Where Next? Data Mining Techniques and Challenges for Trajectory Prediction

Layla Pournajaf

Guest Lecture
CS570: Data Mining
Department of Mathematics and Computer Science
Fall 2015
Trajectory Prediction: Finding Next Location

- Navigational services.
- Traffic management.
- Location-based advertising.
- Simulation.

Trajectory Prediction: Other Variants

- Destination prediction
- Path prediction with known destination
- Path prediction with unknown destination
  - Similar to predicting next N locations

Next Location Prediction: Definitions

Definition 1. A trajectory $T$ is a sequenced set of tuples

$$T = \langle \langle x_1, y_1 \rangle, t_1 \rangle, ..., \langle x_n, y_n \rangle, t_n \rangle$$

where $\langle x_i, y_i \rangle$ indicates a location at time point $t_i$ and $t_i \leq t_{i+1}$.

Definition 2. Given a set of trajectories $O$, and a partial trajectory of a moving object $P$ at current time point $t_c$,

$$P = \langle \langle x_1, y_1 \rangle, t_1 \rangle, ..., \langle x_c, y_c \rangle, t_c \rangle$$

Next Location Prediction problem predicts the object’s location at next time point $t_{c+1}$. 
Next Location Prediction: Steps

Raw Trajectories

Next Location Prediction: Steps

Raw Trajectories

Preprocessed Trajectories

Next Location Prediction: Steps

Raw Trajectories

Preprocessed Trajectories

Prediction Model

Raw Trajectories: GPS Data

Source: www.openstreetmap.org
Real-world data include raw trajectories of continuous GPS coordinates which are noisy and inaccurate!

Source: www.openstreetmap.org
Raw Trajectories: GPS Data

```
<table>
<thead>
<tr>
<th></th>
<th>userid</th>
<th>trajid</th>
<th>latitude</th>
<th>longitude</th>
<th>date</th>
<th>time</th>
<th>time without time zone</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>39.964663</td>
<td>116.31845</td>
<td>2008-10-23</td>
<td>02:53:10</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>0</td>
<td>39.984686</td>
<td>116.318417</td>
<td>2008-10-23</td>
<td>02:53:15</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>0</td>
<td>39.984688</td>
<td>116.318385</td>
<td>2008-10-23</td>
<td>02:53:20</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>0</td>
<td>39.964655</td>
<td>116.318263</td>
<td>2008-10-23</td>
<td>02:53:25</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>0</td>
<td>39.984611</td>
<td>116.318026</td>
<td>2008-10-23</td>
<td>02:53:30</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>0</td>
<td>0</td>
<td>39.984608</td>
<td>116.317761</td>
<td>2008-10-23</td>
<td>02:53:35</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>0</td>
<td>0</td>
<td>39.964563</td>
<td>116.317517</td>
<td>2008-10-23</td>
<td>02:53:40</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>0</td>
<td>0</td>
<td>39.984539</td>
<td>116.317294</td>
<td>2008-10-23</td>
<td>02:53:45</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>0</td>
<td>0</td>
<td>39.984606</td>
<td>116.317065</td>
<td>2008-10-23</td>
<td>02:53:50</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>0</td>
<td>0</td>
<td>39.984568</td>
<td>116.316911</td>
<td>2008-10-23</td>
<td>02:53:55</td>
<td></td>
</tr>
</tbody>
</table>
```
Next Location Prediction: Steps

Raw Trajectories

Preprocessed Trajectories

Next Location Prediction: Preprocessing

- Discretizing Time
  - 30 seconds, one hour

- Discretizing Location
  - Grid-based
  - Mining Frequent Regions
    - Clustering
    - Semantic-based

Preprocessing Trajectories: Grids

Map of Beijing with $30 \times 30$ grid overlay: Each cell $\approx 1.78\text{km}^2$

Source: Xue, Andy Yuan, et al. "Destination prediction by sub-trajectory synthesis and privacy protection against such prediction." ICDE 2013.
Preprocessing Trajectories: Mining Frequent Regions

- Clustering
  - DBScan
  - Hierarchical Clustering

- Semantic-based
  - Using points of interests

<table>
<thead>
<tr>
<th>usrid</th>
<th>trajid</th>
<th>time</th>
<th>region</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>157</td>
<td>11</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>157</td>
<td>12</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>157</td>
<td>13</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>158</td>
<td>10</td>
</tr>
<tr>
<td>5</td>
<td>2</td>
<td>159</td>
<td>10</td>
</tr>
<tr>
<td>6</td>
<td>2</td>
<td>159</td>
<td>11</td>
</tr>
<tr>
<td>7</td>
<td>2</td>
<td>159</td>
<td>12</td>
</tr>
<tr>
<td>8</td>
<td>2</td>
<td>166</td>
<td>13</td>
</tr>
<tr>
<td>9</td>
<td>2</td>
<td>161</td>
<td>14</td>
</tr>
<tr>
<td>10</td>
<td>2</td>
<td>161</td>
<td>13</td>
</tr>
</tbody>
</table>

Next Location Prediction: Steps

Raw Trajectories

Preprocessed Trajectories

Prediction Models

Next Location Prediction: Methods

- Personalized / Individual-based:
  - Utilize only the history of one object to predict its future locations

- General:
  - Utilize the history of all objects to predict future locations
Next Location Prediction: Methods

- Model-based (formulate the movement of moving objects using mathematical models)
  - Markov Models
  - Recursive Motion Function (Y. Tao et. al., ACM SIGMOD 2004)

- Pattern-based (exploit pattern mining algorithms for prediction)
  - Sequential Pattern Mining (G. Yavas et. al., DKE 2005)
  - Trajectory Pattern Mining

- Hybrid
  - Recursive Motion Function + Sequential Pattern Mining (H. Jeung et. al., ICDE 2008)
Next Location Prediction: Methods

- Model-based (formulate the movement of moving objects using mathematical models)
  - Markov Models
    - Recursive Motion Function (Y. Tao et. al., ACM SIGMOD 2004)

- Pattern-based (exploit pattern mining algorithms for prediction)
  - Sequential Pattern Mining (G. Yavas et. al., DKE 2005)
  - Trajectory Pattern Mining

- Hybrid
  - Recursive Motion Function + Sequential Pattern Mining (H. Jeung et. al., ICDE 2008)
Next Location Prediction: Methods

- Model-based (formulate the movement of moving objects using mathematical models)
  - Markov Models
    - Recursive Motion Function (Y. Tao et. al., ACM SIGMOD 2004)

- Pattern-based (exploit pattern mining algorithms for prediction)
  - Sequential Pattern Mining (G. Yavas et. al., DKE 2005)
    - Trajectory Pattern Mining

- Hybrid
  - Recursive Motion Function + Sequential Pattern Mining (H. Jeung et. al., ICDE 2008)
Markov Process: Definition

- The **Markov property** is the independence of the future from the past, given the present.

- More formally:

  \[ P(X_n = x_n | X_{n-1} = x_{n-1}, \ldots, X_0 = x_0) = P(X_n = x_n | X_{n-1} = x_{n-1}) \]

- A stochastic (random) process has the Markov property if the **conditional probability distribution** of future states of the process (conditional on both past and present states) depends only upon the present state, not on the sequence of events that preceded it. A process with this property is called a **Markov process**. (Source: Wikipedia)
Building a Simple Markov Model

(a) $3 \times 3$ grid on the example

(b) $3 \times 3$, Markov model

Source: Xue, Andy Yuan, et al. "Destination prediction by sub-trajectory synthesis and privacy protection against such prediction." ICDE 2013.
Markov Model: Transition Probabilities

\[ p_{45} = \frac{2}{3} \]

\[ p_{56} = \frac{1}{3} \]

Source: Xue, Andy Yuan, et al. "Destination prediction by sub-trajectory synthesis and privacy protection against such prediction." ICDE 2013.
Markov Model: Transition Matrix

\[ M = \begin{pmatrix}
0 & p_{12} & 0 & p_{14} & 0 & 0 & 0 & 0 & 0 \\
p_{21} & 0 & p_{23} & 0 & p_{25} & 0 & 0 & 0 & 0 \\
0 & p_{32} & 0 & 0 & 0 & p_{36} & 0 & 0 & 0 \\
p_{41} & 0 & 0 & 0 & p_{45} & 0 & p_{47} & 0 & 0 \\
0 & p_{52} & 0 & p_{54} & 0 & p_{56} & 0 & p_{58} & 0 \\
0 & 0 & p_{63} & 0 & p_{65} & 0 & 0 & 0 & p_{69} \\
0 & 0 & 0 & p_{74} & 0 & 0 & 0 & p_{78} & 0 \\
0 & 0 & 0 & 0 & p_{85} & 0 & p_{87} & 0 & p_{89} \\
0 & 0 & 0 & 0 & 0 & p_{96} & 0 & p_{98} & 0
\end{pmatrix} \]

Source: Xue, Andy Yuan, et al. "Destination prediction by sub-trajectory synthesis and privacy protection against such prediction." ICDE 2013.
Partial Trajectory:
\(<r_1, t_1>, <r_2, t_2>, ..., <r_c, t_c>, ..., <?, t_{c+1}>\)

Prediction:
• Having a partial trajectory (discretized) including the current region \(r_c\), find the most probable region at time point \(t_{c+1}\)

\[
\arg\max_{r_{c+1} \in \{r_1, r_2, ..., r_n\}} P(R_{c+1} = r_{c+1} | R_c = r_c) = \arg\max_{r_{c+1} \in \{r_1, r_2, ..., r_n\}} P(r_{c+1} | r_c)
\]
Next Location Prediction: Methods

- Model-based (formulate the movement of moving objects using mathematical models)
  - **Markov Models**
    - Recursive Motion Function (Y. Tao et. al., ACM SIGMOD 2004)

- Pattern-based (exploit pattern mining algorithms for prediction)
  - Sequential Pattern Mining (G. Yavas et. al., DKE 2005)
  - **Trajectory Pattern Mining**

- Hybrid
  - Recursive Motion Function + Sequential Pattern Mining (H. Jeung et. al., ICDE 2008)
Next Location Prediction: Methods

- Model-based (formulate the movement of moving objects using mathematical models)
  - Markov Models
    - Recursive Motion Function (Y. Tao et. al., ACM SIGMOD 2004)

- Pattern-based (exploit pattern mining algorithms for prediction)
  - Sequential Pattern Mining (G. Yavas et. al., DKE 2005)
  - Trajectory Pattern Mining

- Hybrid
  - Recursive Motion Function + Sequential Pattern Mining (H. Jeung et. al., ICDE 2008)
1. Preprocess raw trajectories and extract frequent sequential patterns (T-Pattern)
Next Location Prediction: Trajectory Pattern Mining

1. Preprocess raw trajectories and extract frequent sequential patterns (T-Pattern)

2. Build a Prefix Tree (T-Pattern Tree)
Next Location Prediction: Trajectory Pattern Mining

1. Preprocess raw trajectories and extract frequent sequential patterns (T-Pattern)
2. Build a Prefix Tree (T-Pattern Tree)
3. Predict Next Location

Trajectory Pattern Mining: Extract T-Patterns

\[ \langle x_1, y_1, t_1 \rangle, \langle x_2, y_2, t_2 \rangle, \ldots, \langle x_n, y_n, t_n \rangle \]

\[ R_0 \xrightarrow{\alpha_1} R_1 \xrightarrow{\alpha_2} \ldots \xrightarrow{\alpha_n} R_n \]

Two points match if one falls within a spatial neighborhood $N()$ of the other.
Trajectory Pattern Mining: Extract T-Patterns

\(<x_1, y_1, t_1>, <x_2, y_2, t_2>, \ldots, <x_n, y_n, t_n>\)

- Two points match if one falls within a **spatial neighborhood** \(N()\) of the other
- Two transition times match if their **temporal difference** is \(\leq \tau\)
Two points match if one falls within a **spatial neighborhood** $N()$ of the other.

Two transition times match if their **temporal difference** is $\leq \tau$

$$<x_1, y_1, t_1>, <x_2, y_2, t_2>, ..., <x_n, y_n, t_n>$$
Two points match if one falls within a spatial neighborhood $N()$ of the other.

Two transition times match if their temporal difference is $\leq \tau$.

Calculate support for each $T$-pattern.
Generating all association rules from each T-pattern and using them to build a classifier is too expensive.
To avoid the rules generation, the T-Pattern set is organized as a prefix tree.

For Each node $v$

- **Id** identifies the node $v$
- **Region** is a spatial component of the T-Pattern
- **Support** is the support of the T-pattern

For Each edge $j$

$[a,b]$ correspond to the time interval $\alpha_n$ of the T-Pattern
To avoid the rules generation, the T-Pattern set is organized as a prefix tree.

For Each node $v$

- **Id** identifies the node $v$
- **Region** is a spatial component of the T-Pattern
- **Support** is the support of the T-pattern

For Each edge $j$

$[a,b]$ correspond to the time interval $\alpha_n$ of the T-Pattern

(represented both the T-patterns:

$\langle \rangle_A \langle 9, 15 \rangle_B$ supp:31

$\langle \rangle_A \langle 9, 12 \rangle_B \langle 10, 56 \rangle_{B'}$ supp:31)
To avoid the rules generation, the T-Pattern set is organized as a prefix tree.

For Each node $v$

- **Id** identifies the node $v$
- **Region** is a spatial component of the T-Pattern
- **Support** is the support of the T-pattern

For Each edge $j$

$[a, b]$ correspond to the time interval $\alpha_n$ of the T-Pattern

$(\text{Root}, \langle 4, A, 31 \rangle, t_c)$,
$\langle \langle 4, A, 31 \rangle, \langle 9, B, 31 \rangle, [9, 15] \rangle$,
$\langle \langle 9, B, 31 \rangle, \langle 10, E, 21 \rangle, [10, 56] \rangle$

represents both the $T$-patterns:

$\langle(), A\rangle \langle(9, 15), B\rangle \supp: 31$
$\langle(), A\rangle \langle(9, 12), B\rangle \langle(10, 56), E\rangle \supp: 21$
To avoid the rules generation, the T-Pattern set is organized as a prefix tree.

For Each node $v$

- **Id** identifies the node $v$
- **Region** is a spatial component of the T-Pattern
- **Support** is the support of the T-pattern

For Each edge $j$

$[a,b]$ correspond to the time interval $\alpha_n$ of the T-Pattern
Three steps:
1. Search for best match
2. Candidate generation
3. Make predictions

Trajectory Pattern Mining: Predict Next Location
Trajectory Pattern Mining: Predict Next Location

Three steps:
1. Search for best match
2. Candidate generation
3. Make predictions
Trajectory Pattern Mining: Predict Next Location

Three steps:
1. Search for best match
2. Candidate generation
3. Make predictions
Three steps:
1. Search for best match
2. Candidate generation
3. Make predictions
Three steps:
1. Search for best match
2. Candidate generation
3. Make predictions
Three steps:
1. Search for best match
2. Candidate generation
3. Make predictions
Three steps:
1. Search for best match
2. Candidate generation
3. Make predictions

The **Best Match** is the path having:
- the maximum path score using the time and location matching
- at least one admissible prediction.
Next Location Prediction: Evaluation Criteria

- Prediction errors (distance and time)
- Prediction accuracy (precision and recall)
- Prediction rate
References