Map-matching

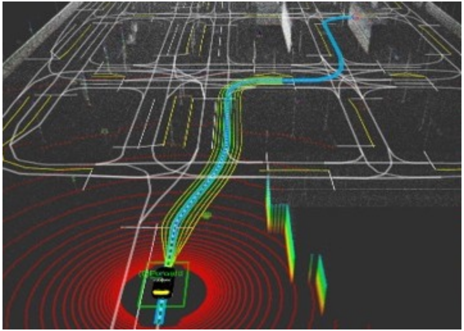
T. Akamatsu, G. Gress, K. Huneycutt, S. Omura
Academic Mentor: Dr. Kano
Industry Mentor: Dr. Yamazaki

July 15, 2022
g-RIPS

Map-matching

Given GPS trajectory data and a road map, **map-matching** is the process of determining the route on the map that corresponds to the trajectory data.

Web mapping services



Autonomous Vehicles [H]

Problem Statement

Problem Statement

Let us fix $d \geq 2$ (but almost everywhere we consider the case $d = 2$).

Definition (Trajectory)

A **trajectory** Tr is a sequence of points $\mathbf{p} = (p_1, p_2, \dots, p_n)$ where $p_i \in \mathbb{R}^d$ for $1 \leq i \leq n$ equipped with

- a sequence $t(\mathbf{p}) = (t_1, \dots, t_n)$ such that $t_i \in \mathbb{R}^+$ for $1 \leq i \leq n$ and $t_1 < t_2 < \cdots < t_n$, called the **timestamp** of **p**,
- a sequence $\mathrm{spd}(\mathbf{p}) = (\mathrm{spd}_1, \dots, \mathrm{spd}_n)$ such that $\mathrm{spd}_i \in \mathbb{R}^+$ for $1 \leq i \leq n$, called the **speed** of **p** (optional),
- a sequence $u(\mathbf{p}) = (u_1, \dots, u_n)$ such that $u_i \in \mathbb{R}^d$ and $\|u\| = 1$ for $1 \leq i \leq n$, called the **direction** of **p** (optional).

Problem Statement

Definition (Road Network)

A **road network** (also known as a map) is a directed graph $G = (V, E)$ consists of the set V (resp. E) of vertices (resp. edges) with an embedding $\phi : |G| \to \mathbb{R}^d$ of the geometric realization $|G|$ of G. We will identify G and the image $\phi(|G|)$ by ϕ as long as there is no confusion.

Problem Statement

Problem Statement

Definition (Route)

A **route** r on a road network $G = (V, E)$ is a sequence of connected edges $(e_1, e_2, \ldots, e_n) \subset E$, i.e. the head of e_i coincides with the tail of e_{i+1} for each $i = 1, 2, \ldots, n-1$. Let R denote the set of all routes.

Problem Statement

Problem Statement

Problem Statement

Definition (Map-Matching)

Given a road network $G = (V, E)$ and a trajectory Tr, the map-matching, $\mathcal{MR}_G(Tr)$, is the route that is the argument of the minimum of some function $L : R \to \mathbb{R}^+$, called the **loss function.**

Approaches to Map-Matching

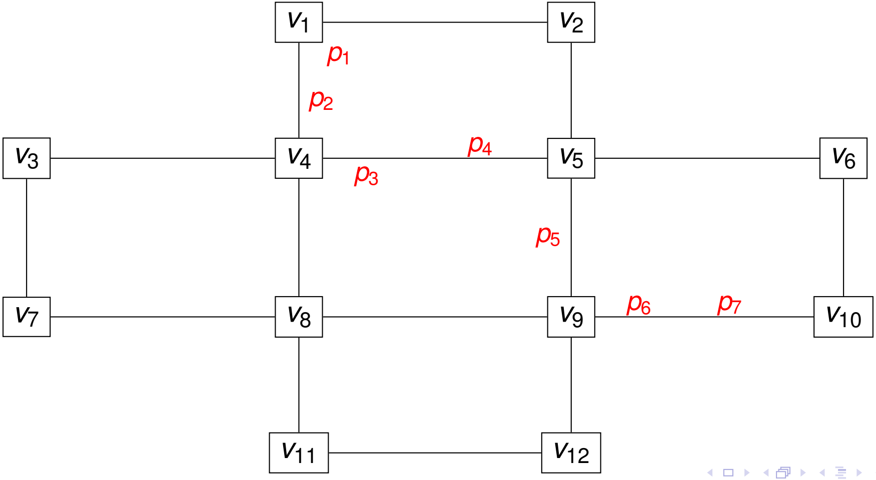
Geometric

- Point-to-point method
- Point-to-curve method

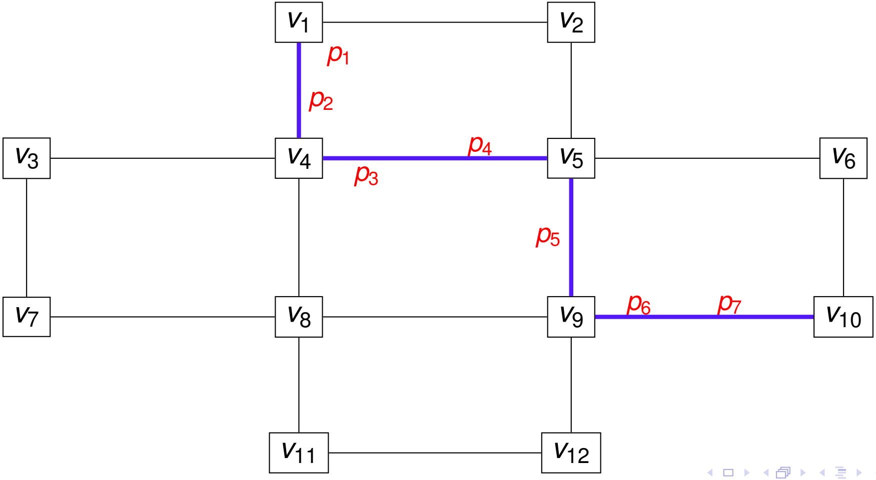
Data-Driven

- Hidden Markov model

Point-to-Curve Method

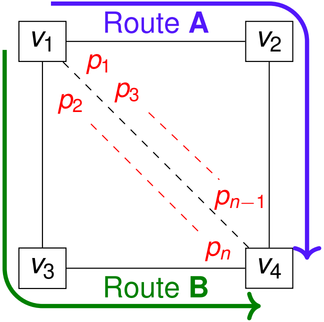


Point-to-Curve Method



Problem & Our strategy

- A square model.
  - $V = \{v_1, v_2, v_3, v_4\}$,
    $E = \{v_1v_2, v_2v_4, v_1v_3, v_3v_4\}$.
  - $\mathbf{p} = \{p_1, \ldots, p_n\}$: trajectory points
    which are located near the diagonal
    w/ coordinates and timestamps.
  - Route **A** $= \{v_1v_2, v_2v_4\}$.
  - Route **B** $= \{v_1v_3, v_3v_4\}$.



Strategy: Construction of the "**trajectory-to-route**"-type method.

"Wasserstein" method

Definition ((L^1-)Wasserstein distance ("W_1 *distance*"))

Let (X, d) be a complete and separable metric space.
For $\mu, \nu \in \mathscr{P}(X) := \{$ all (Borel) probability measures on (X, d) w/ finite support $\}$, define

$$W_1(\mu, \nu) := \min_{\pi \in \Pi(\mu,\nu)} \sum_{x \in X} \sum_{y \in X} d(x, y)\pi(x, y),$$

where $\pi \in \Pi(\mu, \nu) \; :\Leftrightarrow \; {}^{\forall}x, y \in X, \; \sum_{y \in X} \pi(x, y) = \mu(x), \; \sum_{x \in X} \pi(x, y) = \nu(y)$.

- W_1 distance is a distance function on $\mathscr{P}(X)$, i.e. quantifies the differences between two probability measures.
- W_1 distance can be calculated by linear programming (under our setting conditions).
- W_1 distance is also called "**Earth-Mover's distance**" or "**Word-Mover's distance**" (in areas such as Natural Language Processing).

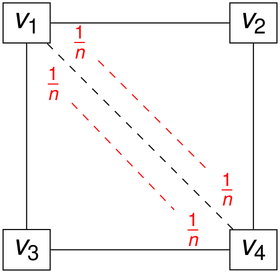


Figure: A prob. meas. $\mu_{\mathbf{p}}$ associated w/ the trajectory **p**. A weight $1/n$ is placed on each trajectory point; $\mu_{\mathbf{p}} := (1/n)\sum_{i=1}^{n} \delta_{p_i}$.

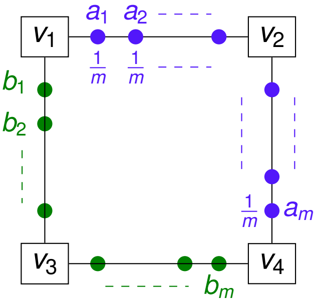


Figure: Prob. meas.s $\nu_{\mathbf{A}} = \nu_{\mathbf{A},m}$ and $\nu_{\mathbf{B}} = \nu_{\mathbf{B},m}$ associated w/ the route **A** and **B**; $\nu_{\mathbf{A}} := (1/m)\sum_{j=1}^{m} \delta_{a_j}$, $\nu_{\mathbf{B}} := (1/m)\sum_{j=1}^{m} \delta_{b_j}$.

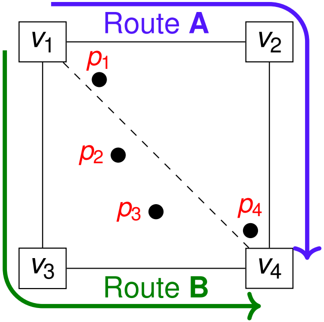
We define $\varphi(A) = \varphi(A, m) := W_1(\mu_{\mathbf{p}}, \nu_{\mathbf{A}})$, $\varphi(B) = \varphi(B, m) := W_1(\mu_{\mathbf{p}}, \nu_{\mathbf{B}})$.

$\rightsquigarrow$ If we obtain $\varphi(A) < \varphi(B)$, then we conclude that the route **A** is the true route.

Further problem: Construction of W_1 method taking speed and direction information into account.

- Modification of transport way (, i.e. the objective function of W_1 distance).
  - Loss function of W_1 method: $\sum_{x \in X} \sum_{y \in X} d(x, y)\pi(x, y)$.
  - Modified W_1 distance is likely to be difficult to handle.
- Modification of probability measures $\mu_{\mathbf{p}}$ (or $\nu_{\mathbf{A}}$ and $\nu_{\mathbf{B}}$).
  - We are trying to modify $\mu_{\mathbf{p}}$ using information from speed and direction information. (Under consideration...)

"Electrical charge" method



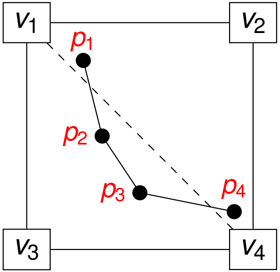


Figure: Connecting trajectory points.

- Considering not only trajectory points, but also the entire polyline.
- Comparing it with the entirety of each route.

"Electrical charge" method

1. Giving the candidate routes and polyline opposing electrical charges.
2. Choosing the route which exerts the most force on the polyline as the true route.

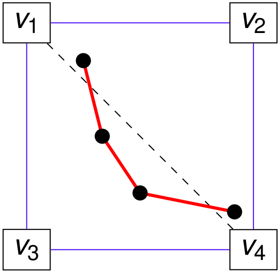


Figure: Giving electrical charges.

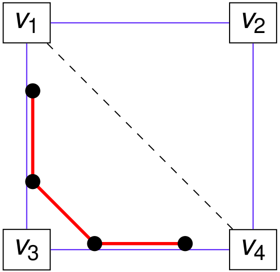


Figure: Moving to "closer" route.

"Electrical charge" method

Further problem: Taking into account information such as

- speed,
- direction,
- error.

- Varying the electric density instead of assuming uniformity.

Jupyter Demonstration