培育数字经济新动能　助推数字中国新发展
2021数字中国创新大赛大数据赛道—城市管理大数据专题
Digital China Innovation Contest, DCIC 2021

早高峰共享单车潮汐点的群智优化

参赛编号：CS-GXDC-0288

目录

一、作品主要产出

作品主要产出

01 区域划分方法

- 使用geohash对共享单车订单的最近停车围栏查找进行优化
- 基于围栏大小设置停车允许范围
- 选择HDBSCAN作为停车围栏区域划分策略

02 三种潮汐区域识别方法

- 按区域内自行车“留存数量”排序（车多）
- 按区域内自行车“留存密度”排序（车挤）
- 按区域内自行车 留存数量与留存密度的综合指标排序（又多又挤）

03 “削峰填谷”引导算法及引导系统设计

- 提出引导算法概念
- 引导算法网页版demo展示
- 一个完整的引导系统设计
- 奖励规则的商业猜想

04 “乱停车”现象及治理建议

- 基于数据的共享单车锁具、车体物理状态异常监测“乱停车”
- 实地考察“乱停车”集中区域，提出具体治理建议
- 在部分“乱停车”现象集中且距围栏过远地区增设停车围栏.对于过于狭窄无法增设停车围栏的路段提出治理建议
- 引导用户前往最近停车围栏

二、赛题分析

赛题分析——问题需求

▶项目的商业运用

在早高峰(7:00-9:00)，部分繁忙地段时常发生因共享单车**围栏爆满或缺乏**导致的共享单车**占道及”乱停车”问题。**

通过对哈啰单车调度人员的调查发现，目前较为常用的，解决”借不到”、”还不进”问题的方案是通过调度车全天将单车**淤积严重地区**的共享单车运送至**临近非淤积点，较为耗费人力物力**

赛题分析——问题需求

▶项目的商业运用

就上述问题，我们对共享单车**订单数据**及共享单车**围栏坐标**进行数据科学探索，**使用算法得出潮汐现象最严重的区域。**设计了一个完整的**引导系统和调度算法展示DEMO**，通过该系统**减少调度车使用次数，节约成本。**同时，对”**乱停车**”现象进行一定的识别，并提供备选解决方案。

赛题分析——问题需求

▶ 城市地理信息的特征及因公开数据源缺失导致的问题

由于障碍物（楼房,宽马路等）和交通规则的存在，城市中的地理信息与传统数据科学中的空间不同。城市抽象为数学概念后是一个**二维的、有向的、有权的、存在环的、局部非处处联通**的图。所以不能简单地使用点与点之间的地理直线距离进行计算。

由于**缺乏自行车可行驶道路的点到停车围栏几何区域的图数据**，只能简单地采用**直线距离**判断当前开/关锁的共享单车应该归属的停车围栏。但在**共享单车围栏聚类**及后续的**调度优化**，**“乱停车”现象识别**过程中，我们均考虑到了城市地理信息特征的问题。

赛题分析——数据分析

▶数据清洗

1.锁具异常行为检测及异常数据清理：

使用Microsoft SQL Server进行数据预览，发现共有4136辆车存在连续开锁、连续关锁等锁具异常（原因可能时锁具损坏,电量不足等）现象。

以BICYCLE_ID为001ca978928d0e762aaede9118e3c7e6的自行车为例：

3	001ca978928d0e762aaede9118e3c...	24.468531	118.098985	0	2020-12-25 07:38:41.0000000	ws7gpqm	218384	3469	13.6616667384733
4	001ca978928d0e762aaede9118e3c...	24.467766	118.092034	0	2020-12-25 07:43:38.0000000	ws7gpnp	268792	3617	62.3311982540882
5	001ca978928d0e762aaede9118e3c...	24.467886	118.09202	0	2020-12-25 07:43:52.0000000	ws7gpnp	268791	3617	75.583958079798
6	001ca978928d0e762aaede9118e3c...	24.467932	118.092116	0	2020-12-25 07:43:56.0000000	ws7gpq2	268790	3617	78.6065244871626
7	001ca978928d0e762aaede9118e3c...	24.467969	118.09205	1	2020-12-25 07:44:04.0000000	ws7gpq2	237305	3617	83.9087947504957

可以发现从7:38:41开始产生了4个连续开锁数据，并在很短时间内产生了关锁数据。

赛题分析——数据分析

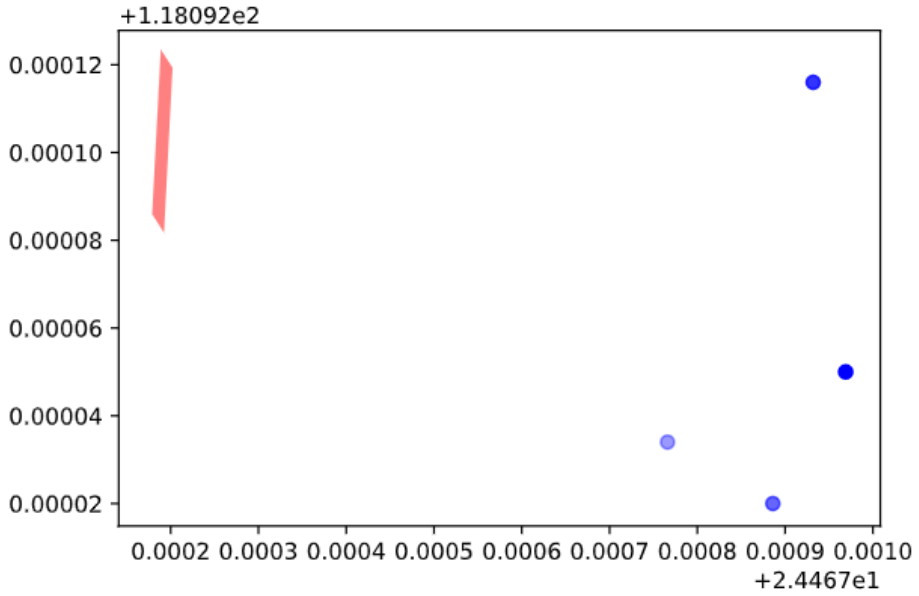
数据清洗

1.锁具异常行为检测及异常数据清理：

经可视化可以看出该车在异常数据发生期间(后四个开锁数据和最后一个关锁数据)是持续移动的，而从静止状态转为移动状态的起始数据点是第一个开锁数据（7时38分）。

故可以去除由7:43:38开始的开锁记录（保留7:38:41的记录和7:44的关锁记录），并将该车加入异常列表。

同理，对于连续的关锁数据,仅保留最后一条，即由移动状态转为静止状态的数据。



赛题分析——数据分析

▶数据清洗

2.去除7时前与9时后的数据：

原始自行车订单数据中存在的订单时间为早6时至10时，而需要研究的是早高峰时段(7:00-9:00)之间存在潮汐现象的共享单车围栏，故直接移除7:00之前和9:00之后的数据。

数据清洗结果：

可用于后续数据挖掘与算法分析的共享单车开/关锁数据数量由原先的585292个下降到了339810个，原始数据中共有约41.9%的数据不可用于后续分析（所处时间不在研究范围内，或为异常数据）。

赛题分析——数据分析

▶使用geohash对共享单车订单的最近停车围栏查找进行优化

Geohash的原理是将经纬度编码至字符串，以等长宽方格进行划分。字符串长度越长、划分精度越高，优化也越明显。

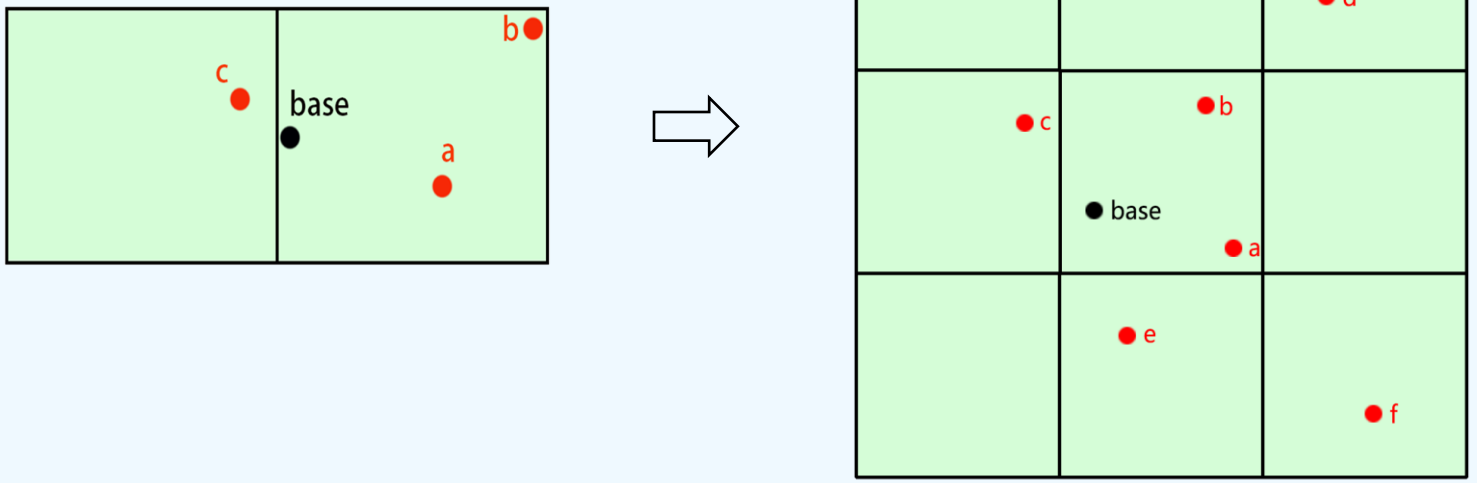
对于本次竞赛数据，过长的geohash直接加入字典进行查找会导致较大的共享单车围栏**无法被全覆盖**以至于在其中部分区域的车辆**无法被计入**。所以我们在进行了综合考虑后(长度最大的共享单车围栏(展鸿路_L_B10002)约为84米. 使用7位geohash即153 * 153 方格进行四角定位最大可覆盖306 * 306的围栏)选择了**7位geohash**, 解决了围栏中的全覆盖问题. 但仍有4894个离围栏过远(超过153米至306米)的共享单车未被围栏匹配

使用字典树数据结构可以解决这个问题，但是7位geohash无法覆盖到的自行车数据**仅占总数据量不到1%，且可直接归为“乱停车”无需计入任何围栏**，所以对于所有数据使用字典树查找反而会大大增加计算时间.

赛题分析——数据分析

▶使用geohash对共享单车订单的最近停车围栏查找进行优化

为了避免geohash的**边缘问题**：将a识别为离base最近的点，而非c点，对于每一个围栏四边形顶点都将其**本身编码和相邻的8个编码**加入计算过程。

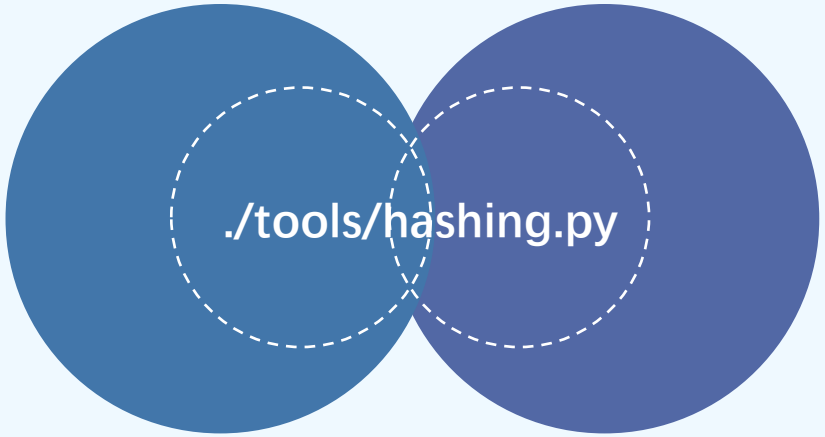


使用geohash前时间复杂度:O(n):n=count(fences); 使用geohash后时间复杂度: xO(1): x远小于n

赛题分析——数据分析

▶使用geohash对共享单车订单的最近停车围栏查找进行优化

1.每一个**自行车**订单对应的geohash编码。



2.每一个**围栏**对应的36个可能重复的geohash值，将其反向映射得到每个geohash编码关联的围栏FID。

赛题分析——数据分析

▶数据完善小结

对于围栏数据：

新增中心点坐标LATITUDE，LONGITUDE。将原有的五个坐标解包为10个维度（LATITUDE_0, LATITUDE_1等），方便读取。

字段名	字段释义
FID	功能同FENCE_ID
ROAD	当前共享单车围栏所属的街道名
LENGTH, WIDTH	当前共享单车围栏长宽
AREA	当前共享单车围栏的面积

赛题分析——数据分析

▶ 数据完善小结

对于自行车数据：

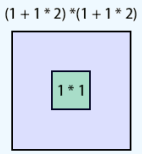
新增字段：GRID, BID, NEAREST_FENCE, DISTANCE, DAY

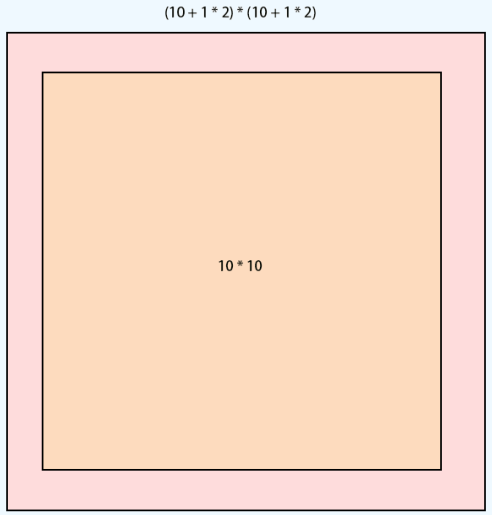
字段名	字段释义
GRID	自行车点位对应的geohash
BID	根据index形成的编号，使每条数据都具有独立性
NEAREST_FENCE	最近的共享单车围栏对应的FID
DISTANCE	到最近共享单车围栏(边缘)的距离
DAY	对应的日期(21, 22, 23, 24, 25)

部分数据的NEAREST_FENCE 和DISTANCE距离为-1，意味着该点位周围153-306米内没有任何共享单车围栏。在实际生活中可将其作为“乱停车”数据的一部分。

赛题分析——基于围栏大小设置停车允许范围

基于围栏大小设置停车允许范围





不同围栏的大小差距非常之大，延伸相同长度之后覆盖的额外范围差距也非常大。

例如，1 * 1围栏向外扩张1米后，额外增加的面积只有9 – 1 = 8 平方米，而 10* 10围栏向外扩张1米后，额外增加的面积变为144-100=44平方米。

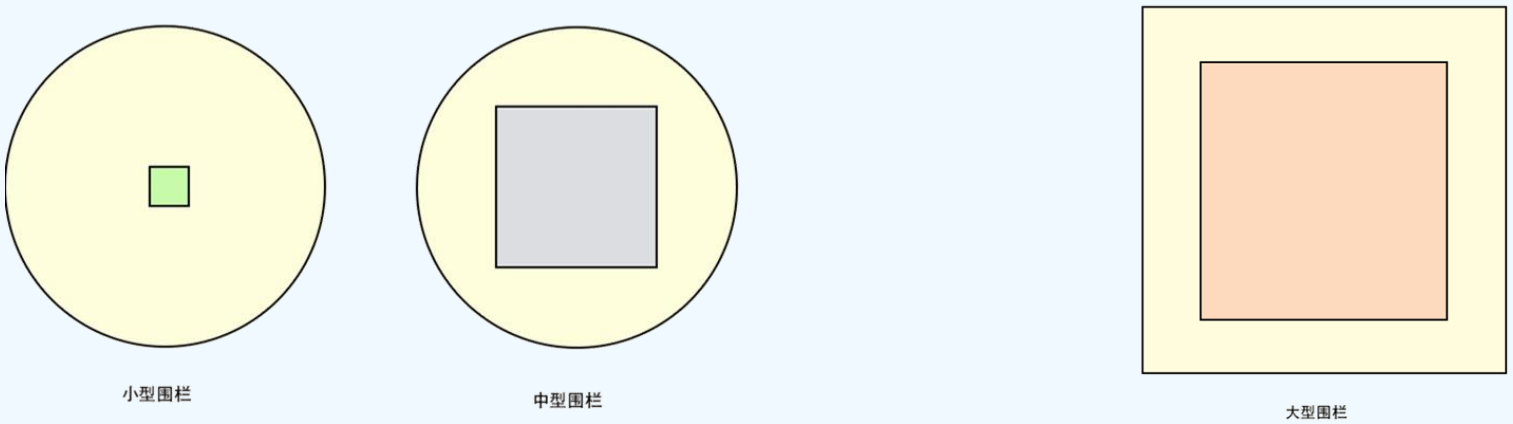
大型围栏外围调度难度远远高于中小型围栏，对大型围栏做特殊处理**减少外围可容忍的额外停车距离。**

赛题分析——基于围栏大小设置停车允许范围

▶基于围栏大小设置停车允许范围

加入CENTER_DISTANCE维度：车辆与围栏中心点的距离。将非异常数据拆分为**可计入停车围栏流量的点**和**无法计入的点。**

无法计入点定义为：离停车围栏中心点40米外（针对中小型共享单车围栏），或离停车边界20米外（针对大型共享单车围栏）的点。

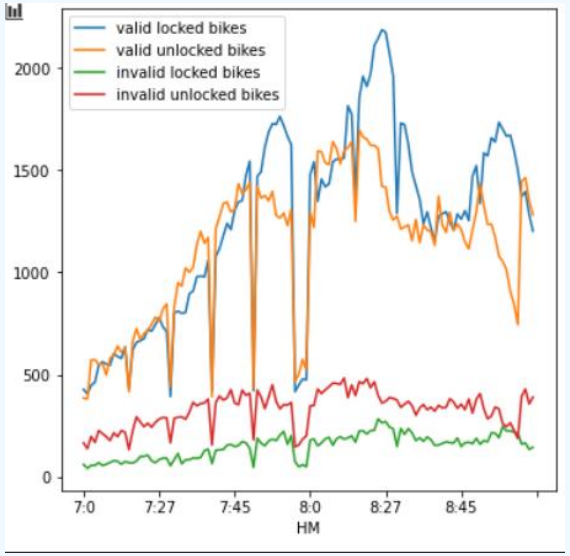


进行剔除后, 数据量由339810降低为282383, 产生了57427个停车区域外(不计入流量)的点, 即”乱停车”点

赛题分析——开/关锁数据的进一步挖掘与可视化

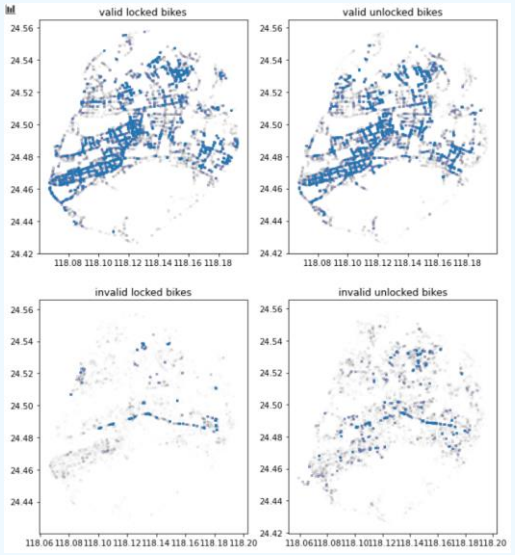
“乱停车”现象集中时段和地区的探索

按“时间-停放量”可视化



在允许范围外停车/取车量与在允许范围内停车/取车量的变化趋势没有特别明显的不同

按“停车/取车数据的空间分布”可视化

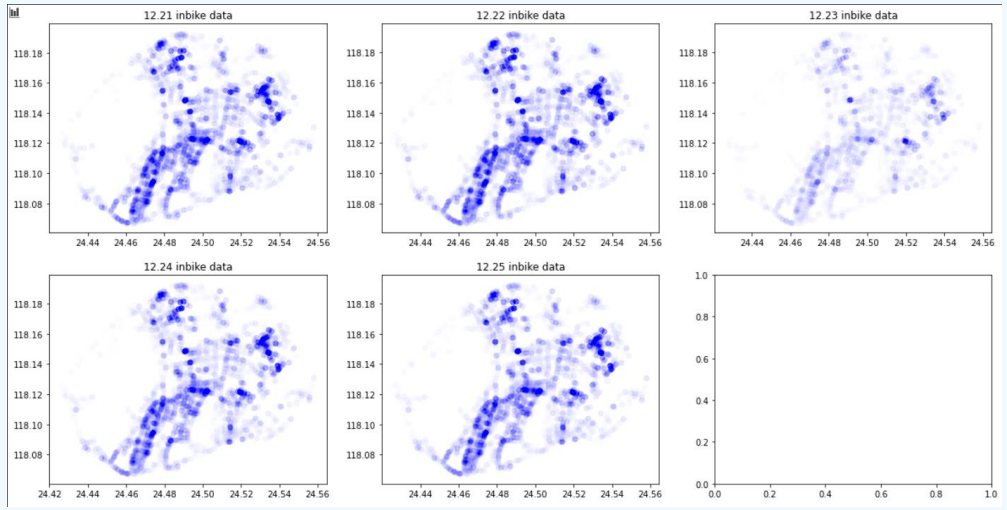


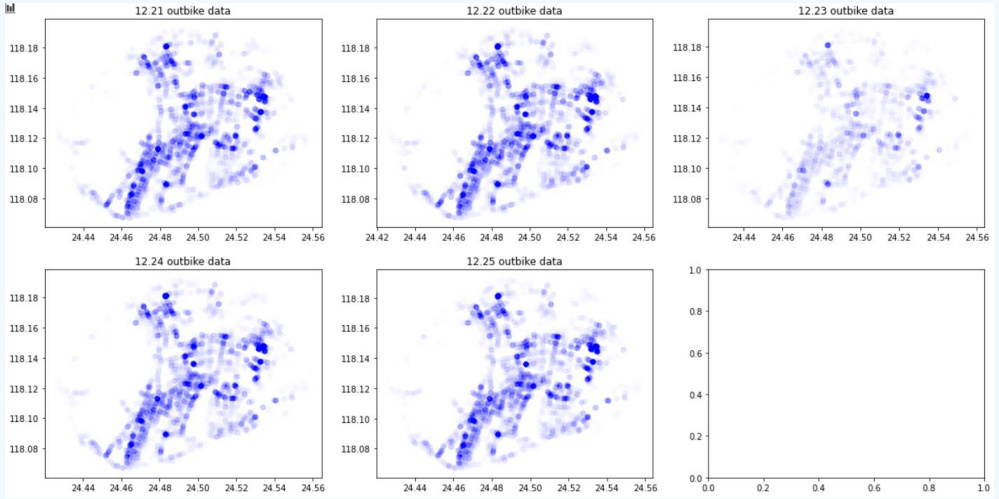
有部分区域”乱停车”现象较为严重。

赛题分析——开/关锁数据的进一步挖掘与可视化

不同日期下的开/关锁数据分布

对12月21日至12月25日的数据进行可视化分析，对比分析是否会因为圣诞节的缘故而导致早高峰潮汐现象地点不同。对比发现12.21-12.25开/关锁数据点分布都十分类似，故不对单日时间进行特殊处理.





三、模型选择

模型选择——停车围栏区域划分策略的选择

▶目前常用的，有较明显缺陷的停车围栏区域划分策略

01 网格划分

网格划分的原理是将地图变成一个个“方块”，将同方块内的点作为同一个类进行处理。

02 街道名划分

以街道名为划分标准，进行区域划分。

03 Kmeans聚类

Kmeans非常适合处理空间直线距离=实际距离且已知分类树凸集的情况。

04 DBSCAN

DBSCAN是目前较为常用的，聚类不规则形状簇的算法。

模型选择——停车围栏区域划分策略的选择

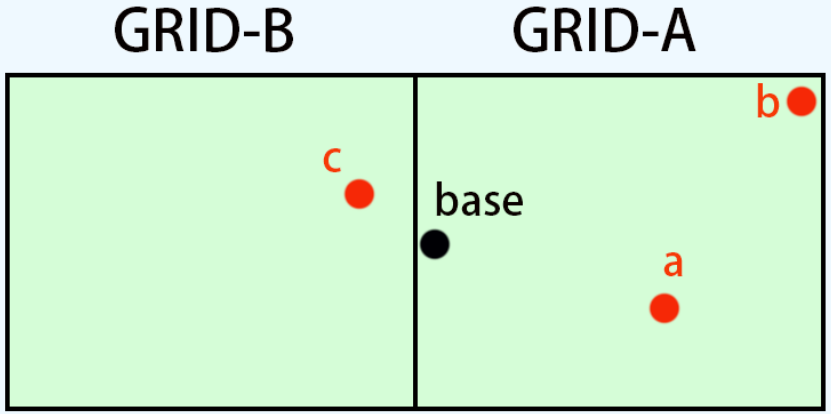
▶ 网格划分方法尝试及缺陷

在使用固定地理网格划分时, a、b、c、base四个点会出现以下**错误划分**:

Cluster A: (a, b, base)

Cluster B: (c)

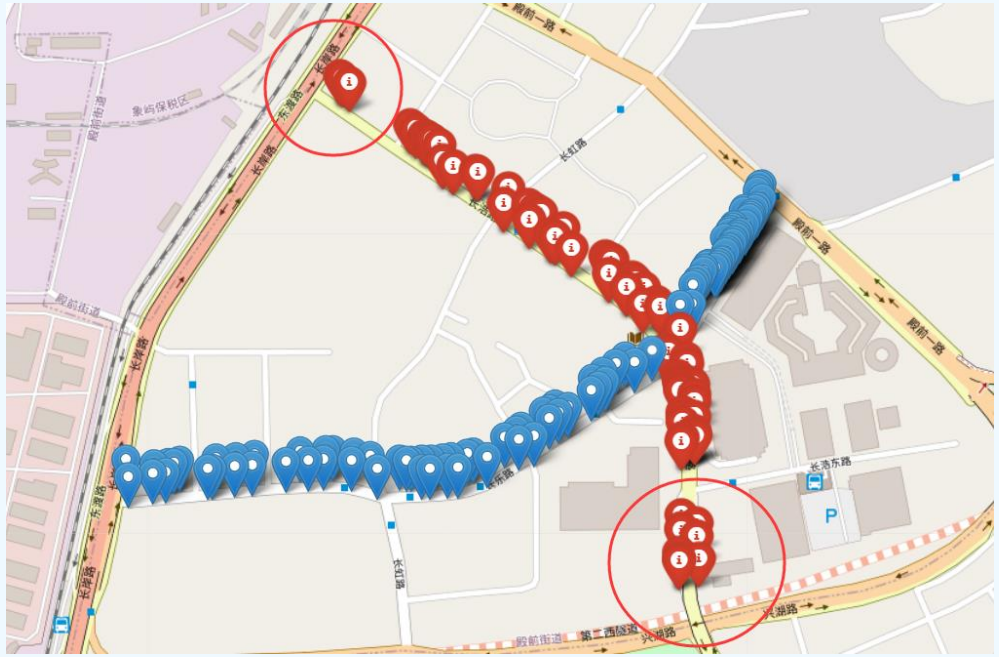
而事实上base与c距离远远小于base与a、b间距离，不应将base与c归为不相同的划分而将base与a、b当作同一个簇。



模型选择——停车围栏区域划分策略的选择

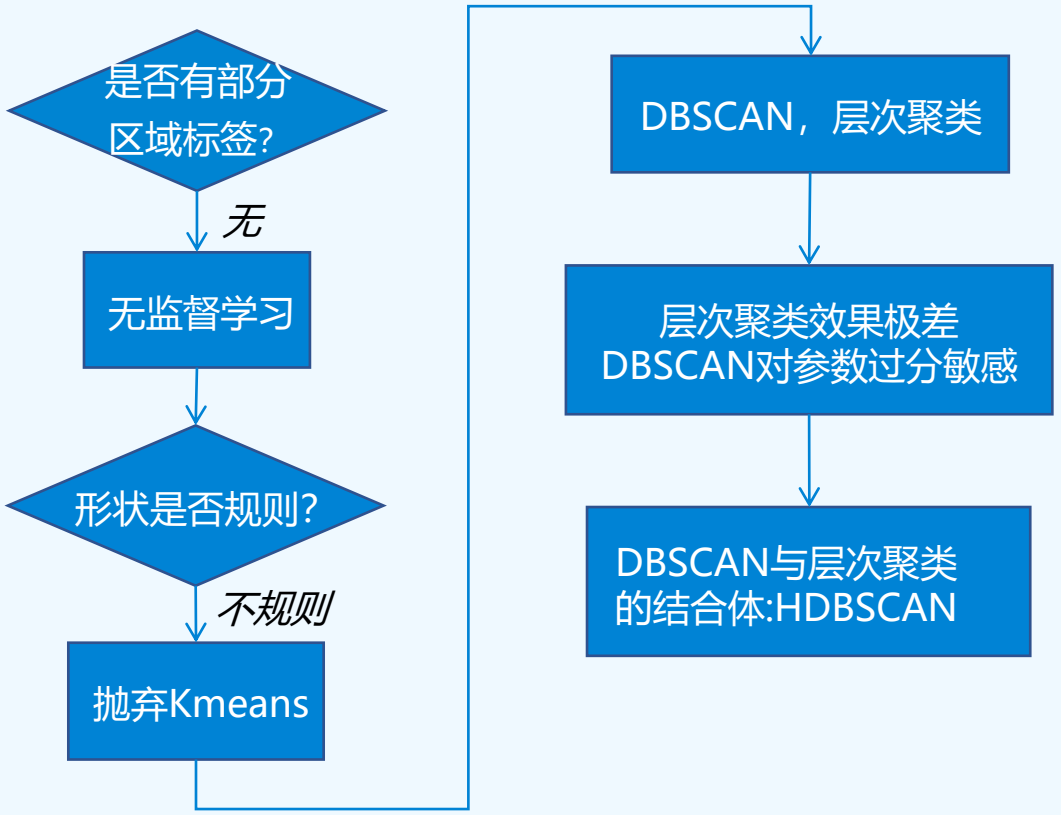
▶街道名划分法的尝试及缺陷

以“长乐路”与“长浩路”为例，部分区域（如红圈所示）的共享单车围栏存在非常明显的与同街道其余点**分离**的情况，而在长浩路与长乐路的交界处（包含部分长乐路停车围栏与部分长浩路停车围栏），停车围栏的**离散程度极低**，应作为同一簇处理。



模型选择——停车围栏区域划分策略的选择

▶本小组关于停车围栏区域划分策略的选择的思路



模型选择——停车围栏区域划分策略的选择

Kmeans聚类划分方法的尝试及缺陷

在实际地理信息中，经常出现“内外环”，“两条直线距离极近的平行街道不互通”的状况。即，出现**不连通的非凸集合**时，会出现错误的分类。

错误分类

正确分类

并且在本次项目中，我们无法确定簇数量即n_clusters无法定义。

模型选择——停车围栏区域划分策略的选择

DBSCAN(密度聚类)的尝试与缺陷

本项目曾经尝试过使用该算法进行聚类，发现了以下问题：

1. DBSCAN**对epsilon参数极为敏感**（该参数的细微变化都会导致结果截然不同，而如果不知道恰当的epsilon，则无法进行聚类）

2. 存在**链式传导**，即只要有少量的点断开，就会导致本应被分为同一个簇的点分裂为多个簇。

模型选择——停车围栏区域划分策略的选择

▶ 最终方案采取的划分方式: HDBSCAN

HDBSCAN(Hierarchical Density-Based Spatial Clustering) 是**DBSCAN与层次聚类算法的结合体**，可用于发现任意形状的簇，且在存在少量不连续点时，也不会错误地将簇分裂。

DBSCAN和HDBSCAN聚类的效果区别

模型选择——停车围栏区域划分策略的选择

基于HDBSCAN区域划分结果

```
x = list(zip(fences['F_LATITUDE'].tolist(), fences['F_LONGITUDE'].tolist()))
cluster = hdbscan.HDBSCAN(min_cluster_size=2, gen_min_span_tree=True, cluster_selection_epsilon = 0.0003).fit(x).labels_

fences['CLUSTER'] = cluster
cluster_counter = collections.Counter(cluster)
print("{} fences in {} clusters".format(len(fences), len(cluster_counter)))
sorted_cluster =sorted(cluster_counter, key=lambda x: cluster_counter[x], reverse = True)
max_index = sorted_cluster[0] if sorted_cluster[0] != -1 else sorted_cluster[1]
print("Cluster {} has maximun number of fences, with fence number {}".format(max_index, cluster_counter[max_index]))
```

```
14071 fences in 1733 clusters
Cluster 1565 has maximun number of fences, with fence number 100
```

基于HDBSCAN区域划分结果生成了**1733个簇**和cluster = -1的**离群点集**。我们在原始HDBSCAN的基础上增加了若两簇距离小于33米(规格小于双向六车道+绿化带的道路)或中间无道路横穿的围栏应判断为同一区域)则合并簇的规则

```
fences['CLUSTER'] = cluster
cur = len(cluster_counter) - 1
# 将离群点(hdbscan分类为-1)提出来变成一个单独的簇,而不都是-1
for i in range(len(fences['CLUSTER'])):
    if fences.loc[i, 'CLUSTER'] == -1:
        fences.loc[i, 'CLUSTER'] = cur
        cur += 1
print("Number of clustering after let all outlier points become a single cluster: {}".format(cur))
```

```
Number of clustering after let all outlier points become a single cluster: 3109
```

再将每个离群点单独作为一个簇，最终**簇数量为3109**。

四、方案介绍及应用设计

方案介绍——应用数据

应用数据

经过数据预处理，得到bikes_data.csv 和 fence_position.csv两个文件。

bikes_data.csv

字段名	数据类型	字段释义
DAY	INT	数据所在的日(21-25)
BID	INT(UNIQUE)	唯一订单代号
BICYCLE_ID	STRING	自行车ID
LATITUDE	FLOAT	当前纬度
LONGITUDE	FLOAT	当前经度
LOCK_STATUS	INT(BOOL)	开/关锁状态(0开1关)
UPDATE_TIME	STRING	数据发生时间
GRID	STRING	坐标geohash编码
NEAREST_FENCE	INT	离坐标点最近的围栏编码
DISTANCE	FLOAT	离最近围栏的距离
MKTIME	FLOAT	数据发生时间离era的秒数(后续没有使用)
FENCE_ID	STRING	围栏名

方案介绍——应用数据

应用数据

fence_position.csv

字段名	数据类型	字段释义
LATITUDE_X	FLOAT	围栏顶点经度（X：0-4）
LONGITUDE_X	FLOAT	围栏顶点纬度（X：0-4）
LATITUDE	FLOAT	围栏中心点经度
LONGITUDE	FLOAT	围栏中心点纬度
ROAD	STRING	围栏所在路名
AREA	FLOAT	围栏面积
FID	INT(UNIQUE)	围栏唯一标识（作用同FENCE_ID）
LENGTH	FLOAT	围栏长度
WIDTH	FLOAT	围栏宽度

方案介绍——潮汐区域识别方法

▶潮汐区域识别方法

01 “车多” 按区域内自行车**“留存数量”**进行排序 ★ 19.2469

02 “车挤” 按区域内自行车**“留存密度”**进行排序 ★ 16.9893

03 “又多又挤” 按区域内自行车**留存数量与留存密度的综合指标**进行排序 ★ 20.2442

留存数量FLOW：簇中所有日期早高峰时期的留存流量总和

留存密度FLOW_DENSITY：簇中留存流量与共享单车围栏面积和的比

方案介绍——结果及结果分析

“车多” - 按区域内自行车“留存数量”指标

```
1.  sorted_cluster_detail = cluster_detail.sort_values("FLOW", ascending = False)
2.  sorted_cluster_detail.head(40)
```

```
top_40_flow = sorted_cluster_detail.index[:40]
top_40_flow_numbers = sum(sorted_cluster_detail['FID'][:40])
print("Number of fence in top 40 crowded clusters is :{}".format(top_40_flow_numbers))
```

```
Number of fence in top 40 crowded clusters is :559
```

将cluster_detail按照FLOW字段(自行车留存数量)降序排列，观察前40个区域，这40个地区中,一共包括559个共享单车围栏数量。

用folium可视化潮汐停车围栏位置:

提交结果:

result.txt	2021-03-14 02:19:35	19.2469

方案介绍——结果及结果分析

“车挤” - 按照区域内自行车”留存密度”指标

```
top_40_density = sorted_density_detail.index[:40]
top_40_density_numbers = sum(sorted_density_detail['FID'][:40])
print("Number of fence in top 40 crowded clusters is :{}".format(top_40_density_numbers))
```

```
Number of fence in top 40 crowded clusters is :65
```

将cluster_detail按照FLOW_DENSITY字段进行排序，观察前40个区域，这40区域内只有64个围栏。观察到这些数据种大部分留存密度较大的围栏都是单独成簇，即距离其他围栏较远。

用folium可视化潮汐停车围栏位置:

提交结果：

result.txt	2021-03-14 03:26:32	16.9893

方案介绍——结果及结果分析

▶ “又多又挤” – 按“留存数量”与”留存密度”的综合指标

```
1.  flow_zscore = (cluster_detail['FLOW'] - cluster_detail['FLOW'].mean())/cluster_detail['FLOW'].std()
2.  density_zscore = (cluster_detail['FLOW_DENSITY'] - cluster_detail['FLOW_DENSITY'].mean())/cluster_detail['FLOW_DENSITY'].std()
3.  cluster_detail['MIXED_SCORE'] = flow_zscore + density_zscore
```

将留存流量和留存流量密度采用z-score标准化后相加，以此排序。这样可以兼顾留存流量和留存流量密度两个指标。

```
export_res(top_40_mix, '../result.txt', fences)
f = pd.read_csv('../result.txt', sep = '|')
sum(f['FENCE_TYPE'])

353
```

前40区域共包含353个围栏。

用folium可视化潮汐停车围栏位置:

提交结果： *(最终答案)*

result.txt	2021-03-13 05:03:08	20.2442

方案介绍——结果及结果分析

“又多又挤”指标下的聚类结果分析及调度可行性探讨

经地图放大和实地调研后发现, 这些潮汐簇所处区域均为企业密集区域/学校/商业区

如五一文化广场周边,双十中学周边,金源大厦周边,软件园2/3期等.符合实际情况

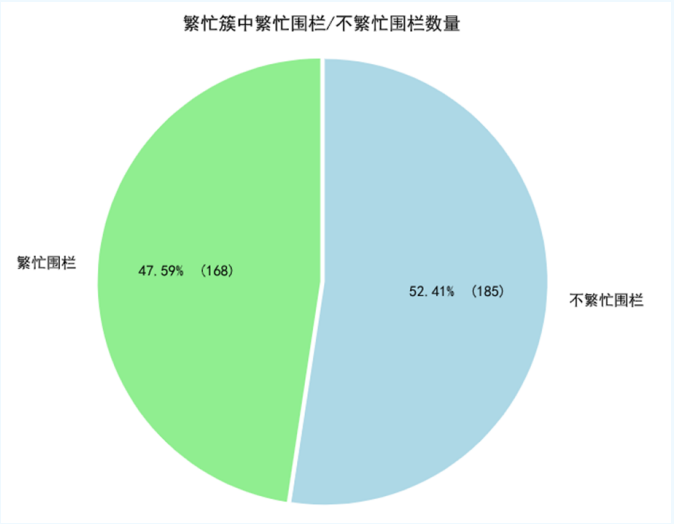
方案介绍——结果及结果分析

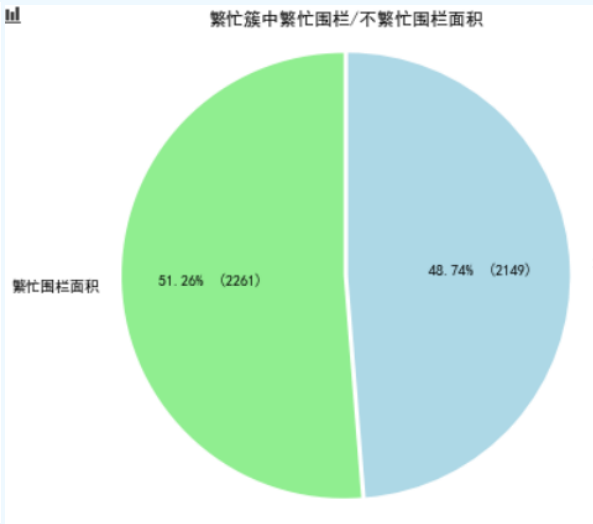
“又多又挤”指标下的聚类结果分析及调度可行性探讨

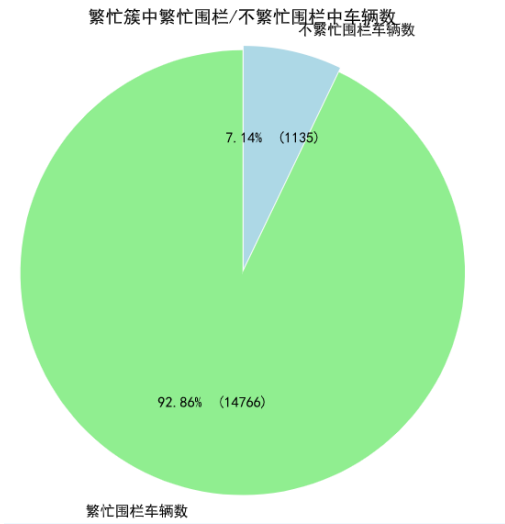
在混合指标下排名第800的共享单车围栏的拥挤程度已经可以接受(五日留存流量27, 留存密度1.51, 综合分数=1.616992),所以我们将”又多又挤”程度排名在800之前的围栏称为”繁忙围栏”,其余称为”不繁忙围栏”,对前40繁忙簇中353个围栏进行繁忙度区分后发现其中有52.41%属于不繁忙围栏

而剩下的47.59%繁忙围栏中存放了92.86%(14766辆)的滞留单车

选取一个繁忙簇进行观察, 以证明相邻或相近的停车围栏是否存在着较大潮汐程度差距







方案介绍——结果及结果分析

“又多又挤”指标下的聚类结果分析及调度可行性探讨

选取一个拥有35个停车围栏的密集簇进行观察，将综合分数高于8的围栏标记为黑色，其余繁忙围栏(1.61 < 综合分数<8)标记为红色，将不繁忙围栏标记为绿色，发现距离相近甚至相邻的围栏中繁忙程度也有极大差距

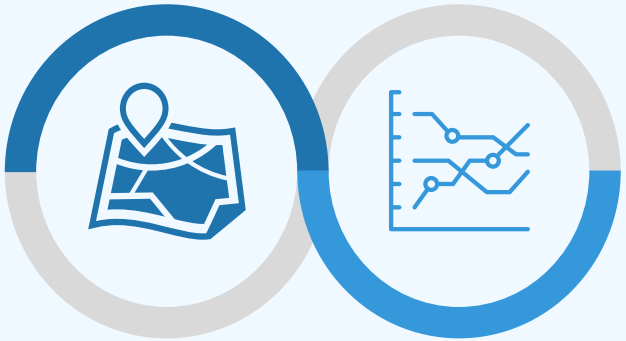
这充分说明了**繁忙簇中距离相近的围栏中也存在着部分较为空闲的围栏可供潮汐围栏分流，为后续的共享单车调度提供了可行性支持**

应用设计——针对“还不进”问题的用户调度算法

▶寻找“不繁忙”单车围栏的方法

经过对哈啰单车调度人员的调查，我们得知共享单车调度是全天进行的。所以实际上的”拥挤围栏”和算法中的”繁忙围栏”可能存在着差距: 若某个围栏刚刚实行过单车调度,那么这个围栏在之后的一段时间其实是不拥挤的。我们应使用当前自行车存量和存量密度代替之前计算的流量与流量密度作为指标进行调度。

所以可以将面临”停车难”问题的用户引导至**较近的**，符合以下规则（或符合以下规则之一）的停车围栏:



现存车辆较少且车辆密度较小

预计入流量及预计入密度较小

* 预计入流量/入密度可通过将共享单车围栏的流量数据**按照时间段划分后**进行时间序列数据分析。此次因缺乏较为长期的时间序列数据,故没有在模拟引导中体现

应用设计——针对“还不进”问题的用户调度算法

▶调度过程实施步骤

geohash → **Dijkstra** → **移除异常值** → **加权结果**

采用geohash算法获取当前坐标点的geohash，并列举出**当前geohash单元格及临近八个单元格**的所有围栏。

使用Dijkstra算法算出当前自行车所在点与这些围栏之间在**有向有权图中的最短路径。**

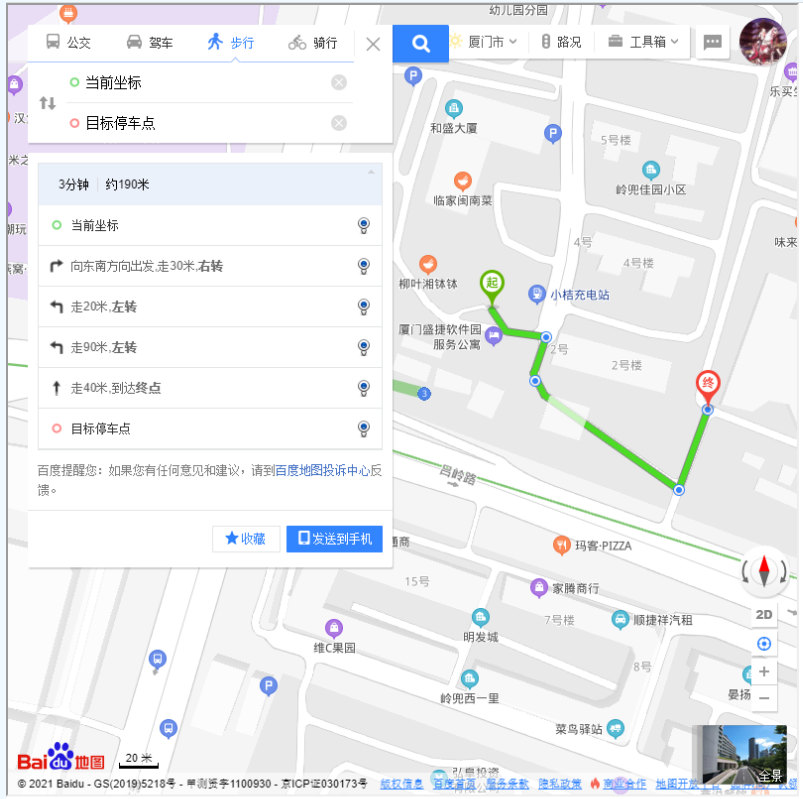
抛弃用户**接受范围(max_distance)外的围栏**，即移除最短路径距离大于该值的围栏。

将停车围栏“车多且挤”的程度(现存车辆与现存车辆密度的zscore之和, 若已知预计入流量,可用预计入流量代替”多且挤”程度，或将其作为一个新的维度)、当前距离、正流量天数（可不用）标准化，分别乘以不同的**正经验权重k1、k2、k3。**设“多且挤”程度为x1、当前距离为x2、正流量天数为x3，那么最终得到分数**s = k1x2 + k2x2 + k3x3**，将用户引导至s**最小**的点。需要注意的是调度决策做出之后需要将目标围栏的车辆数暂时+1防止出现相近时间多车同时调度导致的全部车辆涌向同一个点的问题. 若接受调度的车辆最终关锁点不属于目标围栏,则目标围栏车辆计数-1

应用设计——针对“还不进”问题的用户调度算法

▶ 调度算法模拟网页版demo展示(bike_app应用)

本次设置k1 = 0.3，k2 = 0.5，k3 = 0, max_distance = 500，实际应用中应根据具体情况修改。



当前纬度: 24.490992 当前经度: 118.181635 最大行走距离(0米以上): 500

调度

	FID	ACTIVE_DAYS	DENSITY	FLOW	MIXED_SCORE	DISTANCE	LATITUDE	LONGITUDE	SCORE
16	1758	0	-0.761995	-9	-0.735531	189	24.490702	118.183098	-1.670227
8	9678	0	-2.615640	-18	-1.849687	370	24.488785	118.183478	-0.758149
5	9679	0	-2.221277	-17	-1.648761	393	24.488745	118.183476	-0.515803
15	9677	4	0.622175	5	0.383801	255	24.490640	118.183468	-0.313108
2	9731	1	-0.317668	-7	-0.474674	364	24.491127	118.183547	-0.179060
6	9680	0	-1.857759	-15	-1.420987	431	24.488430	118.183441	-0.153871
12	1752	0	-0.552064	-6	-0.531159	395	24.488707	118.183476	-0.088582
1	9732	0	0.000000	0	-0.068293	376	24.490747	118.183530	-0.056621
3	9675	1	-0.128349	-1	-0.160314	369	24.490716	118.183468	-0.026446
7	9733	0	-0.958979	-14	-1.013553	436	24.488272	118.183422	0.033104
10	1754	2	0.000000	0	-0.068293	371	24.490680	118.183471	0.137478
9	1751	0	-0.254171	-3	-0.290776	435	24.488336	118.183430	0.292628
13	9681	1	-0.368869	-3	-0.337734	432	24.488385	118.183436	0.368686
4	1753	4	0.689647	5	0.411424	377	24.490686	118.183524	0.589108
11	1757	5	1.477587	20	1.326136	319	24.491269	118.183139	0.618057
14	1759	3	0.259898	3	0.156535	440	24.488622	118.182973	0.840312
0	9750	3	0.587687	4	0.330206	437	24.488528	118.182975	0.882493

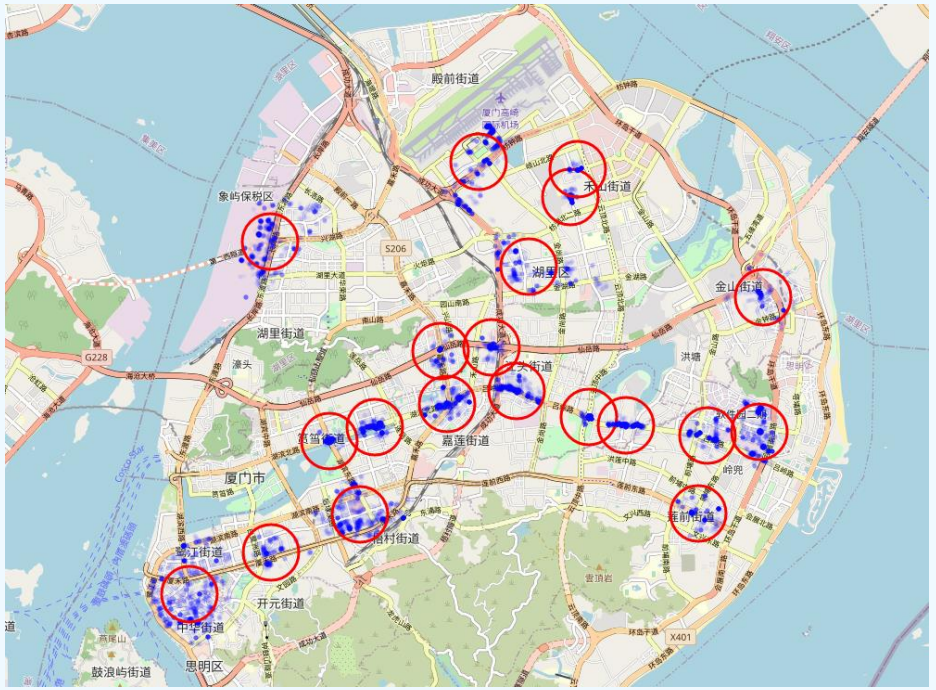
即将导航前往共享单车围栏会展路(塔埔路至吕岭路段)_L_2, 该围栏单车流量为-9, 流量密度为-0.7619954224846628, 距离当前位置189米

结果将位于(24.490992, 118.181635)的共享单车引导至189米外的会展路(塔埔路至吕岭路段)_L_2单车围栏

应用设计——基于早高峰”乱停车”数据的建议

▶ “乱停车”现象集中点

使用HDBSCAN（min_cluster_size=500，意为一个簇的最小数据量为500），对前面所定义的停车允许范围外的自行车（包括开锁和关锁。开锁处若为允许范围外，说明在7时之前出现了“乱停车”现象）进行聚类后，发现存在19处较为集中的“乱停车”点。



应用设计——基于早高峰”乱停车”数据的建议

“乱停车”具体现象及建议

现象详情：

以成功大道与吕岭路交界处”乱停车”集中点为例。可见停车围栏与“乱停车”现象密集区域直线距离在50米外，且没有直线连通路径。需要跨过吕岭路并转至对面龙伏路。

实施建议（除调度外）：

若交通条件允许，应在泰和花园处设置停车围栏，以方便市民停放共享单车。

应用设计——基于早高峰”乱停车”数据的建议

“乱停车”具体现象及建议

现象详情：

在中山路步行街附近(中华街道)的“乱停车”现象也较为类似：较近的共享单车围栏分布在镇海路、厦禾路和轮渡一侧。而该区域道路狭窄，车流量大，店铺众多，共享单车是此处最为便捷的出行方式。

应用设计——基于早高峰”乱停车”数据的建议

“乱停车”具体现象及建议

现象详情：

为此我们前往该处进行了实地调查，调查发现：

1.咨询执勤交警得知，中山路步行街内不允许骑自行车进入。

2.思明南路与思明西路部分未开业的商铺门口存在不少随意停放的共享单车。

应用设计——基于早高峰”乱停车”数据的建议

“乱停车”具体现象及建议

现象详情：

3.如右图所示，思明南路、思明北路、思明东路、思明西路道路狭窄，两侧均为商铺，不适合设立共享单车停车围栏。

4.该区域小巷极多，调度车辆(在之前对于调度人员的调查中发现调度车辆均为小型卡车)难以进入，调度人员也很难在纵横交错的小巷中准确找到被随意停放的共享单车。

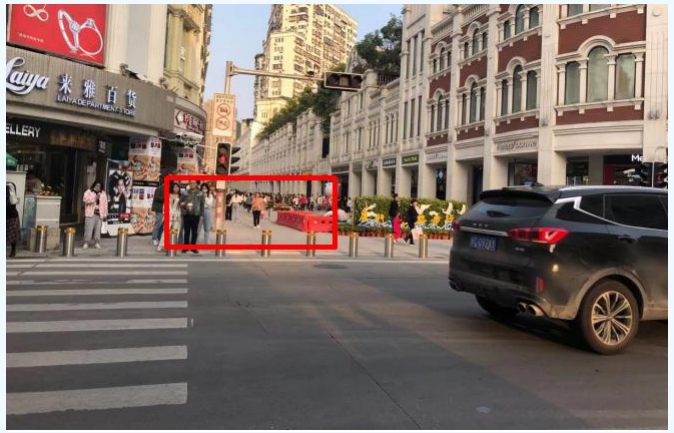
应用设计——基于早高峰”乱停车”数据的建议

“乱停车”具体现象及建议

实施建议（除调度外）：

1.观察到中山路步行街入口处较为宽敞，是否可以在市政规划及市容允许的前提下于此处设立共享单车围栏？例如可以在图的红色方框位置（花坛后方和围挡处）增加共享单车停车位。

2.因小巷内散落共享单车的收集难度过高，应适当提高“乱停车”的处罚力度，并使用后续的调度系统进行引导



应用设计——共享单车引导系统设计

系统组成

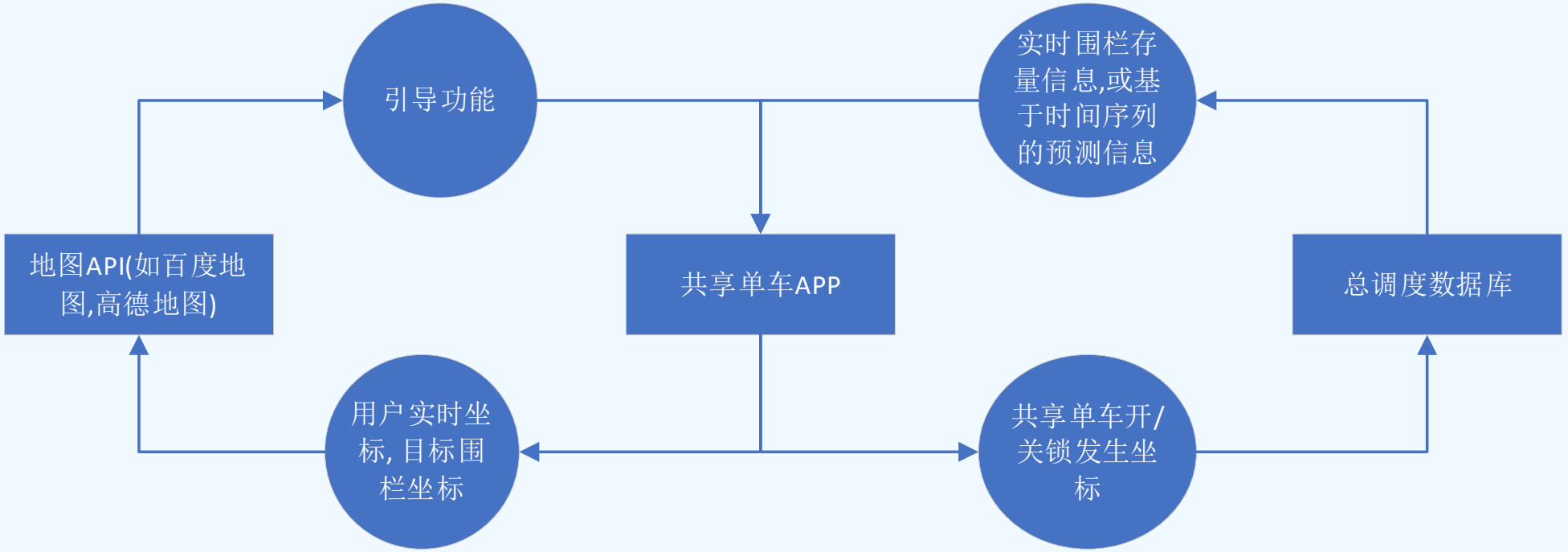
01 总调度数据库(可以考虑放置在i厦门)及其API：提供实时围栏存量信息或最新基于时间序列的预测信息供多个共享单车商家调用,避免企业间信息不互通导致的调度混乱问题。并可接收共享单车商家方回传的用户开/关锁信息供持续改进

02 地图API(百度地图,高德地图等):提供城市图数据及最短路算法计算出的距离;提供实时寻路功能

03 共享单车商家方：从总数据库处请求实时停放情况;使用app接入地图API获取当前点到各个围栏的实际距离，综合实时围栏存量等信息计算出围栏分数s并联合地图API实时寻路功能进行引导;开/关锁信息传递;奖惩措施确定

应用设计——共享单车引导系统设计

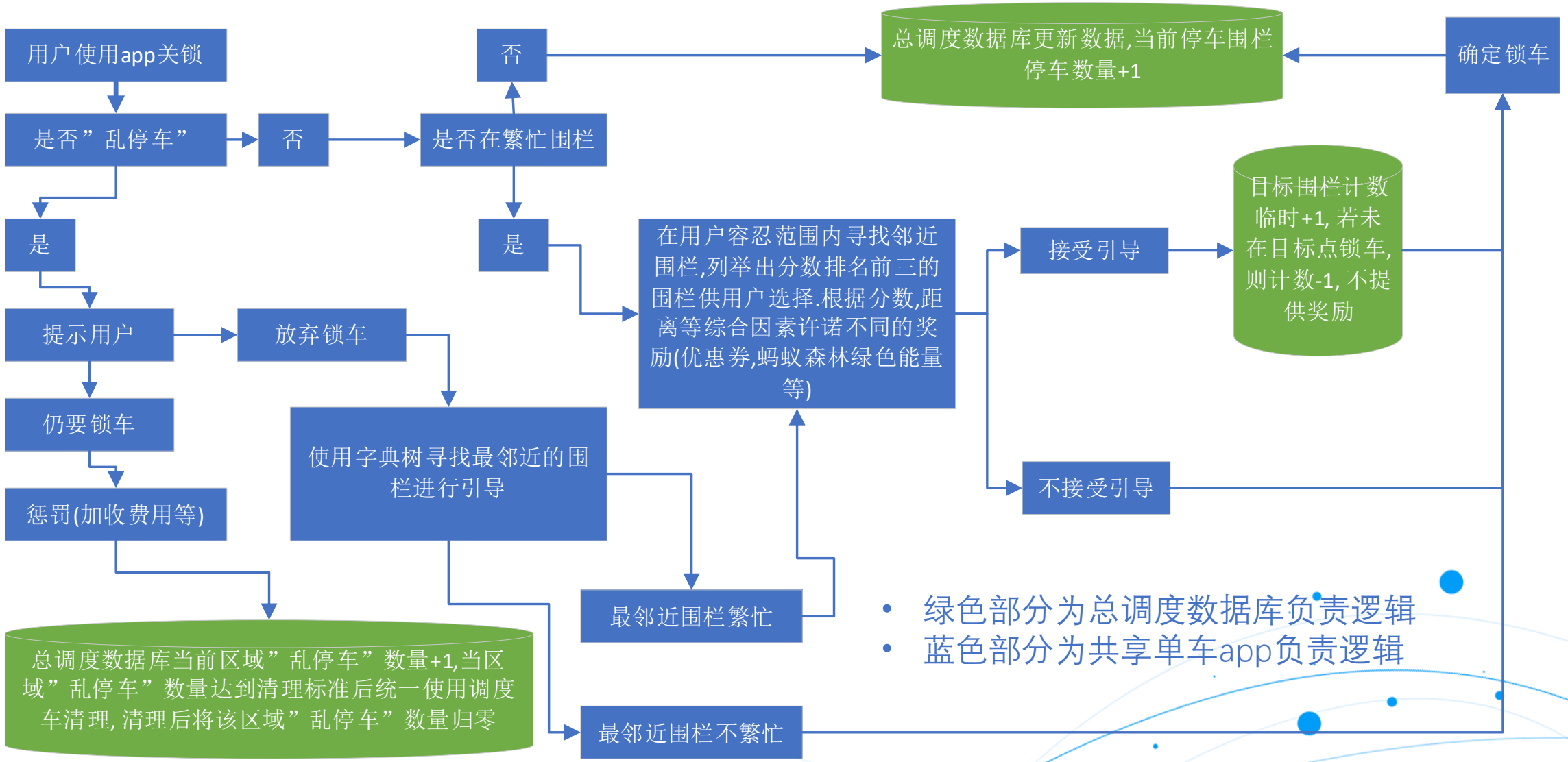
▶系统元素间的关系



总调度数据库存储有所有共享单车商家提供的单车开/关锁信息,但共享单车app不应将用户隐私数据提供给总调度数据库,减少用户隐私数据泄露风险.

应用设计——共享单车引导系统设计

▶用户在进行”关锁”动作时引导系统做出的判断



- 绿色部分为总调度数据库负责逻辑
- 蓝色部分为共享单车app负责逻辑

设计补充——对服从引导用户奖励规则的商业猜想

1. 将“乱停车”用户引导至最近停车围栏的调度过程，不提供奖励。在用户到达停车围栏后才再次判定该围栏是否为潮汐围栏并进行二次引导

这防止部分用户为了获取奖励故意在停车围栏较远的地方进行锁车操作，杜绝了“薅羊毛”的可能

2. 用户可获得的最高奖励与s最小的围栏与当前用户所属潮汐围栏的距离有关

即，s最低的围栏距离越远，用户可以获得的奖励上限越高。s 的计算过程综合了各种因素，目前看来较为空闲的围栏的s并不一定是最低的， 若在计算过程中使用了时间序列预测（如，提前半个小时开始将预计为潮汐围栏的车辆调度至其余围栏），则更可以防止“薅羊毛”行为。

五、总结思考及商业价值

总结思考——作品价值

▶基于共享单车商家和城市规划层面

01 针对“潮汐现象”进行“削峰填谷”，使共享单车得到合理的调度，减少调度车使用次数,节约成本

02 针对“开/关锁状态”分析，能够发现共享单车是否存在故障现象，方便商家及时检查维修损坏车辆，保证共享单车的可用性。

03 通过研究共享单车停放数据得出的用户通勤特征，结合实地考察分析“乱停车”现象，可以帮助设立更多合理的停车点，用以减少“乱停车”现象，美化市容市貌。

04 通过数据交互实现市政与多企业联动,提高共享单车交通治理效率;企业可通过优惠券的形式进行”留客”

总结思考——作品价值

▶基于用户层面

01 缓解用户找不到最近停车围栏被迫”乱停车”被罚款的问题,改善用户体验

02 可以大大减少在例如下班高峰期、节假日的用餐时间等用车拥挤时段用户“还不进”车的问题，使得用户的生活出行变得更加便利、高效。

03 以服从引导提供优惠券或app游戏数值的形式,减少用户因”还不进”导致的烦躁感,为生活提供额外趣味

Thank You!

培育数字经济新动能　助推数字中国新发展

2021数字中国创新大赛大数据赛道—城市管理大数据专题

Digital China Innovation Contest, DCIC 2021

主办单位：数字中国建设峰会组委会

承办单位：福建省数字福建建设领导小组办公室

厦门市人民政府

数字中国研究院（福建）

协办单位：厦门市工业和信息化局

厦门市城市管理行政执法局

执行单位：厦门信息产业和信息化研究院（厦门大学）

厦门市信息中心（厦门市大数据中心）

厦门市规划数字技术研究中心

支持单位：厦门市交通运行监测指挥中心

数字福建城市交通大数据研究所