Universiteit Leiden
The Netherlands

Urban Computing

Dr. Mitra Baratchi

28 September 2020

Leiden Institute of Advanced Computer Science - Leiden University

Fourth Session: Urban Computing - Processing spatio–temporal data

Table of Contents

Preliminaries

Table of content

Examples

Real-world processes being studied in many domains are inherently spatio-temporal in nature including:

- Climate science
- Neuroscience
- Social sciences
- Transportation
- Earth sciences

Example

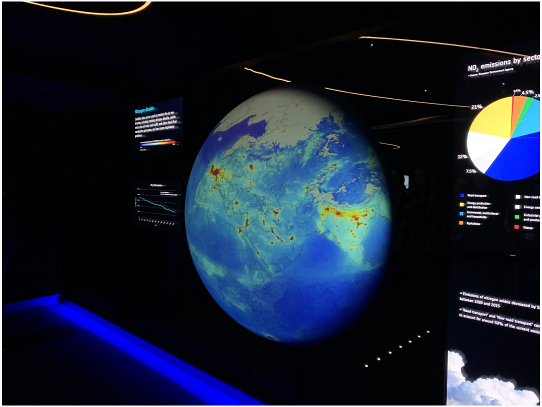


Figure 1: Example spatio-temporal data, NO2 emissions

Essence of spatio-temporal data

- **Temporal and spatial auto-correlation:** Nearby values in space and time tend to be alike
- **Spatial heterogeneity:** as we move away from a central point similarities decrease
- **Temporal non-stationarity:** as time passes similarities decrease
- **Multiple-scale patterns:** Daily (temporal scale 1) and seasonal (temporal scale 2) patterns within a patch of land (spatial scale 1) within a landscape (spatial scale 2)

What are spatio-temporal datasets?

- Spatio-temporal databases are an extension of spatial databases
- A spatio-temporal database embodies spatial, temporal, and spatio-temporal database concepts:
  - Geometry changing over time
  - Location of objects moving over invariant geometry

Spatio-temporal phenomena

1. **Spatio-temporal processes:** variables which are dependent on space and time $\leftarrow$
   - Weather
   - Population
2. **Moving object:** an object moving over space
   - People's trajectories
   - Cars' trajectories

How can we deal with spatio-temporal data?

- How did we deal with spatial data?
- Can we extend those methods to spatio-temporal data?

Spatio-temporal processes

Correspondence of spatial and spatio-temporal processes:

Spatial	Spatio-temporal
Geo-statistical	Spatio-temporal point referenced
Spatial point	Spatio-temporal event
Lattice	Spatio-temporal raster

Spatio-temporal processes

Correspondence of spatial and spatio-temporal processes:

Spatial	**Spatio-temporal**
Geo-statistical	**Spatio-temporal point referenced**
Spatial point	Spatio-temporal event
Lattice	Spatio-temporal raster

Spatio-temporal point reference data

- Measurements of a continuous spatio-temporal field over a set of fixed reference points in space and time
    - Meteorological variables
    - Temperature
    - Humidity

Spatio-temporal processes

Correspondence of spatial and spatio-temporal processes:

Spatial	Spatio-temporal
Geo-statistical	Spatio-temporal point referenced
Spatial point	**Spatio-temporal event**
Lattice	Spatio-temporal raster

Spatio-temporal event processes

- Random points in space and time denoting where and when the event occurred
  - Crime event
  - Road accidents

Spatio-temporal processes

Correspondence of spatial and spatio-temporal processes:

Spatial	Spatio-temporal
Geo-statistical	Spatio-temporal point referenced
Spatial point	Spatio-temporal event
Lattice	**Spatio-temporal raster**

Spatio-temporal raster processes

- Aggregated values over discrete regions of space and periods of time
    - Demographic information
    - Population increase in a city over a year

Spatio-temporal phenomena

1. **Spatio-temporal processes:** variables which are dependent on space and time
   - Weather
   - Population
2. **Moving object:** an object moving over space $\leftarrow$
   - People's trajectories
   - Cars' trajectories

Moving objects

- **Trajectories:** Multi-dimensional sequences containing a temporally ordered list of locations visited by the moving object
- What can we do by analysis of trajectory data?
  - **Studying moving objects:** Can we cluster a collection of trajectories into a small set of representative groups?
  - **Studying locations:** Are there frequent sequences of locations within the trajectories that are traversed by multiple moving bodies?

Table of content

Data types (processes) and data instances

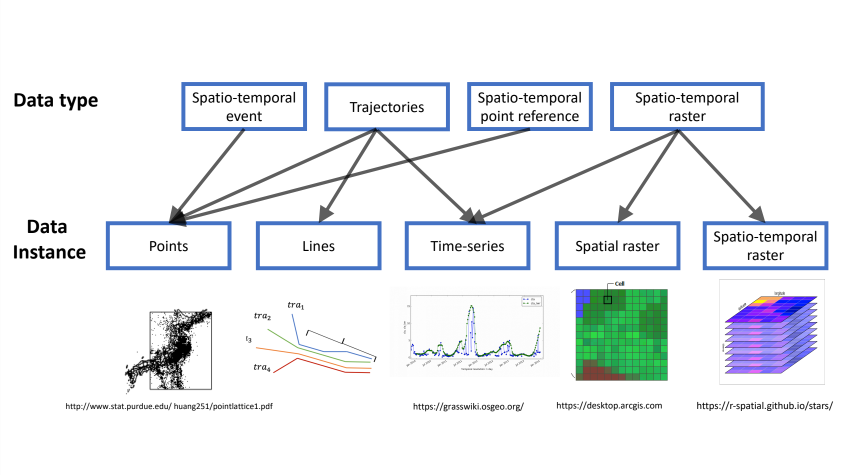


Figure 2: Spatio-temporal data instances and data types that can be used to represent them to algorithms as data instances

Methods for processing spatio-temporal data

Spatio-temporal statistics

Many statistical methods designed for spatial data can be extended to the spatio-temporal data:

- Spatio-temporal auto-correlation
- Space-time forecasting (auto-regressive models)
- Spatio-temporal kriging (interpolation)
- Spatio-temporal k-function (e.g., k-nearest neighbors)
- ...

Table of content

Auto-regressive models for spatio-temporal data

Y_n , Y_t are vectors of dependent variables of size n. ϕ, λ, ρ are model parameters. c is a constant. ϵ represents the noise term. W_n is the spatial weights matrix

- **Auto-regressive**
  - $y_t = c + \sum_{\tau=1}^{p} \phi_\tau y_{t-\tau} + \epsilon_t$
- **Spatial Auto-Regressive model (SAR)**
  - $y_n = c + \lambda \sum_{m \neq n} w_{n,m} y_m + \epsilon_n,$
  - $w_{n,m} y_n$ is referred to as the spatial lag term in the models
  - How we use W determines global and local effect
- **Space-Time Autoregressive model (STAR)**
  - $y_{n,t} = c + \sum_{\tau=1}^{p} (\phi_\tau y_{n,t-\tau} + \lambda_\tau \sum_{m \neq n} w_{n,m} y_{m,t-\tau}) + \epsilon_{n,t}$ [1]

Exercise: try to derive the equivalent if a spatio-temporal moving average model

[1] With STAR typically the degree of dynamics in time and space is also defined (e.g., STAR(1,1) defines autoregressive dynamics with one time lag and one spatial lag)

Methods for processing moving object data (spatio-temporal trajectories)

How does trajectory data look like?

Time

location

Mobile Entity ID, something else

Id	Timestamp	Location-long	Location-lat
2635	1997-07-24 20:50:00	– 149.007	63.809
2635	1997-07-24 21:23:35	– 148.897	63.766
2635	1997-07-27 22:30:23	– 148.967	63.824
2635	1997-07-31 02:52:48	– 149.026	63.803
2635	1997-08-03 01:47:04	– 149.046	63.795

GPS

	Scannerid	Time	DeviceAddress
2	1324	[illegible]	[illegible]
3	1324	1435505646	724290cde6af4645
4	1306	1435492293	6f1cf35f0bae2726
5	1293	1435513780	0c6185a840a4ee04
6	1293	1435513780	1eae7d8fde8dc50a
7	1297	1434550416	58884273fc450e86
8	1297	1434550412	6404531599f094e9
9	1297	1434550418	6e05f4edfdf08391
10	1297	1434550419	a04138e1aff48b03
11	1297	1434559015	6ea0d773c851ed56
12	1297	1434550018	58884273fc450e86

WiFi scans

Trajectory data, moving object data

- **Lagrangian motion data:** Allows collecting data of the movement of one entity globally
  - GPS
- **Eulerian motion data:** Allows collecting data of movement of many entities in restricted spaces
  - Wifi scanning
  - RFID
  - Video surveillance

What are different ways we can look at trajectory data?

We can query a trajectory dataset in different ways. Thus, we can study the data in different ways.

Query type	**Location**	**Entity**	**time**
1	**Fixed**	**Fixed**	Variable
2	**Fixed**	Variable	Variable
3	Variable	**Fixed**	Variable
4	Variable	Variable	Variable

Table 1: Different ways of looking at trajectory data

Patterns to extract from moving object data

Each type of query allows extracting a different type of pattern:

- **Individual**
    - Frequent
    - Periodic
    - Outliers
- **Social**
    - Flock
    - Leadership
    - Convergence
    - Encounter
- **Spatial**
    - Spatial interactions
    - Spatial functions

Table of content

Pre-processing trajectory data

- **In which ways** can we pre-process trajectory data?
  - Reduce the size of data $\rightarrow$ **Trajectory compression**
  - Remove noise $\rightarrow$ **Trajectory filtering**
  - Create workable instances $\rightarrow$ **Trajectory segmentation**

Trajectory compression

- **Goal:** reducing the dimensionality of the trajectory
- **Task:** Reducing the size of trajectory while preserving the precision
- **Good for:**
  - Efficiency (**computationally**) in pattern mining
  - Efficiency (**energy consumption**) in data collection procedure: the location of an object can be reported to the server when the precision reduces according to an error threshold.
  - Efficiency (**storage**)
- **Essence:** finding appropriate techniques and error measures for use in algorithms and performance evaluation.

Techniques for trajectory compression

- Uniform sampling
- Douglas-Peuker $\leftarrow$
- TD-TR
- Window-based algorithms (sliding window, open window, etc.)
- ...

Douglas-Peuker, Also known as Ramer-Douglas-Peucker

- Widely used in cartography and computer graphics
- Tries to estimate the original trajectory with one that has smaller number of points
- Iterative end-point fit algorithm
  - **Recursively** divides the line and approximates based on an error threshold
  - The **optimization** problem is formulated such that it minimizes the "area" between the original function and the approximate line segments
- Douglas-Peuker does not necessarily find a globally optimal solution

Douglas-Peuker approach

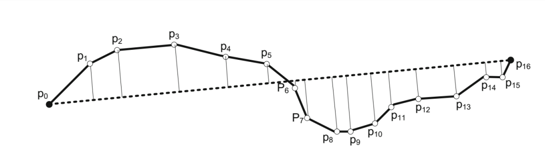


Figure 3: Step 1

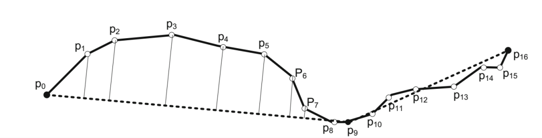


Figure 4: Step 2

Trajectory compression

Error metrics used for implementing trajectory compression:

- **Euclidean distance:** perpendicular distance between a point and a line
  - Only takes into account the geometric aspect of the trajectory representation without considering the temporal characteristics
- **Time synchronized euclidean distance:** Is a **time-distance** ratio metric
  - $SED(A, B, C) = \sqrt{(x'_B - x_B)^2 + (y'_B - y_B)^2}$
  - where $x'_B = x_A + \frac{x_c - x_A}{t_c - t_A}(t_B - t_A)$ and $y'_B = y_A + \frac{y_c - y_A}{t_c - t_A}(t_B - t_A)$

Trajectory compression: Mode of operation

- **Batch:**
  - Leads to high quality approximation due to access to full trajectories
  - It is not practical in many applications
- **Online:**
  - Typically limits the scope within a window
  - Certain trajectory properties can be preserved based on the application's needs
  - Intelligently select some negligible location points to retain a satisfactory approximated trajectory

Trajectory compression: Sliding window algorithm

- **Main idea:** Fitting the location points in a growing sliding window with a valid line segment
- Continues to grow the sliding window until the approximation error exceeds some threshold

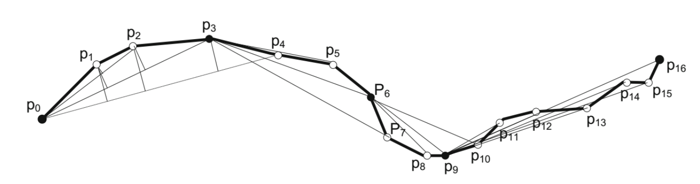


Figure 5: Sliding window algorithm

Trajectory filtering

- Spatial trajectories are often noisy because of the sensing technology
- Filtering techniques are used to smooth the noise and potentially decrease the error in the measurements
- This noise is different from the ϵ we had in the autoregressive models
- Trajectory model:
    - $\mathbf{z}_i = \mathbf{x_i} + \mathbf{v_i}$ → Measurement
    - $\mathbf{x}_i = (x_i, y_i)$ → True position
    - $\mathbf{v}_i \in N(0, R)$ → Noise

Trajectory filtering

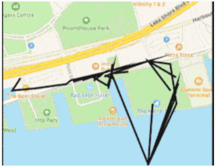


Figure 6: Raw noisy data, $\mathbf{Z}$

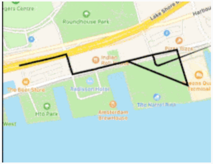


Figure 7: True position $\mathbf{X}$

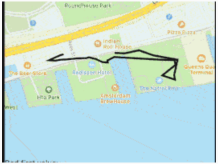


Figure 8: Estimated position $\hat{\mathbf{X}}$

Techniques for trajectory filtering

- Median filter
- Mean filter
- Kalman filter
- Particle filter
- ...

Filtering techniques

- Mean filter
  - $\hat{\mathbf{x}}_i = \frac{1}{n} \sum_{j=i-n+1}^{i} \mathbf{z}_j$
- Median filter
  - $\hat{\mathbf{x}}_i = median\{\mathbf{z}_{i-n+1}, \mathbf{z}_{i-n+2}, ..., \mathbf{z}_{i-1}, \mathbf{z}_i\}$

Mean and Median Filter

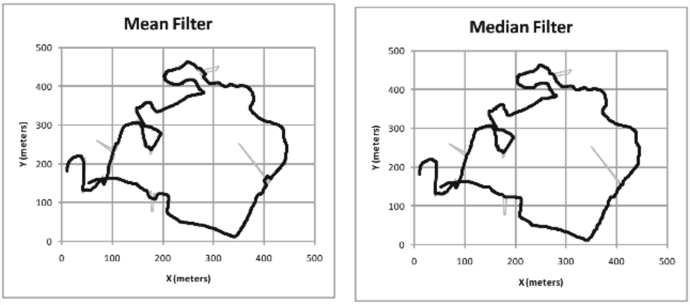


Figure 9: The result of applying the mean and the median filters

[2]

[2] Yu Zheng and Xiaofang Zhou. *Computing with spatial trajectories*. Springer Science & Business Media, 2011.

Properties of filters

- **Mean filter:**
  - Causal $\rightarrow$ depends on the values in the past
  - If the trajectory changes suddenly the effect on the trajectory is only gradually seen $\rightarrow$ It introduces a lag
  - Sensitive to outliers
- **Median filter:**
  - Not sensitive to outliers

Median and mean filters

- **Advantage:**
  - Simple and effective in smoothing trajectories
- **Disadvantages:**
  - Both suffer from the lag problem
  - They are not designed to help estimate higher order variables like speed and acceleration
  - In fact they might reduce the estimation accuracy of higher order variables

Advanced filters

- Advanced techniques that reduce lag and estimate the trajectory based on more than just location information
- **State-space models:**
  - Kalman filter
  - Particle filter

State and observations

- **States:** Things that you cannot measure directly but are interested in estimating
- Examples:
  - The true location
  - The true speed
- **Observations:** Noisy measurements from sensors
- Examples
  - GPS fixes
  - Acceleration

Kalman Filter

- **First use:** estimating trajectory of a space craft to the moon and back (There is no GPS trajectory in the space!)
- **General idea:** estimating the state variables from noisy observations by incorporating the physical domain knowledge $\rightarrow$ **Optimal estimation algorithm**
  - true location
  - speed
  - acceleration
- **Applications:**
  - Error correction
  - Data fusion: When measurements are available from various sensors but mixed with noise

Kalman filter

- Formulation of Kalman filter makes a distinction between what is **measured** as observations and what is **estimated** as states
    - **Measurement model:** How measurements are related to the states
    - **Dynamics model:** How previous states are related to future states

Measurement model

$$\mathbf{z}_i = x_i + v_i$$

v_i → Noise vector

$$\mathbf{x}_i = (x_i, y_i)$$

$$\mathbf{v}_i \in N(0, R)$$

Original trajectory model

$$\mathbf{z}_i = \begin{bmatrix} z_i^{(x)} \\ z_i^{(y)} \end{bmatrix}$$

Noisy measurements of the trajectory data

$$\mathbf{x}_i = \begin{bmatrix} x_i \\ y_i \\ s_i^{(x)} \\ s_i^{(y)} \end{bmatrix} \begin{matrix} \left.\begin{matrix} \\ \\ \end{matrix}\right\} \text{True unknown coordinates} \\ \left.\begin{matrix} \\ \\ \end{matrix}\right\} \text{True unknown velocity} \end{matrix}$$

State vector

$$\mathbf{z}_i = H_i \mathbf{x}_i + \mathbf{v}_i$$

Relationship between measurement and state vector

$$H_i = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

Measurement matrix

- Kalman filter gives estimates for the state vector $\mathbf{x}_i$
- H_i is the measurement matrix translating between x_i and z_i and matching the dimensionality of z_i and v_i

Dynamics model

$$\mathbf{x}_i = \phi_{i-1}\mathbf{x}_{i-1} + \mathbf{w}_{i-1}$$

Change with time

$$\phi_{i-1} = \begin{bmatrix} 1 & 0 & \Delta t_i & 0 \\ 0 & 1 & 0 & \Delta t_i \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

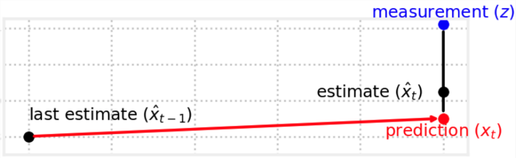
System matrix

- Approximates how the state vector $\mathbf{x}_i$ changes with time
- $\mathbf{w}_i$ is the Gaussian noise term

Kalman filter

A two-step algorithm that

- **Step 1:** Using the dynamics model extrapolates the current state to the next state
- **Step 2:** Incorporates the current measurement to make new estimates (weighted average of predicted state and the measurement)



3

[3] image source: (Roger R. Labbe. *Kalman and Bayesian Filters in Python*. Available: https://github.com/rlabbe/Kalman-and-Bayesian-Filters-in-Python. 2015)

Kalman filter

Advantages:

- No lag effect
- Richer state vector (velocity and location)
- It can incorporate more **physical knowledge** explaining how speed, time and displacement are related to each other
- It can be used to incorporate input from **other sensors**
- It can be used to incorporate **uncertainty** (using a covariance matrix)

Kalman filter

Limitations:

- To initialize the filter we need to have assumptions about the initial state and the uncertainty of the initial state
- The requirement is having a linear dynamic model
- It uses continuous variables without having a way to represent discrete variables like:
    - The mode of transportation
    - Activity

Particle filter

- Also makes disctinction between **measurement** and **dynamics model**
- To formulate these models it does not limit itself to physical movement parameters
- Has less strict assumptions about the linearity of equations and the noise model
- More **general** and **less efficient**

Particle filter

- **Measurement model:**
  - A conditional Gaussian distribution with covariance matrix R_i
  - $p(\mathbf{z}_i|\mathbf{x}_i) = N((x_i, y_i), R_i)$,
- **Dynamics model:**
  - Probability distribution $p(\mathbf{x}_i|\mathbf{x}_{i-1})$
  - It samples from the dynamics models
  - Instead of formalizing it we generate random samples of $\mathbf{x}_{i+1}$ from $\mathbf{x}_i$
  - Each generated sample is referred to as a **particle**
- Computation time and accuracy both depend on the number of particles

Stops and Moves

- Trajectories are considered as a collection of stops and moves[4]
- For many applications semantics of points in trajectories are more important than shapes
- Interest regions
    - Stay points
    - Activity regions
    - The path between two points of interest

[4] Andrey Tietbohl Palma et al. "A clustering-based approach for discovering interesting places in trajectories". In: *Proceedings of the 2008 ACM symposium on Applied computing*. ACM. 2008, pp. 863–868.

Stops and moves

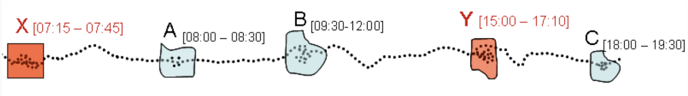


Figure 10: Stops and moves in a trajectory[5]

[5]Image source: (Palma et al., "A clustering-based approach for discovering interesting places in trajectories")

Not only a spatial clustering task

- **Challenge:**
  - We cannot only look at where point are clustered spatially
  - We want to find places that one trajectory has stopped but not only the overlap of a lot of trajectories
  - We want to find meaningful stops where a lot of trajectories stop and not any random stop
- Example approach: based on DBSCAN clustering[6]

[6] Palma et al., "A clustering-based approach for discovering interesting places in trajectories".

Lessons learned

- **Spatio-temporal processes:**
  - Extension of spatial process (geo-statistic, point, lattice processes)
  - Spatio-temporal auto-regressive as a combination of auto-regressive and spatial auto-regressive
- **Moving objects:**
  - Technology allows collection of trajectory data of moving object data in different ways:
    - **Lagrangian:** One individual visiting many locations
    - **Eulerian:** Many individuals passing one location
  - Different patterns can be extracted from data based on how we query the ID of moving objects and locations

Lessons learned (continued)

- **Trajectory pre-processing:**
  - **Trajectory compression:** summarize the trajectory data to key points, save space, save communication, efficient processing
    - Douglas-peuker (batch mode)
    - Window-based (online)
  - **Trajectory filtering:** GPS sensors produce noisy and only approximate location data
    - **Mean, Median filters:** simple, lag problem
    - **State-space filters:** defining a measurement (measurement, state relation) and dynamics model (past state, future state relation)
    - Kalman filter (physics laws, inflexible), Particle filter (flexible, slow)
  - **Trajectory segmentation:** Extracting region of interest by extending DBSCAN clustering

Table of content

End of theory!