Exploring Mobility Datasets

Yannis Theodoridis & Nikos Pelekis

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

Apr. 2013

From bulks of location data to useful trajectory aggregations and patterns

part I: OLAP analysis

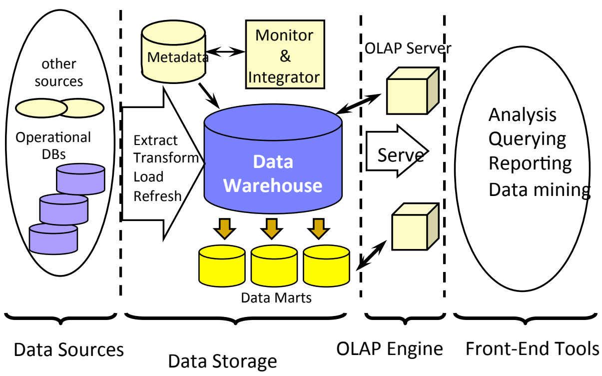
Key questions that arise

- What kind of **analysis** is suitable for mobility data?
  - In particular, trajectories of moving objects?
  - How does infrastructure (e.g. road network) affect this analysis?
- Which patterns / models can be extracted out of them?
  - Clusters, frequent patterns, anomalies / outliers, etc.
  - How to compute such patterns / models efficiently?
- Can we aid analysis by visual artifacts?
  - How should we visualize the mined patterns/models?

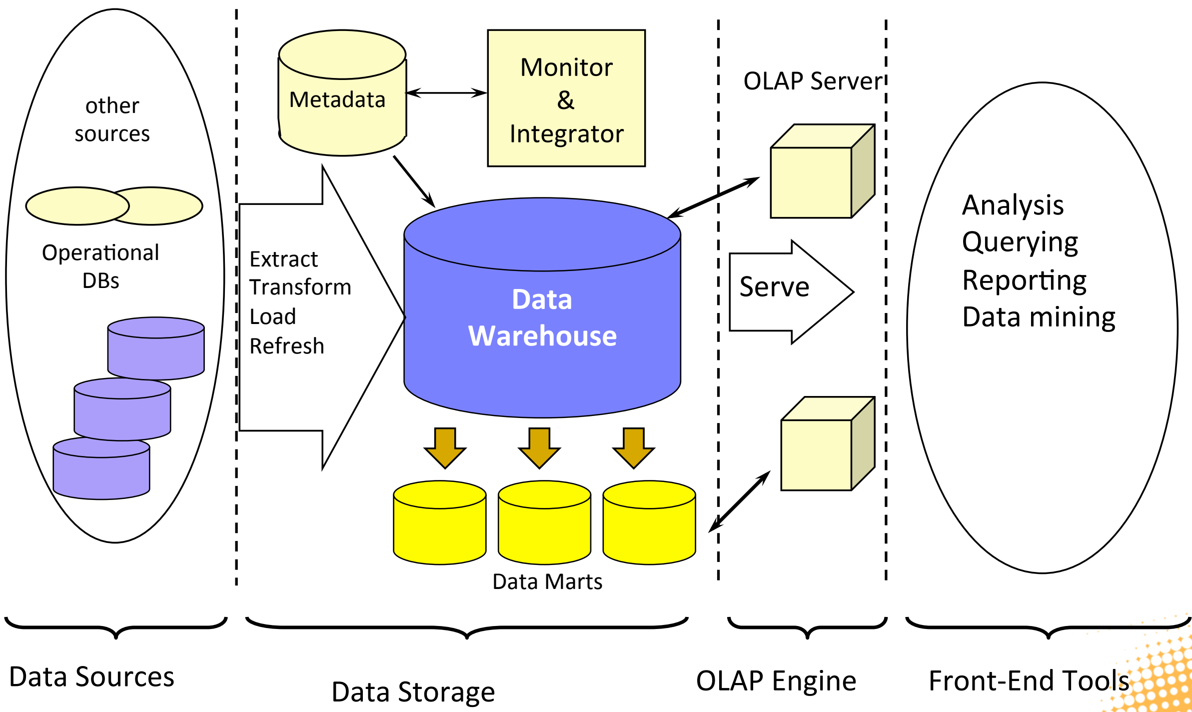
Data warehousing (DW)

- Widely investigated for conventional, non-spatial data.
- A widely accepted definition:
  - A Data Warehouse (DW) is a subject-oriented, integrated, time-variable, non-volatile information system aiming at decision making.

    B. Inmon (1992) *Building the Data Warehouse*. 1st Edition. Wiley and Sons.

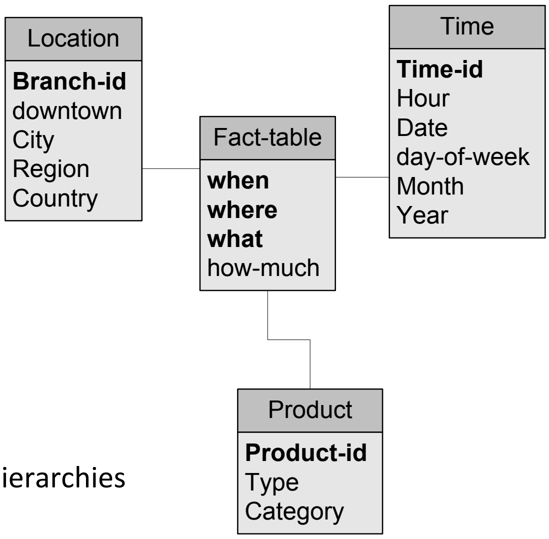


DW architecture



Aggregating DB information: Data Cubes

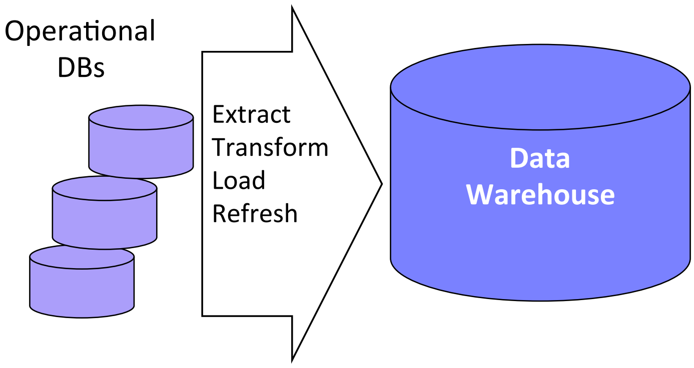
- Aggregated information from DBs is stored in **data cubes** [Gray et al. DMKD '97]
  - Feeded from DB via an Extract-Transform-Load (ETL) procedure
  - Technically, a collection of relations (if relational model is adopted)
- Typical structure: **star schema**
  - Several **dimension tables** with their hierarchies
  - One **fact table** with **measures**
  - Variation: constellation schema (more than one fact tables)



ETL example

- DB schema

```
product (product_ID,
   type, category)
location (branch_ID,
   downtown, city,
   region, country)
sales-transaction (
   timestamp, product_ID,
   branch_ID, units_sold,
   unit_price)
```

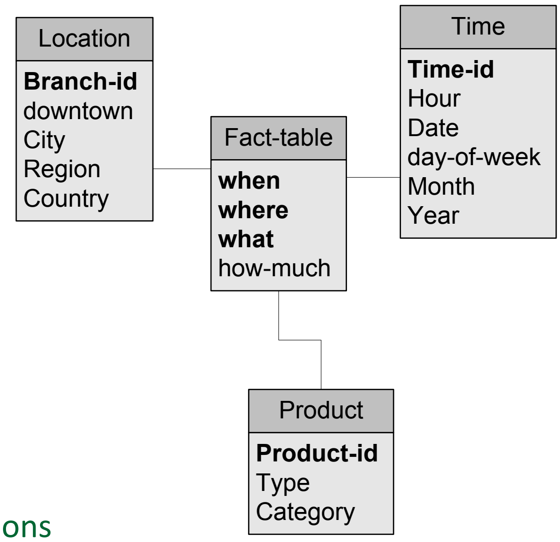


- ETL query

```
INSERT INTO sales
   ( SELECT datetime(timestamp) AS when,
            branch_ID AS where, product_ID AS what,
     sum(units_sold*unit_price) AS how-much
     FROM sales-transaction
     GROUP BY when, where, what
     HAVING how-much > 0 )
```

OLAP operations on data cubes

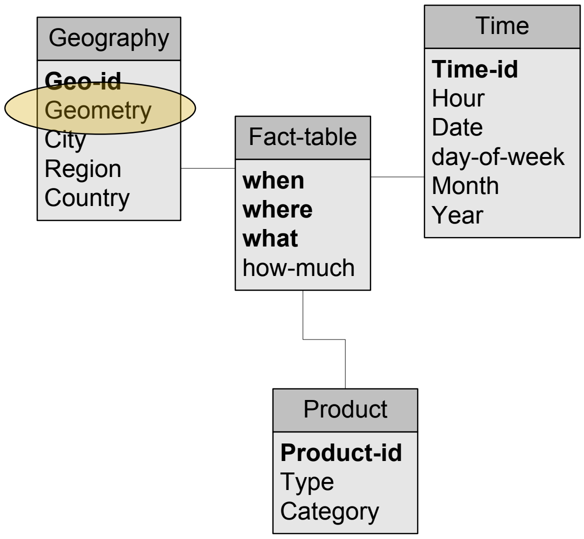
- A sequence of operations:
  - (**roll-up**) “What was the total turnover (“how-much” measure) per month and per city?”
  - (**slice**) “Especially in March, what was the turnover per city?”
  - (**drill-down**) “Especially in March, what was the turnover on weekdays vs. weekends?”
  - (**cross-over**) “Display the DB records that support the above result.”



- Degree of efficiency of OLAP operations depends on the type of measures
  - distributive vs. algebraic vs. holistic

Data cubes for spatial data

- **Spatial data cubes** [Han et al. PAKDD’98]
  - Dimensions
    - Spatial (e.g. Geography) vs.
    - non-spatial /thematic (e.g. Time, Product)
  - Measures:
    - Numerical vs. Spatial



Data cubes for trajectory data

Space
Space-partition-id
Geometry
District
City
State
Country

Fact-table
Interval-id
Space-partition-id
Object-profile-id
count-trajectories
count-objects
avg-distance-traveled
avg-travel-duration
abg-speed
avg-abs-acceleration

Time
Interval-id
Interval
Hour
Date
Month
Quarter
Year
day-of-week
rush-hour

Object-Profile
Object-profile-id
Gender
Birthyear
Profession
Marital-status
Device-type

[Orlando et al. JCSE'07]
[Marketos et al. MobiDE'08]

- **3 Dimensions**:
  - Space, Time, Object-Profile (demographics, technographics)
- **Several (numeric) measures**:
  - Distinct count of trajectories, users, …
  - Average value of distance traveled, travel duration, speed, abs (acceleration), …

Issues that arise

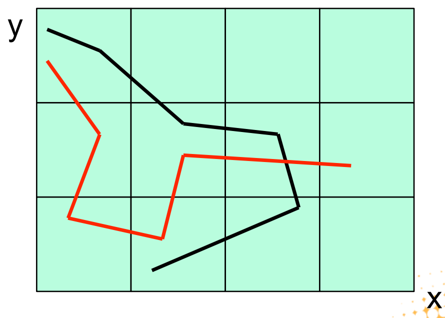
- During ETL:
  - how to **efficiently** feed the fact table?
    - Aggregations over the MOD
- During OLAP:
  - how to address the "**distinct count problem**"?
    - the same trajectory may pass multiple times from the same cell

ETL processing: loading

- Loading data into the dimension tables ➔ straightforward
  - Of course, choosing a reasonable **resolution** in space/time is critical
  - (as usual) tradeoff between quality and usage of resources

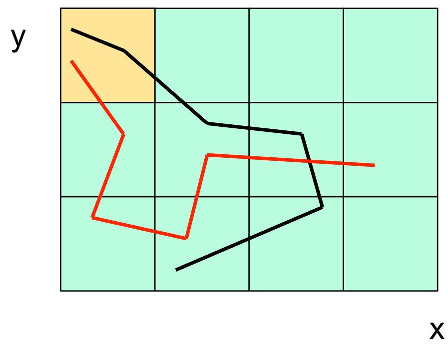
ETL processing: loading

- Loading data into the fact table ➔ complex, expensive
  - Fill in the measures with the appropriate numeric values
  - In order to calculate the measures, we have to extract the portions of the trajectories that fit into the base cells of the cube
    - alternative solutions:
      - cell-oriented
      - trajectory-oriented



ETL processing: algorithms

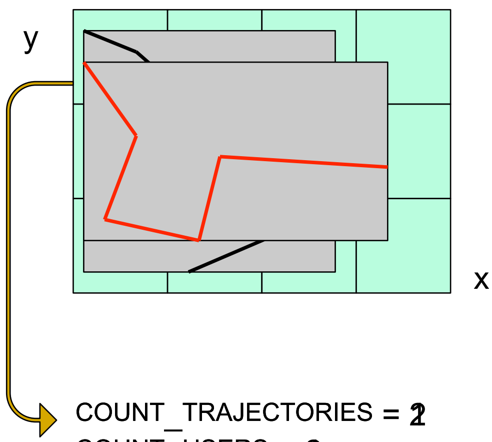
- **Cell-oriented approach (COA)**
  - Search for the portions of trajectories that reside inside a s/t cell
    - **spatiotemporal range query**
    - efficiently supported by the **TB-tree** [Pfoser et al. 2000]
  - Decompose the trajectory portions with respect to the user profiles they belong to
  - Compute measures for this cell
  - Repeat for the next cells



COUNT_TRAJECTORIES = 2
COUNT_USERS = 2
…

ETL processing: algorithms

- **Trajectory-oriented approach (TOA)**
  - Discover the s/t cells where each trajectory resides in
    - Prune by using the trajectory MBR
  - Compute measures for each cell
  - Repeat for the next trajectories



COUNT_TRAJECTORIES = 2
COUNT_USERS = 2
…

OLAP (aggregation in space/time)

- The problem:
  - A trajectory may contribute to several cells
  - What happens when rolling-up?
- The “**distinct count problem**” (Tao et al. 2004)

The distinct count problem

At the lowest hierarchy level:

count of trajectories in $R_{1,1}$ = 2

count of trajectories in $R_{1,2}$ = 2

count of trajectories in $R_{2,1}$ = 1

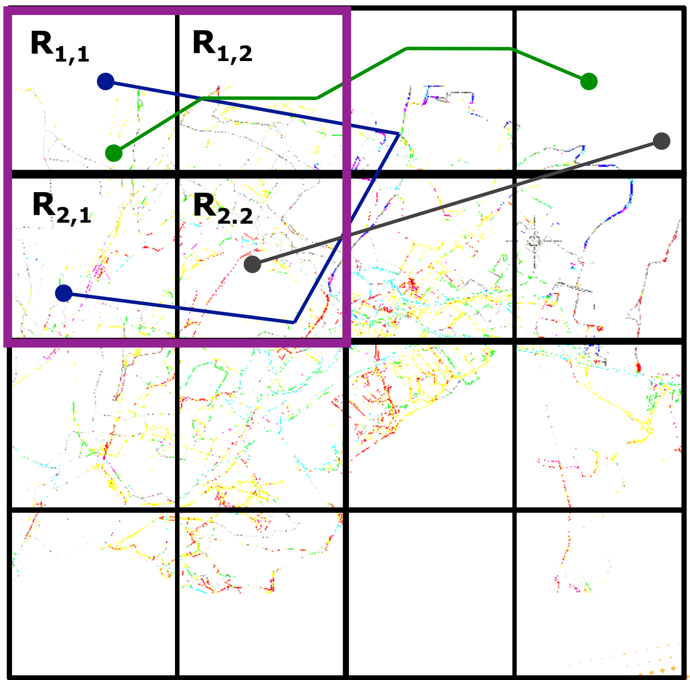
count of trajectories in $R_{2,2}$ = 2

Roll up in R =
$R_{1,1} \cup R_{1,2} \cup R_{2,1} \cup R_{2,2}$

count of trajectories in R = 7
(according to traditional roll-up)

whereas the correct is 3 !!

Any idea how to estimate the correct answer?



The distinct count problem

At the lowest hierarchy level:

count of trajectories in $R_{1,1}$ = 2

count of trajectories in $R_{1,2}$ = 2

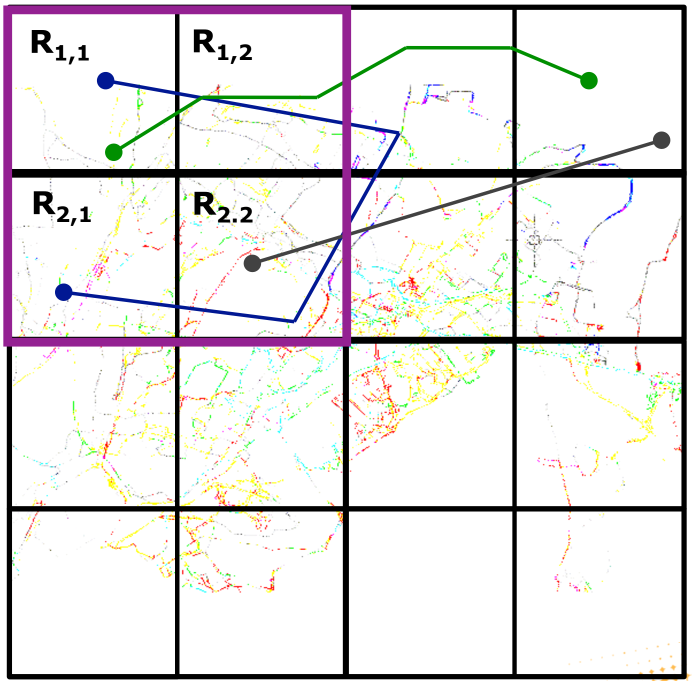
count of trajectories in $R_{2,1}$ = 1

count of trajectories in $R_{2,2}$ = 2

Roll up in R =
$R_{1,1} \cup R_{1,2} \cup R_{2,1} \cup R_{2,2}$

count of trajectories in R = 7
(according to traditional roll-up)

whereas the correct is 3 !!



A (suboptimal) solution:
(Orlando et al. 2007a; 2007b)

"Keep a note on the border between cells"

Case study

Observe and analyze traffic flow during a week in Milano

U. Venice & U. Piraeus,
GeoPKDD final meeting, Pisa, May 2009

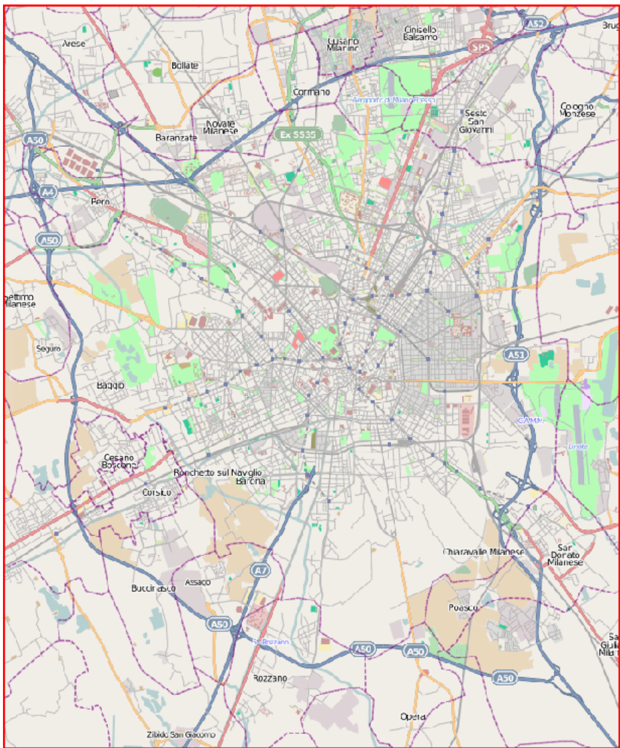
T-Warehouse tool (Leonardi et al. 2010)

Typical kinds of analysis (from end-users' point of view)

- How does traffic flow and speed change along the week?
  - Q1: Where does the highest traffic appear? at what hour?
  - **A1: unclassified choropleth map (for a specific period of time)**
  - Q2: What exactly happens at the road network level?
  - **A2: drill-downs in space and/or time**
  - Q3: How does movement propagate from place to place?
  - **A3: data cube measures' correlation (speed vs. presence)**

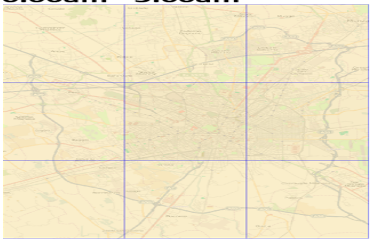
Milano dataset

- What?
  - 2M observations (GPS recordings)
    - for 7 days (Sun. 1 - Sat. 7 April '07)
  - 200K trajectories (after reconstruction)
- How?
  - Stored in Hermes MOD engine
  - Feeding a trajectory data cube

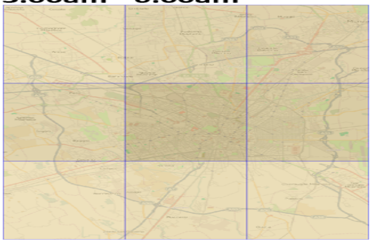


Presence on Tuesday
(aggregated level)

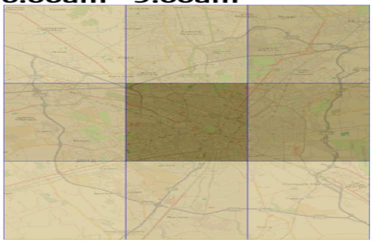
0.00am - 3.00am



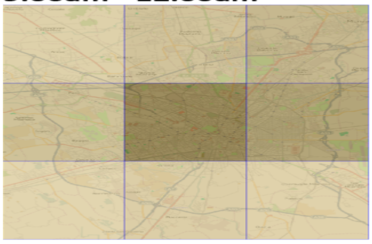
3.00am - 6.00am



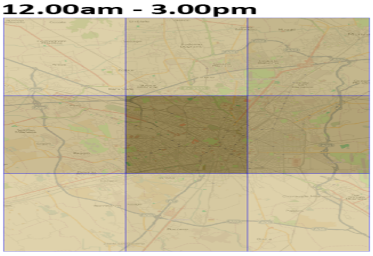
6.00am - 9.00am



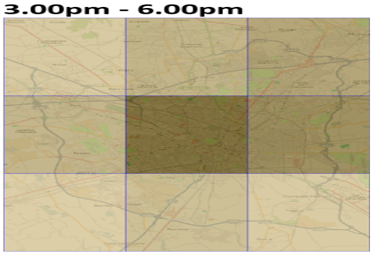
9.00am - 12.00am



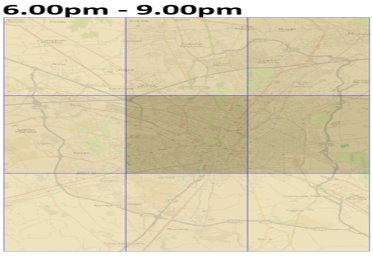
12.00am - 3.00pm



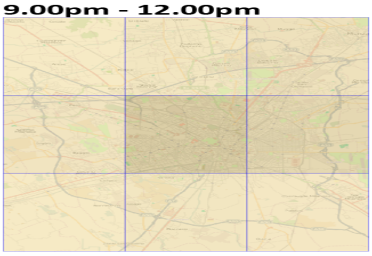
3.00pm - 6.00pm



6.00pm - 9.00pm

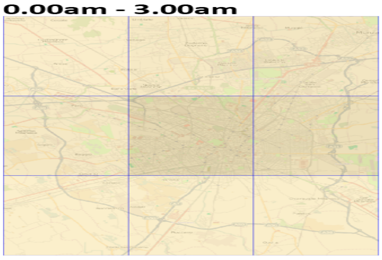


9.00pm - 12.00pm

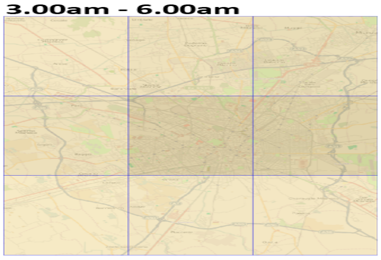


Presence on Saturday
(aggregated level)

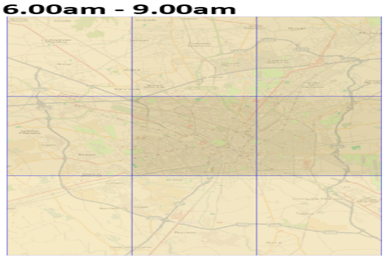
0.00am - 3.00am



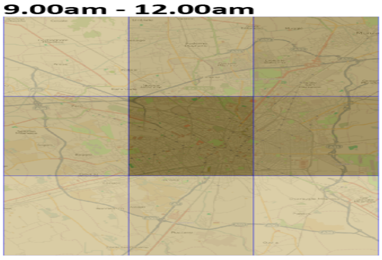
3.00am - 6.00am



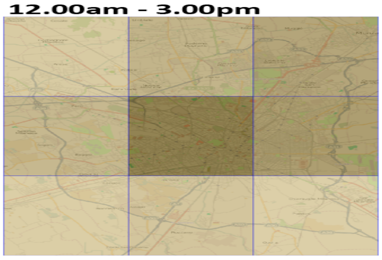
6.00am - 9.00am



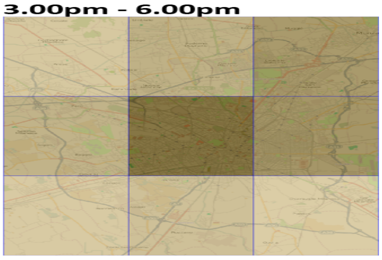
9.00am - 12.00am



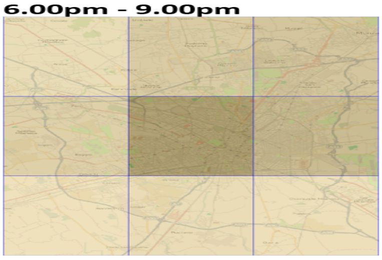
12.00am - 3.00pm



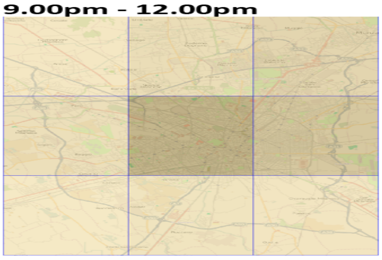
3.00pm - 6.00pm



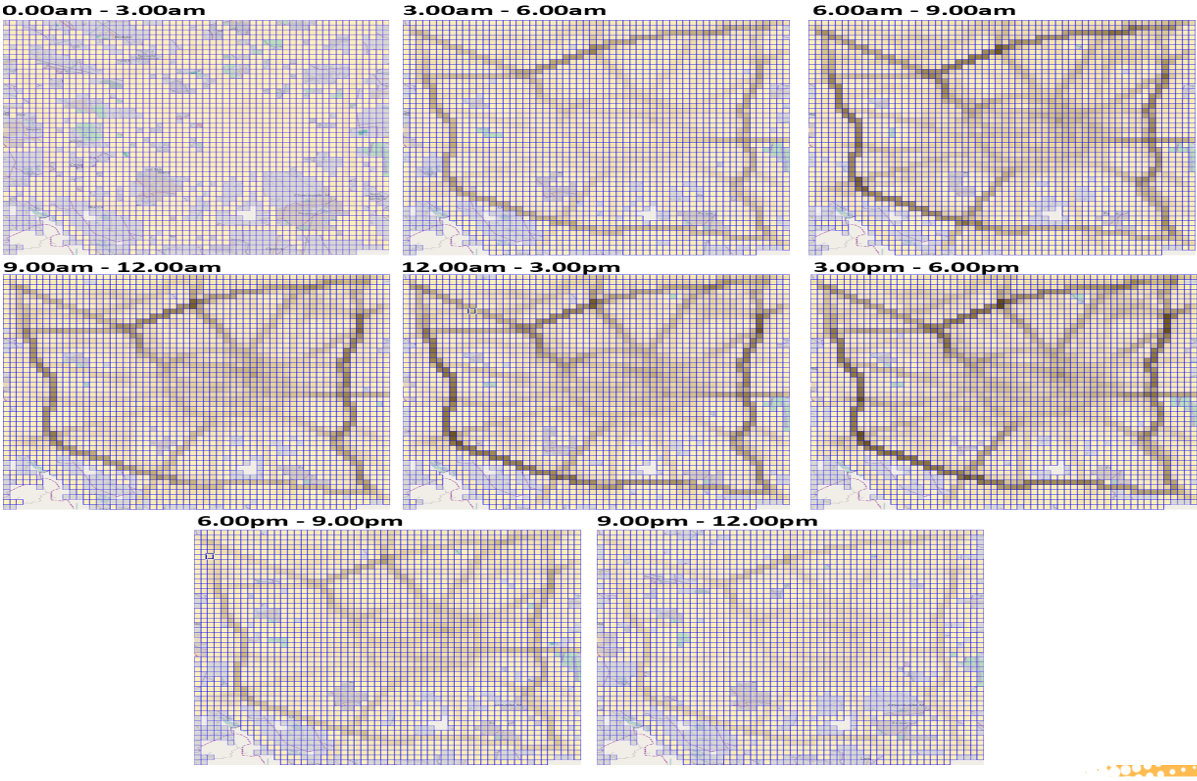
6.00pm - 9.00pm



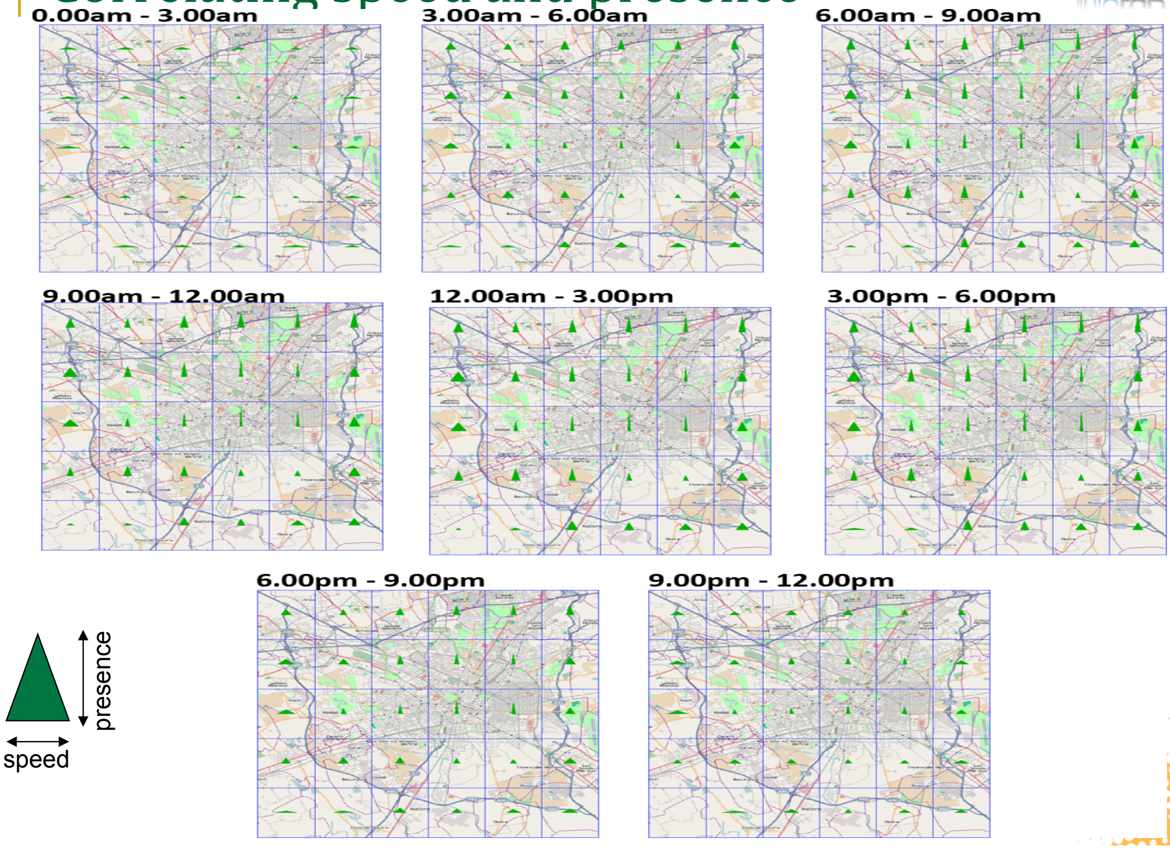
9.00pm - 12.00pm



Presence on Tuesday
(road network level)



Correlating speed and presence



Conclusions on Part I

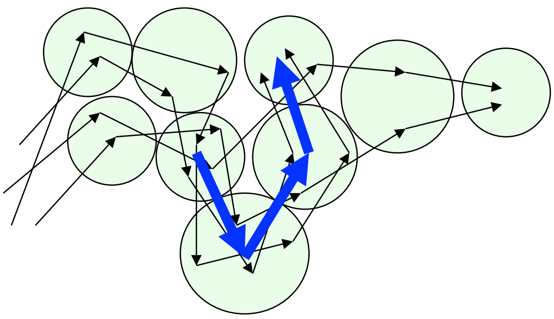
- (Explorative) OLAP analysis over mobility data is a key tool for urban planning, etc.

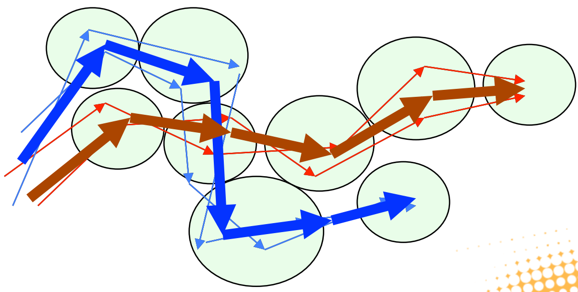
- Research challenges
  - Take network constraints into consideration
    - e.g. grid vs. graph (road network) partitioning at the Space dimension
  - Support “semantic trajectories” → semantic trajectory warehouses

part II: KDD

KDD process over mobility data

- Knowledge discovery from mobility data
  - *"the opportunity to discover, from the **digital traces** of human activity, the **knowledge** that makes us comprehend timely and precisely the way we live, the way we use our time and our land"*
    [Giannotti & Pedreschi, 2008] [Giannotti et al. 2008]



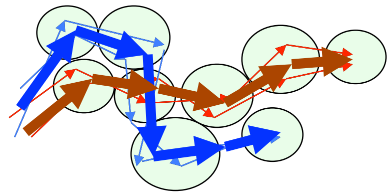


Key questions that arise

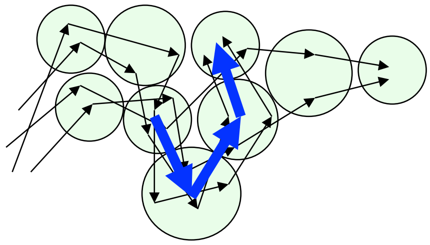
- What kind of analysis is suitable for mobility data?
  - In particular, trajectories of moving objects?
  - How does infrastructure (e.g. road network) affect this analysis?
- Which **patterns / models** can be extracted out of them?
  - Clusters, frequent patterns, anomalies / outliers, etc.
  - How to compute such patterns / models efficiently?
- Can we aid analysis by visual artifacts?
  - How should we visualize the mined patterns/models?

Examples of mobility data mining

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



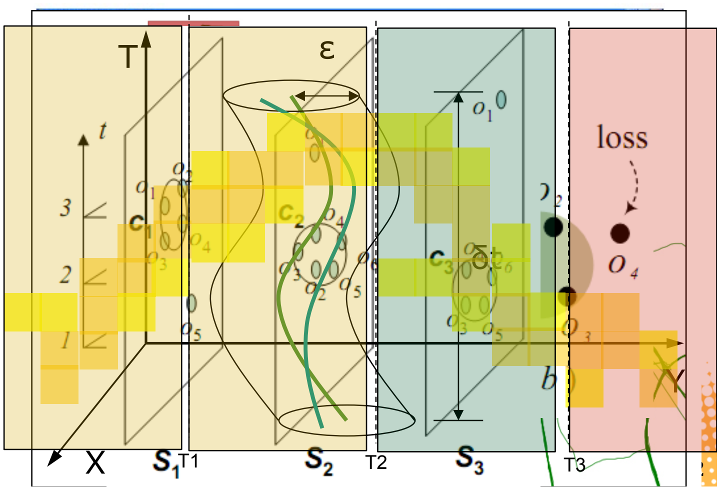
- **Frequent pattern mining**
  - Identify 'frequent' or 'popular' patterns
  - Discover hot spots, hot paths, etc.



- **Trajectory classification**
  - Assign trajectories to predefined classes
  - Find rules that may predict future behavior of moving objects
- **Trajectory sampling**
  - Out of the full population, select some representatives

Applications of mobility data mining

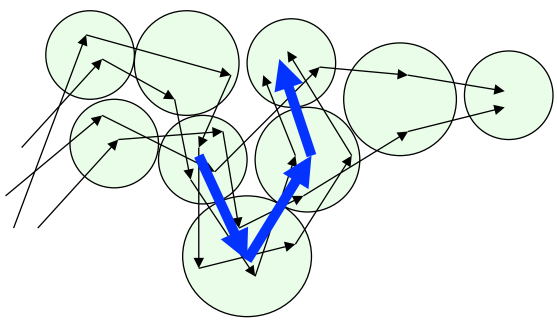
- Exploiting on "mobility patterns"
  - **Hot-spots** (popular places) [Giannotti et al. 2007]
  - **T-Patterns** [Giannotti et al. 2007]
  - **Hot motion paths** [Sacharidis et al. 2008]
  - **Typical trajectories** [Lee et al. 2007]
  - **Moving clusters** [Kalnis et al. 2005]
  - **Flocks & Leaders** [Benkert et al. 2008]
  - **Convoys** [Jeung et al. 2008]
  - **Centroid trajectories** [Pelekis et al. 2009-10]



Frequent pattern mining

“Frequent pattern mining” techniques

- Technical objectives:
  - Identify ‘frequent’ or ‘popular’ patterns
  - Discover hot spots, hot paths, etc.
- Related work:
  - Hot-spots (popular places) [Giannotti et al. 2007]
  - T-Patterns [Giannotti et al. 2007]
  - Hot motion paths [Sacharidis et al. 2008]



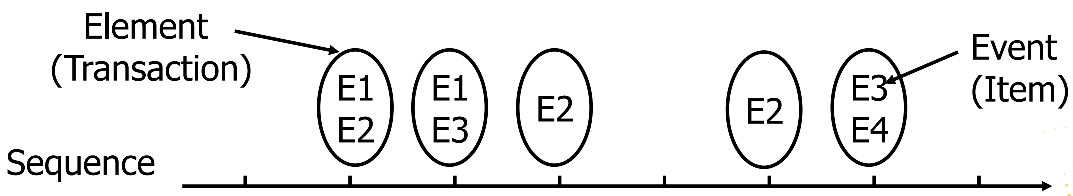
A general definition

- The settings:
  - A dataset of entities $D = \{e_1, e_2, \ldots, e_N\}$
  - Each entity consists of a (temporal) sequence $e_i = <e_{i1}, \ldots, e_{iM}>$ where e_{ij} belongs to a set of items $I = \{I_1, \ldots, I_K\}$
- The objective goal:
  - Find sequences of items $<\ldots, I_i, I_j, \ldots>$ which appear in this order frequently (i.e., at least d times) in the dataset. Such a sequence is called a **frequent pattern** in D

Examples of Sequence Data

original slide from (Tan et al. 2004)

Sequence Database	Sequence	Element (Transaction)	Event (Item)
Customer	Purchase history of a given customer	A set of items bought by a customer at time t	Books, diary products, CDs, etc
Web Data	Browsing activity of a particular Web visitor	A collection of files viewed by a Web visitor after a single mouse click	Home page, index page, contact info, etc
Event data	History of events generated by a given sensor	Events triggered by a sensor at time t	Types of alarms generated by sensors
Genome sequences	DNA sequence of a particular species	An element of the DNA sequence	Bases A,T,G,C



Sequential Pattern Mining: Example

original slide from (Tan et al. 2004)

Object	Timestamp	Events
A	1	1,2,4
A	2	2,3
A	3	5
B	1	1,2
B	2	2,3,4
C	1	1, 2
C	2	2,3,4
C	3	2,4,5
D	1	2
D	2	3, 4
D	3	4, 5
E	1	1, 3
E	2	2, 4, 5

Minsup = 50%

Examples of Frequent Sub-sequences:

< {1,2} >	s=60%
< {2,3} >	s=60%
< {2,4}>	s=80%
< {3} {5}>	s=80%
< {1} {2} >	s=80%
< {2} {2} >	s=60%
< {1} {2,3} >	s=60%
< {2} {2,3} >	s=60%
< {1,2} {2,3} >	s=60%

Sequential Pattern Mining: Challenge

original slide from (Tan et al. 2004)

- Given a sequence: `<{a b} {c d e} {f} {g h i}>`
  - Examples of subsequences:

    `<{a} {c d} {f} {g} >`, `< {c d e} >`, `< {b} {g} >`, etc.

- How many k-subsequences can be extracted from a given n-sequence?

`<{a b} {c d e} {f} {g h i}>   n = 9`

`k=4:   Y _ _ Y Y _ _ _ Y`

`<{a}   {d e}   {i}>`

Answer :

$$\binom{n}{k} = \binom{9}{4} = 126$$

Extracting Sequential Patterns

original slide from (Tan et al. 2004)

- Given n events: $i_1, i_2, i_3, \ldots, i_n$
- Candidate 1-subsequences:

  $<\{i_1\}>, <\{i_2\}>, <\{i_3\}>, \ldots, <\{i_n\}>$

- Candidate 2-subsequences:

  $<\{i_1, i_2\}>, <\{i_1, i_3\}>, \ldots, <\{i_1\} \{i_1\}>, <\{i_1\} \{i_2\}>, \ldots, <\{i_{n-1}\} \{i_n\}>$

- Candidate 3-subsequences:

  $<\{i_1, i_2, i_3\}>, <\{i_1, i_2, i_4\}>, \ldots, <\{i_1, i_2\} \{i_1\}>, <\{i_1, i_2\} \{i_2\}>, \ldots,$

  $<\{i_1\} \{i_1, i_2\}>, <\{i_1\} \{i_1, i_3\}>, \ldots, <\{i_1\} \{i_1\} \{i_1\}>, <\{i_1\} \{i_1\} \{i_2\}>, \ldots$

- … by appropriately pruning at each step! (**A-priori style of thinking**)

What is the "A-priori style of thinking"?

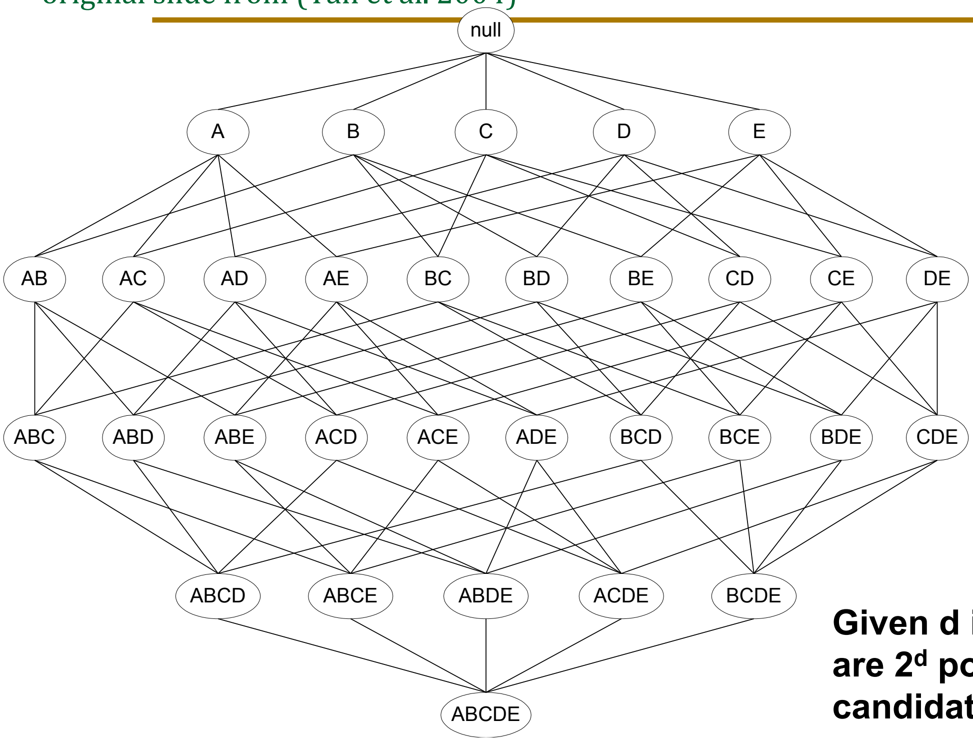
original slide from (Tan et al. 2004)

- **Itemset**: A collection of one or more items
  - Example: {Milk, Bread, Diaper}
- **k-itemset**: An itemset that contains k items
- **Support**: Fraction of transactions that contain an itemset
  - e.g. s({Milk, Bread, Diaper}) = 2/5
- **Frequent Itemset**:
  - An itemset whose support is greater than or equal to a minsup threshold
- **Frequent Itemset Generation**
  - Generate all itemsets whose support ≥ minsup
  - Computationally expensive!

TID	*Items*
1	Bread, Milk
2	Bread, Diaper, Beer, Eggs
3	Milk, Diaper, Beer, Coke
4	Bread, Milk, Diaper, Beer
5	Bread, Milk, Diaper, Coke

Frequent Itemset Generation

original slide from (Tan et al. 2004)



Given d items, there are 2^d possible candidate itemsets

Reducing Number of Candidates

original slide from (Tan et al. 2004)

- Apriori principle:
  - **If an itemset is frequent, then all of its subsets must also be frequent**

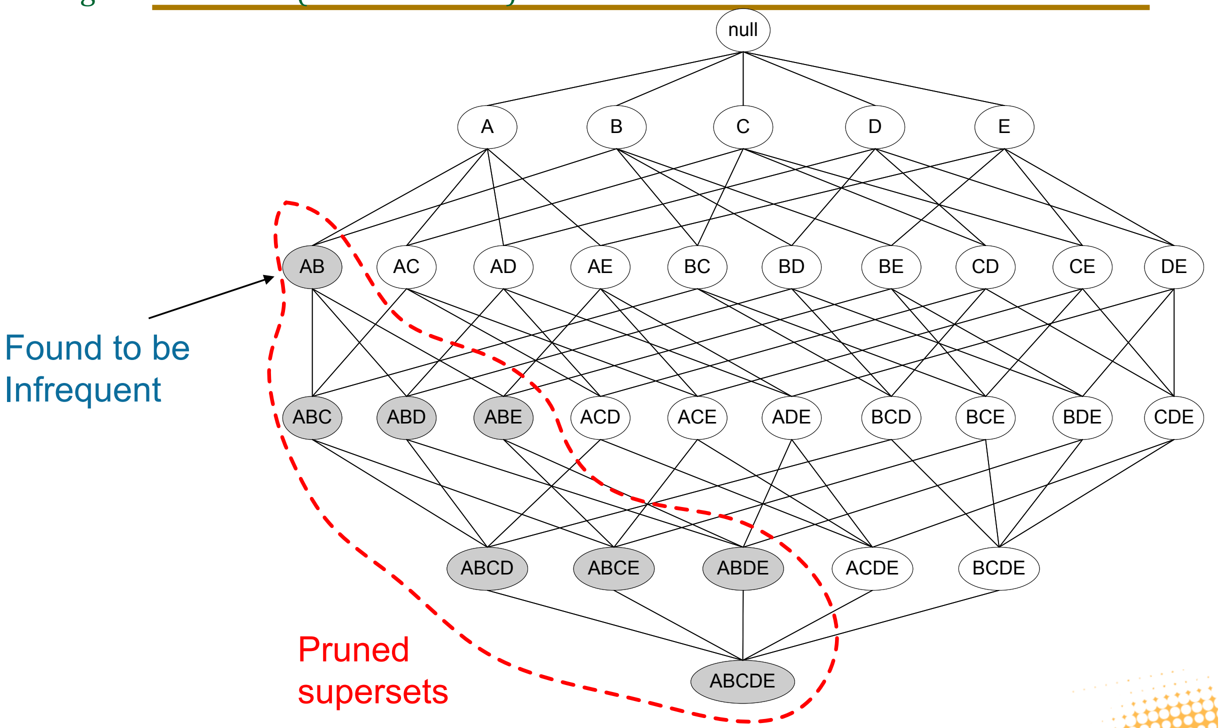
- Apriori principle holds due to the following property of the support measure:

$$\forall X, Y : (X \subseteq Y) \Rightarrow s(X) \geq s(Y)$$

  - Support of an itemset never exceeds the support of its subsets
  - This is known as the anti-monotone property of support

Illustrating Apriori Principle

original slide from (Tan et al. 2004)



Illustrating Apriori Principle

original slide from (Tan et al. 2004)

Item	Count
Bread	**4**
Coke	**2**
Milk	**4**
Beer	**3**
Diaper	**4**
Eggs	**1**

Items (1-itemsets)

Itemset	Count
{Bread,Milk}	**3**
{Bread,Beer}	**2**
{Bread,Diaper}	**3**
{Milk,Beer}	**2**
{Milk,Diaper}	**3**
{Beer,Diaper}	**3**

Pairs (2-itemsets)

(No need to generate candidates involving Coke or Eggs)

Minimum Support = 3

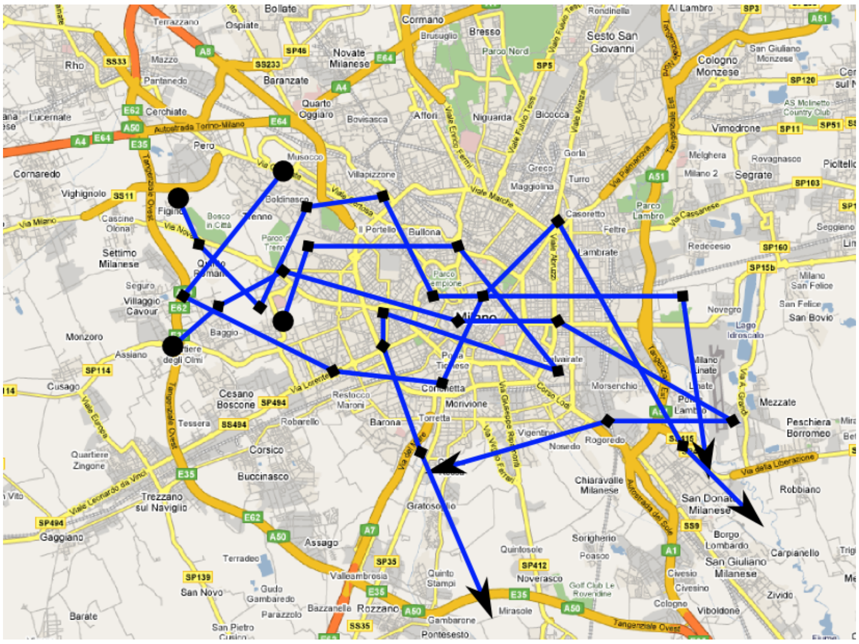
Triplets (3-itemsets)

Itemset	Count
{Bread,Milk,Diaper}	**3**

If every subset is considered,
$^6C1 + {}^6C2 + {}^6C3 = 41$
With support-based pruning,
$6 + 6 + 1 = 13$

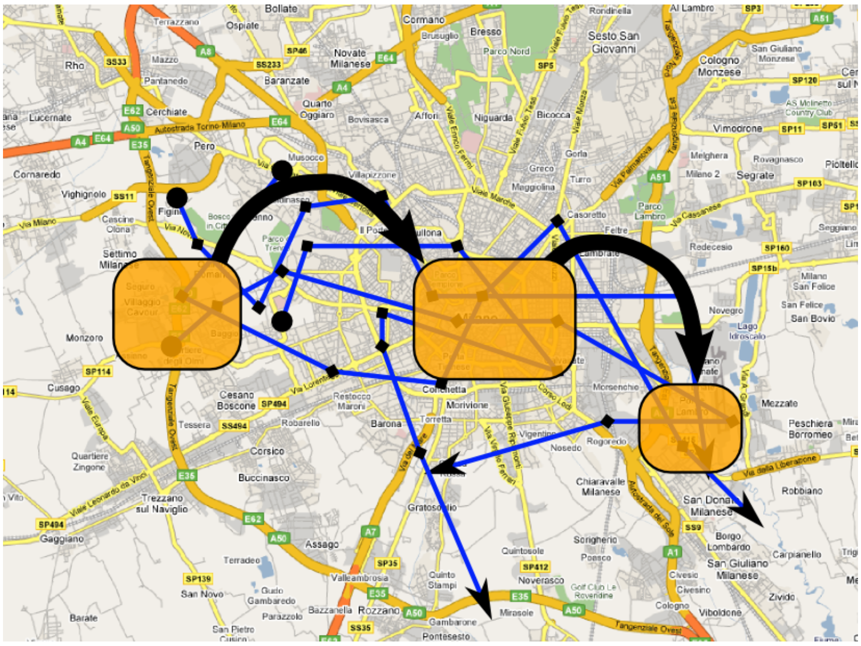
Back to mobility data…

- What is a frequent pattern for trajectories?



T-patterns

- [Giannotti et al. 2007] **T-pattern** is a sequence of visited regions, **frequently** visited in the **specified order** with **similar transition times**



T-Pattern discovery

Input: Trajectory Dataset

Output: T-Patterns

Intermediate result: Regions of Interest

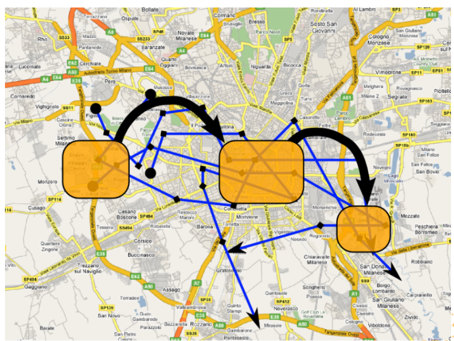
T-Pattern definitions

- A **Trajectory Pattern** (T-pattern) is a pair (**s**, α):
  - **s** = <(x_0,y_0),..., (x_k,y_k)> is a sequence of k+1 point locations
  - α = <α_1,..., α_k> are the respective transition times (*annotations*)

  also written as:

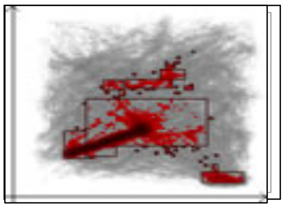
$$(x_0, y_0) \xrightarrow{\alpha_1} (x_1, y_1) \xrightarrow{\alpha_2} \cdots \xrightarrow{\alpha_k} (x_k, y_k)$$

- A T-pattern T_p **occurs** in a trajectory T if T contains a sub-sequence S, such that:
  - *spatial closeness*
    - each point in T_p is **close** to a point in S
  - *temporal closeness*
    - transition times in T_p are **similar** to those in S

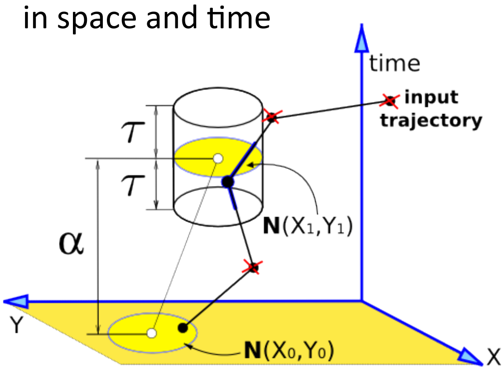


T-Pattern discovery in 3-steps

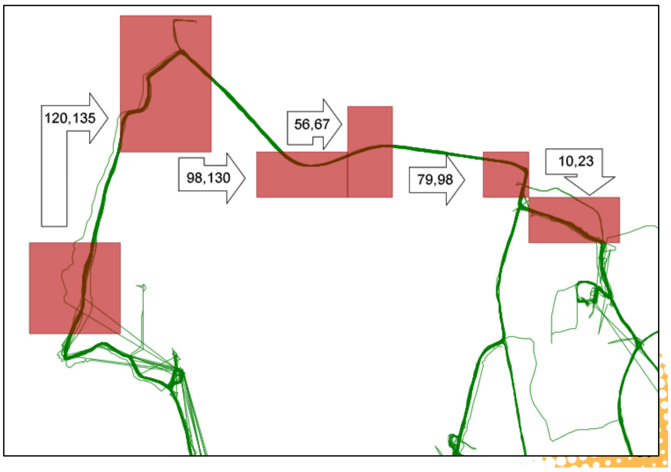
Step 1- Find Regions of Interest



Step 2- Find similar Trajectories in space and time



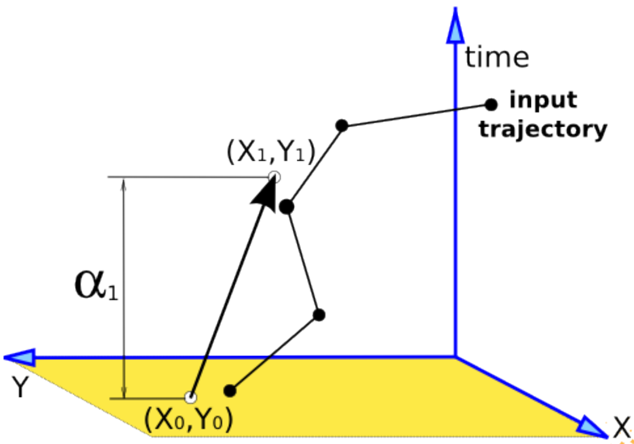
Step 3- Extract patterns with high support



T-Pattern: *approximate* occurrence

- Two points are close to each other if one falls within a **spatial neighborhood N()** of the other
- Two transition times are similar to each other if their **temporal difference is ≤ τ**
- Example:

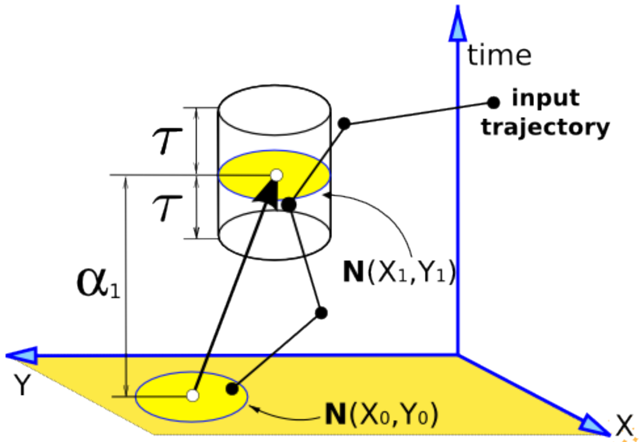
$$(x_0, y_0) \xrightarrow{\alpha_1} (x_1, y_1)$$



T-Pattern: *approximate* occurrence

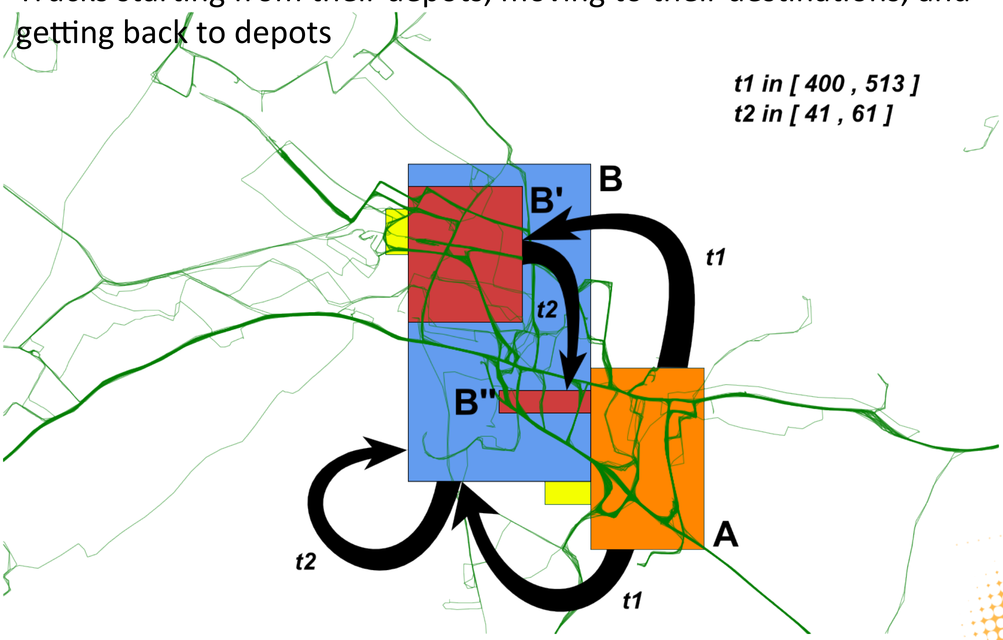
- Two points are close to each other if one falls within a **spatial neighborhood N()** of the other
- Two transition times are similar to each other if their **temporal difference is ≤ τ**
- Example:

$$(x_0, y_0) \xrightarrow{\alpha_1} (x_1, y_1)$$



T-pattern mining on work...

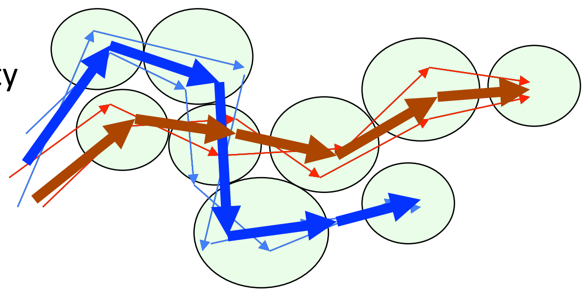
- Athens trucks – 273 trajectories (source: www.rtreeportal.org)
- Trucks starting from their depots, moving to their destinations, and getting back to depots



Trajectory clustering

"Trajectory clustering" techniques

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



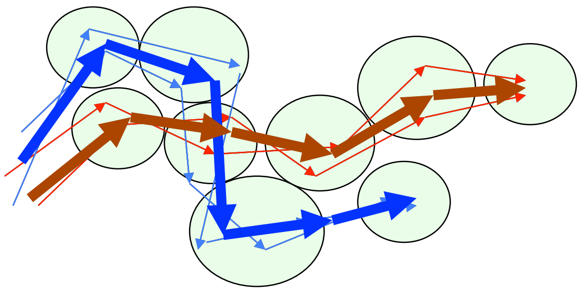
- Related work:
  - Moving clusters [Kalnis et al. 2005]
  - Typical [Lee et al. 2007] vs. Centroid trajectories [Pelekis et al. 2009]
  - Flocks & Leaders [Benkert et al. 2008]; Convoys [Jeung et al. 2008]

A general definition

- The settings:
  - A dataset of entities $D = \{e_1, e_2, \dots, e_N\}$
  - For each pair of entities, a distance $Dist(e_{ij})$ can be measured (hence, a NxN distance matrix is potentially formed)
    - (hopefully) the distance measure $Dist(e_{ij})$ should be a **metric**.
- The objective goal:
  - Partition entities of D into K groups (**clusters**), $G_1, \dots, G_K$ with the following properties:
    - $\cup G_i = D$, $G_i \cap G_j = \emptyset$
    - The intra-cluster (inter-cluster) distance between entities is minimized (maximized, resp.), as better as possible

Back to mobility data…

- Questions:
  - Which distance between trajectories? How do we define intra- and inter-cluster distances?
  - Which kind of clustering?
    - Partitioning (like K-means)? Density-based (like DBSCAN or OPTICS)?
  - How does a cluster 'centroid' look like in our case?
    - A "trajectory" representing the trajectories of a cluster, as better as possible



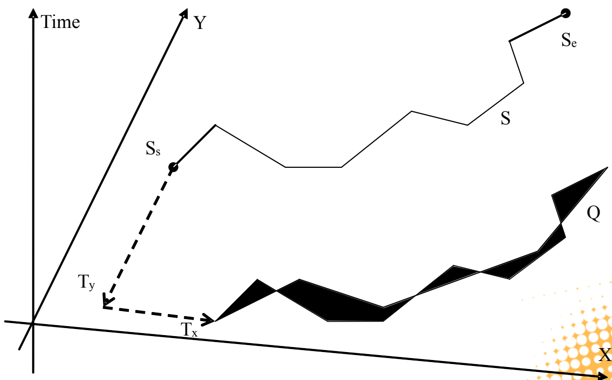
Which distance?

- A possible solution: average Euclidean distance between (sub-) trajectories

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

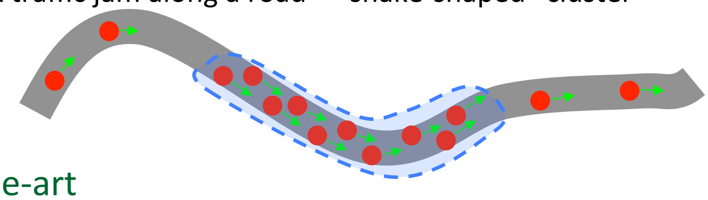
distance between moving objects τ1 and τ2 at time *t*

- "Synchronized" behaviour distance
  - Similar trajectories ➔ in similar places at similar timestamps
- Good news: it is a **metric**
  - Result: efficient indexing, e.g. [Frentzos et al. 2007]



Which kind of clustering?

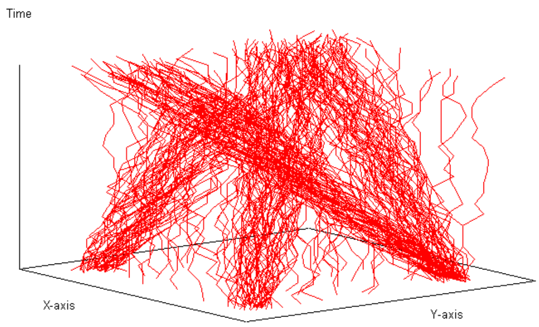
- General requirements:
  - Tolerance to noise; Low computational cost; Applicability to complex, possibly non-vectorial data; Non-spherical clusters; etc.
    - E.g.: A traffic jam along a road = "snake-shaped" cluster

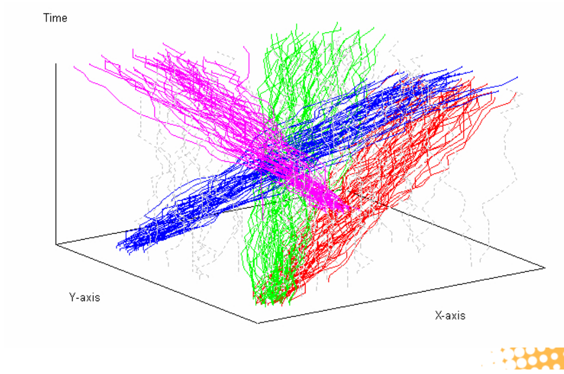


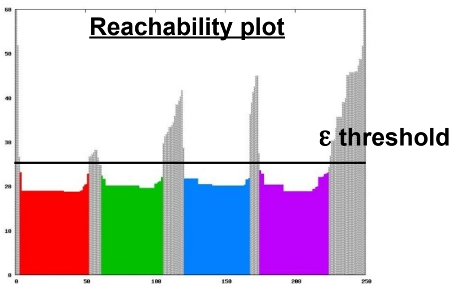
- State-of-the-art
  - Density-based clustering: **T-OPTICS** [Nanni & Pedreschi, 2006]
  - Partition-based clustering: **TRACLUS** [Lee et al. 2007], **CenTR-I-FCM** [Pelekis et al. 2009, 2011]

T-OPTICS

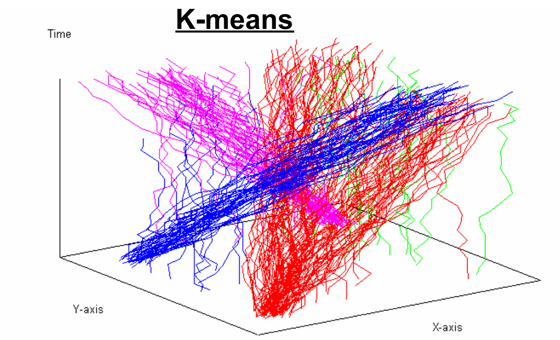
- Builds upon OPTICS
- Keywords:
  - distance, core trajectories, reachability
- Reachability plot (valleys and hills)
  - Valleys → clusters !!

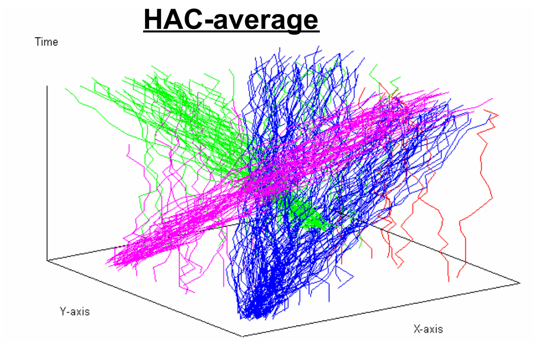


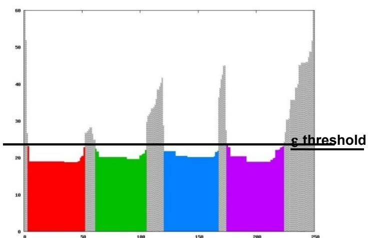




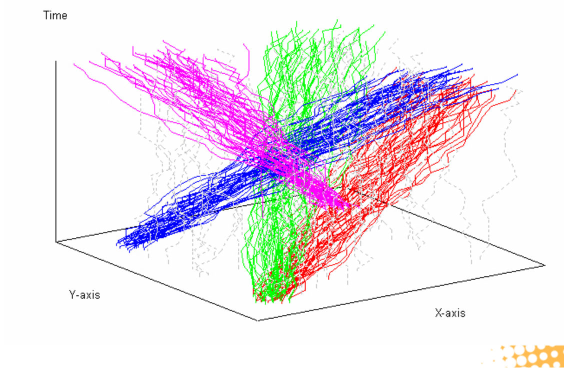
T-OPTICS vs. HAC & K-means





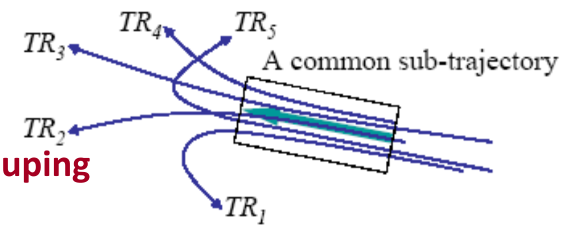


T-OPTICS

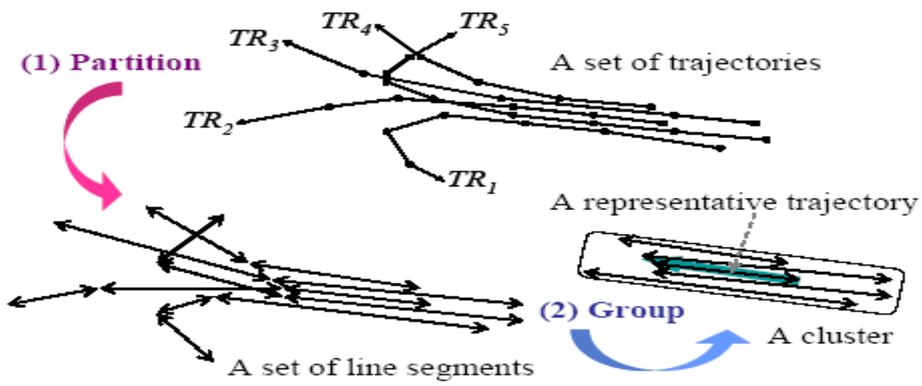


TRACLUS: Partition-and-Group

- Discovers similar portions of trajectories (sub-trajectories)



- Two phases: **partitioning** and **grouping**

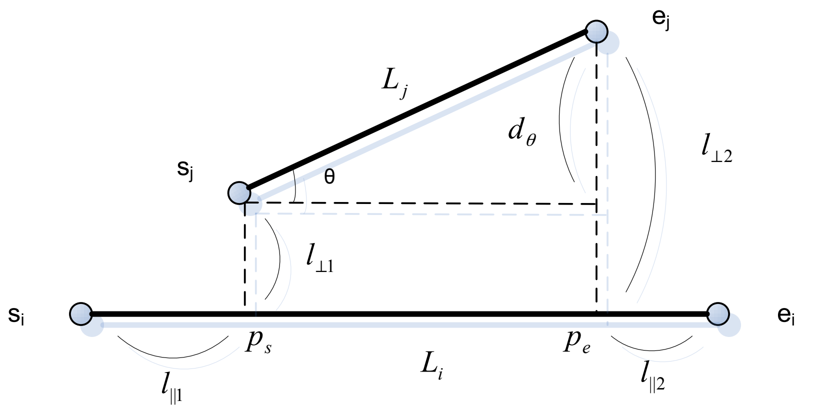


TRACLUS – partitioning phase

- Step 1: Trajectory reconstruction – finding the characteristic points
  - using the Minimum Description Length (MDL) principle.
    - *"the best hypothesis for a given set of data is the one that leads to the best compression of the data"*
- Step 2: Trajectory partitioning into segments

TRACLUS – grouping phase

- Step 3: Trajectory segment grouping
  - using DBSCAN



$$d_\perp = \frac{l_{\perp 1}^2 + l_{\perp 2}^2}{l_{\perp 1} + l_{\perp 2}}$$

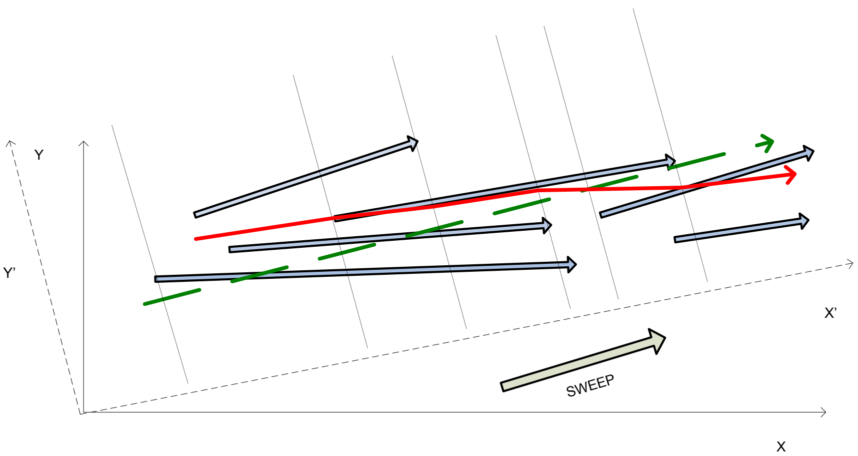
$$d_\| = MIN(l_{\|1} + l_{\|2})$$

$$d_\theta = \| L_j \| \times \sin(\theta)$$

$$dist(L_i, L_j) = w_\perp \times d_\perp(L_i, L_j) + w_\| \times d_\|(L_i, L_j) + w_\theta \times d_\theta(L_i, L_j)$$

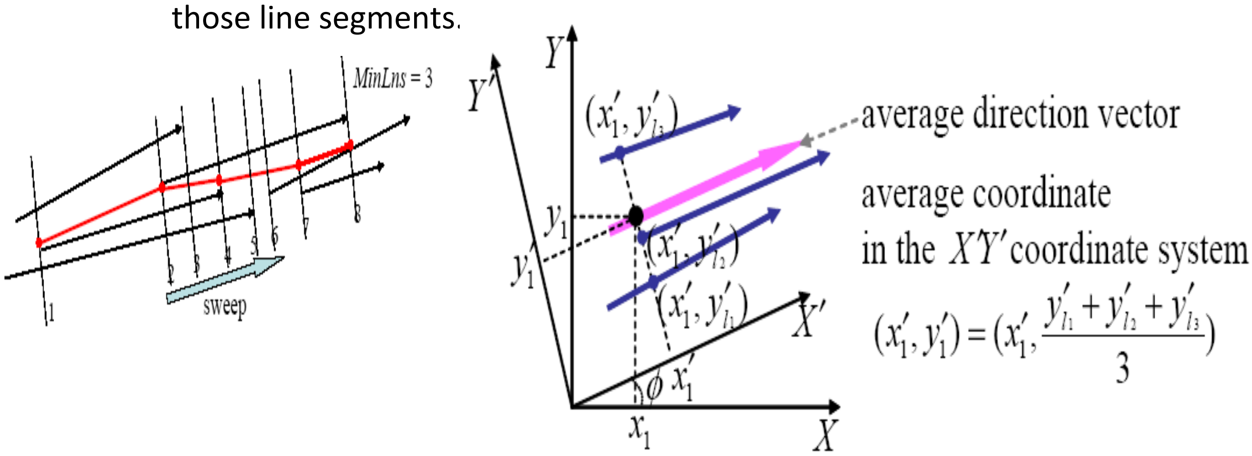
TRACLUS – grouping phase

- Step 4: finding the representative trajectory for each grouping

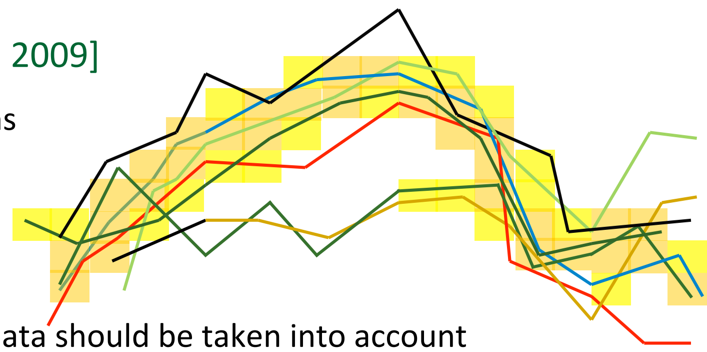


TRACLUS – representative trajectory

- The representative trajectory of the cluster:
  - Compute the average direction vector and rotate axes
  - Sort starting / ending points by the coordinate of the rotated axis
  - While scanning starting / ending points in the sorted order, count the number of line segments and compute the average coordinate of those line segments.



CenTR-I-FCM: Clustering under uncertainty



- CenTR-I-FCM [Pelekis et al. 2009]
  - 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)

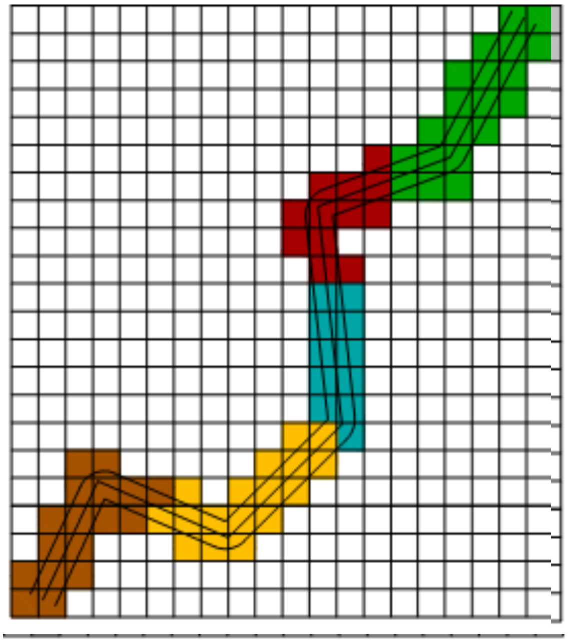
Step 1: trajectories as intuitionistic fuzzy vectors

- Settings:
  - a grid partitioning of space
  - a target dimension p << # timestamps
- **Approximate trajectory**
  - a sequence of p regions (i.e., sets of cells crossed by the trajectory)

$$\overline{T}_i = \langle r_{i,1}, \ldots, r_{i,p} \rangle$$

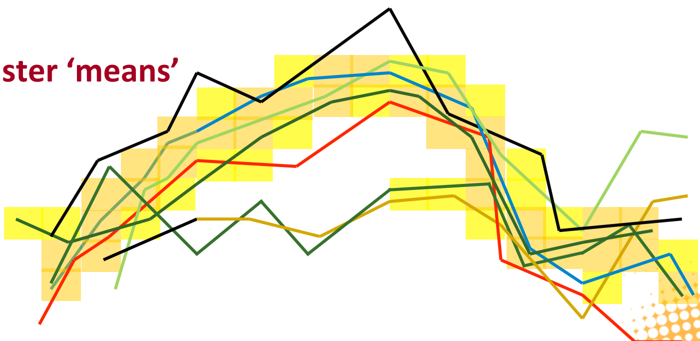
- **Uncertain Trajectory** (UnTra)
  - the ε-buffer of the approximate trajectory

$$\mathrm{UnTra}(\overline{T}_i) = \langle ur_{i,1}, \ldots, ur_{i,p} \rangle$$

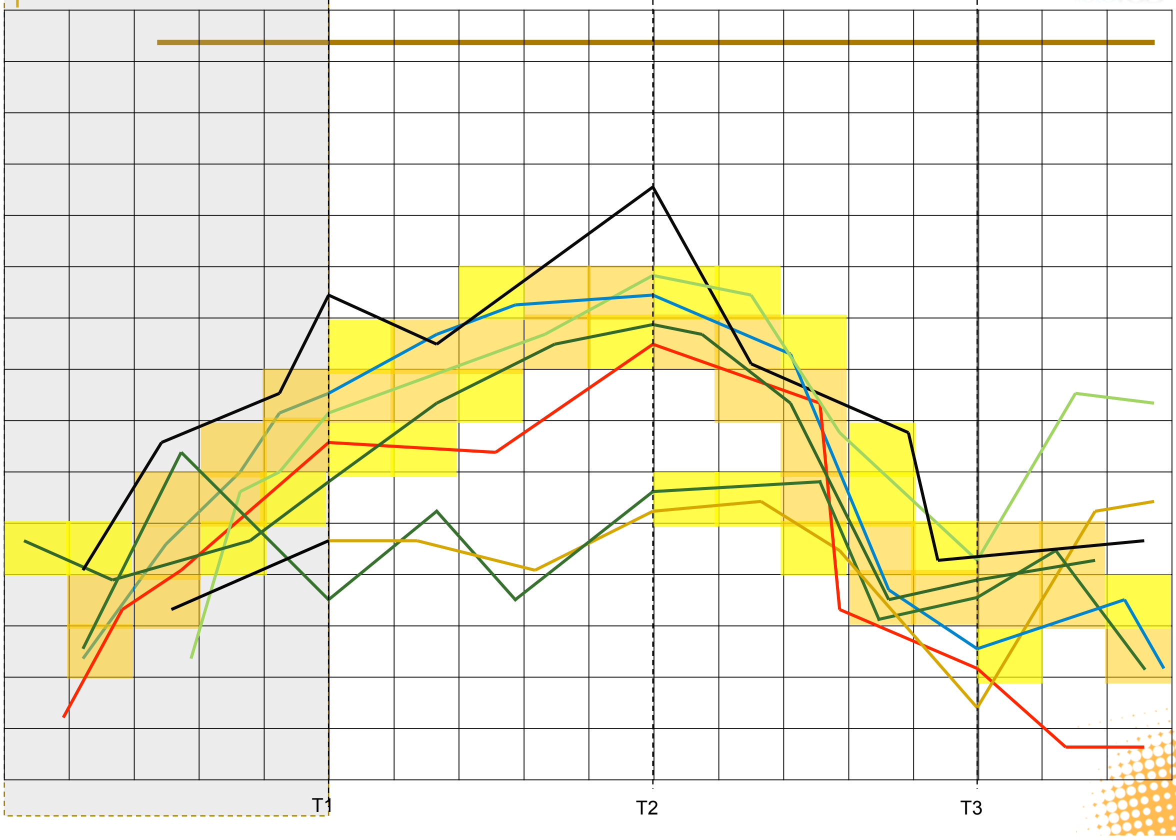


Steps 2-3: clustering using 'centroids'

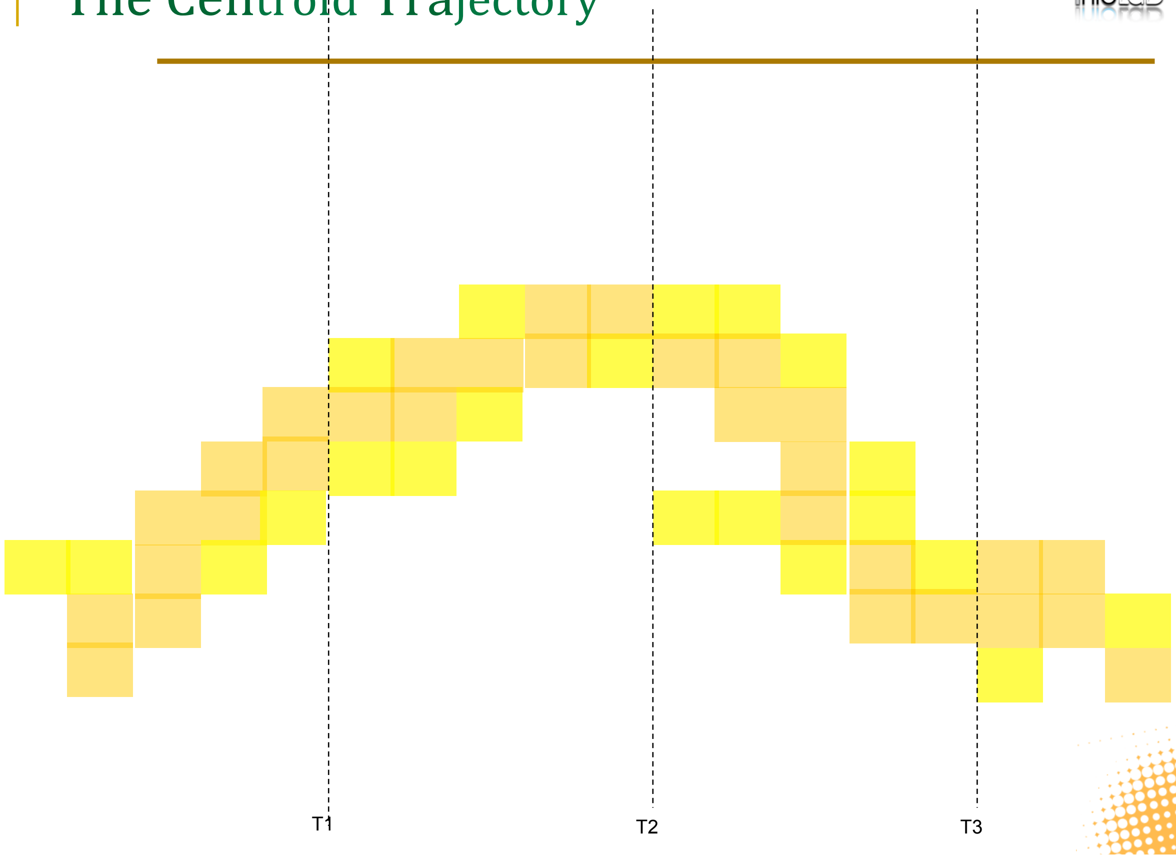
- Step 2 – discover the **centroid** of a bundle of trajectories
  - adopt a local similarity function to identify common sub-trajectories (concurrent existence in space-time), and
  - follow a region growing approach according to density
- Step 3 - clustering
  - adopt Fuzzy-C-Means (FCM), an extension of k-means for clustering uncertain data
  - **using CenTRa as the cluster 'means'**



Algorithm CenTra: An example



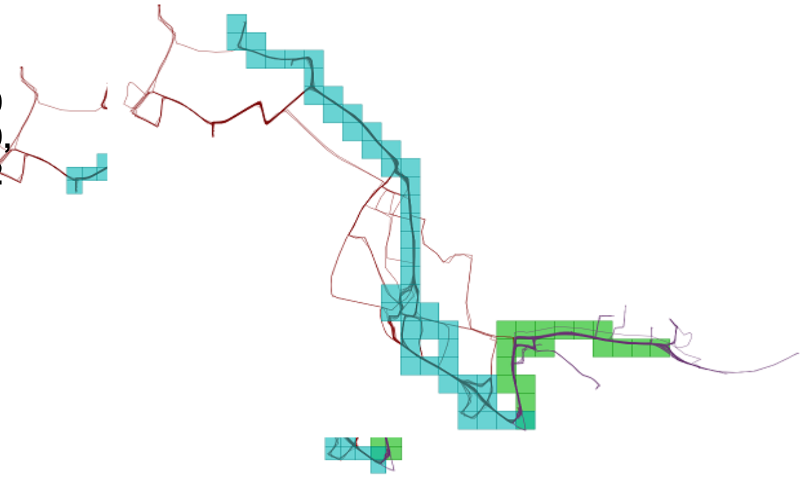
The Centroid Trajectory

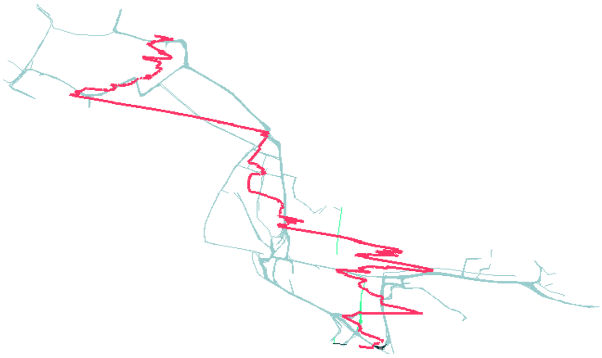


Quality of centroid

TRACLUS vs. CenTra

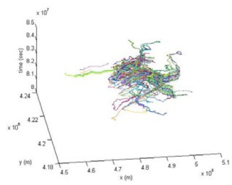
***cell size*=1.3%, *ε*=0, *δ*=0.09**
***cell size*=1.3%, *ε*=0, *δ*=0.09,**
***cell size*=2.8%, *ε*=0, *δ*=0.02**

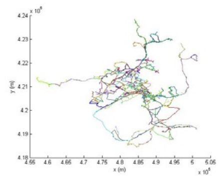




CenTR-I-FCM on work…

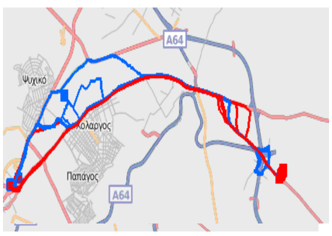
- Settings
  - Dataset: ’Athens trucks’ MOD (www.rtreeportal.org)
    - 50 trucks, 1100 trajectories, 112,300 position records
  - Use CommonGIS [Andrienko et al., 2007] to identify real clusters

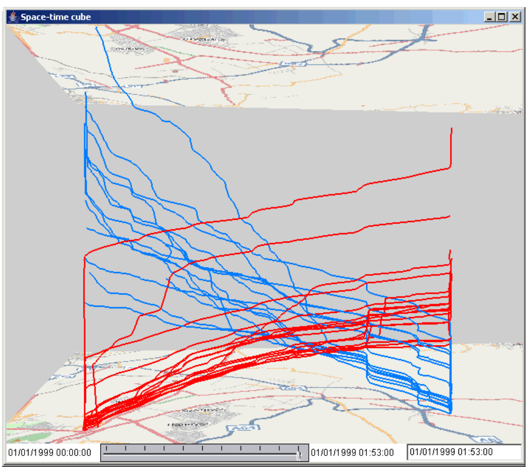




“Round trips” clusters

“Linear” clusters

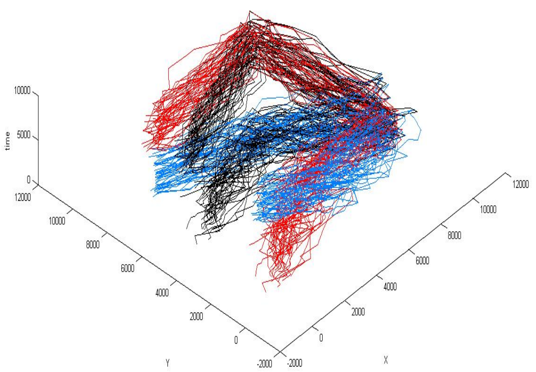


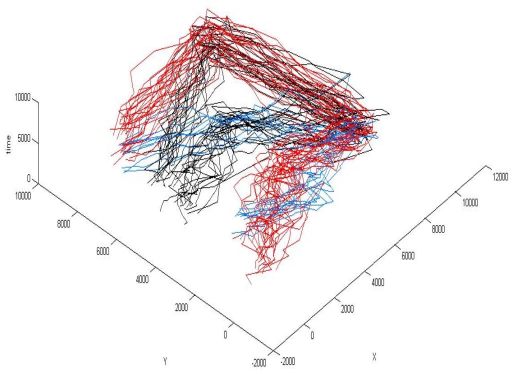


Trajectory sampling

Sampling trajectory datasets

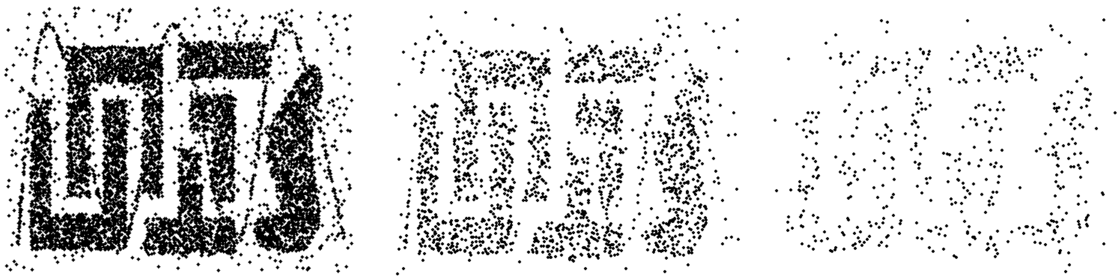
- Can we get the gist of a real large MOD by visualizing it? Can we do this automatically?
- If yes, we can
  - extrapolate the query results from queries in the sampled MOD
  - discover mobility patterns working with a "representative" subset





A general definition

- **Sampling** is the main technique employed for data selection.
  - It is often used for both the preliminary investigation of the data and the final data analysis.
  - Statisticians sample because obtaining the entire set of data of interest is too expensive or time consuming.



8000 points | 2000 Points | 500 Points

A general definition (cont.)

- The key principle for effective sampling is the following:
  - using a sample will work almost as well as using the entire data sets, if the sample is representative
  - A sample is representative if it has approximately the same property (of interest) as the original set of data

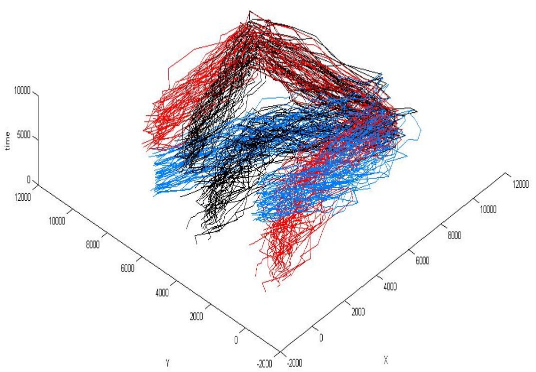
- As such, sampling is also used in data mining because **processing the entire set of data of interest is too expensive or time consuming.**

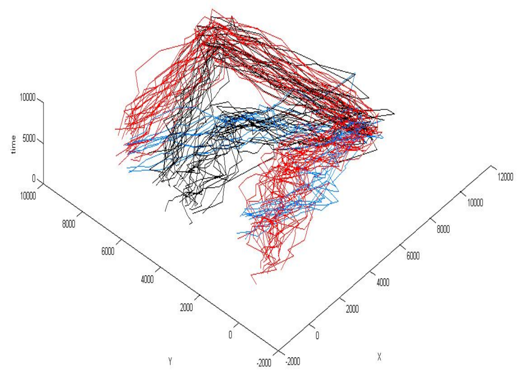
Types of sampling

- Simple Random Sampling
  - There is an equal probability of selecting any particular item
- Stratified sampling
  - Split the data into several partitions; then draw random samples from each partition
- Sampling with vs. without replacement
  - As each item is selected, it remains at (vs. it is removed from) the population
    - In sampling with replacement, the same object can be picked up more than once!

Back to mobility data…

- How can we select some out of the entire population of trajectories?
- Recall that …
  - “A sample is representative if it has approximately the same property (of interest) as the original set of data”

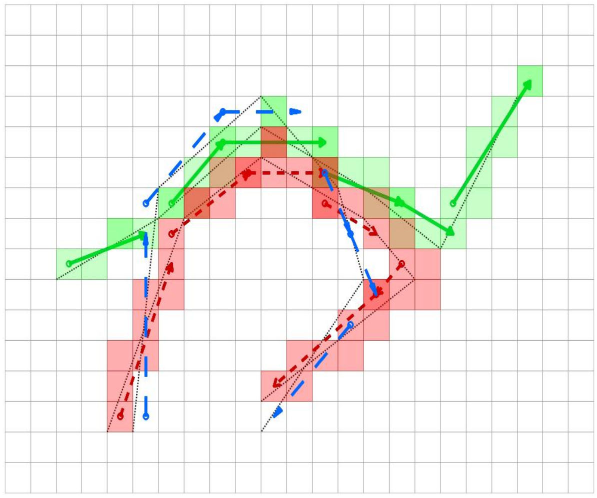




Vector representation of trajectories

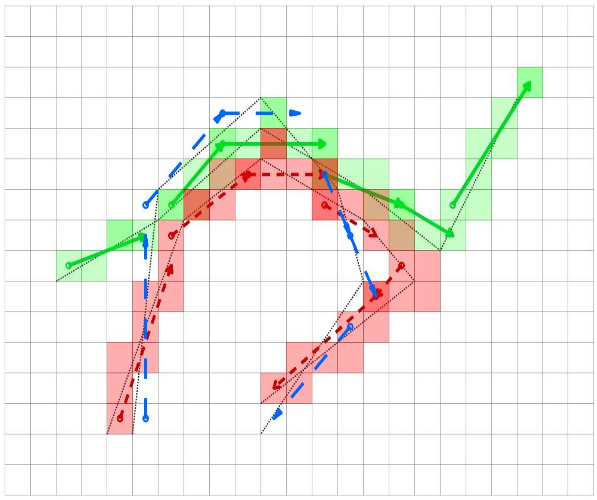
- Settings (as before):
  - a grid partitioning of space
  - a target dimension *p* << # timestamps
- **Approximate trajectory** (ApTra)
  - consists of *p* "directed regions", which are pairs of
    - region (i.e., set of cells crossed by the trajectory) and
    - region's direction (defined wrt. its ending cells)

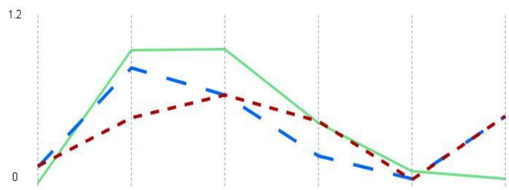
$$\bar{T}_i = <(r_{i,1}, \vec{d}_{i,1}),\ \ldots,\ (r_{i,p}, \vec{d}_{i,p})>$$



"Representative" trajectories

- "**Representativeness**" of a trajectory
  - the number of other trajectories that are **similar** to it
- Technically:
  - A **voting process** applied to each directed region $dr_{i,j}$
  - A directed region is voted by an ApTra in the dataset according to their distance
  - Thus, a 3rd value ("representative-ness") is attached to each directed region
- The result: **Representative trajectory** (ReTra)
  - a set of p triplets $(r_{i,j}, \vec{d}_{i,j}, v(dr_{i,j}))$





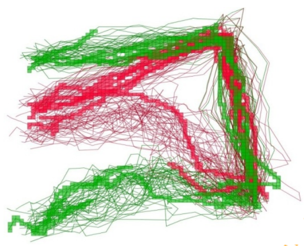
T-Sampling problem formulation

- For each sub-trajectory in a dataset D, we can calculate its trajectory representativeness descriptor Vi(D)
- **Motivation**: Selecting the top-voted sub-trajectories is not the best idea for making a sampling set !!
- Definition of the T-sampling problem:
  - Optimization problem: find an appropriate subset S of D, which maximizes the function SR(S): $SR(S) = \sum_{i=1}^{N} S_i \bullet V_i(D) \bullet (1 - V_i(S))$
    - Si is equal to 1 (0) when (sub-)trajectory Ti belongs (does not belong, resp.) to the sampling set.
  - **Meaning:** the number of trajectories in D that find their representatives in S is maximized

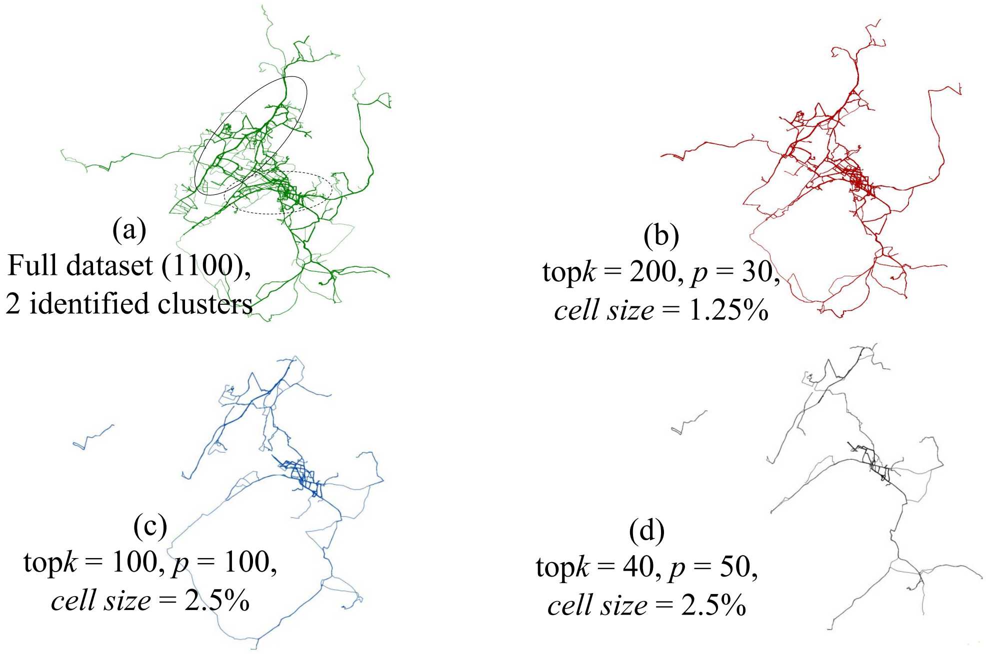
T-sampling on work…

- How "good" is the sample produced by T-sampling?
- … where "good" means …
  - Can we visualize real-world datasets using only a subset?
  - Does the sample preserve the hidden mobility patterns?





T-sampling on work…



(a)
Full dataset (1100),
2 identified clusters

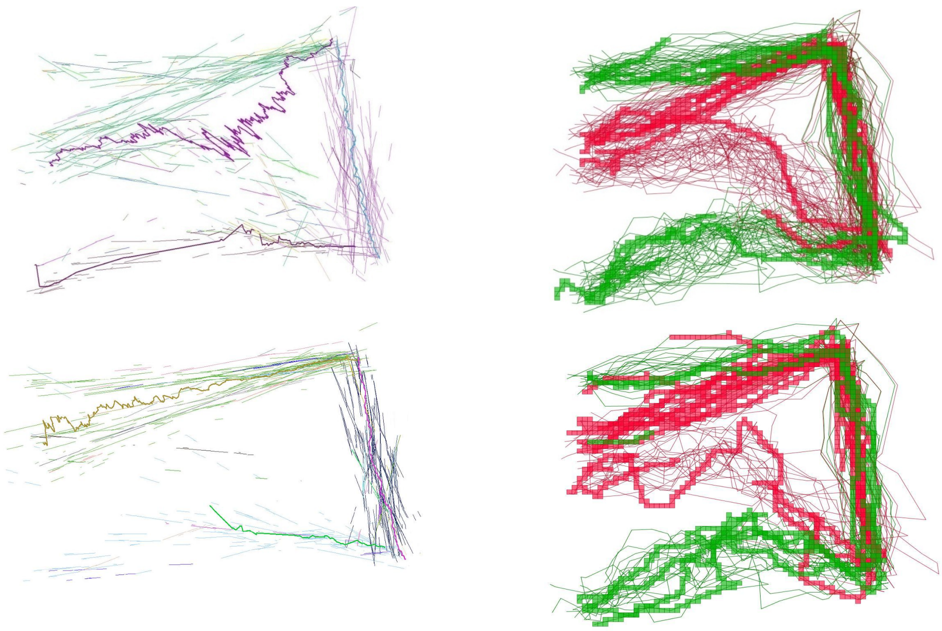
(b)
top*k* = 200, *p* = 30,
cell size = 1.25%

(c)
top*k* = 100, *p* = 100,
cell size = 2.5%

(d)
top*k* = 40, *p* = 50,
cell size = 2.5%

T-sampling on work…

- Preservation of mobility patterns



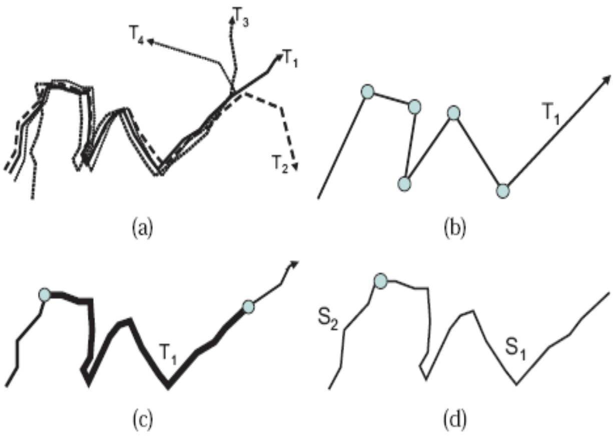
Before

After
Sampling

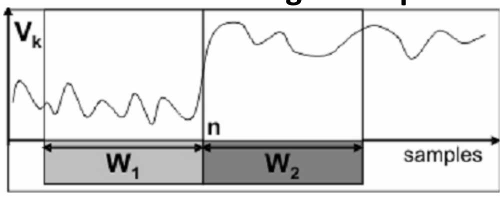
TRACLUS representatives [Lee et al. 2007] and **CenTra** centroids [Pelekis et al. 2009]

“Representative” **sub**-trajectories

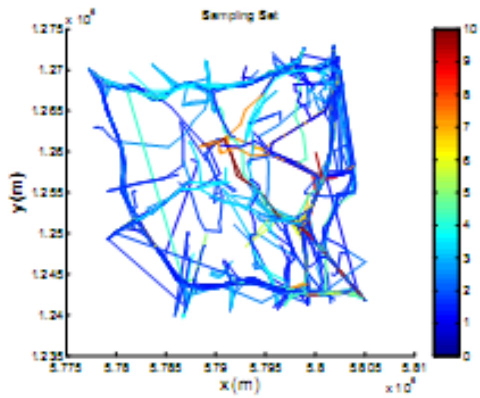
Trajectory Segmentation and Sampling



Continuous Voting Descriptors

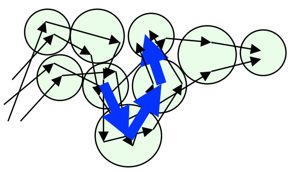


Application to Milano GPS dataset

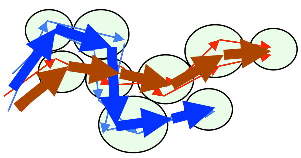


Research challenges in mobility data mining

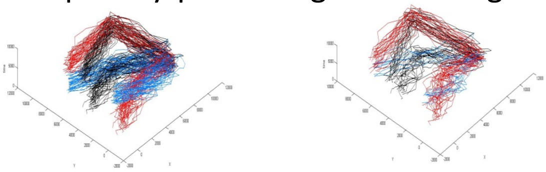
- Frequent pattern mining
  - What about a hierarchy of T-patterns, from more to less general? e.g.
    - coarser level: from north to downtown in 1 hour
    - finer level: from highway A to ring in 20 min.



- Trajectory clustering
  - (as usual) find the optimal number ‘k’ of clusters
  - incremental clustering



- Trajectory sampling
  - Could samples be used for privacy-preserving data mining?



part III: Visual Analytics

Key questions that arise

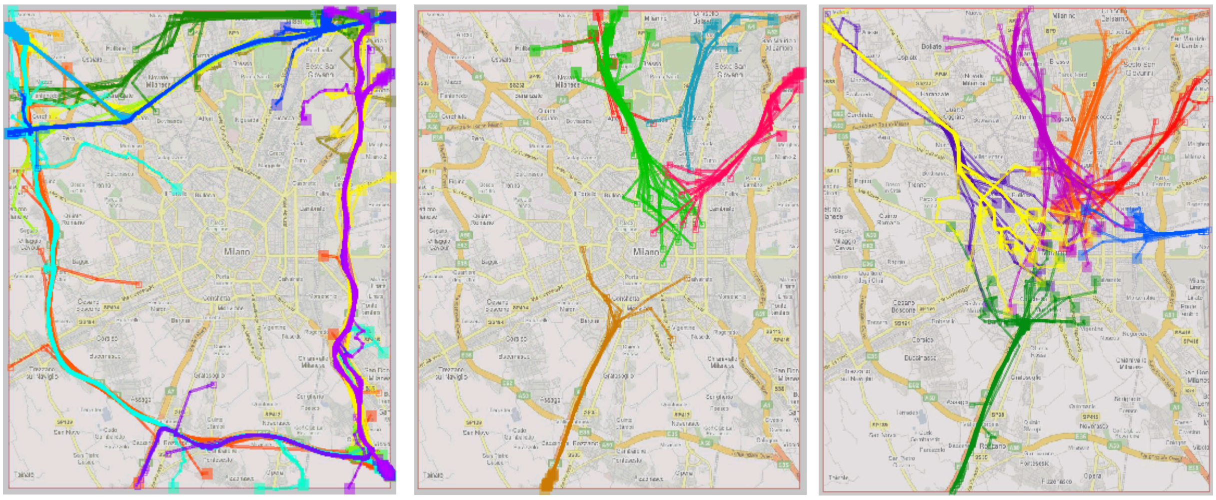
- What kind of analysis is suitable for mobility data?
  - In particular, trajectories of moving objects?
  - How does infrastructure (e.g. road network) affect this analysis?
- Which patterns / models can be extracted out of them?
  - Clusters, frequent patterns, anomalies / outliers, etc.
  - How to compute such patterns / models efficiently?
- Can we aid analysis by **visual artifacts**?
  - How should we visualize the mined patterns/models?

Visual analytics for mobility data

- A synergy of
  - interactive visualization,
  - database processing and
  - data mining
- helps to make sense from large amounts of movement data by interactive, visually-driven exploratory data analysis

- *The source of the following screenshots is CommonGIS® VA toolkit by N. & G. Andrienko, Fraunhofer IAIS.*

Examples of clusters of trajectories



- What is an appropriate way to visualize groups of trajectories?

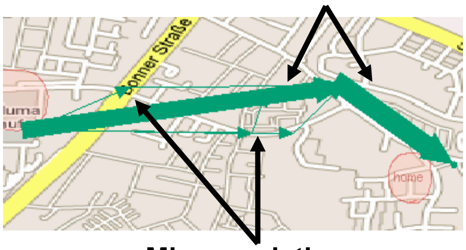
Summarizing a bunch of trajectories

1) Trajectories → sequences of
   "moves" between "places"

2) For each pair of "places", compute
   the number of "moves"

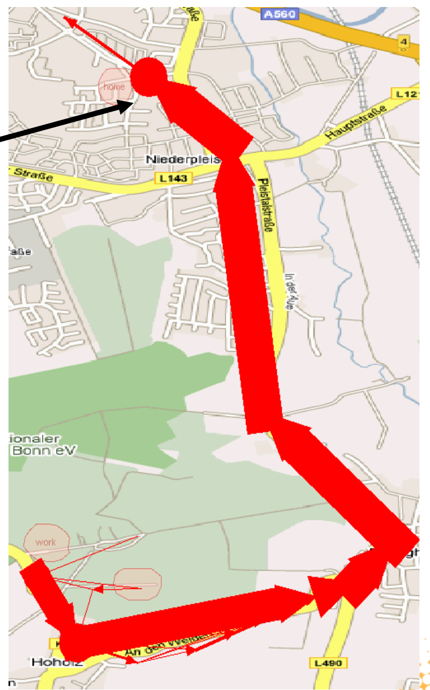
3) Represent "moves" by vectors
   (arrows) with proportional widths

Major flow

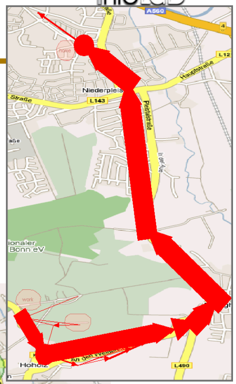


Minor variations

Many small moves

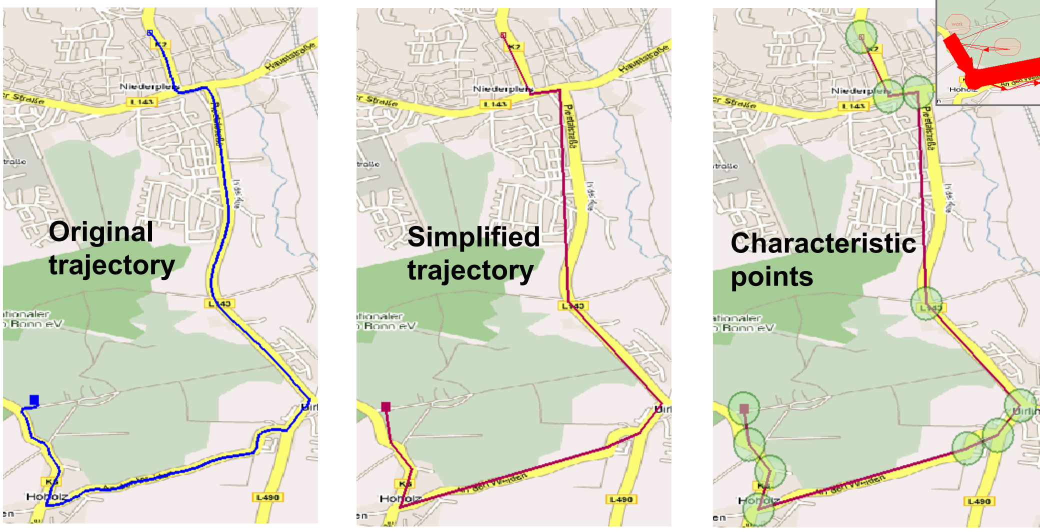


Defining “places”

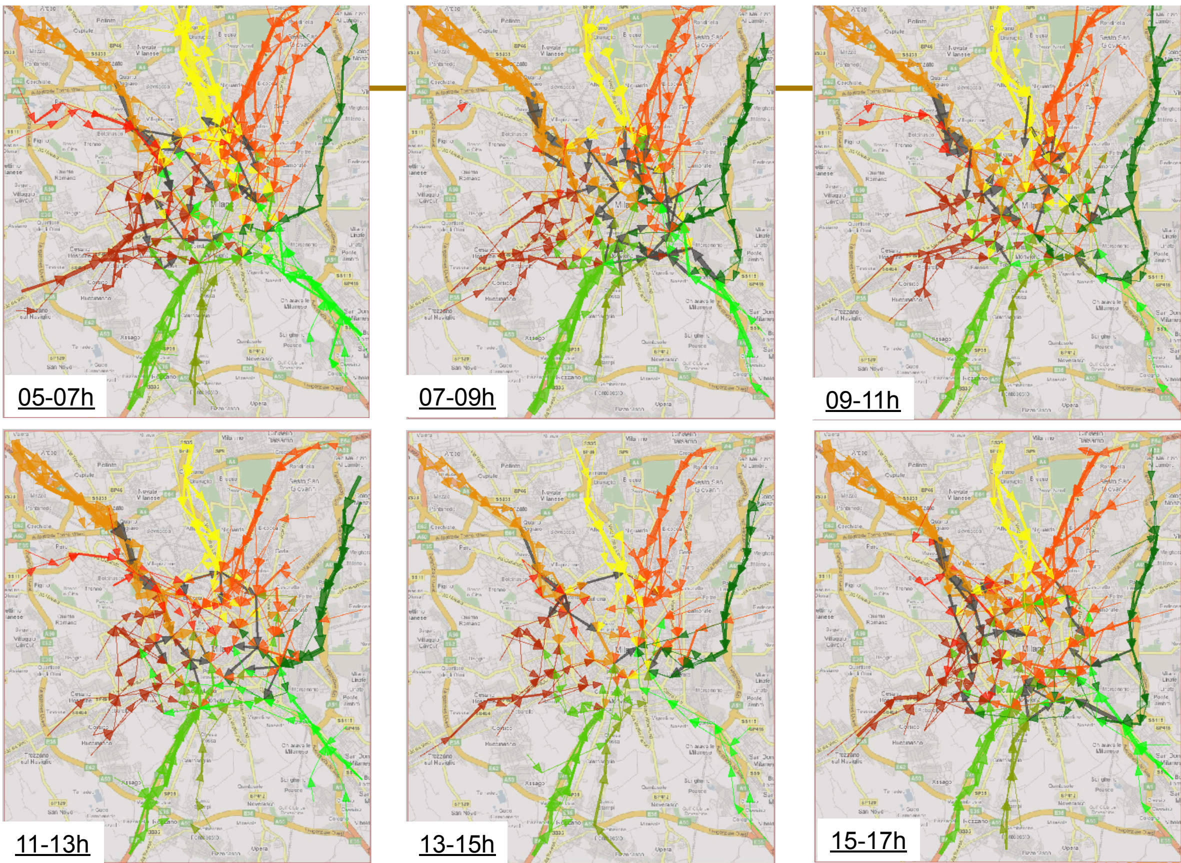


1) Extract characteristic points from all trajectories

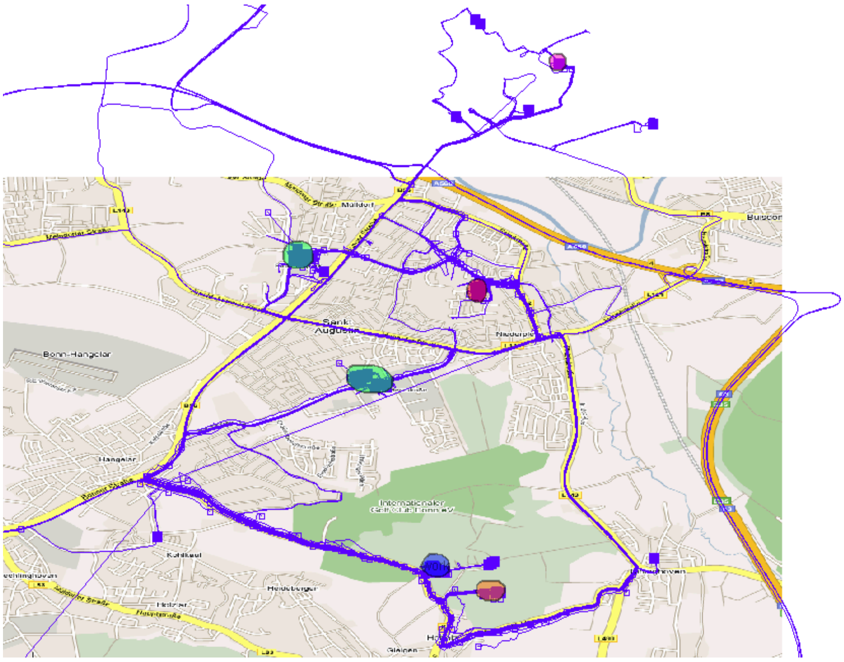
2) Build areas (e.g. circles) around groups of points and isolated points



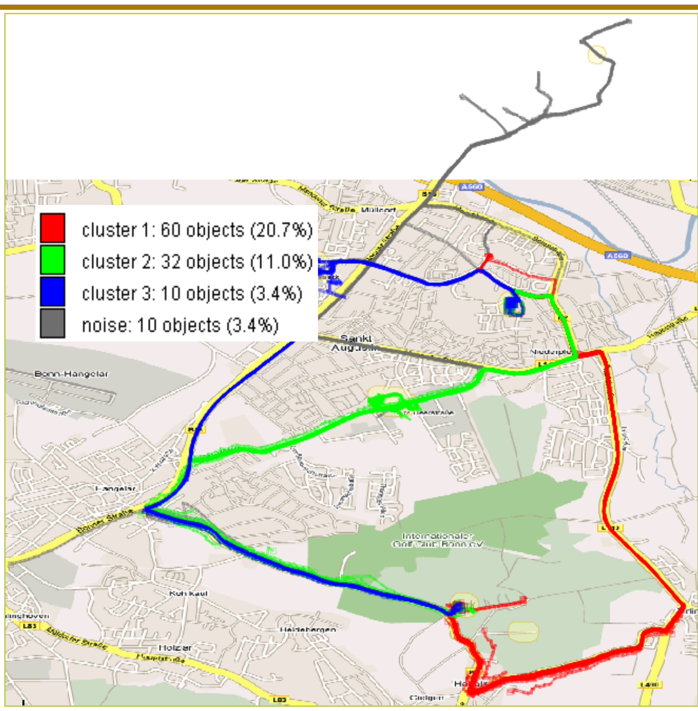
Exploration of the use of the most popular routes towards the centre by times of the day



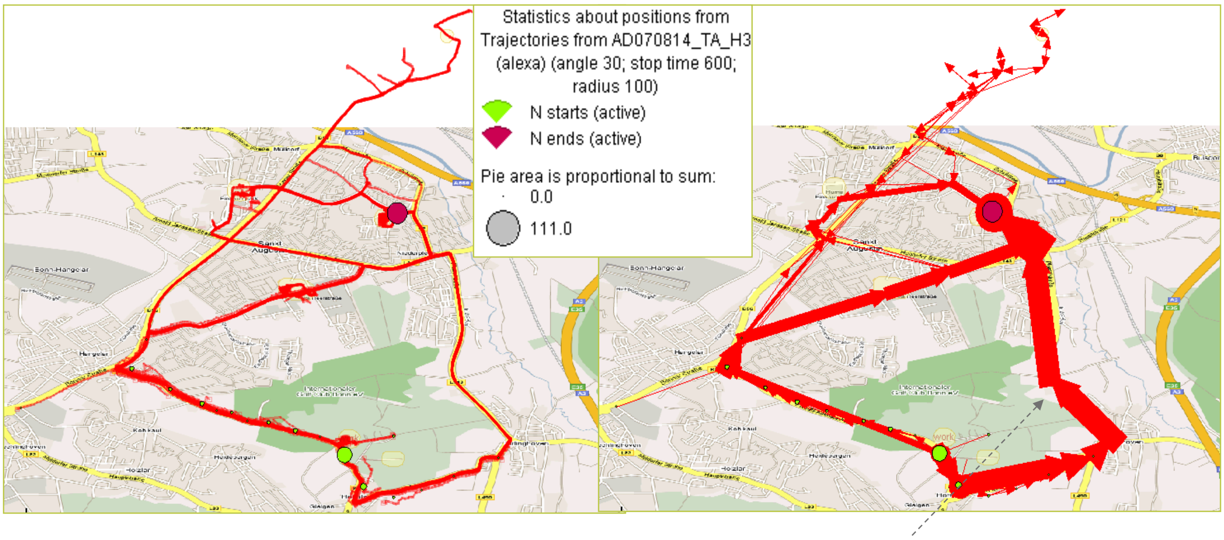
Looking for frequent stops & moves



Clustering typical routes

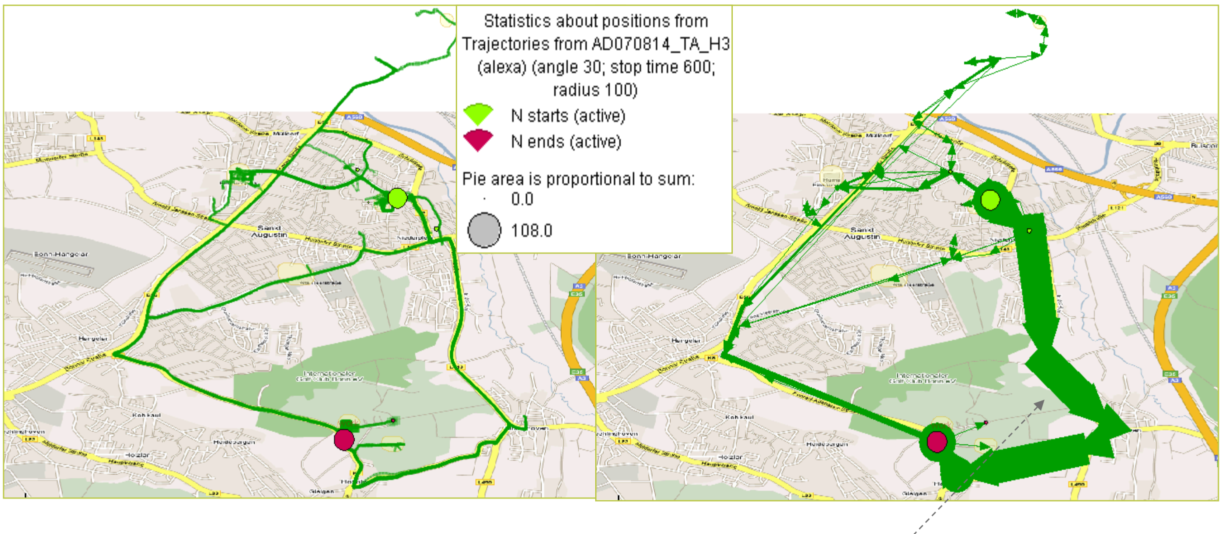


Cluster 1: from work to home



Observation: the eastern route is chosen more often

Cluster 2: from home to work



Observation: the eastern route is chosen **much** more often

Conclusions on part III

- Visual analysis over mobility data is a key tool for most applications

- Research challenges
  - Progressive refinement through visually-driven exploration
    - Progressively adopting different similarity functions
    - Progressive clustering by sampling

Summarizing…

Conclusions & Future trends

- (Privacy-preserving) Mobility Data Mining strives for a win-win situation
  - Obtaining the advantages of collective mobility knowledge without disclosing inadvertently any individual mobility knowledge.
  - Interdisciplinary effort: ***solutions can only be obtained via an alliance of technology, legal regulations, and social norms*** (Rakesh Agrawal)

- Challenge: **Mobile social networks**
  - Facebook, Twitter, etc.: currently, 1 billion users of social media; what if their movement is added?
  - Towards complex social networks of moving interacting objects.

Questions

Reading list

Mobility Data Warehousing

- Han, J. et al. (1998) Selective Materialization: An Efficient Method for Spatial Data Cube Construction. Proceedings of PAKDD.
- Jensen, C.S. et al. (2001) Location-Based Services: A Database Perspective. Proceedings of Scandinavian GIS.
- Jensen, C.S. et al. (2004) Multidimensional data modeling for location-based services, The VLDB Journal, 13: 1–21.
- Leonardi, L. et al. (2010) T-Warehouse: Visual OLAP analysis on trajectory data. Proceedings of ICDE.
- Leonardi, L. et al. (2009) Frequent Spatio-Temporal Patterns in Trajectory Data Warehouses. Proceedings of ACM SAC.
- Marketos, G. et al. (2008) Building Real World Trajectory Warehouses. Proceedings of MobiDE.

Mobility Data Warehousing (cont.)

- Marketos, G. and Y. Theodoridis (2010) Ad-hoc OLAP on Trajectory Data. Proceedings of MDM.
- Orlando, S. et al. (2007a) Spatio-Temporal Aggregations in Trajectory Data Warehouses. Proceedings of DaWaK.
- Orlando, S. et al. (2007b) Trajectory Data Warehouses: Design and Implementation Issues. J. Computing Science & Engineering, 1: 211-232.
- Pelekis, N. et al. (2008) Towards Trajectory Data Warehouses. Chapter in Mobility, Data Mining and Privacy: Geographic Knowledge Discovery. Springer-Verlag. 2008.
- Shekhar, S. et al. (2001) Map Cube: a Visualization Tool for Spatial Data Warehouses, Chapter in Geographic Data Mining and Knowledge Discovery. Taylor and Francis.
- Tao, Y. et al. (2004) Spatio-Temporal Aggregation Using Sketches. Proceedings of ICDE.

Trajectory Pattern Querying

- Benkert, M. et al. (2008) Reporting Flock Patterns. Computational Geometry, 41: 111-125.
- Frentzos, E. et al. (2007) Index-based Most Similar Trajectory Search. Proceedings of ICDE.
- Gudmundsson, J. and M. van Kreveld (2006) Computing longest duration flocks in trajectory data. Proceedings of ACM-GIS.
- Hu, H. et al. (2005) A Generic Framework for Monitoring Continuous Spatial Queries over Moving Objects. Proceedings of ACM SIGMOD.
- Papadias, D. et al. (2003) Query Processing in Spatial Network Databases. Proceedings of VLDB.
- Pelekis, N. et al. (2007) Similarity Search in Trajectory Databases. Proceedings of TIME.
- Tao, Y. et al. (2002) Continuous Nearest Neighbor Search. Proceedings of VLDB.

Frequent Pattern Mining

- Cao, H. et al. (2005) Mining frequent spatio-temporal sequential patterns. Proceedings of ICDM.
- Giannotti, F. et al. (2006) Efficient Mining of Temporally Annotated Sequences. Proceedings of SDM.
- Giannotti, F. et al. (2007) Trajectory Pattern Mining. Proceedings of KDD.
- Hadjieleftheriou, M. et al. (2005) Complex Spatio-Temporal Pattern Queries. Proceedings of VLDB.
- van Kreveld, M. et al. (2007) Efficient Detection of Motion Patterns in Spatio-Temporal Data Sets. GeoInformatica, 11: 195-215.
- Laube, P. et al. (2005) Discovering relative motion patterns in groups of moving point objects. Int. Journal of Geographical Information Science, 19: 639-668.
- Li, X. et al. (2007) Traffic density-based discovery of hot routes in road networks. Proceedings of SSTD.

Frequent Pattern Mining (cont.)

- du Mouza, C. and Rigaux, P. (2005) Mobility Patterns. GeoInformatica, 9: 297-319.
- Nakata, T. and Takeuchi, J. (2004) Mining traffic data from probe-car system for travel time prediction. Proceedings of KDD.
- Qu, Y. et al. (2003) Supporting Movement Pattern Queries in User-Specified Scales. IEEE Transactions on Knowledge and Data Engineering, 15: 26-42.
- Shekhar, S. et al. (2001) Data mining and visualization of twin-cities traffic data. Technical Report, TR-01-015, University of Minnesota.

Trajectory Clustering - Outlier Detection

- Alon, J. Et al. (2003) Disovering Clusters in Motion Time-series Data. Proceedings of CVPR.
- Gadez, I.V. et al. (2000) A General Probabilistic Framework for Clustering Individuals and Objects. Proceedings of KDD.
- Gaffney, S. and Smyth, P. (1999) Trajectory Clustering with Mixtures of Regression Models, Proceedings of KDD.
- Hadjieleftheriou, M. et al. (2003). On-Line Discovery of Dense Areas in Spatio-temporal Databases. Proceedings of SSTD.
- Kalnis, P. et al. (2005) On Discovering Moving Clusters in Spatio-temporal Data. Proceedings of SSTD.
- Lee, J.-G., Han, J., Li, X. (2007) Trajectory Clustering: A Partition-and-Group Framework, Proceedings of ACM SIGMOD.
- Li, X. et al. (2006) Motion-Alert: Automatic Anomaly Detection in Massive Moving Objects. Proceedings of ISI.
- Nanni, M. and Pedreschi, D. (2006) Time-focused clustering of trajectories of moving objects. J. of Intelligent Information Systems, 27: 267-289.

Trajectory Clustering - Outlier Detection (cont.)

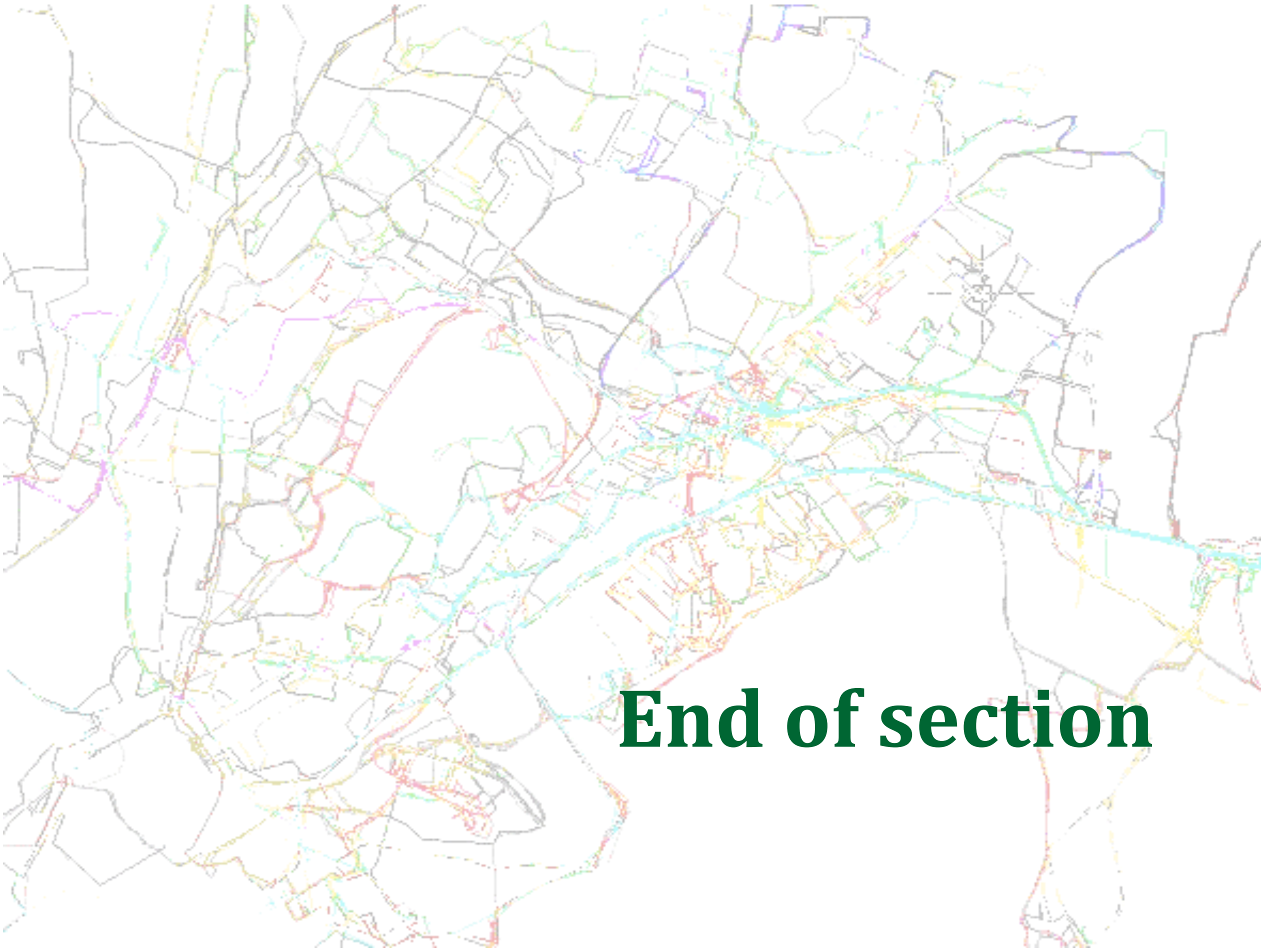
- Pelekis, N. et al. (2009) Clustering Trajectories of Moving Objects in an Uncertain World. Proceedings of ICDM.
- Sacharidis, D. et al. (2008). On-line discovery of hot motion paths. Proceedings of EDBT.
- Vlachos, M. et al. (2002) Discovering Similar Multidimensional Trajectories. Proceedings of ICDE.
- Ying, X., Xu, Z., Yin, W. G. (2009). Cluster-Based Congestion Outlier Detection Method on Trajectory Data. Proceedings of FSKD.

Trajectory sampling

- Panagiotakis, C. et al. (2009) Trajectory Voting and Classification Based on Spatiotemporal Similarity in Moving Object Databases. Proceedings of IDA.
- Panagiotakis, C. et al. (2011) Segmentation and Sampling of Moving Object Trajectories based on Representativeness. IEEE TKDE.
- Pelekis, N. et al. (2010) Unsupervised Trajectory Sampling. Proceedings of ECML/PKDD.

Visual Analytics over Mobility Data

- Andrienko, N. and Andrienko, G. (2007) Designing Visual Analytics Methods for Massive Collections of Movement Data. Cartographica, 42: 117-138.
- Laube, P. et al. (2005) Discovering Relative Motion Patterns in Groups of Moving Point Objects. Int. J. Geographical Information Science, 19: 639-668.
- Rinzivillo, S. et al. (2008) Visually-driven analysis of movement data by progressive clustering. J. of Information Visualization, 7: 225-239.



End of section