Data Science Take-Home-Assignment Report

William Wong

October 11, 2017

1 Detection of Stops

1.1 The Kalman Filter

To detect when a user stops, I primarily use a Kalman filter to track his/her position along with the velocity in two dimensions. The (hidden) internal state $X(t)$ evolves according to

$$X(t+1) = FX(t) + W(t)\,, \tag{1}$$

where $X(t)$ contains the two dimensional longitude and latitude, along with the velocities

$$X(t) = \begin{bmatrix} x_1(t) \\ x_2(t) \\ v_1(t) \\ v_2(t) \end{bmatrix}\,. \tag{2}$$

F is the forward-transition model

$$F = \begin{bmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}\,, \tag{3}$$

and $W(t) \sim \mathcal{N}(0, Q(t))$.

The observation at time t is $Y(t)$

$$Y(t) = HX(t) + V(t)\,, \tag{4}$$

where $V(t) \sim \mathcal{N}(0, R(t))$.

We use the `pykalman` library in Python to implement the Kalman filter. We consider a user has stopped when the absolute value of the estimated rates of change are both less than some parameter.

Sample results obtained from the Kalman filter are shown in Fig. 1, where the following quantities are plotted as a function of time:

1. raw latitude data and the estimated latitude.

2. the estimated rate of change of latitude.
3. raw longitude data and the estimated longitude.
4. the estimated rate of change of longitude.
5. whether the user has stopped.

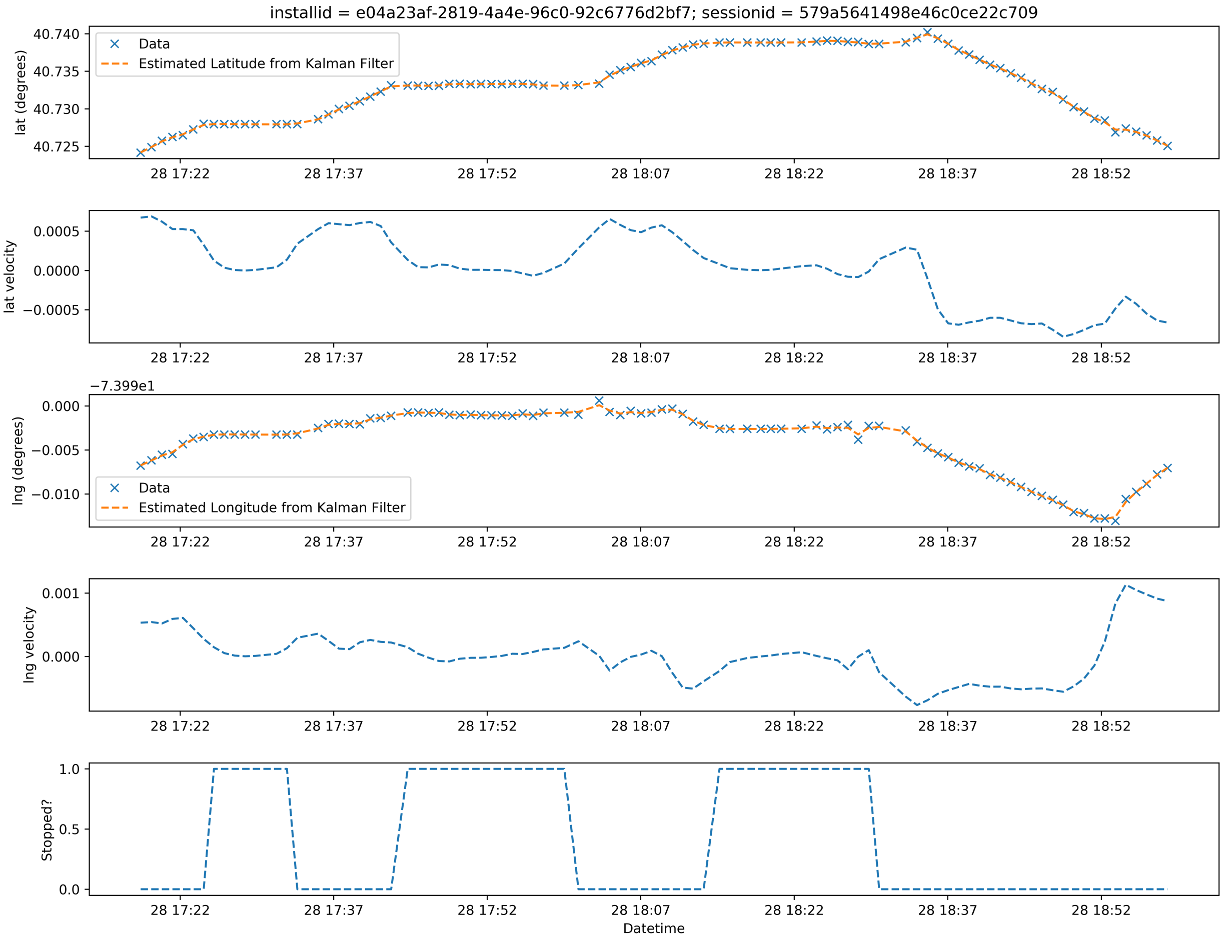

Figure 1: Sample results obtained from the Kalman filter.

1.2 The DBSCAN Algorithm

We also applied the DBSCAN algorithm in Python to estimate the clusters when users stop. I use the haversine metric and the ball-tree algorithm to calculate great-circle distances between points. Sample results obtained from using the DBSCAN algorithm are shown in Fig. 2

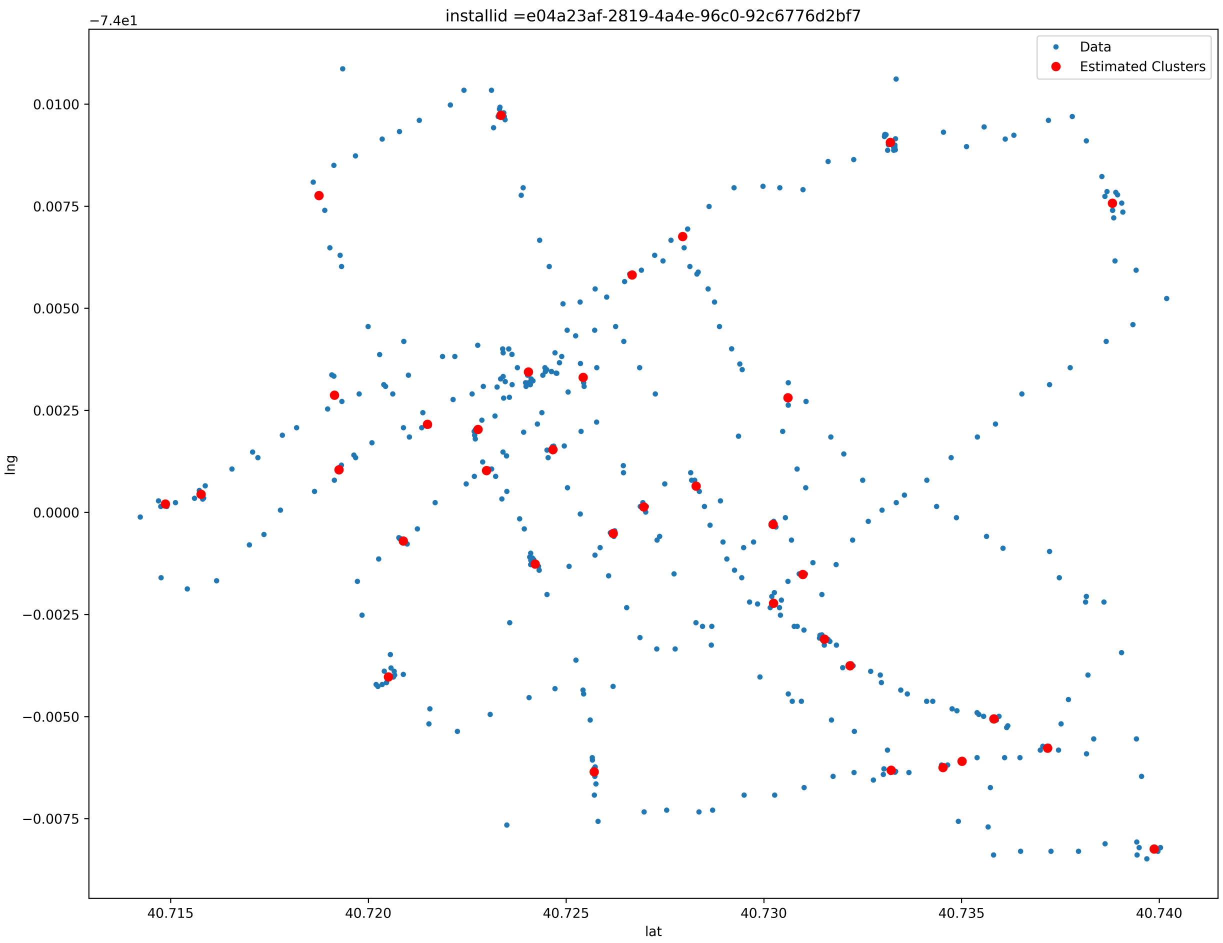

Figure 2: Sample results obtained from using the DBSCAN algorithm.