Route Planning and Visualization Using Geo-Social Media Data

Hsun-Ping Hsieh¹, Thomas Sandholm², and Cheng-Te Li³

¹ National Taiwan University, Taiwan  
² HP Labs, Palo Alto, CA, USA  
³ Academia Sinica, Taiwan

ICWSM 2014 Tutorial
Geo-Social Media

- GPS-equipped mobile devices

- Location-acquisition services

- Foursquare
- Yelp
- Flickr
- Facebook Places
- Brightkite
- Gowalla
This Tutorial: Planning and Visualizing Geo-Social Media Data

• **Goal:** introduce methods and tools to exploit and present geo-social media data

• **Tutorial has two parts**
  
  – **Part 1: Route Planning using Geo-Social Trajectories**
    
    (60 mins) • Provide a broad summary of a series of recent advances on the route planning problem
    
    – How to construct a preferable trip route for users?
    – How to deal with data uncertainty in geo trajectories?

  – **Part 2: Programming Geo Data Visualizations**
    
    (60 mins) • Give a technical introduction and practical advice on how to use various tools for visualizing geo-social data
    
    – How to position markers on a map in a scalable way?
    – How to produce beautiful heatmaps and create interactive maps?
Part I: Route Planning Using Geo-Social Trajectories
About This Route Planning Part

• It is a about one hour introduction on route planning
  – We do not anticipate to cover everything relevant to this topic
  – We will highlight the trend, categorize different types of research problems and approaches

• The tutorial slides can be downloaded via:
  – http://mslab.csie.ntu.edu.tw/icwsm14tut/
Route Planning: Outline

• **Overview of Route Planning** (10 mins)
  – Background and Introduction
  – Geo-social Trajectory Data: Categories, Challenges
  – Route Planning: Query, Approaches

• **Route Planning Using GPS Trajectories** (20 mins)
  – Graph Search
  – Pattern Mining/Matching
  – Prediction/Recommendation

• **Route Planning Using Uncertain Trajectories** (20 mins)
  – Deal with Data Uncertainty
  – On Check-in Data and Geo-tagged Photos

• **Location Recommendation** (5 mins)

• **Conclusions and Future Directions** (5 mins)
Geo-Social Trajectory

• Trajectory
  – A sequence of location data points with
    • Latitude-longitude records
    • Time stamps

  – Represent the spatial-temporal human activities

<table>
<thead>
<tr>
<th>ID</th>
<th>Timestamp</th>
<th>Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>“Peter”</td>
<td>2010-04-02 13:12</td>
<td>37.5, -122.5</td>
</tr>
<tr>
<td>“Peter”</td>
<td>2010-04-02 15:22</td>
<td>37.2, -123.5</td>
</tr>
<tr>
<td>…</td>
<td>…</td>
<td>…</td>
</tr>
</tbody>
</table>

Human movement  Taxi movement  Animal movement
Sources of Geo-Social Trajectory Data

• GPS Devices
  – In-car GPS
  – Personal GPS logger

• Location-based services + mobile phone
  – Check-in actions and records
  – E.g. Facebook, Foursquare, Twitter

• Digital Camera
  – Geo-tagged photos
  – E.g. Flickr, Instagram, Panoramio

Such user mobility records reveal how people travel around an area!
Route Planning

• Given
  – A set of historical trajectory data
  – Route Query: depict the user needs
    • Info. about the desired places along the constructed routes

• Goal: construct/recommend
  – A list of preferable routes
    • Satisfy the query requirements as much as possible
## Route Planning: Examples

<table>
<thead>
<tr>
<th>Query</th>
<th>Preferable Routes</th>
<th>Illustration</th>
</tr>
</thead>
</table>
| 1) A set of locations  
2) Time span of route | A route pass through these locations within time span | ![Illustration 1] |
| 1) A source loc.  
2) A destination loc.  
3) A number of route length | A route starting from source and arrive at the destination, with length satisfied | ![Illustration 2] |
| 1) A city or an area  
2) A set of labels of interests | A route in such area, which contains locations possessing such labels | ![Illustration 3] |
## Category of Geo-Social Trajectory Data

<table>
<thead>
<tr>
<th></th>
<th>GPS Trajectory</th>
<th>Uncertain Trajectory</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Data Source</strong></td>
<td>GPS recorders, in-car GPS tracer</td>
<td>Check-in actions in LBS Meta info in Geo-tagged Photos</td>
</tr>
<tr>
<td><strong>Data Points</strong></td>
<td>Simply Geographical Coordinates Points</td>
<td>Point of Interests (POI) e.g. landmark, restaurant</td>
</tr>
<tr>
<td><strong>Property</strong></td>
<td>Smoothly and continuously record every fixed distance/time period</td>
<td>Discretely and sparsely performed by users in LBS and taken by cameras</td>
</tr>
<tr>
<td><strong>Sample Rate</strong></td>
<td>High</td>
<td>Low</td>
</tr>
</tbody>
</table>
Challenges on Route Planning

• GPS Trajectory
  – How to find meaningful and/or popular places?
  – How to tackle efficiently million-scale geo-data points for query processing?

• Uncertain Trajectory
  – Do not detail the sequences of movement
  – Raise uncertainty between consecutive points

Check-in records

Geo-tagged Photos

Time Square

Rockefeller Center

Grand Central Station

Apple Store
Route Planning: Overview

Route Planning Systems

Planning Approach
- Graph Search
- Pattern Mining
- Inference & Learning

Next Location
- Location Prediction
- Location Recommendation

Preprocessing
- Indexing & Simplification
- Clustering & Classification

Geo-Social Data
- GPS Trajectories
- Location Check-in Data
- Geo-tagged Photos

Uncertain Trajectories

Location Query

Context Query

Social Query

Routes or Next Location
Route Query: Location Query

- **Location Query**
  - **Required Locations**: needed to be pass thru
  - **Visiting Order**: order of required locations
  - **Geo-Distance**: geographical range or the tolerable distance between locations

![Diagram of Route Query]

- 4km
- 2km
Route Query: Context Query

- **Context Query**
  - **Visiting/Stay Time**: whether the visiting time of a location is *proper* or the staying time of a location
  - **Transit Time**: the time for transiting between locations
  - **Travel Duration**: the total traveling duration in the route
  - **Financial Cost**: the budget of a route
  - **Top-K retrieval**: whether or not to return top-k preferable routes

- $t_{\text{vis}} = 4\text{pm}$
- $t_{\text{sta}} = 2\text{hr}$
- $t_{\text{tan}} = 1\text{hr}$
- $t_{\text{dur}} = 10\text{hr}$
- $c_{1} = 3\text{USD}$
- $c_{2} = 10\text{USD}$
- $c_{3} = 1\text{USD}$
- $c_{4} = 0\text{USD}$
- $c_{\text{total}} = 14\text{USD} < c_{\text{budget}} = 15\text{USD}$
Route Query: Social Query

- Social Query
  - Popularity of locations
  - User Preference: whether or not to consider user’s past visiting history
  - Group or Social factor: group trips or the locations that friends had ever visited
  - Activity Labels: specifying the labels or types of locations in the route

Query = \{theater, restaurant, park\}

<table>
<thead>
<tr>
<th>Grp Mem.</th>
<th>List of desired locations</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>{a, b, c, d, e}</td>
</tr>
<tr>
<td>B</td>
<td>{b, f, g}</td>
</tr>
<tr>
<td>C</td>
<td>{c, e, f, g}</td>
</tr>
<tr>
<td>D</td>
<td>{a, c, d, f}</td>
</tr>
</tbody>
</table>
Approach Overview: Graph Search

• **Graph** Construction G

<table>
<thead>
<tr>
<th></th>
<th>Trajectory Data</th>
<th>Road Net</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nodes</td>
<td>Locations</td>
<td>Road Segments</td>
</tr>
<tr>
<td>Edges</td>
<td>Traversal</td>
<td>Intersection</td>
</tr>
<tr>
<td>Node Weights</td>
<td>Popularity / Satisfaction / Traffic</td>
<td></td>
</tr>
<tr>
<td>Edge Weight</td>
<td>Transition Probability / Frequency</td>
<td></td>
</tr>
</tbody>
</table>

• Design an **objective function** $f(r)$ based on query, e.g.
  – E.g. visiting/transition popularity, label cover
  – With some **constraints**, e.g. travel time, financial cost
• Find a route/path $r$ in $G$ such that $f(r)$ is optimized
Approach Overview: Pattern Matching/Mining

- Each trajectory = a **sequence** of geo-points / locations

- **Pattern Mining**
  - Mining the **frequent subsequences** constrained by the query requirements
  - Subsequence **Pruning**: keep closed ones (to save complexity)
  - Subsequence **Merge**: from local route to global route

- **Pattern Matching**
  - Find individuals **with similar behaviors of movements**
  - **Nearest-nearest query** processing (given some locations)
Approach Overview: Recommendation/Prediction

• Given an existing sub-route, successively predict/recommend the next locations
  – Till the user requirement is satisfied
    • E.g. Route Length $k$, Travel Time. Arrive the destination

• Select the next locations
  – Unsupervised method
    • Location info. E.g. popularity, density, incoming flow
    • Estimate the probability $P(\text{candidateLoc} | \text{curSubRoute})$
  – Supervised method
    • Choose a set of candidate locations
    • Extract route/Location-aware features
    • Apply supervised learning methods e.g. SVM
# Approaches on Different Trajectories

with the corresponding studies described in this tutorial

<table>
<thead>
<tr>
<th>GPS Trajectory</th>
<th>Uncertain Trajectory</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Graph Search</strong></td>
<td>Deal with Uncertainty &amp; Route Search</td>
</tr>
<tr>
<td>[Chen’11] [Zheng’11]</td>
<td>[Zheng’12] [Wei’12] [Hsieh’14]</td>
</tr>
<tr>
<td><strong>Pattern Mining</strong></td>
<td>Trip Planning Using Geo-tagged Photos</td>
</tr>
<tr>
<td>[Tang’11] [Tang’12]</td>
<td>[Lu’10] [Cao’12]</td>
</tr>
<tr>
<td><strong>Prediction/Recommend</strong></td>
<td>Loc. Predict/Recommend</td>
</tr>
<tr>
<td>[Jeung’08] [Xue’13]</td>
<td>[Noulas’12] [Ye’11] [Yuan’13] [Karamshuk’13]</td>
</tr>
</tbody>
</table>
# Queries of Targeted Studies in This Tutorial

<table>
<thead>
<tr>
<th>GSP Trajectory Data</th>
<th>Location Query</th>
<th>Context Query</th>
<th>Social Query</th>
</tr>
</thead>
<tbody>
<tr>
<td>[Tang’13]</td>
<td>■</td>
<td>■</td>
<td>■</td>
</tr>
<tr>
<td>[Chen’11]</td>
<td>■</td>
<td>■</td>
<td>■</td>
</tr>
<tr>
<td>[Zheng’11]</td>
<td>■</td>
<td>■</td>
<td>■</td>
</tr>
<tr>
<td>[Tang’11]</td>
<td>■</td>
<td>■</td>
<td>■</td>
</tr>
<tr>
<td>[Jeung’08]</td>
<td>■</td>
<td>■</td>
<td>■</td>
</tr>
<tr>
<td>[Xue’13]</td>
<td>■</td>
<td>■</td>
<td>■</td>
</tr>
<tr>
<td>[Wei’12]</td>
<td>■</td>
<td>■</td>
<td>■</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Uncertain Trajectory Data</th>
<th>Location Query</th>
<th>Context Query</th>
<th>Social Query</th>
</tr>
</thead>
<tbody>
<tr>
<td>[Hsieh’14]</td>
<td>■</td>
<td>■</td>
<td>■</td>
</tr>
<tr>
<td>[Zheng’12]</td>
<td>■</td>
<td>■</td>
<td>■</td>
</tr>
<tr>
<td>[Cao’12]</td>
<td>■</td>
<td>■</td>
<td>■</td>
</tr>
<tr>
<td>[Lu’10]</td>
<td>■</td>
<td>■</td>
<td>■</td>
</tr>
</tbody>
</table>
Route Planning: Outline

• Overview of Route Planning (10 mins)
  – Background and Introduction
  – Geo-social Trajectory Data: Categories, Challenges
  – Route Planning: Query, Approaches

• Route Planning Using GPS Trajectories (25 mins)
  – Graph Search
  – Pattern Mining/Matching
  – Prediction/Recommendation

• Route Planning Using Uncertain Trajectories (25 mins)
  – Deal with Data Uncertainty
  – On Check-in Data and Geo-tagged Photos

• Location Recommendation (10 mins)

• Conclusions and Future Directions (5 mins)
Route Planning Using GPS Trajectories

- **GPS Trajectory, Graph Search**

- **GPS Trajectory, Pattern Mining/Matching**

- **GPS Trajectory, Prediction/Inference**
  - A. Y. Xue et al. Destination Prediction by Sub-Trajectory Synthesis and Privacy Protection Against Such Prediction, In ICDE 2013
Discovering Popular Routes [Chen’11]

• Given a source and a destination location, find the most popular route in between

Count the number of trajectories on different paths connecting the two locations

However,
Not easy to find such well-divided groups

There could be no trajectory connecting two locations at all
Discovering Popular Routes

• Construct a transfer network
  – Node: an intersection of trajectories or just the end locations
  – Edge: at least one contiguous trajectory from A to B without any other transfer nodes in between
Discovering Popular Routes

- Turning Probability
  \[ Pr_d(n_i \rightarrow n_j) = \frac{\sum_{\text{traj} \in (n_i, n_j)} \text{func}(\text{traj}, d)}{\sum_{\text{traj} \in \text{all outgoing edges}} \text{func}(\text{traj}, d)} \]
  \[ \text{func}(\text{traj}, d) = \exp(-\text{dist}_s(\text{traj}, d)) \]

- Popularity indicator

- Route Popularity
  \[ n_i.\text{popularity}(d) = Pr^t(n_i \rightarrow d) \]
  \[ \rho(R) = \prod_{j=1}^{i} n_j.\text{popularity}(d) \]

- Exploit the Dijkstra’s algorithm to find the route with maximum popularity
  – Expanding in a BFS way
Learning Travel Recommendations [Zheng’11]

- User might want to know the classical travel routes in a unfamiliar city
- How to model the interestingness of locations?
- How to exploit location interestingness for recommend routes?
  - A three-step solution is proposed

**Step 1: Modeling Human Location History**

\[ LocH = (s_1 \xrightarrow{\Delta t_1} s_2 \xrightarrow{\Delta t_2} \ldots \xrightarrow{\Delta t_{n-1}} s_n); \Delta t_i = s_{i+1} \cdot arvT - s_i \cdot levT \]
1. Stay point detection

2. Hierarchical clustering

3. Graph Building

Shared Hierarchical Framework

- Stands for a stay point $S$
- Stands for a stay point cluster $c_{ij}$

GPS Logs of User 1
GPS Logs of User 2
GPS Logs of User $i$
GPS Logs of User $i+1$
GPS Logs of User $n-1$
GPS Logs of User $n$
Step 2: HITS-based Interestingness

- Mutual reinforcement relationship
  - A user with rich travel knowledge are more likely to visit more interesting locations
  - A interesting location would be accessed by many users with rich travel knowledge

- A HITS-based inference model
  - Users are hub nodes
  - Locations are authority nodes
  - Topic is the geo-region
Step 3: Detect Interesting Travel Route

• Three factors determining the score of a sequence
  – Travel experiences (hub scores) of the users taking the sequence
  – The location interests (authority scores) weighted by
  – The probability that people would take a specific sequence

The classical score of sequence $A \rightarrow C$:

$$S_{AC} = 5 \times \left( \frac{5}{7} \times a_A + \frac{5}{8} a_C \right) + \sum_{u_k \in U_{AC}} h^k.$$ 

$a_A$: Authority score of location A
$a_C$: Authority score of location C
$h^k$: User k’s hub score

\[ \begin{align*}
\widehat{S}_A &= \frac{5}{7} \times a_A + \frac{5}{8} a_C + \sum_{u_k \in U_{AC}} h^k, \\
\widehat{S}_C &= \frac{5}{7} \times a_A + \frac{5}{8} a_C + \sum_{u_k \in U_{AC}} h^k.
\end{align*} \]
Route Planning Using GPS Trajectories

- **GPS Trajectory, Graph Search**

- **GPS Trajectory, Pattern Mining/Matching**
  - L. A. Tang et al. A Framework of Traveling Companion Discovery on Trajectory Data Streams. In *ACM TIST 2013*.

- **GPS Trajectory, Prediction/Inference**
k-Nearest Neighboring Trajectory (k-NNT) Query [Tang’11]

- Query the top $k$ trajectories with the minimum aggregated distance to the given locations
- The trajectories may not exactly pass those locations
The Aggregate Distance in $k$-NNT

• Find out the closest matching pair from a trajectory to each query point and sum up the distance

\[
\begin{align*}
\text{dist}(R_1, q_1) &= \text{dist}(p_{1,2}, q_1) = 20 \text{ m} \\
\text{dist}(R_1, q_2) &= \text{dist}(p_{1,3}, q_2) = 50 \text{ m} \\
\text{dist}(R_1, q_3) &= \text{dist}(p_{1,5}, q_3) = 15 \text{ m} \\
\text{dist}(R_1, Q) &= \sum \text{dist}(R_1, q_i) = 85 \text{ m} \\
\text{dist}(R_2, q_1) &= \text{dist}(p_{2,3}, q_1) = 30 \text{ m} \\
\text{dist}(R_2, q_2) &= \text{dist}(p_{2,4}, q_2) = 5 \text{ m} \\
\text{dist}(R_2, q_3) &= \text{dist}(p_{2,6}, q_3) = 40 \text{ m} \\
\text{dist}(R_2, Q) &= \sum \text{dist}(R_2, q_i) = 75 \text{ m}
\end{align*}
\]
Candidate Generation and Verification

- Step 1: find k-NN points using best-first-based local heap
- Step 2: generate the candidate trajectories by global heap
Search based on the global heap

Candidate Set

Global Heap

Individual Heaps

\[ <p_{1,2}, q_1> \]

\[ <p_{1,4}, q_2> \]

\[ <p_{1,6}, q_3> \]
Search based on the global heap

**Candidate Set**

**Global Heap**

\[<p_{1,4}, q_2>\]

\[<p_{1,6}, q_3>\]

\[<p_{1,2}, q_1>\]

**Individual Heaps**

\[h_1\]

\[h_2\]

\[h_3\]
Search based on the global heap

Candidate Set

Global Heap

Individual Heaps

$R_1$: $<p_{1,4}, q_2>$ (Partial Match)

$h_1$ $<p_{1,6}, q_3>$

$h_2$ $<p_{5,5}, q_2>$

$h_3$ $<p_{1,2}, q_1>$

......

......
Search based on the global heap

Candidate Set

Global Heap

Individual Heaps

\( R_1: \langle p_{1,4}, q_2 \rangle \langle p_{1,6}, q_3 \rangle \) (Partial Match)

\( R_5: \) (Partial Match)
Search based on the global heap

Candidate Set

Global Heap

\(<p_{1,2}, q_1>, <p_{4,4}, q_2>, <p_{1,5}, q_3>\>

Stop criteria: when there is \(k\) full-matching candidates – Property 1: The candidate set is complete if \(G\) has popped out \(k\) full-matching candidates (In this example \(k=1\))
Candidate Verification

- The full-matching candidate may **not** be the final $k$-NNT
- The system has to retrieve the partial-matching trajectories ($R_4$ and $R_5$) to compute their aggregate distance (I/O cost)
- Question: can we compute a **lower-bound** for $R_4$ and $R_5$ without retrieving their details?

Candidate Set (unit: m)

- $R_1$: $<p_{1,2}, q_1>$: 35, $<p_{1,4}, q_2>$: 27, $<p_{1,6}, q_3>$: 33.
- $R_4$: $<p_{4,5}, q_3>$: 32.
- $R_5$: $<p_{5,5}, q_2>$: 15.

Global Heap (unit: m)

- $<p_{1,5}, q_3>$: 37, $<p_{4,4}, q_2>$: 40, $<p_{1,1}, q_1>$: 42

$$\delta = \text{dist}(R_1, Q) = 95 \text{ m}$$

$$\text{LB}(R_4, Q) = 42 + 40 + 32 = 114 \text{ m} > \delta \text{ (pruned)}$$

$$\text{LB}(R_5, Q) = 42 + 15 + 37 = 94 \text{ m} < \delta$$
Discovery of Traveling Companions [Tang’12]

- Study the partnership in trajectory streams
  - Discover the group of objects that move together (with similar patterns of movements), i.e., traveling companions
  - E.g. migration path, driving direction, travel paths
- We can recommend routes to from your companion based on historical traveling trajs

Size threshold = 4
Duration threshold = 4 snapshots

\{o_1, o_2, o_3, o_4\} is the traveling companion
Traveling Companion Discovery

• Let $\delta_s = \text{size threshold}$ and $\delta_t = \text{duration threshold}$, a group of objects $q$ is traveling companion if:
  – The members of $q$ are density connected by themselves for a period $t$ where $t \geq \delta_t$
  – $\text{size}(q) \geq \delta_s$

• Clustering-and-Intersection Approach
  – Clustering the objects in each snapshot
  – Intersecting the clusters to generate companion candidates, if the candidates meet the size and time requirements, output them as companion
Clustering-and-Intersection ($\delta_t=40m$, $\delta_s=4$)

Once we found $r_1$'s objects in $c_1$, stop the intersection; do not add the un-closed candidates.

$r_1 = \{o_1, o_2, o_3, o_4\}, 10 m$
$r_2 = \{o_6, o_7, o_8, o_9, o_{10}\}, 10 m$

$r_1 = \{o_1, o_2, o_3, o_4\}, 20 m$
$r_2 = \{o_6, o_7, o_8, o_9, o_{10}\}, 20 m$

$r_3 = \{o_1, o_2, o_3, o_4, o_5, o_6, o_7, o_8, o_9, o_{10}\}, 10 m$

$R$'s size: 9
Intersect: 0

$r_1 = \{o_1, o_2, o_3, o_4\}, 30 m$
$r_2 = \{o_8, o_9, o_{10}\}, 30 m$

$r_3 = \{o_1, o_2, o_3, o_4, o_5\}, 20 m$

$R$'s size: 23
Intersect: 11

$R$'s size: 14
Intersect: 29
Route Planning Using GPS Trajectories

• GPS Trajectory, Graph Search

• GPS Trajectory, Pattern Mining/Matching

• GPS Trajectory, Prediction/Inference
  – A. Y. Xue et al. Destination Prediction by Sub-Trajectory Synthesis and Privacy Protection Against Such Prediction, In *ICDE* 2013
Sub-Trajectory Synthesis [Xue’13]

- **Goal:** predict the future routes based on
  - The current location $c$
  - The trajectory from the start location $s$ to $c$

- **Example**
  - $T_1 = \{l_1, l_2, l_5, l_6, l_9\}$
  - $T_2 = \{l_6, l_3, l_2\}$
  - $T_3 = \{l_4, l_5, l_8\}$
  - $T_4 = \{l_9, l_8, l_7\}$
  - $T_5 = \{l_1, l_4, l_7\}$

A user travels from $l_1$ to $l_4$  
=> Predict the future traj $l_7$ and $l_8$
Query traj $\{l_1, l_2, l_3\}$  
=> No predicted traj due to lack of training data
Sub-Trajectory Synthesis

• Step 1: Partition and group points into grid cells
• Step 2: Decompose historical trajs into sub-trajectories
• Step 3: Construct transition matrix $M$

Deal with data sparsity problem
Sub-Trajectory Synthesis

• Step 4: Synthesis
  – Starting from $n_1$, the probability of traveling to $n_9$?
  – Shortest Path is 4: $p_{1 \rightarrow 9} = M^4_{1,9}$
  – $M^4$: transition between cells with distance 4

• Consider **detour** (within 1.2 times shortest path)
  – Users may travel either distance 4 or 5 ($\text{ceil}(4 \times 1.2)$) to reach $n_9$
    
    
    

    

    $p_{1 \rightarrow 9} = M^4_{1,9} + M^5_{1,9}$
Sub-Trajectory Synthesis

• Given a user’s current route $T^p = \{n_1, n_4, n_5\}$

• The probability of $n_9$:

\[
P(n_9 | T^p) = P(n_9 | n_1, n_4, n_5) = \frac{\text{Bayes’ Rule}}{P(T^p)} = \frac{p_{5\rightarrow 9}}{p_{1\rightarrow 9}} \cdot P(T^p)
\]

\[
= \left( \frac{M^2_{5,9}}{M^4_{1,9} + M^5_{1,9}} \right) \times (p_{14} \cdot p_{15})
\]

• Summary

  – Training: build Markov model and matrix multiplications, to prepare probs
  – Prediction: compute the prob: $P(n_k | T^p)$
  – Consecutively predict next locations as the route recommendation
Hybrid Prediction Model [Jeung’08]

- Given a sequence of movement
  - \(<(l_1,t_1), (l_2,t_2), ..., (l_n,t_n)>\)

- What are the locations at future times \(t_F\)
  - \(t_F\): just the next timestamp to \(t_n\) **Distant Time Query**
  - \(t_F\): any future timestamp **Non-distant Time Query**

- The Prediction Model
  - Detection of frequent regions
  - Discovery of trajectory patterns
  - Ranking-based Prediction algorithms
Approach to Predict Future Trajectories

• **Step 1: Frequent Regions**
  - Decomposing a whole trajectory into sub-trajectories
  - Grouping positions at time offset $k$ in each sub-traj
  - Applying a clustering method (e.g. DBSCAN)

• **Step 2: Trajectory Patterns**

\[
R_{t_1}^{j_1}, R_{t_2}^{j_2}, ..., R_{t_m}^{j_m} \rightarrow c \rightarrow R_{t_n}^{j_n} \ (t_1 < t_2 < ... < t_m < t_n)
\]
Query Processing: Near Future

- **Step 3: Forward algorithm** \( t_q - t_c \leq d \)
  - An object is likely to follow the trend of recent movements

- **Candidate filtering**
  \[
  R_{k_1}^{j_1}, R_{k_2}^{j_2}, \ldots, R_{k_m}^{j_m} \xrightarrow{c} R_{k_n}^{j_n}
  \]

- **Ranking candidates**
  - \( Sim(\text{the current movements, the premise of each pattern}) \times \text{confidence} \)
  - Current movements are more important
Query Processing: Distant Future

- **Step 3: Backward algorithm** $t_q - t_c > d$
  - The recent movements are not so important for prediction

- **Candidate filtering**
  
  \[ R_{k_1}^j, R_{k_2}^j, \ldots, R_{k_m}^j \xrightarrow{c} R_{k_n}^j \]

- **Ranking candidates**
  
  - \(( Sim(\text{premise}) \times \text{penalty} + Sim(\text{consequence}) ) \times \text{confidence} \)
  
  - As the query time $\uparrow$, the importance of recent movements $\downarrow$
Route Planning: Outline

• Overview of Route Planning
  – Background and Introduction
  – Geo-social Trajectory Data: Categories, Challenges
  – Route Planning: Query, Approaches

• Route Planning Using GPS Trajectories
  – Graph Search
  – Pattern Mining/Matching
  – Prediction/Recommendation

• Route Planning Using Uncertain Trajectories
  – Deal with Data Uncertainty
  – On Check-in Data and Geo-tagged Photos

• Location Recommendation

• Conclusions and Future Directions
Route Planning Using Uncertain Trajectories

- **Deal with Data Uncertainty (Check-in Data)**

- **Check-in Data & Geo-tagged Photos**
Construct Popular Route from Uncertain Trajectories [Wei’12]

• Using location check-in data

• Given
  – A sequence of locations
  – A time span

• Find
  – The top-k popular routes which sequentially pass through the locations within the specified time span

• The Proposed
  – Routable Graph: deal with data uncertainty (offline)
  – Route Inference: construct the top-k popular routes (online)
 ROUTABLE GRAPH

• Space Partition: divide a space into non-overlapping cells with a given cell length

• Trajectory Indexing

• Region
  – A connected geographical area
  – Merge connected cells to form a region

• Observation
  – Area A1 and area A2 are possibly connected if tra1 and tra2 follow the same route
    • A1 and A2 are spatially close
    • Similar travel times
      – A1 → A3: 10 mins
      – A2 → A3: 11 mins
Routeable Graph

- Spatio-temporally correlated relation between trajs
  - Spatially close

Region: Connected geographical area

Edges in each region

Edges between regions

Routingable Graph
Route Search

General Idea:

Global Route

Local Route
Reducing Uncertainty of Low-Sampling Rate Trajectories [Zheng’12]

- Can we reduce the uncertainty caused by low-sampling-rate before the trajectories undergo further processing?
- Can we estimate its original route from the samples?
- Basic: exploit the historical trajectories collectively
Reducing Uncertainty

• Problem Statement
  – Given: (a) a collection of historical trajectories, (b) a road network, (c) a user-specified low-sampling-rate trajectory
  – Find: a few possible routes of the query trajectory

• Route Inference
  – Step 1: Search for reference trajectories
    • Select relevant trajs helpful in inferring the route of query
  – Step 2: Local route inference
    • Infer the routes between consecutive samples of query
  – Step 3: Global route inference
    • Infer the whole routes by connecting the local routes
Step 1: Reference Trajectories

- Relevant Trajectory Search
  - Simple reference trajectory
  - Spliced reference trajectory
Step 2: Local Route Inference

- **Traverse Graph** based approach

  Use the \( k \) shortest paths of this graph as the candidate local possible route of the query

- **Nearest Neighbor** based approach

  Consider all the reference points in Euclidean space
  Find a continuous hops with shortest Euclidean distance between query points
  Recursively search for kNN of the current position and jump to one of the kNNs
Step 3: Global Route Inference

- Connect the candidate local routes between consecutive samples to form the global route.
- The quality of a global route:
  - \( f(R_i) \): the popularity of each local route.
  - \( g(R_i, R_{i+1}) \): the transition confidence function for the connections.
- Find the subset of global routes maximizing global score:
  - Solved by Dynamic Programming.
Route Planning Using Uncertain Trajectories

• Deal with Data Uncertainty (Check-in Data)

• Check-in Data & Geo-tagged Photos
10:00 AM
Current Location

Time Square
11:00 AM

Central Park

Madison Square Garden

SOHO

World Trade Center

Wall Street

MOMA

Empire State Building

16:00 PM

19:00 PM

21:00 PM

The Statue of Liberty
Proper Visiting Time of a Place [Hsieh’14]

- Given a source and/or destination location, and the current time, can we recommend a route, in which each location can be visited with a pleasant experience

- Pleasant visiting of places should consider visiting time, e.g.
  - People usually visit the Empire State Building from about 12:00 to the mid night (night view is popular)
  - People tend to visit the Madison Square Garden in the early evening for a basketball game
  - The proper time to visit the Central Park is during daytime
  - Time Square is preferred from afternoon to midnight.
Time-Sensitive Route [Hsieh’14]

- The goodness function of a route should consider:
  - **Popularity** of a place: visiting probability
  - **Visiting order** of places: N-gram
  - **Proper visiting time** of a place: Temporal Visiting Distribution (TVD)
  - **Proper transit time** between places: Duration Distribution (DD)

**Using Symmetric KL Divergence to measure the difference between $\mathcal{N}(\mu, \sigma^2)$ and TVD**

$$f(s) = \alpha \times \left( \frac{f_{\text{visit}}(s) + f_{\text{duration}}(s)}{2} \right) + (1 - \alpha) \times f_{\text{order}}(s)$$
Route Construction: 
**Guidance Search**

The *heuristic satisfaction function* $f^*(l)$ direct the search algorithm to move toward destination.

*Step 1.* For each candidate $l_c$, calculate $g(l_c) = f(l_s \rightarrow l_c)$.

*Step 2.* Choose the best $f(l_c \rightarrow l_d)$.

*Step 3.* Calculate the distance of $d(l_c \rightarrow l_d)$.

*Step 4.* $h(l) = \max_{(l \rightarrow l_d) \in S(l \rightarrow l_d)} \{\sqrt{f(l_c \rightarrow l_d)} \times (1 - d(l_c \rightarrow l_d))\}$

*Step 5.* $f^*(l) = (1 - \beta) \times g(l) + \beta \times h(l)$
Route Construction: 

*Guidance Search*

The *Backward Checking mechanism* find path which a higher satisfaction score

Keep track of all scores, \( f^* (l_s \rightarrow l_c) \)

When considering 3\(^{rd}\) location, the Guidance Search can go back to re-choose 2\(^{nd}\) location if we find there exists \( f^* (l_s \rightarrow l_c) \) which is larger than all \( f^* (l_s \rightarrow l_2 \rightarrow l_c) \)
Photo2Trip: Generating Routes from Geo-tagged Photos [Lu’10]

• Scenarios
  – “Want to have a one-day trip in an unfamiliar city, Beijing. Any route suggestion to visit famous places?”
  – “I am going to visit the Forbidden City in Beijing, with 3 hours. What’s the route within the palace?”

• Expected result
  – One-day trip in Beijing: 3 hours in Forbidden City → 2 hours in Tian An Men Square → 2 hours in Qian Men.
Geo-tagged Photos as Uncertain Trajectories

- **Incomplete/uncertain paths** within a destination
  - Tourists take photos discretely and upload a small portion
  - Collectively use **fragments** of photos from tourists & merge them

- Given a city, travel duration (and other user preference)
  - Find a route of destination locations **satisfying travel duration**
  - For each destination, find a travel **path** with a **stay time**

- **Photo2Trip**
  - Step 1: Destination Discovery
    - Geo Clustering, Cluster Size to filter
  - **Step 2: Internal Path Discovery**
  - Step 3: Trip Planning
    - Build Destination Graph, Dynamic Programming to find the optimal route
Internal Path Discovery

• 3 Steps from fragments to a path & estimate stay time

• Step 1: Estimating Path Score = Quality + Popularity
  – Path Quality = Path Density + Path Span
    • Path Density = # of photos per length
    • Path Span = max Euclidean distance among 2 photos of the path
  – Path Popularity = # of tourists involving in the path (# of fragments)
  – Use to filter out candidate paths

Path A: large path span, high path density
Path B: large path span, low path density
Path C: short path span, high path density
**Path Quality: A > B, C**
Internal Path Discovery (cont.)

- **Step 2: Fragment Merging**
  - Merge 2 fragments if the distance between anchor photos is small
    - **Anchor photos**: the closest photo pairs

- **Step 3: Path Discovery**
  - Examine every pair of fragments
    - Test if a pair is ok to merge
    - Select the one with the **highest path quality** as the candidate path
Stay Time Discovery

• Timeline alignment
  – A path is merged from different fragments
    • Photos taken in the same position in diff fragments might have diff times
  – Exploit Anchor Photos to align paths to get the right time span

• Finally, for a destination, we can derive paths with different stay times for route planning
Keyword-aware Route Search [Cao’12]

• Consider a user who wants to a one-day trip for an unfamiliar city. She might pose a query:
  – Find the most popular route from my hotel such that it passes by “shopping mall”, “restaurant”, and “museum”, and time spent on the road is within 4hr

• **Keyword-aware Route Query**
  (a) Start and end locations (hotel)
  (b) A set of keywords (shopping mall, restaurant, museum)
  (c) Budget limit (with 4hr)
  (d) A function $f$ calculating the score of a route (popularity)

• Goals: Satisfying (a)(b)(c) and optimizing $f$
Graph and Route: Illustration

• Graph

Objective score: \( o(v_0, v_1)=4 \)
Budget score: \( b(v_0, v_1)=1 \)

A location associated with a set of keywords

• Route

Query = \(<v_0, v_7, \{t_1, t_2, t_3\}, 8>\)
– Objective score
– Budget score
– Feasible route

\( OS(R) = 2+1+1 = 4 \)
\( BS(R) = 2+2+3 = 7 \)

\( OS(R_2) = 9, BS(R_2)=5 \)
Approach to Keyword-aware Route Query

- **Brute Force**: enumerating all candidate paths
- **The OSScaling Algorithm**
  - Reduce # of partial paths to avoids exhaustive search
  - Scale objective scores into integers using parameter $\varepsilon$:
    - Scaling factor: $\theta = \frac{\varepsilon o_{\text{min}} b_{\text{min}}}{\Delta}$ where $\Delta = \text{budget limit}$
    - Objective score scaling: $\hat{o}(v_i, v_j) = \left\lfloor \frac{o(v_i, v_j)}{\theta} \right\rfloor$

Query = $<v_0, v_7, \{t_1, t_2\}, 10>$, $\varepsilon = 0.5$

$\theta = \frac{\varepsilon o_{\text{min}} b_{\text{min}}}{\Delta} = \frac{0.5 \times 1 \times 1}{10} = \frac{1}{20}$
Node Label and Label Domination

- **Node**
  - Each node $v_i$ has a list of labels, each of which corresponds to a path $\tau$ from the starting node $v$ to $v_i$.
  - $\tau$ is the query keywords covered by $\tau$
  - $\tau$ is the scaled objective score
  - $\tau$ is the original objective score
  - $\tau$ is the budget score

- **Label Domination**

- **Label Order**

$L_4^0 = \langle \{t_1, t_2, t_4\}, 100, 5, 7\rangle$

$L_4^1 = \langle \{t_1, t_2, t_3\}, 120, 6, 11\rangle$

$L_7^0 = \langle \{t_1, t_2, t_3, t_4\}, 120, 6, 10\rangle$
The OSScaling Algorithm

Query = \langle v_0, v_7, \{t_1, t_2\}, 10 \rangle, \varepsilon = 0.5

\begin{align*}
L_6^0 \cdot BS + BS(\sigma_{6,7}) &= 11 > \Delta \\
\text{Label domination: When generating a new label } L_i^k, \text{ if it cannot be dominated by other labels on node } v_i, \text{ delete the labels dominated by it.}
\end{align*}
Route Planning: Outline

• Overview of Route Planning (10 mins)
  – Background and Introduction
  – Geo-social Trajectory Data: Categories, Challenges
  – Route Planning: Query, Approaches

• Route Planning Using GPS Trajectories (25 mins)
  – Graph Search
  – Pattern Mining/Matching
  – Prediction/Recommendation

• Route Planning Using Uncertain Trajectories (25 mins)
  – Deal with Data Uncertainty
  – On Check-in Data and Geo-tagged Photos

• Location Recommendation (10 mins)

• Conclusions and Future Directions (5 mins)
Location Recommendation vs. Location Prediction

• Location Recommendation
  – Recommend NEW locations (never visited before)

• Location Prediction
  – Predict the next existing locations (had ever visited)

• General considered factors
  – Current location info
  – Current time
  – User history/preference
  – Social interaction

Route Planning can be viewed as the successive applications of the mix of location recommendation and location prediction.
Check-in at New Venues

- People seek to discover new locations
  - 80%-90% of visited places are new
  - 60%-80% of check-ins occur at new places
Location Recommendation

General LocRec Methods

• **Popularity**: rank locations using # of check-ins
• **Content Filtering**: using venue type preference
• **Social Filtering**: rank locations using # of check-ins by friends
• **Home Distance**: geo-distance from home
• **K-NN User Similarity** (CF)
• **Place Network** (Item Similarity)
• **Matrix Factorization**
A Random Walk Around The City

1. RWR finds a balance between collective check-in behaviors (graph structure) and personalized bias.
2. RWR can be applied to users with no check-ins. (cold start)

<table>
<thead>
<tr>
<th>Method</th>
<th>APR</th>
<th>Precision@10</th>
<th>Recall@10</th>
</tr>
</thead>
<tbody>
<tr>
<td>Random</td>
<td>0.500</td>
<td>0.000</td>
<td>0.003</td>
</tr>
<tr>
<td>Popular</td>
<td>0.228</td>
<td>0.026</td>
<td>0.089</td>
</tr>
<tr>
<td>Activity</td>
<td>0.228</td>
<td>0.025</td>
<td>0.087</td>
</tr>
<tr>
<td>Home</td>
<td>0.383</td>
<td>0.008</td>
<td>0.026</td>
</tr>
<tr>
<td>Social</td>
<td>0.392</td>
<td>0.015</td>
<td>0.049</td>
</tr>
<tr>
<td>kNN</td>
<td>0.443</td>
<td>0.003</td>
<td>0.011</td>
</tr>
<tr>
<td>PlaceNet</td>
<td>0.337</td>
<td>0.026</td>
<td>0.077</td>
</tr>
<tr>
<td>MF</td>
<td>0.281</td>
<td>0.004</td>
<td>0.014</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>City</th>
<th>Foursquare</th>
<th>Gowalla</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>popularity</td>
<td>rwr</td>
</tr>
<tr>
<td>Austin</td>
<td>0.235</td>
<td>0.222</td>
</tr>
<tr>
<td>Boston</td>
<td>0.204</td>
<td>0.196</td>
</tr>
<tr>
<td>Dallas</td>
<td>0.247</td>
<td>0.232</td>
</tr>
<tr>
<td>Denver</td>
<td>0.233</td>
<td>0.200</td>
</tr>
<tr>
<td>London</td>
<td>0.264</td>
<td>0.262</td>
</tr>
<tr>
<td>Los Angeles</td>
<td>0.212</td>
<td>0.196</td>
</tr>
<tr>
<td>New York</td>
<td><strong>0.192</strong></td>
<td><strong>0.185</strong></td>
</tr>
<tr>
<td>Paris</td>
<td>0.265</td>
<td>0.256</td>
</tr>
<tr>
<td>San Francisco</td>
<td>0.208</td>
<td>0.200</td>
</tr>
<tr>
<td>Seattle</td>
<td>0.238</td>
<td>0.218</td>
</tr>
<tr>
<td>Seoul</td>
<td>0.210</td>
<td>0.226</td>
</tr>
<tr>
<td><strong>Average</strong></td>
<td><strong>0.228</strong></td>
<td><strong>0.217</strong></td>
</tr>
</tbody>
</table>
Geo-Influence for Collaborative Point-of-Interest Recommendation [Ye’11]

• Location-based Social Network

• Can we exploit (a) geographical influence between POIs, (b) user POI preference, and (c) social network for recommendation?

• A CF-based recommender is proposed

\[ S_{i,j} = (1 - \alpha - \beta)S_{i,j}^u + \alpha S_{i,j}^s + \beta S_{i,j}^g \]
Geographical Influence

- Measure **how likely** two of a user’s check-in POIs **within a given distance**
- Use **Power-law distribution** to model the check-in probability to the distance between POIs visited by a user
- Given user i and her check-in history $L_i$:

\[
Pr[L_i] = \prod_{l_m, l_n \in L_i \land m \neq n} Pr[d(l_m, l_n)]
\]

- For a new location $l_j$, we have the probability for user i to check in $l_j$:

\[
Pr[l_j | L_i] = \frac{Pr[l_j \cup L_i]}{Pr[L_i]} = \frac{Pr[L_i] \times \prod_{l_y \in L_i} Pr[d(l_j, l_y)]}{Pr[L_i]} = \prod_{l_y \in L_i} Pr[d(l_j, l_y)]
\]
User-based CF & Social-based CF

• User-based CF
  – Based on user similarity
    • $\hat{c}_{i,j}$: the predicted check-in probability
    • $w_{i,k}$: the similarity of user i and user k

• Social-based CF
  – Based on recommendation from friends

  $\hat{c}_{i,j} = \frac{\sum_{u_k \in F_i} SI_{k,i} \cdot c_{k,j}}{\sum_{u_k \in F_i} SI_{k,i}}$

  • Friends have closer social tie
  • Friends show more similar check-in behaviors

\[
SI_{k,i} = \eta \cdot \frac{|F_k \cap F_i|}{|F_k \cup F_i|} + (1 - \eta) \cdot \frac{|L_k \cap L_i|}{|L_k \cup L_i|}
\]
Time-aware POI Recommendation [Yuan’13]

• Based on LBSN, can we further explore users’ temporal behavior to recommend locations?
  – E.g. A user is more likely to go to a restaurant rather than a bar for lunch at noon
  – E.g. A user is more likely to go to a bar rather than a library at midnight

• A time-aware CF-based recommender
  – *Temporal Influence*: personal temporal behaviors
  – *Spatial Influence*: visit nearby locations
Incorporating Temporal Influence to User-based CF

• Check-in Representation
  – User-POI matrix \(\rightarrow\) user-time-POI cube

• Recommendation Formula

\[
\hat{C}_{u,l} = \frac{\sum_v w_{u,v} c_{v,l}}{\sum_v w_{u,v}} \rightarrow \hat{C}_{u,t,l} = \frac{\sum_v w_{u,v}^{(t)} c_{v,t,l}}{\sum_v w_{u,v}^{(t)}}
\]

• Similarity Computation

\[
w_{u,v} = \frac{\sum_l c_{u,l} c_{v,l}}{\sqrt{\sum_l c_{u,l}^2} \sqrt{\sum_l c_{v,l}^2}} \rightarrow w_{u,v}^{(t)} = \frac{\sum_{t=1}^{T} \sum_{l=1}^{L} c_{u,t,l} c_{v,t,l}}{\sqrt{\sum_{t=1}^{T} \sum_{l=1}^{L} c_{u,t,l}^2} \sqrt{\sum_{t=1}^{T} \sum_{l=1}^{L} c_{v,t,l}}}
\]

90
Incorporating Spatial Influence

• The willingness of a user to visit a dis km far-away POI, modeled by power-law distribution
  \[ wi(dis) = a \times dis^k \]

• Conditional Probability (currently at li)
  \[ p(l_j | l_i) = \frac{wi(dis(l_i, l_j))}{\sum_{l_k \in L, l_k \neq l_i} wi(dis(l_i, l_k))} \]

• Recommend by spatial influence
  \[ \hat{c}^{(s)}_{u,l} = P(l | L_u) \propto P(l)P(L_u | l) = P(l) \prod_{l' \in L_u} P(l' | l) \]

• Unified Framework of Recommendation
  \[ c_{u,t,l} = \alpha \times \bar{c}^{(t)}_{u,t,l} + (1 - \alpha) \times \bar{c}^{(s)}_{u,t,l} \]
Route Planning: Outline

• **Overview of Route Planning** (10 mins)
  – Background and Introduction
  – Geo-social Trajectory Data: Categories, Challenges
  – Route Planning: Query, Approaches

• **Route Planning Using GPS Trajectories** (25 mins)
  – Graph Search
  – Pattern Mining/Matching
  – Prediction/Recommendation

• **Route Planning Using Uncertain Trajectories** (25 mins)
  – Deal with Data Uncertainty
  – On Check-in Data and Geo-tagged Photos

• **Location Recommendation** (10 mins)

• **Conclusions and Future Directions** (5 mins)
Route Planning: Summary

Data
- GPS Trajectory Data
- Uncertain Trajectory
- Check-in Data
- Uncertain Trajectory: Geo-tagged Photos

Query
- Social Query
- Context Query
- Location Query

Approach
- Graph Search
- Pattern Matching & Mining
- Prediction/Recommendation
Route Planning: Opportunities

• Urban Computing
  – Apply **real-time** and **heterogeneous urban sensors data**
    • E.g. **air quality, traffic flow, weather info,**
  – Enable smarter / fine-grained route planning, e.g.

<table>
<thead>
<tr>
<th>Potential Tasks</th>
<th>Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>Find fast driving routes</td>
<td>Instant Traffic Flow</td>
</tr>
<tr>
<td>Recommend healthy/fresh venues</td>
<td>Air Quality Monitoring</td>
</tr>
<tr>
<td>Provide routes with commuting time</td>
<td>Public Transportation and Road Network</td>
</tr>
<tr>
<td>Recommend weather-based locations</td>
<td>Temperature, weather conditions</td>
</tr>
</tbody>
</table>
Route Planning: Opportunities (cont.)

• **Microblogging**
  – Social contents provide rich **user experience** and **event information** on locations
    • E.g. venue opinion, collective sentiments
    • E.g. exhibitions, ball games, movies, concerts
  – Planning routes based on sentiments and events

<table>
<thead>
<tr>
<th>Potential Tasks</th>
<th>Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>Planning routes maximizing rating</td>
<td>Venue rating</td>
</tr>
<tr>
<td>Event-centric route planning</td>
<td>Event location, time, and schedule</td>
</tr>
<tr>
<td>Recommend “positive” locations</td>
<td>Sentiments detected</td>
</tr>
<tr>
<td>Coordinate the group trip members</td>
<td>Location-based social network</td>
</tr>
</tbody>
</table>