MIIS 2022

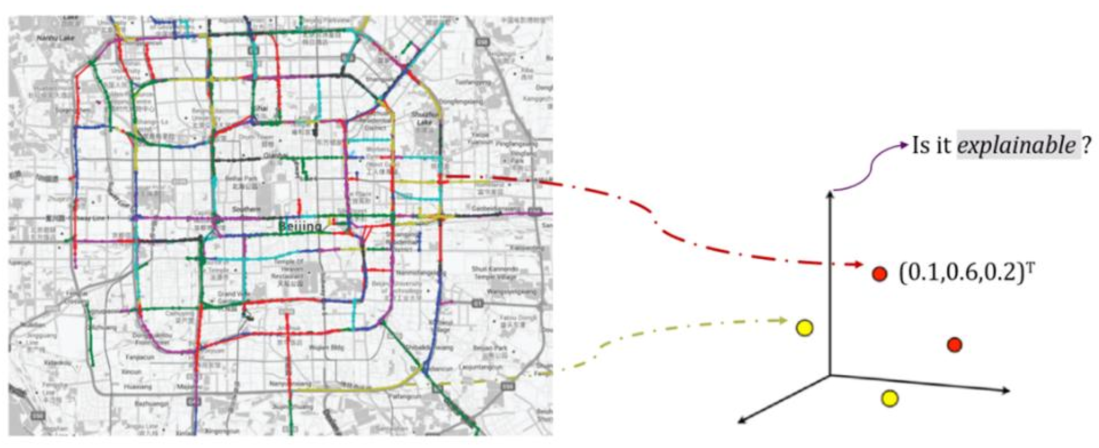
Explainable Trajectory Representation Based On Dictionary Learning

Yuanbo Tang Yang Li*
Tsinghua-Berkeley Shenzhen Institute, Tsinghua University
* Corresponding author

Introduction

Background.

- Trajectory representation learning provides great opportunities to understand vehicular traffic patterns.
- Downstream tasks includes Trajectory compression, Trip time estimation, Public transportation route planning, etc[1].



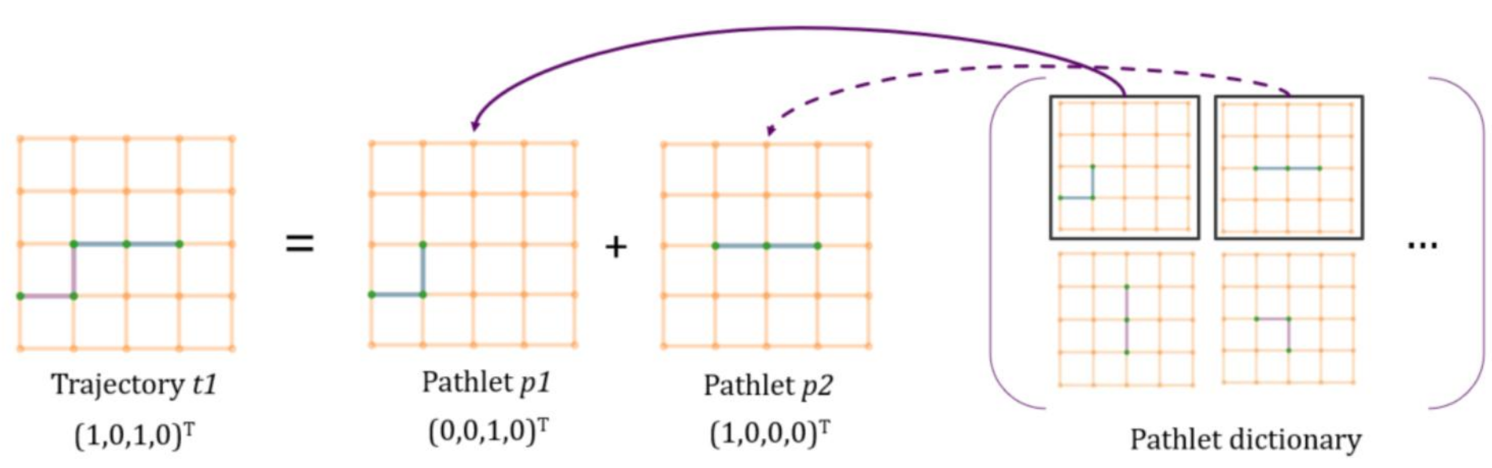
Transforming a trajectory into an embedding vector

Motivation.

- Embedding generated by deep-learning method is usually a dense vector whose dimension lacks semantic information.
- It is difficult to interpret the learned representation and use in applications[2].

Purpose.

- Extract common trajectory segments(called pathlet[3]) as a dictionary.
- Represent trajectory by concatenating pathlets from this dictionary.
- Generate semantic trajectory representation vectors, each dimension corresponding to a mobility pattern.



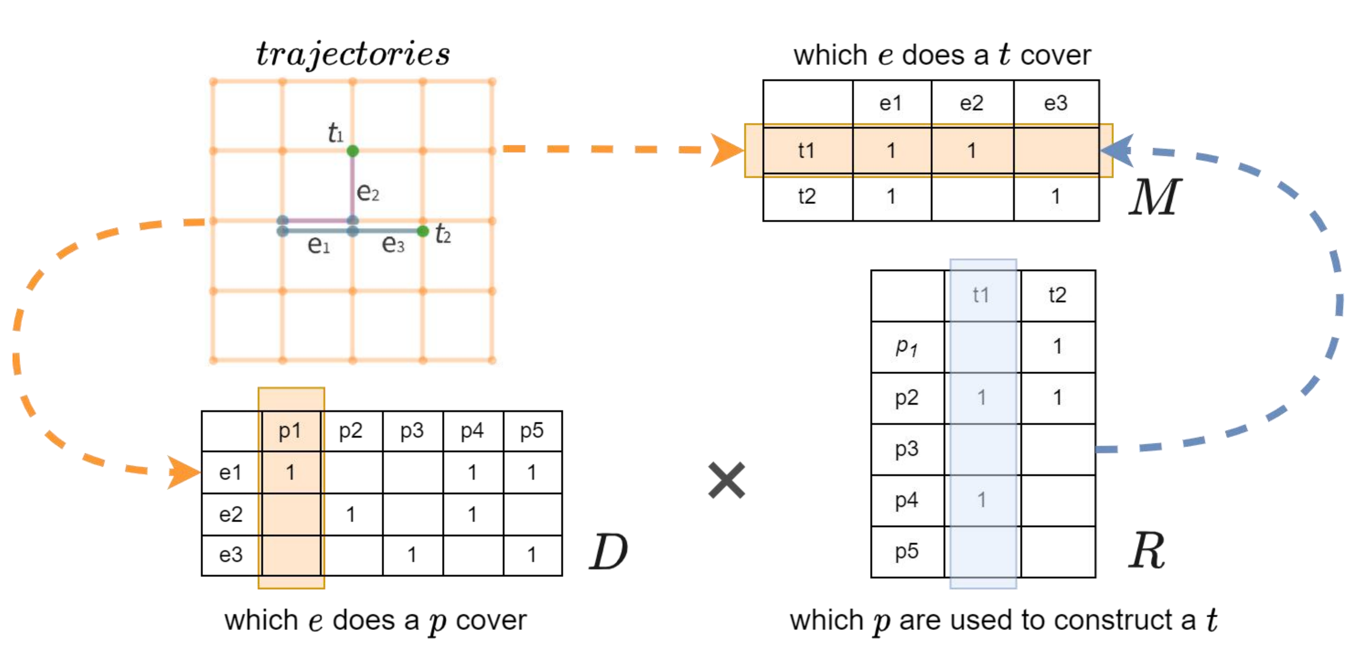
Represent a trajectory by concatenating pathlets from dictionary

Evaluation.

- This dictionary should be able to reconstruct all trajectories.
- Smaller dictionary is better, which means less redundant information.
- Average number of pathlets used to reconstruct trajectory should be as small as possible.

Problem Formulation

Terminology. Given a trajectory dataset T on roadmap $G =< E, V >$, $P(t)$ is used to describe all possible subpath p of $t \in T$ and $\overline{P} = \bigcup_{t\in T} P(t)$ refers to the whole pathlet space. The task is to find best pathlet dictionary $P \subset \overline{P}$ according to evaluation indicators.



Matrix are used to formulate the objective and constraint

- R is decision matrix, corresponding to dictionary P.
- Matrix D and M record the cover relationship between $T, E, \overline{P}$

Based on this, the problem can be formulated as follows:

The size of dictionary — Trade off hyper parameter

$$\min_{R_{i,j}\in\{0,1\}} \sum_i max(R_{i,:}) + \lambda * \sum_i \sum_j |R_{i,j}|$$

$$s.t.\ DR \geq M$$

Constraint: each trajectory should be covered — The number of pathlets used to reconstruct trajectories

Algorithm

1. Relax the binary constraint and get the fractional solution R^*
2. Obtain the rounded solution R^r as follows:

Algorithm 1 randomized rounding

Input: M: trajectory matrix; D: pathlet matrix; R_0: initial solution; ϵ, θ: hyper parameters;

Output: Optimal binary matrix R^r

1: # Step1, we compute the fractional solution R^* using gradient descend.
2: initial $R_0 = 0$;
3: **repeat**
4: compute gradient directions $g_k = \bigtriangledown f(R_k)$;
5: update the decision matrix $R_{k+1} = R_k - \alpha g_k$;
6: clip the result to make sure $0 \leq R_k \leq 1$;
7: **until** $(|f(R_k) - f(R_{k-1})| < \epsilon)$
8: #Step2, we compute rounded solution R^r based on R^*.
9: $P(R^r_{i,j} = 1) = min(1, \theta R^*_{i,j})$

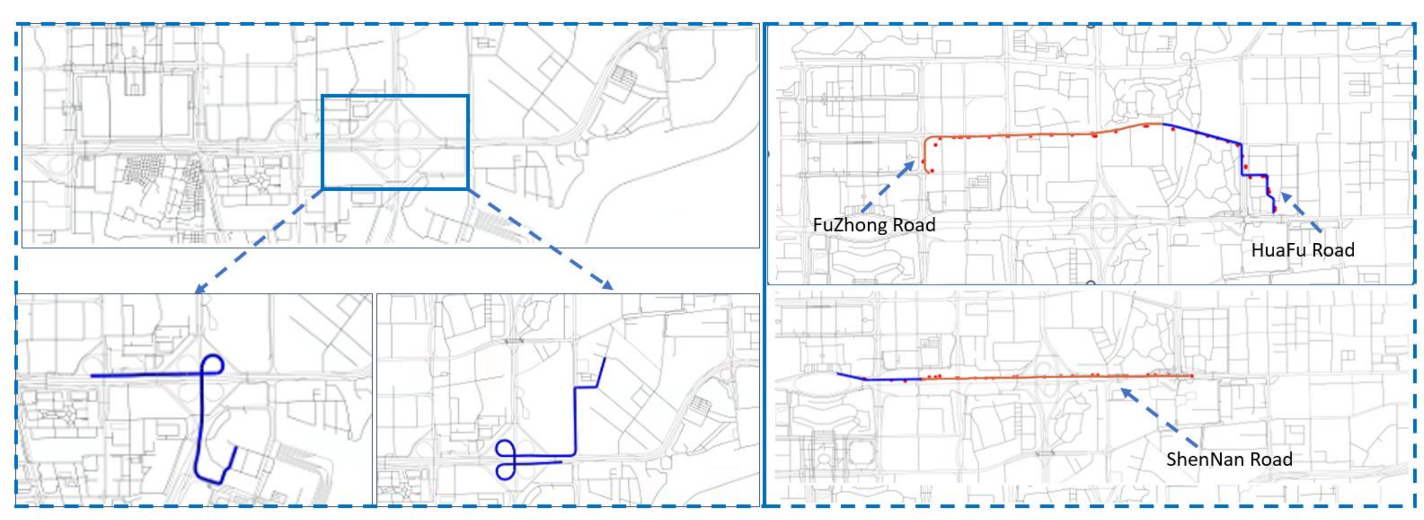
Probability bound. Given the size of dataset $|T|$, trajectory matrix M, pathlet matrix D and trade off parameter λ. Then for constant parameter θ, we have the following bound on the cost of R^r:

$$P[C(R^r) \leq 2\theta\frac{\lambda+1}{\lambda}C(R^*)\ and\ DR^* \geq M] \geq \frac{1}{2} - |T|e^{-\theta}$$

Experiment & Result

Case study.

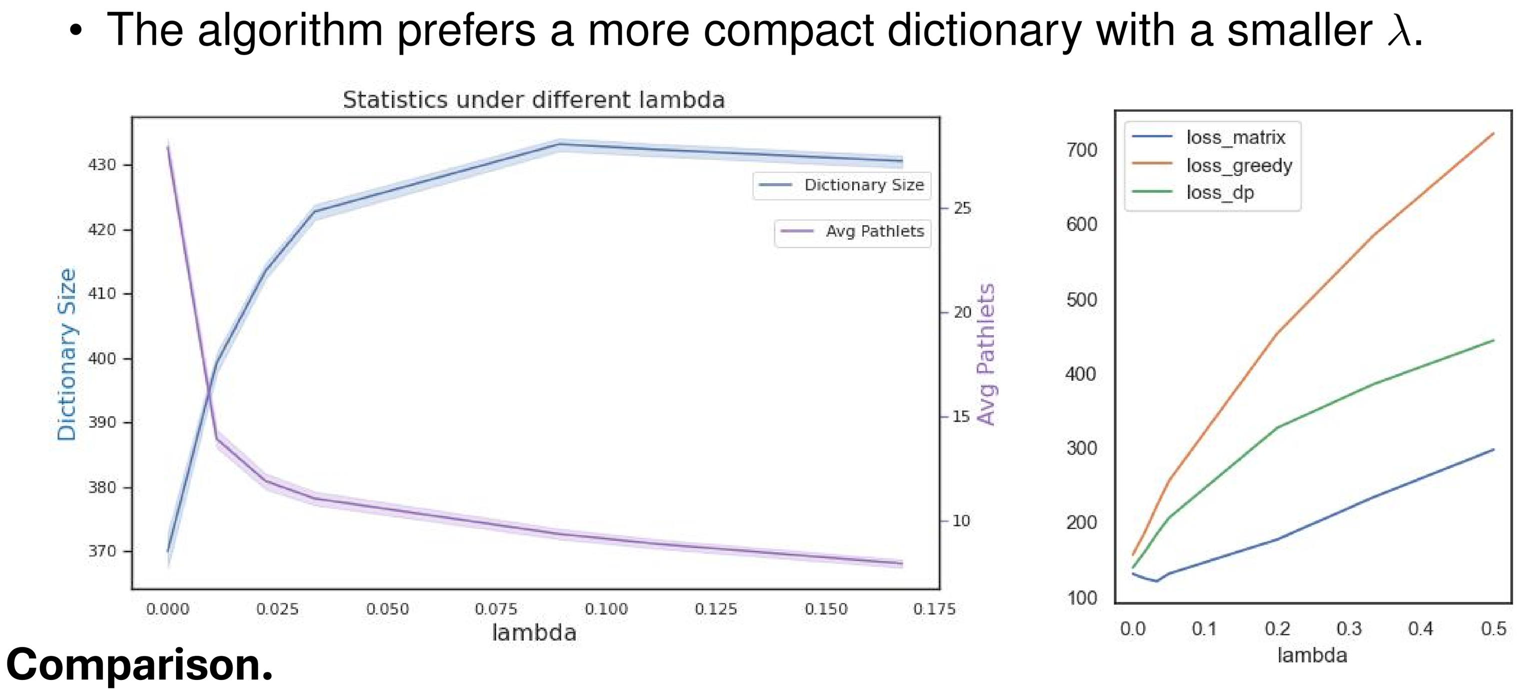
- Pathlets are visualized to verify if common patterns are found.
- Pathlets in Fig(a) refers to turning around or turning left on the overpass, which is consistent with our cognition in life.
- Fig(b) illustrates how a trajectory is reconstructed by pathlets.



(a)Common mobility pattern on an overpass (b)Trajectory decomposition using pathlets

Effect of λ.

- The average number of pathlets need to construct trajectory decreases as λ increases, which means longer pathlet are selected.
- The algorithm prefers a more compact dictionary with a smaller λ.



Comparison.

- Proposed method outperforms heuristic method and dynamic programming method[3] under different lambda on synthesis dataset .

Conclusion

- A novel dictionary learning based method with theoretical probability bound analysis is proposed to solve the trajectory representation problem.
- Explainable trajectory representation are generated, providing a deeper insight into mobility patterns.
- This method will be applied in more real-world applications to verify its generality in future work.

Reference

[1] T.-Y. Fu and W.-C. Lee, “Trembr: Exploring road networks for trajectory representation learning,” in ACM Transactions on Intelligent Systems and Technology, vol. 11, pp. 1–25.
[2] P. Y. e. al, “T3s: Effective representation learning for trajectory similarity computation,” in 2021 IEEE 37th International Conference on Data Engineering (ICDE). IEEE, 2021.
[3] C. Chen . e. al, “Pathlet Learning for Compressing and Planning Trajectories,” in Proceedings of the 21st ACM SIGSPATIAL.