PublicTransitSnapper:

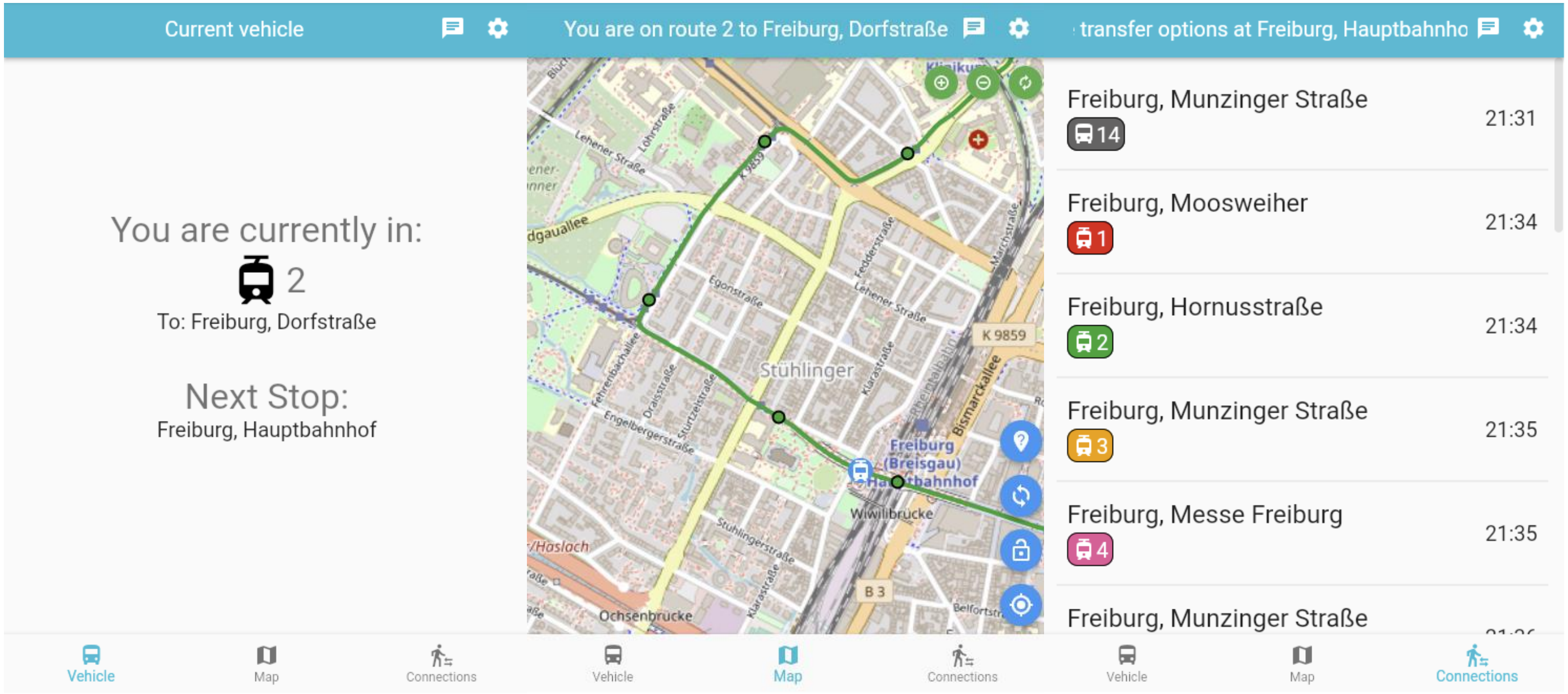
Dynamic Map-Matching To Public Transit Vehicles

Bachelor‘s Thesis by Robin Wu

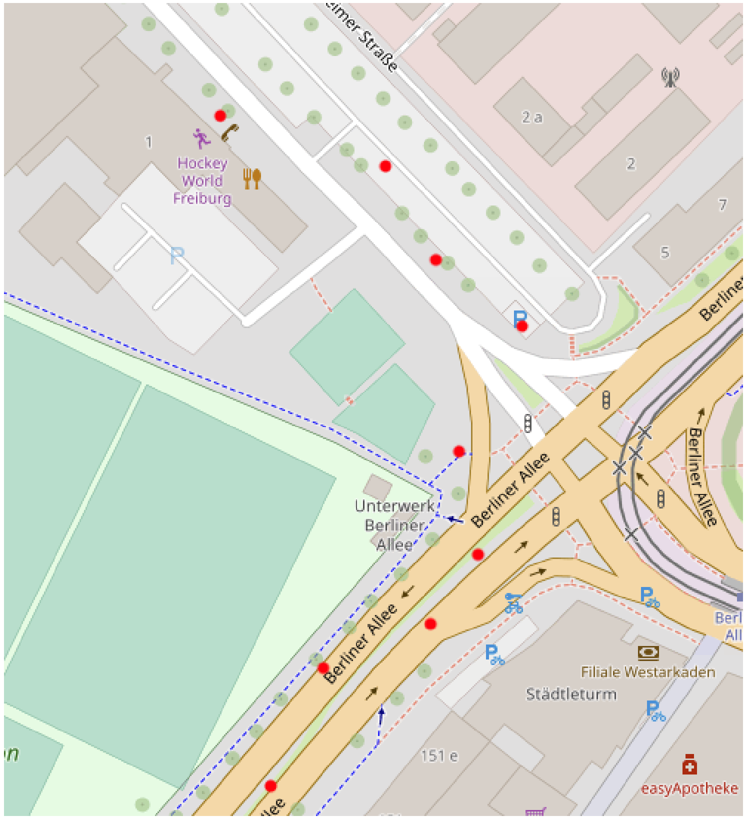
Department of Computer Science
Chair of Algorithms and Data Structures
Albert-Ludwigs-University of Freiburg

UNI FREIBURG

Introduction

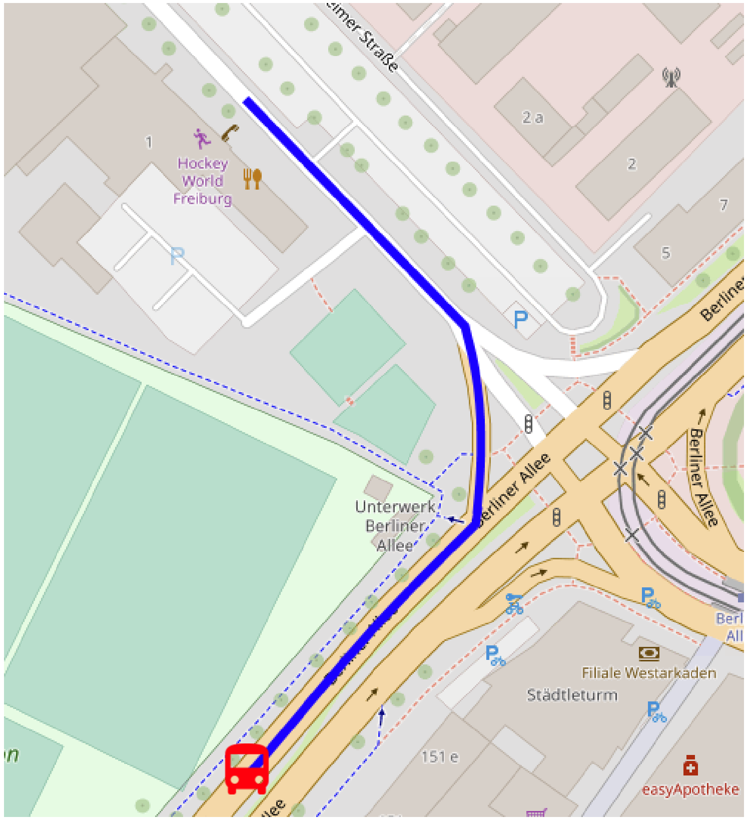


Problem Definition



n timestamped GPS Points

$g = [pt_1, \ldots, pt_n]$



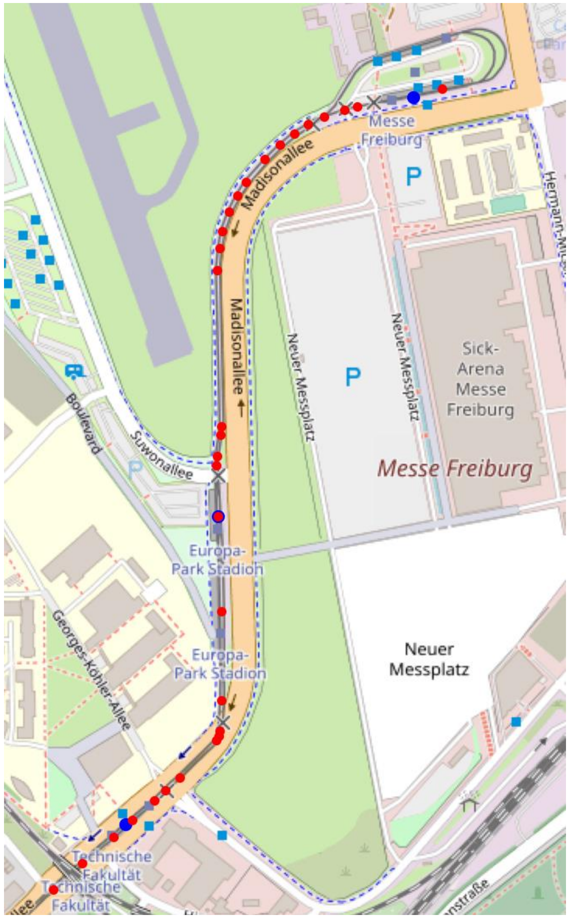
Most likely public transit vehicle

Approach

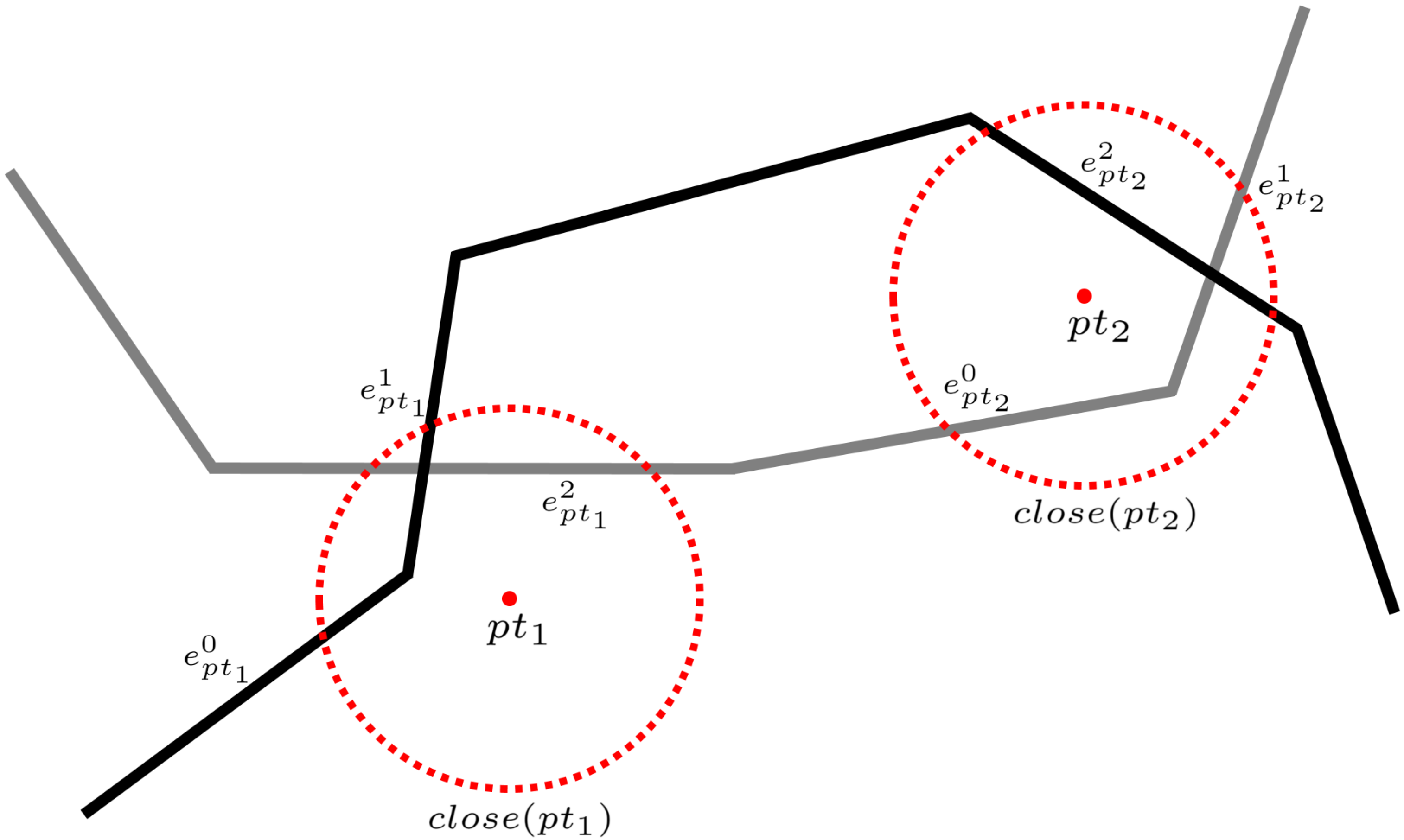
GTFS Dataset

- General Transit Feed Specification (GTFS)

- Each trip is described by:
  - Shape
  - Route
  - Service
    - Active weekdays
    - Exception dates
  - Stops
    - Location
    - Stop times



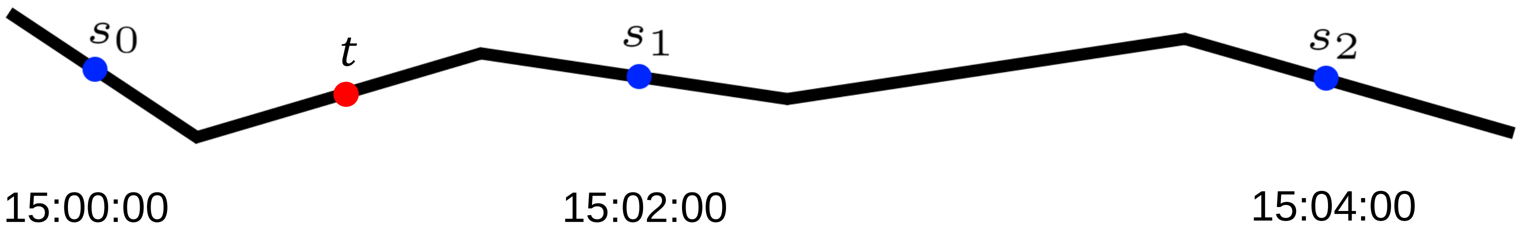
Close Edges


$e^2_{pt_2}$
$e^1_{pt_2}$
pt_2
$e^1_{pt_1}$
$e^0_{pt_2}$
$e^2_{pt_1}$
$close(pt_2)$
pt_1
$e^0_{pt_1}$
$close(pt_1)$

Active Edges

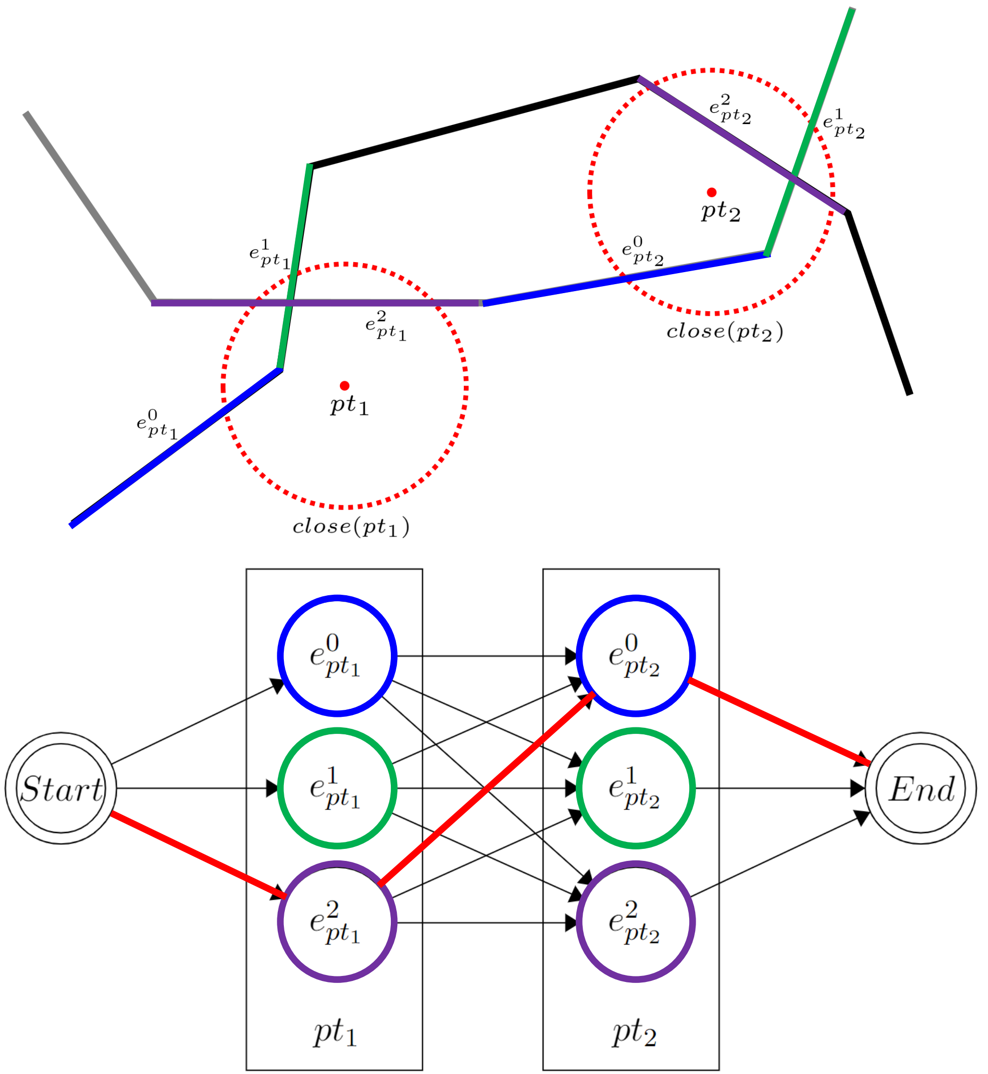
- Check service for the given date

- Time information available for stops



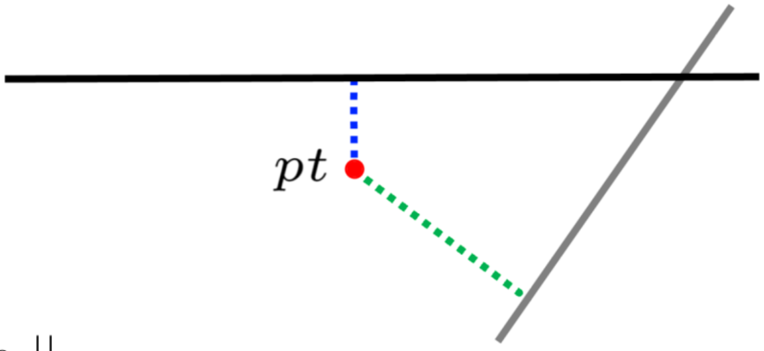
- Check $time_{start} - \varepsilon \leq t \leq time_{end} + \delta$

Hidden Markov Model (HMM)



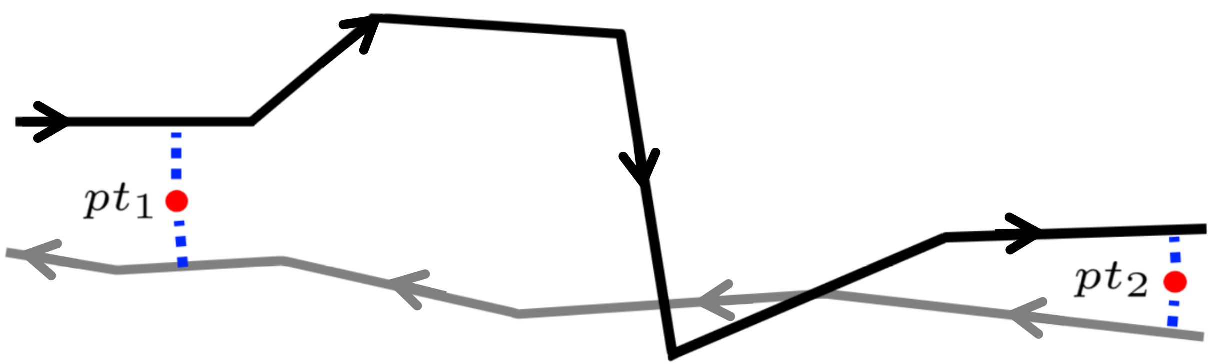
HMM Cost Functions

- Convert to log-space → summation
- $C_{\text{emission}}(e|pt) = ||e, pt||$

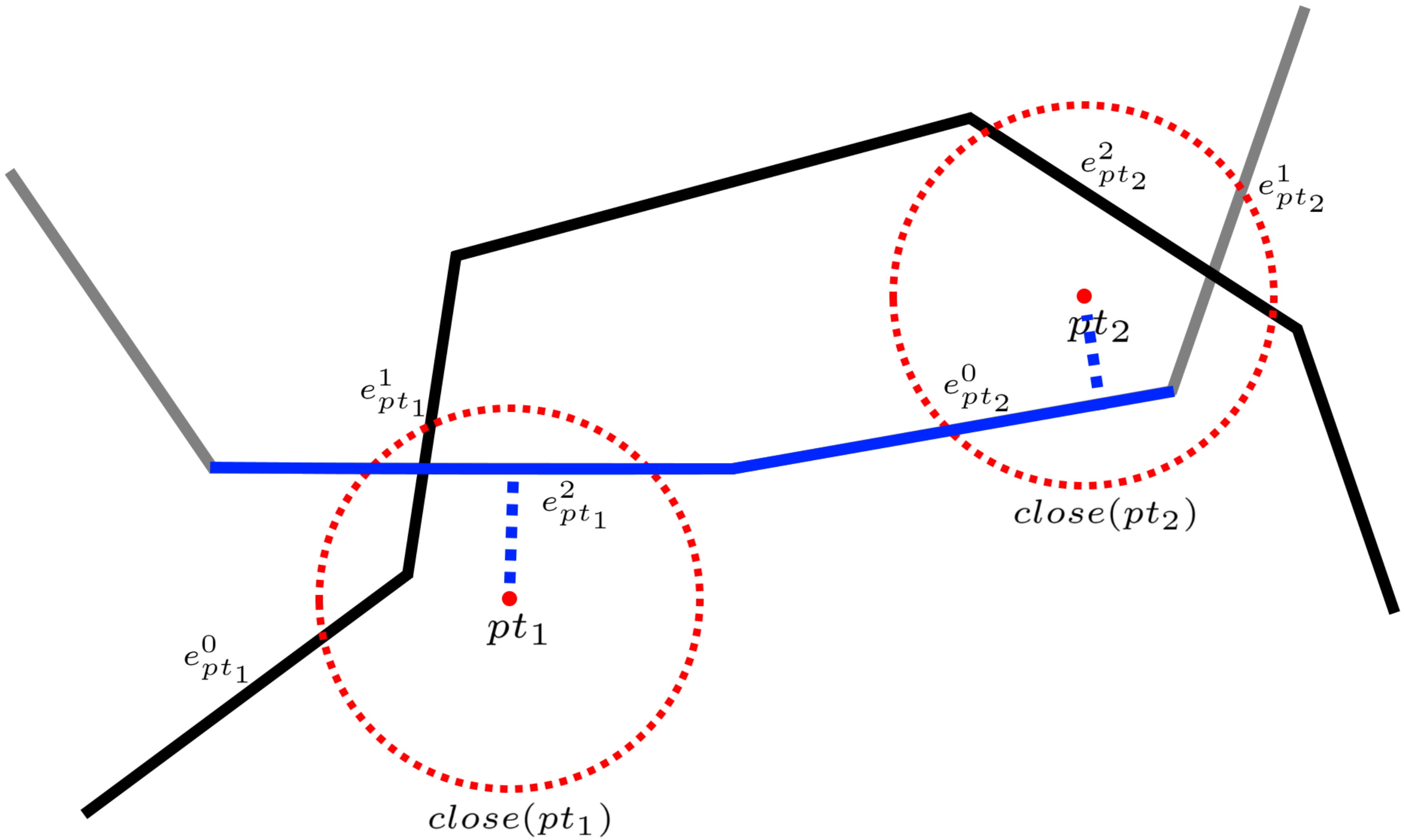


- $C_{\text{transition}}(e_1 \rightarrow e_2) = ||e_1|| + ||e_2||$

$$+ ||shortest\_path((end(e_1), start(e_2))||$$
$$+ direction\_penalty(e_1, e_2)$$

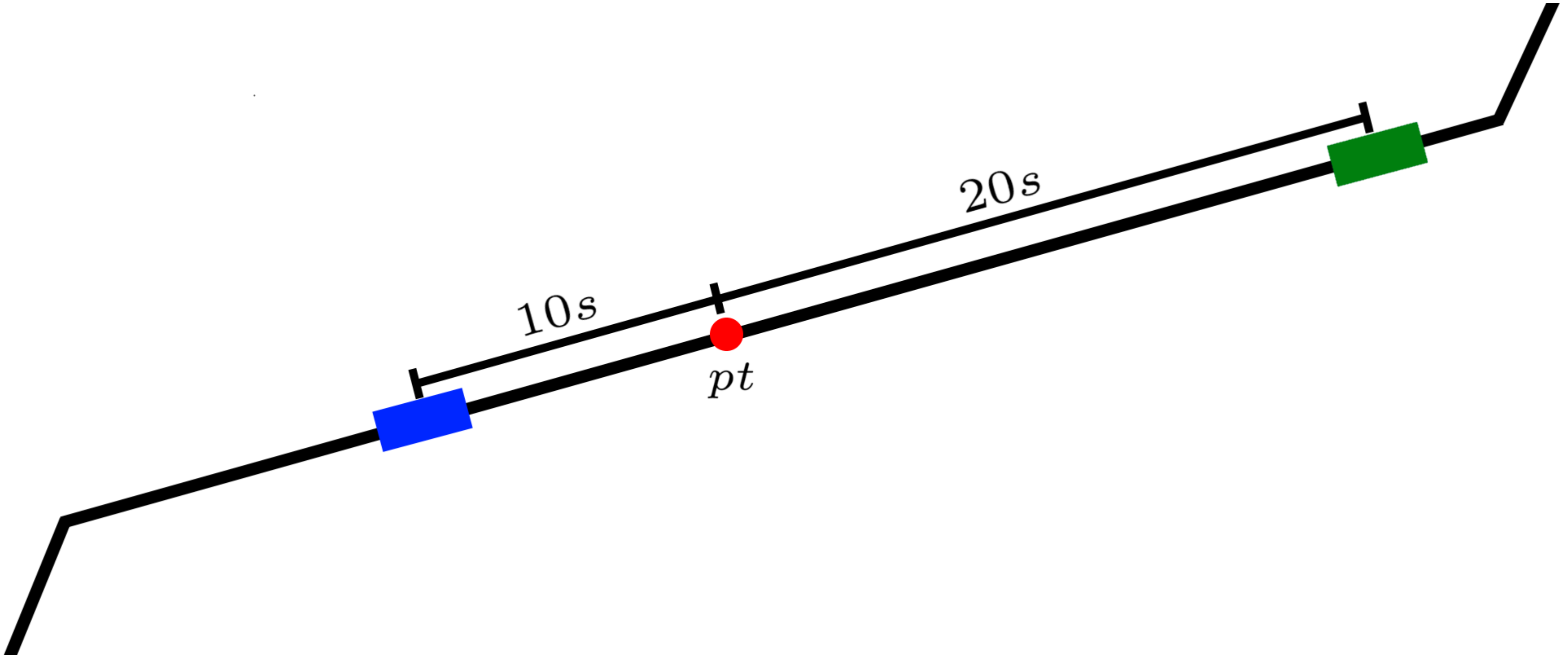


Most Likely Shape



Adding Time Data

- Multiple trips can be active at the same time



Algorithm

- Find all close edges
- Filter the close edges for active edges
- Determine the most likely shape with a HMM
- Select the most likely trip from the most likely shape

Evaluation

Evaluation Method

- Generate own test dataset with GTFS dataset
  - Generate test data for each trip with multiple tests
  - User travels on a trip for 4 stops with 10 GPS Points
  - Generate a timestamp for each GPS Point

- Add noise
  - GPS inaccuracy $\mathcal{N} \sim (0, 16)$
  - Stop times $\max(0, \mathcal{N} \sim (0, 60))$
  - Timestamps $\mathcal{N} \sim (0, 30)$

Accuracy Measure

$$\mathbb{1}(test) = \begin{cases} 1, & \text{if matched correct trip} \\ 0, & \text{otherwise} \end{cases}$$

- Number of tests depends on number of stops

$$\text{accuracy} = \frac{1}{|\text{Test Data}|} \sum_{td \in \text{Test Data}} \left(\frac{1}{|td|} \sum_{test \in td} \mathbb{1}(test) \right)$$

Evaluation Datasets

Dataset	total trips	tram	bus	funicular	train
Freiburg	19,153	9,063	10,090	—	—
SWEG	733	—	—	—	733
Zürich	33,178	—	31,971	1,206	—

Baseline Algorithms

	Average Accuracy		Average Run-Time	
Dataset	Baseline	BaselineHMM	Baseline	BaselineHMM
Freiburg	0.4%	0.8%	0.06s	0.67s
SWEG	0.8%	1.6%	0.002s	0.037s
Zürich	0.2%	0.4%	0.06s	0.70s

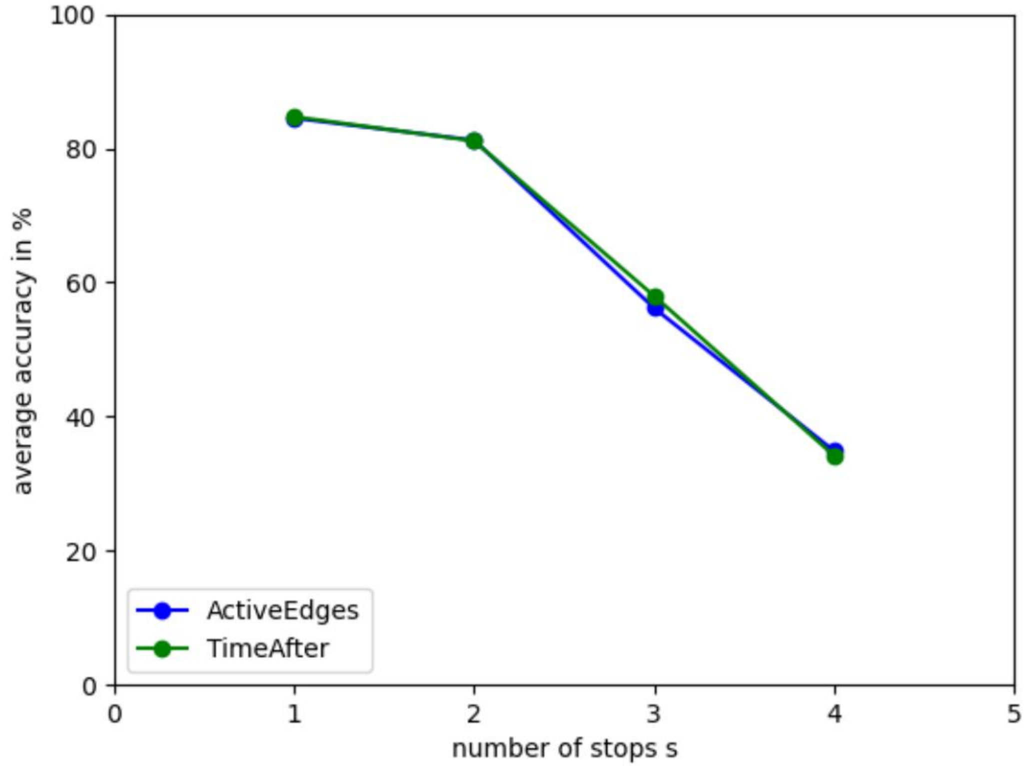
Evaluation Results

	Average Accuracy		Average Run-Time	
Dataset	ActiveEdges	TimeAfter	ActiveEdges	TimeAfter
Freiburg	91.2%	91.3%	0.246s	0.241s
SWEG	32.5%	32.3%	0.018s	0.018s
Zürich	92.3%	94.8%	0.233s	0.238s

with allowed time "slack" $\varepsilon = 1, \delta = 5$ min

SWEG Dataset

- Trains have higher distances between stops



Reducing the number of stops in the test data for the SWEG dataset

Live Demo

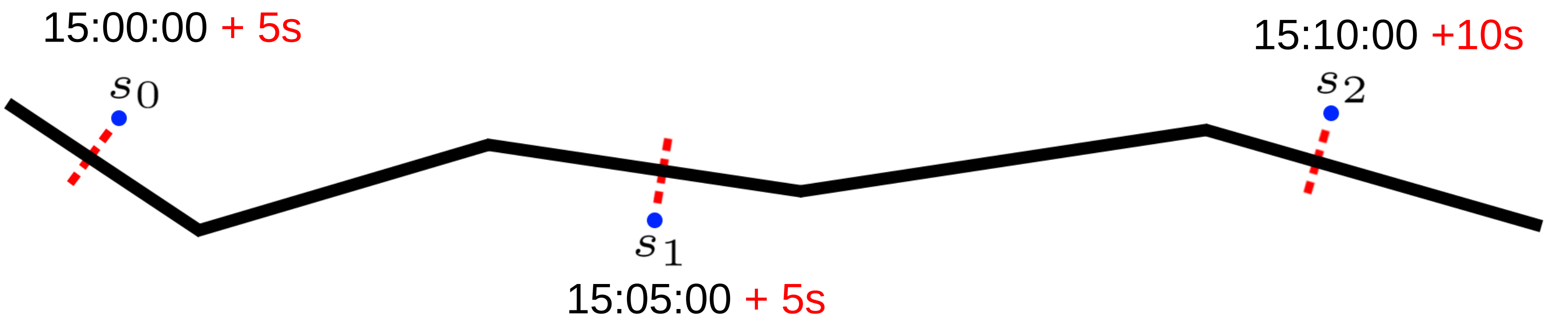
Real-Time Data

- GTFS Realtime

- Update for a trip:

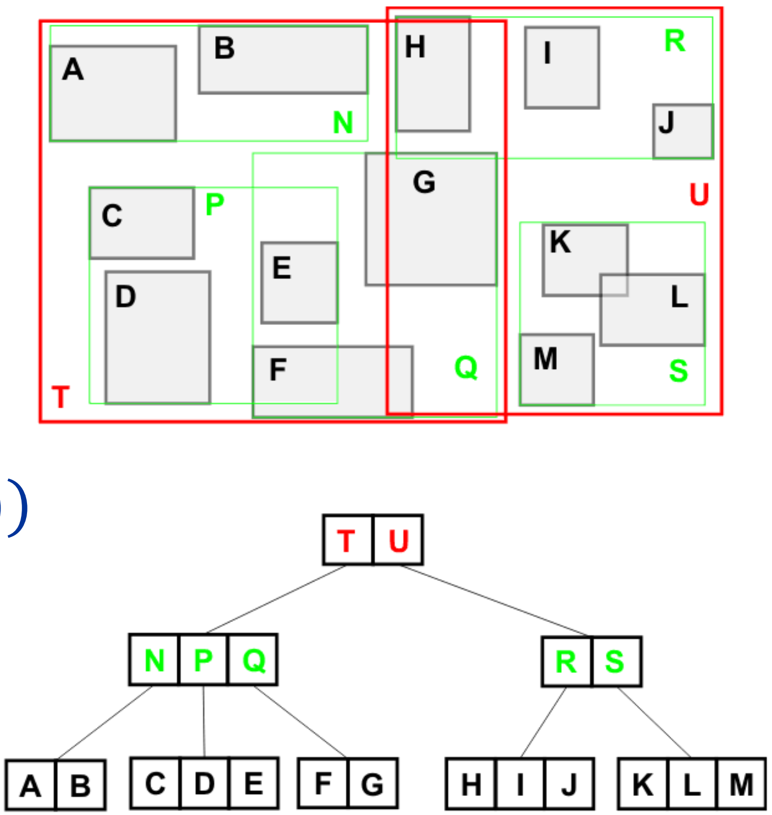
  [“stop_sequence” : 1, “departure” : {“delay” :   5}
   “stop_sequence” : 3, “departure” : {“delay” : 10}]

- Apply the delay to the stop times



R-Tree

- A graph does not support efficient spatial queries
- Insert edges into an R-Tree
- Lookup on average $O(\log(n))$



https://cglab.ca/~cdillaba/comp5409_project/R_Trees.html

“Overtime” Trips

- Trips can be in “overtime”
  - a trip runs on Monday from 23:30:00 till 25:30:00 (Tuesday 01:30:00)

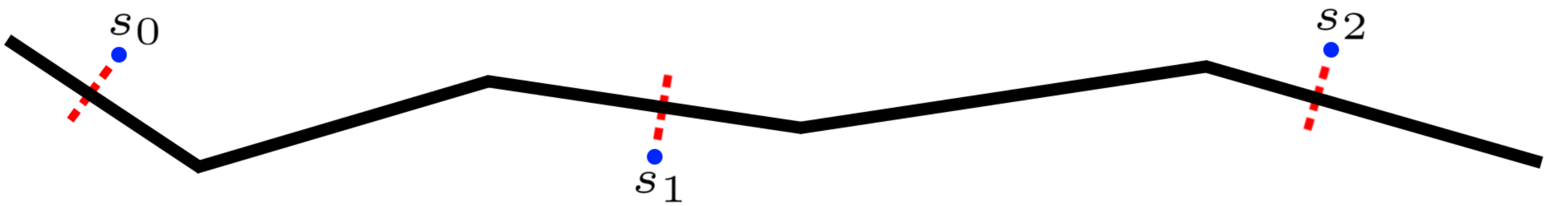
    → active weekdays only contains Monday

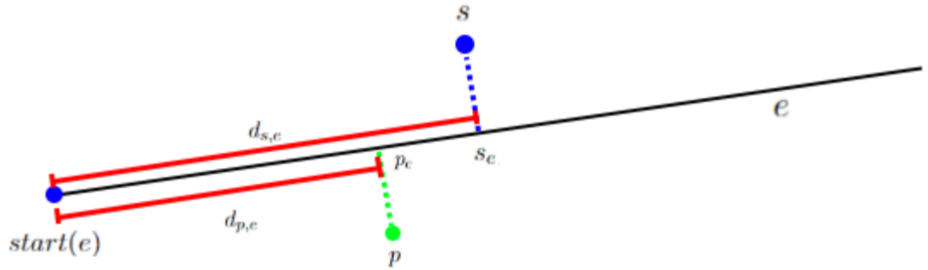
- Checking active weekdays from the service can fail
  - User on Tuesday 01:00:00

- Generate {(0,23,False),(1,0,True),(1,1,True)}

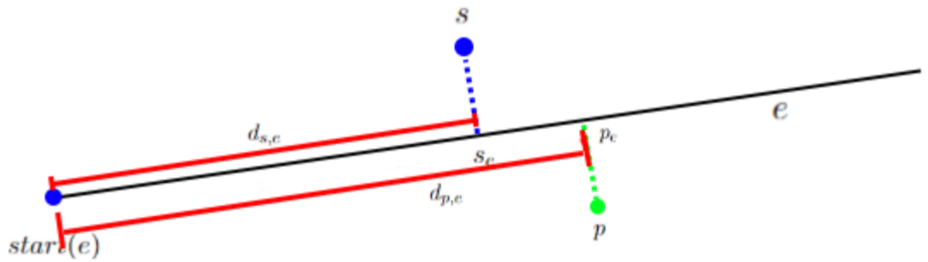
- For the user check (1,0,False) and (1,0,True)

Determine the Next Stop





(a) $d_{s,e} > d_{p,e}$



(b) $d_{s,e} < d_{p,e}$

Storage and Memory Consumption

Dataset	GTFS size	precompute size	precompute time	docker memory usage
Freiburg	28.3MB	36.1MB	1.65s	384MiB
SWEG	4.6MB	5.6MB	0.27s	148MiB
Zürich	45.5MB	58.0MB	2.47s	627MiB

“Broken” Trips

```
Trip: 573.T0.11-3-I-j22-1.6.R

Trip active on: ['monday', 'tuesday', 'wednesday', 'thursday', 'friday']

Freiburg, Munzinger Straße       arrival time: 06:22:00  departure time: 06:22:00
Freiburg, VAG-Zentrum            arrival time: 06:23:00  departure time: 06:23:00
Freiburg, Am Lindenwäldle        arrival time: 06:25:00  departure time: 06:25:00
Freiburg, Bugginger Straße       arrival time: 06:26:00  departure time: 06:26:00
Freiburg, Rohrgraben             arrival time: 06:27:00  departure time: 06:27:00
Freiburg, Bissierstraße          arrival time: 06:29:00  departure time: 06:29:00
Freiburg, Runzmattenweg          arrival time: 06:31:00  departure time: 06:31:00
Freiburg, Rathaus im Stühlinger  arrival time: 06:32:00  departure time: 06:32:00
Freiburg, Eschholzstraße         arrival time: 06:34:00  departure time: 06:34:00
Freiburg, Hauptbahnhof           arrival time: 06:35:00  departure time: 06:35:00
                                      .
                                      .
                                      .

Trip: 586.T0.11-3-I-j22-1.3.R

Trip active on: ['monday', 'tuesday', 'wednesday', 'thursday', 'friday']

Freiburg, Am Lindenwäldle        arrival time: 06:25:00  departure time: 06:25:00
Freiburg, Bugginger Straße       arrival time: 06:26:00  departure time: 06:26:00
Freiburg, Rohrgraben             arrival time: 06:27:00  departure time: 06:27:00
Freiburg, Bissierstraße          arrival time: 06:29:00  departure time: 06:29:00
Freiburg, Runzmattenweg          arrival time: 06:31:00  departure time: 06:31:00
Freiburg, Rathaus im Stühlinger  arrival time: 06:32:00  departure time: 06:32:00
```

Two partly indistinguishable trips in the Freiburg dataset