Mobility Data Management & Exploration

Ch. 07. Mobility Data Mining and Knowledge Discovery

Nikos Pelekis & Yannis Theodoridis

InfoLab | University of Piraeus | Greece
infolab.cs.unipi.gr

v.2014.05

Chapter outline

7.1. Clustering in Mobility Data

7.2. Moving Clusters for Capturing Collective Mobility Behavior

7.3. Sequence Pattern Mining in Mobility Data

7.4. Prediction and Classification in Mobility Data

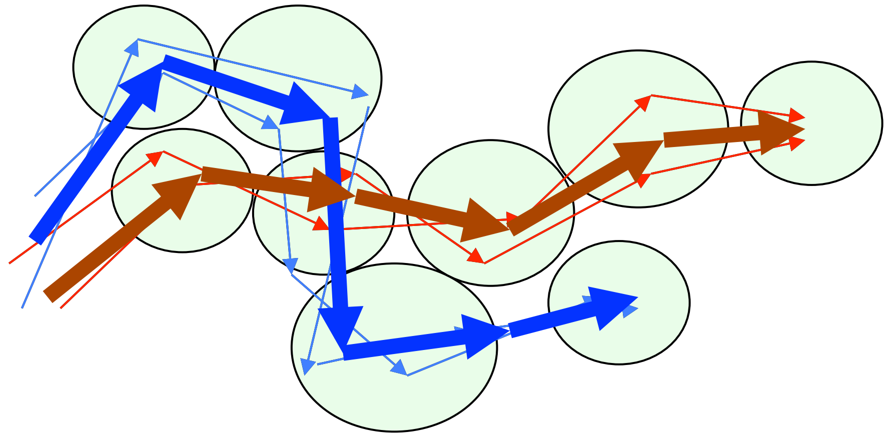
7.5. Summary

7.1.

Clustering in mobility data

Trajectory clustering

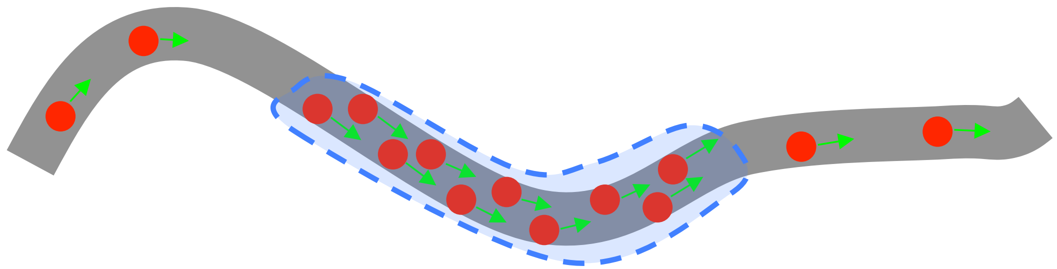
- Technical objectives:
  - Cluster trajectories w.r.t. similarity
    - For each cluster, find its 'centroid' or 'representative'
  - Discover moving clusters (flocks), outliers, etc.



- Issues:
  - Which distance between trajectories? recall Chap. 6
  - Which kind of clustering? Partitioning (K-means-like) vs. Density-based (DBSCAN- or OPTICS- like) solutions
  - How does a cluster 'centroid' or 'representative' look like?

Trajectory clustering (cont.)

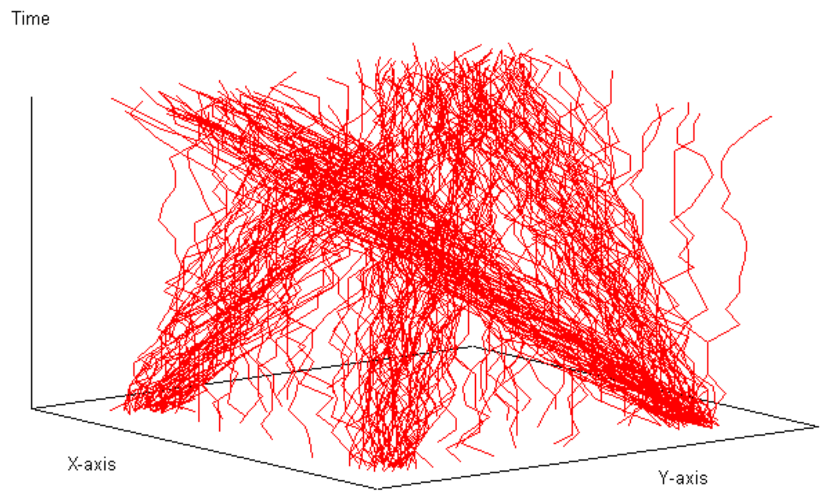
- General requirements:
  - Tolerance to noise; Low computational cost; Applicability to complex, possibly non-vectorial data; Non-spherical clusters; etc.
    - E.g.: A traffic jam along a road = "snake-shaped" cluster

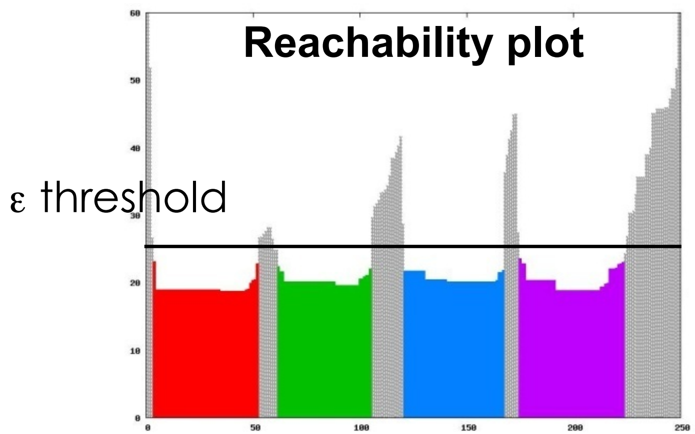


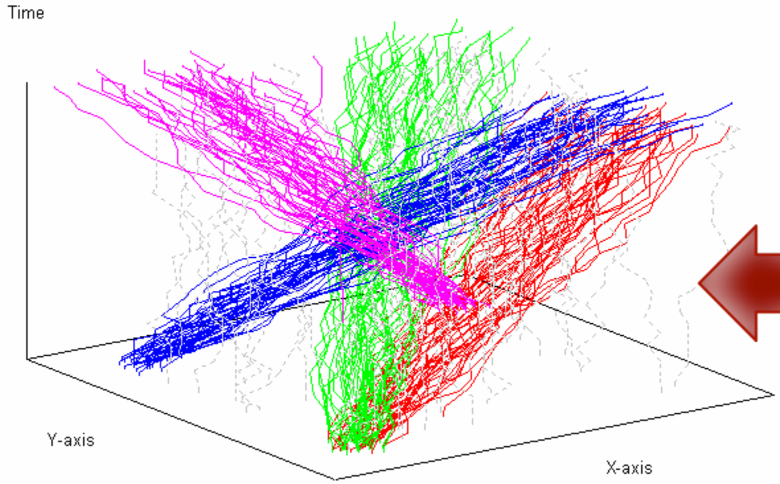
- State-of-the-art
  - Clustering on entire trajectories: **T-OPTICS** (2006), **CenTR-I-FCM** (2009)
  - Clustering on sub-trajectories: **TRACLUS** (2007), **NEAT** (2012)

T-OPTICS (Trajectory OPTICS)

- Builds upon OPTICS and the DISSIM function between trajectories
- Reachability plot (valleys and hills)
  - Valleys → clusters
  - Hills → noise

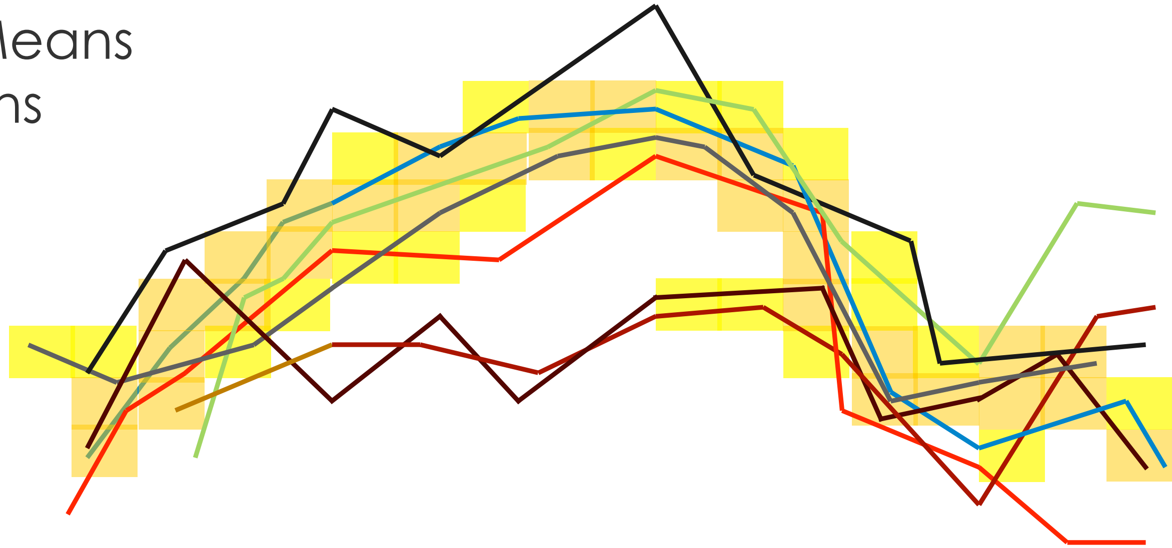






CenTR-I-FCM (Clustering under uncertainty)

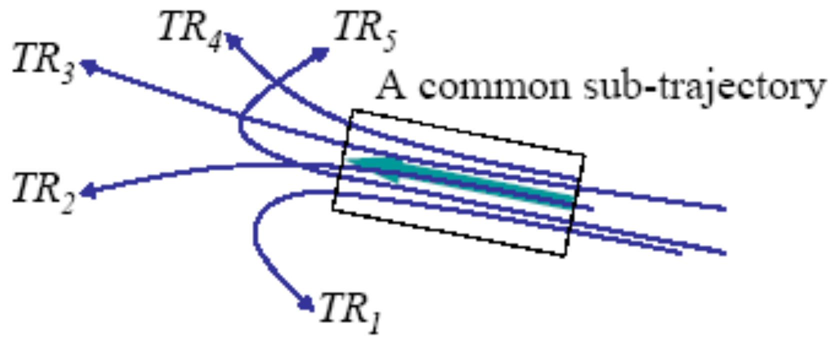
- Builds upon Fuzzy-C-Means (a variation of K-means for uncertain data)
- Motivation:
  - uncertainty of trajectory data should be taken into account



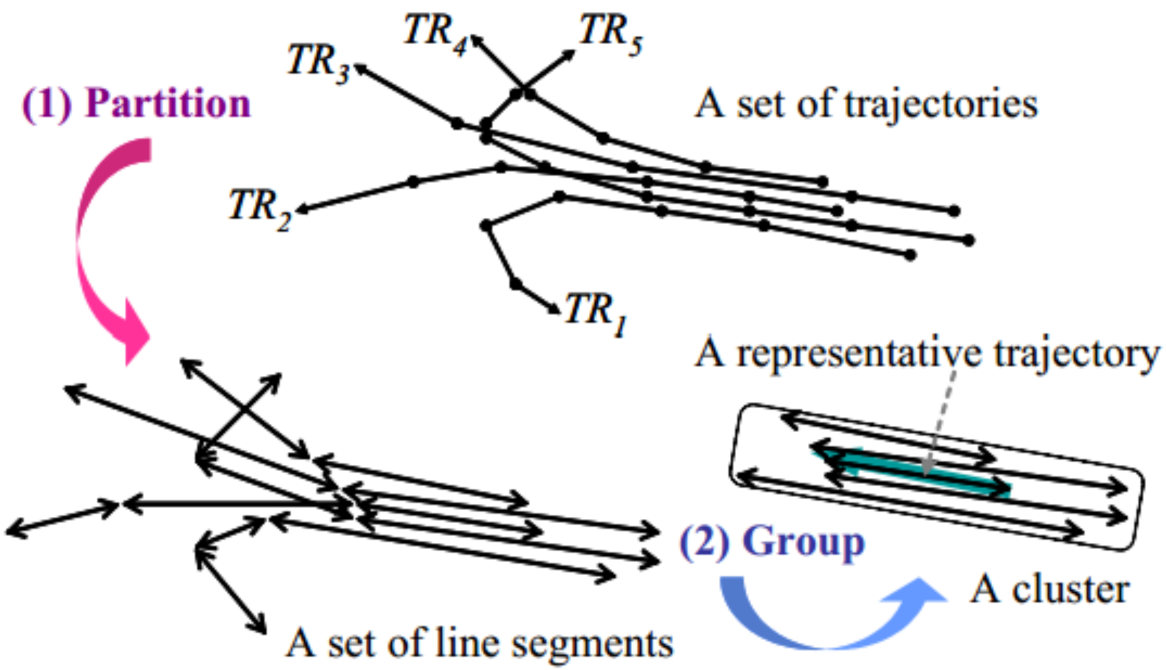
- Three phases:
  - Step 1: **mapping** of trajectories in an intuitionistic fuzzy vector space
  - Step 2: **discovering the centroid** of a bundle of trajectories (algorithm CenTra)
  - Step 3: **clustering** trajectories under uncertainty (algorithm CenTR-I-FCM)

TRACLUS (Trajectory Clustering)

- Discovers similar portions of trajectories (sub-trajectories)

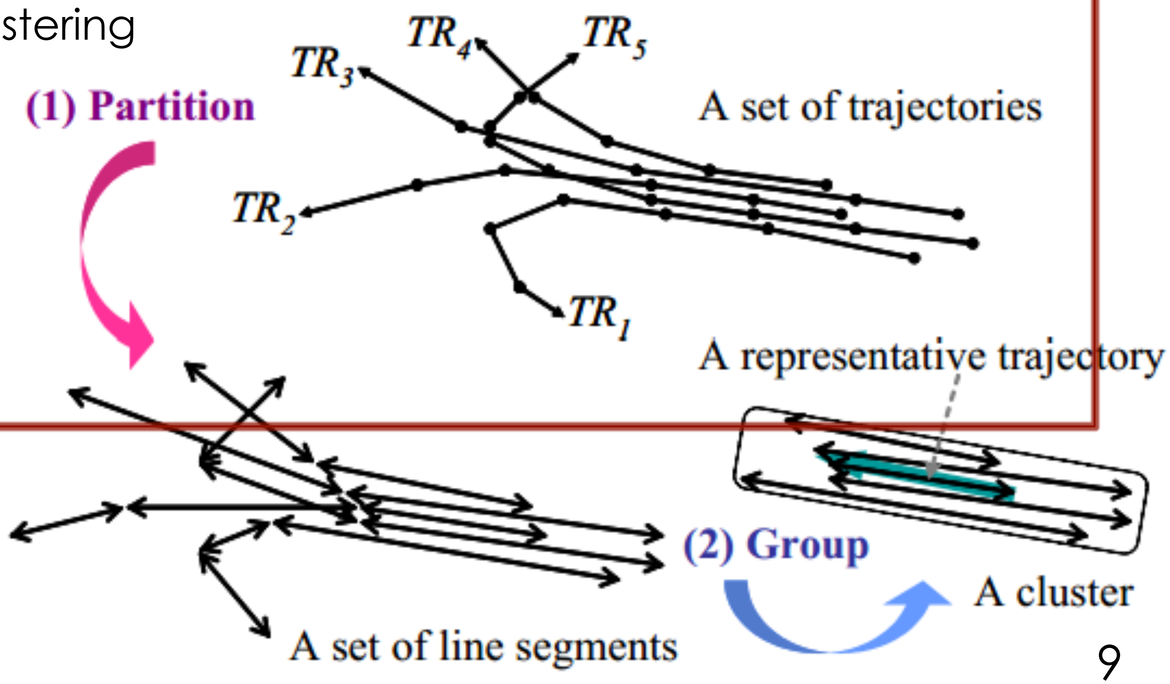


- Works in two phases:
  - **partitioning**
  - **grouping**



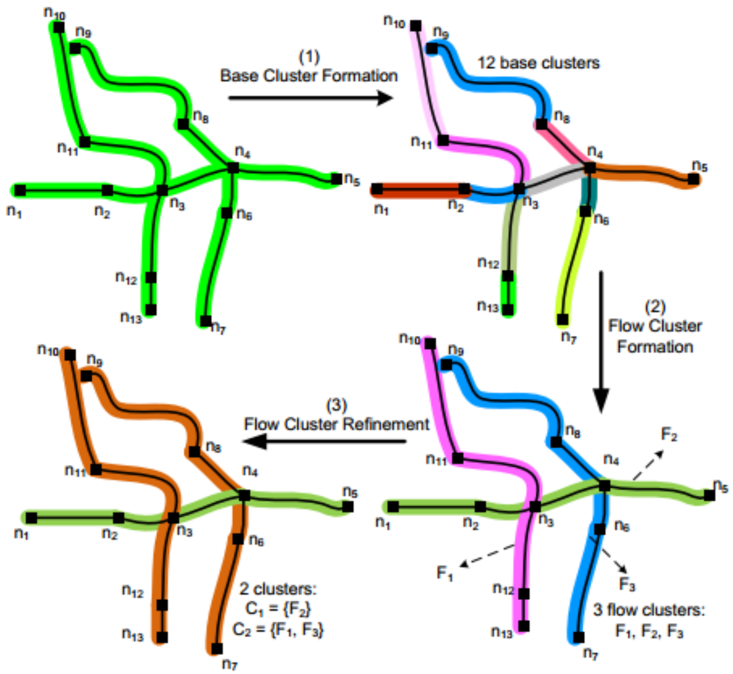
TRACLUS (cont.)

<u>Algorithm TRACLUS</u>
Input: A dataset of trajectories D={tr_1, …, tr_N}
Output: (1) A set of clusters C={C_1, …, C_O}, (2) A set of representative trajectories
/* Partitioning phase */
1. for each tr_i in D do
2. Execute trajectory partitioning; Get a set S of line segments as the result;
3. Accumulate S into a set D';
/* Grouping phase */
4. Execute Line Segment clustering for D';
Get a set C of clusters as the result;
5. for each C_j in C do
6. Execute RTG;
Get a representative trajectory as the result;

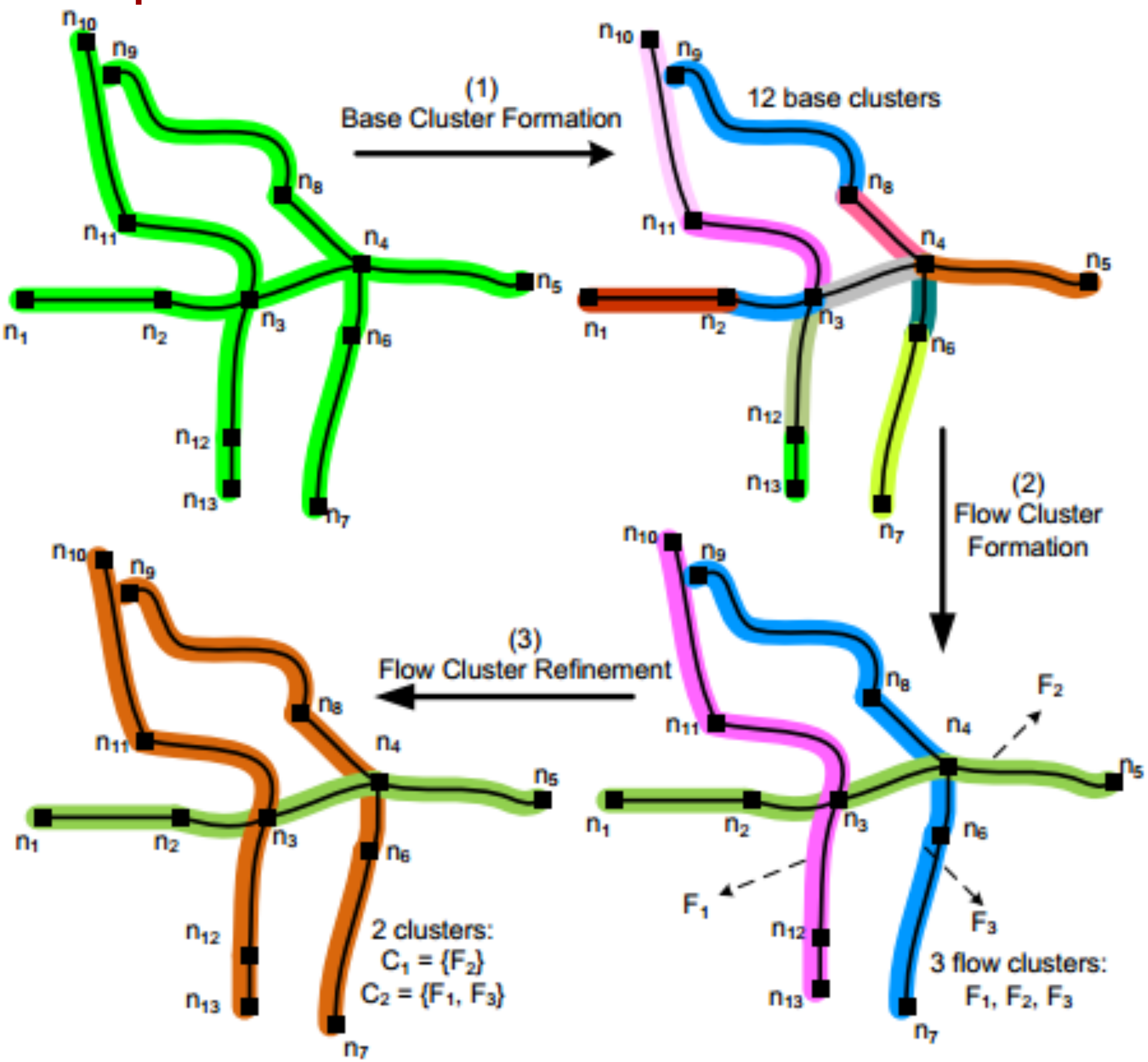


NEAT

- Clusters trajectories moving on road network.
- Works in three phases:
  - **base cluster formation**: partitions trajectories into t-fragments, where each one lies on a single road segment, and forms base clusters, each containing the t-fragments that lie on the same road segment
  - **flow cluster formation**: combines base clusters w.r.t. merging selectivity
    - function that takes into account flow, density and road speed factors
  - **flow cluster refinement**: compresses flow clusters (DBSCAN variant)

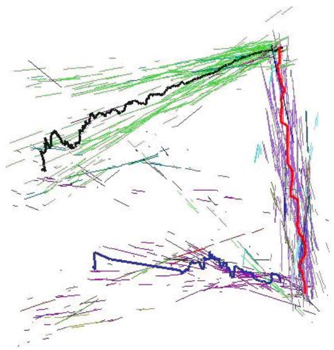


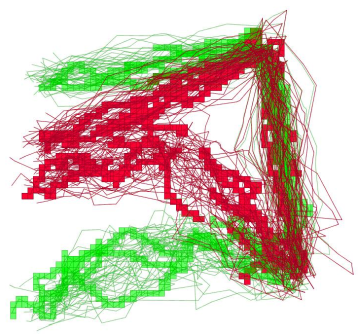
NEAT example



Finding representatives

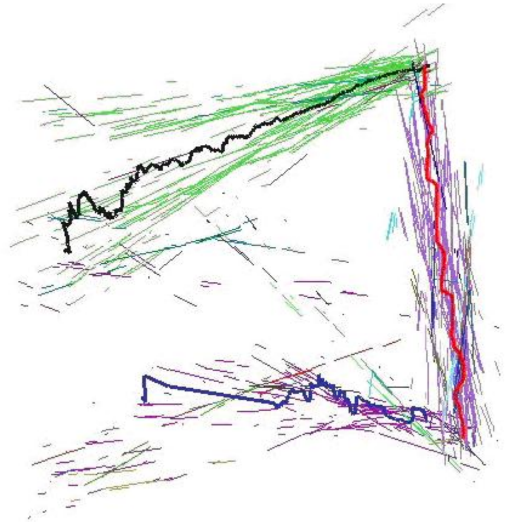
- … in trajectory datasets
- Recall one of the trajectory clustering issues:
  - How does a cluster ‘centroid’ or ‘representative’ look like?
- Some of the trajectory clustering algorithms inherently provide such ‘centroids’ or ‘representatives’ (actually, artificial trajectory-like shapes)
  - TRACLUS representatives
  - CenTR-I-FCM centroids



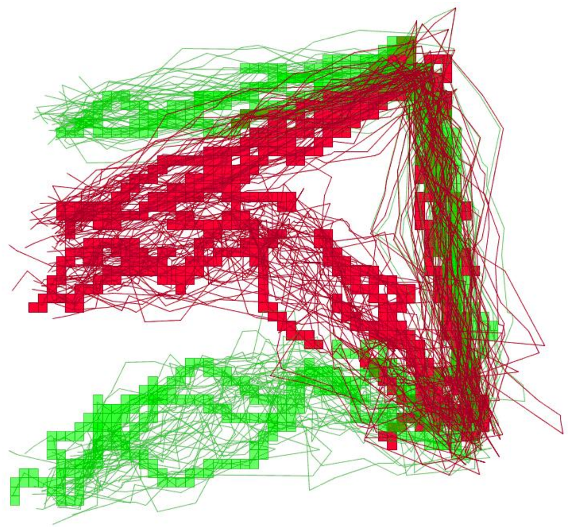


Finding representatives (cont.)

- **TRACLUS representatives**
  - three thick lines (black, red, blue)
  - compositions of segments (sub-trajectories)
  - a trajectory may be split into different clusters

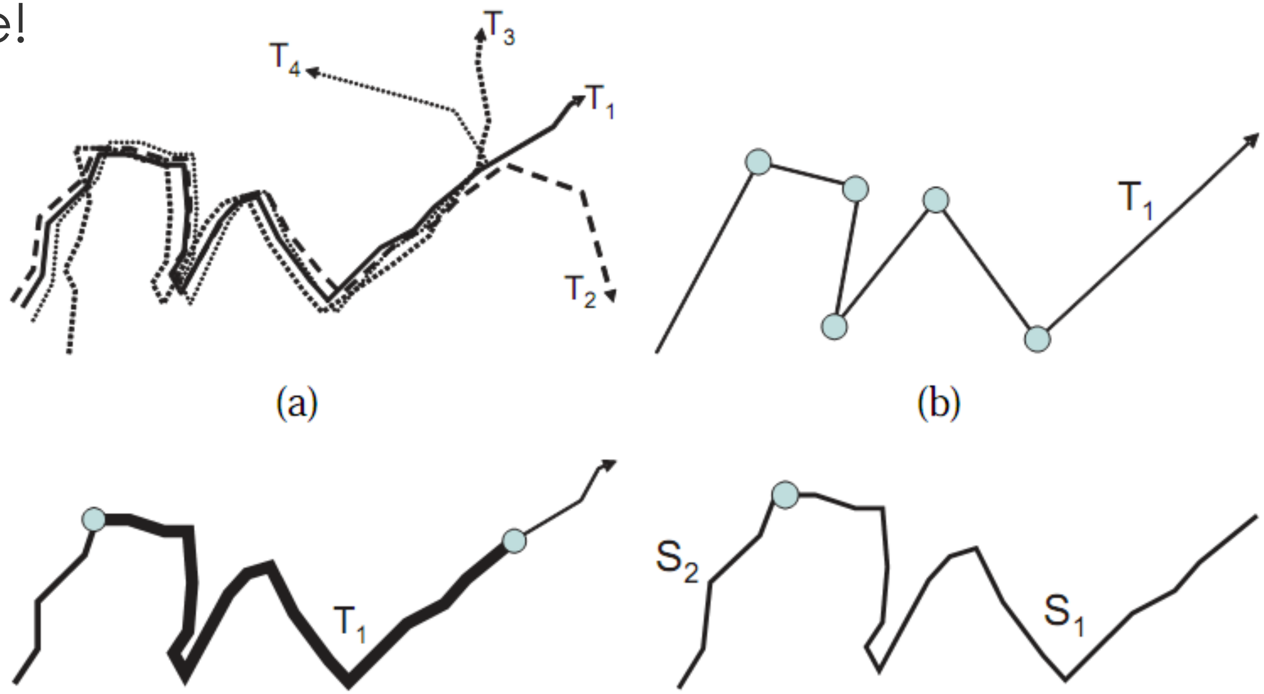


- **CenTR-I-FCM centroids**
  - two sets of cells (green, red)
  - catch the overall complex mobility patterns



Finding representatives (cont.)

- Another approach: **T-sampling** (2010, 2012)
  - Samples the top-k most representative trajectories …
  - … following a deterministic voting methodology
- Trajectory segmentation is neighborhood- rather than geometry-aware!
- Example:



(a) (b)

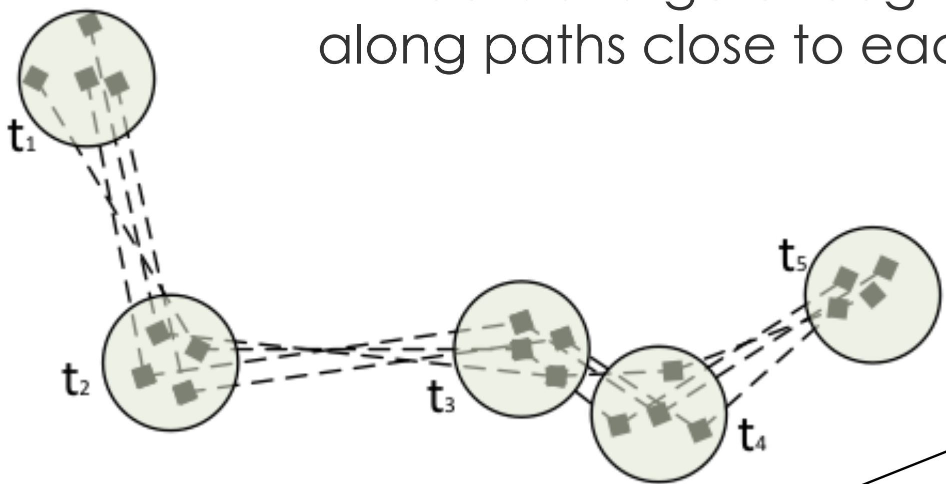
(c) (d)

7.2.

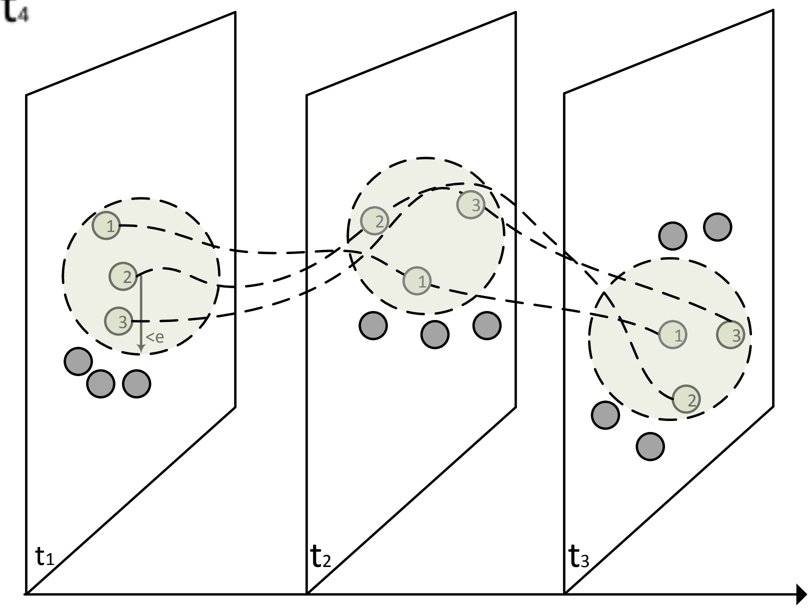
Moving clusters for capturing collective mobility behavior

Flocks and variants

- **Flock**: a large enough subset of objects moving along paths close to each other for a certain time

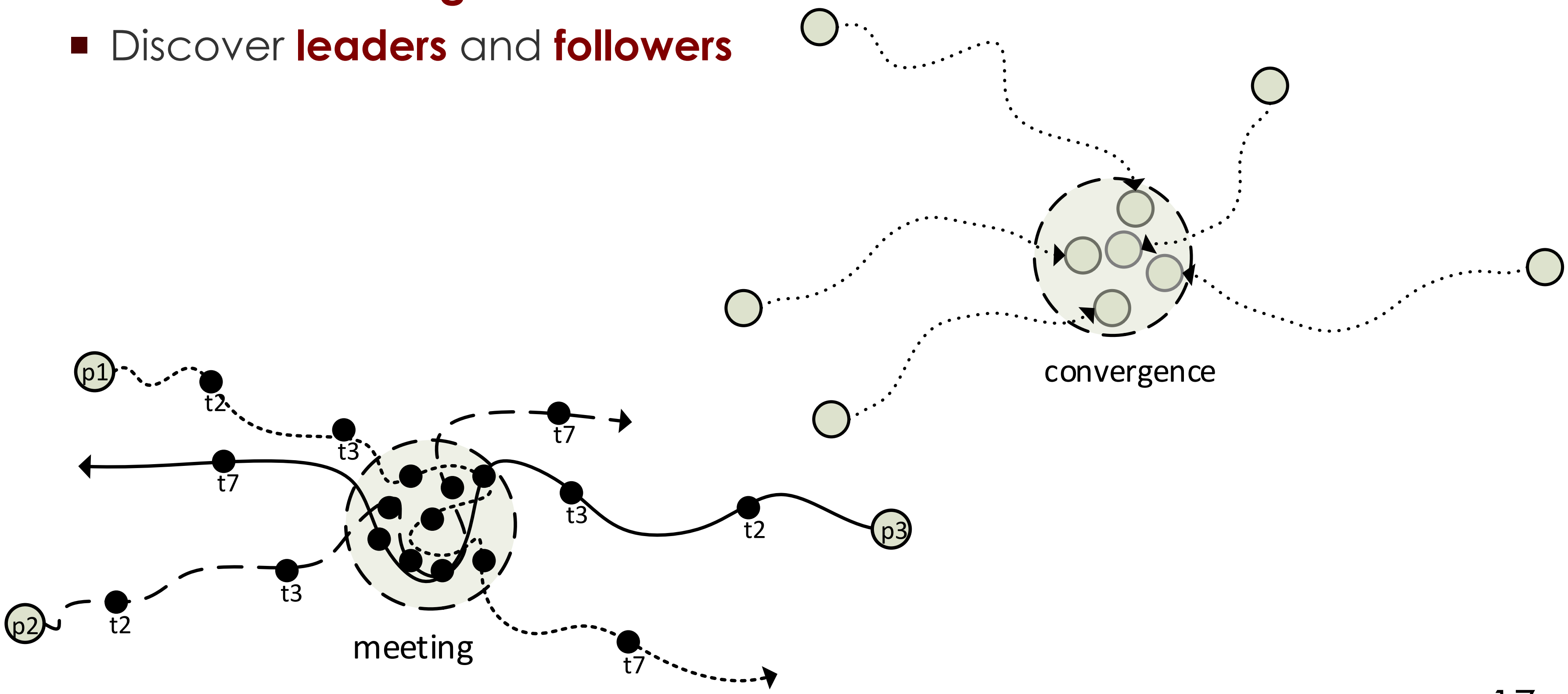


- Side-effect:
  the **lossy-flock problem**



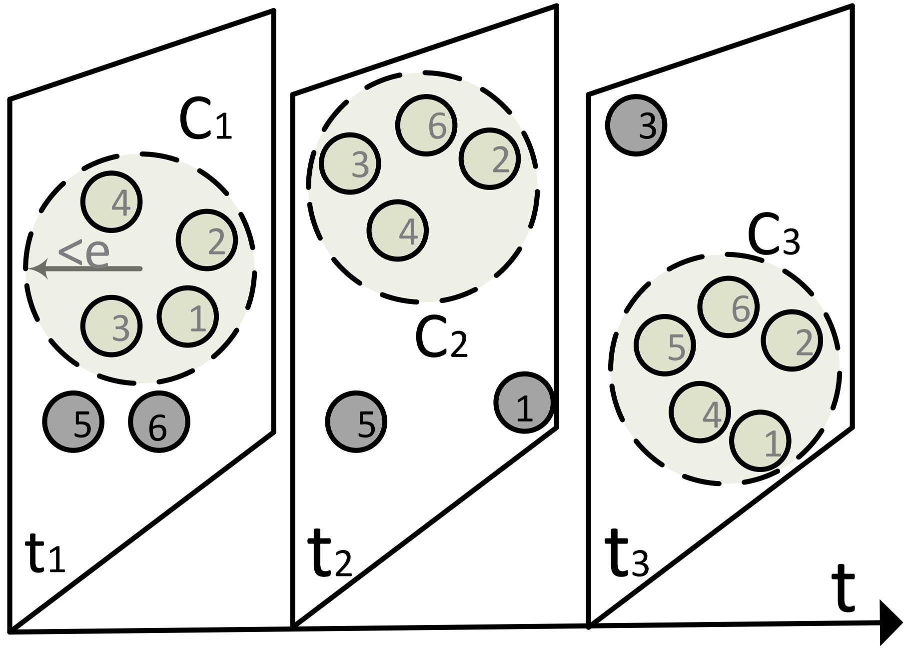
Flocks and variants (cont.)

- Interesting problems arise over the flock concept:
  - Identify long flock patterns (**top-k longest flock pattern discovery**)
  - Discover **meetings** (fixed- vs. varying- versions)
  - Discover **convergences**
  - Discover **leaders** and **followers**



Moving clusters

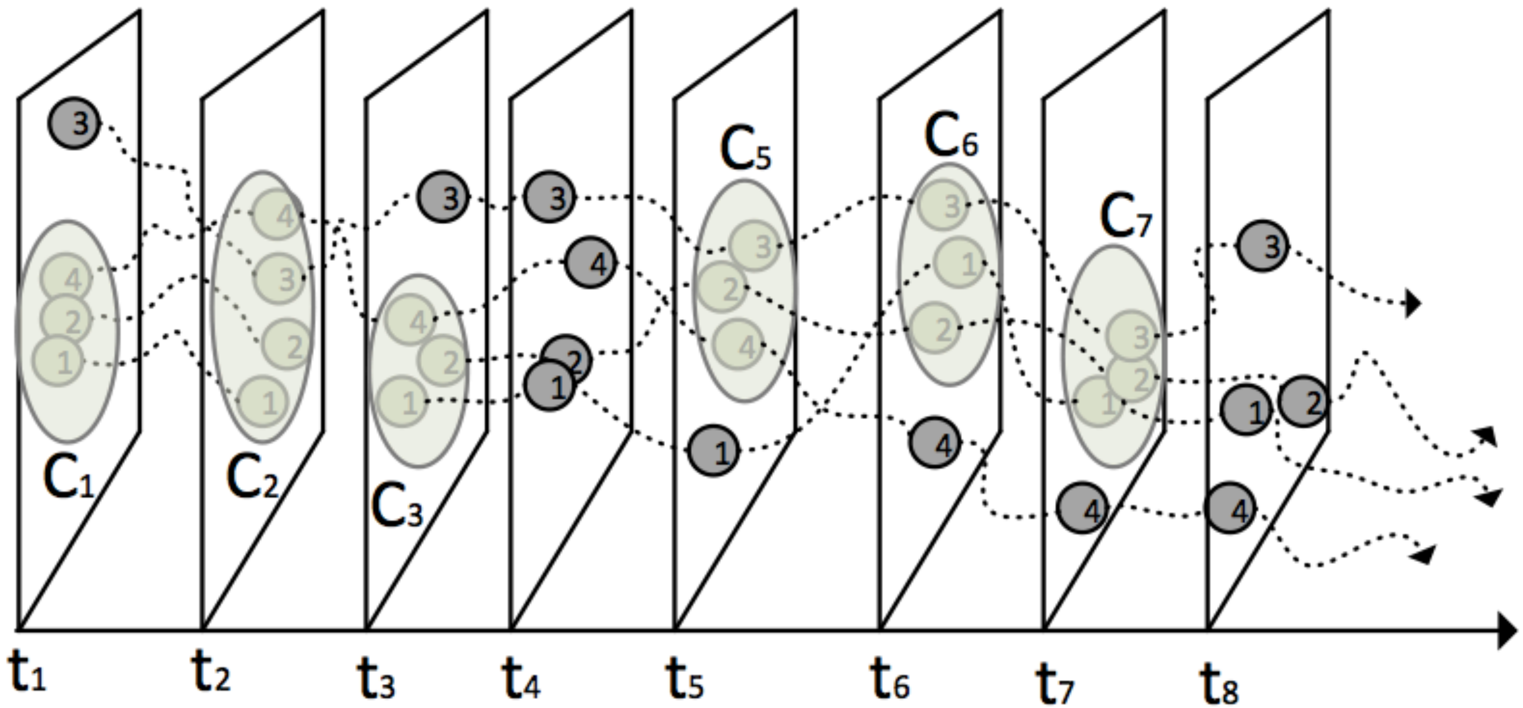
- **Moving cluster**: a sequence of spatial clusters in consecutive timepoints that keep a large percentage of common objects
  - a kind of varying flock
- Example:



- Issue:
  - The 'consecutive time' constraint may result in the loss of interesting patterns

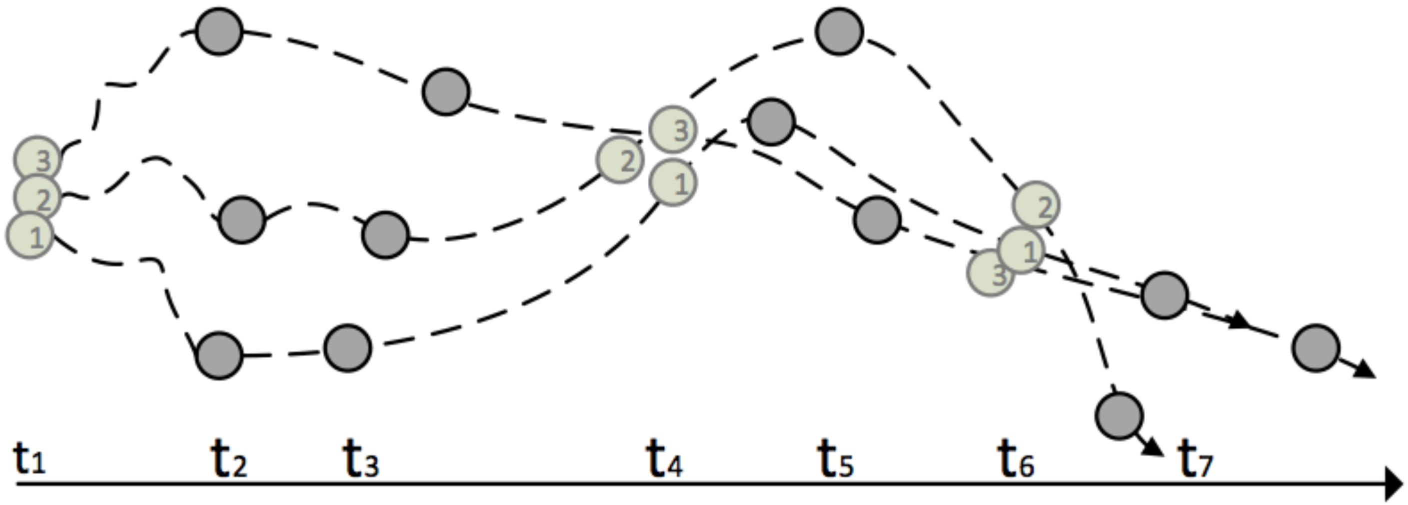
Convoys

- **Convoy**: a group of objects that has at least *m* objects, which are density-connected with respect to a distance threshold *e*, during *k* consecutive timepoints



Group patterns and Swarms

- **Group pattern**: a set of moving objects that travel within a radius for certain timestamps that maybe non-consecutive
  - actually, a time-relaxed flock pattern
- **Swarm**: a collection of moving objects with cardinality at least m, that are part of the same cluster for at least k timepoints
  - timestamps are not required to be consecutive
  -



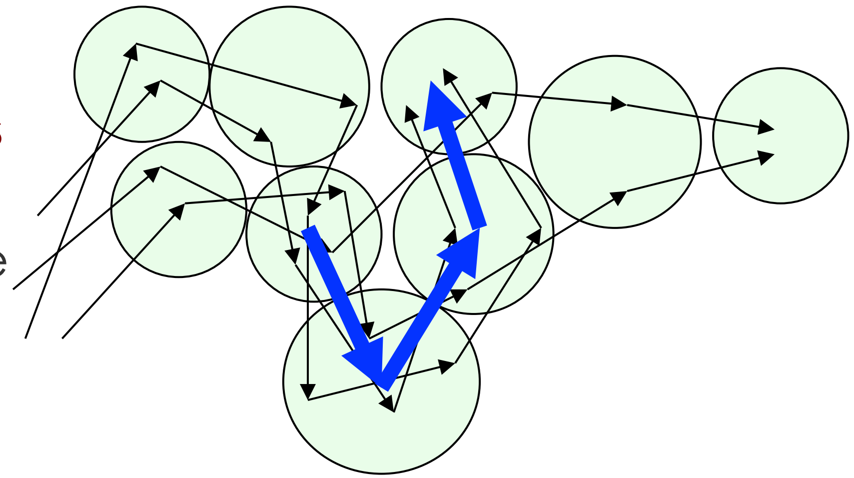
7.3.

Sequence pattern mining in mobility data

Frequent pattern mining

- Technical objectives:
  - Identify ‘frequent’ or ‘popular’ patterns
  - Discover hot spots, hot paths, etc.

- Two groups of approaches:
  - techniques that identify regularities in the behavior of a single user: **Periodic patterns** (2007)
  - techniques that reveal collective sequential behavior of a set of users: **T-Patterns** (2007)



T-patterns

- **T-pattern** is a sequence of visited regions, frequently visited in a specific order with similar transition times
- Example:
  - TP1: <(), A> <(9,15), B> (supp:31)
  - TP2: <(), A> <(4,20), C> (supp:26)
  - TP3: <(), A> <(9,12), B> <(10,56), D> (supp:21)

Algorithm T-Pattern (with static regions of interest - RoI)
Input: (1) a set of input trajectories T, (2) a grid G_0, (3) a minimum support/ density threshold δ, (4) a radius of spatial neighborhoods ε, (5) a temporal threshold τ.
Output: A set of pairs (S, A) of sequences of regions with temporal annotations.

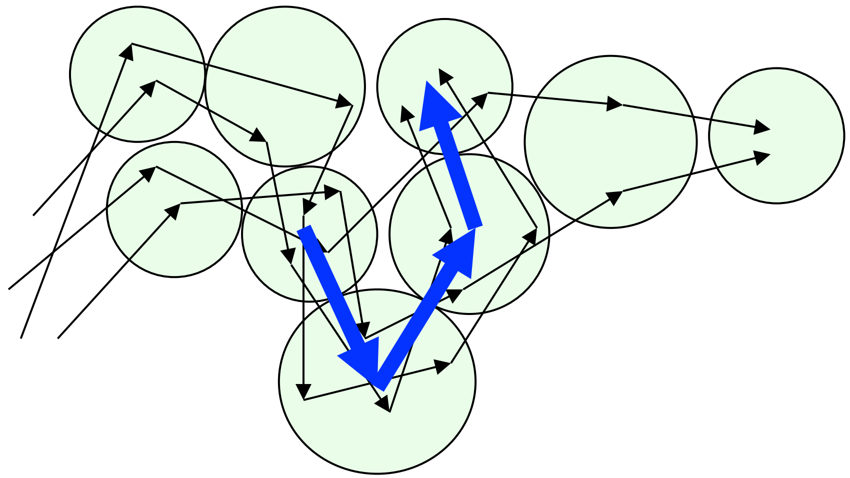
1. G=ComputeDensity(T, G_0, ε);
2. RoI=PopularRegions(G, δ);
3. D=Translate(T, RoI);
4. TAS_mining(D, δ, τ);

7.4.

Prediction and classification in mobility data

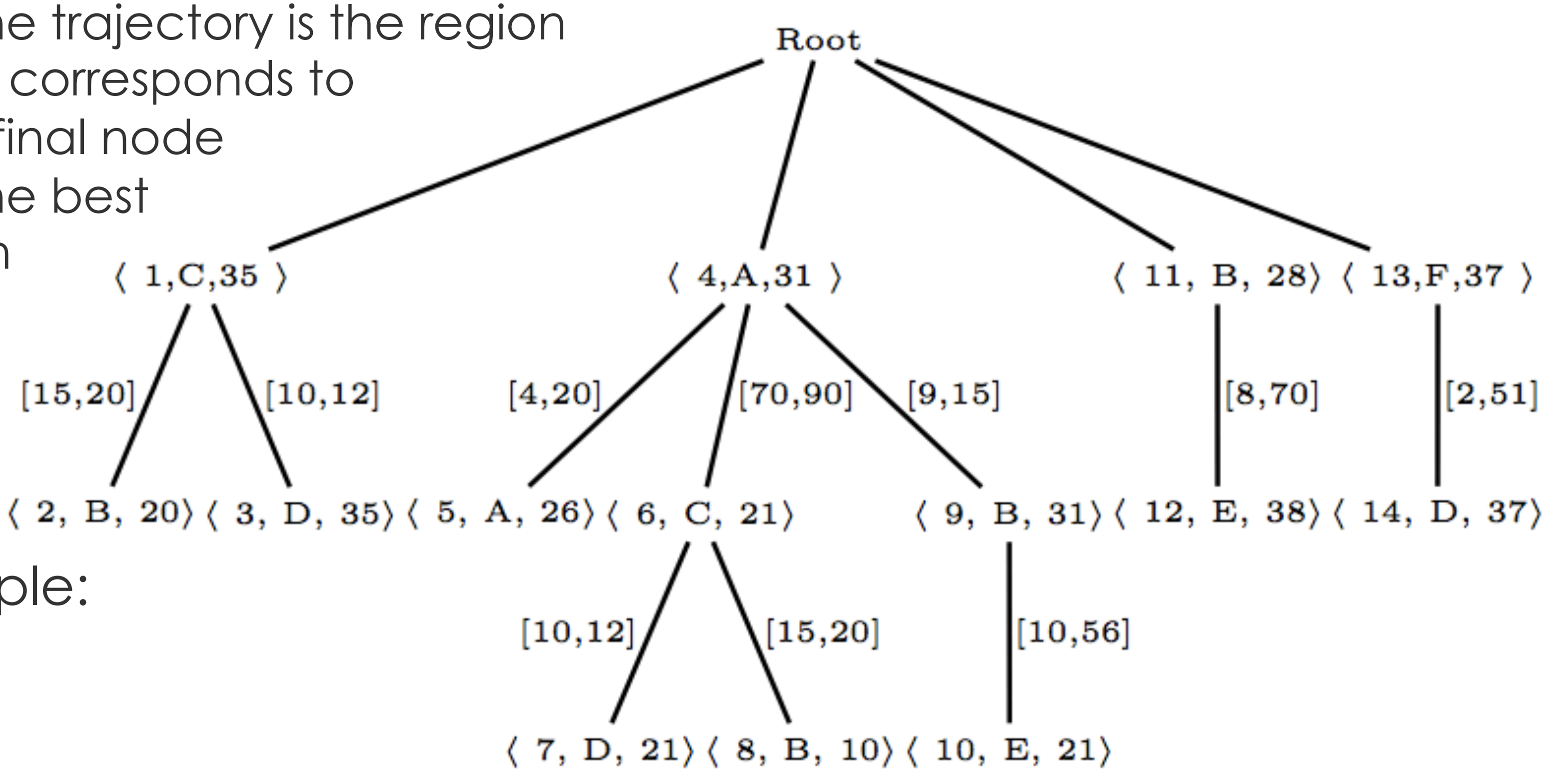
Prediction and Classification

- Goal: to predict the future location of a moving object
- Possible solutions:
  - Naïve : extrapolate w.r.t. current location and velocity vector
  - Alternative: build a prediction model
- State-of-the-art technique:
  - **WhereNext** (2009)



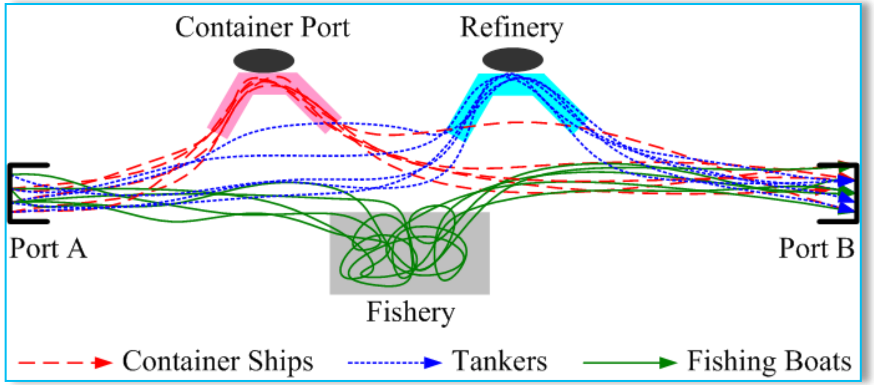
WhereNext

- Builds upon the T-pattern concept: extracts a set of T-patterns and builds a T-pattern tree
  - the best path is found for a given trajectory
  - the predicted future location of the trajectory is the region that corresponds to the final node of the best path
- Example:

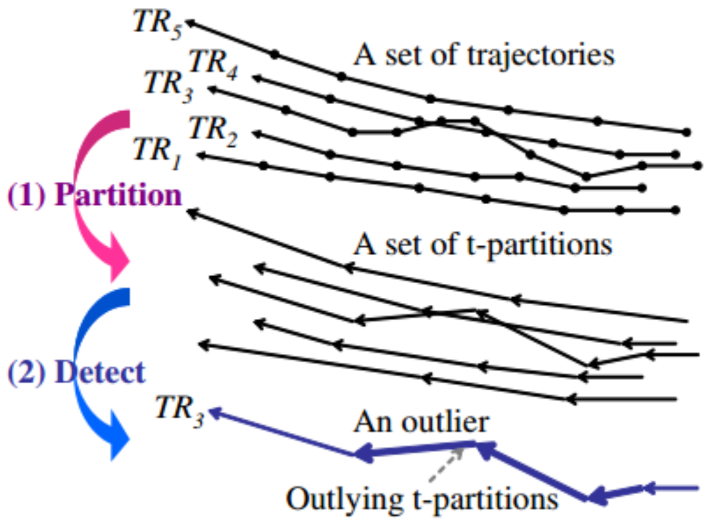


Classification and Outlier detection

- **Classification** aims to predict the class label of a moving object based on its trajectories (and eventually other features)
  - State-of-the-art: **TRACLASS** (2008)

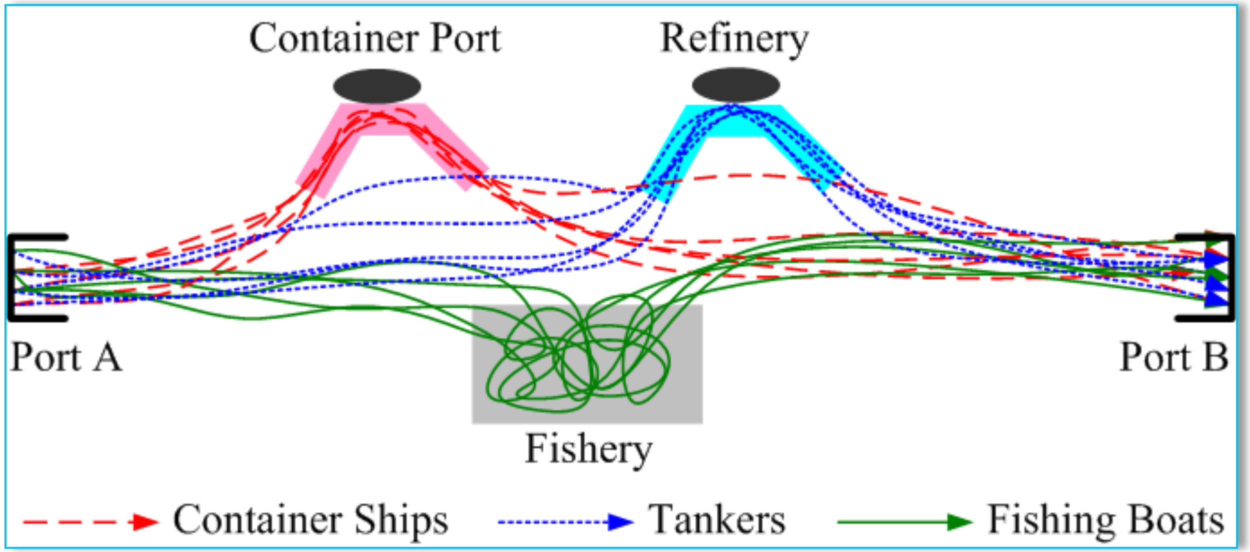


- **Outlier detection** aims to detect among a set of trajectories, those that behave differently from their neighbors
  - State-of-the-art: **TRAOD** (2008)



TRACLASS

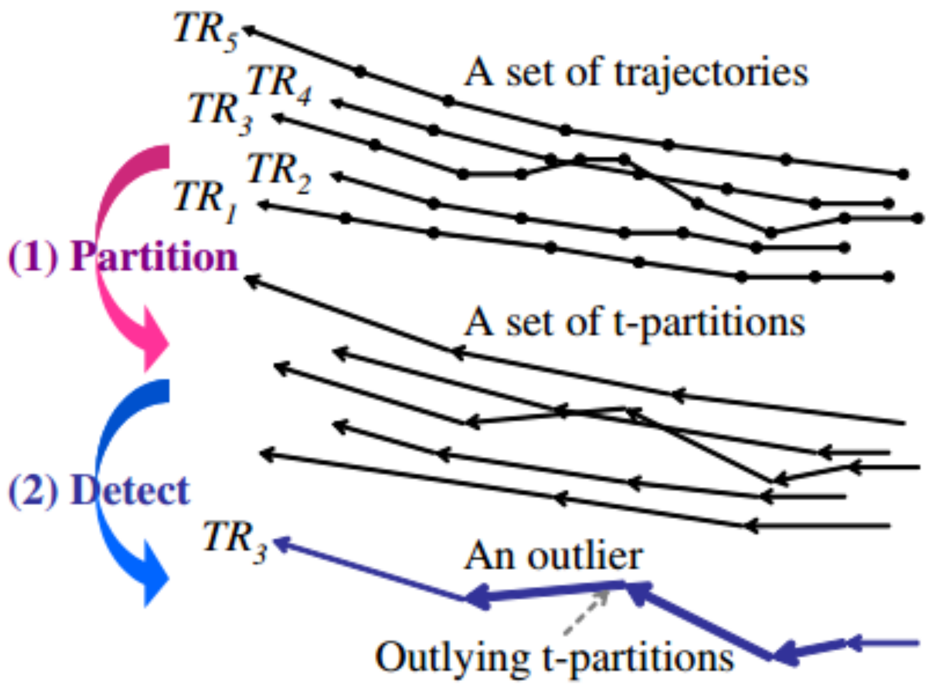
- **TRACLASS** (Trajectory Classification) works in three phases:
  1. Partitions trajectories based on their shapes (using a TRACLUS variant)
  2. Discovers regions that contain sub-trajectories mostly from one class (hierarchical region-based clustering)
     - Tradeoff between homogeneity and conciseness
  3. Discovers common movement patterns for each class of sub-trajectories (trajectory-based clustering)
- Example:



TRAOD

- **TRAOD** (Trajectory Outlier Detection) works in two phases:
  - Partitioning phase: trajectories are segmented into t-partitions (sub-trajectories); recall TRACLUS
  - Detection phase: a trajectory is considered outlier if it contains a sufficient number of outlying t-partitions

- Example:

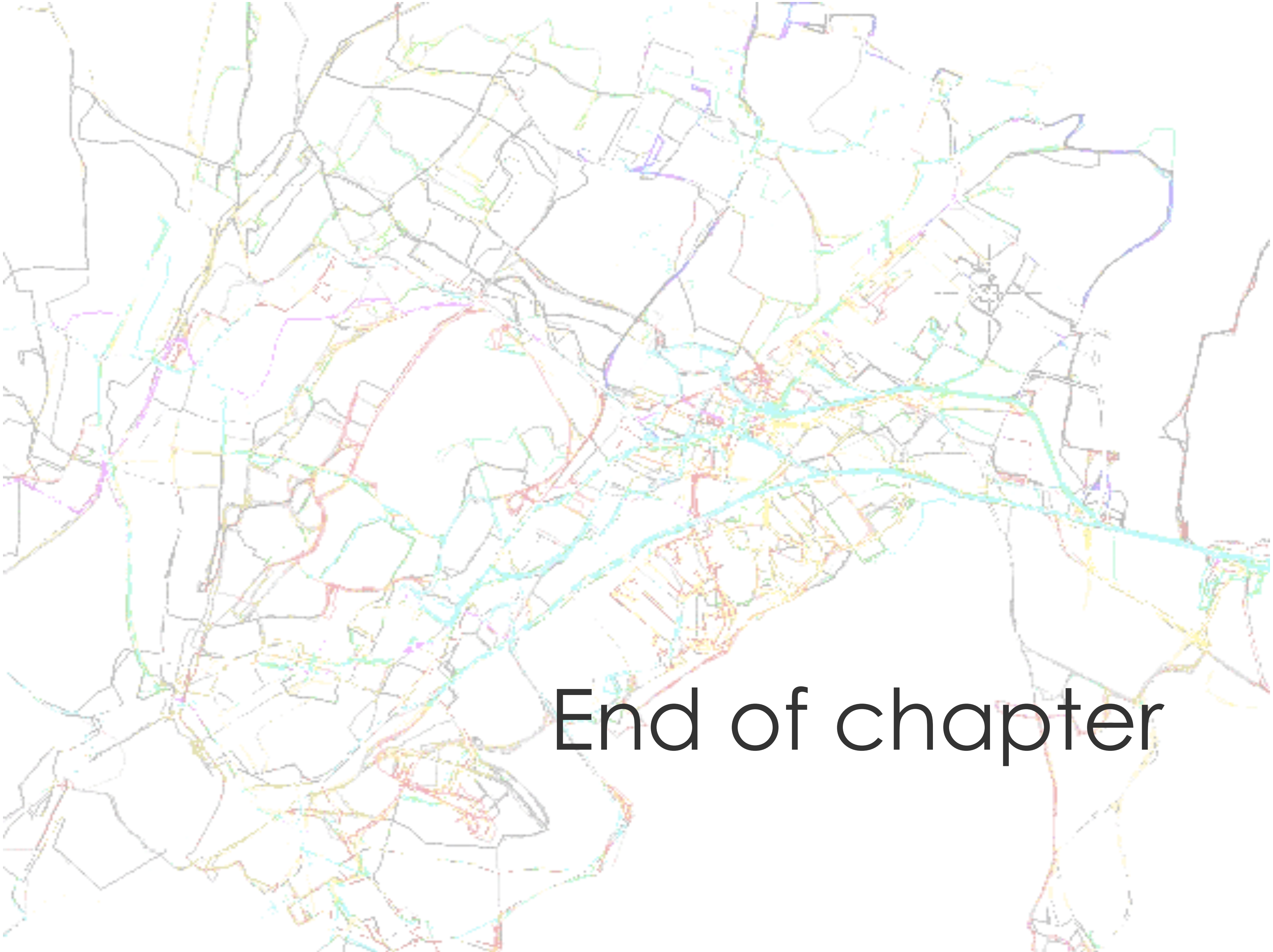


7.5.

Summary

Summarizing …

- Knowledge discovery in trajectory databases discovers behavioral patterns of moving objects
- In this chapter, we presented state-of-the-art techniques for:
  - Clustering a set of trajectories
  - Discovering collective behaviors (flocks, moving clusters, etc.)
  - Predicting the future location of moving objects
  - Classification and outlier detection



End of chapter