

# 基于时空轨迹的热点区域提取

王侃<sup>1\*</sup>, 梅克进<sup>2</sup>, 朱家辉<sup>2</sup>, 牛新征<sup>2</sup>

(1. 中国电子科技集团公司第十研究所 成都 610000; 2. 电子科技大学计算机科学与工程学院 成都 611731)

**【摘要】** 轨迹聚类算法可以广泛地应用在交通管理中, 利用轨迹聚类算法找出车辆轨迹热点区域对交通部门规划管理交通出行有重要指导意义。目前的轨迹聚类算法多以空间相似性进行度量, 不能体现不同时间段的轨迹热点区域划分情况。针对上述问题, 该文结合时间因素, 提出了一种时空轨迹的热点区域提取算法。首先, 对传统的密度峰值聚类算法进行了改进, 考虑了计算密度的线性和非线性部分, 改进了密度的计算方法; 同时, 改进了簇类中心的选取方法, 能够自动地选取簇类中心; 在此基础上, 提出了聚类融合算法, 过滤了不合适的聚类和多余聚类; 最后利用DB检验量来检测提取效果。实验结果表明, 相比于传统的聚类算法, 本文算法能更有效地提取时空轨迹的热点区域。

**关键词** 聚类; 密度; 热点区域; 时空轨迹

中图分类号 TP301.6

文献标志码 A

doi:10.3969/j.issn.1001-0548.2019.06.018

## Hotspots Extraction Based on Spatial-Temporal Trajectory Data

WANG Kan<sup>1\*</sup>, MEI Ke-jin<sup>2</sup>, ZHU Jia-hui<sup>2</sup>, and NIU Xin-zheng<sup>2</sup>

(1. The 10th Research Institute of China Electronics Technology Group Corporation Chengdu 610000;

2. School of Computer Science and Engineering, University of Electronic Science and Technology of China Chengdu 611731)

**Abstract** Trajectory clustering algorithm can be widely used in traffic management. Finding the vehicle trajectory hotspots by using trajectory clustering algorithm has important guiding significance for traffic planning and management of traffic travel. Current trajectory clustering algorithms are mostly measured by spatial similarity, which cannot reflect the division of trajectory hotspots in different time periods. In response to the above problems, this paper proposes a hotspot region extraction algorithm for spatio-temporal trajectory, combined with the factor of time. Firstly, the traditional density peak clustering algorithm and the density calculation method are improved by considering the linear and nonlinear parts of the calculated density. At the same time, the method of choosing cluster center is modified to enable it to automatically select the cluster center. On the basis of the above, we propose a clustering fusion algorithm to filter inappropriate clusters and redundant clusters and use the DB index to detect the division results. The experimental results show that our algorithm can extract the hot spots of spatio-temporal trajectories more effectively than the traditional clustering algorithms.

**Key words** clustering; density; hotspots; spatio-temporal trajectory

近年来, 交通路网不断扩张, 交通出行压力逐渐增加, 尤其在上下班高峰期间, 经常出现大面积拥堵的现象。对车辆轨迹进行分析, 找出车辆的运行规律, 有助于交通运营部门协调交通运输。随着配备了GPS传感器和存储硬件的WiFi嵌入式移动设备的大量使用, 每天都会产生大量的存储着移动对象位置及时间戳的GPS轨迹数据。如何从大量的轨迹数据中挖掘出有用的价值是一个重要问题。轨迹聚类是分析GPS数据特征的主要方法之一, 路网上的轨迹可以划分为许多不同的簇, 通过对轨迹内部运动模式和特征信息的分析, 确定轨迹间的相似度,

将相似度较高的轨迹归入一簇, 通过对轨迹簇的分析找出时空轨迹热点区域, 从而挖掘出轨迹中的潜在规律, 进而解决复杂路网中存在的实际问题。时空轨迹热点区域的主要特点即是在一段时间内存在许多空间相似性高的轨迹簇, 因此在时空轨迹热点区域的提取问题中经常用到轨迹聚类的相关算法。目前的轨迹聚类算法多基于传统的空间聚类方法, 其不足之处在于经常将某一时间段内没有轨迹或只有少量轨迹的区域识别为轨迹热点区域。对此, 本文提出了一种基于聚类的时空轨迹热点区域提取算法, 在传统的空间聚类方法的基础上, 考虑了时间

收稿日期: 2018-11-24; 修回日期: 2019-01-19

基金项目: 四川省科技计划(2017FZ0094); 部级基金

作者简介: 王侃(1986-), 男, 博士, 高级工程师, 主要从事信号、开源以及地理空间情报方面的研究. E-mail: 306616278@qq.com

维度的特性,能够提取不同时间段的轨迹热点区域。

## 1 相关工作

聚类分析是数据挖掘用于知识发现的一个常用手段。它的目标是将数据集划分为一些组,这些组通常被称为簇,同一个簇中的数据点之间的相似度高要高于其他簇中的数据点。目前,轨迹聚类方法多采用传统的空间聚类方法<sup>[1-3]</sup>,如K-means通过不断更新簇类中心,将数据点聚类到相近的簇中,达到划分簇类的目的。然而它存在3个问题:1) 必须用户事先指定划分簇类数量 $k$ ,存在较大的主观因素;2) 初始簇心的选择差异,直接影响最终的聚类结果;3) 不能识别非球形的簇,存在着局限性。而基于密度聚类算法<sup>[4-6]</sup>能识别任意形状的簇,但密度阈值的设定比较关键,对于不同的数据集,密度阈值调整十分困难。对此,文献[7]提出了选取数据点距离升序排序的前2%作为阈值来对高密度点进行选取,然而它的参数并不具有适应性,对不同的数据的划分效果差异较大。此外,部分研究通过数据预处理,消除冗余数据的影响;改变提取轨迹中特征点的方法,将对数据点的提取转变为提取轨迹段,从而更好地保留数据的结构特征;通过改变距离度量,更好地度量数据点之间的相似度,找出相似度较高的数据点(轨迹段)。如文献[8]对真实场景目标轨迹的有效性判断与聚类进行研究,通过轨迹的预处理,消除噪声影响。同时基于轨迹上点坐标值的方差、目标相邻两帧的运动方向以及角度、轨迹的长度3种信息来确定轨迹的有效性,当确认轨迹有效后,采取K-means算法将空间距离相近的轨迹聚到一个聚类中。文献[9]基于传统的空间聚类算法中选取轨迹采样点的局限性,提出了基于轨迹结构相似性的聚类算法,其通过计算轨迹的每个拐角,利用拐角阈值将轨迹划分为轨迹段,对目标轨迹进行结构化处理,计算结构化相似度,将具有相似度较高的轨迹段聚成一个类组。该方法从全局角度把握轨迹的特性和运动趋势,提高了轨迹聚类的精度。但上述方法并没有解决在实际交通中轨迹出现在不同时间段的问题。

为了解决上述问题,相关工作者在研究轨迹聚类问题中加入了时间度量。文献[10]定义了时间和空间距离,结合时间和空间邻域形成时空邻域,计算空间和时间阈值,在时空领域内找到满足空间和时间阈值长度的轨迹,将它们作为聚类核心,通过扩展其领域内直接密度可达的线段,得到最终的聚类

结果。文献[11]通过对时间进行预处理,解决了时间和空间度量不同的问题,通过计算找出具有时空相似性的轨迹,将它们聚成一个聚类。上述方法通过重新定义距离等方式,利用密度峰值聚类算法找出轨迹的时空相似性,得到轨迹的热点区域,但在高密度点的选取上偏向人为因素,有时会存在较大误差。因此,本文提出一种基于时空轨迹的热点区域提取算法,首先对传统的密度峰值聚类算法<sup>[7,12]</sup>做了两个改进:1) 改进密度的计算方法,采用加权密度计算的方法,考虑线性和非线性因素,使得密度点的区分更加明显;2) 改进了簇类中心的选取方法,将轨迹点的密度值和与更高密度点之间的距离值相乘,选取比拐点大的值作为簇类中心进行聚类。最后采用聚类融合算法,过滤距离较近的聚类或者多余聚类,从而划分出一系列的热点区域。

## 2 问题描述

为了便于描述问题,给出热点区域的定义:

在时空轨迹数据集 $r_i$ 中, $r_i.S$ 和 $r_i.E$ 分别表示轨迹出现的开始和结束时刻,在 $[r_i.S, r_i.E]$ 中具有较高时空相似性的时空轨迹簇称为热点区域,并且属于同一个热点区域的样本点较其与其他热点区域的样本点具有更高的时空相似性。

考虑将一个时空轨迹数据集 $T = \{TR_1, TR_2, \dots, TR_{u_{ajs}}\}$ 划分为一系列时空轨迹簇,其中 $TR_i = \langle p_{i,1}, p_{i,2}, \dots, p_{i,n} \rangle$ , $p_{i,j} = (l_{i,j}, t_{i,j})$ 表示在 $t_{i,j}$ 时车辆的位置。时空轨迹簇的质量通常通过簇间分离和簇内同质性来评估。通常,较大的簇间分离和较高程度的簇内同质性表明更准确的聚类。本文采用广泛用于集群验证的Davies-bouldin(DB)索引来衡量道路网络中的聚类质量。

首先计算类内数据到簇质心的平均距离 $C_i$ ,其代表簇类 $i$ 中各点的分散程度:

$$\bar{C}_i = \sqrt{\frac{1}{T_i} \sum_{j=1}^{T_i} \|X_j - A_i\|} \quad (1)$$

式中, $X_j$ 代表簇类 $i$ 中第 $j$ 个数据点; $A_i$ 为簇类 $i$ 的质心; $T_i$ 是簇类 $i$ 中数据的个数; $\|M - N\|$ 代表 $M$ 到 $N$ 的欧式距离。定义DB指标:

$$DB = \frac{1}{k} \sum_{i=1}^k \max_{j \neq i} \left( \frac{\bar{C}_i + \bar{C}_j}{\|A_i - A_j\|} \right) \quad (2)$$

问题定义:给定一组时空轨迹 $T = \{TR_1, TR_2, \dots, TR_{u_{ajs}}\}$ ,希望提取到使DB指标达到最小的多个轨迹热点区域 $H = \{H_1, H_2, \dots, H_H\}$ 。

在现实生活中, 车辆的轨迹随时都会产生。不同时段的轨迹反映了不同的交通状况。例如, 工作日产生的轨迹通常聚集在大型企业、公司和学校周围, 而节假日的轨迹通常遍布整个商场、娱乐场所等。传统空间轨迹聚类考虑了所有历史轨迹但忽略了轨迹的时间维度, 这在一定程度上影响了轨迹分割的准确性, 因为有时它将一段时间内没有轨迹的区域视为轨迹密集区域。如图1所示, 轨迹密集区域存在于  $t_1$ 、 $t_2$  和  $t_3$ 、 $t_4$  之间, 其他时段都不存在轨迹数据, 传统的基于空间的交通分区划分方法难以识别出这种分布特性, 而结合了空间和时间的聚类方法能有效地提取出时空中的热点区域。

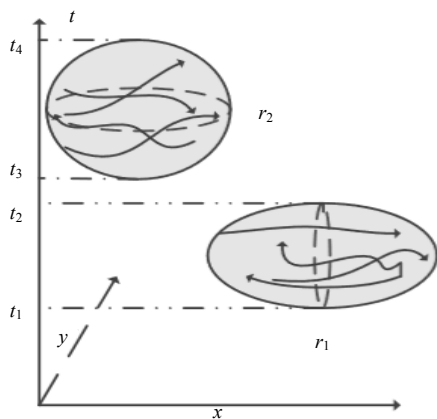


图1 热点区域

### 3 基于时空轨迹的热点区域提取算法

本文提出了基于时空轨迹的热点区域提取算法 (TSTHE), 有效且准确地将车辆的轨迹划分为时空区域中的一系列热点区域。其主要思想概括如下:

首先, 通过z-score在每个维度上对所有轨迹点进行归一化, 以消除由尺寸差异引起的误差。然后, 根据改进的密度峰值聚类提取出一系列时空区域。最后, 对时空区域进行聚类融合, 以将整个时空区域分成一系列热点区域。

识别被低密度所围绕的高密度区域是基于密度的聚类方法的核心思想。它基于以下两种假设:

- 1) 聚类中心周围邻居点的密度不高于其密度;
- 2) 聚类中心相对于更高密度点之间的间隔较大;

通过计算出每个样本点的局部密度  $\rho_i$  和与更高密度点之间的距离  $\delta_i$ , 找到那些局部密度较大且与更高密度点之间的距离较大的点, 将它们定义为聚类中心。传统的密度峰值方法对距离的计算比较简单, 但轨迹点之间的相关性并不全靠线性距离来关联。如一个数据点与另一个数据点之间的影响为1,

那么它与另一个距离为两倍的点之间的影响不一定为1/2, 其关系往往是非线性的。与此同时, 选取轨迹真实密度的截断阈值很难估计, 不同的截断阈值对数据点的密度判断会造成很大影响。对此, 本文考虑距离度量中存在线性和非线性关系。通过给两种关系赋予不同的权值, 以区分不同距离度量的影响效果。同时将数据连续化, 用轨迹点间的最大距离代替截断阈值, 解决了采用截断阈值对实际有差异的轨迹点的密度判断相同的问题。本文提出的两种密度计算方法如下:

1) 轨迹点间的直接距离可以通过线性表达式计算出来, 考虑到距离度量中的线性因素, 对任何轨迹点  $p_i$ , 密度  $\rho_{i,1}$  被定义为:

$$\rho_{i,1} = \sum_j^{\text{num}} (d_{\max} - \|p_i - p_j\|) \quad (3)$$

2) 轨迹点间的距离关系仅仅靠线性因素来判断不够准确, 通常较远的点相互之间的影响较小, 考虑到距离度量中的非线性因素, 对于任何轨迹点  $p_i$ , 密度  $\rho_{i,2}$  被定义为:

$$\rho_{i,2} = \sum_j^{\text{num}} e^{-\left(\frac{\|p_i - p_j\|}{d_{\max}}\right)^2} \quad (4)$$

式中,  $d_{\max}$  表示所有轨迹点之间的最大时间距离; num是所有轨迹点的数量;  $\|p_i - p_j\|$  表示  $p_i$  和  $p_j$  的欧式距离:

$$\begin{cases} \rho'_{i,1} = \frac{\max_{i=1}^{\text{num}} \rho_{i,1} - \rho_{i,1}}{\max_{i=1}^{\text{num}} \rho_{i,1} - \min_{i=1}^{\text{num}} \rho_{i,1}} \\ \rho'_{i,2} = \frac{\max_{i=1}^{\text{num}} \rho_{i,2} - \rho_{i,2}}{\max_{i=1}^{\text{num}} \rho_{i,2} - \min_{i=1}^{\text{num}} \rho_{i,2}} \end{cases} \quad (5)$$

对两种密度进行加权结合得到该点的密度:

$$\rho_i = \omega \rho'_{i,1} + (1 - \omega) \rho'_{i,2} \quad (6)$$

本文默认  $\omega$  为0.5,  $\sigma_i$  定义为:

$$\sigma_i = \begin{cases} \min_{j: \rho_j > \rho_i} \|p_i - p_j\| & \exists \rho_j > \rho_i \\ \max_{j=1,2,\dots,\text{num}} \|p_i - p_j\| & \text{其他} \end{cases} \quad (7)$$

传统的密度峰值聚类算法是将同时具有较高密度  $\rho_i$  和更远距离  $\sigma_i$  的点识别为聚类中心<sup>[7]</sup>, 但对  $\rho_i$  和  $\sigma_i$  的选取一般是人为主观判断, 划分的合理性难以解释。对此, 本文采用手肘法和箱型法相结合的方式来确定初步聚类数量和聚类中心。定义每个样本点  $p_i$  的密度特征  $\alpha_i$  为:

$$\alpha_i = \rho_i \sigma_i \quad (8)$$

$\alpha_i$  越高, 被视为聚类中心的几率越大。将  $\alpha_i$  进行降序排序, 如图2所示。其中横轴代表某个轨迹点

$p_i$ , 纵轴代表轨迹点  $p_i$  对应的  $\alpha_i$  值, 将拐点取为肘部, 选取拐点左侧的  $\alpha_i$  加入集合  $S_1$  中, 则  $S_1$  中的点对应的  $p_i$  可作为候选的簇类中心, 但其中包含点的数量较多, 对簇的划分过于细致, 并且还有一部分区分度不够明显的点, 如该点的密度大但距离小, 或者该点的距离大但密度小, 那么它们的乘积  $\alpha_i$  也会较高, 则该点也有可能被视为簇类中心。因此, 结合箱型法对  $S_1$  集合中的候选簇类中心进行初步过滤。首先取所有轨迹点  $p_i$  对应的  $\alpha_i$  放入集合  $I$  中, 然后找出集合  $I$  中的四分位数  $Q_i = i(n+1)$ , 其中  $n$  为轨迹集合  $T$  中元素的个数, 然后计算上四分位数 ( $Q_3$ ) 和下四分位数 ( $Q_1$ ), 并计算出四分位距  $IQR = Q_3 - Q_1$ , 定义异常值截断阈值  $d_{Th}$  为:

$$d_{Th} = Q_3 + 1.5IQR \quad (9)$$

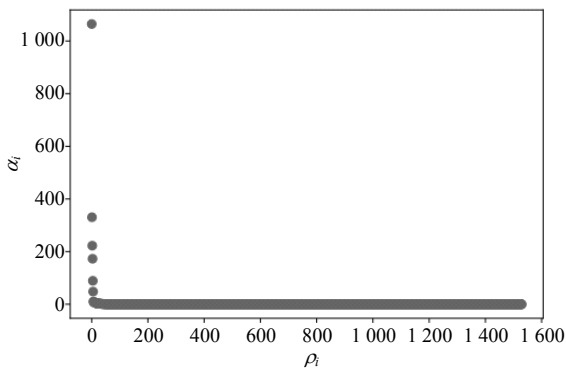


图2  $\rho_i$  与  $\sigma_i$  排序值的序列图

当  $\alpha_i$  超过截断阈值  $d_{Th}$  时, 该点为上分位异常值, 上分位异常值的大小远远超出了集合中其他元素的大小。在本文中,  $\alpha_i$  值越大, 它对应的轨迹点  $p_i$  被视为簇类中心的概率就越高。因此, 需要在集合  $I$  中找到  $\alpha_i$  值超过  $d_{Th}$  的点, 将这些点对应的轨迹点  $p_i$  放入  $S_2$  集合中, 对  $S_1$  和  $S_2$  集合取交集得到  $S$  集合, 则  $S$  集合中元素的  $\alpha_i$  都较大, 它们对应的轨迹点  $p_i$  可以被视为初步的簇类中心。然后, 将每个剩余的点分配到与它最近的高密度邻居相同的簇类中, 形成初步的轨迹时空区域。

本文算法并没有对簇类的划分给出限制, 可能出现初始簇类中心点的数量大于实际簇类数量的情况, 导致过度聚类, 因此需要对初始簇类进行融合。本文对簇类的划分指定了一个融合阈值  $d_c$ , 并根据簇类质心的距离与融合阈值之间的关系判断是否融合: 当簇类间质心距离小于该阈值时, 将这两个簇类融合成一个簇类, 从而划分出符合要求的簇类。聚类融合如图3所示。

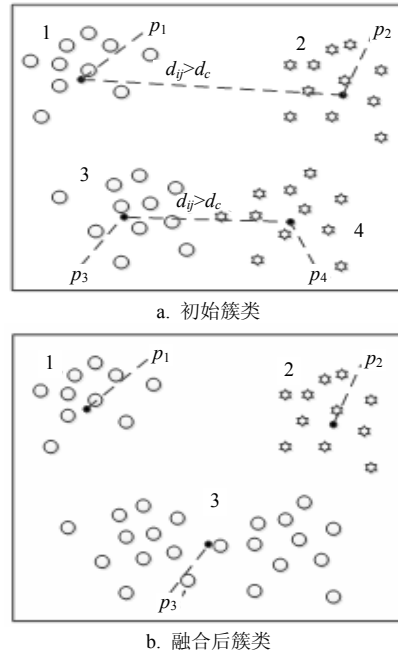


图3 聚类融合

在图3a中, 初始划分簇为4类, 对簇类进行编号。其中簇类1和簇类2的质心距离超过融合阈值  $d_c$ , 两簇不融合, 而簇类3和簇类4的质心距离小  $d_c$ , 两簇融合成一个簇类3, 同时簇心更新为  $p'_3$  点, 如图3b所示。此时, 继续判断剩余簇类是否满足融合条件, 若满足, 按照初始方式重新编号并重复上述融合过程; 若不满足, 则剩余的簇类即为最终的聚类结果。在此方法中阈值的选择不同, 可以得到不同相似程度时空热点区域。由文献[7]的假设可知, 邻居的平均数量约为数据集中总点数的1%~2%, 基于这种假设, 本文将融合阈值  $d_c$  设置为任意两个样本距离集合的升序排序的2%。

TSTHE算法的伪代码如下。其中6)~11)行用于计算轨迹点  $p_i$  的密度  $\rho_i$  和与更高密度点的距离  $\sigma_i$ , 12)~18)行利用手肘法和箱型图找出初步的轨迹聚类中心, 19)行将每个剩余的点分配到与它最近的高密度邻居相同的簇类中, 形成初步的轨迹时空区域, 20)~22)行筛去多余的簇类, 提取出轨迹的热点区域。

TSTHE算法

输入: 时空轨迹  $T = \{TR_1, TR_2, \dots, TR_{trajs}\}$ ;

输出: 热点区域集  $H = \{H_0, H_1, \dots, H_{area}\}$ ;

- 1) 化相似矩阵  $distList = \phi$
- 2)  $distList[i][j] \leftarrow$  计算任意两点  $p_i, p_j$  之间距离  $\|p_i, p_j\|$
- 3)  $d_{max} \leftarrow$  找出最大距离  $distList$
- 4) for  $p_i$  in  $T$  do

- 5)  $\rho_{i,1} \leftarrow \sum_j^{num} (d_{max} - \text{distList}[i][j])$
- 6)  $\rho_{i,2} \leftarrow \sum_j e^{-\frac{\text{dist}_T(p_i, p_j)^2}{d_{max}}}$
- 7)  $\rho'_{i,1}, \rho'_{i,2} \leftarrow$  由式(5)标准化密度
- 8) 密度  $\rho_i = \omega\rho'_{i,1} + (1-\omega)\rho'_{i,2}$
- 9) 计算  $\sigma_i$ //计算  $p_i$  点与更高密度点的距离
- 10)  $\alpha_i = \rho_i\sigma_i$
- 11)  $S_1 \leftarrow p_i$ //将拐点左侧的轨迹点加入集合  $S_1$
- 12) 异常值截断阈值  $d_{Th} = Q_3 + 1.5IQR$
- 13)  $S_2 \leftarrow p_i$ //将大于阈值的轨迹点加入集合  $S_2$
- 14) 初始簇类中心集合  $C \leftarrow S_1 \cap S_2$
- 15)  $C \leftarrow p_i$ //剩余的点  $p_i$  被分配到与它最近的高密度邻居相同的簇类中
- 16)  $H \leftarrow$  将质心距离小于距离阈值  $d_c$  的簇类融合
- 17) return  $H$

### 4 实验评估

在本节中, 所有算法都在Python2.7中实现, 所有实验都在配备IntelCorei5CPU@2.3 GHz和8 GB内存的macOS工作站上进行。实验数据集包含来自50辆不同卡车的1 100条轨迹, 这些数据都提取于希腊雅典周围, 其中有94 098个位置记录, 包括卡车标识符, 日期和时间以及地理坐标信息。数据集可在<http://chorochronos.datastories.org/?q=node/10>上公开获取。

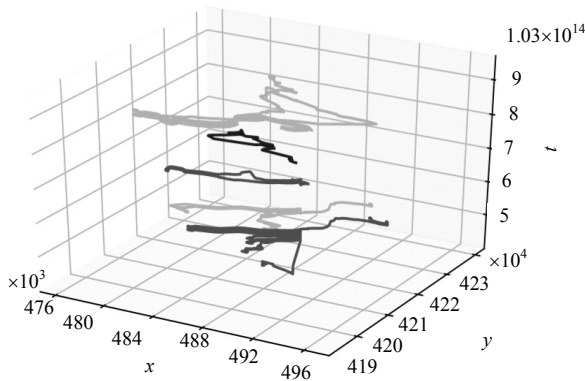


图4 TSTHE算法的热点区域提取结果

用基于时空轨迹的热点区域提取算法对20条轨迹进行热点区域提取, 如图4所示, 在一定区域内的轨迹基于时间维度的特性, 被划分成为不同的热点区域。为验证热点区域的有效性, 将传统的聚类算法K-means、DBSCAN运用于时空轨迹聚类中, 并与本文提出的基于时空轨迹的热点区域提取算法(TSTHE)对热点区域进行提取, 所获得热点区域的

质量由DB指标权衡, DB指标与热点区域的质量成反比。图5展示了3种算法随轨迹数量变化所提取热点区域的DB值。

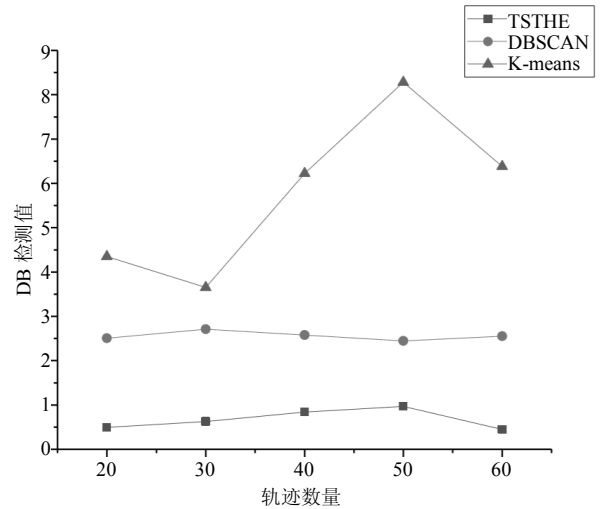


图5 3种算法的结果对比

从图5中可以看出, TSTHE算法的DB指标比DBSCAN和K-means算法都要小, 这是由于TSTHE算法通过线性密度计算和非线性密度计算加权得到轨迹点的真实密度, 更加准确的度量了轨迹点的时空相似性。同时对高密度点的选取进行了优化, 使得初始簇心的选取更加合理。并且对初步的聚类进行了过滤融合, 筛去了划分不合理的簇类, 提取效果与其它算法相比有很大提升, 同时在处理不同数量的轨迹时同样表现出了良好的适应性。

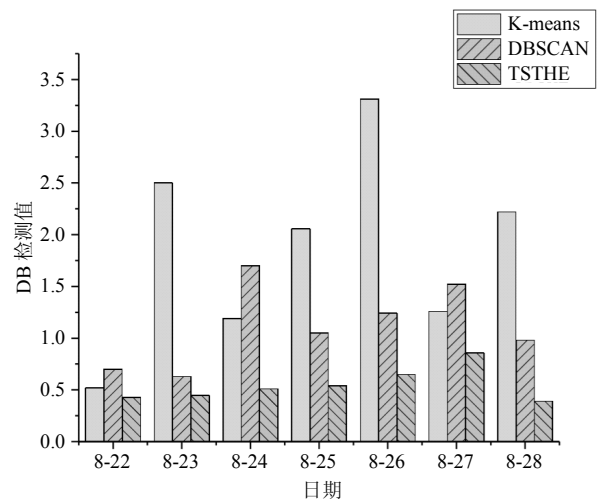


图6 3种算法的结果对比

进一步地, 为了避免不同时段的车辆轨迹带来的影响, 随机地选取了数据集中连续7天的数据进行试验, DB指标如图6所示, 从上到下3条柱状图分别代表DBSCAN、K-means、TSTHE算法的DB检测值。从图中可以看出, 对不同时段的轨迹提取热点区域,

K-means算法和DBSCAN算法的DB指标难以区分优劣,但TSTHE的DB指标要明显小于其它两种算法,说明本文提出基于时空轨迹的热点提取算法在不同时间段的轨迹数据上均具有有效性。

## 5 结束语

本文提出的基于时空轨迹的热点区域提取算法,采用密度加权方式改进了传统的轨迹算法中相似性度量单一的问题,同时利用聚类融合方法将轨迹集自动划分成恰当的簇类,解决了聚类簇数难以确定的问题。实验结果表明,与现有算法相比,本文的划分效果有了较大提升。通过对车辆轨迹数据进行聚类分析,找出具有时空相似性的轨迹区域,进而将轨迹区域划分为一系列热点区域,对合理分配警力,缓解交通高峰期的压力有重要指导意义。

### 参 考 文 献

- [1] JAMES M Q. Some methods for classification and analysis of multivariate observations[C]//Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability. Oakland, USA: [s.n.], 1967, 1: 281-297.
- [2] HUANG Zhe-xue, NG M K. A fuzzy kmodes algorithm for clustering categorical data[J]. IEEE Transactions on Fuzzy Systems, 1999, 7(4): 446-452.
- [3] WANG Bo-wen, YIN Jun, HUA Qi, et al. Parallelizing  $k$ -means-based clustering on spark[C]//2016 International Conference on In Advanced Cloud and Big Data (CBD). [S.l.]: IEEE, 2016: 31-36.
- [4] MARTIN E, KRIEGEL H P, SANDER J, et al. Density-based spatial clustering of applications with noise[C]//International Conference Knowledge Discovery and Data Mining. 1996, 240: 226-231.
- [5] MIHAEL A, BREUNIG MM, KRIEGEL H P. Optics: Ordering points to identify the clustering structure[C]//Proceedings ACM SIGMOD International Conference on Management of Data. Philadelphia, Pennsylvania, USA: ACM, 1999, 28: 49-60.
- [6] KAI M T, MARK J C. Density-ratio based clustering for discovering clusters with varying densities[J]. Pattern Recognition, 2016, 60: 983-997.
- [7] ALEX R, ALESSANDRO L. Machine learning clustering by fast search and find of density peaks[J]. Science, 2014, 344(6191): 1492.
- [8] PAN Qi-ming, CHENG Yong-mei, TAO Yang, et al. Automatic validating and clustering method for trajectories of moving objects in real scene[J]. Application Research of Computers, 2007, 24(4): 158-154.
- [9] YUAN Guan, XIA Shi-xiong, ZHANG Lei, et al. Trajectory clustering algorithm based on structural similarity[J]. Journal on Communications, 2011, 32(9): 103-110.
- [10] DI W U, DU Y Y, YI J W, et al. Density-based spatiotemporal clustering analysis of trajectories[J]. Journal of Geo-Information Science, 2015, 17(10): 1162-1172.
- [11] SHI Lu-kui, ZHANG Yan-ru, ZHANG Xin. Trajectory data clustering algorithm based on spatio-temporal pattern[J]. Journal of Computer Applications, 2017, 37(3): 854-859.
- [12] WANG Shu-liang, WANG Da-kui, LI Cao-yuan, et al. Clustering by fast search and find of density peaks with data field[J]. Chinese Journal of Electronics, 2016, 25(3): 397-402.

编辑 叶芳