Learning to Rank Paths in Spatial Networks

Yang, Sean Bin; Yang, Bin

Publication date:
2019

Citation for published version (APA):
Learning to Rank Paths in Spatial Networks

Sean Bin Yang and Bin Yang
Department of Computer Science, Aalborg University, Denmark
(sean_yang@cs.au.dk)

Introduction
A routing service quality study shows that local drivers often choose paths that are neither shortest nor fastest, rendering classic routing algorithms often impractical in many real world routing scenarios.

In addition, commercial navigation systems, such as Google Maps and TomTom, often follow a similar strategy by suggesting multiple candidate paths to drivers, although the criteria for selecting the candidate paths are often confidential.

Challenges:
- Constructing an appropriate training path set \( \mathcal{PS} \) is non-trivial.
- Effective training models often rely on meaningful feature representation of input data—how to learning path representation.

Our approach:
- **Training Data Generation**: A compact set of diversified paths using trajectories as training data.
- **Path Representation**: An end-end deep learning framework is presented to solve the regression problem.
  - A spatial network embedding is proposed to embed each vertex to a feature vector by considering the road network topology.
  - Since a path is represented by a sequence of vertices, recurrent neural network is applied to model the sequence.
- **The RNN** finally outputs an estimated similarity score, which is compared against the ground truth similarity.

Solution Overview
- We propose a data-driven ranking framework **PathRank**, which ranks candidate paths by taking into account the paths used by local drivers in their historical trajectories.
- Most importantly, **PathRank** models ranking candidate paths as a “regression” problem—for each candidate path, **PathRank** estimates a ranking score for the candidate path.

Training Data Generation
- We proceed to elaborate how to generate a set of training paths for a trajectory path \( P \) from source \( s \) to destination \( d \).
- We propose the strategy using the diversified top-k shortest paths.

```
Algorithm 1: Top-k Diversified Paths
Input: Road network \( G \), source \( s \), destination \( d \), integer \( k \), similarity threshold \( \delta \)
Output: The diversified top-k paths \( \mathcal{DKPS} \)
1. Add the shortest path \( P_1 \) into \( \mathcal{DKPS} \);
2. while \( \mathcal{DKPS} < k \) do
3. Identify the next shortest path \( P_i \);
4. Boolean tag \( \leftarrow \) true;
5. for each path \( P \in \mathcal{DKPS} \) do
6. if sim \((P_i, P) \geq \delta \) then
7. tag \( \leftarrow \) false;
8. Break;
9. if tag then
10. Add \( P_i \) into \( \mathcal{DKPS} \);
11. return \( \mathcal{DKPS} \);
```

Experiments

**Experiments Setup**
- **Road Network and Trajectories**: North Jutland, Denmark, 180 million GPS records from 183 vehicles.
- **Ground Truth Data**: For each trajectory \( P_r \), we generate two sets of training paths: Top-\( k \) shortest paths \( \{TkDI\} \) and diversified top-\( k \) shortest paths \( \{D-TkDI\} \).
  - For each training path \( P \), we employ weighted Jaccard similarity \( \text{WeightedJaccard}(P, P_r) \) as \( P_r \)’s ground truth ranking score.
- **Evaluation Metrics**:
  - Mean Absolute Error (MAE) and Mean Absolute Relative Error (MARE)
  - Kendall Rank Correlation Coefficient (\( \tau \)) and Spearman’s Rank Correlation Coefficient (\( \rho \))

**Experiments Results**
- Table 1 shows that (1) when using the diversified top-k paths for training, we achieve higher accuracy compared to when using top-k paths; (2) a larger embedding feature size \( M \) achieves better results.
- Table 2 shows the results. In addition, PR-A2 achieves better accuracy than does PR-A1, meaning that updating embedding matrix \( B \) is useful.

<table>
<thead>
<tr>
<th>Strategies</th>
<th>( M )</th>
<th>MAE</th>
<th>MARE</th>
<th>( \tau )</th>
<th>( \rho )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( TkDI )</td>
<td>64</td>
<td>0.1433</td>
<td>0.2300</td>
<td>0.6638</td>
<td>0.7044</td>
</tr>
<tr>
<td></td>
<td>128</td>
<td>0.1168</td>
<td>0.1875</td>
<td>0.6913</td>
<td>0.7330</td>
</tr>
<tr>
<td>( D-TkDI )</td>
<td>64</td>
<td>0.1140</td>
<td>0.1830</td>
<td>0.6959</td>
<td>0.7346</td>
</tr>
<tr>
<td></td>
<td>128</td>
<td>0.0955</td>
<td>0.1533</td>
<td>0.7077</td>
<td>0.7492</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Strategies</th>
<th>( M )</th>
<th>MAE</th>
<th>MARE</th>
<th>( \tau )</th>
<th>( \rho )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( TkDI )</td>
<td>64</td>
<td>0.1163</td>
<td>0.1868</td>
<td>0.6835</td>
<td>0.7256</td>
</tr>
<tr>
<td></td>
<td>128</td>
<td>0.1130</td>
<td>0.1814</td>
<td>0.7082</td>
<td>0.7481</td>
</tr>
<tr>
<td>( D-TkDI )</td>
<td>64</td>
<td>0.0940</td>
<td>0.1509</td>
<td>0.7144</td>
<td>0.7532</td>
</tr>
<tr>
<td></td>
<td>128</td>
<td>0.0855</td>
<td>0.1373</td>
<td>0.7339</td>
<td>0.7731</td>
</tr>
</tbody>
</table>

**PathRank**

```
Training Data
P v v v v
P v v v v
P v v v v

Path Representation:
GRU GRU GRU
GRU GRU GRU

Hidden States
\( H_1 \rightarrow H_2 \rightarrow H_3 \rightarrow \cdots \rightarrow H_k \)

Vertex Embedding:
(v1, v2, v3) \to \{v\}

Recurrent Neural Network (RNN):

Training Data
P v v v v
P v v v v
P v v v v

PathRank
( ) , sim , P P P

Source, Destination
Training Data Generation
Candidate Paths
Spatial Network Embedding
RNN
Candidate Paths with Estimated Ranking Scores
```

```
```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```