



## 基于密度核心的出租车载客轨迹聚类算法

田智慧<sup>1,2</sup>, 马占宇<sup>1</sup>, 魏海涛<sup>2</sup>

(1. 郑州大学 信息工程学院, 郑州 450001; 2. 郑州大学 地球科学与技术学院, 郑州 450052)

**摘要:** 目前常见的轨迹聚类大多基于 OPTICS、DBSCAN 和 K-means 等算法, 但这些聚类方法的时间复杂度随着轨迹数量的增加会大幅上升。针对该问题, 提出一种基于密度核心的轨迹聚类算法。通过引入密度核心的概念, 设计轨迹密度计算函数以获取聚类簇的致密核心轨迹, 同时利用出租车载客轨迹自身的方向和速度等属性提取轨迹特征点, 减少轨迹数据量。在此基础上, 根据聚类簇中致密核心轨迹与参与聚类轨迹的相似度距离判断轨迹的匹配程度, 进而聚合相似轨迹, 并将聚类结果储存在聚类节点中。实验结果表明, 与 TRACCLUS 和 OPTICS 聚类算法相比, 该算法能够得到更准确的聚类效果, 并且时间效率更高。

**关键词:** DBSCAN 算法; 特征点; 密度核心; 相似度距离; 轨迹聚类

开放科学(资源服务)标志码(OSID):



**中文引用格式:** 田智慧, 马占宇, 魏海涛. 基于密度核心的出租车载客轨迹聚类算法[J]. 计算机工程, 2021, 47(2): 133-138.

**英文引用格式:** TIAN Zhihui, MA Zhanyu, WEI Haitao. Taxi passenger trajectory clustering algorithm based on density core[J]. Computer Engineering, 2021, 47(2): 133-138.

### Taxi Passenger Trajectory Clustering Algorithm Based on Density Core

TIAN Zhihui<sup>1,2</sup>, MA Zhanyu<sup>1</sup>, WEI Haitao<sup>2</sup>

(1. College of Information Engineering, Zhengzhou University, Zhengzhou 450001, China;

2. College of Earth Science and Technology, Zhengzhou University, Zhengzhou 450052, China)

**[Abstract]** The existing trajectory clustering methods are mostly based on OPTICS, DBSCAN, and K-means clustering algorithms, etc., but their time complexity soars with the increase of the number of trajectories. To address the problem, this paper proposes a trajectory clustering algorithm based on density core. By introducing the concept of density core, a trajectory density calculation function is designed to obtain the dense core trajectory of the cluster. At the same time, the attributes of the taxi passenger trajectory, including the direction and speed, are used to extract the trajectory feature points to reduce the amount of trajectory data. Then based on the similarity distance between the dense core trajectories in the cluster and the participating clustering trajectories, the matching degree of the trajectories is judged, and then similar trajectories are aggregated. The clustering results are stored in the cluster nodes. Experimental results show that the proposed algorithm is more accurate and efficient than TRACCLUS, OPTICS and other clustering algorithms.

**[Key words]** DBSCAN algorithm; feature point; density core; similarity distance; trajectory clustering

DOI: 10.19678/j.issn.1000-3428.0057015

### 0 概述

近年来, 数据挖掘、无线通信、移动定位等技术发展迅速, 持有 GPS 定位信息的移动设备以及基于位置信息服务(Location Information Service, LBS)的终端应用产生了海量时空轨迹数据<sup>[1]</sup>。这些时空数据蕴含着丰富的语义信息, 可用于提取移动对象的运动特征模式和预测移动对象的运动行为<sup>[2-4]</sup>, 已成

为目前空间数据挖掘方面<sup>[5-7]</sup>的研究热点。轨迹大数据可以分为两大类: 一类是由出租车以及其他携带 GPS 交通工具生成的坐标序列; 另一类是移动物体利用特殊装置(如信号基站、连接 WiFi 信号)而产生的带有时间戳的地点坐标序列<sup>[8]</sup>等。这些轨迹数据包含行人的出行规律<sup>[9]</sup>、交通拥堵规律<sup>[10]</sup>和社会活动模式等信息。轨迹聚类的目的是挖掘轨迹大数据的移动模式, 通过分析聚类结果得到移动对象的出

**基金项目:** 河南省重点研发与推广专项(科技攻关)(192102210124)。

**作者简介:** 田智慧(1965—), 男, 教授、博士, 主研方向为数据挖掘、深度学习; 马占宇, 硕士研究生; 魏海涛(通信作者), 讲师、博士。

**收稿日期:** 2019-12-25 **修回日期:** 2020-02-15 **E-mail:** 15110846644@163.com

行规律。其中,研究出租车轨迹聚类对城市规划和出租管理、出租车调度以及城市发展具有积极意义。

文献[11]提出了移动微聚类(Moving Micro Clustering, MMC)的概念,通过应用BIRCH算法中的微聚类思想来改进移动对象的聚类过程,并利用聚类特征描述MMC的特征信息。文献[12]对OPTICS算法加以改进,提出一种基于时间维度的T-OPTICS轨迹聚类算法。该算法增加了轨迹的时间语义信息,提高了聚类结果在时间维度的准确性。文献[13]构建分段及归组轨迹聚类框架TRACCLUS,利用提取到的轨迹特征点将轨迹分为多段,使粒度更细致,在此基础上利用DBSCAN聚类算法再进行归组,以避免丢失部分轨迹信息。但是DBSCAN聚类算法对参数敏感,需要利用专家先验知识获得参数并反复调整得到最优值。文献[14]提出基于路网的聚类算法框架NETSCAN,将路段聚类为一组密集路径簇后,再根据密集路径之间的相似性度量对轨迹进行聚类。文献[15]提出一种基于MFTSM的轨迹聚类算法,利用基于区域计算的位置距离解决轨迹的连续性问题。文献[16]提出一种T-CLUS轨迹聚类方法,在将原始轨迹划分成粒度较小的子段后再进行聚类,得到子轨迹对象的增广聚类序列,根据可达距离图分析不同参数的聚类效果。文献[17]提出一种基于结构相似度的轨迹聚类算法,通过提取轨迹方向、速度、转角和位置等组成的结构特征信息计算相似性度量,并基于结构相似性度量进行轨迹聚类。文献[18]在聚类算法上采用外包矩形作为核心轨迹的搜索邻域,同时重新定义轨迹核心距离与轨迹可达距离,通过邻接表代替空间索引,从而降低算法的复杂度。

现有的轨迹聚类大多基于DBSCAN、OPTICS和K-means等算法,聚类时需要计算所有轨迹或者轨迹段相互间的相似性距离,在面向海量出租车载客轨迹数据时会消耗大量的计算资源,降低算法效率。本文通过引入密度核心<sup>[19]</sup>的概念,提出一种新的载客轨迹聚类算法。计算轨迹点的变向角和速度,根据变向角阈值、累积变向角阈值和速度阈值提取轨迹特征点,以减少轨迹的数据量,加快计算速度。同时,根据聚类节点中致密核心轨迹与参与聚类轨迹的相似度距离判断轨迹的匹配程度,从而聚合相似轨迹,提高聚类的执行效率。

## 1 轨迹数据处理

### 1.1 车辆轨迹模型

出租车轨迹数据一般包含出租车营运设备编号(ID)、经纬度信息、出租车营运状态、卫星定位时间和车辆速度等属性,表1给出了出租车轨迹数据的部分属性信息。根据营运状态可将出租车轨迹分为载客轨迹和空载轨迹。载客时出租车营运状态记为1,空载时出租车营运状态记为0。为研究出租车在不同营运状态下的移动模式,本文以载客轨迹为研究对象。同时

为更好地描述轨迹,对轨迹及其相关属性进行形式化描述。以TD表示载客轨迹集合, $TD=\{T_1, T_2, \dots, T_N\}$ 。载客轨迹T是由出租车在连续的时间段内行驶生成的载客轨迹点组成的集合,表示为 $T=\{p_i | 1 \leq i \leq n\}$ ,其中,n为出租车载客轨迹点的个数, $p_i$ 为载客轨迹点。轨迹点 $p_i$ 是由经度、维度和时间戳组成的三元组,表示为 $p_i=(\text{latitude}, \text{longitude}, \text{timestamp})$ 。

表1 出租车轨迹属性信息

Table 1 Attribute information of taxi trajectory

ID	纬度/(°)	经度/(°)	方向/(°)	时间戳	状态	速度/(km·h <sup>-1</sup> )
TP429	34.644 903	113.668 202	0	2017/2/26 7:03	1	0
T1289	34.807 632	113.697 377	0	2017/2/26 7:08	1	47
TU255	34.758 598	113.712 393	0	2017/2/26 7:13	1	36
TV670	34.794 873	113.686 215	0	2017/2/26 7:18	1	0

### 1.2 轨迹清洗

出租车在行驶过程中,所获取GPS信息的准确度会受到各种地理环境或者天气因素的影响,生成的轨迹数据可能会包含一些噪声数据(如偏移点、回溯点和丢失点)。如图1所示,出租车在轨迹点 $p_7$ 偏离正确轨迹方向。由于噪声数据会影响轨迹聚类的精度,因此需要对轨迹数据进行降噪处理。轨迹降噪的目的就是寻找到轨迹原始数据中的噪声数据并对其进行平滑处理,以此消除噪声值。本文采用卡尔曼滤波法对出租车轨迹数据进行降噪处理。卡尔曼滤波模型是一种线性最优滤波器,本文利用该模型构建一个预测模型,以准确反映出出租车轨迹前一个状态和当前状态的关系。在此基础上,根据噪声点的误差值对出租车轨迹进行清洗,最终消除噪声点。

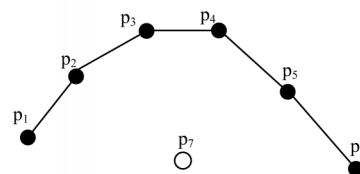


图1 噪声数据示意图

Fig.1 Schematic diagram of noise data

### 1.3 轨迹特征点提取

出租车在行驶过程中,GPS采样的频率较高,导致轨迹中存在大量的冗余数据点,这些重复的轨迹点在进行存储和计算时会消耗大量的资源。提取轨迹特征点的目的是在保证轨迹形状基本不变的情况下尽可能减少重复轨迹点的个数。目前的轨迹特征点提取方法主要是选取道路交汇口轨迹采样点和方向变化较大的轨迹采样点,如文献[16-18]方法,其选择轨迹点方向变化超过预先设置的角度阈值的采样点作为特征点。

这种方法虽然可以选取到有关键点,但忽略了轨迹方向的累积变化对特征点选取的影响。当角度阈值设置过大时,在选取轨迹特征点的过程中会丢失部分有价值的轨迹采样点。如图2所示,由于载客轨迹的所有变向角 $\{\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6\}$ 都小于角度阈值,因此这些采样点不能被选取为轨迹特征点,显然轨迹丢失了部分特征点。为解决这一问题,本文在角度阈值的基础上增加了累积角度阈值和速度阈值,以更精准地选取轨迹的特征点,如图3所示。

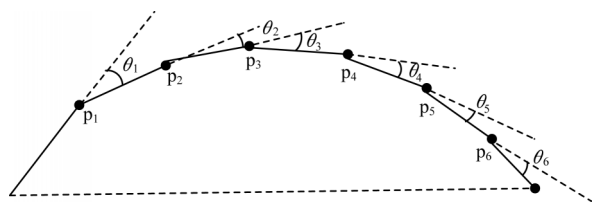


图2 累积变向角

Fig.2 Cumulative turning angles

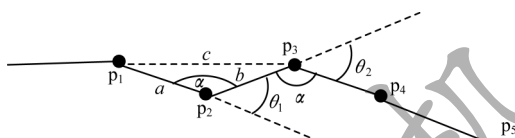


图3 轨迹特征点

Fig.3 Trajectory feature points

相邻轨迹采样点 $p_1 \sim p_3$ 的速度分别表示为 $V_{p_1} \sim V_{p_3}$ , 轨迹点 $p_2$ 的夹角表示为 $\alpha$ , 轨迹段长度 $a, b, c$ 表示轨迹点 $p_2$ 邻边和对边的长度, 夹角 $\alpha$ 的计算公式如下:

$$\alpha = \arccos \frac{a^2 + b^2 - c^2}{2ab} \quad (1)$$

轨迹点的变向角反映了轨迹运动的变化趋势。轨迹采样点 $p_2, p_3$ 的变向角分别表示为 $\theta_1$ 和 $\theta_2$ 。为便于轨迹运动趋势的相似度计算, 指定外向角 $\theta_1$ 为正值, 内向角 $\theta_2$ 为负值。由式(1)可以得到变向角 $\theta$ 的计算式如下:

$$\theta = \begin{cases} \pi - \alpha, & \vec{a} \times \vec{b} \geq 0 \\ \alpha - \pi, & \vec{a} \times \vec{b} < 0 \end{cases} \quad (2)$$

轨迹采样点的速度变化一定程度上反映了出租车轨迹的运动模式, 轨迹采样点 $p_i$ 的速度变化可用式(3)表示:

$$a_{p_i} = |V_{p_i} - V_{p_{i-1}}| + |V_{p_{i+1}} - V_{p_i}| \quad (3)$$

轨迹特征点提取是轨迹聚类的基础, 影响到轨迹聚类的准确率和执行效率。本文借鉴文献[17]中的角度阈值算法并加以改进, 在角度阈值的基础上增加累积角度阈值和速度阈值, 以更精准地保留轨迹的细节信息。本文设计的特征点提取算法步骤如下:

**步骤1** 遍历轨迹中的点序列, 提取道路交汇点作为轨迹的特征点, 利用式(1)、式(2)计算轨迹点的变向角, 选取变向角大于 $\omega_1$ 的轨迹点确定为轨迹特征点。

**步骤2** 如果轨迹点累积的变向角 $\sum_{i=1}^n |\theta_i|$ 大于累

积变向角阈值 $\omega_2$ , 那么该点也会被保留为特征点。在此基础上, 利用式(3)计算轨迹点的变速大小 $a_{p_i}$ , 选取大于速度阈值 $\varepsilon$ 的采样点作为轨迹特征点, 并将这些特征点组成的轨迹作为轨迹聚类的基础单位。

## 2 基于密度核心的轨迹聚类算法

### 2.1 DSP距离

SP(Sum-of-Pairs)距离<sup>[20]</sup>表示两条轨迹匹配点之间距离的总和, 其要求两条轨迹具有相同数量的点, 并且按照两条轨迹上点的顺序一一对应, 将匹配点的距离求和作为轨迹相似性度量。由于该度量方法要求两条具有相同点数量的轨迹, 因此不适用于轨迹长度不同的出租车载客轨迹。本文对SP距离进行改进, 提出DSP(Double Sum-of-Pairs)距离, 在计算两条轨迹的相似度时, 根据每条轨迹自身的轨迹点去匹配另一条轨迹的点, 每个轨迹点匹配到的是另一条轨迹上距离自己最小的轨迹点。因此, 两条轨迹相互之间匹配到的轨迹点并不相同。由于双向计算轨迹相似度, 避免了偶然性, 因此可使匹配轨迹的相似度更精确。以轨迹 $T_A$ 和 $T_B$ 为例: 轨迹 $T_A$ 到轨迹 $T_B$ 的距离如图4(a)所示, 以轨迹 $T_A$ 的每个点到轨迹 $T_B$ 中点的最短距离和的均值表示; 轨迹 $T_B$ 到轨迹 $T_A$ 的距离如图4(b)表示, 以轨迹 $T_B$ 的每个点到轨迹 $T_A$ 中点的最短距离和的均值表示。

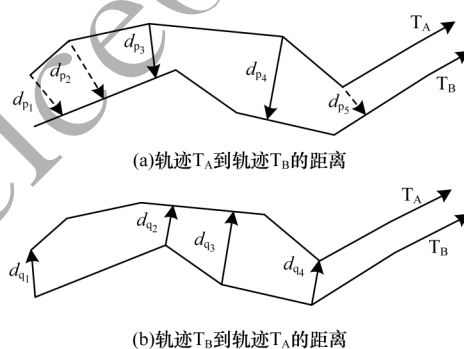


图4 轨迹对之间的相互距离

Fig.4 Mutual distance between trajectory pairs

DSP距离的形式化描述如下: 给定两条轨迹 $T_A$ 和 $T_B$ ,  $T_A = \{p_1, p_2, \dots, p_n\}$ ,  $T_B = \{q_1, q_2, \dots, q_m\}$ 。轨迹 $T_A$ 中任意一点 $p_i$ 到轨迹 $T_B$ 的距离定义为 $p_i$ 轨迹点与 $T_B = \{q_1, q_2, \dots, q_m\}$ 中任意轨迹点的最小距离, 如式(4)所示:

$$d_{p_i} = \min(d(p_i, q_j)) \quad (4)$$

其中,  $q_j (1 \leq j \leq m)$  表示轨迹 $T_B$ 中的一个轨迹点。由于出租车轨迹的长度不同且受轨迹方向的影响, 因此轨迹 $T_B$ 中的点 $q_j$ 到轨迹 $T_A$ 的距离不同于上述情况。轨迹 $T_B$ 中任意一点 $q_j$ 与轨迹 $T_A$ 的最小距离如式(5)所示:

$$d_{q_j} = \min(d(q_j, p_i)) \quad (5)$$



定义轨迹  $T_A$  到轨迹  $T_B$  的距离  $d_{AB}$  为  $T_A$  中所有点到  $T_B$  的距离和的均值,如式(6)所示:

$$d_{AB} = \frac{\sum_{i=0}^n d_{p_i}}{n} \quad (6)$$

同理可得轨迹  $T_B$  到轨迹  $T_A$  的距离  $d_{BA}$ ,如式(7)所示:

$$d_{BA} = \frac{\sum_{j=0}^m d_{q_j}}{m} \quad (7)$$

对  $d_{AB}$  和  $d_{BA}$  求和并取平均值可以得到 DSP 相似性度量,如式(8)所示:

$$S_{AB} = (d_{AB} + d_{BA})/2 \quad (8)$$

## 2.2 C-Tra 聚类算法

在传统的 DBSCAN、OPTICS 等聚类算法中,查询邻域内的轨迹对象数目需要扫描一次全部轨迹数据,轨迹聚合时需要再扫描一次全部数据,当数据量增大时,聚类的时间效率就会大幅下降。为降低聚类算法的复杂度,本文提出一种新的聚类算法 C-Tra。该算法定义聚类节点作为一种数据结构来储存聚类过程中生成的聚类簇,每一个节点储存一个类簇。所有的聚类节点构成轨迹聚类集合,如式(9)所示:

$$C = \{c_1, c_2, \dots, c_M\} \quad (9)$$

聚类节点包含类簇的细节信息,由轨迹列表 CTrList、致密核心轨迹  $h$ 、轨迹数量  $n$  构成,如式(10)所示:

$$c_i = (CTrList, h, n) \quad (10)$$

其中,轨迹列表 CTrList 为聚类簇中所有轨迹组成的集合,如式(11)所示,列表中的轨迹随着聚类过程满足聚类条件而动态增加,直到聚类完成,轨迹列表停止更新。

$$CTrList = [T_1, T_2, \dots, T_n] \quad (11)$$

聚类节点中的致密核心轨迹  $h$  是轨迹密度最高的轨迹,轨迹  $T_i$  的轨迹密度如式(12)所示:

$$\rho_i = \sum_j \chi(S_{T_i, T_j} - d_c) \quad (12)$$

其中:  $T_j$  表示聚类簇中的其他轨迹;当  $x < 0$  时,  $\chi(x) = 1$ ,而在其他情况下,  $\chi(x) = 0$ ;  $d_c$  为截至距离。

在本文设计的 C-Tra 算法中,参与聚类的轨迹  $T$  与每一个聚类节点的致密核心轨迹的相似度距离集合如式(13)所示:

$$\text{simlist} = \{s_1, s_2, \dots, s_M\} \quad (13)$$

在 simlist 集合中,最小的相似度距离  $s_i$  如果小于聚类阈值,则轨迹  $T$  可分配到此距离对应的聚类节点中。当所有参与聚类的轨迹聚合到对应的聚类节点,即完成轨迹聚类。在 C-Tra 聚类算法中,聚类节点的致密核心轨迹需要动态确定,每当有新的轨迹添加到聚类节点中,节点就会根据轨迹列表中全部轨迹的密度重新确定致密核心轨迹,以保证聚类结果的准确性。

C-Tra 聚类算法的伪代码如下:

输入 出租车轨迹集合  $TD = \{T_1, T_2, \dots, T_M\}$ , 聚类阈值  $\theta$ , 截至距离  $d_c$

输出 轨迹聚类集合  $C = \{c_1, c_2, \dots, c_M\}$

1. Create the first cluster node  $c_1 = ([T_1], T_1, 1)$

2.  $C = [c_1]$

3.  $M = 1$

4. for each  $T_i \in TD$  do

5.  $l = T_i$

6. for each  $k \in M$  do

7.  $s_k = S_{h_k, l}$

8. End for

9.  $d = \min(\text{simlist})$

// Index value of class cluster

10.  $F = \text{indexmin}(\text{simlist})$

11. if  $d < \theta$  then

12. add  $l$  into the list CTrList of  $c_F$

13.  $n = n + 1$

14. for each  $T_p \in \text{CTrList}$  do

15.  $\rho_p = \sum_q \chi(S_{T_p, T_q} - d_c)$

// trajectory index value with the highest density

16.  $e = \max \rho_p$

17.  $h = T_e$

// Update cluster nodes

18.  $c_F = (\text{CTrList}, h, n)$

19. else

20.  $M = M + 1$

21.  $c_M = ([T_i], T_i, 1)$

22. end if

23. end for

该算法的具体步骤如下:

**步骤 1** 选择参与聚类的第一个轨迹  $T_1$  并将其放入第一个聚类节点  $c_1$  中  $([T_1], T_1, 1)$ , 此时聚类节点的数量  $M=1$  (见算法第 1 行~第 3 行)。

**步骤 2** 选择轨迹列表中的轨迹线  $T_i (i=1, 2, \dots, N)$ , 分别计算轨迹  $T_i$  和所有聚类节点  $c_k (k=1, 2, \dots, M)$  的致密核心轨迹  $h_k$  的相似度轨迹距离  $s_{T_i, h_k}$ , 选择相似度  $d = \min(s_{T_i, h_k})$  最小的聚类节点  $c_F$ 。(见算法第 4 行~第 10 行)

**步骤 3** 如果  $d$  小于聚类阈值  $\theta$ , 则将  $T_i$  添加到聚类节点  $c_F$  (见算法第 11 行~第 18 行)。

**步骤 4** 如果  $d$  不小于聚类阈值  $\theta$ , 则创建一个新的聚类簇  $c_M ([T_i], T_i, 1)$ ,  $M=M+1$  (见算法第 19 行~第 23 行)。

## 3 实验与结果分析

为验证 C-Tra 聚类算法的性能,在河南省超算中心的 CPU 节点上进行实验。该节点拥有 2 颗 CPU 处理器,内存为 128 GB。实验数据为郑州市出租车 GPS 轨迹数据,时间为 2017 年 2 月 26 日,这些数据是 1.2 万余辆出租车的 GPS 记录,每条记录包含约 30 个字段信息,包括经纬度、状态、速度、方向、温度、湿度等。经过对轨迹数据的预处理,选取设备编号、运营状态、纬度、经度、时间戳这五个属性存储在 ORACLE 11g 中进行实验。

### 3.1 特征点提取前后的存储空间对比

依次选取不同数量的载客轨迹数据,如表 2 所示,将这些轨迹数据在提取特征点前后分别存储于 .txt 文件中。选取参数  $\omega_1=25$ 、 $\omega_2=60$ 、 $\varepsilon=65$  提取轨迹特征点。

特征点提取前后轨迹数据存储空间对比如表 2 所示。可以看出,通过提取特征点能够大幅降低轨迹存储的数据量,减小内存空间的消耗。

表 2 特征点提取前后存储空间对比

Table 2 Comparison of storage space before and after feature point extraction

轨迹数量/条	存储空间/KB	
	特征点提取前	特征点提取后
1 000	2 641	1 380
5 000	11 926	5 678
10 000	23 770	12 497
20 000	46 574	25 149
30 000	70 143	37 175

3.2 聚类结果与效率评价

C-Tra 聚类算法的特性是快速高效,适用于大数据量的载客轨迹。本文选择郑州市 2017 年 2 月 26 日午高峰

(11:00—14:00)的出租车轨迹数据,选取其中 10 000 条轨迹作为实验数据集进行验证。对本文 C-Tra 算法与 TRACLUS、OPTICS 算法的实验结果进行对比。表 3 列出了 3 种算法的时间复杂度,其中,  $M$  表示聚类簇的数目。图 5 给出了 3 种算法的聚类效果,可以看出,本文算法可以实现对载客轨迹的聚类,且聚类的轨迹更加集中,聚类效果更为明显。由于本文算法使用整体轨迹进行聚类,因此聚类后轨迹的长度较长,保留了轨迹的整体信息,能够准确反映载客轨迹的热点路径。

表 3 3 种算法的时间复杂度对比

Table 3 Comparison of time complexity of three algorithms

算法	时间复杂度
TRACLUS 算法	$O(N^2)$
OPTICS 算法	$O(N^2)$
C-Tra 算法	$O(M \cdot N)$

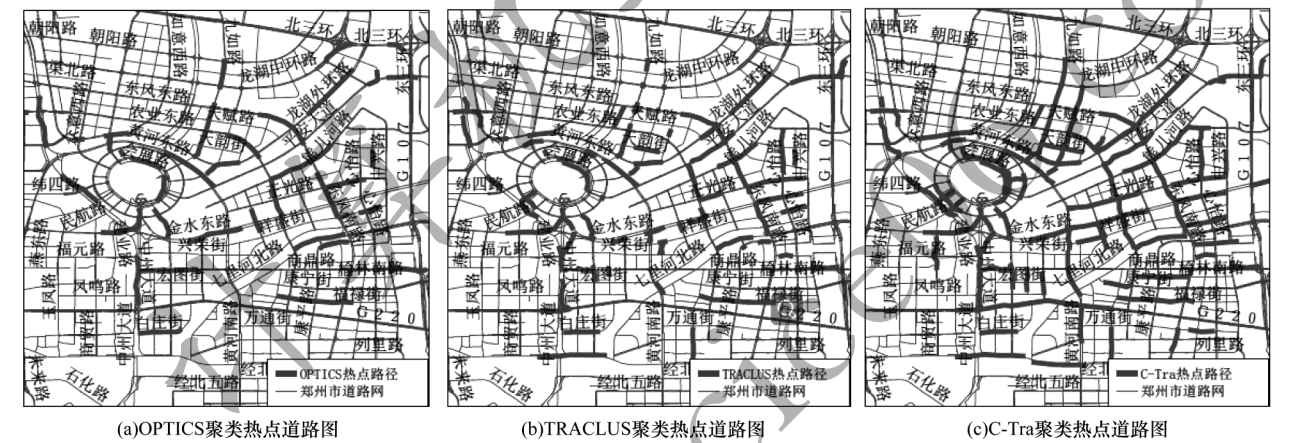


图 5 3 种算法的聚类效果对比

Fig.5 Comparison of clustering effect of three algorithms

3 种算法在出租车载客数据实验时的相关参数和运行时间如表 4 所示。可以看出,相对于 TRACLUS 和 OPTICS 算法,C-Tra 算法具有更高的执行效率。

表 4 3 种算法的参数设置和运行时间对比

Table 4 Comparison of parameter setting and running time of three algorithms

算法	参数	运行时间/s
TRACLUS 算法	$\text{MinPts}=19, \epsilon=160$	54.37
OPTICS 算法	$\text{MinPts}=36, \epsilon=31.5$	60.80
C-Tra 算法	$\theta=0.6, d_c=0.3$	16.73

4 结束语

本文研究出租车载客轨迹聚类问题,考虑载客轨迹特性并从提高算法执行效率角度出发,提出具有线性复杂度的聚类算法 C-Tra。根据轨迹的方向和速度

提取特征点以精简轨迹,通过改进 SP 距离使算法适用于长度不同的出租车载客轨迹,同时利用致密核心轨迹计算动态聚类节点的核心轨迹,完成轨迹聚类。实验结果表明,与 TRACLUS 和 OPTICS 算法相比,本文算法对于出租车轨迹聚类具有较高的执行效率,可为城市规划管理和交通拥堵治理提供借鉴。

参考文献

[ 1 ] FENG Huifang, BAI Fengshan, XU Youji. Urban traffic perception and critical node identification of road network based on trajectory big data [J]. Journal of Transportation Systems Engineering and Information Technology, 2018, 18(3): 42-47, 54. (in Chinese)  
冯慧芳,柏凤山,徐有基. 基于轨迹大数据的城市交通感知和路网关键节点识别[J]. 交通运输系统工程与信息, 2018, 18(3): 42-47, 54.

- [2] CHENG Zhiyuan. Traffic hotspot analysis based on trajectory clustering[D]. Chengdu: University of Electronic Science and Technology of China, 2018. (in Chinese)  
程智源. 基于轨迹聚类的交通热点分析[D]. 成都: 电子科技大学, 2018.
- [3] JIANG Hailin. Study on the temporal and spatial characteristics of urban residents' travel based on taxi data[D]. Wuhan: Wuhan University, 2018. (in Chinese)  
姜海林. 基于出租车数据的城市居民出行时空特征研究[D]. 武汉: 武汉大学, 2018.
- [4] ZHAO Xin. Research on mining urban hotspots and hot paths based on spatiotemporal constraint[D]. Chongqing: Chongqing University, 2017. (in Chinese)  
赵欣. 基于时空约束的城市热点区域与热点路径挖掘[D]. 重庆: 重庆大学, 2017.
- [5] GUAN Xuefeng, ZENG Yumei. Research progress and trends of parallel processing, analysis, and mining of big spatiotemporal data[J]. Progress in Geography, 2018, 37(10): 1314-1327. (in Chinese)  
关雪峰, 曾宇媚. 时空大数据背景下并行数据处理分析挖掘的进展及趋势[J]. 地理科学进展, 2018, 37(10): 1314-1327.
- [6] TIAN Zhiyong. Research on recharging characteristics of electric vehicle taxis based on trajectory data mining[D]. Wuhan: Huazhong University of Science and Technology, 2018. (in Chinese)  
田智勇. 基于轨迹数据挖掘的新能源出租车充电特征研究[D]. 武汉: 华中科技大学, 2018.
- [7] ZHANG Zhining. Research on trajectory data mining based on K-means and DBSCAN[J]. China Strategic Emerging Industry, 2017(44): 113-114. (in Chinese)  
张致宁. 基于K-means和DBSCAN的轨迹数据挖掘研究[J]. 中国战略新兴产业, 2017(44): 113-114.
- [8] XU Qiang. Research and scheme design of indoor location and trajectory prediction technology based on WiFi and map matching fusion[D]. Nanjing: Nanjing University of Posts and Telecommunications, 2018. (in Chinese)  
徐强. 基于WiFi与地图匹配融合的室内定位及轨迹预测技术研究及方案设计[D]. 南京: 南京邮电大学, 2018.
- [9] FENG Qisen. Mining of residents' trip hot routes and areas based on taxi trajectory[D]. Chongqing: Chongqing University, 2016. (in Chinese)  
冯琦森. 基于出租车轨迹的居民出行热点路径和区域挖掘[D]. 重庆: 重庆大学, 2016.
- [10] XU Haiyang. Temporal and spatial characteristics analysis of taxi urban traffic[D]. Nanjing: Nanjing University, 2019. (in Chinese)  
徐海洋. 出租车城市交通时空特征分析[D]. 南京: 南京大学, 2019.
- [11] LI Yifan, HAN Jiawei, YANG Jiong. Clustering moving objects[C]//Proceedings of KDD'04. New York, USA: ACM Press, 2004: 617-622.
- [12] NANNI M, PEDRESCHI D. Time-focused clustering of trajectories of moving objects[J]. Journal of Intelligent Information Systems, 2006, 27(3): 267-289.
- [13] LEE J G, HAN J, WHANG K Y. Trajectory clustering: a partition-and-group framework[C]//Proceedings of ACM SIGMOD International Conference on Management of Data. New York, USA: ACM Press, 2007: 593-604.
- [14] KHARRAT A, POPA I S, ZEITOUNI K, et al. Clustering algorithm for network constraint trajectories[M]//RUAS A, GOLD C. Headway in spatial data handling. Berlin, Germany: Springer, 2008: 631-647.
- [15] YU Qingying, LUO Yonglong, CHEN Chuanming, et al. Trajectory similarity clustering based on multi-feature distance measurement[J]. Applied Intelligence, 2019, 49(6): 2315-2338.
- [16] ZHANG Yanling, LIU Jinpeng, JIANG Baoqing. Partition and clustering for sub-trajectories of moving objects[J]. Computer Engineering and Applications, 2009, 45(10): 65-68. (in Chinese)  
张延玲, 刘金鹏, 姜保庆. 移动对象子轨迹段分割与聚类算法[J]. 计算机工程与应用, 2009, 45(10): 65-68.
- [17] YUAN Guan, XIA Shixiong, ZHANG Lei, et al. Trajectory clustering algorithm based on structural similarity[J]. Journal on Communications, 2011, 32(9): 103-110. (in Chinese)  
袁冠, 夏士雄, 张磊, 等. 基于结构相似度的轨迹聚类算法[J]. 通信学报, 2011, 32(9): 103-110.
- [18] YANG Shuliang, BI Shuoben, NKUNZIMANA A, et al. Spatial clustering method for taxi passenger trajectory[J]. Computer Engineering and Applications, 2018, 54(14): 249-255. (in Chinese)  
杨树亮, 毕硕本, NKUNZIMANA A, 等. 一种出租车载客轨迹空间聚类方法[J]. 计算机工程与应用, 2018, 54(14): 249-255.
- [19] RODRIGUEZ A, LAIO A. Clustering by fast search and find of density peaks[J]. Science, 2014, 344(6191): 1492-1496.
- [20] DAN G. Algorithms on stings, trees, and sequences[J]. ACM SIGACT News, 1997, 28(4): 41-60.