

文章编号:1001-9081(2006)12-2985-03

挖掘多粒度时间下异步周期的模式

夏卓群¹, 程 显², 梁涤青¹

(1. 长沙理工大学 计算机与通信工程学院, 湖南 长沙 410076;

2. 浙江大学 计算机科学与技术学院, 浙江 杭州 310027)

(xiazhuoqun@tom.com)

摘要: 把异步周期和多时间粒度下的时态模型结合起来研究, 并利用异步周期的特点提出了一种有效的挖掘算法。算法先找到所有的有效时间节, 再通过有效时间节求出最长的有效时间段。实验表明所提出的算法是稳定而有效的。

关键词: 数据挖掘; 异步周期; 多粒度; 时间序列

中图分类号: TP311.131 **文献标识码:**A

Discovery of asynchronous periodic patterns with multiple granularities

XIA Zhuo-qun¹, CHENG Yu², LIANG Di-qing¹

(1. College of Computer and Communication Engineering, Changsha University of Science and Technology,
Changsha Hunan 410076, China;

2. College of Computer Science, Zhejiang Gongshang University, Hangzhou Zhejiang 310035, China)

Abstract: An algorithm for efficient mining of asynchronous periodic patterns with multiple granularities by exploring some interesting properties related to asynchronous periodicity was proposed. The algorithm generated all valid time section at first, and then found the longest valid time segment by using valid time section. The performance study shows that the proposed method is stable and efficient.

Key words: data mining; asynchronous periodicity; multiple granularities; time series

0 引言

周期模式的挖掘是数据挖掘领域近年来兴起的一个研究热点, 有很重要的实用价值。以前大部分周期模式的表示方法为符号串或数学曲线, 但有时用传统的方法不能表示出某些周期模式。例如某公司每个月 1 号给员工发工资, 连续两次发工资的时间间隔为 28 到 31 天不等。这种周期就不方便用符号串或数学曲线表示, 必须用多时间粒度来表示周期模式。此外用多时间粒度来表示周期模式也能更加直观地表示信息。本文中使用了基于日历的模式来表示周期。例如(年:2004, 月:*, 日:15)就是一个基于日历的模式, 表示 2004 年每个月的 15 号。在这里用“*”来代表任何一个在这个时间可能出现的数字。上面发工资的例子的周期模式可以表示为(月:*, 日:1)。

现实世界中大部分的周期都不是这种数学意义上的严格周期, 因为噪音的关系周期可能在时间上发生平移。把这种在时间上平移了的周期称为异步周期。例如一个超市每个星期补充一次大米, 进货的时间是每个星期一, 但有个星期的大米销售特别的好, 在星期四大米就买光了, 超市只好在星期五提前进货。这以后大米进货的频率回复到了一个星期一次, 但进货的时间移到了星期五。当平移的时间间隔在一定范围之内时, 我们仍希望发现这种周期模式。定义两个阈值 min_rep 、 max_dis , 用来验证有效时间节和有效时间段。在一连续的时间段事件的时间发生呈完美的周期规律(即数学意义上的周期), 如果这段时间里发生的事件数不小于 min_rep ,

则称这段连续时间为有效时间节。借鉴上面超市的例子, 如 $min_rep = 5$, 在 2003 年的第 1 到第 7 个星期的星期一超市都会进买进大米, 表示为(年:2003, 周:1—7, 日:1), 而从第 8 个星期开始在星期五买进大米, 表示为(年:2003, 周:8—, 日:5), 显然在 1 ~ 7 个星期买进大米这个事件发生的时间是呈完美周期规律的(周期为一个星期), 在这段时间里事件重复的次数为 $7 > 5$, 那么 1 ~ 7 个星期为一个有效时间节。在一个连续时间段内包含了若干的有效时间节, 如果任意相邻两个有效时间节的间隔小于或等于 max_dis , 则称这段连续时间为有效时间段。周期重复次数最多的时间段称为最长有效时间段。在有效时间节中, 事件的发生必须符合完美的周期规律并重复 min_rep 次, 以体现这节时间的重要性和时间的周期性。另一方面两个有效时间节之间的间隔必须在 max_dis 之内, 如果超出这个范围就认为事件的行为已发生改变, 而不是因为中间有噪音和突发情况。 Max_dis 是衡量这两种情况的边界。我们提出的算法将发现符合这两个阈值的周期模式和最长的有效时间段。

在时序数据的周期挖掘方面已经取得了很多研究成果, 大部分研究都集中在符号串和数学曲线来表示周期模式。文献[1]中引入了部分周期的概念, 并提出了一个在时间序列中寻找部分周期的高效算法——“最大子模式命中集算法”。文献[2]中提出了一种在没有预先知道周期的情况下, 利用自相关函数来探测周期长度的方法。文献[3]与我们的工作相似, 提出了异步周期的概念和挖掘算法, 但它是使用传统的符号串方法来表示周期模式。文献[4]中提出了层次周期的

收稿日期:2006-06-12; 修订日期:2006-09-01

作者简介: 夏卓群(1977-), 男, 湖南益阳人, 讲师, 主要研究方向: 数据挖掘、粗糙集; 程显(1977-), 男, 湖南湘潭人, 博士研究生, 主要研究方向: 数据挖掘、粗糙集; 梁涤青(1980-), 男, 湖南娄底人, 助教, 硕士研究生, 主要研究方向: 数据挖掘。

概念,对周期具有层次性进行了分析,并提出了挖掘高层周期的算法。

在与多时间粒度相关的数据挖掘的研究方面,文献[5]系统地研究了时态型和时间粒度的有关理论。文献[6]中提出了在多粒度时间下挖掘时态模式的四种算法,并用实验对它们的性能做了详细的对比。本文中借鉴了文献[6]中的时态模式表示方法。文献[7]中提出了用时间粒度自动机来发现频繁模式的算法。

1 建立模型

引进时间模板的概念,时间模板 R 可表示如下:

$$R = (f_n:D_n, \dots, f_2:D_2, f_1:D_1)$$

f_i 代表时间粒度(年,月,日等); D_i 是一个正整数的子集,它是对应时间粒度 f_i 的取值范围。当 D_i 显而易见时可以省略,如当 f_i 为月时则 $D_i = \{1 \sim 12\}$ 可以省略。

对应模板 $R = (f_n:D_n, \dots, f_2:D_2, f_1:D_1)$, 输入 Ψ 为 $\{T_1, T_2, \dots, T_n\}$, 其中 $T_i = (t_n, \dots, t_2, t_1)$ 代表事件发生的时间。例如对应 $R = (\text{年}, \text{月}, \text{天})$, $T_i = (2004, 7, 1)$ 表示事件发生的时间是 2004 年 7 月 1 日。有效时间段模式可以表示 (p_n, \dots, p_2, p_1) , 其中 $p_1 \in D_1, p_i \in 2^{D_i} \cup \{\ast\}$, \ast 代表 D_i 中的任意一个整数。例如对应模板 $R = (\text{年}, \text{周}, \text{日})$ ($2 \leq i \leq n$) 在 2003 年的第一第 7 个星期一超市都会买进大米, 表示为 $(2003, 1-7, 1)$ 。对于一个输入 $T = (t_n, \dots, t_2, t_1)$ 和模式 $P = (p_n, \dots, p_2, p_1)$, 若 $t_i = p_i$, 或 $p_i = \ast$, 或 $t_i \in p_i$ ($1 \leq i \leq n$), 则称模式 P 覆盖输入 T 。有效时间段模式表示为它包含的有效时间段模式的并, 如 $(2003, 1-7, 1) \cup (2003, 7-4)$ 。最长有效时间段的模式就是要找的周期模式。

2 找出有效时间段

首先产生候选有效时间段模式集。对应模板 $R = (f_n:D_n, \dots, f_2:D_2, f_1:D_1)$, 输入 Ψ 为 $\{T_1, T_2, \dots, T_n\}$ 对于输入的每一个元素 $T_i = (t_n, \dots, t_2, t_1)$, 可以产生一个候选模式 (\ast, \dots, \ast, t_1) 把它添加到候选模式集 Cdt 中。例如时间模板为(年,月,日), 对于输入 $T_i = (2004, 7, 11)$, 产生候选模式 $(\ast, \ast, 11)$ 。

找出所有的有效时间段。利用一个长度为 $|D_n| \times \dots \times |D_2| \times |D_1|$ 的数组 A , 初始化 A 中的所有元素为 0, 对输入 Ψ 中的每个元素 T 通过哈希函数 $G(T) = (t_n - m_n) \times |D_{n-1}| \times \dots \times |D_2| \times |D_1| + \dots + (t_2 - m_2) \times |D_1| + (t_1 - m_1)$ (m_i 为 D_i 中最小的整数), 映射到数组 A 上, 令 $A[G(T)] = 1$ 。那么输入的元素和数组 A 中等于 1 的元素就成了一一对应的关系。这样虽然增加了 $|D_n| \times \dots \times |D_2| \times |D_1|$ 个单位的空间, 但查找与模式 P 覆盖的输入时只需查找模式覆盖的点, 而不用扫描整个输入。对于候选模式 $P = (\ast, \dots, \ast, p_1)$ ($p_1 \in D_1$) 按时间顺序查找它覆盖的时间点, 最先查找 (m_n, \dots, m_2, m_1) , 其中 m_i 为 D_i 中最小的数 ($2 \leq i \leq n$)。若这个点有输入, 则把时间段长度加 1, 查找模式覆盖的下一个时间点; 若这个点没有输入则比较当前时间段的长度是否大于或等于 min_rep , 若是则为有效时间段, 否则不是。然后开始寻找下一个覆盖的时间点。

3 取得最长有效时间段

前面找出的有效时间段可能会重叠, 为了找到最长的有

效时间段, 并不能把有效时间段简单的相连, 而必须选择其中一些有效时间段或者是有效时间段的某个部分来组成最长有效时间段。在这里先介绍一个概念: 如果一个有效时间段的末尾紧接着一个与模式匹配的周期, 或是有与之时间间隔小于 max_dis 的有效时间段存在, 则称此时间段是可扩展的。

本文提出的 VTS(Valid Time Section) 算法思想是: 顺序扫描有效时间段覆盖的时间区间的数据, 当扫描到的有效时间段中每个周期的第一个位置时, 对候选时间段集合中的时间段实施下面两个扩展操作:

(1) 若候选时间段集合中的时间段 X 的结束位置与当前位置(指当前扫描的位置)之间的时间间隔不大于 max_dis , 且当前位置到当前时间段的结束位置之间至少包含了 min_rep 个周期, 则把以当前位置为开始位置并包含了 min_rep 个周期的时间节接入到这个时间段, 形成一个新的时间段 Y , 把时间段 Y 放到候选时间段集合中。称这种扩展为跳跃式扩展。例如当 $min_rep = 3$, $max_dis = 5$ 天, 如图 1(b) 所示, 当扫描到位置 31 时, 时间段 $X1$ 的结束位置 28 与当前位置的时间间隔为 2 天, 小于 max_dis , 把时间节 S 接入到时间段 $X1$ 中, 得到了一个新的时间段 $Y2$ 。之所以接入的是 min_rep 个周期的时间节是因为: ① 至少要包含了 min_rep 个周期的时间节才是有效时间段; ② 需要考虑各种时间段扩展的各种情况。

(2) 若候选时间段集合中的时间段 X 的结束位置与当前位置相差 1, 则将时间段 X 中的最后一个时间段扩展一个周期长度, 形成一个新的时间段 Y , 把时间段 Y 放到候选时间段集合中。称这种扩展为单步扩展。如图 1(c) 所示, 当扫描到位置 29 时, 时间段 $X1$ 的结束位置与周期 $D10$ 相差 1, 扩展时间段 $X1$ 得到了新的时间段 $Y2$ 。

当扫描到时间区间的最后位置, 包含周期次数最多的时间段就是最长的有效时间段。

我们注意到, 如果一个可扩展的有效时间段的末尾位置, 既存在一个紧接的与模式匹配的周期, 又有多个与之时间间隔小于 max_dis 的有效时间段存在, 例如前面提到的时间段 X , 那么可以知道增加的候选时间段的数量是随着有效时间段的数量呈指数形式增长的。所以一个有效的裁剪方法是非常重要的。

我们提出的裁剪方法就是要尽量减少新增加到候选时间段集合的时间段数量, 并最大可能删除掉候选时间段集合中不可能成为最长有效时间段的时间段。在算法 VTS 中主要使用了以下裁剪方法:

1) 在可以被施加操作(1)的时间段集中选出一个周期重复次数最多的时间段施加操作(1)。

2) 在候选时间段集合中选出可以被施加操作(2)的所有时间段中, 选出一个周期重复次数最多的时间段执行操作(2)。在候选时间段集合中删除其余可以被施加操作(2)的时间段。

3) 在候选时间段集合中不可扩展的时间段移到不可扩展的时间段集合中。

对文献[3]中 SB 算法的改进:

1) SB 算法中没有首先求出所有有效时间段, 所以在执行跳跃式扩展时接入只包含一个周期的时间节到候选时间段。那么, 这个新接入的时间节就不可能扩展成为一个有效的时间节, 相应的这个时间段也就不能成为有效的时间段。假设 $min_rep = 5$, $max_dis = 6$, 如图 1(d) 所示, 在算法 SB 中时间段 $X2$ 执行跳跃式扩展就可能产生一个时间段 $Y3$, 而根

据图 1(a)中可以看出, Y_3 是不能成为有效的时间段的。VTS 算法利用了前面求得的有效时间段的信息, 在执行跳跃式扩展时, 必须是在现有时间段后加入一个有效的时间节(包含了 min_rep 个匹配周期的时间节), 这样就避免了算法 SB 中的无效时间段的产生。

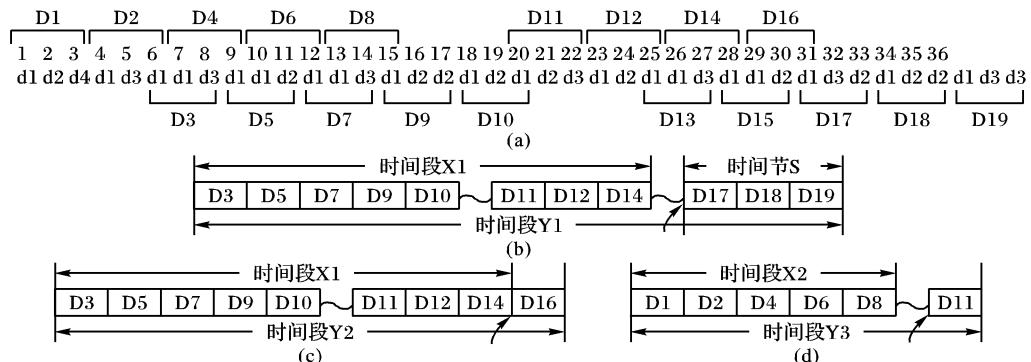


图 1 有效时间段的选取

4 实验

算法都是用 C 实现, 运行在 CPU 为 PIV 1.8G, 256M 内存, 操作系统为 Windows 2000 的 PC 上。

4.1 实际数据

股票数据是现实生活中一种非常典型的时间序列, 把所提出的算法应用到实际的股票以验证算法的有效性。

对 1991 年 1 月 2 日到 2002 年 9 月 19 日的深发展 A 的数据进行了挖掘。把一个交易日作为基本时间粒度, 比较每个交易日的开盘价与收盘价进行比较得出股票的涨跌信息。

实验中发现了一些有趣的模式, 例如: 设置 $len = 1$ (len 为周期长度), $min_rep = 3$, $max_dis = 5$ 时, 这支股票的持续上涨最长的时间段为 1991 年 9 月 6 日 ~ 1991 年 10 月 12 日。当设置 $len = 5$, $min_rep = 3$, $max_dis = 21$, 发现 1995 年 9 月 5 日到 1997 年 6 月 12 日每隔一周至少出现一次上涨。

为了进一步分析算法的效率, 使用了 2M 个合成数据。

4.2 合成数据

合成数据的产生方法如下: 1) 生成的每个时间节包含的周期个数服从数学期望为 200 的几何分布; 2) 在每个时间节后面插入一段干扰噪音, 噪音的长度服从数学期望为 200s 的几何分布。重复这两步直到产生 2M 的数据。

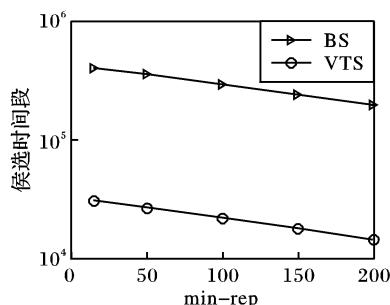
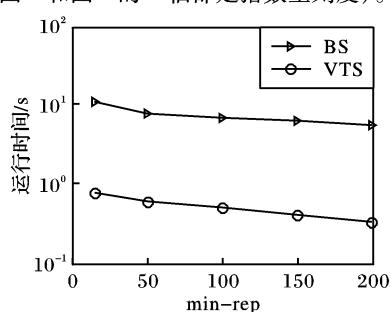
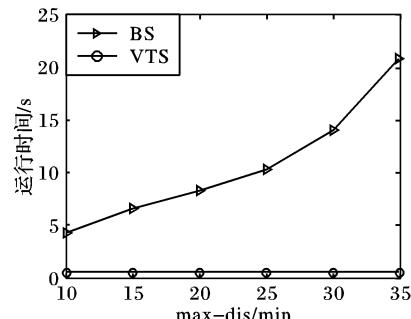


图 2 最小重复次数对候选时间段的影响

在图 2 和图 3 所表示的实验中, 阈值 max_dis 的设置方法为 $max_dis = 600 + min_rep \times 4$ 。从图 2 中可以看出, 在各种 min_rep 值的情况下, SB 算法所产生的候选时间段数都比算法 VTS 所产生的候选时间段数多了大概一个数量级。这是因为算法 VTS 利用了有效时间段的信息增强了对候选时间段的裁剪能力, 阻止了大量无效候选模式的产生。而候选模式的数量是影响算法效率的关键所在, 算法 VTS 产生的候选时间段数量相比算法 SB 减少了, 所以算法 VTS 的执行时间

2) 所建立的模型可以发现的周期规律可能只在所挖掘的整个时间范围内的一部分出现, 例如所挖掘的数据时间范围为一年, 而有周期规律的只有上半年。算法 VTS 利用了模型的这种特性, 只在出现了有效时间段的时间范围内搜索, 缩减了产生候选时间段需要考虑的时间范围。

图 3 min_rep 对生成候选所需时间的影响图 4 max_dis 对算法运行时间的影响

在图 4 所表示的实验中, min_rep 设置为 100, 显然 max_dis 的变化对算法 VTS 并没有很大的影响, 而算法 BS 的运行时间是随着 max_dis 的增加而增加的。这主要是因为 max_dis 增加会大量增加算法 BS 产生的无效候选时间段数, 而对有效时间段的数量影响并不是很大, 所以算法 VTS 候选时间段的数量不会有太大的影响。

5 结语

把多时间粒度下的周期模型和异步周期的挖掘方法结合起来, 研究了异步周期在多时间粒度下的特点和性质。最后提出了挖掘的模型和挖掘算法 VTS, 并与文献[3]中算法 BS 在运行效率上做了详细的对比。VTS 利用了有效时间段的信息裁剪了候选时间段的数量, 缩减了产生候选周期需要考虑的时间范围, 提高了运行速度。

(下转第 2990 页)

故由(4)和(5)式有 $\forall k \in K, i \in I, \bar{a}_{ki} = a_{ki}$, 即 (\bar{W}, \bar{U}) 使得(1)式成立, 故 $(\bar{W}, \bar{U}) \in W(p)$ 。

证毕。

本定理直接给出了连接权矩阵对的解析表达式, 故同时也解决了给定的训练模式集 $set = \{(A_k, B_k) | k \in K\}$ 是否可以成为 $Max-T_L FBAM$ 的平衡态集的判定问题, 我们只需把 (\bar{W}, \bar{U}) 代入(1)式, 看是否使其成立即可。

3 实验

实验步骤:(1)设能成为 $Max-T_L FBAM$ 的平衡态集的训练模式集, 如表1所示。

(2)利用本文提出的学习算法, 计算得到 $Max-TL FBAM$ 的连接权矩阵对, 如表2所示。

(3)根据文献[1]的网络模型($Max-Min FBAM$)和模糊Hebb学习规则以及表1中的训练模式集, 计算得到连接权矩阵对, 如表3所示。但是此时表1中的训练模式集不能被该网络完整可靠地存储。

表1 能成为 $Max-TL FBAM$ 的平衡态集的训练模式对集

k	A_k				B_k			
1	0.2	0.3	0.4	0.5	0.2	0.3	0.4	0.5
2	0.4	0.5	0.6	0.7	0.4	0.5	0.6	0.7
3	0.5	0.6	0.7	0.8	0.5	0.6	0.7	0.8
4	0.3	0.4	0.5	0.6	0.3	0.4	0.5	0.6
5	0.6	0.7	0.8	0.9	0.6	0.7	0.8	0.9

表2 连接权矩阵对1

$W_{4 \times 4}$				$U_{4 \times 4}$			
1	1	1	1	1	1	1	1
0.9	1	1	1	0.9	1	1	1
0.8	0.9	1	1	0.8	0.9	1	1
0.7	0.8	0.9	1	0.7	0.8	0.9	1

表3 连接权矩阵对2

$W_{4 \times 4}$				$U_{4 \times 4}$			
0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6
0.6	0.7	0.7	0.7	0.6	0.7	0.7	0.7
0.6	0.7	0.8	0.8	0.6	0.7	0.8	0.8
0.6	0.7	0.8	0.9	0.6	0.7	0.8	0.9

实验结果表明:(1)只要任意给定的模式对集能成为 $Max-$

$T_L FBAM$ 的平衡态集, 则本文提出的算法一定能找到所有使该模式对集成为平衡态集的连接权矩阵对中的最大者。(2)在实验中能被 $Max-T_L FBAM$ 可靠存储的模式对集却不能被 $Max-Min FBAM$ 可靠存储, 这也说明了 $Max-T_L FBAM$ 的价值。

4 结语

本文提出的学习算法直接给出了连接权的解析表达式, 故可以用作判别给定模式对集有否可能成为相应的 $Max-T_L FBAM$ 的平衡态集, 对带有此最大连接权矩阵对的网络的性能分析也提供了方便。另外, 该学习算法对其他的 t -模 T , $Max-T FBAM$ 也适用。

参考文献:

- [1] KOSKO B. Bidirectional associative memory in fuzzy expert systems [M]. Addison Wesley, 1987.
- [2] KOSKO B. Bidirectional associative memory[J]. IEEE Transactions on System, Man and Cybernetics, 1988, 18(1): 49 - 60.
- [3] 舒桂清, 肖平. 模糊联想记忆网络的增强学习算法[J]. 中国图形图像学报, 2003, 8(1): 84 - 89.
- [4] BELOHLAVEK R. Fuzzy logical bi-directional associative memory[J]. Information Science, 2000, 128(1): 91 - 103.
- [5] CHENG QS, FAN ZT. The stability problem for fuzzy bi-directional associative memories[J]. Fuzzy sets and systems, 2002, 132(1): 83 - 90.
- [6] 范周田, 钟义信. 模糊双向联想记忆网络的收敛性[J]. 电子学报, 2000, 28(4): 127 - 130.
- [7] 孙学全, 冯英浚. 多层感知器的灵敏度分析[J]. 计算机学报, 2001, 24(9): 952 - 958.
- [8] 陈松灿, 夏开军. 内连式复值双向联想记忆模型及性能分析[J]. 软件学报, 2002, 13(3): 433 - 437.
- [9] 修春波, 刘向东, 张宇河, 等. 一种新的双向联想记忆的学习算法[J]. 小型微型计算机系统, 2005, 26(6): 976 - 978.
- [10] 王敏, 王士同, 吴小俊. 新模糊形态学联想记忆网络的初步研究[J]. 电子学报, 2003, 31(5): 690 - 693.
- [11] 李换琴, 万百五. 大规模前馈神经网络的一种有效学习算法及其应用[J]. 信息与控制, 2003, 32(5): 403 - 407.
- [12] 程思蔚, 徐蔚鸿, 王勇, 等. 基于爱因斯坦 t -模的模糊联想记忆网络的学习算法[J]. 计算机工程与应用, 2006, 42(15): 40 - 41, 44.
- [13] 王士同. 神经模糊系统及其应用[M]. 北京: 北京航空航天大学, 1998.

(上接第 2987 页)

仍有一些多粒度时间下的数据挖掘问题值得我们研究。如:如何自动选取最合适的模板来表示周期模式, 多粒度时间下近似周期的挖掘算法, 噪音的处理, 时态自动机在周期挖掘的应用。我们将继续进行这方面的研究。

参考文献:

- [1] HAN J, DONG G, YIN Y. Efficient Mining of Partial Periodic Patterns in Time Series Databases[A]. Proceedings of 1999 International Conference on Data Engineering[C]. Sydney, Australia, 1999.
- [2] BERBERIDIS C, WALID AG, ATALLAH M. Multiple and Partial Periodicity Mining in Time Series Databases[A]. Proceedings of 15th European Conference on Artificial Intelligence (ECAI 2002) [C]. Lyon, France: IOS Press, 2002. 370 - 374.
- [3] YANG J, WANG W, YU PS. Mining Asynchronous Periodic Patterns

in Time Series Data[J]. IEEE Transactions Knowledge and Data Engineering, 2003, 15(3): 613 - 28.

- [4] YANG J, WANG W, YU PS. Discovering Higher Order Periodic Patterns[J]. Knowledge and Information Systems, 2004, 6(3): 243 - 268.
- [5] 孟志青. 时态数据采掘中的时态型与时间粒度研究[J]. 湘潭大学自然科学学报, 2000, 22(3): 1 - 4.
- [6] LI YG, WANG X, JAJOEDIA S. Discovering Temporal Patterns in Multiple Granularities[A]. Lecture Notes in Computer Science[C]. London, UK: Springer-Verlag, 2001, Vol 2007: 5 - 19.
- [7] BETTINI C, WANG X, JAJOEDIA S, et al. Discovering Frequent Event Patterns with Multiple Granularities in Time Sequences[J]. IEEE Transactions on Knowledge and Data Engineering, 1998, 10(2): 222 - 237.