Learning to Rank Paths in Spatial Networks

Yang, Sean Bin; Yang, Bin

Publication date:
2019

Citation for published version (APA):
Yang, S. B., & Yang, B. (2019). *Learning to Rank Paths in Spatial Networks*. Poster presented at 2020 IEEE 36th International Conference on Data Engineering (ICDE), Texas, United States.

Learning to Rank Paths in Spatial Networks

AALBORG UNIVERSITY DENMARK

Sean Bin Yang and Bin Yang

Department of Computer Science, Aalborg University, Denmark

{seany, byang}@cs.aau.dk

daisy Center for Data-Intensive Systems

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:

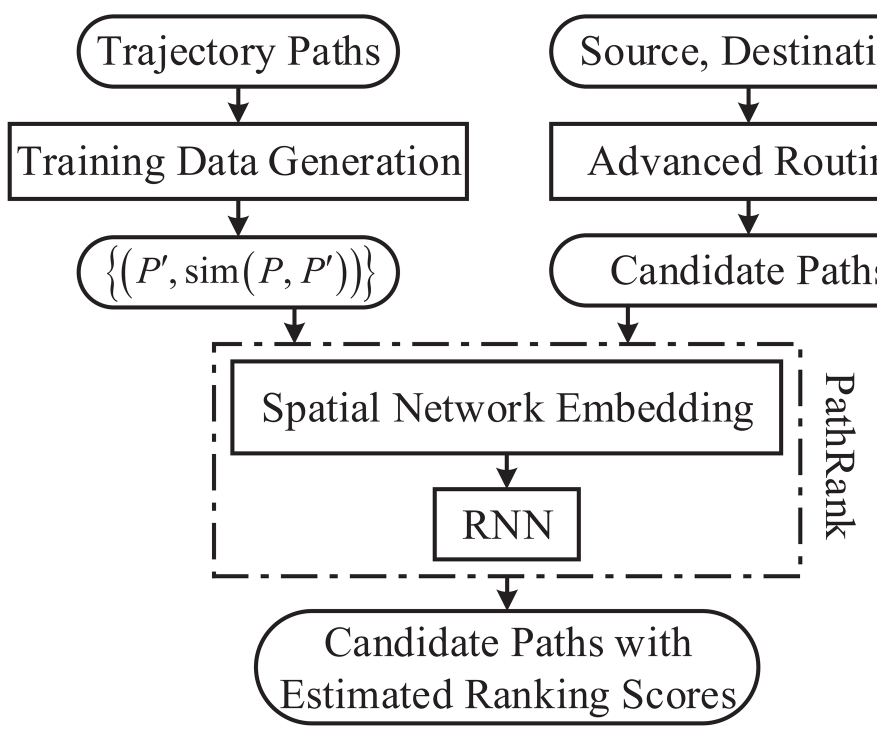
- Constructing an appropriate training path set $\mathcal{PS}$ is non-trivial.
- Effective training models often rely on meaningful feature representation of input data–how to learning path representation.

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 theirs 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.
- **Solution Overview.**



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.

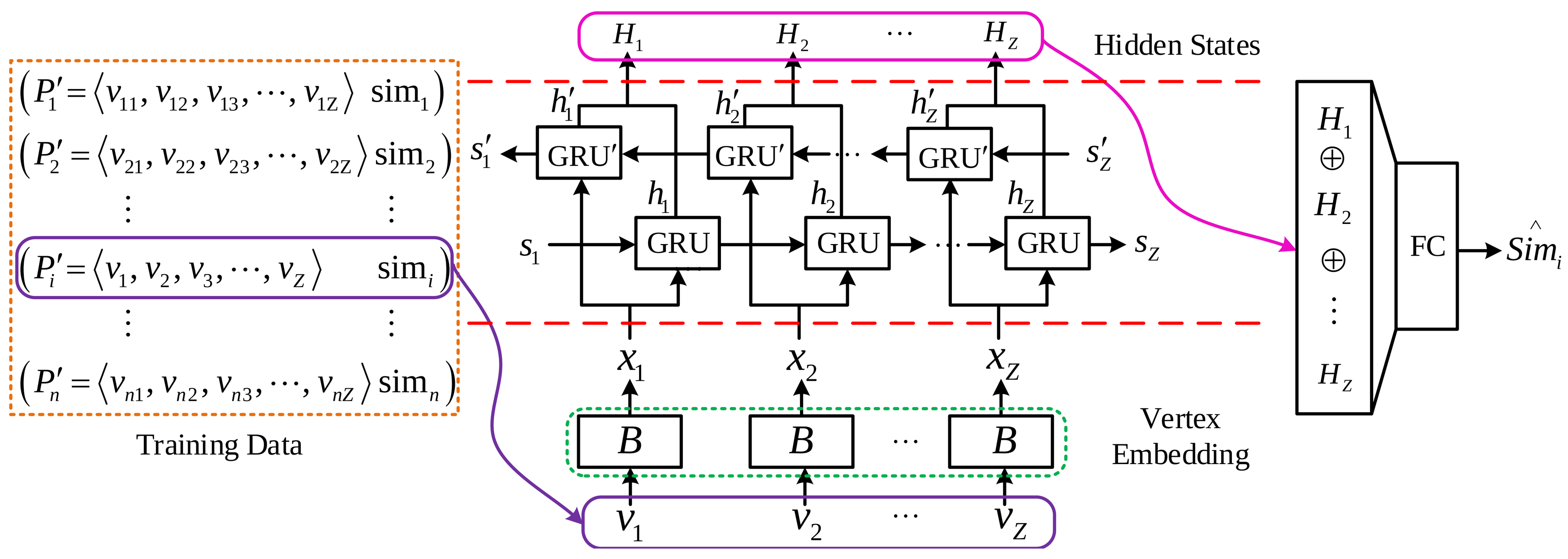
Algorithm 1: Top-k Diversified Paths

Input: Road network G, source s, destination d, integer k, similarity threshold δ

Output: The diversified top-k paths: $DkPS$

```
Add the shortest path P_1 into DkPS;
while DkPS < k do
    Identify the next shortest path P_i ;
    Boolean tag ← true;
    for each path P ∈ DkPS do
        if sim(P_i, P) ≥ δ then
            tag ← false;
            Break;
    if tag then
        Add P_i into DkPS;
return DkPS;
```

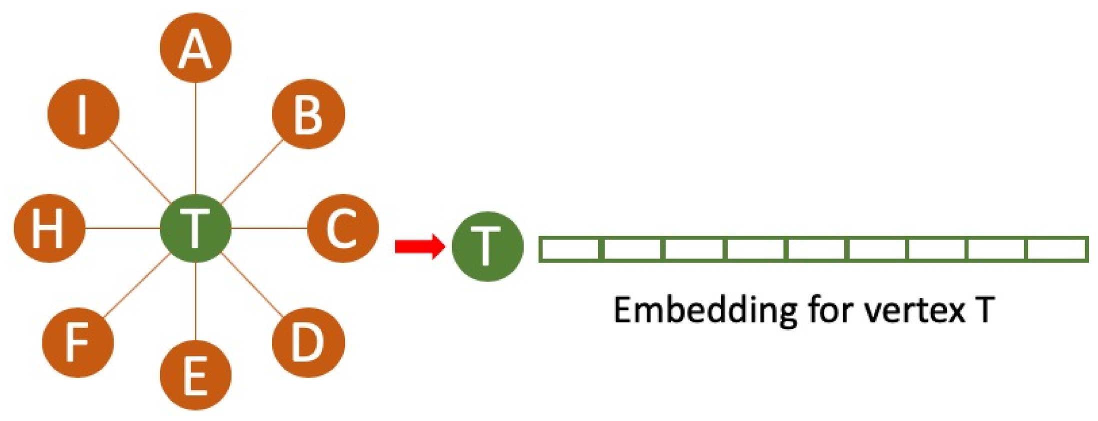
PathRank



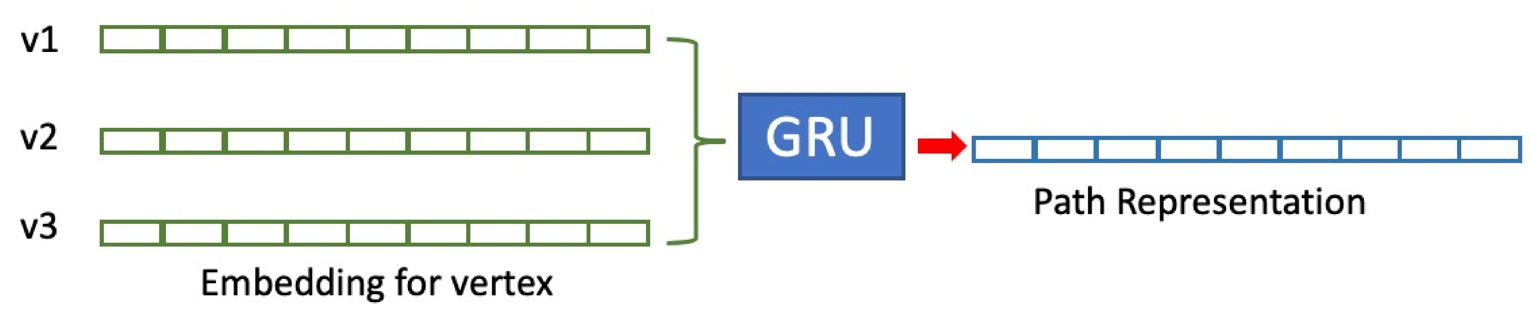
: *PathRank* Overview.

Vertex Embedding:

- Node2vec is used to embed road network and initialize vertex embedding layer.



Recurrent Neural Network (RNN):



Experiments

Experiments Setup

- **Road Network and Trajectories:** North Jutland, Denmark, 180 million GPS records from 183 vehicles.
- **Ground Truth Data:** For each trajectory P_T. 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 *WeightedJaccard*(P, P_T) as P's *ground truth* ranking score.
- **Evaluation Metrics:**
  - ⋆ Mean Absolute Error (MAE) and Mean Absolute Relative Error (MARE)
  - ⋆ Kendall Rank Correlation Coefficient (τ) and Spearman's Rank Correlation Coefficient (ρ)

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 1: Training Data Generation Strategies, *PR-A1*

Strategies	M	MAE	MARE	τ	ρ
TkDI	64	0.1433	0.2300	0.6638	0.7044
	128	0.1168	0.1875	0.6913	0.7330
D-TkDI	64	0.1140	0.1830	0.6959	0.7346
	128	**0.0955**	**0.1533**	**0.7077**	**0.7492**

Table 2: Training Data Generation Strategies, *PR-A2*

Strategies	M	MAE	MARE	τ	ρ
TkDI	64	0.1163	0.1868	0.6835	0.7256
	128	0.1130	0.1814	0.7082	0.7481
D-TkDI	64	0.0940	0.1509	0.7144	0.7532
	128	**0.0855**	**0.1373**	**0.7339**	**0.7731**