|  |  |
| --- | --- |
| 算法1: 基于局部方向中心性度量的分簇机制 | |
|  | 输入: 轨迹数据集  *:* 寻找最近邻点的数量  *:* 用于确定 值的比例系数 |
|  | 输出: 带标签的簇 |
| 1: | **for**  in **do** |
| 2: | 寻找每个轨迹点的最近邻点 KNN () |
| 3: | 计算轨迹点与最近邻点KNN()形成的角度 |
| 4: |  |
| 5: | **end for** |
| 6: | 对所有降序排列 |
| 7: |  |
| 8: | 定义内部点集合  边界点集合 |
| 9: | **for**  in **do** |
| 10: | **if**  **then** |
| 11: | 添加  到 |
| 12: | **else** |
| 13: | 添加  到 |
| 14: | **end if** |
| 15: | **end for** |
| 16: | /\*计算可达距离\*/ |
| 17: | **for**  in **do** |
| 18: | for all  in |
| 19: | **end for** |
| 20: | /\*根据可达距离连接成簇\*/ |
| 21: | 初始化一个空的列表用于存储簇 |
| 22: | **for**  in  **do** |
| 23: | **if**  **then** |
| 24: | 将  添加到  所在的簇 中 |
| 25: | **end if** |
| 26: | **end for** |
| 27: | 为内部点簇分配田标签field |
| 28: | 为边界点分配路标签road |
| 29: | **return** |

|  |  |
| --- | --- |
| 算法2：基于空间距离特征的簇边界重置策略 | |
|  | 输入：初步分簇结果  :田点邻域点密度阈值  ：乘法比例系数 |
|  | 输出: 簇边界重置结果 |
| 1: | /\* 计算空间距离 \*/ |
| 2: | 计算田路间的距离 ,,  轨迹点间的距离矩阵 |
| 3: | /\* 簇边界重置策略 \*/ |
| 4: | Initialize |
| 5: | **for****from** 0 **to** |
| 6: | **if** label () = field **then** |
| 7: |  |
| 8: | **if**  **then** |
| 9: |  |
| 10: | **end if** |
| 11: | **end if** |
| 12: | **end for** |
| 13: |  |
| 14: | **for****from** 0 **to** |
| 15: | **if**  **then** |
| 16: |  |
| 17: | **if** count > 0 **then** |
| 18: |  |
| 19: | **end if** |
| 20: | **end if** |
| 21: | **end for** |
| 22: |  |
| 23: | **for****from** 0 **to** |
| 24: | **if**  = field **then** |
| 25: |  |
| 26： | **if** count > 0 **then** |
| 27： |  |
| 28： | **end if** |
| 29： | **end if** |
| 30： | **end for** |
| 31： |  |