**1 DATASETS**

We have access to two order fleets, which is from ele.me company. The details about the data are given in Table. 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: Shop Accept Order & Rider Accept Order, Arrive Shop , Pick Up, Arrival User |

Table 1. Two Fleets

**• Rider GPS:** We access Rider GPS trace data recorded by an APP platform, which is mandatory to be installed on every riders’ phone. Therefore, one trace data includes id, longitude, latitude, time, device detail.

**• Order:** The data we get was collected from the company’s database, and was carefully selected to match each order. One order contains rider-id, tracking-id, shop-id, user GPS, shop GPS, timestamps (shop accept, rider accept, rider arrive shop, rider pick up, rider arrive user)

**2 METHODOLOGY**

This section describes how we divide rider GPS track to match each order to get the delivery distance. We first show the behavior of order’s location points via rider’s GPS track with a simple example. Second, we elaborate on timestamp to infer a matching method. Finally, we use the matched GPS track fragment to calculate delivery distance.

**2.1 Simple Example**

To illustrate how does the matching works in our system, Fig. 1 gives an GPS track map of rider and one of the orders which was being delivered at that time. We find some special features to ensure a correct match. The rider is near shop location when GPS time was arrived shop time. The order travel along with rider before delivery finish.

Fig. 1. Location points via GPS track

|  |
| --- |
| Blue line: Rider gps track  Left green point: Shop location  Right green point: User location |

**2.1 Matching Method**

In order to calculate the delivery distance for each order, we need to match the trajectory information of delivery rider with every order. The items we pick up for matching are given in Table 2.

Table 2. Match Items

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

Using these items, we can divide GPS track into different fragments for each order (Attention that this is a one to multiple match, because rider can deliver more than one order at the same time). After that we get our matching result, format like Table 3.

Table 3. Matching Format

|  |  |
| --- | --- |
| **Order** | **Rider** |
| Tracking ID | Rider ID  GPS Timestamp |

**2.1 Distance Calculation**

In this subsection, we describe the calculation method that turn GPS timestamp into delivery distance.

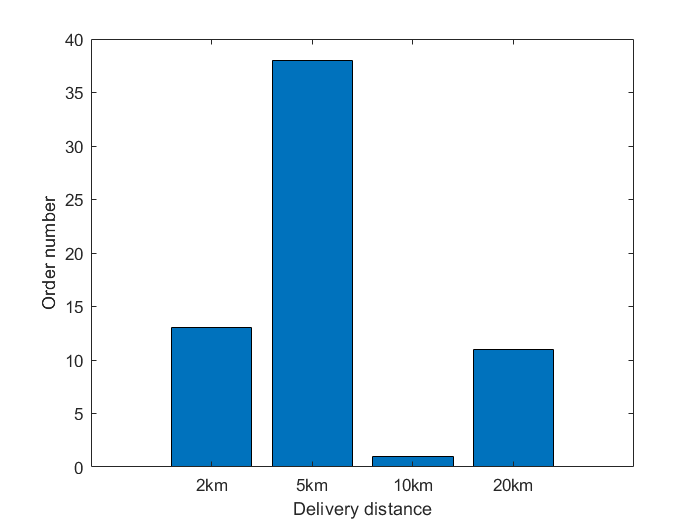
After extracting the rider GPS track information corresponding to the completed order, calculate the distance between the position of the rider and the rider position of the previous position for each of the tracks except the start time, and accumulate the rider's meal of the order by accumulating Distance, the specific calculation formula is as follows:

After the matching and calculation is completed, the meal distance and GPS timestamp information corresponding to each order are counted and written into the form dest.csv.

**3 MEASUREMENT RESULTS AND ANALYSIS**

In this section, we first briefly evaluate our matching method with detail data since it is the foundation of delivery distance calculation. We generally divided the delivery distance into four levels 2km, 5km, 10km, 20km. The order number in different levels are shown in Fig. 2.

Fig. 2. Delivery distance



We will divide the problems encountered into two categories, expected problems and unexpected anomalies.in Table 3.

Table 3. Anomaly Categorization

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

Redundant order: There is an order with a delivery distance of 0 in the result, and the corresponding GPS timestamp cannot be intercepted for this order. The reason for the match failure is that the rider did not reach the start (stop) position of the order during the meal (delivery) time. Since the GPS information is the only representation of the rider's position, and the platform is the recorded order time point time of the rider's click confirmation. In actual circumstances, the rider may click to confirm before the arrival, and the platform prevents the advance ride by judging the rider's position.