Trajectory analysis

Ivan Kukanov

Joensuu, 2014
Semantic Trajectory Mining for Location Prediction

Josh Jia-Ching Ying
Tz-Chiao Weng
Vincent S. Tseng
Taiwan
Wang-Chien Lee
Wang-Chien Lee
USA

Copyright © 2011 ACM
The next location prediction approaches

- Personal-based prediction [1]
- General-based prediction [2,3]

Note: further consider General-based prediction
Semantic Trajectory Mining for Location Prediction

• predicting the next location of a user’s movement

Input:
• GPS user’s trajectories
Geographical and Semantic similarity

- **Example:** \( T_1 \) and \( T_2 \) – geographically,
  \( T_2 \) and \( T_3 \) – semantically
SemanPredict framework

• Offline and Online Modules

• Novel prediction strategy: Semantic and Geographic trajectory patterns

• MIT reality dataset [4]
  – 106 mobile users
  – Over 500,000 hours
SemanPredict: offline module

1. Data preprocessing
2. Semantic mining
3. Geographic mining
SemanPredict: offline module

1. Data preprocessing:

GPS trajectories to *stay locations sequence* [1, 5]

Output: $T_1 = \langle S_0, S_1, S_2 \rangle$; $T_2 = \langle S_0, S_4, S_3 \rangle$; $T_3 = \langle S_0, S_5, S_4, S_3 \rangle$; ...
2. Semantic mining:

2.1. Assign *semantic labels* to stay locations using gazetteer [6];

- **Example:**

  **Input:** $T_1=\langle S_0, S_1, S_2 \rangle$; $T_2=\langle S_0, S_4, S_3 \rangle$; $T_3=\langle S_0, S_5, S_4, S_3 \rangle$; ...

  **Output:** $T_1=\langle \text{School, Park, Stadium} \rangle$; $T_2=\langle \text{School, Bank, Hospital} \rangle$; $T_3=\langle \text{School, Unknown, Bank, Hospital} \rangle$; ...

---

*SemanPredict*: offline module
2. Semantic mining:

- Example: result *semantic labels*

**SemanPredict**: offline module
2. Semantic mining

2.2. Similar User Clustering

- Complete linkage clustering

**Similarity measure:** \( \text{Maximal Semantic Trajectory Pattern Similarity} \) [6]
**SemanPredict**: offline module

- **Example**: Longest Common Sequence (LCS)

\[ P = \langle \text{School, Park, Stadium} \rangle, \quad Q = \langle \text{School, Bank, Park, Shop} \rangle \]

\[ \text{LCS}(P, Q) = \langle \text{School, Park} \rangle \]

\[ \text{ratio}(P) = \frac{1 + 1}{3} = \frac{2}{3} \quad \text{ratio}(Q) = \frac{1 + 1}{4} = \frac{1}{2} \]

\[ \text{dist}(P, Q) = \frac{1}{2} \cdot \left( \frac{2}{3} + \frac{1}{2} \right) = \frac{7}{12} \]
**SemanPredict**: offline module

3. Geographic mining

**Input:**
- stay location sequence
- clusters from semantic mining

3.1. Grouping Users’ Stay Location Sequences

*(Prefix-Span [9])*

**Output:** clusters of stay location sequence based on semantic clusters
SemanPredict: offline module

- Recap

**Input:** GPS user’s trajectories
**SemanPredict**: online module

Semantic Trajectory Mining for Location Prediction
SemanPredict: online module

- **Input:** the trajectory of the user’s recent moves
SemanPredict: online module

- Matching score:

\[ \text{Score} = \beta \times \text{GeographicScore} + (1 - \beta) \times \text{SemanticScore}, \]
where \(0 < \beta \leq 1\)

Note: how well the ‘user behavior’ matches model
Summarizing Trajectories into k-Primary Corridors: A Summary of Results

Michael R. Evans
Dev Oliver
Francis Harvey
USA

Copyright © 2012 ACM
Why k-Primary Corridors?

- Identifying potential avenues for access
- Determine bus, metro and railway systems corridors

Traffic jam?!!!
Basic concepts

• Road network
  – Graph $G = \{V, E\}$
  – Nodes = road intersections

• Track
  – GPS trace to series of nodes and edges of road network

• Primary corridor
  – ‘Representative track of group of other tracks’ (Medoid)

Summarizing Trajectories into k-Primary Corridors: A Summary of Results
• Example:

Road Network

7-Primary Corridors

Summarizing Trajectories into k-Primary Corridors: A Summary of Results
Problem statement

• Given
  – Road Network: $G = \{V, E\}$
  – Collection of Tracks: $T$
  – Number $k$

• Find
  – $k$-Primary Corridors

• Constraints
  – Each primary corridor is track itself from set $T$
  – $G$ is a connected graph with nonnegative weights
Solution

• Input: see Given
  1. Compute the *track similarity matrix*

<table>
<thead>
<tr>
<th></th>
<th>Track 1</th>
<th>Track 2</th>
<th>Track 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Track 1</td>
<td>0</td>
<td>1.16</td>
<td>3</td>
</tr>
<tr>
<td>Track 2</td>
<td>1.16</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>Track 3</td>
<td>3</td>
<td>2</td>
<td>0</td>
</tr>
</tbody>
</table>


Note: *bottleneck* is the track similarity matrix computation
**Example:** time comparison of steps in k-Primary Corridor problem (100 nodes/tracks, k=5)

<table>
<thead>
<tr>
<th>Steps of k-PC</th>
<th>Runtime</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Track Similarity Matrix</td>
<td>46.56 sec</td>
<td>94.1%</td>
</tr>
<tr>
<td>Partitioning / Clustering</td>
<td>2.96 sec</td>
<td>5.9%</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>49.52 sec</strong></td>
<td><strong>100%</strong></td>
</tr>
</tbody>
</table>

Summarizing Trajectories into k-Primary Corridors: A Summary of Results
Approaches to compute the Track Similarity Matrix

– **Graph-Node Track Similarity** (naïve approach)

1. Consider all track pairs \((t_i,t_j)\)
2. For each pair of nodes in tracks \((t_i,t_j)\) run Dijkstra

Note: \(O(N^2 \cdot N^2 \cdot N \cdot \log N)\)
• **Example**: track similarity matrix computation
Example: track similarity matrix computation (cont.)

Transform to graph (for simplicity all weights equal 1)
• **Example:** track similarity matrix computation (cont.)

Calculate **track similarity function**

\[
s(t_i, t_j) = \frac{1}{|t_i|} \sum_{n \in t_i} \min_{m \in t_j} \left[ \text{ShortestPath}(n, m) \right]
\]

\(n, m\) – nodes of tracks \(t_i, t_j\) respectively

<table>
<thead>
<tr>
<th></th>
<th>Track 1</th>
<th>Track 2</th>
<th>Track 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Track 1</td>
<td>0</td>
<td>1.16</td>
<td>3</td>
</tr>
<tr>
<td>Track 2</td>
<td>1.16</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>Track 3</td>
<td>3</td>
<td>2</td>
<td>0</td>
</tr>
</tbody>
</table>
Approaches to compute the Track Similarity Matrix

– Matrix-Element Track Similarity

1. Consider all track pairs \((i, j)\)

2. Attach a **virtual node** to track \(i\)

3. Run **single** Dijkstra from the virtual node to all the nodes in track \(j\)

4. Compute track similarity metric

**Note:** \(O(N^2 \cdot N \cdot \log N)\)
• **Example:** track similarity matrix computation

Summarizing Trajectories into k-Primary Corridors: A Summary of Results
• **Example:** track similarity matrix computation (cont.)

Calculate **track similarity function**

\[
s(t_i, t_j) = \frac{1}{|t_i|} \sum_{m \in t_j} \text{dist}[m]
\]

\(\text{dist}[m]\) – distance from node \(m\) in tracks \(t_j\) to its closest node in track \(t_i\)
Detecting Road Intersections from GPS Traces

Alireza Fathi
John Krumm
Microsoft Research
USA

2010
Problem definition

1. GPS data collecting
   – Expensive:
     Specialized vehicle
     Keep up with changes in the road
   
   – Chip:
     Regular vehicle
     Detect road intersection
Problem definition

**Input:** GPS data of regular vehicles

2. **Infer the road network from GPS traces:**
   - Graph $G = \{V, E\}$
   - Nodes = road intersections

**Goal:** detect road intersections
Example: GPS data of regular vehicles

(a) all GPS data

(b) one intersection

Detecting Road Intersections from GPS Traces
Example: Local Shape Descriptor with 32 bins

• Sliding over GPS data to detect intersection
Training phase

**Input:** examples of GPS trace with intersections and non-intersections

1. Sliding by *shape descriptor* over examples
2. Map the *bins to the feature vector*
3. Learn classifier (Adaboost [8])
Example: result on test dataset
Thank you for your attention!
References


2. E. H.-C. Lu and V. S. Tseng. Mining Cluster-Based Mobile Sequential Patterns in Location-Based Service Environments. MDM, 2009


References

