doi:10.3969/j.issn.1000-436x.2013.01.002

社会网络中时空周期行为模式挖掘算法

胡玉鹏[1,2]，罗昊[1]，林亚平[1]，秦拯[1]，尹波[1]
(1. 湖南大学 信息科学与工程学院，湖南 长沙 410082；2. 国防科学技术大学 计算机学院，湖南 长沙 410073)

摘 要：提出了一种层次二部图行为模式分析模型以及相应的挖掘算法，可获取潜在的时空周期行为模式，同时能克服以往算法的子集漏选问题。在此基础上，所设计的地点获取算法可以获取近似最小地点控制子集，尽早对少量地点进行监控。实验表明算法能全面地抽取周期地点子集，获取近似的最小地点控制子集，挖掘出常用地点以覆盖大部分周期行为个体。

关键词：社会网络；层次二部图；时空周期行为模式；最小地点控制子集
中图分类号：TP393 **文献标识码**：A **文章编号**：1000-436X(2013)01-0008-11

Spatio-temporal periodic behavior mining algorithm for social networks

HU Yu-peng[1,2], LUO Hao[1], LIN Ya-ping[1], QIN Zheng[1], YIN Bo[1]
(1.College of Information Science and Engineering, Hunan University, Changsha 410082, China;
2.School of Computer, National University of Defense T nology, Changsha 410073, China)

Abstract: A hierarchical bipartite graph based model and a mining algorithm were presented to obtain the potential spatio-temporal periodic behavior, meanwhile avoided the subset omitting problem in previous schemes. Then the location analysis algorithm was designed to achieve the nearly inimum location dominating subset, it could monitor the small portion of the locations as early as possible. Finally experiments results show that the algorithms can find out the periodic location set as well as obtain nearly minimum location dominating subset, so as to cover the major portion of objectives using those popular locations.

Key words: social network; hierarchical bipartite graph; spatio-temporal periodic behavior pattern; minimum location dominating subset

1 引言

社会网络分析(social network analysis)作为研究社会网络动态结构及演化规律的热点研究领域，其潜在的广泛应用前景(如恐怖活动监控、金融危机预测、舆情分析和国际关系研究等)已经吸引了众多学者的极大关注[1,2]。目前，社会网络挖掘研究大多集中在各类复杂社会关系的量化表征、多维关联分析以及动态演化分析方面[3~11]：文献[3]研究基于主题的社会影响力分析模型，关注于如何量化不同主题下用户之间的影响力；文献[4]设计了一种社会网络可视化方法，用于监控分析恐怖组织活动；文献[5]依据同名的不同人物具有不同的社会网络思想，通过图谱分割来进行社会网络的自动聚类，从而实

收稿日期：2011-12-05；**修回日期**：2012-05-07
基金项目：国家自然科学基金资助项目(61272546, 61173038, 61070194)；湖南省自然科学基金资助项目(10JJ4042)；中国博士后基金资助项目（2012m512071）；湖南省科技计划重点基金资助项目（2011FJ2003）；中央高校基本科研业务费基金资助项目（2011-036）
Foundation Items: The National Natural Science Foundation of China (61272546, 61173038, 61070194); The Natural Science Foundation of Hunan Province(10JJ4042); The Postdoctoral Foundation of China (2012m512071); The Science and Technology Key Projects of Hunan Province (2011FJ2003); The Basic Foundation of Central College of China (2011-036)

现人名检索结果的重名消解。文献[6~8]通过抽取邮件习惯或电话通信行为，研究社会群体内部的特有组织结构关系；文献[9]主要研究了动态网络随机图的生成模型及演化算法；文献[10]扩展了一般的随机块建模方法，允许个体有多种类型，这对于个体在不同社团中的多重角色的结构研究奠定了一定的基础；文献[11]提出了时间相关的用户行为因子模型，可对动态的社会网络关系进行模拟。

尽管已有不少的社会网络研究成果，然而对于社会网络动态交互行为模式的研究尚且不多。随着通信技术的进步以及 Web 社交网络的迅猛发展，一些用户信息逐渐透明化，包括用户 GPS 位置、邮件习惯等数据。通过掌握这些社会网络动态交互过程中形成的个人数据流，可以有效监控或预测人们的行为轨迹，挖掘潜在的交互行为模式，进而可以有针对性地提供相关服务(如位置服务、动态推荐服务)，或者实施行为监控(如敏感事件监控)。近几年，虽然文献[12~14]通过对 Web 社会网络数据流进行分析，宏观上对社会网络各种行为进行分类统计，但是对更深层次的行为模式信息的研究尚未涉及。

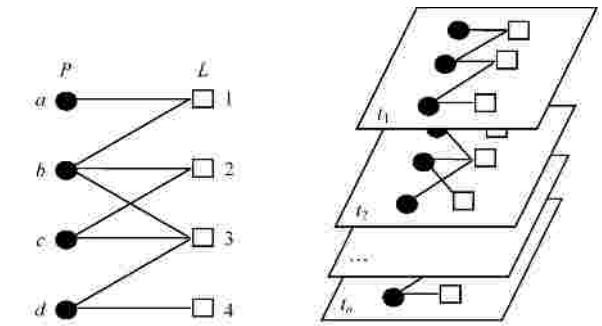
Lahiri 等率先提出周期子图挖掘问题[15,16]，用于研究动态社会网络中的周期行为模式挖掘问题，挖掘动态交互数据流中所包含着一些周期或类周期行为(如演员之间的周期合作、动物的周期迁徙等行为)。然而，Lahiri 等提出的算法(以下简称 LM 算法)主要从时间维度进行分析挖掘，并未涉及空间维度的研究，而在实际应用中，大部分社会网络行为数据往往在时间和空间维度上同时具有周期(如每逢春节在父母家团聚、球友之间在某运动中心的定期活动等)。

因此，本文基于行为数据在时间和空间维度上的双重关联性，提出了一种基于层次二部图(hierarchical bipartite graph)的时空周期行为模式挖掘算法(SPBMA, spatio-temporal periodic behavior mining algorithm)，该算法能够针对社会中人们的出行习惯、移动轨迹，对动态社会网络数据进行时间片切片，并将整个社会网络转化为一个模式二叉树，能在多项式时间复杂度内完成挖掘任务，克服 LM 算法的子集漏选问题，获取代表时空周期行为模式的闭合周期地点子集。任何一个闭合周期地点子集都能反映某个群体行为轨迹在时间和空间维度上的内在联系。然后，基于 SPBMA 得到的周期地点集合，设计最小地点控制子集获取算法，算法能得到覆盖全部或大多数个体日常行为的最小地点控制集，可应用到公共安全、恶性行为/事件监控领域。

2　层次二部图模型

目前已有的频繁子图挖掘算法大多基于 Apriori[17]和 FP-growth[18]方法，往往只考虑在时间维度挖掘，没有涉及到空间维度。本节建立一种层次二部图分析模型，可以同时体现行为数据的时间和空间属性，从而便于对时空周期行为模式进行挖掘。

首先基于二部图的一般特性，对社会网络数据进行划分。图 1(a)表示的是单一时间片上(某 Δt 时间段内)从事某一社会活动的时空行为分析模型，将活动对象或行为者和活动地点归纳为 2 类不同的节点子集 P 和 L，同类子集内部节点互不相连，从而构建子集 P 和 L 相互关联的二部图 $G(P$ 和 L 之间的链接表示关联属性)。由于将挖掘算法规约在二部图的二维空间中，故能保证具有较低的时间复杂度。进而，通过设定时间跨度 Δt(如按照常人的生活习惯设定以天为单位)，基于时间切片 $\{t_1,t_2,\cdots,t_n\}$ 对连续输入的社会网络数据流进行顺序切分，即每个 $t_i(i=1,2,\cdots,n)$ 的跨度为 Δt，从而得到如图 1(b)所示的分层行为模式挖掘模型。这样每个单一时间片都是一个二部图。



(a) 单一时间片上的二部图模型　　(b) 多时间片的层次二部图模型

图 1　基于层次二部图的行为模式分析模型

本文的首要任务就是挖掘这种层次二部图中潜在的周期行为模式，本文定义周期行为模式为：与特定行为者集合 P 关联的周期地点集合。因此，基于该层次二部图模型，只需要关注与某特定 P 集关联的地点集合 L，通过 SPBMA 获取时间片序列 $\{t_1,t_2,\cdots,t_n\}$ 中的周期地点子集即可。本文第二个任务是对所有的周期地点子集做进一步处理，获取最小地点控制子集，可应用到恶性行为/事件监控领

域；如图 1(a)中的地点子集 L^*={1-某超市,3-车站}为一个最小地点控制集，用最少的地点个数能覆盖整个行为群体 P，则对 L^*进行监控即可对整个行为群体 P 进行控制，这样通过获取整个时间序列 $\{t_1,t_2,\cdots,t_n\}$中的周期最小地点控制子集，可以尽早地对少量个体进行监控，防止传染性行为或恶性事件的扩散。

3 基本定义

基于上节所设计的层次二部图行为模式挖掘模型，本节对相关算法需要用到的基本概念进行定义。

定义 1 二部图动态网络。二部图动态网络 $G=<G_1,\cdots,G_T>$，$G_i=(E_1,\cdots,E_n)$ 表示所在时间片(time-step)为 t_i (i=1,2,$\cdots$,T，以小时或天为单位)，且边数为 n 的二部图。其中，E_j 是 G_i 中的边，$E_j=(P_j, L_j)$，j=1,2,$\cdots$,n，P_j 代表边所关联的行为者，L_j 代表关联的地点。

定义 2 频繁地点子集。对于任意地点子集 $\mathbb{L}$，其在 G 中的地点支持集 $S(\mathbb{L})$ 表示：所有包含地点集 $\mathbb{L}$ 的 G_i 对应的时间片 t_i 的集合，标记为 $S(\mathbb{L})=\{t_i|\mathbb{L}\subseteq G_i(L)\}$，$G_i$ (L)表示 G_i 包含的地点集合，i=1,2,$\cdots$,T；且令 $|S(\mathbb{L})|$ 表示 $\mathbb{L}$ 的支持度，即 $S(\mathbb{L})$ 的元素个数。设 $s(1\leq s\leq T)$表示最小值支持度，若满足 $|S(\mathbb{L})|\geq s$，则称 $\mathbb{L}$ 是 G 的频繁地点子集。

定义 3 周期地点子集。对于任意地点子集 $\mathbb{L}$，若 $S(\mathbb{L})\neq\varnothing$ 且对于任意 i，i=1,2,$\cdots$,T，满足 $t_{i+1}-t_i=r$，r 为正整数常量，则称 $S(\mathbb{L})$ 为 $\mathbb{L}$ 的一个周期支持集。如果 $\mathbb{L}$ 至少存在一个周期支持集满足 $|S(\mathbb{L})|\geq s$，σ 是最小支持度，则 $\mathbb{L}$ 是一个周期地点子集，常数 r 是 $\mathbb{L}$ 的周期，表示群体行为的周期。每个不同的周期地点子集可以有不同的起始时间，不同的支持度，不同的周期，从而拥有多重周期。

令 $S=\{\mathbb{L}_s\}$ 表示 G 中支持度为 σ 的所有周期地点子集 $\mathbb{L}$ 的集合，易知对于任何周期地点子集 $\mathbb{L}\in S$，其子集都是周期的并且属于 S。可见这种向下封闭的性质存在着较大的冗余，为了尽可能去除冗余且不丢失地点信息，下面定义最大周期地点子集和闭合周期地点子集。

定义 4 最大周期地点子集。若不存在周期地点子集 $L^*\in S$，使得 $\mathbb{L}\subset L^*$，则称周期地点子集 $\mathbb{L}$，$\mathbb{L}\in S$，是一个支持度为 s 的最大周期地点子集，即在该支持度上 $\mathbb{L}$ 无法再加入新的地点。且令 $S_{\max}$，$S_{\max}\subseteq S$，表示支持度为 s 的所有最大周期地点子集的集合。

定义 5 闭合周期地点子集。若周期地点子集 $\mathbb{L}\in S$ 的周期为 p，且当 $s'\geq s$ 时，$\mathbb{L}$ 是一个最大周期地点子集，则称 $\mathbb{L}$ 在周期为 r 时是闭合的。令 S_{closed}($S_{\text{closed}}\subseteq S$)，表示最小支持度为 s 的所有闭合周期地点子集的集合。

由于 $S_{\max}\subseteq S_{\text{closed}}\subseteq S$，且 $S_{\max}$ 和 S 都可以被 S_{closed} 覆盖。因此，只需要计算出所有支持度下的所有闭合周期地点子集就可以得到所有的周期地点子集。下面给出具体挖掘算法中需要用到的相关定义。

定义 6 最小地点控制子集。对于地点子集 $\mathbb{L}\subseteq L$，行为者集 P，满足 $\forall p\in P$，$\exists l\in\mathbb{L}$ 与之关联，则称 $\mathbb{L}$ 是 P 的地点控制子集。若 $\mathbb{L}_i$ (i=1,2,$\cdots$)均是 P 的一个地点控制集，令 $\mathbb{L}_{\min}(P)$ 表示 P 的最小地点控制集，则有 $\mathbb{L}_{\min}(P)=\arg\max_{\mathbb{L}_i}\{f(\mathbb{L}_i)=P(\mathbb{L}_i)/\sum_{l\in\mathbb{L}_i}\deg(l)\}$。其中，$P(\mathbb{L}_i)$ 表示与地点子集 $\mathbb{L}_i$ 关联的不同行为者的数量，$\deg(l)$表示地点 l 的度，即与地点集 l 关联的行为者的数量。

易知 $f(\mathbb{L}_i)\leq 1$，表示 $\mathbb{L}_i$ 的访问重叠因子，值越小则表示单个行为者平均访问的地点数越多，值越大表示单个行为者平均访问的地点数越少。如图 1(a)的最小地点控制子集 L^*={1,3} 的重叠因子为 $f(L^*)$=0.8，表示单个行为者平均访问的地点数比较少，只有行为者 b 同时访问了 1 和 3，其他行为者都只访问了一个地点。

定义 7 模式二叉树。模式二叉树 PB-Tree (pattern binary tree)中的每个节点表示为 $N=<\mathbb{L}_N$, $\{D_N\}$, N_l, $N_r>$，N 包含一个地点子集 $\mathbb{L}_N$ ($\mathbb{L}_N$ 可以是一个时间片的图 G_i 所包含的地点集的子集)、一个左节点 N_l、一个右节点 N_r 以及一个描述器集$\{D_N\}$。N_l 包含的地点子集 $\mathbb{L}_{N_l}$ 必须满足 $\mathbb{L}_{N_l}\subset\mathbb{L}_N$，$N_r$ 包含的地点子集 $\mathbb{L}_{N_r}$ 必须满足 $\mathbb{L}_{N_r}\not\subseteq\mathbb{L}_N$。每个节点对都附带一个描述器集合$\{D_N\}$，包含一系列描述器，用来产生闭合周期子集。

定义 8 描述器。描述器 $D_k=<S_k(\mathbb{L})$, $r>$，(k=1,2,$\cdots$)，其中，$D_k\in\{D_N\}$; $S_k(\mathbb{L})$ 表示当前节点 N 包含的地点子集 $\mathbb{L}$ 当前部分支持集，r 是周期。且令 $S_{k\text{-last}}$ 表示支持集 $S_k(\mathbb{L})$ 中的最后一个时间元素 t_i，令 NEXT(D_k)=$S_{k\text{-last}}+r$，表示 PB-Tree 更新过程中 D_k 能获取的下一个周期时间片值 t_{i+r}。

4　行为模式挖掘算法

本节对行为模式挖掘算法（SPBMA）和最小地点控制子集获取算法进行详细描述。总体的行为模式挖掘算法如图 2 所示，通过不断地读入时间片 G_i，执行 SPBMA 进行模式二叉树迭代更新，从而将动态网络 G 映射到一个模式二叉树 PB-Tree。SPBMA 通过不断读入时间片 G_i，在遍历 PB-Tree 过程中进行相关操作，将符合条件的时间片地点子集作为新的节点插入到 PB-Tree 中，或更新当前节点的描述器，最终挖掘出所有闭合周期子集的集合 S_{closed}。由于 SPBMA 采用二叉树先序遍历方法，不需要文献 LM 算法采用的散列 Map 方法来定位节点，克服了周期子集漏选问题。在此基础上，基于所有闭合周期子图，由 4.2 节中的最小地点控制子集获取算法，分析得到周期最小地点控制子集或近似子集。

总体挖掘算法(G)：
输入： G
输出： S_{closed}
1) 初始化，$S_{\text{closed}}=\{\}$; $N_{\text{root}}=\text{new node}(G_1(L))$
2) for each G_i in G
3)　SPBMA($N_{\text{root}}, G_i(L)$)
4) return S_{closed}

图 2　总体的行为模式挖掘算法

4.1　SPBMA

本节首先介绍 SPBMA。SPBMA 模式二叉树算法用于计算所有的闭合周期地点集，该算法包含 2 个子算法：节点插入算法 INSERT_NODE 和描述器更新算法 UPDATE_DESCRIPTORS，这 2 种算法将分别在 4.1.1 节和 4.1.2 节中进行阐述。

SPBMA 从读第一个时间片 G_1 开始，初始化 PB-Tree，且将包含的地点集合 $G_1(L)$作为 PB-Tree 的根节点 N_{root}，并创建其第一个描述器 D_1=<{t_1}, r=0>，t_1 是 G_1 的时间值。然后从下一个时间片 G_2 开始，对于读入的每个新时间片 G_i 调用模式二叉树算法 SPBMA (N,$G_i(L)$)，如算法 1 所示。算法 1 中，对于每个新的时间片 G_i，SPBMA 都采用先序遍历方式递归执行，具体过程如下：

1) 首先，从根节点 N_{root} 开始，比较当前节点 N 存储的地点子集 $\mathbb{L}_N$ 与对应 G_i 的地点集合 L 即可，若 $\mathbb{L}_N \subseteq G_i(L)$，根据定义 7 可知，只需更新当前节点 N 及其左子树的描述器集合即可，然后判断 N 的右子节点 N_r 是否为空，若不为空，则对其右子树调用 SPBMA (N_r,$G_i(L)$)；若 N_r 为空且 G_i 未使用过(注：当且仅当 PB-Tree 中不存在节点 N' 满足 $\mathbb{L}_{N'}=G_i(L)$，则称图 G_i 及地点集 $G_i(L)$未被使用)，则调用节点插入算法 INSERT_NODE(right,$G_i(L)$)，直到遍历结束。

2) 若 $G_i(L)\subset\mathbb{L}_N$，则先判断 N 的左子节点 N_l 是否为空，若 N_l 为空且 G_i 未使用过，则调用节点插入算法 INSERT_NODE(left,$G_i(L)$)；若 N_l 不为空则先对其左子树调用 SPBMA(N_l,$G_i(L)$)；然后判断 N 的右子节点 N_r 是否为空，若 N_r 为空且 G_i 未使用过，则调用节点插入算法 INSERT_NODE(right,$G_i(L)$)，N_r 不为空则对其右子树调用 SPBMA(N_r,$G_i(L)$)，直到遍历结束。

3) 若 $\mathbb{L}_N$ 和 $G_i(L)$ 不满足上述 2 个条件，则计算其公共子集 C, $C=\mathbb{L}_N \cap G_i(L)$，先判断 N 的左子节点 N_l 是否为空，若 N_l 为空，则调用节点插入算法 INSERT_NODE(left,C)；N_l 不为空则先对其左子树调用 SPBMA(N_l,MCS)，然后判断 N 的右子节点 N_r 是否为空，若 N_r 为空且 G_i 未使用过，则调用节点插入算法 INSERT_NODE(right,$G_i(L)$)，若 N_r 不为空则对其右子树调用 SPBMA(N_r,$G_i(L)$)，直到遍历结束。

图 3 给出了模式二叉树算法示例。其中，inherit, remove, output 表示对节点描述器的操作。整体上 SPBMA 采用先序遍历，通过对 PB-Tree 进行操作，然后计算周期地点集。图 3(a)中 t_1~t_5 是 5 个时间片，G_1~G_5 分别是这 5 个时间片对应的行为轨迹二部图，最小支持度为 s=3。与文献[15,16]提出的算法不同的是，SPBMA 只需要关注 L 中的某些地点，对一系列地点序列进行分析，不需要关注行为者集合的变化。图 3(a)表示初始化的地点数据，其中，t_4 的地点集为空，表示没有行为者在该时间段进行相关活动，t_1、t_2、t_3、t_5 这 4 个状态分别对应图 3(b)~图 3(e)，这 4 个状态均表示已更新的当前时间片状态。PB-Tree 对于每个新的时间片都更新一次，首先以根节点 N_{root} 为目标节点调用更新算法 SPBMA (N_{root}, $G_i(L)$)。

以图 3 中更新时间片 t_3 为例，如图 3(d)所示，以图 3(c)为基础，读入地点集 $G_3(L)$，首先以根节点 N_{root} 作为目标节点调用更新算法 SPBMA(N_{root}, $G_3(L)$)，$\mathbb{L}_{N_{\text{root}}}$={1,2,3,4,5}，$G_3(L)$={1,4,5,6}，$C=\mathbb{L}_{N_{\text{root}}}\cap G_3(L)$={1, 4, 5}，由于根节点的左子树不为空，则进入递

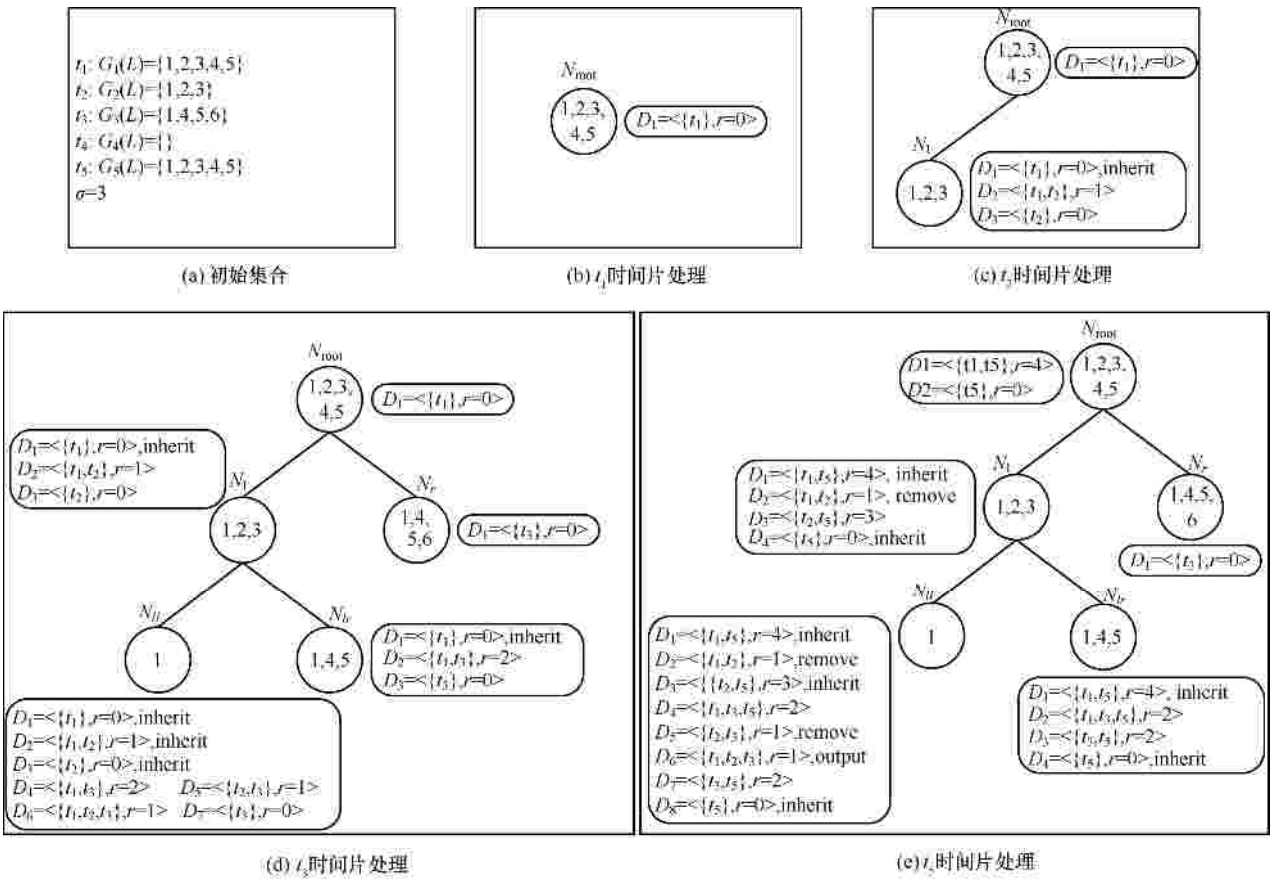


图 3 模式二叉树 SPBMA 算法示例

归指令：以根节点的左子节点 N_l 作为目标节点调用更新算法 SPBMA(N_l,C)，**其中，** $\mathbb{L}_{N_l}=\{1,2,3\}$，$C$={1,4,5}；**而** $C'=\mathbb{L}_{N_l} \cap M=\{1\}$，**因目标节点 N_l 没有左子节点和右子节点，故依次调用节点插入算法** INSERT_NODE(left, C')、INSERT_ NODE(right, C)(C **未使用**)，**则** SPBMA(N_l, C)**执行结束，跳出递归指令；这时目标节点变回根节点 N_{root}，其右子节点 N_r 为空，故调用节点插入算法** INSERT_NODE (right,$G_3(L)$)，**执行** SPBMA (N_{root},$G_3(L)$)**结束。类似地，对于时间片 t_5，则只需要以根节点 N_{root} 作为目标节点调用** SPBMA **以及描述器更新算法** UPDATE_ DESCRIPTORS (N_{root},$G_5(L)$)，**当满足支持度时，则说明节点为一个闭合周期地点子集，需要添加到 S_{closed}，并输出**(output)**相应的描述器，详细的描述器更新过程将在** 4.1.2 **节中阐述。**

本文的 SPBMA **与** LM **算法不同的是本文采用二叉模式树数据结构，所以不需要** LM **算法中的散列** Map **来保存节点路径以便于查找，从而可以避免** LM **算法出现子集漏选的问题。例如，** $\mathbb{L}_{N_{\text{root}}}=\{1,2,3,4,5\}$ **是根节点 N_{root} 地点集，新读入地点集为** $G_2(L)$={1,2,3}，**与根节点交集为** $C=\mathbb{L}_{N_{\text{root}}} \cap G_2(L)=\{1,2,3\}$，**且树中不存在该节点，即散列** Map **查找不到，则需要将** $G_2(L)$={1,2,3}**作为根节点的一个子节点 N_l 插入，** $\mathbb{L}_{N_l}=\{1,2,3\}$。**而读入下一时间片的地点集为** $G_3(L)$={1,4,5,6}，**同理作为根节点的另一个子节点 N_r 插入。**LM **算法到此处将执行完毕。但是** $\mathbb{L}_{N_l}$ **和** $\mathbb{L}_{N_r}$ **还有公共子集** $\mathbb{L}_{N_l} \cap \mathbb{L}_{N_r}=\{1\}$，**则被** LM **算法所遗漏，而本文的** SPBMA **则可以避免此问题。**

算法 1 SPBMA(N,$G_i(L)$)

输入：节点 N、地点子集

1) if $\mathbb{L}_N \subseteq G_i(L)$
2) {
3) UPDATE_DESCRIPTORS(N,$G_i(L)$);
4) if N_r **不为空**
5) SPBMA(N_r, $G_i(L)$);
6) else if N_r **为空 且** G_i **未使用**
7) INSERT_NODE(right,$G_i(L)$);
8) }

```
9) else if  G_i(L) ⊂ 𝕃_N
10) {
11)     if N_l 不为空
12)         SPBMA(N_l,G_i(L));
13)     if N_r 不为空
14)         SPBMA (N_r,G_i(L));
15)     else if N_r 为空 且 G_i 未使用
16)         INSERT_NODE(right,G_i(L));
17) }
18) else
19) {
20)     C ← 𝕃_N ∩ G_i(L) ; //C 表示公共子集
21)     if C 为空
22)         if N_r 不为空
23)             SPBMA (N_r,G_i(L));
24)         else return;
25)     else
26)     {
27)         if N_l 不为空
28)             SPBMA (N_l,C);
29)         else INSERT_NODE(left,C)
30)         if N_r 不为空
31)             SPBMA (N_r,C);
32)         else if N_r 为空 且 G_i 未使用
33)             INSERT_NODE(right,G_i(L));
34)     }
35) }
```

4.1.1　节点插入算法

SPBMA 包含的节点插入子算法用于对 PB-Tree 插入节点，算法涉及如何避免 PB-Tree 中产生冗余节点以及描述器的继承问题。

具体的插入算法的伪代码如算法 2 所示。由定义 7 可知，PB-Tree 中任意节点 N 的左子树中所有节点包含的子集均是 $\mathbb{L}_N$ 的子集。为了避免 PB-Tree 中出现重复节点，降低冗余度，算法规定只有在当前节点 N 的左子节点 N_l 或右子节点 N_r 为空，且要插入的子集 L^* 未被使用过时，才能将 L^* 作为新的地点集插入。算法插入左子节点和插入右子节点的区别在于：左子节点需要继承当前父节点 N 的所有描述器集合$\{D_N\}$，而右子节点只需要继承$\{D_N\}$中从 N 的父节点继承过来的那一部分描述器。因为 $\mathbb{L}_{N_l} \subset \mathbb{L}_N$，所以 $\mathbb{L}_{N_r}$ 必定在 $\mathbb{L}_N$ 出现过的所有时间片中出现，故需要全部继承$\{D_N\}$，包含全部相关的时间片；而 $\mathbb{L}_{N_r} \not\subseteq \mathbb{L}_N$，同时$\{D_N\}$中有继承过来的部分描述器，对于 N 的父节点包含的地点集 L^* 中出现过的时间片，$\mathbb{L}_{N_r}$ 与 $\mathbb{L}_N$ 也必定都出现，故 N_r 只需继承 N 的描述器集合中也是继承过来的那一部分描述器信息。

每个新插入的节点 N_{new} 在继承了对应的描述器后，还需要对描述器集$\{D_{\text{new}}\}$执行一次更新算法 UPDATE_DESCRIPTORS(N_{new},$G_i(L)$)，具体流程如 4.1.2 节的算法 3 所示。

算法 2　INSERT_NODE(*type*,$G_i(L)$)

输入：节点类型 *type*(left 或 right)、地点子集

```
1) if G_i(L)已被使用
2)     return;
3)  N_new=new Node(); 𝕃_N_new ← G_i(L) ;N_r=NULL;
N_l=NULL;
4) if type== “ left ”
5) {
6)     N_l←N_new;
7)     {D_l}继承全部{D_N};
8)     UPDATE_DESCRIPTORS(N_l,G_i(L));
9) }
10) else
11) {
12) N_r←N_new;
13) {D_r}继承{D_N}中从 N 的父节点继承得到的所有描述器;
14) UPDATE_DESCRIPTORS(N_r,G_i(L));
15) }
```

以图 3 中更新时间片 t_3(图 3(d))为例，需要先后插入 3 个新节点，即 INSERT_NODE(left,C')，INSERT_NODE(right, C)，INSERT_NODE(right, $G_3(L)$) (C'、C 和 $G_3(L)$均未使用)。由于 C'是作为当前节点(图 3(d)中的 N_l)的左子节点插入的，故需要全部继承 N_l 的描述器集合$\{D_l\}$=$\{D_1$=<$\{t_1\}$, r=0>, inherit;D_2=<$\{t_1,t_2\}$,r=1>;D_3=<$\{t_2\}$,r=0>$\}$，然后更新并添加新的描述器(具体见描述器更新算法 3)，故得到新的节点 N_{ll}。而 C 和 $G_3(L)$则是作为右子节点插入的，所以只需要分别继承其父节点的 N_l 描述器集中也是继承得到的部分描述器(即有 inherit 标识的描述器)，并更新描述器集即可。其中，C 是作为当前节点 N_l 的右子节点插入的，故只需要继承中$\{D_l\}$中是继承过来的部分描述器，即$\{D_1$=<$\{t_1\}$,r=0>,

inherit;}并更新描述器集即可得到新的节点 N_{lr}。而 $G_3(L)$是作为根节点的 N_{root} 右子节点插入的，而 N_{root} 的描述器集中没有继承得到的描述器，所以直接更新自身的描述器集即可得到 N_r。

类似地，对于更新时间片 t_5(见图 3(e))，更新算法进行到以节点 N_r、$\mathbb{L}_{N_r}=\{1,4,5,6\}$ 为目标节点时，$G_5(L)=\{1,2,3,4,5\}$，则公共子集还是 $C=\mathbb{L}_{N_r} \cap G_5(L)=\{1,4,5\}$，但树中已存在节点 N_{lr}，其地点集也为 $\{1,4,5\}$，故 C 已使用，没有新节点插入。

4.1.2 描述器集合更新算法

描述器集合更新算法是用来获取闭合周期子集的，具体算法伪代码如算法 3 所示。对于节点任意节点 N 的描述器集$\{D_N\}$，$S_k(\mathbb{L})$ 表示$\{D_N\}$中描述器 D_k (k>0)的支持集。每个新描述器 D_k 初始化的支持度都为 1(即支持集中只有 1 个时间元素)，周期都为 0。

如算法 3 所示，描述器更新算法的具体流程如下。

1) 首先，在更新当前节点时需要与父节点的描述器同步(即父节点描述器集有改动时，当前节点也要做相应改动)。

2) 当用新支持时间片 t_i 对描述器集$\{D_N\}$更新时，首先对于所有满足 r=0 并且 $S_{k\text{-last}}\neq t_i$ 的描述器，若判断该描述器是继承得到的，则产生新的描述器 D_{new}=<$\{t,t_i\}$,$r=t_i-t$>(t 是描述器中已存在的时间元素)，并添加到描述器集$\{D_N\}=\{D_N\}\cup D_{\text{new}}$(如算法 3 的 4)~13))。若该描述器是非继承得到的，则将 t_i 作为最后一个元素插入描述器的支持集并且令 $r=t_i-S_{k\text{-last}}$。对于其他 $r\neq 0$ 的描述器，则先将 t_i 与$\{D_N\}$中所有不等于 t_i 的支持时间 t 分别产生新的描述器 D_{new}，则将 D_{new} 合并到$\{D_N\}$(如算法 3 的 14)~20)，若描述器集$\{D_N\}$中已存在描述器 $D_k=D_{\text{new}}$，则合并时集合$\{D_N\}$不发生变化); 然后对所有描述器更新一次，更新过程如下：若 D_k 满足 t_i=NEXT(D_k)，则将 t_i 插入到支持集 D_k 的最后；若 D_k 满足 NEXT(D_k)<t_i，同时支持集合元素个数$|S_k(\mathbb{L})|\geq s$，则表明相应的节点为一个闭合周期地点子集，并添加到 S_{closed}，同时输出(output) D_k；若 D_k 满足 NEXT(D_k)<t_i，同时支持集合元素个数$|S_k(\mathbb{L})|<s$，则移除(remove)该描述器(如算法 3 的 21)~26))。

3) 特别地，当描述器集$\{D_N\}$更新完后，$\{D_N\}$中需要添加新的描述器 D_{new}=<$\{t_i\}$,r=0>，若已存在相同描述器则不添加。

4) 最后，若当前节点 N 的描述器集已更新，则需要对其左子树中所有节点进行更新。因为 N 的左子树中所有节点包含的地点子集都是 $\mathbb{L}_N$ 的子集(见定义 7)。同理，若 N 中有描述器 D_k 被移除，则 N 的左子树中所有继承了描述器 D_k 的节点都要移除描述器 D_k。

算法 3 UPDATE_DESCRIPTORS(N,$G_i(L)$)

输入：节点 N、$G_i(L)$;

输出：描述器 D_k

```
1) 与节点 N 的父节点描述器集继承同步(继承规则见节点插入算法)
2) for each D_k in {D_N}
3) {
4)    if r=0 且 t≠t_i// t 表示 S_k(L) 中支持的时间值
5)       if D_k 是继承得到的
6)          {
7)            D_new=<{t,t_i},r=t_i- t>;//建新描述器
8)            {D_N} ={D_N} ∪ D_new; //添加新描述器
9)          }
10)       else   {
11)             ADD_TO_LAST( S_k(L) ,t_i);
12)             r=t_i- S_k-last;
13)         }
14)    else if r≠0
15)   {
16)       for each t≠t_i in  S_k(L)
17)          {
18)          D_new=<{t,t_i},r=t_i- t>；//建新描述器
19)          {D_N} ={D_N} ∪ D_new ;//添加新描述器
20)          }//End of for
21)      if t_i=NEXT(D_k)//t_i是G_i(L)对应的时间片
22)          ADD_TO_LAST( S_k(L) ,t_i);//插入到支持集 S_k(L) 的最后
23)      else if (NEXT(D_k)<t_i)
24)         if  |S_k(L)|≥s
25)              S_closed ← S_closed ∪{L_N } ；OUTPUT (D_k);}//S_closed 为挖掘算法的最终输出
26)         else REMOVE(D_k);
27)   }
28) }//End of for
```

29) D_{new}=<{t_i},r=0>;//建新描述器

30) {D_N} ={D_N} ∪ D_{new}; //添加新描述器

31) if N_l 不为空

32) if N_{lr} 不为空

33) UPDATE_DESCRIPTORS(N_{lr},G_i(L));

34) else UPDATE_DESCRIPTORS(N_l,G_i(L))

以图 3 中更新时间片 t_5 为例进行算法说明。G_5(L)={1,2,3,4,5}，首先以根节点 N_{root} 为目标节点，其描述器集{D_1=<{t_1},r=0>}直接更新后变为{D_1=<{t_1,t_5},r=4>}，然后新增描述器 D_2=<{t_5},r=0>。由于 N_{root} 的描述器集已更新，所以需要对 N_{root} 左子树中所有节点进行描述器更新。对于节点 N_l，其地点集为{1,2,3}，为了先与父节点 N_{root} 描述器集合继承同步，需要更新已继承的描述器 D_1=<{t_1,t_5},r=4>,inherit，并继承新描述器 D_4=<{t_5},r=0>,inherit。然后更新 r=0 的描述器 D_3=<{t_2},r=0>得到 D_3=<{t_2,t_5},r=3>；且执行代码 16)~20)时，对于 $r\neq 0$ 的描述器 D_1=<{t_1,t_5},r=4>, inherit 和 D_2=<{t_1,t_2},r=1>中包含的不等于 t_5 的时间片值 t_1 和 t_2 将分别产生新的描述器 D_{new1}=<{t_1,t_5}, r=4>和 D_{new2}=<{t_2,t_5},r=3>，由于描述器集中已有相同的描述器，故合并时不发生变化。且对于描述器 D_2=<{t_1,t_2},r=1>，由于 NEXT(D_2)= t_3<t_5，且其支持度为 2<s，故需要移除 D_2，且其左子树中所有继承了该描述器的节点均需要移除(remove)该描述器。进而执行 29)~30)，添加新描述器 D_{new3}=<{t_5}, r=0>，由于已经存在相同描述器，故合并时描述器集合不发生变化。最后得到的 N_l 描述器集合如图 3(e)所示。

最后，对于子节点 N_{ll}，地点集为{1}，其描述器 D_6=<{t_1,t_2,t_3},r=1>，满足 NEXT(D_6)<t_5，但其支持度为 3=s，故表明该节点为一个闭合周期地点子集，并输出其相应的 D_6，合并{1}到 S_{closed}，N_{ll} 的其他操作与节点 N_l 类似。

4.2　周期最小地点控制子集近似算法

计算最小控制子集通常是一个 NP 难问题[19]，为了兼顾整个算法的执行效率，且达到近似完全控制，本文设计了基于贪心算法的最小地点控制子集获取算法(MLDSA, minimum location dominating set algorithm)。根据定义 6，该算法可通过获取二部图中度最大的前 ß 个地点集合来获取近似周期最小地点控制子集。MLDSA 基于 SPBMA 得到的闭合周期地点子集的集合 S_{closed} 进行分析计算，基本的算法思路为：MLDSA 先对全部周期地点集包含的地点按二部图中的度排序，即把地点 l 在各个时间片 G_i 中的度求和并进行排序，去掉重复出现的地点，得到 $S^*_{closed}=\{l\}$。然后，把度最大的第一个地点 l^* 加入到周期最小地点控制子集 $\wp_{min}(P)$。进而在剩下的周期地点集中把使得 $(\sum_{l\in\wp_{min}(P)}\deg(l))+\deg(l')$ 最大的地点 l' 加入到 $\wp_{min}(P)$ 中，以此类推，直到 $\wp_{min}(P)$ 中元素个数达到 ß 或者覆盖个体集 P。需要说明，S_{closed} 中包含的是不同 r 的所有闭合周期子集，实际中，需要根据不同的 r 值来挖掘频繁地点子集，这样可以获取覆盖某个具体周期行为模式(即由 r 所确定)的地点集。

5　算法分析与实验评估

5.1　算法复杂度分析

若 m 表示地点数据集 L 的大小，由于 SPBMA 最终输出的是 S_{closed} 闭合周期地点集，在二叉树中，最坏情况下，节点的数量最多只有 2^m- 1 个；然而，要出现最坏情况必须是满模式二叉树，即所有节点都和右父节点有交集且互不包含，因此，由于活动地点的扩散性和随机性，通常难以出现满二叉树的情况，尤其是个体数量比较大时，所有时间片均包含活动地点子集的可能性极小。如图 3 中只有 5 个节点，远小于 2^5- 1，因此，本文令二叉树中的平均节点数为 M，M<<(2^m- 1)，用来衡量实际的二叉树规模。此外，根据文献[15, 16]，证明时间片数为 T 的动态网络中闭合周期子集数最多为 $?(T^2\ln(T/s))$。所以，模式二叉树中节点数量为 $\max\{M, ?(T^2\ln(T/s))\}$，受到地点周期子集数和时间片数的双重约束。对每个时间片，SPBMA 都要对树中所有节点遍历一次，并且求一次最大公共子集(时间复杂度为 $O(E)$[20]，E 为边数，SPBMA 针对的数据集中没有孤立节点)，所以 SPBMA 的时间复杂度为 $O(ET(\max\{M, ?(T^2\ln(T/s))\}))$。因此，SPBMA 对于周期子集数少并且时间片多的数据集，即 M<$?(T^2\ln(T/s))\}$时，其时间复杂度为 $?(ET^3\ln(T/s))\}$，则与 LM[15]算法复杂度一致。反之，算法对于周期子集数非常多且时间片较少的数据集，即 $M\geq ?(T^2\ln(T/s))\}$时，其时间复杂度为 $O(ETM)$。

对于 SPBMA 的空间复杂度，由于整个模式二叉树都是维持在内存中间，所以模式二叉树的大小即为算法所需要的空间复杂度。假设模式二叉树中每个节点所需要的内存空间为 $O(1)$，则 SPBMA 所需要的

空间复杂度为 $O(\max\{M, ?(T^2\ln(T/s))\})$。

5.2 实验评估

本节设计实验，基于多个数据集对算法进行总体评估。本文采用微软亚洲研究院的 GPS 位置数据集(MS GPS)[21,22]和学生运动轨迹数据集(Students)进行测试。其中，MS GPS 数据集是微软亚洲研究院统计的 GPS 数据集经过相关处理后得到的，包括 2008 年至 2010 年间 165 个人的 GPS 数据(本实验从中抽取 250 天的连续数据)。而 Students 数据集则是对湖南大学多名研究生和本科生配备了 i-gotU 微型轨迹记录仪，记录其 250 天的移动数据。测试时默认的时间片间隔为 1 天(即 T=250)，并取每 10min 记录的每一条记录作为一个地点记录(MS GPS 约抽取 594×10^4 条记录，Stduents 约抽取 216×10^4 条记录)，即研究对象一天可以去多个地点，最小支持度 s=3，并且为了对行为模式进行适当的模糊化，设定当经度(或纬度)的偏差不小于 0.005(约 300m)时，才判定个体的地点已改变(该参数可用来调节行为模式挖掘的精确度)。测试前还需要对数据集进行预处理，只保留研究个体编号(记录仪编号)、GPS 经纬度、时间等信息。

表 1 显示了算法的平均运算时间、内存及子集个数对比情况。在 3.0GHz 的 PC 机上运行该实验，实验采用相同的数据集，分别用 2 种算法进行子集抽取，实验运行 5 次取平均值作为最后实验结果。在测试 LM 算法时，需要先用本文的层次二部图模型对数据集进行预处理，将地点集和行为个体分割，然后用修改后的 LM 算法对地点集进行挖掘。表 1 中可以发现 MS GPS 数据集消耗的时间和内存都比较大，这是因为该数据集蕴含的地点集比 Students 的要大，这表明个体的活动范围比较广，其中，MS GPS 数据集大约有 250 多个地点，Students 只有 175 个。由表 1 可知，本文提出的 SPBMA 在 2 个数据集上分别求出的闭合周期子集都要比 LM 算法多出近 15%，这也是导致 SPBMA 运算和内存开销上比 LM 算法高的原因之一。虽然 LM 算法采用散列 Map 来快速定位节点，但是仍然要对已定位的节点进行描述器更新操作；而 SPBMA 更关注于时空周期行为模式挖掘，虽然由于进行了遍历操作而导致算法性能上不如 LM 算法，但是可以避免子集漏选问题。

特别需要说明的是，由于没有采用快速散列 Map 以及需要处理更多的地点集，导致 SPBMA 比 LM 算法的执行时间长，即 SPBMA 约 1.5s 处理 1×10^4 条记录，但是 SPBMA 与 LM 算法的时间复杂度是相同的。因为分析得到的核心二叉树节点数量远小于(T^2=62 500)，如 MS GPS 数据集只得到约 500 个节点的模式二叉树，远小于 62 500，因此，按照 5.1 节中的分析，其时间复杂度与 LM 相当。

图 4 给出了 s=3 时闭合周期子集数随时间片 T 的变化曲线。文献[15,16]中已证明时间片数为 T 的动态网络中闭合周期子集数最多为 $?(T^2\ln(T/s))$，但是由于各种行为具有不同的周期，图 4 中实际获取的子集个数比最坏的理论值要低，如 MS GPS 数据集实际最大的闭合周期子集个数约为 111 348，Students 数据集最多约为 21 356 个闭合周期子集。同时，随着时间的延长，获取的子集个数越多。闭合周期子集数量大的主要原因是存在大量重复的地点子集，因为同一个地点可能在不同的 r、不同的开始时间 t 情况下，均为周期地点而被多次输出。为了将 2 个数据集统一，图 4 中最多分析了 250 天的数据。由于 Students 数据集包含的地点集数量比 MS GPS 要小许多，所以获取的 Students 数据集的闭合周期子集数要比 MS GPS 数据集的闭合周期子集数少。

图 5 显示了地点控制子集对周期行为个体的覆盖情况。图 5 显示了当 s 取不同值，而 r=3 时(大部分周期行为都是间隔周期较短的，周期越大，周期地点子集数越小；表 1 是所有 r 值的子集总数)，SPBMA 和 LM 算法获取不同周期的闭合周期子集数量(注：仍然包含一些重复地点子集)。固定 r 值是为了对进行某类周期行为的个体群进行研究，从而对其进行监控。当行为周期 s 变化时，能获取的周期子集数量迅速下降，这表明通过增大 s，可以

表 1 SPBMA 与 LM 算法的性能比较

数据集	SPBMA			LM 算法		
	运行时间/s	占用内存/MB	闭合周期子集数/个	运行时间/s	占用内存/MB	闭合周期子集数/个
MS GPS	725	167	111 348	619	114	81 780
Students	227	102	21 356	124	68	18 078

过滤掉表 1 中获取的大量伪周期行为，即真正的周期行为应该是持续相当一段时间以上的，其中，s=5 时，闭合周期子集数量只有 500 左右。同时可见，随着 s 增大时，LM 算法由于存在漏选问题，其获取子集数比 SPBMA 低约 15%~19%。

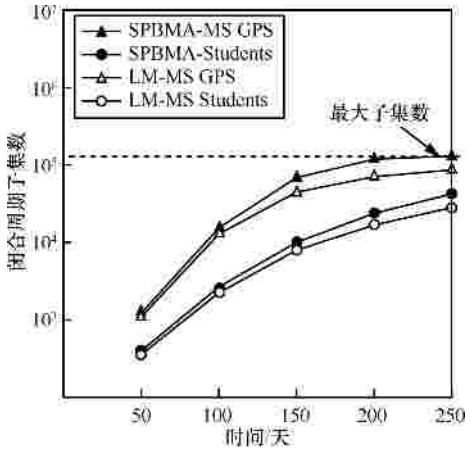


图 4　闭合周期子集数随时间变化

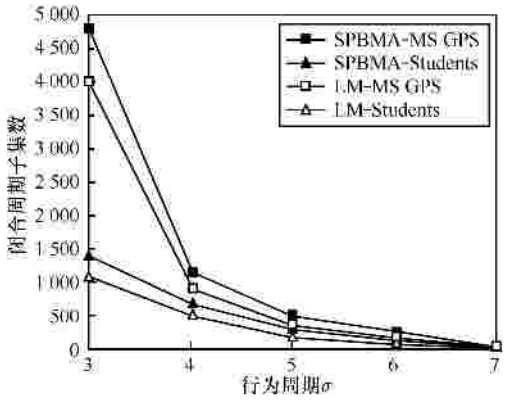


图 5　闭合周期子集数随 s 的变化(r=3)

图 6 给出了不同地点集门限值时的个体覆盖率情况。设定 r=3，s=5 时，实验采用 4.2 节中的最小地点控制子集获取算法获取 ß 个地点集。当 ß 增大时，可以发现对个体的覆盖率变高；且覆盖率主要在前 40~50 个地点变化较快，后面变化较慢，这表明大部分个体访问的是少量的常用地点，可通过获取少量的常用地点，即可近似覆盖大部分个体，防止恶性事件的扩散。然而，由于 LM 算法漏选了一些关键地点子集，导致一些节点度较大的地点没有入选，如 Students 数据集中漏选了“岳麓书院”和“南郊公园”等关键地点，所以其最高覆盖率只有约 0.75，比 SPBMA 要低；而 SPBMA 个体覆盖率最高接近于 0.99，且主要集中在前 50 个常用地点。同时，由于 Students 数据集的地点数没有 MS GPS 多，个体活动范围相对较小，故相同 ß 值时其覆盖率相对较高。

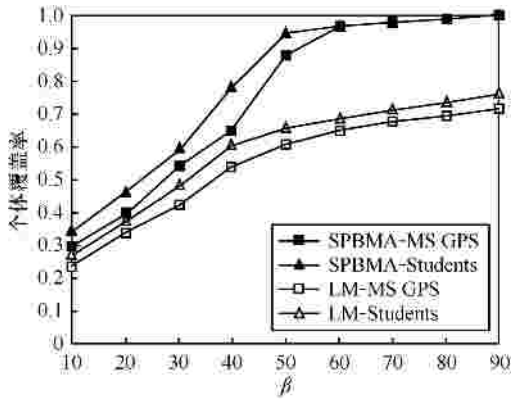


图 6　个体覆盖率随 ß 变化(r=3, s=5)

6　结束语

本文提出了时空周期行为轨迹挖掘问题，提出一种层次二部图行为模式分析模型和 SPBMA，获取潜在的周期行为模式，同时克服了以往方案的子集漏选问题。在此基础上，设计 MLDSA，该方法可以获取最小地点控制子集，能尽早地对少量地点进行监控。实验表明算法能获取近似的最小地点控制子集，挖掘出常用地点以覆盖大部分个体行为。下一步的工作为如何降低算法实现的时间和空间开销（如利用快速匹配和搜索思想），在算法实现上做进一步的优化。

参考文献：

[1] BILL H. Analyzing online social networks[J]. Communications of the ACM, 2008, 51(11):14-16.

[2] JOHN B, STEFAN D. The future of social networks on the internet: the need for semantics[J]. IEEE Internet Computing, 2007, 11(6): 86-90.

[3] TANG J, SUN J M, WANG C. Social influence analysis in large-scale networks[A]. Proceedings of the 15th ACM SIGKDD'09[C]. New York, USA, 2009. 807-816.

[4] 杨育彬, 李宁, 张瑶. 基于社会网络可视化分析的数据挖掘[J]. 软件学报, 2008, 19(8):1980-1994.
YANG Y B, LI N, ZHANG Y. Networked data mining based on social network visualizations[J]. Journal of Software, 2008,19(8):1980-1994.

[5] 郎君, 秦兵, 宋巍等. 基于社会网络的人名检索结果重名消解[J]. 计算机学报, 2009, 32(7):1365-1374.
LANG J, QIN B, SONG W, *et al*. Person name disambiguation of searching results using social network[J]. Chinese Journal of Computers, 2009, 32(7):1365-1374.

[6] CHAPANOND A, KRISHNAMOORTHYMS, YENER B. Graph theoretic and spectral analysis of Enron email data[J]. Comput Math

Organ Theory, 2005, 11(3):265-281.

[7] DIESNER J, CARLEY K M. Exploration of communication networks from the enron email corpus[A]. Proceedings of the 2005 SIAM Workshop On Link Analysis, Counterterrorism and Security[C]. California, USA, 2005. 3-14.

[8] NANAVATI A, GURUMURTHY S, DAS G, *et al*. On the structural properties of massive telecom call graphs: findings and implications[A]. Proceedings of the 15th ACM International Conference on Information and Knowledge Management[C]. New York, USA, 2006. 435-444.

[9] BORGWARDT K M, KRIEGEL H P, WACKERSREUTHER P. Pattern mining in frequent dynamic subgraphs[A]. Proceedings of the 6th IEEE Intl Conf on Data Mining[C]. Hong Kong, China, 2006. 818-822.

[10] WOLFE A P, JENSEN D. Playing multiple roles: discoveri g overlapping roles in social networks[A]. ICML-04 Workshop on Statistical Relational Learning and Its Connections to Other Fields[C]. Banff, Canada, 2004. 20-29.

[11] TAN C H, TANG J, SUN J M, *et al*. Social action tracking via noise tolerant time-varying factor graphs[A]. Proceedings of the Sixteenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (SIGKDD'2010)[C]. New York, USA, 2010. 1049-1058.

[12] FABRÍCIO B, TIAGO R, MEEYOUNG C, *et al*. Characterizing user behavior in online social networks[A]. Proceedings of the 9th ACM SIGCOMM Conference on Internet Measurement Conference[C]. New York, USA, 2009. 49-62.

[13] PARAG S, MATTHEW R. Yes, there is a correlation: from ial networks to personal behavior on the Web[A]. Proceedings of the 17th International Conference on World Wide Web[C]. New York, USA, 2008. 655-664.

[14] MARCELO M, JUSSARA A, VIRGÍLIO A. Identifying user behavior in online social networks[A]. Proceedings of the 1st Workshop on Social Network Systems[C]. New York, USA, 2008. 1-6.

[15] LAHIRI M, BERGER-WOLF T Y. Mining periodic behavior in dynamic social networks[A]. Proceedings of the 8th IEEE International Conference on Data Mining (ICDM 2008)[C]. Pisa, Italy, 2008. 373-382.

[16] LAHIRI M, BERGER-WOLF T Y. Periodic subgraph mining in dynamic networks[J]. Journal of Knowledge and Information Systems, 2010, 24(3):467-497.

[17] RAKESH A, RAMAKRISHNAN S. Fast algorithms for mining association rules in large databases[A]. Proceedings of the 20th International Conference on Very Large Data Bases[C]. VLDB, Santiago, Chile, 1994. 487-499.

[18] HAN J W, PEI J, YIN Y W. Mining frequent patterns without candidate generation[A]. ACM SIGMOD[C]. Dalls, USA, 2000.1-12.

[19] FABRIZIO G. A note on the complexity of minimum dominating set[J]. Journal of Discrete Algorithms, 2006, 4(2):209-214.

[20] DICKINSON P J, BUNKE H, DADEJ A, *et al*. On graphs with unique node labels[A]. IAPR[C]. Fisciano, Italy, 2003.13-23.

[21] ZHENG Y, LI Q N, CHEN Y K, *et al*. Understanding mobility based on GPS data[A]. Proceedings of ACM Conference on Ubiquitous Computing (UbiComp 2008)[C]. Seoul, Korea, 2008.312-321.

[22] ZHENG Y, ZHANG L Z, XIE X, *et al*. Mining interesting locations and travel sequences from GPS trajectories[A]. Proceedings of International Conference on World Wild Web (WWW 2009)[C]. Madrid, Spain, 2009. 791-800.

作者简介：

胡玉鹏（1981-），男，湖南祁东人，博士，湖南大学副教授，国防科学技术大学在站博士后，主要研究方向为网络存储系统、社会网络与无线网络。

罗昊（1987-），男，湖南株洲人，湖南大学硕士生，主要研究方向为数据挖掘。

林亚平（1955-），男，湖南邵阳人，博士，湖南大学教授、博士生导师，主要研究方向为计算机网络与机器学习。

秦拯（1969-），男，湖南祁东人，博士，湖南大学教授、博士生导师，主要研究方向为网络安全、云计算和软件工程等。

尹波（1983-），女，湖南株洲人，湖南大学博士生，主要研究方向为数据挖掘、传感器网络等。