THE UNIVERSITY OF TEXAS AT DALLAS

Driving pattern analysis of Emergency Vehicle (EV)

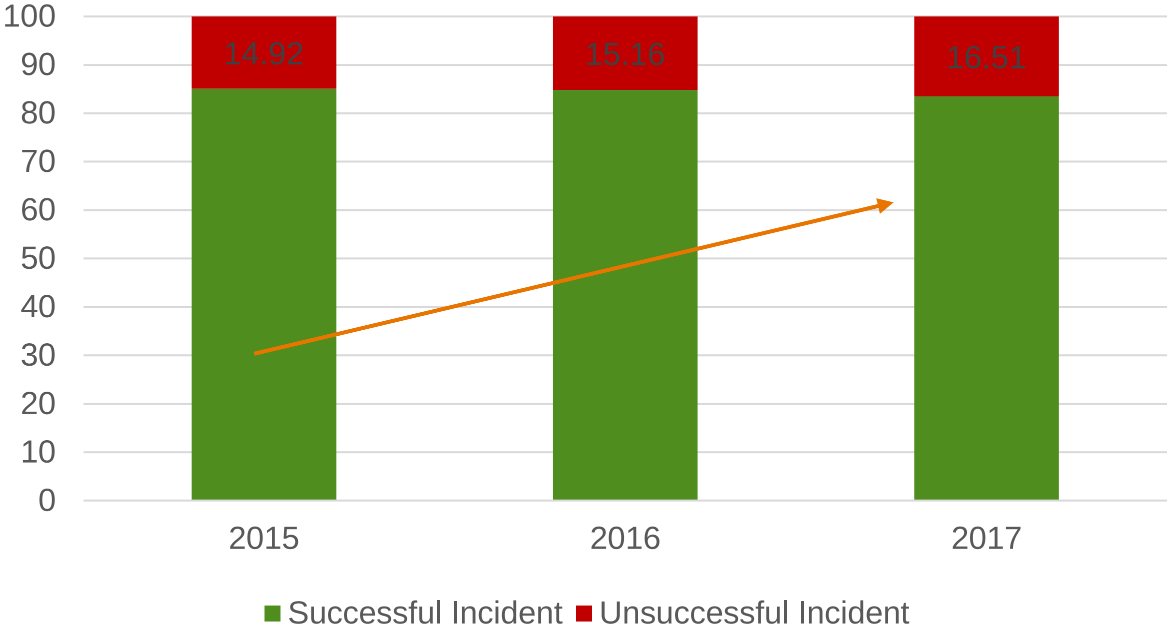
Yalin Yang, Yanan Wu, Dr. May Yuan, Geospatial Information Sciences

Background Introduction

- Response time?

8 minutes

Arrive



- Successful Incident
  As long as one EV arrives within eight minutes.

- Unsuccessful Incident
  All dispatched EVs fail to arrive within eight minutes.

Research Purpose: What causes the delay of emergency vehicles

Data

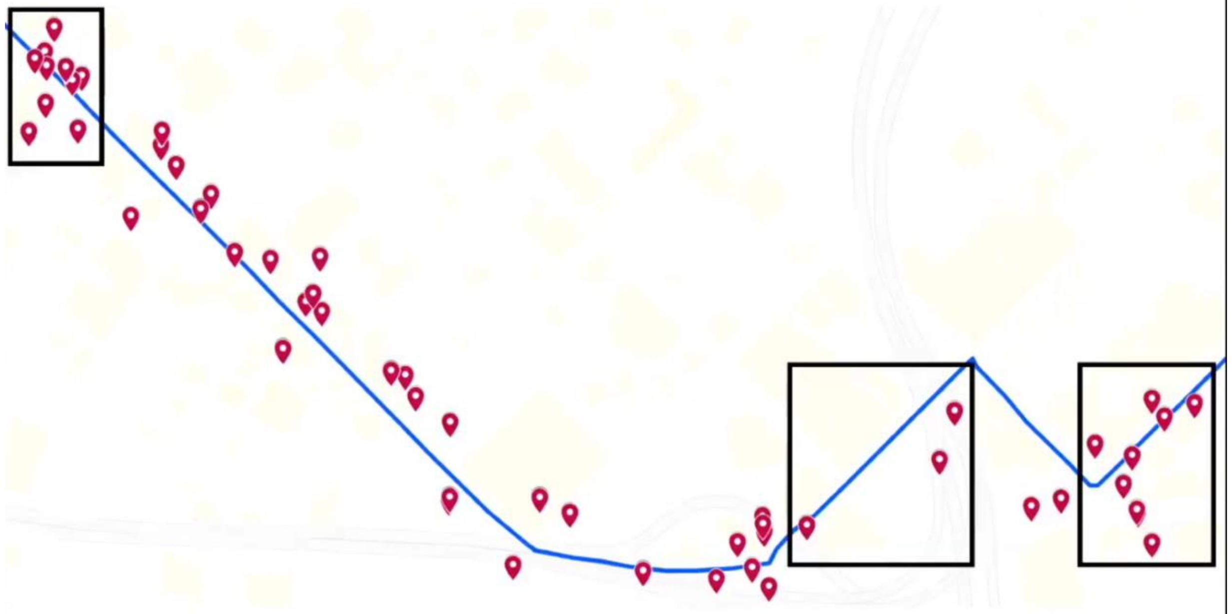
- From October 2015 to November 2017
- 2,325,360 GPS Records From Emergency Vehicles
- 887,825 Emergency Runs.
- 532,653 Incidents.

Research Question:

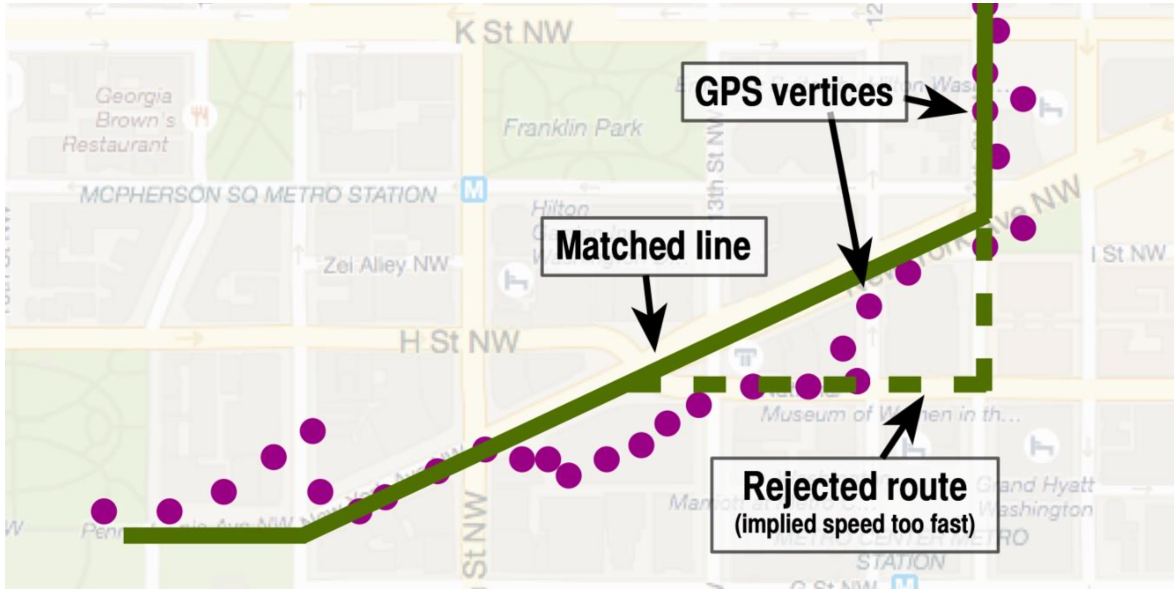
1. What information we can get from these records?
2. How can we achieve a better response strategy?

Step1: Restore trajectories from GPS records (Map-Matching)

- Given a sequence of GPS signals, find the most probable sequence of road segments
  - Noise (Random effects)
  - Sparseness (Signal of Cross Positioning)
  - Physical Constrain (Speed / Road Property, etc.)



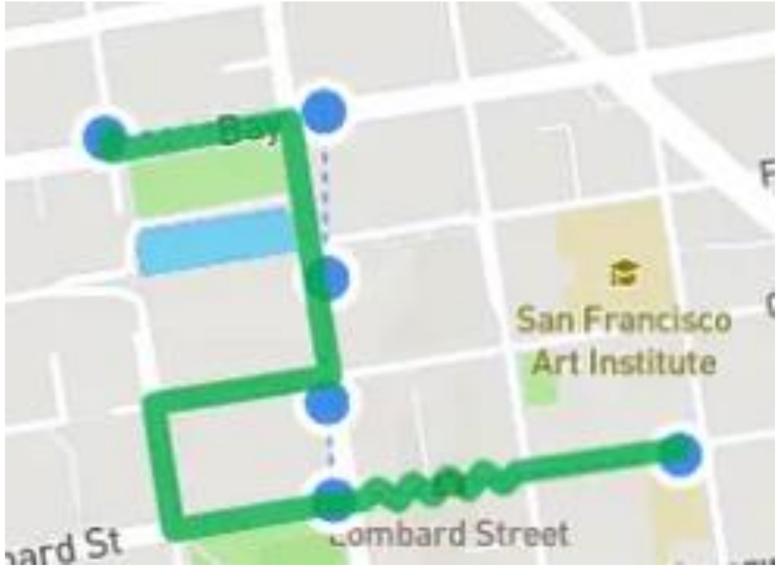
Map data ©2017 Google

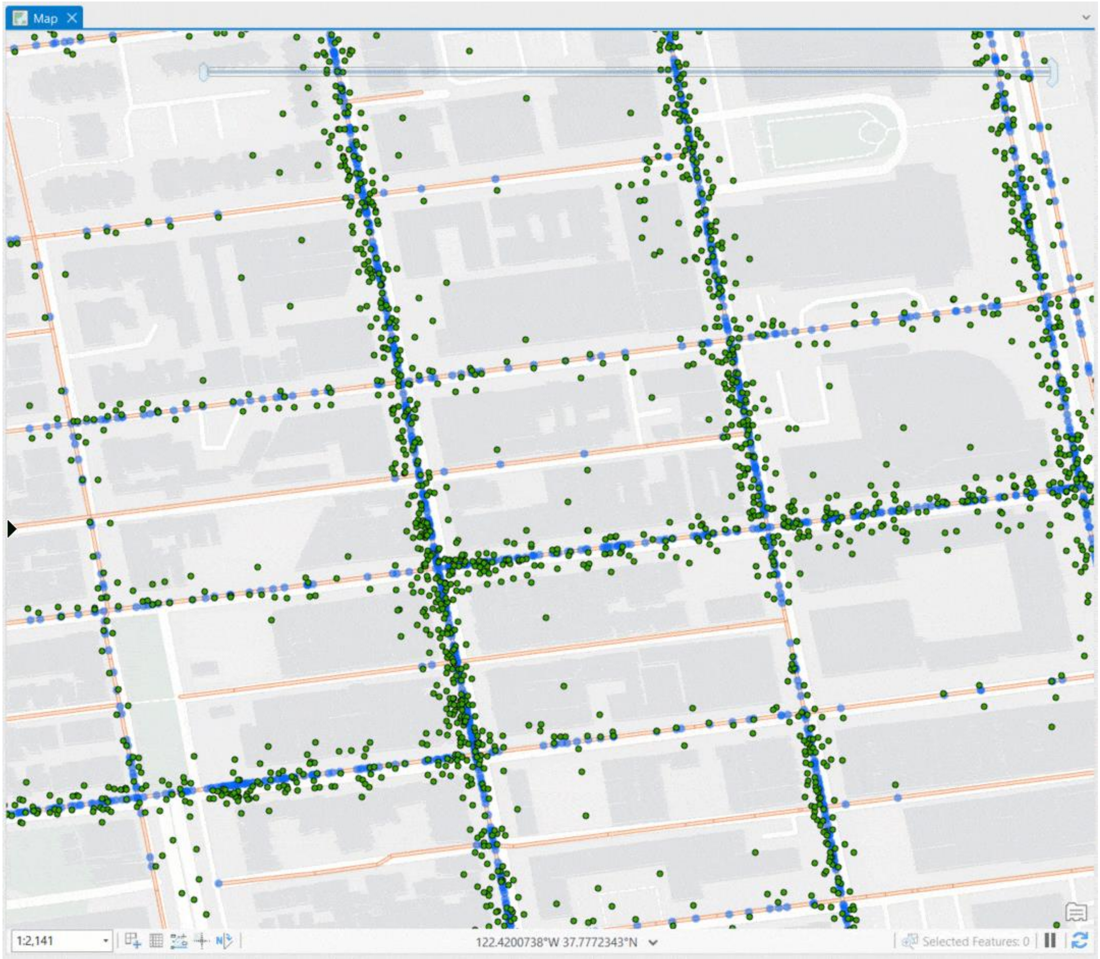


Naïve way:

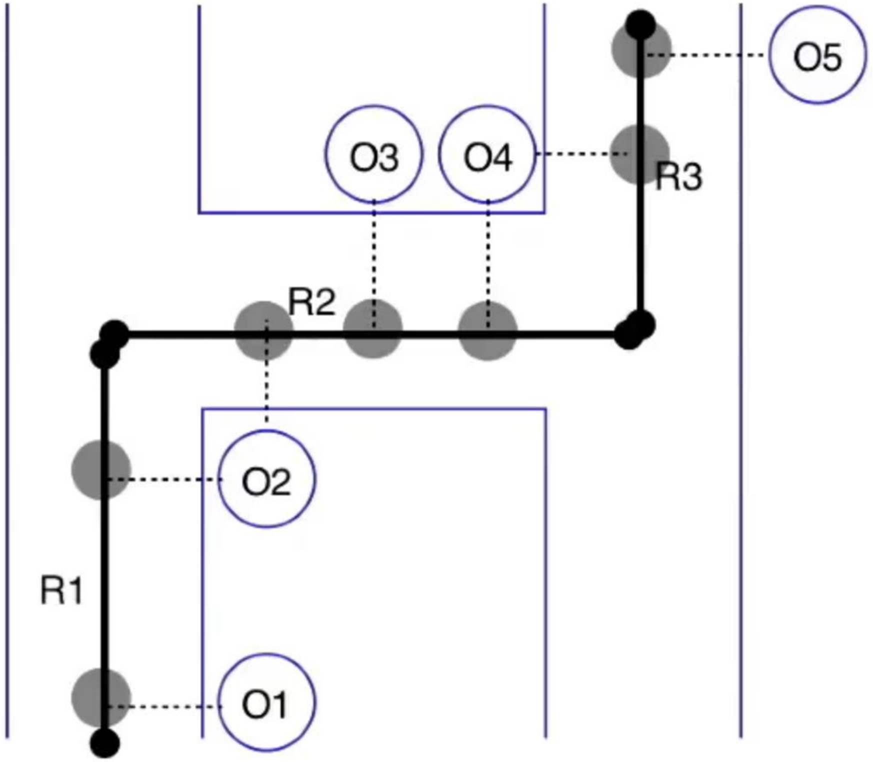
1. Snap(Project) GPS Points to the closest road with a distance threshold

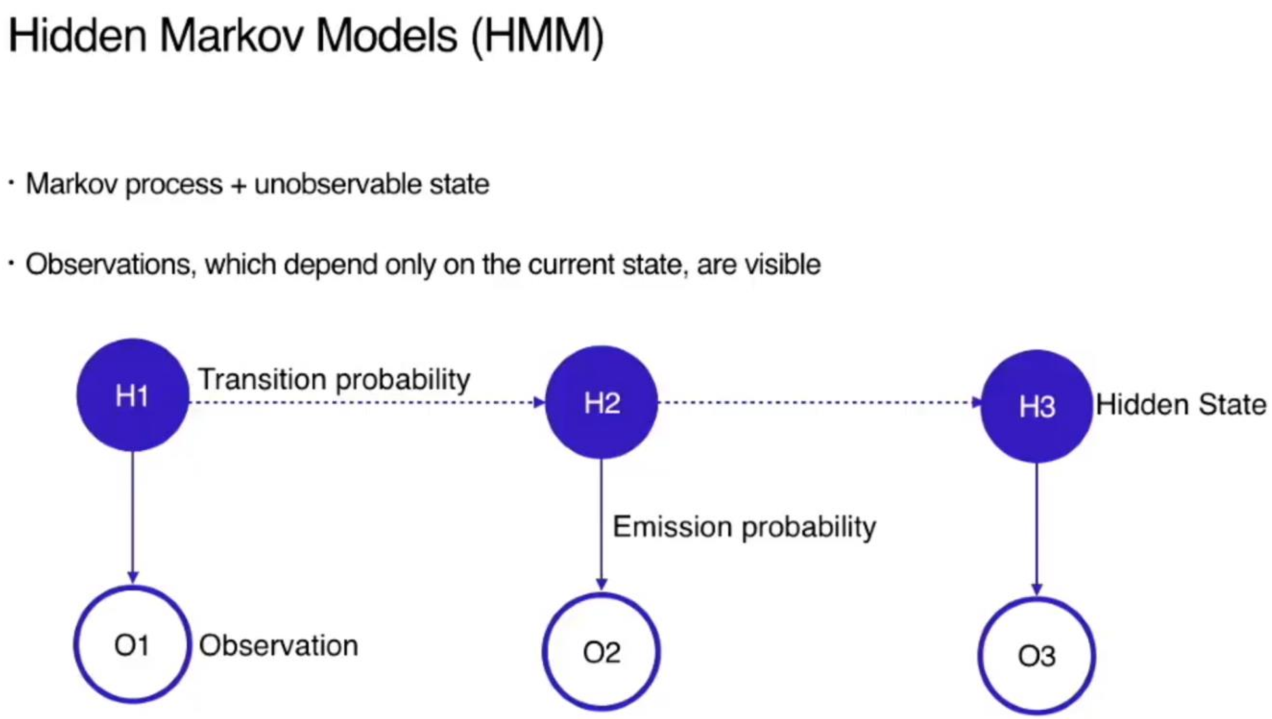
    1. Wired Path generated on a nested road network





Map-Matching using the Hidden Markov Model (Presented by Uber, 2017)





Observe States: GPS points
Hidden States: Points on Roads (Projected Points)

Emission Probability
Gaussian distribution of distance

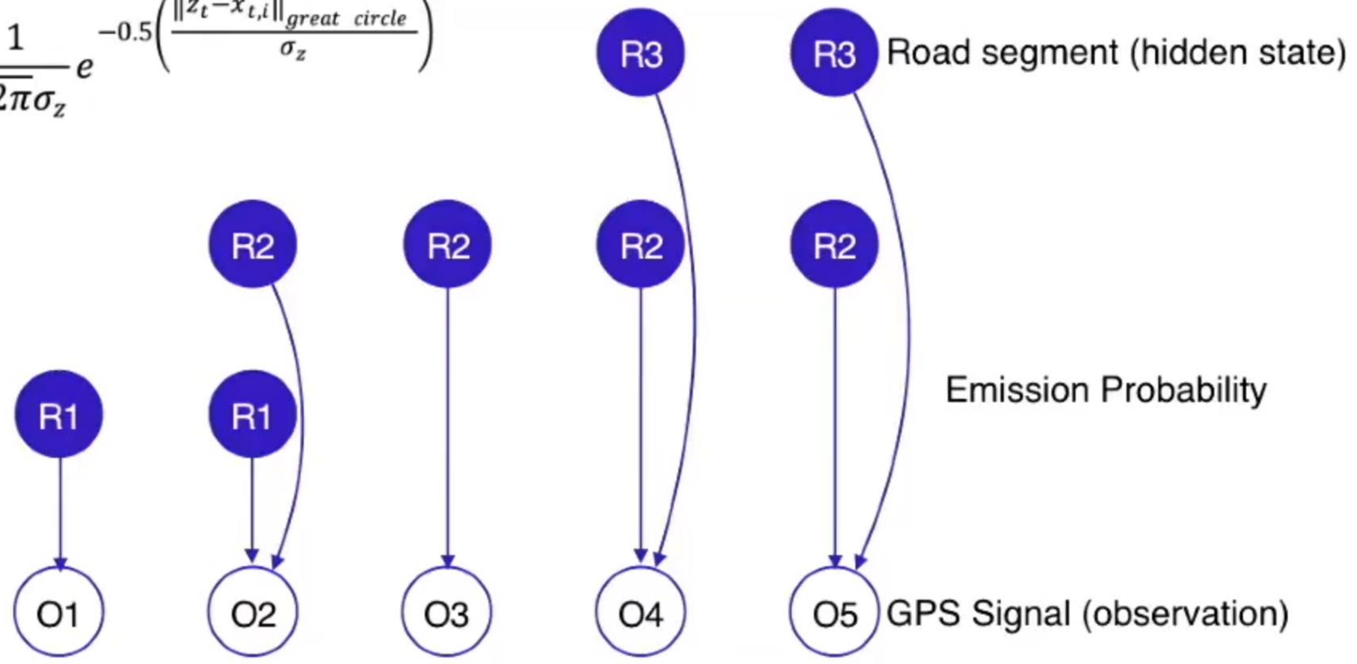
z_t: *GPS signal*
x_t: ***projected point*** *on the road*

Only consider within 200 meters

$$\sigma_z = 1.4826 \ \mathrm{median}_t \left(\left\| z_t - x_{t,i^*} \right\|_{great\ circle} \right) \qquad (5)$$

For our test data, this value was $\sigma_z = 4.07$ meters, which is a reasonable value for GPS noise.

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



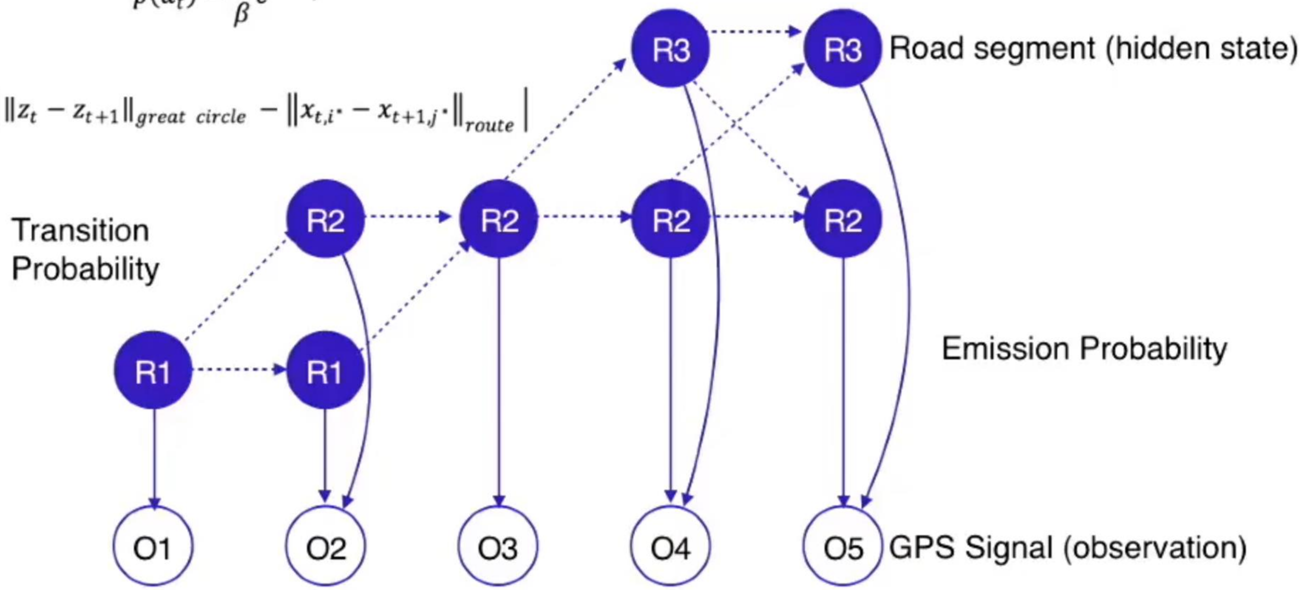
Transition Probability
Inverse Exponential distribution of distance difference of two consecutive points

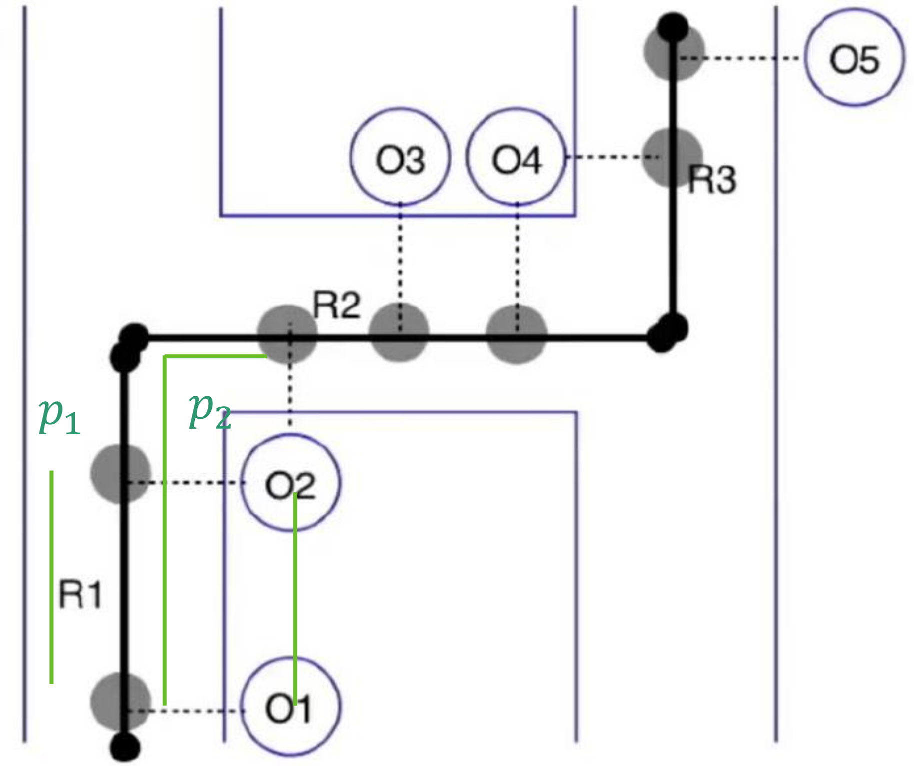
$$\beta = \frac{1}{\ln(2)} \mathrm{median}_t \left(\left| \|z_t - z_{t+1}\|_{great\ \ circle} - \|x_{t,i^*} - x_{t+1,j^*}\|_{route} \right| \right)$$

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

Here

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



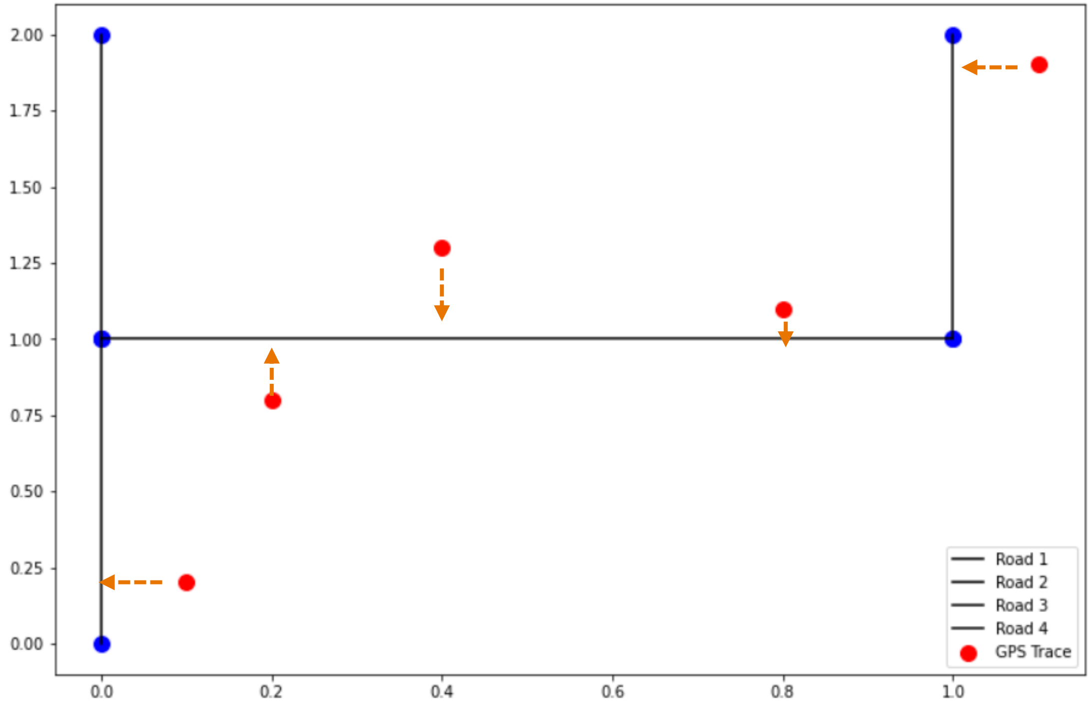


Implement

Snap

```
        # Find the road segment with the closest poi
        closest_road_idx = np.argmin([shortest_distar

        # Add the closest road segment to the path
        path.append(closest_road_idx)

    return path

# Run map matching algorithm
path = map_matching_close_distance(gps_trace, [road1,

# Print the path
print("Most likely path: ", path)
```

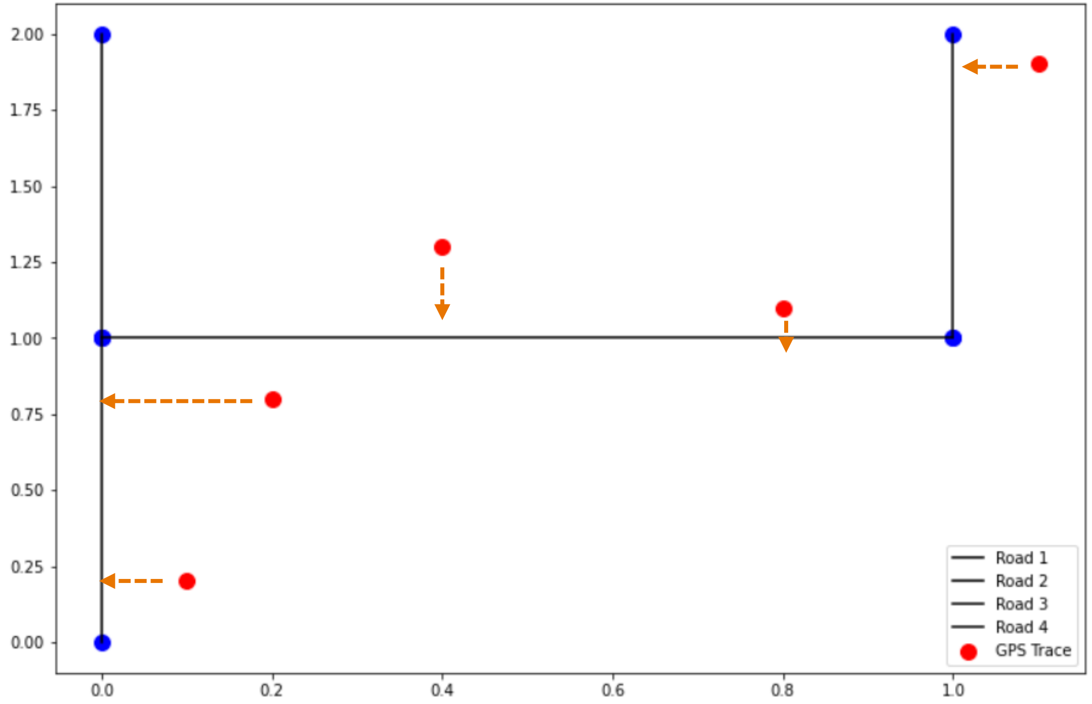
```
Most likely path:  [0, 1, 1, 1, 3]
```



HMM

```
    for j in range(4):
        temp = np.zeros(4)
        for k in range(4):
            temp[k] = viterbi_mat[i-1,k] * trans_mat[i-1,k,j
        viterbi_mat[i,j] = np.max(temp)
        backpointers[i-1,j] = np.argmax(temp)

# Find most likely sequence of hidden states
path = [np.argmax(viterbi_mat[-1])]
for i in range(len(gps_trace)-2, -1, -1):
    path.append(backpointers[i, path[-1]])
path.reverse()

# Print results
print("Most likely sequence of road segments: ", path)
```

```
Most likely sequence of road segments:  [0, 0, 1, 1, 3]
```



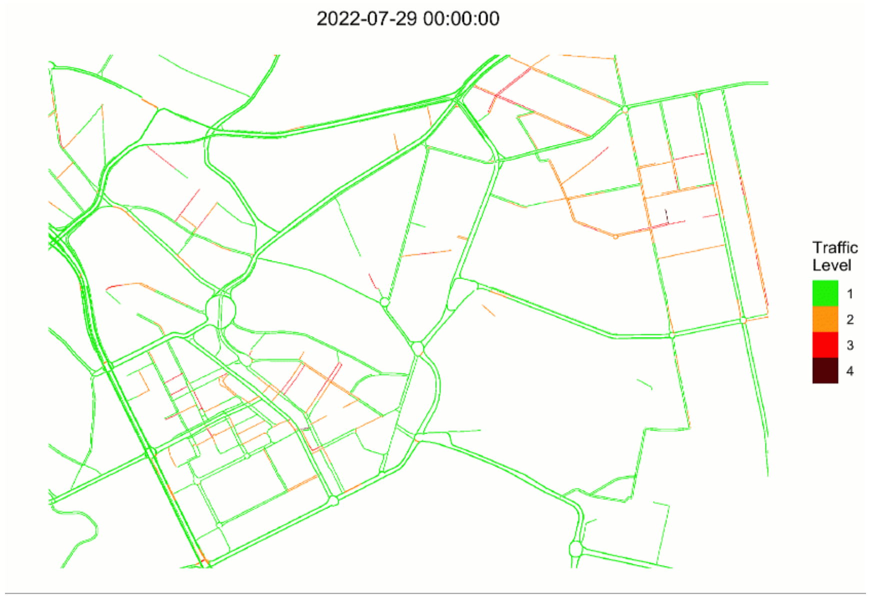
Still have breaks

Future Work

1. Fix the break problem to restore the true path

2. Find the shortest distance path and shortest driving time path from $s \rightarrow t$

3. Propose an index to measure the difference between the True path and the suggested path (shortest driving time path), figure out the reason behind it



Thank you! Questions?

Yalin Yang, yxy180050@utdallas.edu
May Yuan, myuan@utdallas.edu