一种时间序列频繁模式挖掘算法及其在 WSAN 行为预测中的应用

万 里 廖建新 朱晓民
(北京邮电大学网络与交换技术国家重点实验室 北京 100876)
(东信北邮信息技术有限公司 北京 100083)

摘 要：该文提出 FPM(Frequent Pattern Mining)算法充分考虑频繁模式在时间序列中出现次数和分布。基于这些不同分布的频繁模式扩展 MAMC(Mixed memory Aggregation Markov Chain)模型提出 FMAMC(Frequent pattern based Mixed memory Aggregation Markov Chain)模型。将 FPM 和 FMAMC 应用到实际的智能楼宇项目中，证明和现有算法相比 FPM 算法具有较好的时间性能，FMAMC 模型能够比 MAMC 模型更准确的预测 WSAN 节点行为。

关键词：数据挖掘；时间序列；频繁模式挖掘；无线传感器自组织网络节点行为预测；智能楼宇
中图分类号：TP393 **文献标识码**：A **文章编号**：1009-5896(2010)03-0682-05
DOI:10.3724/SP.J.1146.2009.00300

Time Series Frequent Pattern Mining Algorithm and its Application to WSAN Behavior Prediction

Wan Li Liao Jian-xin Zhu Xiao-min
(*State Key Laboratory of Networking and Switching Technology*, *Beijing University of Posts and Telecommunications*, *Beijing* 100876, *China*)
(*EBUPT Information Technology Co. Ltd*, *Beijing* 100083, *China*)

Abstract: A frequent pattern mining algorithm FPM (Frequent Pattern Mining) is proposed. FPM not only considered the frequency but also the distribution of the frequent pattern along the time series. Based on these different types of frequent patterns, MAMC (Mixed memory Aggregation Markov Chan) is extended to FMAMC (Frequent pattern based Mixed memory Aggregation Markov Chan) model. The proposed algorithm and model are applied to a smart building project, experiment and practice both demonstrate FPM is efficient than existing algorithms and FMAMC model could more accurately predict the node behavior in WSAN than MAMC.

Key words: Data mining; Time series, Frequent Pattern Mining (FPM); WSAN behavior prediction; Smart building

1 引言

如何对 WSAN 所采集的海量数据进行分析，已成为一个重要研究课题[1]。WSAN 节点行为预测是指根据各节点采集的历史数据预测下一时刻将采集数据的节点。已有聚类、关联规则挖掘等数据挖掘技术被应用到 WSAN 节点行为预测的研究中[1,2]。虽然可以用传统的时间序列挖掘算法和模型对 WSAN 采集的时间序列进行分析，但是，WSAN 采集的时间序列数据具周期性强和事件间的时序关联具有时间延迟的特点[2]。基于频繁模式的时间序列预测模型是目前一个研究热点[3,4]，但现有方法中没有考虑频繁模式在时间轴上的分布。

本文提出了一种考虑时间分布的频繁模式挖掘算法 FPM(Frequent Pattern Mining)和基于频繁模式及其时间分布的 FMAMC(Frequent pattern based Mixed memory Aggregation Markov Chain)预测模。在学习 FMAMC 模型时不仅根据事件序列在时间序列中出现的次数区分了频繁模式和噪声，并且根据频繁模式在时间轴上的不同分布区分其频繁程度。仿真实验和实际应用证明，本文提出的 FPM(Frequent Pattern Mining)算法能够快速发现时间序列中的频繁模式，FMAMC 模型和已有 MAMC 模型相比能更准确的预测时间序列中出现的事件。

2 频繁模式和 MAMC

频繁模式是指在时间序列中出现次数大于最小支持度的子序列。片段模式(episode)和序列模式(sequential pattern)均是频繁时间序列模式：前者表

2009-03-09 收到，2009-09-03 改回
国家杰出青年科学基金(60525110)，国家 973 计划项目(2007CB307100，2007CB307103)和电子信息产业发展基金(基于 3G 的移动业务应用系统)资助课题
通信作者：万里 wanly@ebupt.com

示某个子序列在一个时间序列中频繁出现，子序列出现一次其支持度增加 1；后者表示某个子序列在一个时间序列集合中频繁出现，子序列在集合中的某个时间序列出现一次或多次，其支持度只增加 1。

本文所研究的 FMAMC 模型将不同时间分布的频繁模式引入到MAMC模型中，MAMC模型[5−7]结合 Aggregate Markov(AM)和 Mixed Memory Markov(MMM)模型。AM 模型对状态空间中的状态进行分类，当前时刻的状态由它和上一个时刻的状态所属分类决定。MMM 描述状态之间转移时延，当前状态由 l 个时刻前状态决定。假设有时间序列 $X=\{x_1,x_2,\cdots,x_t,\cdots,x_T\}$，其中 $t\in\{1,\cdots,T\}$，$x_t\in E=\{e_1,e_2,\cdots,e_n\}$ 为当前时刻状态，E 为状态空间。则 MAMC 模型为

$$p(x_t\mid x_{t-1},\cdots,x_{t-L})=\sum_{l=1}^{L}p(l)p^l(x_t\mid x_{t-l})=\sum_{l=1}^{L}p(l)\sum_{k=1}^{K}p(x_t\mid k)p^l(k\mid x_{t-l})\quad(1)$$

其中 L 为最大时延，K 为状态空间中状态分类，$p(l)$ 表示时延长度为 l 的概率，$p^l(k\mid x_{t-l})$ 表示 l 个时刻前的状态 x_{t-l} 和当前时刻状态所属分类为 k 的概率，$p(x_t\mid k)$ 表示如果当前时刻状态所属分类为 k，当前状态为 x_t 的概率。

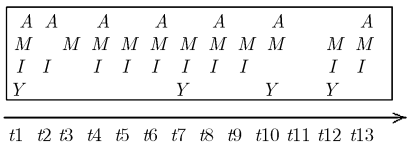
3　频繁模式发现

用矩阵表示多维时间序列，图 1(a)所示矩阵每一行对应一维，每一列对应时间轴上一个时刻，两个事件之间的时间间隔为两个事件所在列数差的绝对值。

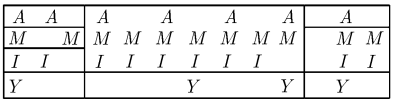
3.1 多维时间序列划分

FPM 算法根据 MDL(Minimum Description Length)原则划分多维时间序列，以描述某种划分所需编码长度为目标函数，所需编码长度越小的划分越好。和已有的基于 MDL 的时间序列分片算法不同[8]，本文的编码方法考虑了事件间时间间隔分布，而不只考虑事件出现次数分布。如图 1(b) 所示，图 1(a)中的时间序列被分为 3 个时间片，每个时间片内出现周期相似的事件序列被分到同一组。式(2)表示划分模型 M_i 对应时间片 S_i 的概率，X_j 为划分模型 M_i 中一个分组，$p(X_j)$ 为分组 X_j 中事件出现的概率分布，$p_{\tau(E_m)}(X_j)$ 为事件 E_m 后紧跟长度为 τ 的时间间隔在分组 X_j 中出现的概率分布，$n(E,S_i)$ 表示在分片 S_i 中事件 E 出现次数。

$$P(S_i\mid M_i)=\prod_{j=1}^{l}\prod_{E\in X_j}\prod_{m=1}^{n(E,S_i)}(p(X_j)p_{\tau(E_m)}(X_j))\quad(2)$$



(a) 矩阵表示多维时间序列



(b) 基于 MDL 原则对 (a) 中时间序列分片

图 1 时间序列表示方法及分片实例

$$\mathrm{LD}(S_i\mid M_i)=-\log_2 P(S_i\mid M_i)\quad(3)$$

$$\mathrm{LM}(M_i)=l\log_2 m+m\log_2 m\quad(4)$$

$$\mathrm{LS}(S_i,M_i)=\mathrm{LD}(S_i\mid M_i)+\mathrm{LM}(S_i\mid M_i)\quad(5)$$

$$\mathrm{TS}(S,M)=k\log_2 n+\sum_{i=1}^{k}\mathrm{LS}(S_i,M_i)\quad(6)$$

根据信息论，描述概率为 p 的事件需要编码长度为 $-\log_2 p$。$\mathrm{LD}(S_i\mid M_i)$ 表示模型 M_i 描述分片 S_i 需要比特数，$\mathrm{LM}(M_i)$ 为描述模型 M_i 本身需要的比特数，其中 l 为分片中分组个数，m 为分片中事件类型个数($m\log_2 m$ 比特用于描述 M_i 中不同类型事件所在行的排列顺序，$l\log_2 m$ 比特用于描述 m 个类型的事件在某种固定排列顺序上的分组)。长度为 n 的时间序列 $T=\{t_1,t_2,\cdots,t_n\}$ 被划分为 k 个分片需要确定 k 个边界，因为长度为 n 的时间序列有 n 个可能作为分片边界的时刻($\{t_2,t_3,\cdots,t_{n+1}\}$)，描述其中 k 个时刻作为边界的事件所需编码长度为

$$-k\log_2(1/n)=k\log_2 n$$

式(5)是描述每个时间分片所需编码长度，式(6)是描述整个时间序列分片所需编码长度。

GreedySegment 采用贪心算法架构，首先假设所有有事件发生的时刻均为分片边界，计算每个边界被移除后整个分片方案编码长度($\mathrm{TS}(S,M)$)的变化量($\Delta(b_i)$)，每次迭代移除使得编码长度减少最多的边界，当移除任何边界均无法减少编码长度时结束迭代。

GreedySegment 算法具体步骤如下：

输入：长度为 n 的时间序列 T

输出：编码代价最小的分片集合

1：For　T 中每个有事件出现的时刻 b_i

2：根据式(5)计算分片 $S[b_{i-1},b_i]$，$S[b_i,b_{i+1}]$，$S[b_{i-1},\ b_{i+1}]$ 的编码代 $\mathrm{LS}(S[b_{i-1},b_i],M_{i-1})$，$\mathrm{LS}(S[b_i,b_{i+1}],M_i)$，$\mathrm{LS}(S[b_{i-1},b_{i+1}],M_{i+1})$，计算移除边界 b_i 使得编码长度变化的增量：

$$\Delta(b_i)=\mathrm{LS}(S[b_{i-1},b_i],M_{i-1})+\log_2 n-\mathrm{LS}(S[b_i,b_{i+1}],M_i)-\log_2 n-\mathrm{LS}(S[b_{i-1},b_{i+1}],M_{i+1})-\log_2 n$$

3： 插入 $< b_i, \Delta(b_i) >$ 到哈希表 H
4：While H 不为空
5： 找到最小的 $\Delta(b_i)$
6： If $\Delta(b_i) < 0$
7： 移除 b_i 并更新 $\Delta(b_{i-1})$， $\Delta(b_{i+1})$
8： Else if $\Delta(b_i) \geq 0$
9： 终止 while 循环
10：输出 H

3.2 FPM 算法

FPM 算法的基本思想是基于长度为 l 的频繁模式找出长度为 $l+1$ 的频繁模式，算法从 $l=1$ 开始迭代，第 l 次迭代得到长度为 l 的频繁序列，将这些频繁序列作为前缀，在第 $l+1$ 次迭代中所找到的频繁项和这些长度为 l 的前缀组成长度为 $l+1$ 的频繁序列。计算过程中，根据候选频繁模式之间的包含关系将它们按格的形式存储。

FPM 算法具体步骤如下：

输入：时间序列 T，最小频繁片段支持度 $\min \mathrm{spt}_e$，最小频繁序列支持度 $\min \mathrm{spt}_s$

输出：所有极大频繁模式

1：初始化格 L $=\{\phi\}$，当前频繁序列前缀 $p_l = \phi$，迭代次数 $l=1$

2：调用 S=GreedySegment(T)

3：调用 FPM_Lattice(S， p_l， l)

4：输出格 L 中的所有频繁模式

其中，FPM_Lattice 是一叠代函数，具体步骤如下：

输入：时间序列分片集合 S，当前频繁序列前缀 p_l，迭代次数 l

1：计算当前时间分片集合 $S=\{S_1, S_2, \cdots, S_n\}$ 中所有事件的支持度 $\mathrm{spt}(e_i)$=$\{S_1(e_i), S_2(e_i), \cdots, S_n(e_i)\}$

2：若 $|\mathrm{spt}(e_i)| > \min \mathrm{spt}_s$ 或 $\exists S_j(e_i) > \min \mathrm{spt}_e$ 输出 $p_l = p_l + e_i$ 到格 L，并更新格 L

3：For(每个输出到 L 的 p_l)

4： if (p_l 在当前 L 中 $|\mathrm{spt}(p_l)| > \min \mathrm{spt}_s$ 或 $\exists S_j(p_l) > \min \mathrm{spt}_e$)

5： $S = S \mid e_i$ (e_i 为 p_l 中最后一个时刻的事件)， $l = l+1$

6： 调用 FPM_Lattice(S, p_l, l)

$S_j(e_i)$ ($S_j(p_l)$)表示事件 e_i (序列 p_l)在时间分片 S_j 中出现的次数， $|\mathrm{spt}(e_i)|$ ($|\mathrm{spt}(p_l)|$)表示 $\mathrm{spt}(e_i)$ ($\mathrm{spt}(p_l)$)中不为 0 元素的个数。 $\min \mathrm{spt}_s$ 和 $\min \mathrm{spt}_e$ 分别为最小频繁序列支持度和最小频繁片段支持度。 $S \mid e_i$ 表示 e_i 在时间分片集合 S 上的投影，即 $\forall S_j \in S$， $\mathrm{pos}(e_i, S_j)$ 表示 e_i 出现在时间分片 S_j 中的位置，则 $\exists S_j' \in S \mid e_i$， $S_j' = \{\mathrm{subSegment}(k, S_{jm}) \mid k \in \mathrm{pos}(e_i, S_{jm}), S_{jm} \in S_j\}$，其中 $\mathrm{subSegment}(k, S_{jm})$ 表示分片 S_{jm} 中从位置 k 到分片结束的子分片。

4 FMAMC 模型学习

如表 1 所示，利用最小频繁序列支持度 $\min \mathrm{spt}_s$ 和最小频繁片段支持度 $\min \mathrm{spt}_e$ 可以将时间序列中所有的子序列分为 4 类。将时间序列 S 中的每个事件用这 4 种类型进行标注，则得到具有类别标注的时间序列 $AS = \{< E_1, C_1 >, < E_2, C_2 >, \cdots, < E_T, C_T >\}$，其中 $E_i \in E$， $C_i \in K$， E 表示时间序列中事件类型的集合， K 表示表 1 中定义的 4 种事件分类。本文按照模式类型的频繁程度，由高到低地定义各频繁模式的优先级(完全频繁模式 > 频繁序列模式 > 频繁片段模式)，当一个事件被多种频繁模式包含时选择最高优先级模式类型进行标注。

表 1 根据 min spt$_s$ 和 min spt$_e$ 区分出 3 类频繁模式和噪声

	$> \min \mathrm{spt}_s$	$< \min \mathrm{spt}_s$
$> \min \mathrm{spt}_e$	完全频繁模式	频繁片段模式
$< \min \mathrm{spt}_e$	频繁序列模式	噪声

根据 component-wise EM-likelihood 算法[7]，从标注时间序列 AS 中估计 FMAMC 模型(式(1))参数。给定 FMAMC 模型 Φ，在时间序列中观测到 $X = \{x_1, x_2, \cdots, x_T\}$ 的概率是

$$p(X \mid \Phi) = \prod_{t=1}^{T} \log_2 \sum_{l=1}^{L} p(l) \sum_{k=1}^{K} p^l(k \mid x_{t-l}) p(x_t \mid k) \quad (7)$$

在模型 Φ 下 X 的对数似然值为

$$L(\Phi \mid X) \equiv \log_2 p(X \mid \Phi) = \sum_{s_0, s_1, \ldots, s_L = 1}^{S} N_{s_0, s_1, \cdots, s_L} \cdot \log_2 \sum_{l=1}^{L} p(l) \sum_{k=1}^{K} p^l(k \mid s_l) p(s_0 \mid k) \quad (8)$$

其中 $s_0, s_1, \cdots, s_L$ 表示当前时刻 t 及前 L 个时刻的事件序列 $(x_t = s_0, x_{t-1} = s_1, \cdots, x_{t-L} = s_L)$， $N_{s_0, s_1, \cdots, s_L}$ 表示该序列在时间序列中出现次数。式(8)中 L 为隐含变量， $K=4$，使式(8)取值最大的参数 L , $p(l)$， $p^l(k \mid s_l)$， $p(s_0 \mid k)$ 即为所求。

5 实验

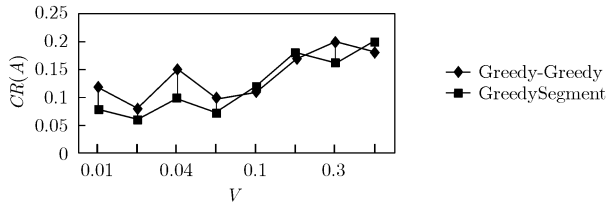
本文定义压缩率评价算法对时间序列分片的效果： $CR(A) = \dfrac{TS(S, M_A)}{TS(S, M_o)}$，其中 M_A 表示算法 A 输出的分片方案， S 为时间序列， $TS(S, M_A)$ 由式(6)计算得到。 M_o 表示一个时间间隔为一个分片的方案。 $CR(A) \in [0,1]$， $CR(A)$ 越小算法分片效果越好。

用准确率(precision)和召回率(recall)评价预测模型，好的预测模型当召回率增加时准确率也保持较高的水平。用 $\min \mathrm{spt}_s$ 和 $\min \mathrm{spt}_e$ 分别表示最小相

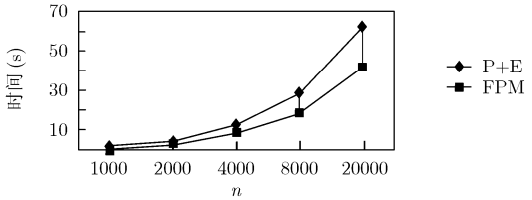
对频繁序列支持度和最小相对频繁片段支持度，如$(\min \mathrm{spt}_s,\ \min \mathrm{spt}_e)$。

5.1 仿真数据

按文献[8]的方法生成仿真数据集：仿真参数包括时间序列长度 n，时间序列维度 m，时间分片个数 k 和噪声级别 V。图 2(a)中数据集仿真参数：n=5000, m=10, k=20, V={0.01,0.02,0.04,0.08,0.1, 0.2, 0.3, 0.4}。图 2(b)中数据集仿真参数：n={1000，2000，4000，8000，20000}, m=10, k=20, V=0.02，最小相对支持度(0.16，0.2)。



(a) 不同噪声级别下压缩率比较



(b) FPM 和 P+E 算法时间效率比较

图 2 仿真数据集上时间序列划分算法和频繁模式挖掘算法实验

比较 GreedySegment 和 Greedy-Greedy[8]时间序列划分算法时间效率与压缩效果，从图 2(a)可以看出，GreedySegment 对时间序列的分片效果好于 Greedy-Greedy。将 FPM 和 PrefixSpan+WinMiner (P+W)进行比较。PrefixSpan+WinMiner 分别采用 PrefixSpan[9]和 WinMiner[10]挖掘出序列模式和片段模式，图 2(b)说明在时间序列较长时(频繁模式较多)FPM 算法具有较为明显的时间性能优势。

5.2 真实数据实验

本文实现了智能楼宇 WSAN 数据分析系统 EventMiner。实验采用的 WSAN 具有 6 个感应器节点，采集事件周期为 1 min。EventMiner 的核心功能之一是发现 WSAN 采集事件序列中的频繁模式，并预测下一时刻 WSAN 采集的事件。

对 2008 年 6 到 11 月采集的数据进行了分析，数据集中每条记录包括时间戳和在该时刻 6 个传感器节点所采集的事件。将 6 到 11 月的数据分为 3 个数据集，如表 2 所示。

由图 3 可以看出，本文提出的 GreedySegment 算法在数据集较小时，时间性能和 Greedy-Greedy 算法相近，当数据集较大时，GreedySegment 时间性能略低于 Greedy-Greedy 算法，其原因在于 GreedySegment 比 Greedy-Greedy 多了计算相邻事件间时间间隔分布的步骤。但从压缩率来看，Greedy Segment 的压缩效率远高于 Greedy-Greedy 算法，由此可见对周期性较强的时间序列进行分片时采用 GreedySegment 更为有效。

表 2 数据集

数据集	记录数	月份(月)
Dataset1	43359	6
Dataset2	87939	7,8
Dataset3	131072	9–11

由于结果类似，图 4 中只列出在 Dataset3 上 FPM 和 P+W 算法时间性能比较结果，可以看出 FPM 算法的时间性能优于 PrefixSpan+WinMiner，特别是在最小支持度较小时 FPM 优势更为明显，因为此时有更多的频繁模式输出。

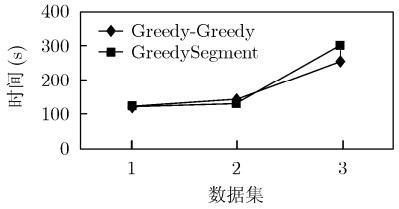
本文在 6 到 11 月数据集上比较了 FMAMC 和 MAMC 模型的预测性能。将前 5 个月的数据作为训练集，第 6 个月的数据作为测试集。由于在各种最小支持度设置下得到结果类似，图 5 只列出最小相对支持度为(0.16，0.2)的实验结果。

根据图 5(a)所示结果，选择 L=6，L=7，L=8 建立 FMAMC 模型，从图 5(b)看出 FMAMC 模型的预测效果明显优于 MAMC(K=1, L=7)。FMAMC 模型在召回率达到 40%时准确率仍在 60%以上，召回率达到 50%的时准确率仍保持在 30%以上，可见，在学习预测模型的时候区分频繁模式和噪声是必要的，FMAMC 对实际数据的预测是准确有效的。

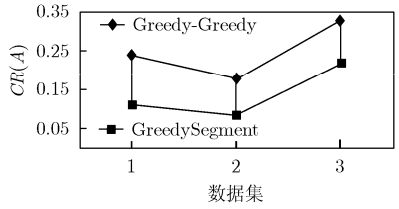
6　结束语

FPM 是一种考虑频繁模式分布的挖掘算法，对时间序列分片时，FPM 保证每个分片内同类型事件出现周期相似。基于频繁模式及其时间分布，扩展 MAMC 模型为 FMAMC 模型，用以描述时间序列中事件之间的转移概率及转移时间延迟。实验和实践证明，考虑频繁模式在时间轴上的分布是必要的，和已有算法相比，FPM 算法能够更高效的发现频繁模式。FMAMC 比 MAMC 能更准确地预测 WSAN 节点行为。

下一步工作将集中在挖掘事件间空间-时序(spatial-temporal)关系模式上。



(a) 分段算法时间性能



(b) 分段算法压缩率

图 3 在 Dataset1 至 3 上比较分段算法

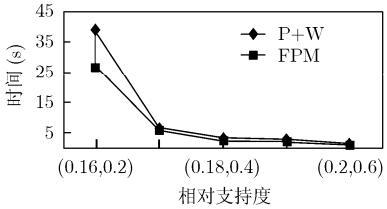
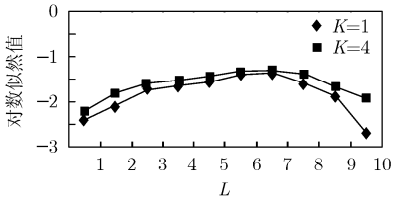
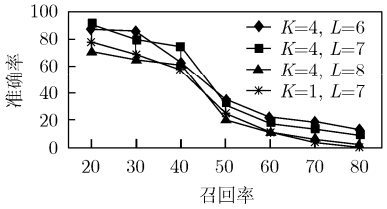


图 4 在 Dataset3 上比较 FPM 和 P+E 算法时间性能



(a) 不同时间延迟 L 对应对数似然值



(b) 不同参数设置下预测模型比较

图 5 在 6 到 11 月数据上比较预测模型性能

参 考 文 献

[1] Boukerche A. Handbook of Algorithms for Wireless Networking and Mobile Computing. Chapman & Hall/ CRC, 2005.

[2] Boukerche A and Samarah S. A novel algorithm for mining association rules in wireless Ad hoc sensor networks. *IEEE Transactions on Parallel and Distributed Systems*, 2008, 19(7): 143–160.

[3] Laxman S. Stream prediction using a generative model based on frequent episodes in event. Knowledge Discovery and Data Mining Conference, Las Vegas, US. Aug. 24–27, 2008: 101–110.

[4] Laxman S, Sastry P S, and Unnikrishnan K P. Discovering frequent episodes and learning Hidden Markov Models: A formal connection. *IEEE Transactions on Knowledge and Data Engineering*, 2005, 17(11): 1505–1517.

[5] Chudova D and Smyth P. Pattern discovery in sequences under a Markovian assumption. Knowledge Discovery and Data Mining Conference, Alberta, Canada, July 17–19 2002: 109–118.

[6] Alon J, Sclaroff S, Kollios G, and Pavlovic V. Discovering clusters in motion time series data. Computer Vision and Pattern Recognition Conference, Wisconsin, U S, June 2003: I-375-I–381.

[7] Wang X and Kabán A. A dynamic bibliometric model for identifying online communities. *Journal of Data Mining Knowledge Discovery*, 2008, 10(3): 42–68.

[8] Kiernan J and Terzi E. Constructing comprehensive summaries of large event sequences. Knowledge Discovery and Data Mining Conference, Las Vegas, U.S. Aug. 24–27, 2008: 131–140.

[9] Pei J, Han J, Pinto H, Chen Q, Dayal U, and Hsu M C. PrefixSpan: Mining sequential patterns efficiently by prefix-projected pattern growth. International Conference of Data Engineering, Heidelberg, Germany, 2001: 215–224.

[10] M´eger N and Rigotti C. Constraint-based mining of episode rules and optimal window sizes. 8th European Conference on Principles and Practice of Knowledge Discovery in Databases, Pisa, Italy, 2004: 313–324.

万 里： 男，1981年生，博士生，卡耐基梅隆大学(Carnegie Mellon University)访问学者，研究方向为网络智能化、人工智能、信号处理.

廖建新： 男，1965 年生，教授，博士生导师，研究方向为网络智能化.

朱晓民： 男，1974 年生，副教授，硕士生导师，研究方向为智能网、下一代业务网络、3G 核心网、协议工程等.