Subtrajectory Clustering : Models and Algorithms

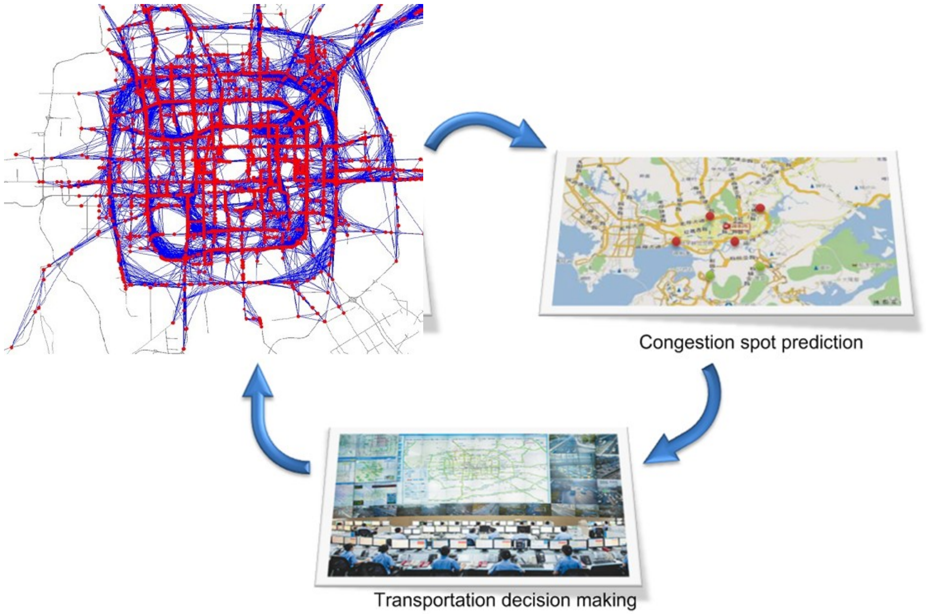
Pankaj K. Agarwal[1]   Kyle Fox[2]   Kamesh Munagala[1]

Abhinandan Nath[1]   Jiangwei Pan[3]   Erin Taylor[1]

[1] Duke University
[2] UT Dallas
[3] Facebook

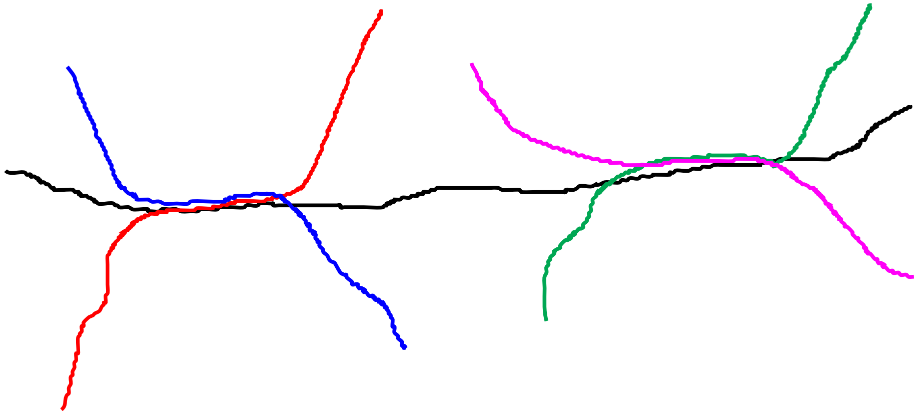
Introduction

- Huge trajectory data
  - GPS traces, sensors …
  - Improve decision making
  - Gain useful insights
- Noisy and incomplete
- Gives rise to several computational challenges

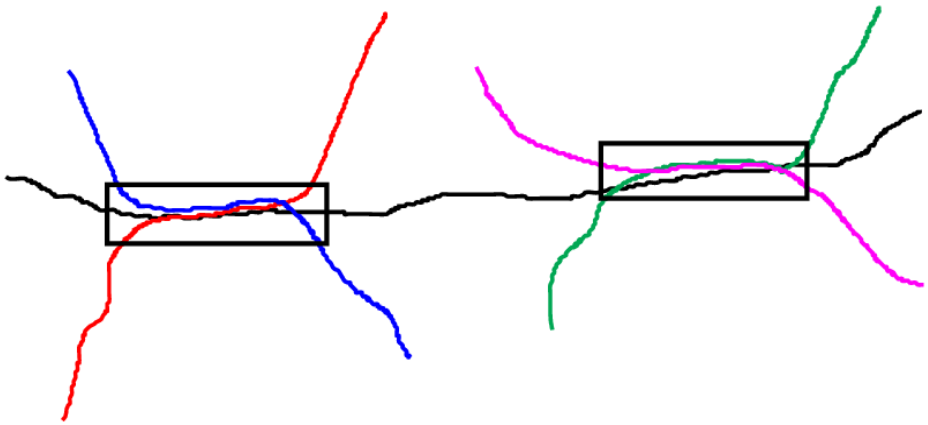


[developer.huawei.com]

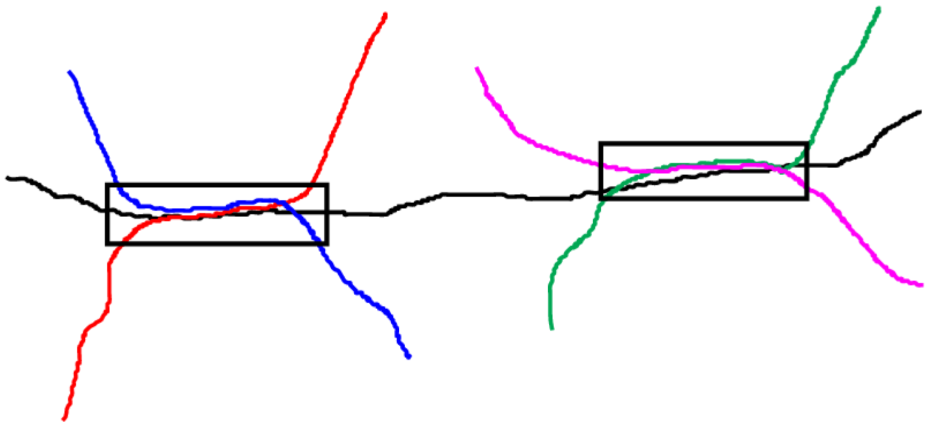
Motivation



Motivation



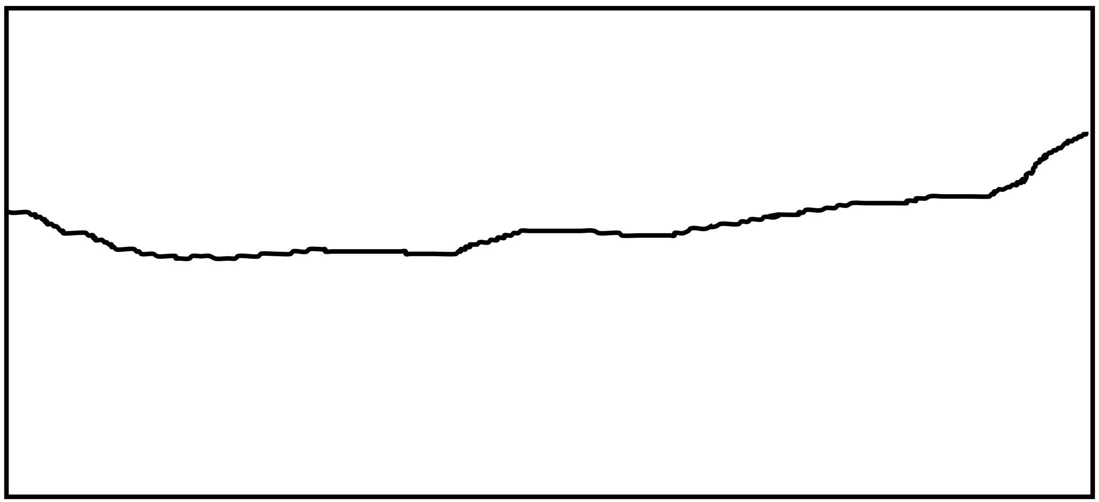
Motivation



- Subtrajectory clusters capture common portions
- Different from clustering trajectories as a whole

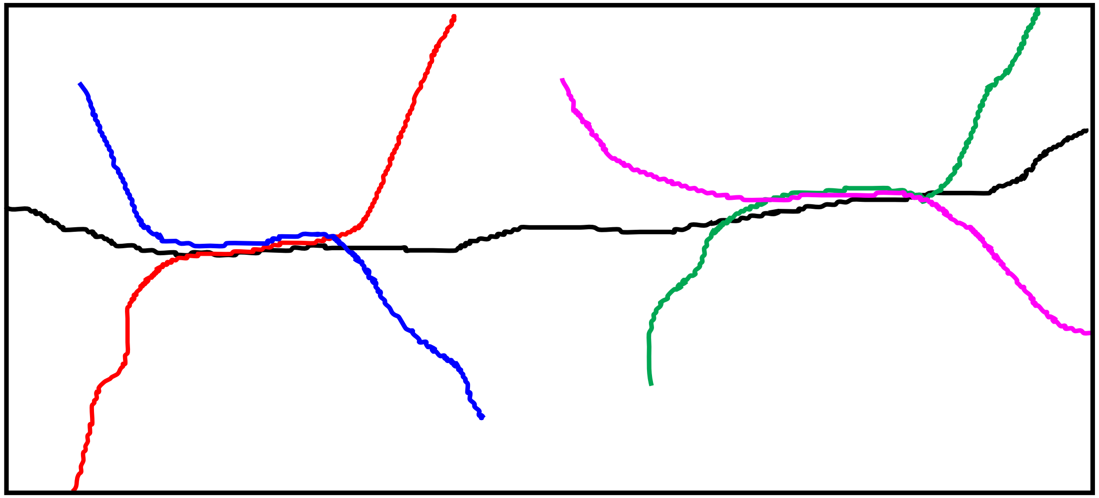
Motivation

- Extract high-level shared structure from *large* trajectory data sets



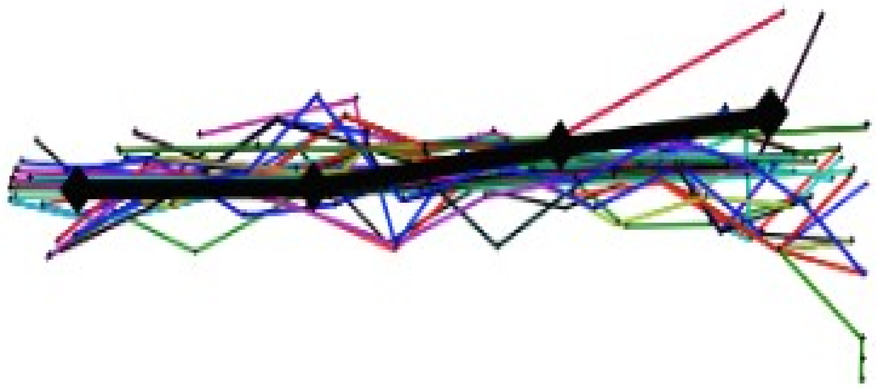
Motivation

- Extract high-level shared structure from *large* trajectory data sets



- Leverage ***wisdom of the crowd***

Pathlet



Representative ***pathlet*** for each cluster

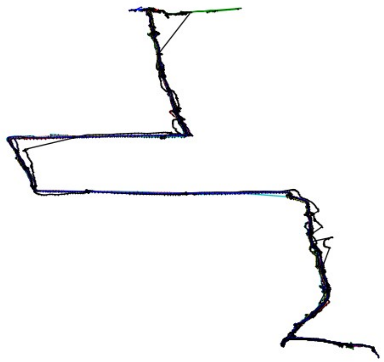
- Cluster “center”
- Pathlet is a curve, not necessarily part of the input

Application of pathlets

- Compression of large trajectory data [Chen et al. 2013]
  - Hope that each trajectory can be reconstructed with small no. of pathlets
  - Small pathlet dictionary - non-linear dimension reduction
- Can provide semantic information
  - Useful for anomaly detection [Sung et al. 2012]
- Reconstructing road network from trajectory data [Li et al. 2013; Buchin et al. 2017]

Our contribution

- Model for subtrajectory clustering
  - Robust to noise and missing data
  - Data-driven clusters and pathlets



- NP-hardness of subtrajectory clustering problem
- Provably-efficient approximation algorithms
  - Faster algorithms for **realistic inputs**
- Experimental results

Previous work

- Graph setting – no noise or gaps [Chen et al. 2013]
- Based only on point density [Panagiotakis et al. 2012]
- Restricted to line segments [Lee et al. 2007]
- Search for pre-defined patterns [Fan et al. 2016; Tang et al. 2013; Wang et al. 2015; Zheng et al. 2013]

None of these have provable performance guarantees

Outline of talk

- Model and problem formulation
- Algorithms
- Experiments

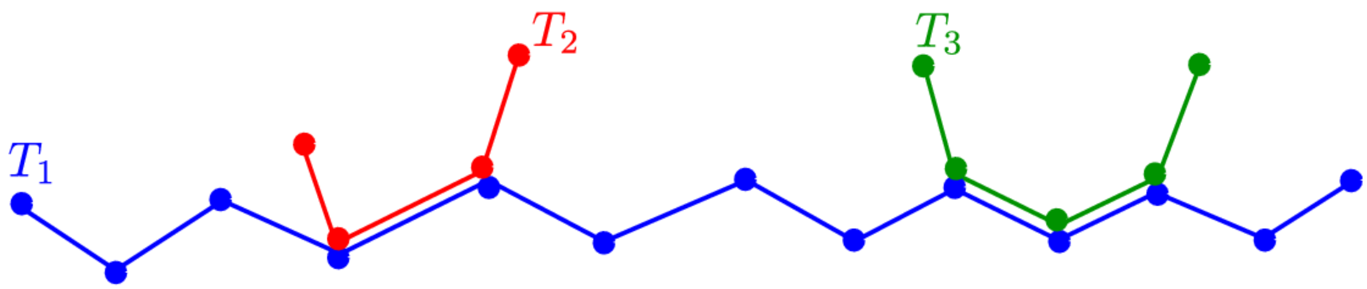
Outline of talk

- Model and problem formulation

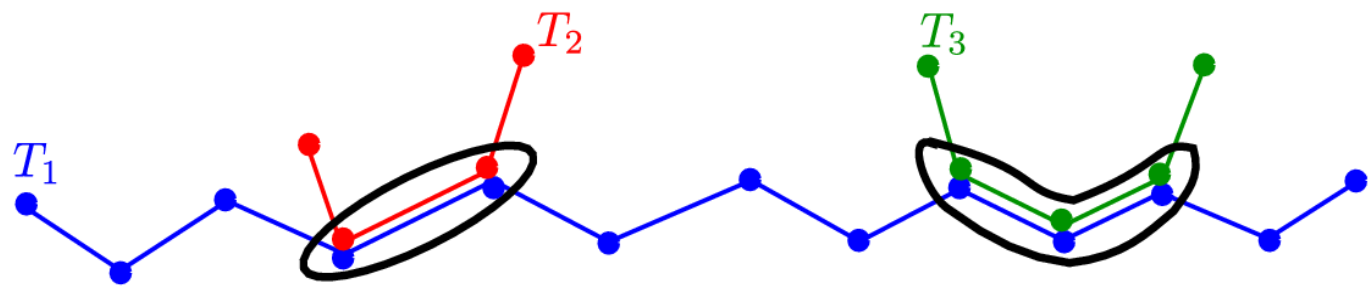
Input

- Trajectories : $\boldsymbol{\mathcal{T} = \{T_1, \ldots, T_n\}}$
- Each trajectory is sequence of points $\boldsymbol{\langle p_1, p_2, \ldots \rangle}$ in $\mathbb{R}^{\mathbf{2}}$
  - Subtrajectory is subsequence of traj.
- Let $\mathbb{X} = \cup_i \boldsymbol{T_i}$ be all trajectory points, $|\mathbb{X}| = \boldsymbol{m}$

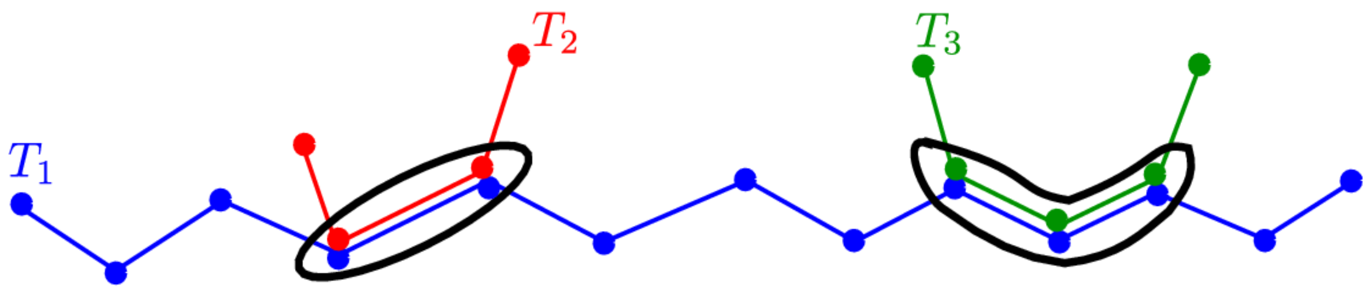
Objective function



Objective function



Objective function

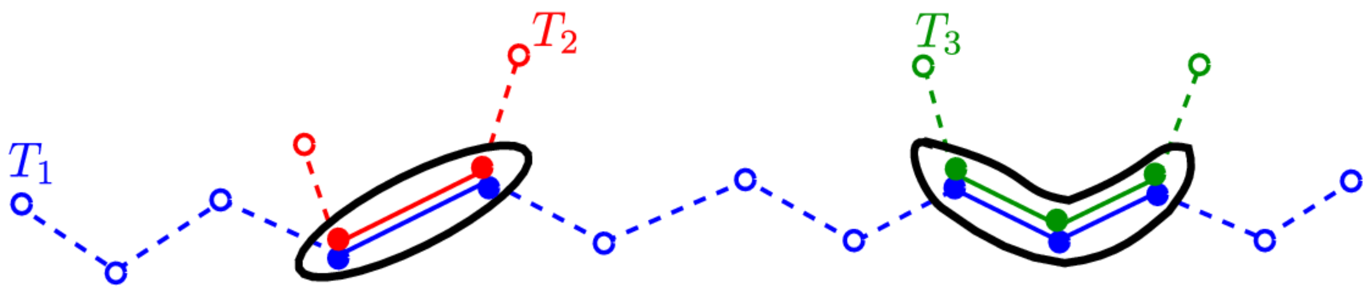


$$|\mathcal{P}| + \sum_{P \in \mathcal{P}} \sum_{S \in \mathcal{T}(P)} d(S, P)$$

Need small # pathlets

Measure of cluster quality

Objective function

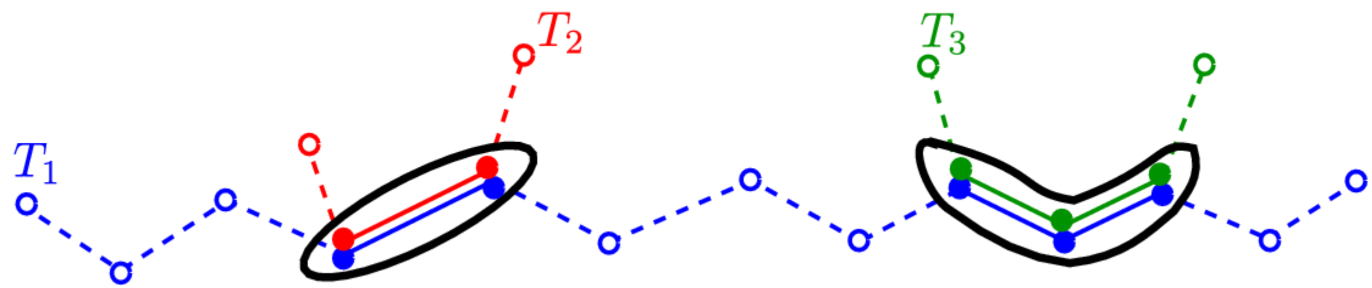


$$|\mathcal{P}| + \sum_{P \in \mathcal{P}} \sum_{S \in \mathcal{T}(P)} d(S, P)$$

Need small # pathlets

Measure of cluster quality

Objective function



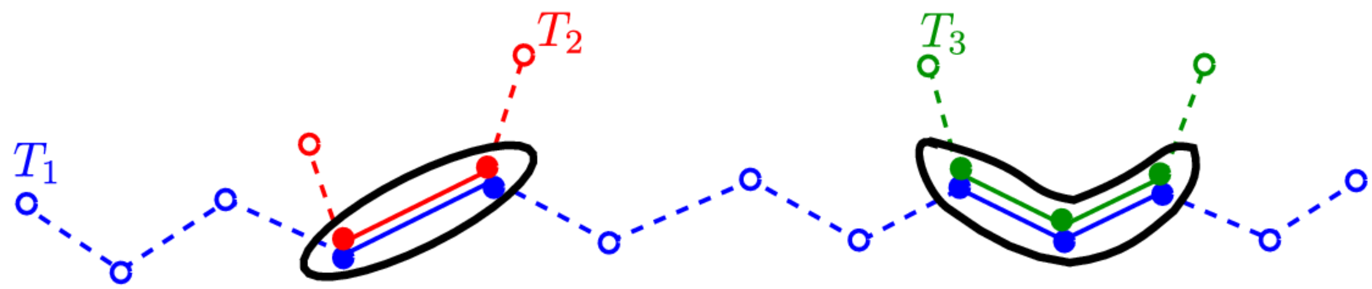
$$|\mathcal{P}| + \sum_{P \in \mathcal{P}} \sum_{S \in \mathcal{T}(P)} d(S, P) + \sum_{T \in \mathcal{T}} \phi(T)$$

Need small # pathlets

Measure of cluster quality

Fraction of points unassigned for each trajectory : "gaps"

Objective function



$$c_1\,|\mathcal{P}| \quad + c_2 \sum_{P \in \mathcal{P}} \sum_{S \in \mathcal{T}(P)} d(S, P) \quad + \; c_3 \sum_{T \in \mathcal{T}} \phi(T)$$

A note on the distance

We use **discrete Fréchet distance**

Given $T_1 = \langle p_1, p_2, \ldots \rangle$ and $T_2 = \langle q_1, q_2, \ldots \rangle$

- **Correspondence** $C \subseteq T_1 \times T_2$ s.t. every pt. in at least one pair

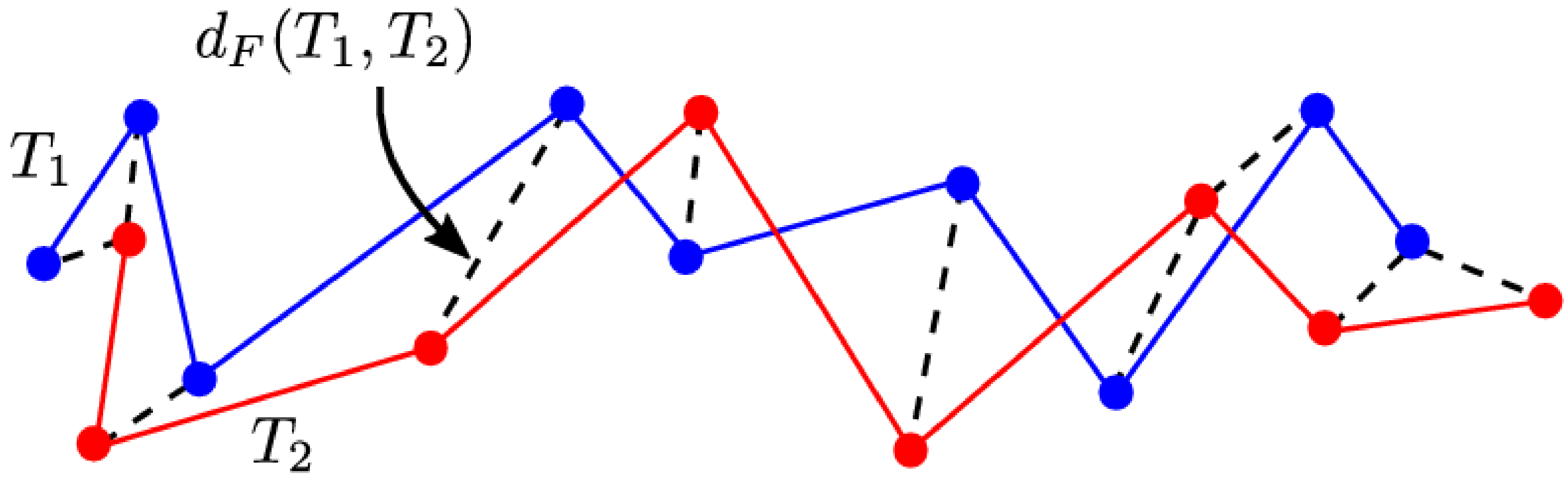
- C is **monotone** if for all $(p_i, q_{i'}), (p_j, q_{j'}) \in C$,

$$j \geq i \Rightarrow j' \geq i'$$

Discrete Fréchet distance

$$d_F(T_1, T_2) = \min_{C \in \mathbb{C}} \max_{p,q \in C} ||p - q||$$

$\mathbb{C}$: Set of all monotonone correspondencess b/w T_1, T_2



Choosing pathlets

Given $\mathcal{T}$, goal is to choose $\mathcal{P}^*$ from set of candidate pathlets $\mathbb{P}$ to minimize objective function

- If $\mathbb{P}$ is given as input : **pathlet-cover problem**

- If $\mathbb{P}$ not given but assumed to be (uncountably) infinite set of all trajectories in plane : **subtrajectory-clustering problem**

Outline of talk

- Model and problem formulation
- Algorithms
- Experiments

Outline of talk

- Algorithms

Basic idea

- Reduce to **set-cover**

- Solve using greedy algorithm : gives $\boldsymbol{O(\log |\mathbb{X}|)}$ approximation

- **Challenge :** implementing greedy step efficiently

Set-cover

Input :

- Set system $(X, \mathcal{S})$
- Weight $w : \mathcal{S} \to \mathbb{R}^+$

Goal is to find $\mathcal{C} \subseteq \mathcal{S}$ of minimum total weight such that $\bigcup \mathcal{C} = X$

From pathlet-cover to set-cover

$(\mathcal{T}, \mathbb{P}, d) \rightarrow (X, \mathcal{S}, w)$

- $X \leftarrow \bigcup_{T \in \mathcal{T}} T$
- $\mathcal{S}$ has two kinds of sets :
  - For all $p \in X$, $\{p\}$ with $w(\{p\}) = c_3/|T^{(p)}|$ where $p \in T^{(p)}$

From pathlet-cover to set-cover

$$(\mathcal{T}, \mathbb{P}, d) \to (X, \mathcal{S}, w)$$

- $X \leftarrow \bigcup_{T \in \mathcal{T}} T$
- $\mathcal{S}$ has two kinds of sets :
  - For all $P \in \mathbb{P}$ and for any set of subtraj. $\mathcal{R}$,

$$S(P, \mathcal{R}) = \{p \in S \mid S \in \mathcal{R}\}$$

with

$$w(S(P, \mathcal{R})) = c_1 + c_2 \sum_{S \in \mathcal{R}} d(S, P)$$

From pathlet-cover to set-cover

$$(\mathcal{T}, \mathbb{P}, d) \rightarrow (X, \mathcal{S}, w)$$

- $X \leftarrow \bigcup_{T \in \mathcal{T}} T$
- $\mathcal{S}$ has two kinds of sets :
  - For all $P \in \mathbb{P}$ and for any set of subtraj. $\mathcal{R}$,

$$\boxed{S(P, \mathcal{R}) = \{p \in S \mid S \in \mathcal{R}\}}$$

with

Exponential # sets :
cannot construct explicitly!!

$$c_2 \sum_{S \in \mathcal{R}} d(S, P)$$

From pathlet-cover to set-cover

Theorem : There exists bijection between feasible solutions of $(X, \mathcal{S}, w)$ and $(\mathcal{T}, \mathbb{P}, d)$ with same weight and cost across bijection

- For sets of form $\{p\}$: leave p unassigned, and vice versa

- For sets of form $S(P, \mathcal{R})$: assign subtraj. in $\mathcal{R}$ to P , and vice versa

Greedy algorithm for set-cover

Initialize $\mathcal{C} = \{\}$

- At each step add to $\mathcal{C}$ the set in $\mathcal{S}$ that maximizes the *coverage-to-weight ratio*
- Stop when all points are covered

Implementing greedy step

- For $S(P, \mathcal{R})$ let $\rho(P, \mathcal{R})$ denote coverage-to-weight ratio

Implementing greedy step

- For $S(P, \mathcal{R})$ let $\rho(P, \mathcal{R})$ denote coverage-to-weight ratio

$$\rho(P, \mathcal{R}) = \frac{\sum_{S \in \mathcal{R}} |\hat{S}|}{c_1 + c_2 \sum_{S \in \mathcal{R}} d(S, P)}$$

where $\hat{S}$ is set of uncovered pts. of S

Implementing greedy step

- For $\mathcal{S}(P,\mathcal{R})$ let $\rho(P,\mathcal{R})$ denote coverage-to-weight ratio

- Let
$$\mathcal{T}_P = \arg\max_{\mathcal{R}:\mathcal{S}(P,\mathcal{R})\in\mathcal{S}} \rho(P,\mathcal{R})$$
$$P^* = \arg\max_{P\in\mathbb{P}} \rho(P,\mathcal{T}_P)$$

Implementing greedy step

- For $S(P,\mathcal{R})$ let $\rho(P,\mathcal{R})$ denote coverage-to-weight ratio

- Let $$\mathcal{T}_P = \arg\max_{\mathcal{R}:S(P,\mathcal{R})\in\mathcal{S}} \rho(P,\mathcal{R})$$ $$P^* = \arg\max_{P\in\mathbb{P}} \rho(P,\mathcal{T}_P)$$

- For each uncovered pt. p let $\rho(p) = \frac{|T^{(p)}|}{c_2}$ and $$p^* = \arg\max_{p} \rho(p)$$

Implementing greedy step

At every greedy step

- Pick $S(P^*, \mathcal{T}_{P^*})$ or $\{p^*\}$whichever has higher ρ
- Update $\mathcal{T}_P, P^*, p^*$ accordingly

Computing/updating $\mathcal{T}_P$

Form of $\rho(P, \mathcal{R})$ permits to

- Break ρ into contribution corresponding to each traj.
- Independently choose "best" subtraj. from each traj.

Computing/updating $\mathcal{T}_P$

Form of $\rho(P, \mathcal{R})$ permits to

- Break ρ into contribution corresponding to each traj.
- Independently choose "best" subtraj. from each traj.

$\boxed{\text{$\mathcal{T}_P$ can be computed/updated in poly-time without explicitly constructing sets $S(P, \mathcal{R})$!!}}$

Our result

Let $|\mathbb{P}| = b$

- **Theorem :** The greedy algorithm computes a $O(\log m)$ -approximate solution to the pathlet-cover problem in $\tilde{O}(bm^3)$ time

Subtrajectory clustering

Set of candidate pathlets not given, assumed to be ***all possible trajectories***

Reducing # candidate pathlets

- d satisfies triangle inequality :
  - Let candidate pathlets be subtraj. of input traj.
  - # candidate pathlets is $O(m^2)$
  - Optimal solution cost increases by factor of 2

Reducing # candidate pathlets

- d satisfies triangle inequality :
  - Let candidate pathlets be subtraj. of input traj.
  - # candidate pathlets is $O(m^2)$
  - Optimal solution cost increases by factor of 2

- $d = d_F$:
  - Reduce # candidate pathlets to $O(m)$
  - Cost increases by factor of $O(\log m)$

Improved running time

- For realistic inputs can further cut down on # assignments need to consider

- **Theorem :** For realistic curves using Fréchet distance, can compute $O(\log^2 m)$-approximate solution to the subtrajectory clustering problem in $\tilde{O}(m^2)$ time

Outline of talk

- Model and problem formulation
- Algorithms
- Experiments

Outline of talk

- Experiments

Data sets

Real data sets :

- Beijing taxi data [Tsinghua University]
  - 28,000 cabs over 4 days
  - 9 mil. points
  - Incomplete and sparse

Data sets

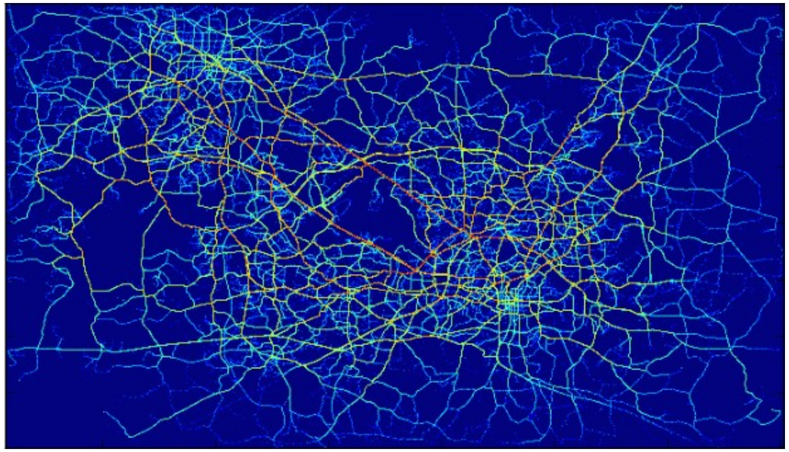
Real data sets :

- GeoLife [Microsoft Research Asia]
  - Pedestrian data of 182 users over 4 years
  - ~2,600 trajs.
  - ~1.5 mill. pts.
- Cycling
  - 37 traj.
  - 106,000 pts.
  - Has self-intersections and loops

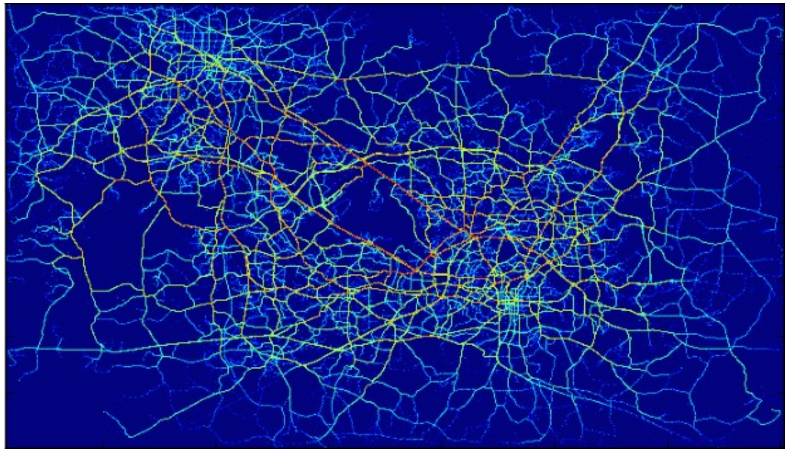
Data sets

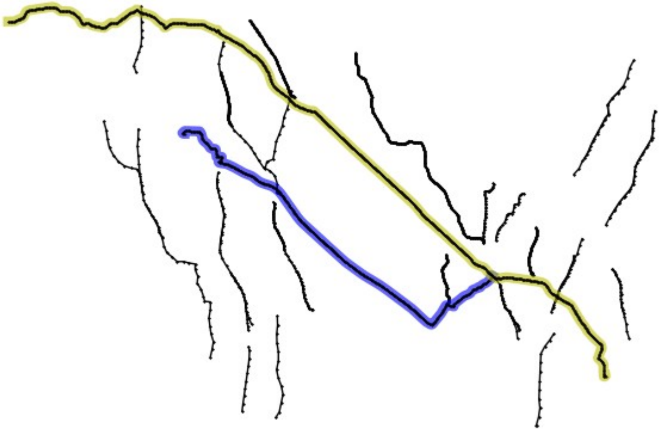
Synthetic data sets :

- RTP
  - Traffic data generated by web-based tool [http://mntg.cs.umn.edu/tg/index.php]
  - Research Triangle in NC
  - ~20,000 traj.
  - ~1 mill. pts.

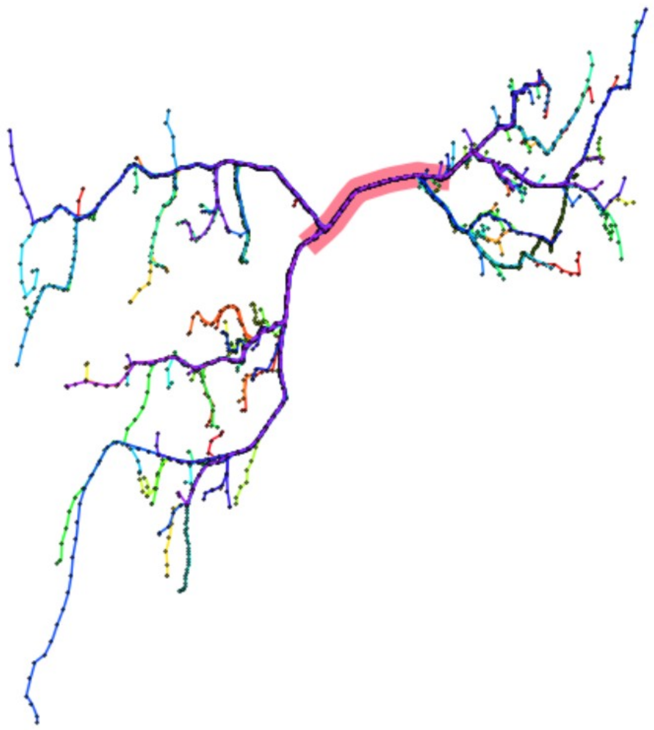


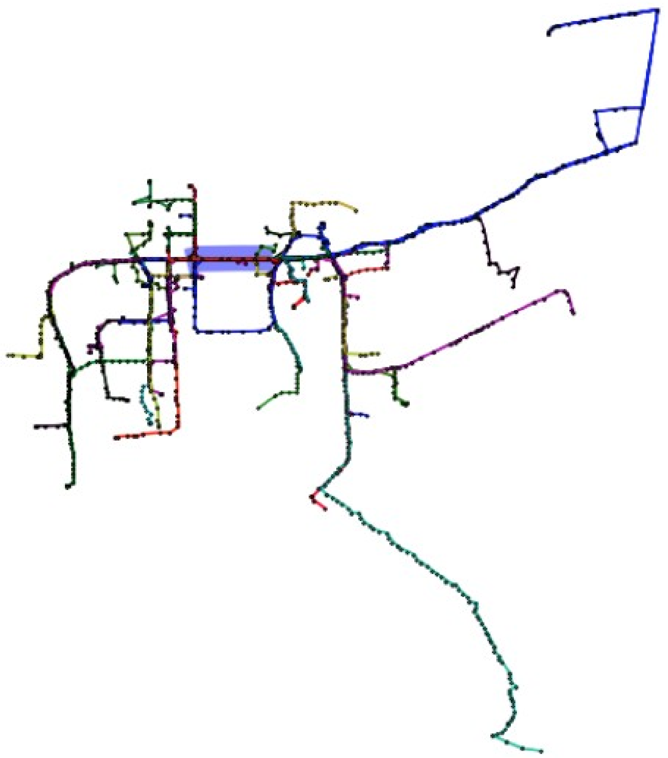
Dense & popular regions



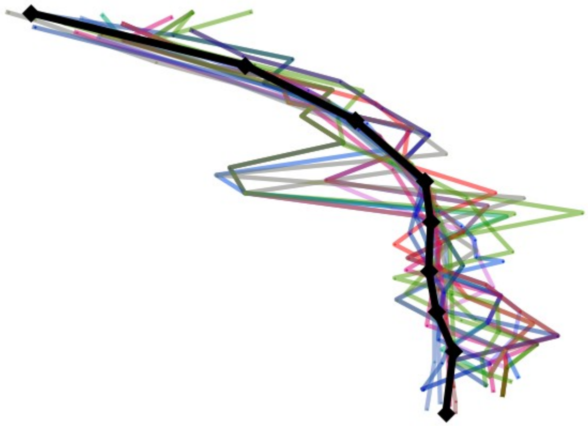


Common trajectory portions

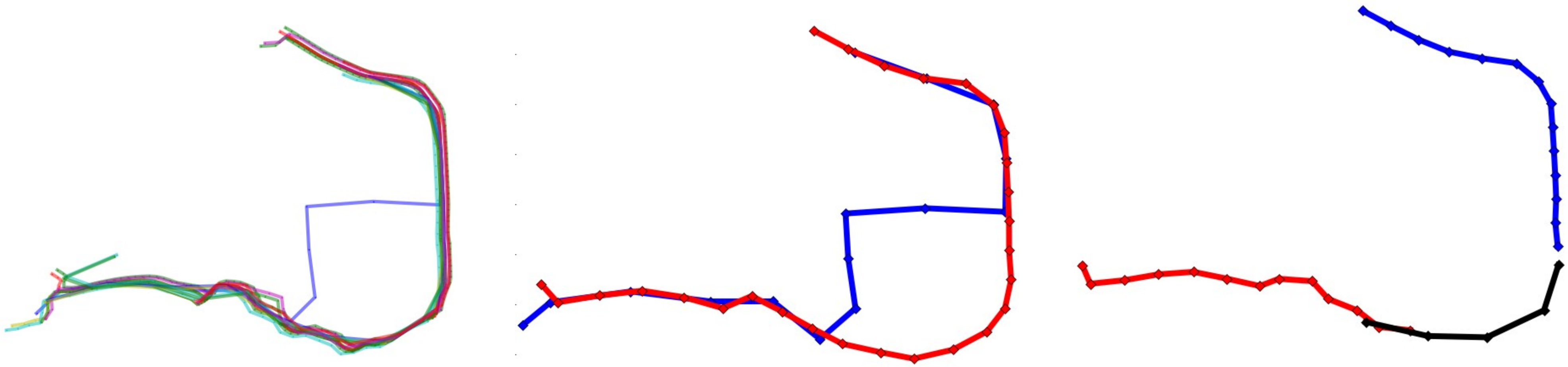




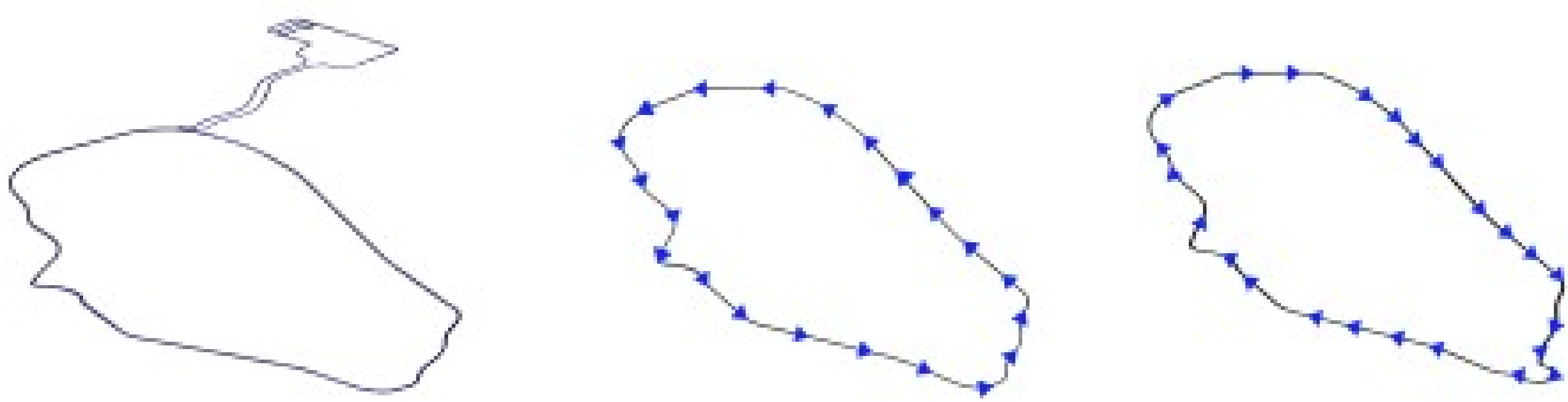
Handling noise



Gaps



Data-driven pathlets



Conclusion

- Proposed new models and algorithms for subtrajectory clustering
  - Theoretical analysis
  - Experiments

- Future directions
  - Clustering trajectories under Fréchet (and other) distances
    - k-means, k-medians, k-center objective ??
  - Going “beyond worst-case analysis”

Thank you!