Trajectory Pattern Mining in Social Media

Ling-Yin Wei (魏綾音)

Advisor: Prof. Wen-Chih Peng
Co-Advisor: Prof. Wang-Chien Lee

Institute of Computer Science and Engineering
National Chiao Tung University

Outline

- **Introduction**
- **Research work**
  - Exploring Pattern-Aware Travel Routes for Trajectory Search
  - Constructing Popular Routes from Uncertain Trajectories
- **Conclusion**
- **Future work**

Introduction (1/2)

- **Social media is everywhere**
  - E.g., GPS logs, geo-tagged photos, check-ins etc.
    - Spatial information
    - Temporal information

Introduction (2/2)

- **Trajectory**
  - A sequence of geo-locations

Work 1: Exploring Pattern-Aware Travel Routes for Trajectory Search

Introduction (1/3)

- **Existing GPS-related web services**
  - Query: a spatial range (e.g., Kenting)
  - Output: a set of trajectories

Which trajectories are interesting?

Introduction (2/3)

- **Existing GPS-related web services**
  - Query: a spatial range (e.g., Kenting)
  - Output: a ranked list of trajectories

Manual Voting

Introduction (3/3)

- **User preferences**
  - In-breadth trip
    - Visit as many regions-of-interest (ROIs) as possible
  - In-depth trip
    - Stop a few ROIs for an in-depth visit

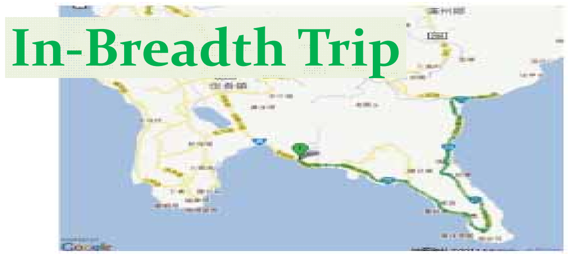
Problem Definition

- **Data**
  - Travel trajectory dataset
    - E.g.,

tid	index	time	lon	lat
1	1	1269130206	120.910332	23.865391
1	2	1269130210	120.910332	23.865391

⋮

- **Query**
  - A spatial range
  - A user-preference of depth/breadth
- **Output**
  - Top K trajectories from historical data
    - Passed the query range



Related Work

- **Trajectory search**
  - Search by a trajectory [Chen et al. SIGMOD'05, Trajcevski et al. ACM GIS'07]
  - Search by locations [Chen et al. SIGMOD'10]

- **Trip planning**
  - Fastest/Popular routes from a source to a destination [Yuan et al. IEEE TKDE'12, Chen et al. ICDE'11]
  - Visiting sequences of regions-of-interest (ROIs) [Zheng et al. WWW'09, Kurashima et al. CIKM'10, Arase et al. ACM MM'10, Yin et al. SDM'11]

Issues (1/2)

- **How to formulate a trajectory score?**
- **Observation**
  - Trip planning
    - A travel route passes through some regions-of-interest (ROIs)

Issues (2/2)

- **How to efficiently search top K trajectories from historical trajectories?**
- **Idea**
  - Only compute the scores of candidate trajectories
  - Iteratively reduce the searching space

Framework

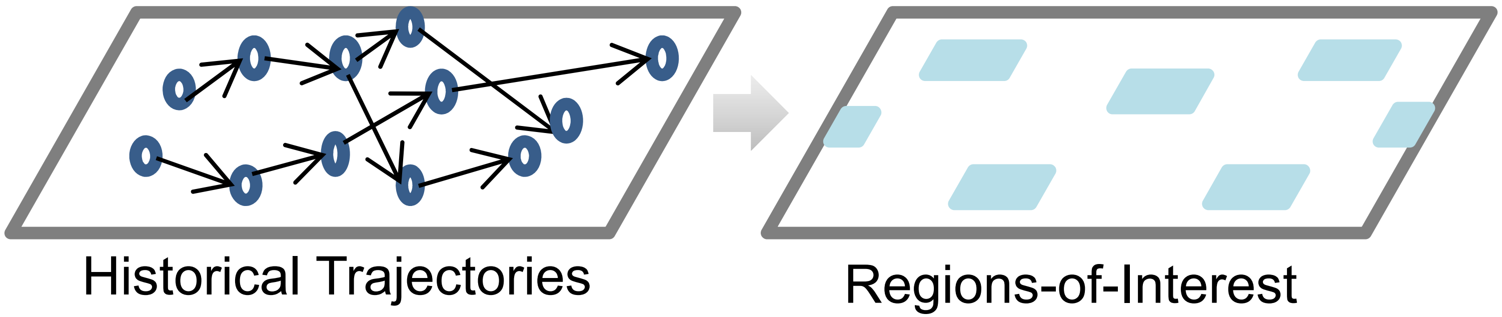
- **Travel behavior exploration (off-line)**
  - Regions-of-interest (ROIs) discovery
  - ROIs scoring
    - User movement graph
- **Trajectory search (on-line)**
  - Trajectory score functions
    - Breadth-trip score
    - Depth-trip score
  - Bounded trajectory search algorithm

Travel Behavior Exploration (1/4)

Phase 1: ROIs discovery 從路徑建立ROI

- **Divide a 2D space into non-overlapping cells**
  - Uncertainty of GPS logs
    - Discover ROIs instead of points-of-interest (POIs)
- **Determine a set of ROIs**
  - A cell is a ROI if

    the number of distinct trajectories that passed the cell is greater than a user-specified threshold



Travel Behavior Exploration (2/4)

Phase 2: ROIs scoring 將路徑轉為ROI順序

- **Transform trajectories**
  - Transform each trajectory into a sequence of ROIs
    - E.g.,

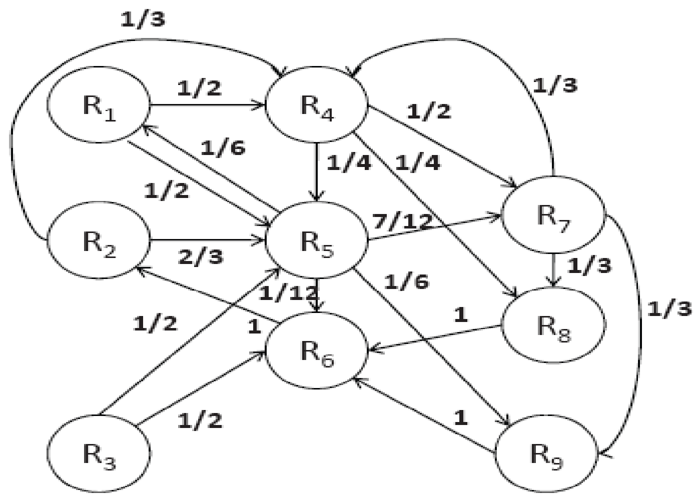
Tid	Sequence of ROIs (Travel Route)
Tr_1	$R_1 \dashrightarrow R_4 \dashrightarrow R_7$
Tr_2	$R_2 \dashrightarrow R_5 \dashrightarrow R_7 \dashrightarrow R_4 \dashrightarrow R_5 \dashrightarrow R_6$

- **Construct a user movement graph**
  - Travel sequential relationships

Travel Behavior Exploration (3/4)

- **User movement graph**

Tid	Sequence of ROIs (Travel Route)
Tr_1	$R_1 \dashrightarrow R_4 \dashrightarrow R_7$
Tr_2	$R_2 \dashrightarrow R_5 \dashrightarrow R_7 \dashrightarrow R_4 \dashrightarrow R_5 \dashrightarrow R_6$
Tr_3	$R_1 \dashrightarrow R_5 \dashrightarrow R_7$
Tr_4	$R_2 \dashrightarrow R_4 \dashrightarrow R_8$
Tr_5	$R_2 \dashrightarrow R_5 \dashrightarrow R_7$
Tr_6	$R_5 \dashrightarrow R_7 \dashrightarrow R_8 \dashrightarrow R_6$
Tr_7	$R_3 \dashrightarrow R_5 \dashrightarrow R_1$
Tr_8	$R_3 \dashrightarrow R_6 \dashrightarrow R_2$
Tr_9	$R_5 \dashrightarrow R_9 \dashrightarrow R_6$
Tr_{10}	$R_4 \dashrightarrow R_7 \dashrightarrow R_9$



  - Weight on edge $<R_i,R_j>$ is

$$w_{<R_i,R_j>} = \frac{\displaystyle\sum_{Tr_h \in TRD, <R_i,R_j> \subset Tr_h} \frac{1}{deg^+(R_i|Tr_h)}}{\left|\displaystyle\bigcup_{R_l \in \Re, R_i \neq R_l} \{Tr_k | Tr_k \in TRD, < R_i, R_l > \subset Tr_k\}\right|}$$

where $deg^+(R_i|Tr) = |\{R_j| < R_i, R_j > \subset Tr, R_i \neq R_j\}|$

Travel Behavior Exploration (4/4)

- **Attractive scores of ROIs**
  - Iteratively compute it until scores converge
  - $$\begin{pmatrix} S^i(R_1) \\ S^i(R_2) \\ \vdots \\ S^i(R_m) \end{pmatrix} = M \cdot \begin{pmatrix} 1 \\ S^{i-1}(R_1) \\ \vdots \\ S^{i-1}(R_m) \end{pmatrix}$$

    Si(R1) = (1-a)*1 + a*w<R1,R1>*Si-1(R1) + a*w<R2,R1>*Si-1(R2)…

  where

  - $$M = \begin{pmatrix} 1-\alpha & \alpha \cdot w_{<R_1,R_1>} & \cdots & \alpha \cdot w_{<R_m,R_1>} \\ 1-\alpha & \alpha \cdot w_{<R_1,R_2>} & \cdots & \alpha \cdot w_{<R_m,R_2>} \\ \vdots & & \ddots & \\ 1-\alpha & \alpha \cdot w_{<R_1,R_m>} & \cdots & \alpha \cdot w_{<R_m,R_m>} \end{pmatrix}$$
  - parameter $\alpha \in [0, 1)$
  - $w_{<R_i,R_i>} = 0$

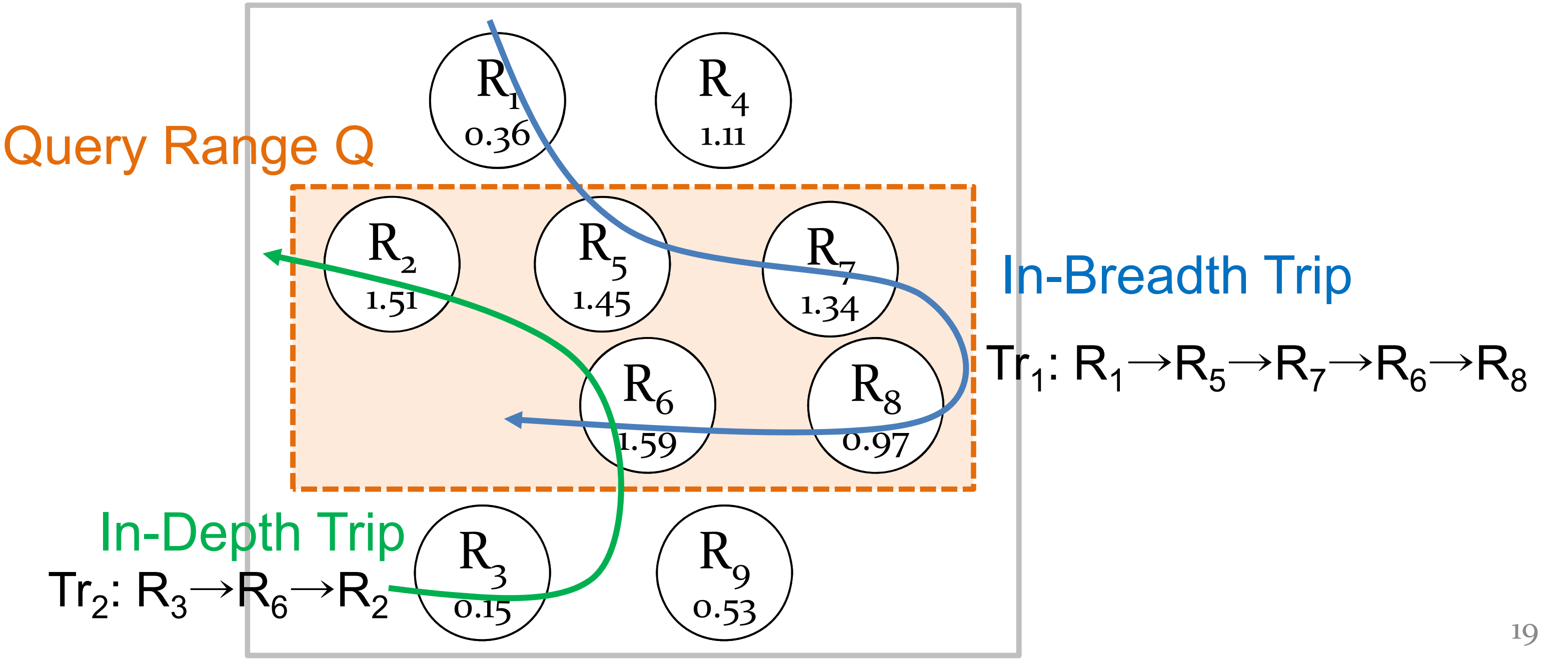
Framework

- **Travel behavior exploration (off-line)**
  - Regions-of-interest (ROIs) discovery
  - ROIs scoring
    - User movement graph

- **Trajectory search (on-line)**
  - Trajectory score functions
    - Breadth-trip score
    - Depth-trip score
  - Bounded trajectory search (BTS) algorithm

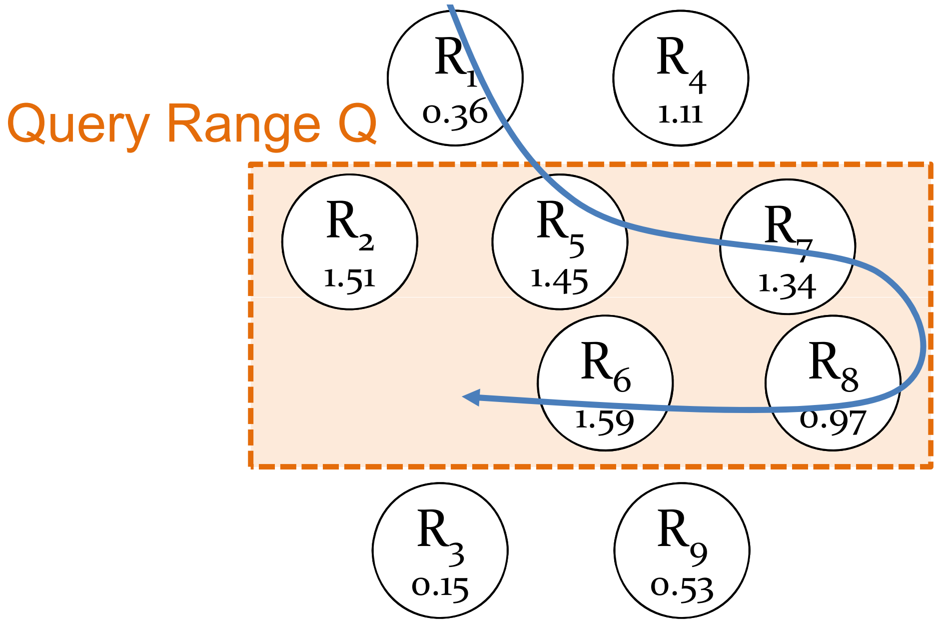
Trajectory Score Function (1/2)

- **In-breadth trip**
  - Visit as many ROIs as possible
- **In-depth trip**
  - Stop a few ROIs for an in-depth visit



Trajectory Score Function (2/2)

- **Breadth-trip score**
  - $BT(Tr) = \sum_{\{R_i | R_i \in Tr.VS \cap R_Q\}} S(R_i)$
- **Depth-trip score**
  - $DT(Tr) = \frac{1}{|\{R_i | R_i \in Tr.VS \cap R_Q\}|} \sum_{\{R_i | R_i \in Tr.VS \cap R_Q\}} S(R_i)$



Tr_1: $R_1 \rightarrow R_5 \rightarrow R_7 \rightarrow R_6 \rightarrow R_8$

$BT(Tr_1) = 1.45+1.34+1.59+0.97=5.35$

$DT(Tr_1) = (1.45+1.34+1.59+0.97)/4$
$= 1.338$

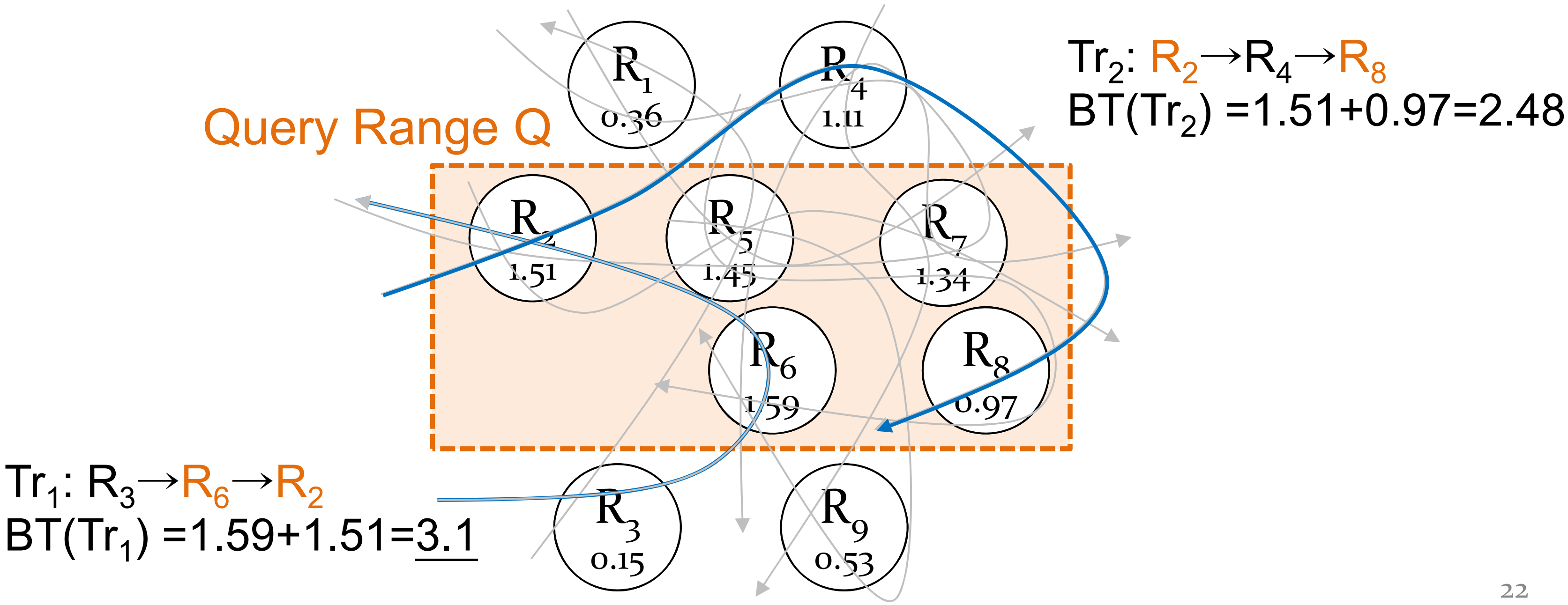
Trajectory Search (1/2)

- **Naïve method**
  - Calculate the scores of the trajectories passing query range Q and then rank them

Tid	Sequence of ROIs (Travel Route)
Tr_1	$R_1 \dashrightarrow R_4 \dashrightarrow R_7$
Tr_2	$R_2 \dashrightarrow R_5 \dashrightarrow R_7 \dashrightarrow R_4 \dashrightarrow R_5 \dashrightarrow R_6$
Tr_3	$R_1 \dashrightarrow R_5 \dashrightarrow R_7$
Tr_4	$R_2 \dashrightarrow R_4 \dashrightarrow R_8$
Tr_5	$R_2 \dashrightarrow R_5 \dashrightarrow R_7$
Tr_6	$R_5 \dashrightarrow R_7 \dashrightarrow R_8 \dashrightarrow R_6$
Tr_7	$R_3 \dashrightarrow R_5 \dashrightarrow R_1$
Tr_8	$R_3 \dashrightarrow R_6 \dashrightarrow R_2$
Tr_9	$R_5 \dashrightarrow R_9 \dashrightarrow R_6$
Tr_{10}	$R_4 \dashrightarrow R_7 \dashrightarrow R_9$

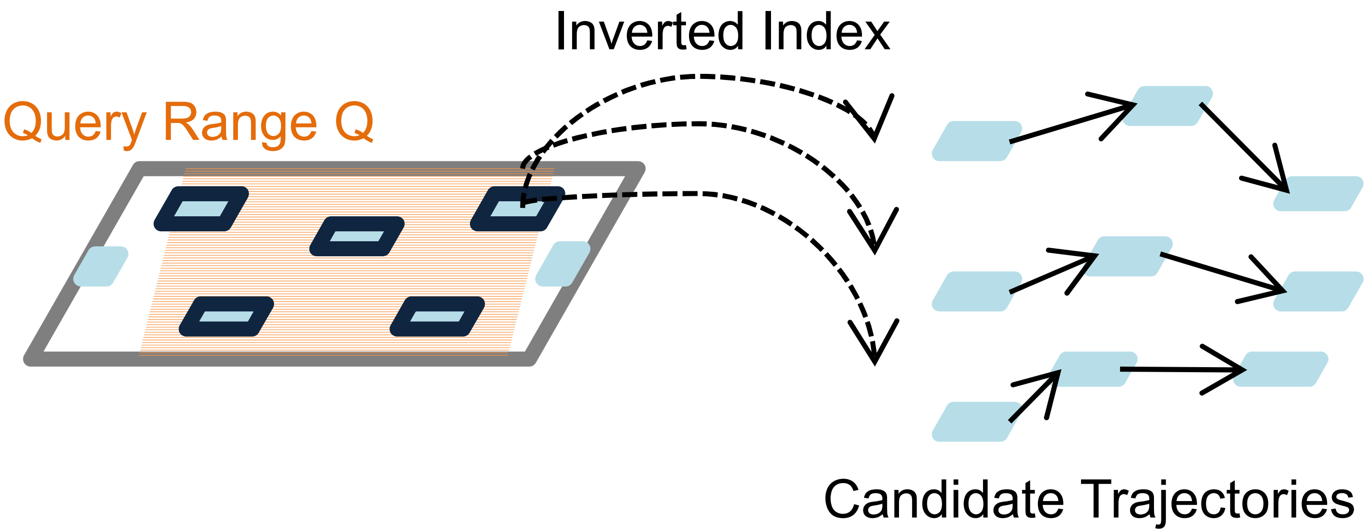
Trajectory Search (2/2)

- **Idea**
  - Only compute the scores of candidate trajectories
- **Observation**
  - A trajectory has a higher score if it passed ROIs having higher attractive scores

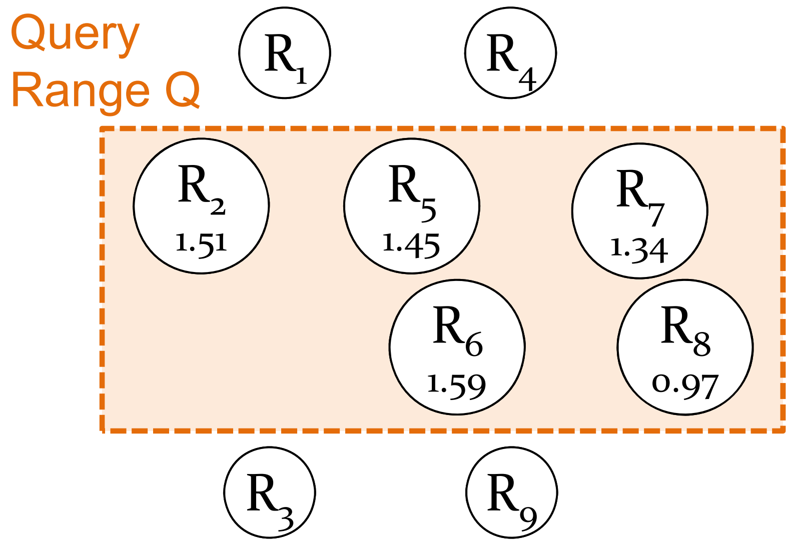


Bounded Trajectory Search (1/3)

- **Idea**
  - Scan ROIs in the query range to select candidate trajectories by ROIs' attractive scores
  - Iteratively reduce the searching space by updating the score at rank K in the set of candidate trajectories



Example: Top 5 Trajectories



Tid	Sequence of ROIs (Travel Route)	DT
Tr_1	$R_1 \dashrightarrow R_4 \dashrightarrow R_7$	
Tr_2	$R_2 \dashrightarrow R_5 \dashrightarrow R_7 \dashrightarrow R_4 \dashrightarrow R_5 \dashrightarrow R_6$	1.48
Tr_3	$R_1 \dashrightarrow R_5 \dashrightarrow R_7$	1.395
Tr_4	$R_2 \dashrightarrow R_4 \dashrightarrow R_8$	1.24
Tr_5	$R_2 \dashrightarrow R_5 \dashrightarrow R_7$	1.433
Tr_6	$R_5 \dashrightarrow R_7 \dashrightarrow R_8 \dashrightarrow R_6$	1.338
Tr_7	$R_3 \dashrightarrow R_5 \dashrightarrow R_1$	1.45
Tr_8	$R_3 \dashrightarrow R_6 \dashrightarrow R_2$	1.55
Tr_9	$R_5 \dashrightarrow R_9 \dashrightarrow R_6$	1.52
Tr_{10}	$R_4 \dashrightarrow R_7 \dashrightarrow R_9$	

Scan ROIs in Q by attractive scores:

$S(R_6) = 1.59$ ⇨ $\{Tr_2, Tr_6, Tr_8, Tr_9\}$

$S(R_2) = 1.51$ ⇨ $\{Tr_4, Tr_5\}$

SBR$_T$ $S(R_5) = 1.45$ ⇨ $\{Tr_3, Tr_7\}$

SBR$_T$ $S(R_7) = 1.34$

SBR$_T$ $S(R_8) = 0.97$

Round	T (Candidate trajectories)	LBS_T	SBR_T
Initial	$\emptyset$	∞	R_8

Top 5 trajectories: $\{Tr_8, Tr_9, Tr_2, Tr_7, Tr_5\}$

Bounded Trajectory Search (2/3)

- **Lower bound of the scores of candidate trajectories**
  - $LBS_T = DT(Tr)$ or $BT(Tr)$
    where T is the candidate trajectories
    and $DT(Tr)$ or $BT(Tr)$ is the score at rank k in T
- **Search bound for examining ROIs w.r.t. LBS_T**
  - $SBR_T = R_l^s$
    where for trajectory score function DT,
    $$l = \max\{i | S(R_i^s) \geq LBS_T, \forall i \in \mathbb{N}, 1 \leq i \leq |R_Q|\}$$
    and for trajectory score function BT,
    $$l = \max\{i | \sum_{i \leq j \leq |R_Q|} S(R_j^s) \geq LBS_T, \forall i \in \mathbb{N}, 1 \leq i \leq |R_Q|\}$$

Bounded Trajectory Search (3/3)

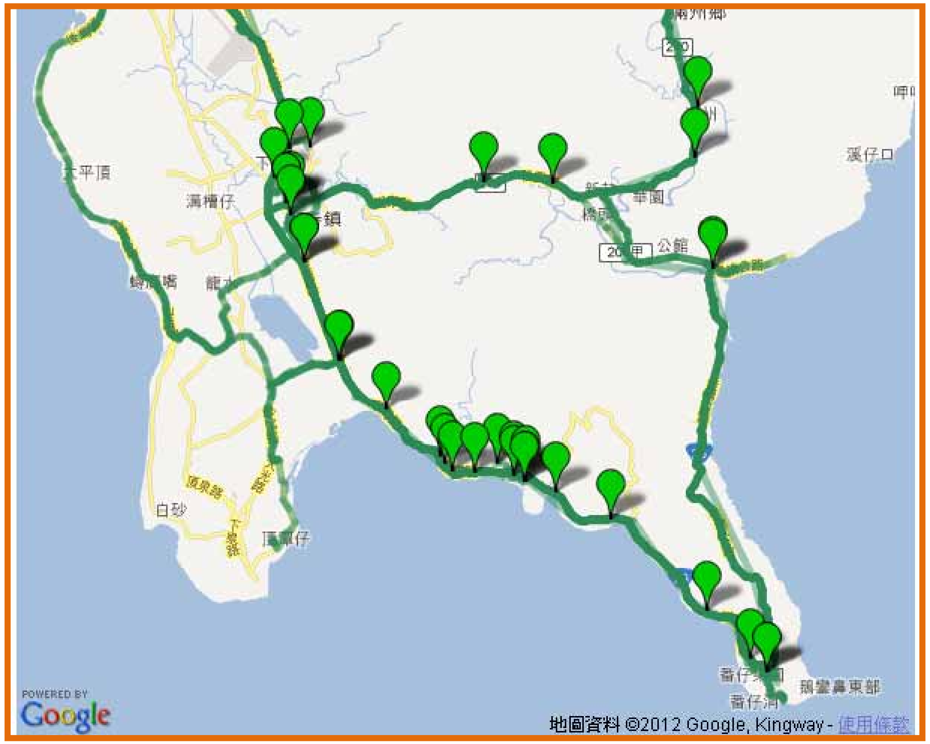
- **Sort ROIs in decreasing order by attractive scores**
- **Scan ROIs from R^s_1 to $SBR_T = R^s_j$**
  - The set of candidate trajectories, T
    - Add the trajectories passed R^s_i into T and calculate their scores
  - Update LBS_T in T if $|T|>k$
  - Update SBR_T w.r.t. LBS_T
- **Thm: Correctness of algorithm BTS**
  - Proofs in Chap. 2.5.3.

Experiments

- **Trajectory data**
  - Travel websites: EveryTrail, Bikemap
  - Taiwan
    - 6,548 trajectories
    - 1,301,192 GPS logs
- **Parameters**
  - Grid length: 300 meters
  - Minimum density threshold: 10
  - Rank-threshold K: 10

Top 1 Trajectory in Kenting (1/2)

- **Query: Kenting**

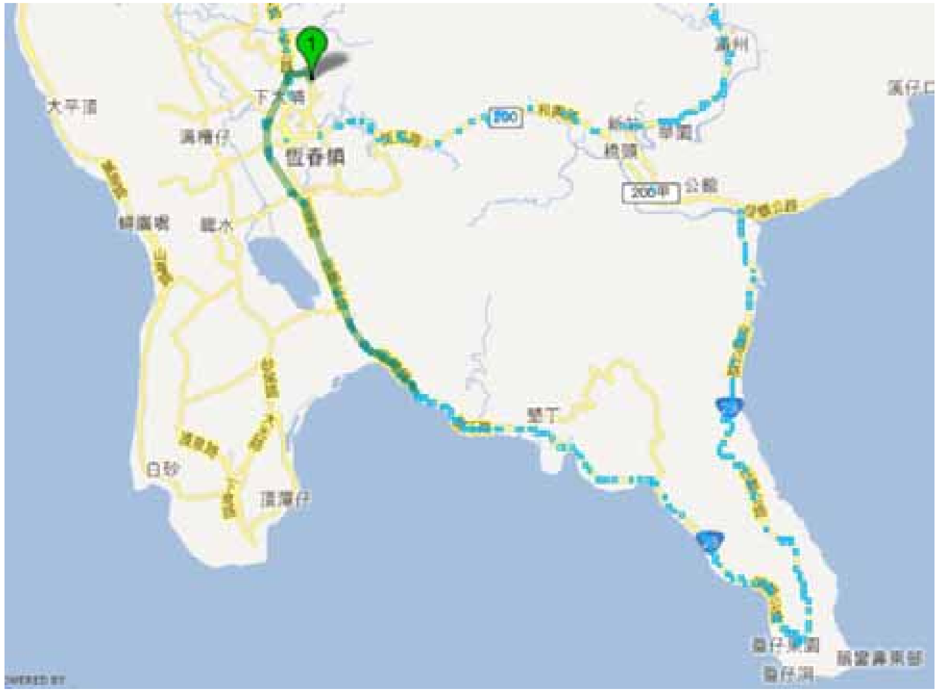


Attractions

Top 1 Trajectory in Kenting (2/2)

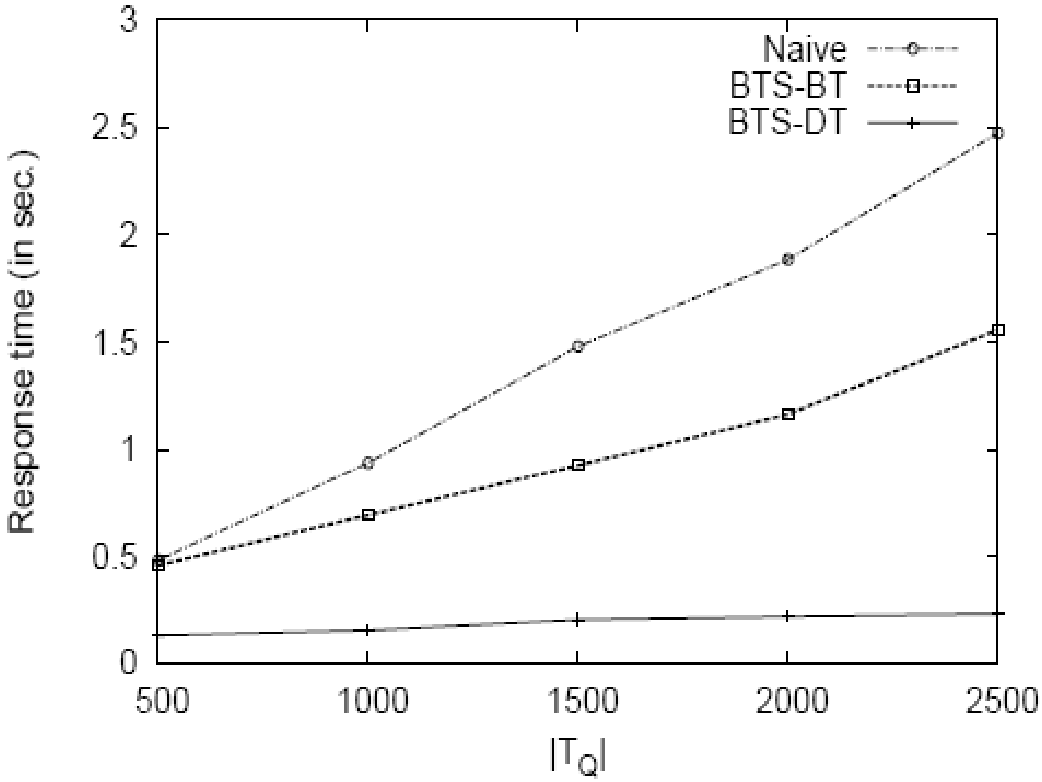
- In-Breadth Trip

- In-Depth Trip



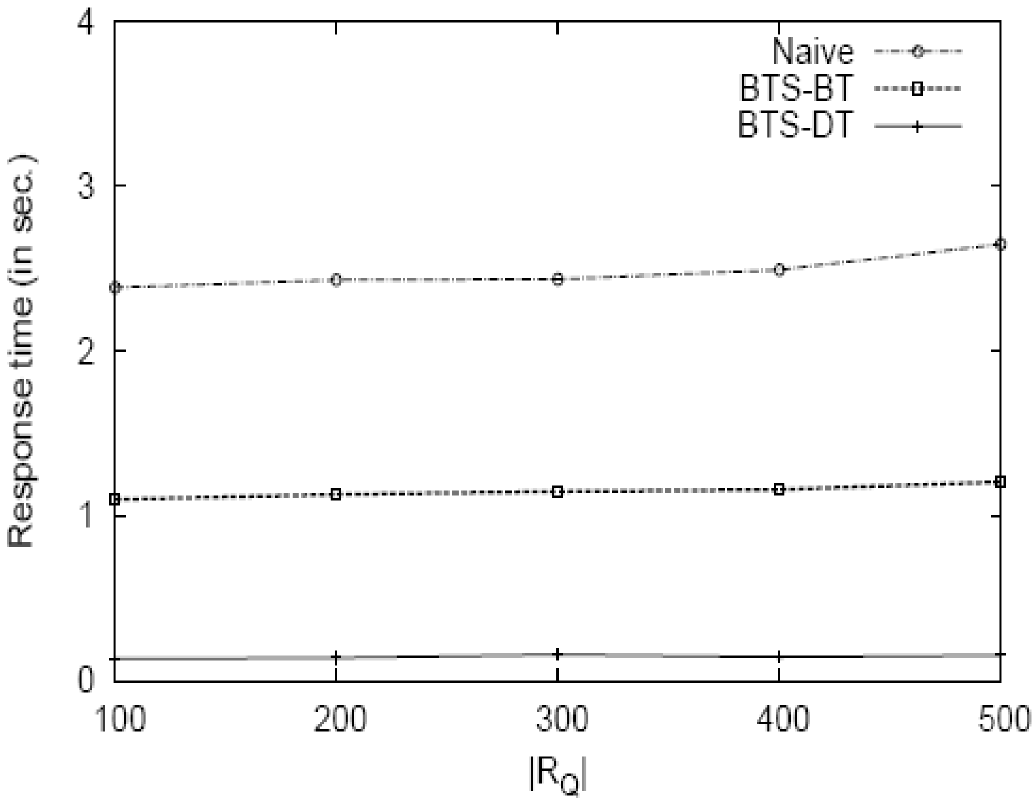
Performance Evaluation (1/3)

- **Impact of the number of the trajectories crossing query range Q, $|T_Q|$**
  - The number of ROIs in Q, $|R_Q|$=500



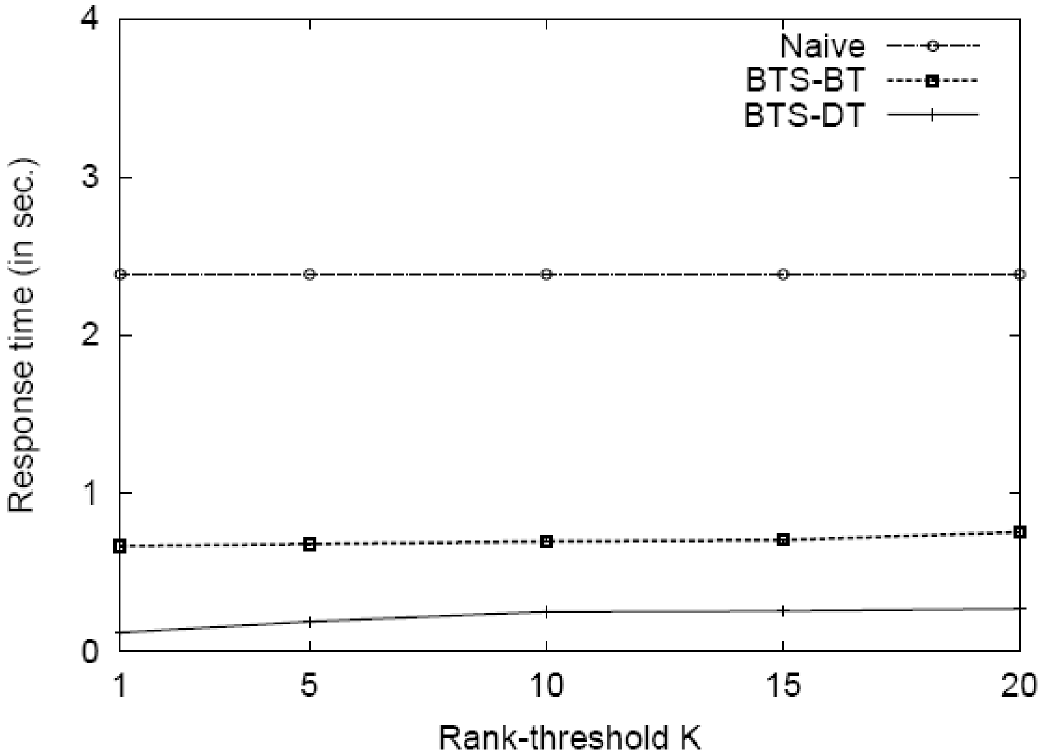
Performance Evaluation (2/3)

- **Impact of the number of ROIs in query range Q, $|R_Q|$**
  - $|T_Q|$=3,892



Performance Evaluation (3/3)

- **Impact of the rank-threshold, K**
  - $|T_Q|$=1,600
  - $|R_Q|$=100



Summary

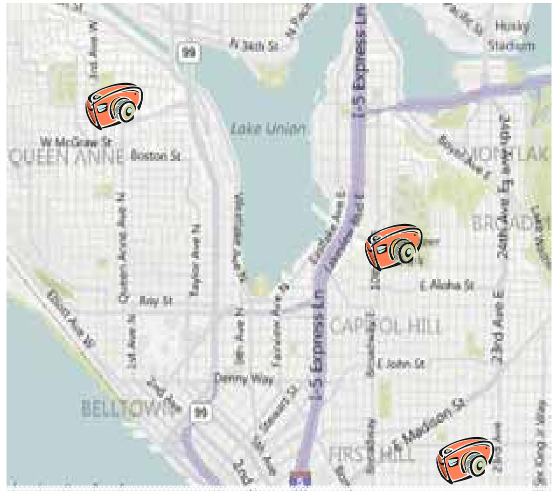
- Presented a novel trajectory search framework for trip planning by considering different user preferences
- Employed a user movement graph to capture travel patterns hidden in trajectory datasets and developed an algorithm to score ROIs
- Developed an algorithm BTS for efficiently retrieving the top K trajectories

Work 2: Constructing Popular Routes from Uncertain Trajectories

Introduction (1/2)

- **Trajectory data**
  - Geo-tagged Flickr photo traces
  - Taxicabs' trajectories
  - Foursquare check-in sequences

- **Uncertainty of trajectories**
  - Energy saving
  - Features of applications



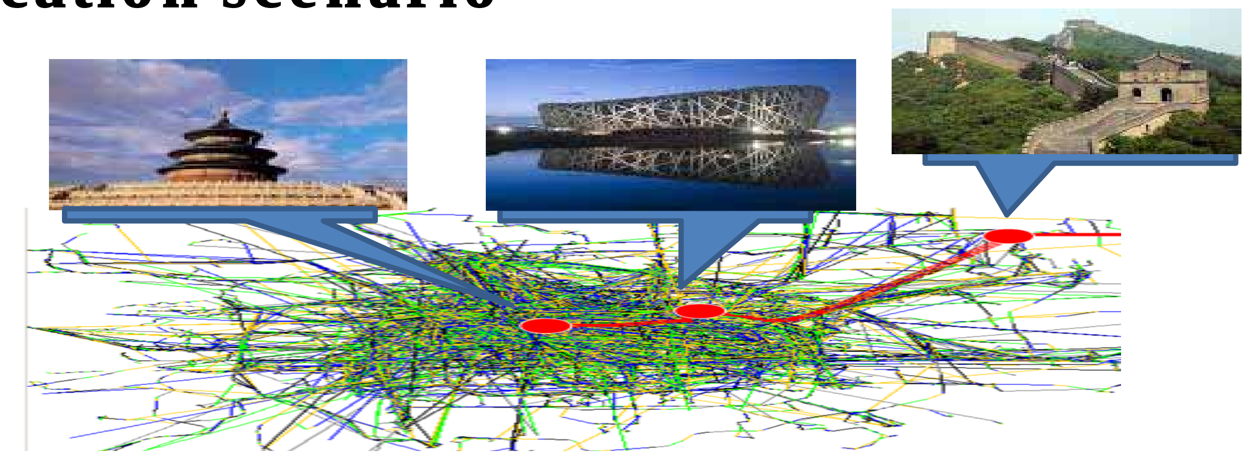
Uncertain Trajectories

Introduction (2/2)

- **Question**
  - What are popular routes between two consecutive geo-locations?

Road Network 2D Free Space

- **Application scenario**



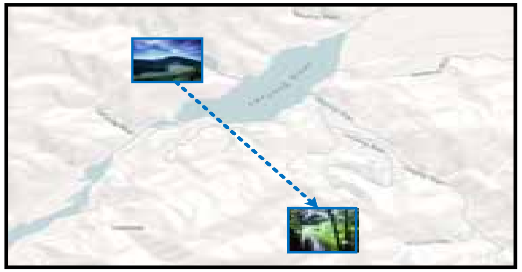
Problem Definition

- **Data**
  - Uncertain trajectory dataset
  - ~~Road networks~~
- **Query**
  - A geo-location sequence q: $q_1 \rightarrow \ldots \rightarrow q_n$ and $q_{i+1}.t - q_i.t \leq \Delta t$ where Δt is a time span
- **Output**
  - Top K popular routes
    - Sequentially traverse the geo-locations

Challenges (1/2)

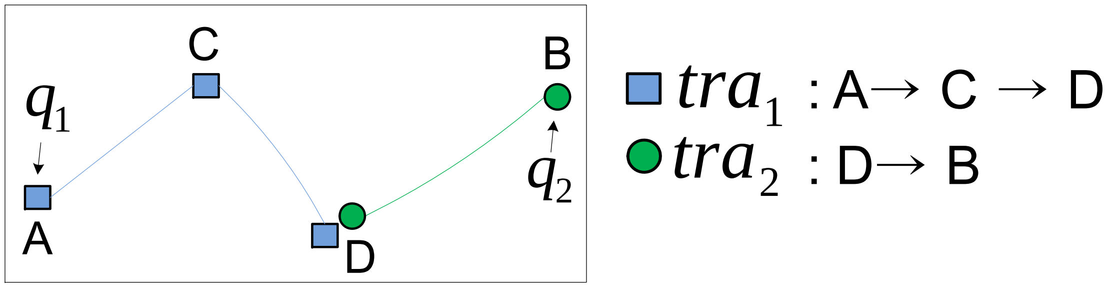
- **Road network information is not applicable or not available**

Road Network



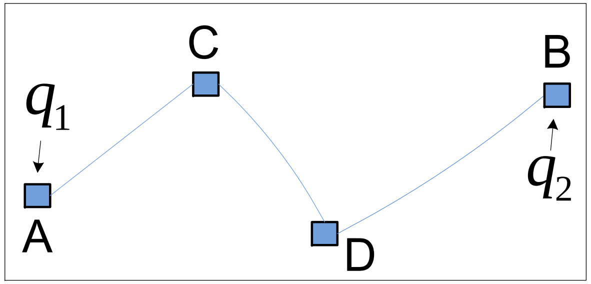
2D Free Space

- **No historical trajectory passes through the given geo-location sequence**



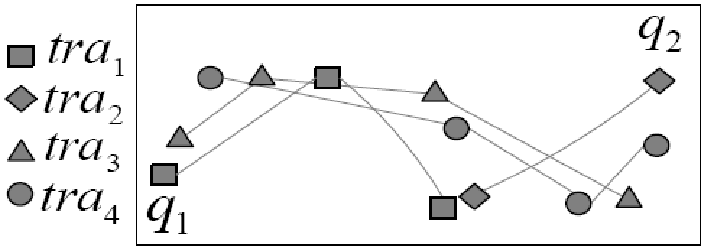
Challenges (2/2)

- **Uncertain trajectories passing through the given geo-location sequence are not detailed routes**

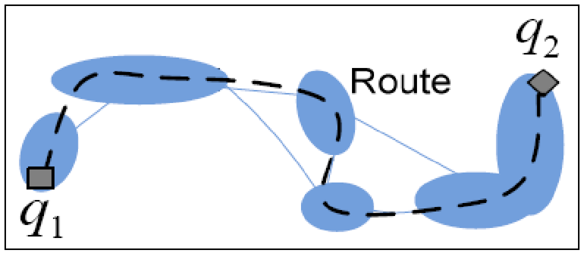


$tra_1: A \rightarrow C \rightarrow D \rightarrow B$

- **Which uncertain trajectories have similar movements?**
  - Infer popular routes based on collective knowledge



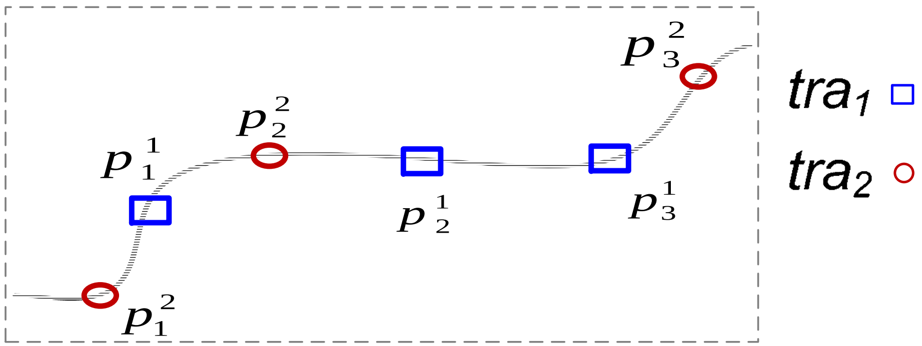
Uncertain trajectories



Mutual reinforcement

Related Work (1/3)

- **Trajectory clustering**
  - 2D free space [Lee et al.SIGMOD'07, Pelekis et al. ICDM'09]
    - Data: high-sampled trajectories
    - Issue for uncertain trajectories
      - Different geo-spatial representations of a route



  - Road network [Kuijpers et al. SSTD'09]
    - Data: low-sampled trajectories
    - Disadvantage: do no detail routes

Related Work (2/3)

- **Trip planning**
  - Data: high-sampled/low-sampled trajectories
    [Zheng et al. WWW'09/Kurashima et al. CIKM'10, Arase et al. MM'10, Yin et al. SDM'11]
  - Output: visiting sequences of ROIs
  - Disadvantage: do not detail routes

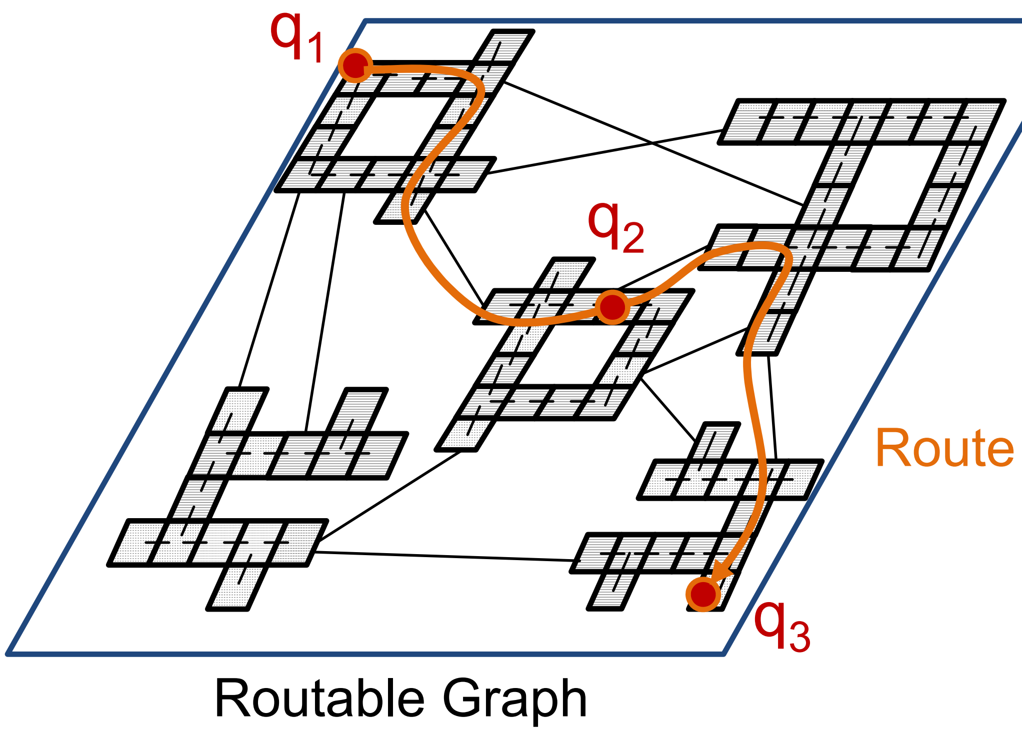
Related Work (3/3)

- **Route planning**
  - Query: two or more geo-locations
  - Output: Top K routes according to the geo-locations

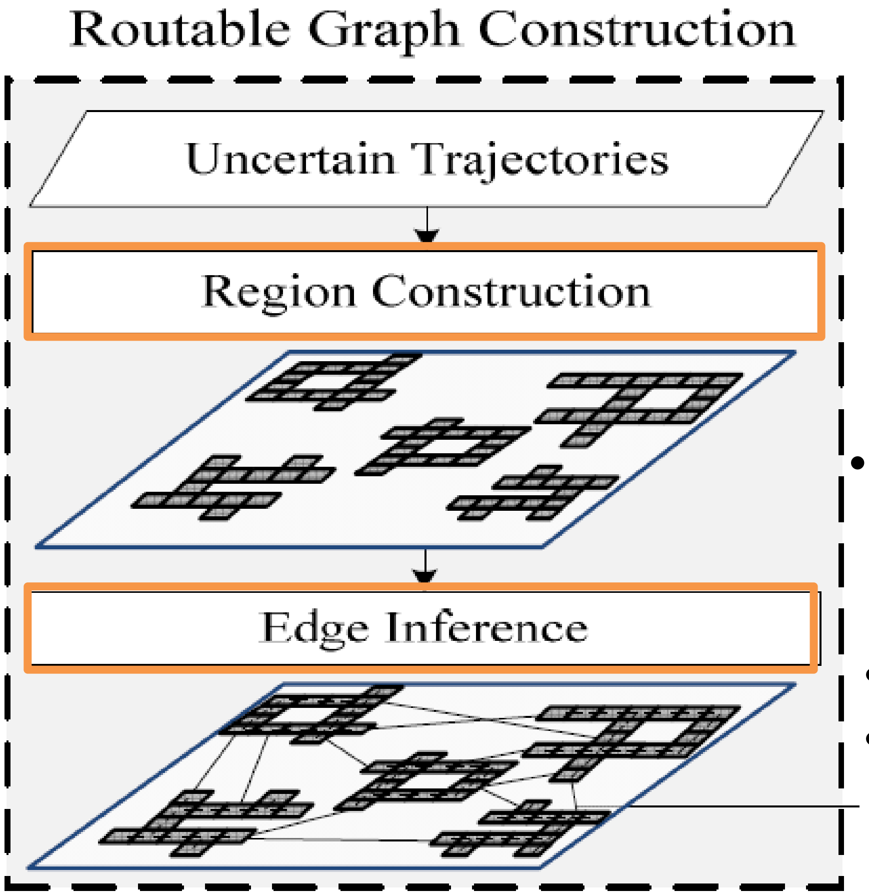
	Trajectory - Sampling Rate	Road Network	Ranking Criteria	Output
Chen et al. SIGMOD'10	High	V	Distance	Historical trajectories
Chen et al. ICDE'11	High		Popularity	Synthetic routes
Yuan et al. TKDE'12	Low	V	Fastest	Synthetic routes
Zheng et al. ICDE'12	Low	V	Popularity	Synthetic routes
Ours	Low		Popularity	Synthetic routes

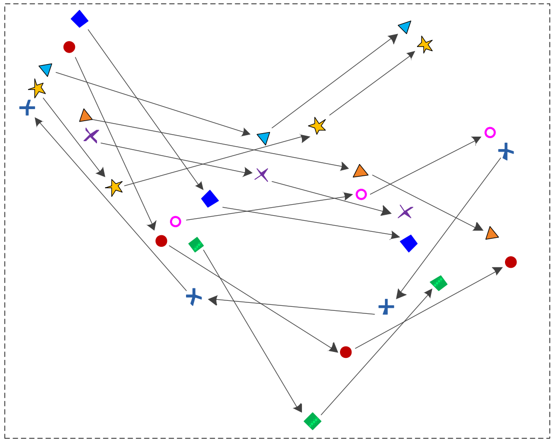
Framework

- Routable graph construction (off-line)
- Route inference (on-line)



Framework Overview



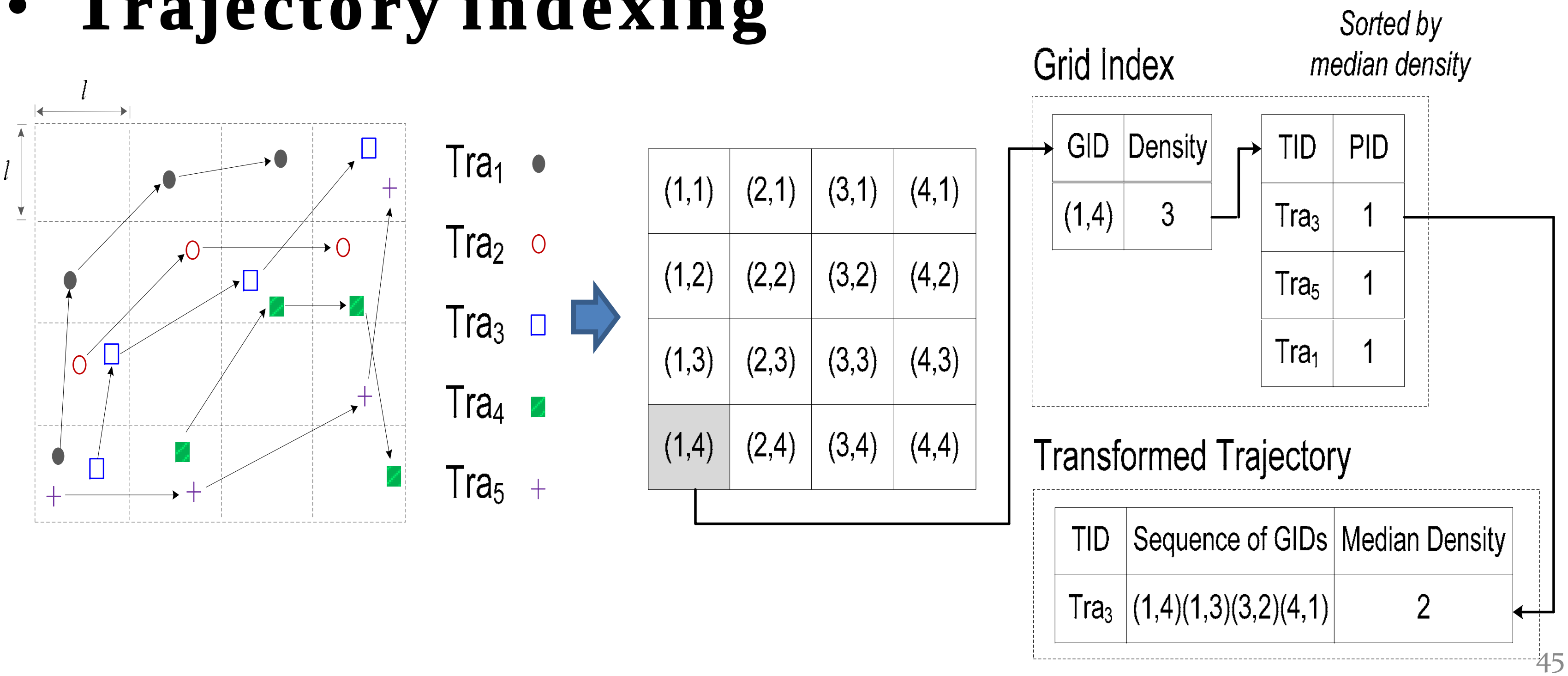


- Region: a connected geographical area

- Edges inside a region
- Edges between regions

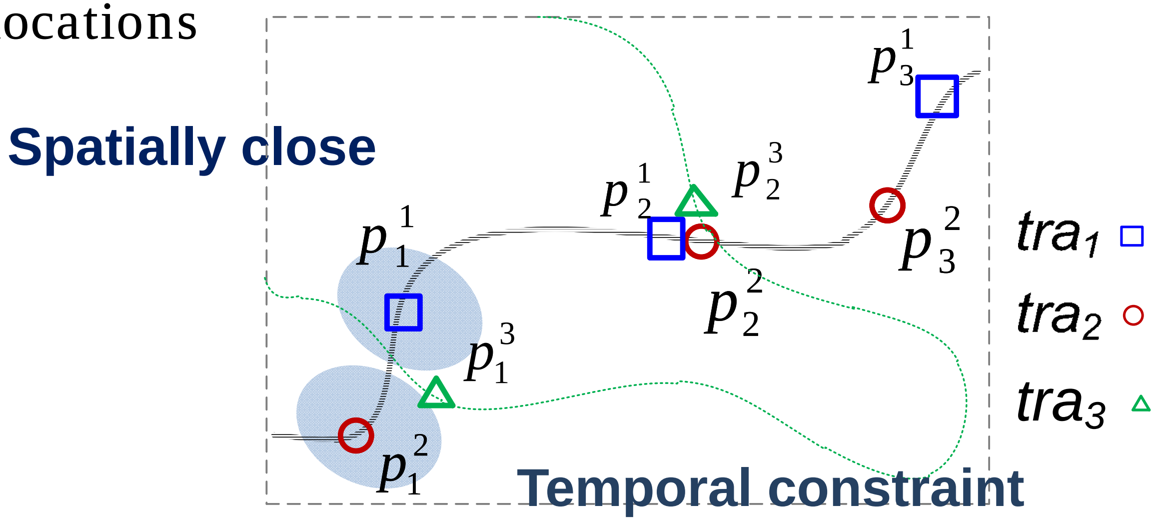
Routable Graph Construction (1/3)

- **Space partition**
  - Divide a space into non-overlapping cells with a given cell length
- **Trajectory indexing**



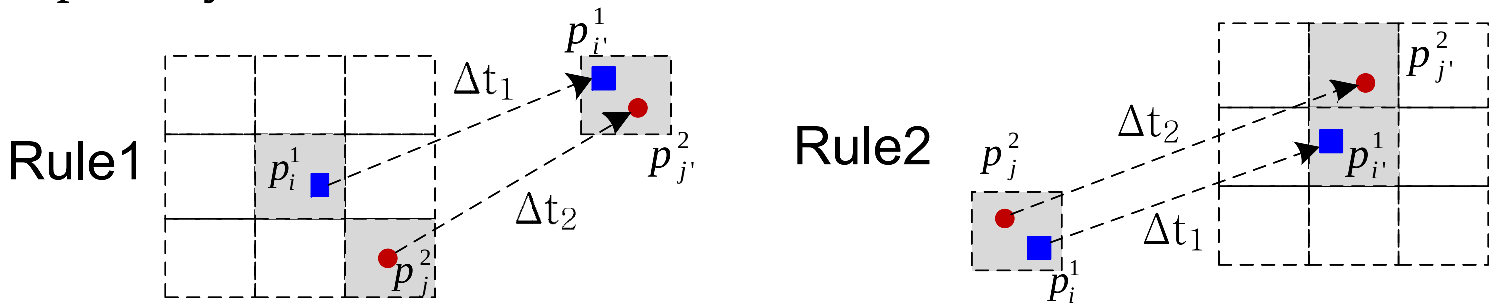
Routable Graph Construction (2/3)

- **Region**
  - A connected geographical area
- **Idea**
  - Merge connected cells to form a region
- **Observation**
  - Tra_1 and Tra_2 follow the same route but have different sampled geo-locations



Routable Graph Construction (3/3)

- **Spatio-temporally correlated relation between trajectories**
  - Spatially close



  - Temporal constraint θ
    - $\frac{|\Delta t_1 - \Delta t_2|}{\max\{\Delta t_1, \Delta t_2\}} \leq \theta$
- **Connection support of a cell pair**
  - Minimum connection support C

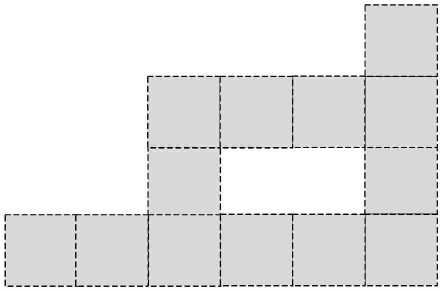
Edge Inference (1/2)

[Edges in a region]

Step 1: Let a region be a bidirectional graph first

Step 2: Infer the *direction*, *travel time* and *support* between each two consecutive cells

- Multiple shortest routes between consecutive points and then propagate the weight evenly
- Aggregate the weights
- Estimate the travel time of each edge



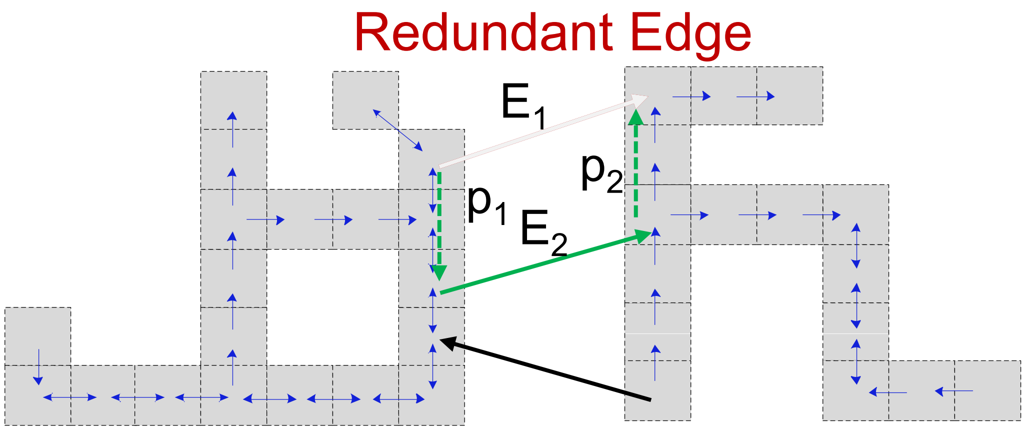
Edge Inference (2/2)

[Edges between regions]

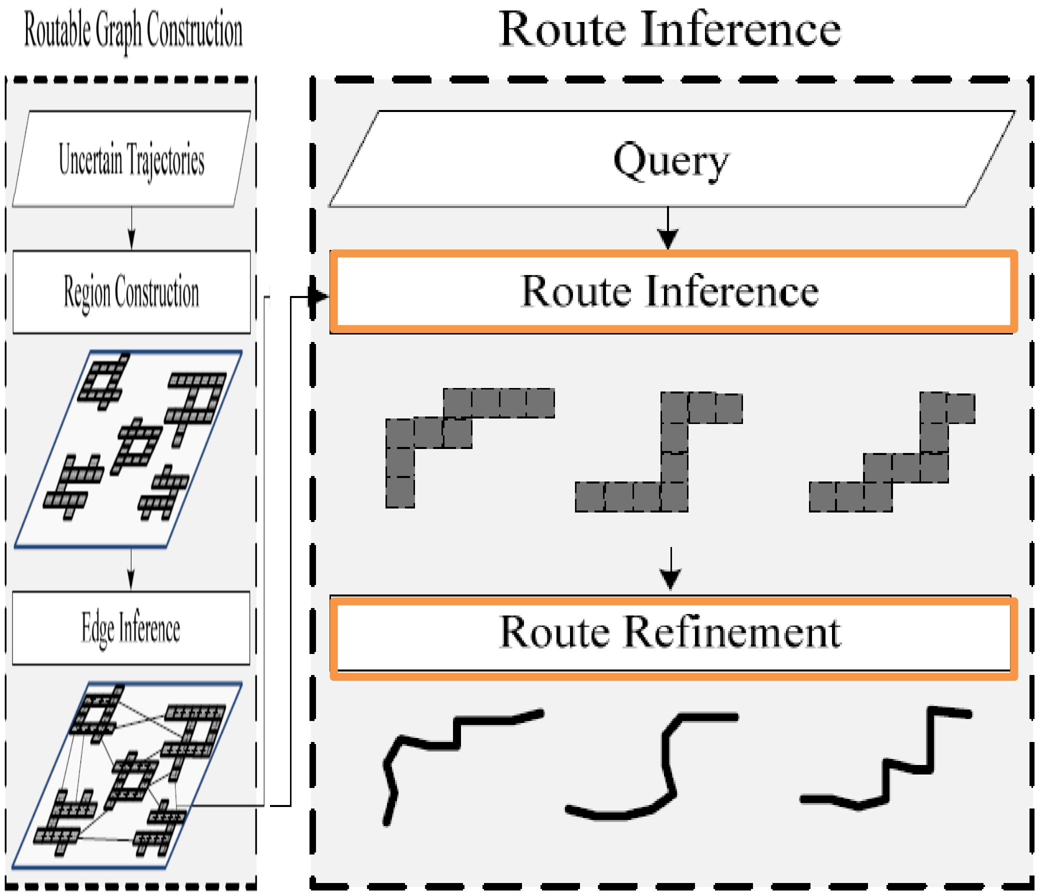
Step 1: Build edges between two cells in different regions by trajectories

Step 2: Replace the redundant edges by constructed edges

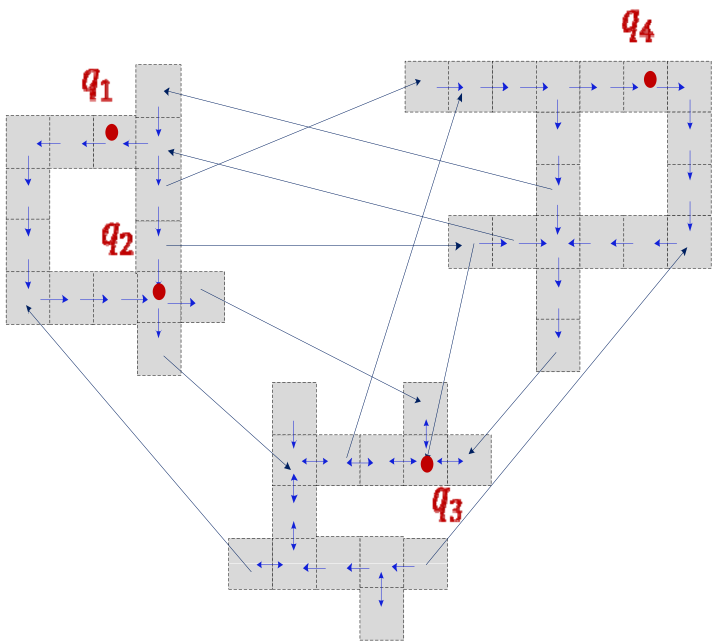
- Redundant edge
  - Exist at least one edge E_2 s.t. E_2' support > E_1's support
  - The travel time of the path $p_1 \rightarrow E_2 \rightarrow p_2$ and the travel time of edge E_1 are similar
- Propagate weights to multiple edges



Route Inference (1/2)



- Local route search
- Global route search



Route Inference (2/2)

- **Route score**
  - Given a graph $G = (V, E)$, a route $\wp: \wp_1 \rightarrow \wp_2 \rightarrow \cdots \rightarrow \wp_m$, the score of the route is

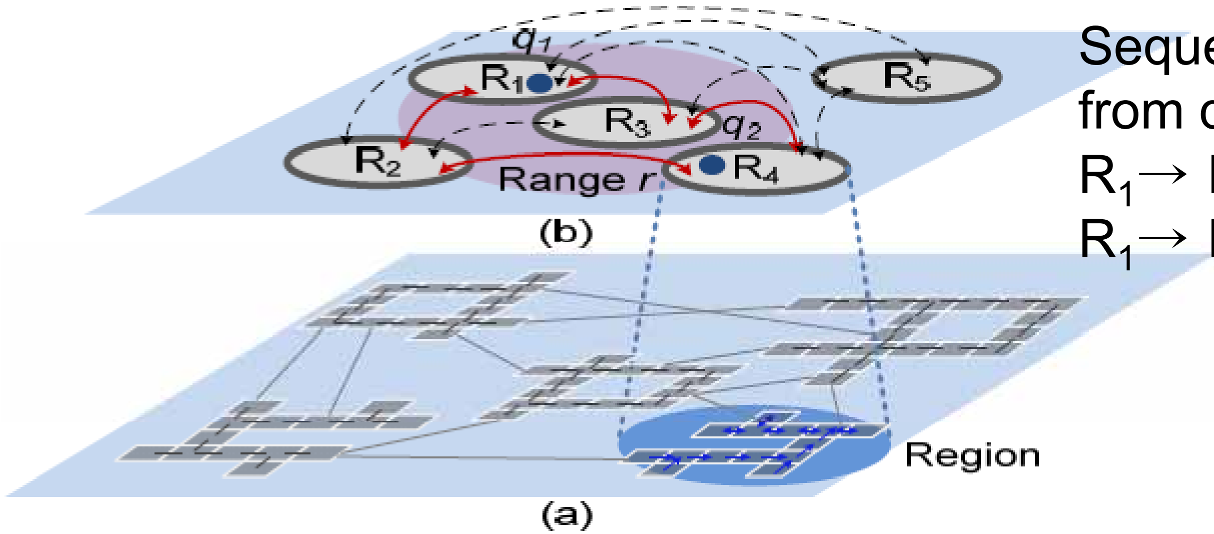
$$f(\wp) = \sum_{i=1}^{m} \rho(\wp_i)$$

where $\wp_i: g_{i_1} \longrightarrow g_{i_2} \longrightarrow \cdots \longrightarrow g_{i_j}$

and $\rho(\wp_i) = \frac{1}{j-1} | \cup_{k=1}^{j-1} \{tra | g_{i_k} \rightarrow g_{i_{k+1}} \in tra\} |$

Local Route Search

- **Goal**
  - Top K local routes between two consecutive geo-locations q_i, q_{i+1}
- **Approach**
  - Determine qualified visiting sequences of regions by travel times
  - A*-like routing algorithm
    - $\hat{f}(\wp) = \rho(\wp_c) + h(\wp_f)$ where a route $\wp: \wp_c \rightarrow \wp_f$



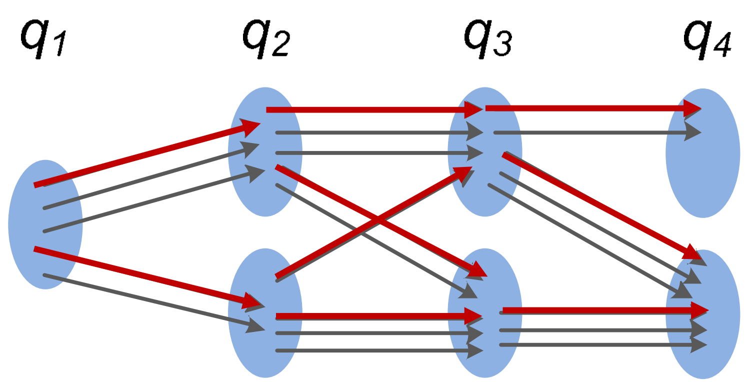
Sequences of Regions from q_1 to q_2:

$R_1 \rightarrow R_2 \rightarrow R_4$

$R_1 \rightarrow R_3 \rightarrow R_4$

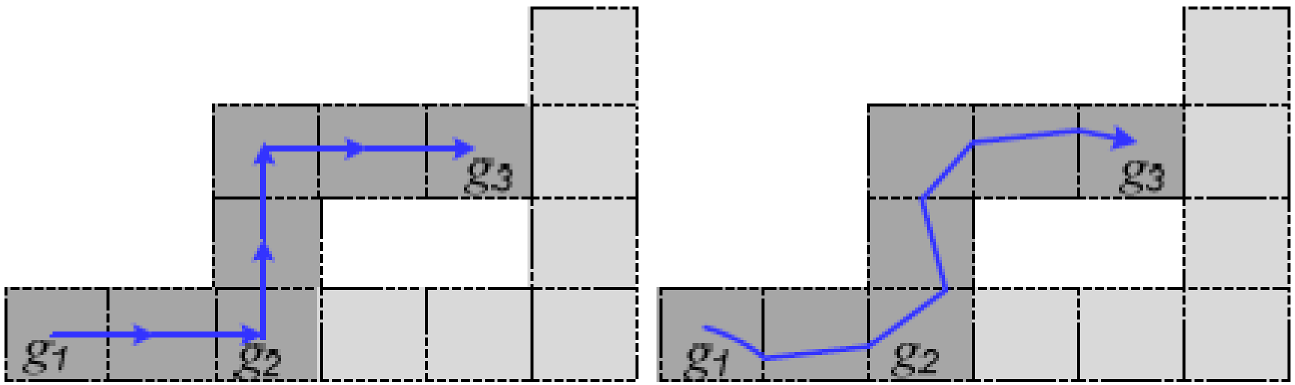
Global Route Search

- **Input**
  - Local routes between any two consecutive geo-locations
- **Output**
  - Top K global routes
- **Branch-and-bound search approach**
  - E.g., Top 1 global route



Route Refinement

- **Input**
  - Top K global routes: sequences of cells
- **Output**
  - Top K routes: sequences of segments
- **Approach**
  - Select GPS logs for each grid
  - Adopt linear regression to derive regression lines



Experiments

- **Real dataset**
  - Check-in records in Manhattan: 6,600 trajectories
  - GPS logs in Beijing: 15,000 trajectories
- **Effectiveness evaluation**
  - Inferred routes
    - Error:
      - T: $\mathrm{NDTW}(\mathrm{T},\mathrm{T}') = \mathrm{Avg}_{p_k \in \mathrm{T}} \min_{tra \in \mathrm{T}'} \mathrm{NDTW}(p_k, tra)$
      - T': top K trajectories (ground truth)
- **Efficiency evaluation**
  - Query time
- **Competitor**
  - MPR [Chen et al. ICDE'11]

Visualization in Manhattan

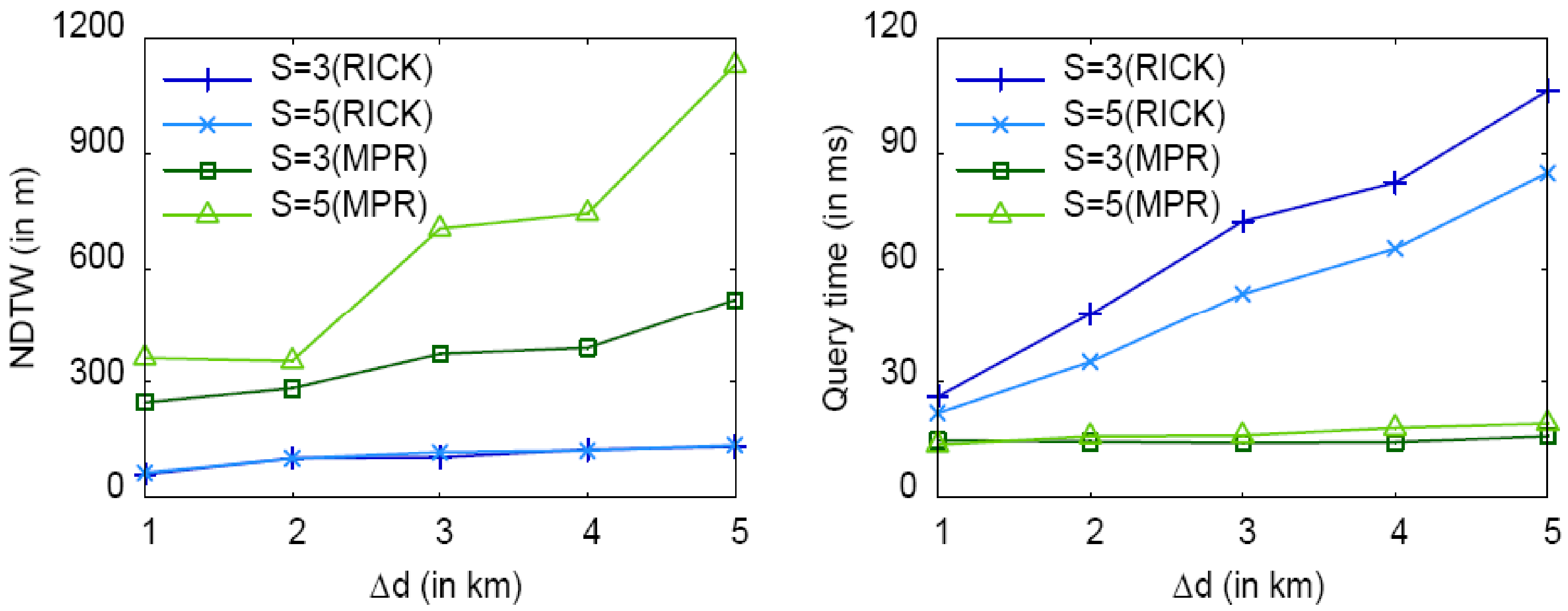
- Cell length: 500 m
- Minimum connection support: 3
- Temporal constraint: 0.2
- Time span Δt: 40 minutes

Routable Graph

Top 1 Popular Route

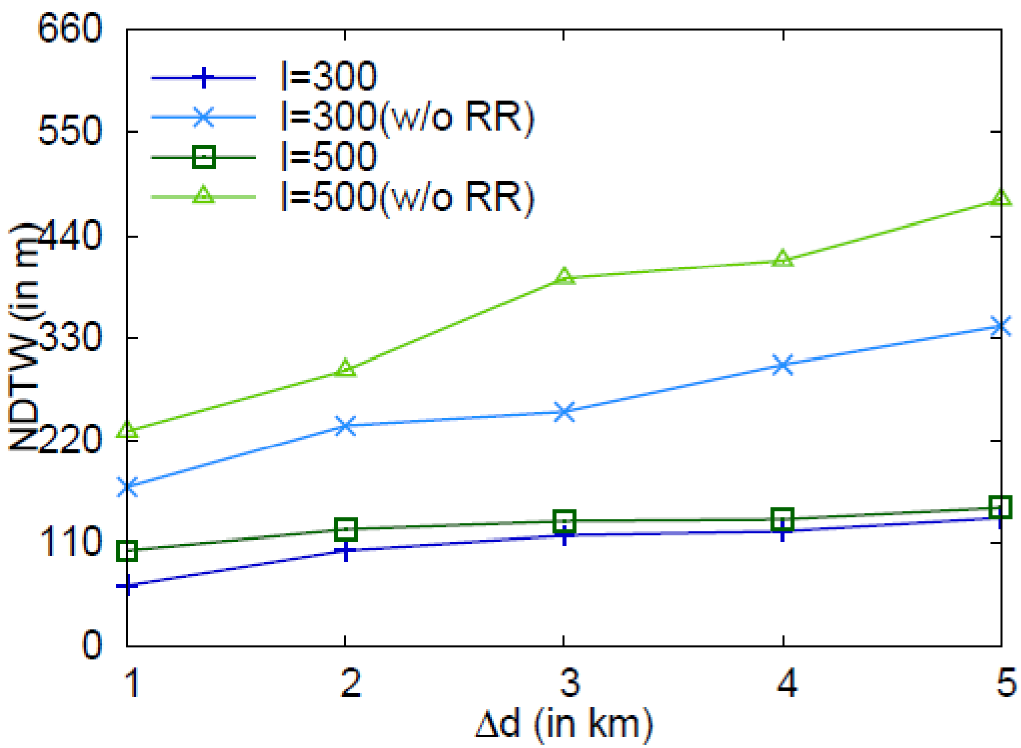
Performance Comparison

- **Competitor: MPR** [Chen et al. ICDE'11]
- **Parameters**
  - |q|:2, K:1, cell length: 300 m
- **Factors**
  - sampling rate (S), query distance (Δd)



Effectiveness of Route Refinement

- **Parameters**
  - K:1
  - |q|: 2



Summary

- **Developed a route inference framework without the aid of road networks**
- **Proposed a routable graph by exploring spatio-temporal correlations among uncertain trajectories**
- **Developed a routing algorithm to construct the top K popular routes**

Conclusion

- **Proposed a novel trajectory search framework for trip planning**
  - Search pattern-aware travel routes from historical trajectories with considering a user-preference of depth/breadth
  - Developed an efficient bounded trajectory search algorithm

- **Proposed a route inference framework without the aid of road network information**
  - Infer popular routes from uncertain trajectories
  - Developed an effective approach to construct a routable graph
  - Developed efficient and effective approaches to infer popular routes