软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software,2012,23(5):1183−1194 [doi: 10.3724/SP.J.1001.2012.04094]

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel/Fax: +86-10-62562563

基于情节规则匹配的数据流预测*

朱辉生[1,2+], 汪 卫[2], 施伯乐[2]

[1](泰州师范高等专科学校 江苏 泰州 225300)

[2](复旦大学 计算机科学技术学院,上海 200433)

Data Stream Prediction Based on Episode Rule Matching

ZHU Hui-Sheng[1,2+], WANG Wei[2], SHI Bai-Le[2]

[1](Taizhou Teachers College, Taizhou 225300, China)

[2](School of Computer Science, Fudan University, Shanghai 200433, China)

+ Corresponding author: E-mail: zhs@fudan.edu.cn

Zhu HS, Wang W, Shi BL. Data stream prediction based on episode rule matching. *Journal of Software*, 2012, 23(5):1183−1194. http://www.jos.org.cn/1000-9825/4094.htm

Abstract: This paper proposes an algorithm called Predictor. This algorithm uses an automaton per matched episode rule with general form. With the aim of finding the latest minimal and non-overlapping occurrence of all antecedents, Predictor simultaneously tracks the state transition of each automaton by a single scanning of data stream, which can not only map the boundless streaming data into the finite state space but also avoid over-matching episode rules. In addition, the results of Predictor contain the occurring intervals and occurring probabilities of future episodes. Theoretical analysis and experimental evaluation demonstrate Predictor has higher prediction efficiency and prediction precision.

Key words: data stream; episode rule; the latest minimal and non-overlapping occurrence; prediction

摘 要: 提出了一种数据流预测算法 Predictor.该算法为每个待匹配的一般形式的情节规则分别使用了一个自动机,通过单遍扫描数据流来同时跟踪这些自动机的状态变迁,以搜索每个规则前件最近的最小且非重叠发生.这样不仅将无界的数据流映射到有限的状态空间,而且避免了对情节规则的过于匹配.另外,算法预测的结果是未来多个情节的发生区间和发生概率.理论分析和实验评估表明,Predictor 具有较高的预测效率和预测精度.

关键词: 数据流;情节规则;最近的最小且非重叠发生;预测

中图法分类号: TP311 **文献标识码**: A

数据流是由一系列值对(事件类型、发生时间)构成的序列[1],具有高速、无界、连续、时变的特点,并已广泛出现在网络安全监控[2]、金融证券管理[3]、事务日志分析[4]、传感器数据处理[5]等应用领域.数据流与静态数据库截然不同的特点,使得许多面向传统数据库的数据挖掘算法很难直接应用于数据流的环境,这就为数据流的分析带来了巨大的挑战.然而,历史流数据(实际是一个事件序列)中蕴含着大量的信息,研究历史流数据的潜在规律并应用这些规律对未来流数据作出预测,能够为许多现实应用提供重要的决策支持.

* 基金项目: 国家自然科学基金(61003001, 61103009); 国家重点基础研究发展计划(973)(2005CB321905)
收稿时间: 2011-03-28; 定稿时间: 2011-07-21

近年来,数据流预测的研究得到了学术界和工业界的广泛关注,并取得了一定的研究成果,学者们提出了许多基于回归分析的预测算法[5−9]和基于规则匹配的预测算法[10−14].其中,回归分析方法首先利用历史流数据来确定含有待定系数的函数表达式以建立反映自变量与因变量关系的回归模型,然后根据回归模型对数据流上一个固定区间的数据进行预测.这种方法预测速度快,但不能预测数据流上的非线性数据,因而具有一定的应用局限性;规则匹配方法是首先利用历史流数据来抽取其中反映两个情节之间因果联系的情节规则,然后利用这些情节规则在数据流上进行在线匹配,从而实现对未来流数据的预测.这种方法可以用来预测线性和非线性数据,应用范围较广,但现有的算法存在规则形式严格、预测区间受限或过时、规则过于匹配等问题.上述事实促成了本文的研究动机.下面通过一个实例来说明我们的研究目的.

假设γ=(⟨*AAB*⟩,⟨*CD*⟩,260,90%,8)是基于某图书馆 Web 服务器上一个历史文档阅读序列而得到的一个情节规则,其中,字母符号表示某个文档名,⟨*AAB*⟩和⟨*CD*⟩分别是规则γ的前件情节和后件情节;260 和90%分别是规则γ的支持度和置信度;7 是规则γ的窗口宽度,即情节⟨*AABCD*⟩必须在 7 个时间单位内发生.当前该 Web 服务器上的文档阅读流为 *DS*1=⟨(*A*,1),(*A*,2),(*A*,3),(*B*,4),(*A*,5),(*B*,6),(*A*,7),(*B*,8),(*C*,9),…⟩,其中的数字表示某读者对相应文档的开始阅读时间.通过在当前阅读流 *DS* 上匹配给定的规则γ,我们可以推知在区间[10,11]上将有 90%的概率出现 *D*,这种对读者未来阅读行为的预测将有助于图书馆向读者提供个性化的推荐服务.然而,在这个文档阅读流上基于情节规则的匹配来预测读者未来的阅读序列却遇到了如下挑战:

第一,一个情节规则的前件可能在当前流上发生多次,但并非所有的发生对预测产生意义.就数据流 *DS*1 而言,根据最小发生[15]、非重叠发生[16]、最小且非重叠发生[17]的定义,规则γ的前件⟨*AAB*⟩分别有以下 3 个发生集:

(1) 最小发生集{[2,4],[3,6],[5,8]}:该发生集的基数最大,表示在数据流上可能会对规则前件过于匹配,从而影响了预测的效率.

(2) 非重叠发生集{[1,4],[5,8]}:该发生集的基数最小,但其中的每个区间未必是最小发生,如区间[1,4]上存在⟨*AAB*⟩的另一次发生[2,4],因此容易导致遗漏预测,从而影响了预测的精度.例如,假设情节规则γ'=(⟨*AAB*⟩,⟨*CD*⟩,260,90%,9),当前数据流 *DS*2=⟨(*A*,1),(*A*,2),(*A*,3),(*B*,4),(*A*,5),(*B*,6),(*A*,7),(*F*,8),(*C*,9),…⟩,因规则前件⟨*AAB*⟩的非重叠发生集为{[1,4]},区间[1,4]的起始时间 1 与截止时刻 9 的间隔为 9,已达到规则γ'的窗口宽度,故无法预测 *D* 将出现在区间[10,10]上.

(3) 最小且非重叠发生集{[2,4],[5,8]}:该发生集不仅基数最小,而且其中的每个区间都是最小发生.显然,这种考虑既避免了对规则前件的过于匹配,也不会造成预测的遗漏.在这样的发生集中,最近一次发生[5,8]才是我们所关心的对预测产生意义的一次发生.因此,数据流上情节规则匹配的关键问题之一就是如何查找规则前件最近的最小且非重叠发生.

第二,在截止时刻之前,规则后件可能不完整地出现在规则前件的最近一次发生之后.例如,对于规则γ而言,在前件⟨*AAB*⟩的最近一次发生之后,后件⟨*CD*⟩中的⟨*C*⟩也已在截止时刻之前发生,此时预测结果不应是整个后件⟨*CD*⟩,而应是后件中余下的事件类型⟨*D*⟩.因此,数据流上情节规则匹配的关键问题之二就是如何在匹配规则前件的同时也能检查规则后件的发生情况.

第三,在实际应用中,待匹配的情节规则可能有多个,如何在数据流上同时查找多个规则前件最近的最小且非重叠发生,这将是我们面临的关键问题之三.

由上述分析可知,数据流上情节规则匹配的核心是数据流的查询问题,它类似于 XML 文档流上的订阅机制管理[18,19]和数据流上的复杂事件处理[20,21].但是订阅机制管理旨在数据流上检索用户感兴趣的 XML 文档,由于缺乏针对时间的运算符,因而也就无法表达带有时间约束的情节规则;而复杂事件处理旨在数据流上优化事件序列的查询计划以满足指定的时空资源约束等,它需要维护事件序列的每次发生,并非是事件序列的最后一次发生.因此,订阅机制管理及复杂事件处理的研究成果并不能直接解决我们的问题.

为此,本文提出了一种新的基于情节规则匹配的数据流预测算法 Predictor.该算法为每个待匹配的一般形式的情节规则分别使用了一个自动机,通过单遍扫描数据流来同时跟踪这些自动机的状态变迁,以搜索每个前件的最近的最小且非重叠发生,这样不仅将无界的数据流映射到有限的状态空间,而且避免了对情节规则的过

于匹配.另外,Predictor 预测的结果是未来多个情节的发生区间和发生概率.理论分析和实验评估证明,Predictor 具有较高的预测效率和预测精度.

1 相关工作

回归分析是在大量观察数据的基础上,利用数理统计方法建立因变量与自变量之间的关系函数表达式,从而预测事物未来的发展趋势.正因为如此,许多学者围绕基于回归模型的数据流预测问题展开了相应的研究,并取得了一定的成果,例如,Fletcher等人[5]将传感器的数据输出模拟成一个线性的随机系统,通过研究历史传感器数据的变化趋势来预测其中可能丢失的值;Jain 等人[6]以车辆跟踪信息、电力负荷、网络流量数据为例,提出了采用滤波对变化的数据流值进行预测的方法;Papadimitriou 等人[7]提出了 SPRINT 算法来增量发现 n 个数据流中数据之间的关联关系,并挖掘出多个数据流中反映数据流集合变化趋势的关键的隐藏变量;Pokrajac等人[8]提出了一种基于时空自回归模型的方法来分析历史数据上的小样本,并预测未来时刻的时空数据;Lazarevic 和 Kanapady[9]基于线性回归模型,使用曲线拟合的方法近似描绘了数据的变化规律.

尽管回归分析的方法预测速度快,但是它并不能预测数据流中的非线性数据.为此,许多学者又提出了基于规则匹配的数据流预测方法.2008 年,Laxman 等人[10]提出了基于事件序列上频繁情节构造的生成模型来预测数据流的方法.该方法分为两个阶段:第 1 阶段为训练阶段,首先,基于非重叠发生的情节支持度定义,根据目标事件类型 Y 在历史流数据中发现所有先于 Y 且发生在窗口宽度 W 中的频繁情节集 F^s,并将每个频繁情节表示为一个隐马尔可夫模型 HMM(hidden Markov model),然后,基于训练数据集和 EM 算法构造由多个 HMM 组成的混合 HMM,从而得到一个用来描述历史流数据特性的生成模型;第 2 阶段为预测阶段,首先,根据生成模型计算数据流上最近的长度为 W 的子序列的出现概率,然后,在这个 W 窗口中查找属于 F^s 的频繁情节的最近一次非重叠发生,若存在这样的一个发生,则表示目标事件类型 Y 将会出现在截止时刻的下一时间点上.显然,该方法存在如下不足:第一,待匹配的规则形式(前件为某频繁情节,后件为目标事件类型)严格,无法同时预测多个目标事件类型的发生;第二,预测阶段旨在查找情节并非最小的最近一次非重叠发生,容易导致预测区间过时;第三,只能预测数据流上一个时间点的数据,无法预测多个时间区间上若干情节的发生情况.2007 年,Cho 等人[11]提出了一个数据流预测算法 ToFel.该算法将待匹配规则的前件看作是一个拓扑图,并将数据流的演化看成是一个由符合拓扑关系的所有事件组成的队列,通过在队列上查找规则前件最近的最小发生来进行预测.由于队列维护时记录了规则前件的多次不完整发生,所以 ToFel 的时间代价大.另外,因一个情节规则对应了一个拓扑图,故 ToFel 的存储代价大.为此,Cho 等人[12]于 2008 年提出了一种采用后向检索规则前件策略的数据流预测算法 CBS-Tree.该算法以一个叶子节点个数固定的完全二叉树来存储和维护数据流,按照与规则前件拓扑结构相反的顺序在该完全二叉树中查找前件最近的最小发生以实现预测,算法所需的存储空间只与所有规则中最大的前件窗口宽度有关,而与待匹配的规则个数无关.为了提高算法 CBS-Tree 和 ToFel 的预测性能,Cho 等人在 2010 年首先引入两个优化技术来修剪前件的非最小发生以及区间超过规则窗口宽度的发生,提出了 CBS-Tree 的改进算法[13];然后,引入了一个优化技术来避免不必要的队列维护,提出了针对 ToFel 的改进算法 DeMO[13].然而,Cho 等人提出的算法也存在一些不足:第一,匹配规则形式严格,因为它限定了规则的前件必须为单映射情节,即不能含有相同的事件类型;第二,预测区间可能过时,因为它们在找到规则前件最近的最小发生时,给出的预测区间是规则的窗口宽度减去该次发生的起始时间,显然,这个预测区间可能出现在截止时刻之前,达不到预测的目的;第三,规则过于匹配,因为它考虑的是前件的最小发生,数据流上这样的发生次数要远远多于非重叠的发生.另外,这些文献中均未提及对多个规则同时匹配的实现细节.

为了选择最好的模型来预测 Web 文档,Yang 等人[14]首先对比了基于不同关联规则构建的序列分类器的差异,然后提出了评价预测质量的两个尺度:一个尺度是规则前件的选择方法,可以从当前的文档访问序列中分别选择其子集、子序列、最近子序列、子串、最近子串作为规则的前件;另一个尺度是整个规则的选择方法,可以分别根据最长匹配、最高置信度、最小估计误差来选择一个匹配规则.实验结果表明,选择最近子串作为前件并且选择最小估计误差规则作为匹配规则,将会达到最好的预测质量.然而,文献中并未给出在 Web 文档访问

流上如何对选定规则进行匹配的具体细节.

数据流上情节规则匹配的核心是数据流的查询问题.针对 XML 文档流的查询,Altinel 和 Franklin[18]提出了基于有限状态机、使用无冗余操作符来表达 XML 路径的方法,Peng 和 Chawath[19]提出了基于有限状态机、使用谓词来表达 XML 路径的方法.但这些方法缺乏针对时间的运算符,因而也就无法表达带有时间约束的情节规则.针对数据流上的复杂事件处理,Viglas 和 Naughton[20]研究了在数据流流速和连接运算符代价已知的前提下,如何来优化查询计划;Wu[21]等人提出了数据流上的查询模型 SASE,该模型借助基于栈结构的有限状态机和序列运算符、时间窗口运算符、否定运算符来动态维护待查序列的每次发生,并通过修剪策略来优化查询计划.

2 预备知识

2.1 基本概念

定义 1(事件、数据流、事件序列). 给定一个事件类型集$\varepsilon=\{E_1,E_2,\ldots,E_n\}$,一个事件就是一个二元组$(E,t)$,其中,$E\in\varepsilon$,$t$ 表示该事件的发生时间.定义在ε上的数据流 *DS* 是由无数个事件按发生时间先后排列的序列,表示为$DS=\langle(E_1,t_1),(E_2,t_2),\ldots,(E_s,t_s),\ldots\rangle$,其中,$t_i<t_j(1\leqslant i<j\leqslant s)$,$t_s$ 为截止时刻.数据流 *DS* 上的一个事件序列 *ES* 是由有限个事件按发生时间先后排列的序列,表示为 $ES=\langle(E'_1,t'_1),(E'_2,t'_2),...,(E'_k,t'_k)\rangle$, 其中, $t'_i<t'_j$ $(1\leqslant i<j\leqslant k)$.

定义 2(情节、单映射情节). 一个情节α是由若干事件类型组成的序列,表示为$\alpha=\langle E_1E_2\ldots E_k\rangle$,其中,元素 E_i $(1\leqslant i\leqslant k)\in\varepsilon$且对于所有的 i 和 $j(1\leqslant i<j\leqslant k)$满足 E_i 总是排列在 E_j 之前.情节α中的元素个数称为α的长度,记为$|\alpha|$.若一个情节中不含有相同的事件类型,则该情节称为单映射情节.

定义 3(单重情节、多重情节、全重情节). 设情节$\alpha=\langle E_1,E_2,\ldots E_k\rangle$:(1) 若 $E_1\neq E_2$,则α是一个重度为 1 的单重情节;(2) 若存在一个 $i(2\leqslant i<k)$满足 $E_1=E_2=\ldots=E_i$ 且 $E_i\neq E_{i+1}$,则α是一个重度为 i 的多重情节;(3) 若 $E_1=E_2=\ldots=E_k$,则α是一个全重情节.

定义 4(后缀、串接). 给定情节$\alpha=\langle E_1E_2\ldots E_k\rangle(k>1)$,称情节$\langle E_iE_{i+1}\ldots E_k\rangle(2\leqslant i\leqslant k)$是$\alpha$的 i-后缀,记为 $suffix(\alpha,i)$.给定情节$\alpha=\langle E_1E_2\ldots E_m\rangle$和 $\beta=\langle E'_1E'_2...E'_k\rangle$,则 $\langle E_1E_2...E_mE'_1E'_2...E'_k\rangle$ 称为α与β的串接,记为 $concat(\alpha,\beta)$.

定义 5(发生). 给定当前数据流 *DS* 和情节$\alpha=\langle E_1E_2\ldots E_k\rangle$,若 *DS* 上至少存在 1 个事件序列 $ES=\langle(E_1,t_1),(E_2,t_2),\ldots,(E_k,t_k)\rangle$,满足 $t_i<t_{i+1}(1\leqslant i\leqslant k-1)$,则称 *DS* 上发生(或出现)了情节$\alpha$,区间$[t_1,t_k]$称为$\alpha$在 *DS* 上的一次发生,其中,$t_1$ 和 t_k 分别称为该发生的起始时间和终止时间.

定义 6(最小发生). 设$[t_s,t_e]$是情节α在当前数据流 *DS* 上的一次发生,若 *DS* 上不存在α的另一次发生$[t'_s,t'_e]$,使得 $t_s<t'_s$ 且 $t'_e\leqslant t_e$,或 $t_s\leqslant t'_s$ 且 $t'_e<t_e$,即$[t'_s,t'_e]\subset[t_s,t_e]$,则称$[t_s,t_e]$是$\alpha$在 *DS* 上的一次最小发生.

定义 7(非重叠发生). 设$[t_s,t_e]$和$[t'_s,t'_e]$是情节α在当前数据流 *DS* 上的两次发生,若 $t_e<t'_s$ 或 $t'_e<t_s$,则称$[t_s,t_e]$和$[t'_s,t'_e]$是α在 *DS* 上的非重叠发生.

定义 8(最小且非重叠发生). 设$[t_s,t_e]$和$[t'_s,t'_e]$是情节α在当前数据流 *DS* 上的两次发生,若 $t_e<t'_s$ 或$t'_e<t_s$,且$[t_s,t_e]$和$[t'_s,t'_e]$都是α的最小发生,则$[t_s,t_e]$和$[t'_s,t'_e]$是α在 *DS* 上的最小且非重叠发生.

定义 9(最近的最小且非重叠发生). 情节α在当前数据流 *DS* 上所有最小且非重叠发生中的最后一次发生称为情节α在 *DS* 上最近的最小且非重叠发生.

定义 10(情节的支持度). 情节α在当前数据流 *DS* 上所有最小且非重叠发生组成的最大集合的基数称为α的支持度,记为$\alpha.\mathrm{sup}$.

定义 11(情节规则). 一个情节规则γ是一个五元组(l,r,s,c,w),其中,l,r,s,c,w 分别称为γ的前件、后件、支持度、置信度和窗口宽度.γ的支持度用于衡量γ的统计特性,它等于情节 $concat(\gamma.l,\gamma.r)$的支持度;$\gamma$的置信度用于衡量$\gamma$的可信程度,它等于情节 $concat(\gamma.l,\gamma.r)$的支持度与情节 $concat(\gamma.l)$的支持度的比值;γ的窗口宽度用于衡量γ的时间特性,表示情节 $concat(\gamma.l,\gamma.r)$发生时终止时间与起始时间的最大差值,亦即规则的前件和后件必须在指定的窗口宽度内发生.

2.2 问题描述

给定一组具有一般形式的情节规则和当前数据流 DS,则问题可以描述为:设计一个基于情节规则匹配的数据流预测算法,要求:(1) 只需单遍扫描 DS,以查找各规则前件最近的最小且非重叠发生;(2) 给出所有可能的预测结果.

3 数据流预测算法 Predictor

3.1 算法描述

本文抽出的算法 Predictor 旨在根据给定的情节规则集 R,通过单遍扫描当前数据流 DS 以查找每个规则 $r(r\in R)$前件最近的最小且非重叠发生,从而预测未来情节的发生区间及发生概率.为了能将无界的数据流映射到有限的状态空间,并同时跟踪多个情节规则前件在数据流上的发生,算法为每个待匹配情节规则分别使用了一个自动机.为方便起见,我们将对应情节规则γ的自动机简称为自动机γ,情节规则γ的前件与后件串接后所形成情节的第 i 个事件类型简称为$\gamma[i]$,则自动机γ的第 i 个状态对应着事件类型$\gamma[i]$.为了快速地访问所有自动机,我们借助了一个特殊的数据结构——$waits$[22].$waits$ 是一个含有$|\varepsilon|$(ε为数据流上的事件类型集)个元素的列表,每个元素 $waits(A)(A\in\varepsilon)$是一个由值对$(\gamma,j)$组成的集合,其中,$\gamma$为自动机名,$j$ 为该自动机的某个状态序号.若$(\gamma,j)\in waits(A)$,则表示自动机γ正在等待事件类型 A 在数据流上出现而进入状态 j.也就是说,通过 $waits(A)$可以访问正在等待事件类型 A 在数据流上出现而进入相应状态的所有自动机,即 $waits(A)$是访问这些自动机的一个入口.另外,算法 Predictor 还使用了表 1 中给出的几个符号.

Table 1 Some symbols and their explanations in algorithm Predictor

表 1 算法 Predictor 中的几个符号及含义

符号	含义
$\gamma.l$	规则γ的前件
$\gamma.r$	规则γ的后件
$\gamma.rep$	$\gamma.l$的重度
$\gamma.ind$	$\gamma.l$与$\gamma.r$串接后的情节中各事件类型的序号
$\gamma.tq$	记录$\gamma[1]$在数据流上出现时间的循环队列,其最大长度为$\gamma.rep$
$\gamma.ts$	$\gamma.l$在数据流上出现一次最小且非重叠发生时的起始时间
$\gamma.te$	$\gamma.l$在数据流上出现一次最小且非重叠发生时的终止时间

算法 Predictor 的基本思想是,将数据流的预测分为 3 个阶段:第 1 阶段为初始化阶段,即初始化各个自动机γ,将值对$(\gamma,1)$添加至集合 $waits(\gamma[1])$中,表示所有自动机正在等待其对应规则前件的第 1 个事件类型在数据流上出现而进入第 1 状态;第 2 个阶段为匹配阶段,按序扫描数据流上的各个事件,若当前出现事件(E_i,t_i),则将入口 $waits(E_i)$中的所有自动机转移至相应状态,并为各自动机进入下一状态或重新初始化做好准备工作,一旦自动机γ已经进入第$|\gamma.l|$个状态,则表示规则γ的前件已经发生,然后将检查其后件的发生或重新初始化自动机γ;第 3 阶段为预测阶段,依据各自动机在截止时刻前的最后状态来输出预测结果.

为了保证自动机状态转移的连续性,当事件类型 E_i 在数据流上出现时,我们不仅要从集合 $waits(E_i)$中删除元素(γ,j),表示已将自动机γ转移至状态 j,而且要将元素$(\gamma,j+1)$添加至集合 $waits(\gamma[j+1])$中,以便在数据流上出现事件类型$\gamma[j+1]$时能够将自动机γ转移至状态$j+1$.当集合 $waits(E_i)$为空时,表示针对事件(E_i,t_i)的出现的相关处理已经结束.然而,这样的处理方法在规则γ的前件或后件为非单映射情节时却发生了错误.例如,设$(\gamma,j)\in waits(E_i)$且$\gamma[j+1]=E_i$,当事件(E_i,t_i)出现在数据流上时,由于 $waits(\gamma[j+1])=waits(E_i)$,则自动机$\gamma$被连续转移至状态 j 和 $j+1$.为了避免因一个事件的出现而使自动机发生多次的状态转移,正确的处理方法是:在事件(E_i,t_i)出现时,首先删除 $waits(E_i)$中的元素(γ,j),然后将$(\gamma,j+1)$临时保存在一个集合 bag 中,只有在集合 $waits(E_i)$为空并处理数据流上的下一个事件之前,才将 bag 中的元素全部移至 $waits(\gamma[j+1])$中.下面是 Predictor 的伪代码.

Algorithm. *Predictor*(R,DS).

Input: R: A set of episode rules.

DS: The current data stream $\langle(E_1,t_1),(E_2,t_2),...,(E_n,t_n),...\rangle$.

Output: R': A set of prediction results.

```
1:  Let bag=∅
2:  For each e∈ε do
3:      Let waits(e)=∅
4:  For each γ∈R do
5:      Add (γ,1) to waits(γ[1])
6:      Let γ.ts=γ.ind=0
7:      Let γ.te=−1
8:  For i=1 to n do
9:      For each (γ,j)∈waits(E_i) do
10:         If j≤|γ.l|
11:             MatchLHS(γ,j)
12:         Else
13:             MatchRHS(γ,j)
14: R′=Reporter(R)
15: Return R′
```

3.2 前件匹配

算法 Predictor 的关键是如何查找所有规则前件最近的最小且非重叠发生.对于自动机γ而言,其初始化时$\gamma.ts$=0,$\gamma.te$=−1,表示前件$\gamma.l$ 在数据流上尚无一次发生.自动机γ重新初始化的目的旨在检查前件$\gamma.l$ 的下一次发生,此时,$\gamma.te$ 保存着$\gamma.l$ 上一次发生的终止时间,只有当自动机γ再次进入状态$|\gamma.l|$时,$\gamma.te$ 才被赋以新值.所以,前件$\gamma.l$ 在数据流上发生一次的条件是$\gamma.te\geqslant\gamma.ts$.

当前件匹配时,若$(\gamma,1)\in waits(E_i)$,则当事件(E_i,t_i)在数据流上出现时,自动机γ转移至第 1 状态.然而 t_i 未必是$\gamma.l$ 当前可能存在的一次发生的起始时间,因为我们总是考虑$\gamma.l$ 的最小且非重叠发生.为了能够正确记录$\gamma.l$ 一次发生的起始时间$\gamma.ts$,我们为每个规则γ设置了一个最大长度为$\gamma.rep$(即前件$\gamma.l$ 的重度)的循环队列$\gamma.tq$,下面分 3 种情形来讨论$\gamma.tq$ 的工作原理:

情形 1:$\gamma.l$ 是一个长度等于 1 的单重情节.当$\gamma[1]$出现时,$\gamma.tq$ 记录了$\gamma[1]$的发生时间,并且自动机γ转移至状态$|\gamma.l|$.此时,将$\gamma.tq$ 中的队首元素赋给$\gamma.ts$,并清空队列$\gamma.tq$,则$\gamma.ts$ 一定是$\gamma.l$ 刚刚发生的起始时间(也是终止时间).

情形 2:$\gamma.l$ 是一个多重情节或长度大于 1 的单重情节.令$\gamma.l$ 中第 1 种与$\gamma[1]$不同的事件类型为 E',则$\gamma.tq$ 记录了在 E'出现之前$\gamma[1]$先后发生的时间,当自动机γ进入状态$|\gamma.l|$时,将$\gamma.tq$ 中的队首元素赋给$\gamma.ts$,并清空队列$\gamma.tq$,此时,$\gamma.ts$ 一定是$\gamma.l$ 当前这次发生的起始时间.此种情形下,为了使$\gamma.tq$ 能够记录在 E'出现之前$\gamma[1]$最近一些发生的时间,应该等到 E'出现时(而不是$\gamma[1]$出现第$\gamma.rep$ 次时)从 $waits(\gamma[1])$中删除$(\gamma,\gamma.rep)$.

情形 3:$\gamma.l$ 是一个长度为 k 的全重情节.表示$\gamma.l$ 中不存在一个与$\gamma[1]$不同的事件类型,则$\gamma.tq$ 记录了$\gamma[1]$先后发生的时间,当自动机γ进入状态$|\gamma.l|$时,将$\gamma.tq$ 中的队首元素赋给$\gamma.ts$,并清空队列$\gamma.tq$,此时,$\gamma.ts$ 一定是$\gamma.l$ 当前这次发生的起始时间.

为便于理解,下面通过一个示例来说明前件匹配的过程.设情节规则γ的前件为$\langle AAB\rangle$,当前数据流为 $DS1$,则扫描 $DS1$ 至事件$(B,4)$时,队列$\gamma.tq$ 的队首元素为 2,自动机γ已进入第 3 状态,得到$\langle AAB\rangle$的第 1 次最小发生[2,4];接着,清空队列$\gamma.tq$ 并重新初始化自动机γ,当扫描 $DS1$ 至事件$(B,8)$时,队列$\gamma.tq$ 的队首元素为 5,自动机γ再次进入第 3 状态,得到$\langle AAB\rangle$的第 2 次最小发生[5,8],它是与[2,4]非重叠的一次发生,也是$\langle AAB\rangle$在 $DS1$ 上的最近一次发生.下面是前件匹配的伪代码.

Procedure *MatchLHS*(γ,j)

Input: γ: An automaton corresponding to episode rule γ.

j: A state into which automaton γ is transiting.

Objective: Transit automaton γ into state *j*.

1: If $j \leqslant \gamma.rep$
2: Add t_i to $\gamma.tq$
3: If $j \neq \gamma.rep$
4: Remove (γ,j) from $waits(E_i)$
5: Else if $\gamma.rep=|\gamma.l|$
6: Remove (γ,j) from $waits(E_i)$
7: If $j=\gamma.rep+1$
8: Remove $(\gamma,j-1)$ from $waits(E_i)$
9: Let $j'=j+1$
10: If $j=|\gamma.l|$
11: Let $j'=1$
12: Let $\gamma.ts=\gamma.tq[1]$
13: Let $\gamma.te=t_i$
14: Let $\gamma.ind=j$
15: Empty $\gamma.tq$
16: Add $(\gamma,j+1)$ to $waits(\gamma[j+1])$
17: If $\gamma[j']=E_i$
18: Add (γ,j') to bag
19: Else
20: Add (γ,j') to $waits(\gamma[j'])$

3.3 后件匹配

由于对预测产生意义的是规则前件最近的最小且非重叠发生以及随后可能出现的规则后件的不完整发生,所以我们在检查规则前件下一次发生的同时,还需要检查规则后件的发生情况.

设待匹配的情节规则为γ,当$\gamma.te \geqslant \gamma.ts$ 时,自动机γ已转移至状态$|\gamma.l|$,表示前件$\gamma.l$ 在数据流上出现了一次发生,此时,必须将$(\gamma,|\gamma.l|+1)$添加至集合 $waits[|\gamma.l|+1]$中,以便于随后对后件$\gamma.r$ 发生情况的检查.考虑到一个集合$waits[E_i]$中可能会同时存在元素$((\gamma,j_1)$和(γ,j_2),其中,$j_1 \leqslant |\gamma.l|$,$j_2>|\gamma.l|$,随着事件类型 E_i 的出现,自动机γ要分别转移至状态 j_1 和状态 j_2,这相当于自动机γ同时产生了两个例程,一个用于前件的匹配,另一个用于后件的匹配.当发生如下两种情形时,后件的匹配必须重新开始或立即停止:

情形 1:在后件匹配期间出现了前件的一次发生.这种情形下,无论后件的发生情况如何,都应重新开始后件的匹配.

情形 2:出现了后件的一次完整发生.这种情形下,无论随后是否存在后件的其他完整或不完整发生,都应立即停止后件的匹配,因为后件的一次完整发生意味着前件的当前发生对预测不会产生任何意义.

下面给出后件匹配的伪代码.

Procedure *MatchRHS*(γ,j)

Input: γ: An automaton corresponding to episode rule γ;
j: A state into which automaton γ is transiting.
Objective: transit automaton γ into state j.
1: If $\gamma.te \geqslant \gamma.ts$ and $j \leqslant |\gamma.l|+|\gamma.r|$
2: Remove (γ,j) from $waits(E_i)$
3: Let $j'=j+1$
4: Let $\gamma.ind=j$
5: If $\gamma[j']=E_i$
6: Add (γ,j') to bag
7: Else
8: Add (γ,j') to $waits(\gamma[j'])$

3.4 结果输出

在实际应用中,当前数据流与历史流数据有着许多相同的特性,根据历史流数据抽取得到的一个情节规则,若它具有较高的置信度,则基于该情节规则在数据流上的匹配而预测的未来情节也具有等同于置信度的发生概率.基于这样一个事实,我们可以根据匹配结束后满足条件"存在前件最近的最小且非重叠发生,且随后不存在后件的完整发生"的情节规则来预测未来情节的发生区间及发生概率.下面给出结果输出的伪代码.

Procedure *Reporter*(*R*)

Input: R: A set of episode rules after matching over DS.

Output: R': A set of prediction results.

1: Let $R'=\varnothing$

2: For each $\gamma\in R$ do

3: If $\gamma.te\geqslant\gamma.ts$ and $(t_n-\gamma.ts)\leqslant\gamma.w$

4: Let $\alpha=concat(\gamma.l,\gamma.r)$

5: Let $f=suffix(\alpha,\gamma.ind+1)$

6: Let $t=\gamma.w-(t_n-\gamma.ts)$

7: Let $p=\gamma.c$

8: Add (f,t,p) to R'

9: Return R'

4 算法复杂度分析

设ε为给定的事件类型集,R 为给定的情节规则集,DS 为给定的数据流,则算法 Predictor 的复杂度分析如下:

定理 1. Predictor 的时间复杂度为 $O(|R|\cdot|DS|)$.

证明:对$|R|$个情节规则的匹配是算法的主要时间代价,匹配时每个情节规则使用了一个自动机,每个自动机需要时间 $O(|DS|)$以发现其对应规则前件的最近的最小且非重叠发生,所以算法 Predictor 的时间复杂度为 $O(|R|\cdot|DS|)$. □

定理 2. Predictor 的空间复杂度为 $O(|\varepsilon|+|R|)$.

证明:用于规则匹配的自动机访问入口及自动机的个数分别为$|\varepsilon|$,$|R|$,前件匹配或后件匹配时,*bag* 中最多临时存储了$|R|$个情节规则信息,结果输出时最多包括$|R|$个输出结果.因此,算法 Predictor 的空间复杂度为 $O(|\varepsilon|+|R|)$. □

5 实验评估

我们通过 6 组实验对比了 Predictor 与算法 DeMO[13]的预测精度和时空性能.实验采用的硬件环境为 2.13 GHz Intel(R) Core(TM) i3 CPU,内存 2 GB,操作系统为 Windows XP,程序采用 Java 实现.

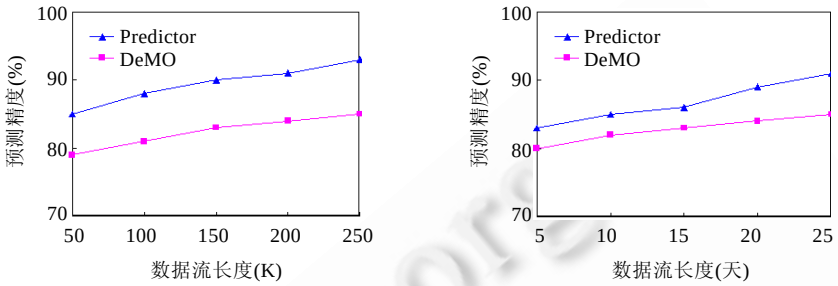
5.1 数据集

对于合成数据集,我们首先使用 IBM 合成数据生成器 Quest Market-Basket 的修改版生成了每个交易为单个项的交易序列,通过设置 D=0.001,C=300000,N=20,S=300000,其中,参数 D 表示交易序列的个数(单位为 1 000),C 表示每个交易序列中交易的平均个数,N 表示所有交易项的类型种数(单位为 1 000),S 为最长交易序列中交易的平均个数,这样就得到了一个 20 000 种交易项类型上的由 300 000 个交易组成的交易序列.然后,为该交易序列中的每个交易依次赋上一个连续的正整数以作为每个交易发生的时间戳,这样,我们就构造了一个 20K 种事件类型上的由 300K 个事件组成的事件序列.

对于真实数据集,考虑到作为国内最具影响力的知识传播与数字化学习平台,中国知网 CNKI[23]为全社会提供了最丰富、最全面的文献资源.为了能够发现 CNKI 中相关文献之间的引用关系,并为广大学者展开相关研究提供最可靠的个性化服务,我们选用了 CNKI 的一个 Web 服务器上从 2010 年 11 月 1 日至 2010 年 11 月 30 日的日志数据,该日志数据包括了相关读者对 132 885 种不同文献的 211 665 个阅读记录.

5.2 实验结果

实验 1(预测精度 vs. 数据流长度). 为了考察算法在数据流上预测的准确性,我们将预测精度定义为预测结果中的正确数占预测结果总数的比例.实验时采用类似于机器学习中的交叉验证方法,从300K合成数据集和30天真实数据集中分别随机地选择5个长度不等的事件序列作为当前数据流 DS_i,而原数据集上 DS_i 后面的剩余序列将作为判断算法预测是否正确的验证序列.我们将300K合成数据集(支持度和置信度分别设定为800和60%)和30天真实数据集(支持度和置信度分别设定为7和60%)上抽取的4 478和315个情节规则作为待匹配的情节规则,从而得到数据流长度对算法预测精度的影响,如图1所示.



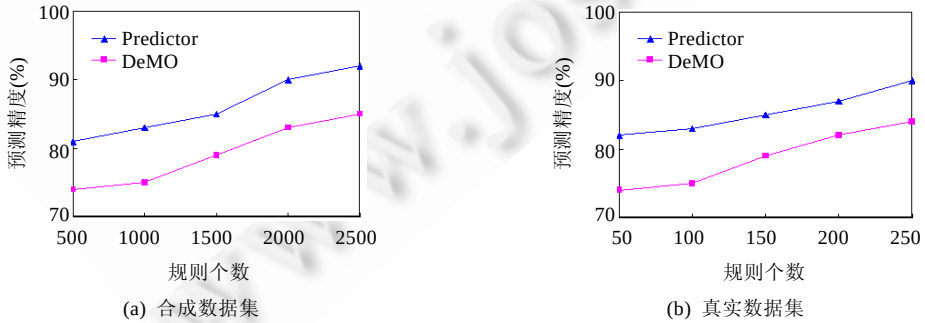
(a) 合成数据集 (b) 真实数据集

Fig.1 Prediction precision vs. length of data stream

图1 预测精度 vs. 数据流长度

可以看出,Predictor与DeMO在各个数据流上均具有较高的预测精度,这是因为产生待匹配规则的历史流数据与当前数据流具有很大的相似性(如各事件类型的分布);而且随着数据流长度的增加,两种算法的预测精度都在提高,这是由于在数据流长度增加时两种算法发现了更多规则前件最近的最小且非重叠发生,但正确预测的比例增速更快.我们还观察到,Predictor的预测精度要优于DeMO,这是因为DeMO是通过查找待匹配规则前件最近的最小发生来预测后件的发生情况,一方面,在当前数据流上可能存在规则前件许多重叠的最小发生,因而导致了对该情节的过于匹配(即过拟合);另一方面,只要存在规则前件最近的最小发生,DeMO就给出了一个预测结果,而这个预测结果可能不是当前数据流未来情节的一次发生.与DeMO不同的是,Predictor是根据待匹配规则前件最近的最小且非重叠发生(避免了对规则前件的过于匹配),结合数据流的截止时刻和情节规则的窗口宽度来预测未来情节的发生情况.

实验 2(预测精度 vs. 规则个数). 我们首先从300K合成数据集和30天真实数据集中分别选择一个长度为200K的事件序列作为两个当前数据流 DS_1 和 DS_2,并将原数据集上 DS_i 后面的剩余序列作为判断算法预测正确与否的验证序列,然后,分别从4 478个合成数据情节规则和315个真实数据情节规则中随机选择各自的子集作为待匹配的情节规则,从而得到规则个数对算法预测精度的影响,如图2所示.



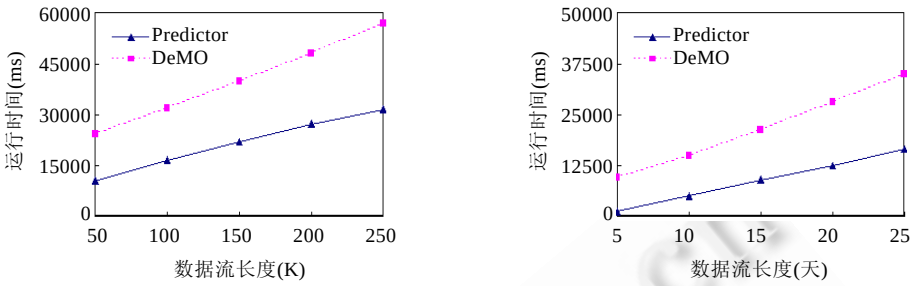
(a) 合成数据集 (b) 真实数据集

Fig.2 Prediction precision vs. number of rules

图2 预测精度 vs. 规则个数

可以看出,两种算法均具有较高的预测精度,而且随着规则个数的增加,两种算法的预测精度都在提高.同时我们还观察到,Predictor 比 DeMO 具有更高的的预测精度.产生这些现象的原因与实验 1 的解释相同.

实验 3(运行时间 vs. 数据流长度). 由实验 1,我们得到了数据流长度对算法预测时间的影响,如图 3 所示.



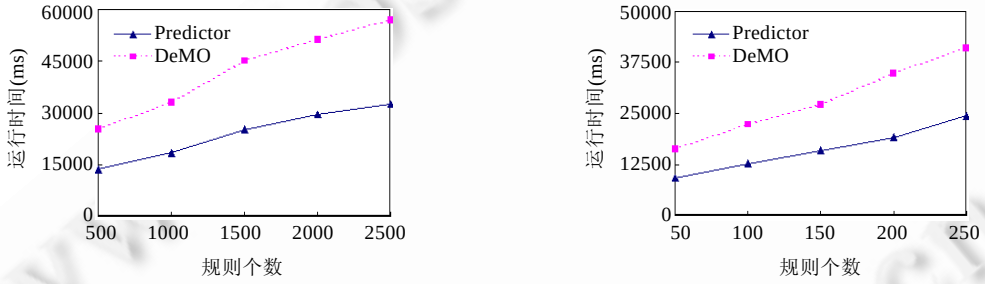
(a) 合成数据集 (b) 真实数据集

Fig.3 Running time vs. length of data stream

图 3 运行时间 vs. 数据流长度

可以看出,两种算法的预测时间都随着数据流长度的增加而增加,但 Predictor 要优于 DeMO,这是因为后者存在对规则前件的过于匹配问题.

实验 4(运行时间 vs. 规则个数). 由实验 2,我们得到了规则个数对算法预测时间的影响,如图 4 所示.

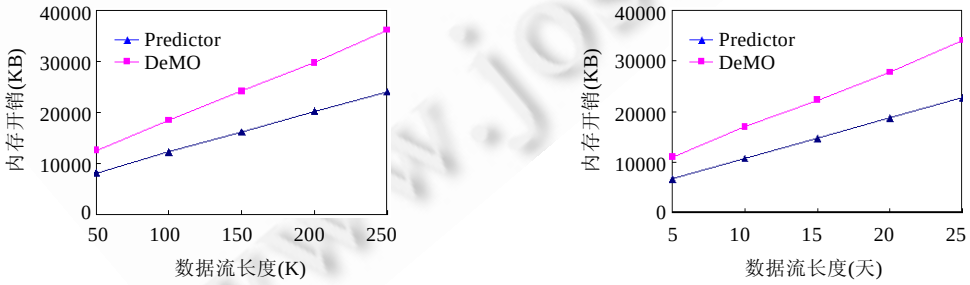


(a) 合成数据集 (b) 真实数据集

Fig.4 Running time vs. number of rules

图 4 运行时间 vs. 规则个数

实验 5(内存开销 vs. 数据流长度). 由实验 1,我们得到了数据流长度对算法内存开销的影响,如图 5 所示.可以看出,两种算法的内存开销都随着数据流长度的增加而增加,但 Predictor 要优于 DeMO,这也是因为后者存在对规则前件的过于匹配问题.

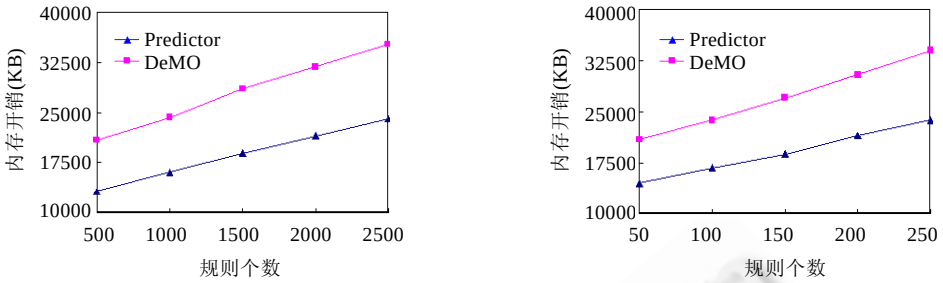


(a) 合成数据集 (b) 真实数据集

Fig.5 Memory requirement vs. length of data stream

图 5 内存开销 vs. 数据流长度

实验 6(内存开销 vs. 规则个数). 由实验 2,我们得到了如图 6 所示的规则个数对算法内存开销的影响.



(a) 合成数据集 (b) 真实数据集

Fig.6 Memory requirement vs. number of rules

图 6 内存开销 vs. 规则个数

6 结 论

研究历史流数据的潜在规律并应用这些规律对未来流数据作出预测,能够为许多现实应用提供重要的决策支持.针对现有数据流预测算法存在的不足,本文提出了一种新的基于情节规则匹配的数据流预测算法 Predictor.该算法为每个待匹配的一般形式的情节规则分别使用了一个自动机,通过单遍扫描数据流来同时跟踪这些自动机的状态变迁,以搜索每个规则前件最近的最小且非重叠发生.这样不仅将无界的数据流映射到有限的状态空间,而且避免了对情节规则的过于匹配.另外,Predictor 预测的结果是未来多个情节的发生区间和发生概率.理论分析和实验评估表明,Predictor 具有较高的预测效率和预测精度.

References:

[1] Babcock B, Babu S, Datar M, Motwani R, Widom J. Models and issues in data stream systems. In: Popa L, ed. Proc. of the 21st ACM SIGACT-SIGMOD-SIGART Symp. on Principles of Database Systems. New York: ACM Press, 2002. 1−16. [doi: 10.1145/543613.543615]

[2] Julisch K, Dacier M. Mining intrusion detection alarms for actionable knowledge. In: Proc. of the 8th ACM SIGKDD Int'l Conf. on Knowledge Discovery in Data Mining. New York: ACM Press, 2002. 366−375. [doi: 10.1145/775047.775101]

[3] Ng A, Fu AW. Mining frequent episodes for relating financial events and stock trends. In: Whang KY, Jeon J, Shim K, Srivastava J, eds. Proc. of the 7th Pacific-Asia Conf. on Knowledge Discovery and Data Mining. 2003. 27−39. [doi: 10.1007/3-540-36175-8_4]

[4] Cortes C, Fisher K, Pregibon D, Rogers A. Hancock: A language for extracting signatures from data streams. In: Proc. of the 6th ACM SIGKDD Int'l Conf. on Knowledge Discovery in Data Mining. New York: ACM Press, 2000. 9−17. [doi: 10.1145/347090.347094]

[5] Fletcher AK, Rangan S, Goyal VK. Estimation from lossy sensor data: Jump linear modeling and Kalman filtering. In: Proc. of the 3rd Int'l Symp. on Information Processing in Sensor Networks. 2004. 251−258. [doi: 10.1145/984622.984659]

[6] Jain A, Chang EY, Wang YF. Adaptive stream resource management using Kalman Filter. In: Weikum G, Konig AC, Debloch S, eds. Proc. of the 2004 ACM SIGMOD Int'l Conf. on Management of Data. New York: ACM Press, 2004. 11−22. [doi: 10.1145/1007568.1007573]

[7] Papadimitriou S, Sun JM, Faloutsos C. Streaming pattern discovery in multiple time series. In: Bohm K, Jensen CS, Haas LM, Kersten ML, Larson PA, Ooi BC, eds. Proc. of the 31st Int'l Conf. on Very Large Data Bases. Toronto: Morgan Kaufmann Publishers, 2005. 697−708.

[8] Pokrajac D, Hoskinson RL, Obradovic Z. Modeling spatial temporal data with a short observation history. Knowledge and Information Systems, 2003,5(3):368−386. [doi: 10.1007/s10115-002-0094-1]

[9] Lazarevic A, Kanapady R, Kamath C. Effective localized regression for damage detection in large complex mechanical structures. In: Kim W, Kohavi R, Gehrke J, DuMouchel W, eds. Proc. of the 10th ACM SIGKDD Int'l Conf. on Knowledge Discovery in Data Mining. New York: ACM Press, 2004. 450−459. [doi: 10.1145/1014052.1014103]

[10] Laxman S, Tankasali V, White RW. Stream prediction using a generative model based on frequent episodes in event sequences. In: Li Y, Liu B, Sarawagi S, eds. Proc. of the 14th ACM SIGKDD Int'l Conf. on Knowledge Discovery in Data Mining. New York: ACM Press, 2008. 453−461. [doi: 10.1145/1401890.1401947]

[11] Cho CW, Zheng Y, Chen ALP. Continuously matching episode rules for predicting future events over event streams. In: Dong GZ, Lin XM, Wang W, Yang Y, Yu JX, eds. Proc. of the 9th Asia-Pacific Web Conf. and the 8th Int'l Conf. on Web-Age Information Management. 2007. 884−891. [doi: 10.1007/978-3-540-72524-4_91]

[12] Cho CW, Zheng Y, Wu YH, Chen ALP. A tree-based approach for event prediction using episode rules over event streams. In: Bhowmick SS, Kung J, Wagner R, eds. Proc. of the 19th Int'l Conf. on Database and Expert Systems Applications. 2008. 225−240. [doi: 10.1007/978-3-540-85654-2_24]

[13] Cho CW, Wu YH, Yen SJ, Zheng Y, Chen AL P. On-Line rule matching for event prediction. The VLDB Journal, Online First, 2010. [doi: 10.1007/s00778-010-0197-3]

[14] Yang Q, Li TY, Wang K. Building association-rule based sequential classifiers for Web-document prediction. Data Mining and Knowledge Discovery, 2004,8(3):253−273. [doi: 10.1023/B:DAMI.0000023675.04946.f1]

[15] Mannila H, Toivonen H. Discovering generalized episodes using minimal occurrences. In: Simoudis E, Han JW, Fayyad UM, eds. Proc. of the 2nd ACM SIGKDD Int'l Conf. on Knowledge Discovery in Data Mining. New York: ACM Press, 1996. 146−151.

[16] Laxman S, Sastry PS, Unnikrishnan KP. Discovering frequent episodes and learning hidden Markov models: A formal connection. IEEE Trans. on Knowledge and Data Engineering, 2005,17(11):1505−1517. [doi: 10.1109/TKDE.2005.181]

[17] Zhu HS, Wang P, He XM, Li YJ, Wang W, Shi BL. Efficient episode mining with minimal and non-overlapping occurrences. In: Webb GI, Liu B, Zhang CQ, Gunopulos D, Wu XD, eds. Proc. of the 10th Int'l Conf. on Data Mining. Sydney: IEEE Computer Society, 2010. 1211−1216. [doi: 10.1109/ICDM.2010.25]

[18] Altinel M, Franklin MJ. Efficient filtering of XML documents for selective dissemination of information. In: Abbadi AE, Brodie ML, Chakravarthy S, Dayal U, Kamel N, Schlageter G, Whang KY, eds. Proc. of the 26th Int'l Conf. on Very Large Data Bases. Toronto: Morgan Kaufmann Publishers, 2000. 53−64.

[19] Peng F, Chawathe SS. XPath queries on streaming data. In: Halevy AY, Ives ZG, Doan AH, eds. Proc. of the 2003 ACM SIGMOD Int'l Conf. on Management of Data. New York: ACM Press, 2003. 431−442. [doi: 10.1145/872757.872810]

[20] Viglas SD, Naughton JF. Rate-Based query optimization for streaming information sources. In: Franklin MJ, Moon B, Ailamaki A, eds. Proc. of the 2002 ACM SIGMOD Int'l Conf. on Management of Data. New York: ACM Press, 2002. 37−48. [doi: 10.1145/564691.564697]

[21] Wu E, Diao YL, Rizvi S. High-Performance complex event processing over streams. In: Chaudhuri S, Hristidis V, Polyzotis N, eds. Proc. of the 2006 ACM SIGMOD Int'l Conf. on Management of Data. New York: ACM Press, 2006. 407−418. [doi: 10.1145/1142473.1142520]

[22] Mannila H, Toivonen H, Verkamo AI. Discovery of frequent episodes in event sequences. Data Mining and Knowledge Discovery, 1997,1(3):259−289. [doi: 10.1023/A:1009748302351]

[23] http://www.cnki.net

朱辉生(1968－),男,江苏泰州人,博士,副教授,主要研究领域为数据挖掘,数据流分析.

汪卫(1970－),男,博士,教授,博士生导师,CCF 高级会员,主要研究领域为复杂结构数据管理,数据挖掘,数据安全.

施伯乐(1936－),男,博士,教授,博士生导师,主要研究领域为数据库,知识库.