

Analyse et fouille de données de trajectoires d'objets mobiles

Thèse présentée et soutenue publiquement par
Mohamed Khalil EL MAHRSI

30 septembre 2013

Devant le jury composé de :

M. Talel ABDESSALEM	(Président du jury)
Mme Barbara HAMMER	(Rapporteur)
Mme Karine ZEITOUNI	(Rapporteur)
M. Pierre BORGNAT	(Examineur)
M. Etienne CÔME	(Examineur)
M. Ludovic DENOYER	(Examineur)
M. Cédric DU MOUZA	(Examineur)
M. Fabrice ROSSI	(Directeur de thèse)



The Traffic Congestion Problem

- Traffic congestion and road jams
 - Frustrating travel delays
 - Economical losses
 - Environmental damage
- Countermeasures are needed
 - Infrastructure improvement
 - Prohibiting/favoring specific routes
- Based on the analysis of drivers' behavior



How is Road Traffic Monitored?

- Traffic counters/recorders
 - Expensive
 - Partially deployed
 - Count traffic on their local section
- Consequences:
 - Incomplete vision of traffic
 - A valuable information is missed: vehicles' identities



Main Motivation: Trajectory Analysis as a Complement?

- Why not collect the trajectories of vehicles moving on the road network...
 - Many fleet management companies already do this
 - Commuters can contribute their trajectories

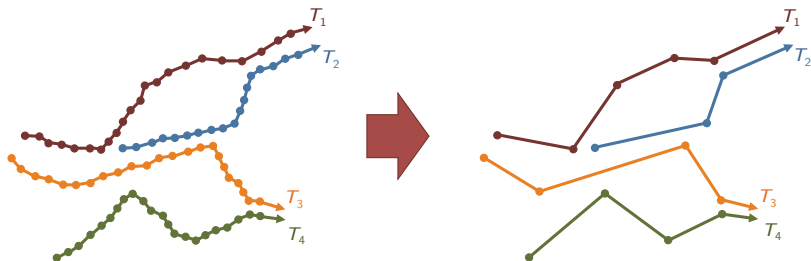
Main Motivation: Trajectory Analysis as a Complement?

- ... and analyze them to discover
 - Groups of vehicles that followed the same routes
 - Groups of roads that are often traveled together during a considerable number of commutes
 - Etc.



But...

- Modern devices can sample their positions at high rates
 - At such rates, the data are inherently redundant
- Transmitting and storing the entirety of the trajectories are impractical
 - Important space requirements
 - Computational overheads
- We have to intelligently reduce the size of the data



Research Problems Explored in this Thesis

Main objective:

Clustering Trajectory Data in Road Network Environments

How to discover meaningful groupings of “similar” trajectories and road segments in the specific context of road networks?

But first, a small detour:

Sampling Trajectory Data Streams

How to reduce the size of trajectory data streams while trying to preserve the most of their spatiotemporal features?

- 1 Context and Motivations
- 2 Sampling Trajectory Data Streams
- 3 Graph-Based Clustering of Network-Constrained Trajectory Data
- 4 Co-Clustering Network-Constrained Trajectory Data
- 5 Conclusions, Future Work and Open Issues

- 1 Context and Motivations
- 2 Sampling Trajectory Data Streams**
- 3 Graph-Based Clustering of Network-Constrained Trajectory Data
- 4 Co-Clustering Network-Constrained Trajectory Data
- 5 Conclusions, Future Work and Open Issues

(Raw) Trajectory

A trajectory T is a series of discrete, timestamped positions:

$$T = \langle id, \{P_1(t_1, x_1, y_1), P_2(t_2, x_2, y_2), \dots, P_i(t_i, x_i, y_i), \dots\} \rangle$$

- id : identifier
- t_i : timestamp (time of capture)
- (x_i, y_i) : coordinates (in the Euclidean space)

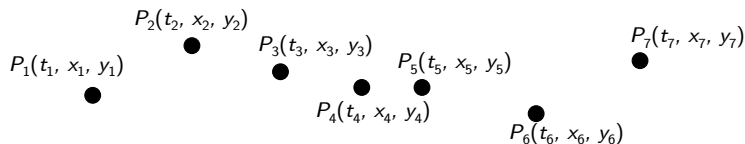


Figure : Illustration of a raw trajectory

(Raw) Trajectory

A trajectory T is a series of discrete, timestamped positions:

$$T = \langle id, \{P_1(t_1, x_1, y_1), P_2(t_2, x_2, y_2), \dots, P_i(t_i, x_i, y_i), \dots\} \rangle$$

- id : identifier
- t_i : timestamp (time of capture)
- (x_i, y_i) : coordinates (in the Euclidean space)
- Interpolation is used to approximate missing positions

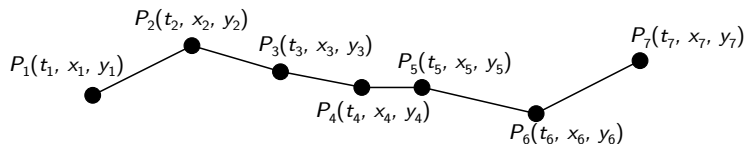


Figure : Illustration of a linearly-interpolated trajectory

Compressed (Sampled) Trajectory

Given a trajectory T , a compressed trajectory T_C of T is a subset of the original points forming T , such as:

- T_C covers T from start to finish
- $\forall P_i \in T_C, P_i \in T$

- Objectives
 - Reduce data size (obviously)
 - Small, preferably configurable approximation errors
- Constraints
 - On-the-fly processing
 - Low computational complexity
 - Low in-memory complexity

- Classic sampling techniques are inadequate
 - They overlook the spatiotemporal properties of the trajectories
- Two types of trajectory oriented sampling techniques
 - Configurable approximation errors but high complexity
 - Low complexity but no guarantees for approximation errors
- To the best of our knowledge: no approaches combining low complexity and configurable approximation errors

The Spatiotemporal Stream Sampling (STSS) Algorithm

[El Mahrsi et al., 2010]

- Intuition: use linear prediction to guess forthcoming positions
- The accuracy of the prediction (w.r.t. a threshold d_{Thres}) guides the sampling process

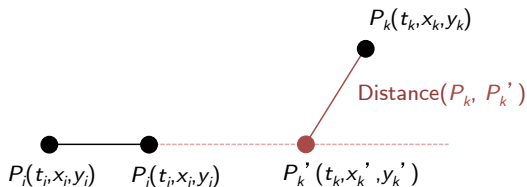


Figure : Linear prediction of incoming positions

P_1
○

Legend: -○- real trajectory —●— sampled trajectory prediction

Figure : Illustration of the functioning of the STSS algorithm

STSS: How it Works

P_1



Legend: -○- real trajectory —●— sampled trajectory prediction

Figure : Illustration of the functioning of the STSS algorithm

STSS: How it Works



Legend: -○- real trajectory —●— sampled trajectory prediction

Figure : Illustration of the functioning of the STSS algorithm

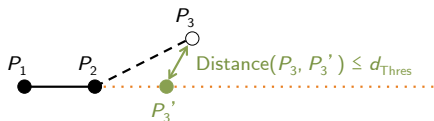
STSS: How it Works



Legend: -○- real trajectory —●— sampled trajectory prediction

Figure : Illustration of the functioning of the STSS algorithm

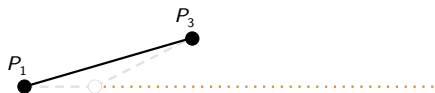
STSS: How it Works



Legend: -○- real trajectory —●— sampled trajectory prediction

Figure : Illustration of the functioning of the STSS algorithm

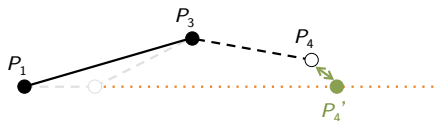
STSS: How it Works



Legend: -○- real trajectory —●— sampled trajectory prediction

Figure : Illustration of the functioning of the STSS algorithm

STSS: How it Works



Legend: -○- real trajectory —●— sampled trajectory prediction

Figure : Illustration of the functioning of the STSS algorithm

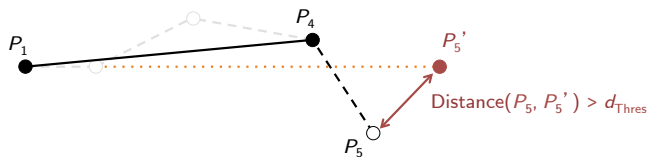
STSS: How it Works



Legend: -○- real trajectory —●— sampled trajectory prediction

Figure : Illustration of the functioning of the STSS algorithm

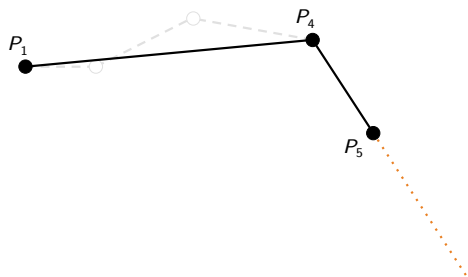
STSS: How it Works



Legend: -○- real trajectory ●- sampled trajectory prediction

Figure : Illustration of the functioning of the STSS algorithm

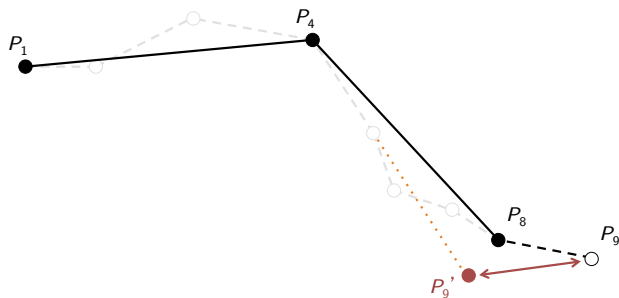
STSS: How it Works



Legend: -○- real trajectory —●— sampled trajectory prediction

Figure : Illustration of the functioning of the STSS algorithm

STSS: How it Works



Legend: -○- real trajectory —●— sampled trajectory prediction

Figure : Illustration of the functioning of the STSS algorithm

STSS: How it Works

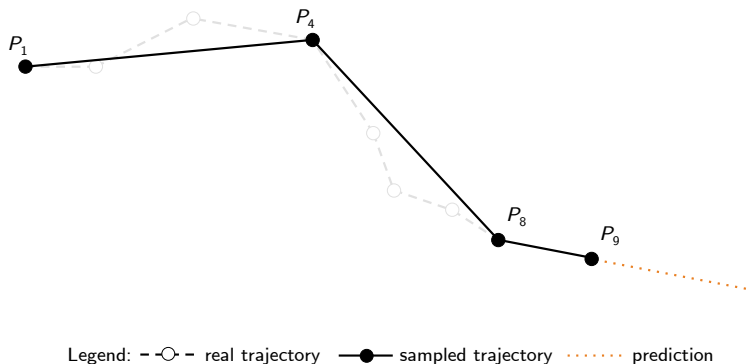
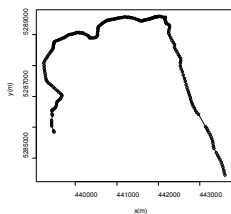
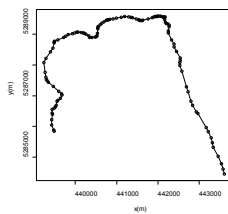


Figure : Illustration of the functioning of the STSS algorithm

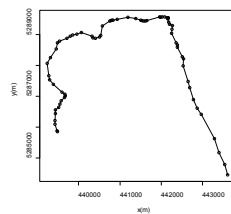
STSS in Action



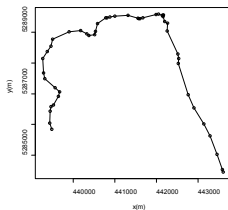
(a) Original trajectory
(228 points)



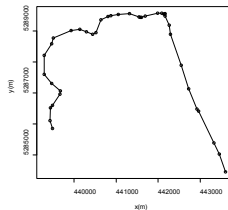
(b) Tolerated error: 10m
(117 points|comp. ratio: 1.9:1)



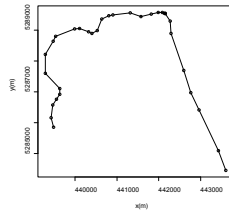
(c) Tolerated error: 50m
(72 points|comp. ratio: 3.2:1)



(d) Tolerated error: 100m
(49 points|comp. ratio: 4.6:1)



(e) Tolerated error: 150m
(40 points|comp. ratio: 5.7:1)



(f) Tolerated error: 200m
(32 points|comp. ratio: 7.1:1)

Figure : Example of a trajectory sampled with different error tolerances

- Single-pass, on-the-fly algorithm
- Linear computational complexity
- Constant in-memory complexity
- Easy to configure (only one parameter)
- Guaranteed upper bound for compression errors

Experimental Results: Comparison with TD-TR and OPW-TR [Meratnia and de By, 2004]

- Dataset
 - 5263 trajectories
 - 367691 data points (1 position/15 sec)
- The competition
 - TD-TR: offline, recursive partitioning, quadratic complexity
 - OPW-TR: on-the-fly, opening window, quadratic complexity
- Evaluation criteria
 - Percentage of retained data = $\frac{\text{size of the output data}}{\text{size of the input data}}$
 - Approximation error (distance between real points and their approximation)

Experimental Results: Percentage of Retained Data

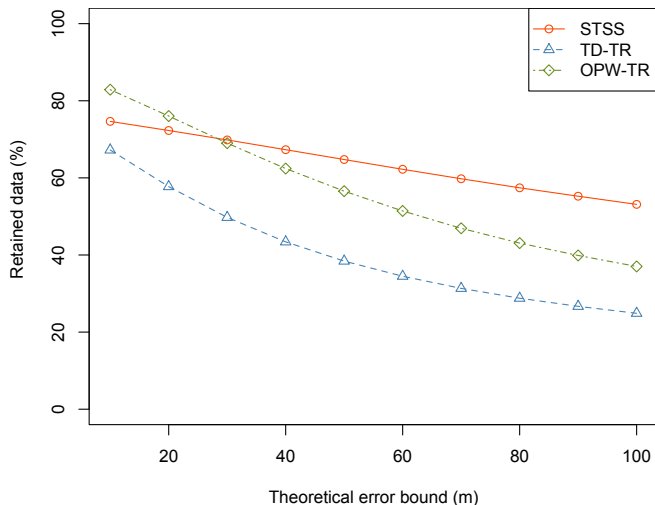


Figure : Percentages of retained data achieved by STSS, TD-TR and OPW-TR for different error tolerances

Experimental Results: Approximation Errors

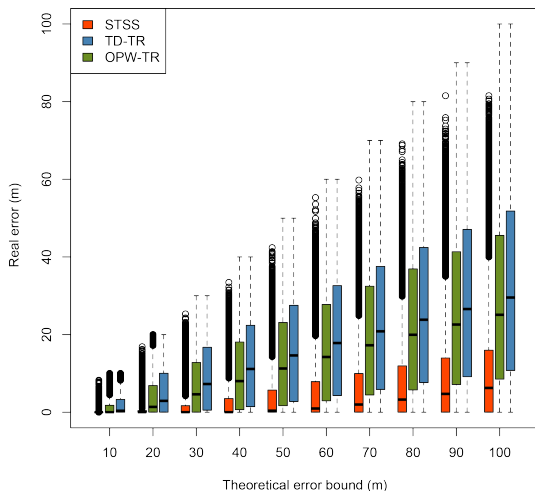


Figure : Distribution of the approximation errors resulting from applying STSS, TD-TR and OPW-TR for different error tolerances

Outline

- 1 Context and Motivations
- 2 Sampling Trajectory Data Streams
- 3 Graph-Based Clustering of Network-Constrained Trajectory Data**
- 4 Co-Clustering Network-Constrained Trajectory Data
- 5 Conclusions, Future Work and Open Issues

Existing Work on Trajectory Clustering

- Two main research areas
 - Distance and similarity measures
 - Clustering algorithms
- In both areas
 - For trajectories moving freely in a Euclidean space
 - For network-constrained trajectories
- Observations on existing trajectory clustering techniques
 - Density-based clustering
 - Flat clustering
 - A promising new trend: graph-based analysis [Guo et al., 2010]

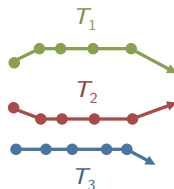


Figure : Effect of the underlying network on trajectory similarity

Existing Work on Trajectory Clustering

- Two main research areas
 - Distance and similarity measures
 - Clustering algorithms
- In both areas
 - For trajectories moving freely in a Euclidean space
 - For network-constrained trajectories
- Observations on existing trajectory clustering techniques
 - Density-based clustering
 - Flat clustering
 - A promising new trend: graph-based analysis [Guo et al., 2010]

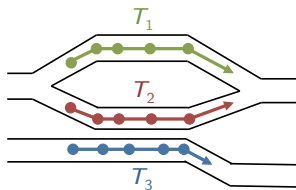


Figure : Effect of the underlying network on trajectory similarity

Road Network

The road network is represented as a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{S})$

- Vertices (\mathcal{V}): intersections and terminal points
- Edges (\mathcal{S}): road segments (with travel direction)

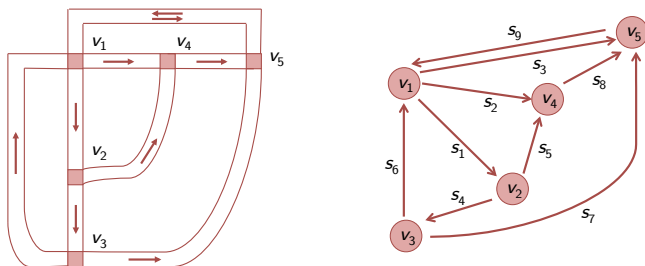


Figure : A road network and its graph representation

(Network-Constrained) Trajectory

A trajectory T is represented symbolically, as the sequence of traveled road segments:

$$T = \langle id, \{s_1, s_2, \dots, s_l\} \rangle$$

- $\forall 1 \leq i < l, s_i$ and s_{i+1} are connected

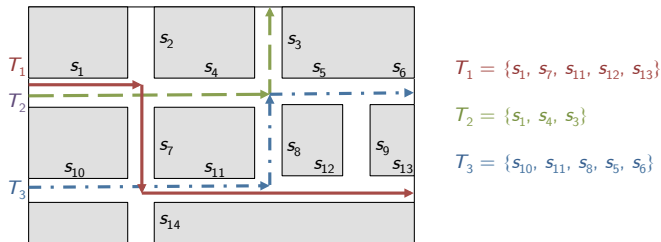


Figure : Example of three trajectories moving on a road network

Measuring the Similarity Between Trajectories

[El Mahrsi and Rossi, 2012a, El Mahrsi and Rossi, 2012c]

- Cosine similarity is used to measure the resemblance between trajectories

$$\text{Similarity}(T_i, T_j) = \frac{T_i \cdot T_j}{\|T_i\| \|T_j\|} = \frac{\sum_{s \in \mathcal{S}} \omega_{s, T_i} \times \omega_{s, T_j}}{\sqrt{\sum_{s \in \mathcal{S}} \omega_{s, T_i}^2} \times \sqrt{\sum_{s \in \mathcal{S}} \omega_{s, T_j}^2}}$$

- Road segments are weighted based on:
 - Their spatial length
 - Their frequency in the set of trajectories \mathcal{T}

$$\omega_{s, T} = \frac{n_{s, T} \times \text{length}(s)}{\sum_{s' \in \mathcal{T}} n_{s', T} \times \text{length}(s')} \times \log \frac{|\mathcal{T}|}{|\{T_i : s \in T_i\}|}$$

Trajectory Similarity Graph

- A weighted graph $\mathcal{G}_T(\mathcal{T}, \mathcal{E}_T, \mathcal{W}_T)$ is used to model relationships between trajectories

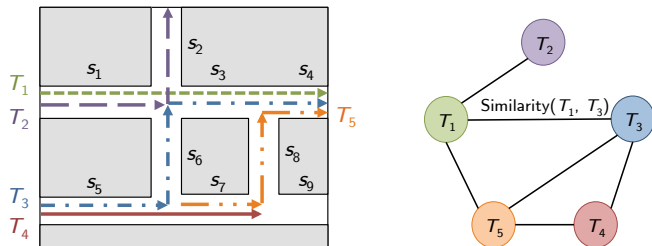


Figure : Example of a trajectory similarity graph

Clustering the Similarity Graph

- We used an implementation of the algorithm in [Noack and Rotta, 2009]
 - Based on modularity optimization [Newman, 2006]
 - Greedy hierarchical agglomerative clustering
 - Combined with multi-level refinement
- Input: trajectory similarity graph
- Output: a hierarchy of nested vertex (trajectory) clusters

Case Study: The Data



(a) 14 trajectories



(b) 19 trajectories



(c) 20 trajectories



(d) 20 trajectories



(e) 12 trajectories

Figure : The case study dataset is formed of 85 artificial trajectories divided into 5 pre-established and interacting clusters

Case Study: Hierarchy of Trajectory Clusters

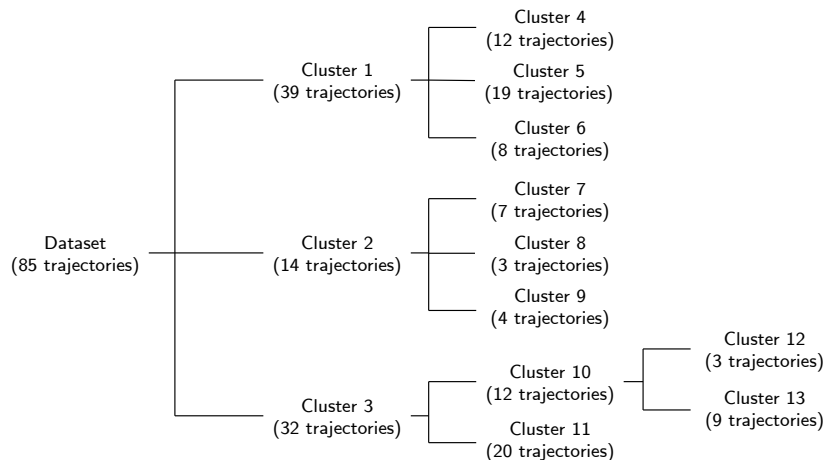


Figure : Hierarchy of trajectory clusters discovered through graph-based clustering

Case Study: High Level Trajectory Clusters



(a) Cluster 1
(39 trajectories)



(b) Cluster 2
(14 trajectories)



(c) Cluster 3
(32 trajectories)

Figure : Trajectory clusters in the highest level of hierarchy

Case Study: Refinement of Trajectory Clusters



(a) Cluster 1
(39 trajectories)



(b) Cluster 4
(12 trajectories)



(c) Cluster 5
(19 trajectories)



(d) Cluster 6
(8 trajectories)

Figure : Refinement of cluster 1 into its three sub-clusters

- Experimental setting
 - 9 artificial datasets containing labeled clusters
 - Clusters can present interactions with each other
- Evaluation based on external criteria
 - Adjusted Rand Index [Hubert and Arabie, 1985]
 - Purity and entropy [Zhao and Karypis, 2002]

Table : Characteristics of the labeled datasets

Dataset	Clusters	Trajectories	Road network
1	9	158	Oldenburg
2	10	163	Oldenburg
3	11	141	Oldenburg
4	6	86	Oldenburg
5	6	91	Oldenburg
6	6	110	Oldenburg
7	12	205	San Joaquin
8	11	190	San Joaquin
9	12	203	San Joaquin

Table : Adjusted Rand Index

Dataset	Discovered clusters	Adjusted Rand Index	
		NNCluster Baseline	Modularity
1	9 (9)	0.902	1
2	10 (10)	0.881	1
3	11 (11)	0.764	0.873
4	6 (6)	1	1
5	6 (6)	1	1
6	6 (6)	1	1
7	14 (12)	0.618	0.961
8	12 (11)	0.921	0.971
9	10 (12)	0.752	0.889

Table : Purity and entropy

Dataset	Discovered clusters	Purity		Entropy	
		NNCluster Baseline	Modularity	NNCluster Baseline	Modularity
1	9 (9)	0.924	1	0.062	0
2	10 (10)	0.902	1	0.059	0
3	11 (11)	0.823	0.915	0.113	0.064
4	6 (6)	1	1	0	0
5	6 (6)	1	1	0	0
6	6 (6)	1	1	0	0
7	14 (12)	0.712	1	0.185	0
8	12 (11)	0.942	1	0.038	0
9	10 (12)	0.778	0.872	0.136	0.075

Extension to Road Segment Clustering

- Clustering road segments is equally important
- Motivations:
 - Characterize the roles they play in the road network
 - Predict how traffic congestion propagates



(a) Cluster 4
(12 trajectories)



(b) Cluster 5
(19 trajectories)



(c) Cluster 6
(8 trajectories)

Figure : Trajectory clusters are clearly “supported” by groups of road segments

Road Segment Clustering

[El Mahrsi and Rossi, 2012b, El Mahrsi and Rossi, 2013]

- We proceed by analogy to the trajectory case
 - Cosine similarity is used to measure segment resemblances
 - A weighted graph $\mathcal{G}_S(\mathcal{S}, \mathcal{E}_S, \mathcal{W}_S)$ depicts segment interactions
 - The same clustering algorithm is used to cluster the graph

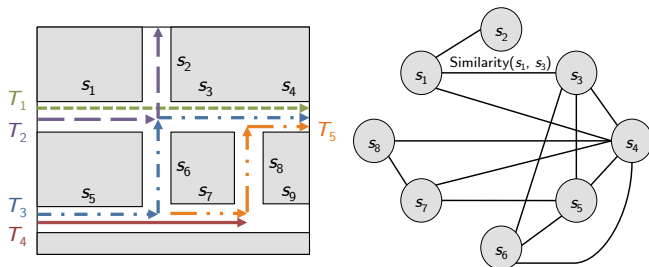


Figure : Example of a road segment similarity graph

How to Interpret Road Segment Clusters?

- We did discover clusters, but...



(a)



(b)



(c)



(d)



(e)



(f)

Figure : Examples of road segment clusters discovered through graph-based segment clustering

- Duality between trajectory clustering and segment clustering
- Road segment clusters are hard to interpret “on their own”
 - Due to lack of context
 - Easier to interpret in the light of trajectory clusters
 - Left to the initiative of the user
- Instead of considering trajectories and road segments separately, consider clustering both at the same time

Outline

- 1 Context and Motivations
- 2 Sampling Trajectory Data Streams
- 3 Graph-Based Clustering of Network-Constrained Trajectory Data
- 4 Co-Clustering Network-Constrained Trajectory Data**
- 5 Conclusions, Future Work and Open Issues

Co-Clustering Network-Constrained Trajectory Data

Joint work w/ Romain Guigourès and Marc Boullé (Orange Labs) [El Mahrsi et al., 2013]

- Objective: cluster trajectories and road segments simultaneously
- Equivalent to considering a bipartite graph $\mathcal{G}(\mathcal{T}, \mathcal{S}, \mathcal{E})$ representing interactions between trajectories and segments

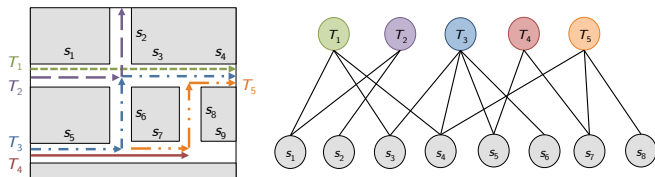
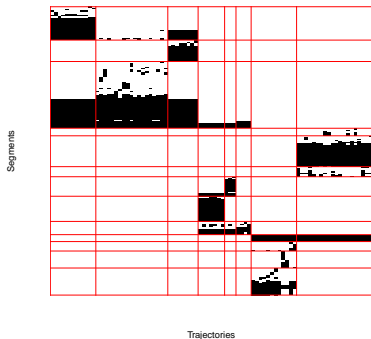


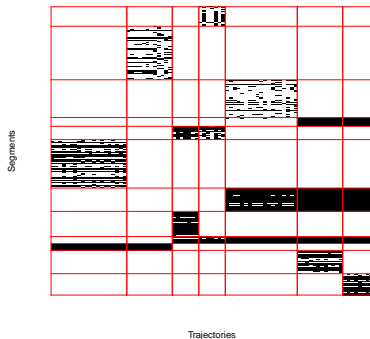
Figure : Bipartite graph of interactions between trajectories and road segments

- MODL co-clustering is applied to the adjacency matrix of the bipartite graph
 - Based on Bayesian model selection with a hierarchical prior
 - Rearrange rows and columns into homogeneously dense blocks
- Output: a set of co-clusters, each is the intersection of
 - A trajectory cluster
 - A road segment cluster

Back to the Case Study



(a) Modularity-based approach



(b) Co-clustering approach

Figure : Adjacency matrix of the bipartite graph, rearranged based on the clusters discovered by both approaches

Characterizing Traffic Using Trajectory/Segment Co-Clusters

- We use the discovered co-clusters' contribution to mutual information to guide the interpretation

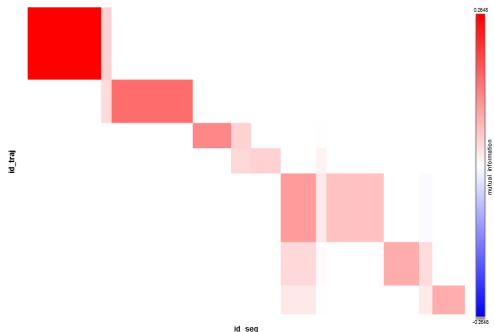


Figure : Contribution to mutual information of the co-clusters discovered in the case study dataset. Trajectory clusters (7 clusters) are depicted on the rows and road segment clusters (12 clusters) on the columns

Characterizing Traffic: Peripheral Road Segments



(a) 34 segments



(b) 40 segments



(c) 77 segments

Figure : Examples of “secondary” road segment clusters leading to peripheral areas of the road network and visited exclusively by single groups of trajectories

Characterizing Traffic: Hub Road Segments



(a) Hub segment cluster
(11 segments)



(b) Trajectory cluster
(20 trajectories)



(c) Trajectory cluster
(12 trajectories)

Figure : A hub road segment traveled by two different trajectory clusters with different departures and destinations

Outline

- 1 Context and Motivations
- 2 Sampling Trajectory Data Streams
- 3 Graph-Based Clustering of Network-Constrained Trajectory Data
- 4 Co-Clustering Network-Constrained Trajectory Data
- 5 Conclusions, Future Work and Open Issues**

- STSS, a fast on-the-fly algorithm for sampling trajectory streams with configurable approximation errors
[El Mahrsi et al., 2010]
- Graph-based approaches to clustering trajectories in road networks
[El Mahrsi and Rossi, 2012c, El Mahrsi and Rossi, 2012a, El Mahrsi and Rossi, 2012b, El Mahrsi and Rossi, 2013]
- An approach to simultaneous co-clustering of trajectories and road segments
[El Mahrsi et al., 2013]






- Noise sensitivity
- Presence of the road network
- Effect on querying

- Better evaluation of the approaches
 - On real datasets
 - With more realistic data generators
- Effect of varying the clustering algorithms
- Integration of time in the clustering process
- “Social-oriented” clustering of mobility data

List of Publications

- [1] M. K. El Mahrsi, C. Potier, G. Hébrail, and F. Rossi, "Spatiotemporal sampling for trajectory streams," in *SAC'10: Proceedings of the 2010 ACM Symposium on Applied Computing*, (New York, NY, USA), pp. 1627-1628, ACM, 2010. (Poster)
- [2] M. K. El Mahrsi and F. Rossi, "Modularity-Based Clustering for Network-Constrained Trajectories," in *Proceedings of the 20th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN 2012)*, (Bruges, Belgium), pp. 471-476, Apr. 2012.
- [3] M. K. El Mahrsi and F. Rossi, "Graph-Based Approaches to Clustering Network- Constrained Trajectory Data," in *Proceedings of the Workshop on New Frontiers in Mining Complex Patterns (NFMCP 2012)*, (Bristol, UK), pp. 184-195, Sept. 2012.
- [4] M. K. El Mahrsi and F. Rossi, "Clustering par optimisation de la modularité pour trajectoires d'objets mobiles," in *Actes des 8èmes journées francophones Mobilité et Ubiquité*, (Anglet, France), pp. 12-22, Cepaduès Éditions, Jun. 2012.
- [5] M. K. El Mahrsi, R. Guigourès, F. Rossi, and M. Boullé, "Classifications croisées de données de trajectoires contraintes par un réseau routier," in *Actes de 13ème Conférence Internationale Francophone sur l'Extraction et gestion des connaissances (EGC'2013)*, vol. RNTI-E-24, (Toulouse, France), pp. 341-352, Hermann-Éditions, Feb. 2013.
- [6] M. K. El Mahrsi and F. Rossi, "Graph-based approaches to clustering network-constrained trajectory data," in *New Frontiers in Mining Complex Patterns*, vol. 7765 of *Lecture Notes in Computer Science*, pp. 124-137, Springer Berlin Heidelberg, 2013.
- [7] M. K. El Mahrsi, R. Guigourès, F. Rossi, and M. Boullé, "Co-Clustering Network-Constrained Trajectory Data," Submitted to AKDM-5 (Advances in Knowledge Discovery and Management Vol. 5).

References I

-  Boullé, M. (2011). Data grid models for preparation and modeling in supervised learning. In Hands-On Pattern Recognition: Challenges in Machine Learning, vol. 1, pages 99–130. Microtome.
-  El Mahrsi, M. K., Guigourès, R., Rossi, F., and Boullé, M. (2013). Classifications croisées de données de trajectoires contraintes par un réseau routier. In Vrain, C., Péninou, A., and Sedes, F., editors, Actes de 13ème Conférence Internationale Francophone sur l'Extraction et gestion des connaissances (EGC'2013), volume RNTI-E-24, pages 341–352, Toulouse, France. Hermann-Éditions.
-  El Mahrsi, M. K., Potier, C., Hébrail, G., and Rossi, F. (2010). Spatiotemporal sampling for trajectory streams. In SAC '10: Proceedings of the 2010 ACM Symposium on Applied Computing, pages 1627–1628, New York, NY, USA. ACM.
-  El Mahrsi, M. K. and Rossi, F. (2012a). Clustering par optimisation de la modularité pour trajectoires d'objets mobiles. In UbiMob'12, pages 12–22.
-  El Mahrsi, M. K. and Rossi, F. (2012b). Graph-Based Approaches to Clustering Network-Constrained Trajectory Data. In Proceedings of the Workshop on New Frontiers in Mining Complex Patterns (NFMCP 2012), pages 184–195, Bristol, UK.

References II



El Mahrsi, M. K. and Rossi, F. (2012c). Modularity-Based Clustering for Network-Constrained Trajectories. In Proceedings of the 20th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN 2012), pages 471–476, Bruges, Belgium.



El Mahrsi, M. K. and Rossi, F. (2013). Graph-based approaches to clustering network-constrained trajectory data. In Appice, A., Ceci, M., Loglisci, C., Manco, G., Masciari, E., and Ras, Z., editors, New Frontiers in Mining Complex Patterns, volume 7765 of Lecture Notes in Computer Science, pages 124–137. Springer Berlin Heidelberg.



Guo, D., Liu, S., and Jin, H. (2010). A graph-based approach to vehicle trajectory analysis. J. Locat. Based Serv., 4:183–199.







Hubert, L. and Arabie, P. (1985). Comparing partitions. Journal of Classification, 2:193–218.



Meratnia, N. and de By, R. A. (2004). Spatiotemporal compression techniques for moving point objects. In Bertino, E., Christodoulakis, S., Plexousakis, D., Christophides, V., Koubarakis, M., Böhm, K., and Ferrari, E., editors, EDBT, volume 2992 of Lecture Notes in Computer Science, pages 765–782. Springer.



Newman, M. E. J. (2006). Modularity and community structure in networks. Proceedings of the National Academy of Sciences, 103(23):8577–8582.

-  Noack, A. and Rotta, R. (2009). Multi-level algorithms for modularity clustering. In Proceedings of the 8th International Symposium on Experimental Algorithms, SEA '09, pages 257–268, Berlin, Heidelberg. Springer-Verlag.
-  Potamias, M., Patrourmpas, K., and Sellis, T. (2006). Sampling trajectory streams with spatiotemporal criteria. In Proceedings of the 18th International Conference on Scientific and Statistical Database Management, SSDBM '06, pages 275–284, Washington, DC, USA. IEEE Computer Society.
-  Roh, G.-P. and Hwang, S.-w. (2010). Nncluster: An efficient clustering algorithm for road network trajectories. In Database Systems for Advanced Applications, volume 5982 of Lecture Notes in Computer Science, pages 47–61. Springer Berlin - Heidelberg.
-  Zhao, Y. and Karypis, G. (2002). Criterion functions for document clustering: Experiments and analysis. Technical report.

STSS Vs. STTrace [Potamias et al., 2006]

- Athens trucks dataset
 - 276 trajectories
 - 112203 data points (1 position/30 sec)
- STTrace: on-the-fly, no error guarantees (but storage space guarantee)
- Comparison for the same percentage of retained data
- Evaluation criteria
 - Average approximation error

$$\text{Average Approximation Error} = \frac{1}{\sum_{T \in \mathcal{T}} |T|} \times \sum_{T \in \mathcal{T}} \sum_{P_i \in T} \text{distance}(P_i, P'_i)$$

- Maximum approximation error

$$\text{Maximum Approximation Error} = \max_{T \in \mathcal{T}} (\max_{P_i \in T} (\text{distance}(P_i, P'_i)))$$

STSS Vs. STTrace: Average Approximation Error

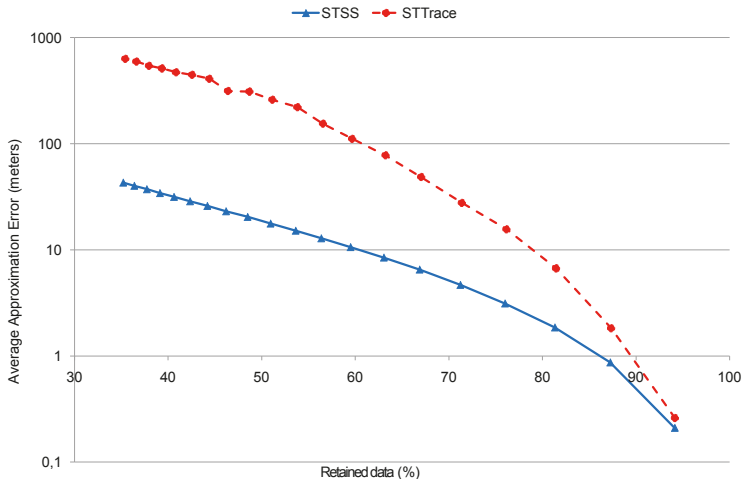


Figure : Average Approximation Errors resulting from STSS and STTrace sampling

STSS Vs. STTrace: Maximum Approximation Error

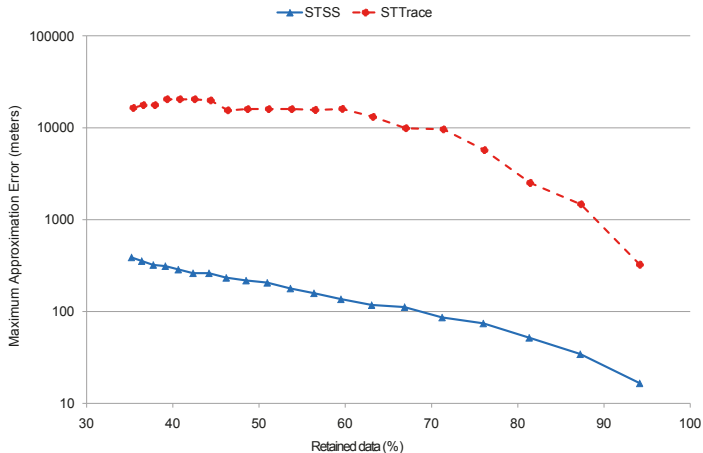


Figure : Maximum Approximation Errors resulting from STSS and STTrace sampling

Why Modularity-Based Community Detection?

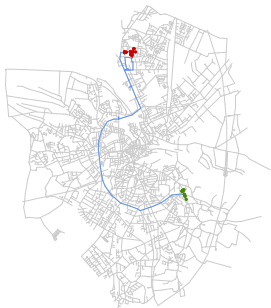
- Efficiency and effectiveness observed in practice
- Non-parametric
- Robustness to the presence of high degrees
- The implementation we used produces a hierarchy of nested clusters
 - Recursive descent based on the statistical significance of the partitions

How Do We Generate Our Labeled Datasets?

- When generating a cluster
 - A set of neighbor vertices is selected as the starting area
 - A set of neighbor vertices is selected as the destination area
 - For each trajectory, a vertex is chosen randomly in each set and the trajectory is generated as the shortest path between them
- Clusters are generated based on patterns we considered as relevant

Cluster Patterns: Inverted Clusters

- The starting area of one cluster is the destination area of the other



(a)

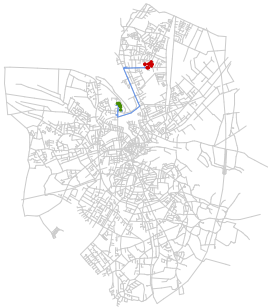


(b)

Figure : Example of inverted clusters

Cluster Patterns: Converging Clusters

- The clusters depart from different areas and arrive to the same destination area



(a)

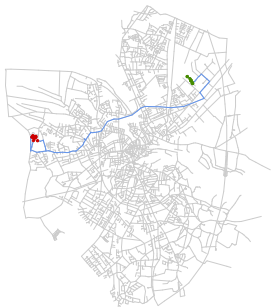


(b)

Figure : Example of converging clusters

Cluster Patterns: Diverging Clusters

- The clusters depart from the same area and arrive to different destinations



(a)



(b)

Figure : Example of diverging clusters

Modularity Vs. Spectral Clustering (Trajectory Case)

Table : Adjusted Rand Index

Dataset	Discovered clusters	Adjusted Rand Index	
		Spectral	Modularity
1	9 (9)	1	1
2	10 (10)	1	1
3	11 (11)	0.802	0.873
4	6 (6)	1	1
5	6 (6)	0.974	1
6	6 (6)	1	1
7	14 (12)	0.961	0.961
8	12 (11)	0.942	0.971
9	10 (12)	0.889	0.889

Table : Entropy and Purity

Dataset	Discovered clusters	Purity		Entropy	
		Spectral	Modularity	Spectral	Modularity
1	9 (9)	1	1	0	0
2	10 (10)	1	1	0	0
3	11 (11)	0.837	0.915	0.106	0.064
4	6 (6)	1	1	0	0
5	6 (6)	0.989	1	0.0233	0
6	6 (6)	1	1	0	0
7	14 (12)	1	1	0	0
8	12 (11)	0.963	1	0.021	0
9	10 (12)	0.872	0.872	0.075	0.075

Internal Quality Criteria

- Inspired by Intra-Cluster Inertia
- Sum of average trajectory intra-cluster overlaps

$$Q(\mathcal{C}_T) = \sum_{C \in \mathcal{C}_T} \frac{1}{|C|} \sum_{T_i, T_j \in C} \frac{\sum_{s \in T_i, s \in T_j} \text{length}(s)}{\sum_{s \in T_i} \text{length}(s)}$$

- Sum of average road segment intra-cluster overlaps

$$Q(\mathcal{C}_S) = \sum_{C \in \mathcal{C}_S} \frac{1}{|C|} \sum_{s_i, s_j \in C} \frac{|\{T \in \mathcal{T} : s_i \in T \wedge s_j \in T\}|}{|\{T \in \mathcal{T} : s_i \in T \vee s_j \in T\}|}$$

Similarity Between Road Segments

- Road segments are considered as bags-of-trajectories
- Weights are assigned to trajectories based on the number of segments they visit

$$\omega_{T,s} = \frac{n_{s,T}}{\sum_{T' \in \mathcal{T}} n_{s,T'}} \times \log \frac{|\mathcal{S}|}{|s' \in \mathcal{S} : s' \in T|}$$

- Segment resemblance is measured through cosine similarity

$$\text{Similarity}(s_i, s_j) = \frac{\sum_{T \in \mathcal{T}} \omega_{T,s_i} \times \omega_{T,s_j}}{\sqrt{\sum_{T \in \mathcal{T}} \omega_{T,s_i}^2} \times \sqrt{\sum_{T \in \mathcal{T}} \omega_{T,s_j}^2}}$$

Modularity Vs. Spectral Clustering (Segment Case)

- Comparison on 5 artificial datasets (composed of 100 trajectories each)
- Based on the sum of average road segment intra-cluster overlaps

$$Q(\mathcal{C}_S) = \sum_{C \in \mathcal{C}_S} \frac{1}{|C|} \sum_{s_i, s_j \in C} \frac{|\{T \in \mathcal{T} : s_i \in T \wedge s_j \in T\}|}{|\{T \in \mathcal{T} : s_i \in T \vee s_j \in T\}|}$$

Table : Characteristics of the five synthetic datasets

Dataset	Number of segments	Number of edges in the similarity graph
1	2562	79811
2	2394	100270
3	2587	110095
4	2477	87023
5	2348	80659

Modularity Vs. Spectral Clustering (Segment Case)

Table : Sum of average segment intra-cluster overlaps

Dataset	Number of discovered clusters	Intra-cluster overlaps	
		Spectral	Modularity
1	23	685.82	657.20
2	21	556.22	524.46
3	20	623.21	561.09
4	22	647.56	594.76
5	26	684.81	666.24