ACM SIGSPATIAL China 2017 @长沙

基于神经网络的轨迹建模
Modeling Trajectory with Recurrent Neural Networks

孙未未
复旦大学
2017-11-18

Motivation：

大量低价值密度的历史轨迹数据如何保存？

1. 概要（size小）
2. 通用（支持应用多）

对轨迹数据的分布进行建模

Modeling Trajectory with Recurrent Neural Networks (IJCAI 2017)

Hao Wu†, Ziyang Chen†, Weiwei Sun†, Baihua Zheng‡, Wei Wang†

†Fudan University, Shanghai, China

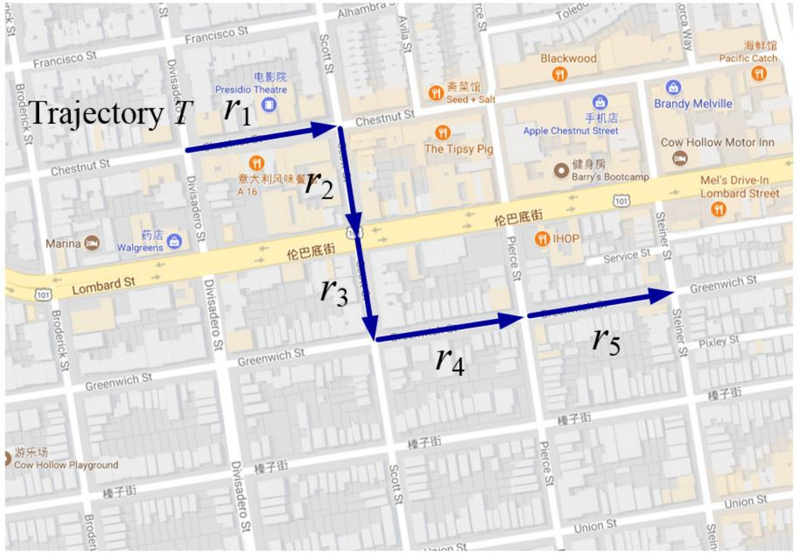
‡Singapore Management University, Singapore

FUDAN UNIVERSITY 1905

SMU SINGAPORE MANAGEMENT UNIVERSITY

Trajectory

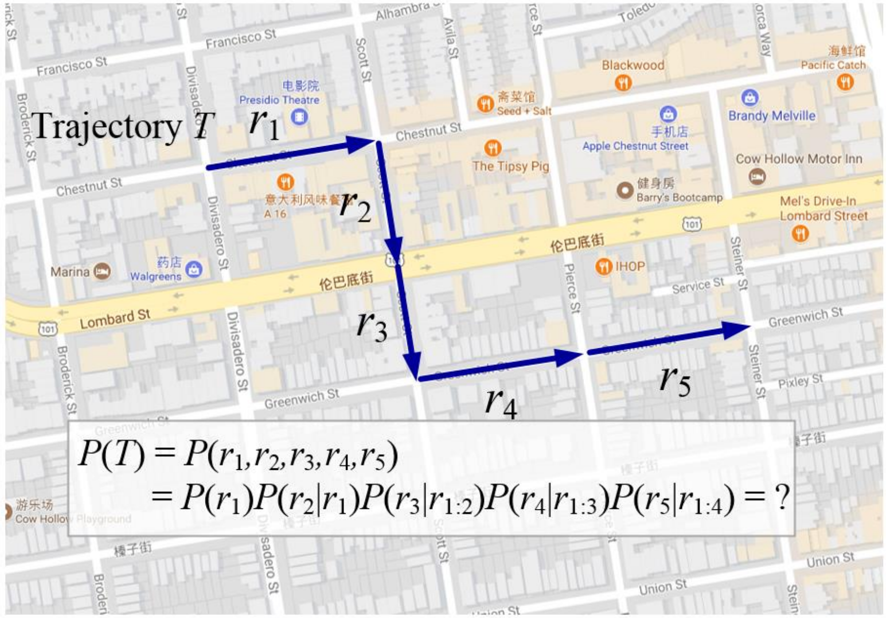
- [**Trajectory**] A (vehicle) trajectory is a sequence of edges in the road network $r_1 \to r_2 \to \cdots \to r_k$, where each two consecutive road segments in the trajectory are connected in the graph.



Trajectory modeling task

- **Objective**: Given a set of historical trajectories, build a probabilistic model to model the distribution of the trajectory data.

$$P(T) = P(r_1, r_2, \dots, r_k) = P(r_1)\prod_{i=1}^{k-1} P(r_{i+1}|r_{1:i})$$



Routing decision / transition probability

The probability to choose the next road given the fact of having travelled from r_1 to r_i

What can trajectory model do

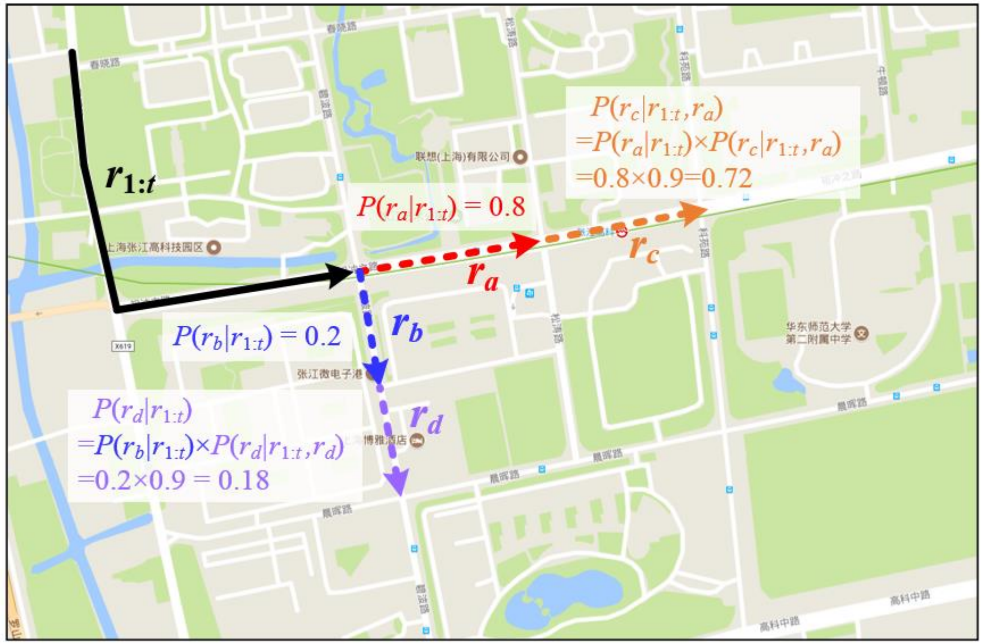
Applications:

- Trajectory prediction (turning / multi-step prediction)
- Trajectory Compression
- Navigation (route planning / recommendation)
- Trajectory recovery (reducing uncertainly)
- Outlier detection (greedy taxi driver)
- Generation (trajectory simulation)
- …

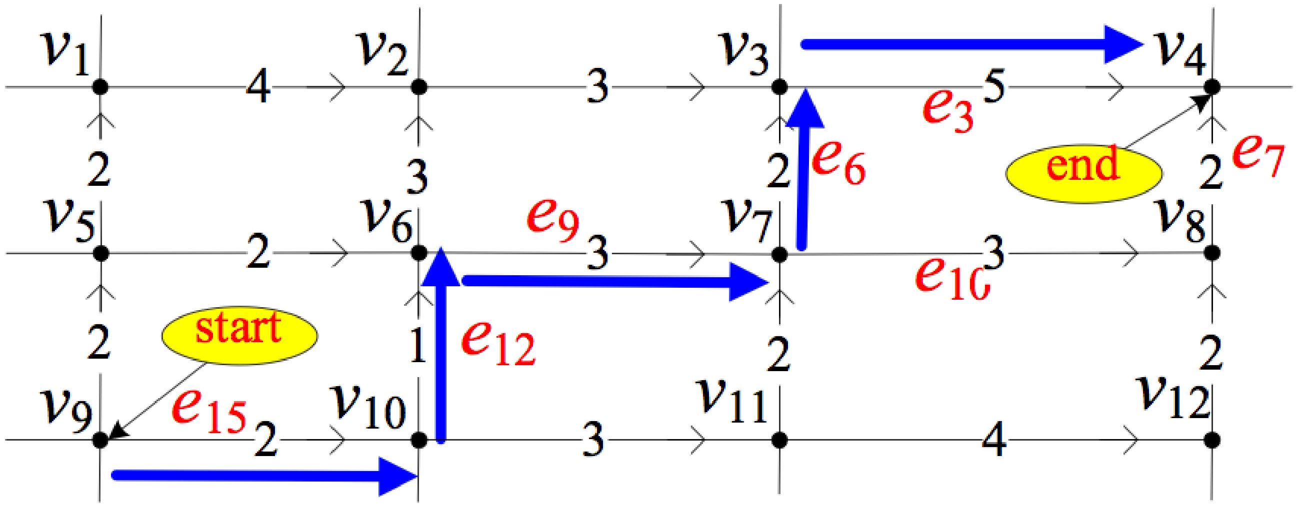
Trajectory prediction

- One-step transition probability $P(r_{i+1}|r_{1:i})$
- Iteratively multiply transition probabilities to get further steps' prediction

Example of two-step prediction



Trajectory Compression

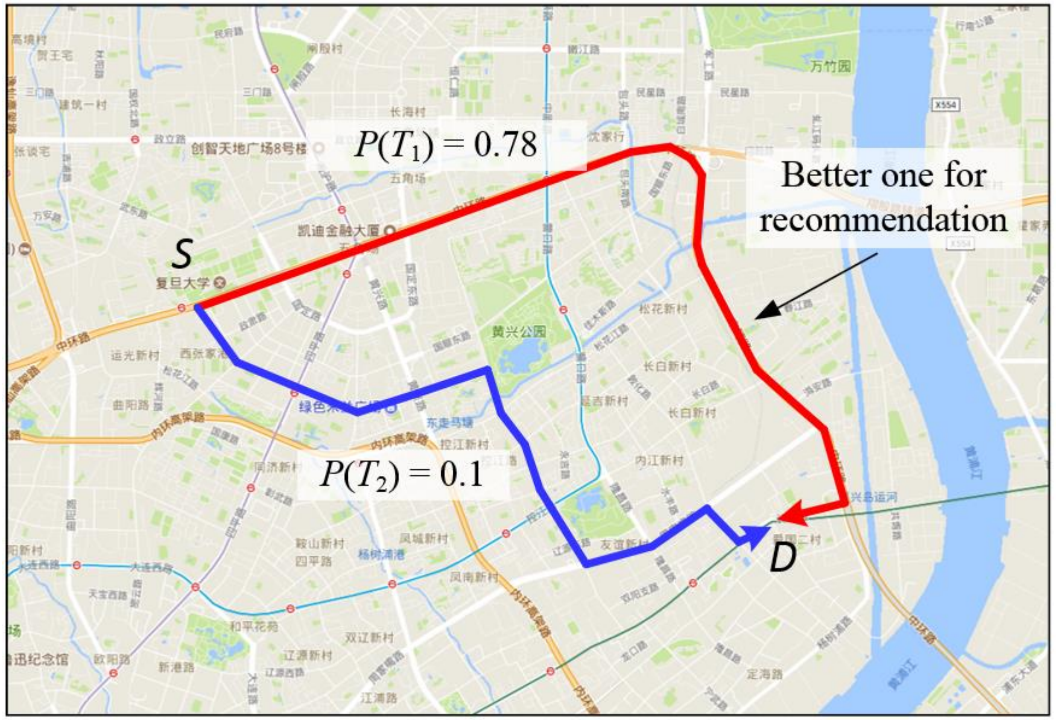


Edge sequence representation: $\langle e_{15}, e_{12}, e_9, e_6, e_3 \rangle$

Compressed representation: $\langle e_{15}, e_3 \rangle$

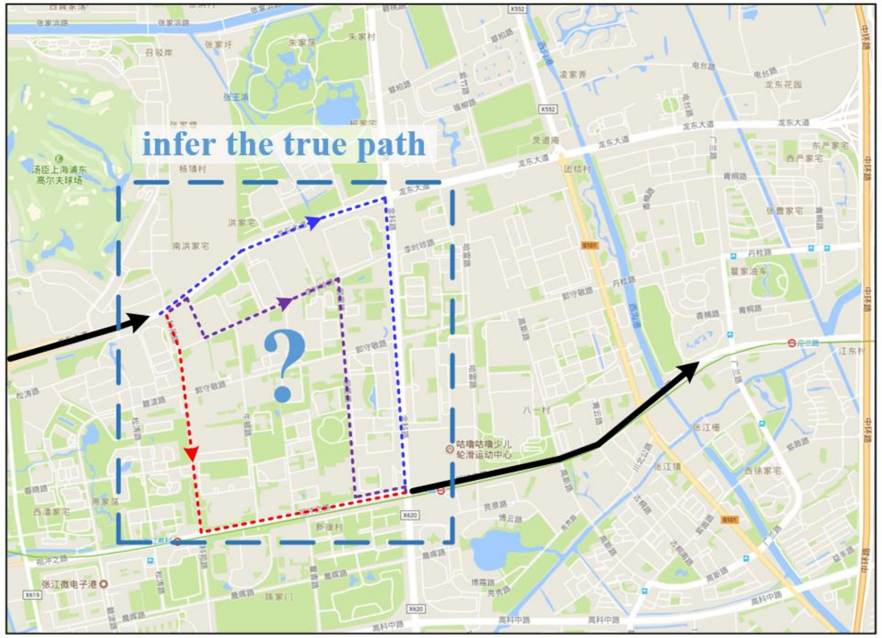
Route planning / recommendation

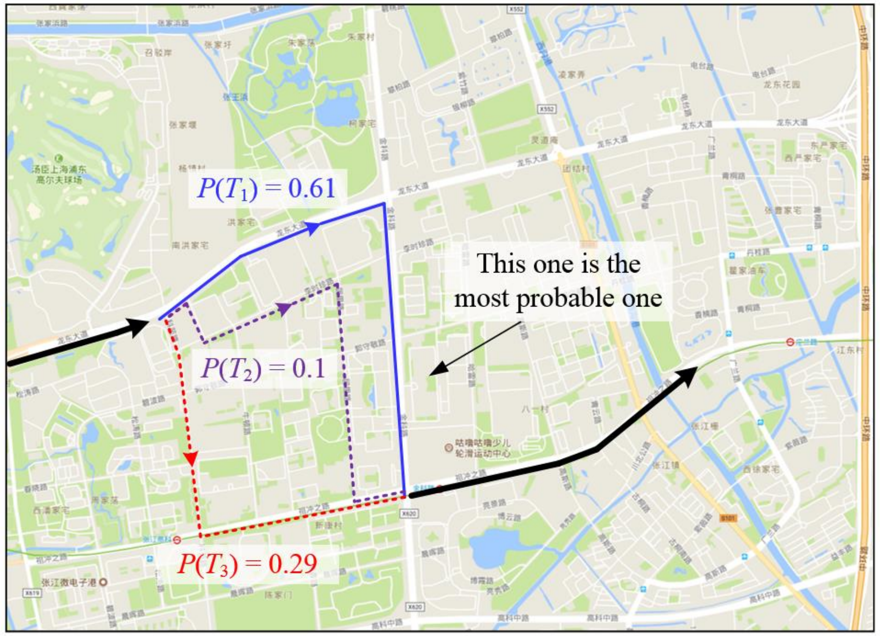
- A trajectory with higher probability indicates the path is preferred by many people in the historical data.



Trajectory recovery

- Missing path caused by
  - Low sampling rate
  - Signal loss (e.g., tunnel, broken device)
- Which will cause the data uncertainty
- Recover the missing part by the trajectory having the highest probability

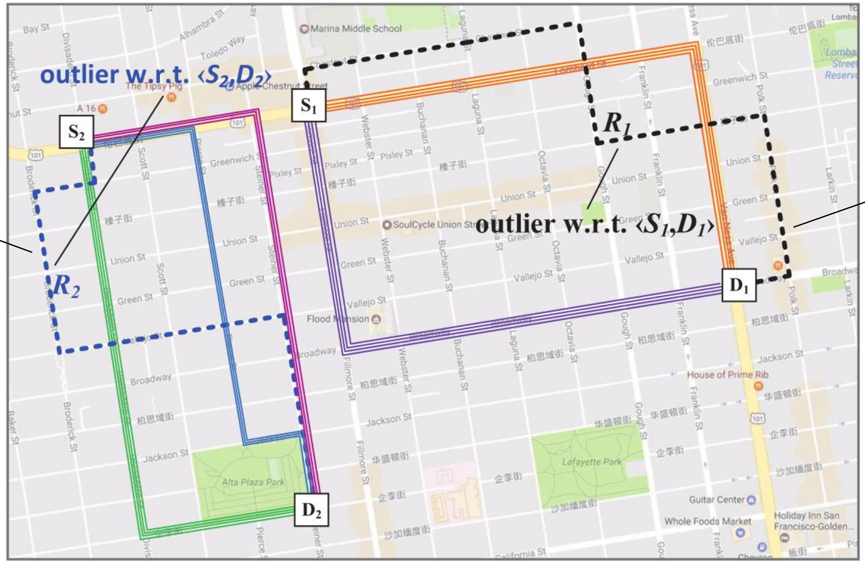




Outlier / Anomaly detection

- An outlier often has very low probability density
- Warn the passenger for paying attention to the greedy driver (potential outlier)

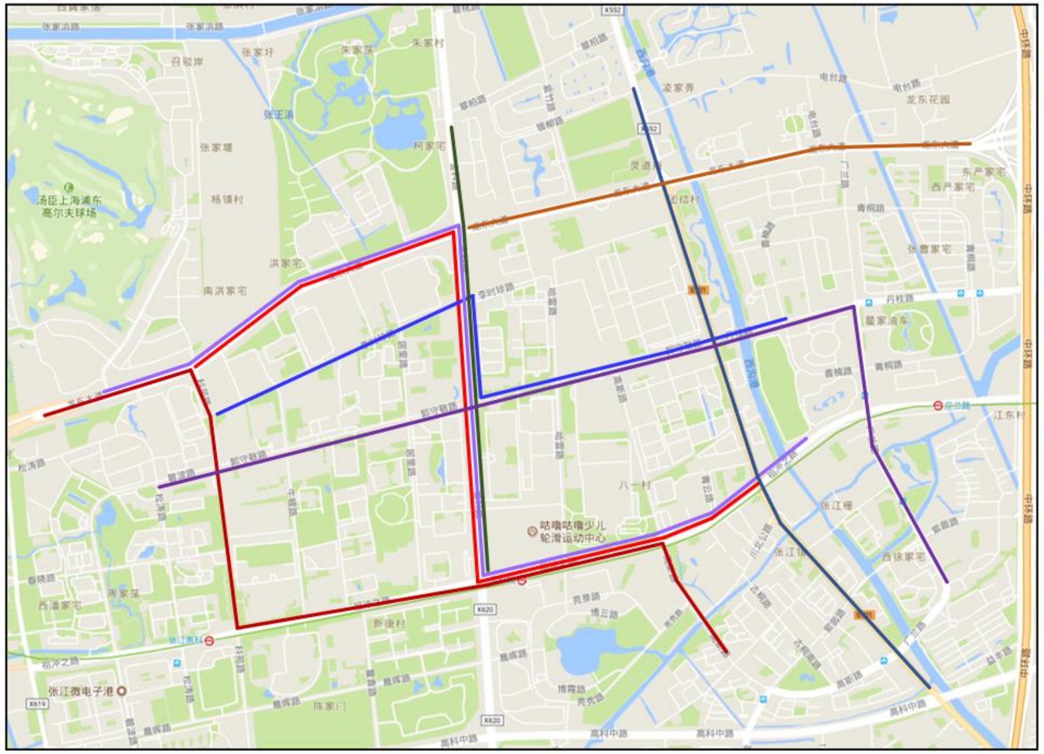
$P(R_1|S_1, D_1) = 0.0006$



$P(R_1|S_1, D_1) = 0.001$

Trajectory generation

- Generation task
- Trajectory data simulation

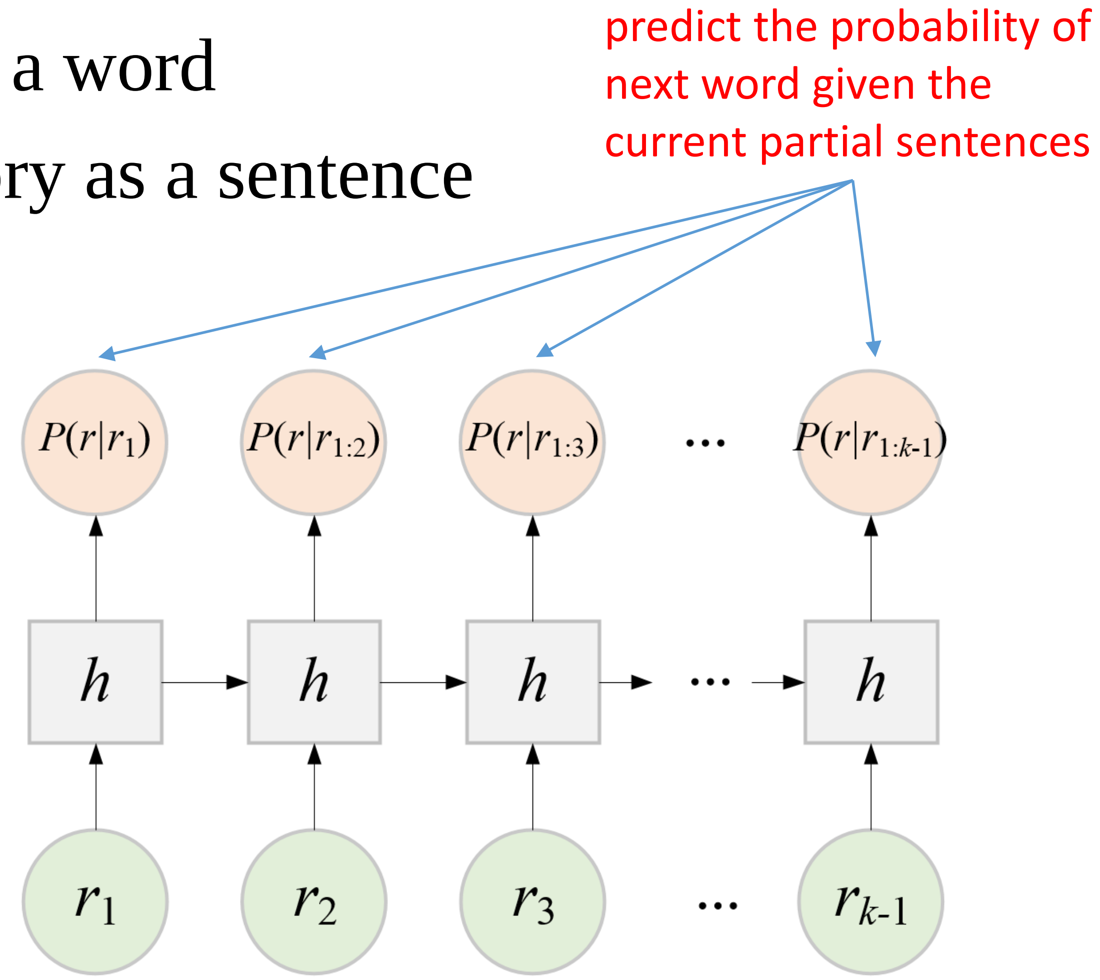


Existing works

- Markov chain / N-gram
  - Shallow model -> Fail to capture long-term dependency
  - Suffer data sparsity problem
- Inverse reinforcement learning
  - Bayesian IRL (***BIRL***) [Zheng and Ni, 2014] [Ramachandran and Amir, 2007]
    - Still Markov chain
  - Maximum entropy IRL (***MEIRL***) [Ziebart *et al.*, 2008a], [Ziebart *et al.*, 2008a]
    - Too few parameters / Insufficient model capacity

Neural Language model (RNN-based)

- Regard the road as a word
- Regard the trajectory as a sentence



Difference

- Language: transition *from word to word* is free
- Trajectory: only *adjacent roads* can be transited to

- Topological constraints on the predicted states

- Could RNN automatically learn such constraints?

Theorem for the limitation of vanilla RNN

1. A lower error ε (the summation of the assigned probability of all illegal states)
2. A larger number of roads $|E|$ (city scale)

will all increase the number of hidden units in RNN with the lower bound

$$\left[\frac{1}{2\|\omega\|_2} \cdot \log\left(\left(\frac{1}{\varepsilon} - 1\right)\left(\frac{|E|}{|\mathcal{S}^+|} - 1\right)\right)\right]^2$$

Which means:

- Harder to train
- Slower model
- Later convergence
- Larger memory consumption

Our Models

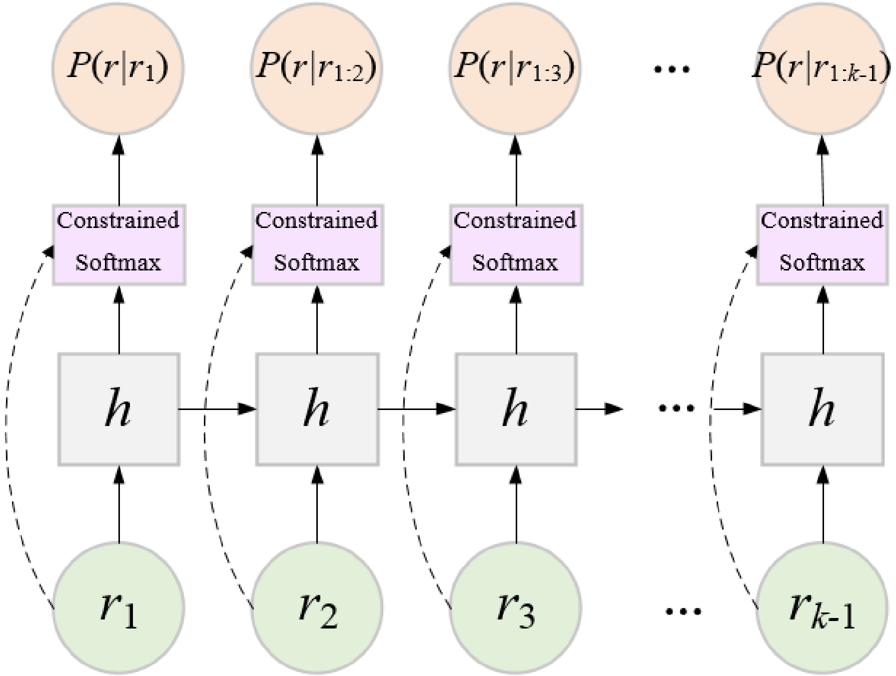
- Constraint State Space Recurrent Neural Networks (CSSRNN)
  - Less parameters
  - Robust to overfitting
  - Performs good if we no not have so much data
- Latent Prediction Information Recurrent Neural Networks (LPIRNN)
  - More parameters
  - Better representation power
  - Performs good if we have enough training data

Constrained State Space RNN(CSSRNN)

- Manually input the topological constraints into the model through the *constrained softmax.*

$$\mathcal{M}_{ij} = \begin{cases} 1 & \text{if } r_i \text{ can reach } r_j \\ 0 & \text{otherwise} \end{cases}$$

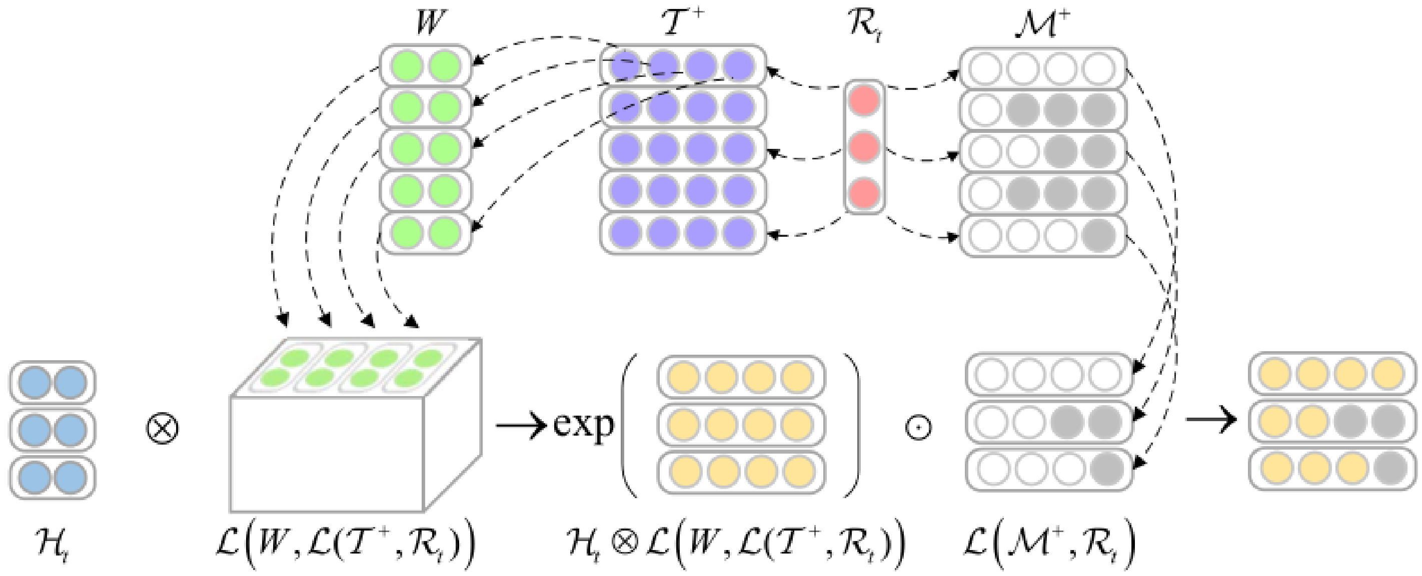
$$p(\tilde{r}_{t+1}|r_{1:t}) = \mathcal{C}(Wh_t + b, r_t) = \frac{\exp(Wh_t + b) \odot \mathcal{M}_i}{||\exp(Wh_t + b) \odot \mathcal{M}_i||_1}$$



The gradients w.r.t. illegal states will be blocked by the mask

Fast convergence!

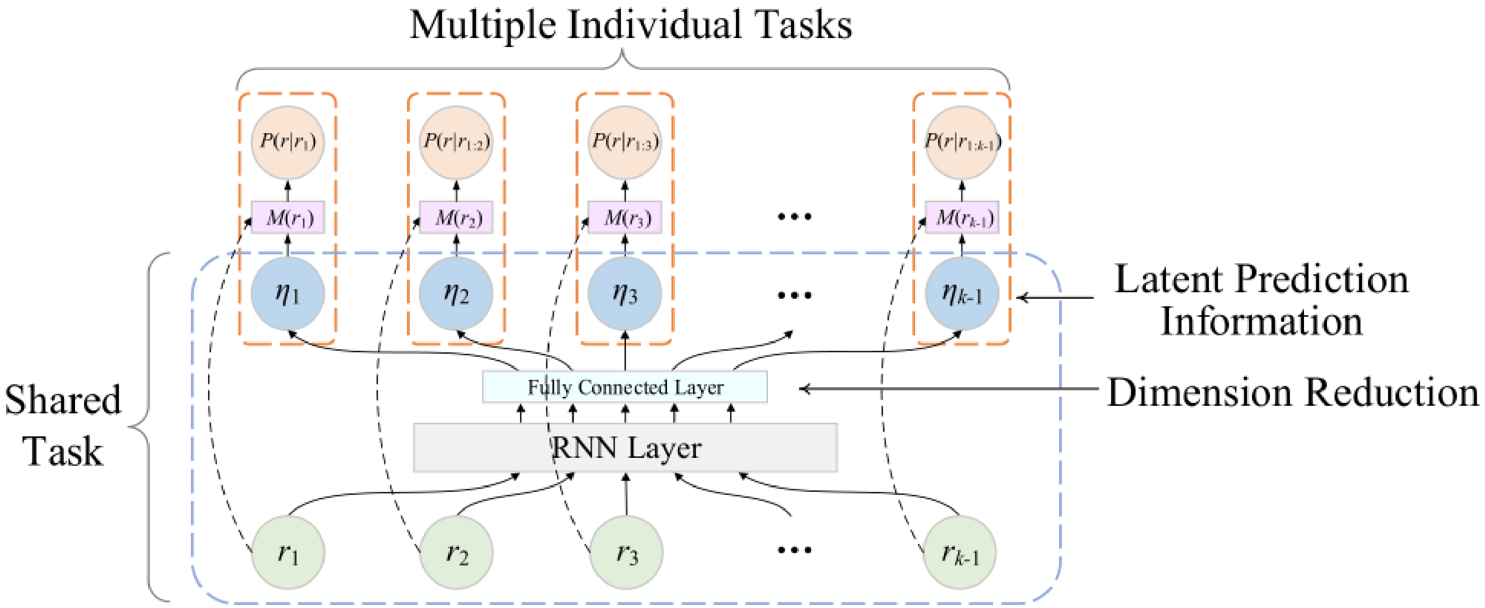
Potential speed up



$$\frac{\exp(\mathcal{H}_t \otimes \mathcal{L}(\mathcal{W}, \mathcal{L}(\mathcal{T}^+, \mathcal{R}_t)) \odot \mathcal{L}(\mathcal{M}^+, \mathcal{R}_t)}{\sum_{dim=2}(\exp(\mathcal{H}_t \otimes \mathcal{L}(\mathcal{W}, \mathcal{L}(\mathcal{T}^+, \mathcal{R}_t)) \odot \mathcal{L}(\mathcal{M}^+, \mathcal{R}_t))}$$

Latent Prediction Information RNN (LPIRNN)

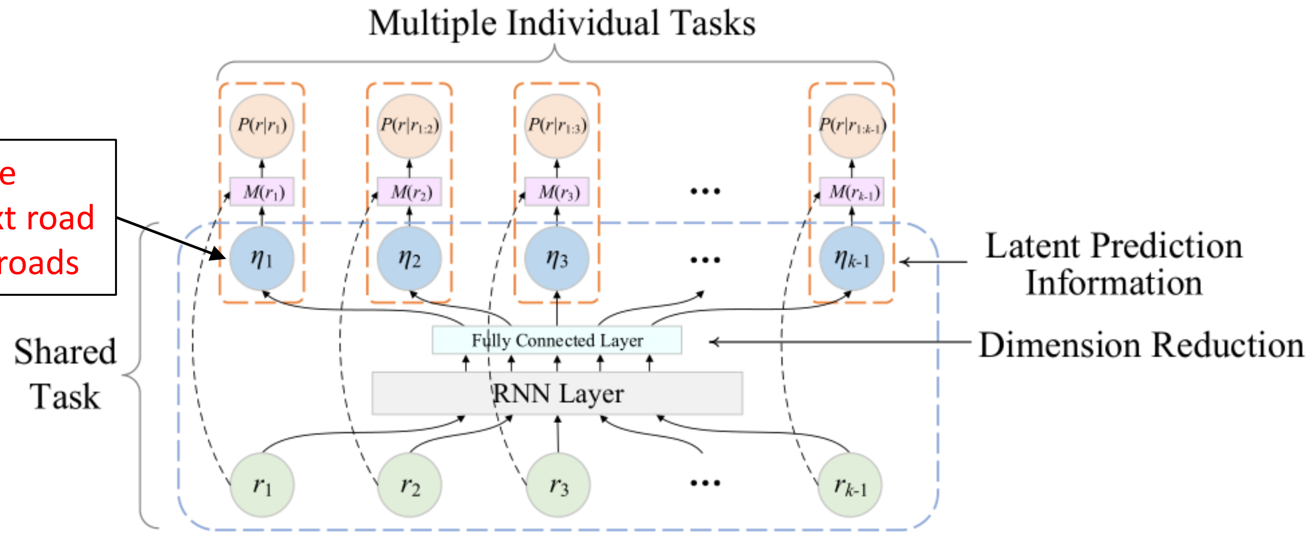
- Decompose the full probability into several individual tasks
- Shared task layer: homogeneous prediction task
  - Across all roads
- Several Individual models: heterogeneous prediction task
  - Road-specific model



Latent Prediction Information RNN (LPIRNN)

- Decompose the full probability into several individual tasks
- Shared task layer: homogeneous prediction task
  - Predict the direction of next road (latent)
- Individual task layer: heterogeneous prediction task
  - Predict the road among the roads that can be transited from r_i

First: predict the direction of next road given previous roads

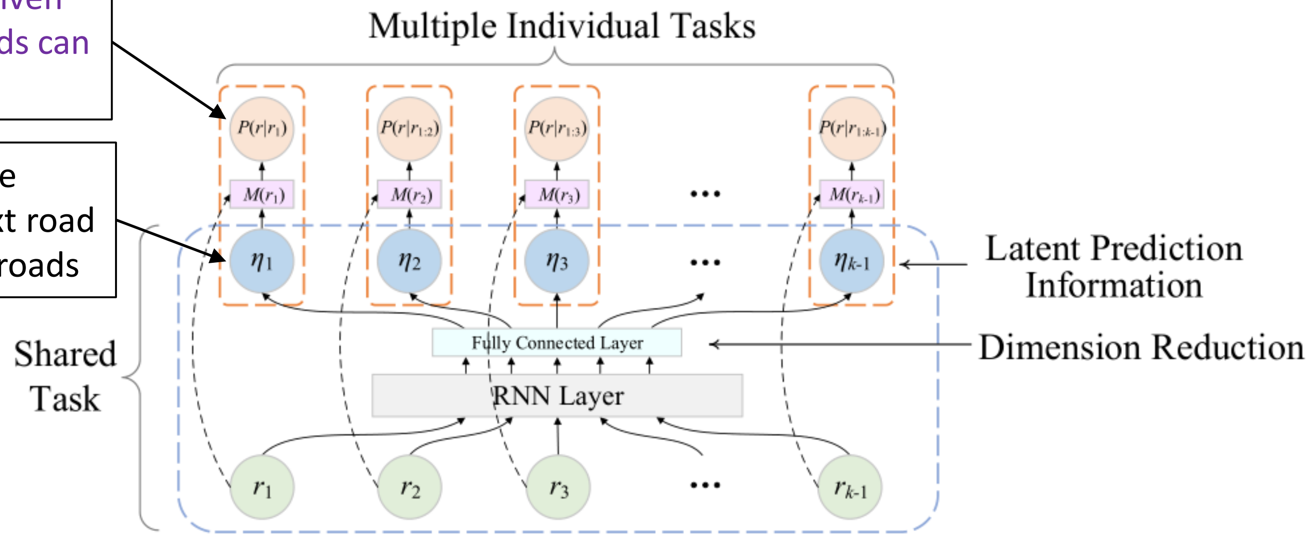


Latent Prediction Information RNN (LPIRNN)

- Decompose the full probability into several individual tasks
- Shared task layer: homogeneous prediction task
  - Predict the direction of next road (latent)
- Individual task layer: heterogeneous prediction task
  - Predict the road among the roads that can be transited from r_i

Second: predict next road given the direction among all roads can be legally transited to.

First: predict the direction of next road given previous roads

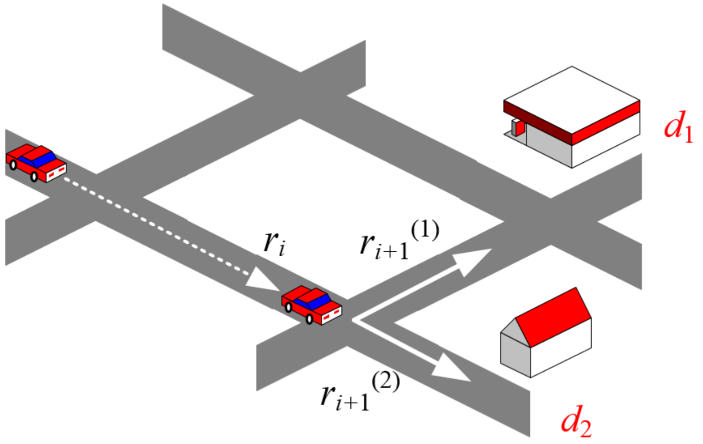


Another Task (with destination)

- **Objective**: Given a set of historical trajectories, build a probabilistic model to model the distribution of the trajectory data given the fact of ***destination***.

$$P(T|d) = P(r_1|d) \prod_{i=1}^{k-1} P(r_{i+1}|r_{1:i}, d)$$

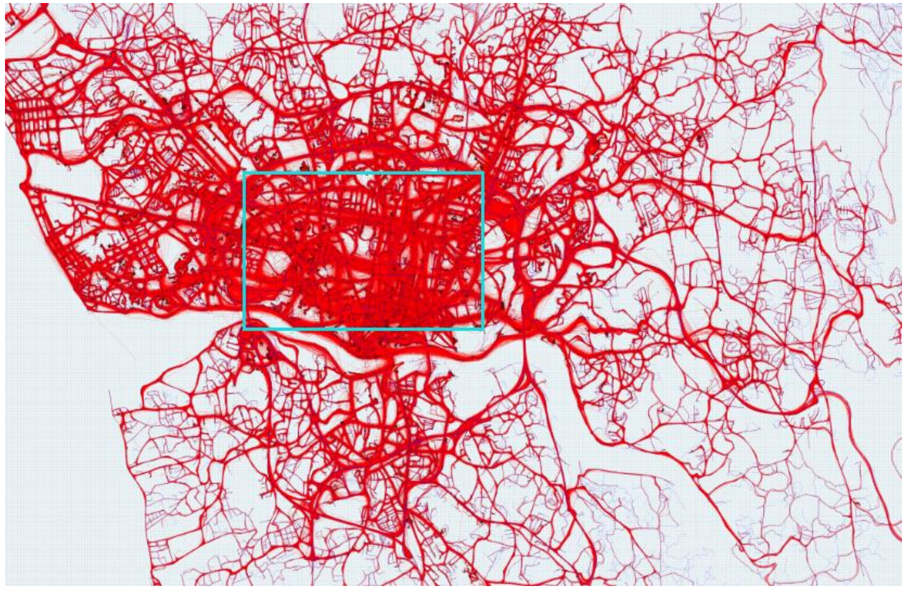
- Intuitively, the routing decision $P(r_{i+1}|r_{1:i}, d)$ is correlated to where the d is.



Experiments

	# Edges	# Vertices	# Trajectories	# Samples per edge
PT_{large}	40,267	18,157	859,195	21.3
PT_{small}	6,117	3,182	486,268	79.5
SH_{large}	60,200	28,620	3,709,666	61.6
SH_{small}	8,075	3,632	757,032	93.8

- Dataset: Porto (public dataset), Shanghai
- Evaluation metrics:
  - Negative log-likelihood $NLL = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{k_i-1}\log P(r_{j+1}|r_{1:j})$
  - Prediction accuracy $ACC = \frac{1}{\sum_{i=1}^{N} k_i}\sum_{j=1}^{k_i-1}\mathbf{1}\{\text{argmax}_{r\in E}P(r|r_{1:j}) = r_{j+1}\}$



Model Configuration

RNN cell	LSTM
Layer	1
Hidden unit	400~600
Initialization	Uniform [-0.03, 0.03]
Optimizer	RMSProp
Embedding dimension	400~600
Learning rate	1e-4 with decay rate at 0.9
Dropout	0.1
Gradient clippling	By norm at 1.0

Baselines

- N-gram
  - bi/tri/4-gram
- BIRL [Zheng and Ni, 2014]
- MEIRL [Ziebart et al., 2008a]
- Vanilla RNN-based language model
  - using LSTM with the same hidden unit setting

Overall Evaluation

Task	Without Destination								With Destination							
Dateset	PT_{small}		PT_{large}		SH_{small}		SH_{large}		PT_{small}		PT_{large}		SH_{small}		SH_{large}	
Metric	NLL	ACC	NLL	ACC	NLL	ACC	NLL	ACC	NLL	ACC	NLL	ACC	NLL	ACC	NLL	ACC
Bi-gram	8.32	90.43%	9.55	90.69%	9.51	83.76%	9.22	85.57%	6.20	94.15%	8.91	92.30%	7.40	88.54%	7.38	89.11%
Tri-gram	7.97	90.89%	9.15	91.15%	9.04	84.60%	8.76	86.26%	6.20	93.88%	8.92	91.99%	7.34	87.81%	7.29	88.60%
4-gram	7.75	91.21%	8.91	91.43%	8.71	85.24%	8.47	86.77%	6.21	93.57%	8.93	91.66%	7.31	87.02%	7.24	88.04%
BIRL	–	–	–	–	–	–	–	–	5.84	95.53%	–	–	6.67	91.42%	–	–
MERIL	–	–	–	–	–	–	–	–	7.84	93.70%	8.87	93.23%	7.28	91.19%	6.59	92.00%
RNN	7.77	92.27%	9.97	92.21%	8.92	86.60%	11.52	86.99%	3.74	97.13%	5.63	96.65%	5.27	93.58%	5.67	94.42%
CSSRNN	7.00	92.32%	**8.13**	**92.36%**	8.11	86.56%	**7.93**	87.83%	3.21	97.16%	**3.96**	96.89%	**4.21**	94.10%	3.97	**94.9%**
LPIRNN	**6.98**	**92.33%**	8.27	92.31%	**7.91**	**86.81%**	7.94	**87.84%**	**3.12**	**97.21%**	3.98	**96.97%**	4.22	**94.15%**	**3.96**	94.88%

Table 2: The results of two trajectory modeling tasks under four datasets.

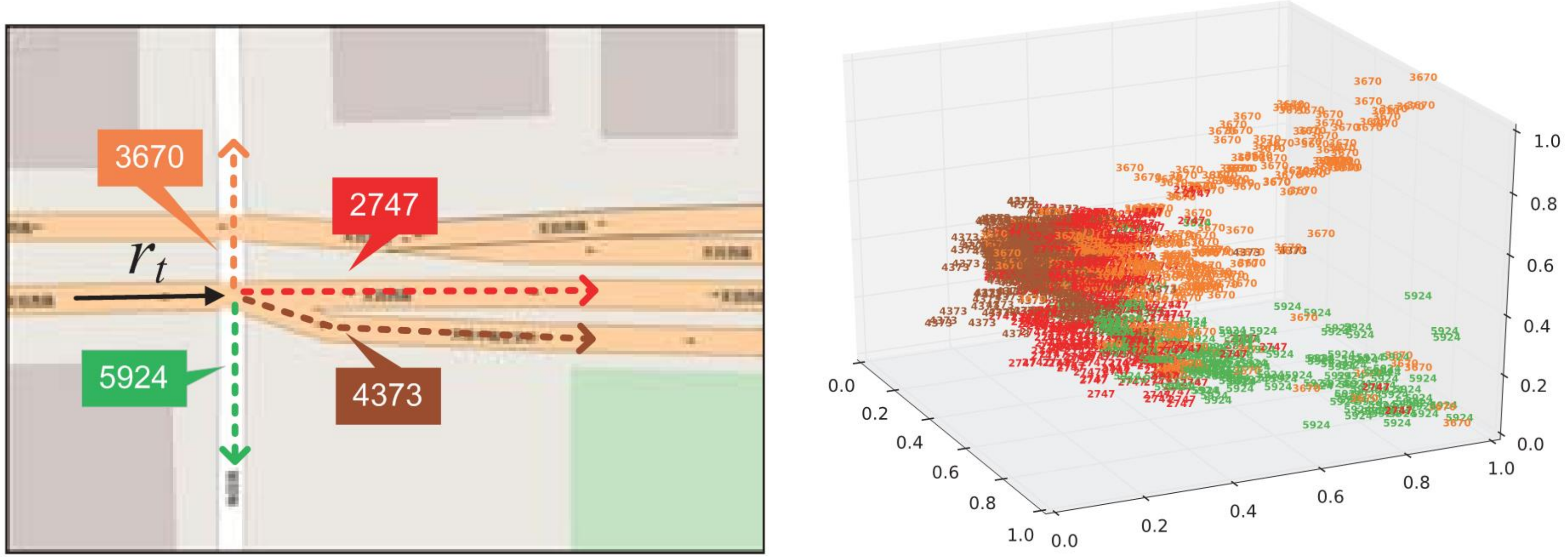
Efficiency of Speed-up Strategy in CSSRNN

- One GTX1080

state size	10K	20K	30K	40K	50K	60K
No speed-up (#traj/sec)	662	334	221	166	131	109
With speed-up (#traj/sec)	4563	4555	4588	4556	4582	4578
Speed-up Ratio	6.89	13.64	20.76	27.45	34.98	42.00

Table 5: The results of speed-up strategy under different sizes of states. The evaluation metric is the number of trajectories the model can process per second.

Understanding Latent Prediction Information



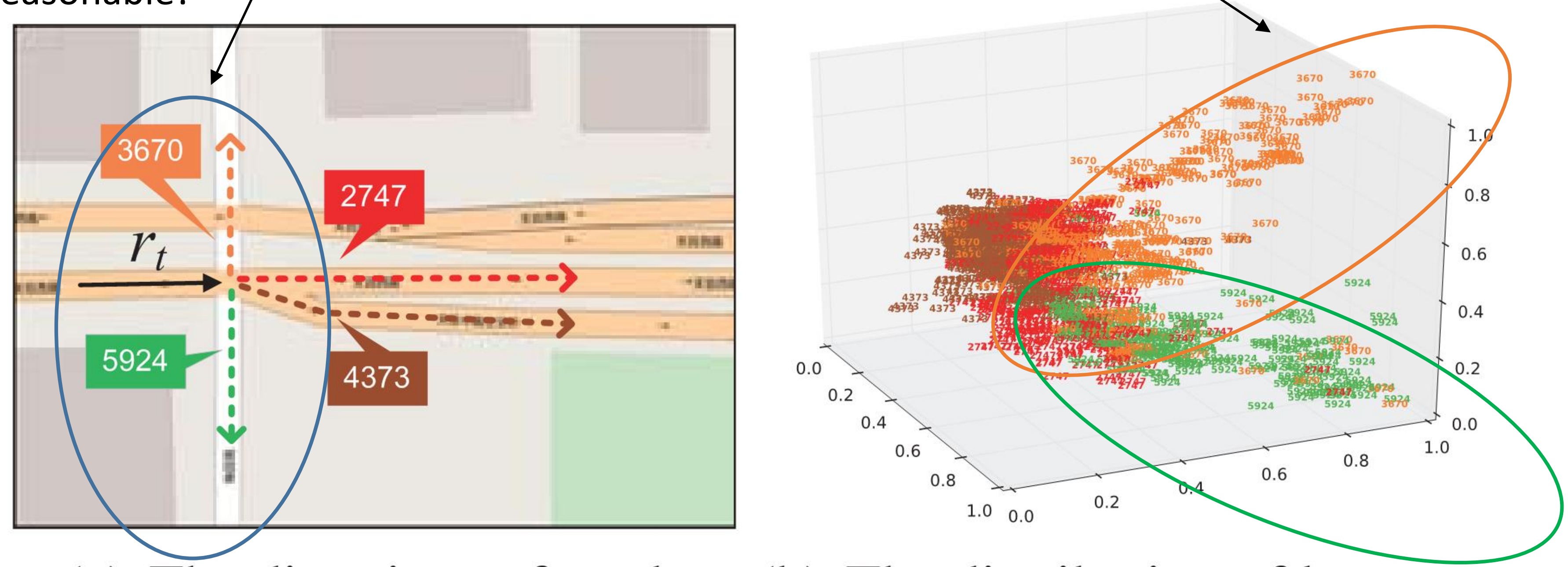
(a) The directions of roads (b) The distribution of latent prediction information via PCA

Figure 3: The visualization of latent prediction prediction.

Understanding Latent Prediction Information

Easy for a predictor (classifier) to separate them

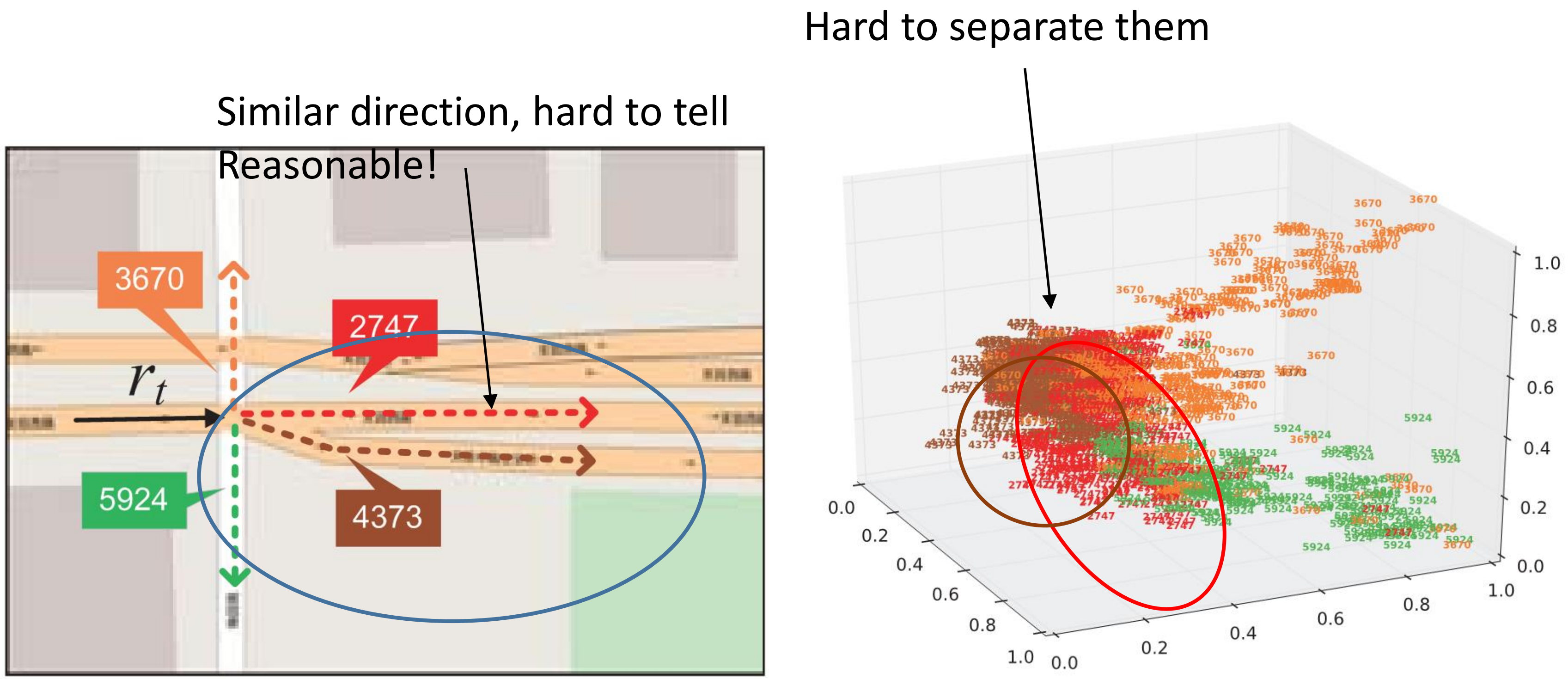
Different direction, easy to predict
Reasonable!



(a) The directions of roads (b) The distribution of latent prediction information via PCA

Figure 3: The visualization of latent prediction prediction.

Understanding Latent Prediction Information



(a) The directions of roads (b) The distribution of latent prediction information via PCA

Figure 3: The visualization of latent prediction prediction.

Visualizing the Embeddings of Destinations

- t-SNE

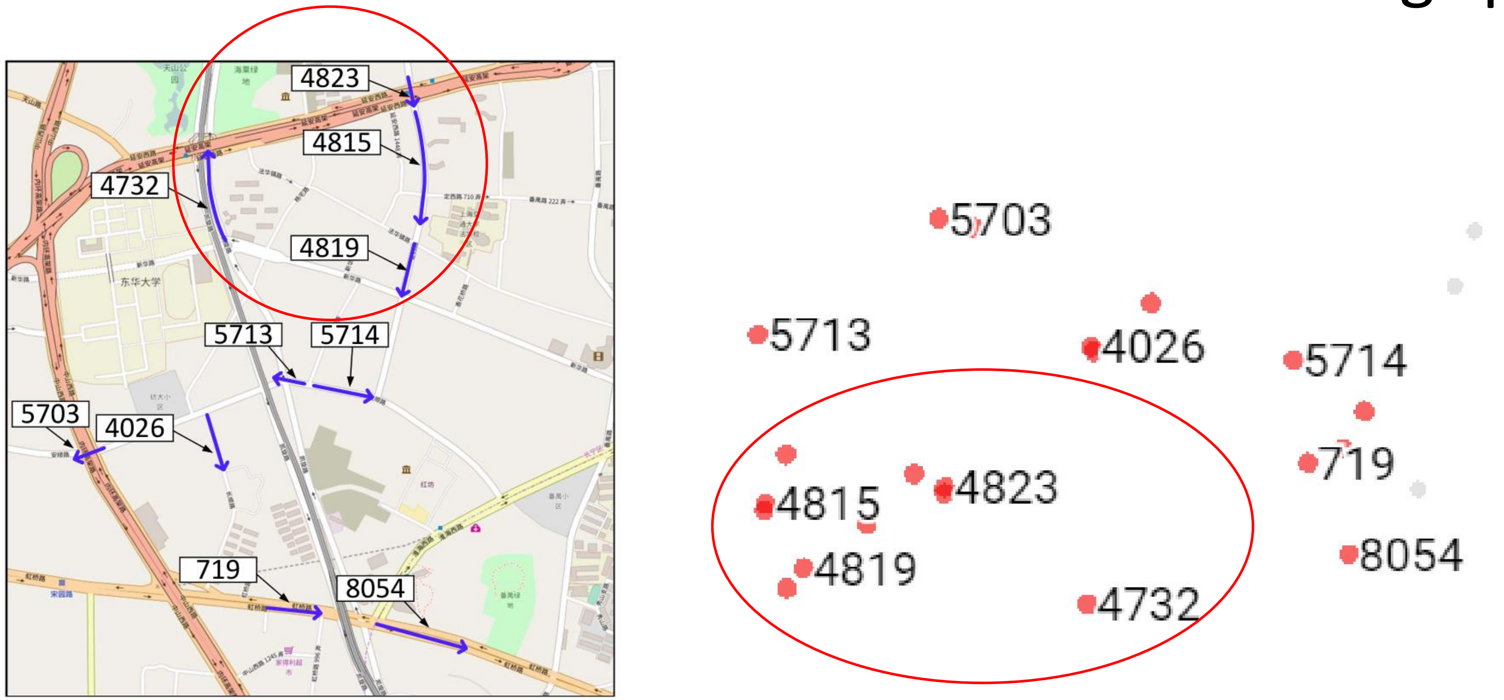
Clustered in the embedding space

(a) Roads in the map (b) Trained destination embeddings

Figure 4: Visualization of the trained destination embeddings.

Visualizing the Embeddings of Destinations

- the relative closeness can be reflected in the embedding space

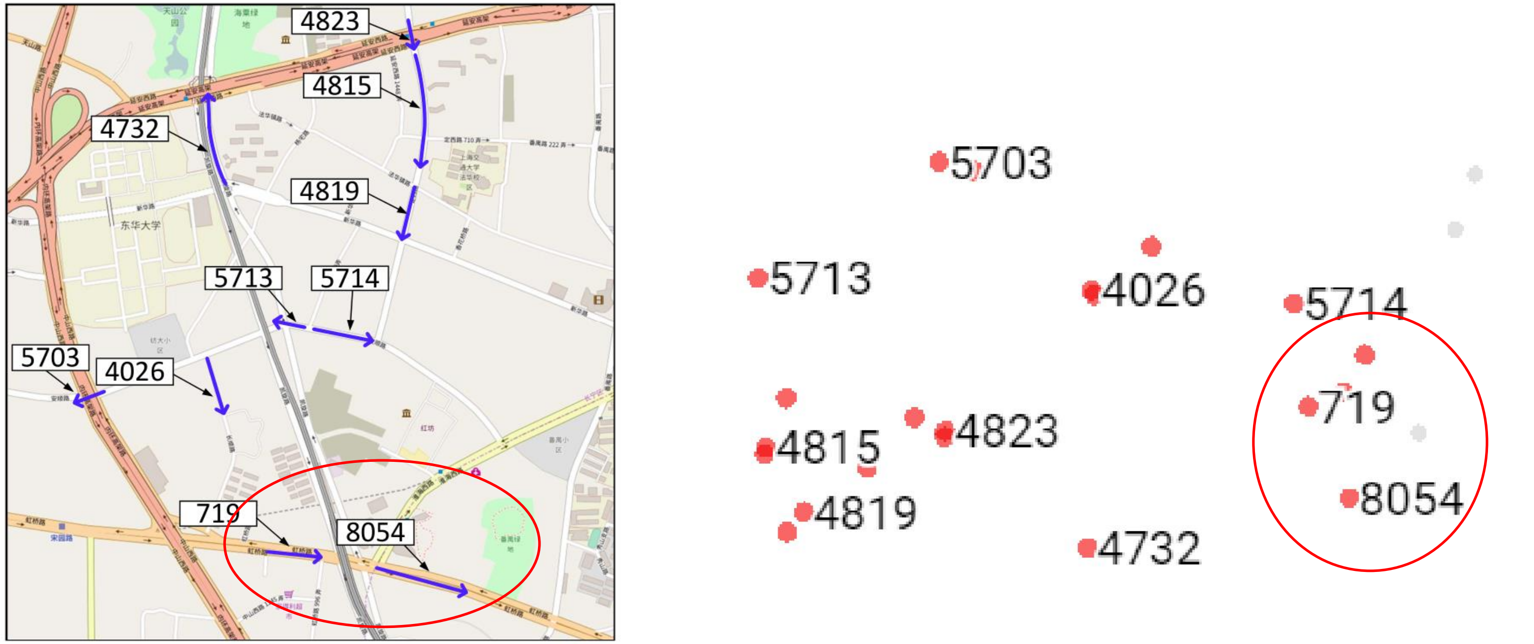


(a) Roads in the map (b) Trained destination embeddings

Figure 4: Visualization of the trained destination embeddings.

Visualizing the Embeddings of Destinations

- the relative closeness can be reflected in the embedding space

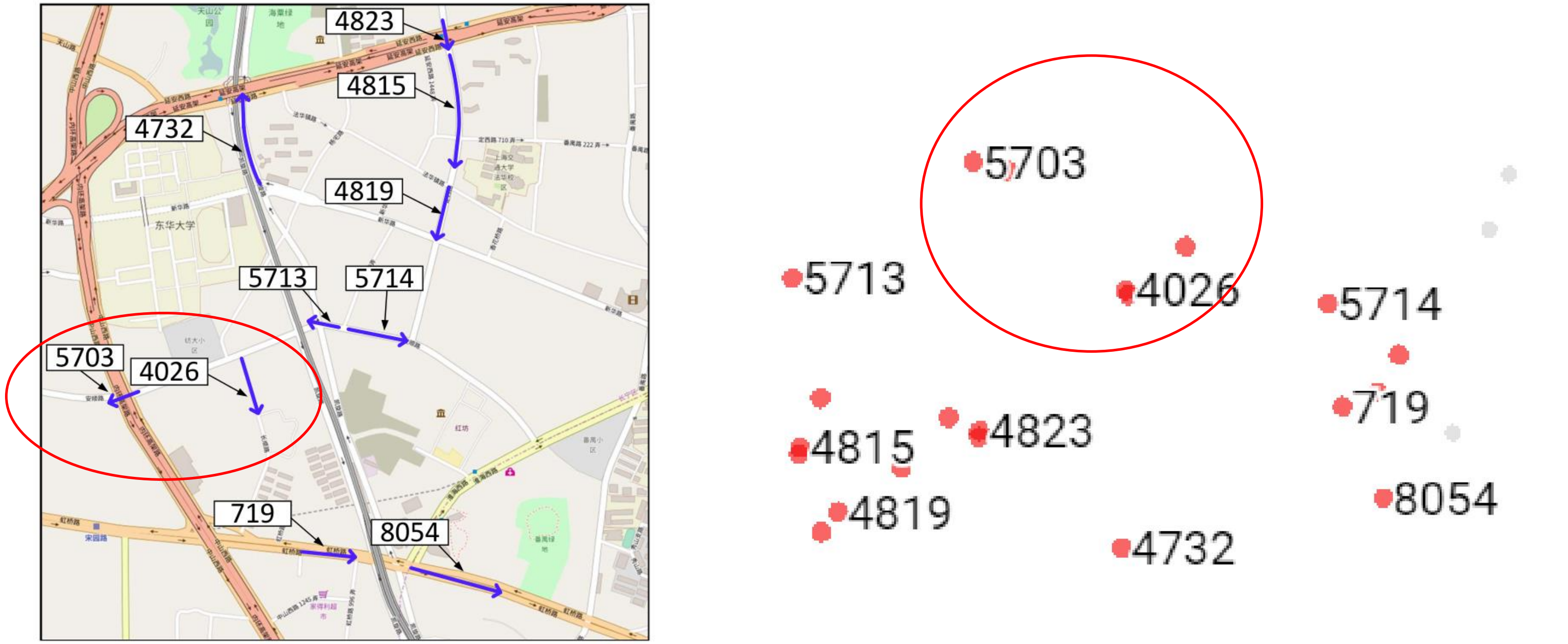


(a) Roads in the map (b) Trained destination embeddings

Figure 4: Visualization of the trained destination embeddings.

Visualizing the Embeddings of Destinations

- the relative closeness can be reflected in the embedding space

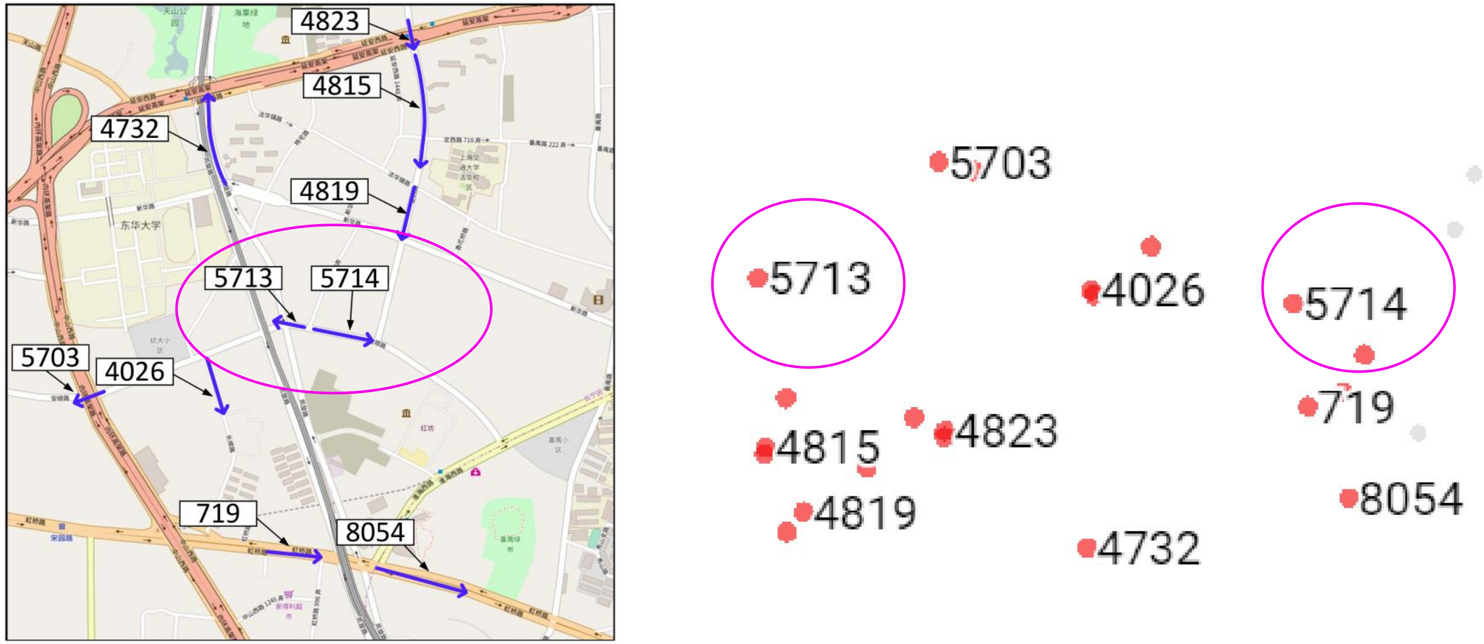


(a) Roads in the map (b) Trained destination embeddings

Figure 4: Visualization of the trained destination embeddings.

Visualizing the Embeddings of Destinations

- Not so close for 5713 & 5714 in the embedding space. Why?

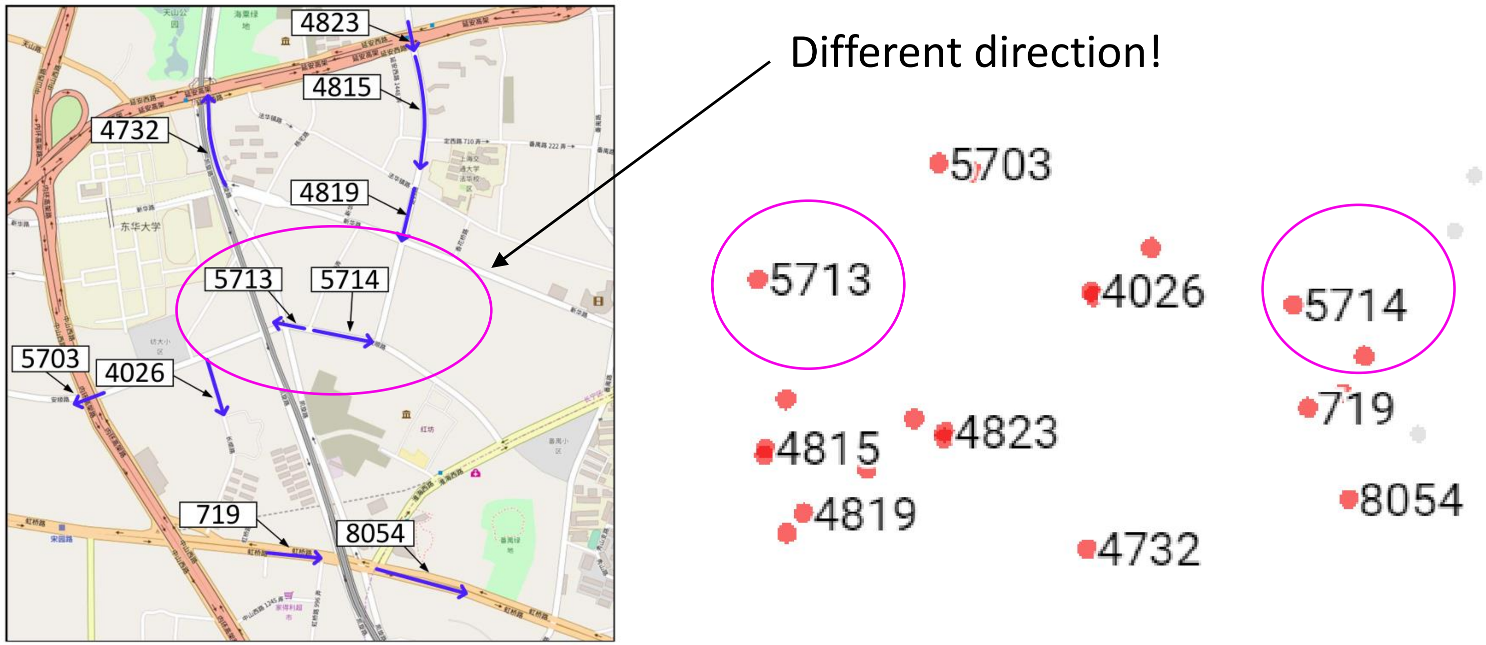


(a) Roads in the map (b) Trained destination embeddings

Figure 4: Visualization of the trained destination embeddings.

Visualizing the Embeddings of Destinations

- Capture both direction & spatial information



(a) Roads in the map (b) Trained destination embeddings

Figure 4: Visualization of the trained destination embeddings.

Conclusion

- Theoretically prove the limitation of the RNN language model for modeling trajectory
- Propose two models for effectively & efficiently model trajectory data as well as being irrelevant to the scale of the road network
- Use four real world datasets to evaluate our models and the results justify the effectiveness of our approaches

Motivation：
大量低价值密度的历史轨迹数据如何保存？

1. 概要（size小）
2. 通用（支持应用多）

对轨迹数据的分布进行建模：

1. 上海规模的城市，几十MB
2. 支持轨迹预测、压缩，路径推荐、补全，异常检测，……，等应用

谢谢！

Hao Wu, Ziyang Cheng, Weiwei Sun, Baihua Zheng, Wei Wang. Modeling Trajectories with Recurrent Neural Networks. IJCAI 2017: 3083-3090.

Code: https://github.com/wuhao5688/RNN-TrajModel

- *Environment*
  - *Python 2/3*
  - *Tensorflow 0.12.0*
  - *Linux*

孙未未　博士
wwsun@fudan.edu.cn
复旦大学
计算机科学技术学院 教授 博士生导师
移动数据管理实验室 主任