Paper Methodology

Ivan Mendoza Vázquez / Gustavo Álvarez Coello / Andrés Baquero Larriva

2023-06-25

Home Detection Methodology

The Resulting locations of the student homes have to be mined from mobility data after these are processed following a multi-step procedure. The required steps are summarized through the flow chart found in 1, then a deeper explanation follows in the next sections.

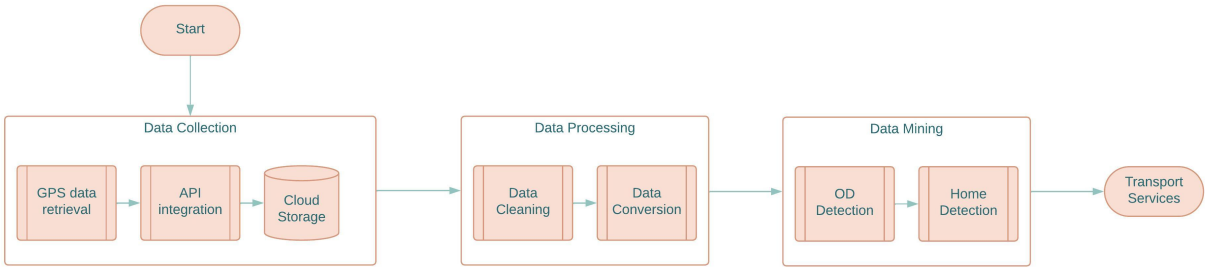


Figure 1: Multi-step procedure methodology for Big Data Home Detection

In a few words, data are collected by a mobile app which uses the Global Positioning System (GPS) sensors of the mobile device, so that they can be stored in a remote server via an Application Programming Interface (API). Then, these unprocessed data are cleaned to filter out outliers and transformed to a suitable formaat for further calculations. At last, origins and destinations (OD) are detected by some segmentation algorithm and later classified as potential home locations. Some potential transport services after acquiring this information are planned out to complete the flow.

Mobile Data Collection

The first step, assuming a mobile app exists so that it can collect GPS data, is storing mobility data together with temporal and identity information in a remote database. Each observation i (a labeled spatiotemporal data point) will have the following structure:

$$p_i = (x_i, y_i, t_i)$$

that is, in its simplest form, it consists on the location's coordinates (x, y) of the tracked user at a specific time stamp t. A more complete version of the observation is:

$$p_i = (uid_i, lat_i, lon_i, alt_i, date_i, t_i, dow_i, acc_i)$$

- *uid*, is a user's unique identifier, normally a MD5 hash string which guarantees data is collected anonymously, but it still makes possible to know the user in the database.
- *lat. lon, alt*, the point's latitude, longitude and altitude coordinates.
- *date*, a day-month-year string
- *t*, a 24-hour local time format for a continuous variable after applying this formula:

$$t = hours + \frac{minutes}{60} + \frac{seconds}{3600}$$

- *dow*, the day of the week the observation was captured, where 1 is Sunday
- *acc*, accuracy in meters of the measurement as reported by the sensor API, lower values produce more accurate measurements.

The set P_a of data point observations collected in real-time by a user's mobile device, is stored in this format in a remote server for further off-line procedures. However, this format is not yet suitable for most calculations, so that further processing is required as described in the next steps.

Data Processing

In order to avoid bias in the calculation of travel destinations, outliers are removed considering position accuracy and temporal constraints. A subset P_a consists of more refined "valid" observations, selected through the following criteria.

$$P_b = \{p_i \in P_a | acc_i < \alpha\}$$

where α is a parameter that denotes the maximum allowed accuracy error in meters. Moreover, users can travel to destinations found in other regions, countries and even continents. For the purpose of this research, home locations must be found inside a study region at a medium-sized city level, so a bounding box defined by $[lon_{min}, lon_{max}, lat_{min}, lat_{max}]$ constraints the valid observation previously found.

$$P_c = \{p_i \in P_b | (lat_{min} < lat_i < lat_{max}) \wedge (lon_{min} < lon_i < lon_{max})\}$$

Because this research is intended to provide insights about transport services for students in a College Community, a time period where users are expected to regularly visit the campus must be selected. Then, a cutoff date defined by $[date_{min}, date_{max}]$ is used to select the final subset P.

$$P = \{p_i \in P_c | (date_{min} < date_i < date_{max})\}$$

At last, since OD detection requires computation of distances, a Cartesian projection is a better approach. For the selected study region the UTM-17S is chosen, so that longitude, latitude and altitude measures are transformed to euclidean space coordinates (x, y, z). Since dates and accuracies are not needed anymore, each observation p_i in the resulting processed set P becomes:

$$p_i = (uid_i, x_i, y_i, z_i, t_i, dow_i)$$

Table 1: Sample observations from the Spatiotemporal Dataset

	uid	x	y	z	t	dow
5	614820f9e6	717002.3	9677061	2616.9	23.30222	2
6	614820f9e6	717002.8	9677062	2616.9	23.30444	2
7	614820f9e6	717007.8	9677053	2616.9	23.30500	2
8	614820f9e6	717007.8	9677053	2616.9	23.30500	2
9	614820f9e6	716998.5	9677059	2616.9	23.33444	2
10	614820f9e6	716998.5	9677059	2616.9	23.33444	2
11	614820f9e6	716998.2	9677059	2616.9	23.33500	2
12	614820f9e6	717175.5	9676911	2617.3	6.65667	3
13	614820f9e6	717175.5	9676911	2617.3	6.65667	3
14	614820f9e6	717175.5	9676911	2617.3	6.65667	3

Dataset Description

The full dataset used in this study consists of spatiotemporal data, collected by a dedicated tracking mobile app for a College Community. It involves data from *728* users during one moth period, namely from May 20 to June 20, 2023. The results of the further analysis have proven 30 days of mobility data per user to be sufficient to identify home locations. The number of observations after the data cleaning process is above *11 million*, so that good infrastructure of the cloud storage is required to handle this volume of big data.

A sample from the spatiotemporal dataset is presented below for the five attributes mentioned earlier:

The bounding box for the region of Cuenca, Ecuador contains longitudes between -79,084789233 and -78,933588295, and latitudes between -2,938030323 and -2,865347073. A picture of the sample of collected points on a map at scale 1:50000 is given in 2, where it is shown that complete travel trajectories can be retrieved from data. The sampling frecuency of the GPS, that is the time difference between measures was not fixed but around 2 seconds on average.

Figure 2: Sample of collected points for the region of Cuenca, Ecuador.

Table 2: Sample observations from the Spatiotemporal Dataset

	uid	x	y	z	t	dow	distance	dt	speed
2	614820f9e6	717002.8	9677062	2616.9	23.30444	2	0.0006438	0.00222	0.2900193
3	614820f9e6	717007.8	9677053	2616.9	23.30500	2	0.0104787	0.00056	18.7118802
5	614820f9e6	716998.5	9677059	2616.9	23.33444	2	0.0112072	0.02944	0.3806784
7	614820f9e6	716998.2	9677059	2616.9	23.33500	2	0.0002437	0.00056	0.4350924
11	614820f9e6	717228.9	9676883	2617.3	6.65778	3	0.0602266	0.00111	54.2582027
12	614820f9e6	717271.2	9676865	2617.3	6.65861	3	0.0456285	0.00083	54.9740577
16	614820f9e6	717304.9	9676851	2617.3	6.65944	3	0.0365996	0.00083	44.0958588
17	614820f9e6	717338.8	9676843	2617.3	6.66056	3	0.0350002	0.00112	31.2501966
18	614820f9e6	717375.9	9676831	2617.3	6.66167	3	0.0390131	0.00111	35.1468991
19	614820f9e6	717408.8	9676819	2617.3	6.66250	3	0.0349841	0.00083	42.1494802

Computing new attributes

Some additional features must be added to the existing dataset so that travel behavior can be evaluated.

Let p_i be the i^{th} observation in a spatiotemporal dataset P. Then, the cumulative distance $D_{a,n}$ for a trajectory starting at point a and consisting of the next n observations is defined as:

$$D_{a,n} = \sum_{i=a+1}^{a+n} d_{i,i-1}$$

where $d_{i,j}$ is the euclidean distance between two observations.

$$d_{i,j} = ((x_i - x_j)^2 + (y_i - y_j)^2)^{1/2}$$

that is, the sum of distances between proximate points; then, instant speed at observation i^{th} is computed by:

$$s_i = \frac{d_{i,i-1}}{t_i - t_{i-1}}$$

The resulting extended dataset has the following structure after making the mentioned computations, and then filtering out consecutive measures taken at the same time stamp (possibly duplicates); also the very first observation has to be removed in order to avoid division by zero in the speed calculation. Distances have been transformed to *km* so that speed unit is *km/h*.

Finally, a last filtering procedure removes observations with extreme speeds, as values beyond *130km/h* are very improbable (by checking speed limits within the studied region). These data are often related to extreme distances between proximate points, which can occur due to GPS bad measures.

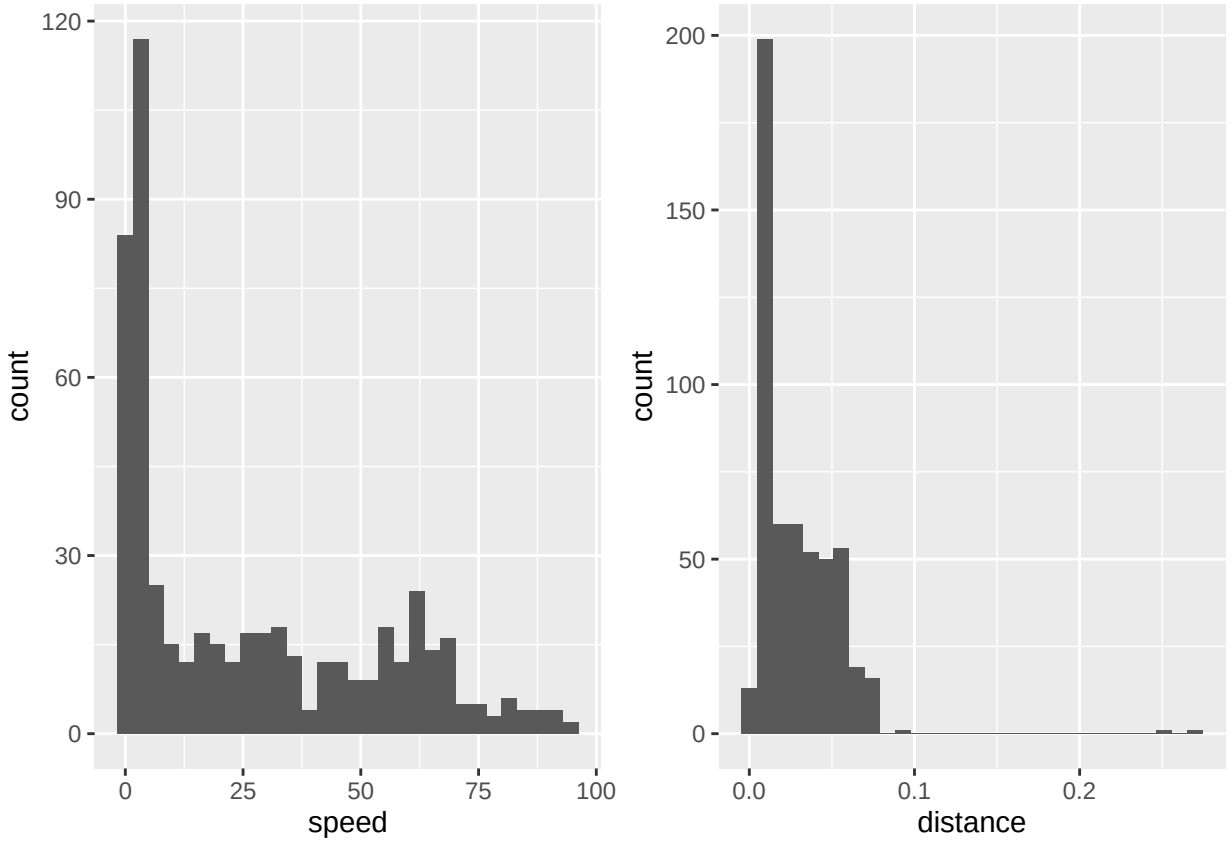
Data Mining

Data traces of each user must be segmented into individual travels (geometries) so than OD's can be found at the start and end points. In order to aggregate data into trajectories, an heuristic considering low-speed hot spots as destinations is now described.

Taken a single day displacements for a single user, the speeds and distances between proximate points variations that take place when traveling, staying on destination, changing to a different travel mode may allow to detect a trip's endpoints. The speed and distance distributions are shown in @ref(fig:speed-plot).

Table 3: Sample trips for one single user and day

uid	tripid	ox	oy	dz	departure	dx	dy	dz	arrival	tdistance	t
614820f9e6	1	717228.9	9676883	2617.3	6.65778	722204.0	9677167	2540.9	7.03611	6.2172335	0.
614820f9e6	2	722212.4	9677154	2539.8	8.66944	722206.1	9677236	2538.8	9.12222	0.4251312	2.
614820f9e6	3	722207.4	9677177	2538.0	11.11917	722225.2	9677174	2539.6	11.26194	0.1162438	2.
614820f9e6	4	722213.5	9677163	2540.7	12.24444	722191.8	9677168	2544.8	13.12417	0.7033840	1.
614820f9e6	5	722207.1	9677172	2545.2	14.10444	716997.5	9677065	2617.0	16.63361	7.7671695	3.



As seen in the previous figure, most of the speeds and distances are closer to zero, probably because users spend most of their time walking or staying in one place before starting the next trip. The changes in speed during the day can be seen in @ref(fig:speed-hour-plot).

This means that, data points can be segmented into individual travels by detecting those locations when moving at very low speeds (staying still), with respect to a given speed tolerance ϵ.

It can be assumed that actual destinations are found in intervals where users are not moving for a minimum amount of time t_{min}, that is in the "valleys" shown in last figure; in contrast to traffic lights that will also produce zero speeds but will last only a few seconds. The following plot allows appreciating points merged into trips (clusters) for ϵ=1km/h and t_{min}= 12 minutes @ref(fig:speed-hour2-plot). Increasing t_{min} will merge nearby trips into larger ones

At last, the fist (oldest) and last (newest) point in each cluster indicate the origin and destination, as well as the departure and arrival times (from the time stamps of these points). The travel time is simply the difference between the arrival and departure times; also, the cumulative distance of all points in a cluster indicates the trip travel distance. The following table gives a glimpse of the aggregate date into individual travels.

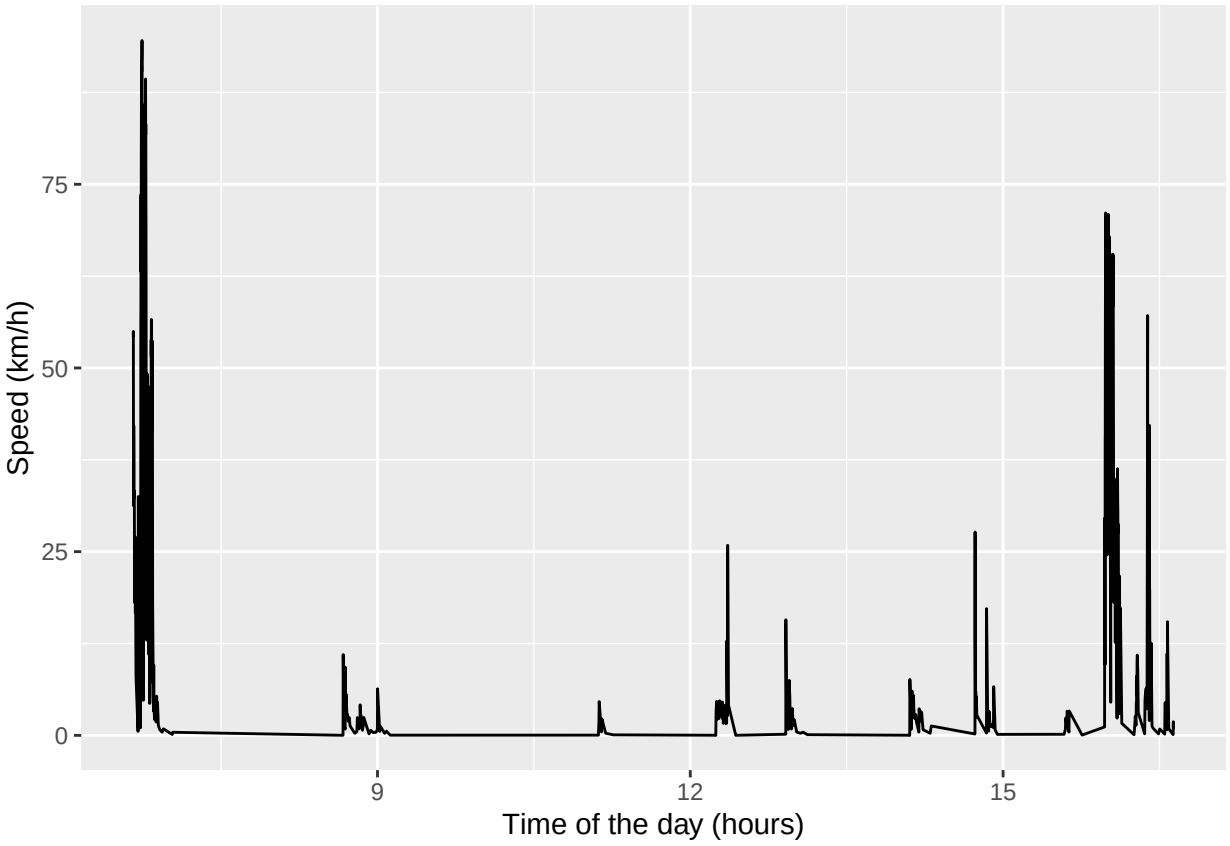


Figure 3: Speed variations along a single day.

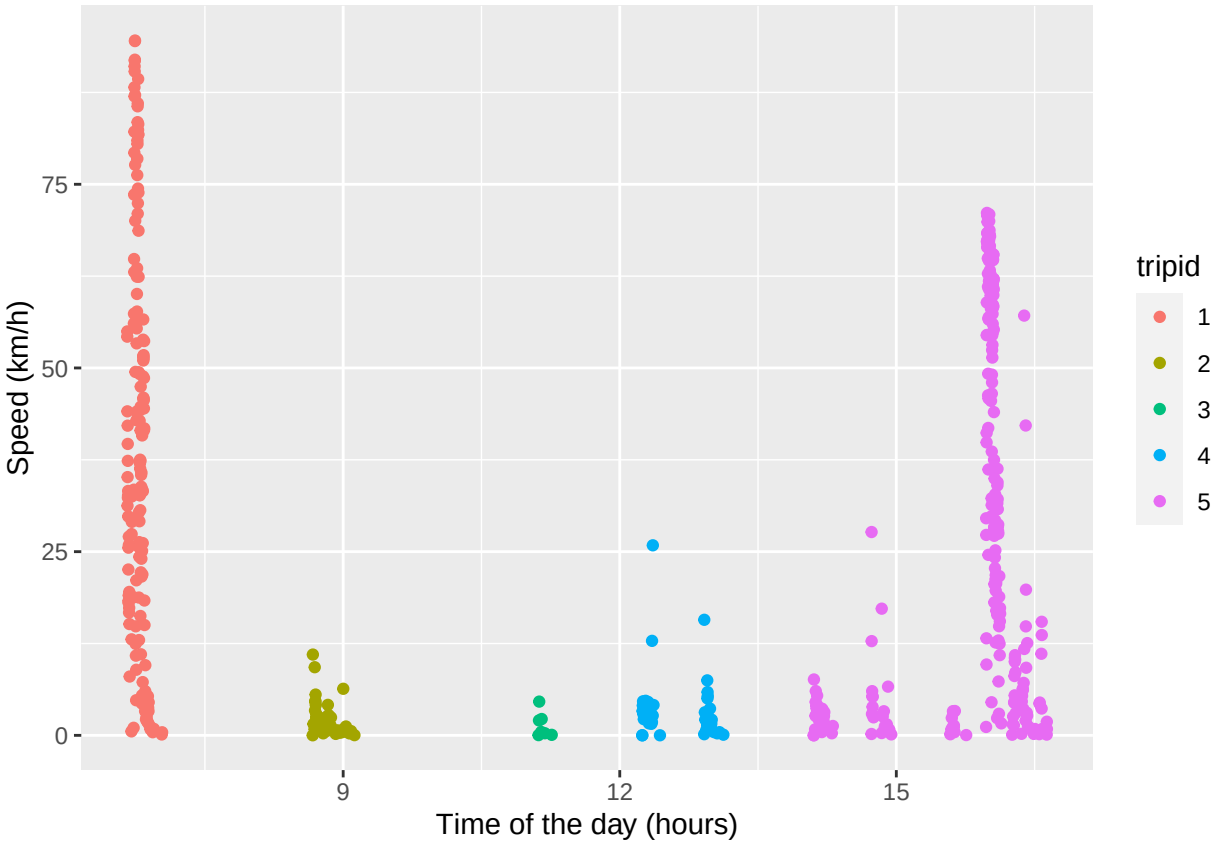


Figure 4: Speed variations along a single day.

The resulting segmentation allows OD's to be detected. Their coordinates are given in the table as attributes "dx" and "dy" for destinations locations, and "ox" and "oy" for the origins. Figure 2 presents on a map at scale 1:25000 the resulting user's destinations (as red spots). It can be noticed that as locations are repeatedly visited, some points could be merged into a single destination, this can be done by density-based clustering techniques; however this is not necessary for the upcoming analysis.

Figure 5: Sample destinations for one single user and day.

Home Detection

After destinations have been detected, another heuristic can be used to identify a user's home.