

基于时空图的移动对象聚集模式挖掘方法^{*}

张峻铭, 李静林, 王尚广, 刘志晗, 袁泉, 杨放春



(交换与智能控制国家重点实验室(北京邮电大学), 北京 100876)

通讯作者: 张峻铭, E-mail: longmumin@gmail.com

摘要: 移动对象聚集模式是指由移动对象参与的一组群体事件,通常用来预测交通系统中出现的异常现象.然而由于海量移动轨迹数据的产生,已有的研究方法难以准确、高效地挖掘特定的聚集模式.为此,提出一种基于时空图的移动对象聚集模式挖掘方法.该方法首先通过改进的空间聚类算法(DBScan)分析轨迹数据,从而获得移动对象聚类;然后,利用时空图模型代替单独存储轨迹数据的方式,用于实时观测移动对象聚类的时空变化特征.最后提出基于最大完全子图查找的聚集检索算法及其改进算法,用于查找满足时空约束的最大完全子图.基于真实大规模轨迹数据集上的实验结果表明,所提出的方法在移动对象聚集模式挖掘的准确性和高效性方面优于其他方法.

关键词: 聚集模式挖掘;时空图;轨迹数据

中图分类号: TP311

中文引用格式: 张峻铭,李静林,王尚广,刘志晗,袁泉,杨放春.基于时空图的移动对象聚集模式挖掘方法.软件学报,2016,27(2): 348-362. <http://www.jos.org.cn/1000-9825/4797.htm>

英文引用格式: Zhang JM, Li JL, Wang SG, Liu ZH, Yuan Q, Yang FC. Mining moving object gathering pattern method via spatio-temporal graph. Ruan Jian Xue Bao/Journal of Software, 2016,27(2):348-362 (in Chinese). <http://www.jos.org.cn/1000-9825/4797.htm>

Mining Moving Object Gathering Pattern Method Via Spatio-Temporal Graph

ZHANG Jun-Ming, LI Jing-Lin, WANG Shang-Guang, LIU Zhi-Han, YUAN Quan, YANG Fang-Chun

(State Key Laboratory of Networking and Switching Technology (Beijing University of Posts and Telecommunications), Beijing 100876, China)

Abstract: Moving object gathering pattern represents a group event or incident that involves congregation of moving objects, enabling the prediction of anomalies in traffic system. However, effectively and efficiently discovering the specific gathering pattern remains a challenging issue since the large number of moving objects generate high volume of trajectory data. In order to address this issue, this article proposes a moving object gathering pattern mining method that aims to support the mining of gathering patterns by using spatio-temporal graph. In this method, firstly an improved density based clustering algorithm (DBScan) is used to collect the moving object clusters. Then, a spatio-temporal graph is maintained rather than storing the spatial coordinates to obtain the spatio-temporal changes in real time. Finally, a gathering mining algorithm and its improved version are developed by searching the maximal complete graphs which meet the spatio-temporal constraints. The effectiveness and efficiency of the proposed methods are outperformed other existing methods on both real and large trajectory data.

Key words: gathering pattern mining; spatio-temporal graph; trajectory data

近年来,随着卫星定位技术的普及,越来越多的移动对象都安装了卫星定位系统.这项技术使我们获得了大

* 基金项目: 国家自然科学基金(61202435); 国家高技术研究发展计划(863)(2012AA111601); 北京市自然科学基金(4132048)
Foundation item: National Natural Science Foundation of China (61202435); National High-Tech R&D Program of China (863) (2012AA111601); Beijing Natural Science Foundation of China (4132048)

收稿时间: 2014-01-07; 修改时间: 2014-04-03; 采用时间: 2014-11-25; jos 在线出版时间: 2015-11-03

CNKI 网络优先出版: 2015-11-04 17:10:00, <http://www.cnki.net/kcms/detail/11.2560.TP.20151104.1710.001.html>

量的轨迹数据(也就是常说的时空数据),通过分析这些数据,使得获取移动对象的行为成为可能.基于此,国内外的许多研究人员正在通过挖掘移动对象的移动模式,为交通系统提供路况分析、线路规划等服务^[1,2](如,基于轨迹数据挖掘的实时路况导航服务).

在上述这些研究中,一个重要的研究方向就是如何挖掘移动对象聚集模式(本文简称聚集模式).聚集模式^[3]可以看成由一系列事件或者事故引起的一组移动对象的聚合,比如发生了车祸路段由车辆聚集引起的拥堵.为了统计或预测交通系统中可能出现的事故或者拥堵等现象,研究人员提出了聚集模式挖掘方法.该方法的基本原理是:通过移动对象的轨迹数据分析某一区域出现的密集移动对象群组,统计发生的移动对象聚集.其具体实现方法是:通过空间网格(grid)检索移动对象轨迹,查找密集的移动对象群组,再使用 TAD 算法检验每一个移动对象群组能否形成聚集.

为了通过海量的移动轨迹数据挖掘移动对象聚集模式,flock^[4]被定义为一组连续行驶 k 个时间片的群组,这个群组包含在一个固定大小的圆形区域中.而 Gudmundsson 等人^[5]用 flock 研究一个移动对象群组的最长共同行驶时间.Li 等人^[6]在 flock 的基础上提出了更加灵活的 swarm,它允许群组被包含在任意形状的区域中,其中任意两个对象的距离都小于阈值,群组中移动对象的共同行驶时间不需要连续,使用聚类算法挖掘轨迹中的移动对象群组.采用比以往宽松的移动群组定义方式,能够更广泛地挖掘移动对象群组.可以看出,上述研究大多不能观测动态变化的移动群组,特别是在使用流数据的环境下.Tang 等人^[7]提出一种增量式的算法来挖掘行驶中的伙伴,通过轨迹流可以动态地分析出移动对象的行驶伙伴,行驶伙伴可以用来进行移动对象间的资源分配、智能拼车与安全管控.文献[8-11]通过分析北京市的出租车轨迹数据来研究该城市交通系统中出现的异常现象.Sanjay 等人^[11]进一步通过汽车轨迹的聚合来模拟行驶路线,解释异常现象出现的原因.另外,通过相似性判断不同移动对象是否具有相同的移动模式^[12,13],Shi 等人^[13]使用分段聚类算法对轨迹进行先分段再聚类,挖掘兵棋作战系统中移动对象的运动态势.Jeung 等人^[14]提出了一种通过兴趣区来检测移动对象间是否具有相同的移动模式.这种方法通过检测不同对象经过兴趣区的时序序列的相似性来实现.Zheng 等人^[3]定义了移动对象聚集模式的概念,并且使用一种移动群组检测的算法来挖掘移动对象的聚集模式.聚集模式可以用来统计或预测交通系统中可能出现的事故或者拥堵等现象.但是,现有的研究只给出了聚集模式的发现算法,通过分析轨迹数据中的移动群组来发现移动对象聚集模式.这种方法在进行特定时间或者空间的聚集模式挖掘时,每次都需要重新遍历全部轨迹数据.比如,通过时间进行挖掘时,需要首先提取所需时间的轨迹数据,对其中的移动物体聚类后进行聚集模式发现.这样会耗费大量的时间,在交通系统中不能用来观测实时路况信息,预测也缺乏准确性.有些研究^[15-18]通过时间序列分析移动对象历史、现在和将来的位置.比如,Ding 等人^[19]设计了一种基于 R 树索引结构 NDTR-Tree,能够动态地索引和维护移动对象的当前及将来位置.Geraldine 等人^[20]提出了由移动对象组成的复杂网络,通过这个网络分析对象的移动特性.Mondo^[21]提出了强语义时空图模型,使用图来表示移动对象的时间和空间的变化特征.但是,这些研究只考虑单个移动对象的移动位置等信息,没有考虑移动群组的变化趋势;同时,由于移动群组中对象间都有很强的时空约束,传统的数据结构并不能完全适应聚集模式挖掘的时空约束条件.

针对上述不足,为了从海量的轨迹数据中准确、高效地挖掘特定的聚集模式,本文提出一种基于时空图的聚集模式挖掘方法,其核心是:通过使用最大完全子图查找算法检索由移动对象聚类组成的时空图,找出满足时空约束条件的最大完全子图,从而准确、高效地挖掘给定时间或者位置的聚集模式,可以为实时路况分析、路线规划等服务.主要贡献如下:(1) 引入了时空图数据模型.该模型由移动对象聚类构成,图中的每个节点除包含组成聚类的移动对象的信息外,还包含对应聚类的形成时间和位置,每条边记录两个聚类间的时空关系.时空图能够准确地反映移动对象聚类的时空变化特征.(2) 基于时空图,提出了移动对象聚集检索算法 GR.该算法使用最大完全子图代表移动对象聚集,通过查找完全子图的方式找出满足时空约束的移动对象聚集集合.根据移动对象聚集的特点,我们又提出了一种改进的移动对象聚集检索算法 GR+.它使用高效的极大团检索算法来查找最大完全子图,通过剪枝策略降低搜索空间,提高了移动对象聚集的检索效率.(3) 提出了一种基于时空图的移动对象聚集模式挖掘方法.首先,通过实时的轨迹数据对移动对象进行空间聚类;然后,使用移动对象聚类组成

时空图;最后,根据时空图利用聚集检索算法高效地查找满足时空约束条件的移动对象聚集集合.(4) 为了验证基于时空图的移动对象聚集模式挖掘方法,本文基于真实的出租车轨迹数据构建了仿真实验,通过准确性指标与效率指标的对比表明:我们的方法在准确性方面提供了较高的精度,在效率方面优于现有的研究.

本文第1节给出相关概念的定义和移动对象聚集模式挖掘方法,其中,第1.1节列出本文用到的相关概念和符号,第1.2节详述移动对象聚集模式挖掘方法,第1.2.1对移动对象进行聚类,第1.2.2节创建时空图,第1.2.3节提出聚集检索算法.第2节对本文所述方法进行验证,其中,第2.1节设置实验环境,第2.2节对方法的准确性进行对比,第2.3节对方法的效率进行对比.第3节总结全文.

1 研究问题

在本节中,我们首先定义了本文用到的相关概念和符号,然后详述了移动对象聚集模式挖掘方法.

1.1 相关定义

定义 1(轨迹快照). 设移动对象数据库是 O_{DB} ,与它相对应的时间数据库是 T_{DB} ,轨迹快照为 $S_i = \{(v_i, t_i) | v_i \in O_{DB}, t_i \in T_{DB}\}$,其中, S_i 是对应时间片 t_i 的轨迹快照,属于轨迹快照集合 S 的子集.

一个轨迹快照 S_i 是在 t_i 时刻一组移动对象及其位置的集合,不是所有在集合 O_{DB} 中的移动对象都会出现在 t_i 时刻的轨迹快照内.给出一个轨迹快照 S_i ,使用空间聚类算法找出在 S_i 上的密集的移动对象群组,如果这个群组的成员超过给定的阈值,这个群组就形成一个聚类.

定义 2(移动对象聚类). 令集合 $c_i = \{o_1, o_2, \dots, o_k\}$ 是对应于每一个时间片 $i(1 \leq i \leq k)$ 的移动对象集合.对于两个给定的阈值 m_c 和 k_c ,当满足以下条件时, c_i 是一个移动对象聚类:

- (1) c_i 中的成员数量 $size(c_i) \geq m_c$;
- (2) c_i 存在的一段时间,即生存时间 $time(c_i) \geq k_c$;
- (3) c_i 包含的任意两个移动对象 o_k 和 o_p 的距离不大于 δ ,可以表示为 $Dist(o_k, o_p) \leq \delta$,其中, $\forall o_k, o_p \in O_{DB}$.

移动对象聚类就是在给定的时间片上由一组移动对象形成的指定大小和形状的群组,其中的两个对象的距离都小于给定的阈值.根据定义 2,我们给出移动对象聚集的定义.

定义 3(移动对象聚集). 设一个移动对象聚类集合为 C_g ,当满足以下条件时, C_g 是一个移动对象聚集:

- (1) 对于时间阈值 μ ,移动对象聚类的形成时间 $c.t, |c_i.t - c_j.t| < \mu, c_i, c_j \in C_g$;
- (2) 对于给定的距离阈值 $\delta, D_H(c_i, c_j) \leq \delta, c_i, c_j \in C_g$;
- (3) 对于给定的成员阈值 $m_g, |C_g| \geq m_g$.

图 1 所示为 7 个移动对象在 5 个时间片内的移动轨迹,从图中可以看出, o_2, o_3, o_5 形成了一个移动对象聚类,而 o_4, o_6, o_7 形成了另一个聚类.虽然起始点不一样,但是两个移动对象聚类都驶向同一个方向,并且他们共同行驶了 4 个时间片,最终形成了一个聚集.表 1 列出了文中用到的符号.

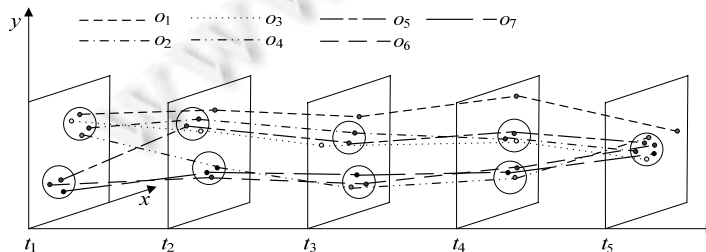


Fig.1 Trajectories of moving objects graph

图 1 移动对象轨迹图

Table 1 Table of notations

表 1 符号含义表

符号	解释	符号	解释
O_{DB}	移动对象数据库	m_c	移动对象聚类成员阈值
T_{DB}	时间域数据库	M_g	移动对象聚集成员阈值
o	O_{DB} 中的移动对象	δ	距离阈值
t	T_{DB} 中的时间点	G	时空图
S_i	t_i 时刻的轨迹快照	$D_H(c_i, c_j)$	两聚类间的 Hausdorff 距离
c_i	t_i 时刻的移动对象聚类	$n.l$	节点 n 对应聚类的形成位置
k_c	移动对象聚类生存时间阈值	$n.t$	节点 n 对应聚类的形成时间
μ	时间阈值	$N(n)$	节点 n 的所有邻居节点

1.2 移动对象聚集模式挖掘方法

本文所述方法的结构框架图如图 2 所示.它主要包含 3 个部分:移动对象聚类、时空图创建和移动对象聚集检索.

- 移动对象聚类.该过程(图 2 左 1 列)首先获得实时的轨迹数据(如汽车轨迹),按照一定的时间间隔划分时间片,将移动对象的轨迹分配到每一个时间片对应的轨迹快照中,得到轨迹快照后,使用基于密度的空间聚类算法对每一个快照中的移动对象进行空间聚类,使得聚类中任意两个移动对象的距离小于给定的阈值 m_c ,并且这个聚类的生存时间大于给定的阈值 k_c .
- 时空图创建.提取移动对象聚类的信息生成节点,每一节点包含组成聚类的移动对象信息,当聚类的形成时间和位置新加入一个节点时,遍历图中已有节点,如果两者的形成时间满足时间约束,那么连接这两个节点,每条边的权值为这两个节点对应的移动对象聚类之间的距离,时空图由移动对象聚类生成的节点和它们之间的边组成(图 2 左 2 列).
- 移动对象聚集检索.收到服务请求时,根据请求的位置或者时间信息在时空图中查找最大完全子图集合,将得到的最大完全子图集合进行聚集验证,如果集合中的最大完全子图满足聚集的时空性约束,则将其加入到候选集合中,然后对候选集合进行相似性验证,根据聚集的相似性定义移除重复的或者相似的项,得到聚集集合(图 2 左 3 列).

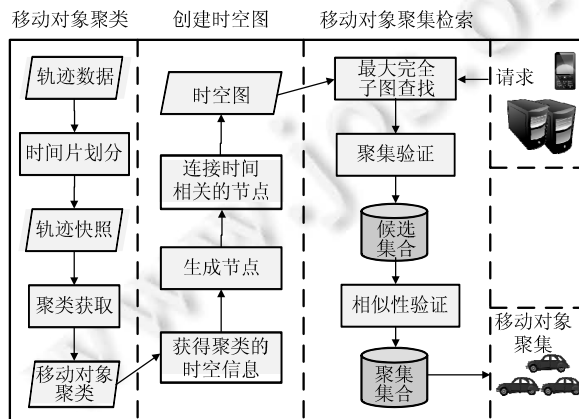


Fig.2 Structure of moving object gathering pattern mining method

图 2 移动对象聚集模式挖掘方法框架

1.2.1 移动对象聚类

移动对象的聚类算法使用的是一种基于密度的空间聚类算法,其搜索空间是历史和实时更新的轨迹数据,也就是我们所说的(O_{DB}, T_{DB}).在本文中,我们使用一种改进的 DBScan 算法来获得移动对象聚类.DBScan^[22]能够根据一个空间集合的子集的密度大小来判断这个子集是否为一个聚类,由于空间聚类特性,它能够检测出任意

形状的聚类.改进的 DBScan 算法由时间聚类和空间聚类两个过程组成.

通过时间聚类,把移动对象数据库 O_{DB} 划分为 $|T_{DB}|$ 个空间子集,每一个子集对应一个时间片 T_i .也就是说,将对象 $\{o_1, o_2, \dots, o_i, \dots, o_{T_{DB}}\}$ 的轨迹按照时间片划分到每一个快照 S_i ,其中, $1 < i < T_{DB}$.移动轨迹数据库被划分为快照集后,需要对每一个快照中的移动对象进行空间聚类.该过程扫描每个快照中包含的轨迹,计算一个对象和现有集合间的距离,如果一个对象 o_i 与一个集合的距离 $h(o_i, c')$ 小于给定的距离阈值 δ ,就把该对象加入 c' .快照扫描结束后,如果集合 c' 的成员数大于阈值 m_c ,并且它的生存时间大于时间阈值 k_c ,这个集合就形成一个移动对象聚类.计算一个对象和集合间的距离时使用的是 Hausdorff 距离,它广泛应用于计算机视觉领域和图像领域的点集间的距离计算公式.给定两个点集 C_1 和 C_2 ,它们之间的 Hausdorff 距离 $D_H(C_1, C_2)$ 可以定义为

$$D_H(C_1, C_2) = \max \left\{ \max_{c_{i1} \in C_1} \min_{c_{i2} \in C_2} d(c_{i1}, c_{i2}), \max_{c_{i2} \in C_2} \min_{c_{i1} \in C_1} d(c_{i1}, c_{i2}) \right\} \quad (1)$$

例 1:图 3 所示为 5 个移动对象的轨迹,我们使用 $\{o_1, \dots, o_5\}$ 来标记每一个移动对象.假设所有对象都只行驶 6 个时间片,对象 $\{o_1, \dots, o_5\}$ 在每一个时间片的位置都被标记为图中的黑点.聚类生存时间阈值 k_c 等于 3,聚类成员阈值 m_c 等于 2,距离阈值 δ 的大小如图所示, U 代表已经存在的移动对象的集合, c'_1 和 c'_2 是已经存在的临时移动对象群组, c_1 和 c_2 是移动对象聚类.系统在 t_1 时刻更新一个轨迹快照 S_1 ,移动对象 o_1 出现,我们把它加入到移动对象集合 U 中,在图中的每一个时刻都会更新一个轨迹快照,系统根据快照中移动对象的信息更新 U, c'_1, c'_2, c_1 和 c_2 .在时刻 t_2, o_2 和 o_3 出现,移动对象集合 U 更新为 $U = \{o_1, o_2, o_3\}$.在时刻 t_3 ,一个新的移动对象 o_5 出现,由于它与 o_2 之间的距离小于距离阈值 δ ,我们可以得到两个临时的移动群组集合 $c'_1 = \{o_1, o_3\}$ 和 $c'_2 = \{o_2, o_5\}$.最后,可以得到两个移动对象聚类 $c_1 = \{o_1, o_3\}$ 和 $c_2 = \{o_2, o_5\}$.因为它们都包含两个移动对象成员,作为临时移动群组的共同行驶时间超过时间阈值 k_c .需要注意的是,在时刻 t_5 出现的移动对象 o_4 不能被加入移动聚类 c_2 ,因为它在临时移动群组 $c'_2 = \{o_2, o_4, o_5\}$ 中只经过两个时间片,小于给定的时间阈值.

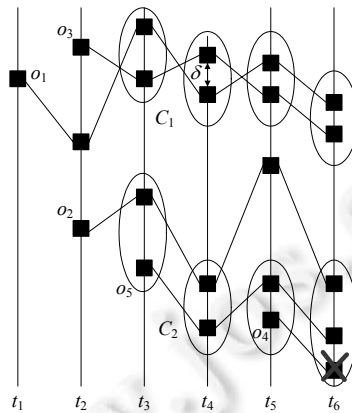


Fig.3 A sample graph of clustering moving objects at different timestamps

图 3 多时刻移动对象聚类示意图

1.2.2 创建时空图

一个时空图包含有限的节点和边,其中,节点对应一个移动对象聚类,连接节点的边记录聚类间的时空关系.聚类形成的时间和位置信息存储在聚类对应的节点中,如果两个聚类出现在同一个或相邻的时间片内,它们对应的节点间会有一条边,边的权值是两个聚类间的距离.详细定义如下所述:

定义 4(时空图). 集合 $N = \{n_1, n_2, \dots, n_n\}$ 代表图中的点集,对应的移动对象聚类 $C = \{c_1, c_2, \dots, c_n\}$,集合 $E = \{e_1, e_2, \dots, e_n\}$ 代表图中的边集,对应移动对象聚类间的时空关系,时空图 $G = (N, E)$ 由点集 N 和边集 E 组成.

定义 5(节点). 给定一个移动对象聚类 c_i ,它的形成时间 t 和位置 l ,那么节点 $n = \{t, l, \{o_i\} | \{o_i\} \in c_i\}$ 是时空图中的一个节点.

图中的每一个节点都会存储它对应的移动对象聚类的形成时间和位置,主要目的是:

- (1) 节点的位置信息可以使我们知道它所对应的移动对象聚类的形成位置,通过位置信息可以获得某一特定地点的聚集模式.同理,根据聚类的时间信息能够获得特定时间范围内的聚集模式.
- (2) 通过 t 和 l ,可以根据时空图设计聚集检索算法,更高效地进行检索聚集数.举例来说,我们想查询在 14:20 时在北京中关村附近的聚集,以往的方法需要应用聚类算法对轨迹数据进行扫描,多次迭代查找这一时间和地点的聚集.但是通过时空图,系统只需要根据时间信息先找到对应的节点集,搜索节点集对应的聚集,再对聚集进行位置分析即可,减少了计算复杂度.

定义 6(边). 给定两个图中的节点 n_i 和 n_j ,如果 $|n_i.t - n_j.t| < \mu$,那么 $e=(n_i, n_j)$ 是时空图中连接节点 n_i 和 n_j 的一条边,它的权值 $e.weight = D_H(c(n_i), c(n_j))$.

时间阈值 μ 用来衡量两个移动对象聚类是否在一个或者相邻的时间片内, μ 的值越大,聚集就越容易形成,同时估算也会相对不准确.另外,我们令图中边的权值为两个移动对象聚类的 Hausdorff 距离.

例 2:图 4 所示为时空图的一个例子.从图中我们可以看出,节点 A_1 有 4 个邻居节点,这意味着有 4 个移动对象聚类与 A_1 处于同一个或者相邻时间片内.如果我们将距离阈值设为 $\delta=400$,可以得到 3 个与 A_1 在同一区域的移动对象聚类,如图中与 A_1 用粗线连接的 3 个节点.

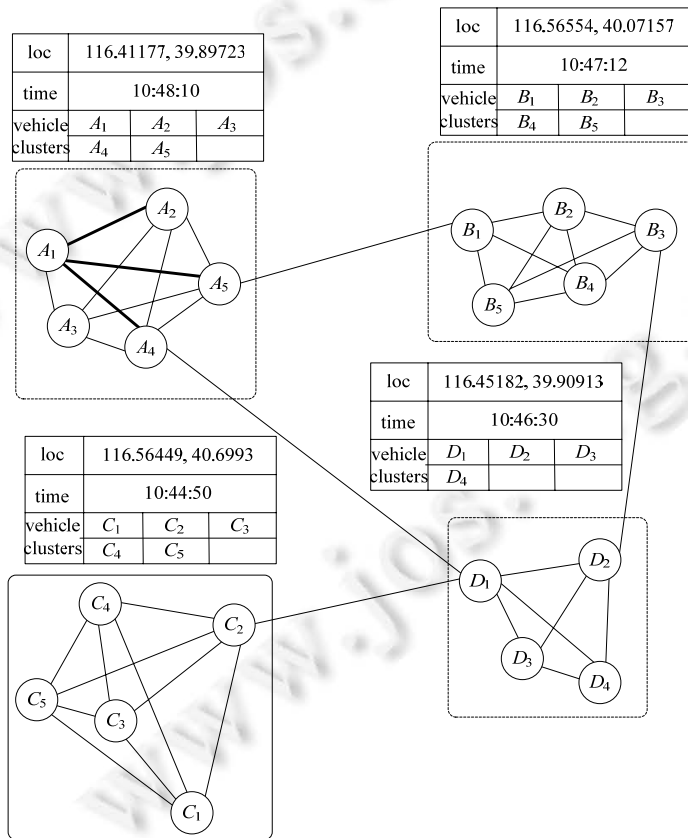


Fig.4 An example of spatio-temporal graph

图 4 时空图的一个例子

算法 1 列出了建立一个时空图所需的步骤:

- 首先,我们得到一个移动对象聚类的集合 $C = \{c_i\}^n$.集合中的每一个 c_i 都代表一个聚类,它同时记录着这个聚类的形成时间和位置.对于每一个 $c_i \in C, c_i.t$ 是其形成时间, $c_i.l$ 是其位置(算法 1 第 1 行~第 3 行),

图中的每一个节点都对应着一个移动对象聚类(算法 1 第 4 行),所有存在的聚类构成了图中的点集.

- 然后,我们检验 c_i 与图中所有节点所对应的聚类是否在同一或者相邻的时间片内(算法 1 第 5 行).
- 对于图中的每一个节点 n_j ,如果它们和节点 n_i 形成的时间差值小于 μ ,那么我们在它们之间创建一条边,把两个聚类之间的距离赋给这条边的权值(算法 1 第 7 行~第 10 行).

时空图使用的是无向图结构.

算法 1. 创建时空图.

输入:移动对象聚类集合 $C=\{c_i\}^n$,时间阈值 μ .

输出:时空图 $G=(N,E)$.

1. **for** $i=1$ to n **do**
2. $c_i.l \leftarrow \text{GetLocation}(c_i)$;
3. $c_i.t \leftarrow \text{GetTime}(c_i)$;
4. $n_i \leftarrow \text{CreateNewNode}(G, c_i, c_i.l, c_i.t)$;
5. $C'_i \leftarrow \text{FindQualifiedNeighbor}(c_i, G, \mu)$;
6. **if** $C'_i \neq \emptyset$ **then**
7. **foreach** vehicle cluster c_j in C'_i **do**
8. **if not** $\text{ExistEdge}(c_i, c_j)$ **then**
9. $e_i \leftarrow \text{AddEdge}(c_i, c_j)$;
10. $e_i.\text{weight} \leftarrow \text{GetDistance}(c_i, c_j)$;

1.2.3 移动对象聚集检索算法

从时空图中找出具有给定时空约束(时间或者位置信息)的节点集合,搜索集合中每一个节点的最大完全子图,得到一个最大完全子图集合.根据定义 3,一个聚集中的所有聚类都必须在相同或者相邻时间片,且在一个区域内.需要对集合中的每一个最大完全子图进行验证,得到满足时空约束的聚集集合.由此,本文提出一种查找最大完全子图的聚集检索算法 GR.然而,由于时空图规模的增大,最大完全子图检索需要很高的计算代价.为了降低算法的时间复杂度,本文使用高效的极大团检索算法 Bron-Kerbosch 降低检索的时间消耗,使用剪枝策略减少搜索空间,并据此提出一种改进的聚集检索算法 GR+.

1) 聚集检索算法 GR

根据定义 3,可以通过查找时空图中最大完全子图的方式来获得移动对象聚集数.为了找出包含给定节点的最大完全图,一个简单的方法是循环遍历与这个节点相连的子图.下面将给出一些定义和性质.

定义 7(完全图). 给定一个图 $G=(N,E)$, $\forall n_i, n_j \in N, \exists e=(n_i, n_j)$, 使得 $e \in E$, 那么 G 是一个完全图.

定理 1. 设移动对象聚集 $C_g = \{c_{g1}, c_{g2}, \dots, c_{gi}\}$ 和一个时空图 $G=(N,E)$, 存在一个与 C_g 对应的最大完全子图 $G' \subseteq G$, 使得 $C_g \subseteq G'$.

证明: 根据定义 4, $\forall c_{gi}, c_{gj} \in C_g$, 都有 $|c_{gi}.t - c_{gj}.t| < \mu$, 对应时空图 $G=(N,E)$ 中的一个子图 $G'=(N',E')$, 其中 $G' \subseteq G$. $\forall n_i, n_j \in N', \exists e=(n_i, n_j)$, 使得 $e \in E'$, 其中, c_{gi} 对应于 n_i , c_{gj} 对应于 n_j . 由定义 7, G' 是一个完全图. 根据定理 4, 不存在一个对应于移动对象聚集 C_g 的子图 G'' , 使得 $G'' \subseteq G'$. 所以, 对于一个移动对象聚集 C_g , 存在一个最大完全子图 $G' \subseteq G$, 使得 $C_g \subseteq G'$. \square

根据以上定义和定理,给出基于时空图的移动对象聚集检索算法 GR(以时间请求为例).它首先根据查询聚集的时间从时空图中找出一个点集(算法 2 第 1 行).对于点集中的每一个点,我们都找出包含这个点的最大完全图集合,然后验证它的大小是否超过成员阈值(算法 2 第 2 行~第 6 行).对于合格的完全图,我们验证它所包含的边的权值是否大于距离阈值(算法 2 第 7 行~第 9 行).如果这个完全图满足了移动对象聚集的定义,把它加入到候选集合 Ga' 中(算法 2 第 10 行、第 11 行).一旦点集 N 中的每一个点导出的聚集都已经加入候选集合,就验证这个候选集中的候选项是否具有相似性,删除重复和相似的项,然后返回合格的集合 Ga (算法 2 第 12 行、第 13 行).

算法 2. GR 算法.

输入:时空图 G ,聚集成员阈值 m_g ,距离阈值 δ ,请求时间 t .

输出:移动对象聚集集合 G_a .

1. $N \leftarrow \text{GetNode}(G, t)$;
2. **foreach** n in N **do**
3. $G' = \text{GetMaximalCompleteGraph}(G, n)$;
4. **foreach** $g \in G'$ **do**
5. $\text{edgeSize} \leftarrow \text{GetSize}(g)$;
6. **if** $\text{edgeSize} > m_g$ **then**
7. **foreach** edge e **do**
8. **if** $e.\text{weight} < \delta$ **then**
9. $\text{qualifiedEdgeNum}++$;
10. **if** $\text{qualifiedEdgeNum} > m_g$ **then**
11. $G_a' \leftarrow g$;
12. $G_a \leftarrow \text{VerifySimilarity}(G_a')$;
13. **return** G_a ;

算法中的 $\text{GetMaximalCompleteGraph}()$ 需要找出时空图 G 中节点 n 导出的所有最大完全图.由于找某一节点的完全图是一个 NP 难问题,需要大量的循环来遍历时空图 G ;同时,时空图包含所有的移动对象轨迹,遍历一遍需要耗费大量的时间,这会使得检索极其缓慢.为了提高查找最大完全子图过程的效率,本文又提出一种改进的 GR+算法.

2) 聚集检索算法 GR+

为了提高聚集检索过程中最大完全子图查找的效率,本文根据极大团检索 Bron-Kerbosch 算法^[23]提出一种高效的聚集检索算法.BK 算法是一种广泛使用的高效的极大团检索算法,它被认为是现有算法中较高效的一种算法^[24].根据搜索空间的特点,制定有效的剪枝策略,提出了一种改进的使用极大团检索算法和剪枝策略的算法 GR+.首先,我们给出极大团的定义.

定义 8(极大团). 给定一个无向图 $G=(N, E)$,如果 $\exists N' \in N, \exists E' \in E$,使得 $G'=(N', E')$ 是一个完全图,则称 G' 是一个团.如果 $\neg \exists n \in N$ 且 $n \in N'$,使得 $G''=\{N' \cup \{n\}, E''\}$ 是一个完全图,则称 G' 是一个极大团.

由于时空图包含所有对象的轨迹,形成了庞大的搜索空间.但是极大团检索算法需要多次遍历时空图,为了减少搜索空间,提高算法的效率,我们根据以下定理提出剪枝规则:

定理 2. 对于一个 t 时刻形成的移动对象聚集 C_g ,不可能存在一个在 t' 时刻形成的聚集,使得 $C_g \subset C'_g$, 其中, $t=t'$.

证明:假设存在两个聚集集合 $C_g=\{c_{g1}, c_{g2}, \dots, c_{gi}\}$ 和 $C'_g=\{c_{g1}, c_{g2}, \dots, c_{gi}, c_{gi+1}\}$,它们的形成时间 $t=t'$.

由于 $c_{gi+1} \notin C_g$,所以 c_{gi+1} 与 C_g 不在同一个时间片或者相邻的时间片,这与 $t=t'$ 相矛盾.所以在时刻 t 有且只有一个移动对象聚集 C_g . \square

定义 9(团的相似性). 令 G' 与 G'' 是时空图 G 中的两个极大团,那么对于给定的差异阈值 $\theta(0 < \theta \leq 1)$,如果 $\frac{|G' \cap G''|}{|G' \cup G''|} \geq \theta$, 那么 G' 与 G'' 相似.

定理 3. 对于任意一个在时空图 G 中的极大团 G' ,不存在另一个极大团 G'' ,使得 G' 与 G'' 相似,记为 $G' \sim G''$.

证明:假设存在两个极大团 $G'=\{n_1, n_2, \dots, n_i\}$ 和 $G''=\{n_1, n_2, \dots, n_j\}$,使得 $G' \sim G''$,那么存在 $n_m \in (G' \cap G'')$,使得 $n_m \in G'$ 且 $n_m \in G''$,这与定义 7 矛盾.所以,对于时空图 G 中的极大团 G' ,不存在另一个极大团 G'' ,使得 G' 与 G'' 相似.证毕. \square

在图 4 中,通过节点 A_1 和 A_5 ,都可以得到极大团 $\{A_1, A_2, A_3, A_4, A_5\}$.算法过程中过滤掉重复或者相似的极大团,

可以降低 GR 算法中验证聚集的候选集合时的搜索空间(算法 3 中第 4 行~第 11 行).

规则 1. 设一个极大团为 $G'=(N',E')$, fG' 和 bG' 分别为集合 $\{N'\}$ 中节点序号的最小值和最大值,对于候选集合中的任一极大团 $G''=(N'',E'')$,如果满足 $[fG',bG'] \cap [fG'',bG''] \neq \emptyset$,则验证 G'' 将是否与 G' 相似.

验证一个极大团是否能够形成移动对象聚集,需要检验其合格的边数是否大于给定的阈值.对于一个较大的候选集集合,就需要较长的计算时间.为了避免这个问题,我们提出以下定理:

定理 4. 设一个候选极大团 $G'=(N',E')$,其中, $|E'|$ 是极大团中的边数, $|SE'|$ 是已经搜索过的边数, $|QE'|$ 是已经搜索过的边中权值小于距离阈值的边数, m_g 是成员阈值.如果满足 $|E'| - |SE'| + |QE'| < m_g$,则该极大团不是移动对象聚集.

根据移动对象聚集的定义,定理 4 容易证明.注意到如果减少在验证候选集时所需比较的边的条数,可以减少搜索的范围,从而有规则 2.

规则 2. 设一个候选极大团的边集满足 $|E'| - |SE'| + |QE'| < m_g$,从候选集中移除该极大团.

据此,本文提出一种改进的移动聚集检索算法 GR+,如算法 3(以时间请求为例)所示.它首先根据查询请求的时间从时空图中找出一个点集(算法 3 第 2 行).将这个点集传给子过程 *ProcBK*,该过程使用 BK 算法获得点集中每一个点导出的极大团,并将该集合赋给 G' (算法 3 第 25 行~第 34 行).使用规则 1 缩小需要检验的极大团候选集合(算法 3 第 8 行),将满足规则 1 中条件的项与经过筛选的极大团进行相似性比较,如果与 G'' 中的极大团不相似,则将其加入 G'' (算法 3 第 9 行、第 10 行);不满足规则 1 中条件的项则直接加入 G'' (算法 3 第 11 行、第 12 行).对候选集合中的每一个极大团,验证其大小是否超过成员阈值 m_g ,如果合格,则验证其所包含的边的权值是否大于距离阈值(算法 3 第 13 行~第 21 行),其中,如果一个极大团满足规则 2 中的剪枝规则,则将其从候选集合中移除.如果该团满足了移动对象聚集的时空约束,则将其加入到候选集合 G_a 中(算法 3 第 22 行~第 24 行).

算法 3. GR+算法.

输入:时空图 G ,聚集成员阈值 m_g ,距离阈值 δ ,请求时间 t .

输出:移动对象聚集集合 G_a .

1. 令 G' 为候选极大团集合, G'' 为经过筛选的极大团集合
2. $N \leftarrow \text{GetNode}(G, t)$;
3. 调用子过程 *ProcBK*(\emptyset, N, \emptyset),不断调用递归函数 *ProcBK*(C, T, D)如第 19 行~第 28 行
 $G' \leftarrow \text{ProcBK}(\emptyset, N, \emptyset)$;
4. **foreach** $g' \in G'$ **do**
5. **if** $G'' = \emptyset$ **then**
6. $G'' \leftarrow g'$;
7. **foreach** $g'' \in G''$ **do**
8. **if** $[f g', b g'] \cap [f g'', b g''] \neq \emptyset$ **then**
9. **if** *VerifyCliqueSimilarity*(g', g'') = false **then** //如果 g' 与 g'' 不相似
10. $G'' \leftarrow g''$;
11. **else**
12. $G'' \leftarrow g''$;
13. **foreach** $g'' \in G''$ **do**
14. $\text{edgeSize} \leftarrow \text{GetSize}(g'')$;
15. **if** $\text{edgeSize} > m_g$ **then**
16. **foreach** $\text{edge } e$ **do**
17. **if** $e.\text{weight} < \delta$ **then**
18. $\text{qualifiedEdgeNum}++$;
19. $\text{searchedEdges}++$;

```

20.         if  $(|e|-searchedEdges+qualifiedEdgeNum)<m_g$  then
21.             break;
22.         if  $qualifiedEdgeNum>m_g$  then
23.              $Ga\leftarrow g''$ ;
24. return  $Ga$ ;
子过程:  $ProcBK(C,T,D)$ 
25. 令  $(v_p)$  为节点  $v_p$  的所有邻居节点
26. if  $T=\emptyset$  and  $D=\emptyset$  then
27. 将  $C$  输出为一个极大团;
28. return;
29. 从  $(T\cup D)$  中选择一个枢轴节点  $v_p$ ;
30.  $T'\leftarrow T\setminus N(v_p)$ ;
31. foreach  $v\in T'$  do
32.    $T\leftarrow T\setminus\{v\}$ ;
33.   call  $ProcBK(C\cup\{v\},T\cap N(v),D\cap N(v))$ ;
34.    $D\leftarrow D\cup\{v\}$ ;

```

例 3:我们使用图 4 的例子来简单地说明我们的聚集模式挖掘过程,其中,成员阈值 $m_g=3$,距离阈值 $\delta=200$.给定一个点集 $N=\{A_1,A_3,C_4\}$,对于其中每一个点,使用 BK 算法找到包含这个点的所有极大团,得到极大团集合:

$$G'=\{\{A_1,A_2,A_3,A_4,A_5\},\{A_1,A_2,A_3,A_4,A_5\},\{C_1,C_2,C_3,C_4,C_5\}\}.$$

使用规则 1 对极大团集合进行剪枝,得到 $G'=\{\{A_1,A_2,A_3,A_4,A_5\},\{C_1,C_2,C_3,C_4,C_5\}\}$.接下来验证 G' 中的极大团是否符合聚集要求的时空约束条件,所有集合的大小都大于成员阈值,但是集合 $\{A_1,A_2,A_3,A_4,A_5\}$ 中 A_3 与 A_5 的距离大于距离阈值,将 A_3 从集合中去掉,得到移动对象聚集集合 $Ga=\{\{A_1,A_2,A_4,A_5\},\{C_1,C_2,C_3,C_4,C_5\}\}$.

3) 算法性能分析

GR 算法使用原始的极大团发现算法,对于满足时间或者空间约束的 n 个节点,需要在图中遍历 2^n 个子集,在检验每一个子集时,需要检验 $n(n-1)/2$ 条边,所以极大团发现算法的时间复杂度为 $O(n^2 2^n)$.然后,需要对 m 个极大团中的每一个进行验证,看其中的 k 条边是否满足距离阈值要求,从而检验该极大团是否能够形成聚集,时间复杂度为 $O(mk)$.最后,GR 算法总时间复杂度为 $O(n^2 2^n + mk)$.

GR+算法使用具有较高速率的 BK 算法,时间复杂度为 $O(3^{n/3})^{[24]}$;同时,GR+算法使用规则 1 过滤 m 个极大团中的相似的项,使用规则 2 过滤不满足边集要求的极大团,使得验证极大团的复杂度降为 $O(\log(mk))$.最后,GR+算法总时间复杂度为 $O(3^{n/3} + \log(mk))$.

实际情况中,GR 每次查询时只需遍历时空图中的移动对象聚类节点,而无需遍历全部的轨迹数据,较 Crowd-TAD 具有较高的查询效率.同时,GR+算法又使用了高效的极大团检索算法和剪枝策略,使得检索的时间复杂度进一步优化.因此,GR+算法是一种高效的聚集检索算法.

2 实验比较与分析

为了验证移动对象聚集模式挖掘方法的准确性和高效性,我们在真实的出租车轨迹数据集下进行了实验,并与现有的移动对象聚集模式挖掘算法^[3]进行比较.

2.1 实验建立

我们使用真实世界的汽车轨迹数据^[25]来验证移动对象聚集模式挖掘方法的准确性和高效性.该数据集包含了 752MB 的汽车轨迹数据,是从北京 10 357 辆出租车采集的一周的行驶轨迹(<http://research.microsoft.com/apps/pubs/?id=152883>).图 5 是轨迹数据集的一个示例图,随机抽取 1 200 辆汽车的轨迹,画出每辆车的行驶轨迹位置形成该图.图中的黑点为汽车的行驶位置,较黑的线段是车辆通行量较大的线路,黑团为较易形成密集群组

的地段.近期与本文最相近的研究是移动对象聚集模式挖掘算法 Crowd-TAD.该算法由文献[3]提出,使用增量式算法逐步扩展移动对象序列.本文通过与 Crowd-TAD 的对比来说明本文所提出方法的准确性和高效性.实验的硬件平台是 4 核 Intel Xeon CPU(2.00GHz),16G 内存,运行 Linux 操作系统,所有的程序均使用 C++实现.



Fig.5 Trajectory dataset graph

图 5 轨迹数据集图

2.2 准确性对比

首先,我们通过计算某个地区或者某个时间段的移动对象聚集数来与之前的研究^[3]进行准确性对比.GR 与 GR+在检索的准确性方面没有差别,在此选用 GR+算法与 Crowd-TAD 算法进行准确性对比.这一组实验中默认参数的设置为:移动对象数据库 $|O_{DB}|=10357$,时间数据库 T_{DB} 划分为 $(7 \times 24 \times 360)$ 个时间点,每个时间片 $t_p=10$,移动对象聚集成员阈值 $m_g=8$,聚类的生存时间阈值 $k_c=20$,移动对象聚集成员阈值 $m_c=8$,时间阈值 $\mu=20$,距离阈值 $\delta=800$.

图 6(a)给出了单日、工作日和周末的移动对象聚集数的平均值.实验收集一周 7 天内所有的移动对象聚集数后求出一天的平均值,将 7 天的轨迹数据分为工作日和周末两类,计算出相应的平均聚集数,结果如图所示.可以看出,GR+与 Crowd-TAD 在单日、工作日、周末的移动对象聚集数基本是相同的.这说明本方法提出的移动对象聚集模式挖掘方法在准确性方面与现有的研究具有相同的准确度.我们也注意到,周末的聚集数远大于工作日的聚集数.这是因为大多数人在工作日一般都是早上上班、晚上下班,一天中多数时间都待在工作地点,所以聚集主要在这两个时间段形成.但是周末许多人都会驾车出入商场、餐厅以及其他娱乐场所,这些是较容易形成聚集的地方.所以,在工作日形成的聚集数比周末少很多.

图 6(b)是 7 天中每个小时的平均聚集数.根据实际的交通情况得知,北京市的交通在一天中的上下班高峰期经常会出现拥堵情况.如图所示,一天中的聚集数在 14:00 出现峰值,但是在 23:00~7:00 都会保持在较低水平.因为我们使用的是出租车数据,每天中午都会有较多的人打车,但是晚上出门的人则较少.我们还可以观察到,在 18:00 时也会出现一个次高的峰值,因为这个时段是下班高峰期.这个结果基本与北京市的交通情况一致.

第 3 个实验统计 6 个不同区域在单日的平均聚集数,分别是东单、天宁寺桥、中关村、国贸桥、马家楼桥和延庆,这 6 个地点包括 3 个较易发生拥堵的地区和 3 个通行相对顺畅的地区.分别获得这 6 个地区 7 天的聚集数平均值,结果如图 6(c)所示.可以发现,发生聚集数发生最多的地区是东单.主要原因是东单靠近天安门,车流量较大,很容易形成汇聚的情况;相反,延庆的聚集数较少,这是因为延庆在北京郊区,面积大,车流量少,所以不容易形成聚集.我们还可以发现:在 3 个容易发生拥堵的地区,移动对象聚集数差别不大.因为在北京的繁华地区车流量都很大,所以汽车的平均行驶速度也相对缓慢,容易发生聚集.

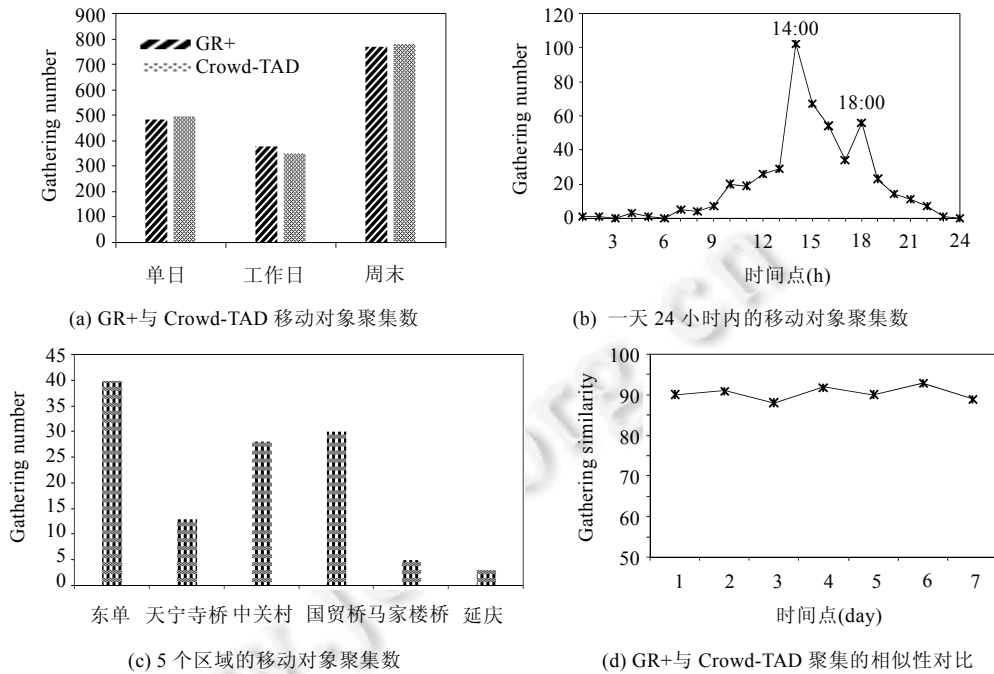


Fig.6 Average gathering number of GR+ vs. Crowd-TAD

图 6 GR+和 Crowd-TAD 平均聚集数比较

最后一个实验对 GR+算法和 Crowd-TAD 算法挖掘出的聚集的相似性进行验证,图 6(d)是 7 天中两种算法得出的聚集的平均相似值.我们提取一周 7 天中的聚集集合,根据定义 9,使用 $sim=|g_1 \cap g_2|/|g_1 \cup g_2|$ 来计算两个聚集 g_1 和 g_2 的相似性,其中 g_1 和 g_2 分别为 GR+算法和 Crowd-TAD 算法得出的同一聚集.在图 6(d)中可以发现,使用两种算法挖掘出的聚集集合具有较高的相似性.

2.3 检索效率对比

在这一组实验中,我们主要通过改变聚集成员阈值 m_g 和移动对象聚类数 $|C|$ 与已有方法进行效率对比,使用的默认参数是移动对象数据库 $|O_{DB}|=10357$,时间数据库 T_{DB} 划分为 $(7 \times 24 \times 360)$ 个时间点,每个时间片 $t_p=10$,移动对象聚类成员阈值 $m_c=8$,聚类的生存时间阈值 $k_c=20$,时间阈值 $\mu=20$,距离阈值 $\delta=800$.用 4 种算法进行效率对比:(a) Brute-force 从轨迹数据中提取包含所需时间和地点的轨迹数据并组成轨迹快照,循环遍历轨迹快照不断扩大移动对象群组,直到发现移动对象聚集;(b) Crowd-TAD 从轨迹数据中抽取出全部快照,根据时间和地点使用 Crowd-TAD 算法检索聚集数;(c) GR 算法;(d) GR+用改进的极大团检索及剪枝策略实现的 GR 算法.每一个实验均随机选取 100 个时刻和 100 个地点进行检索,重复 10 次并计算聚集检索时间的平均值.

图 7(a)所示为算法在移动聚类数 $|C|=7000$,变化聚集成员阈值 m_g 时的运行时间.从图中可以看出,当 m_g 增大时,brute-force 和 Crowd-TAD 的运行时间都有一定程度的升高,但是 GR 的运行时间基本保持不变.主要原因是,brute-force 和 Crowd-TAD 实现都需要大量的循环来遍历轨迹快照寻找群组,但是 GR 搜索依据的是时空流的结构,无论 m_g 如何变化,运行时间都只与图的结构有关.图 7(b)所示为变化聚类数 $|C|$,保持 $m_g=7$ 不变时 4 种算法的运行时间.可以明显地看出,随着 $|C|$ 的增大,4 种算法的运行时间都随着增加,但是 GR 算法的运行时间仍然远小于另外两种算法.聚类数 $|C|$ 增大,一定会导致时空图规模的上升,检索的时间也随之增大;同时,brute-force 和 Crowd-TAD 遍历的次数也会增加,但是 GR 算法仍然保持着较高的运行效率.从图中我们也可以明显看出,GR+的运行时间低于 GR.这是因为 GR+采用了高效极大团检索算法,剪枝策略也使搜索空间减小.根据这一组实验,我们可以看出 GR+算法拥有较高的检索效率.

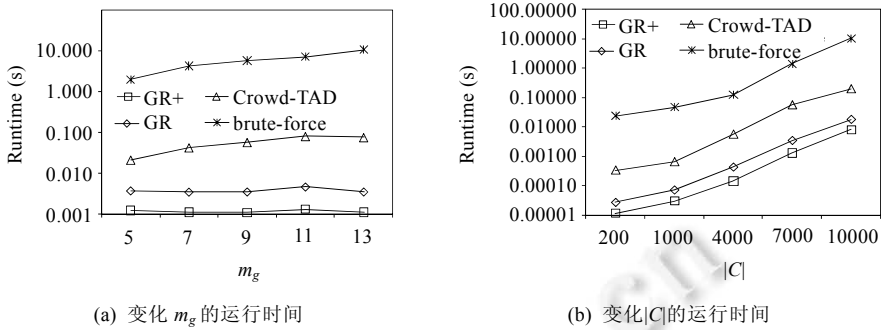


Fig.7 Running time of four gathering retrieving algorithm

图 7 4 种聚集检索算法的运行时间

图 8 分别是使用剪枝和不使用剪枝的 GR+算法.使用与上述实验一样的方法,图 8(a)固定聚类数 $|C|$ 改变阈值 m_g ,移动对象聚集的检索时间随着 m_g 的增大基本保持不变.在图 8(b)中,检索时间增长较快.从这两个图中我们可以看出,使用剪枝策略的 GR+算法的运行效率明显高于一般的 GR+算法.这是因为剪枝使搜索空间减小,加快了搜索的收敛速度.

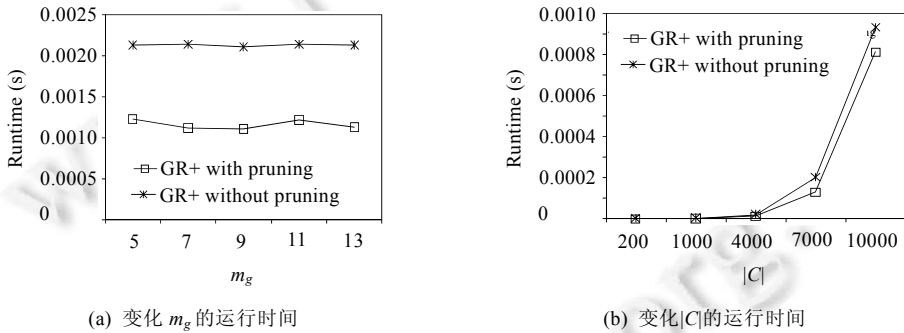


Fig.8 Running time of GR+ with pruning and without pruning

图 8 使用剪枝策略与不使用剪枝策略的 GR+算法的运行时间

图 9 给出了当一个新的移动快照加入导致聚类数 $|C|$ 增大时,GR 更新时空图的时间和 Crowd-TAD 扩展一个移动群组(crowd)的时间.

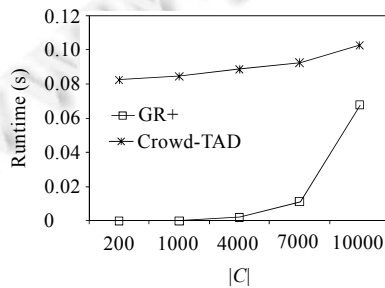


Fig.9 Updating snapshot time of GR+ vs. Crowd-TAD

图 9 GR+与 Crowd-TAD 的更新快照时间对比

如前所述,Crowd-TAD 算法只遍历给定时间的相邻轨迹快照,所以它的运行时间基本保持不变.GR 需要维

护一个时空图,所以当聚类数 $|C|$ 增大时,更新图的时间会呈指数级增长.可以明显看出,GR 的时间的增长速度大于 Crowd-TAD,但是总体时间还是小于 Crowd-TAD.为了避免维护时空图给检索带来的额外的时间开销,将时空图放在线下维护,每次获得一个新的移动快照时都传给线下系统,这样就可以不影响总体查询的效率.

3 结 论

移动对象聚集模式挖掘是移动模式挖掘中一个重要的研究问题.通过分析移动对象的聚集模式,我们可以实时观测城市的交通状况,提供路线规划服务.然而,目前关于移动对象聚集模式的研究主要关注如何发现聚集模式,挖掘特定时空约束条件下的聚集模式,需要遍历所有的轨迹数据,极大地增加了挖掘的时间开销.此外,现有的移动对象索引机制只支持单个移动对象,不能分析多个群组以及它们的移动趋势.已有的数据模型不能适应群组间强时空约束的条件,缺乏必要的灵活性,影响了聚集模式挖掘的效率.

为了解决上述问题,本文提出了一种移动对象聚集模式挖掘方法.该方法首先对 DBScan 算法进行了改进,以根据轨迹数据中的时空特性对移动对象进行聚类.然后,为了更加灵活、高效地挖掘满足时空约束条件的聚集模式,提出了一种时空图模型.该模型由移动对象聚类组成,可以实时观测移动对象聚类的变化情况.最后,该方法根据聚集的特点提出了一种改进的聚集检索算法以提高检索效率.基于北京市真实的汽车轨迹数据上的实验结果表明:与其他方法相比,本文提出的移动对象聚集模式挖掘方法具有更高的准确性和检索效率.

致谢 我们向提供出租车行驶轨迹数据集的微软亚洲研究院的郑宇博士表示感谢.

References:

- [1] Dieter P, Jensen CS. Indexing of network constrained moving objects. In: Proc. of the 11th ACM Int'l Symp. on Advances in Geographic Information Systems. New Orleans: ACM Press, 2003. 25–32. [doi: 10.1145/956676.956680]
- [2] Bart K, Othman W. Modeling uncertainty of moving objects on road networks via space–time prisms. Int'l Journal of Geographical Information Science, 2009,23(9):1095–1117. [doi: 10.1080/13658810802097485]
- [3] Zheng K, Zheng Y, Yuan NJ, Shang S. On discovery of gathering patterns from trajectories. In: Proc. of the 29th IEEE Int'l Conf. on Data Engineering. Brisbane: IEEE, 2013. 242–253. [doi: 10.1109/ICDE.2013.6544829]
- [4] Patrick L, Imfeld S. Analyzing relative motion within groups of trackable moving point objects. In: Proc. of the 2nd Int'l Conf. on Geographic Information Science. Boulder: Springer-Verlag, 2002. 132–144. [doi: 10.1007/3-540-45799-2_10]
- [5] Gudmundsson J, van Kreveld M. Computing longest duration flocks intrajjectory data. In: Proc. of the 18th ACM Int'l Symp. on Advances in Geographic Information Systems. Arlington: ACM Press, 2006. 35–42. [doi: 10.1145/1183471.1183479]
- [6] Li ZH, Ding BL, Han JW, Kays R. Swarm: Mining relaxed temporal moving object clusters. Proc. of the VLDB Endowment, 2010, 3(1-2):723–734. [doi: 10.14778/1920841.1920843]
- [7] Tang LA, Zheng Y, Yuan J, Han JW, Leung A, Hung CC, Peng WC. On discovery of traveling companions from streaming trajectories. In: Proc. of the 28th IEEE Int'l Conf. on Data Engineering. Washington: IEEE, 2012. 186–197. [doi: 10.1109/ICDE.2012.33]
- [8] Liu W, Zheng Y, Chawla S, Yuan J, Xie X. Discovering spatio-temporal causal interactions in traffic data streams. In: Proc. of the 17th Int'l Conf. on Knowledge Discovery and Data Mining. San Diego: ACM Press, 2011. 1010–1018. [doi: 10.1145/2020408.2020571]
- [9] Pang LXL, Chawla S, Liu W, Zheng Y. On mining anomalous patterns in road traffic streams. In: Proc. of the 7th Int'l Conf. on Advanced Data Mining and Applications. Beijing: Springer-Verlag, 2011. 237–251. [doi: 10.1007/978-3-642-25856-5_18]
- [10] Pang LX, Chawla S, Liu W, Zheng Y. On detection of emerging anomalous traffic patterns using GPS data. Data & Knowledge Engineering, 2013,87:357–373. [doi: 10.1016/j.datak.2013.05.002]
- [11] Chawla S, Zheng Y, Hu JF. Inferring the root cause in road traffic anomalies. In: Proc. of the 12th IEEE Int'l Conf. on Data Mining. Brussels: IEEE, 2012. 141–150. [doi: 10.1109/ICDM.2012.104]
- [12] Yu YW, Wang Q, Wang XD, Wang H, He J. Online clustering for trajectory data stream of moving objects. Computer Science and Information Systems, 2013,10(3):1293–1317. [doi: 10.2298/CSIS120723049Y]
- [13] Shi CL, Gan WY, Wu L, Zhang MJ, Tang YB. Clustering trajectories of entities in computer wargames. Ruan Jian Xue Bao/Journal of Software, 2013,24(3):465–475 (in Chinese with English abstract). <http://www.jos.org.cn/1000-9825/4248.htm> [doi: 10.3724/SP.J.1001.2013.04248]

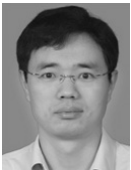
- [14] Jeung H, Yiu ML, Jensen CS. Trajectory pattern mining. In: Proc. of the 13th Int'l Conf. on Knowledge Discovery and Data Mining. San Jose: ACM Press, 2007. 330–339. [doi: 10.1145/1281192.1281230]
- [15] Šaltenis S, Jensen CS, Leutenegger ST, Lopez MA. Indexing the positions of continuously moving objects. In: Proc. of the 2000 Int'l Conf. on Management of Data. Dallas: ACM Press, 2000. 331–342. [doi: 10.1145/342009.335427]
- [16] Demiryurek U, Pan B, Banaei-Kashani F, Shahabi C. Towards modeling the traffic data on road networks. In: Proc. of the 2nd Workshop on Computational Transportation Science. Seattle: ACM Press, 2009. 13–18. [doi: 10.1145/1645373.1645376]
- [17] Hartmut G, de Almeida T, Ding ZM. Modeling and querying moving objects in networks. Int'l Journal on Very Large Data Bases, 2006,15(2):165–190. [doi: 10.1007/s00778-005-0152-x]
- [18] de Almeida VT, Güting RH. Indexing the trajectories of moving objects in networks. GeoInformatica, 2005,9(1):33–60. [doi: 10.1007/s10707-004-5621-7]
- [19] Ding ZM, Li XN, Yu B. Indexing the historical, current, and future locations of network-constrained moving objects. Ruan Jian Xue Bao/Journal of Software, 2009,20(12):3193–3204 (in Chinese with English abstract). <http://www.jos.org.cn/1000-9825/3400.htm> [doi: 10.3724/SP.J.1001.2009.03400]
- [20] Del Mondo G, Stell JG, Claramunt C, Thibaud R. A graph model for spatio-temporal evolution. Journal of Universal Computer Science, 2010,16(11):1452–1477.
- [21] Del Mondo G, Rodríguez MA, Claramunt C, Bravo L, Thibaud R. Modeling consistency of spatio-temporal graphs. Data & Knowledge Engineering, 2013,84:59–80. [doi: 10.1016/j.datak.2012.12.007]
- [22] Ester M, Kriegl HP, Jörg S, Xu XW. A density-based algorithm for discovering clusters in large spatial databases with noise. In: Proc. of the 2nd Int'l Conf. on Knowledge Discovery and Data Mining. Portland: ACM Press, 1996. 226–231.
- [23] Bron C, Kerbosch J. Algorithm 457: Finding all cliques of an undirected graph. Communications of the ACM, 1973,16(9):575–577. [doi: 10.1145/362342.362367]
- [24] Cazals F, Karande C. A note on the problem of reporting maximal cliques. Theoretical Computer Science, 2008,407(1-3):564–568.
- [25] Yuan J, Zheng Y, Zhang CY, Xie WL, Xie X, Huang Y. T-Drive: Driving directions based on taxi trajectories. In: Proc. of the 18th ACM Int'l Symp. on Advances in Geographic Information Systems. San Jose: ACM Press, 2010. 99–108.

附中参考文献:

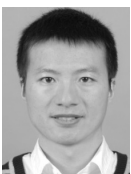
- [13] 石崇林, 谿文燕, 吴琳, 张茂军, 唐宇波. 计算机兵棋作战实体轨迹聚类算法. 软件学报, 2013, 24(3): 465–475. <http://www.jos.org.cn/1000-9825/4248.htm> [doi: 10.3724/SP.J.1001.2013.04248]
- [19] 丁志明, 李肖南, 余波. 网络受限移动对象过去、现在及将来位置的索引. 软件学报, 2009, 20(12): 3193–3204. <http://www.jos.org.cn/1000-9825/3400.htm> [doi: 10.3724/SP.J.1001.2009.03400]



张峻铭(1987—),男,河南郑州人,博士生,主要研究领域为轨迹数据计算,时空数据挖掘。



李静林(1975—),男,博士,副教授,CCF 专业会员,主要研究领域为融合网络,服务支撑环境。



王尚广(1982—),男,博士,副教授,CCF 高级会员,主要研究领域为服务计算,网络安全。



刘志喆(1979—),男,讲师,CCF 专业会员,主要研究领域为融合网络服务支撑环境关键技术。



袁泉(1988—),男,博士生,CCF 学生会员,主要研究领域为车载网络。



杨放春(1957—),男,博士,教授,博士生导师,CCF 杰出会员,主要研究领域为服务计算,网络智能化,网络安全,通信软件。