出租车轨迹数据分析

随着移动互联网络、卫星定位技术、WIFI 定位技术、蓝牙定位技术等位置采集技术的高速发展，轨迹数据大量产生，轨迹数据是指带有时间戳标记的一系列位置的集合，常见的轨迹数据有出租车轨迹数据、手机数据、志愿者数据集、公交车轨迹数据、签到数据等。滴滴出行公司采集到的轨迹数据主要是出租车轨迹数据，出租车轨迹数据中包含着大量的有关人类活动、出行规律、城市功能区域的信息，研究出租车轨迹数据有助于解决智能交通、智能旅游推荐、城市社区结构划分等问题。可以辅助打造智能程度较高的智慧城市，更好的响应“城市让生活更美好”的口号。

在轨迹数据的研究中，主要分为两大类的研究内容，一类是研究轨迹数据挖掘中的方法论，另一类是研究轨迹数据在具体的生活场景中的应用。轨迹数据研究方法包括轨迹数据预处理方法、轨迹数据管理方法、轨迹数据隐私保护方法、轨迹分类、异常检测等方法。

数据预处理方法有数据清洗、路网匹配、轨迹压缩、轨迹分段。数据清洗指的是对于数据中质量较差的数据做删除或是替换；路网匹配是指将轨迹数据定位到实际的地图的路网中；轨迹数据压缩是对原始数据进行压缩处理，从而获得一条占用更少空间，且误差在允许范围内的压缩轨迹，可以解决由于轨迹数据量过大而导致的一些问题；轨迹分段主要解决的问题是轨迹数据可能因为时间间隔大、位于不同行驶路段等原因而导致特征有所不同，因此需要对轨迹进行分段处理。

轨迹数据隐私保护指的是保护个人或是机构等实体的不愿被其他实体所获知的信息。轨迹数据中包含的有直接隐私信息以及间接隐私信息，直接隐私信息指含有一些敏感位置、敏感行程的轨迹数据，间接隐私信息指从中推测出用户的个人信息或私密行为的轨迹数据。这两类数据都需要做一些处理以保护隐私，这样得到的数据就是可发布的脱敏数据了。

轨迹数据在具体的生活场景中的应用则更具有多样性。具体的应用场景有智能交通、环境监测、旅游推荐、城市功能区识别、研究人类活动模式等。

本实验主要通过分析成都市的出租车轨迹数据以及订单数据获取有关成都市社区结构划分、交通道路情况的信息，并且结合实际情况对于分析结果做出解释，并在已有的分析结果的基础上对于市民出行、出租车运营、城市规划等领域内的问题提出有针对性的建议。

1.数据说明

本实验中所使用的数据是来自滴滴出行“盖亚”数据开放计划的。应滴滴出行盖亚计划开放宣言共享原则的第七条的要求，首先标明本实验中数据来源：https://gaia.didichuxing.com。数据开放计划中开放的数据有 2018 年五六月成都 POI 检索数据、2016 年 10 月西安市二环局部区域轨迹数据、2016 年 11 月西安市二环局部区域轨迹数据、2016 年 10 月成都市二环局部区域轨迹数据、2016 年 11 月成都市二环局部区域轨迹数据。本实验中主要使用的是 2016 年 11 月成都市二环局部区域轨迹数据。

2016 年 11 月成都市二环局部区域轨迹数据包含两部分的内容，一部分是滴滴的全样本轨迹数据、另一部分是订单数据。

图 1 是原始数据的部分截图。图 1（a）是部分轨迹数据图，图 1（b）是部分订单数据图。从图 1 中可以看出，两类数据都是以 csv 文件格式进行存储的，各个属性列之间以逗号分隔开，其中轨迹数据包含五个属性，订单数据包含七个属性。轨迹数据的数据采样精度为 2~4 秒，五个属性分别是行驶的滴滴车辆的司机的 ID，车辆该次行程的订单 ID，进行轨迹点采样时的时间戳，采样时车辆所处位置的经度，采样时车辆所处位置的纬度，属性值均是以 String 类型存储。订单数据的七个属性分别是车辆该次行程的订单 ID，车辆该次行程开始时的时间戳，车辆该次行程结束时的时间戳，该次行程中乘客上车位置的经度，该次行程中乘客上车位置的纬度，该次行程中乘客下车位置的经度，该次行程中乘客下车位置的纬度，属性值同样也都是以 String 类型存储的。

观察图 1 中的数据还可以发现司机 ID 与订单 ID 的属性值都是看上去没有意义的字符串，这是滴滴出行在进行数据开放时已经对司机以及订单信息进行了加密脱敏匿名化处理，做过脱敏处理的数据可以比较好的保护用户的隐私。也使得有关用户特征的分析工作难以开展，所以本实验的分析中不包含涉及到用户特征信息的分析内容。

图 1 原始数据

此外，开放的轨迹数据也是已经做了绑路处理的数据。所谓绑路处理，是一种常见的对于轨迹数据需要进行的预处理手段，之所以需要进行绑路处理，是因为目前常用的各种定位方式或多或少的都会存在着一定的误差，因此在进行轨迹采样时有可能会使得采样点偏离实际的路网中的道路，甚至落在湖泊、铁路、泥沼等不可能行车的位置点。绑路处理对于提升轨迹数据的质量是很有效的，如图 2 所示，图 2（a）中的轨迹是根据未经绑路处理的轨迹点得到的行驶路线，图 2（b）中的轨迹则是利用绑路处理后的数据得到的行驶路线。从图 2 中的对比可以发现，绑路处理显著提升了原始数据的质量。

图 2

2. 数据获取

滴滴出行开放的盖亚数据主要有两种获取方式，一种是参与数据开放计划进行申请，另一种方式是在滴滴云开放平台租用包含有盖亚数据的服务器。

第一种方式是在数据开放计划的网站 https://outreach.didichuxing.com/app-vue/dataList 的数据详情页中申请。本实验采用的是第二种方式，在滴滴云平台创建包含有盖亚数据的服务器，创建服务器的网页链接为 https://app.didiyun.com/#/dc2/add，在选择服务器类型时，选择包含有盖亚数据的服务器即可，如图 3 所示，选择“gaia-CentOS”类型的镜像。创建好服务器之后，通过命令行找到“/data”，data 文件夹下有两个文件夹 chengdu 和 xian，chengdu 文件夹中包含的压缩包是 2016 年 10 月成都市二环局部区域轨迹数据以及 2016 年 11 月成都市二环局部区域轨迹数据的压缩数据，解压后可以得到本实验所使用的数据。

图 3 选用包含有盖亚数据的服务器

3. 数据预处理

本部分的数据预处理的工作的主要目的是提升原始数据的质量。在数据说明中已经提到，数据经过了绑路处理，所以该部分主要是处理缺失值、重复值以及

异常值。首先是对于缺失值的处理，查看原始数据中的缺失值数量的多少，使用的方法就是对比将数据中的缺失值所在行全部去除前后的数据行的数目，即可得到缺失值的数量。代码如下所示。

以数据 order_20161101 为例，两次输出的关于 dataframe 的统计结果均是：[209422 rows x 7 columns]。这表明原始的数据集中并没有属性值缺失的情况。

之后处理数据中的重复值，本实验中对于重复值的定义是存在两行数据，这两行数据所有列的属性值都是相同的，则认定两行数据是重复值，处理办法就是对于若干个重复的行，最后仅保留其中一行数据。

在轨迹数据中之所以不仅仅使用位置信息判定为重复的原因是车辆可能在某一次行程的不同时间点经过同一经纬度标识的位置，比如图 4 中轨迹所示的情况，司机为了调整行驶方向进行了 U-Turn；之所以不仅仅使用订单以及时间信息判定是否为重复数据是因为在订单信息以及时间信息相同但地理位置信息不同的情况下更有可能是出现了异常值，对时间信息或者订单信息进行了错误的记录，而不是重复值出现的情况。

图 4 车辆因为 U-turn 在一次行程中多次经过某一位置

判定订单数据中是否出现了重复值也采用判定两行数据是否完全相同的理由与轨迹数据中为何采用此法类似，不做赘述。进行重复值判定与去除的代码如下所示。

同样是以数据 order_20161101 为例，前后两次打印出的输出的关于 dataframe 的统计结果分别是[209422 rows x 7 columns]和[181172 rows x 7 columns]，根据前后输出的统计结果可以得知，在 order_20161101 中，被去除的重复的数据有 28250 条。

通过观察图 1（a）中的数据，可以发现轨迹数据是按照行程顺序排列的，如果在一次行程的数据中夹杂了不属于该行程的数据，那么这条数据就是异常值。因此首先检测时间错误记录的异常值，检测的是记录的时间与数据描述中描述的采样时间间隔为 2~4 秒这一事实不一致的错误记录。以数据 gps_20161101 为例，在进行错误记录检测的过程中发现数据中不止有时间的错误记录（如图 5（a）所示，其中 v[204]与 v[203]之间的时间间隔为 1 秒，是错误记录，错误记录需要被纠正），还有一些被遗漏掉的数据条目（如图 5（b）所示，其中被圈起来的两个条目之间的时间间隔为 6 秒，而其余相邻条目间的时间间隔多为 3 秒，因此将这两条记录视为中间缺失了一条记录的数据，由于本实验使用的数据采样间隔较小，并且缺失记录的概率并不大，所以缺失一条记录的部分可以不做补全）。

图 5 轨迹数据中的错误记录与遗漏记录

关于错误记录的数据的处理部分的代码如下所示。

检查数据条目 i 是否是错误记录时，在索引不越界情况下，使用第 i-1 条，第 i+1 条以及第 i+2 条数据来辅助检查。这种检查方式建立在已经观察到轨迹数据是按照不同的行程逐条排列的基础上。判别两条相邻的数据行是否属于同一行程的主要方式是看两个数据行的订单信息是否相同。具体的处理的思路如下：

1. 首先判断条目 i 与条目 i-1 是否属于同一行程，属于同一行程则执行 2，否则执行 3

2. 判断条目 i 与条目 i-1 之间的采样时间间隔 t_i 和 t_{i-1} 是否在 2~4 秒的范围内，如果在此范围内则无需调整数据（说明 i 未错误记录）。如果采样时间间隔不在 2~4 秒内，条目 i 与 i+1 属于同一行程，且条目 i 与 i+1 的采样时间间隔也不在 2~4 秒内，则调整采样时间点 t_i 为 t_{i+1}-3（因为期望间隔时间为 3 秒）；如果 i 与 i+1 不属于同一行程，则调整 t_i 为 t_{i-1}+3。其余的情况无需调整。之所以如此调整，是为了在错误记录的影响被限制在一个条目的数据的时间偏差的基础上尽可能修正错误记录。当 t_i 和 t_{i-1} 间隔不在 2~4 秒内时，之所以没有考虑条目 i+2 的影响是因为出现错误记录的概率比较小，在四条数据内出现两条错误记录的概率就更加小了，因此将该事件视为不会发生的事件。
3. 当条目 i 与 i-1 不属于同一行程时，根据对原始数据的观察，可以做出断言，i 与 i+1 以及 i+2 属于同一行程，这是因为出租车的一段行程的时间应该是大于 12 秒的。此时之所以要引入条目 i+2，是因为当条目 i 与条目 i+1 的采样时间间隔异常时不引入其他数据无法判别是哪一条数据被错误记录了，当发现条目 i+1 属于错误记录时，无需处理，因为会在下一个循环中被处理，如果是条目 i 属于错误记录，整采样时间点 t_i 为 t_{i+1}-3。

在处理的整个循环中，每次循环结束都可以保证已循环过的数据中的错误记录被处理了，而接下来的循环中的处理错误记录的逻辑也是建立在这一基础上的。

接下来要做的异常值处理工作主要是检查数据取值是否在值域内。之所以将这个检查（称之为检查二，前一个检查称之为检查一）放在后面，是因为检查一中出现的错误记录可能在值域外，但是是可以被纠正的，所以先行纠正。在纠正过后仍然位于值域外的数据则采取舍弃的处理方法。检查二检查的是时间数据以及地理位置数据，时间数据的值域是与文件名相关的，盖亚数据中，每天的轨迹数据以及订单数据分别被命名为 gps_[date]和 order_[date]，读取文件名可以得到文件中时间数据的上下界。上界是 date 00：00：00，下界是 date 23：59：59。由于成都的地理边界是形状不规则的，不能直接抽象成某种容易确定边界的几何形状，所以不采用判断经纬度坐标是否在值域内的方法来确定异常值。判断地理位置是否异常的方法是通过逆地理编码服务将经纬度信息转化成对应的位置信息。以经度为 104.11225、纬度为 30.66703 的位置为例，调用逆地理编码服务并且输出逆地理编码结果的代码如下所示。

输出的结果的部分内容的截图如图 6 所示。

图 6 部分逆地理编码结果

从图 6 中可以看到得到的返回结果中包含着逆编码后的以文本形式表示的地理位置信息，其中 city 属性的属性值是成都市，为了判断数据集中的位置是否位于成都市，只需要判断逆编码结果中的 city 属性的属性值是否为成都市。执行检查二的代码如下所示。

4.数据分析

4.1 出租车区域推荐以及交通管理建议

首先根据订单数据中的上下客位置的经纬度信息进行关于上下客位置点的分析。首先在实际的地图中分别对上客点和下客点绘制热力图。调用绘制热力图的方法（方法来源：http://lbsyun.baidu.com/jsdemo.htm#c1_15），绘制热力图时主要调整的参数是 point 以及 points 这两个参数，point 参数表示的是地图初始化显示时的中心点，根据盖亚数据描述的数据表示的地理位置范围的东南西北四个边界点[30.727818,104.043333]、[30.726490,104.129076]、[30.655191,104.129591]、[30.652828,104.042102]计算出中心点坐标[30.690581,104.086025],points 是展示在热点图中的数据点的集合，数据点的内容包括数据点的经纬度以及数据点的权重，本实验中的数据点表示的是一次上客或者一次下客的数据，所以 points 中的点的权重均设置为 1，热力图中颜色由深到浅表示数据点的集中到稀疏，通过高亮的形式展示乘客热衷的上客区域和下客区域。

工作日与休息日时城市中的人流量与流动规律会因为上班族是否工作而有所不同，所以以 11 月第一周的数据为例，绘制热力图，如图 7 所示。其中（a）、（b）、（c）、（d）四幅图分别表示的是第一周工作日上客点、周末上客点、工作日下客点、周末下客点的热力图。从中可以看出，粗粒度的来看四幅图的大致轮廓是一直的，不同之处在于工作日累计五天的出行人数要多于周末两天的出行人数。这说明从整个市区这个精度来看，工作日与周末的人流规律较为相似、上客点与下客点的区域差异不大。

图 7 第一周数据热力图

为了更加详尽的展示每周七天的滴滴出行数据反映的滴滴车运行情况，统计一周内各天完成的行程数。如图 8 所示，其中每天的行程数是计算得到的 11 月的数据的均值，从中可以看到，每天的行程数大致在 194300 到 195100 之间，整体的浮动并不大，其中周五与周六的行程数最多。

图 8 一周内各天行程数

为了对数据做进一步的解读，绘制其中星期三、星期五以及星期六的各时间段的行程数图表，其中每个时间段跨度为两个小时，如图 9 所示。

图 9 通过控制台上传文件

通过比较星期三、星期五以及星期六的各个时间段的行程数，可以发现这几天的不同时间段的行程数的变化规律是基本相似的，星期六的数据与其余两天的不同之处在于星期六的行程数峰值是在 18-20 时这段时间内达到峰值，而其余两天行程数则是在 14-16 时达到峰值。同时图 8 中反映出，周六的行程数是一周内最多的一天，这也与生活常识相符，因为周六是休息日，而且第二天星期天也是不需要上班的，所以出行的行程数最多，而且峰值是在 18-20 时的黄金时间，也是出行的人群为了享受生活而出行的侧面印证，同时这也就要求交通管理部门要在星期六的黄金时间段安排更多的交通警察以及协警来加强对交通的管理，从而避免由于出行量大且傍晚光线较差而发生一些交通事故。从图 9 中还可以发现，在全天的各个时间段中，只有在 8-10 时这个时间段，星期六的行程数是明显少于其余两天的，这是因为 8-10 时是工作日的上班高峰时段，所以有很多行程对于上班族来说是不得不发生的，而休息日的时候没有了上班的压力，这个时间段的行程数就有了明显的减少，因此对于出租车师傅来说，这段时间可以用来休息，

如果出工的话也要找到距离上客热门区域更近的区域，从而提高接到客人的可能性。

为了找到上客热门区域，采用聚类算法与热力图可视化方法结合的方式。使用聚类算法是因为在位置数据中位置可以由经纬度表示，通过聚类可以将地理位置相近的位置点聚类到同一个簇中，聚类结果得到的多个簇则代表多个地理区域，其中包含位置点最多的几个簇就是需要找到的上客热门区域。由于地图上的数据点呈圆形以及一些不规则形状分布，所以属于非凸型数据集，因而聚类算法不能选择 k-means 及其部分变种算法，采用 DBSCAN 算法，这是因为该算法可以对任意形状的稠密数据集进行聚类，包括凸型数据集以及非凸型数据集；该算法可以在聚类的同时发现异常点，对数据集中的异常点不敏感；聚类结果没有偏倚，因为不需要像 k-means 算法那样指定聚类得到的簇的数目。但是 DBSCAN 在数据集较大时聚类收敛时间较长而且对于计算机的计算资源要求较高，所以本实验中不采用所有的数据进行聚类，因为内存资源不足以支持完成聚类。聚类的部分代码如下所示。

通过聚类可以得到附近的数据点最多的位置是锦江区的春熙路、盐市口、督院街、天府广场区域，其次是东门大桥、合江亭区域，再少一些的是青羊区的人民公园、汪家拐、少城区域。再通过热力图的扩放来找到上客点热力图中高亮较为密集的区域，也就是上客热门区域。一些热门区域如图 10 所示。

图 10 热力图中的上客热门区域

根据热力图中的内容可以看到，最热门的上客区域主要位于锦江区，其次是青羊区以及金牛区。所以推荐滴滴网约车签约司机主要活动在锦江区的春熙路、天府广场、高地中心、合江亭，青羊区的人民公园、城隍庙，金牛区的抚琴、营门口等热门区域，更有可能接到乘客的订单。

与此同时，出租车行程密集的区域也是交通难于管理的区域，所以建议交通管理部门可以抽调一些交通压力小的区域的人力来协助管理热门区域的交通状况。

4.2 城市规划建议

城市规划是处理城市及其邻近区域的工程建设、经济、社会、土地利用布局以及对未来发展预测的专门学问。本实验中涉及到城市范围较大，所以以一些局部地区为例作分析并且提出相应的城市规划的建议。

本实验选定的区域是成都站附近。之所以选定成都站，一方面是因为成都站历史悠久而且经过了很多次的改造，另一方面是因为成都站附近的上下客区域的热力图给人的第一印象是不符合对成都站不了解的人群的常识的。成都站附近的上下客区域热力图相似，所以选择其中一幅进行展示，其中下客区域热力图如图 11 所示。图 11 中可以看到，在成都站（即图 11 中的火车北站，成都站原名火车北站）16 万平方米的建筑面积上，滴滴车辆主要的出没的地点就在成都站东北角。

图 11 成都站区域下客点热力图

对于这个现象首先提出两个疑问，为什么成都站的北方要比南方发生了更多的滴滴车上下客事件？为什么在北方只有东北角有较多的滴滴车上下客事件？

为了解决第一个问题，首先对比成都站的南北广场，经过查阅资料可以得知北广场主要包括长途客运枢纽、公交场站、高架北广场。北广场有独立的出租车

上下客区和社会车辆停车场，主要承担公交、出租车、长途客运。南广场有出租车上客区和公交站场，主要承担地铁客运（https://baike.baidu.com/item/成都站）。南北广场的一个重要差距就在于南广场有地铁客运手段，而北广场没有，在现代化的城市建设中，地铁已经是人们出行的一个非常重要的选择，会分流巨大的人流量，所以成都站北方要比南方要有更多的滴滴车上下客事件。这暴露出的问题是北广场没有地铁交通，这会使得该侧的周边居民以及一些该侧出站的乘客的不便之处。所以应该在接下来的建设中为北广场添加地铁设施。另一方面，为了更好的分析南北两侧的差异，还查看了南北两侧的全景图，如图 12 所示。

图 12 成都站南北侧对比图

从图 12 中可以看到南北两侧的建设情况是有一定的差距的，这是因为 2015 年时，南广场以及东西两侧进行了棚户区改造，而北广场的改造还在进行中，这也是为北广场添加地铁设施的一个重要理由，因为广场改造完成，添加地铁设施可以将人流中的一部分引流到北侧，一方面可以降低南侧的客运压力和交通压力，另一方面也有利于北侧的新兴区域的发展。另一个为北广场添加地铁设施的重要原因是如果地铁建成，那么南北广场应该同属一个地铁站点，这样南北广场连通问题也就自然而然的解决了，可以避免重庆北站南北广场不连通那样的“悲剧”出现。所以可以为城市规划提出的一个建议就是在北侧增加地铁。

接下来分析第二个问题，因为考虑的是北侧这一侧的问题，所以查看北侧的全景图。如图 13 所示。

图 13 成都站北侧全景图

结合图 13 以及图 11 可以看到，成都站北侧之所以滴滴乘车的事件集中发生在东北角，是因为北侧有一面大围墙，而且围墙与外界连接的出口在东北角。这就导致了北侧出来的未使用成都站服务设施的公交、出租车、长途客运的乘客都会被集中到东北角。同时又可以从图 13（a）中看出，西北角区域与居民区连接紧密，所以在西北角开围墙增设出口的设想也不够实际，另外由于围墙的力学结构，在围墙中部开口也不现实，所以一方面是增设地铁设施，另一方面应该在现有的停车条件情况下增设停车位，并且在东北角区域也增设出租车、网约车管理的设施，比如指定上下客区域，增设围栏使乘客和车辆都更加有序。

成都市属于闻名的旅游城市，同时也是现代化程度较高的城市，所以为了更好的进行城市规划，还应该针对商业区、旅游景点、教育场所、其余的重要的交通枢纽等区域进行分析和民意调查。

除了热门区域的规划与改造以外，识别城市中的交通拥堵区域也是城市规划中一个热门研究内容。交通拥堵会导致通勤时间的大幅增加，影响居民出行计划，会影响驾驶人心情，使得他们心烦意乱，因此在一定程度上也加大了交通事故发生的可能性。因此如何找到城市中的交通拥堵区域，并且在道路政策、道路设计等方面对相应问题做出调整也是城市规划中的重要组成内容。

本案例中，利用车的行驶轨迹数据结合我国城市道路交通拥堵评价标准来找到成都市中的一些拥堵区域。将车速均速不大于 10km/h 的路段视为处在严重拥堵区域的路段。通过轨迹数据确定某一时刻车辆所在的街道与行驶车速，车速可以根据相邻轨迹点之间的距离与时间间隔计算得到，由于相邻轨迹点之间的采样时间较短，所以车辆行驶的距离不会过长，因此可以简化相邻轨迹点之间的距离的计算方式，将模型简化为一条直线路线，这样可以在计算距离时节省大量时间，

又因为采样时间间隔较短，所以产生的误差较小。查找拥堵路段时确定位置与速度的部分代码如下所示：

根据这种方式找到的一些拥堵路段有武都路、人民北路、人民南路、倒桑树街、红照壁街、锦里中路等。找到这些易拥挤的路段之后就可以结合道路的实际情况进行道路扩容、交通政策限流、收费限流等手段来改善交通拥堵的情况了。

应滴滴出行盖亚计划开放宣言共享原则的第七条的要求，特别感谢滴滴出行提供的数据，数据来源：滴滴出行“盖亚”数据开放计划。

案例中的图表和代码请参考：

赵卫东. 机器学习案例实战. 北京：人民邮电出版社，2019