1)=\frac{N_{[T_1,T_2]}(I_{i3}\bigwedge I_{j5}\bigwedge(I_{k7})_1)}{N_{[T_1,T_2]}(I_{i3}\bigwedge I_{j5})}. \tag{7}$$

2　T-FS-tree时态关联规则挖掘算法

现有的时态关联规则常常是布尔型属性或将定量属性进行区间离散化后挖掘获取的,且仅考虑了同

一时间内属性项之间的关联. 本文侧重于发现不同事务不同时间下发生的多维事件间的关联,这种多维多时间关联规则更加符合现实事物,具有一定的预测作用,体现了时态数据的变化趋势.

针对时间序列数据,首先进行降维离散化,将连续数值型属性随时间在数值上的波动转换为状态值来获取趋势特征. 将属性增量变化趋势分别映射到相应的属性域中,由定量属性转变为定性属性. 属性趋势信息的获取对这种有一定预测作用的关联规则的挖掘质量起着重要的作用,为实现多元时间序列数据集的降维离散化处理,本文采用基于区域极值点的时间序列聚类算法[14],其基本思路是:首先提取原始时间序列中的极值点以实现时间序列的降维压缩处理,然后结合时间序列相似性度量标准,实现时间序列的聚类离散化处理,获取有时间标记的状态值.

这里多维时态关联规则的挖掘与其他关联规则一样,寻找满足某种时态约束频繁发生的模式序列. 考虑树结构在规则挖掘中无需产生候选项集等优势,提出一种新的基于频繁项集树的时态关联规则挖掘算法,其结构构建与频繁项集挖掘同时进行,提高了规则挖掘效率. 整个算法的核心仍然是寻找频集,基本思想如下:在多元时间序列数据集降维离散化的基础上,将所得离散时态事务集转换为布尔离散时态矩阵;根据布尔离散时态矩阵和向量运算得到时态频繁1-项集和频繁2-项集;由所得时态频繁项集(考虑项集之间的时态关系)构建初始频繁项集树,包含任意两个频繁1-项集间的关联关系,用于频繁$k(k \geqslant 3)$-项集的生成;由初始频繁项集树得到完整频繁项集树;遍历所得完整频繁项集树,得到所有时态频繁项集;由所得频繁项集生成强时态关联规则.

T-FS-tree算法的具体过程描述如下.

输入:多元时间序列数据集$S = \{s_1, s_2, \cdots, s_i, \cdots, s_n\}$,最小支持度minsup,最小置信度minconf,最小重要度minimp;

输出:强时态关联规则.

Step 1 获取布尔离散时态矩阵$X_B^{[T_1,T_2]}$. 为避免对原有事务集的多次扫描,由定义3将离散时态事务集$D_z^{[T_1,T_2]}$中的项集列属性转化为布尔数据形式,得到布尔离散时态矩阵,以便于利用向量运算提高算法效率.

Step 2 生成频繁1-项集集合F_1. 假定模式序列集$M_Z = \{I_{11}, I_{12}, \cdots, I_{1q_1}, \cdots, I_{i1}, I_{i2}, \cdots, I_{iq_i}, \cdots, I_{n1}, I_{n2}, \cdots, I_{nq_n}\}$,其中$n$表示时间序列的属性个数. 根据定义3,每个项对应一个列向量$\mathrm{CV}_{iq_i} = (b_1^{iq_i}, \cdots, b_l^{iq_i}, \cdots, b_N^{iq_i})^{\mathrm{T}}$. 若$\sum_{l=1}^{N} b_l^{iq_i} \geqslant N \times \mathrm{minsup}$,则称项集$\{I_{iq_i}\}$为频繁1-项集,否则为非频繁1-项集,所有频繁1-项集构成集合F_1.

Step 3 生成频繁2-项集集合F_2. 将F_1中不同属性的项集两两连接,以项集$\{I_{i_1j_1}\}$和$\{I_{i_2j_2}\}(i_1, i_2 = 1, 2, \cdots, n)$为例,假定它们为频繁1-项集,$i_1$表示时间序列$s_{i_1}$,$j_1$表示时间序列$s_{i_1}$中第$j_1$个离散化状态,$i_2$表示时间序列$s_{i_2}$,$j_2$表示时间序列$s_{i_2}$中第$j_2$个离散化状态. 若$i_1 \neq i_2$,即项集$\{I_{i_1j_1}\}$和$\{I_{i_2j_2}\}$是不同属性的项集,则连接得到2-项集$\{I_{i_1j_1}, I_{i_2j_2}\}$. 根据定义4,在频繁1-项集的时间区间成立的基础上,对第2项依次加上时间区间,判断频繁2-项集所满足的时态关联关系.

Step 4 构建初始频繁项集树. 首先创建根节点Root,其包含1个数据域,离散时态事务集$D_z^{[T_1,T_2]}$的时间区间$[T_1, T_2]$;然后将所得到的频繁1-项集集合F_1和频繁2-项集集合F_2依次插入到树结构中. 具体方法是:将F_1中的频繁1-项集依次插入到树结构中,构建频繁项集树的第1层,其中每个节点由项、列向量、生命周期、有效时间集合4个数据域构成. 对于频繁2-项集集合F_2,以$\{I_{i_1j_1}, I_{i_2j_2}\}$为例,首先遍历第1层节点,找到包含项集$\{I_{i_1j_1}\}$的节点,并以此节点为父节点,构建其子节点,用于对项集$\{I_{i_2j_2}\}$和$\{I_{i_1j_1}, I_{i_2j_2}\}$对应的列向量、生命周期、有效时间集合的存储,重复该过程,直到所有的频繁2-项集全部插入到树结构中为止.

Step 5 由初始频繁项集树构建完整频繁项集树. 初始频繁项集树存储了所有频繁1-项集和频繁2-项集在数据库中的关联关系. 对于任意频繁$k(k \geqslant 3)$-项集的生成,首先查找所含前$k-2$个项均相同的两频繁$k-1$-项集$F_{k-1}{}' = \{F_{k-2}', I_{i_1j_1}\}$和$F_{k-1}{}'' = \{F_{k-2}'', I_{i_2j_2}\}$,其中项集$F_{k-2}'$和$F_{k-2}''$所含项均相同. 因此,任意频繁$k$-项集的生成,均可等价为两频繁$k-1$-项集所含最后项$I_{i_1j_1}$和$I_{i_2j_2}$的连接,若两项存在关联关系,则需进一步判断是否存在频繁k-项集$\{F_{k-2}, I_{i_1j_1}, I_{i_2j_2}\}$;若两项不存在关联关系,则无需进行连接判断,提高算法效率.

假定当前树结构有k层子节点,有:

1) 树结构最后一层即为第k层,其包含m个叶子节点$\{\mathrm{node}_k^1, \cdots, \mathrm{node}_k^i, \cdots, \mathrm{node}_k^m\}(1 \leqslant i \leqslant m)$,其中$\mathrm{node}_k^i$表示树结构第$k$层中第$i$个叶子节点,所含项为$I_{\mathrm{node}_k^i}$. 叶子节点$\mathrm{node}_k^i$向上到达根节点分支上所有节点中的项构成频繁$k-1$项集$\{I_{\mathrm{node}_2^i},$

$I_{\mathrm{node}_3^i},\cdots,I_{\mathrm{node}_k^i}\},\ k\geqslant 3$.

2) 第k层中叶子节点node_k^i,其所含项为$I_{\mathrm{node}_k^i}$,以树结构第2层中包含项$I_{\mathrm{node}_k^i}$的节点为父节点,按照自左向右的顺序在第3层中找到该父节点的l个子节点,得到子节点集合$\mathrm{Node}_3=\{\mathrm{node}_3^1,\cdots,\mathrm{node}_3^j,\cdots,\mathrm{node}_3^l\}(1\leqslant j\leqslant l)$,其中$\mathrm{node}_3^j$表示集合中第$j$个子节点,所含项为$I_{\mathrm{node}_3^j}$. 在频繁$k-1$项集$\{I_{\mathrm{node}_2^i},I_{\mathrm{node}_3^i},\cdots,I_{\mathrm{node}_k^i}\}$的基础上,在树结构第2层中找到包含项$I_{\mathrm{node}_3^j}$的节点,并连接得到候选$k$-项集$\{F_{k-1},I_{\mathrm{node}_3^j}\}$. 根据定义4,在频繁$k-1$项集的时间区间成立的基础上,对第$k$项$(I_{\mathrm{node}_3^j})$依次加上时间区间,判断频繁$k$-项集所满足的时态关联关系. 重复2),直到子节点集合为空或无频繁项集生成为止.

按照自上而下的顺序依次遍历树结构的所有分支,得到所有频繁k-项集:从根节点开始,依次遍历直到频繁项集树的第$k+1$层,存储所遍历节点中的项,获得所有的频繁项集.

Step 6　生成时态关联规则. 由频繁项集、最小支持度minsup、最小置信度minsup和最小重要度minimp,得到时态关联规则为

$$\begin{aligned}&\mathrm{Rule}^{[T_1,T_2]}:\\&(I_{i_1j_1},[T_b^1,T_e^1])\bigwedge\cdots\bigwedge(I_{i_vj_v},[T_b^v,T_e^v])\bigwedge\cdots\bigwedge\\&(I_{i_{n-1}j_{n-1}},[T_b^{n-1},T_e^{n-1}])\overset{\mathrm{lc}(I_{\mathrm{Rule}^{[T_1,T_2]}})}{\longrightarrow}\\&(I_{i_nj_n},[T_b^n,T_e^n]),\sup(\mathrm{Rule}^{[T_1,T_2]}),\\&\mathrm{conf}(\mathrm{Rule}^{[T_1,T_2]}),\mathrm{imp}(\mathrm{Rule}^{[T_1,T_2]}).\end{aligned}\tag{8}$$

令$I_{\mathrm{Rule}^{[T_1,T_2]}}=\{X,Y\}=\{I_{i_1j_1},\cdots,I_{i_vj_v},\cdots,I_{i_nj_n}\}(1\leqslant v\leqslant n)$表示存在于规则$\mathrm{Rule}^{[T_1,T_2]}$中的项集,$X=\{I_{i_1j_1},\cdots,I_{i_vj_v},\cdots,I_{i_{n-1}j_{n-1}}\}$表示规则前件,$Y=\{I_{i_nj_n}\}$表示规则后件. 假定$t(T_1\leqslant t\leqslant T_2)$时刻为规则后件所含项$I_{i_nj_n}$的开始时间,即$t=T_b^n$,且规则所含项集的有效时间集合为$\mathrm{lt}(I_{\mathrm{Rule}^{[T_1,T_2]}})=\{\Delta t_{i_1j_1},\cdots,\Delta t_{i_vj_v},\cdots,\Delta t_{i_nj_n}\}$,其中$\Delta t_{i_vj_v}$表示项$I_{i_vj_v}$在生命周期$\mathrm{lc}(I_{\mathrm{Rule}^{[T_1,T_2]}})$中的有效时间,则规则后件有效时间区间的结束时间为$T_e^n=t+\Delta t_{i_nj_n}$. 第$v$个规则前件有效时间区间的起始时间可分为两种情况:

1) 当$I_{i_vj_v}$与$I_{i_{v+1}j_{v+1}}$同时发生时,$T_b^v=T_b^{v+1}$;

2) 当$I_{i_vj_v}$与$I_{i_{v+1}j_{v+1}}$相邻发生时,$T_b^v=t-\sum_{u=v}^{n-1}\Delta t_{i_uj_u}$.

第v个规则前件有效时间区间的结束时间同样可分为两种情况:

1) 当$I_{i_vj_v}$与$I_{i_{v+1}j_{v+1}}$同时发生时,$T_e^v=T_e^{v+1}$;

2) 当$I_{i_vj_v}$与$I_{i_{v+1}j_{v+1}}$相邻发生时,$T_e^v=t-\sum_{u=v+1}^{n-1}\Delta t_{i_uj_u}$.

规则支持度$\sup(\mathrm{Rule}^{[T_1,T_2]})$、规则置信度$\mathrm{conf}(\mathrm{Rule}^{[T_1,T_2]})$和规则重要度$\mathrm{imp}(\mathrm{Rule}^{[T_1,T_2]})$的定义如下:

$$\sup(\mathrm{Rule}^{[T_1,T_2]})=\frac{N_{[T_1,T_2]}(I_{\mathrm{Rule}^{[T_1,T_2]}})}{|D_z^{[T_1,T_2]}|},\tag{9}$$

$$\mathrm{conf}(\mathrm{Rule}^{[T_1,T_2]})=\frac{N_{[T_1,T_2]}(I_{\mathrm{Rule}^{[T_1,T_2]}})}{N_{[T_1,T_2]}(X)},\tag{10}$$

$$\mathrm{imp}(\mathrm{Rule}^{[T_1,T_2]})=\log\Big(\frac{N_{[T_1,T_2]}(I_{\mathrm{Rule}^{[T_1,T_2]}})}{N_{[T_1,T_2]}(X)\times N_{[T_1,T_2]}(Y)}\Big).\tag{11}$$

其中:$|D_z^{[T_1,T_2]}|$表示离散事务数据集中的事务数,$N_{[T_1,T_2]}(I_{\mathrm{Rule}^{[T_1,T_2]}})$、$N_{[T_1,T_2]}(X)$和$N_{[T_1,T_2]}(Y)$分别表示离散事务数据集$D_z^{[T_1,T_2]}$中包含项集$I_{\mathrm{Rule}^{[T_1,T_2]}}$、$X$和$Y$的事务数. 若规则同时满足

$$\begin{aligned}\sup(\mathrm{Rule}^{[T_1,T_2]})&\geqslant\mathrm{minsup},\\\mathrm{conf}(\mathrm{Rule}^{[T_1,T_2]})&\geqslant\mathrm{minconf},\\\mathrm{imp}(\mathrm{Rule}^{[T_1,T_2]})&\geqslant\mathrm{minimp},\end{aligned}$$

则称候选规则$\mathrm{Rule}^{[T_1,T_2]}$为强关联规则.

Step 7　规则剪枝并输出强时态关联规则. 对于两候选时态关联规则$\mathrm{Rule1}^{[T_1,T_2]}$和$\mathrm{Rule2}^{[T_1,T_2]}$,$I_{\mathrm{Rule1}^{[T_1,T_2]}}=\{X_1,Y_1\}$表示存在于规则$\mathrm{Rule1}^{[T_1,T_2]}$中的项集,$I_{\mathrm{Rule1}^{[T_1,T_2]}}=\{X_2,Y_2\}$表示存在于规则$\mathrm{Rule2}^{[T_1,T_2]}$中的项集. 如果两规则满足下述条件之一,则表明规则$\mathrm{Rule1}^{[T_1,T_2]}$和$\mathrm{Rule2}^{[T_1,T_2]}$存在冗余,需要进行规则的剪枝:

1) 两规则前件项集存在包含关系$(X_1\subseteq X_2$或$X_2\subseteq X_1)$;规则后件相同$(Y_1=Y_2)$;

2) 两规则生命周期相同$(\mathrm{lc}(I_{\mathrm{Rule1}^{[T_1,T_2]}})=\mathrm{lc}(I_{\mathrm{Rule2}^{[T_1,T_2]}}))$;

3) 两规则所含项集的时态关系一致.

若规则$\mathrm{Rule1}^{[T_1,T_2]}$满足如下的条件之一,则需要对规则Rule2进行剪枝:

1) $\sup(\mathrm{Rule1}^{[T_1,T_2]})>\sup(\mathrm{Rule2}^{[T_1,T_2]})$;

2) $\sup(\mathrm{Rule1}^{[T_1,T_2]})=\sup(\mathrm{Rule2}^{[T_1,T_2]})$,
$\mathrm{conf}(\mathrm{Rule1}^{[T_1,T_2]})>\mathrm{conf}(\mathrm{Rule2}^{[T_1,T_2]})$;

3) $\mathrm{conf}(\mathrm{Rule1}^{[T_1,T_2]})=\mathrm{conf}(\mathrm{Rule2}^{[T_1,T_2]})$,
$\sup(\mathrm{Rule1}^{[T_1,T_2]})=\sup(\mathrm{Rule2}^{[T_1,T_2]})$,
$\mathrm{imp}(\mathrm{Rule1}^{[T_1,T_2]})>\mathrm{imp}(\mathrm{Rule2}^{[T_1,T_2]})$;

4) $\mathrm{conf}(\mathrm{Rule1}^{[T_1,T_2]})=\mathrm{conf}(\mathrm{Rule2}^{[T_1,T_2]})$,
$\sup(\mathrm{Rule1}^{[T_1,T_2]})=\sup(\mathrm{Rule2}^{[T_1,T_2]})$,

$$\mathrm{imp}(\mathrm{Rule1}^{[T_1,T_2]}) = \mathrm{imp}(\mathrm{Rule2}^{[T_1,T_2]}),$$
$$X_1 \subseteq X_2.$$

即规则$\mathrm{Rule1}^{[T_1,T_2]}$前件中所含项集数少于规则$\mathrm{Rule2}^{[T_1,T_2]}$,剪枝后得到的时态关联规则为强时态关联规则.

3 性能评估

为了验证T-FS-tree算法的性能,以UCI数据库[15]中的4个时间序列数据集(Air quality, Synthetic ăcontrolăchart, Dow jones index, Japanese vowels)作为实验数据. 分别设计对比4种不同时态关联规则挖掘方案,如表1所示. 在进行时态关联规则挖掘之前,采用基于关键点的时间序列相似性聚类算法对时间序列数据进行降维离散化处理,从算法复杂度、运行时间和参数变化影响等角度进行对比实验.

表1 时态关联规则挖掘算法

方案	核心算法
1	Apriori算法[2]
2	FP-growth算法[6]
3	基于FP-树的时态关联规则的分区挖掘算法[5]
4	本文T-FS-tree算法

3.1 复杂度的理论对比分析

对于数据容量为$n \times m$的时间序列数据集S,m表示序列样本数,n表示序列属性个数,假定$|F_k|$、$|C_k|$分别表示频繁k-项集和候选k-项集的个数,在最坏的情况下,每个时间序列属性会被划分为n个状态区间. 为验证T-FS-tree算法的效率,在上述条件下,从理论角度对比了4种方案在时态关联规则生成过程中对原数据库的扫描次数、时间复杂度和空间复杂度,分析结果见表2.

表2 算法复杂度对比

方案	原数据库扫描次数	时间复杂度	空间复杂度
1	k	$O\Big(n \times m^2 + \sum_{k\geqslant 2} \lvert F_k\rvert \times \lvert F_k\rvert + \lvert C_k\rvert \times (m+1)\Big)$	$O(n \times m + \lvert F_k\rvert + \lvert C_k\rvert)$
2	2	$O(n \times m \times (3(m+1)/2))$	$O(2 \times (n \times m) + \lvert F_k\rvert)$
3	2	$O(n \times m \times (3(m+1)/2))$	$O(3 \times (n \times m) + \lvert F_k\rvert)$
4	1	$O((n \times m) \times (1+k))$	$O((1+m) \times (n \times m) + \lvert F_k\rvert)$

由表2可见,在扫描原有数据库方面,方案1需要多次扫描数据库生成候选项集,方案2和方案3需2次扫描,方案4 仅需1次扫描. 利用频繁1-项集和频繁2-项集构造初始的频繁项集树的第1层和第2层,在此基础上寻找对应k层的具有时态关联关系的频繁k-项集,避免遍历树的所有分支,方案4的时间复杂度明显低于其他3种情况;而在空间复杂度方面,由于方案4在数据预处理阶段需要对离散时态事务集进行转换,其空间复杂度稍高于方案2和方案3,但仍好于方案1,从理论上可能说明所提出的方案4(T-FS-tree算法)具有较好的性能.

3.2 参数和数量变化对T-FS-tree算法的影响

为验证支持度、置信度参数和数据量变化对T-FS-tree算法效率的影响,基于4个时间序列数据集,在不同的样本数量下,随着最小支持度和最小置信度参数的变化,对比了算法的运行时间. 图1为不同支持度和数量下的算法运行时间变化,图2为不同置信度和数量下的算法运行时间变化.

随着数据量的增加,由图1(a)可见,算法T-FS-tree在不同支持度下的运行时间均较小,且变化较为平缓,但当数据量较大且支持度较小时,运行时间变化较大;由图1(b)可见,在支持度等于0.35及更小的情况下,算法T-FS-tree的运行时间相对较大,而在其他情况下变化平缓且运行时间较少;由图1(c)可见,除支持度等于0.1的情况,在其他支持度下,算法T-FS-tree所需运行时间变化很小;由图1(d)可见,不同支持度下算法T-FS-tree所需运行时间不同,这是由于该数据集分布变化较大引起的,除支持度等于0.3的情况,其他支持度下算法的运行时间变化平稳. 通过对比4种数据集下的运行结果可以发现,随着数据量的增加,算法T-FS-tree的运行时间也逐渐增加,但变化趋势较为平缓,时间幅值变化不大,表明算法T-FS-tree具有较好的可扩展性,当最小支持度为0.1时,相比于其他情况,会得到更多的频繁项集,增加了构建频繁项集树及规则生成所需时间,因此,为了保证算法具有较高的效率和鲁棒性,最小支持度不应设定太小.

由图2(a)可见,对于不同的样本数量,在不同的置信度下,算法的运行时间变化不明显,而且随着数据量的增加,运行时间也逐渐增加;由图2(b)和图2(d)可见,对于不同的样本数量,在不同的置信度下,算法的运行时间虽然存在差距,但差距不大,运行时间同样会随数据量的增加而增加;由图2(c)可见,随着数据量的增加,算法T-FS-tree的运行时间也逐渐增加,但变化趋势较为平缓,当数据量为400时,算法运行时间变化明显增大,这可能是与数据分

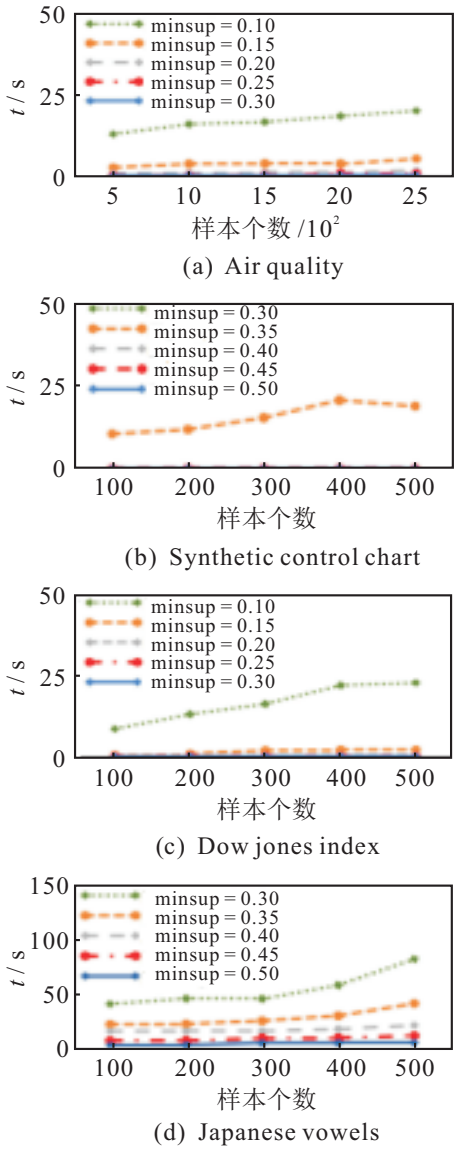


图 1　不同支持度和数据量下算法运行时间的变化

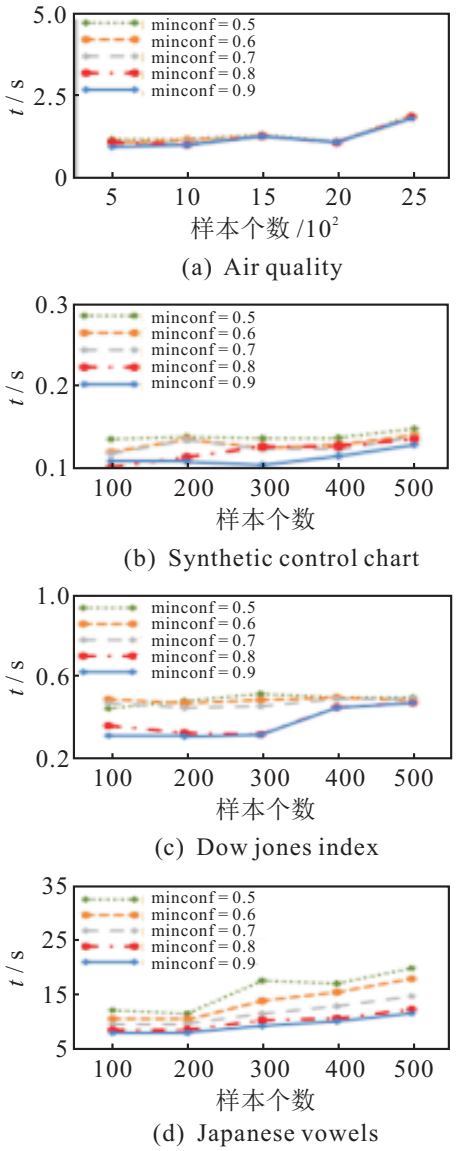


图 2　不同置信度和数据量下算法运行时间的变化

布有关. 因此,同样可以表明算法T-FS-tree具有较好的可扩展性;而在相同数量的条件下,不同置信度下,算法T-FS-tree在数据集Air quality和Dow jones index上的运行时间变化不明显,而对于数据集Synthetic controlăchart和Japanese vowels,虽然运行时间存在一定差异,但变化范围不大,因此可以说明,在相同数量情况下,置信度参数变化对算法T-FS-tree的影响不大,算法具有较好的鲁棒性.

3.3　不同方案的性能对比

为验证T-FS-tree算法的性能,在多个不同支持度的条件下,分别对比4种方案产生的时态关联规则数的变化. 如图3所示,当支持度小于等于0.25时,方案4所得时态关联规则数均大于其他方案,这是由于算法T-FS-tree能够产生更多不同时态关系的规则,且支持度越小,各方案规则数差距相对越大,这是由于数据集中相邻时态关系规则支持度相对较小引起的;当支持度较大时,各方案所得规则数均较少且差距不大. 因此,为了减少时态关联规则的数量,选择合适的最小支持度至关重要.

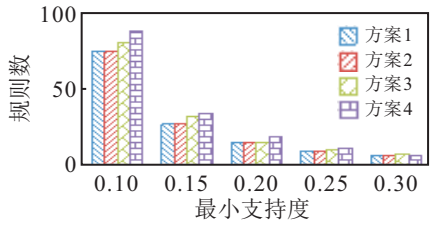
3.4　规则时态关系的实例验证

为验证T-FS-tree算法对时序数据中时态规律的描述,使用Air quality数据集中部分数据进行时态关联规则的挖掘,所得的部分时态关联规则如表3所示.

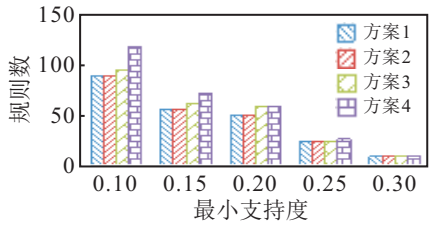
为了更好地理解规则的物理意义,对上述时态关联规则进行明确的解释,以表3最后一条时态关联规则为例,有

$$(31|[15,16]) \bigwedge ((22|[16,19]) \overset{[1,19]}{\longrightarrow} (62|[16,19])),$$
$$\text{sup} = 0.17,\ \text{conf} = 1, \text{imp} = 0.60. \tag{12}$$

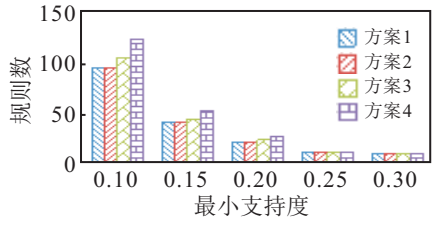
规则前件包含项31、22以及它们发生的有效时间区间[15,16]和[16,19],规则的后件包含项62以及它发生的有效时间区间[16,19]. 根据时间序列变量离散化得到的项集,可以对应到时间序列的序列片断模式,如表4所示. 因此,规则可以进一步表示为



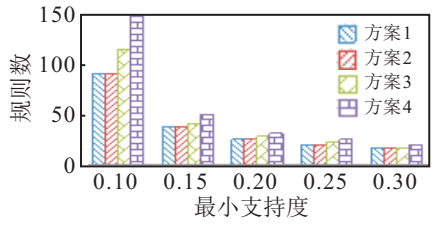
(a) Air quality



(b) Synthetic control chart



(c) Dow jones index



(d) Japanese vowels

图 3 时态关联规则数的变化

$$(\mathrm{NOx(GT)}\{\bar{k}=-31.6;\ \Delta\bar{y}=-92.4\}|$$
$$[2004.03.11.05,\ 2004.03.11.06]),$$
$$(\mathrm{PT08.S2(NMHC)}\{\bar{k}=74.25;\ \Delta\bar{y}=144.2\}| \quad (13)$$
$$[2004.03.11.06,\ 2004.03.11.09]) \quad (14)$$
$$\xrightarrow{[2004.03.10.18,\ 2004.03.11.12]}$$
$$(\mathrm{PT08.S5(O3)}\{\bar{k}=80.8;\Delta\bar{y}=911.1\}|$$
$$[2004.03.11.06,\ 2004.03.11.09]),$$
$$\mathrm{up}=0.17,\ \mathrm{conf}=1,\ \mathrm{imp}=0.60. \quad (15)$$

其中: $\bar{k}$ 为斜率均值; $\Delta\bar{y}$ 为幅值变化均值. 但形如式(13)的规则还是难以理解其时间序列的变化趋势, 为此, 进一步给出序列片断模式的语义描述如表5所示.

根据表5的语义描述, 上述规则可以进一步描述为: 在2004年3月10日18时～2004年3月11日12时的时间区间中, 时间序列NOx(GT)如果在时间2004年3月11日1时～2004年3月11日2时内(即时间区间[15,16])按照快速小幅下降的趋势变化, 而时间序列PT08.S2(NMHC)在时间2004年3月11日2时～2004年3月11日5时内(即时间区间[16,19])按照剧烈中幅上升的趋势变化, 则可以得到时间序列PT08.S5(O3)在时间2004年3月11日2时～2004年3月11日5时内(即时间区间[16, 19]), 会按照剧烈中幅上升的趋势变化. 图4更形象地展示了该规则在生命周期中的变化趋势.

表 3 时态关联规则挖掘算法

关联规则	支持度	置信度	重要度	生命周期	有效时间集合	后件开始时间
$(14,[35,40])\xrightarrow{[19,48]}(64,[35,40])$	0.17	1	0.78	[19, 48]	5, 5	35
$(22,[17,20])\xrightarrow{[5,37]}(62,[17,20]$	0.25	1	0.60	[5, 37]	3, 3	17
$(23,[29,30])\bigwedge(53,[29,30])\xrightarrow{[15,50]}(63,[29,30]$	0.17	1	0.78	[15, 50]	1, 1, 1	29
$(31,[15,16])\bigwedge(22,[16,19])\xrightarrow{[1,19]}(62,[16,19])$	0.17	1	0.60	[1, 19]	1, 3, 3	16

表 4 离散化项集的序列片段模式

时间序列变量	离散化项	斜率均值	斜率对应角度	幅值变化均值	归一化变化幅值均值
PT08.S2（NMHC）	22	74.25	89.25	144.2	0.48
NOx(GT)	31	−31.6	−87.1	−92.4	−0.16
PT08.S5(O3)	62	80.8	89.3	911.1	0.45

表 5 序列片断模式的语义描述

斜率对应角度范围	语义描述	归一化变化幅值范围	语义描述
[−90, −60]	剧烈下降	[−1, −0.6]	大幅下降
[−60, −30]	快速下降	[−0.6, −0.3]	中幅下降
[−30, 0]	平稳下降	[−0.3, 0]	小幅下降
[0, 30]	平稳上升	[0, 0.3]	小幅上升
[30, 60]	快速上升	[0.3, 0.6]	中幅上升
[60, 90]	剧烈上升	[0.6, 1]	大幅上升

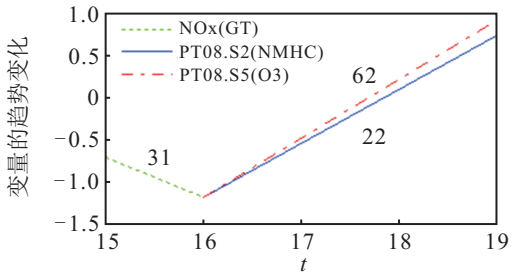


图 4 规则变化趋势

为了评估该规则的时态关系, 与实际时态数据中的变化趋势进行对比验证, 匹配结果如图 5 所示. 虚线框中的变化趋势对应于图 4 表示的规则变化趋势, 可以看到, 规则表示的时间趋势能够较好地表示原有时间序列数据的变化趋势. 在规则的生命周期 [1, 19] 中, 该规则表示的变化趋势出现两次, 在时间区间 [1, 5] 中时间序列 NOx(GT) 为快速下降的小幅变化趋势, 在时间区间 [5, 12] 中时间序列 PT08.S2(NMHC) 为剧烈上升的中幅变化趋势, 同时时间序列 PT08.S5(O3) 也为剧烈上升的中幅变化趋势, 在时间区间 [15, 19] 中也有类似的趋势变化, 因此, 本文所得的时态关联规则挖掘算法能够较好地反映时间序列中的趋势变化.

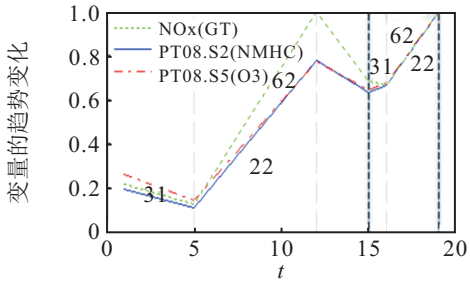


图 5　时间序列匹配结果

4　结　论

为了更有效地挖掘多维事务间的时态关联关系, 本文提出了一种新的基于频繁项集树的时态关联规则挖掘算法. 采用一种新的存储结构 —— 频繁项集树结构, 可以避免产生候选项集和有效减少扫描数据库的计算成本. 同时, 该结构构建与频繁项集挖掘同时进行, 提高了频繁项集挖掘效率. 以此为基础, 充分考虑了项集之间的时态关系, 挖掘不同时间区间的事务间的时态关联规则. 为验证算法的性能, 采用 UCI 数据库中的多个时间序列数据集进行仿真实验, 与其他算法相比, 所提出算法在规则挖掘效率和解释性方面具有较好的优势. 如何确定最小支持度和置信度等问题有待进一步研究.

参考文献 (References)

[1] Ning H, Yuan H, Lu Z, et al. Research on association rules mining with temporal restraint[C]. IEEE Conf on Industrial Electronics and Applications. Piscataway NJ: IEEE, 2007: 1600-1602.

[2] Mao G. Mining temporal association rules in network traffic data[J]. Int J of Future Computer and Communication, 2014, 3(1): 55-59.

[3] Schlüter T, Conrad S. Mining several kinds of temporal association rules enhanced by tree structures[C]. The 2nd Int Conf on Information, Process, and Knowledge Management. Piscataway NJ: IEEE, 2010: 86-93.

[4] Yin K C, Hsieh Y L, Yang D L, et al. Association rule mining considering local frequent patterns with temporal intervals[J]. Applied Mathematics & Information Sciences, 2014, 8(4): 1879-1890.

[5] 马慧, 汤庸, 潘炎. 一种基于FP-树的时态关联规则的分区挖掘方法[J]. 计算机工程, 2006, 32(17): 132-134.
(Ma H, Tang Y, Pan Y. A FP-tree based partition mining approach to discovering temporal association rules[J]. Computer Engineering, 2006, 32(17): 132-134.)

[6] Pankaj G, Sagar B B. Discovering weighted calendar-based temporal relationship rules using frequent pattern Tree[J]. Indian J of Science and Technology, 2016, 28(9): 1-6.

[7] Rashid M M, Gondal I, Kamruzzaman J. Mining associated patterns from wireless sensor networks[J]. IEEE Trans on Computers, 2015, 64(7): 1998-2011.

[8] 秦亮曦, 史忠植. 多时间序列跨事务关联分析研究 [J]. 计算机工程与应用, 2005, 41(27): 10-12.
(Qin L X, Shi Z Z. Research on multiple time series inter-transactional association analysis[J]. Computer Engineering and Applications, 2005, 41(27): 10-12.)

[9] Chen C H, Lan G C, Hong T P, et al. Mining fuzzy temporal association rules by item lifespans[J]. Applied Soft Computing, 2016, 41: 265-274.

[10] Mohd Khairudin N, Mustapha A, Ahmad M H. Effect of temporal relationships in associative rule mining for web log data[J]. The Scientific World J, 2014: 121-130.

[11] Chen Y C, Peng W C, Lee S Y. CEMiner —— An efficient algorithm for mining closed patterns from time interval-based data[C]. Int Conf on Data Mining. Piscataway NJ: IEEE Computer Society, 2011:121-130.

[12] Ruan G, Zhang H, Plale B. Parallel and quantitative sequential pattern mining for large-scale interval-based temporal data[C]. IEEE Int Conf on Big Data. New York: IEEE, 2014: 32-39.

[13] Talha A M, Junejo I N. Dynamic scene understanding using temporal association rules[J]. Image and Vision Computing, 2014, 32(12): 1102-1116.

[14] 孙雅, 李志华. 基于区域极值点的时间序列聚类算法 [J]. 计算机工程, 2015, 41(5): 33-37.
(Sun Y, Li Z H. Clustering algorithm for time series based on locally extreme point[J]. Computer Engineering, 2015, 41(5): 33-37.)

[15] UCI machine learning repository[DB/OL]. [2016-10-12]. http://archive.ics.uci.edu/ml/.

(责任编辑：郑晓蕾)