Learned Index for Similar Trajectory Search in Distributed In-Memory System

Hani Ramadhan*, Hudzaifah E. Nursantio, Joonho Kwon

Pusan National University

Presented at Korean Database Conference 2019

October 11th, 2019, Chuncheon, South Korea

datalab
data science laboratory
@ PNU

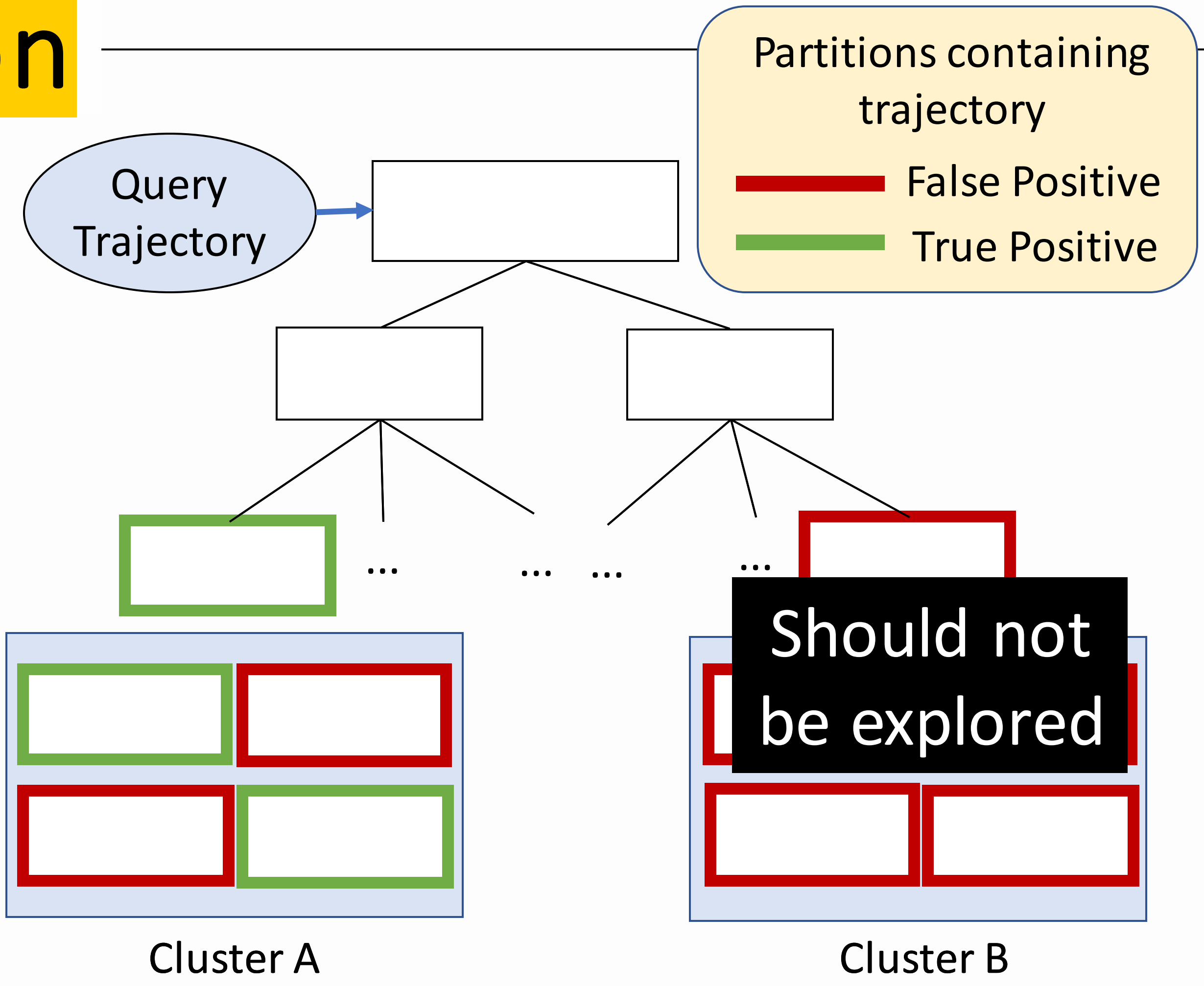
부산대학교
PUSAN NATIONAL UNIVERSITY

Outline

1. Problem
2. Challenges
3. Proposed Method
4. Experiment
5. Conclusion

Problem & Motivation

- Existing systems [1][2] on similar trajectory search still suffer from false positive results before verification
  - Inefficient query processing
- Machine learning-based index [3][4] (learned index) may provide better approximation of the trajectory location in partition



[1] D. Xie and F. P. J. M. Li, "Distributed Trajectory Similarity Search," PVLDB 2017.
[2] Z. Shang, G. Li and Z. Bao, "DITA: Distributed In-Memory Trajectory Analytics," in SIGMOD 2018.
[3] T. Kraska, A. Beutel, E. Chi, J. Dean and N. Polyzotis, "The Case for Learned Index Structures," in SIGMOD 2018.
[4] T. Kraska, M. Alizadeh, A. Beutel, E. Chi, A. Kristo, G. Leclerc, S. Madden, H. Mao and V. Nathan, "SageDB: A Learned Database System," in CIDR 2019.

Related Works

- [1] (uses segmentation) and **DITA [2] (uses global-local partitioning)** still suffers inefficiency of exploring false positive partitions
- Learned index improves the time efficiency and the index size from the traditional indexing (disk-based) [3]
- Different learned index is deployed for nearest neighbor search of large dataset of points [5].
  - Using a representation of codebook, proved for large dataset of points
  - Contextually, it is a modified similar search. But, still not applied to trajectory

[1] D. Xie and F. P. J. M. Li, "Distributed Trajectory Similarity Search," PVLDB 2017.
[2] Z. Shang, G. Li and Z. Bao, "DITA: Distributed In-Memory Trajectory Analytics," in SIGMOD 2018.
[3] T. Kraska, A. Beutel, E. Chi, J. Dean and N. Polyzotis, "The Case for Learned Index Structures," in SIGMOD 2018
[5] C.-Y. Chiu, A. Prayoonwong and Y.-C. Liao, "Learning to Index for Nearest Neighbor Search," IEEE Transactions on Pattern Analysis and Machine Intelligence, pp. 1-15, 2019.

Contributions

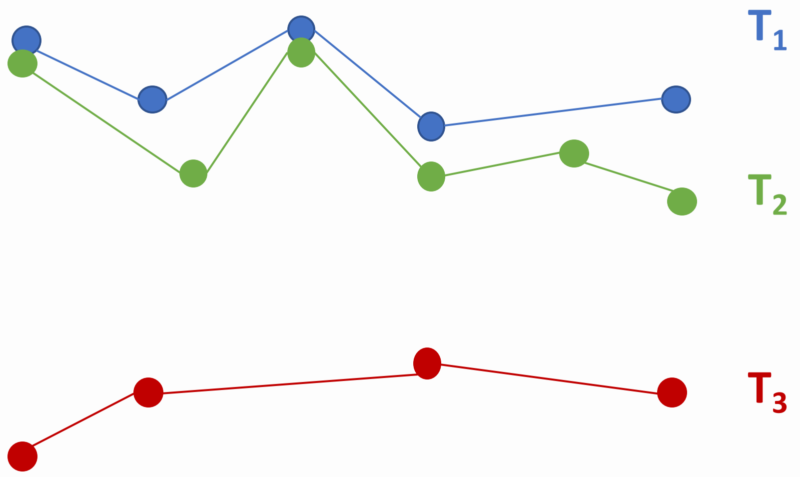
- We provide a learned index for an existing similarity trajectory search indexing (DITA) to minimize the exploration of irrelevant partitions/cluster thus improving the efficiency of the query processing
- We develop a probabilistic distance, applicable to learned index, to model the similarity between trajectories

Proposed Method

Modify the DITA indexing approach

The **probabilistic distance** to model the similarity between trajectories

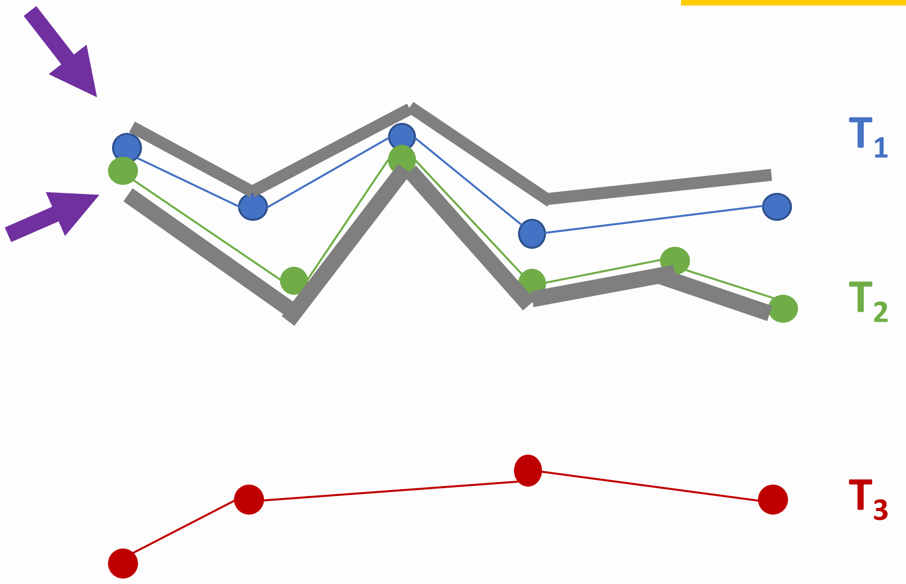
Trajectory Similarity Search Query



- Task:
  - Given a query trajectory T_Q
  - With a distance threshold of τ
  - Return all trajectories in set $\mathcal{T}$ whose distance $\leq \tau$

$$SimTS_{\tau}^{T_Q} = \{\langle T_s \rangle, T_s \in \mathcal{T}\}$$

Trajectory Similarity Search Query



Distance (DTW)	T_1	T_2	T_3
T_1	0	0.4	10
T_2	0.4	0	9
T_3	10	9	0

- Example query:
  - Given T_1 as query to set $\{T_1, T_2, T_3\}$
  - Using threshold $\tau = 0.5$
  - Result: $\{T_1, T_2\}$

$$\mathtt{DTW}(T,Q) = \begin{cases} \sum_{i=1}^{m} \mathtt{dist}(t_i, q_1) & \text{if } n = 1 \\ \sum_{j=1}^{n} \mathtt{dist}(t_1, q_j) & \text{if } m = 1 \\ \mathtt{dist}(t_m, q_n) + \min\big(\mathtt{DTW}(T^{m-1}, Q^{n-1}), & \\ \quad \mathtt{DTW}(T^{m-1}, Q), \mathtt{DTW}(T, Q^{n-1})\big) & \text{otherwise} \end{cases}$$

DTW → widely used in trajectory similarity functions in many experiments [6]

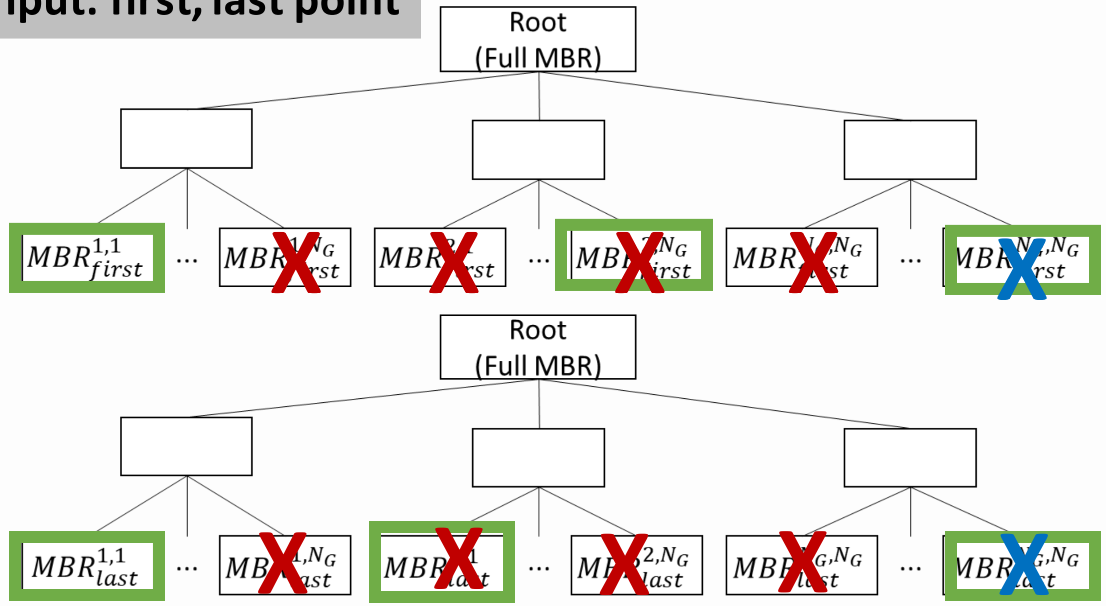
[6] C. S. Myers and L. R. Rabiner. A comparative study of several dynamic timewarping algorithms for connected-word recognition. Bell System Technical Journal, 60:1389–1409, 1981

DITA Comparison with/out learned Index

Global Partition

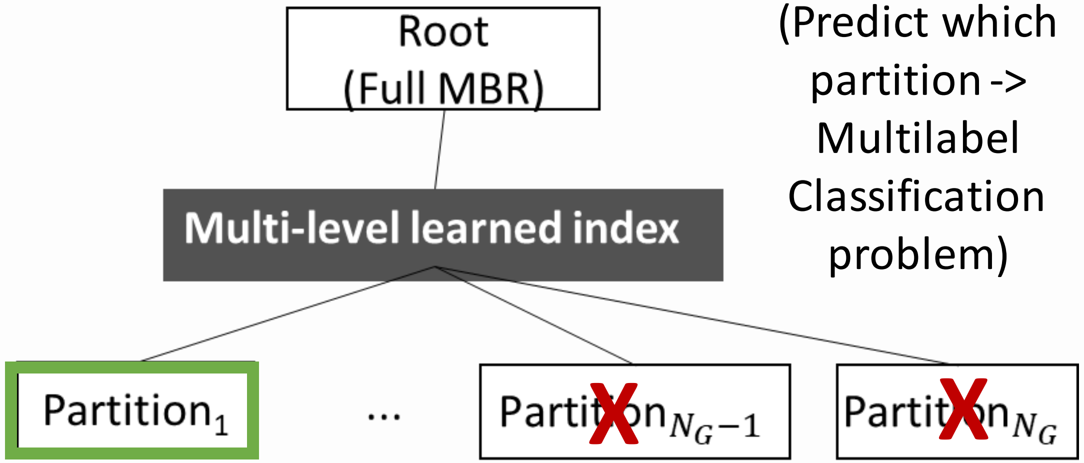
≤ threshold

Input: first, last point



1. Compute distance between first and last point to each global partition separately (2 trees)
2. Then, explore a global partition if first+last **total distance** less than treshold

Input: first, last, k pivot point



(Predict which partition -> Multilabel Classification problem)

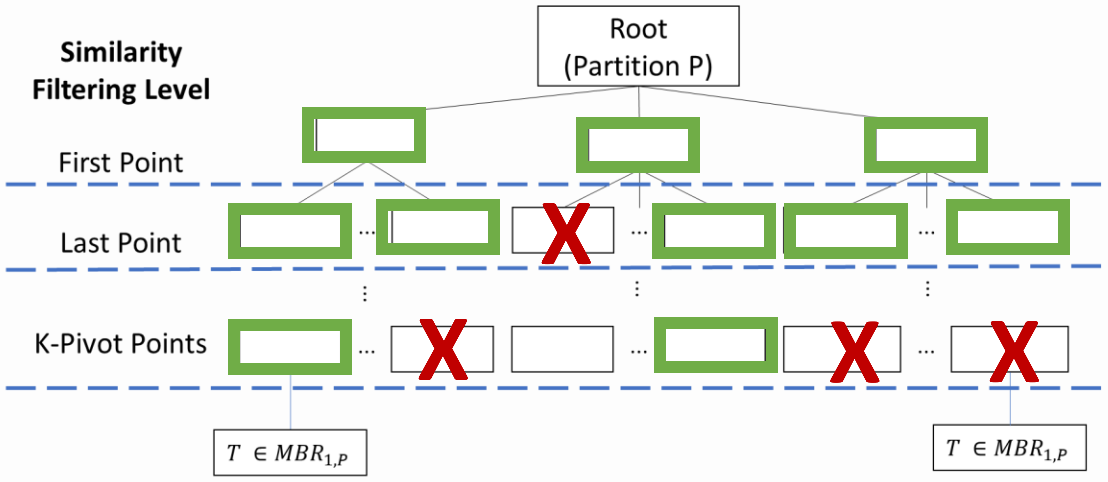
1. Compute similarity between first, last, and pivot points to each global partition (only 1 time)
2. Then, explore a global partition if their its distance less than treshold

DITA Comparison with/out learned Index

Local Partition

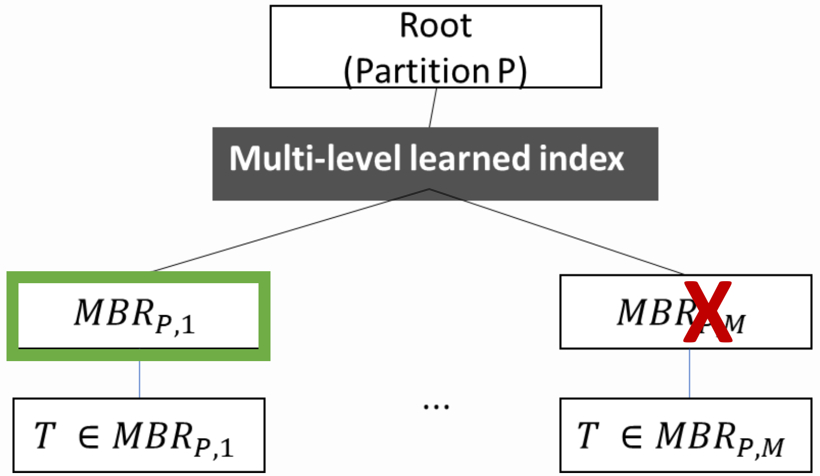
≤ threshold

Input: first, last, k pivot points



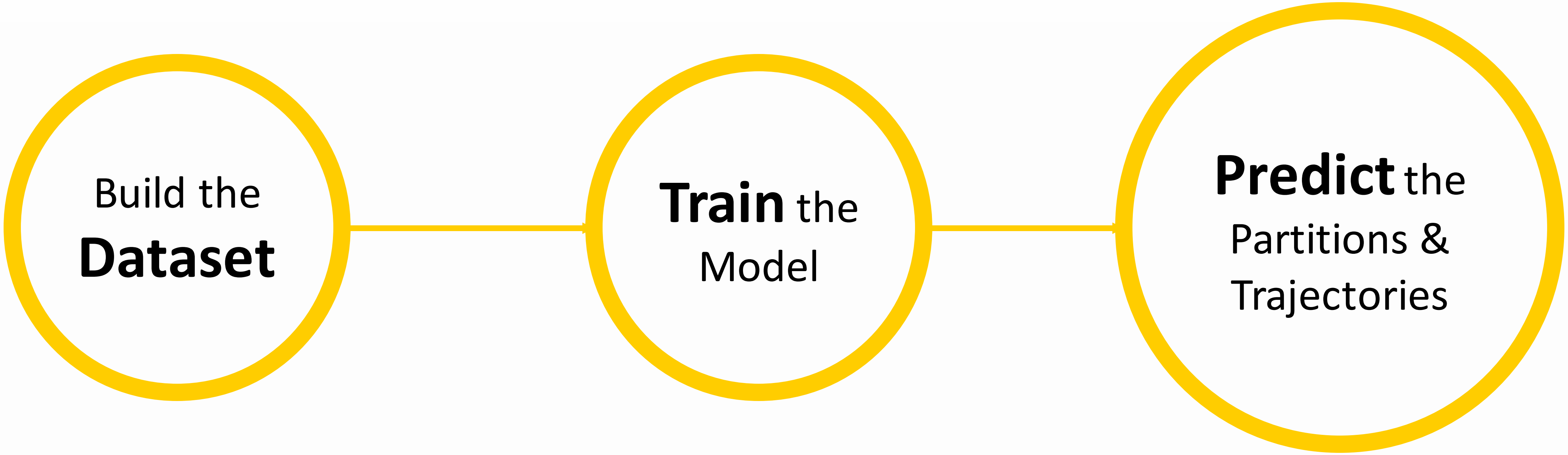
DITA Local Index

Have to dive 2+k levels of partition indexing (+pruning)



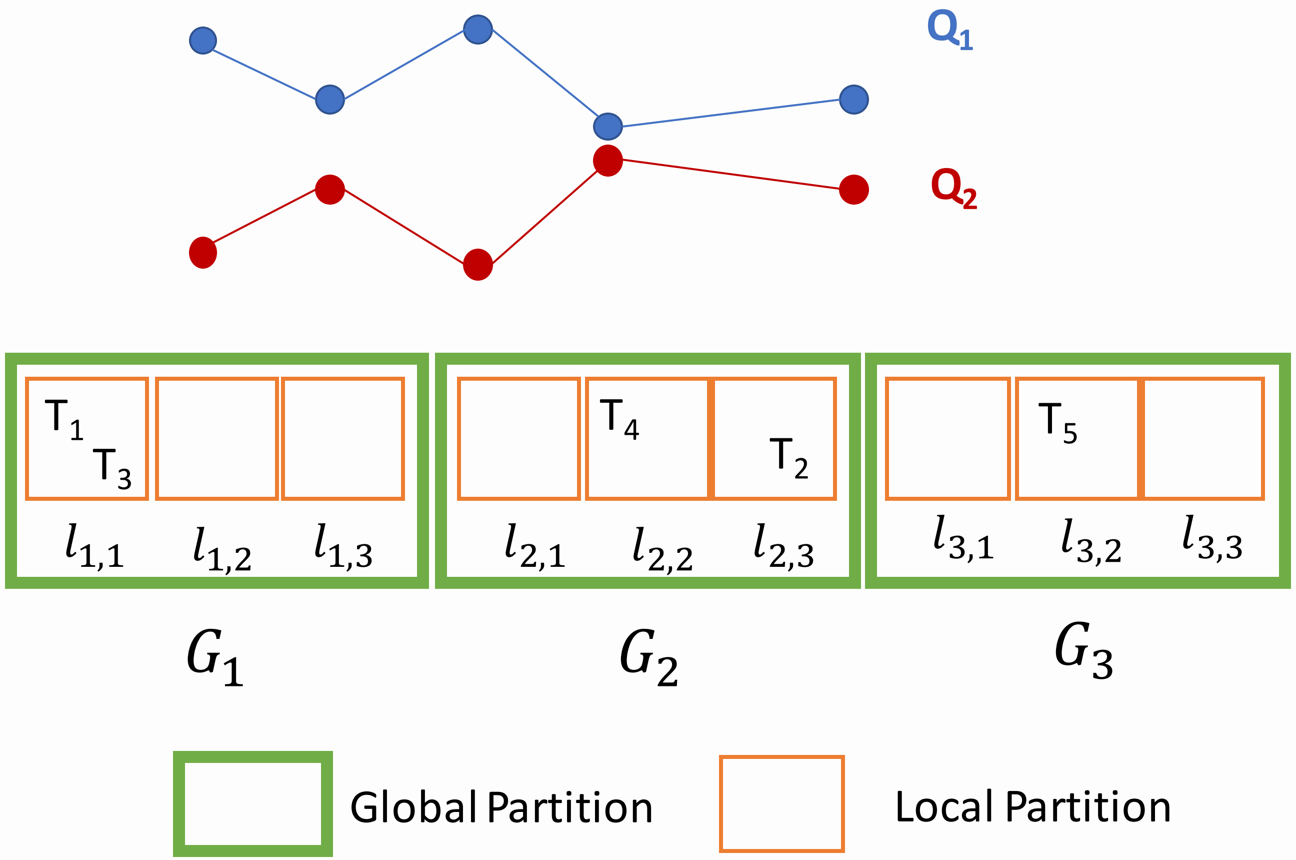
Goes to the local partitions that contains similar trajectories directly from single inference

Learned Index Approach for Trajectory Similarity Search



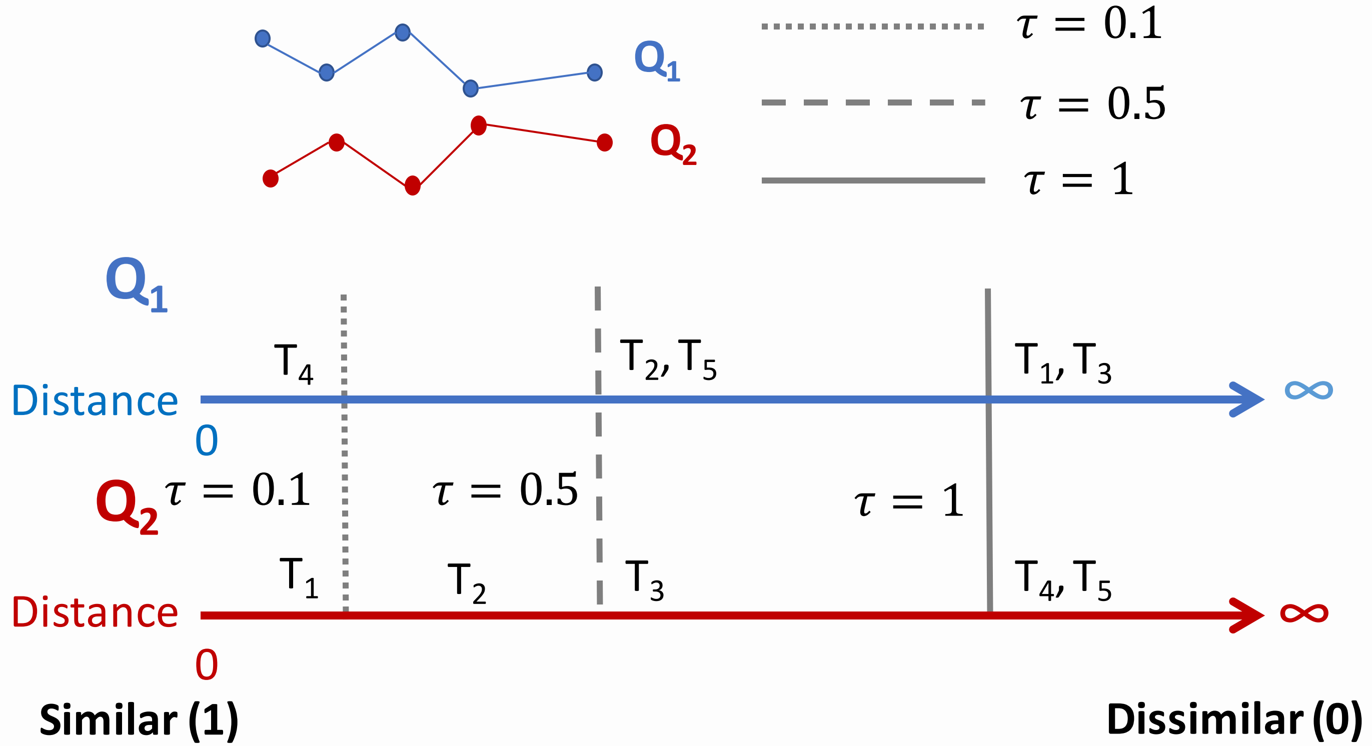
Build Dataset for Similarity Search Query

- Example:
  - Query Set $\mathcal{Q}$: $\{Q_1, Q_2\}$
  - Trajectory set $\mathcal{T}$: $\{T_1, T_2, T_3, T_4, T_5\}$
  - $H = \{0.1, 0.5, 1\}$

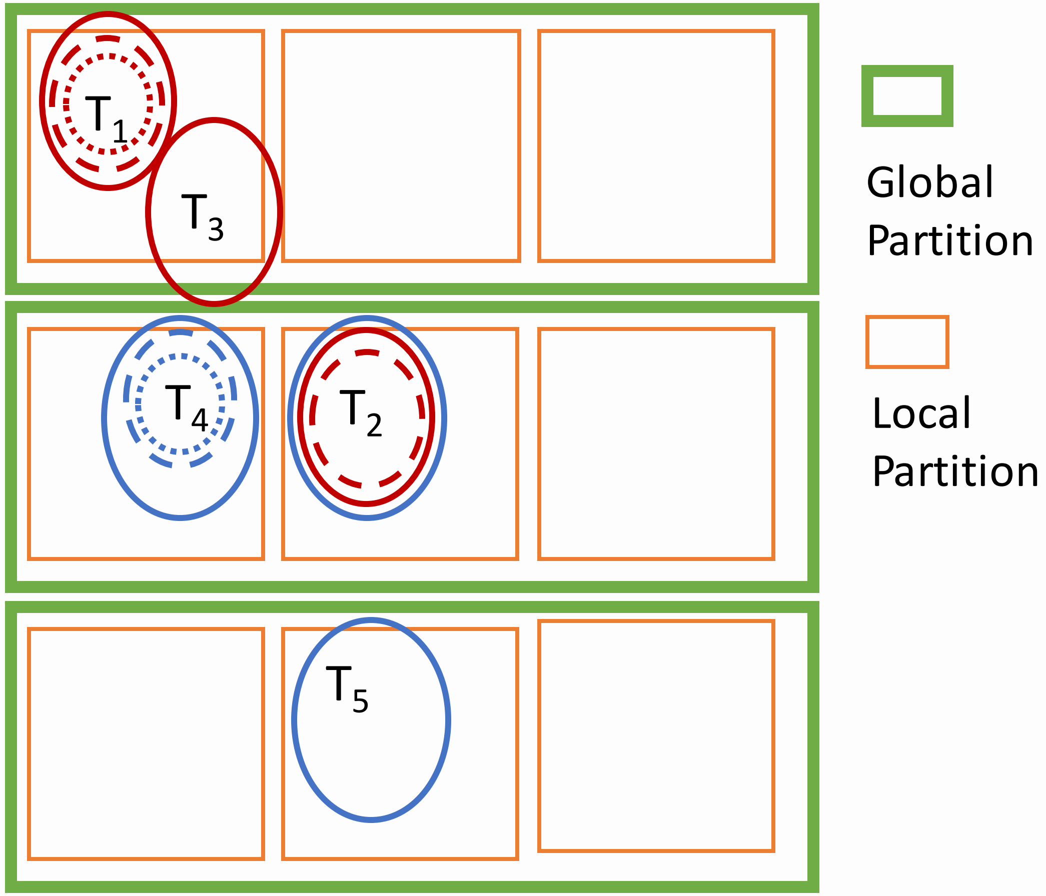


Build Dataset for Similarity Search Query

- Query result

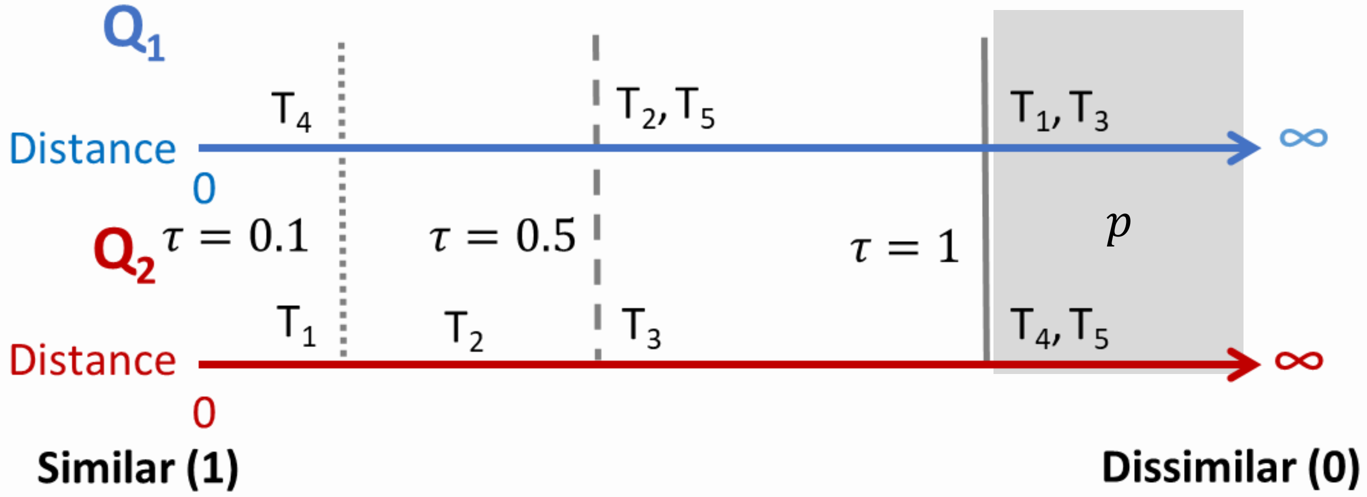


Learn this using Machine Learning!



Probabilistic Distance $d_p(T_Q|T_C)$

- A parameter p → uncovered threshold from H
- Also interchangable with the partitions
  - $T_C \leftrightarrow G_i \leftrightarrow l_{i,j}$



$$d_p(T_Q|T_C) = \begin{cases} 1, & T_C \in SimTS_{\tau}^{T_Q}, \tau = \min(H) \\ p, & T_C \notin SimTS_{\tau}^{T_Q}, \tau = max(H) \\ p + \dfrac{max(H) - \tau_{i-1}}{max(H)} \times (1-p), & T_C \in SimTS_{\tau}^{T_Q} \wedge T_C \notin SimTS_{\tau-1}^{T_Q} \end{cases}$$

Example: $p = 0.05$

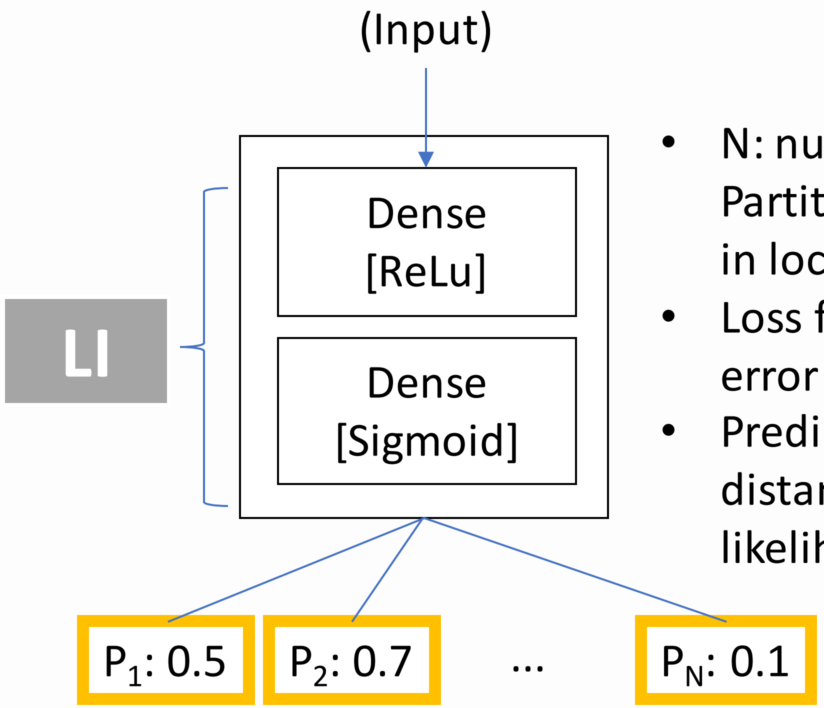
$d_p(Q_1|T_1) = 0.05$

$d_p(Q_2|T_2) = 0.905$

$d_p(Q_1|G_3) = 0.48$

$d_p(Q_2|l_{2,1}) = 1$

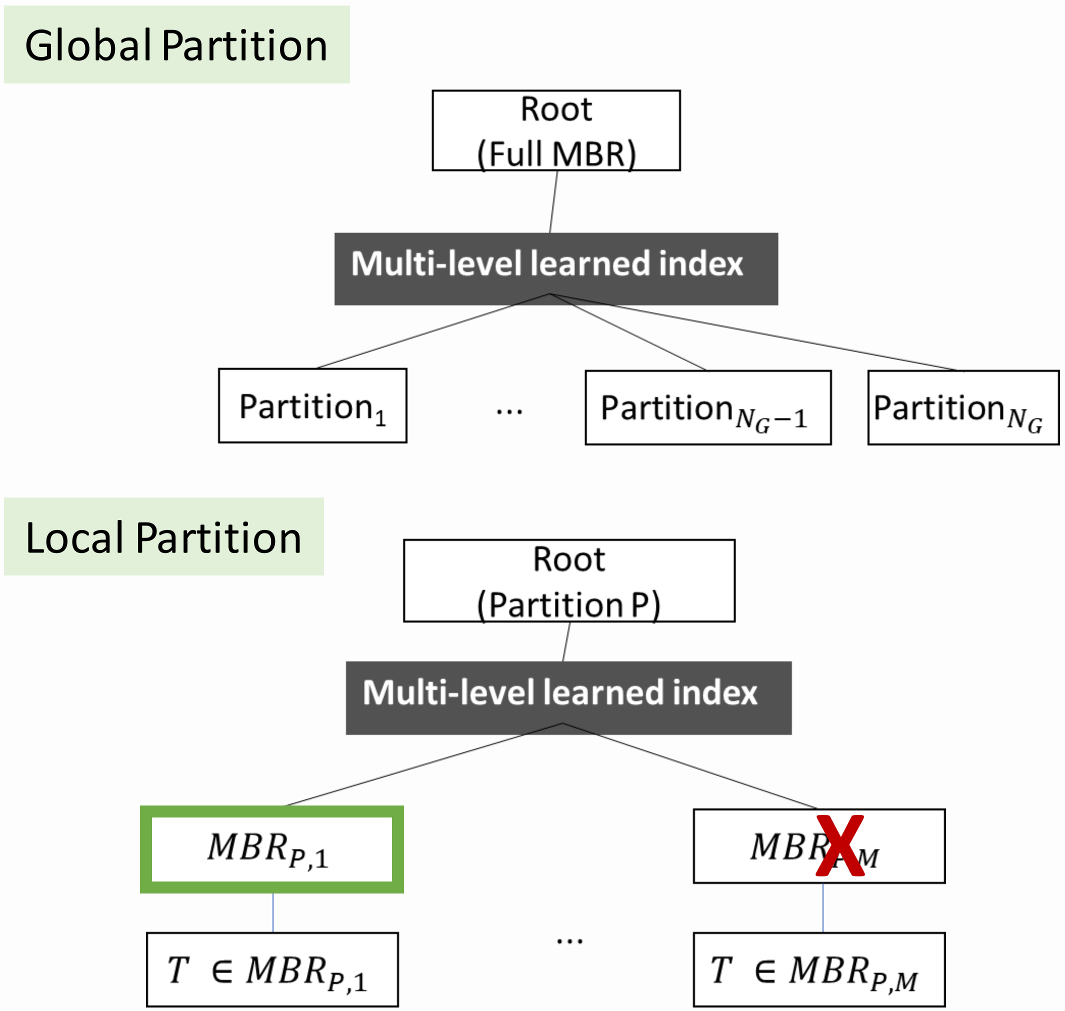
Train the ML Model (Single Level)



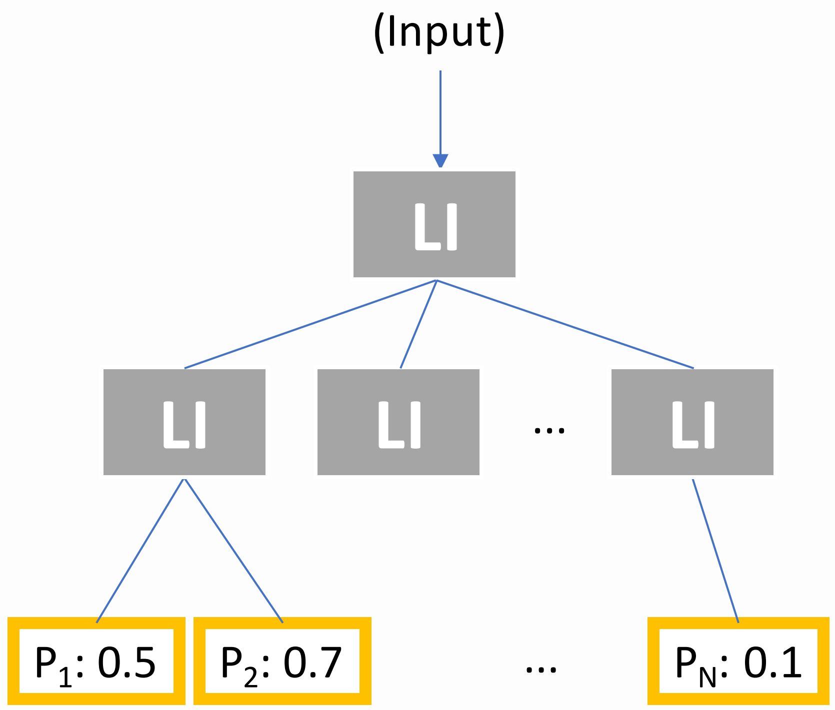
- N: num of Global/Local Partitions, num of Trajectory in local partitions
- Loss function: Mean absolute error
- Predicting probabilistic distance: similar to predicting likelihood of multilabel [0..1]

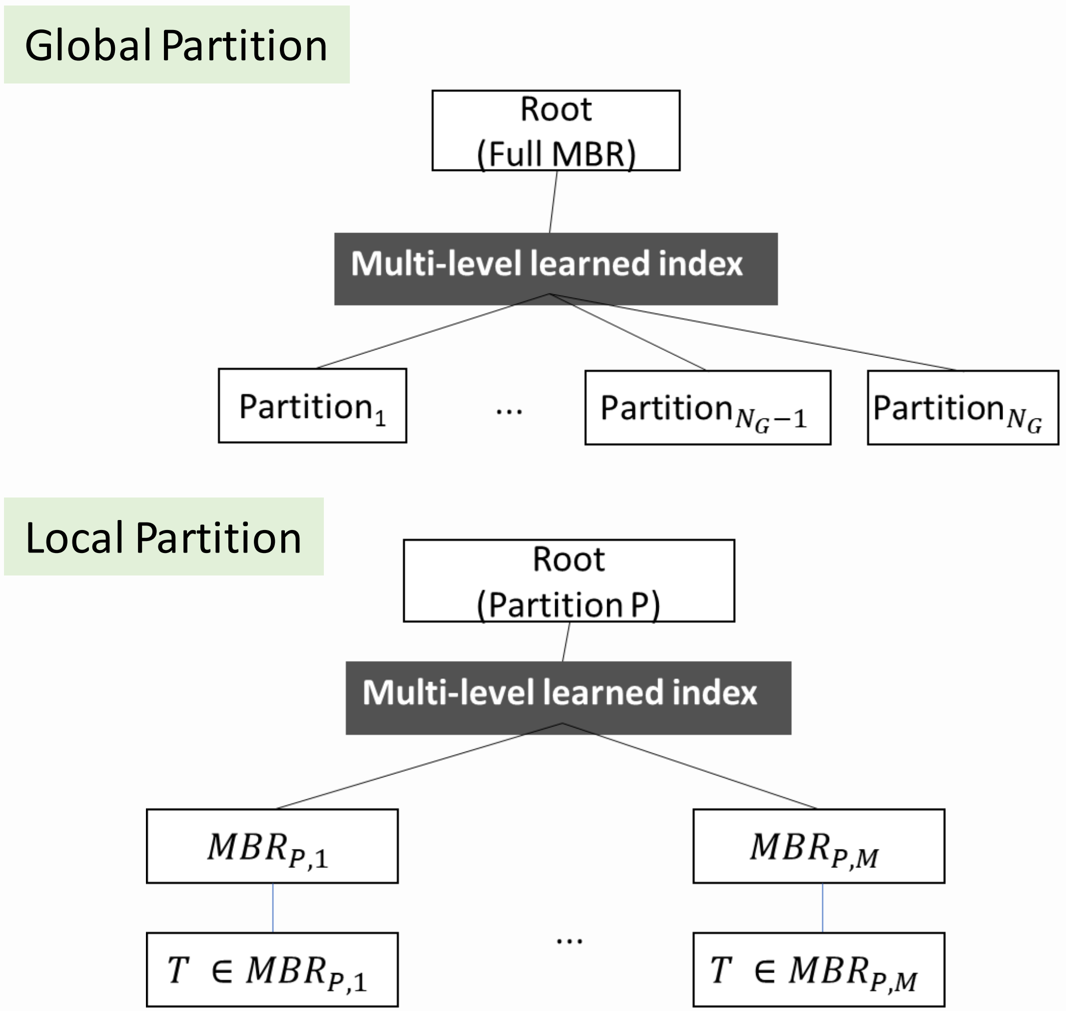
Sample input:

- First point (x_f, y_f)
- Last point (x_l, y_l)
- K pivot points $\{(x_1,y_1), (x_2,y_2),\ldots, (x_k,y_k)\}$



Train the ML Model (Multi-level)





Predict the Partitions & Trajectories

- Suppose we have a query with different threshold τ'
- Find All $T_i \in \mathcal{T}$ that satisfies $d_p(T_Q|T_i) \geq d_p(T_i|T_Q, \tau')$
  - T_i = similar trajectories/partitions containing
  - $d_p(T_i|T_Q, \tau') = \begin{cases} \frac{max(H) - \tau'}{max(H)} \times (1-p), & \tau' \leq max(H) \\ \frac{max(H)}{\tau'} \times p, & \tau' > max(H) \end{cases}$

Example: $\tau' = 3$
$p = 0.05, max(H) = 7.5$
$d_p(T_i|T_Q, \tau') = 0.6333$

$d_p(Q_A|G_3) = 0.47$
$d_p(Q_A|l_{2,1}) = 0.97$

G_3 is NOT IN the result
$l_{2,1}$ is IN the result

Experiment Setup

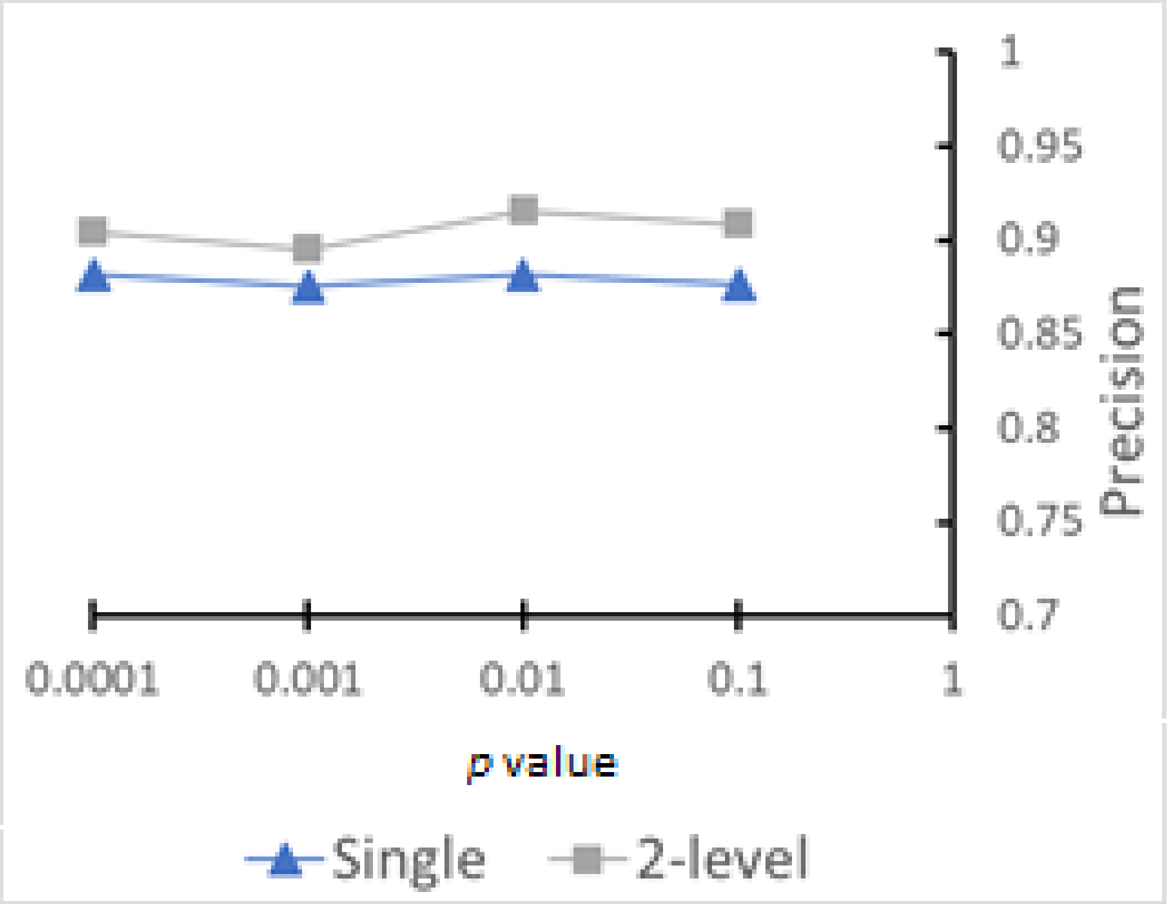
- Dataset: DITA example trajectories of taxi driving. Using (https://github.com/TsinghuaDatabaseGroup/DITA)
  - 10,000 trajectories
- Input: first point, last point, pivot point (k=1)
- Model type: Simple Deep Neural Network
  - 1-level model
  - 2-level model, similar to RMI [3]
- Training dataset: random sampling of 1,381 trajectories Q, threshold $H = \{0.001, 0.005, 0.01, 0.05, 0.1, 0.5, 1, 2.5, 5, 7.5, 10\}$, and distance: DTW
- Test dataset: 60 trajectories $\notin Q$ and $H' = \{0.075, 0.03, 0.4, 6\}$
- Evaluation Metric: precision, compared to ground truth (DITA)

[3] T. Kraska, A. Beutel, E. Chi, J. Dean and N. Polyzotis, "The Case for Learned Index Structures," in SIGMOD 2018

Experiment Setup

- Hardware
  - Intel(R) Core(TM) 3.60GHz
  - 16 GB RAM
- Software
  - Hadoop-2.6 for Windows, Spark 2.2.0, TensorFlow and TensorFlow Java API 1.13.1, and Python 3.6
  - Train the model in Python first, then call the model using Scala (*DITA is built using Scala)

Result



- Our developed model nearly achieved the ground truth performance within *p* variations
- The 2-level recursive model has better performance than the single model
  - Slightly similar to DITA original structure, however more complex

Discussion and Future Works

- This is still preliminary work
- The input of the ML model (the first, last, and pivot points) may not quite represent the trajectory for machine learning
  - Trajectory representation using Vector [6] & Cluster [7]
  - However, implementation for in-memory approach still need to be discussed

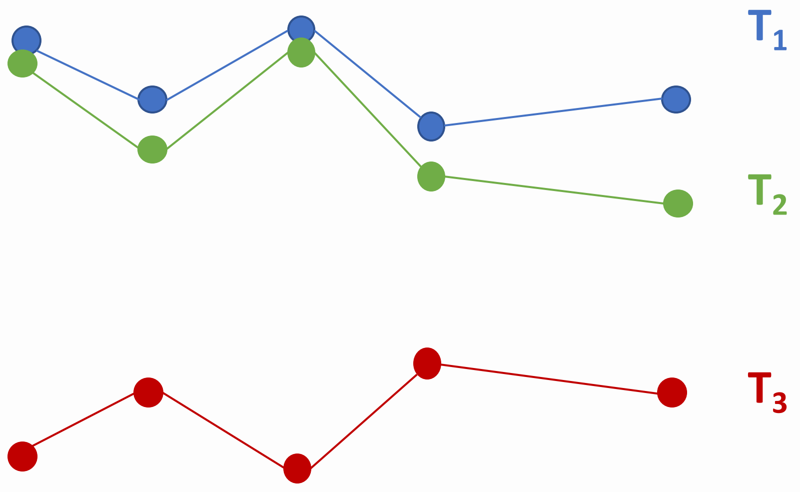
[6] X.Li, K.Zhao, G. Cong, C.S. Jensen, W. Wei, "Deep Representation Learning for Trajectory Similarity Computation," ICDE 2018.
[7] D. Yao, C. Zhang, Z. Zhu, J. Huang, J. Bi, "Trajectory Clustering via Deep Representation Learning," in IJCNN 2017.

Conclusion

- We developed a learned index approach for an existing similarity trajectory search indexing (DITA) to minimize the exploration of irrelevant partitions/cluster
- We developed a probabilistic distance, applicable to learned index, to model the similarity between trajectories

Thank you for your attention

Trajectory Similarity Search Query



Distance (DTW)	T_1	T_2	T_3
T_1	0	0.4	10
T_2	0.4	0	9
T_3	10	9	0

- Task:
  - Given a query trajectory T_Q
  - With a distance threshold of τ
  - Return all trajectories in set $\mathcal{T}$ whose distance $\leq \tau$

    $$SimTS_{\tau}^{T_Q} = \{\langle T_s \rangle, T_s \in \mathcal{T}\}$$
- Example query:
  - Given T_1 as query to set $\{T_1, T_2, T_3\}$
  - Using threshold τ =0.5
  - Result: $\{T_1, T_2\}$
- Learned index: learn the similarity relationship between the trajectories

Build Dataset for Similarity Search Query