Empowering A∗ Search Algorithms with Neural Networks for Personalized Route Recommendation

Jingyuan Wang[*], Ning Wu[*]
Beijing Advanced Innovation Center for BDBC, School of Computer Science and Engineering, Beihang University, Beijing, China
{jywang,wuning}@buaa.edu.cn

Wayne Xin Zhao[†]
School of Information, Renmin University of China
Beijing, China
batmanfly@gmail.com

Fanzhang Peng, Xin Lin
MOE Engineering Research Center of ACAT, School of Computer Science and Engineering, Beihang University
Beijing, China
{pengfanzhang,sweeneylin}@buaa.edu.cn

KDD 2019

张峻伟
2019/7/5

Introduction

- ***Personalized Route Recommendation (PRR)***

Given the road network, PRR aims to generate user-specific route suggestions on instant queries about the path planning from a source to a destination

Pathfinding problem on graphs

- Heuristic search algorithms
- Machine learning methods

Is there a principled way to combine the merits of both kinds of approaches

The main idea of our solution is to **automatically learn the cost functions in A∗ algorithms**, which is the key of heuristic search algorithms.

- ***The Challenge***

  - ◻ How to define a suitable form for the cost in the PRR task.
  - ◻ How to design effective models for implementing cost functions with different purposes, and unify different cost functions for deriving the final cost.
  - ◻ How to utilize rich context or constraint information for improving the task performance.

This paper uses neural networks for improving *A** algorithm in the PRR task. It is able to **automatically learn the cost functions** without handcrafting heuristics. It is able to effectively **utilize context information and characterize complex trajectory characteristics**.

Preliminaries

- ***Definitions and Notations***

DEFINITION 1. ***Road Network***. *A road network is a directed graph* $\mathcal{G} = (\mathcal{L}, \mathcal{E})$, *where* $\mathcal{L}$ *is a vertex set of locations and* $\mathcal{E} \subset \mathcal{L} \times \mathcal{L}$ *is an edge set of road segments. A vertex* $l_i \in \mathcal{L}$ (i.e., *a location*) *represents a road junction or a road end. An edge* $e_{l_i,l_j} = \langle l_i, l_j \rangle \in \mathcal{E}$ *represents a directed road segment from vertex* l_i *to vertex* l_j.

DEFINITION 2. ***Route***. *A route* (a.k.a., *a path*) p *is an ordered sequence of locations connecting the source location* l_s *with the destination location* l_d *with* m *intermediate locations,* i.e., $p : l_s \rightarrow l_1 \rightarrow \ldots \rightarrow l_m \rightarrow l_d$, *where each pair of consecutive locations* $\langle l_i, l_{i+1} \rangle$ *corresponds to a road segment* $e_{l_i, l_{i+1}}$ *in the road network.*

Due to instrumental inaccuracies, the sampled trajectory points may not be well aligned with the locations in L. So the paper preforms the procedure of map matching for aligning trajectory points with locations in L

DEFINITION 3. ***Trajectory***. *A trajectory t is a time-ordered sequence of m locations (after map matching) generated by a user,* i.e., $t : \langle l_1, b_1\rangle \rightarrow \langle l_2, b_2\rangle \rightarrow \ldots \rightarrow \langle l_m, b_m\rangle$, *where b_i is the visit timestamp for location l_i.*

DEFINITION 4. ***Query***. *A query q is a triple $\langle l_s, l_d, b\rangle$ consisting of source location l_s, destination location l_d and departure time b.*

DEFINITION 5. ***Personalized Route Recommendation (PRR)***. *Given a dataset $\mathcal{D}$ consisting of historical trajectories, for a query $q : \langle l_s, l_d, b\rangle$ from user $u \in \mathcal{U}$, we would like to infer the most possible route p^* from l_s to l_d made by user u, formally defined as solving the optimal path with the highest conditional probability:*

$$p^* = \arg\max_p Pr(p|q, u, \mathcal{D}). \tag{1}$$

A heuristic A∗ solution for PRR

- ***Review of A∗ Algorithm***

$$f(n) = g(n) + h(n), \tag{2}$$

where $g(n)$ is the cost of the path from the source to n (we call it *observable cost* since the path is observable), and $h(n)$ is an estimate of the cost required to extend the future path to the goal (we call it *estimated cost* since the actual optimal path is unknown). The key part of A^* is the setting of the heuristic function $h(\cdot)$, which has an important impact on the final performance.

A Simple A^*-based Approach for PRR. Considering our task, the goal is to maximize the conditional probability of $\Pr(p|q, u, \mathcal{D})$. We can equally minimize its negative log: $-\log \Pr(p|q, u, \mathcal{D})$. Given a possible path $p : l_s \to l_1 \to l_2 \cdots \to l_m \to l_d$, consisting of m intermediate locations, we can factorize the path to compute its cost according to the chain rule in probability in the form of

the first-order Markov assumption

$$-\log \Pr(p|q, u, \mathcal{D}) = -\sum_{i=0}^{m} \log \Pr(l_{i+1}|l_s \to l_i, q, u), \tag{3}$$

https://blog.csdn.net/pipisorry/article/details/46618991
https://blog.csdn.net/zhulichen/article/details/78786493

● *The Challenge of A*-based approach*

- ☐ *A** algorithm is a general framework in which cost functions have to be heuristically set. It is difficult to incorporate varying context information.
- ☐ The cost function usually relies on the heuristic computation or estimation, which is easy to suffer from data sparsity.
- ☐ The PRR task is challenging, and a simple heuristic search strategy may not be capable of performing effective pathfinding in practice.

The NASR model

- *Model Overview*

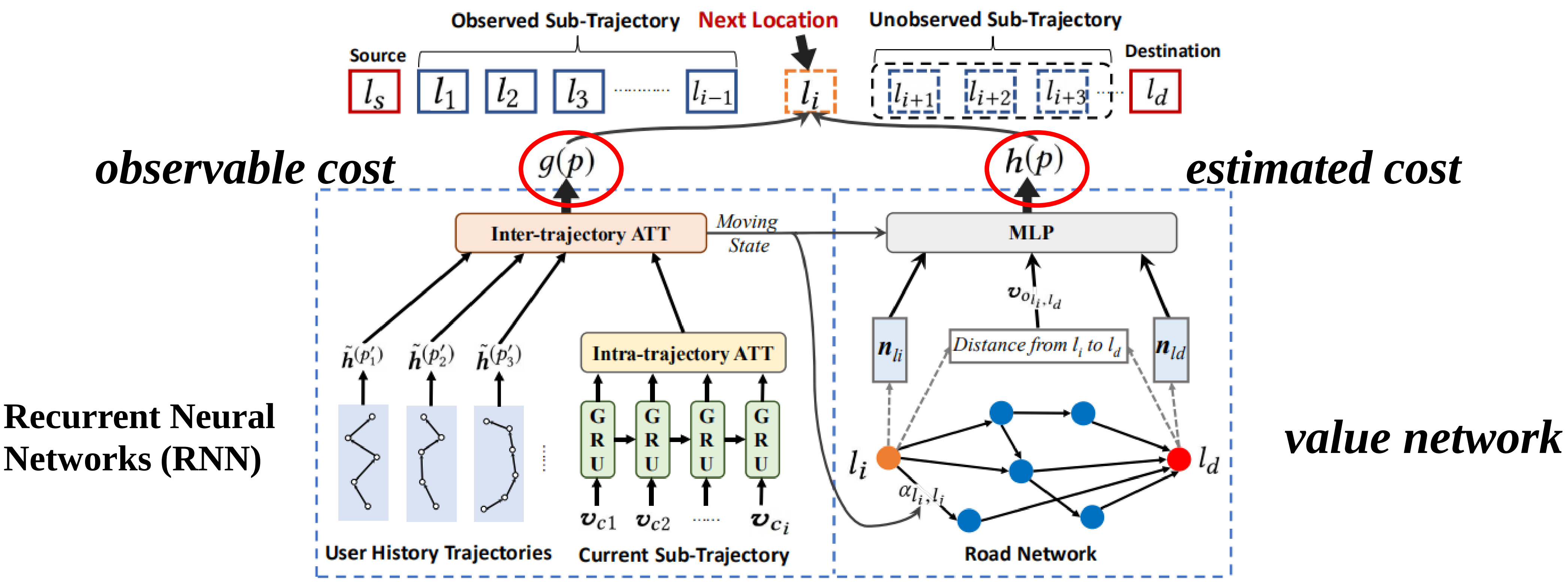
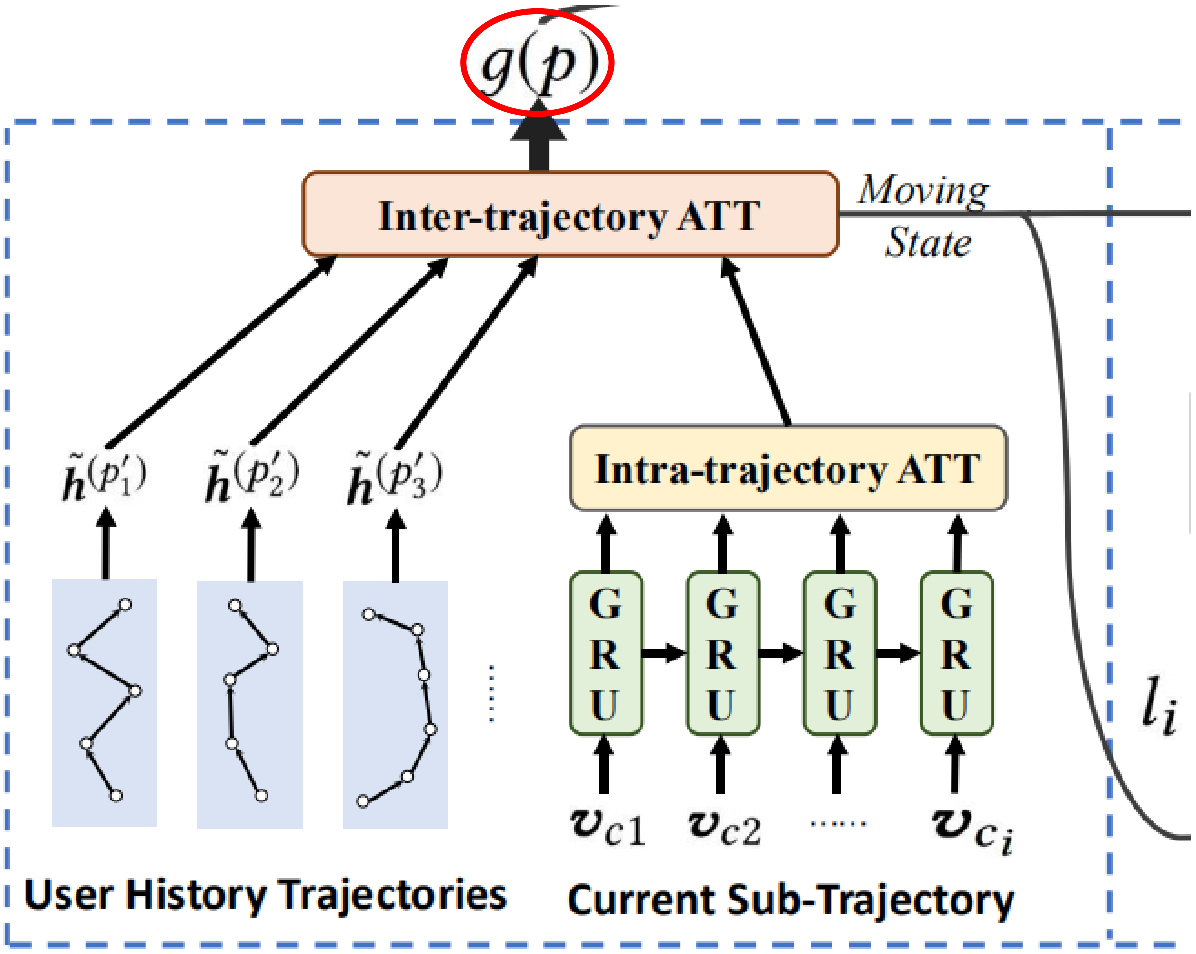


Figure 1: The overall architecture of the NASR model. $g(\cdot)$ **learns the cost from the source to a candidate location, called** ***observable cost***; $h(\cdot)$ **predicts the estimated cost from a candidate location to the destination, called** ***estimated cost.***

● *Modeling the Observable Cost with RNN*

How to effectively learn the conditional transition probabilities $\Pr(l_{k+1}|l_s \rightarrow l_k, q, u)$



Location information

Concatenation

$$v_{x_i} = v_u \| v_{l_i} \| v_{di(b_i)} \| v_{hi(b_i)},$$

User information

Temporal information

Encoding the Observed Sub-Trajectory with RNN

model the trajectory characteristics of users' moving behaviors

$$h_i^{(p)} = \mathrm{GRU}(v_{x_i}, h_{i-1}^{(p)}),$$

the hidden vector produced by the GRU network

the context vector

the moving state of a user at the i-th time step

Enhanced Moving States with Attention Mechanism

two types of attention to improve the learning of moving state by leveraging data dependence

Intra-Trajectory Attention

compute the attention between locations in the same trajectory

$$\tilde{\boldsymbol{h}}_i^{(p)} = \sum_{k=1}^{i} \operatorname{att}\left(\boldsymbol{h}_i^{(p)}, \boldsymbol{h}_k^{(p)}\right) \cdot \boldsymbol{h}_k^{(p)},$$

$$\operatorname{att}\left(\boldsymbol{h}_i^{(p)}, \boldsymbol{h}_k^{(p)}\right) = \frac{\alpha_{i,k}}{\sum_{k'=1}^{i} \exp(\alpha_{i,k'})},$$

$$\alpha_{i,k} = \boldsymbol{w}_1^\top \cdot \tanh\left(\boldsymbol{W}_1 \cdot \boldsymbol{h}_k^{(p)} + \boldsymbol{W}_2 \cdot \boldsymbol{h}_i^{(p)}\right),$$

the parameter vector or matrices to learn

the state representation of the last location for encoding the entire sub-trajectory

Inter-Trajectory Attention

attend it to each of the other historical trajectories as

$$\boldsymbol{h}^{(p)} = \sum_{p' \in \mathcal{P}^u} \text{att}\left(\tilde{\boldsymbol{h}}^{(p)}, \tilde{\boldsymbol{h}}^{(p')}\right) \cdot \tilde{\boldsymbol{h}}^{(p')},$$

is similar to Eq. (8)

denotes the set of historical trajectories generated by *u*

☐ *Observable Cost Computation with the Road Network Constraints.*

$$\Pr(l_i | l_s \to l_{i-1}, q, u) = \frac{\exp\left(z(p^{l_s \to l_i})\right)}{\sum_{l' \in \mathcal{L}_{l_{i-1}}} \exp\left(z(p^{l_s \to l'})\right)},$$

$$g(l_s \to l_i) = -\sum_{j=2}^{i} \log \Pr\left(l_j | l_s \to l_{j-1}, q, u\right).$$

linear transformation function $z(p) = \boldsymbol{w}_2^\top \cdot \boldsymbol{h}^{(p)}$

parameter vector

$$Loss_1 = \sum_{u \in \mathcal{U}} \sum_{p \in \mathcal{P}^u} g(p).$$

● *Modeling the Estimated Cost with* ***Value Networks***

build the value network on **top of an improved graph attention network** with useful context information

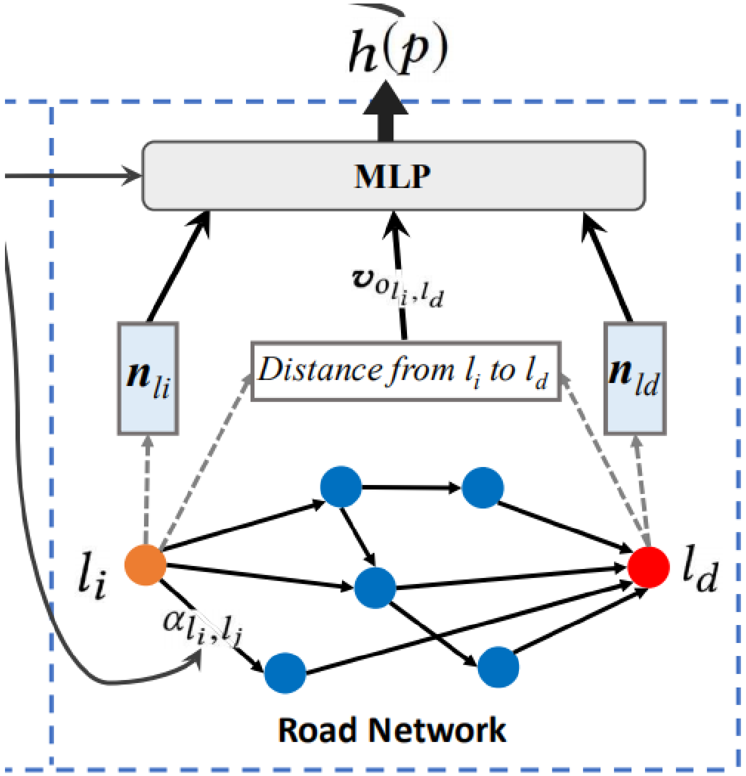
☐ *Improved Graph Attention Networks for Road Networks*

$$\boldsymbol{N}^{(z+1)} = \mathrm{GNN}\left(\boldsymbol{N}^{(z)}\right)$$

Graph ATTention network (GAT)

denotes the matrix consisting of node representations at the *z*-th iteration

$$\boldsymbol{n}_{l_k}^{(0)} = \boldsymbol{v}_{l_k}$$



◻ Context-aware Graph Attention

a node has a larger impact on a nearby node than a faraway node

discretize the distance o_{l_i,l_j} between nodes l_i and l_j into consecutive value bins

$$\alpha_{l_j,l_{j'}} = \frac{\exp\left(\boldsymbol{w}_2^\top \cdot \left(\boldsymbol{W}_3\boldsymbol{n}_{l_j} + \boldsymbol{W}_4\boldsymbol{n}_{l_{j'}} + \boldsymbol{W}_5\boldsymbol{h}^{(p)} + \boldsymbol{W}_6\boldsymbol{v}_{o_{l_j,l_{j'}}}\right)\right)}{\sum_{k\in\mathcal{L}_{l_j}} \exp\left(\boldsymbol{w}_2^\top \cdot \left(\boldsymbol{W}_3\boldsymbol{n}_{l_j} + \boldsymbol{W}_4\boldsymbol{n}_{l_k} + \boldsymbol{W}_5\boldsymbol{h}^{(p)} + \boldsymbol{W}_6\boldsymbol{v}_{o_{l_j,l_k}}\right)\right)}, \quad (14)$$

learnable parameters

The embedding vector for the discretized distance value

$$\boldsymbol{n}_{l_i}^{(z+1)} = \Big\Vert_{a=1}^{A} \mathrm{relu}\left(\sum_{l_j\in\mathcal{L}_{l_i}} \alpha_{l_i,l_j}^{(a)} \boldsymbol{W}^{(a)} \boldsymbol{n}_{l_j}^{(z)}\right)$$

☐ Predicting the Estimated Cost with MLP

We use a Multi-Layer Perceptron component to infer the cost from the candidate location *li* to the destination *ld*

$$h\left(l_i \rightarrow l_d\right) = \mathrm{MLP}\left(\boldsymbol{h}^{(p)}, \boldsymbol{n}_{l_i}, \boldsymbol{n}_{l_d}, \boldsymbol{v}_{o_{l_i, l_d}}\right)$$

the node representations li and ld   embedding for their spatial distance

☐ Temporal Difference Learning for the Estimated Cost

the optimal sub-route from *li* to *ld*, which is a multi-step decision process and difficult to be directly

Markov Decision Process

a state consists of the information of the query q and the current sub-sequence $ls \rightarrow li\rangle$ 1; an action is to select a location li to extend in the route. The standard MDP aims to maximize the future reward, while our task aims to minimize the future cost.

$$c_i = -\log \Pr(l_i | l_s \rightarrow l_{i-1}, q, u),$$

$$\Pr(l_i | l_s \rightarrow l_{i-1}, q, u) = \frac{\exp\left(z(p^{l_s \rightarrow l_i})\right)}{\sum_{l' \in \mathcal{L}_{l_{i-1}}} \exp\left(z(p^{l_s \rightarrow l'})\right)}$$

our purpose is to estimate the future cost

we adopt a popular value-based learning method for optimizing the value networks

$$h(l_i \rightarrow l_d) = \sum_{j=i+1}^{T} \gamma^{j-i-1} c_j$$

discount rate to discounting future cost to the current and T is the timestamp arriving at ld.

$$y_{l_i} = \gamma^n h(l_{i+n} \rightarrow l_d) + \sum_{j=i+1}^{i+n} \gamma^{j-i-1} c_{l_j}$$

the estimated cost using the temporal difference approach

$$\delta_{l_i} = \sum_{i=1}^{T-1} \left\| h(l_i \rightarrow l_d) - y_{l_i} \right\|^2$$

all the observed trajectories over all users

$$Loss_2 = \sum_{u \in \mathcal{U}} \sum_{p \in \mathcal{P}^u} \sum_{l_i \in p} \delta_{l_i}$$

https://blog.csdn.net/LagrangeSK/article/details/81239518

- ***Model Analysis and Learning***
  - the first component utilizes RNNs to characterize the currently generated sub-trajectory for learning observable cost
  - the second component incorporates a value network to predict the estimated cost to arrive at the destination
  - Finally, the two cost values are summed as the final evaluation cost of a candidate location

Merits

- it does not require to manually set functions with heuristics, but automatically learns the functions from data.
- it can utilize various kinds of context information and capture more complicated personalized trajectory characteristics.
- it is able to coordinate and integrate the two components by sharing useful information or parameters in a principled way

- To learn the model parameters, we first pre-train the RNN component.
- Then, we jointly learn the two components using alternative optimization by iterating over the trajectories in training set.
- After model learning, we follow the search procedure of $A*$ algorithm for the PRR task with the evaluation cost computed by our model.

EXPERIMENT

- **Experimental Settings**

Instead of reporting the overall performance on all test trajectories, we generate three types of queries short (10 to 20 locations), medium (20 to 30 locations) and long (more than 30 locations)

- *Datasets*

we use three real-world trajectory datasets. The ***Beijing taxi* trajectory** data is sampled every minute, while the ***Beijing bicycle* dataset** is sampled every 10 seconds. The **Porto taxi dataset** is originally released for a Kaggle trajectory prediction competition with a sampling period of 15 seconds.

- *Evaluation Metrics*

Precision, Recall and F1-score

$$Precision = \frac{|p \cap p'|}{|p'|}, \quad Recall = \frac{|p \cap p'|}{|p|} \quad F1 = \frac{2*P*R}{P+R}$$

Baselines

•*RICK* [30]: It builds a routable graph from uncertain trajectories, and then answers a users online query (a sequence of point locations) by searching top-k routes on the graph.

•*MPR* [4]: It discovers the most popular route from a transfer network based on the popularity indicators in a breadth-first manner.

•*CTRR* [6]: It proposes collaborative travel route recommendation by considering user's personal travel preference.

•*STRNN* [16]: Based on RNNs, it models local temporal and spatial contexts in each layer with transition matrices for different time intervals and geographical distances.

•*DeepMove* [9]: It is a multi-modal embedding RNN that can capture the complicated sequential transitions by jointly embedding the multiple factors that govern the human mobility.

Among these baselines, RICK and MPR are heuristic search based methods, CTRR is a machine learning method, and STRNN and DeepMove are deep learning methods. The parameters in all the models have been optimized using the validation set.

Parameter Settings

The parameters in all the models have been optimized using the validation set.

Results and Analysis

Table 1: Performance comparison using four metrics on three datasets. All the results are better with larger values except the EDT measure. With paired t-test, the improvement of the NASR over all the baselines is significant at the level of 0.01.

Datasets	Metric	Precision						Recall					
	Length	RICK	MPR	CTRR	STRNN	DeepMove	NASR	RICK	MPR	CTRR	STRNN	DeepMove	NASR
Beijing Taxi	Short	0.712	0.347	0.558	0.491	0.742	**0.821**	0.723	0.372	0.164	0.384	0.756	**0.848**
	Medium	0.638	0.253	0.276	0.446	0.642	**0.757**	0.651	0.261	0.067	0.350	0.654	**0.773**
	Long	0.586	0.169	0.194	0.359	0.562	**0.684**	0.589	0.173	0.045	0.214	0.575	**0.709**
Porto Taxi	Short	0.697	0.359	0.701	0.442	0.721	**0.804**	0.705	0.381	0.358	0.372	0.726	**0.832**
	Medium	0.622	0.271	0.416	0.403	0.619	**0.729**	0.634	0.293	0.106	0.326	0.628	**0.754**
	Long	0.565	0.184	0.305	0.340	0.547	**0.657**	0.578	0.198	0.036	0.218	0.568	**0.671**
Beijing Bicycle	Short	0.652	0.303	0.587	0.559	0.673	**0.788**	0.670	0.313	0.272	0.330	0.685	**0.802**
	Medium	0.568	0.217	0.603	0.461	0.582	**0.715**	0.574	0.226	0.142	0.304	0.589	**0.724**
	Long	0.503	0.129	0.613	0.297	0.487	**0.641**	0.519	0.139	0.045	0.206	0.492	**0.663**

Datasets	Metric	F1-score						EDT					
	Length	RICK	MPR	CTRR	STRNN	DeepMove	NASR	RICK	MPR	CTRR	STRNN	DeepMove	NASR
Beijing Taxi	Short	0.717	0.359	0.253	0.431	0.749	**0.834**	4.594	8.287	9.082	7.551	4.362	**3.376**
	Medium	0.644	0.257	0.108	0.392	0.648	**0.765**	8.273	16.321	23.110	14.725	8.730	**5.728**
	Long	0.587	0.171	0.073	0.268	0.568	**0.703**	11.283	25.873	27.493	22.705	12.059	**8.314**
Porto Taxi	Short	0.701	0.370	0.474	0.404	0.723	**0.818**	4.801	8.104	6.935	8.790	4.496	**3.563**
	Medium	0.628	0.282	0.169	0.360	0.623	**0.741**	8.619	15.032	18.294	13.368	8.930	**5.949**
	Long	0.571	0.191	0.065	0.266	0.557	**0.687**	11.379	21.349	31.745	19.603	12.297	**8.572**
Beijing Bicycle	Short	0.661	0.308	0.372	0.414	0.679	**0.795**	5.183	8.924	7.784	7.092	4.629	**3.719**
	Medium	0.571	0.221	0.229	0.367	0.585	**0.720**	8.972	17.497	20.966	14.503	9.039	**6.253**
	Long	0.511	0.134	0.084	0.243	0.489	**0.671**	11.891	22.028	57.997	21.324	12.692	**8.794**

(1) heuristic search methods;

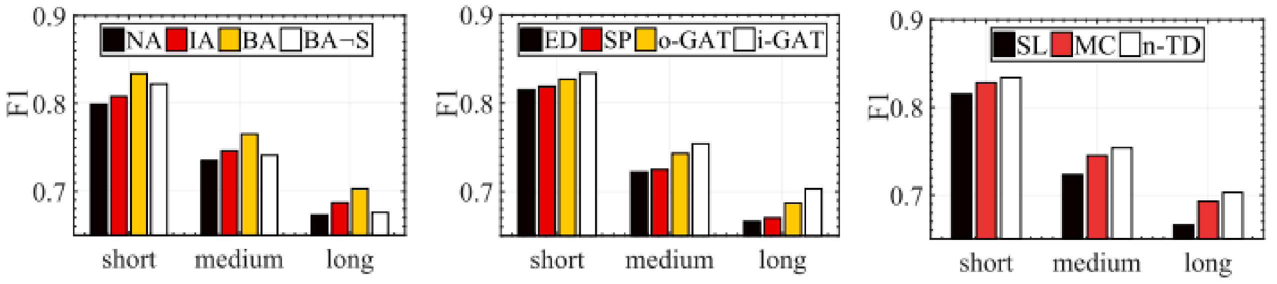
(2) the matrix factorization based method CTRR does not perform better than RICK and MPR

(3) deep learning method DeepMove performs very well among all the baselines.

● Detailed Analysis on Our Model NASR

□ *Effect of the RNN Component*

examine the effect of the RNN component with different variants： without attention (NA), using only intra-trajectory attention (IA) and using both intra- and intertrajectory attention (BA)



(a) Examining the RNN component.

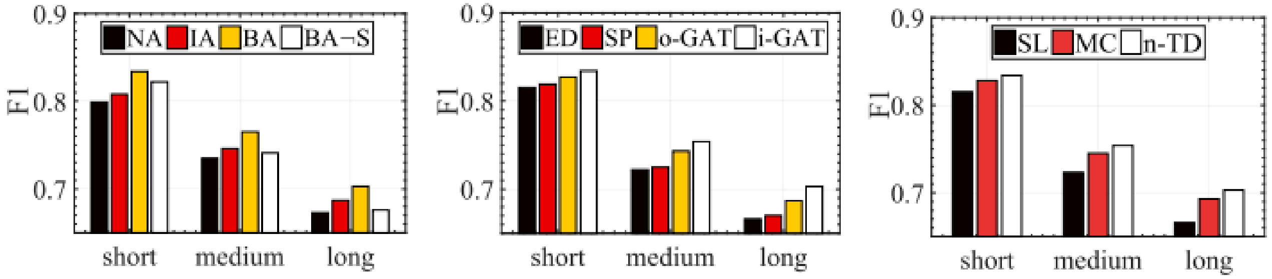
(b) Examining the value network.

(c) Examining the TD learning method.

Figure 2: Detailed analysis of our model on the dataset of Beijing taxi using F1 measure.

Effect of the Value Network.

four variants for the value network as comparisons, including (1) ED using Euclid distance as heuristics, (2) SP using the scalar product between the embeddings of the candidate and destination locations, (3) o-GAT using the original implementation of graph attention networks, and (4) i-GAT using our improved GAT by incorporating context information.



(a) Examining the RNN component. (b) Examining the value network. (c) Examining the TD learning method.

Figure 2: Detailed analysis of our model on the dataset of Beijing taxi using F1 measure.

Effect of Temporal Difference Learning Method

(1)*SL* which directly learns the actual distance between the candidate location and the destination in a supervised way, (2) *MC* which applies Monte Carlo method to generate sampled sequences and trains the model with the cost of these sampled sequences, (3) *n-TD* which uses a TD step number of 5

- **Qualitative Analysis**

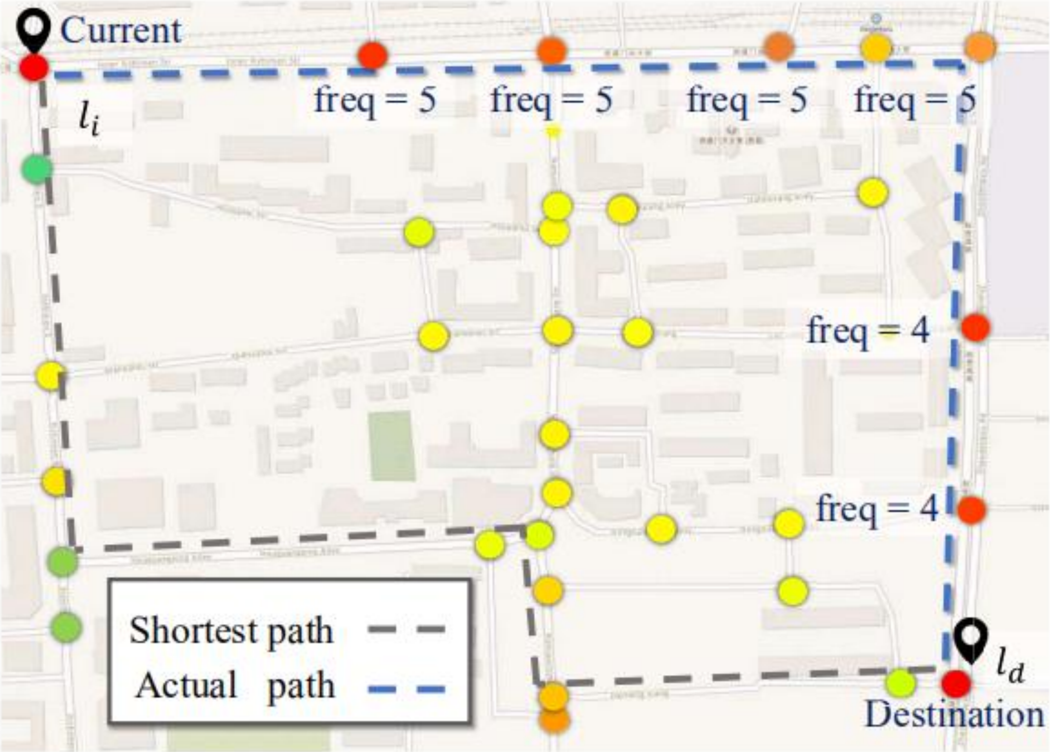


Figure 3: Visualization of the learned association scores using improved graph attention networks. The colored circles denote locations in the road network. A darker color indicates a larger importance degree *w.r.t.* current location l_i and destination l_d. "*freq*" denotes the visit frequency by the user in historical trajectories.

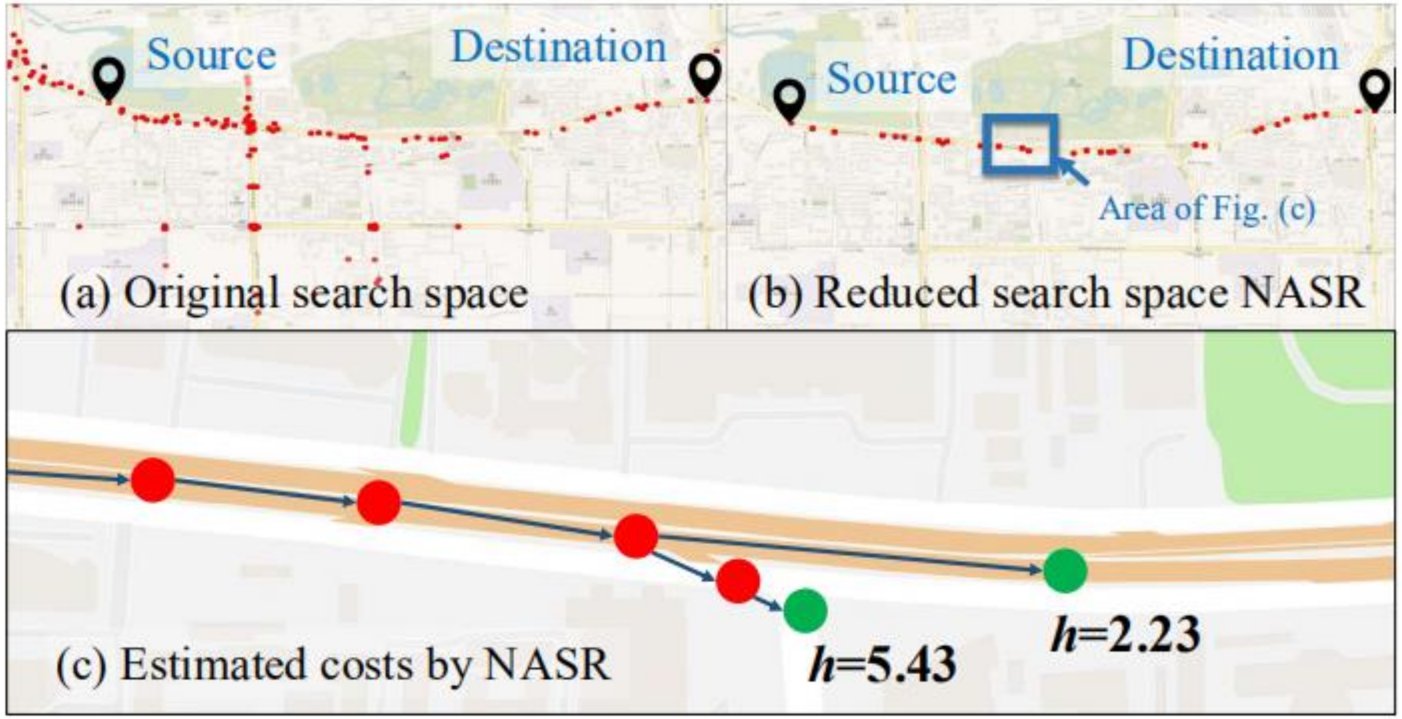


Figure 4: Visualization of the search procedure with the estimated costs by the NASR model. In (c), red points have been already explored and green points are candidate locations to extend in A^* search algorithm.

CONCLUSION AND FUTURE WORK

- The paper first presented a simple A∗ solution for solving the PRR task, and formally defined the suitable form for the search cost
- The author set up two components to learn the two costs respectively, i.e., the RNN component for д(·) and the value network for $h(\cdot)$
- The two components were integrated in a principled way for deriving a more accurate cost of a candidate location for search
- We constructed extensive experiments for verifying the effectiveness and robustness of the proposed model.
- we will consider extending our model to solve the PRR task without road networks

New Idea

- 将传统方法与深度神经网络结合
- 模型中各个部分的重要性的分析


thank
you