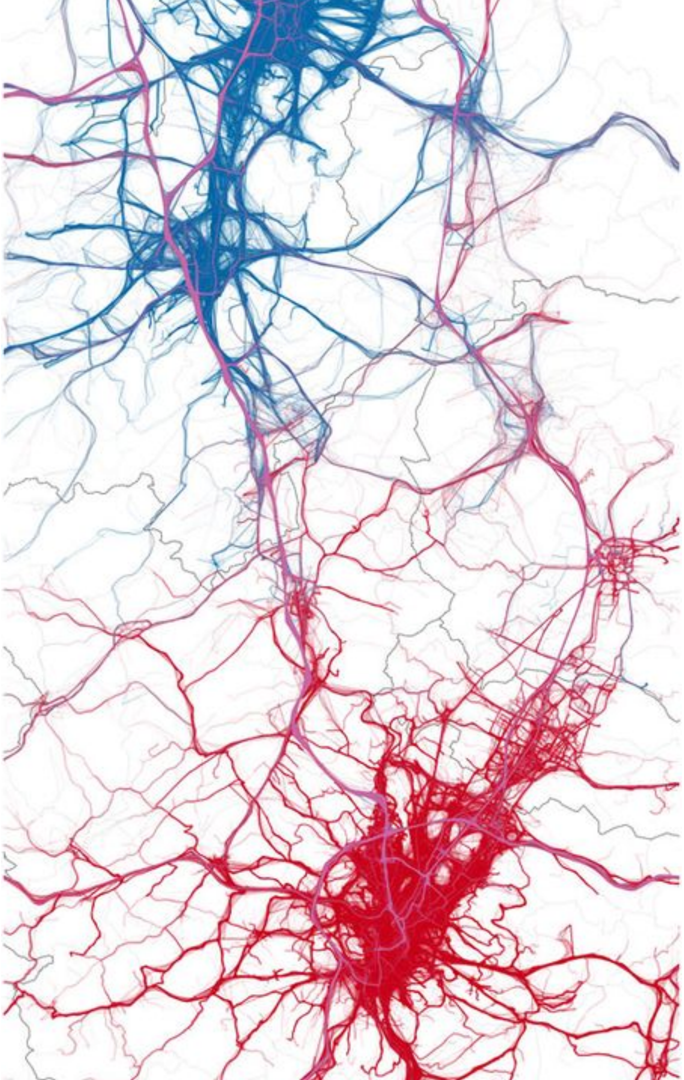


Mobility Patterns

Consiglio Nazionale delle Ricerche

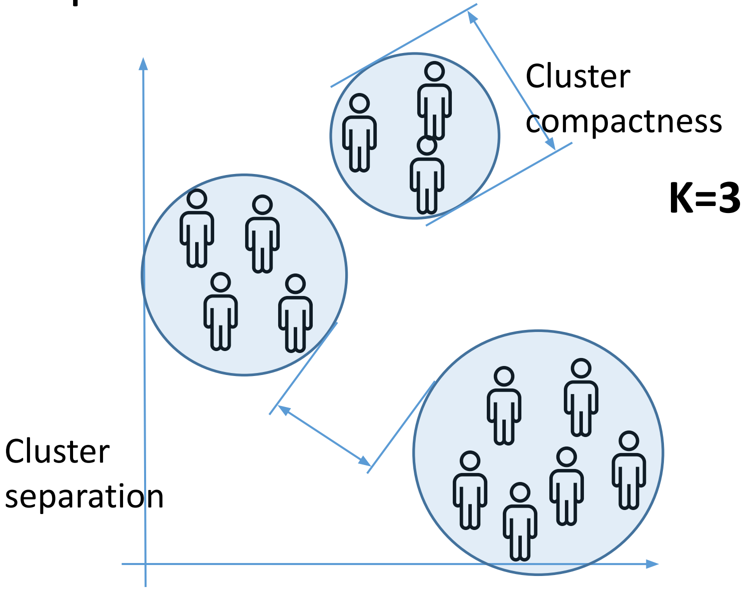
Content of this lesson

- Global patterns: Clustering
  - Trajectory distances
  - Trajectory clustering
- Local patterns
  - Flocks, Convoys & Swarms
  - Moving clusters
  - T-Patterns

Global Patterns

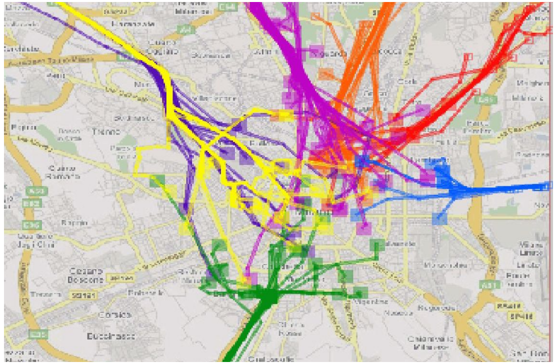
Clustering
(sample K-means family)

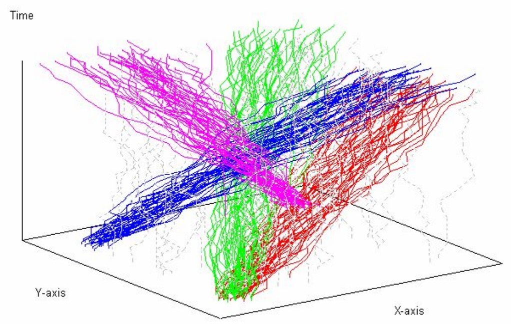
- Find k subgroups that form compact and well-separated clusters



Trajectory clustering

- Trajectories are grouped based on similarity



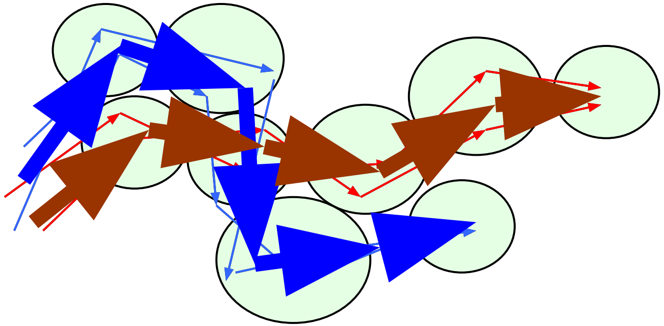


Nanni, Pedreschi. **Time-focused clustering of trajectories of moving objects**. J. of Intelligent Information Systems, 2006

Rinzivillo, Pedreschi, Nanni, Giannotti, Andrienko, Andrienko. **Visually-driven analysis of movement data by progressive clustering**. J. of Information Visualization, 2008

Trajectory Clustering

- Questions:
  - Which distance between trajectories?
  - Which kind of clustering?
  - What is a cluster ‘mean’ in our case?
    - A representative trajectory?



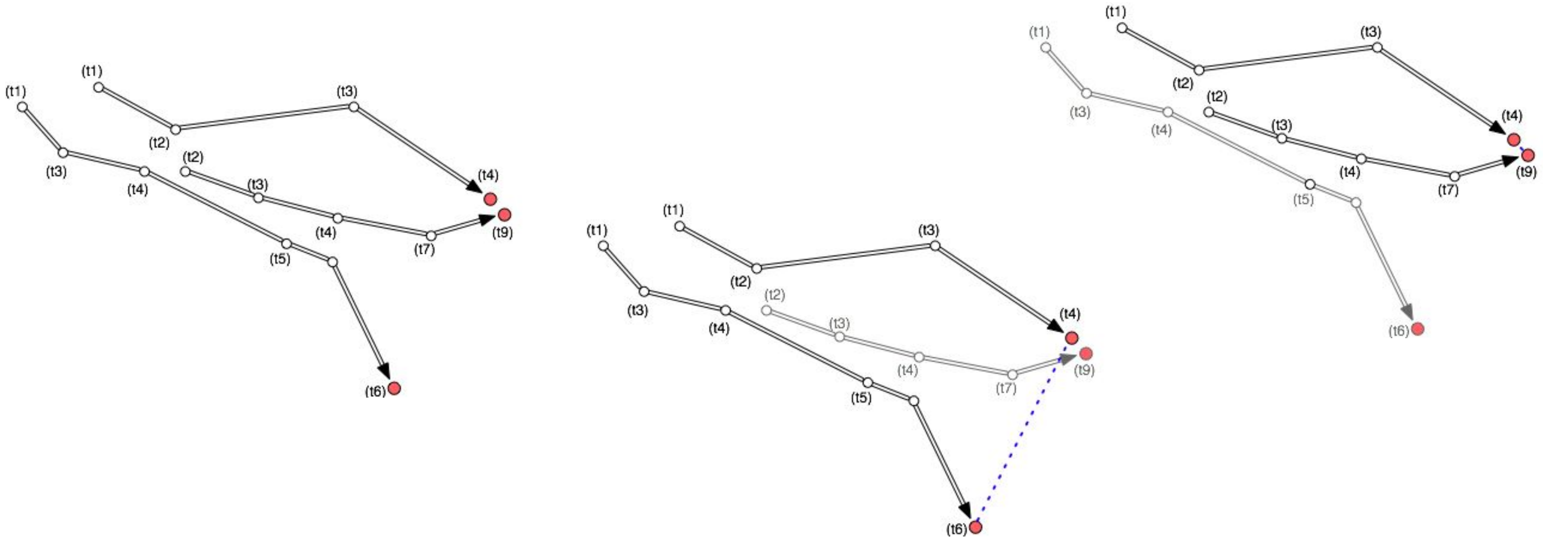
Trajectory Distances

Families of Trajectory Distances

- Trajectory as **set** of points
  - Single-point approaches
  - Hausdorff distance
- Trajectory as **sequence** of points
  - Fréchet distance
  - Time series distances: Euclidean, DTW & LCSS
- Trajectory as **time-stamped sequence** of points
  - Average Euclidean distance

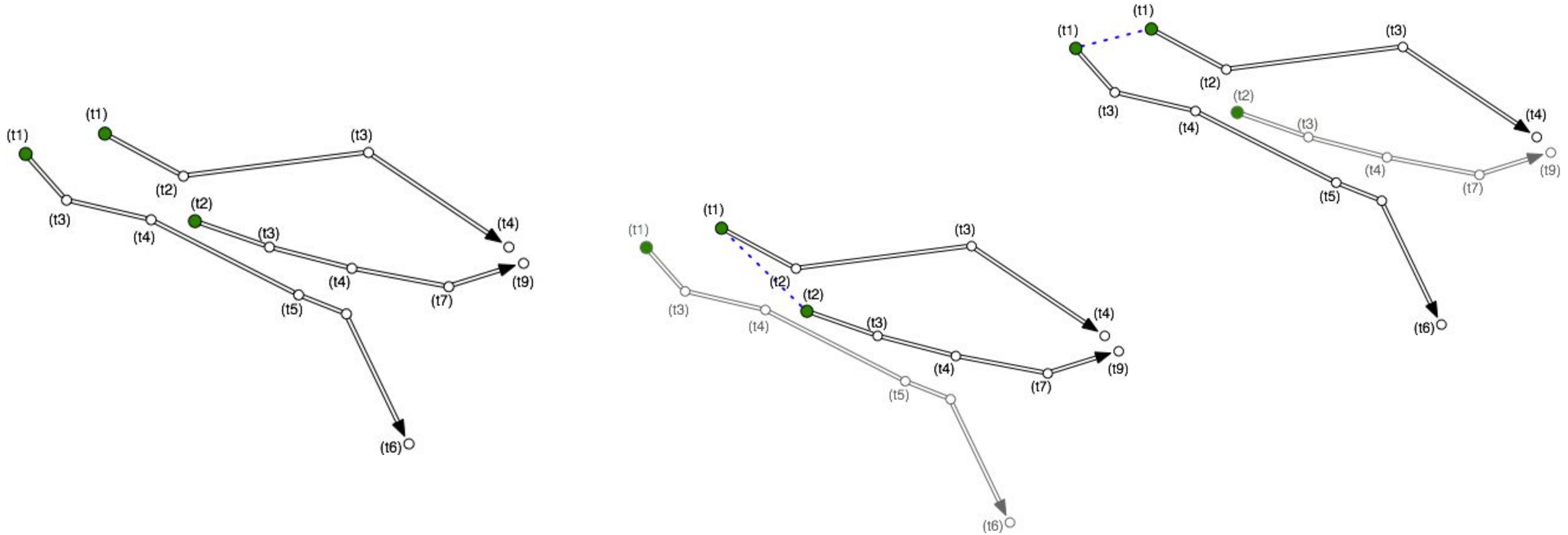
Reduce Trajectories to single points
Common Destination

- Select last point *Plast* for each trajectory
- D(T,T’) = Euclidean(Plast, P’last)



Reduce Trajectories to single points
Common Origin

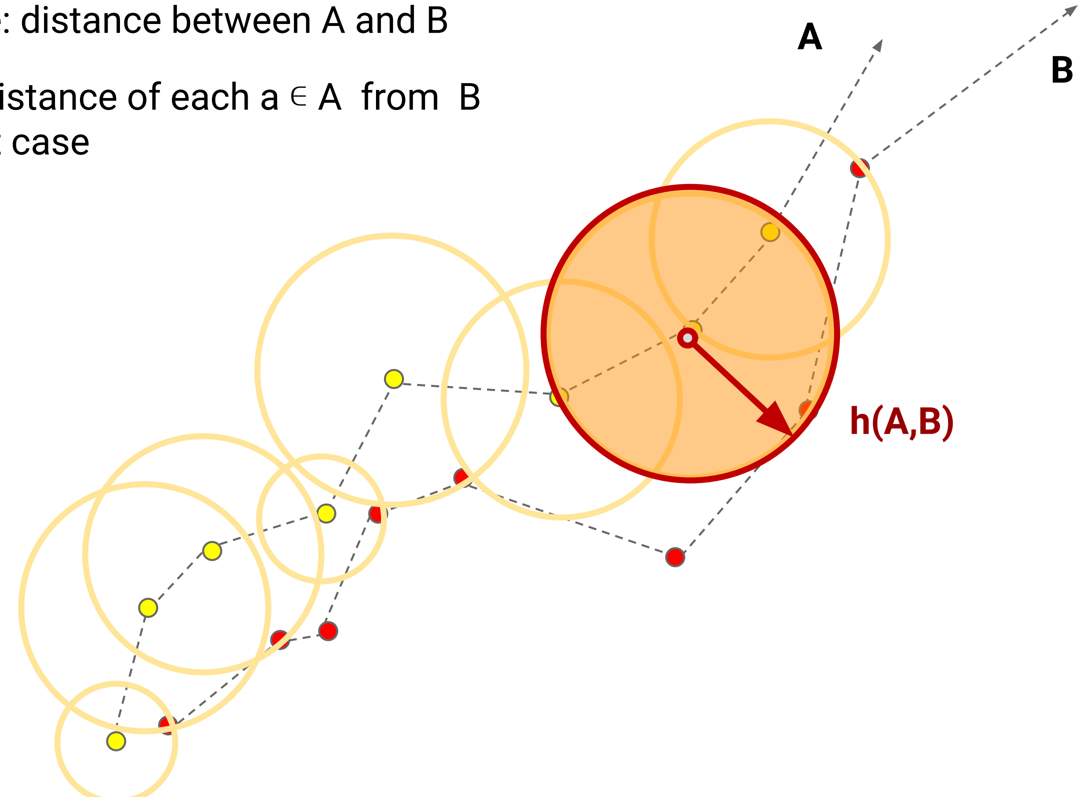
- Select first point *Pfirst* for each trajectory
- D(T,T') = Euclidean(Pfirst, P'first)



Trajectory as set of points
Hausdorff distance

Start from an example: distance between A and B

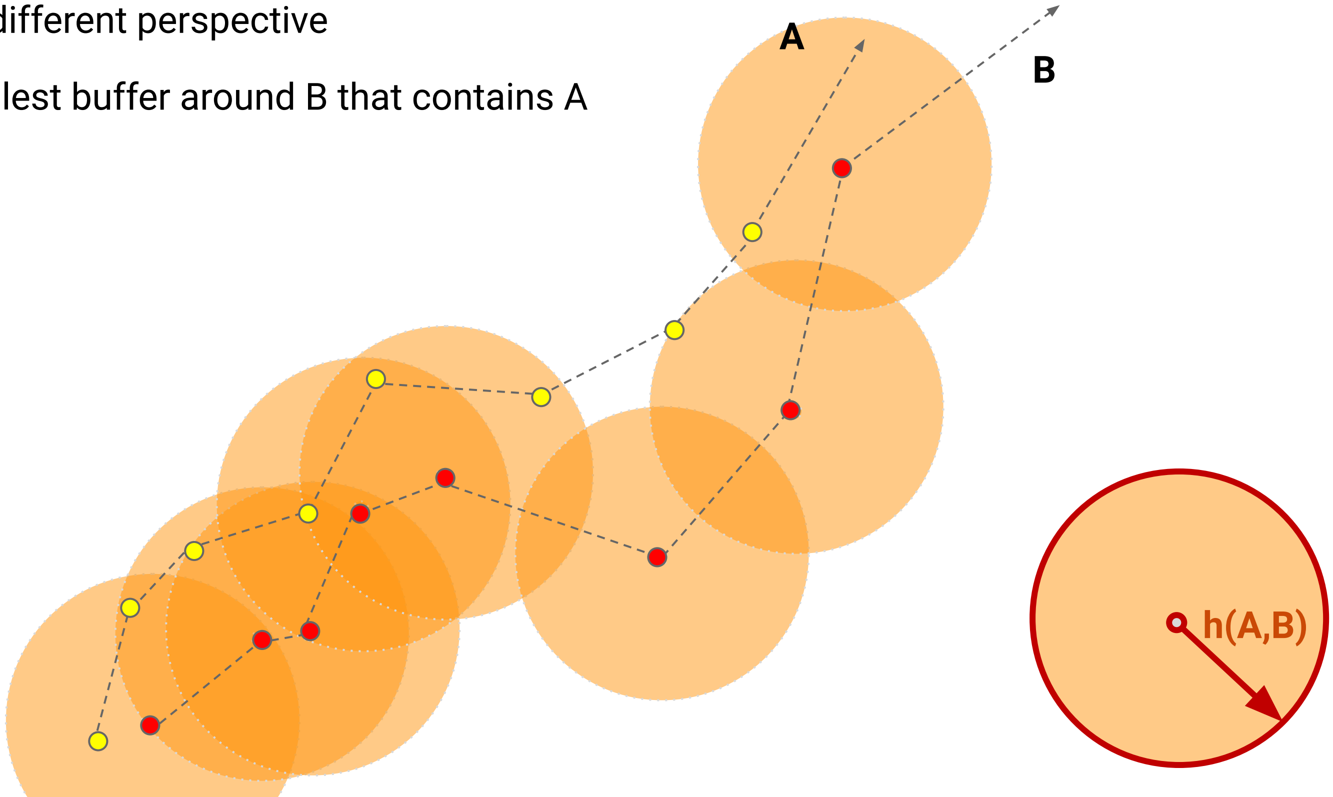
- Find minimum distance of each a ∈ A from B
- Return the worst case



Trajectory as set of points
Hausdorff distance

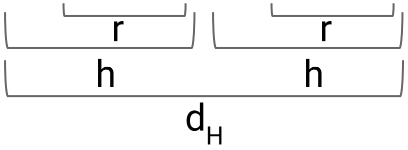
Same concept, from a different perspective

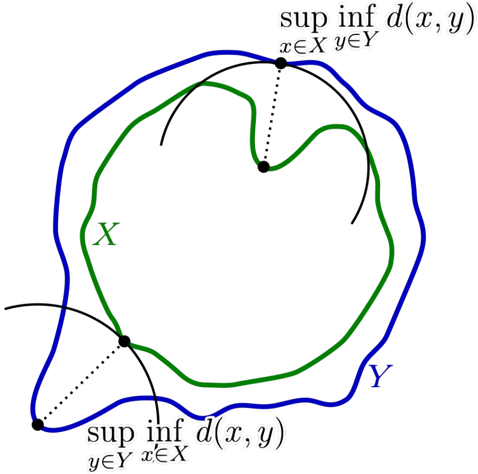
- h(A,B) is the smallest buffer around B that contains A



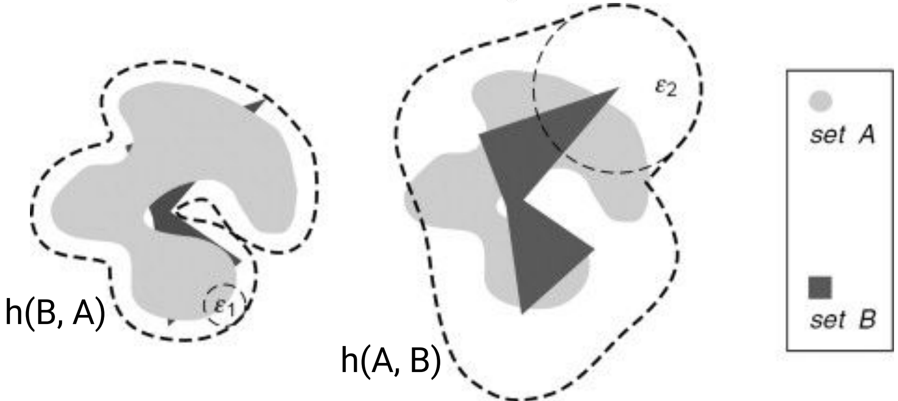
Trajectory as set of points
Hausdorff distance

- Intuition: two sets are close if every point of either set is close to some point of the other set
- Formally, given sets A and B: $d_{\mathrm{H}}(X,Y) = \max\{\sup_{x\in X} \inf_{y\in Y} d(x,y), \sup_{y\in Y} \inf_{x\in X} d(x,y)\}$
  - r (x , B) = inf {d(x , b) : b ∈ B}
  - h(A, B) = sup{r (a, B) : a ∈ A}
  - d_H(A, B) = max { h(A, B), h(B, A) }





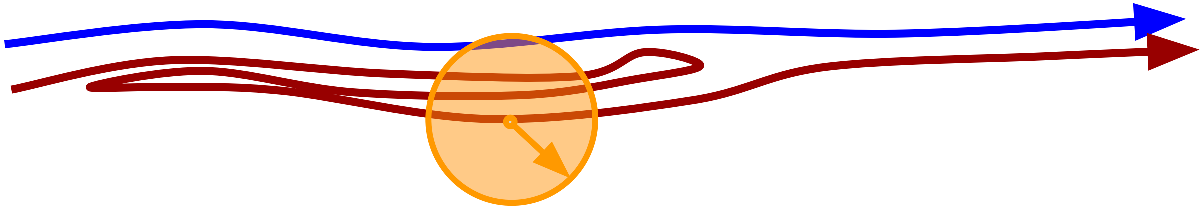
- Equivalently:
  - h(A, B) = minimum buffer radius around B that fully contains A
  - d_H(A, B) = symmetric version of h()



Trajectory as sequence of points

From Hausdorff to Fréchet distance

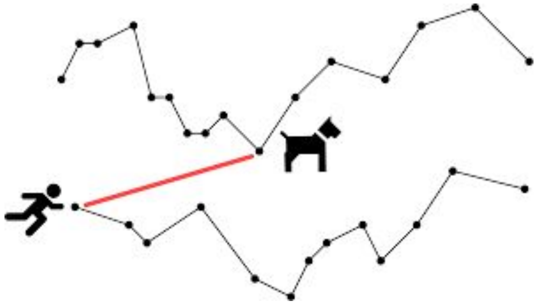
- Applied to trajectories, sometimes Hausdorff distance yields counter-intuitive results
- How far are these?



- Reasonable in a set-oriented view
- Wrong in terms of moving objects

Trajectory as sequence of points
Fréchet distance

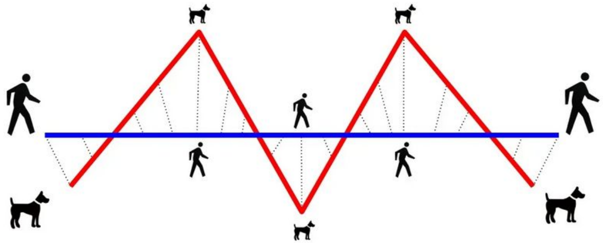
- Intuition: equivalent of Dynamic Time Warping on continuous curves
- Formally:



$$F(A,B) = \inf_{\alpha,\beta} \max_{t\in[0,1]} \left\{ d\Big(A(\alpha(t)),\, B(\beta(t))\Big) \right\}$$

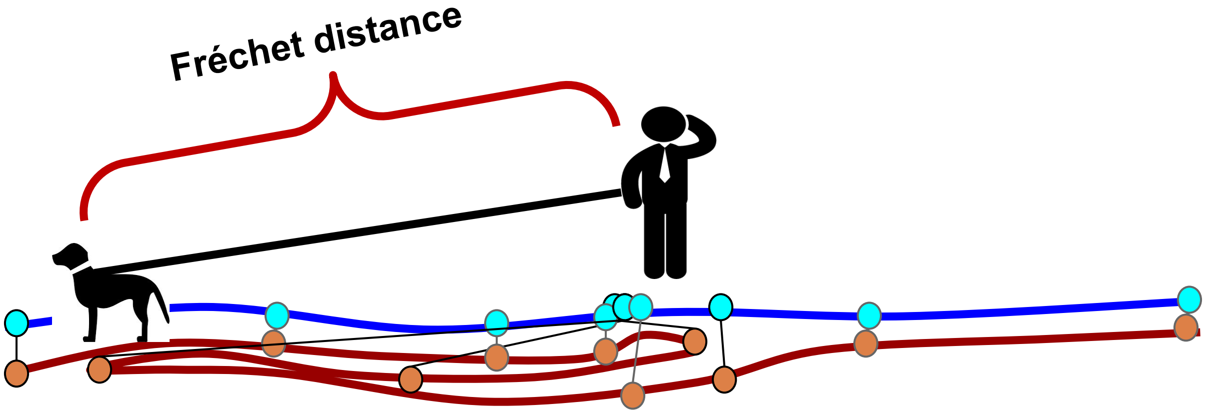
  α and β are non-decreasing mappings from [0,1] to the points along A and B in forward order

- Also described as “minimum leash length”:
  - What is the minimum length of a leash needed to stroll around the dog, given the owner’s and the dog’s trajectories?



Trajectory as sequence of points
Fréchet distance

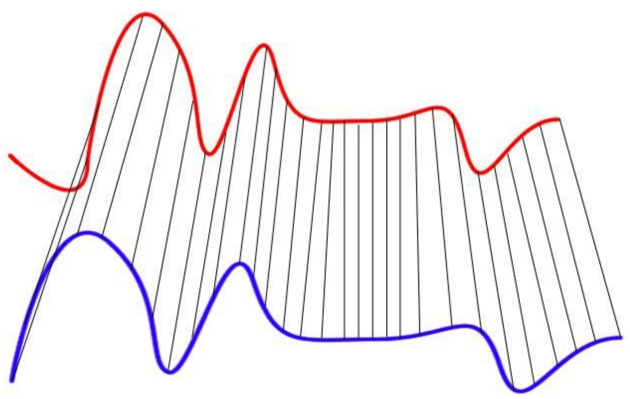
- Back to our example



Trajectory as sequence of points
Time series distances

- Just replace “difference of two values” with “spatial distance of two points”

- Examples:
    - Dynamic Time Warping
        - Very similar to Fréchet!
    - Edit Distance with Real values
        - Similar to DTW, but can remove points



Dynamic Time Warping Matching

- IMPORTANT: most methods in this class assume constant sampling rates

Trajectory as time-stamped sequence of points
Average Euclidean distance

- The trajectory is seen as a continuous spatio-temporal curve
- Positions between input points (the GPS fixes) linearly interpolated

$$D(\tau_1, \tau_2)|_T = \frac{\int_T d(\tau_1(t), \tau_2(t))dt}{|T|}$$

distance between moving objects τ_1 and τ_2 at time t

- "Synchronized" behaviour distance
  - Similar objects = almost always in the same place at the same time
- Computed on the whole trajectory

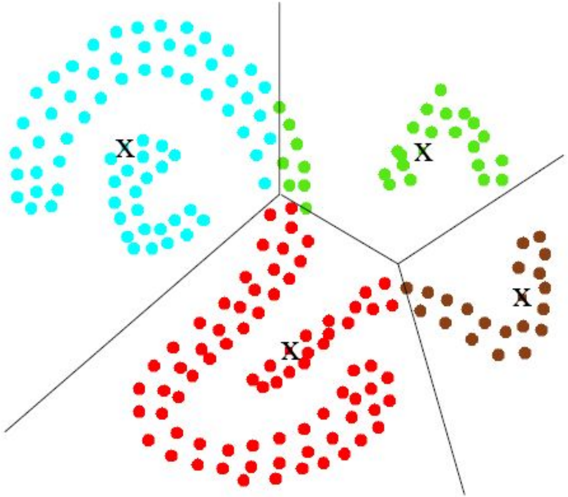
Clustering Algorithms

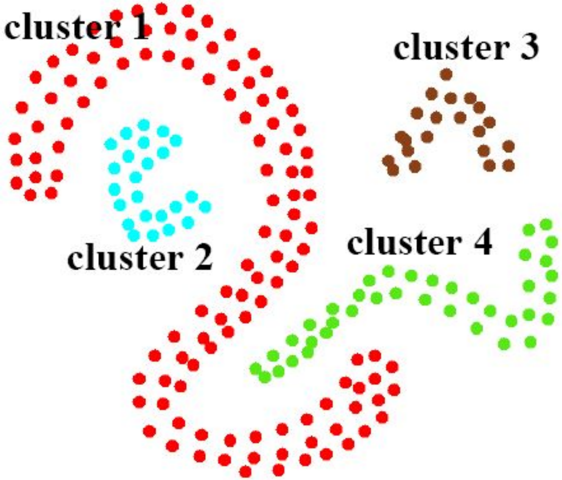
Which kind of clustering method?

- In principle, any distance-based algorithm
- General requirements:
  - Non-spherical clusters should be allowed
    - E.g.: A traffic jam along a road = “snake-shaped” cluster
  - Tolerance to noise
  - Low computational cost
  - Applicability to complex, possibly non-vectorial data
- A suitable candidate: Density-based clustering
  - OPTICS (Ankerst et al., 1999)
  - Evolution of standard DBSCAN

Density Based Clustering
A refresher

K-means	*Density-based*

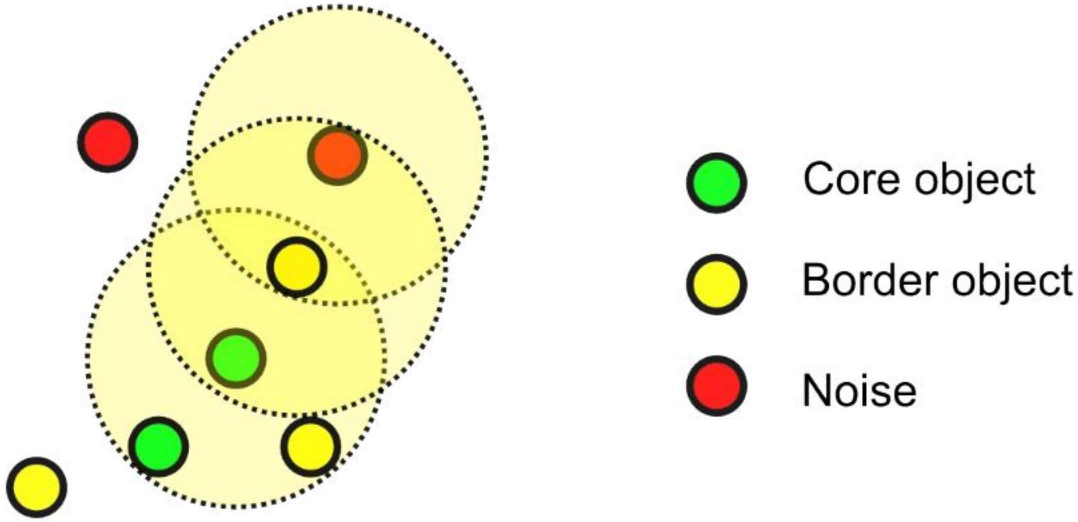




Density Based Clustering

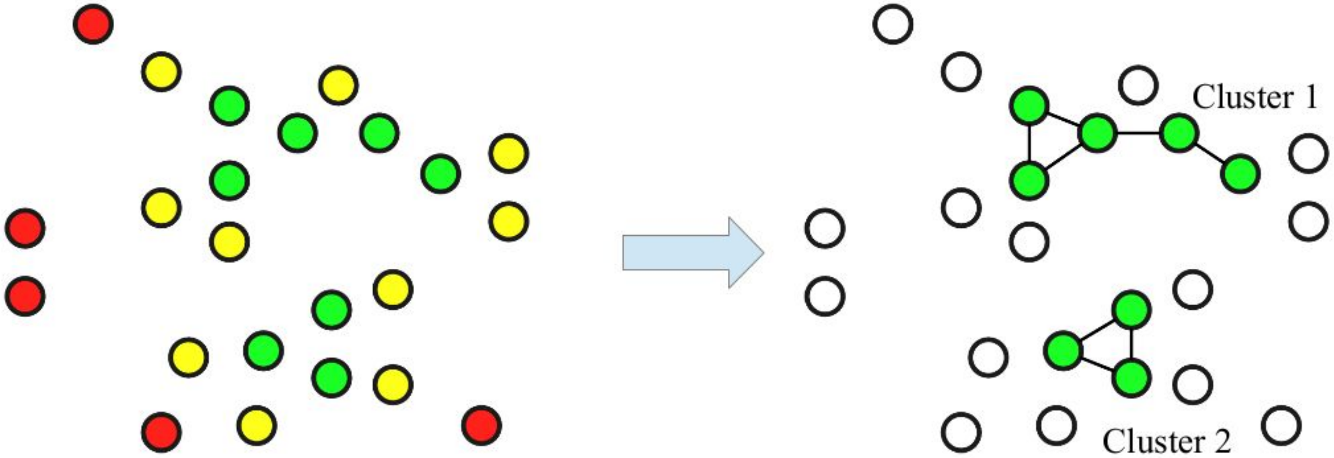
Step 1: label points as core (dense), border and noise

- Based on thresholds R (radius of neighborhood) and min_pts (min number of neighbors)



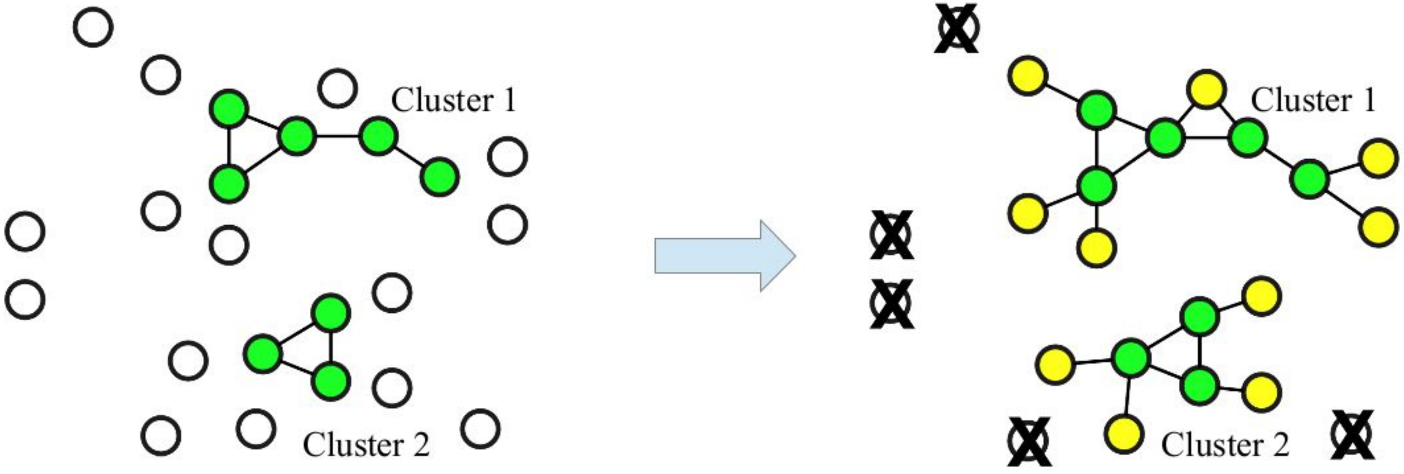
Density Based Clustering

Step 2: connect core objects that are neighbors, and put them in the same cluster



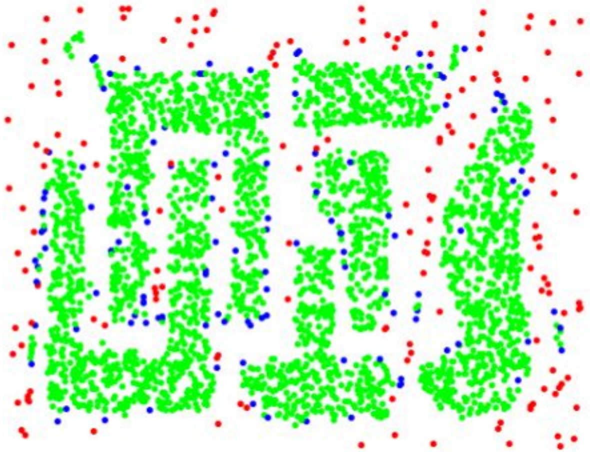
Density Based Clustering

Step 3: associate border objects to (one of) their core(s), and remove noise



Density Based Clustering

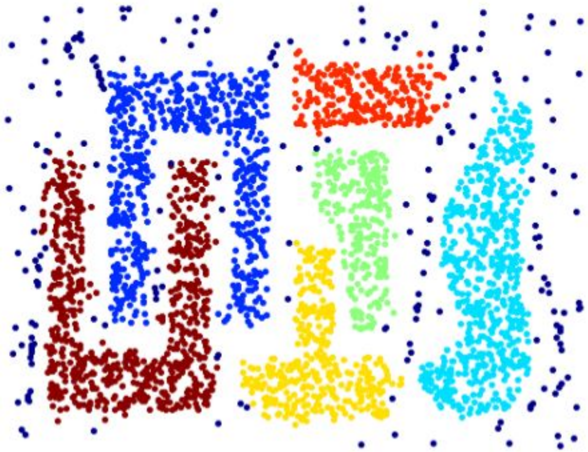
Original Points



Point types: core, border and noise

Density Based Clustering

Original Points

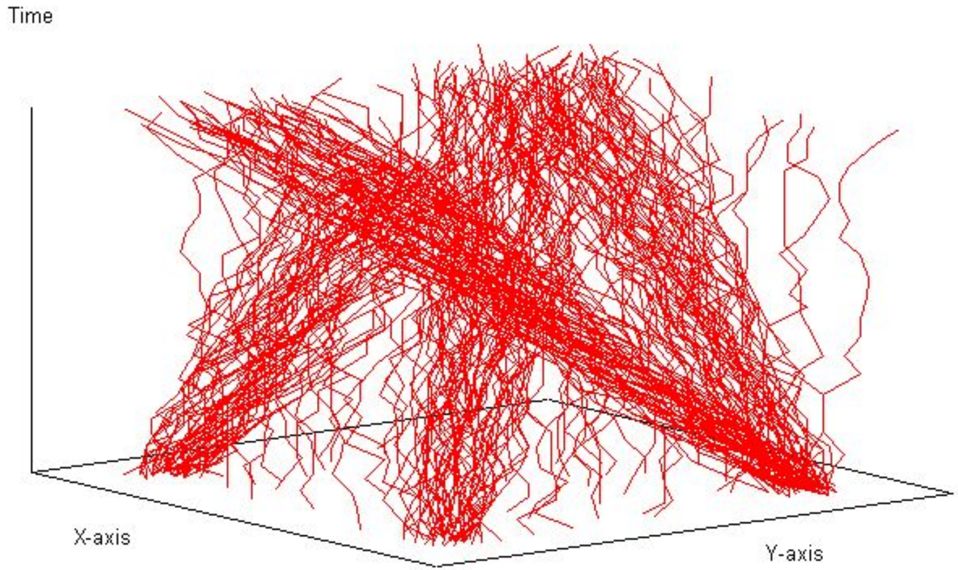


Clusters

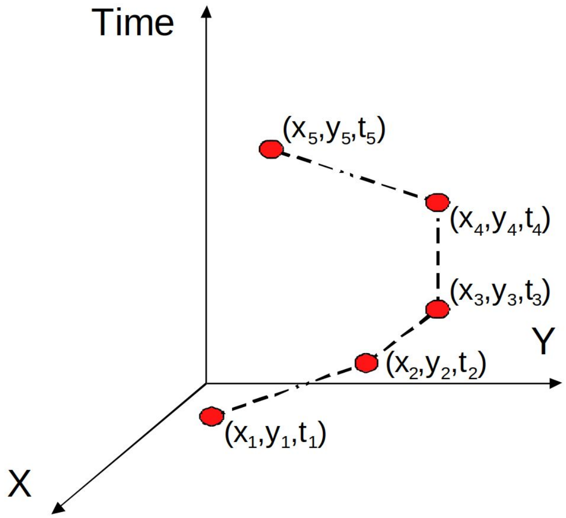
- Resistant to Noise
- Can handle clusters of different shapes and sizes

A sample dataset

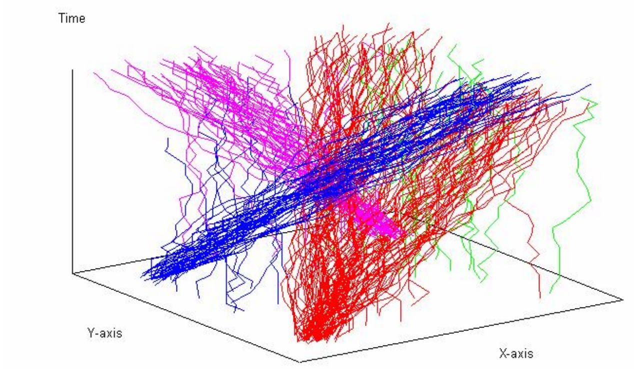
- A set of trajectories forming 4 clusters + noise (synthetic)

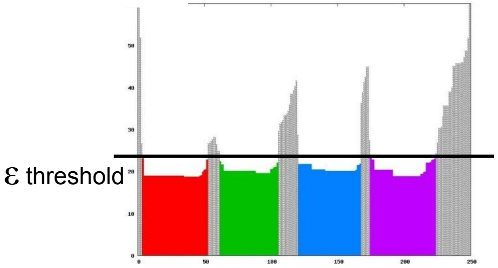


T-OPTICS vs. K-means

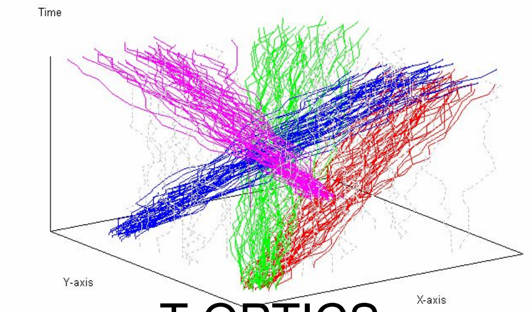


K-means





Reachability plot
(= objects reordering for distance distribution)

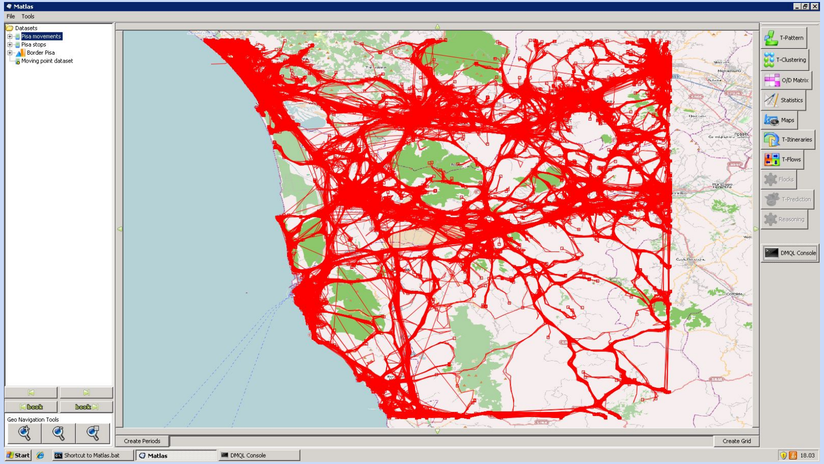


T-OPTICS

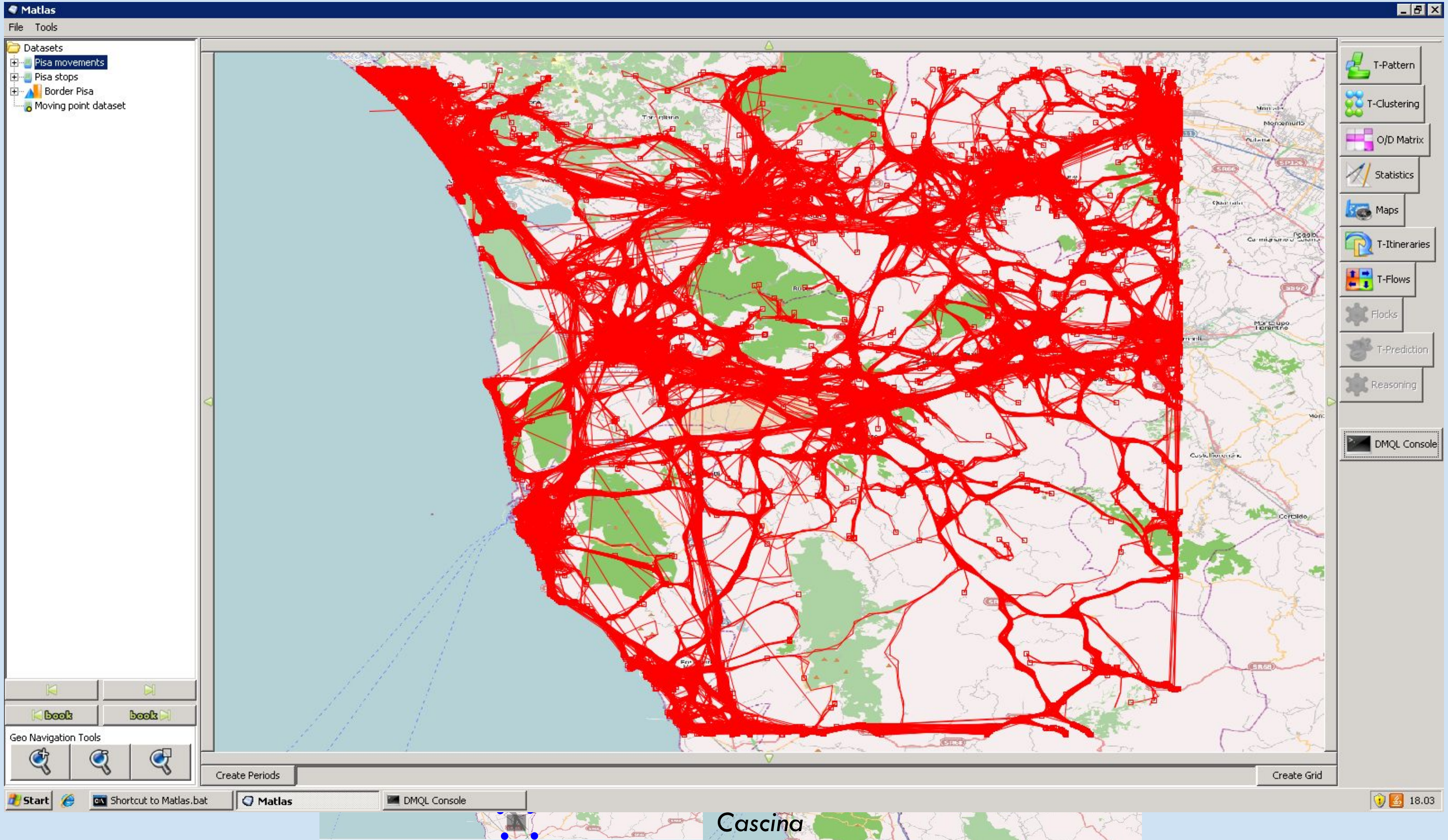
INTERVALLO

What's the source of traffic in Pisa?

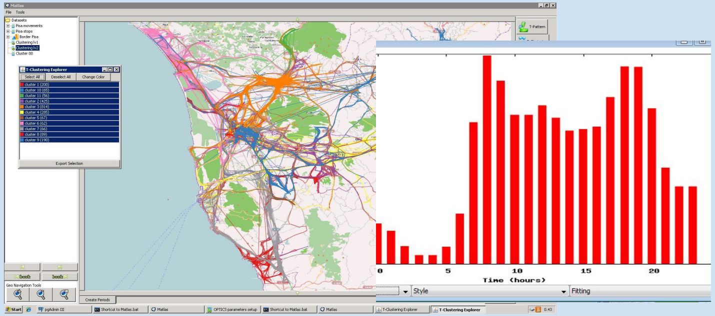
Trajectory clustering at work

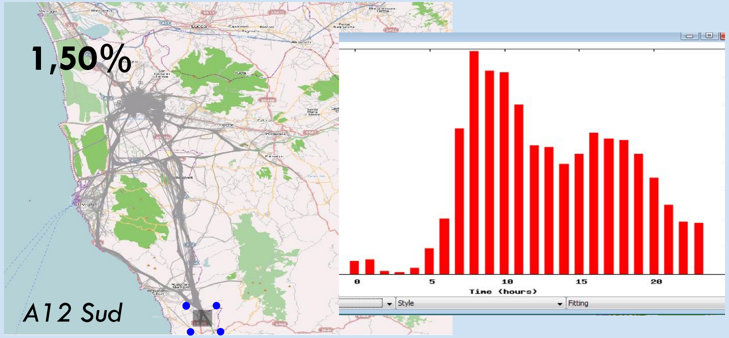


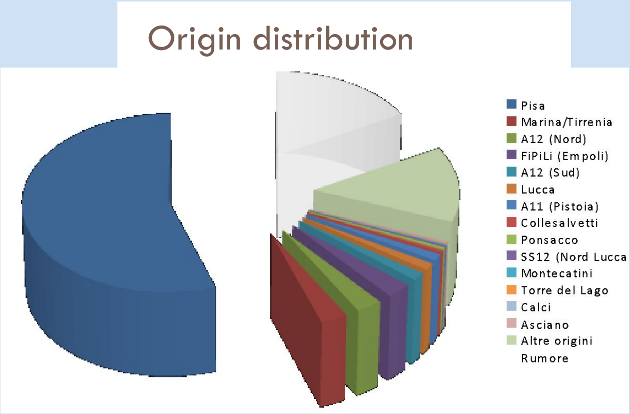
Access patterns using T-clustering

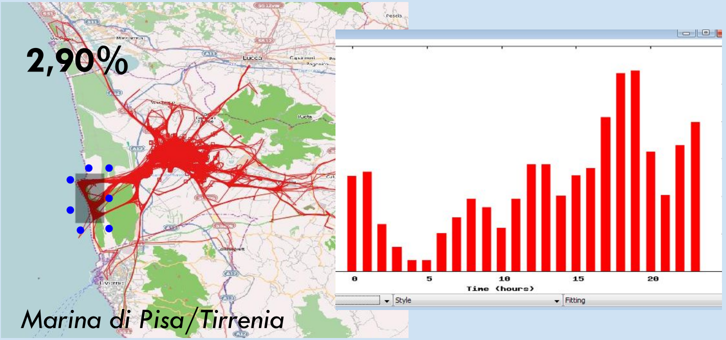


Characterizing the access patterns: origin & time





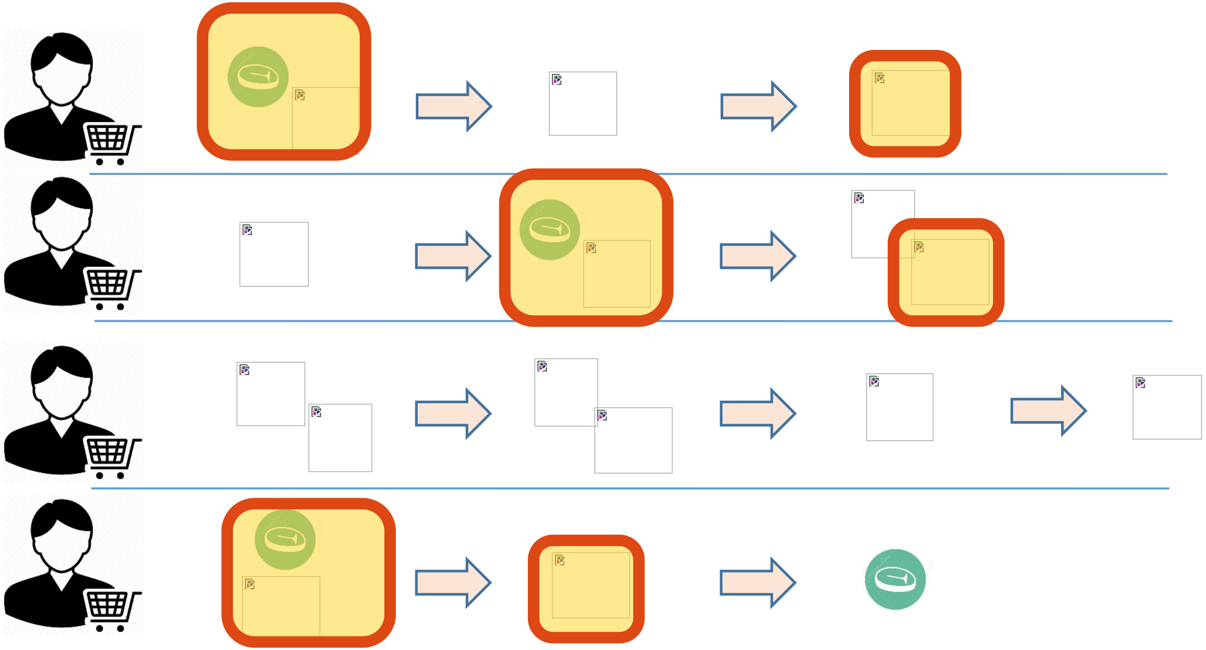




Local Trajectory Patterns

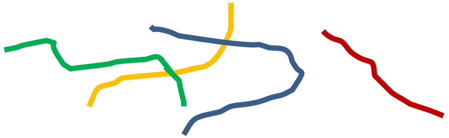
Frequent patterns in sequences

- Frequent sequences (a.k.a. Sequential patterns)
- Input: sequences of events (or of groups)



From trajectories to sequential patterns: the easy way

- Map each trajectory to a sequence of areas
  - Predefined or driven by data



+

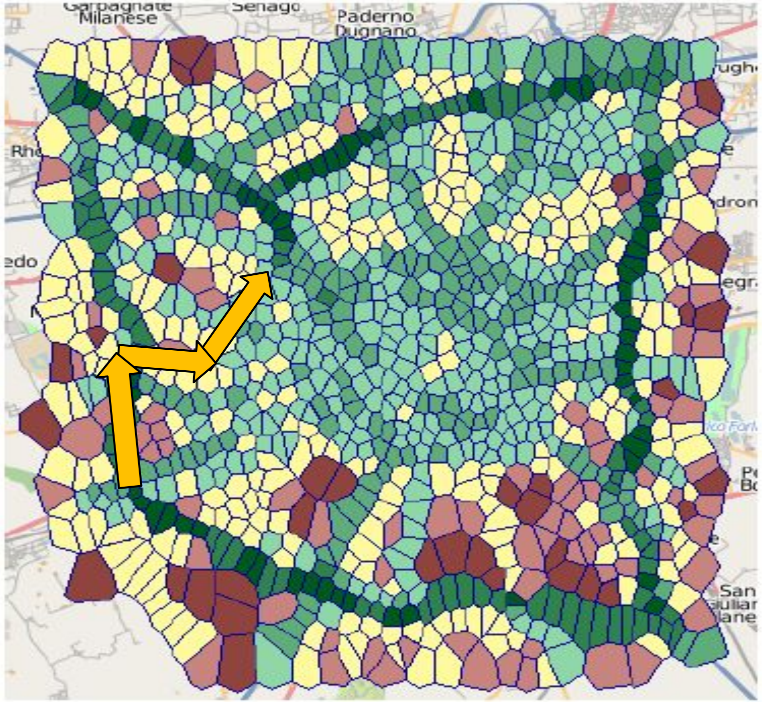
O	I	F	P	Q
A	B	E	H	M
N	D	C	G	L

=

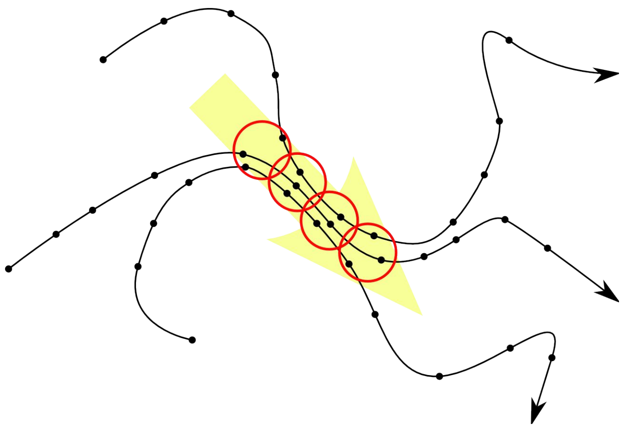
A → B → C
D → B → E → F
C → G → H → E → I
L → M → H

From trajectories to sequential patterns: the easy way

- A “Trajectory frequent pattern” can be defined as sequential pattern over traversed areas

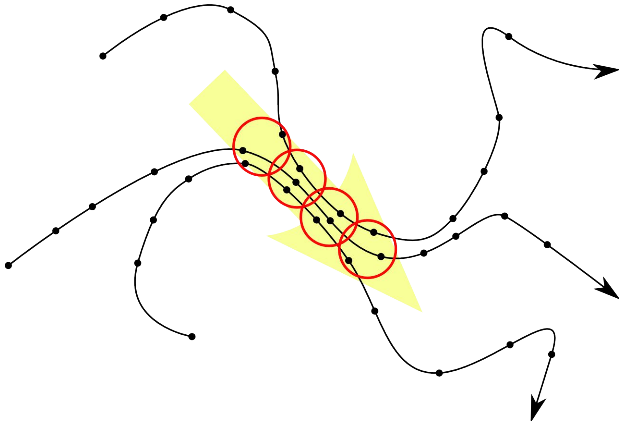


Moving Trajectory Flocks



- Group of objects that move together (close to each other) for a time interval

Moving Trajectory Flocks

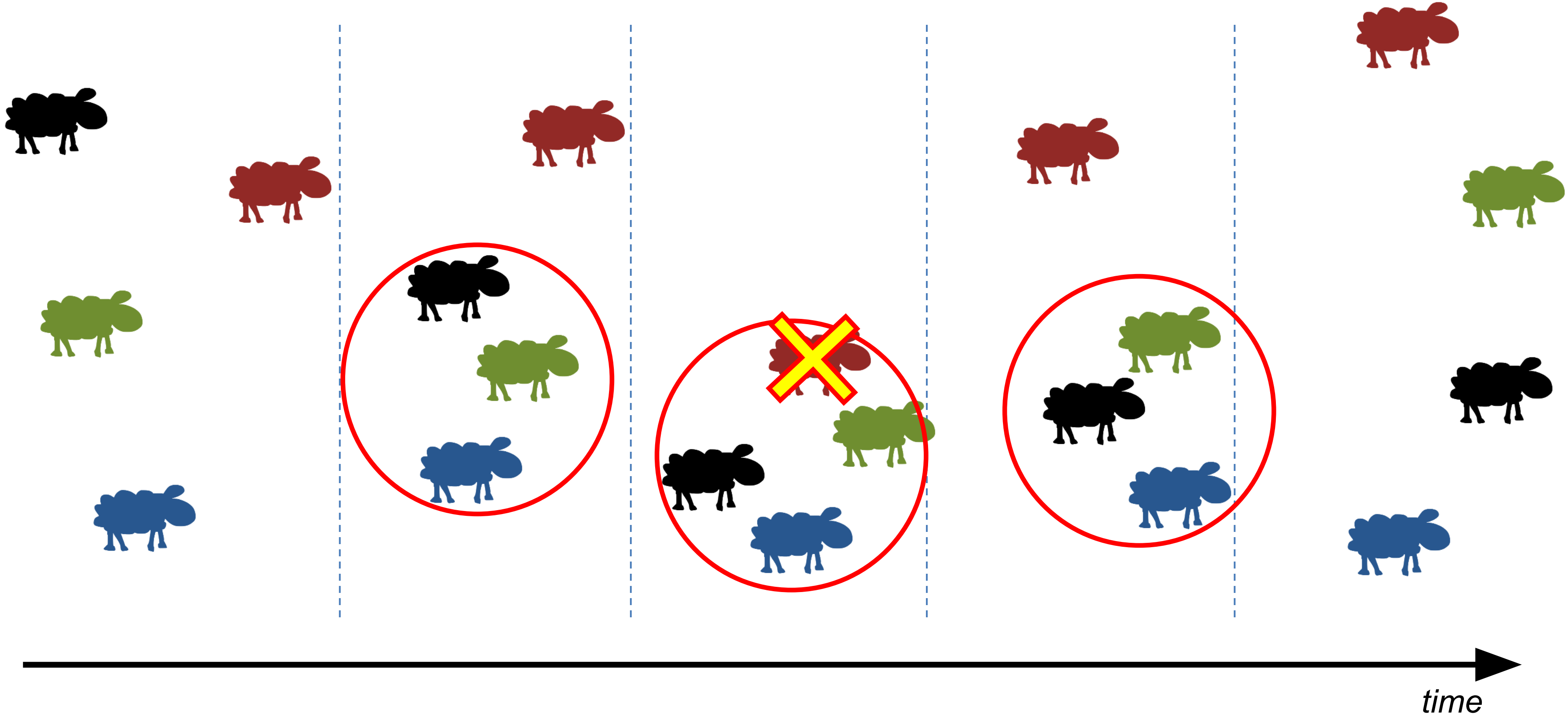


- Group of objects that move together (close to each other) for a time interval

- Discover all possible:
  - sets of objects O, with |O| > min_size and
  - time intervals T, with |T| > min_duration
- such that for all timestamps t ∈T the points in O|t are contained in a circle of radius *r*

M. Wachowicz, R. Ong, C. Renso, M. Nanni: ***Finding moving flock patterns among pedestrians through collective coherence.*** *IJGIS 25(11): 1849-1864 (2011)*

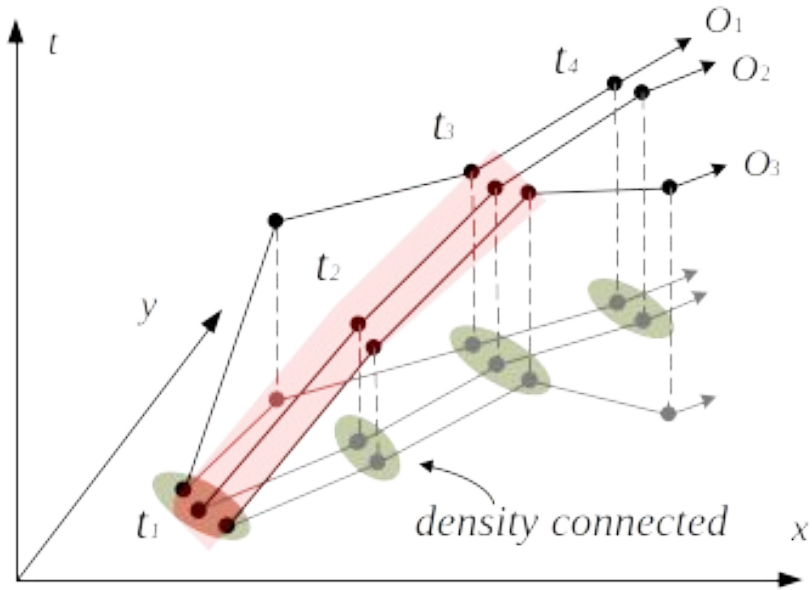
Moving Trajectory Flocks



From Flocks to Convoys

- Given radius r, size m, and time threshold k
  - find all groups of objects so that each group consists of **density-connected objects** w.r.t. r and m
  - during at least k consecutive time points

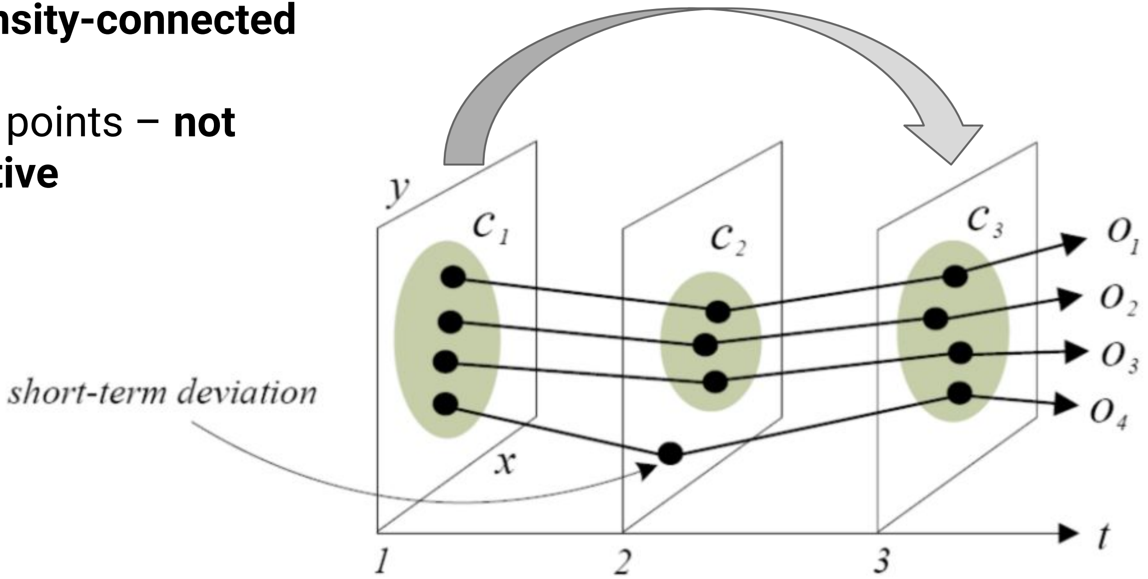
- Basically replace circles with DBSCAN clusters



From Convoys to Swarms

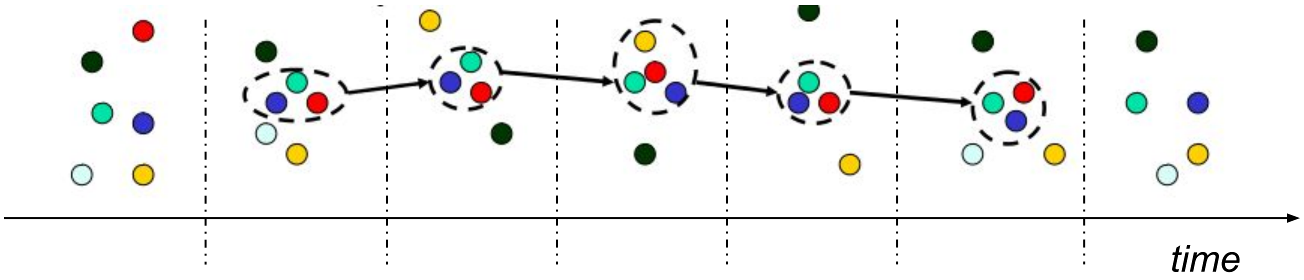
- Given radius r, size m, and time threshold k
  - find all groups of objects so that each group consists of **density-connected objects** w.r.t. r and m
  - during at least k time points – **not necessarily consecutive**

swarm pattern = $\{O_1, O_2, O_3, O_4\}$
over times <1,3>



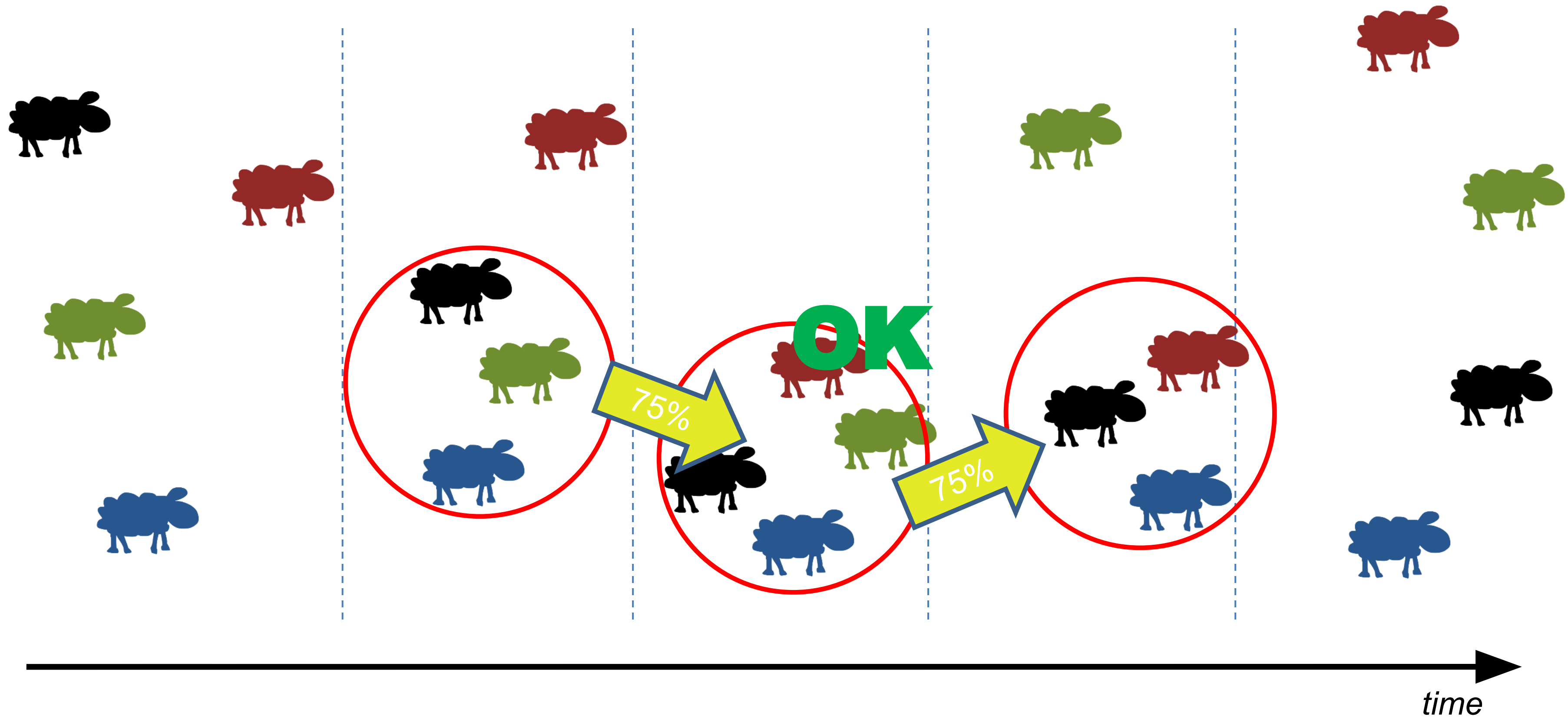
Moving Clusters

- A ***moving cluster*** is a set of objects that move close to each other for a long time interval



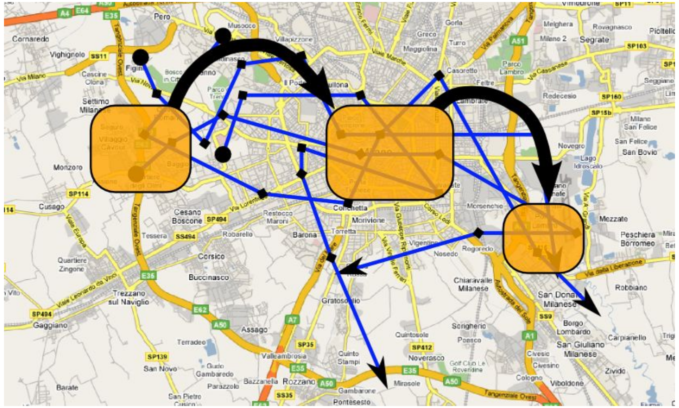
- Formal Definition [Kalnis et al., SSTD'05]:
  - A ***moving cluster*** is a sequence of (snapshot) clusters ***c1, c2, …, ck*** such that for each timestamp i ($1 \leq i < k$): Jaccard(c_i, c_{i+1}) $\geq \theta$
    - Jaccard(c_i, c_{i+1}) = $|c_i \cap c_{i+1}| \;/\; |c_i \cup c_{i+1}|$
    - $0 < \theta \leq 1$
  - Clustering computed with density-based method (DBSCAN)

Moving Clusters



T-Patterns

- A sequence of visited regions, **frequently** visited in the **specified order** with **similar transition times**

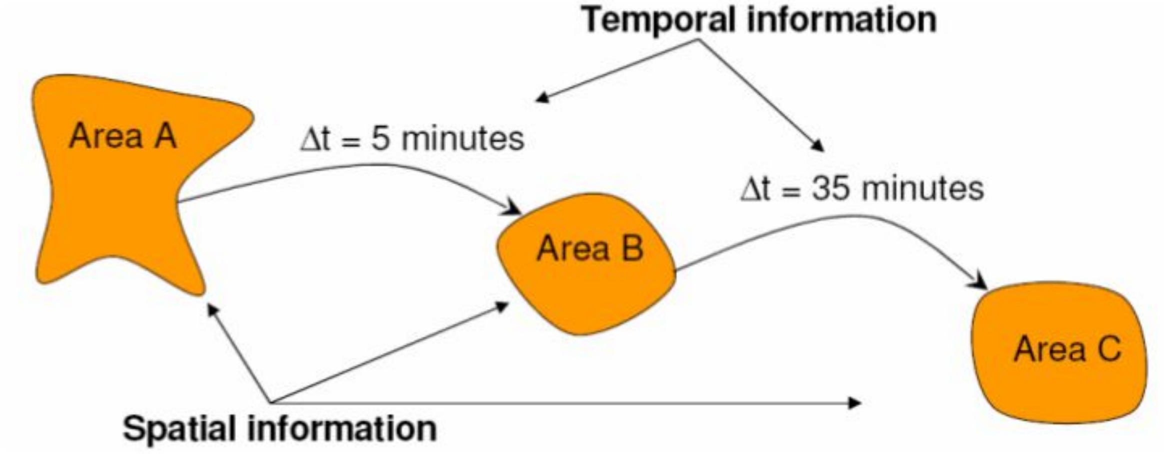


$$A_0 \xrightarrow{t_1} A_1 \xrightarrow{t_2} \dots \; A_{n-1} \xrightarrow{t_n} A_n$$

t_i = transition time, A_i = spatial region

Giannotti, Nanni, Pedreschi, Pinelli. Trajectory pattern mining. Proc. ACM SIGKDD 2007

T-Patterns

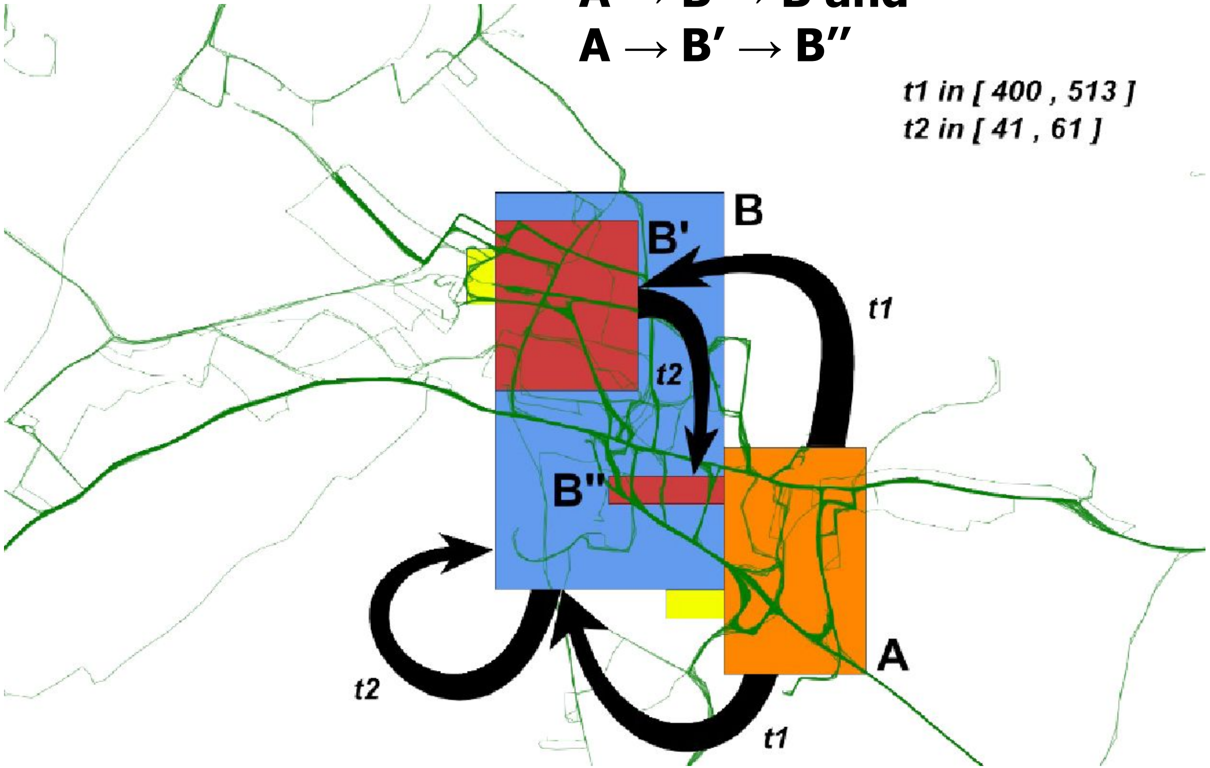


- Key features
  - Includes typical transition times in the output
  - Areas are automatically detected – not “the easy way”

Sample Trajectory Pattern

Data Source: Trucks in Athens (273 trajectories)

A → B → B and
A → B′ → B″

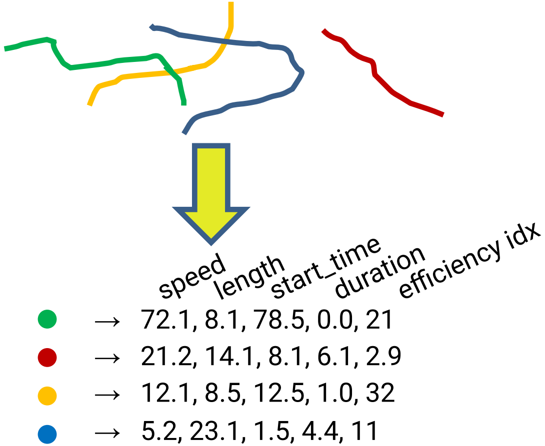


A quick peek into Deep Learning

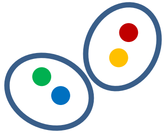
Deep Learning approaches to Trajectory Clustering

Traditional approach

- Preprocess the data to obtain features

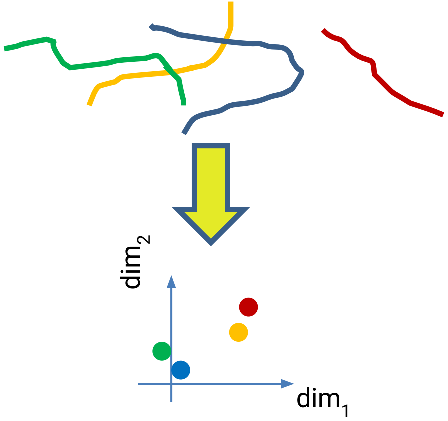


- Clustering over features

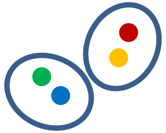


Deep learning approach

- Learning a latent representation (or embeddings)

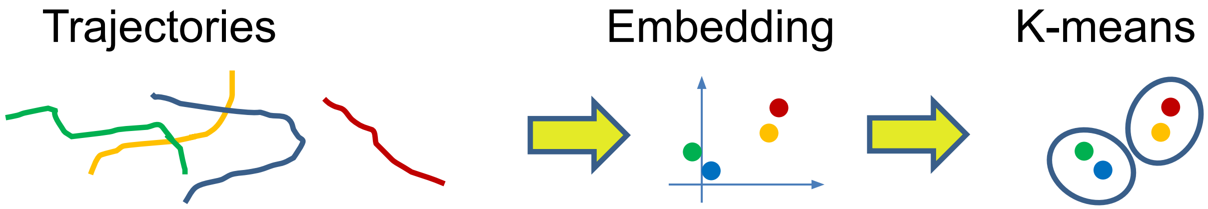


- Clustering over embeddings



Deep Learning approaches to Trajectory Clustering

- Sample approach: DETECT: Deep Trajectory Clustering for Mobility-Behavior Analysis
- Basic idea:

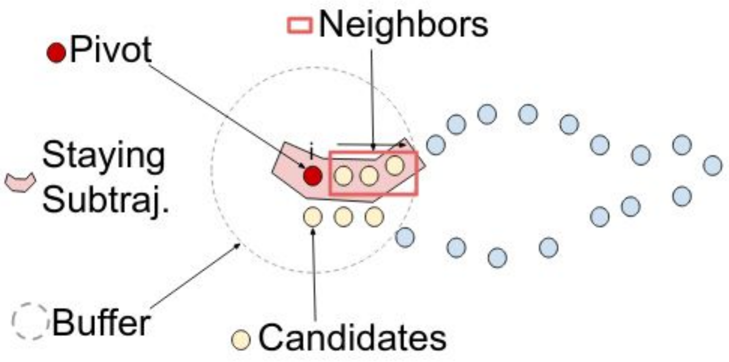


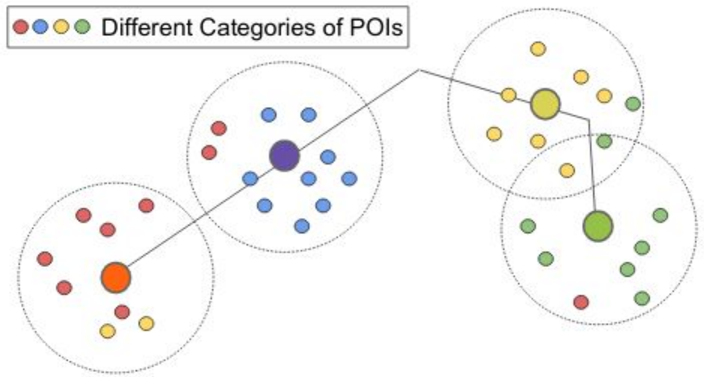
- Integrate the clustering step in the learning of embeddings
- Three steps:
  - Enrich trajectories with context
  - LSTM-based embedding of trajectories
  - Clustering on embeddings

M. Yue, Y. Li, H. Yang, R. Ahuja, Y. Chiang and C. Shahabi, "DETECT: Deep Trajectory Clustering for Mobility-Behavior Analysis," 2019 IEEE International Conference on Big Data (Big Data), 2019, pp. 988-997

DETECT / 1

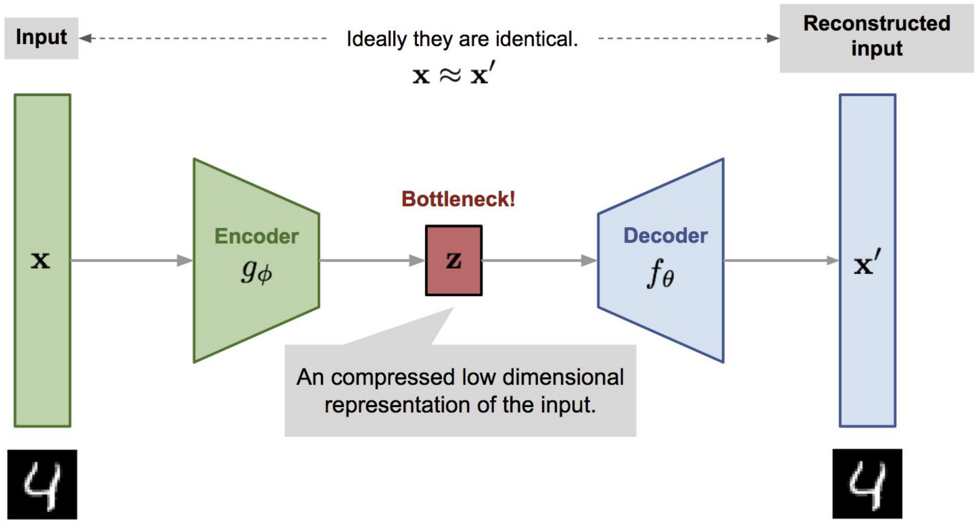
- Enrich trajectories with context
  - Identify stay areas = segment of trajectory where there is no movement, basically a stop
  - Create a buffer around the area
  - Select all points-of-interest located there (hotels, shops, etc.)
  - Compute a feature vector, one feature per PoI category
- Output
  - Traj = < $(x,y,[f_1,\ldots, f_n])$, $(x',y',[f'_1,\ldots, f'_n])$, … >





DETECT / 2

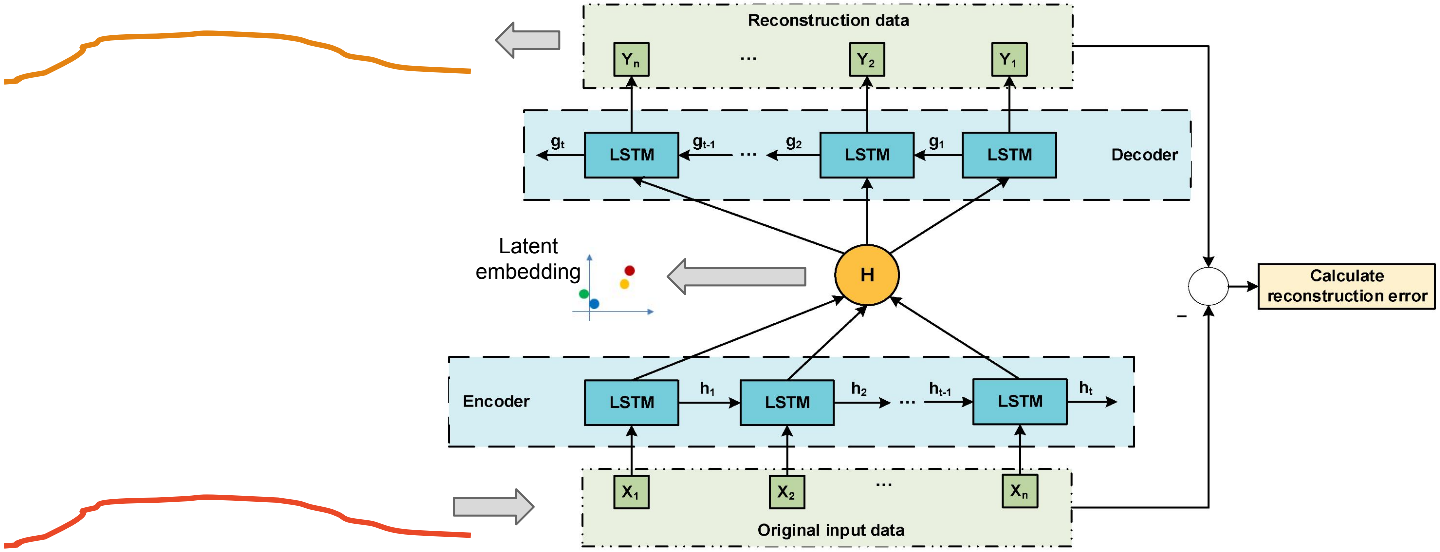
- LSTM-based embedding of trajectories
  - Apply a encoder-decoder schema to the enriched trajectories
  - Use LSTM as basic mechanism



- Objective: minimize the difference between the encoder input and the decoder output

DETECT / 2

- LSTM-based embedding of trajectories
  - Apply a encoder-decoder schema to the enriched trajectories
  - Use LSTM as basic mechanism

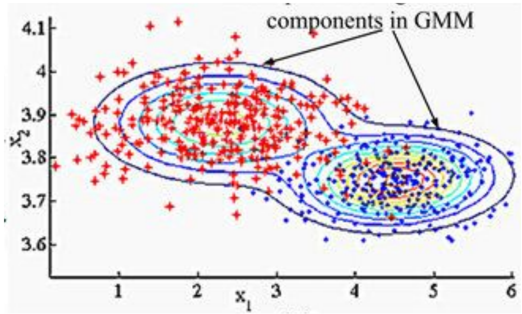


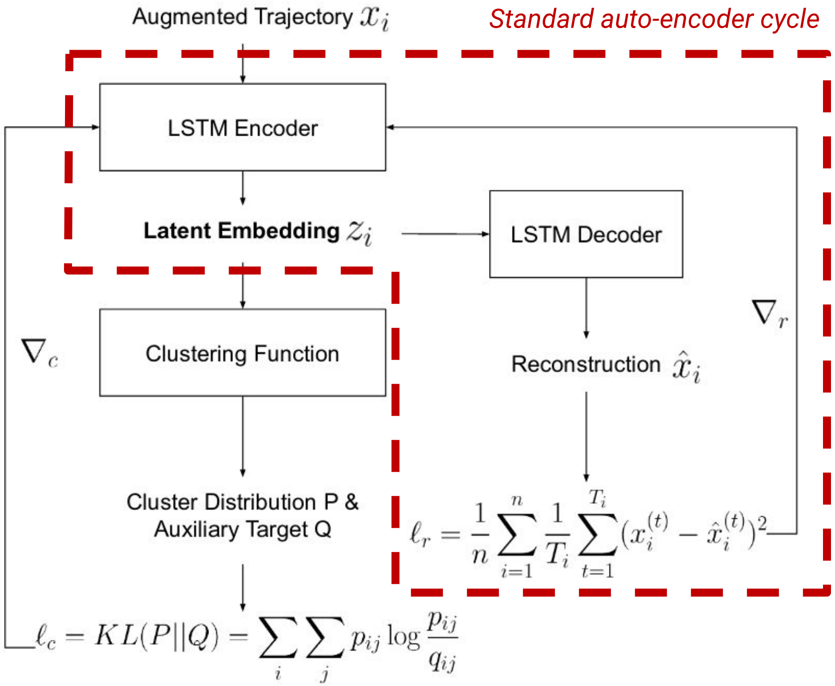
- Objective: minimize the difference between the encoder input and the decoder output

DETECT / 3

- Clustering on embeddings

- Clustering error becomes one term of the overall loss function
- P & Q = points distribution
  - P = real data (embedded)
  - Q = clusters (Student t-distribution around centers)





Homeworks

Food for thought

- **Hausdorff**: let interpret trajectory T=<p_1, p_2, …, p_n> as a polyline, thus containing also the segments (= infinite sets of points) between each pair p_i,p_{i+1}. How can you compute the Hausdorff distance between two trajectories?

- **Local patterns in mobility**: can you find some examples in urban mobility where local patterns might be useful? Frequent sequences? Flocks?

- **Local or global:** we discover that 90% of vehicles in a city pass through the same road segment (maybe a bridge). Is that a local or global pattern? Is there really a difference?

- **The thin line between clusters and flows**: if you take all the trajectories that form a specific flow (same origin and same destination), how many clusters do you expect to find? What is the difference between a flow and a (trajectory) cluster?

to study for the exam

Material

- [paper] Spatio-Temporal Trajectory Similarity Measures: A Comprehensive Survey and Quantitative Study, Danlei Hu et al., arXiv: https://arxiv.org/abs/2303.05012v2
  - Sections 1, 2, 3 (only the measures seen in these slides)
- [paper] Computing longest duration flocks in trajectory data, Joachim Gudmundsson and Marc van Kreveld (2006), GIS '06, https://dl.acm.org/doi/10.1145/1183471.1183479
  - Section 1 (definitions)
- [paper] Discovery of Convoys in Trajectory Databases, Hoyoung Jeung et al., VLDB 2008, https://arxiv.org/abs/1002.0963v1
  - Section 3 (definitions)

to study for the exam

Material

- [paper] On Discovering Moving Clusters in Spatio-temporal Data, Kalnis, P., Mamoulis, N., Bakiras, S. SSTD 2005. https://doi.org/10.1007/11535331_21
  - Sections 1, 2, 4.1 (definitions and basic algorithm)
- [paper] Trajectory pattern mining, Giannotti, Nanni, Pedreschi, Pinelli. KDD 2007. https://dl.acm.org/doi/10.1145/1281192.1281230
  - Section 3 (definitions)
- [paper] DETECT: Deep Trajectory Clustering for Mobility-Behavior Analysis, M. Yue et al. Big Data 2019. https://arxiv.org/abs/2003.0135
  - Section II (focus on definitions and overall approach, not details)