Similarity-based Analysis for Trajectory Data

Kevin Zheng
Outline

• Background
  – What is trajectory
  – Where do they come from
  – Why are they useful
  – Characteristics

• Trajectory similarity search
  – Query classification
  – Trajectory similarity measures
  – Trajectory index

• Similarity-based trajectory mining
  – Popular route mining
  – Co-traveller discovery
  – Trajectory clustering
Outline

• **Background**
  – What is trajectory
  – Where do they come from
  – Why are they useful
  – Characteristics

• **Trajectory similarity search**
  – Query classification
  – Trajectory similarity measures
  – Trajectory index

• **Similarity-based trajectory mining**
  – Popular route mining
  – Co-traveller discovery
  – Trajectory clustering
What is trajectory?

• Historical location records of moving objects
• In mathematics
  – Continuous function: time $\rightarrow$ location
  – Location can be any dimension
• In real applications
  – Locations are sampled periodically
  – A finite sequence of time-stamped locations: $<p_1, t_1>, <p_2, t_2>, ..., <p_n, t_n>$
  – $p$: two or three dimensions (longitude, latitude)
Where is it from?

• GPS module on moving objects
  – Vehicles, mobile phone users, animals
• Online social network
  – Twitter, Flickr, Facebook, Weibo
• Sensors
  – Surveillance cameras, RFID, WiFi
• More …
Who cares about it?

- Government
  - Traffic pattern analysis
  - Public transportation management
  - Urban planning
- Business
  - Location-based service
  - Personalized advertisement & recommendation
  - Taxi company, logistic company
- Scientists & Researchers
  - Zoologist, meteorologist, astronomer
  - Open problems, challenging tasks
- More …
Trajectory data are BIG

• Volume
• Velocity
• Variety
Volume

• In 2010, 1 billion vehicles
  – Taxi, logistic companies keep tracking their vehicles
  – Self-driving car in near future?
• In 2012, 1.08 billion smartphone users
• In 2013, 20 million surveillance cameras in China
• They are generator!
  – The data keep accumulated
Velocity

• Not just huge, they’re being generated quickly

• Vehicle tracking & navigation
  – Re-position every few seconds

• Geo-tagged social media
  – 2 million Flickr photos per day, 5% geo-tagged
  – 100 million posts on Sina Weibo per day, 1-2% geo-tagged
  – 400 million tweets per day, 1% geo-tagged

• Sensors
  – How many cars pass a road camera every day?
Geo-tagged tweets
Variety

• Data source
• Tracking devices
  – Car GPS, smartphones, sensors
• Tracking methods
  – Sampling strategy, sampling rate,
• Spatial length & temporal duration
• Data quality
Research directions

• Scalable, real-time data processing
• Flexible database storage and index
• Effective similarity measures
• Uncertainty management
• Data compression

Key and fundamental research problem: similarity-based analysis
Outline

• Background
  – What is trajectory
  – Where do they come from
  – Why are they useful
  – Characteristics

• Trajectory similarity search
  – Query classification
  – Trajectory similarity measures
  – Trajectory index

• Similarity-based trajectory mining
  – Popular route mining
  – Co-traveller discovery
  – Trajectory clustering
Similarity-based analysis for trajectories

• Core problem: trajectory similarity search
  – Input: a trajectory dataset $D$, a query $Q$
  – Output: a subset of $D$ that are ‘similar’ to $Q$

• Foundation
  – Trajectory similarity measures

• Approach
  – Index and search algorithm

• Application
  – Popular route mining (route recommendation)
  – co-traveller discovery, clustering, classification, etc…
Similarity query classification

• P-query
  – Query: point(s)

• R-query
  – Query: region (spatial & temporal dimension)

• T-query
  – Query: trajectory
P-query (single point)

Query location: $q$
Temporal constraint (optional): $tc = [ts, te]$

$D(q, T) = \min_{p \in T} \text{dist}(q, p)$
$p \in T$ and satisfy $tc$
$\text{dist}(q, p)$:
- $L_p$-norm
- Network distance

P-query (multiple points)

Query locations $Q$: $q_1$, $q_2$, $q_3$, $q_4$

$D(Q,T)$ is an aggregate function of $D(q,T)$

R-query

- Spatial region: $R$
- Temporal interval: $[ts, te]$

Ask for trajectories in a given region during a time interval

T-query

• Query: $T_q$

How to measure their distance?
Trajectory similarity measures

- Many-to-many mapping
- Different semantic/applications
- Different lengths
- Different sampling rates
- Noises
- Temporal dimension?
Classification

<table>
<thead>
<tr>
<th>Consider location only</th>
<th>Consider both location and time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Spatial-only</td>
<td>Spatial-temporal</td>
</tr>
<tr>
<td>Lp-norm</td>
<td>DTW, LCSS, EDR with time constrain</td>
</tr>
<tr>
<td>DTW</td>
<td></td>
</tr>
<tr>
<td>LCSS</td>
<td></td>
</tr>
<tr>
<td>EDR</td>
<td></td>
</tr>
<tr>
<td>OWD</td>
<td>Synchronous Euclidean Distance</td>
</tr>
<tr>
<td>LIP</td>
<td></td>
</tr>
</tbody>
</table>

Based on location samples
- Discrete
- Continuous

Based on line segments or curves
## Classification

<table>
<thead>
<tr>
<th></th>
<th>Spatial-only</th>
<th>Spatial-temporal</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Discrete</strong></td>
<td>Lp-norm</td>
<td>DTW, LCSS, EDR with time constrain</td>
</tr>
<tr>
<td></td>
<td>DTW</td>
<td></td>
</tr>
<tr>
<td></td>
<td>LCSS</td>
<td></td>
</tr>
<tr>
<td></td>
<td>EDR</td>
<td></td>
</tr>
<tr>
<td><strong>Continuous</strong></td>
<td>OWD</td>
<td>Synchronous Euclidean Distance</td>
</tr>
<tr>
<td></td>
<td>LIP</td>
<td></td>
</tr>
</tbody>
</table>
Lp-norm

- Average Lp-norm distance of all matched locations
- 1-to-1 mapping
- Trajectories are of the same length
Lp-norm

• Cannot detect similar trajectories with different sampling rates
• Sensitive to noise
DTW

• Dynamic Time Warping distance
  – Adaptation from time series distance measure
  – Used to handle time shift and scale in time series

• Optimal order-aware alignment between two sequences
  – Goal: minimize the aggregate distance between matched points

• 1-to-many mapping

Yi, Byoung-Kee, Jagadish, HV and Faloutsos, Christos, Efficient retrieval of similar time sequences under time warping. ICDE 1998
DTW for trajectories

• Nothing to do with ‘time’ at all
• Useful when detecting similar trajectories with different sampling rates
• Sensitive to noise
LCSS

- Longest Common Sub-Sequence
- Adaptation of string similarity
  - $\text{Lcss('abcde', 'bd')} = 2$
- Threshold-based equality relationship
  - Two locations are regarded as equal if they’re ‘close’ (compared to a threshold)
- 1-to-(1 or null) mapping

VLACHOS, M., GUNOPULOS, D., AND KOLLIOS, G. Discovering similar multidimensional trajectories. ICDE 2002
LCSS

- Insensitive to noise
- Not easy to define threshold
- May return dissimilar trajectories
EDR

- Edit Distance on Real sequence
- Adaptation from Edit Distance on strings
  - Number of insert, delete, replace needed to convert A into B
- Threshold-based equality relationship
  - Two locations are regarded as equal if they’re ‘close’ (compared to a threshold)

Lei Chen, M. Tamer Ozsu, Vincent Oria, Robust and Fast Similarity Search for Moving Object Trajectories. SIGMOD 2005
EDR

- Value means the number of operations, not “distance between locations”
  - Insensitive to noise
LCSS and EDR

• They are both count-based
  – LCSS counts the number of matched pairs
  – EDR counts the cost of operations needed to fix
    the unmatched pairs
• Higher LCSS, lower EDR
• If cost(replace) = cost(insert) + cost(delete):
  • $EDR(X, Y) = L(X) + L(Y) - 2LCSS(X, Y)$
## Classification

<table>
<thead>
<tr>
<th></th>
<th>Spatial-only</th>
<th>Spatial-temporal</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Discrete</strong></td>
<td>Lp-norm</td>
<td>DTW, LCSS, EDR with time constrain</td>
</tr>
<tr>
<td></td>
<td>DTW</td>
<td></td>
</tr>
<tr>
<td></td>
<td>LCSS</td>
<td></td>
</tr>
<tr>
<td></td>
<td>EDR</td>
<td></td>
</tr>
<tr>
<td><strong>Continuous</strong></td>
<td>OWD</td>
<td>Synchronous Euclidean Distance</td>
</tr>
<tr>
<td></td>
<td>LIP</td>
<td></td>
</tr>
</tbody>
</table>
OWD

• One Way Distance from $T_1$ to $T_2$ is:
  – Integral of the distance from points of $T_1$ to $T_2$
  – Divided by the length of $T_1$

$$D_{owd}(T_1, T_2) = \frac{1}{|T_1|} \left( \int_{p \in T_1} D_{point}(p, T_2) \, dp \right)$$

• Make it into symmetric measure

$$D(T_1, T_2) = \frac{1}{2} \left( D_{owd}(T_1, T_2) + D_{owd}(T_2, T_1) \right)$$

Bin Lin, Jianwen Su, One Way Distance: For Shape Based Similarity Search of Moving Object Trajectories. In Geoinformatica (2008)
OWD example

• Consider one trajectory as piece-wise line segment, and the other as discrete samples
LIP distance

- Locality In-between Polylines

\[ LIP(Q, S) = \sum_{\forall \text{polygon}_i} \text{Area}_i \cdot w_i \]

- \textit{Polygon} is the set of polygons formed between intersection points

\[ w_i = \frac{\text{Length}_Q(I_i, I_{i+1}) + \text{Length}_S(I_i, I_{i+1})}{\text{Length}_Q + \text{Length}_S} \]

LIP distance

- Only work for 2-dimensional trajectories
- Polygon $\rightarrow$ polyhedron: non-trivial change
Spatial-temporal similarity measure

• All the spatial-only similarity measures can incorporate time information in trajectories
• Lp-norm and DTW can apply a temporal constrain for more synchronized alignment
• EDR and LCSS can apply a temporal threshold on top of spatial threshold
## Classification

<table>
<thead>
<tr>
<th></th>
<th>Spatial-only</th>
<th>Spatial-temporal</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Discrete</strong></td>
<td>Lp-norm, DTW, LCSS, EDR</td>
<td>DTW, LCSS, EDR with time constrain</td>
</tr>
<tr>
<td><strong>Continuous</strong></td>
<td>OWD, LIP</td>
<td>Synchronous Euclidean Distance</td>
</tr>
</tbody>
</table>
DTW with temporal constrain

Time tolerance = 2
Spatial-temporal LCSS

• A temporal threshold controls how far in time we can go in order to match two locations

VLACHOS, M., GUNOPULOS, D., AND KOLLIOS, G. Discovering similar multidimensional trajectories. ICDE 2002
## Classification

<table>
<thead>
<tr>
<th>Discrete</th>
<th>Spatial-only</th>
<th>Spatial-temporal</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Lp-norm</td>
<td>DTW, LCSS, EDR with time constrain</td>
</tr>
<tr>
<td></td>
<td>DTW</td>
<td></td>
</tr>
<tr>
<td></td>
<td>LCSS</td>
<td></td>
</tr>
<tr>
<td></td>
<td>EDR</td>
<td></td>
</tr>
<tr>
<td>Continuous</td>
<td>OWD</td>
<td>Synchronous Euclidean Distance</td>
</tr>
<tr>
<td></td>
<td>LIP</td>
<td></td>
</tr>
</tbody>
</table>
SED

• Synchronous Euclidean Distance
  – Euclidean distance between locations at the same time instance of two trajectories
• Regard trajectory as continuous function of time

\[ D(\tau_1, \tau_2)|_T = \frac{\int_T d(\tau_1(t), \tau_2(t)) dt}{|T|} \]


POTAMIAS, M., PATROUMPAS, K., AND SELLIS, T. K. Sampling trajectory streams with spatiotemporal criteria. SSDBM 2006
Virtually create a sample point at \( t=3 \)
Trajectory index

• Similarity measures give the way to calculate the distance between two trajectories

• However …
  – Huge amount of trajectories
  – Linear scan is inefficient
Trajectory index

- 3D R-tree
- STR-tree (Spatio-temporal R-tree)
- TB-tree (Trajecoty Bundle)
- Multi-version R-tree
  - Partition temporal dimension
- Grid-based index
  - Partition spatial dimension
3D R-tree

- Indexing position samples only
  - Cannot answer queries about movements in-between those samples
- Indexing line segments
Problem of R-tree

• Large “dead space”
  – MBB covers a large portion of the space with no data
  – Low pruning power

• Trajectory preservation
  – Line segments are grouped merely by spatial proximity
  – Regardless which trajectory they belong to
  – Retrieving a trajectory requires visits of different paths in the tree
Augmented 3D R-tree

• Augment leaf node with orientation information
• Distance to line segments can be approximated more accurately

STR-tree

- Extension of augmented 3D R-tree
- Insertion
  - Try to keep the line segments belonging to the same trajectory together
  - Find the leaf node containing the predecessor
- Node split
  - Put disconnected segments into new node
  - Put the most recent \textit{backward-connected} segment into new node

TB-tree (Trajectory Bundle)

- A leaf node only contains segments belonging to the same trajectory
- Leaf nodes containing same trajectory are linked
- Strictly preserve trajectories

TB-tree (Trajectory Bundle)
Reflection

• Previous index treat spatial and temporal dimensions equally
  – 3D tree structure
• In trajectory database, temporal dimension is more dynamic
  – New segments are appended to existing trajectories
  – Archived trajectories rarely update
• Indexing spatial and temporal dimensions separately
  – Partition temporal dimension first
  – Partition spatial dimension first
Multi-version R-tree

For each timestamp, an R-tree is created. So, there are many R-trees. These R-trees are indexed.

Query for trajectories in a given region and in a given time interval:
1. The R-tree at the timestamp is found first
2. The trajectories in the specified region are retrieved from the R-tree.

Grid-based index

• Partition space into non-overlapping cells
• Trajectory segments in each cell are indexed on temporal dimension
• Query processing:
  – Spatial filtering
  – Temporal filtering

Grid-based index

Outline

• Background
  – What is trajectory
  – Where do they come from
  – Why are they useful
  – Characteristics

• Trajectory similarity search
  – Query classification
  – Trajectory similarity measures
  – Trajectory index

• Similarity-based trajectory mining
  – Popular route mining
  – Co-traveller discovery
  – Trajectory clustering
Popular route mining

• Shortest path may not be the favourable
• Find the most popular/desirable path using the GPS trajectories of past travellers
• Classification
  – No specific source/destination – hot route discovery
  – Specific source/destination – popular route search
T-pattern

- A set of individual trajectories that share the property of visiting the same sequence of places with similar travel times

\[ T = s_0 \xrightarrow{\alpha_1} s_1 \xrightarrow{\alpha_2} \cdots \xrightarrow{\alpha_n} s_n \]

1. Spatial discretization: discretize space into finite set of regions of interest (RoI)
2. Translate trajectories into sequence of RoIs
3. Adapt sequential pattern mining algorithms with time constrain

Periodic pattern

- Find the repeat pattern for individual’s trajectory

Interesting travel sequence

- A sequence of interesting locations that is travelled by experienced drivers frequently
- Interestingness of a location?
  - How many *experienced* travellers visited it
- Experience of a traveller?
  - How many *interesting* locations she has visited

Interesting travel sequence

- Hyperlink-Induced Topic Search (HITS)
Popular route search

• Given source/destination, find/estimate the most popular route in between

• *Ideally*, we can just count the number of trajectories on different paths connecting the two locations

Popular route discovery

• In real scenarios, it’s not easy to find such well-divided groups

• Even worse, there’s no trajectory connecting two locations at all!
Popular route discovery

- Construct a *transfer network* from raw trajectories as an intermediate result to capture the moving behaviors between locations
  - Node: cluster of turning points
  - Edge: trajectories passing two nodes
Popular route discovery

• Transfer probability

\[ Pr(n_i \rightarrow n_j) = \frac{\text{number of trajectories on } (n_i, n_j)}{\text{number of trajectories on all outgoing edges}} \]

• Find the route with the highest joint transfer probability w.r.t. destination

\[ \rho(R) = \prod_{j=1}^{i} n_j \cdot \text{popularity}(d) \]
Find popular route from uncertain trajectories

• Low-sampling-rate trajectories have high degree of uncertainty

• Uncertain trajectories are prevalent!

• Is it possible to recover the original route given a low-sampling-rate trajectory?

• What if…
  – There is a historical set of uncertain trajectories

Find popular route from uncertain trajectories

• Can we find popular routes for specific s/d using low-sampling-rate trajectories

Infrequent samples on the same path can reinforce each other, and they collectively form a more ‘dense’ trajectory.
Find popular route from uncertain trajectories

- Find PR for a given sequence of locations
- Two-phase approach
  - Local PR construction
  - Global PR search
Find popular route from uncertain trajectories without road networks

- A road network is not always available or applicable

1. Discretize space into disjoint cells
2. Derive the transfer graph using cells
3. Infer frequent ‘virtual edges’ on the graph

Time period-based most frequent path (TPMFP)

- Find the most frequent path for specific s/d and time period

- Desired properties for a MFP
  - Suffix optimal: suffix of a MFP is also a MFP
  - Length insensitive: MFP shouldn’t favor long/short route
  - Bottleneck free: MFP shouldn’t contain infrequent edges
Time period-based most frequent path (TPMFP)

- Footmark graph
  - A weighted sub-graph
  - Edge frequency: number of trajectories reaching $v_d$ during $T$
  - Path frequency: non-decreasingly sorted sequence of edge frequencies

\[
\begin{align*}
&v_1 \rightarrow v_2: 14, v_2 \rightarrow v_3: 10 \\
v_3 \rightarrow v_{12}: 10, v_2 \rightarrow v_{12}: 8 \\
&V_1 \rightarrow v_2 \rightarrow v_{12}: (8, 14) \\
&V_1 \rightarrow v_2 \rightarrow v_3 \rightarrow v_{12}: (10, 10, 14) \\
&V_1 \rightarrow v_{10} \rightarrow v_{11} \rightarrow v_{12}: (1, 21, 21)
\end{align*}
\]
Time period-based most frequent path (TPMFP)

• Compare path frequency
  – More-frequent-than relation (>)
  – \( F > F' \) if their first different value \( f_j > f'_j \)
  – \((10,10,14) > (8,14) > (1,21,21)\)

• Property
  – A total order, so MFP always exist
  – Guarantee the suffix optimal
  – Length of path doesn’t matter a lot
  – Path with infrequent edge frequency be disadvantaged
PR search is more challenging

• Specific source/destination (and time constraint)
  – Data are sparse
  – How to utilize more relevant data?
• Online performance
  – Efficiency is critical
Summary

• Background
  – What is trajectory
  – Where do they come from
  – Why are they useful
  – Characteristics

• Trajectory similarity search
  – Query classification
  – Trajectory similarity measures
  – Trajectory index

• Similarity-based trajectory mining
  – Popular route mining
  – Co-traveller discovery
  – Trajectory clustering
Thank you

• Questions
  – kevinz@itee.uq.edu.au
• DKE group at UQ
• My research