CZ4079 Final Year Project

A Machine Learning-Based Approach to Time-Dependent Shortest Path Queries

Wei Yumou

School of Computer Science and Engineering
Nanyang Technological University

Agenda

Introduction: Problem

Introduction: Problem

- A **dynamic road network** $G = (V, E)$ with a time-dependent weight function $w : E, t \rightarrow \mathbb{R}$

Introduction: Problem

- A **dynamic road network** $G = (V, E)$ with a time-dependent weight function $w : E, t \rightarrow \mathbb{R}$
- A **query** $Q(u, v, t)$ that asks for a shortest path from u to v departing at time moment t

NANYANG TECHNOLOGICAL UNIVERSITY SINGAPORE

Introduction: General Approach

- Traditional **Bellman-Ford or Dijkstra's algorithm** do not work with dynamic edge weights ("the curse of traditionality")

Introduction: General Approach

- Traditional **Bellman-Ford or Dijkstra's algorithm** do not work with dynamic edge weights ("the curse of traditionality")
- The new **machine learning-based approach** draws on collective wisdom of thousands of taxi drivers

Introduction: General Approach

- Traditional **Bellman-Ford or Dijkstra's algorithm** do not work with dynamic edge weights ("the curse of traditionality")
- The new **machine learning-based approach** draws on collective wisdom of thousands of taxi drivers
- **Unsupervised learning** is employed to figure out the time-dependent edge costs

NANYANG TECHNOLOGICAL UNIVERSITY SINGAPORE

Introduction: General Approach

- Traditional **Bellman-Ford or Dijkstra's algorithm** do not work with dynamic edge weights ("the curse of traditionality")
- The new **machine learning-based approach** draws on collective wisdom of thousands of taxi drivers
- **Unsupervised learning** is employed to figure out the time-dependent edge costs
- A modified Dijkstra's algorithm calculates a shortest path on the fly

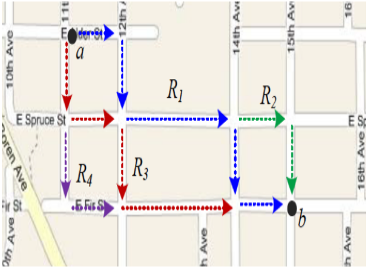
NANYANG TECHNOLOGICAL UNIVERSITY SINGAPORE

Introduction: Challenges

- Arbitrary u and v

Introduction: Challenges

- Arbitrary u and v
- Sparse sample points



Introduction: Challenges

- Arbitrary u and v
- Sparse sample points
- Limited GPS accuracy

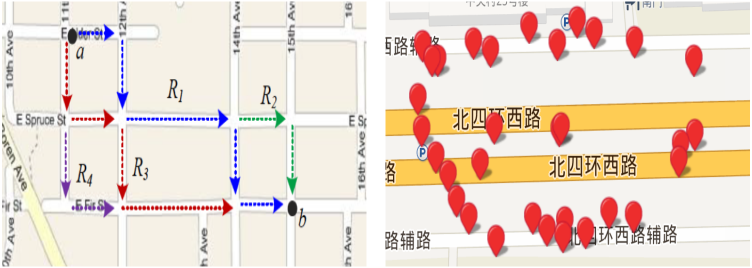


Figure 1: Examples of challenges

NANYANG TECHNOLOGICAL UNIVERSITY SINGAPORE

Agenda

1. Introduction
2. Preliminary Processing

Preliminary Processing: Data Description

- Is collected from Computational Sensing Lab at Tsinghua University
- Contains 83 million GPS records from 8,602 taxis in Beijing during May of 2009

Field	Explanation
CUID	ID for each taxi
UNIX_EPOCH	Unix timestamp
GPS_LONG	Longitude in WGS-84
GPS_LAT	Latitude in WGS-84
HEAD	Heading direction
SPEED	Instantaneous speed (m/s)
OCCUPIED	Hired (1) or not (0)

Table 1: A summary of the seven original fields

NANYANG TECHNOLOGICAL UNIVERSITY SINGAPORE

Preliminary Processing: Reverse Geocoding

Preliminary Processing: Reverse Geocoding

- GPS coordinate translation: 1.34°N, 103.68°E $\rightarrow$ SCSE, NTU

Preliminary Processing: Reverse Geocoding

- GPS coordinate translation: 1.34°N, 103.68°E $\rightarrow$ SCSE, NTU
- China GPS shift problem: WGS84 v.s. BD09

Figure 2: An example of China GPS shift problem

Preliminary Processing: Reverse Geocoding

- GPS coordinate translation: 1.34°N, 103.68°E $\rightarrow$ SCSE, NTU
- China GPS shift problem: WGS84 v.s. BD09
- Solution: WGS84 $\xrightarrow{Baidu\ \ API}$ BD09 $\xrightarrow{Baidu\ \ API}$ Street

Figure 2: An example of China GPS shift problem

Preliminary Processing: Outlier Detection

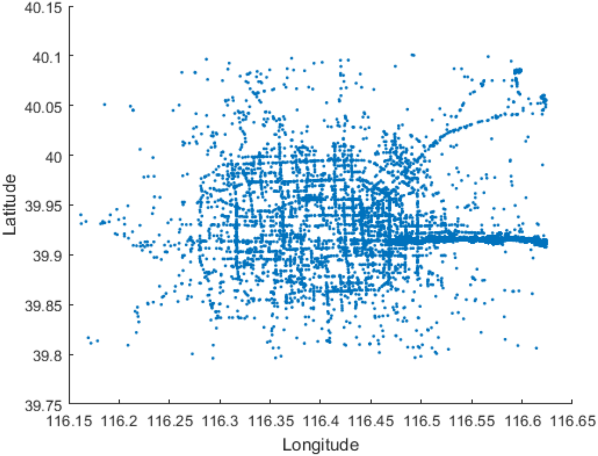


Figure 3: An example of outliers

NANYANG TECHNOLOGICAL UNIVERSITY SINGAPORE

Preliminary Processing: Outlier Detection

Theorem (*Majority Clustering Theorem*)

If a **reasonable reverse geocoder** is used to reverse-geocode a set of GPS data points which are mapped to a particular street *in reality*, then, when plotted on a 2-D plane, majority (more than 50%) of the points must be clustered together to form a rough shape that is similar to the shape of the street that they are supposed to be mapped to.

Preliminary Processing: Outlier Detection

Theorem (*Majority Clustering Theorem*)

If a **reasonable reverse geocoder** is used to reverse-geocode a set of GPS data points which are mapped to a particular street *in reality*, then, when plotted on a 2-D plane, majority (more than 50%) of the points must be clustered together to form a rough shape that is similar to the shape of the street that they are supposed to be mapped to.

Two-step procedure:
Outlier Detection = Outlier Identification + Outlier Removal

Outlier Identification: Clustering

Preliminary Processing: Outlier Detection

Outlier Identification: Clustering

- Sample point concentration $\rightarrow$ cluster concentration

Preliminary Processing: Outlier Detection

Outlier Identification: Clustering

- Sample point concentration $\rightarrow$ cluster concentration
- Top $k\%$ ($k = 50$) largest clusters as groups of correct sample points

Preliminary Processing: Outlier Detection

Outlier Identification: Clustering

- Sample point concentration $\rightarrow$ cluster concentration
- Top $k\%$ ($k = 50$) largest clusters as groups of correct sample points
- 10×10 self-organising feature maps implementation

Preliminary Processing: Outlier Detection

Outlier Identification: Clustering

- Sample point concentration $\rightarrow$ cluster concentration
- Top $k\%$ ($k = 50$) largest clusters as groups of correct sample points
- 10×10 self-organising feature maps implementation

Outlier Removal: Distance Threshold d_{max}

NANYANG TECHNOLOGICAL UNIVERSITY SINGAPORE

Preliminary Processing: Outlier Detection

Outlier Identification: Clustering

- Sample point concentration $\rightarrow$ cluster concentration
- Top $k\%$ ($k = 50$) largest clusters as groups of correct sample points
- 10×10 self-organising feature maps implementation

Outlier Removal: Distance Threshold d_{max}

- Assign sample points to legal centroids no farther than d_{max}

Preliminary Processing: Outlier Detection

Outlier Identification: Clustering

- Sample point concentration $\rightarrow$ cluster concentration
- Top $k\%$ ($k = 50$) largest clusters as groups of correct sample points
- 10×10 self-organising feature maps implementation

Outlier Removal: Distance Threshold d_{max}

- Assign sample points to legal centroids no farther than d_{max}
- Remove all "orphan" sample points

Preliminary Processing: Outlier Detection

Outlier Identification: Clustering

- Sample point concentration $\rightarrow$ cluster concentration
- Top $k\%$ ($k = 50$) largest clusters as groups of correct sample points
- 10×10 self-organising feature maps implementation

Outlier Removal: Distance Threshold d_{max}

- Assign sample points to legal centroids no farther than d_{max}
- Remove all "orphan" sample points
- Use real physical distance on the Earth

NANYANG TECHNOLOGICAL UNIVERSITY SINGAPORE

Preliminary Processing: Outlier Detection

Outlier Identification: Clustering

- Sample point concentration $\rightarrow$ cluster concentration
- Top $k\%$ ($k = 50$) largest clusters as groups of correct sample points
- 10×10 self-organising feature maps implementation

Outlier Removal: Distance Threshold d_{max}

- Assign sample points to legal centroids no farther than d_{max}
- Remove all "orphan" sample points
- Use real physical distance on the Earth
- Set $d_{max} = 30\text{m}$ or 50m

NANYANG TECHNOLOGICAL UNIVERSITY SINGAPORE

Preliminary Processing: Outlier Detection

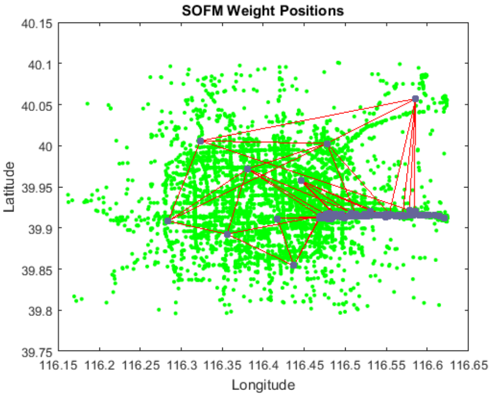


Figure 4: A plot of neuron positions after training

Preliminary Processing: Outlier Detection

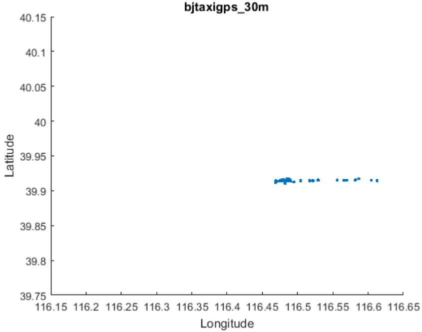


Figure 5: A plot of sample points after outlier removal