Map Matching algorithm

Design purpose

Match the raw GPS data to the corresponding road segments. So, it takes GPS signals as the input, and Positions on a road network as the output.

GPS signals (component)

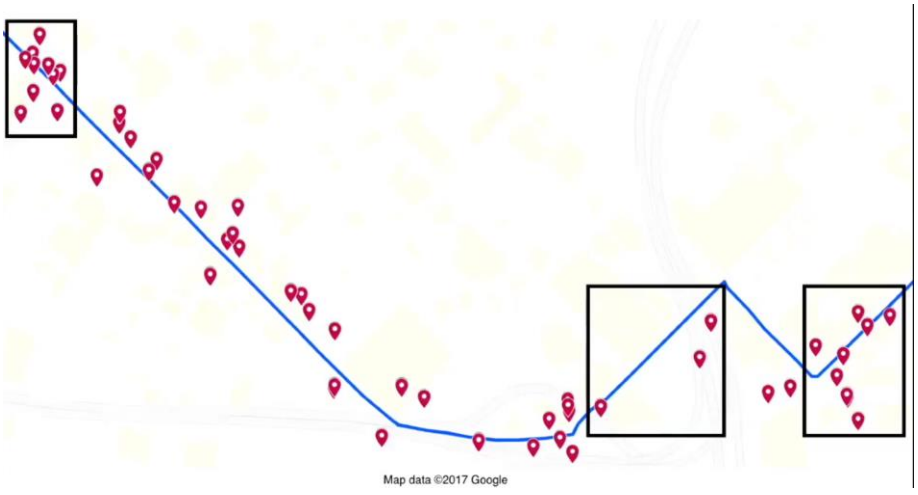
1. Latitude
2. Longitude
3. Speed (how fast it travels)
4. Course (the direction of travels)

Positions on a road network (component)

1. Latitude (On an actual road)
2. Longitude (On an actual road)
3. Road segment ID
4. Road name
5. Direction / heading

Why we need map matching algorithm

Reduce the noise and sparseness of raw GPS data

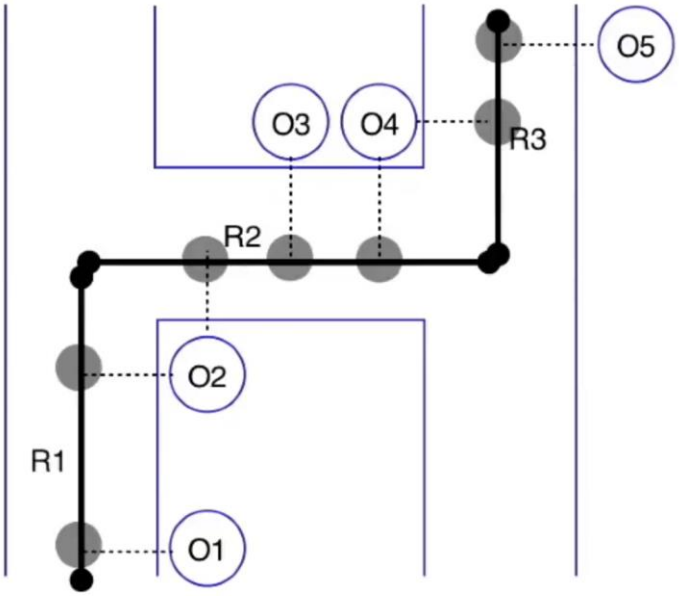


Use cases

1. Online map matching: driver position

2. “offline” map matching: Fare calculation

Map Matching + Hidden Markov model: Candidate’s selection



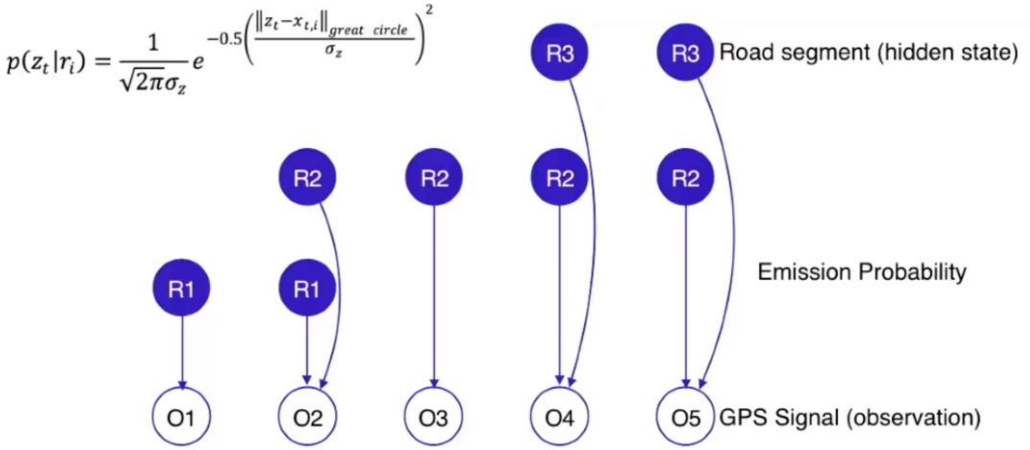
1. Use the k-nearest neighbor look the road candidates for GPS signal using a geospatial index (R-tree or Kd-Tree) – also have a minimum search radius
2. Project GPS signal to the candidate road segment (perpendicular drawn from GPS to road)
3. Setting up the hidden Markov model and calculate the emission probability

$$z_t \; is \; the \; GPS \; signal, \; x_t \; is \; the \; projected \; point \; on \; the \; road$$

$$\sigma_z \; is \; the \; sd \; of \; Gaussian \; GPS \; noise, which \; need \; to \; be \; estimated$$

Map Matching + HMM: Emission Probability

$$p(z_t|r_i) = \frac{1}{\sqrt{2\pi}\sigma_z} e^{-0.5\left(\frac{\|z_t - x_{t,i}\|_{great\ circle}}{\sigma_z}\right)^2}$$



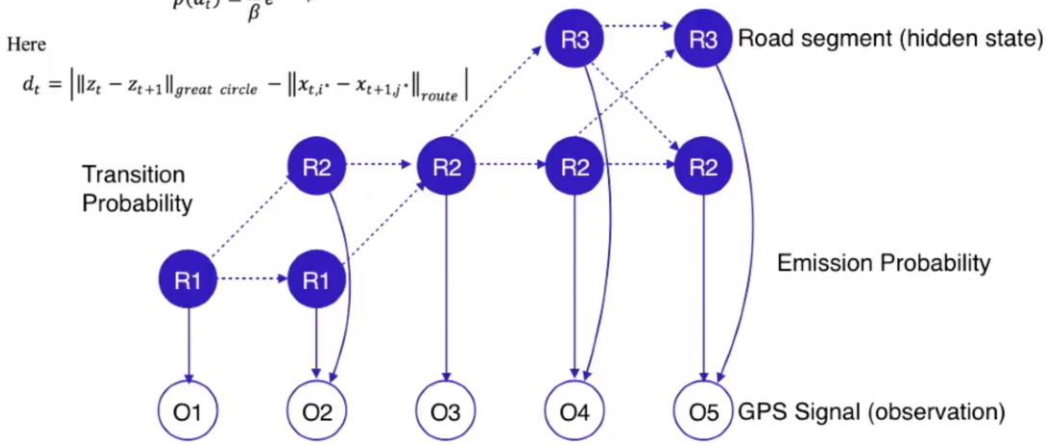
4. Calculate the transaction probability

Map Matching + HMM: Transition Probability

$$p(d_t) = \frac{1}{\beta} e^{-d_t/\beta}$$

Here

$$d_t = \left| \|z_t - z_{t+1}\|_{great\ circle} - \|x_{t,i} - x_{t+1,j}\|_{route} \right|$$



5. Using the Viterbi Algorithm to decode the possible path