

基于速度的轨迹停留点识别算法^①



蔡小路, 曹 阳, 董 蒲

(华南师范大学 计算机学院, 广州 510631)

通讯作者: 曹 阳, E-mail: C_EXE@126.com

摘 要: 轨迹中的停留点识别是将空间轨迹转换为语义轨迹的关键步骤. 当前轨迹停留点识别方法缺少对轨迹记录点时间连续性的考虑, 导致识别出的停留点缺乏时间信息. 同时, 在轨迹点缺失的情况下, 停留点信息也无法被准确识别. 针对上述问题, 本文提出一种基于速度的时空聚类方法, 首先通过缺失轨迹的时空特性确定真实缺失子轨迹, 并根据缺失轨迹的平均速度对其进行插值填充, 再结合轨迹速度特征和时空特性识别轨迹中的停留点. 实验采用 GeoLife 轨迹数据集对所提出的方法进行验证, 结果表明, 该算法能够有效地识别用户的停留点, 并对轨迹中的干扰具有一定的鲁棒性.

关键词: 时空轨迹; 语义轨迹; 时空聚类; 停留点

引用格式: 蔡小路, 曹阳, 董蒲. 基于速度的轨迹停留点识别算法. 计算机系统应用, 2020, 29(4): 214-219. <http://www.c-s-a.org.cn/1003-3254/7367.html>

Velocity-Based Trajectory Stop Point Recognition Algorithm

CAI Xiao-Lu, CAO Yang, DONG Pu

(School of Computer Science, South China Normal University, Guangzhou 510631, China)

Abstract: The recognition of stop point in trajectory is the key step to transform spatial trajectory into semantic trajectory. At present, the recognition method of the stop point lacks the consideration of the time continuity of the record point, which leads to the lack of time information of the identified stop point. At the same time, in the scenario of missing track points, the information of stop point cannot be accurately identified. In order to solve these problems, this study proposes a velocity based spatio-temporal clustering method. Firstly, the real missing sub trajectories are determined by the spatio-temporal characteristics of the missing trajectories, and then the missing trajectories are interpolated according to the average velocity of the missing trajectories. In the experiment, GeoLife trajectory data is used to verify the proposed method. The results show that the algorithm can effectively identify the user's stop point, and has relatively sound robustness to the interference in the trajectory.

Key words: spatio-temporal trajectory; semantic trajectory; spatial-temporal cluster; stop point

随着带有定位模块的智能终端使用量的增加, 移动对象位置数据的获取变得越来越普遍. 而轨迹数据能够反映移动对象的部分隐藏信息^[1], 通过识别轨迹中的隐藏信息加入语义信息, 可以得到体积较小、质量较高、便于理解的带有语义信息的轨迹. 将原始轨迹

转换为语义轨迹为目的位置预测^[2]、城市道路的规划^[3]、充电桩分布的规划^[4]、动物习性的调查^[5]等提供了解决方法. 因此, 对轨迹数据进行语义转换的需求将会变得越来越大. Alvares, Bogorny 等人^[6]提出了一种将原始轨迹转换为语义轨迹的模型, 模型得到了广泛

① 基金项目: 广东省省级科技计划 (2017A070713027)

Foundation item: Science and Technology Plan of Guangdong Province (2017A070713027)

收稿时间: 2019-09-10; 修改时间: 2019-10-10; 采用时间: 2019-10-24; csa 在线出版时间: 2020-04-05

的应用^[7-9]。模型先将物体的移动路径切分为“停留点(stop)”和“移动点(move)”两部分,再将停留点与地理信息库相匹配为停留点附上地理背景信息,然后将这些带有地理信息的停留点按照时间先后顺序排列后即为语义轨迹。因此,原始轨迹转换为语义轨迹的关键是如何准确识别出原始轨迹中的停留点或停留域。

目前关于轨迹停留点识别的算法大致可以分为两大类:静态方法和动态方法^[10]。Alvares, Bogorny 等人^[11]提出挖掘原始的轨迹数据中的语义信息是理解这些数据的前提,认为停靠点是轨迹集中具有重要意义的子序列集合,将轨迹与预先定义好的重要位置的交集视为停靠点。静态算法的主要缺点是用户需要指定有意义的位置。因此,如果事先没有定义一些有意义的特殊位置,静态方法就无法识别出停留点信息^[12]。而动态停留点识别方法则不需要事先知道停留点或停留域位置,也可以发现停留点位置。Ashbrook, Starner 等人^[13]对传统的 K-Means 算法进行改进来提取出轨迹中有意义的点。方法要求首先选取一个初始中心点并定义覆盖半径,将中心点覆盖半径内的点进行标记并计算这些点的均值点。将均值点作为新的中心重复上述操作,直到中心点不再变化,将上述标记的点视为一个停留域。然后从未被标记的点中选择一个初始中心点进行新的停留域的筛选,直到所有点都被标记。K-Means 算法的 K 值和初始中心点的选取是重要问题,选取不当会导致最终的结果产生很大的误差。Zhou, Frankowski 等人^[14]提出了名为 DJ-Cluster 的基于密度和连接的算法,算法核心思想是:计算每一个点的一定距离(EPS)内的邻域点。当点的数量小于 MinPts 时将其标记为噪声。邻域点数目大于 MinPts 时,邻域不属于任何一个已知停留点类时则将其标记为一个新的停留点类,反之合并到所属类中。基于密度聚类的停留点识别算法可以克服 K-Means 算法的缺陷,但是仅考虑了轨迹的空间特征而忽略了时序性。Hwang, Evans^[15]提出了一种“自上而下”的光栅采样方法,该方法采用预先定义的 GPS 采集器进行数据收集,将具有显著区分值的光栅单元作为停留点进行聚类。Palma, Bogorny^[16]提出了 CB-SMoT(基于聚类的停止和轨迹移动)算法来提取已知和未知的停留点。CB-SMoT 算法是一种考虑了时间和空间特征的基于密度的聚类算法,通过选取速度比速度阈值慢的点来生成停留点。Zhao, Xu^[17]指出^[16]中估计参数 Eps 适当值的分位数函数并不能稳定地确定

参数的适当阈值。因此提出了一种改进的 CB-SMoT 算法,提出了一种计算 EPS 参数的方法,但由于需要用户区分低速部分和高速部分,计算起来仍然困难。Yan, Parent^[18]在 CB-SMoT 的基础上加入最短停留时间参数来过滤掉“伪停留点”(例如,短期低速拥堵不应是一个停留点)。Lv, Chen^[19]提出了一种层次聚类算法,首先从 GPS 轨迹中提取访问点,然后对这些访问点进行聚类形成停留域。Chen, Ji^[20]针对 GPS 点的时间序列特性,提出了一种改进的基于密度的聚类算法 T-DBSCAN,并定义了两个新的前提条件(单停留域内的状态连续性和停留点间的时间间断性)作为调整聚类中轨迹点选择的理论基础。Yang, Li^[21]将 Eps 坐标系划分为网格单元。通过遍历数据集,每个数据点根据其经度、纬度和时间分量被分配到相应的网格单元。再计算每个网格单元内的数据点数量,将网格点数大于 MinPts 的单元标记为核心网格单元。从核心网格单元出发,采用广度优先遍历策略搜索其相邻的核心网格,并将最大的相邻核心网格单元集标记为停留点簇。

然而上述方法无法有效地分辨出移动目标是从建筑物穿过还是在建筑内有过停留这一问题,也不能合理地设置时间阈值来聚类缺失记录点。当移动目标于某一地理位置产生活动时,速度会低于整条轨迹的平均速度。针对这一物理特性和前文论述的研究缺陷,本文提出了一种改进的停留点识别框架:先确定轨迹中的缺失子轨迹,再按缺失轨迹的平均速度对缺失轨迹进行插值,然后采用基于速度的时空聚类方法来识别轨迹中的停留点。

1 基于速度特征的轨迹停留点识别算法

1.1 轨迹预处理

由于移动目标物体会在不同的场景中移动,场景的不同导致采集到的轨迹数据特点也会不同。当行人进入建筑物内时,因为信号受到遮挡,无法收集到理想的数据。用户轨迹进入建筑物可以分为两种情况:在建筑物内有停留和穿过建筑物。停留点识别算法应正确地识别出前者而过滤掉后者。当距离一定时,停留时间越长,该轨迹段内的按时记录的点的密度越大;而对于穿过建筑物情景而言,用户的移动速度不存在剧烈变化。

因此针对前文描述的数据缺失或偏差等情况,采用插值法对缺失轨迹的记录点进行填充:将缺失轨迹

前后两端点的时间差作为缺失时间, 两点间的距离作为缺失轨迹的缺失距离. 因为单体建筑范围一般小于 $200 \times 200 \text{ m}^2$, 且为了防止前条轨迹结束点与后一轨迹的起始点小于设定的缺失距离而导致两条轨迹因为插值导致合并造成最终的提取结果产生偏差, 所以对缺失轨迹通过缺失时间的大小来进行区别. 当缺失轨迹的缺失距离小于 200 m, 缺失时间小于 2 小时, 缺失轨迹的前后两条轨迹视为同一条轨迹. 将缺失时间与轨迹记录点时间间隔的比值视为缺失点的数目, 将缺失距离按缺失数目等距插值填充, 然后采用卡尔曼滤波器对填充后的整条轨迹进行平滑处理得到最终的实验数据集.

1.2 停靠点识别算法

本文在 ST-DBSCAN 算法的基础上提出了基于轨迹速度的密度聚类算法 (V-DBSCAN, Velocity-Based DBSCAN). 对于经过预处理后的轨迹点, 采用以下步骤进行处理: 1) 使用时间阈值和速度阈值对实际轨迹点进行筛选, 生成候选停留点邻域. 2) 沿候选停留点邻域的时间轴将其相邻点数大于密度阈值的点进行聚类, 得到最终停留点. 具体算法见算法 1.

算法 1. 基于速度的密度聚类停靠点识别算法

```

1) TrajList = traj_split(AllTrajectory)
2) For Traj in TrajList:
3) Eps = 半径阈值;
4) Mspeed = 平均速度;
5) clusterID = 0;
6) For  $p_i$  in Traj:
7) If  $p_i$  未被标记:
8) 计算  $p_i$  点的邻域;
9) If  $p_i$  邻域点数小于密度阈值:
10)  $p_i \leftarrow \text{noise}$ ;
11) Else:
12) clusterID = clusterID + 1;
13)  $p_i$  邻域点用 clusterID 标记;
14)  $p_i$  邻域点入栈;
15) While 栈非空:
16)  $p_k = \text{stack.pop}()$ ;
17) 计算  $p_k$  点的邻域;
18) If  $p_k$  邻域点数大于密度阈值:
19)  $p_k$  邻域点用 clusterID 标记;
20)  $p_k$  邻域入栈;
21) Else:
22)  $p_k \leftarrow \text{noise}$ ;
22) Return Traj;

```

如算法 1 所示, 算法首先根据轨迹点的加速度、

相邻轨迹速度范围和速度持续时间等特征将轨迹切成速度相近的子轨迹段, 再依次对子轨迹进行停留点识别. 识别算法首先计算子轨迹点集合 *Traj* 的半径阈值 Eps (算法详情见算法 2), 再计算 *Traj* 的整体平均速度 (*Mspeed*). *Traj* 中所有点的初始状态为未标记状态, 从起始记录点开始, 按记录点时间顺序依次对每一个点进行筛选过滤. 如果当前点 (p_i) 处于未标记状态, 则调用 *retrieve_neighbor* 函数, *retrieve_neighbors* 函数根据 Eps 、*T* 和 *Mspeed* 三个条件阈值, 从 *Traj* 中筛选出 p_i 的所有满足条件的相邻点. 如算法 3 所示, 函数依次根据时间、距离和速度三个条件筛选出 p_i 半径阈值 (Eps) 范围内速度小于 $Mspeed * e$ 且与 p_i 的时间间隔小于时间阈值 (*T*) 的轨迹点. 结果集则是 p_i 的候选停留点邻域. 若 p_i 的候选停留点邻域中轨迹点的数量小于 $minP_s$, 将 p_i 标记为噪点, 反之则将 p_i 与其邻域用类 ID 进行标记. 然后再按时间顺序迭代地对 p_i 邻域中未被标记的点进行筛选, 直到 *Traj* 中所有点都被标记.

计算 *Traj* 空间半径阈值 Eps 的函数 *spatial_threshold* 如算法 2 所示. 本方法是受文献[22]提出的 RNN-DBSCAN 算法启发而得出的, 在 RNN-DBSCAN 算法中通过计算每个目标的 *k* 最近邻近点, 建立所有目标的 *k* 邻近点索引表来进行密度聚类. 在 *spatial_threshold* 中, 先求出每一个点与其他任一点的地球球面距离, 筛选距离最短的 *k* 个值放入 *all_kth_distances* 中, 再求得 *all_kth_distances* 的平均值, 返回均值.

算法 2. 计算轨迹半径阈值 (*spatial_threshold*) 函数

```

1) allkth_distances = [];
2) k = Traj 长度开方;
3) For  $p_i$  in Traj:
4) distances = [];
5) For  $p_j$  in Traj:
6) If  $i \neq j$ :
7)  $d = \text{get\_distance}(p_i, p_j)$ ;
8) distances.append(d);
9) distances.sort();
10) knearest_distances = distances[:k];
11) allkth_distances.add(knearest_distances);
12) eps = avg(allkth_distances);
13) Return eps;

```

从原始轨迹点中筛选候选停留点集的过程为: 对连续的两个时刻内的运动状态进行判断, 当 p_k 与 p_i 的时间差小于时间阈值 *T*, 则对两点间的距离进行判断,

若距离小于半径阈值,且 p_k 的速度小于速度阈值 $Mspeed * e$ ($0 < e < 1$),则将 p_k 添加到 p_i 的邻域点集中,并计算下一点是否为 p_i 的邻域点,直到所有点筛选完毕.具体方法如算法3所示.

算法3. 筛选邻域点函数 (retrieve_neighbors)

```

1) neighborhood = []
2) df = points(time_diff(pi,pk) ≤ T)
3) For p in df:
4)   If not pi:
5)     d = get_distance(pi, p);
6)     If d ≤ Ep:
7)       If V(p) ≤ Mspeed * e:
8)         neighborhood = neighborhood.append(p);
9) Return neighborhood;

```

2 实验分析

为了评估 V-DBSCAN 算法的有效性,本章节将其与 ST-DBSCAN 算法^[23]在提取的停留点的数目和准确度方面进行实验比较.实验数据集共两组:第一组数据集为 GeoLife 数据集中 user ID=20 的用户的所有轨迹数据,采样点间隔为 5 s,共 177 683 个记录点,用于验证算法的提取能力,下文称数据一;第二组是志愿者收集的带有停留点标记的轨迹数据,采样时间间隔为 2 s,共 14 479 个记录点,用于验证提取到的停留点的准确性,下文称数据二;两组数据集的属性都相同,包有时间戳、经纬度、海拔基本信息.轨迹数据基本信息如表1所示.

表1 轨迹数据点基本信息表

| 属性 | 数据示例 | 备注 |
|------|---------------------|-------------------------|
| Time | 2018-10-07 11:29:55 | 时间, yyyy-MM-dd HH:MM:SS |
| Lat | 39.984 886 | 纬度,精确到 8 位小数 |
| Lon | 116.318 417 | 经度,精确到 8 位小数点 |
| Ele | 24.044 | 海拔,精确到 3 位小数点 |

2.1 参数设定

为了防止不恰当的参数设定造成最终结果的偏差,本节探索参数对结果的影响,并确定最终的对比实验的参数.此处的实验数据集为另一志愿者室外步行状态下的轨迹点,时间间隔(t)为 2 s,共 12 797 个轨迹记录点.整条轨迹含 26 个人工记录的停留点.

如表2所示,经过多轮实验探索得出:时间阈值设为 30 倍的记录点时间间隔(例如间隔 2 s 记录一个点的情况下,时间阈值设为 60 s),密度阈值设为 1.5 倍的

时间阈值大小,即 45 倍的记录点时间间隔时,提取出的停留点数最接近实际情况.

表2 参数对结果的影响

| $T(*t)$ | 15 | | | 30 | | | 45 | | |
|------------|----|----|----|----|----|----|----|----|----|
| MinPts(*t) | 30 | 45 | 60 | 30 | 45 | 60 | 30 | 45 | 60 |
| Count | 16 | 7 | 3 | 41 | 26 | 15 | 46 | 28 | 15 |

2.2 实验结果

实验一测试两种算法在相同时间、半径阈值,各个密度阈值下,对轨迹中停留点的提取能力,采用数据一作为实验数据来进行对比验证.时间阈值设为 30 倍的记录时间间隔,密度阈值设为记录时间间隔的 45 倍,密度阈值下为 0.75 倍轨迹平均速度.从图1中可以看出相同的实验参数下,V-DBSCAN 算法提取的停留点数都要大于 ST-DBSCAN 算法.这是因为目标用户在实际移动中往往存在两个或多个停留点相隔较近的情况,此种情景下停留点间的移动点的数量会非常少.当 ST-DBSCAN 算法的半径阈值设置过大或密度阈值设置过小时,就会将停留点间的移动点也划分到停留点中,导致相近的两个或多个停留点被识别为一个停留点.针对停留点间的移动点的瞬时速度大于两边停留点的瞬时速度这一物理特征,V-DBSCAN 算法对轨迹点速度进行筛选,只有当轨迹点的速度小于设置的速度阈值时才将该点进行后续的筛选,因此可以对地理距离相隔较近的多个停留点进行很好的划分.因此在识别停留点的提取能力上 V-DBSCAN 算法要强于 ST-DBSCAN 算法.

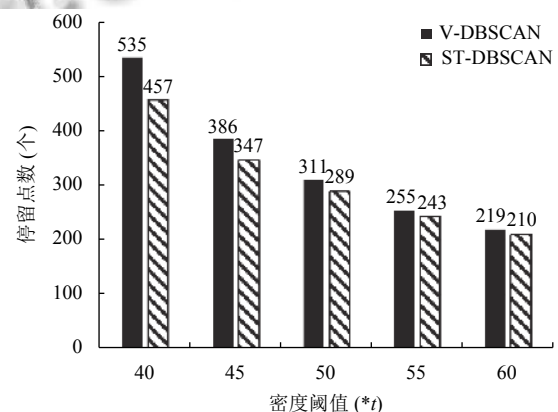


图1 停留点提取数量对比

实验二测试两种算法对轨迹中停留点的提取能力,采用数据二作为实验数据来进行对比验证($T=60$ s, $MinPts=90$, $e=0.75$).两者的对比实验结果如图2所示,

其中黑色大圆点为 V-DBSCAN 算法的结果, 白色小圆点为 ST-DBSCAN 算法的结果。



图2 停留点准确性对比

从图2中可以看出, 当仅考虑对停靠点的提取能力时, V-DBSCAN 相对于 ST-DBSCAN 提取出更多的停靠点, 例如图中 A、B 两点. 其中 A、B 两点的原始轨迹数据由于建筑物的遮挡而缺失, 所以未改进的 ST-DBSCAN 算法无法识别出这两停留点. 因为对于 ST-DBSCAN 算法的时间阈值 T 范围内可能存在多个停留点, 这是 ST-DBSCAN 算法参数不能基于轨迹特征来设定导致的缺陷. 为此本文中提出了基于速度的密度聚类方法, 在 ST-DBSCAN 算法的基础上考虑轨迹点速度来提取停留点, 极大改善了 ST-DBSCAN 算法对停留点提取不完全的问题。

3 总结

本文针对空间轨迹停留点提取过程中的轨迹点缺失和半径阈值难以确定等问题, 提出了基于轨迹点速度的密度聚类算法. 由于传统的基于密度的聚类方法将轨迹的时间和空间两个属性分开来进行考虑, 忽略了两者之间的联系. V-DBSCAN 的设计是通过在一定范围内的缺失轨迹点进行插值填充再根据速度和时间来筛选出轨迹中的停留点. 试验结果表明, 相对于 ST-DBSCAN, V-DBSCAN 一方面被证明具有更好的轨迹聚类能力. 另一方面, V-DBSCAN 可以自动计算出轨迹的 E_p 半径距离, 减少了人为设定参数的不确定性. 但是, 时间阈值和密度阈值仍然要人为设定. 在后续的研究中会考虑本算法对多用户的多条轨迹提取停靠点的处理能力, 并根据不同用户的停靠点计算多个用户间的用户相似性。

参考文献

1 Mazimpaka JD, Timpf S. Trajectory data mining: A review

of methods and applications. *Journal of Spatial Information Science*, 2016, 2016(13): 61–99.

2 Ying JJC, Lee WC, Weng TC, *et al.* Semantic trajectory mining for location prediction. *Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*. Chicago, IL, USA. 2011. 34–43. [doi: 10.1145/2093973.2093980]

3 Chawla S, Zheng Y, Hu JF. Inferring the root cause in road traffic anomalies. *Proceedings of the 2012 IEEE 12th International Conference on Data Mining*. Brussels, Belgium. 2012. 141–150. [doi: 10.1109/ICDM.2012.104]

4 Jirka S, Remke A, Bröring A. enviroCar—crowd sourced traffic and environment data for sustainable mobility. *Proceedings of Environmental Information Systems and Services-Infrastructures and Platforms*. Austria. 2013.

5 De Groeve J, Van de Weghe N, Ranc N, *et al.* Extracting spatio-temporal patterns in animal trajectories: An ecological application of sequence analysis methods. *Methods in Ecology and Evolution*, 2016, 7(3): 369–379. [doi: 10.1111/2041-210X.12453]

6 Alvares LO, Bogorny V, de Macedo JAF, *et al.* Dynamic modeling of trajectory patterns using data mining and reverse engineering. *Proceedings of Tutorials, Posters, Panels and Industrial Contributions at the 26th International Conference on Conceptual Modeling—Volume 83*. Auckland, New Zealand. 2007. 149–154.

7 Restrepo E, Moreno F, Lasheras FS. Identification of frequent touristic routes using the SMoT algorithm. *AIP Conference Proceedings*, 2013, 1558(1): 2325–2328.

8 Moreno FJ, Pineda AF, Fileto R, *et al.* SMOT+: Extending the SMOT algorithm for discovering stops in nested sites. *Computing and Informatics*, 2014, 33(2): 327–342.

9 Arboleda FJM, Bogorny V, Patiño H. SMOT+NCS: Algorithm for detecting non-continuous stops. *Computing and Informatics*, 2017, 36(2): 283–306. [doi: 10.4149/cai_2017_2_283]

10 Luo T, Zheng XW, Xu GL, *et al.* An improved DBSCAN algorithm to detect stops in individual trajectories. *ISPRS International Journal of Geo-Information*, 2017, 6(3): 63. [doi: 10.3390/ijgi6030063]

11 Alvares LO, Bogorny V, Kuijpers B, *et al.* A model for enriching trajectories with semantic geographical information. *Proceedings of the 15th Annual ACM International Symposium on Advances in Geographic Information Systems*. Seattle, WA, USA. 2007. Article No.22.

- 12 Xie KX, Deng K, Zhou XF. From trajectories to activities: A spatio-temporal join approach. Proceedings of the 2009 International Workshop on Location Based Social Networks. Seattle, WA, USA. 2009. 25–32.
- 13 Ashbrook D, Starner T. Using GPS to learn significant locations and predict movement across multiple users. Personal and Ubiquitous Computing, 2003, 7(5): 275–286. [doi: [10.1007/s00779-003-0240-0](https://doi.org/10.1007/s00779-003-0240-0)]
- 14 Zhou CQ, Frankowski D, Ludford P, *et al.* Discovering personally meaningful places: An interactive clustering approach. ACM Transactions on Information Systems (TOIS), 2007, 25(3): Article No. 12. [doi: [10.1145/1247715.1247718](https://doi.org/10.1145/1247715.1247718)]
- 15 Hwang S, Evans C, Hanke T. Detecting stop episodes from GPS trajectories with gaps. In: Thakuriah P, Tilahun N, Zellner M, eds. Seeing Cities Through Big Data. Cham: Springer, 2017. 427–439.
- 16 Palma AT, Bogorny V, Kuijpers B, *et al.* A clustering-based approach for discovering interesting places in trajectories. Proceedings of the 2008 ACM Symposium on Applied Computing. Fortaleza, Ceara, Brazil. 2008. 863–868.
- 17 Zhao XL, Xu WX. A clustering-based approach for discovering interesting places in a single trajectory. Proceedings of the 2009 Second International Conference on Intelligent Computation Technology and Automation. Changsha, China. 2009. 429–432. [doi: [10.1109/ICICTA.2009.569](https://doi.org/10.1109/ICICTA.2009.569)]
- 18 Yan ZX, Parent C, Spaccapietra S, *et al.* A hybrid model and computing platform for spatio-semantic trajectories. Proceedings of the 7th Extended Semantic Web Conference on the Semantic Web: Research and Applications. Heraklion, Crete. 2010. 60–75.
- 19 Lv MQ, Chen L, Xu ZX, *et al.* The discovery of personally semantic places based on trajectory data mining. Neurocomputing, 2016, 173: 1142–1153. [doi: [10.1016/j.neucom.2015.08.071](https://doi.org/10.1016/j.neucom.2015.08.071)]
- 20 Chen W, Ji MH, Wang JM. T-DBSCAN: A spatiotemporal density clustering for GPS trajectory segmentation. International Journal of Online and Biomedical Engineering (iJOE), 2014, 10(6): 19. [doi: [10.3991/ijoe.v10i6.3881](https://doi.org/10.3991/ijoe.v10i6.3881)]
- 21 Yang YL, Li LD, Tang JC, *et al.* The PGST-DBSCAN algorithm for mining clusters from massive spatial-temporal data. Proceedings of 2nd International Conference on Artificial Intelligence and Engineering Applications. Guilin, China. 2017. 67–87.
- 22 Bryant A, Cios K. RNN-DBSCAN: A density-based clustering algorithm using reverse nearest neighbor density estimates. IEEE Transactions on Knowledge and Data Engineering, 2018, 30(6): 1109–1121. [doi: [10.1109/TKDE.2017.2787640](https://doi.org/10.1109/TKDE.2017.2787640)]
- 23 Birant D, Kut A. ST-DBSCAN: An algorithm for clustering spatial-temporal data. Data & Knowledge Engineering, 2007, 60(1): 208–221.