AALBORG UNIVERSITY

Learning to Route

Bin Yang

In collaboration with Razvan-Gabriel Cirstea, Jian Dai, Chenjuan Guo, Jilin Hu, Christian S. Jensen, Tung Kieu, Simon A. Pedersen and Sean B. Yang

Background and Motivation

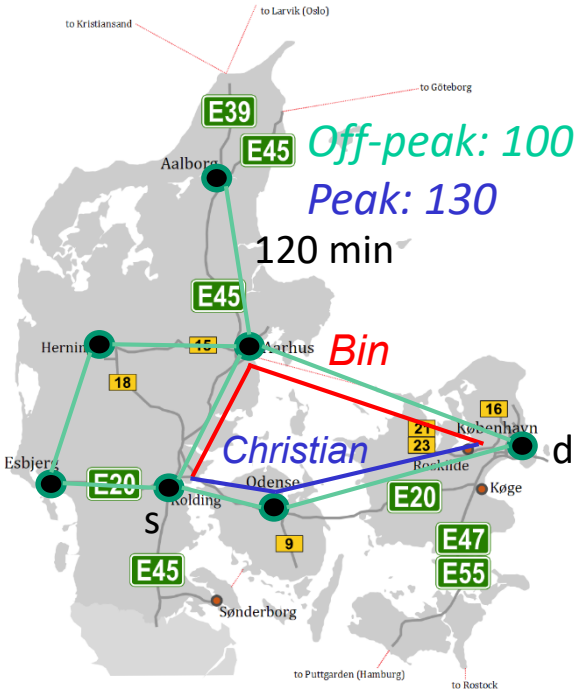
- Vehicular transportation is an important aspect of the daily lives of many people and is also essential to many businesses as well as society as a whole.

- Routing is a core functionality in vehicular transportation.
  - Given an (*s*, *d*) pair, identify a "best" path from *s* to *d*.

- As part of the continued society-wide digitization, more and more trajectory data is becoming available.

- ***Learning to Route*** studies how to best utilize trajectory data to enhance routing quality.

Learning to Route



Off-peak: 100

Peak: 130

120 min

(80, 10%)
(90, 20%)
(100, 50%)
(120, 20%)

L2R

Accurate Travel Costs

Routing Preferences

Data-Intensive Routing

How to best utilize trajectory data to enhance routing quality?

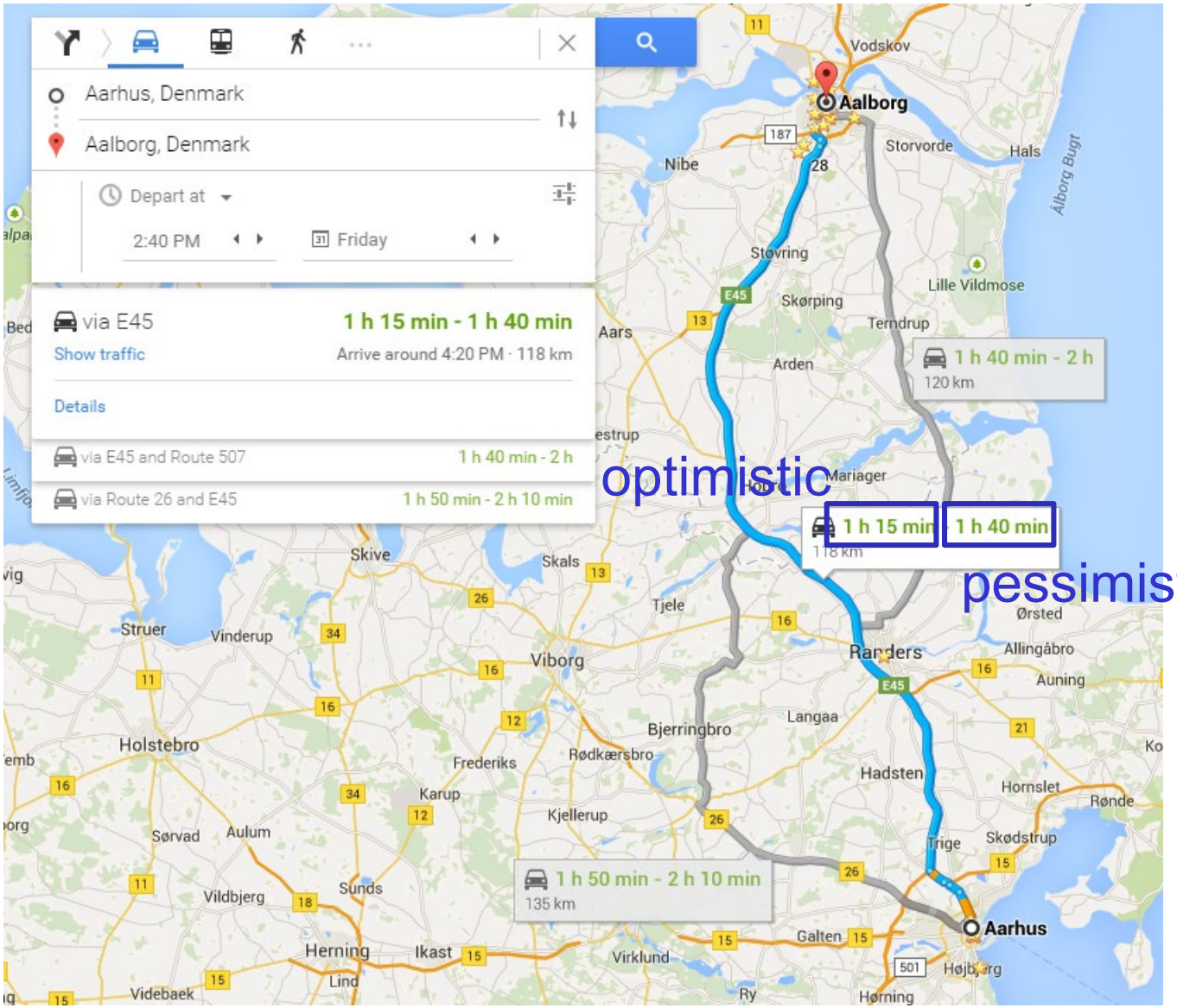
Outline

- Motivation
- Learn accurate travel costs
- Learn routing preferences
- Data-intensive routing

Learn Accurate Travel Costs

- Employ spatio-temporal data to derive accurate travel costs, e.g., travel time and fuel consumption.
- A core challenge is to capture ***travel cost uncertainty***.
  - Google Maps offers three types of travel times: optimistic, pessimistic and best-guess.

Google Maps


Aarhus, Denmark
Aalborg, Denmark
Depart at
2:40 PM
Friday
via E45
1 h 15 min - 1 h 40 min
Show traffic
Arrive around 4:20 PM · 118 km
Details
via E45 and Route 507
1 h 40 min - 2 h
via Route 26 and E45
1 h 50 min - 2 h 10 min
optimistic
pessimistic
1 h 15 min
1 h 40 min
118 km
1 h 40 min - 2 h
120 km
1 h 50 min - 2 h 10 min
135 km
Aalborg
Aarhus
Vodskov
Nibe
Storvorde
Hals
Støvring
Skørping
Lille Vildmose
Terndrup
Aars
Arden
Mariager
Skive
Skals
Tjele
Ørsted
Struer
Vinderup
Viborg
Randers
Allingåbro
Auning
Holstebro
Bjerringbro
Langaa
Frederiks
Rødkærsbro
Hadsten
Hornslet
Rønde
Karup
Kjellerup
Sørvad
Aulum
Trige
Skødstrup
Vildbjerg
Sunds
Herning
Ikast
Virklund
Galten
Videbaek
Lind
Ry
Hørning

Travel Time Distributions

- Our goal is to push the resolution of traffic uncertainty modeling further by providing *travel time distributions*.
  - Considering two paths P_1 and P_2 from home to airport.

Travel Time	[40, 50)	[50, 60)	[60, 70]
P_1	0.3	0.6	0.1
P_2	0.6	0.2	0.2

  - Which path should a self-driving taxi take in order to delivery a passenger to airport within 60 mins?
    - Probability($P_1 \leq 60$) = 0.9
    - Probability($P_2 \leq 60$) = 0.8

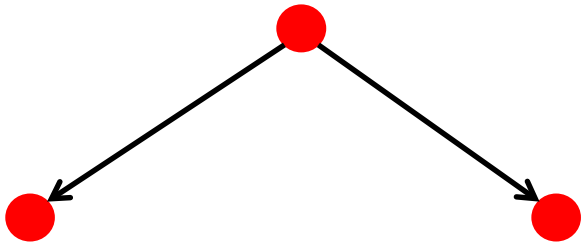
⇨ P_1

  - Google's approach:
    - Both paths have the same optimistic (40) and pessimistic (70) travel time.
  - Traditional, deterministic approach: using expected travel time:
    - P_1 has 53 and P_2 has 51, so P_2 is chosen, which is a bad decision.

Uncertain Graph Models

- Weights are distributions but not deterministic values anymore.

Uncertain Graph Models

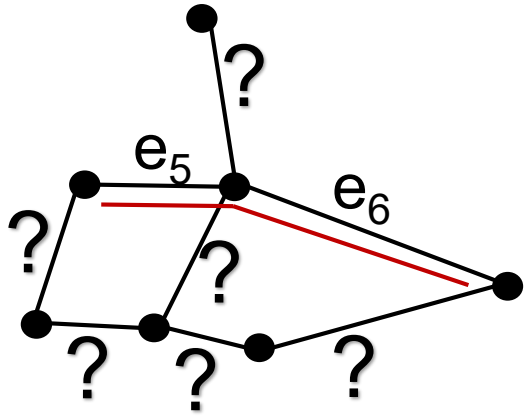


Edge-Centric Model: Weights are assigned to edges.

Path-Centric Model: Weights are assigned to paths.

Edge-Centric Model

- Edge weights, classic way, graph theory.
  - Split trajectories into small pieces that fit edges.
  - Use the small pieces to assign edges with travel time distributions.



- Example: P=<e_5, e_6>
  - Trajectory 1: 10 s, 20 s
  - Trajectory 2: 15 s, 25 s

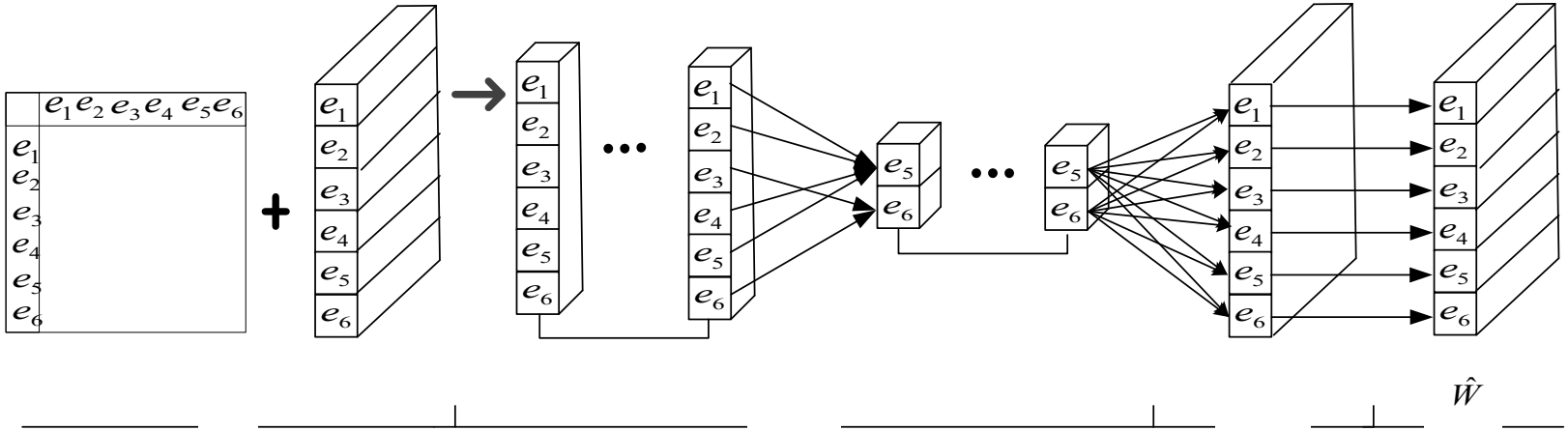
	Cost	Prob
e_5	10	0.5
	15	0.5

	Cost	Prob
e_6	20	0.5
	25	0.5

- Challenge: data sparseness.
  - Even big trajectory data is skewed, which cannot cover all edges.
- Solution: stochastic weight completion.
  - Propagate distributions from edges with trajectories to edges without trajectories.

Weight Completion Architecture

- Autoencoder + Graph Convolutional Neural Network.



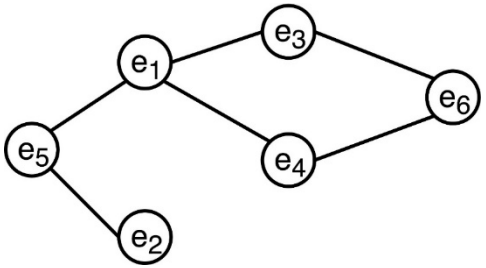
	[5, 10)	[10, 15)	[15, 20)
e_1	?	?	?
e_2	?	?	?
e_3	?	?	?
e_4	?	?	?
e_5	0.3	0.5	0.2
e_6	0.2	0.3	0.5

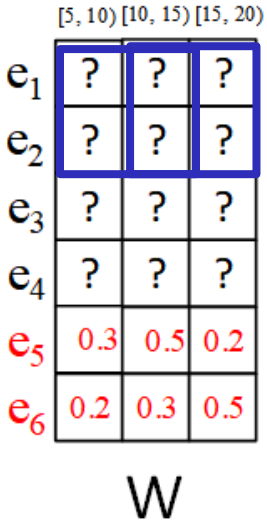
[5, 10)	[10, 15)	[15, 20)
0.4	0.5	0.1
0.1	0.7	0.2
0.4	0.2	0.2
0.2	0.8	0
0.3	0.5	0.2
0.2	0.3	0.5

J. Hu, C. Guo, B. Yang, and C. S. Jensen. Stochastic Weight Completion for Road Networks using Graph Convolutional Networks. ICDE 2019, 1274-1285.

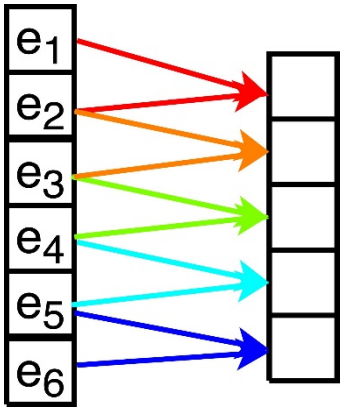
Classic Convolution vs. Graph Convolution

- Classic convolution filter 2×2.





W



- Classic convolution
  - None of the adjacent rows in W is spatially adjacent.
    - Thus, spatial correlations are not captured.
- Graph convolution
  - Considers its neighbors' features by using adjacency matrix A.
  - $W^{(l)}=\delta (A W^{(l-1)} w)$
  - E.g.: when convoluting the features of e_1, graph convolution considers the features of e_3, e_4, and e_5.

Empirical Studies

- Setup:
  - Highway tollgates network (24 edges) and a road network (172 edges).
  - Equi-width histograms with 8 buckets, 96 intervals per day.
- MKLR: Mean KL-Divergence Ratio
  - KL-divergence measures the distance between two distributions.
  - How much we can reduce the KL-divergence compared to a naïve baseline just using histograms obtained from all historical data.

rm	GP	RF	LSM	CNN	DR	GCWC	A-GCWC
0.5	1.00	0.96	1.08	0.55	0.85	0.48	0.48
0.6	1.00	0.97	1.07	0.59	0.68	0.50	0.49
0.7	1.00	0.98	1.26	0.58	0.55	0.50	0.49
0.8	1.00	0.99	1.35	0.66	0.61	0.49	0.49

Path-Centric Model

- Example: P=<e_5, e_6>
  - Trajectory 1: 10 s, 20 s
  - Trajectory 2: 15 s, 25 s

e_5

Cost	Prob
10	0.5
15	0.5

e_6

Cost	Prob
20	0.5
25	0.5

- The distribution of a path is computed by summing the distributions of edges while assuming they are indepdennt.

<e_5, e_6>

Cost	Prob
10, 20	0.25
15, 20	0.25
10, 25	0.25
15, 25	0.25

Joint distribution

P

Cost	Prob
30	0.25
35	0.50
40	0.25

Independence

<e_5, e_6>

Cost	Prob
10, 20	0.5
15, 25	0.5

Path weight

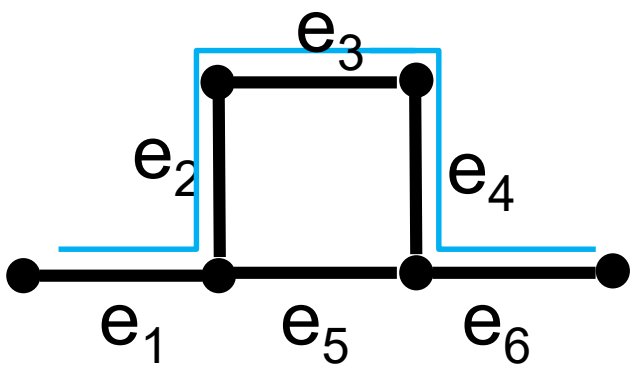
P

Cost	Prob
30	0.5
40	0.5

Dependence

Path-Centric Model

- Better capture cost dependency
  - Path weights: weights are assigned to paths, which maintain the dependency among the edges in the paths.
- Challenge: more than one combination to compute the cost distribution of a path.



Edge weights:

$e_1, e_2, e_3, e_4, e_5, e_6$

Path weights:

$\langle e_1, e_2\rangle, \langle e_4, e_6\rangle, \langle e_3, e_4, e_6\rangle$

More accurate ***More efficient***

The blue path $\langle e_1, e_2, e_3, e_4, e_6\rangle$:

Edge-centric: $e_1 \odot e_2 \odot e_3 \odot e_4 \odot e_6$

Path-centric: $\langle e_1, e_2\rangle \odot \langle e_3, e_4, e_6\rangle$

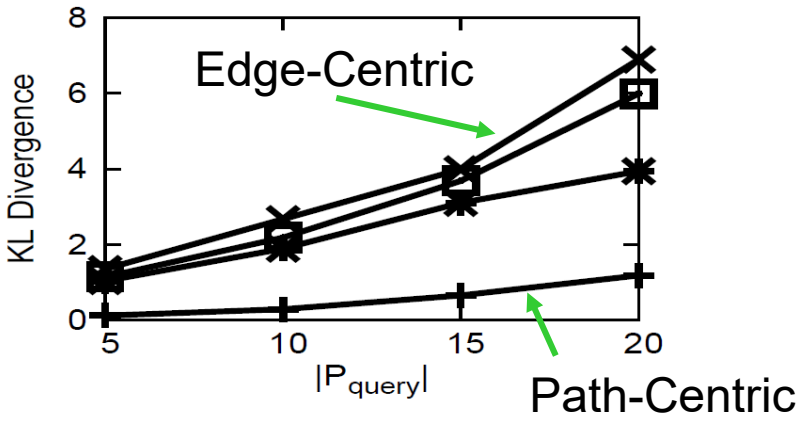
$\langle e_1, e_2\rangle \odot e_3 \odot \langle e_4, e_6\rangle$

$e_1 \odot e_2 \odot \langle e_3, e_4, e_6\rangle$

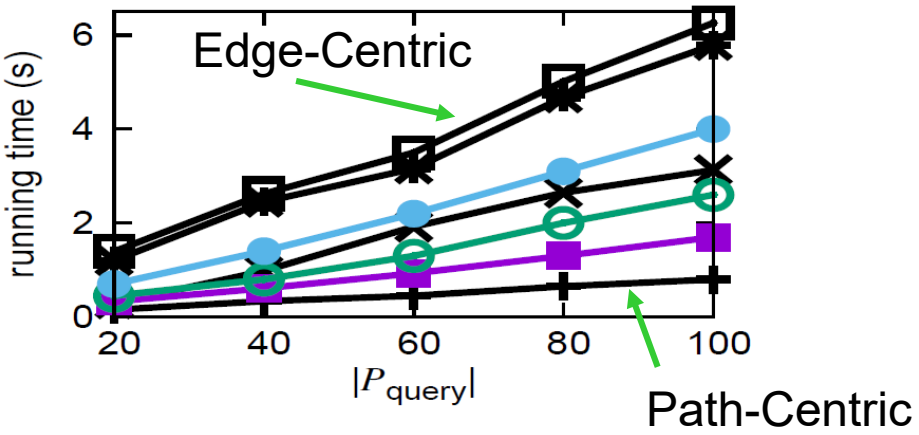
Dai, Yang, Guo, Jensen, Hu. Path Cost Distribution Estimation Using Trajectory Data. PVLDB 10(3): 85-96 (2016).
Yang, Dai, Guo, Jensen, Hu. PACE: A PAth-CEntric Paradigm For Stochastic Path Finding. The VLDB Journal 27(2): 153-178 (2018).

Empirical Studies

- Setup
  - 180 Million GPS records from Denmark.



Accuracy



Efficiency

A forward-looking paradigm that promises higher efficiency and accuracy.

Outline

- Motivation
- Learn accurate travel costs
- Learn routing preferences
- Data-intensive routing

Routing Preferences

- Professional and local drivers often follow paths that are neither fastest nor shortest.

- It is of interest to know why drivers chose such paths.
  - Educate new drivers;
  - Provide personalized navigation;
  - Teach self-driving cars to make good routing decisions.

Green: fastest path.
Red: shortest path.
Blue: a driver's actual path.

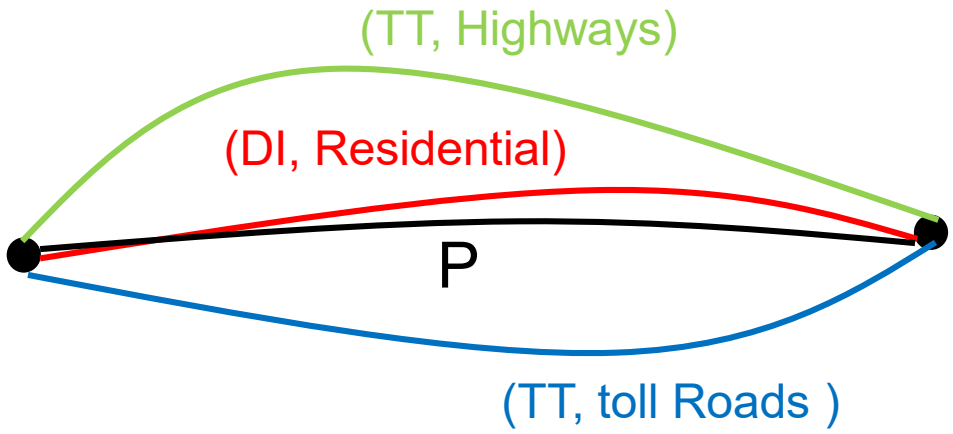
Modeling Routing Preferences

- Consider two categories of features that may affect a driver's routing decisions.
- Travel costs
  - Travel time (TT), distance (DI), fuel consumption (FC).
- Road conditions
  - Highways, residential roads, toll roads;
- Routing preference
  - 2D vector, e.g., V=(TT, Highways)

Learning Preferences

- Given a routing preference V, we are able to identify a corresponding path P_V.
  - V=(TT, Highways): a fastest path that uses highways if possible.

- If V reflects accurately a driver's actual routing preference, P_V should be the same, or very similar, to the path P used by the driver.



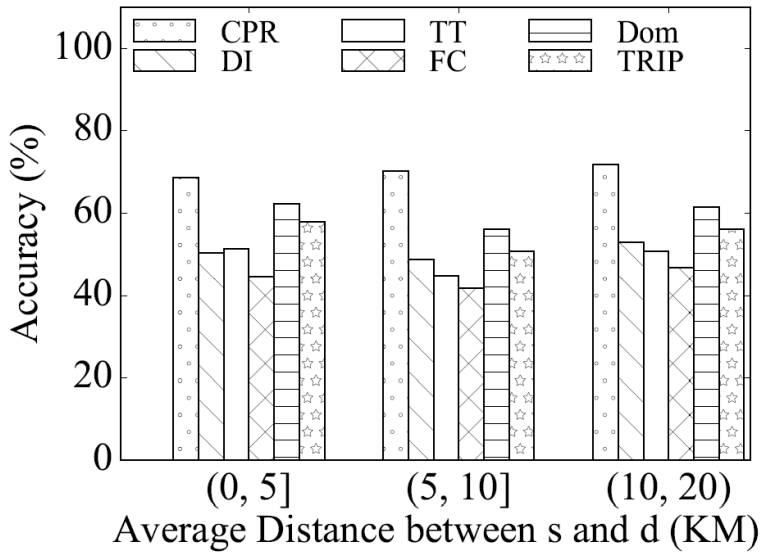
Speed up the learning

- When having *n* travel costs and *m* road conditions.
  - Naïve method: check all *n*m* possible preference vectors.
- Coordinate descent: check only *n*+*m* possible preference vectors.
  - On the first dimension, identify the best travel cost.
    - Is the fastest path, the shortest path, or the most fuel efficient path is the most similar to P?
    - For example, if the shortest path is the most similar to P, DI is chosen.
  - On the second dimension, identify the best road condition.
    - To see whether the shortest path can be further improved when considering different road conditions.
- Checking each preference vector calls for a shortest path finding.
  - Dijkstra's algorithm is slow.
  - Label-constrained contraction hierarchies.
    - Road conditions are treated as labels.

Empirical Studies

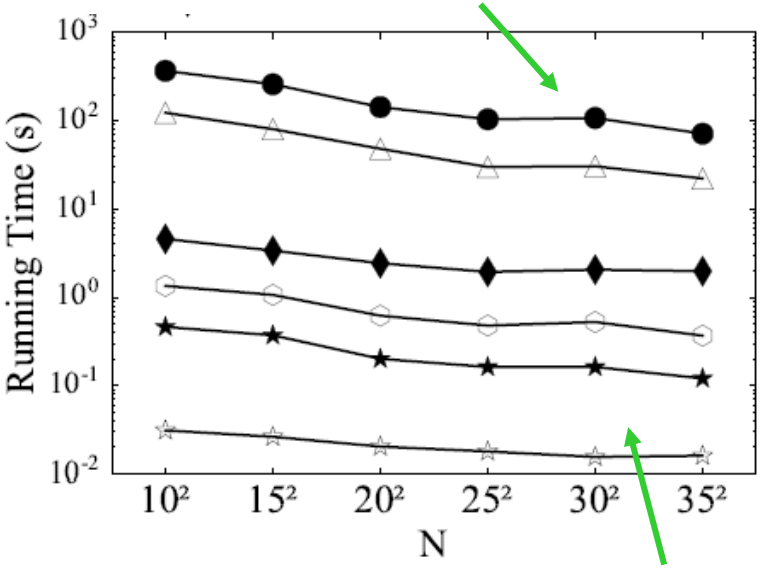
- Setup:
  - 180 Million GPS records from Denmark.
  - 75% data for learning routing preferences. 25% data for testing the accuracy of the learned routing preferences.

Accuracy



Efficiency

Naïve + Dijkstra's alg



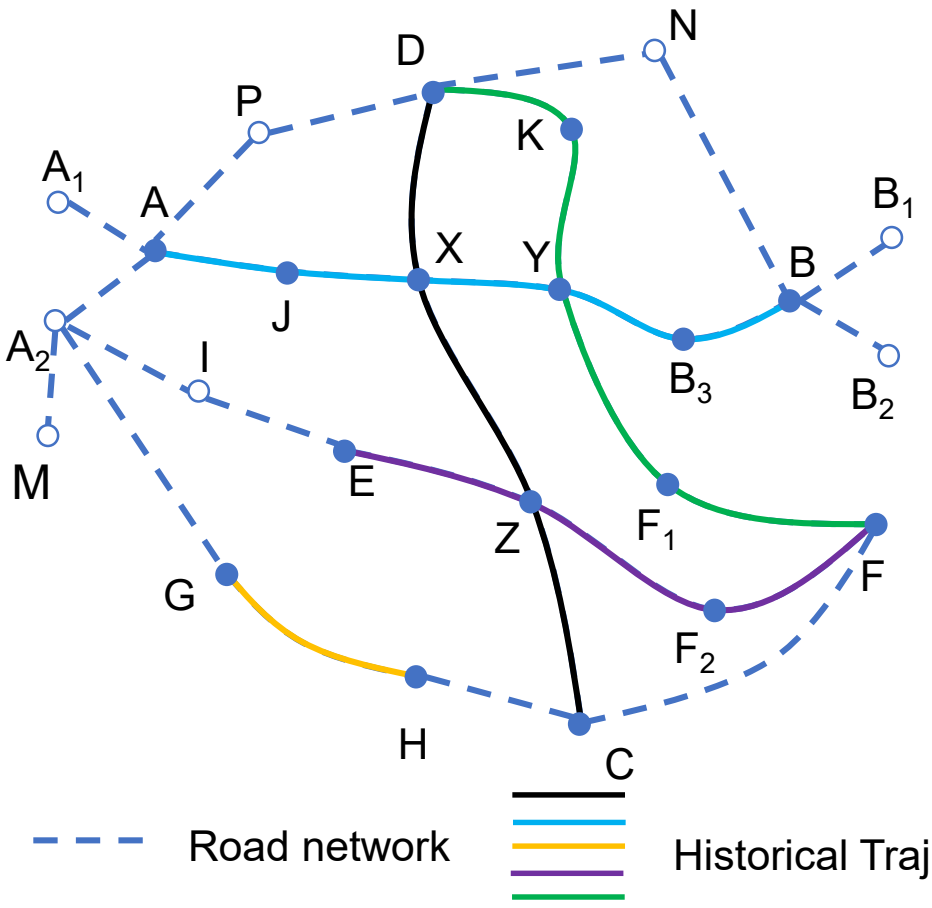
Coordinate descent + Contraction Hierarchies

Outline

- Motivation
- Learn accurate travel costs
- Learn routing preferences
- Data-intensive routing

Trajectory-base Routing

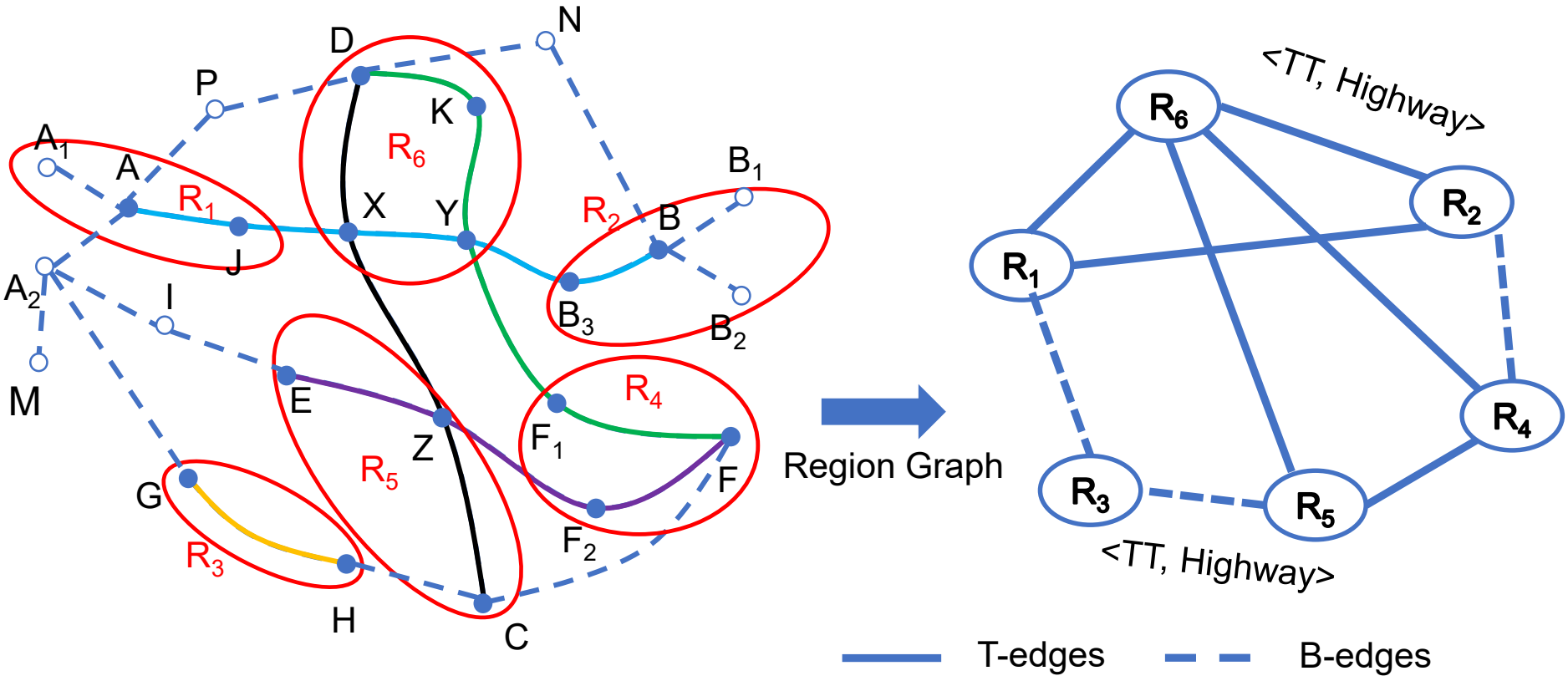
- How to best utilize trajectories from experienced, professional drivers, to recommend routes to new drivers, or self-driving cars?



- Reuse professional drivers' paths
  - Trivial case: D to C
    - Reuse the path <D, X, Z, C> in the black trajectory.
  - Data sparseness challenge!
    - G to B?
- Solution
  - Graph clustering.
  - Semi-supervised preference learning.

Overview

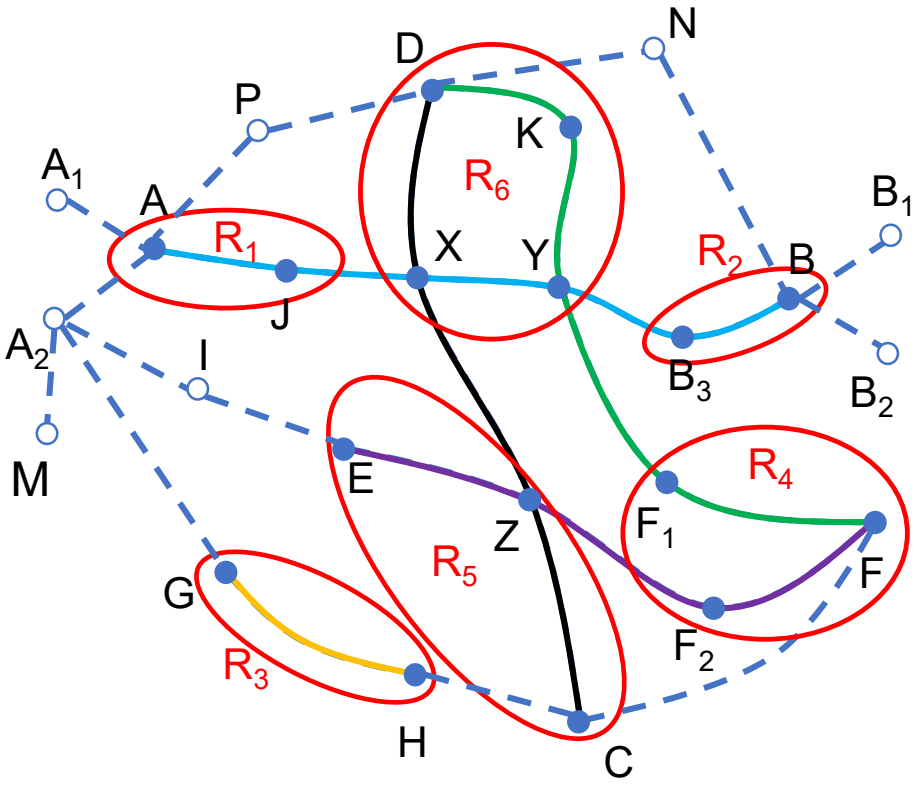
- Cluster vertices into regions and transfer a road network to a region graph.
- Learn a routing preference from T-edges.
- Transfer the preferences from T-edges to similar B-edges.
- Use the transferred preferences to infer paths for B-edges.

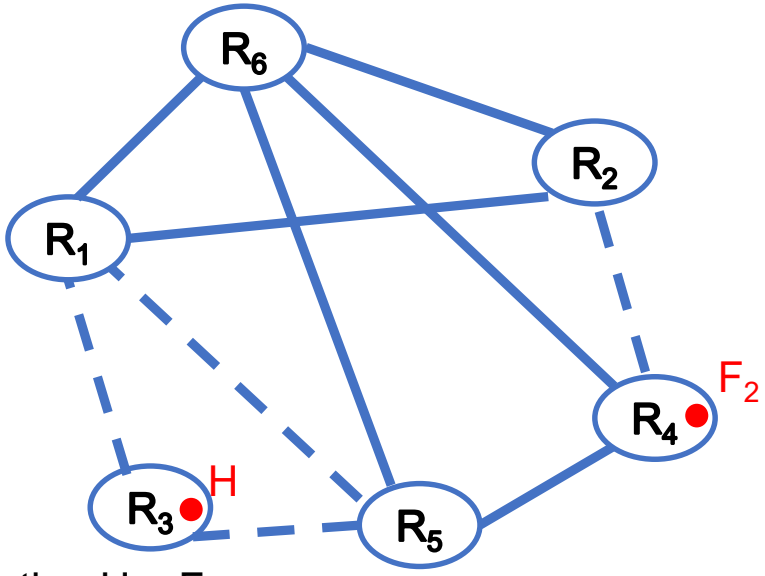


Guo, Yang, Hu, and Jensen. Learning to Route with Sparse Trajectory Sets. ICDE 2018, 1073-1084.

Routing on the Region Graph

- Given arbitrary (s, d) in the original road network,
  - Identify nearest source and destination region.
  - Route in the region graph.
  - Reconstruct paths using the original road network.



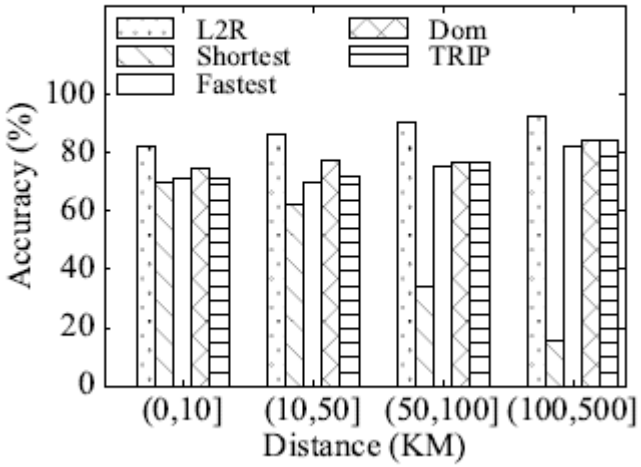


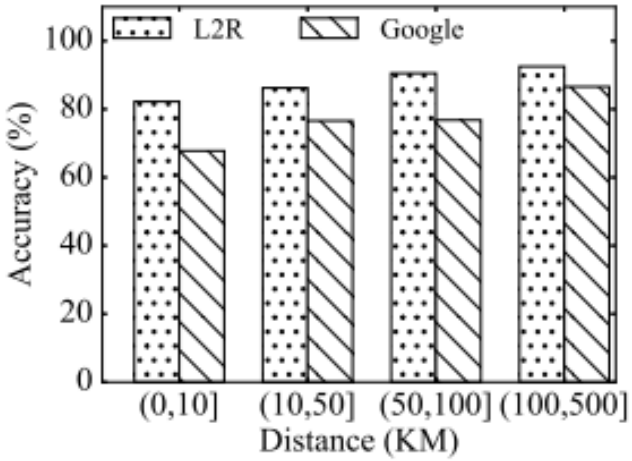
Routing H – F_2:

1. H is in R_3
2. F is in R_4
3. Region graph routing: $R_3 – R_5 – R_4$
4. $R_3 – R_5$ is recovered as H – C
5. $R_5 – R_4$ is recovered as Z – F_2
6. Return H – C – Z – F_2

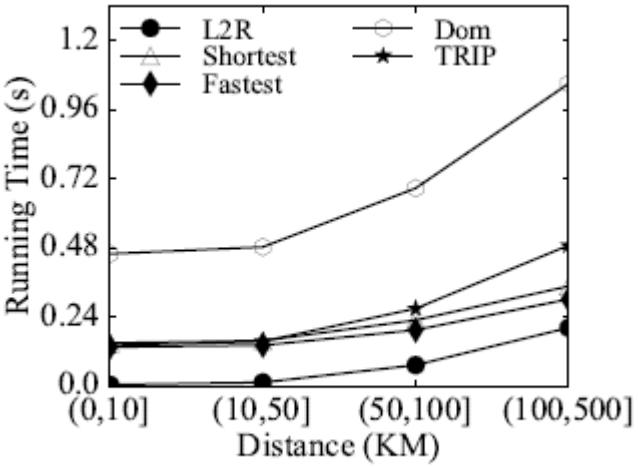
Empirical studies

- Accuracy





- Efficiency

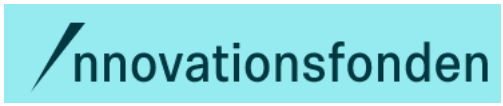


Data Driven Decision Making

- From 4V big data to 3T big knowledge that helps decision making.
- 3T big knowledge
  - Thorough (volume and variety).
    - Conquer the data sparseness challenges.
    - Cover all edges/paths, all time periods, all (s, d) pairs.
  - Timely (velocity).
    - Handle high-speed streaming data, e.g., 100 Hz accelerometer data.
    - Different applications have different requirements.
  - Trustworthy (veracity)
    - Capture traffic uncertainty at high resolution.
    - Make reliable decisions under uncertainty.

Acknowledgment

- Independent research fund Denmark.
- The Danish agency for science and higher education.
- Innovation fund Denmark.
- The Obel family foundation.



References

- Hu, Guo, Yang, Jensen: Stochastic Weight Completion for Road Networks Using Graph Convolutional Networks. ICDE 2019: 1274-1285
- Kieu, Yang, Guo, and Jensen. Outlier Detection for Time Series with Recurrent Autoencoder Ensembles. IJCAI 2019.
- Cirstea, Guo, and Yang. Graph Attention Recurrent Neural Networks for Correlated Time Series Forecasting. MileTS@KDD 2019.
- Guo, Yang, Hu, Jensen: Learning to Route with Sparse Trajectory Sets. ICDE 2018: 1073-1084
- Yang, Dai, Guo, Jensen, Hu. PACE: A PAth-CEntric Paradigm For Stochastic Path Finding. The VLDB Journal 27(2): 153-178 (2018).
- Hu, Yang, Guo, Jensen: Risk-aware path selection with time-varying, uncertain travel costs: a time series approach. VLDB J. 27(2): 179-200 (2018)
- Distinguishing Trajectories from Different Drivers using Incompletely Labeled Trajectories. CIKM 2018: 863-872
- Dai, Yang, Guo, Jensen, Hu. Path Cost Distribution Estimation Using Trajectory Data. PVLDB 10(3): 85-96 (2016).
- Yang, Guo, Ma, Jensen: Toward personalized, context-aware routing. VLDB J. 24(2): 297-318 (2015)
- Dai, Yang, Guo, Ding: Personalized route recommendation using big trajectory data. ICDE 2015: 543-554

Thank you!