软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software,2013,24(8):1816−1835 [doi: 10.3724/SP.J.1001.2013.04340]

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

轨迹数据库中热门区域的发现*

刘奎恩[1], 肖俊超[1], 丁治明[1], 李明树[1,2]

[1](中国科学院 软件研究所 基础软件国家工程研究中心,北京 100190)

[2](计算机科学国家重点实验室(中国科学院 软件研究所),北京 100190)

通讯作者: 刘奎恩, E-mail: kuien@iscas.ac.cn

摘 要: 发现被移动对象频繁造访的热门区域是从轨迹数据库中挖掘运动模式的重要前提,而合理约束热门区域的大小是提高轨迹模式的精确表达能力的关键.研究如何从轨迹数据库找出热门区域及如何限制其大小.定义了带有覆盖范围约束的热门区域,并采用过滤-精炼策略发现热门区域.在过滤阶段,设计了一种基于网格的密集区域发现近似算法以提高发现效率;在精炼阶段,提出了基于趋势和差异性的度量指标,实现了对应区域重构算法及重构参数启发性选择算法,保证了从密集区域中有效提取出符合覆盖范围约束的热门区域.在真实数据集上验证了该工作的有效性.

关键词: 移动对象;轨迹数据库;热门区域;数据挖掘

中图法分类号: TP311 **文献标识码**: A

中文引用格式: 刘奎恩,肖俊超,丁治明,李明树.轨迹数据库中热门区域的发现.软件学报,2013,24(8):1816−1835. http://www.jos.org.cn/1000-9825/4340.htm

英文引用格式: Liu KE, Xiao JC, Ding ZM, Li MS. Discovery of hot region in trajectory databases. Ruan Jian Xue Bao/Journal of Software, 2013,24(8):1816−1835 (in Chinese). http://www.jos.org.cn/1000-9825/4340.htm

Discovery of Hot Region in Trajectory Databases

LIU Kui-En[1], XIAO Jun-Chao[1], DING Zhi-Ming[1], LI Ming-Shu[1,2]

[1](National Research Center of Fundamental Software, Institute of Software, The Chinese Academy of Sciences, Beijing 100190, China)

[2](State Key Laboratory of Computer Science (Institute of Software, The Chinese Academy of Sciences), Beijing 100190, China)

Corresponding author: LIU Kui-En, E-mail: kuien@iscas.ac.cn

Abstract: Mining of the enclosed regions that are visited frequently by moving objects (i.e. hot region) is a critical premise for the discovery of movement patterns from trajectory databases, and restricting their coverage is the key to promote precision and efficiency for representation of trajectory patterns. Given a trajectory database, this paper studies how to discover these hot regions and how to constraint their size. A definition of hot region query with coverage constraints is presented with a filter-refinement framework to construct them. In the filter step, the study introduces a grid-based approximate schema to construction the dense regions efficiently; and in the refinement step, the study proposes two trend-based and dissimilarity-based measures, and designs corresponding algorithms and heuristic parameter selection method to rationally reconstruct the regions under the coverage constraints. Experiments on practical datasets validate the effectiveness of this work.

Key words: moving object; trajectory database; hot region; data mining

管理二维(或高维)空间上移动对象运动信息的轨迹数据库,及其在交通监控、基于位置的服务和移动计算

* 基金项目: 国家自然科学基金(61202064, 61003028, 91124001); 国家科技重大专项(核高基)(2012ZX01039-004); 国家高技术研究发展计划(863)(2013AA01A603); 中国科学院战略性科技先导专项课题(XDA06010600); 中国科学院重点部署项目(KGZD-EW-102-3-3)

收稿时间: 2010-04-01; 修改时间: 2010-07-28, 2012-03-16; 定稿时间: 2012-10-19

等领域的应用,是当前的研究热点.其中,大部分研究[1−8]集中在轨迹模式挖掘上,旨在找出大量移动对象在一个较长时间段内的相似行为,用来指导未来决策.例如,根据居民区的人群每天从家开车到办公区途中经过的热门线路,预测交通状况或改进公交线路.识别被移动对象频繁光顾的热门区域,是从轨迹数据库中挖掘运动模式的关键前提.有了热门区域,每条轨迹(或其分段)可被转化为一个区域(或其标识 ID)的有限序列,用于挖掘出区域的关联规则(association rules)[8]或频繁序列模式(frequent sequential pattern)[6].这里,热门区域和区域序列可被对应地视作数据挖掘中的频繁项集(frequent item)和事务(transaction).轨迹模式上的相邻区域间的连线上标注着转换信息,如频繁度、时间间隔、空间距离或主要路线等.该信息的收集与热门区域挖掘算法密切相关.另外,热门区域自身也能被直接应用.例如,作为交通的有效参考信息,使司机避开热门区域,从而达到缓解交通拥塞的目的.

尽管一些发现轨迹模式的研究工作考虑了热门区域的挖掘方法[2,7],但其主要兴趣集中在如何挖掘模式并将模式用于分类和预测.这些工作假设数据挖掘领域中已有聚类算法[4,9,10]可以轻易发现所需要的热门区域,但是很少考虑热门区域挖掘的效率和面积约束.首先,在真实的轨迹数据库中统计一个给定区域的热度,即所经过该区域不同的轨迹数目,会遇到以下问题:

1) 移动对象的采样位置是离散的,且移动对象的运动速度和采样间隔都是多样的,在两个相邻的更新之间还可能丢失了一些时间点上的采样值.所以在一个连续的空间上跟踪一个移动对象不确定运动是个难题,且在执行聚类运算时,必须设计专门的插值(或投影)操作来为丢失的时间点生成大量的(虚拟)位置.
2) 移动对象的规模随着无线通信技术和定位技术的日益普及而增长,且轨迹数据的规模随着时间增大,在这种背景下,聚类是一个时间消耗性过程.如,在不使用索引技术的情况下(如 R 树),DBSCAN 算法[11]的计算开销与聚类点的规模成平方关系,而且 R 树也被证实其在聚类任意形状时不是一个有效的结构[2].

此外,目前还缺乏热门区域覆盖面积约束的深入研究.该约束对需要精确表达轨迹模式的应用至关重要,比如在文献[12]使用轨迹模式来预测未来位置的工作中,一个城市的整个 CBD 区域或一条数公里长的主干道路都会被当作一个区域,一旦移动对象进入该区域,则很难知道它在这个区域里面的精确位置,而且区域之间线路预测的精确性更加难以保障.直观地给定两条表征从清华大学到人民大学的轨迹模式:模式 1“清华大学→四环路→人民大学”和模式 2“清华大学东门→保福寺桥→中关村一桥→海淀黄庄路口→人民大学东门”,基于紧凑区域挖掘的模式 2 可更精细地刻画移动对象运动,更适用于诸如车辆未来线路预测等实际场景.

热门区域面积过大的原因在于移动对象的运动是连续的,所以仅凭空间区域被访问的频繁度并不足以限制热门区域的大小.如图 1 所示,随着一个对象移动,它将经过一个连续的空间区域,包含在该区域里面的所有子区域的频繁度都将增加.所以,一个能够覆盖连续的高密度子区域的热门区域的边界是难以约束的,我们称其为大区域问题(big-region problem).

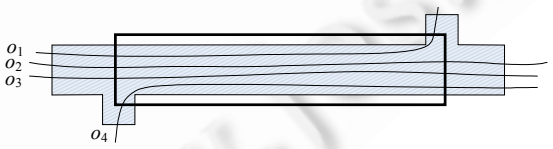


Fig.1 Big-Region problem

图 1 大区域问题

限制区域覆盖范围的合理性在于:可根据用户需求来限定热门区域覆盖范围.比如,根据预测精度(预测结果与实际运行轨迹之间的最大容忍偏差)来设定区域的最大半径,具有实际应用价值;可有效调节热门区域数目规模,达到权衡管理开销和数据覆盖率的目的.对于基于这些热门区域所挖掘出的轨迹模式的精确性和有效性起到积极促进作用,其效果可参见本文相关工作[13].

本文给出带有覆盖范围约束的热门区域定义,并提出一种基于过滤-精炼(filter-refinement)策略的热门区域发现方法.本文的主要贡献包括:

1) 针对大区域问题给出了热门区域的明确定义;
2) 充分考虑了轨迹数据的不确定性和聚类算法的复杂性,设计了一种基于网格的密集区域发现优化算法;
3) 提出了基于趋势和差异性的度量指标,并基于这两个度量指标实现区域重构算法及重构参数启发性选择算法;
4) 基于真实数据集的实验结果验证了本文工作的有效性.

此外,本文部分算法已经以组件方式(即 MOIR/HR[14])部署在移动对象管理平台 MOIR[15]上.

本文第 1 节介绍相关工作.第 2 节给出问题描述.第 3 节和第 4 节分别细述热门区域的发现方法及两个大区域的重构算法.第 5 节给出实验结果.第 6 节总结全文.

1 相关工作

目前,数据挖掘领域已有众多的空间数据聚类算法[4,9,10],如 *k*-Means,BIRCH[16],DBSCAN[11]和 STING[17],而本节侧重于从历史轨迹中发现移动对象密集区域的相关工作,并根据轨迹数据的时间属性将其分为以下两类:

(1) 同步区域:是指从移动对象同步运动的轨迹上发现的密集区域.基本思路是:给定一个轨迹数据库 D,在每个时间点 t 上,生成所有移动对象在该时间点上的位置快照 S,并最终从 S 中发现移动对象的密集群集(cluster)及它们的闭包作为区域边界.

Hadjieleftheriou 等人[18]最早定义了区域密度的概念.一个区域的密度可通过单位面积上特定时间间隔内所经过的移动对象的数目来测量,即 $density(r,\Delta t)=\min_{\Delta t}N/area(r)$,其中, $\min\limits_{\Delta t} N$ 为时间区间 Δt 内任意时间上区域 r 内移动对象的最小数目.然后,Jensen 等人[19]提出了二维的密度直方图(density histogram)及基于离散余弦变换(discrete cosine transform)的压缩形式,进一步改善了密度查询的效率.Verhein 等人[20]提出了类似的密集区域(dense region 或 hot spot)概念,并进一步考虑了密度的变化,扩展定义了大交通量区域(high traffic region)和稳定区域(stationary region).

(2) 异步区域:对应地,我们把那些从移动对象的异步运动轨迹中发现的密集区域称为异步区域.在一些场景里,给定一组移动对象和它们的历史轨迹,即便是这些轨迹具有相似的空间形状,轨迹之间仍可能存在相当大的时间偏差而且找不到明显的周期性属性,我们也难以将它们聚类在一起.所以,从移动对象在每个时间点 t 的位置快照里挖掘同步区域的方法难以适用于该情况.

文献[7,13]给出了如何发现这类区域的方法.比如,在文献[7]给出了关注区域(regions-of-interest)的查找步骤:首先,将移动对象的运动空间分割为一些子区域(比如小的网格单元);然后,每条轨迹投影在这些子区域上,并增加与这条轨迹(可能)相交的子区域的密度;最后,可以通过传统的聚类算法来合并相邻的密集子区域以获得最终的关注区域(也即密集区域).只要这些子区域足够小,该方案就可以提供较好的近似结果.

每时间间隔 T 后反复出现的周期性模式,可以基于同步区域发现的方法获得.即,首先将一个较长的轨迹分成等间隔的片段,然后使用上面的方法在相同偏移的时间点发现密集区域.Jeung 等人[8]采用 DBSCAN 算法[11]从周期性的时间点上发现频繁区域(frequent region),文献[2]引入了一种基于网格的 DBSCAN 近似算法来加速频繁区域(又称 frequent 1-pattern)的获取过程.

此外,机器人导航及计算机视觉等领域中的最新研究采用卡尔曼滤波(Kalman filter)和粒子滤波技术(particle filter)实现对象跟踪预测[21,22],如文献[23]提出的基于卡尔曼滤波的迭代运动函数计算方法(RMF)和文献[24]基于粒子滤波进行的跟踪实验.区别于这类移动对象跟踪预测相关工作[13,23],本文侧重于历史时空信息挖掘,但这些先进方法和技术可被借鉴并用于本文研究问题的扩展与优化.

本文的研究背景符合综述性文献[25]中对空间模式的分类的范畴,但是单纯依靠空间密度进行轨迹分析的方法适应性有限,比如会导致本文所提出的大区域问题.近年来,学者们在空间密度的基础上引入了更多度量因

素(尤其是时间),用于丰富所挖掘的区域的语义.比如,文献[26]引入了滞留时间,定义了停靠点(stay point)概念;文献[27]给出了带有时间、空间属性的密集区域的可视化方法;文献[28]给出了带有时空语义的区域发现框架,以获得诸如兴趣(interest)与活动(activity)等信息.研究表明,引入时间因素(如文献[26])并不能有效解决大区域问题,文献[28]给出了较详细的解释.区别于这些工作,本文侧重于在空间密度基础上引入运动(motion)属性.该研究思路与转向区域(turning region)[29]较为相似,但后者缺少限定区域面积的方法.

虽然文献[18,19]在问题定义中要求限定密集区域面积的大小,$\alpha_1 \leqslant area(r) \leqslant \alpha_2$($\alpha_1,\alpha_2$ 即为面积阈值),但是没有给出有效的解决方案.上述工作都难以有效解决大区域问题.本文注重移动对象的连续运动特性,基于区域的密度和面积阈值定义了一种适用于轨迹模式挖掘的异步区域,并给出了一种基于网格的热门区域快速发现方法及两个针对大区域问题的精炼算法.

2 问题描述

本文研究如何从轨迹数据库中发现热门区域的问题.下面给出该问题的明确描述.

定义 1(运动轨迹). 一个移动对象的历史运动轨迹可由一个长度为 n 的时空点序列 l 组成:$\langle l_1,l_2,\ldots,l_n\rangle$,这里,$l_i$ 是一个三元组(x_i,y_i,t_i),表示了移动对象在时间 t_i 时刻处于位置(x_i,y_i).进而,$l_{i,j}$ 表示轨迹 l 上从时间点 l_i 到时间点 l_j 的一段(其中 $j>i$).

定义 2(区域密度). 给定一个封闭的空间区域 r 和一段时间间隔Δt,该区域的密度(density)可表示为

$$density(r,\Delta t)=\min_{\Delta t} N/area(r).$$

其中,$N(r,\Delta t)$是在时间段Δt 里穿过区域 r 的轨迹的数目,$area(r)$是区域 r 的面积.

基于密度的区域发现技术在捕捉任意形状和抗噪特性上都有优势[11].非规则的区域边界对轨迹模式挖掘至关重要,因为固定形状的边界容器**难以表达移动对象轨迹真实的密度分布,并在实际应用中遇到多种问题,比如,难以覆盖连续的密集区域导致冗余模式问题(redundant pattern problem[2])和遗漏部分数据导致群组丢失问题(flock-lossy problem[30]).

定义 3(热门区域). 空间区域 r 被称为热门区域(hot region),当且仅当它满足以下条件:

(1) 它是密集区域,即它的区域密度大于等于一个密度阈值δ.

(2) 它有一个紧凑的边界约束.例如,它既能覆盖一个半径为α_1 的圆形,又能被一个半径为α_2 的圆形所覆盖,即它的大小受限于一个约束半径区间$[\alpha_1,\alpha_2]$.

(3) 任意两个热门区域不重叠.

条件(1)给出了密集区域的阈值下限,条件(2)保证了所获得的区域是有意义的.理论上,一个足够小的区域或一个足够长的狭窄带状区域,只要一条轨迹从中穿越,其密度就可以超过阈值δ,所以我们需要指定区域边界的上下限.这里,我们使用约束半径而不是文献[18,19]中常用的面积约束,也是因为考虑到移动对象运动模式的实际意义.例如,一条(空间填充)曲线可能穿过很多运动轨迹,其覆盖面积仍然可以满足最大面积约束.而约束半径范围将保证这些极端情况不会出现.为了简化多边形与圆形内含和外包的空间关系判断开销,本文在后面算法描述时将圆形约束转换为:一个热门区域必须能够覆盖一个边长为$2\alpha_1$的正方形,并被一个边长为$2\alpha_2$的正方形所覆盖.这种转换并没有改变其在上述定义中的半径约束意义,所以不影响定义的合理性和算法的有效性.而在具体的应用环境中,可使用更加精细的、支持任意形状的约束边界.条件(3)保证了热门区域的无冗余性.

定义 4(热门区域发现). 给定轨迹数据库 D、密度阈值δ及空间约束α_1和α_2,找出所有符合定义的热门区域.

在发现结果中的任一热门区域将表示成一个封闭空间区域和一组(与该区域相交的)轨迹,其中,轨迹的数目除以空间区域的面积是大于等于密度阈值δ的,且区域范围满足约束区间$[\alpha_1,\alpha_2]$.

** 比如,由有限条线性不等式所定义的线性容器:最小边界矩形(minimum boundary rectangle)和凸包(convex hull).又如,边界由一个二次表达式给出的边界容器:包络球(bounding ball)和椭面(ellipsoid).

3 基于网格的热门区域发现方法

为了提高热门区域发现的效率,本文采用基于网格的方式发现热门区域.其优势在于,发现效率较高,并且重新聚类的开销仅与网格的单元格数目有关,不会随移动对象数目的增多而明显增加.一旦网格建立,轨迹数据库中移动对象和新轨迹记录的增加就不会对热门区域发现造成太大的影响.

整个发现过程可以简化为:先把移动对象的运动空间分割为许多不重叠的单元格;然后,问题转化为如何从这些单元格找出符合热门区域定义的相邻单元格集合.因为每个移动对象的运动轨迹都是可知的(存储在轨迹数据库中),可推测该对象的连续运动并找出其所有可能经过的单元格.通过维护每个单元格被经过的次数,可知每个单元格的密度.那些从未被访问过的单元格被丢弃,以减少聚类的搜索空间;最后,采用过滤-精炼策略查找所有的热门区域.在过滤阶段,依据密度阈值的约束条件合并相邻的单元格,得到任意形状和大小的密集区域(但可能不满足约束半径限制);在精炼阶段,重构超出约束半径上限α_2的密集区域、过滤掉低于约束半径下限α_1的密集区域,直到所有候选结果都符合热门区域定义.

该方案在实施时需要先解决以下问题:

- 由于运动轨迹是离散采样的,存在着严重的采样误差(sampling error problem[31]),所以区域的密度难以统计.即,给定两个连续的采样坐标P_1和P_2(对应时间点t_1和t_2)和移动对象的最大运动速度v_{max},对于时间变量$t_x(t_1<t_x<t_2)$,该移动对象在P_1和P_2坐标点之间所有可能的运动轨迹满足一个椭圆形覆盖区域.如图 2 所示,该椭圆以P_1和P_2为焦点,以$a=v_{max}\times t/2, b=\sqrt{a^2-c^2}$为长、短轴.
- 给定密度分布,不同聚类策略会产生不同的结果.我们需要一种能够捕捉密集的、覆盖面积大的区域发现算法.
- 基于密度的聚类方法不能解决大区域问题(参考图 1 所示).实验结果表明:大区域在所有候选密集区域中占有相当大的比例[13].除了密度以外,我们还需要更多能够反映移动对象运动的时空属性及重构算法来精炼大区域.

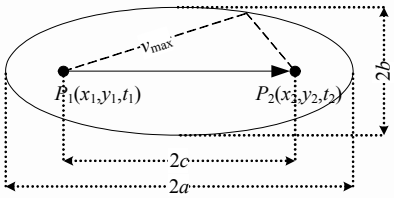


Fig.2 Sampling error problem

图 2 采样误差问题

第 3.1 节和第 3.2 节将分别给出不确定性密度统计方法和热门区域发现算法.第 4 节将给出针对大区域问题提出的两个新的时空属性及基于其上的两种重构算法.

3.1 不确定性密度统计

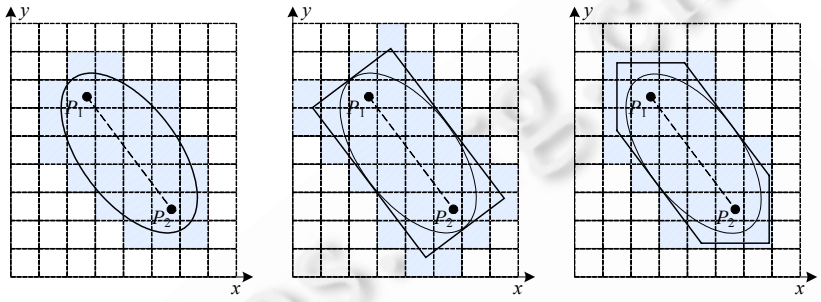
为了简化描述,我们只考虑均匀分布的网格,即每个单元格c都是面积等大的正方形,并存储被访问次数N_c.网格是区域发现过程中的主要数据结构,单元格越小,维护网格结构所消耗的内存越多.所以,单元格的边长(记为ε)不能太小.然后,过大的单元格又将直接导致结果精度降低.这里,我们设置$\varepsilon=\alpha_1$作为单元格边长***,从而获得一个新的阈值$N_{min}=\varepsilon^2\cdot\delta$,表示经过密集区域的轨迹的最小数量.以此方式,密度阈值δ转换为轨迹数目阈值N_{min},便于后面发现算法中进行条件判断.

考虑到数据采样的不确定性(特别是采样误差问题),我们把每个移动对象可能出现的位置都投影到网格的

*** 单元格边长$\varepsilon=\alpha_1$是经验值.实验结果表明:当取该值时,密集区域发现结果中符合热门区域定义的比例最大,且轨迹覆盖率均高于 99%.

单元格上.即,如果一条轨迹的不确定外延和某一单元格相交,则将该单元格的被访问次数加 1.图 3 给出了一个密度统计的例子,其中,P_1 和 P_2 是同一条轨迹上的两个连续的采样点.如图 3(a)所示,椭圆形外延给出了采样点 P_1 和 P_2 之间的移动对象可能出现的范围.所有与该椭圆区域相交的单元格(即图 3(a)中被灰显部分)的被访问次数均加 1.另外,如果一个轨迹多次投影到一个单元格上,且这些投影对应的轨迹段处于不同的时期(即连续两次投影的采样时间点不是连续的),则当作多条轨迹统计;否则,该单元格的被访问次数只增加 1.

直接沿椭圆曲线(可被表示为二次表达式 $Ax^2+Bxy+Cy^2+Dx+Ey+f=0$)遍历所有相交单元格的计算开销较大,所以本文引入了两个线性近似来简化投影范围的边界:最小边界矩形(minimal bounding rectangle,如图 3(b)所示)和对称六边形(bilaterally symmetrical hexagon,如图 3(c)所示).对称六边形是由两个围绕 P_1 和 P_2 的正方形(边长为 $\sqrt{2}b$)及其顶点的连接线所构成的最大封闭区域构建.



(a) 椭圆 (b) 最小边界矩形 (c) 对称六边形

Fig.3 Density projection ellipse and two linear approximations

图 3 密度投影椭圆和两个线性近似

令 E,R,H 分别表示椭圆、最小边界矩形和对称六边形,使用图 2 中所标变量,3 种边界的面积可表示为 $Area(E)=\pi ab$,$Area(R)=4ab$ 和 $Area(H)=4bc+2b^2$,则其紧凑度对比可以由下面不等式给出:

$$\begin{cases} Area(H) \geqslant Area(R), & \text{if } \dfrac{5}{3}c \geqslant a \geqslant c \\ Area(H) < Area(R), & \text{if } a > \dfrac{5}{3}c \end{cases}.$$

根据 a 和 c 所代表的物理意义,$2a$ 表示一个对象可以运动的最大距离,$2c$ 表示 P_1 和 P_2 之间的欧式距离.即,$2a=v_{\max}\times|t_2-t_1|$且 $2c\approx v_{avg}\times|t_2-t_1|$.根据这两个算式,我们可以参考不同地理区域上历史交通状况的先备经验来选择边界,比如在市中心车速较低区域选择对称六边形,在高速公路上选择最小边界矩形.

至此,除了网格结构以外,我们还将获得另外两个数据结构:

- 单元格投影表 HASH⟨*cell*,*trajectories*⟩,存储了投影在单元格(*cell*)上的轨迹集合(*trajectories*);
- 轨迹映射表 HASH⟨*trajectory*,$\overrightarrow{cells}$⟩,将轨迹映射到它所经过的单元格序列($\overrightarrow{cells}$).

网格是空间数据聚类中较为常用的数据索引方法之一,其主要优势在于其快速的处理时间.该时间只与所划分的单元格数目有关,而与数据对象规模无关[4].在本文的聚类和重构算法中,由于单元格密度值会被频繁访问,且单元格数据规模较大,在此情况下,与其他常规结构(如数组、红黑树等)相比,本文将哈希结构用于网格表述可以更好地平衡系统空间利用率与存取效率,即实现“空间换时间”的目的.实验部分验证了该方法的可行性.此外,选择近似最优的哈希函数、负载均衡的空间划分方法和划分粒度,可以进一步提高哈希结构的空间利用率和存取效率,但这不是本文研究重点,不再展开.

3.2 热门区域发现算法

本节给出热门区域发现算法,包括 5 个步骤(见算法 1).

算法 1. 热门区域发现过程.

1. procedure *HotRegionDiscovery* (Grid G, cell with ε, density threshold δ, spatial thresholds α_1 and α_2)
2. /* **Step 1**. **Grid-Based density counting** */
3. $\varepsilon \leftarrow \alpha_1/2, N_{\min} \leftarrow \varepsilon^2 \cdot \delta$
4. Init Grid G with cell width ε
5. project trajectories on G using any methods introduced in Section 3.1
6. /* **Step 2**. **Find out all core cells** */
7. **for** each cell $c \in G$ **do**
8. $c.flag \leftarrow true$
9. **if** $N_c \geqslant N_{\min}$ **then**
10. label c as core cell
11. /* **Step 3**. **Find out all core regions** */
12. $R_{core} \leftarrow \varnothing$
13. **for** each cell $c \in G$ **do**
14. **if** c_i is a core cell and $c.flag$ is true **then**
15. $r \leftarrow$new core-region($\{c\}$)
16. **while** true **do**
17. **if** $\exists$ a core cell $c' \in r.neighbours$ and $c'.flag$ is true **then**
18. $r \leftarrow r \cup \{c'\}, c'.flag \leftarrow false$
19. **else**
20. $R_{core} \leftarrow R_{core} \cup \{r\}$
21. /* **Step 4**. **Find out all dense regions** */
22. $R_{core} \leftarrow \varnothing$, $R_{candidate} \leftarrow R_{core}$
23. **for** each region $r \in R_{candidate}$ **do**
24. $C \leftarrow \{c \in r.neighbours \wedge c.flag\}$ is true
25. $c_0 \leftarrow$the dense cell in C
26. **if** $N_{c_0} \neq 0$ and $\left(\sum_{c \in r} N_c + N_{c_0}\right) \Big/ (|r|+1) \geqslant N_{\min}$ **then**
27. update r as $r \cup \{c_0\}$ in $R_{candidate}$
28. $c_0.flag = false$
29. **else**
30. $R_{dense} \leftarrow R_{dense} \cup \{r\}$
31. remove r from $R_{candidate}$
32. /* **Step 5**. **Refine dense regions till the area constraints are fulfilled** */
33. $R_{hot} \leftarrow refine(R_{dense}, \alpha_1, \alpha_2)$
34. return R_{hot}

Step 1. 统计投影密度.基于上节密度统计方法,可以得到密度阈值 $N_{\min}$ 和每个单元格 c 的被访问次数 N_c.

Step 2. 发现核心单元格.对于每个单元格 c,如果 N_c 大于等于 $N_{\min}$,标识其为核心单元格(core cell).

Step 3. 发现核心区域.通过将所有邻接的核心单元格组合起来(即上下左右和 4 个对角线方向),我们得到一组核心区域(core region).任意核心区域互不重叠.当某单元格被合并入一个核心区域时,其状态被设置为 *false*(即,我们在余下的发现过程中将不需再考虑该单元格).

Step 4. 发现密集区域.对于每个核心区域 r,我们收集 r 中所有单元格所邻接的非核心单元格,并将这组非

核心单元格根据各自被访问次数排序,将 N_c 最大的单元格记为 c_0.如果 c_0 的被访问次数不为 0,且被并入 r 后新区域仍满足密集阈值约束,即 $\left(\sum_{i\in r} N_i + N_{c_0}\right) \Big/ (|c|+1) \geqslant N_{\min}$ 且 $N_{c_0} > 0$, 我们将单元格 c_0 并入 r,并将该单元格状态标识为 *false*;否则,r 作为扩展完毕的密集区域加入结果列表(即算法中 R_{dense}).这些结果被称为密集区域而不是热门区域,这是因为这样获得的结果中可能存在不符合热门区域覆盖范围约束的区域.我们称密集区域中单元格为密集单元格(dense cell),称未被并入任一密集区域的单元格为非密集单元格.这些操作将保证所发现区域的密度和边界都尽可能地大和紧凑.我们对每个核心区域重复这些操作,直到每个核心区域都扩展完毕.

由于引入边界简化算法和散列类的数据结构,区域发现算法效率非常高.实验结果显示,给定一个 300 万坐标点的数据集,本文基于网格的密度聚类算法比使用椭圆边界和基于网格 DBSCAN 聚类的算法要快约 4 000 倍.

Step 5. 发现热门区域.从 Step 4 获得的密集区域可能不满足热门区域定义中的覆盖范围约束条件,需要引入除密度之外,更多合理的时空属性来重构这些结果,以获得最终的热门区域(即算法中 R_{hot}).

第 4 节将介绍两种新的时空属性——趋势性和差异性,以及基于其上的两种重构算法.

4 大区域重构算法

过小的区域可以简单丢弃处理，但如果一个区域太大,如覆盖一条高速公路或一个繁华街区的密集区域,则在其上所搭建的轨迹模式的精确性和有效性将难以保障.本节着重解决大区域问题.

密度阈值δ(或者 $N_{\min}$)影响着大区域在所发现密集区域中占据的比例.提高该阈值可以减少大区域的数目,但同时也降低了区域的覆盖率和有效性.所以,通过调整密度阈值来减少大区域的方法不适用于热门区域发现.此外,那种在单元格合并过程中直接添加一个最大尺寸约束的做法也不可取,因为该方法会将一个大区域分割为多个连续碎片,难以反映移动对象的运动趋势.这些因素促使我们设计新的重构算法来生成更加紧凑、更有意义的热门区域.本节引入两种重构算法来解决大区域问题:基于趋势的区域重构算法(trend-based region reconstruction,简称 TBRR)和基于差异性的区域重构算法(dissimilarity-based region reconstruction,简称 DBRR),并讨论这两种算法中所需参数的选取方法.

4.1 基于趋势的区域重构算法(TBRR)

根据区域发现算法 1,每个大区域都由一组邻近的密集单元格构成.我们观察到,当一个对象进入(退出)一个区域时,它最初(最后)遇到的那些单元格对表征该对象的运动趋势比其他单元格有更大影响.如图 4(a)中例子所示,区域 r_1 中的单元格 A 和 D 扮演着比 B 和 C 更重要的角色,因为移动对象在靠近区域内部的运动趋势很大程度上是被移动对象进入/退出该区域时的位置所主导(或限制).此外,被移动对象频繁访问的单元格也比低密度的单元格更重要.

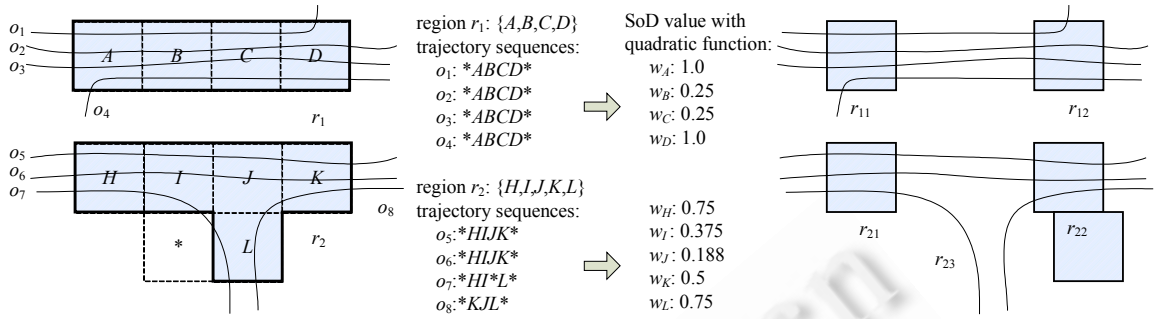
本节引入一个新的度量指标:主导积分(score of domination,简称 SoD).即,当移动对象穿越大区域时,我们可以根据该区域中每个单元格距离入口/出口单元格的距离来度量该单元格对运动趋势的表征能力,进而累计得出每个单元格总的 SoD 值.给定 SoD 阈值 $S_{\min}$,大区域中 SoD 值大于等于该阈值的单元格可以重新聚合为更加紧凑的区域片.图 4 给出了 TBRR 算法的 3 个步骤:

1) 将轨迹转换为单元格序列;
2) 计算每个单元格的 SoD 值;
3) 使用这些 SoD 值来重构大区域.

下面分别介绍这 3 个步骤.

给定一个大区域 r,为了计数每个单元格离其最近的入口/出口单元格的距离,我们将与该区域相交的每个对象 o 的轨迹转换为序列"o:$c_0c_1\ldots c_n$",这里,c_i 可以是该大区域中的一个单元格或一个特殊的符号"*","*"指代整个空间上不包含在该区域中任一单元格,而且根据密度统计方法,序列上相邻项不会重复,即 $c_i \neq c_{i+1}$,$0 \leqslant i < n$.例如,在图 4(b)中,序列"o_7:*HI*L*"表示:移动对象 o_7 停留在区域 r_2 外面(可以是任何地方),在进入单元格 H,L 后被

发现,随后移出了区域 r_2,直到在单元格 L 中再次被发现并最终离开区域 r_2.



(a) 两个大区域 (b) 主导积分(SoD) (c) 重构结果(S_{min}=0.5)

Fig.4 Trend-Based region reconstruction (TBRR) algorithm

图 4 基于趋势的区域重构算法(TBRR)

这里假设 S_r 表示经过大区域 r 的所有轨迹转化的序列.对于 S_r 中任一序列 s 上任一非"*"单元格 c,它在 s 距离入口/出口单元格的距离 $d_{s,c}$ 表示为

$$d_{s,c}=\begin{cases}\min\{distance(c,*)\mid *\in s\}, & \text{if } c\in s\\ \infty, & \text{otherwise}\end{cases}.$$

如果 c 从来没有出现在序列 s 中,则 $d_{s,c}\to\infty$(即 $d_{s,c}^{-1}=0$);否则,$1\geqslant d_{s,c}^{-1}>0$..进而,我们可以得到多个以加权公式来计算 c 的 SoD 值,比如:

- 二次函数: $\omega_c=\dfrac{\sum_{s\in S_r}d_{s,c}^{-2}}{|S_r|}$;
- 指数函数: $\omega_c=\dfrac{\sum_{s\in S_r}2^{1-d_{s,c}}}{|S_r|}$;
- 阶乘函数: $\omega_c=\dfrac{\sum_{s\in S_r}d_{s,c}!^{-1}}{|S_r|}$.

在 TBRR 重构算法中,区域重新聚合的基本思想和区域发现算法 1 基本相同,除了 N_{min} 被替换为 S_{min} 以外.限于篇幅,这里不再赘述.图 4(c)给出一个 S_{min}=0.5 的重构例子.TBRR 算法的有效性与阈值 S_{min} 的取值密切相关.在 TBRR 重构算法中,关键参数 S_{min} 的选择问题将在第 4.3 节讨论.

4.2 基于趋势的区域重构算法(TBRR)

我们观察到的另一个现象是:大区域中覆盖着道路交叉口的单元格应该得到更多的关注.换句话说,任一密集单元格 c 都可能投影了一组轨迹 T_c,与它邻近的单元格上的不尽相同.因为差异性反映了移动对象更精细的分布情况,所以那些与周围邻接单元格差别较大的单元格比同一大区域中的其他单元格更重要.

根据该思想,我们提出另一个度量指标,即差异度(degree of dissimilarity,简称 DoD),来测量给定单元格的整体差异性.在计算 DoD 之前,我们先明确将要用到的一些概念(和对应的变量):

- 邻接关系:给定一个单元格 c,我们把与 c 直接相邻的单元格称为邻接单元格(最多 8 个),所有邻接单元格构成集合 NB_c;同理可得到一个区域 r 的邻接集合 NB_r,即 $NB_r=\{c\mid\exists c_i\in r\wedge c\in NB_{c_i}\wedge c\notin r\}$.
- 覆盖区域:在大区域 r 扩展它的邻接集合 NB_r,并去除被访问次数为 0 的单元格,得到覆盖区域.因为空的邻接单元格(即 $H(c_i)$=0,$c_i\in NB_r$)对热门区域的定义没有贡献,所以我们把所有空的单元格从 NB_c 和 NB_r 中剔除,以减少差异性计算的规模,即 $r'=r\cup\{c|c\in NB_r\wedge H(c)>0\}$.图 5(a)展示了两个覆盖区域的例子 r_1' 和

r_2',其中的虚线框表示 r_1' 和 r_2' 邻接的非空单元格.

- 单元格桶(cell bucket):对于一个给定覆盖区域 r'中的每个密集单元格 c,经过单元格 c 的移动对象组可表示为 $H(c)$,可由第 3.1 节得到的哈希表快速得到.图 5(b)展示了单元格桶的例子,灰色的部分对应于那些大区域外部的邻接单元格.

使用上面的变量,给定大区域 r,其覆盖区域 r'中的任意两个邻接单元格 c_i,c_j 的差异性可用如下公式表示:

$$diss(c_i,c_j)=1-\left|\frac{H(c_i)\cap H(c_j)}{H(c_i)\cup H(c_j)}\right|.$$

考虑到被访问次数多(或称热度)的单元格对保障重构后区域密度更重要,定义单元格 c 的热度如下:

$$hotness(c)=\frac{|H(c)|}{|\cup_{c_i\in r}H(c_i)|}.$$

对于空的单元格,其热度是 0,这也是算法中先删除它们的数学原因.

我们用 $DISS(c)$表示单元格 $c(c\in r)$与它所有的邻接单元格差异性之和,即

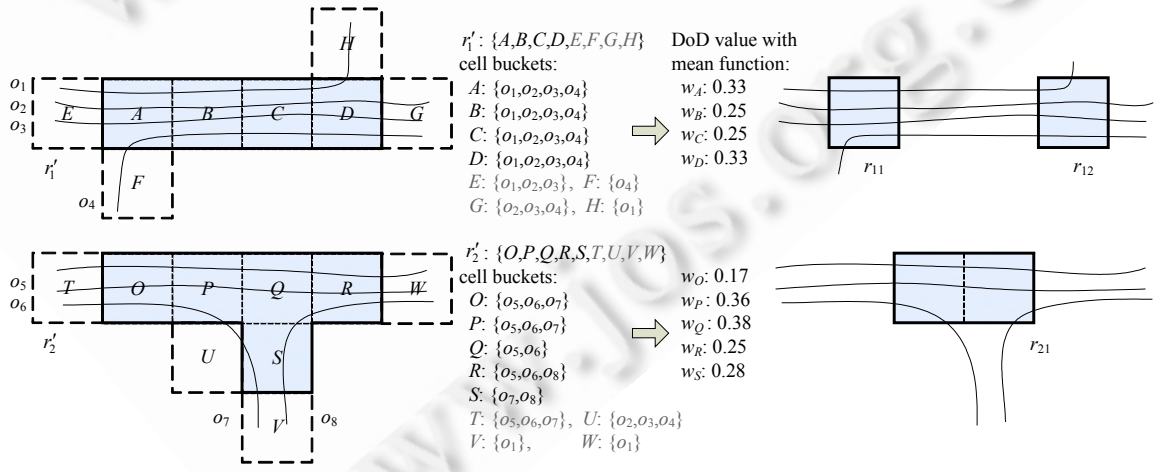
$$DISS(c)=\sum_{c_i\in NB_{c'}}diss(c,c_i),$$

则单元格 $c(c\in r)$的 DoD 值可通过下面类似的聚合公式来计算:

- 平均值估计函数: $\omega_c=\frac{DISS(c)}{|r'|}\cdot hotness(c)$;
- 标准化加权和函数: $\omega_c=\frac{DISS(c)}{\max\{DISS(c_i)\,|\,c_i\in r\}}\cdot hotness(c)$;
- 二阶原点距平方根函数: $\omega_c=\sqrt{\frac{\sum_{c_i\in NB_{c'}}diss(c,c_i)^2}{|r'|}}\cdot hotness(c)$.

给定 DoD 阈值 $D_{\min}$,可采用与 TBRR 和区域发现算法 1 类似的思路,将 DoD 值大于等于 $D_{\min}$ 的单元格重新聚合.

图 5 用上一节中的同一个例子演示了 DBRR 算法的全过程,特别是图 5(c)给出了一个 DBRR 算法进行区域重构的结果,显示出与 TBRR 算法的区别,如路口被优先保留.图 5(c)中,DoD 是基于均值估计函数测量的并设定阈值 $D_{\min}$=0.3.



(a) 两个扩展的大区域 (b) 差异度(DoD) (c) 重构结果($D_{\min}$=0.3)

Fig.5 Dissimilarity-Based region reconstruction (DBRR) algorithm

图 5 差异性区域重构算法(DBRR)

表 1 给出了图 5 中大区域 r_2 的 DoD 值计算例子.每个单元格的 DoD 值可由上面任一聚合函数得到.

Table 1 An example of degree of dissimilarity calculations

表 1 一个 DoD 值计算例子

单元格标签	单元格热度	不同的聚合函数			
		全局和	标准和	平均值	平方根
O	0.75	0.67	0.18	0.17	0.29
P	0.75	1.92	0.52	0.36	0.42
Q	0.75	2.00	0.55	0.38	0.43
R	0.75	0.75	0.20	0.25	0.32
S	0.50	2.75	0.50	0.28	0.31

4.3 重构参数选择

主导积分 SoD 和差异度 DoD 的阈值(即 $S_{\min}$ 和 $D_{\min}$)是影响 TBRR 算法和 DBRR 算法有效性的关键参数.本节将讨论它们的取值问题.

根据实施范围可将重构参数分为两类:全局阈值和单独阈值.

全局阈值是指事先为重构算法指定一个全局固定的阈值,所有大区域均采用这一阈值重构.如果重构结果中仍然有大区域,则使用同样的阈值再次重构这些区域,直到所有结果均符合热门区域定义.但该方法适用性有限,理由如下:

1) 对于不同的数据集或同样的数据集在不同时间阶段,所适用的阈值可能不同;
2) 大区域所覆盖的面积和轨迹模式的复杂度都是不同的,固定的阈值不能适用于多个大区域;
3) 大多数情况下,没有先验知识可以用来启发阈值的选择,所以难以保证给定阈值一定有效(即能够将所有的大区域重构为热门区域).

单独阈值是指为每个大区域选择一个适合该区域的重构阈值.它可以解决全局阈值缺陷,但需要快速、合理地选择取值.

本节给出迭代试探法(recursive cut-and-try method,简称 RCTM)来选取单独阈值和执行重构算法.

假设一个大区域 r 由 n 个单元格组成,每个单元格的权重(SoD 或 DoD)均已得出.以 TBRR 重构算法为例,RCTM 算法步骤如下:

(1) 将 r 中单元格按 SoD 值以降序方式排序,所得权重序列记为 $c_1c_2…c_n$.

(2) 采用启发性方法选中序列中某一单元格(如 c_i)的 SoD 值,作为重构阈值 $S_{\min}$.

(3) 在 r 上执行 TBRR 重构算法,得到区域集合 R.

(4) 如果 R 中仍存在大区域,则对 R 中每个大区域 r',从第 1 步再次迭代执行.

(5) 直到 r 被完全重构为符合定义的热门区域集合为止.迭代中得到的最大 $S_{\min}$ 为该区域 r 的单独阈值.

下面给出两种启发性方法的例子:

- 选择序列中最靠近 p%位置(如 50%)单元格的权重作为推荐的阈值.如,每次迭代时,从权重序列 $c_1c_2…c_m$ 中选择单元格 c_i,$i=\lceil p\%\times m\rceil$.
- 从降序序列中挑选出相邻的权重落差最大的两个单元格,并将其中较大的权重作为推荐阈值.

前一种是定量方法,后一种是定性方法.第 1 种方法旨在有效减小大区域的规模,第 2 种方法则考虑度量值分布的不均匀性.这两种方法的共性是:它们都对区域的面积不敏感.换句话说,无论区域有多大,基于这两种方法的得到的推荐阈值都可以同样有效地执行重构算法.

根据阈值的选择方法,一次重构至少过滤掉 i–1 个单元格(i 为每轮试探中所选阈值的位置),所以每次重构得到的区域集合的总覆盖面积必然小于重构之前.多次迭代后所得结果,要么全部符合热门区域定义,要么为空.即,RCTM 算法是收敛的.例如,使用第 1 种启发方式(令 p%=50%),平均迭代次数在 3~7 次之间.

5 实 验

本节通过以下两部分实验来验证所提出的方法的有效性和效率:

- 第 1 部分:首先,分析不同密度阈值下所发现的密集区域中大区域所占比例;然后,使用不同的重构参数和权重计算函数对密集区域进行重构,分析 TBRR 算法和 DBRR 算法的有效性;最后,收集热门区域发现过程中不同环节的时间开销.
- 第 2 部分:首先给出基于独立阈值进行重构的效果,然后对比不同重构参数选取方法的效率,最后给出热门区域发现实例.

本文的实验环境是在 Windows XP 操作系统上使用 C++所开发,并运行在一台配置为 2.83G Intel 4 核 CPU, 2GB 内存和 250G 硬盘的 PC 机上,所有算法处理都运行在主存空间.

5.1 数据集和度量指标

本实验采用了两组真实数据集,都是从车载 GPS 设备获得的.我们选择这两组数据集主要是因为考虑到它们相异的特性,特别是移动对象数目及数据分布情况的明显区别.每个数据集描述如下:

- Truck:该数据集包含希腊雅典城区 50 辆卡车的 276 条轨迹[32].这些卡车负责运输混凝土到几个建筑工地,每 90 秒左右采样一次,记录了 33 天.
- Taxi:该数据集来自 MOIR 项目[33],记录中国北京城区出租车的轨迹.我们抽取 3 天的数据用于本次实验.该数据集包含 10 283 辆出租车的真实轨迹,采样间隔从几秒到几分钟不等,平均在 5 分钟左右.由于出租车的目的地和路线随机性较大,所以运动模式尤为复杂.

为了研究热门区域的有效性,我们给出 4 个度量指标:

1) 热门区域规模:分别测量重构前和重构后的热门区域数目.
2) 空间覆盖率:重构后热门区域的空间覆盖面积与重构前面积之比.
3) 关联完整度:重构后热门区域之间的关联关系的完整程度,可由每条轨迹平均穿越的热门区域数计算得到.
4) 轨迹覆盖率:热门区域对轨迹的覆盖程度,可由热门区域所相交的轨迹数除以全部轨迹数得到,即

$$trajectoryCoverage=\frac{|\{l\,|\,r\otimes l,\forall r\in\mathbb{R},\forall l\in\mathbb{L}\}|}{|\mathbb{L}|}.$$

这里,$\mathbb{L}$和$\mathbb{R}$表示数据库中的轨迹集合和发现的热门区域集合,用⊗表示相交关系,用||表示有限集合的计数操作.一个理想的热门区域挖掘算法可以用尽可能小且尽可能少的区域来覆盖尽可能多且尽可能完整的轨迹.即前两个指标应尽可能地小,而后两个指标应尽可能地大.

实验结果将与文献[18,19]中定义的密集区域(记 FixRgn[18])和文献[7,12]中提出的流行区域(记为 PopRgn[7])进行对比.文献[18]最早定义了密集区域查询,文献[19]对其进行了完善,它们都采用特定形状(如固定大小的正方形)来约束密集区域范围.流行区域[7,12]与本文所关注的热门区域最具有相似性和可比性,比如,都是基于网格技术进行聚类,且都以轨迹模式挖掘为研究背景.这些工作具有一定代表性与影响力,比如,文献[7]被引用的次数超过 200.

5.2 数据集和度量指标

我们首先实验了基于网格的密集区域挖掘算法,分析基于密度的聚类结果中过大和过小区域的比例(设 α_1=100,α_2=300.比如,无法被 600×600 的正方形所覆盖的区域被判定为大区域.图 6 给出了在 Taxi 和 Truck 两种数据集上采用不同密度阈值($N_{\min}=\delta\times10^4$)得到的密集区域数目.可以看出,当 $N_{\min}$=5 时,对于 Taxi(Truck)数据集,发现的大区域占所有区域的比例为 15.8%(30.9%).为了说明在空间密度基础上引入新的度量指标(SoD 和 DoD)的必要性,我们将本文的大区域重构方法与文献[18,19]中限制区域面积的方法进行了对比.在文献[18,19]的问题定义中,有效的密集区域必须能够被特定的形状和覆盖范围所约束.本节实现了该思路的简化版本(FixRgn[18]).

图 7 给出了两类方法所生成的有效区域的数目.单纯依赖空间密度和特定形状进行聚类将产生大量区域碎片,难以紧凑地表征移动对象的空间分布及运动趋势.由于二者存在着数量级上的差距,下文不再与这类方法进行更多的对比.

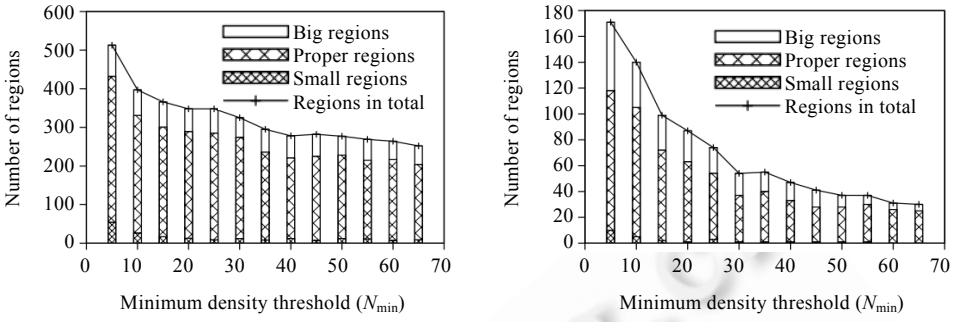


Fig.6 Area distribution of dense region

图 6 密集区域的面积分布

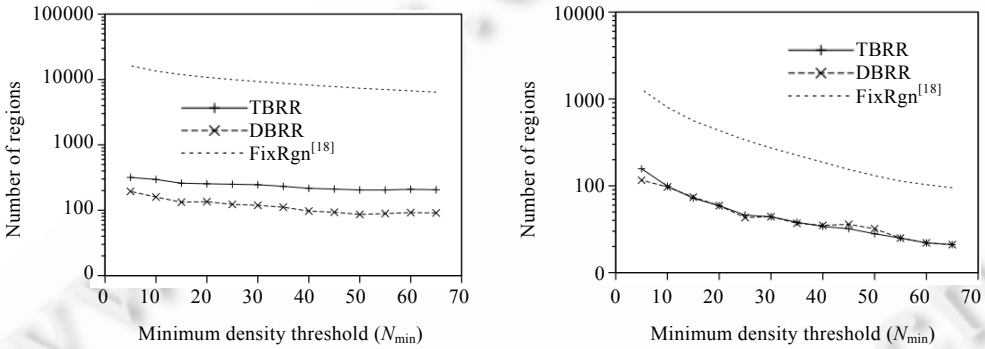


Fig.7 Comparing with dense regions constrained to a certain shape

图 7 与受限于特定形状的密集区域对比

本节设计了两组实验来验证覆盖范围约束条件(即α_1,α_2)对热门区域发现方法有效性的影响.首先固定α_2=600,随着α_1取值的逐渐增大,更多的小区域将被丢弃.这里的小区域还包括在重构过程中从大区域分裂出来的子区域.由于真实交通网络中支道和低级道路的数量要远多于干道,热门区域数量和关联完整度迅速下降.从图 8 上容易发现,热门区域数量和关联完整度的下降趋势与 PopRgn[7]基本一致,而对应的轨迹覆盖率和空间覆盖率下降较为平缓.其中,TBRR 算法在轨迹覆盖率上要优于 DBRR 算法,而 DBRR 算法在空间覆盖率上要优于 TBRR 算法.这是因为 DBRR 算法更容易捕捉到路口中心区域,而 TBRR 算法更容易保留路口岔道,其原理参见图 4 和图 5 示例.此外,从图 8(b)和图 8(f)可以看出,Taxi 上空间覆盖率对α_1的敏感程度要小于 Truck,因为出租车受限于城市路网,而卡车在工地和堆场上的活动范围要更灵活,所以α_1的取值要参照实际环境.

然后固定α_1=100.图 9 给出了α_2对热门区域发现算法有效性的影响.图中显示出几个明显的特征:

1) 随着α_2取值的增大,轨迹覆盖率和空间覆盖率呈现上升趋势;
2) 重构后的热门区域数据逐步逼近不受限的区域(PopRgn[7]);
3) 热门区域的关联完整度要优于 PopRgn[7].

尤其是,最后一项对轨迹模式挖掘较为有利.此外,从图 8 和图 9 中可以看出,Truck 数据集上空间覆盖率表

现出的优势不如 Taxi 明显,这是由于为了方便比较两个数据集采用了同一套重构参数,这套参数显然更适用于 Taxi 数据集.下面通过更多的实验对 TBRR 算法和 DBRR 算法的可行性进行补充.

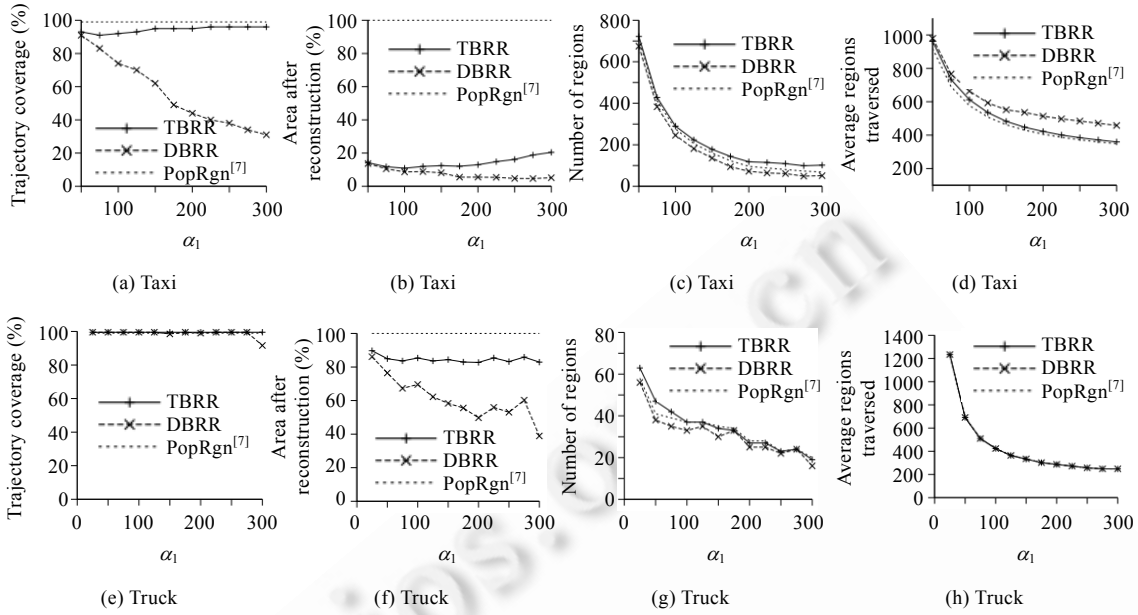


Fig.8 Impact of α_1 on the effectiveness of our algorithms (α_2=600, δ=0.02, *minSoD*=0.03, *minDoD*=0.04)

图 8 α_1 对本文算法有效性的影响(α_2=600,δ=0.02,*minSoD*=0.03,*minDoD*=0.04)

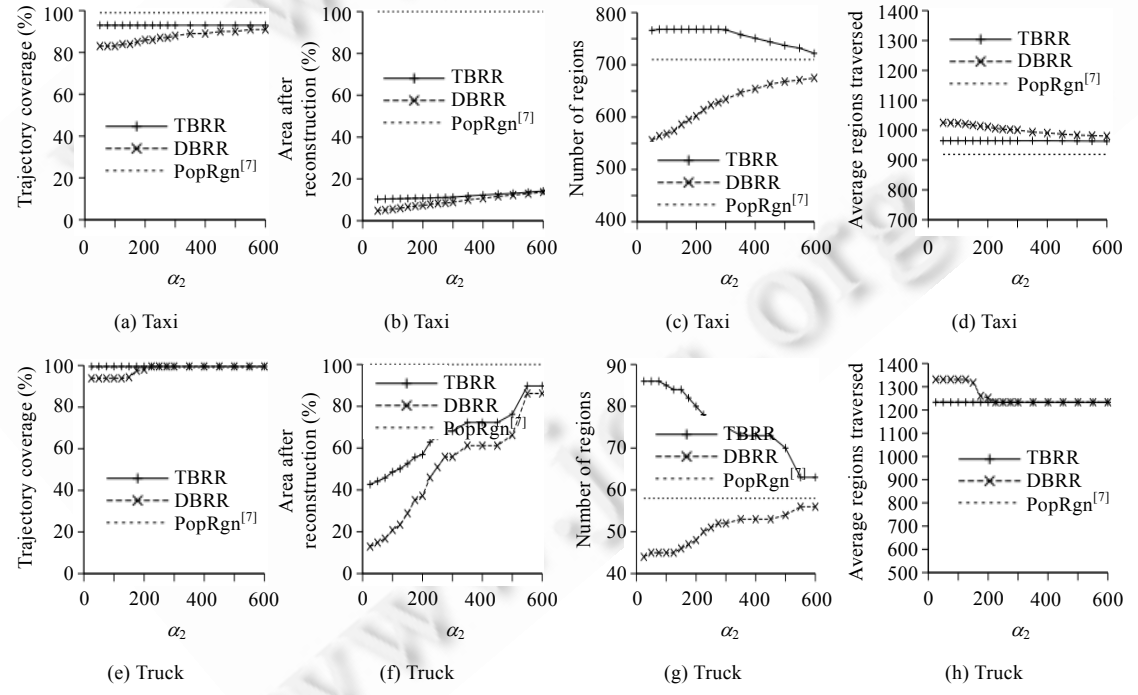


Fig.9 Impact of α_2 on the effectiveness of our algorithms (α_1=50, δ=0.02, *minSoD*=0.03, *minDoD*=0.04)

图 9 α_2 对本文算法有效性的影响(α_1=50,δ=0.02,*minSoD*=0.03,*minDoD*=0.04)

下面,我们对使用 TBRR 算法和 DBRR 算法来解决大区域问题的效率和性能进行了详细实验.这里,设置密度阈值为δ=0.015,空间约束为α_1=100,α_2=300.实验方法是先得到密集区域,然后用不同的全局阈值进行重构.在同一种算法(TBRR 或 DBRR)里,所有大区域重构时都采用相同阈值;如果重构结果中仍然存在大区域,则将这类结果收集起来,反复重构,直到全部合格为止.实验结果表明,一般重构 3~5 次即可得到最终结果;阈值越小,重构次数越多.

图 10 给出了使用 TBRR 算法进行区域重构的结果.从中我们可以得到以下结论:

1) TBRR 方法可有效减少所发现热门区域的面积,比如在 Taxi 数据集上,当 *minSoD*=0.015 时,轨迹覆盖率在 95.2%左右,而总的区域面积只有原来的 10.4%,这极大地提升了热门区域的表达能力和精度.
2) 与原有密集区域相比,重构后所得到的热门区域的数目随着阈值的增加而呈现先扬后抑的趋势.这说明,合适的阈值可以将大区域有效地重构为多个小区域,但过大的阈值会导致一些区域消失.
3) 虽然衰减速度较快的加权函数(如图 10 所示,指数 Expo-和阶乘 Fact-比二次 Quad-函数衰减的速度要快)所重构出来的结果具有更好的轨迹覆盖率,但其总覆盖面积会较大.
4) 全局的 *minSoD* 很难获得,特别是 Taxi 的运动模式复杂,运动趋势很难集中.如,当 Taxi 数据集上取 *minSoD*=0.5 时,轨迹覆盖率已经下降为 88.1%;而在 Truck 数据集上取 *minSoD*=0.5 时,重构后覆盖率仍然在 94.5%之上.

另外我们发现,当重构后区域数目接近原有密集区域时,轨迹覆盖率和区域面积变化开始平缓.这也给合理选择全局阈值带来启示.

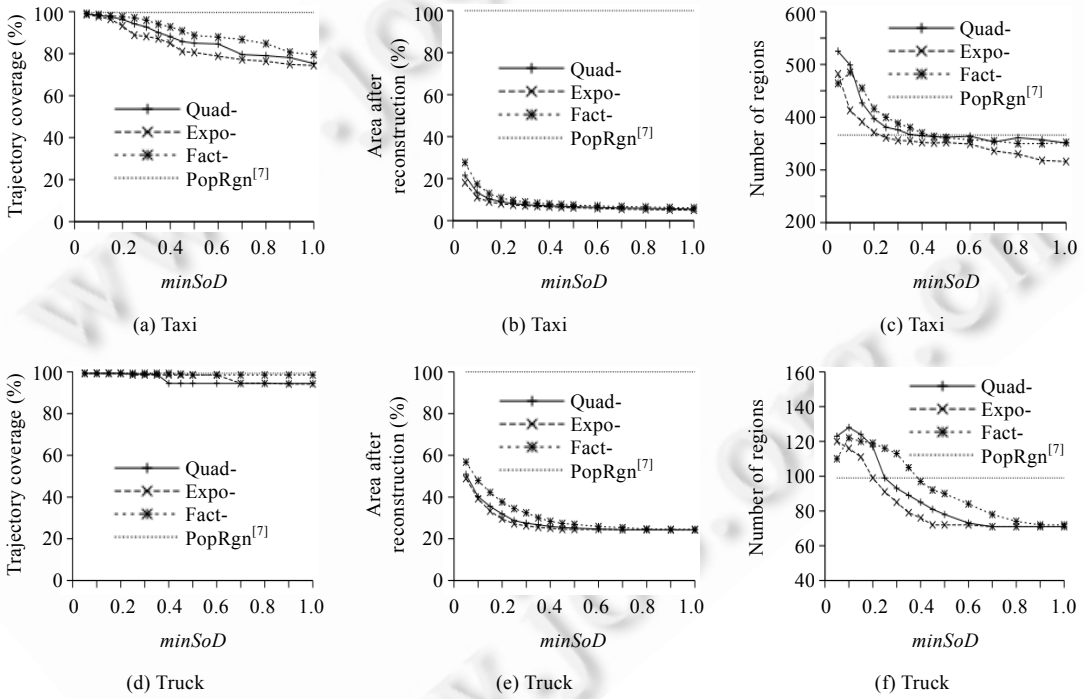


Fig.10 Effect of *minSoD* in TBRR algorithm (α_1=100, α_2=300, δ=0.015)

图 10 TBRR 算法中 *minSoD* 的影响(α_1=100,α_2=300,δ=0.015)

DBRR 算法给出了类似结果(如图 11 所示),但是 DBRR 算法的重构结果对阈值 *minDoD* 更加敏感,所以图中阈值都取得很小.尤为明显的是,Taxi 上的轨迹覆盖率和覆盖面积都下降得非常明显;Truck 上轨迹覆盖率下降得较慢,覆盖面积下降到 20%左右(如图 11(b)所示),而在 Taxi 上则下降到 5%(如图 11(e)所示).这是因为:

1) 道路网络特点不同.北京的交通路网规划为多级均匀网格,所以主要路段的交通量非常大(比如几个环城路和贯穿要道),依据密度所发现的区域边界较大(比如整条近60km的四环路),所以重构后区域面积会急剧下降;而雅典为点带状的星形交通线路,所以大区域的比例和面积相对较小.
2) Taxi 运动的目的性比较随机,差异性分布比较均匀,阈值上细微的差别往往会对重构结果产生较大影响;而 Truck 是从几个仓库到多个建筑工地之间的运动,目的性和路线相对集中.

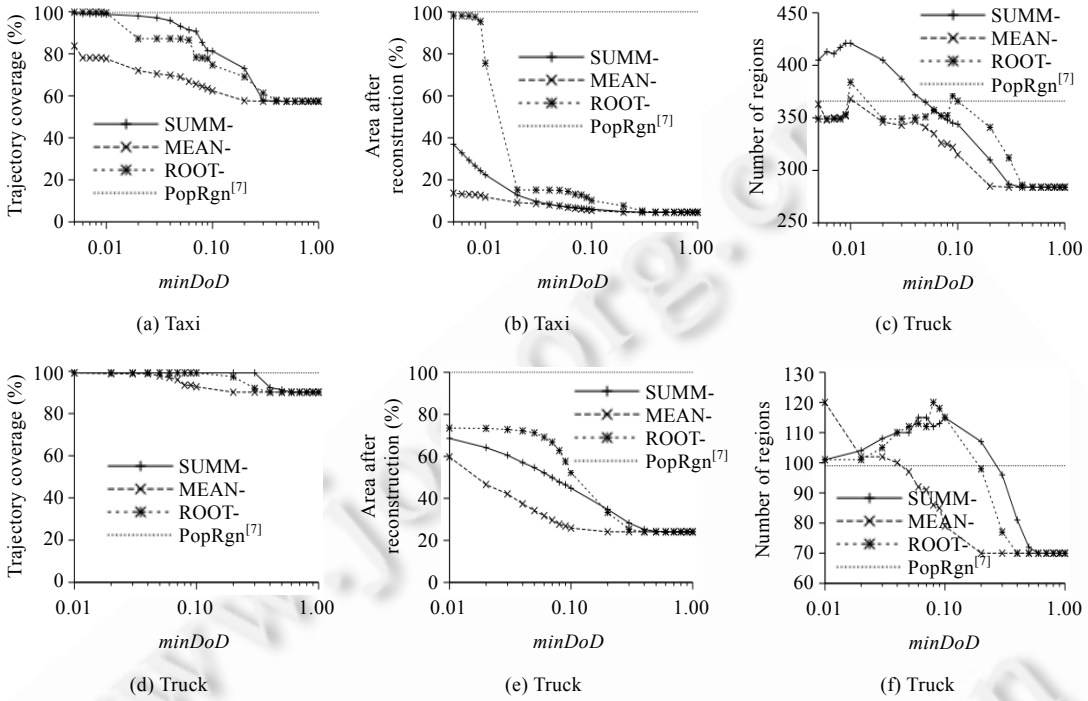


Fig.11 Effect of *minSoD* in DBRR algorithm (α_1=100, α_2=300, δ=0.015)

图 11 DBRR 算法中 *minSoD* 的影响(α_1=100,α_2=300,δ=0.015)

表 2 给出了热门区域发现过程中各环节的平均消耗时间.由表 2 可知,本文给出的区域发现算法有较高的效率.例如,在 Taxi 数据集上,在网格上投影一条轨迹所需的平均时间大约为 0.05ms,获得单个密集区域的平均时间开销大约为 0.2ms.大区域重构是热门区域发现过程占用时间最多的环节,因为这涉及到大量的轨迹集合排序、相交、合并等耗时操作,日后可做进一步改进.

Table 2 Time cost during the discovery process

表 2 发现过程中的时间开销

数据集	加载轨迹	网格投影	密集区域	TBRR	DBRR
北京 Taxi	30.515s	8.453s	1.922s	63s	79s
希腊 Truck	671ms	110ms	31ms	103ms	137ms

以上实验得出如下结论:若给定合适的阈值,大区域的重构算法则可以在保证轨迹覆盖率的情况下有效降低区域的覆盖面积,并给出符合热门区域发现约束的结果.

5.3 重构算法参数选择方法分析

本节实验首先分析了采用单独阈值进行大区域重构的可行性和效率.我们采用迭代试探法 RCTM 来得到每个大区域的单独阈值,并在 Taxi 和 Truck 数据集上对 TBRR 和 DBRR 算法中各计算函数进行了对比实验.表

3 给出了使用单独阈值重构的例子,说明了迭代试探法 RCTM 的可行性.例如,对比图 10 和图 11 中使用全局阈值的 TBRR 算法在 Taxi 数据集上发现的热门区域与表 3 中基于单独阈值的 TBRR 算法结果,在相同的面积覆盖率下,区域覆盖率能够提升 6%~9%.表 3 还给出了单独阈值的平均值和分布区间,可作为全局阈值的参考.

Table 3 An example of individual thresholds (*minSoD*, *minDoD*) used in TBRR (DBRR) algorithm

表 3 TBRR(DBRR)算法中使用单独阈值(*minSoD*,*minDoD*)的效果示例

实验数据集	重构算法	加权函数	重构参数			轨迹覆盖率	空间覆盖率	热门区域数
			平均	最小	最大			
Taxi 数据	TBRR	二次函数	0.109	0.016	0.222	0.881	0.063	392
		指数函数	0.124	0.014	0.295	0.873	0.060	368
		阶乘函数	0.208	0.027	0.371	0.875	0.061	374
	DBRR	标准加权和	0.112	0.007	0.221	0.855	0.058	357
		平均值估计	0.073	0.005	0.173	0.753	0.070	442
		原点距方根	0.237	0.011	0.379	0.784	0.077	544
Truck 数据	TBRR	二次函数	0.117	0.019	0.286	0.989	0.385	129
		指数函数	0.111	0.010	0.206	0.993	0.377	118
		阶乘函数	0.197	0.019	0.344	0.993	0.385	119
	DBRR	标准加权和	0.149	0.004	0.278	0.993	0.363	109
		平均值估计	0.042	0.007	0.096	0.989	0.409	120
		原点距方根	0.162	0.031	0.292	0.973	0.402	125

然后,我们实验了迭代试探法 RCTM 中提出的不同启发策略的效率(即试探次数).表 4 给出了实验结果.在这组实验中,第 1 种方法使用权重序列中的 50%位置作为推荐值.从表 4 可以得出以下结论:

1) 基于启发式的试探方法只需很少的试探(3~7 次)即可找到适合大区域重构的单独阈值;

2) 在 Taxi 数据集上,第 2 种方式的效率要要优于第 1 种方式.原因在于:出租车运动模式的趋势性和差异性差别较大,基于属性落差的单独阈值可以有效地减少搜索空间;而 Truck 数据集上数据分布差别较小,落差较大的权值更可能分布在权重序列的两端,所以基于第 1 种方法的效率更好.

Table 4 Effect of different heuristic methods

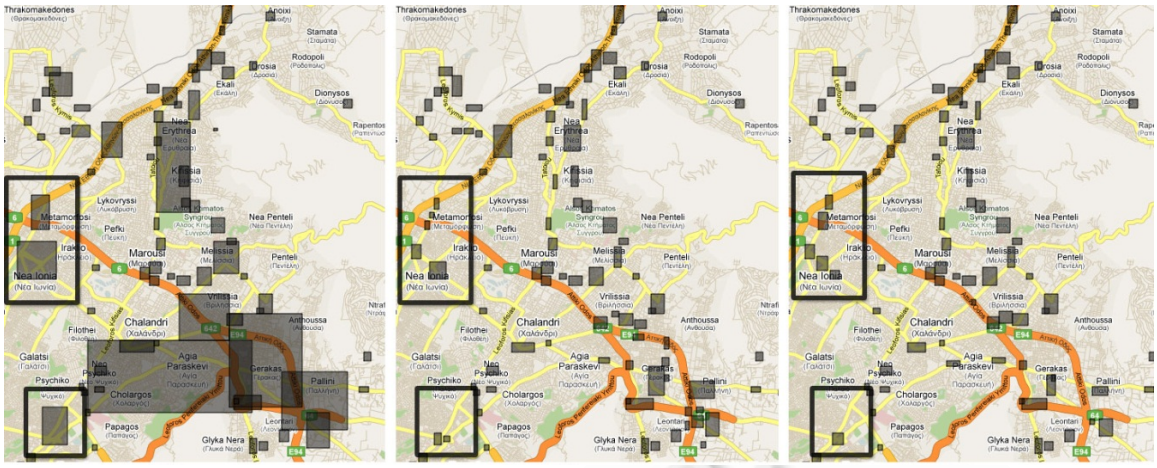
表 4 不同启发方法的影响

实验数据集	重构算法	加权函数	第 1 种启发方法	第 2 种启发方法
Taxi 数据	TBRR	二次函数	3.985	3.923
		指数函数	5.231	3.894
		阶乘函数	5.369	3.886
	DBRR	标准加权和	5.385	3.300
		平均值估计	4.923	4.666
		原点距方根	5.277	3.729
Truck 数据	TBRR	二次函数	3.815	6.309
		指数函数	5.037	5.096
		阶乘函数	5.148	5.309
	DBRR	标准加权和	5.185	4.877
		平均值估计	5.000	5.783
		原点距方根	5.222	5.999

最后,本文给出一个热门区域挖掘的实例,如图 12 所示.图 12(a)为基于密度发现算法得到的密集区域(其最小边界矩形(MBR)以深色方框标出),图 12(b)为在密集区域上执行 TBRR 重构算法后的结果,图 12(c)为在密集区域上执行 DBRR 重构算法后的结果.其中,TBRR 算法采用阶乘函数,DBRR 算法采用标准加权和函数,单独阈值均采用迭代试探法 RCTM 得到.从中我们看到:

1) 图 12(b)和图 12(c)与图 12(a)相比,重构后的热门区域的轨迹覆盖率高达 99:3%,但是总的面积覆盖却只有原来的 38:5%;

2) 图 12(b)和图 12(c)中对应的方框区域显示出 TBRR 算法和 DBRR 算法重构结果的差别.



(a) 密集区域 (b) 基于 TBRR 重构后的区域 (c) 基于 DBRR 重构后的区域

Fig.12 A practical example of hot region discovered in Athens, Greece

图 12 希腊雅典市热门区域发现实例

6 结 论

本文针对大区域问题(big-region problem)定义了带有覆盖范围约束的热门区域,给出了从离散轨迹数据库中发现热门区域的快速近似算法;提出了除区域密度外的两个时空属性:趋势性和差异性,并对应给出两种大区域重构算法以及重构参数选择算法.本文在两组真实轨迹数据集上进行了大量实验,所得到的可视化及量化结果验证了本文提出的热门区域发现算法和两种重构算法的可行性.

致谢 在此,我们向对本文的工作给予支持和建议的同行,尤其是中国科学院软件研究所的许佳捷、李亚光、何凤成在论文撰写及文章排版过程中给予的建议和帮助表示感谢.

References:

[1] Cayirci E, Akyildiz IF. User mobility pattern scheme for location update and paging in wireless systems. IEEE Trans. on Mobile Computing, 2002,1(3):236−247. [doi: 10.1109/TMC.2002.1081758]

[2] Mamoulis N, Cao HP, Kollios G, Hadjieleftheriou M, Tao YF, Cheung DW. Mining, indexing, and querying historical spatiotemporal data. In: Won K, Ron K, Johannes G, William D, eds. Proc. of the 10th ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining (KDD 2004). New York: ACM Press, 2004. 236−245. [doi: 10.1145/1014052.1014080]

[3] Yavaş G, Katsaros D, Ulusoy Ö, Manolopoulos Y. A data mining approach for location prediction in mobile environments. Data & Knowledge Engineering, 2005,54(2):121−146. [doi: 10.1016/j.datak.2004.09.004]

[4] Han JW, Kamber M. Data Mining: Concepts and Techniques. 2nd ed., San Francisco: Morgan Kaufmann Publishers, Inc., 2005.

[5] Wang YD, Lim EP, Hwang SY. Efficient mining of group patterns from user movement data. Data & Knowledge Engineering, 2006,57(3):240−282. [doi: 10.1016/j.datak.2005.04.006]

[6] Yang J, Hu M. Trajpattern: Mining sequential patterns from imprecise trajectories of mobile objects. In: Ioannidis YE, Scholl MH, eds. Advances in Database Technology, Proc. of the 10th Int'l Conf. on Extending Database Technology (EDBT 2006). Munich: Springer-Verlag, 2006. 664−681. [doi: 10.1007/11687238_40]

[7] Giannotti F, Nanni M, Pedreschi D, Pinelli F. Trajectory pattern mining. In: Berkhin P, Caruana R, Wu XD, eds. Proc. of the 13th ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining (KDD 2007). New York: ACM Press, 2007. 330−339. [doi: 10.1145/1281192.1281230]

[8] Jeung HY, Liu Q, Shen HT, Zhou XF. A hybrid prediction model for moving objects. In: Alonso G, Blakeley JA, Chen ALP, eds. Proc. of the 24th Int'l Conf. on Data Engineering (ICDE 2008). Washington: IEEE Computer Society, 2008. 70−79. [doi: 10.1109/ICDE.2008.4497415]

[9] Tan PN, Steinbach M, Kumar V. Introduction to Data Mining. Boston: Addison Wesley, 2005.

[10] Gan GJ, Ma CQ, Wu JH. Data Clustering: Theory, Algorithms, and Applications. Philadelphia: SIAM, 2007.

[11] Ester M, Kriegel HP, Sander J, Xu XW. A density-based algorithm for discovering clusters in large spatial databases with noise. In: Simoudis E, Han JW, Fayyad U, eds. Proc. of the 2nd Int'l Conf. on Knowledge Discovery and Data Mining (KDD'96). Portland: AAAI Press, 1996. 226−231.

[12] Monreale A, Pinelli F, Trasarti R, Giannotti F. WhereNext: A location predictor on trajectory pattern mining. In: John F, Elder IV, Fogelman-Soulié F, Flach PA, Zaki MJ, eds. Proc. of the 15th ACM SIGKDD Int'l Conf. on Knowledge Discovery & Data Mining (KDD 2009). New York: ACM Press, 2009. 637−646. [doi: 10.1145/1557019.1557091]

[13] Liu KE, Deng K, Ding ZM, Zhou XF, Li MS. Pattern-Based moving object tracking. In: Lu F, Xie X, Shaw SL, eds. Proc. of the 2011 Int'l Workshop on Trajectory Data Mining and Analysis (TDMA 2011). New York: ACM Press, 2011. 5−14. [doi: 10.1145/2030080.2030083]

[14] Liu KE, Ding ZM, Li MS. MOIR/HR: Mining of hot regions with coverage constraints. Journal of Computer Research and Development, 2010,47(z1):455−458 (in Chinese with English abstract).

[15] Ding ZM, Guo LM, Liu K, Wu H, Zhou XF. MOIR: A prototype for managing moving objects in road networks. In: Meng XF, Lei H, Grumbach S, Leong HV, eds. Proc. of the 9th Int'l Conf. on Mobile Data Management (MDM 2008). Washington: IEEE Computer Society, 2008. 219−220. [doi: 10.1109/MDM.2008.40]

[16] Zhang T, Ramakrishnan R, Livny M. Birch: An efficient data clustering method for very large databases. In: Widom J, ed. Proc. of the 1996 ACM SIGMOD Int'l Conf. on Management of Data (SIGMOD 1996). New York: ACM Press, 1996. 103−114. [doi: 10.1145/235968.233324]

[17] Wang W, Yang J, Muntz R. STING: A statistical information grid approach to spatial data mining. In: Jarke M, Carey MJ, Dittrich KR, Lochovsky FH, Loucopoulos P, Jeusfeld MA, eds. Proc. of the 23rd Int'l Conf. on Very Large Data Bases. San Francisco: Morgan Kaufmann Publishers, 1997. 186−195.

[18] Hadjieleftheriou M, Kollios G, Gunopulos D, Tsotras VJ. On-Line discovery of dense areas in spatio-temporal databases. In: Hadzilacos T, Manolopoulos Y, Roddick J, Theodoridis Y, eds. Proc. of the 8th Int'l Symp. on Spatial and Temporal Databases (SSTD 2003). Berlin, Heidelberg: Springer- Verlag, 2003. 306−324. [doi: 10.1007/978-3-540-45072-6_18]

[19] Jensen CS, Lin D, Ooi BC, Zhang R. Effective density queries on continuously moving objects. In: Ling L, Andreas R, Kyuyoung W, Jianjun Z, eds. Proc. of the 22nd Int'l Conf. on Data Engineering (ICDE). Atlanta: IEEE Computer Society, 2006. [doi: 10.1109/ICDE.2006.179]

[20] Verhein F, Chawla S. Mining spatio-temporal association rules, sources, sinks, stationary regions and thoroughfares in object mobility databases. In: Lee M, Tan KL, Wuwongse V, eds. Proc. of the 11th Int'l Conf. on Database Systems for Advanced Applications. Berlin, Heidelberg: Springer-Verlag, 2006. 187−201. [doi: 10.1007/11733836_15]

[21] Ristic B, Arulampalm S, Gordon N. Beyond the Kalman Filter: Particle Filters for Tracking Applications. Boston: Artech House Publishers, 2004.

[22] Yilmaz A, Javed O, Shah M. Object tracking: A Survey. ACM Computing Surveys, 2006,38(4):1−45. [doi: 10.1145/1177352.1177355]

[23] Tao YF, Faloutsos C, Papadias D, Liu B. Prediction and indexing of moving objects with unknown motion patterns. 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. 611−622. [doi: 10.1145/1007568.1007637]

[24] Hightower J, Borriello G. Particle filters for location estimation in ubiquitous computing: A case study. In: Davies D, *et al*., eds. Proc. of the 13th Int'l Conf. on Ubiquitous Computing 2011. Berlin, Heidelberg: Springer-Verlag, 2004. 88−106. [doi: 10.1007/978-3-540-30119-6_6]

[25] Zheng Y, Zhou XF. Computing with Spatial Trajectories. New York: Springer-Verlag, 2011. 143−177. [doi: 10.1007/978-1-4614-1629-6]

[26] Zheng VW, Zheng Y, Xie X, Yang Q. Collaborative location and activity recommendations with gps history data. In: Rappa M, Jones P, Freire J, Chakrabarti S, eds. Proc. of the 19th Int'l Conf. on World Wide Web (WWW 2010). New York: ACM Press, 2010. 1029−1038. [doi: 10.1145/1772690.1772795]

[27] Demšar U, Virrantaus K. Space-Time density of trajectories: Exploring spatio-temporal patterns in movement data. Int'l Journal of Geographical Information Science, 2010,24(10):1527−1542. [doi: 10.1080/13658816.2010.511223]

[28] Lu CT, Lei PR, Peng WC, Su IJ. A framework of mining semantic regions from trajectories. In: Yu JX, Kim MH, Unland R, eds. Proc. of the 16th Int'l Conf. on Database Systems for Advanced Applications (DASFAA 2011). Berlin, Heidelberg: Springer-Verlag, 2011. 193−207. [doi: 10.1007/978-3-642-20149-3_16]

[29] Huang GY, Zhang YC, He J, Ding ZM. Efficiently retrieving longest common route patterns of moving objects by summarizing turning regions. In: Huang JZ, Cao LB, Srivastava J, eds. Proc. of the 15th Pacific-Asia Conf. on Knowledge Discovery and Data Mining (PAKDD 2011). Berlin, Heidelberg: Springer-Verlag, 2011. 375−386. [doi: 10.1007/978-3-642-20841-6_31]

[30] Jeung HY, Yiu ML, Zhou XF, Jensen CS, Shen HT. Discovery of convoys in trajectory databases. Proc. of the VLDB Endowment, 2008,1(1):1068−1080.

[31] Brakatsoulas S, Pfoser D, Salas R, Wenk C. On map-matching vehicle tracking data. In: Böhm K, Jensen CS, Haas LM, Kersten ML, Larson PA, Ooi BC, eds. Proc. of the 31st Int'l Conf. on Very Large Data Bases (VLDB 2005). New York: ACM Press, 2005. 853−864.

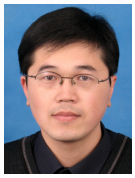
[32] Theodoridis Y. School-Buses and trucks. 2006. http://www.rtreeportal.org/datasets/trajectories/trucks.zip

[33] Liu KN, Deng K, Ding ZM, Li MS, Zhou XF. Moir/mt: Monitoring large-scale road network traffic in real-time. Proc. of the VLDB Endowment, 2009,2(2):1538−1541.

附中文参考文献:

[14] 刘奎恩,丁治明,李明树.MOIR/HR:覆盖区域受限的热门区域挖掘.计算机研究与发展,2010,47(z1):455−458.

刘奎恩(1983－),男,河南扶沟人,博士,助理研究员,CCF 会员,主要研究领域为数据库,分布式数据管理,时空数据及物联网数据管理,时空数据挖掘.
E-mail: kuien@iscas.ac.cn



肖俊超(1978－),男,博士,副研究员,CCF 会员,主要研究领域为软件过程管理,基础软件,数据管理.
E-mail: junchao@nfs.iscas.ac.cn

丁治明(1966－),男,博士,研究员,博士生导师,CCF 高级会员,主要研究领域为数据库与知识库系统,移动及时空数据管理,云计算,物联网,信息检索.
E-mail: zhiming@iscas.ac.cn

李明树(1966－),男,博士,研究员,博士生导师,CCF 高级会员,主要研究领域为软件工程方法,软件过程技术,需求工程,软件工程经济学,可信软件过程.
E-mail: mingshu@iscas.ac.cn