**1.1数据集**

我们从饿了么公司获得了骑手的GPS轨迹数据以及对应的的订单情况数据。详细数据情况见表1。

|  |  |  |
| --- | --- | --- |
|  | **Rider GPS** | **Order** |
| Data Size |  |  |
| # of Daily Records | 2 thousand | 1 hundred |
| Format | Rider ID& Time GPS  Fare Collect Device Detail | Rider ID& Order ID  User& Shop GPS  Timesteps: Restaurant Accept Order & Rider Accept Order, Pick Up, Delivery, Arrival |

Table 1. Two Fleets

**• Rider GPS：**利用骑手手机APP平台在送餐途中自动地记录其GPS信息以及相应设备信息。记录数据包含id, longitude, latitude, time, device detail

**• Order：**对于订单数据，统计每个订单对应的餐厅接单时间、骑手接单时间、骑手到店时间、骑手取餐时间、订单送达时间。除此之外，还统计了骑手ID以及餐厅与客户的GPS信息。

**1.2数据处理**

首先，整理已经获得的数据，对数据信息进行筛选并删除重复的GPS时间戳。我们有对应时间戳的骑手GPS轨迹数据，订单始终点位置以及时间点数据。为了计算每个订单的送餐距离，我们需要提取订单递送骑手的轨迹信息。对此我们利用订单时间点对骑手GPS轨迹进行截取，为每个订单获得其相应的送餐轨迹，其匹配方法见表2。

Table 2. Match Method

|  |  |
| --- | --- |
| **Rider** | **Order** |
| GPS Time | Arrive Restaurant Time  Delivery Time |
| Shop GPS | GPS at Arrive Restaurant |
| User GPS | GPS at Delivery Finish |

在提取完成订单对应骑手GPS轨迹信息以后，对于轨迹中除开始时间外的每个时间戳计算骑手所处位置与前一个时间戳的骑手位置间距离，通过累加得到此订单的骑手送餐距离，具体计算公式如下：

匹配及计算完成后，将每个订单对应的送餐距离以及GPS时间戳信息统计并写入表格dest.csv。

**1.3异常分析**

我们将所遇到的问题分为两类，意料内问题以及异常，详见表3。

Table 3. Anomaly Categorization

|  |  |
| --- | --- |
| **Anomaly Categorization** | **Anomaly Examples** |
| Expected  Unexpected | rider free, timestep loss  redundant order |

冗余订单：结果中有送餐距离为0的订单，及此订单无法截取相应GPS时间戳。匹配失败原因是：在取餐（送达）时间骑手未到达订单开始（终止）位置。由于GPS信息是对骑手位置的唯一表示，而平台是记录的订单时间点是骑手点击确认的时间，实际情况下可能出现骑手到达前点击确认的情况，此时平台因判断骑手位置阻止其提前点击。