

控制与决策

Control and Decision

基于时空加权和密度峰值的轨迹聚类算法

赵嘉, 段发祥, 潘正祥, 王奔, 张翼英, 于华东, 吴润秀

引用本文:

赵嘉, 段发祥, 潘正祥, 等. 基于时空加权和密度峰值的轨迹聚类算法[J]. *控制与决策*, 2026, 41(2): 470-480.

在线阅读 View online: <https://doi.org/10.13195/j.kzyjc.2025.0546>

您可能感兴趣的其他文章

Articles you may be interested in

基于波段影像统计信息量加权K-means聚类的高光谱影像分类

Algorithm based on band statistical information weighted K-means for hyperspectral image classification

控制与决策. 2021, 36(5): 1119-1126 <https://doi.org/10.13195/j.kzyjc.2019.1516>

基于相互邻近度的密度峰值聚类算法

Density peaks clustering based on mutual neighbor degree

控制与决策. 2021, 36(3): 543-552 <https://doi.org/10.13195/j.kzyjc.2019.0795>

基于优化DBSCAN聚类算法的晶圆图预处理

Wafer map preprocessing based on optimized DBSCAN clustering algorithm

控制与决策. 2021, 36(11): 2713-2721 <https://doi.org/10.13195/j.kzyjc.2020.0738>

基于KPCA和G-G聚类的多元时间序列模糊分段

Fuzzy segmentation of multivariate time series with KPCA and G-G clustering

控制与决策. 2021, 36(1): 115-124 <https://doi.org/10.13195/j.kzyjc.2019.0849>

一种基于相对密度和决策图的聚类算法

A novel clustering algorithm based on relative density and decision graph

控制与决策. 2018, 33(11): 1921-1930 <https://doi.org/10.13195/j.kzyjc.2017.0822>

基于时空加权和密度峰值的轨迹聚类算法

赵嘉^{1,2†}, 段发样^{1,2}, 潘正祥³, 王奔⁴, 张翼英⁵, 于华东⁶, 吴润秀^{1,2}

1. 江西水利电力大学 信息工程学院, 南昌 330099;
2. 江西省水利大数据智能处理与预警技术工程研究中心, 南昌 330099;
3. 南京信息工程大学 人工智能学院, 南京 210044;
4. 南瑞集团有限公司, 南京 211000; 5. 天津科技大学 人工智能学院, 天津 300457;
6. 北京智芯微电子科技有限公司, 北京 102200)

摘要: TRACLUS 算法在处理具有复杂时空特性的轨迹数据时, 其采用的分段策略侧重空间几何信息而忽略了时空动态信息, 导致分段精度不高; 空间邻近性的聚类的规则难以有效识别时空耦合模式, 且在处理噪声和密度不均数据方面存在局限性. 鉴于此, 提出基于时空加权和密度峰值的轨迹聚类算法. 该算法使用时空加权分段将自适应时空权重和时空几何距离融入最小描述长度代价函数, 提取包含时空局部特征的子轨迹段; 将时空局部密度融入密度峰值聚类, 并结合局部特征的噪声识别与迭代式类簇中心选择, 提升子轨迹段聚类效果; 将密度筛选和时空连续性约束嵌入代表性轨迹生成, 增强聚类结果可解释性. 在拓扑结构不同的北京和上海出租车数据集上的实验表明: 时空加权分段使轨迹重建误差平均降低 28.15%, 方向偏差平均降低 66%; STW-DP-TRACLUS 算法在 3 种评价指标上综合优于多种传统及先进的轨迹聚类算法, 验证了其在复杂时空轨迹模式挖掘方面的有效性.

关键词: TRACLUS 算法; 轨迹聚类; 密度峰值聚类算法; 时空加权; 最小描述长度; 代表性轨迹

中图分类号: TP181 文献标志码: A

DOI: 10.13195/j.kzyjc.2025.0546

引用格式: 赵嘉, 段发样, 潘正祥, 等. 基于时空加权和密度峰值的轨迹聚类算法 [J]. 控制与决策, 2026, 41(2): 470-480.

Trajectory clustering algorithm based on spatio-temporal weighting and density peak

ZHAO Jia^{1,2†}, DUAN Fa-yang^{1,2}, PAN Jeng-shyang³, WANG Ben⁴, ZHANG Yi-ying⁵, YU Hua-dong⁶, WU Run-xiu^{1,2}

- (1. School of Information Engineering, Jiangxi University of Water Resources and Electric Power, Nanchang 330099, China; 2. Jiangxi Province Engineering Research Center for Intelligent Processing and Early Warning Technology of Water Conservancy Big Data, Nanchang 330099, China; 3. School of Artificial Intelligence, Nanjing University of Information Science and Technology, Nanjing 210044, China; 4. NARI Group Corporation, Nanjing 211000, China; 5. College of Artificial Intelligence, Tianjin University of Science & Technology, Tianjin 300457, China; 6. Beijing Smartchip Microelectronics Technology Company Limited, Beijing 102200, China)

Abstract: The TRACLUS (trajectory clustering: a partition and group framework) algorithm's reliance on purely spatial metrics for its partitioning and clustering phases leads to inaccurate segmentation and a failure to identify coupled spatio-temporal patterns, especially with noisy, density-varied data. To address these limitations, we propose a trajectory clustering algorithm based on spatio-temporal weighting and density peak (STW-DP-TRACLUS). This method enhances the TRACLUS in three key aspects: First, spatio-temporal weighted segmentation integrates adaptive weights and geometric distances into the minimum description length cost function to extract sub-trajectory segments with local features; Second, improved density peak clustering incorporates spatio-temporal local density, novel noise identification, and iterative center selection to boost clustering quality; Third, density filtering and continuity constraints are embedded in the representative trajectory generation to enhance interpretability. Experiments on taxi

收稿日期: 2025-05-27; 录用日期: 2025-08-19.

基金项目: 国家自然科学基金项目 (62466037); 南昌市重大科技攻关项目 (2024zdxm010, 2024zdxm002).

责任编委: 徐泽水.

†通信作者. E-mail: zhaojia@nit.edu.cn.

datasets from Beijing and Shanghai, featuring different network topologies, demonstrate that the proposed method alone reduces reconstruction error by 28.15% and direction deviation by 66% on average. Overall, the STW-DP-TRACLUS outperforms traditional and state-of-the-art algorithms across three evaluation metrics, validating its efficacy in mining complex spatio-temporal patterns.

Keywords: TRACLUS algorithm; trajectory clustering; density peaks clustering algorithm; spatio-temporal weighted; minimum description length; representative trajectories

0 引言

智慧城市建设推动多源感知设备与移动终端的规模化部署, 导致城市时空数据总量呈指数增长^[1]. 其中, 轨迹数据作为有序时空三元组 (x, y, t) 集合, 每一三元组记录移动对象的位置坐标 (x, y) 和时间戳 t , 在交通管理^[2]、公共安全^[3] 和商业服务^[4] 等领域具有关键应用价值. 然而, 轨迹数据存在长程时空耦合^[5] 与密度分布不均^[6] 的特性, 给从中挖掘潜在模式带来了挑战.

轨迹聚类^[7] 作为一种有效的数据挖掘方法, 能够自动将相似轨迹分配至同一类簇. 现有方法主要分为两类^[8]: 基于相似性度量的方法通过定义轨迹间的距离函数, 如最长公共子序列 (longest common subsequence, LCSS)^[9]、动态时间规整 (dynamic time warping, DTW)^[10]、编辑距离 (edit distance)^[11] 等, 将相似轨迹归类; 基于密度的方法则通过识别轨迹在时空维度的密度分布来发现任意形状类簇并有效识别噪声, 例如 DBSCAN (density based spatial clustering of applications with noise) 算法^[12] 及其时空变体^[13-14], 以及密度峰值聚类 (density peaks clustering, DPC) 算法^[15] 及其融合变体^[16-17].

然而, 上述方法大多将完整轨迹作为聚类单元, 难以捕捉局部时空耦合特征^[18]. 分段聚类的代表性算法 TRACLUS (trajectory clustering: A partition and group framework) 算法^[19] 通过最小描述长度 (minimum description length, MDL) 准则对轨迹进行分段, 并利用 DBSCAN 算法对子轨迹段聚类. 尽管该算法能有效发现局部模式, 但存在两个限制: 1) 其轨迹分段与聚类过程均仅依赖空间信息, 忽略了时间维度, 无法挖掘时空耦合信息; 2) 其核心 DBSCAN 算法难以适应密度不均的子轨迹段类簇, 且固定的全局参数易将低密度簇的子轨迹段误判为噪声.

为解决上述限制, 本文提出一种基于时空加权和密度峰值的轨迹聚类算法 (trajectory clustering based on spatio-temporal weighting and density peak, STW-DP-TRACLUS). 该算法在 3 个关键阶段进行创新: 1) 时空加权分段, 在 MDL 代价函数中融入自适应时空权重, 生成反映局部时空特性的子轨迹段; 2) 时空密度峰值聚类: 采用改进的 DPC 算法替代

DBSCAN 算法, 引入时空局部密度和噪声识别机制, 提升对密度不均子轨迹段的聚类鲁棒性; 3) 代表性轨迹生成, 通过密度筛选与时空连续性约束提取高密度代表轨迹, 实现结果可视化.

1 相关工作

1.1 TRACLUS 算法

TRACLUS 算法是轨迹聚类代表性分段框架, 采用先分段再聚类的策略, 以应对全局方法忽略轨迹局部特征的问题. 其核心分为两阶段: 通过识别特征点将原始轨迹分段为子轨迹段; 随后应用 DBSCAN 算法对子轨迹段进行聚类. 该方法优势在于能提取局部共同模式, 并识别噪声子轨迹段, 提高聚类精细度.

为度量子轨迹段间相似性, TRACLUS 算法定义几何距离, 包括垂直距离 d_{\perp} 、平行距离 d_{\parallel} 和角度距离 d_{θ} 三种几何特征分量. 图 1 给出由 s_i 和 e_i 为端点连接而成的子轨迹段 L_i 和由 s_j 和 e_j 为端点连接而成的子轨迹段 L_j , p_s 和 p_e 分别是 s_j 和 e_j 在 L_i 上的投影点.

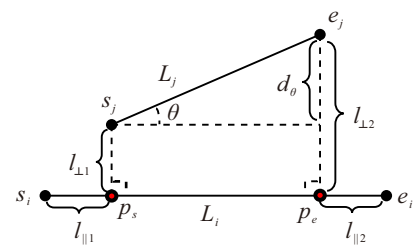


图1 三种几何特征分量

垂直距离 $d_{\perp}(L_i, L_j)$ 计算如下:

$$d_{\perp}(L_i, L_j) = \frac{l_{\perp 1}^2 + l_{\perp 2}^2}{l_{\perp 1} + l_{\perp 2}}. \quad (1)$$

其中: $l_{\perp 1}$ 为 s_j 与 p_s 之间的垂直欧氏距离, $l_{\perp 2}$ 同理. 式 (1) 采用 2 阶 Lehmer 均值, 以突出较大偏移的影响, 从而更有效地捕捉子轨迹段间的几何差异.

平行距离 $d_{\parallel}(L_i, L_j)$ 计算如下:

$$d_{\parallel}(L_i, L_j) = \min(l_{\parallel 1}, l_{\parallel 2}). \quad (2)$$

其中: $l_{\parallel 1}$ 为投影点 p_s 到 L_i 较近端点的欧氏距离, $l_{\parallel 2}$ 同理. 平行距离 $d_{\parallel}(L_i, L_j)$ 用于反映子轨迹段在运动方向上的重叠程度.

角度距离 $d_{\theta}(L_i, L_j)$ 计算如下:

$$d_\theta(L_i, L_j) = \begin{cases} \|\min(L_i, L_j)\| \times \sin(\theta), & 0 \leq \theta < 90^\circ; \\ \|\min(L_i, L_j)\|, & 90^\circ \leq \theta \leq 180^\circ. \end{cases} \quad (3)$$

其中: $\|\min(L_i, L_j)\|$ 为较短子轨迹段的欧氏长度, θ 为 L_i 与 L_j 之间的较小夹角. 式 (3) 从轨迹方向一致性的角度量化差异, 并对较大角度偏差施加更强的惩罚, 以精确反映子轨迹段间的运动模式. 基于以上 3 个距离, L_i 与 L_j 之间的几何距离函数 $\text{dist}(L_i, L_j)$ 定义如下:

$$\text{dist}(L_i, L_j) = w_\perp \cdot d_\perp(L_i, L_j) + w_\parallel \cdot d_\parallel(L_i, L_j) + w_\theta \cdot d_\theta(L_i, L_j), \quad (4)$$

其中默认权重均为 1, 以均衡各分量贡献. 轨迹分段采用 MDL 准则, 最小化总描述代价

$$\text{Cost}(\text{MDL}) = L(H) + L(D|H). \quad (5)$$

给定轨迹 $\text{Tr}(p_1, p_2, \dots, p_n)$, 选取特征点集 $\text{CP}(p_{c_1}, p_{c_2}, \dots, p_{c_k})$, 生成 $k-1$ 个子轨迹段 $\{L_1, L_2, \dots, L_{k-1}\}$. 模型复杂度 $L(H)$ 公式为

$$L(H) = \sum_{i=1}^{k-1} \log_2(\text{len}(L_i)). \quad (6)$$

其中: k 为分段点数, $\text{len}(L_i)$ 为第 i 条子轨迹段的欧氏长度. 同时, $L(D|H)$ 表示为

$$L(D|H) = \sum_{i=1}^{k-1} \sum_{j=p_j}^{p_{j+1}-1} \log_2(d_\perp(L_i, p_j p_{j+1})) + \log_2(d_\theta(L_i, p_j p_{j+1})). \quad (7)$$

其中: $d_\perp(L_i, p_j, p_{j+1})$ 为子轨迹段 L_i 与原始轨迹 $p_j p_{j+1}$ ($1 < j < n-1$) 之间的垂直距离, $d_\theta(L_i, p_j, p_{j+1})$ 为子轨迹段 L_i 与 $p_j p_{j+1}$ 之间的角度距离. 由于全局最优求解计算繁琐, 采用贪心近似策略. 从起点向前延长子轨迹段, 逐点比较当前点是否为特征点的代价. 若特征点代价更小, 则延长当前段; 否则, 固定前一点为特征点并开始新阶段. 该策略能够有效平衡计算效率与分段质量.

1.2 DPC 算法

DPC 算法是一种基于局部密度和相对距离的聚类算法^[20], 该算法的核心优势在于其对参数的敏感性较低, 仅需确定一个截断距离参数和相对距离, 这使得它在处理不同类型和规模的数据集时具有较高的鲁棒性. 该算法基于两个核心假设^[21]: 1) 类簇中心具有较高的局部密度; 2) 不同类簇之间具有较大的相对距离. DPC 算法通过局部密度和相对距离识别类簇中心, 这些假设使 DPC 算法适用于非凸类簇和噪声数据, 在轨迹聚类应用中具有显著潜力.

局部密度 ρ_i 用于衡量样本 i 周围的邻域密度, 由截断距离 d_c 确定, 有

$$\rho_i = \sum_j \chi(d_{ij} - d_c). \quad (8)$$

其中: d_{ij} 为样本 i 与样本 j 之间的欧氏距离; $\chi(x)$ 为指示函数, 当 $x < 0$ 时, $\chi(x) = 1$, 否则 $\chi(x) = 0$, 即等于距离样本 i 小于 d_c 的其他样本的数量. 为获得更平滑密度估计, 采用高斯核函数

$$\rho_i = \sum_j \exp\left(-\frac{d_{ij}^2}{d_c^2}\right). \quad (9)$$

相对距离 δ_i 用于衡量样本 i 与其他密度更高样本的分离程度, 计算公式为

$$\delta_i = \begin{cases} \min_{j: \rho_j > \rho_i} \{d_{ij}\}, & \exists j \text{ s.t. } \rho_j > \rho_i; \\ \max_j \{d_{ij}\}, & \text{otherwise.} \end{cases} \quad (10)$$

如果存在局部密度 ρ 大于 ρ_i 的样本 j , 则 δ_i 定义为样本 i 到所有这些密度更高样本中最邻近者的距离; 否则, δ_i 定义为样本 i 到所有其他样本点的最大距离. 该设计确保类簇中心具有较大的分离值, 从而突出其与其他高密度区域的分离特征.

DPC 算法结合 ρ 和 δ 识别类簇中心, 通常, 类簇中心会同时具有较高的 ρ 值和较高的 δ 值. 为便于选择, 定义一个决策值 γ 为

$$\gamma_i = \rho_i \cdot \delta_i. \quad (11)$$

计算所有样本的决策值 γ 并降序排序, 将 γ 值突出的样本作为类簇中心. 确定类簇中心后, DPC 算法将其余各点分配给距离其最近的且密度高于该点的样本所属的类簇, 完成聚类过程.

2 基于时空加权和密度峰值的轨迹聚类算法

STW-DP-TRACLUS 算法旨在克服 TRACLUS 算法在轨迹分段中时空感知不足的问题, 并改善其在子轨迹聚类中难以有效处理时空耦合与密度噪声的缺陷, 进而实现轨迹数据的有效聚类. STW-DP-TRACLUS 算法整合 3 个主要部分实现其目标, 整体流程框架如图 2 所示.

2.1 时空加权分段

定义 1(自适应时空权重) 对于轨迹 $\text{Tr}(p_1, p_2, \dots, p_n)$ 中的子轨迹段 $L_s(p_s, \dots, p_c)$, 其自适应时空权重为

$$\begin{cases} w_{\text{spatial}}^{[s,c]} = \frac{L_{\text{spatial}}^{[s,c]}}{L_{\text{spatial}}^{[s,c]} + L_{\text{temporal}}^{[s,c]}}, \\ w_{\text{temporal}}^{[s,c]} = \frac{L_{\text{temporal}}^{[s,c]}}{L_{\text{spatial}}^{[s,c]} + L_{\text{temporal}}^{[s,c]}}. \end{cases} \quad (12)$$

其中: $L_{\text{spatial}}^{[s,c]}$ 为该子轨迹段 $L_i(p_s, \dots, p_c)$ 内所有相

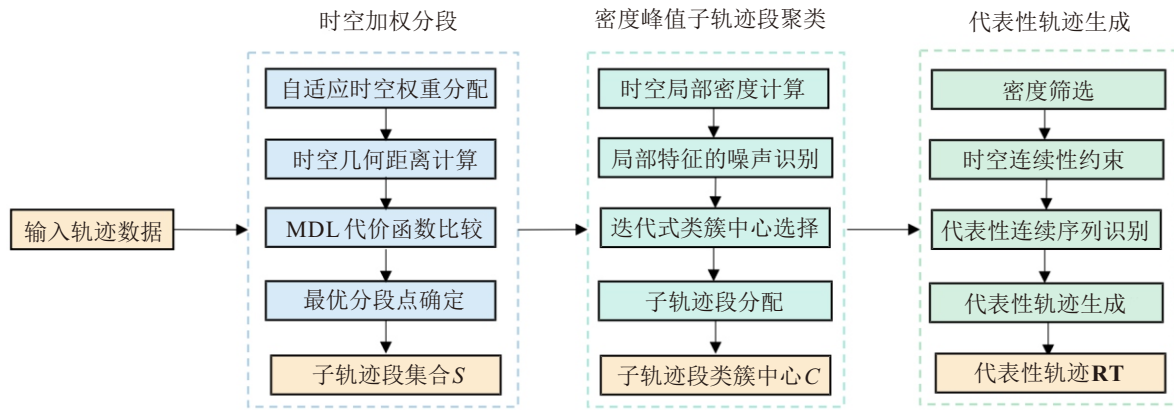


图2 STW-DP-TRACCLUS 算法流程

邻轨迹点间累积的欧氏空间距离总和, 即该子轨迹段的路径长度; $L_{\text{temporal}}^{[s,c]}$ 为该子轨迹段内所有相邻轨迹点间累积的时间差绝对值总和, 即该子轨迹段的总持续时间; $w_{\text{spatial}}^{[s,c]}$ 和 $w_{\text{temporal}}^{[s,c]}$ 根据该子轨迹段的整体时空特性, 作为自适应的调节因子, 用于平衡空间维度与时间维度的相对重要性, 使后续的计算能反映该特定子轨迹段的运动特征。

定义 2 (时空加权长度) 时空加权长度 len_{st} , 旨在为子轨迹段 $L_s(p_s, \dots, p_c)$ 赋予一个能够综合反映其时空跨度与内部运动特性的时空长度, 该长度通过计算该子轨迹段起点 $p_s(x_s, y_s, t_s)$ 与终点 $p_c(x_c, y_c, t_c)$ 之间的时空加权长度来确定, 有

$$\text{len}_{st}^{[s,c]} = w_{\text{spatial}}^{[s,c]} \cdot \sqrt{(x_s - x_c)^2 + (y_s - y_c)^2} + w_{\text{temporal}}^{[s,c]} \cdot |t_s - t_c|. \quad (13)$$

时空加权长度不仅度量子轨迹段首尾轨迹点间的时空距离, 而且通过动态权重间接蕴含该子轨迹段内部的运动特征, 作为评估单个子轨迹段长度的基础, 用于后续计算整个分段方案的模型复杂度。

定义 3 (时空几何距离) 为综合度量两条子轨迹段 L_i 与 L_j 之间的时空相似性, 对 TRACCLUS 算法中的几何距离进行时空扩展, 时空几何距离 $\text{dist}_{st}(L_i, L_j)$ 计算如下:

$$\text{dist}_{st}(L_i, L_j) = w_{\perp} \cdot d_{st\perp}(L_i, L_j) + w_{\parallel} \cdot d_{st\parallel}(L_i, L_j) + w_{\theta} \cdot d_{st\theta}(L_i, L_j), \quad (14)$$

其中 w_{\perp} , w_{\parallel} 和 w_{θ} 分别为满足和为 1 的权重系数, 用以调节各时空几何分量的相对重要性. 3 个核心分量 $d_{st\perp}$, $d_{st\parallel}$, $d_{st\theta}$ 是式 (1) ~ (3) 的时空拓展, 旨在度量子轨迹段间的综合时空相似性。

基于上述定义, 时空加权分段对 MDL 准则分段形成时空拓展. $L(H)$ 不再仅依赖各子轨迹段 L_i 的欧氏长度, 而是基于定义 2 计算的时空加权长度 len_{st} 衡量, 自适应地反映每个子轨迹段内部的时空运动

特性. $L(D|H)$ 则利用定义 3 中提出的时空几何距离 dist_{st} 衡量分段模型与原始轨迹的时空拟合误差. 同样, 由于精确求解全局最优分段仍然面临计算挑战, 时空加权分段同样采用前向贪心比较的近似计算方法进行求解, 在保证计算效率的同时, 获得能更好反映轨迹时空动态的分段结果。

2.2 密度峰值子轨迹段聚类

定义 4 (时空局部密度) 为量化每个子轨迹段在其时空邻域内的聚集程度, 对于子轨迹线段集合 $\mathbf{S} = \{s_1, s_2, \dots, s_n\}$ 中的任意子轨迹线段 s_i , 其时空局部密度 ρ_i 定义为其他所有子轨迹段 s_j 对其密度贡献的总和, 采用高斯核函数进行计算, 有

$$\rho_i = \sum_{j=1, j \neq i}^n \exp\left(-\left(\frac{\text{dist}_{st}(s_i, s_j)}{d_c}\right)^2\right). \quad (15)$$

其中: 时空局部密度 ρ_i 为累加集合 \mathbf{S} 中所有其他子轨迹段 $s_j (j \neq i)$ 对其密度贡献的总和, 该贡献度由高斯核函数确定, 高斯核函数能提供一个平滑的密度估计, 即所有子轨迹段均对密度有贡献, 但贡献度随距离平滑衰减, 这种特性通过刻画不同密度区域的内在结构, 从而在处理密度不均数据和识别噪声方面表现更优; $\text{dist}_{st}(s_i, s_j)$ 为定义 3 中提出的时空几何距离, 捕获子轨迹段 s_i 和 s_j 在空间位置和时间跨度上的相似性; d_c 为带宽参数, 控制密度贡献随距离衰减的速度; ρ_i 量化全局范围内其他子轨迹段对 s_i 的贡献程度, 是后续识别潜在类簇中心和判定噪声的核心依据之一。

定义 5 (局部特征的噪声子轨迹段) 为自动识别并处理在时空分布上表现出局部稀疏性且远离其邻近子轨迹段的异常子轨迹段 s_i , 基于其局部邻域的时空特性定义其噪声属性 noise_i , 有

$$\text{noise}_i = \begin{cases} 1, & \bar{d}_i > d_{\text{th}}, \bar{\rho}_i < \rho_{\text{th}}; \\ 0, & \text{otherwise.} \end{cases} \quad (16)$$

其中: \bar{d}_i 为 s_i 到其 k 近邻的平均时空几何距离, 反映

局部离散度; $\bar{\rho}_i$ 为其 k 近邻的平均局部时空密度, 反映局部聚集水平. 相应的局部特征距离阈值 d_{th} 和局部特征密度阈值 ρ_{th} 由整体数据集中各子轨迹段对应 \bar{d}_i 与 $\bar{\rho}_i$ 值的分布情况确定. 取 \bar{d}_i 分布的较高百分位数和 $\bar{\rho}_i$ 分布的较低百分位数, 以区分正常子轨迹段和噪声子轨迹段. 因此, 若子轨迹段 s_i 同时满足 \bar{d}_i 大于 d_{th} 且 $\bar{\rho}_i$ 小于 ρ_{th} 这两个条件, 则该段被识别并标记为噪声子轨迹段. 在后续的分类簇中心选择和子轨迹段分配过程中, 被标记为噪声的子轨迹段将不被考虑作为潜在的聚类中心, 也不会被分配到任何非噪声类簇中, 而是被直接识别为噪声子轨迹段. 这种机制利用噪声子轨迹段通常与其邻居时空距离较远且所处邻域整体较为稀疏的特性, 通过在聚类核心步骤前将其过滤, 从而有效减少噪声对聚类结构形成的干扰, 增强算法的鲁棒性.

定义 6 (时空因子决策值) 为实现子轨迹段类簇中心在迭代选择过程中的均衡分布, 避免在高密度区域过度聚集, 引入时空因子决策值 $ST(\gamma_i)$ 优化类簇中心的迭代选择过程. 对于一个尚未被选为类簇中心的子轨迹段 s_i , 其时空因子决策值定义为其基础决策值 γ_i 与其到当前已选类簇中心集合 C 中最近类簇中心时空距离的乘积, 即

$$ST(\gamma_i) = \gamma_i \times (\min_{e \in C} \{\text{dist}_{st}(s_i, e)\}). \quad (17)$$

其中: 定义的时空因子决策值 $ST(\gamma_i)$ 结合了子轨迹段 s_i 的基础决策值 γ_i 以及 s_i 到当前已选类簇中心集合 C 的最短时空几何距离. 为解决传统密度峰值算法易在高密度区域内选取过多邻近中心的问题, 本文设计了一种迭代式的类簇中心选择策略, 以确保最终选出的类簇中心在时空维度上具有良好的分离度与代表性. 该过程首先在所有非噪声子轨迹段中, 选择基础决策值 γ_i 最大的子轨迹段作为初始类簇中心. 确定初始中心后, 为避免后续选点与已有中心在时空上过于邻近, 算法引入时空因子决策值 $ST(\gamma_i)$ 指导后续选择. 该决策值将候选点的基础决策值 γ_i 与其到当前已选中心集合的最短时空几何距离相乘, 此距离项充当了一个动态的惩罚因子. 因此, 在后续的每一轮迭代选择中, 算法将优先选取 $ST(\gamma_i)$ 值最高的候选子轨迹段作为新的类簇中心. 该过程循环执行, 直至选出预设数量的中心为止. 此策略通过动态惩罚与已选中心邻近的候选点, 有效抑制了冗余中心的产生, 优化了最终类簇中心集的均衡性与代表性.

2.3 代表性轨迹生成

定义 7 (基于密度阈值的代表性候选子轨迹段

集) 为识别出类簇 C_j 中最具代表性的核心成员, 首先计算该类簇中所有子轨迹段的局部密度 ρ_i , 并确定簇内最大密度值 $\rho_{max} = \max_{i \in C_j} \rho_i$; 然后定义代表性候选子轨迹段集 R_j 为 C_j 中所有满足以下条件的子轨迹段 s_i 的集合: 其局部密度 ρ_i 与最大密度 ρ_{max} 的比值大于预设的密度比例阈值 α ($0 < \alpha < 1$). 数学表示为

$$R_j = \left\{ s_i \in C_j \mid \frac{\rho_i}{\rho_{max}} > \alpha \right\}. \quad (18)$$

其中: 该集合 R_j 包含类簇 C_j 中密度相对最高的子轨迹段, 被视为构建最终代表性轨迹 RT_j 的高质量候选对象; 密度比例阈值 α 控制筛选的严格程度.

定义 8 (时空连续性约束) 此约束是确保最终生成的代表性轨迹具有物理意义和逻辑连贯性的关键环节, 定义如下:

$$\text{cont}(s_i, s_j) = \begin{cases} 1, & \Delta t(s_i, s_j) \leq t_{th}^{\text{cont}}, \\ & d(s_i, s_j) \leq d_{th}^{\text{cont}}; \\ 0, & \text{otherwise.} \end{cases} \quad (19)$$

定义候选集 R_j 中任意两个子轨迹线段 s_i 和 s_j 是否可以在时空上被视为连续, 当且仅当它们之间的时间间隔 $\Delta t(s_i, s_j)$ 和 s_i 终点到 s_j 起点的距离 $d(s_i, s_j)$ 均分别小于预设的时间间隔阈值 t_{th}^{cont} 和端点距离阈值 d_{th}^{cont} 时, 满足时空连续性约束, 记为 $\text{cont}(s_i, s_j) = 1$. 该约束确保只有在时间和空间上都足够接近子轨迹段才可能被连接起来, 从而构建符合运动规律的轨迹.

2.4 算法步骤

输入: 一组原始轨迹 $T = \{\text{Tr}_1, \text{Tr}_2, \dots, \text{Tr}_N\}$, 时空几何距离权重 w_{\perp} , w_{\parallel} 和 w_{θ} , 截断距离 d_c , 噪声识别中的近邻参数 k_{noise} , 局部特征距离阈值 d_{th} 和局部特征密度阈值 ρ_{th} , 目标类簇数量 K , 密度比例阈值 α , 时间间隔阈值 t_{th}^{cont} 和端点距离阈值 d_{th}^{cont} .

输出: 代表性轨迹 $RT = \{RT_1, RT_2, \dots, RT_K\}$.

step 1: 使用式 (1) ~ (7) 和式 (12) ~ (14) 对 T 中每一条轨迹 Tr 进行分段, 得到子轨迹段集合 $S = \{s_1, s_2, \dots, s_{\text{Total}}\}$.

step 2: 使用式 (15), 计算每个 $s_i \in S$ 的时空局部密度 ρ_i , 并使用式 (10) 和 (11) 计算相对距离 δ_i 和基础决策值 γ_i .

step 3: 使用式 (16), 对于 $s_i \in S$, 基于其 k 近邻的平均距离 \bar{d}_i 和平均密度 $\bar{\rho}_i$, 判断并标记其是否为噪声.

step 4: 选择 γ_j 最大且非噪声的 s_j 作为第 1 个中心, 加入类簇中心集合 C . 当类簇中心数小于目标类

簇数量 K 时, 对所有非噪声、非中心、未分配的候选段 s_j 使用式 (17) 计算时空因子决策值 $ST(\gamma_j)$, 选取最大 $ST(\gamma_j)$ 的候选段 s_{new} 加入 C .

step 5: 将剩余非噪声、非中心的子轨迹段分配到最近的类簇中心所属的类簇中, 得到 K 个子轨迹段类簇 $C = \{C_1, C_2, \dots, C_K\}$;

step 6: 使用式 (18) 筛选出高密度候选集 R_j , 在 R_j 中使用式 (19) 寻找代表性的连续子轨迹段 $S_j(s_1, s_2, \dots, s_m)$. 最后构建由 S_j 中所有子轨迹段的原始轨迹端点按顺序连接而成的代表性轨迹 RT_j 并添加到 RT 中.

2.5 时间复杂度分析

设 Tr_{num} 为输入总轨迹数, n 为单条轨迹最大点数, Num 为所有轨迹总点数, S 为分段后子轨迹段总数 ($S < Num$), K 为目标类簇中心数量 ($K \ll S$), k_{noise} 为噪声识别近邻数量.

时空加权分段遍历 Tr_{num} 条轨迹的 n 个点执行贪心比较, MDL 代价计算次数为 $O(n)$, 总复杂度为 $O(Num)$. 密度峰值子轨迹段聚类以 S 个子轨迹段为输入, 主要开销为计算 $S(S-1)/2$ 对时空几何距离, 复杂度为 $O(S^2)$; 局部密度 ρ_i 和相对距离 δ_i 计算也为 $O(S^2)$; 噪声识别为 $O(S \cdot k_{noise})$; 迭代中心选择为 $O(K^2 \cdot S)$; 分配为 $O(S \cdot K)$. 主导项为 $O(S^2)$. 代表性轨迹生成在 K 个类簇上进行高密度候选集筛选和时空连续性搜索, 平均复杂度为 $O(S^2/K)$, 在最差情况下为 $O(S^2)$. 由于本算法旨在分段捕捉轨迹的精确时空动态, 分段将产生大量子轨迹段. 尽管 $S < Num$, 但 S 的数量可使 S^2 的平方增长速度高于 Num 的线性增长. 因此, 第 2 阶段密度峰值子轨迹段聚类过程的 $O(S^2)$ 复杂度将远大于 $O(Num)$ 复杂度. 此外, 在最坏情况下, 第 3 阶段的代表性轨迹生成也可能达到 $O(S^2)$ 级别. 综合上述分析, 本算法的最终时间复杂度为 $O(Num) + O(S^2)$, 主要由 $O(S^2)$ 项主导.

3 实验与结果分析

3.1 实验环境与数据集介绍

本文实验的硬件环境为 Intel i5-13600KF 处理器与 32 GB 内存, 软件环境为 Windows 11 操作系统, 所有算法均基于 Python 3.10 实现.

为验证算法的有效性, 选用 T-Drive 和上海出租车两个真实轨迹数据集. T-Drive 记录了 2008 年 2 月 2 日 ~ 8 日北京 10 357 辆出租车约 1 500 万轨迹点; 上海数据集记录了 2007 年 2 月 20 日上海 4 316 辆出租车近 10 万 GPS 记录. 两数据集经统一预处理:

相邻点时间间隔不超过 300 s、空间距离不超过 2 000 m, 单条轨迹时间至少 4 h 且包含 50 个以上有效点. 预处理后详细信息如表 1 所示. 原始轨迹采用 WGS84 坐标系, 为便于距离计算, 转换为通用横轴墨卡托投影坐标系.

表1 数据集情况描述

相关信息	北京出租车轨迹	上海出租车轨迹
轨迹点数	36 148	37 251
轨迹条数	39	84
时间分布	2008 年 2 月 6 日(除夕)	2007 年 2 月 20 日(正月初三)
空间分布	116.16° ~ 116.60° 39.75° ~ 40.1°	121.40° ~ 121.55° 31.15° ~ 31.30°

3.2 算法效果评估

3.2.1 分段效果评估

为定量评估本文所提出时空加权分段相较于 TRACLUS 算法中 MDL 分段的有效性, 本节从几何保真度和语义一致性两个维度进行对比实验. 几何保真度采用重建误差 (reconstruction error, RE) 进行度量, 该指标计算原始轨迹点与其对应分段模型间的平均几何偏差, RE 值越低表明分段对原始轨迹的结构还原度越高, 其计算公式如下:

$$RE = \frac{1}{N} \sum_{i=1}^N \min_{s_j \in S} d(p_i, s_j). \quad (20)$$

其中: N 为原始轨迹点总数, S 为分割后的子轨迹线段集合, $d(p_i, s_j)$ 为点 p_i 到子轨迹线段 s_j 的最短几何距离.

语义一致性采用方向偏差 (direction deviation, DD) 进行评估, 该指标量化了各子轨迹段内部运动方向的连贯性, DD 值越低表明分段结果越能捕获具有单一运动意图的片段, 计算公式如下:

$$DD = \frac{1}{|S|} \sum_{s_j \in S} \left(\frac{1}{n_j - 1} \sum_{a=s_j}^{e_j-1} \arccos \left(\frac{\vec{V}_j \cdot \vec{v}_a}{|\vec{V}_j| \cdot |\vec{v}_a|} \right) \right), \quad (21)$$

其中 $|S|$ 为分段总数. 对于任意分段 $s_j \in S$, 它对应于原始轨迹点序列 $(p_{s_j}, p_{s_j+1}, \dots, p_{e_j})$, $n_j = e_j - s_j + 1$ 为构成该分段的原始点数量. 该分段的总体方向向量为 $\vec{V}_j = p_{e_j} - p_{s_j}$, 其内部的第 a 个步进向量为 $\vec{v}_a = p_{a+1} - p_a$, 其中 $s_j \leq a < e_j$.

时空加权分段中, 时空几何距离内部各分量权重根据数据特性, 设定为 $w_{\perp} = 0.4$, $w_{\parallel} = 0.3$ 和 $w_{\theta} = 0.3$. 表 2 展示了在两个数据集上原始分段与时空加权分段的指标对比.

如表 2 所示, 在北京和上海出租车数据集上, 时空加权分段使 RE 指标分别降低了 25.4% 和 30.9%,

表2 轨迹分段性能对比

数据集	评估指标	原始分段	时空加权分段	性能变化率/%
北京出租车轨迹	总轨迹数	39	39	—
	总分段数	12 432	16 336	↑31.4
	重建误差/m	0.63	0.47	↓25.4
	方向偏差/(°)	2.72	1.17	↓57.0
	处理时间/s	88.40	137.69	↑55.8
上海出租车轨迹	总轨迹数	84	84	—
	总分段数	18 497	21 826	↑18.0
	重建误差/m	0.081	0.056	↓30.9
	方向偏差/(°)	0.08	0.02	↓75.0
	处理时间/s	92.37	151.96	↑64.5

DD 指标降低了 57.0% 和 75.0%。结果表明, 尽管计算时间略有增加, 但本文方法在不同城市路网环境下均能大幅提升分段的几何保真度与语义一致性, 验证了其普适性与有效性。

3.2.2 聚类效果评估

为系统评估密度峰值子轨迹段聚类性能, 采用轮廓系数 (silhouette coefficient, SC)、Davies-Bouldin Index (DBI) 和 Calinski-Harabasz Index (CHI) 三种内部评估指标进行综合评估, 以期从不同维度对聚类结果的质量进行更全面地评估。

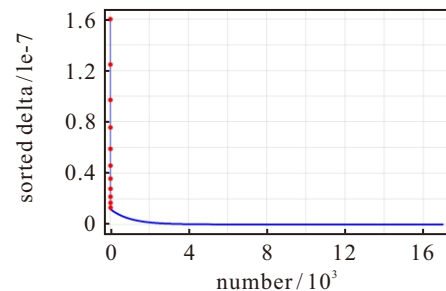
为确保实验对比的公平性, 在北京和上海两个出租车轨迹数据集上对所有算法进行统一的参数优化, 并以寻找最小 DBI 为目标。

对于传统聚类算法 K -means^[22]、DPC 和 AGNES (agglomerative nesting)^[23], 将其应用于 STW-DP-TRACLUS 时空加权分段生成的子轨迹线段集合进行聚类。 K -means、DPC 和 AGNES 算法的类簇数与密度峰值子轨迹段聚类保持一致, 且 DPC 算法的截断距离 d_c 在子轨迹间距离的 [0.1%, 5%] 分位数范围^[24]内网格搜索确定。

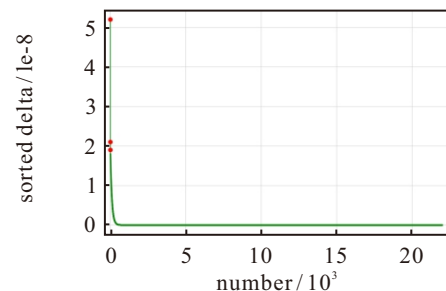
对于轨迹聚类算法, 在两个轨迹数据集上进行对比, 参数选择遵循其文献的推荐与实验配置。Tra-PDPC (trajectory parallel ordering points to identify clustering structure) 算法^[25]的 d_c 参数在距离 [0.1%, 5%] 分位数范围内寻优, 按照其论文建议将空间网格大小分别设置为 $g_{\max}^x=100$ 和 $g_{\max}^y=100$, 网格区域的最小点数阈值 $N_g=15$, 在组合相似度计算中, 欧氏空间距离权重 α 和网格空间距离权重 β 设定为 0.65 和 0.35, 距离函数的权重参数 w_{\perp} 、 w_{\parallel} 和 w_{θ} 均设为 0.33; STFC (spark-based trajectory fast clustering) 算法^[26]中距离函数的权重系数 w_{\perp} 、 w_{\parallel} 、 w_{θ} 和 w_s 均采用原论文建议设为 1, 近邻数 k 在 [2, 50] 区间内^[27]搜索,

类簇合并距离比率 ratio 设为 0.5, 转角阈值 th_{angle} 和速度阈值 th_{speed} 分别在 [60, 120] 和 [30, 80] 范围内优化, 聚类中心识别的密度分位数 th_{ρ} 和距离分位数 th_{delta} 均在 [0.5, 0.9] 区间内进行最优选择; TRACLUS 算法的邻域半径 eps 采用信息熵最小化方法确定, 最小样本数 \min_{mins} 和轨迹基数约束 \min_{tra} 取相同值, 并在 [2, 10] 整数区间内的网格搜索获得最优值; 距离函数的权重参数 w_{\perp} 、 w_{\parallel} 和 w_{θ} 均设为 1; Tra-POPTICS (trajectory parallel ordering points to identify clustering structure) 算法^[28]的最小样本数 \min_{mins} 在 [2, 10] 范围内寻优, 邻域半径 eps 取 k -距离分布 90% ~ 95% 百分位, 类簇提取距离 eps' 在 [0.25eps, eps] 范围内寻优。

密度峰值子轨迹段聚类的截断距离 d_c 在子轨迹间距离 [0.1%, 5%] 分位数范围内取值, 近邻参数 k_{noise} 在 [2, 50] 分位数范围内优化, 局部特征阈值分别取平均局部距离和平均局部密度的 80% 和 20% 分位数, 目标类簇数 K 由决策图确定。如图 3 所示, 北京出租车轨迹数据集中目标类簇数 $K=11$, 上海出租车轨迹数据集中目标类簇数 $K=3$ 。



(a) 北京出租车轨迹



(b) 上海出租车轨迹

图3 决策图

为确定关键参数 k_{noise} 和 d_c , 在各自的范围内进行参数敏感性分析, 其性能随参数变化的详细结果如图 4 所示。上图为 k_{noise} 对聚类性能的影响, 下图为 d_c 对聚类性能的影响。实验结果表明, 两个关键参数对算法性能的影响是可预测的, 其最优值可通过在合理范围内搜索有效确定, 表明密度峰值子轨迹段聚类具有良好的鲁棒性。本次实验中, 最优 k_{noise} 值可

分别选择为 15 和 21, 最优 d_c 值可分别取为 3 573.99 和 1 354.25.

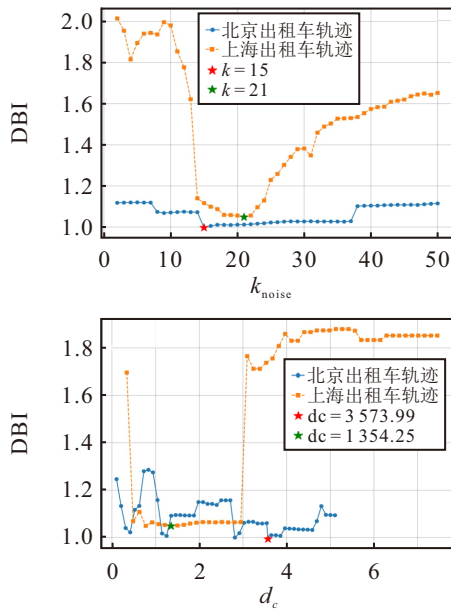


图4 关键参数对聚类性能的影响

为评估算法性能, 本文在两个真实轨迹数据集上进行综合对比实验, 详细结果列于表 3. 其中: 最优结果以加粗字体凸出, 最优参数用“Arg-”表示. 实验结果表明, 本文提出的 STW-DP-TRACLUS 算法在核心指标 DBI 上领先, 其在北京和上海数据集上的最优值分别为 0.994 和 1.048, 显示生成的类簇结构

最合理, 兼顾高内聚与低耦合. 相比之下, 基准算法在特定指标上的优势揭示了其局限性. 例如, K -means 虽在 CHI 指标上领先, 但这反映了该指标与算法对球形簇的共同偏好, 其高 DBI 值表明难以拟合复杂真实轨迹. Tra-PDPC 在北京出租车轨迹数据集上 SC 值最高, 但 DBI 较差, 暗示整体结构欠佳.

STW-DP-TRACLUS 通过融合时空加权分段与密度峰值聚类, 在复杂数据上展现出均衡稳健性能, 验证了其设计有效性, 并能够为交通模式挖掘提供可靠支持.

3.2.3 代表性轨迹分析

在代表性轨迹生成阶段, 为确保轨迹的时空连续性与核心特征, 相关参数设定如下: 时间阈值 t_{th}^{cont} 为 600 s, 允许短时间的衔接间断; 端点距离阈值 d_{th}^{cont} 根据城市路网特性和轨迹采样密度设为 1000 m; 密度比例阈值 α 为 0.9, 以聚焦最具代表性的高密度子轨迹段.

为深入理解本算法聚类效果所揭示的城市交通模式, 对北京和上海出租车轨迹数据集生成的代表性轨迹进行时空特征分析. 在北京数据集中, 算法生成 11 条代表性轨迹; 在上海数据集中, 则生成 3 条. 图 5 与图 6 分别展示了这些轨迹的空间分布, 其详细时空特征信息归纳于表 4.

表3 不同聚类算法的性能对比

数据集	算法	Arg-	DBI	SC	CHI
北京出租车轨迹	K -means	$K = 11$	2.132	0.331	26 630.632
	DPC	$K = 11, d_c = 2 802.61$	1.839	0.165	1 532.298
	AGENS	$K = 11$	2.183	0.259	942.893
	Tra-PDPC	$d_c = 1 607.12$	1.035	0.537	4 243.922
	STFC	$k = 21, th_{angle} = 120, th_{speed} = 60, th_{delta} = 0.7$	1.340	0.129	4 405.218
	TRACLUS	$eps = 32.4, min_{ins} = 6, min_{tra} = 6$	1.072	0.226	2 167.103
	Tra-POPTICS	$eps = 740.36, min_{ins} = 6, eps' = 130.44$	1.512	0.177	1 444.859
	STW-DP-TRACLUS	$K = 11, d_c = 3 573.99, k_{noise} = 15$	0.994	0.434	4 599.642
上海出租车轨迹	K -means	$K = 3$	1.640	0.060	24 774.314
	DPC	$K = 3, d_c = 1 773.95$	2.768	0.166	1 532.298
	AGENS	$K = 3$	1.907	0.034	281.772
	Tra-PDPC	$d_c = 1 383.13$	1.352	0.008	630.543
	STFC	$k = 11, th_{angle} = 100, th_{speed} = 75, th_{delta} = 0.8$	1.916	0.213	1 268.793
	TRACLUS	$eps = 212.2, min_{ins} = 9, min_{tra} = 9$	1.292	0.384	3 127.415
	Tra-POPTICS	$eps = 1 486.02, min_{ins} = 5, eps' = 1 238.39$	2.779	0.030	1 349.699
	STW-DP-TRACLUS	$K = 3, d_c = 1 354.25, k_{noise} = 21$	1.048	0.449	4 917.156

表4 代表性轨迹基本信息

数据集	轨迹名称	起点坐标/(°)	终点坐标/(°)	距离/km	时间段	关联区域
北京出租车轨迹	T_1	(116.381, 39.853)	(116.380, 39.854)	0.15	12:45:45 ~ 12:59:15	马家堡西路
	T_2	(116.384, 39.864)	(116.390, 39.864)	0.50	07:45:56 ~ 09:36:36	南三环西路
	T_3	(116.421, 39.908)	(116.431, 39.902)	2.89	13:00:45 ~ 13:25:45	崇文门外大街
	T_4	(116.328, 39.883)	(116.323, 39.895)	11.16	02:45:07 ~ 04:50:40	北京西站至马连道村公共汽车站
	T_5	(116.536, 39.755)	(116.537, 39.756)	0.12	02:10:58 ~ 05:57:01	新风河路
	T_6	(116.329, 39.903)	(116.334, 39.896)	1.30	16:01:45 ~ 16:05:12	北京西站
	T_7	(116.378, 39.837)	(116.380, 39.839)	0.93	20:55:04 ~ 21:00:35	新发地长途客运站
	T_8	(116.454, 39.892)	(116.455, 39.893)	0.48	17:40:27 ~ 17:44:48	和平村
	T_9	(116.393, 39.864)	(116.395, 39.864)	0.20	02:11:00 ~ 05:39:13	南三环中路
	T_{10}	(116.285, 39.899)	(116.330, 39.888)	9.73	19:16:40 ~ 19:51:09	莲花池西路至北京西站南路
	T_{11}	(116.383, 39.855)	(116.378, 39.855)	0.42	11:02:32 ~ 11:03:02	角门北路
上海出租车轨迹	T_1	(121.546, 31.291)	(121.400, 31.152)	21.93	08:13:24 ~ 17:36:48	杨浦区至闵行区
	T_2	(121.405, 31.210)	(121.407, 31.211)	0.26	08:00:12 ~ 08:20:03	娄山关路地铁站
	T_3	(121.410, 31.199)	(121.407, 31.210)	3.71	14:35:36 ~ 15:46:59	长宁区

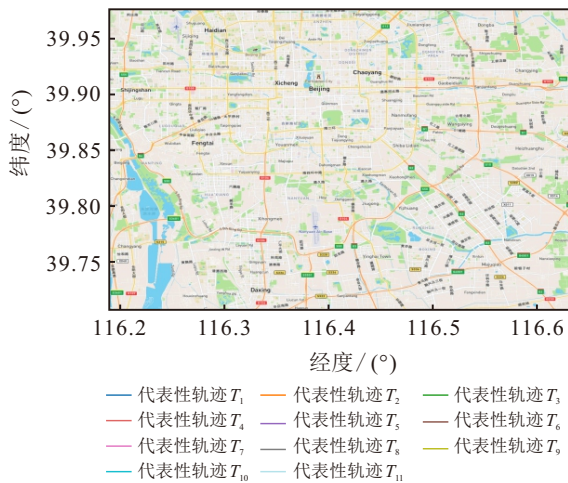


图5 北京出租车代表性轨迹分布情况

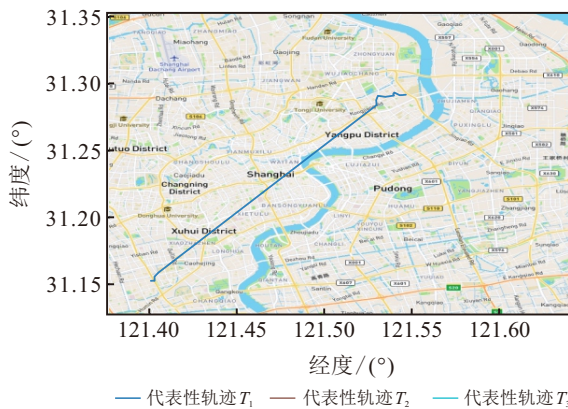


图6 上海出租车代表性轨迹分布情况

北京代表性轨迹分析揭示了除夕期间出行模式的高度多样性,包括围绕大型交通枢纽的集散服务、空间位移极小但持续数小时的候客行为,以及连接

核心商业区的功能性出行.相比之下,上海代表性轨迹展现出不同尺度特征,涵盖跨行政区的长距离通行、轨道交通站点附近的高频短距离接驳,以及城区内部的中等距离本地出行.

两地结果对比显示,北京模式更为复杂,而上海呈现更大空间跨度.这一差异可能源于城市空间结构、节日特点及出租车运营模式的差异.这些发现证实,STW-DP-TRACLUS算法能有效挖掘并区分异构时空模式,其代表性轨迹能够为理解城市交通系统运行特征并优化资源配置提供精确数据洞察.

4 结论

针对TRUCLUS算法在时空动态感知不足及非均匀密度与噪声处理方面的局限性,本文提出了一种基于时空加权和密度峰值的轨迹聚类算法.时空加权分段将MDL代价函数应用到时空维度,对轨迹实现更精确地分段;密度峰值子轨迹段聚类集成时空局部密度、局部特征的噪声识别以及迭代式的中心选择,增强聚类的时空适应性鲁棒性;代表性轨迹生成阶段,提取子轨迹段类簇的核心运动模式并结合可视化手段直观呈现,增强算法的可解释性.尽管STW-DP-TRACLUS算法在挖掘质量上表现出优势,但其计算效率仍是主要挑战.算法的性能瓶颈主要在于第2阶段计算子轨迹段距离矩阵时产生的 $O(S^2)$ 复杂度,限制了其在大规模轨迹数据上的应用.未来工作的重点将通过算法优化^[29-30],进一步提升其运行效率和可扩展性.

参考文献 (References)

- [1] Fadhel M A, Duhaim A M, Saihood A, et al. Comprehensive systematic review of information fusion methods in smart cities and urban environments[J]. *Information Fusion*, 2024, 107: 102317.
- [2] 王康, 李琼琼, 王子洋, 等. 考虑侧倾的无人车 NMPC 轨迹跟踪控制[J]. *控制与决策*, 2022, 37(10): 2535-2542.
(Wang K, Li Q Q, Wang Z Y, et al. Trajectory tracking control for automated vehicle based on NMPC considering vehicle rolling motion[J]. *Control and Decision*, 2022, 37(10): 2535-2542.)
- [3] 宋逸哲, 刘妹琴, 董山玲, 等. 弱通信条件下多水下自主航行器分布式编队轨迹规划[J]. *控制与决策*, 2025, 40(1): 71-79.
(Song Y Z, Liu M Q, Dong S L, et al. Distributed formation trajectory planning method for multiple autonomous underwater vehicles with limited communication[J]. *Control and Decision*, 2025, 40(1): 71-79.)
- [4] 郭戈, 胡峻豪. 区别多种出行方式的城市活动轨迹预测[J]. *控制与决策*, 2023, 38(4): 1022-1030.
(Guo G, Hu J H. Urban activity trajectory prediction with different travel modes[J]. *Control and Decision*, 2023, 38(4): 1022-1030.)
- [5] Gong X R, Huang Z, Wang Y L, et al. High-performance spatiotemporal trajectory matching across heterogeneous data sources[J]. *Future Generation Computer Systems*, 2020, 105: 148-161.
- [6] 江浩斌, 路保松, 李傲雪. 汽车行驶轨迹数据密度峰值聚类算法研究[J]. *汽车工程*, 2023, 45(7): 1153-1162.
(Jiang H B, Lu B S, Li A X. Study on density peaks clustering algorithm of vehicle trajectory data[J]. *Automotive Engineering*, 2023, 45(7): 1153-1162.)
- [7] Ansari M Y, Ahmad A, Khan S S, et al. Spatiotemporal clustering: A review[J]. *Artificial Intelligence Review*, 2020, 53(4): 2381-2423.
- [8] 刘敬一, 彭举, 唐建波, 等. 融合多特征的轨迹数据自适应聚类方法[J]. *地球信息科学学报*, 2023, 25(7): 1363-1377.
(Liu J Y, Peng J, Tang J B, et al. An automatic trajectory clustering method integrating multiple features[J]. *Journal of Geo-information Science*, 2023, 25(7): 1363-1377.)
- [9] Boroumand F, Alesheikh A A, Sharif M, et al. FLCSS: A fuzzy-based longest common subsequence method for uncertainty management in trajectory similarity measures[J]. *Transactions in GIS*, 2022, 26(5): 2244-2262.
- [10] Bai X G, Xie Z X, Xu X F, et al. An adaptive threshold fast DBSCAN algorithm with preserved trajectory feature points for vessel trajectory clustering[J]. *Ocean Engineering*, 2023, 280: 114930.
- [11] Houssou N L J, Guillaume J L, Prigent A. Edit distance with quasi real penalties: A hybrid distance for network-constrained trajectories[C]. *IEEE International Conference on Data Mining Workshops*. Orlando, 2022: 1045-1053.
- [12] Ester M, Kriegel H P, Sander J, et al. A density-based algorithm for discovering clusters in large spatial data bases with noise[C]. *Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining*. Portland: AAAI, 1996: 226-231.
- [13] Bao L W, Liu Z L, Miao R, et al. Spatiotemporal clustering analysis of shared electric vehicles based on trajectory data for sustainable urban governance[J]. *Journal of Cleaner Production*, 2023, 412: 137373.
- [14] Yang Y Q, Cai J H, Yang H F, et al. TAD: A trajectory clustering algorithm based on spatial-temporal density analysis[J]. *Expert Systems with Applications*, 2020, 139: 112846.
- [15] 赵嘉, 陈蔚昌, 肖人彬, 等. 面向流形数据的共享近邻和二阶 K 近邻密度峰值聚类算法[J]. *控制理论与应用*, <http://kns.cnki.net/kcms/detail/44.1240.tp.20240830.1612.081.html>.
(Zhao J, Chen W C, Xiao R B, et al. Density peaks clustering algorithm based on shared nearest neighbor and second-order K nearest neighbor for manifold data[J]. *Control Theory & Applications*, <http://kns.cnki.net/kcms/detail/44.1240.tp.20240830.1612.081.html>.)
- [16] Jiang Q W, Liu Y, Ding Z R, et al. Behavior pattern mining based on spatiotemporal trajectory multidimensional information fusion[J]. *Chinese Journal of Aeronautics*, 2023, 36(4): 387-399.
- [17] 钟超, 刘漫丹, 贺帆. 基于改进密度峰值算法的轨迹聚类[J]. *计算机工程与设计*, 2024, 45(1): 130-138.
(Zhong C, Liu M D, He F. Trajectory clustering based on improved density peak algorithm[J]. *Computer Engineering and Design*, 2024, 45(1): 130-138.)
- [18] Wang X, Wu Z X, Jin B, et al. MDSTF: A multi-dimensional spatio-temporal feature fusion trajectory prediction model for autonomous driving[J]. *Complex & Intelligent Systems*, 2024, 10(5): 6647-6665.
- [19] Lee J G, Han J W, Whang K Y. Trajectory clustering: A partition-and-group framework[C]. *Proceedings of the 2007 ACM SIGMOD International Conference on Management of Data*. Beijing, 2007: 593-604.
- [20] 赵嘉, 王刚, 吕莉, 等. 面向流形数据的测地距离与余弦互逆近邻密度峰值聚类算法[J]. *电子学报*, 2022, 50(11): 2730-2737.
(Zhao J, Wang G, Lv L, et al. Density peaks clustering algorithm based on geodesic distance and cosine mutual reverse nearest neighbors for manifold datasets[J]. *Acta Electronica Sinica*, 2022, 50(11): 2730-2737.)

- [21] Zhao J, Wang G, Pan J S, et al. Density peaks clustering algorithm based on fuzzy and weighted shared neighbor for uneven density datasets[J]. *Pattern Recognition*, 2023, 139: 109406.
- [22] Hartigan J A, Wong M A. Algorithm AS 136: A K -means clustering algorithm[J]. *Applied Statistics*, 1979, 28(1): 100.
- [23] Kaufman L, Rousseeuw P J. Finding groups in data: An introduction to cluster analysis[M]. New York: John Wiley & Sons, 1990: 199-208.
- [24] 吕莉, 赵妞, 肖人彬, 等. 混合近邻和多簇合并的密度峰值聚类算法[J]. *控制与决策*, 2025, 40(7): 2194-2202. (Lv L, Zhao N, Xiao R B, et al. Density peak clustering algorithm with mixed nearest neighbors and multi-cluster merging[J]. *Control and Decision*, 2025, 40(7): 2194-2202.)
- [25] Niu X Z, Zheng Y H, Fournier-Viger P, et al. Parallel grid-based density peak clustering of big trajectory data[J]. *Applied Intelligence*, 2022, 52(15): 17042-17057.
- [26] Wang N, Gao S, Peng X W, et al. Research on fast and parallel clustering method for trajectory data[C]. IEEE 24th International Conference on Parallel and Distributed Systems. Singapore, 2018: 252-258.
- [27] 赵嘉, 姚占峰, 吕莉, 等. 基于相互邻近度的密度峰值聚类算法[J]. *控制与决策*, 2021, 36(3): 543-552. (Zhao J, Yao Z F, Lv L, et al. Density peaks clustering based on mutual neighbor degree[J]. *Control and Decision*, 2021, 36(3): 543-552.)
- [28] Deng Z, Hu Y Y, Zhu M, et al. A scalable and fast OPTICS for clustering trajectory big data[J]. *Cluster Computing*, 2015, 18(2): 549-562.
- [29] 张曦, 李璠, 付雪峰, 等. 随机学习萤火虫算法优化的模糊软子空间聚类算法[J]. *江西师范大学学报: 自然科学版*, 2021, 45(2): 137-144. (Zhang X, Li F, Fu X F, et al. The fuzzy soft subspace clustering algorithm optimized by random learning fire fly algorithm[J]. *Journal of Jiangxi Normal University: Natural Science Edition*, 2021, 45(2): 137-144.)
- [30] 范强, 吕莉, 邱日轩, 等. 反距离加权和微簇合并的密度峰值聚类算法[J]. *南昌工程学院学报*, 2024, 43(6): 43-50. (Fan Q, Lv L, Qiu R X, et al. Density peaks clustering with inverse distance weighting and micro cluster merging[J]. *Journal of Nanchang Institute of Technology*, 2024, 43(6): 43-50.)

作者简介

赵嘉 (1981-), 男, 教授, 博士, 主要研究方向为机器学习、数据挖掘、智能计算, E-mail: zhaojia@nit.edu.cn;

段发祥 (2000-), 男, 硕士生, 主要研究方向为数据挖掘, E-mail: 2023313022@nit.edu.cn;

潘正祥 (1961-), 男, 教授, 博士, 主要研究方向为人工智能、智能计算, E-mail: jspan@ieee.org;

王奔 (1985-), 男, 高级工程师, 硕士, 主要研究方向为电力信息通信、人工智能, E-mail: 19603595@qq.com;

张翼英 (1973-), 男, 教授, 博士, 主要研究方向为物联网及其安全、智能电网、大数据、安全密钥管理和车联网, E-mail: yiyinzhang@tust.edu.cn;

于华东 (1977-), 男, 高级工程师, 硕士, 主要研究方向为自动化, E-mail: 12551008@qq.com;

吴润秀 (1971-), 女, 教授, 硕士, 主要研究方向为群智能算法及应用, E-mail: wrx@nit.edu.cn.