Finding Structurally and Temporally Similar Trajectories in Graphs

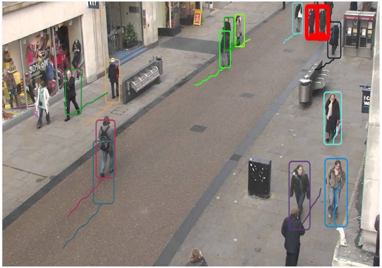
Roberto Grossi [1] **Andrea Marino** [2] **Shima Moghtasedi** [1]

[1]Dipartimento di Informatica, Università di Pisa, Italy

[2]Dipartimento di Statistica, Informatica, Applicazioni "G. Parenti", Università di Firenze, Italy

SEA 2020


EXCURSION
JOURNEY
ROAD
EXPLORE
ADVENTURE
ATTRACTIONS
WALK
CRUISE
TRANSPORT
GUIDE
HOLIDAY
AIRPLANE
TRIP
WALK
VACATION
EXPEDITION
QUEST
AGENCY
TOUR
TOURISM
VOYAGE
HOTEL
QUEST
TRAVEL
ROAD



OutLine

1. Modeling trajectories
2. Modeling similarity taking into account both time and space
3. Modeling query
4. Computing efficiently similarity

Part I

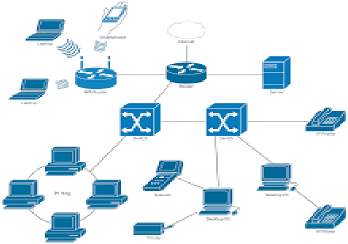
Modeling trajectories

Movement Representation

✓ Graph with location information

✓ Spatial-Temporal Trajectories

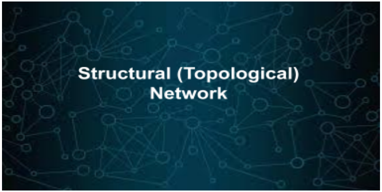
Movement Representation



✓ Graph with location information
✓ Spatial-Temporal Trajectories

Movement Representation

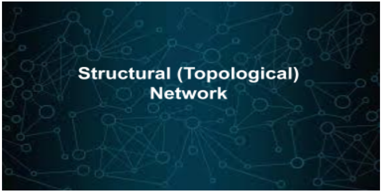
✓ Graph with location information
✓ Spatial-Temporal Trajectories



✓ Topology of the network with no spatial information
✓ Structural-Temporal Trajectories
✓ Low-dimensional indexing challenges
✓ Less overall size of data

Movement Representation

✓ Graph with location information
✓ Spatial-Temporal Trajectories



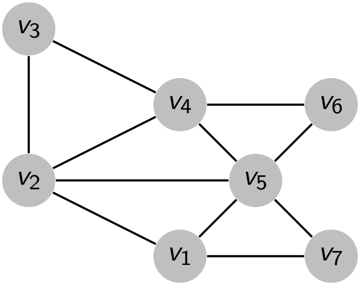
✓ Topology of the network with no spatial information
✓ Structural-Temporal Trajectories
✓ Low-dimensional indexing challenges
✓ Less overall size of data

Trajectory: sequence of nodes on networks with no spatial information.

Definition

Let $G = (V, E)$ is a connected and undirected graph
We use the topology of the graph G to define a trajectory.

A trajectory (Timed walk) is a time-stamped sequence of nodes

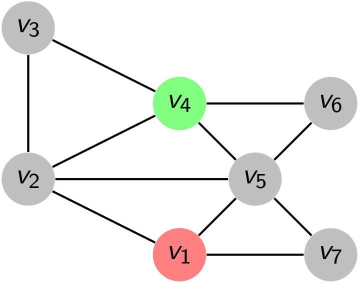


$T_1 = \langle \rangle$
$T_2 = \langle \rangle$

Definition

Let $G = (V, E)$ is a connected and undirected graph
We use the topology of the graph G to define a trajectory.

A trajectory (Timed walk) is a time-stamped sequence of nodes

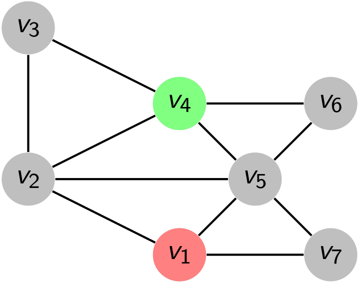


$t = 0$
$T_1 = \langle (v_1, [0, 0]) \rangle$
$T_2 = \langle (v_4, [0, 0]) \rangle$

Definition

Let $G = (V, E)$ is a connected and undirected graph
We use the topology of the graph G to define a trajectory.

A trajectory (Timed walk) is a time-stamped sequence of nodes

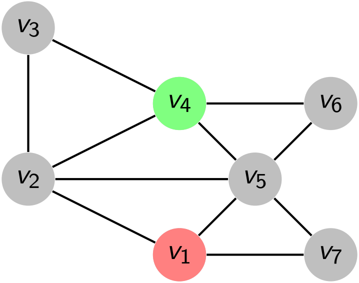


$t = 1$
$T_1 = \langle (v_1, [0, 1]) \rangle$
$T_2 = \langle (v_4, [0, 1]) \rangle$

Definition

Let $G = (V, E)$ is a connected and undirected graph
We use the topology of the graph G to define a trajectory.

A trajectory (Timed walk) is a time-stamped sequence of nodes

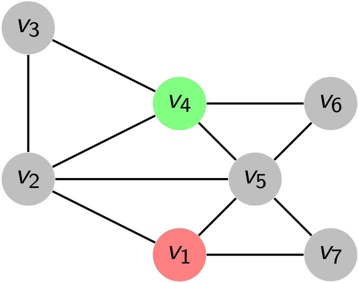


$t = 2$
$T_1 = \langle (v_1, [0, 2]) \rangle$
$T_2 = \langle (v_4, [0, 2]) \rangle$

Definition

Let $G = (V, E)$ is a connected and undirected graph
We use the topology of the graph G to define a trajectory.

A trajectory (Timed walk) is a time-stamped sequence of nodes

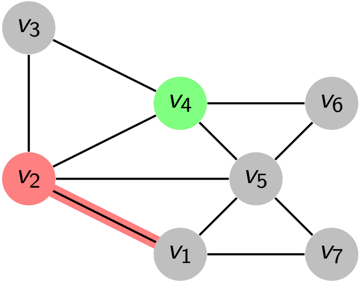


$t = 3$
$T_1 = \langle(v_1, [0, 3])\rangle$
$T_2 = \langle(v_4, [0, 3])\rangle$

Definition

Let $G = (V, E)$ is a connected and undirected graph
We use the topology of the graph G to define a trajectory.

A trajectory (Timed walk) is a time-stamped sequence of nodes

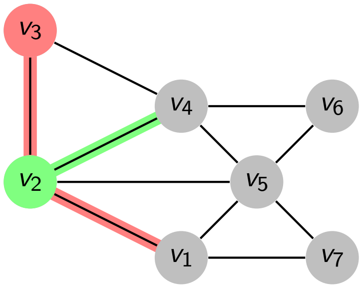


$t = 4$
$T_1 = \langle (v_1, [0, 3]), (v_2, [4, 4]) \rangle$
$T_2 = \langle (v_4, [0, 4]) \rangle$

Definition

Let $G = (V, E)$ is a connected and undirected graph
We use the topology of the graph G to define a trajectory.

A trajectory (Timed walk) is a time-stamped sequence of nodes

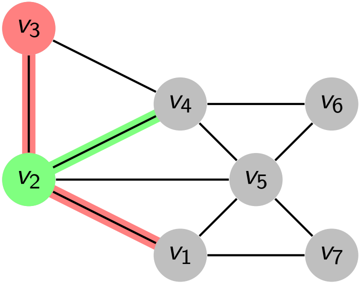


$t = 5$
$T_1 = \langle (v_1, [0,3]), (v_2, [4,4]), (v_3, [5,5]) \rangle$
$T_2 = \langle (v_4, [0,4])(v_2, [5,5]) \rangle$

Definition

Let $G = (V, E)$ is a connected and undirected graph
We use the topology of the graph G to define a trajectory.

A trajectory (Timed walk) is a time-stamped sequence of nodes

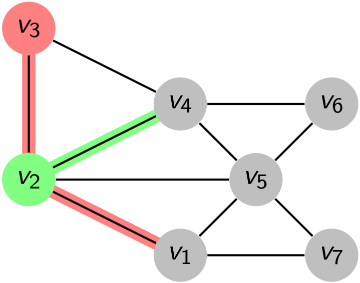


$t = 6$
$T_1 = \langle(v_1, [0, 3]), (v_2, [4, 4]), (v_3, [5, 6])\rangle$
$T_2 = \langle(v_4, [0, 4])(v_2, [5, 6])\rangle$

Definition

Let $G = (V, E)$ is a connected and undirected graph
We use the topology of the graph G to define a trajectory.

A trajectory (Timed walk) is a time-stamped sequence of nodes

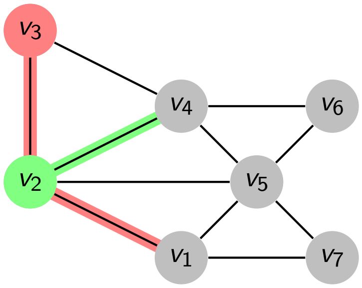


$t = 7$
$T_1 = \langle (v_1, [0, 3]), (v_2, [4, 4]), (v_3, [5, 7]) \rangle$
$T_2 = \langle (v_4, [0, 4])(v_2, [5, 7]) \rangle$

Definition

Let $G = (V, E)$ is a connected and undirected graph
We use the topology of the graph G to define a trajectory.

A trajectory (Timed walk) is a time-stamped sequence of nodes

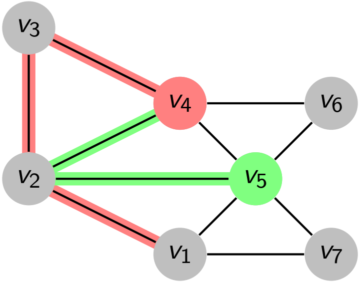


$t = 8$
$T_1 = \langle (v_1, [0, 3]), (v_2, [4, 4]), (v_3, [5, 8]) \rangle$
$T_2 = \langle (v_4, [0, 4])(v_2, [5, 8]) \rangle$

Definition

Let $G = (V, E)$ is a connected and undirected graph
We use the topology of the graph G to define a trajectory.

A trajectory (Timed walk) is a time-stamped sequence of nodes

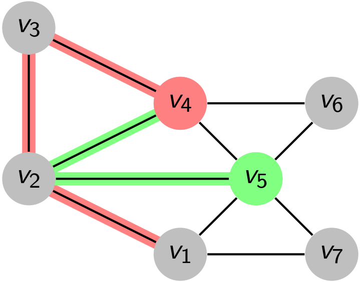


$t = 9$
$T_1 = \langle (v_1, [0, 3]), (v_2, [4, 4]), (v_3, [5, 8]), (v_4, [9, 9]) \rangle$
$T_2 = \langle (v_4, [0, 4])(v_2, [5, 8])(v_5, [9, 9]) \rangle$

Definition

Let $G = (V, E)$ is a connected and undirected graph
We use the topology of the graph G to define a trajectory.

A trajectory (Timed walk) is a time-stamped sequence of nodes

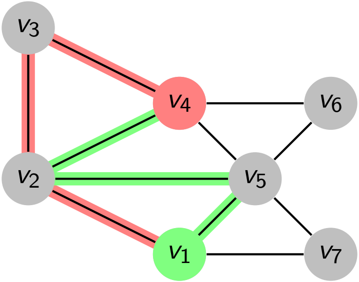


$t = 10$
$T_1 = \langle(v_1, [0, 3]), (v_2, [4, 4]), (v_3, [5, 8]), (v_4, [9, 10])\rangle$
$T_2 = \langle(v_4, [0, 4])(v_2, [5, 8])(v_5, [9, 10])\rangle$

Definition

Let $G = (V, E)$ is a connected and undirected graph
We use the topology of the graph G to define a trajectory.

A trajectory (Timed walk) is a time-stamped sequence of nodes

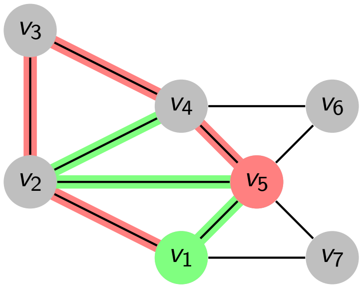


$t = 11$
$T_1 = \langle (v_1, [0,3]), (v_2, [4,4]), (v_3, [5,8]), (v_4, [9,11]) \rangle$
$T_2 = \langle (v_4, [0,4])(v_2, [5,8])(v_5, [9,10])(v_1, [11,11]) \rangle$

Definition

Let $G = (V, E)$ is a connected and undirected graph
We use the topology of the graph G to define a trajectory.

A trajectory (Timed walk) is a time-stamped sequence of nodes

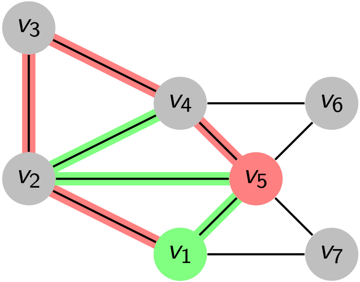


$t = 12$
$T_1 = \langle (v_1, [0,3]), (v_2, [4,4]), (v_3, [5,8]), (v_4, [9,11]), (v_5, [12,12]) \rangle$
$T_2 = \langle (v_4, [0,4])(v_2, [5,8])(v_5, [9,10])(v_1, [11,12]) \rangle$

Definition

Let $G = (V, E)$ is a connected and undirected graph
We use the topology of the graph G to define a trajectory.

A trajectory (Timed walk) is a time-stamped sequence of nodes

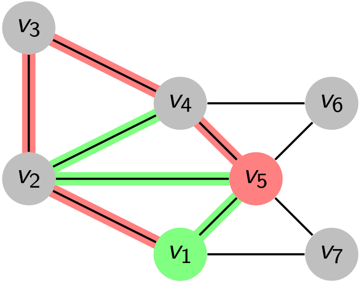


$t = 13$
$T_1 = \langle(v_1, [0, 3]), (v_2, [4, 4]), (v_3, [5, 8]), (v_4, [9, 11]), (v_5, [12, 13])\rangle$
$T_2 = \langle(v_4, [0, 4])(v_2, [5, 8])(v_5, [9, 10])(v_1, [11, 13])\rangle$

Definition

Let $G = (V, E)$ is a connected and undirected graph
We use the topology of the graph G to define a trajectory.

A trajectory (Timed walk) is a time-stamped sequence of nodes

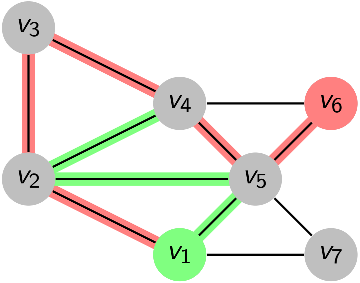


$t = 14$
$T_1 = \langle (v_1, [0, 3]), (v_2, [4, 4]), (v_3, [5, 8]), (v_4, [9, 11]), (v_5, [12, 14]) \rangle$
$T_2 = \langle (v_4, [0, 4])(v_2, [5, 8])(v_5, [9, 10])(v_1, [11, 14]) \rangle$

Definition

Let $G = (V, E)$ is a connected and undirected graph
We use the topology of the graph G to define a trajectory.

A trajectory (Timed walk) is a time-stamped sequence of nodes

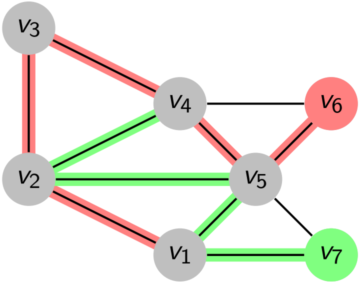


$t = 15$
$T_1 =$
$\langle (v_1, [0,3]), (v_2, [4,4]), (v_3, [5,8]), (v_4, [9,11]), (v_5, [12,14]), (v_6, [15,15]) \rangle$
$T_2 = \langle (v_4, [0,4])(v_2, [5,8])(v_5, [9,10])(v_1, [11,15]) \rangle$

Definition

Let $G = (V, E)$ is a connected and undirected graph
We use the topology of the graph G to define a trajectory.

A trajectory (Timed walk) is a time-stamped sequence of nodes

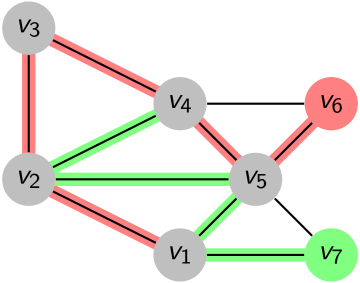


$t = 16$
$T_1 =$
$\langle(v_1, [0, 3]), (v_2, [4, 4]), (v_3, [5, 8]), (v_4, [9, 11]), (v_5, [12, 14]), (v_6, [15, 16])\rangle$
$T_2 = \langle(v_4, [0, 4])(v_2, [5, 8])(v_5, [9, 10])(v_1, [11, 15])(v_7, [16, 16])\rangle$

Definition

Let $G = (V, E)$ is a connected and undirected graph
We use the topology of the graph G to define a trajectory.

A trajectory (Timed walk) is a time-stamped sequence of nodes



$t = 17$
$T_1 =$
$\langle(v_1, [0, 3]), (v_2, [4, 4]), (v_3, [5, 8]), (v_4, [9, 11]), (v_5, [12, 14]), (v_6, [15, 17])\rangle$
$T_2 = \langle(v_4, [0, 4])(v_2, [5, 8])(v_5, [9, 10])(v_1, [11, 15])(v_7, [16, 17])\rangle$

Part II

Modeling Similarity

Similarity Requirements

TEMPORAL

ARBITRARY LENGTH

STRUCTURAL

FAST

GRAPH METRIC

PROXIMITY

Similarity Measures

State-of-the-art	**Temporal**	**Proximity**	**Properties**	**Input**	**Complexity time**
Won et al. [WKBL09]	×	×	Jaccard similarity based	Two strings as trajectories	$O(\ell^2)$
Xia et al. [XWZ+11]	✓	×	Jaccard similarity based	Two strings as trajectories	$O(\ell^2)$
Hwang et al. [HKL06]	✓	✓	Pair to pair distance computation only at **specific predefined points**	Two trajectories with **the same length**, implicitly	$\boldsymbol{O(\ell)}$
Tiakas et al. [TPN+06]	✓	✓	Spatial and temporal distance computation in separate way (Liner combination)	Two trajectories with **the same length**	$\boldsymbol{O(\ell)}$
Shang et al. [SDZ+14]	✓	✓	LCSS based Liner combination of spatial and temporal distance	Two trajectories	$O(\ell^2)$
Tiakas et al. [TR15]	✓	✓	Linear combination of spatial and temporal distance	A set of query points and a trajectory	$O(\ell^2)$

Network-based similarity measures for trajectories of ℓ nodes.

Spatiotemporal function	References
$\sigma * \mathcal{D}_s(Q, T) + (1 - \sigma) * \mathcal{D}_t(Q, T)$	[TPN+06, SDZ+14, SCW+17, TR15]
$(\mathcal{D}_s(Q, T) + \sigma * \mathcal{D}_t(Q, T))/2$	[CBKK07]
$(\mathcal{D}_s(Q, T)/\sigma + 1) * (\mathcal{D}_t(Q, T) + 1)$	[CBKK07]
$\mathcal{D}_s(Q, T) * \mathcal{D}_t(Q, T)$	[XWZ+11, SXW+15]
$\sigma * \mathcal{D}_s(Q, T) + \mathcal{D}_t(Q, T)$	[HKL06]

Spatio-temporal similarity functions: the parameter σ controls the relative importance of the spatial and temporal similarities.

Spatiotemporal function	References
$\sigma * \mathcal{D}_s(Q,T) + (1-\sigma) * \mathcal{D}_t(Q,T)$	[TPN+06, SDZ+14, SCW+17, TR15]
$(\mathcal{D}_s(Q,T) + \sigma * \mathcal{D}_t(Q,T))/2$	[CBKK07]
$(\mathcal{D}_s(Q,T)/\sigma + 1) * (\mathcal{D}_t(Q,T) + 1)$	[CBKK07]
$\mathcal{D}_s(Q,T) * \mathcal{D}_t(Q,T)$	[XWZ+11, SXW+15]
$\sigma * \mathcal{D}_s(Q,T) + \mathcal{D}_t(Q,T)$	[HKL06]

Spatio-temporal similarity functions: the parameter σ controls the relative importance of the spatial and temporal similarities.

|| Linear-time || Arbitrary Length || Proximity || Single Function ||

Constructing Similarity Function

Time Restricted Trajectory

Given a trajectory T and a time interval $t = [s, e]$, the time restricted trajectory $T[t]$ is the sequence of pairs $(u_i, t_i) \in T$ such that $t_i = [s_i, e_i]$ has overlap with $t = [s, e]$ (i.e. $t_i \cap t \neq 0$).

$T = \langle(v_1, [0, 3]), (v_2, [4, 4]), (v_3, [5, 8]), (v_4, [9, 11]), (v_5, [12, 14])\rangle$
For time interval $t = [4, 11]$, we have :
$T[t] = \langle(v_2, [4, 4]), (v_3, [5, 8]), (v_4, [9, 11])\rangle$

Similarity Function Definition

The distance between a *node* v and a *trajectory* T within a *time interval* t:

Node-Trajectory distance

$$dist(v, T, t) = \frac{\min_{(u_i, t_i) \in T[t]} d(v, u_i)}{D_G}$$

I. $dist(v, T, t) = 0 \iff \exists \; i \in t \; | T(i) = v.$

II. $dist(v, T, t) = 1 \iff \forall i \in t, \;\; d(v, T(i)) = D_G.$

Similarity Function Definition

The distance between a *node* v and a *trajectory* T within a *time interval* t:

Node-Trajectory distance

$$dist(v, T, t) = \frac{\min_{(u_i,t_i)\in T[t]} d(v, u_i)}{D_G}$$

Trajectory-Trajectory Similarity

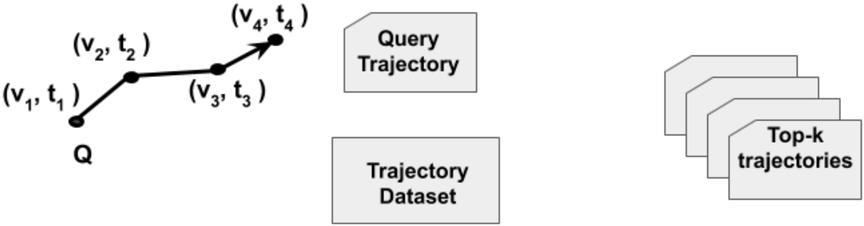
$$Sim(Q, T, t) = \frac{\sum_{(v_i,t_i)\in Q[t]} |t_i| \times e^{-dist(v_i,T,t_i)}}{|t|}$$

Four random trajectories in a dataset of trajectories moving in Milan. The trajectory with the red color is a query. The green trajectory is the most similar one to the query.

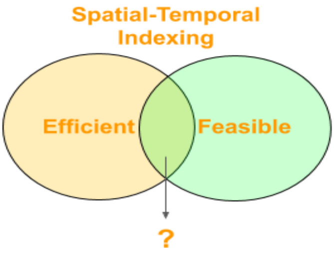
Part III

Modeling Query

k-MsTraj Query



NTrajI Indexing: Structural-Temporal Trajectories



Spatial Indexing

Variation of R-tree: [CSZ$^+$10, TZX$^+$11]

Temporal Indexing

Based on the B-tree: [LTCN13, PZO$^+$10]

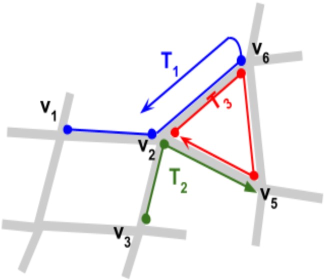
Neighborhood Trajectory Indexing (NTrajI)

Projection Set

Given the set of trajectories $\mathcal{T}$, for each $v \in V$ we define the projection set:

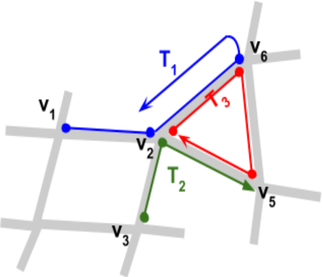
$\forall v \in V \; \forall T \in \mathcal{T} \rightarrow$
$S_v = \{(t, T) \mid (u, t) \in T \text{ and } u \in \{v\} \cup N(v) \text{ and } T \in \mathcal{T}\}$

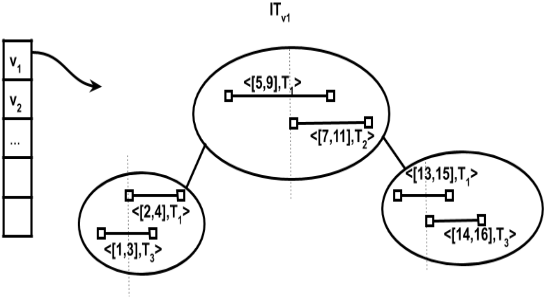
$\mathcal{T} = \{T_1, T_2, T_3\}$
$T_1 = \langle (v_1, [2, 4]), (v_2, [5, 9]), (v_4, [10, 12]),$
$(v_2, [13, 15]) \rangle$
$T_2 = \langle (v_3, [1, 6]), (v_2, [7, 11]), (v_5, [12, 16]) \rangle$
$T_3 = \langle (v_2, [1, 3]), (v_4, [4, 7]), (v_5, [8, 13]),$



NTrajI Structure

$S_{v_1} = \{([1,3], T_3), ([2,4], T_1), ([5,9], T_1),$
$([7,11], T_2), ([13,15], T_1), ([14,16], T_3)\}$





BaseLine

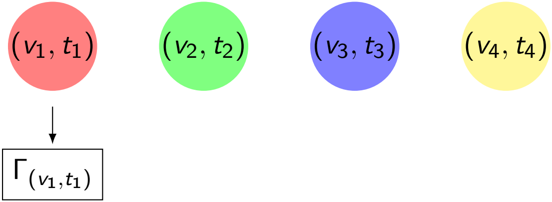
Input: Given a dataset $\mathcal{T}$ over the graph G,
a query trajectory Q and time interval t.
Output: k-MSTraj

BaseLine

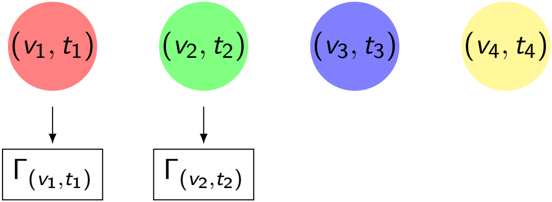
Input: Given a dataset $\mathcal{T}$ over the graph G,
a query trajectory Q and time interval t.
Output: k-MSTraj

(v_1, t_1) (v_2, t_2) (v_3, t_3) (v_4, t_4)

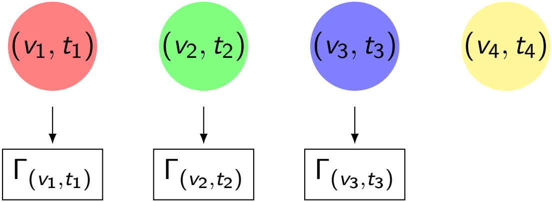
BaseLine



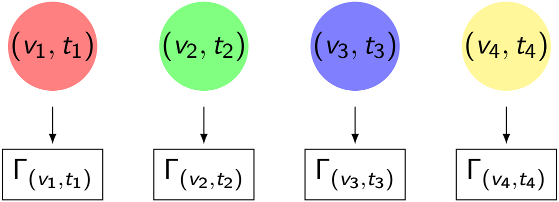
BaseLine



BaseLine

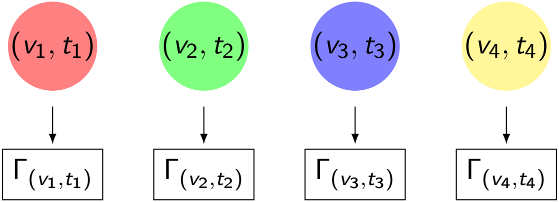


BaseLine



(v_1, t_1)
(v_2, t_2)
(v_3, t_3)
(v_4, t_4)
$\Gamma_{(v_1,t_1)}$
$\Gamma_{(v_2,t_2)}$
$\Gamma_{(v_3,t_3)}$
$\Gamma_{(v_4,t_4)}$

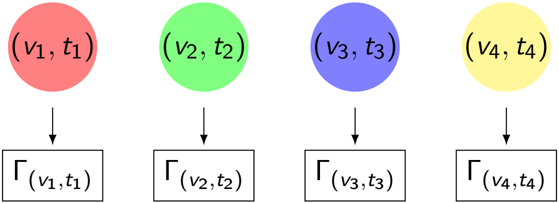
BaseLine



$$\boxed{\Gamma = \cup_{(v_i,t_i)\in Q}\Gamma_{(v_i,t_i)}}$$

BaseLine

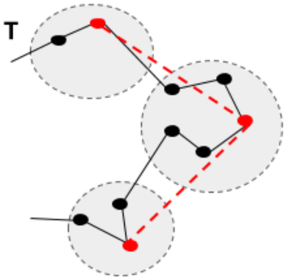
Input: Given a dataset $\mathcal{T}$ over the graph G,
a query trajectory Q and time interval t.
Output: k-MSTraj



$\boxed{\Gamma = \cup_{(v_i,t_i)\in Q}\Gamma_{(v_i,t_i)}}$

For each trajectory $T \in \Gamma$: computing $Sim(Q, T, t)$

k-MsTraj: Approach



Shrinking approach

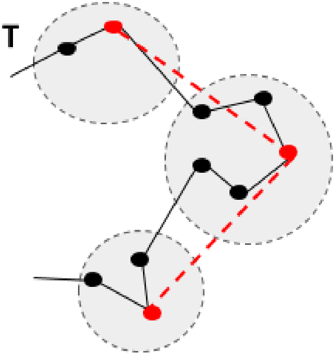
Idea

- ✓ Presenting a comprehensive representation of trajectories
- ✓ Reducing the number of shortest path distance computations
- ✓ Reducing query processing time

Part IV

Efficiently Query Processing: Shrinking Technique

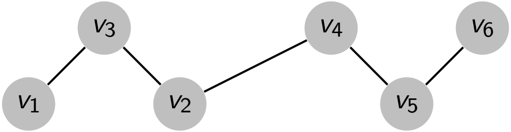
Shrinking Technique



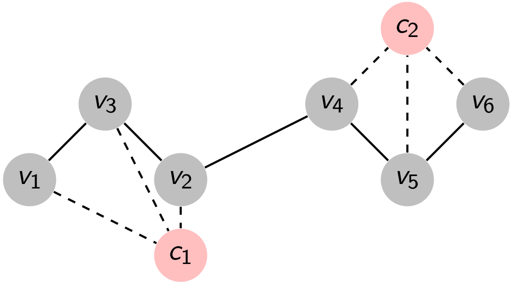
[ANFA15]

- ✓ Approximate solution for k-NN trajectories to the set of query points.
- ✓ Estimate all query node as the centroid of the convex hull of query points.
- ✓ Spatial domain.

Shrinking Approach: Basic Idea



Shrinking Approach: Basic Idea



Voronoi Diagram for graphs (*VDG*)

- ✓ $G = (V, E)$, let V be a set of n vertices on G
- ✓ Center nodes: $\mathcal{C} = \{c_1, c_2, \ldots c_h\} \subset V$ $\boxed{\text{Most frequent nodes in } V}$
- ✓ A node $u \in V$ is in the group corresponding to a center $c_i \in \mathcal{C}$ if $d(u, c_i) \leq d(u, c_j)$ for each $c_j \in \mathcal{C}$ with $i \neq j$

Voronoi Diagram for graphs (*VDG*)

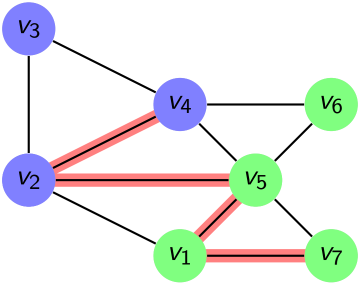
- ✓ $G = (V, E)$, let V be a set of n vertices on G
- ✓ Center nodes: $\mathcal{C} = \{c_1, c_2, \dots c_h\} \subset V$ $\boxed{\text{Most frequent nodes in } V}$
- ✓ A node $u \in V$ is in the group corresponding to a center $c_i \in \mathcal{C}$ if $d(u, c_i) \leq d(u, c_j)$ for each $c_j \in \mathcal{C}$ with $i \neq j$

$\boxed{\text{Precomputing the shortest path distances in linear time}}$

$\boxed{O(n^2) \text{ shortest path distance precomputations in [SDZ}^{+}\text{14]}}$

Shrunk Trajectory

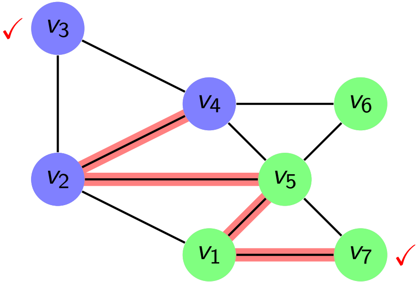
Shrunk trajectory $\hat{T}$ is shrink(T'), which recursively merges any pair $(c_i, t_i), (c_{i+1}, t_{i+1}) \in T'$ as $(c_i, t_i + t_{i+1})$ when $c_i = c_{i+1}$ and t_i, t_{i+1} are two consecutive time intervals. T' is trajectory T represented by $c_i \in \mathcal{C}$.



$$T = \langle (v_4, [0, 4]), (v_2, [5, 8]), (v_5, [9, 12]), (v_1, [13, 15]), (v_7, [16, 17]) \rangle$$

Shrunk Trajectory

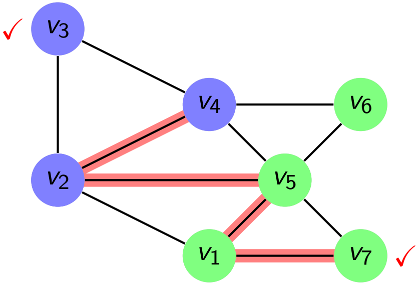
Shrunk trajectory $\hat{T}$ is shrink(T'), which recursively merges any pair $(c_i, t_i), (c_{i+1}, t_{i+1}) \in T'$ as $(c_i, t_i + t_{i+1})$ when $c_i = c_{i+1}$ and t_i, t_{i+1} are two consecutive time intervals. T' is trajectory T represented by $c_i \in \mathcal{C}$.



$T = \langle (v_4, [0,4]), (v_2, [5,8]), (v_5, [9,12]), (v_1, [13,15]), (v_7, [16,17]) \rangle$
$T' = \langle (v_3, [0,4]), (v_3, [5,8]), (v_3, [9,12]), (v_7, [13,15]), (v_7, [16,17]) \rangle$

Shrunk Trajectory

Shrunk trajectory $\hat{T}$ is shrink(T'), which recursively merges any pair $(c_i, t_i), (c_{i+1}, t_{i+1}) \in T'$ as $(c_i, t_i + t_{i+1})$ when $c_i = c_{i+1}$ and t_i, t_{i+1} are two consecutive time intervals. T' is trajectory T represented by $c_i \in \mathcal{C}$.



$T = \langle (v_4, [0, 4]), (v_2, [5, 8]), (v_5, [9, 12]), (v_1, [13, 15]), (v_7, [16, 17]) \rangle$
$T' = \langle (v_3, [0, 4]), (v_3, [5, 8]), (v_3, [9, 12]), (v_7, [13, 15]), (v_7, [16, 17]) \rangle$
$shrink(T') = \hat{T} = \langle (v_3, [0, 12]), (v_7, [13, 17]) \rangle$

VoTrajI Construction: Extension of NTrajI

Given the set of trajectories $\mathcal{T}$, for each Voronoi center node $c \in \mathcal{C}$ we define the projection set:

$\forall c \in \mathcal{C}; \forall T \in \mathcal{T} \rightarrow S_c = \{(t, T) \mid (v, t) \in T \text{ and } v \in g \text{ and } g.\mathcal{C} = c \text{ and } T \in \mathcal{T}\}$

$\forall c \in \mathcal{C} \rightarrow IT_c$: an Interval tree storing $(t, T) \in S_c$

Query Processing

For a query trajectory Q within time interval t,

First: Making shrunk trajectory of $Q[t] \rightarrow \hat{Q}[t]$

Second: By making use of VoNTrajI, for each $(c_i, t_i) \in \hat{Q}[t]$ find trajectories traversing the nodes belonging to the group with center c_i within t_i $\rightarrow$ $\boxed{\tilde{\Gamma} = \cup_{(c_i,t_i)\in\hat{Q}[t]}\Gamma_{(c_i,t_i)}}$

SHQ

For each $T \in \tilde{\Gamma}$:

(I) Compute $Sim(\hat{Q}, T, t)$, as an estimation of $Sim(Q, T, t)$

SHQT

For each $T \in \tilde{\Gamma}$:

(I) Making shrunk trajectory of $T \rightarrow (\hat{T})$

(II) Compute $Sim(\hat{Q}, \hat{T}, t)$, as an estimation of $Sim(Q, T, t)$

SHQ

For each $T \in \tilde{\Gamma}$:

$$Sim(Q, T, t) \rightarrow Sim(\hat{Q}, T, t)$$

$$\boxed{d(u, v)_{u \in T, v \in Q} \rightarrow d(u, c_j)_{u \in T, c_j \in \hat{Q}}}$$

$$\boxed{d \rightarrow \bar{d}}$$

Lemma

$$\bar{d} \leq d \leq 3\bar{d}$$

SHQT

For each $T \in \tilde{\Gamma}$:

$$Sim(Q, T, t) \rightarrow Sim(\hat{Q}, \hat{T}, t)$$

$$\boxed{d(u, v)_{u \in T, v \in Q} \rightarrow d(c_i, c_j)_{c_i \in \hat{T}, c_j \in \hat{Q}}}$$

$$\boxed{d \rightarrow \tilde{d}}$$

Lemma

$$\tilde{d} \leq 2\bar{d}$$

SHQ

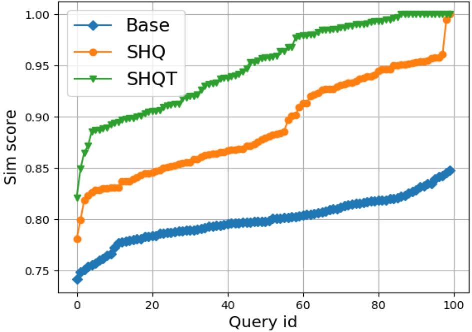
Lemma

$$\bar{d} \leq d \leq 3\bar{d}$$

SHQT

Lemma

$$\tilde{d} \leq 2\bar{d}$$



Experiments: Datasets

Dataset Name	#trajectories	#nodes	#edges	Diameter
Facebook Dataset	1000	4039	88234	8
Milan Dataset	16166	3000	130071	5
Rome Dataset	7755	473	10524	6

Summary of Datasets

Experimental Evaluation

Q1: How fast is getting the answer for a query, i.e. how much is the query time?

Q2: How fast is the preprocessing time?

Q3: How good is the quality of the solution found if compared with the exact solution?

Experimental Evaluation

Q1: How fast is getting the answer for a query, i.e. how much is the query time?

Q2: How fast is the preprocessing time?

Q3: How good is the quality of the solution found if compared with the exact solution?

Running Time: Q1

Datasets	BASE	SHQ	SHQT
Facebook	1.09	0.74	0.43
Milan	380.03	376.15	85.48
Rome	26.19	19.42	15.83

The average time for answering a query for each proposed method on each dataset

Datasets	BASE	SHQ	SHQT
Milan	9786.39	9968.98	9968.98
Rome	7504.37	6569.84	6569.84

The average number of trajectories in the candidate set in each method.

Experimental Evaluation

Q1: How fast is getting the answer for a query, i.e. how much is the query time?

Q2: How fast is the preprocessing time?

Q3: How good is the quality of the solution found if compared with the exact solution?

Preprocessing Time: Q2

Dataset	NTrajI	VoTrajI	Distance precomputing	Shrinking trajectories and building Voronoi diagram
Facebook	185.19	0.51	0.48	0.62
Milan	1716.44	13.50	0.27	10.21
Rome	69.25	0.24	0.005	0.56

Preprocessing time (in sec.)

Experimental Evaluation

Q1: How fast is getting the answer for a query, i.e. how much is the query time?

Q2: How fast is the preprocessing time?

Q3: How good is the quality of the solution found if compared with the exact solution?

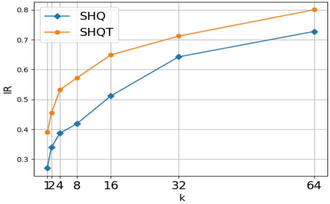
Precision Evaluation: Q3

$\gamma_1 \rightarrow$ Exact output set of k-MsTraj query
$\gamma_2 \rightarrow$ Estimated output set of k-MsTraj query

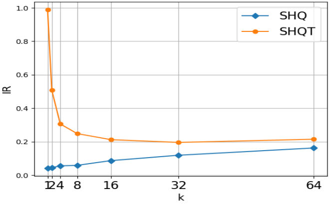
$$\mathrm{SSR}(\gamma_1, \gamma_2) = \frac{\sum_{T \in \gamma_1} Sim(Q, T, t)}{\sum_{S \in \gamma_2} Sim(Q, S, t)}$$

$$\mathrm{IR}(\gamma_1, \gamma_2) = \frac{|\gamma_1 \cap \gamma_2|}{k}.$$

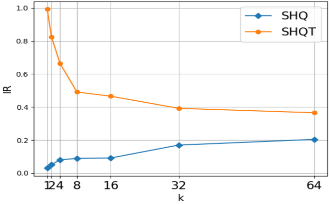
IR Ratio



Facebook



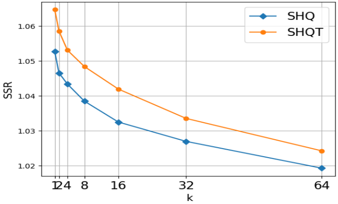
Milan



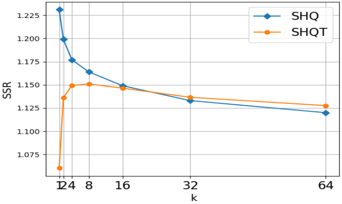
Rome

IR values show better performance for SHQT in comparison with SHQ.

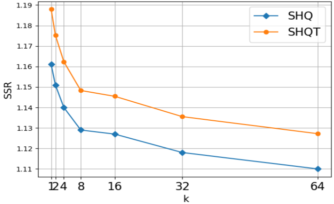
SSR Ratio



Facebook



Milan



Rome

SSR values for both SHQ and SHQT are almost close to 1.

Conclusion

- ✓ SHQ and SHQT are effective w.r.t the BaseLine method.
- ✓ SHQT provides a fast solution with an acceptable precision.
- ✓ SHQ is more efficient, querying long trajectories
- ✓ SHQ is more efficient, facing with a set of long trajectories

References I

Mohammad Reza Abbasifard, Hassan Naderi, Zohreh Fallahnejad, and Omid Isfahani Alamdari, *Approximate aggregate nearest neighbor search on moving objects trajectories*, Journal of Central South University (2015).

Jae-Woo Chang, Rabindra Bista, Young-Chang Kim, and Yong-Ki Kim, *Spatio-temporal similarity measure algorithm for moving objects on spatial networks*, Springer, 2007.

Zaiben Chen, Heng Tao Shen, Xiaofang Zhou, Yu Zheng, and Xing Xie, *Searching trajectories by locations: an efficiency study*, Proceedings of the 2010 ACM SIGMOD International Conference on Management of data, ACM, 2010, pp. 255–266.

Jung-Rae Hwang, Hye-Young Kang, and Ki-Joune Li, *Searching for similar trajectories on road networks using spatio-temporal similarity*, East European Conference on Advances in Databases and Information Systems, Springer, 2006, pp. 282–295.

Wuman Luo, Haoyu Tan, Lei Chen, and Lionel M Ni, *Finding time period-based most frequent path in big trajectory data*, Proceedings of the 2013 ACM SIGMOD international conference on management of data, ACM, 2013.

Iulian Sandu Popa, Karine Zeitouni, Vincent Oria, Dominique Barth, and Sandrine Vial, *Parinet: A tunable access method for in-network trajectories*, 2010 IEEE 26th International Conference on Data Engineering (ICDE 2010), IEEE, 2010, pp. 177–188.

Shuo Shang, Lisi Chen, Zhewei Wei, Christian S Jensen, Kai Zheng, and Panos Kalnis, *Trajectory similarity join in spatial networks*, Proceedings of the VLDB Endowment (2017).

Shuo Shang, Ruogu Ding, Kai Zheng, Christian S Jensen, Panos Kalnis, and Xiaofang Zhou, *Personalized trajectory matching in spatial networks*, The VLDB Journal **23** (2014), no. 3, 449–468.

References II

- Wenqiang Sha, Yingyuan Xiao, Hongya Wang, Yukun Li, and Xiaoye Wang, *Searching for spatio-temporal similar trajectories on road networks using Network Voronoi Diagram*, Springer, 2015.
- Eleftherios Tiakas, Apostolos N Papadopoulos, Alexandros Nanopoulos, Yannis Manolopoulos, Dragan Stojanovic, and Slobodanka Djordjevic-Kajan, *Trajectory similarity search in spatial networks*, IEEE, 2006.
- Eleftherios Tiakas and Dimitrios Rafailidis, *Scalable trajectory similarity search based on locations in spatial networks*, Model and Data Engineering, Springer, 2015, pp. 213–224.
- Lu-An Tang, Yu Zheng, Xing Xie, Jing Yuan, Xiao Yu, and Jiawei Han, *Retrieving k-nearest neighboring trajectories by a set of point locations*, International Symposium on Spatial and Temporal Databases, Springer, 2011, pp. 223–241.
- Jung-Im Won, Sang-Wook Kim, Ji-Haeng Baek, and Junghoon Lee, *Trajectory clustering in road network environment*, IEEE, 2009.
- Ying Xia, Guo-Yin Wang, Xu Zhang, Gyoung-Bae Kim, and Hae-Young Bae, *Spatio-temporal similarity measure for network constrained trajectory data*, International Journal of Computational Intelligence Systems 4 (2011), no. 5, 1070–1079.


THANK YOU