Splitter: Mining Fine-Grained Sequential Patterns in Semantic Trajectories

Chao Zhang¹, Jiawei Han¹, Lidan Shou², Jiajun Lu¹, Thomas La Porta³

czhang82@Illinois.edu

¹ University of Illinois at Urbana-Champaign
² Zhejiang University
³ Penn State University
Sequential pattern: a subsequence that matches at least \( n \) sequences (\( n \) is the support threshold) in the database.

Sequence Database:
C -> A -> B
A -> C -> B -> A
A -> B
B -> C

Sequential Pattern (\( n=3 \)):
A -> B
How do We Define Sequential Movement Pattern?

- Can we define it as a *place sequence* that matches at least \( n \) trajectories? No.
- Due to space continuity, similar places need to be grouped to collectively form frequent patterns.
How do We Define Sequential Movement Pattern?

- Meaningful patterns must satisfy three constraints:
  - Semantic consistency
  - Spatial compactness
  - Temporal continuity

Sequential Pattern: $G_1 \rightarrow G_2 \rightarrow G_3$
Existing Approaches

• Trajectory Pattern Mining $[1, 2, 3]$:
  ‣ Partition the space into small grids
  ‣ Group the places in the same grid (or several neighboring grids)
  ‣ Mine frequent sequential patterns.

Existing Approaches

• Drawbacks of rigid space partitioning:
  ‣ It suffers from the sharp boundary problem.
  ‣ It is hard to pre-specify the partition granularity.
An Overview of Splitter

- Splitter is a two-step approach.
  - Step 1: mining coarse patterns that satisfy the semantic and temporal constraints.
  - Step 2: splitting each coarse patterns into fine-grained ones to meet the spatial constraint.
Mining Coarse Patterns

- Group the places by category such that the places having the same category go to the same group.
  - We obtain groups like office, gym, restaurant, etc.
  - Each group can be viewed as an independent item.
Mining Coarse Patterns

- Transform trajectories into item sequences, by mapping the place ids to the group ids.

<table>
<thead>
<tr>
<th>Object</th>
<th>Semantic Trajectory</th>
</tr>
</thead>
<tbody>
<tr>
<td>$o_1$</td>
<td>\langle(p_3, 0), (p_1, 10), (p_7, 30), (p_9, 40)\rangle</td>
</tr>
<tr>
<td>$o_2$</td>
<td>\langle(p_5, 0), (p_7, 30), (p_2, 360), (p_7, 400), (p_{10}, 420)\rangle</td>
</tr>
<tr>
<td>$o_3$</td>
<td>\langle(p_3, 0), (p_6, 30)\rangle</td>
</tr>
<tr>
<td>$o_4$</td>
<td>\langle(p_2, 0), (p_1, 120), (p_6, 140), (p_8, 150), (p_{11}, 180)\rangle</td>
</tr>
<tr>
<td>$o_5$</td>
<td>\langle(p_{12}, 50), (p_8, 80), (p_{11}, 120), (p_4, 210)\rangle</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Object</th>
<th>Timestamped item sequence</th>
</tr>
</thead>
<tbody>
<tr>
<td>$o_1$</td>
<td>\langle(G_2, 0), (G_1, 10), (G_2, 30), (G_3, 40)\rangle</td>
</tr>
<tr>
<td>$o_2$</td>
<td>\langle(G_1, 0), (G_2, 30), (G_1, 360), (G_2, 400), (G_3, 420)\rangle</td>
</tr>
<tr>
<td>$o_3$</td>
<td>\langle(G_2, 0), (G_3, 30)\rangle</td>
</tr>
<tr>
<td>$o_4$</td>
<td>\langle(G_1, 0), (G_1, 120), (G_3, 140), (G_2, 150), (G_3, 180)\rangle</td>
</tr>
<tr>
<td>$o_5$</td>
<td>\langle(G_2, 50), (G_2, 80), (G_3, 120), (G_1, 210)\rangle</td>
</tr>
</tbody>
</table>
Mining Coarse Patterns

- We modify PrefixSpan by using the full projection principle.
  - It guarantees the result patterns satisfy the time constraint.
  - It extracts the snippets (place sequences) for each coarse pattern.

![Diagram showing place sequences](image-url)
What do We Have Now?

- A set of coarse patterns.
- The snippets for each coarse pattern.
Splitting Coarse Patterns

- We find fine-grained patterns by merging close snippets.
  - Each snippet is mapped to a weighted high-dimensional point (e.g., length-2 snippets are mapped to 4D points).
  - Detect dense and compact clusters in the high-dimensional space spanned by the snippets.
Splitting Coarse Patterns

- Finding snippet clusters via *weighted snippet shift*:
Splitting Coarse Patterns

• Top-down pattern discovery:
  ‣ Start with an initially large bandwidth.
  ‣ Gradually dampen the bandwidth and find patterns on-the-fly.
  ‣ Terminate until no more pattern can be found.

• We introduce a divide-and-conquer strategy to speed up the top-down discovery process.
Experimental Data

- A Foursquare check-in data set:
  - ~15K users in New York.
  - ~50K places.
  - 15 categories.

- Two synthetic data sets generated by the Brinkhoff’s network-based generator.
Compared Methods

• Grid
  ▸ Trajectory pattern mining [1] based on space partitioning.
• HC
  ▸ Group the places via top-down hierarchical clustering.
  ▸ Mine movement patterns using PrefixSpan.

Example Coarse Patterns

- Support threshold \( n = 100 \)

<table>
<thead>
<tr>
<th>length=2</th>
<th>Pattern</th>
<th>Sup</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Shop (\rightarrow) Food</td>
<td>1819</td>
</tr>
<tr>
<td></td>
<td>Food (\rightarrow) Shop</td>
<td>1464</td>
</tr>
<tr>
<td></td>
<td>Professional (\rightarrow) Nightlife Spot</td>
<td>1121</td>
</tr>
<tr>
<td></td>
<td>Outdoor (\rightarrow) Food</td>
<td>947</td>
</tr>
<tr>
<td></td>
<td>Residence (\rightarrow) College &amp; University</td>
<td>647</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>length=3</th>
<th>Pattern</th>
<th>Sup</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Shop (\rightarrow) Food (\rightarrow) Shop</td>
<td>262</td>
</tr>
<tr>
<td></td>
<td>Professional (\rightarrow) Food (\rightarrow) Nightlife Spot</td>
<td>240</td>
</tr>
<tr>
<td></td>
<td>Entertainment (\rightarrow) Food (\rightarrow) Shop</td>
<td>178</td>
</tr>
<tr>
<td></td>
<td>Transportation (\rightarrow) Shop (\rightarrow) Shop</td>
<td>174</td>
</tr>
<tr>
<td></td>
<td>Residence (\rightarrow) Outdoor (\rightarrow) Food</td>
<td>163</td>
</tr>
</tbody>
</table>
Example Fine-Grained Patterns

- Length-2 patterns for Shop -> Food:

![Diagram showing shops and restaurants with associated support values](Diagram)
Effectiveness Comparison

- Varying the support threshold:

(a) Coverage w.r.t. $\sigma$.

(b) Pattern number w.r.t. $\sigma$. 

Graphs showing the relationship between coverage and pattern number with varying $\sigma$.
Efficiency Comparison

- Varying the support threshold and the time interval:

(a) Running time w.r.t. $\sigma$.

(b) Running time w.r.t. $\Delta t$. 
Efficiency Comparison

- Efficiency on the synthetic data sets:

(a) Running time on S1K.
(b) Running time on S10K.
Summary

• Finding fine-grained sequential movement patterns is a critical yet challenging task.

• We develop a two-step method for mining fine-grained sequential movement patterns in semantic trajectories.
  ‣ Step 1: mining coarse patterns
  ‣ Step 2: splitting each coarse pattern into fine-grained ones.

• Our method significantly outperforms existing ones in both effectiveness and efficiency.