Urban Computing

Dr. Mitra Baratchi

Leiden Institute of Advanced Computer Science - Leiden University

February 14, 2019

Second Session: Urban Computing - Processing Time-series Data

Agenda for this session

- Part 1: Preliminaries on time-series data
  - What does time-series data look like?
  - How do we represent time-series data?
- Part 2: Techniques for processing time-series data
  - Forecasting
  - Classification
- Part 3: Assignment

Part 1: Preliminaries on time-series data

Why do we care about time-series data

Time-series data are ubiquitous...
What types of data do we have in form of time-series for Urban computing research

- Temperature
- Humidity
- Number of people, cars passing a road
- Price of houses
- Sensor measurements

How does this data look like?

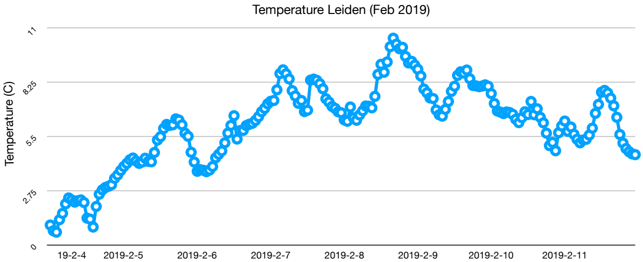


Figure: Temperature in Leiden during the month of February so far [1]

[1] data source: https://www.meteoblue.com

How does this data look like?

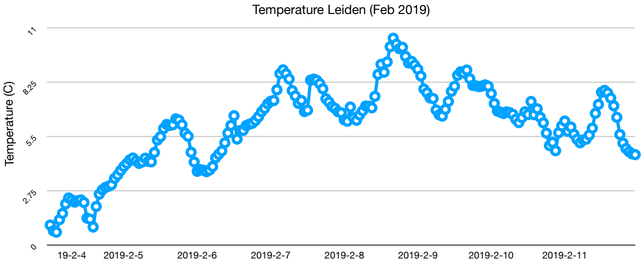


Figure: Temperature in Leiden during the month of February so far [1]

How many dimensions the data have?

[1] data source: https://www.meteoblue.com

How does this data look like?

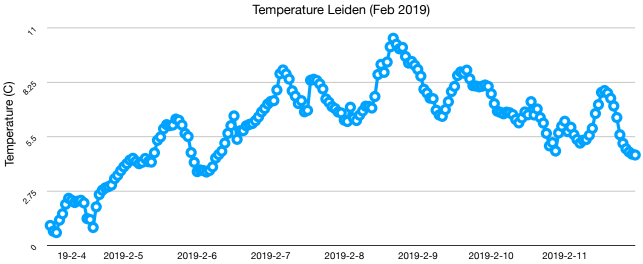


Figure: Temperature in Leiden during the month of February so far [1]

How many dimensions the data have? Length over time defines the dimensions, $\rightarrow$ many (even infinite)

[1] data source: https://www.meteoblue.com

How does this data look like?

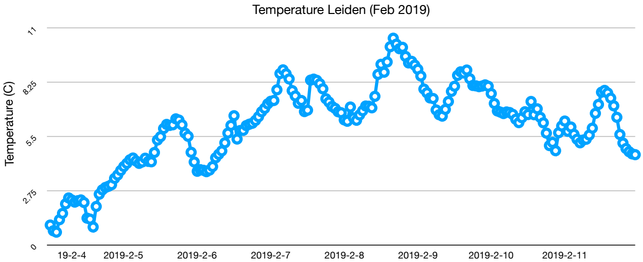


Figure: Temperature in Leiden during the month of February so far [1]

How many dimensions the data have? Length over time defines the dimensions, $\rightarrow$ many (even infinite)

How would you use this data for predicting the temperature of the following days?

[1] data source: https://www.meteoblue.com

Time-series versus signal

- By nature all the data we get is discrete. We can make it continuous by interpolation.
- Time series data is a signal variation over time...

Who has so far developed methods, algorithms for working with such data?

- Signal processing experts
- Statisticians

What can we do with such data?

- Predict?

What can we do with such data?

- Predict? (Better say forecast)

What can we do with such data?

- Predict? (Better say forecast)
- Classify

What can we do with such data?

- Predict? (Better say forecast)
- Classify
- Find patterns, clusters, outliers

What can we do with such data?

- Predict? (Better say forecast)
- Classify
- Find patterns, clusters, outliers
- Query

Two approaches to deal with or **represent** data

How do we represent time-series data in order to process it?

- **Approach 1**: Take it as it is.

Two approaches to deal with or **represent** data

How do we represent time-series data in order to process it?

- **Approach 1**: Take it as it is.
  - Represent it in time domain.
  - Main issue: (Time-series data is high dimensional $\rightarrow$ very difficult to work with)

Two approaches to deal with or **represent** data

How do we represent time-series data in order to process it?

- **Approach 1**: Take it as it is.
  - Represent it in time domain.
  - Main issue: (Time-series data is high dimensional $\rightarrow$ very difficult to work with)
- **Approach 2**: Represent it in a format that is more understandable or easier to work with. Representation techniques are designed to reduce the dimensionality of data as much as possible.

Two approaches to deal with or **represent** data

How do we represent time-series data in order to process it?

- **Approach 1**: Take it as it is.
  - Represent it in time domain.
  - Main issue: (Time-series data is high dimensional $\rightarrow$ very difficult to work with)
- **Approach 2**: Represent it in a format that is more understandable or easier to work with. Representation techniques are designed to reduce the dimensionality of data as much as possible.
  - Frequency domain
  - Time-frequency domain
  - ...

Approach 2-example 1

Fourier transform

- What is Fourier transform?
- What does it do?
- Why is it useful (in math, in engineering, etc)?
- How can it be useful in Urban Computing?

What is Fourier transform?

The basic elements:

Fourier theory shows that **all signals** (periodic and non-periodic) can be decomposed into a linear combination of sine waves defined based on their amplitude (A), period ($\frac{2\pi}{\omega}$), and phase (ϕ)

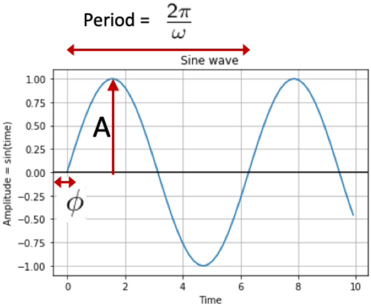


Figure: A sine wave, basic element of Fourier transform

$$Asin(\omega t + \phi)$$

Fourier transform in one image

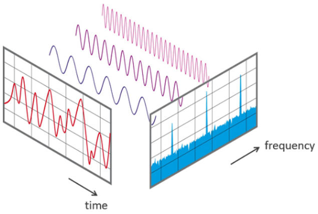


Figure: View of a signal in time and frequency domain[2]

[2] source: http://www.nti-audio.com/portals/0/pic/news/FFT-Time-Frequency-View-540.png

Why is it useful?

The main intuition:

If the frequency domain view is **sparse**, we can leverage the sparsity in different ways. (e.g. create new features for classification, compress the signal, ...)

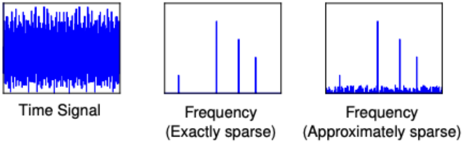


Figure: Different views of a signal and levels of sparsity. [3]

Question we should seek to answer before using a frequency domain transformation:

Does a transformation give us a sparser, thus, more understandable representation?

[3] Source: https://groups.csail.mit.edu/netmit/sFFT/slidesEric.pdf

Why is it useful?

Intuition behind frequency

- **Change, speed of change**: If change has a repetitive pattern we see it better in the frequency domain
- How can we use frequency analysis in urban computing?
  - Typically any phenomenon with a periodic pattern can be captured in the frequency domain
    - Periodicity in trajectory data (daily, weekly, seasonal, yearly patterns)
    - Activities with periodic patterns from accelerometer data (walking, running, biking)
    - Forecasting
    - Compressing data

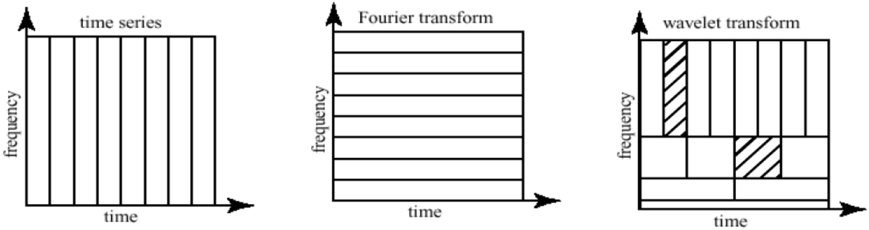
Approach 2-example 2

Wavelet transform

- Fourier analysis tells you **what** frequency components are strong in a signal, but not where in the signal (frequency view)
- Wavelet tells you **what** frequency components and also **where** they happen in a signal (time + frequency view)
- Useful for multi-resolution analysis

Time, Frequency, Frequency-time domains

4



- Lower frequency components take more time
- Higher frequency components take less time

[4] http://www.cerm.unifi.it/EUcourse2001/Gunther$_l$ecturenotes.pdf

Example case

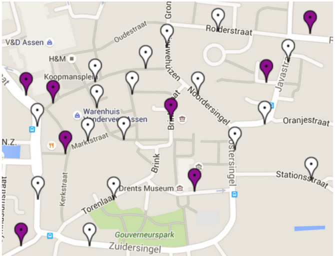


Figure: Assen sensor setup

We collected WiFi data from a city during TT festival.

- What would you do to see what happened in the city during the festival?
- How would you automate the process of detecting things that changed during the festival?

Multi-resolution analysis using Wavelets

Multiresolution analysis on visits of people to TT festival.

When and **how strongly** the number of visitors **changed**?

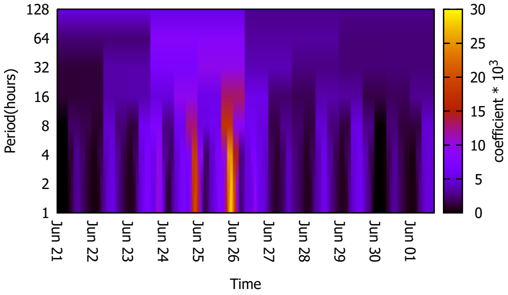


Figure: [PCB+17]

Example: Two approaches for dealing with the same problem

How do you find important periods from one person's trajectory data?

Example: Two approaches for dealing with the same problem

How do you find important periods from one person's trajectory data?

- **Method 1: Time domain analysis**
- **Method 2: Frequency domain analysis**

Method 1: Autocorrelation function

- **Auto**-correlation function (correlation of data with itself)
- The value of the autocorrelation function in (τ) can be interpreted as the self-similarity score of a time series when shifted (τ) timestamps

$$ACF_\tau = \frac{1}{T} \sum_{t=1}^{t=T-\tau(orT)} {}^5 (x_t - \overline{x})(x_{t+\tau} - \overline{x}).,\tau = 0, 1, 2, ..., T^6$$

[5] T is used in circular autocorrelation

[6] max value of $\tau canbesmaller$

Circular autocorrelation function

For implementing circular autocorrelation we use a shift operation from the end of time-series to its beginning

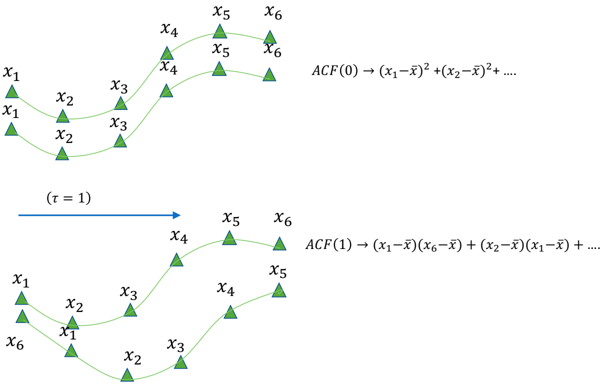


Figure: Calculating autocorrelation in different lags

Finding periodicity using autocorrelation function

Once ACF is visualized in a graph, the peaks on the autocorrelation graph can show the periods of repetitive behavior

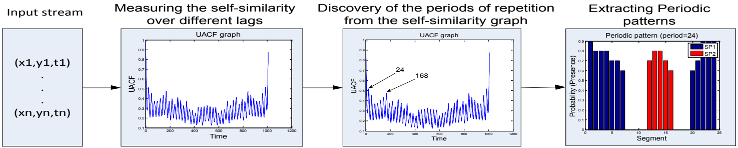


Figure: Finding periodic patterns using autocorrelation function [BMH14]

Method 2: Periodogram

- A periodogram is used to identify the dominant periods (or frequencies) of a time series.
- After performing Fourier transform the sum of squared coefficinets in each period is used to create the periodogram

Periodogram

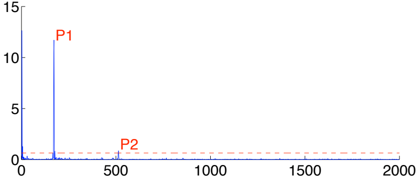


Figure: Periodogram

[LDH+10]

Why you need to know different methods

Each method has its pros and cons (typically, they complement each other in some way)

- In practice, on real data both of them fail in someway
- Fourier transform often suffers from the low resolution problem in the low frequency region, hence it provides poor estimation of large periods. (this is referred to as the **spectral leakage** problem)
- False positives can appear in periodogram that are caused by noise
- Autocorrelation offers accurate estimation for both short and large periods. However, It is more difficult to set the significance threshold for finding important periods.

Many more different methods for representing time-series data in alternative domains

[WMD+13]

- Discrete Cosine transform
- Discrete Fourier transform
- Discrete Wavelet transform
- Piecewise aggregate approximation
- Piecewise cloud approximation
- ...

What effects of time exist?

Some effects we would like to capture in a representation based on the task we have in mind

- **When** things happen?
- **How long** do they last?
- How do they **repeat**?
- How do they **follow** each other?
- When things start to **appear/disappear**?
- When and how things **change**?

Part 2: Techniques for processing time-series data

Classical forecasting using time-series

Problem:
Given $x_1, x_2, x_3, ...x_t$ forecast the value of $x_{t+1}, x_{t+2}...x_{t+n}$
Forecast horizon depending on the value n:

- Short-term
- Medium-term
- Long-term

Autoregressive models

- Classical models widely used by statisticians
- The **auto**-regressive model specifies that the output variable depends linearly on its **own previous values** and on a stochastic term
- Assumption: Having a stationary process
  - Time series is said to be strictly stationary if its properties are not affected by a change in the time origin. OR Joint probability distribution of $x_t, x_{t+1}, \ldots, x_{t+n}$ is equal to $x_{t+k}, x_{t+k+1}, \ldots, x_{t+k+n}$
  - In a more strict sense, a stationary time series exhibits similar statistical behavior in time and this is often characterized as a constant probability distribution in time

Regression, Auto-regressive, Moving average

- **Regression**
  - $Y_i = c + \phi X_i + \epsilon_i$
- **Autoregressive**
  - $X_t = c + \sum_{i=1}^{p} \phi_i X_{t-i} + \epsilon_t$
- **Moving average**
  - $X_t = c + \sum_{i=1}^{q} \phi_i \epsilon_{t-i}$
  - Literally moving average, (i.e.) average value of previous values of the time-series
- **Auto-Regressive Moving Average (ARMA)**
  - $X_t = c + \sum_{i=1}^{q} \phi_i \epsilon_{t-i} + \sum_{i=1}^{p} \phi_i X_{t-i}$

$\rightarrow$ c is constant, ϕ is model parameter, ϵ is white noise

Typical patterns in time-series that should be considered

How far can you go ahead in time:

- Seasonality
- Periodicity
- Trends

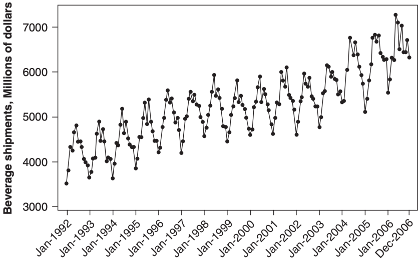


Figure: Time series with trend and periodicity [BJRL15]

Some other examples of time-series forecasting models [MJK15]

- Autoregressive integrated moving average (ARIMA)
- Seasonal ARIMA (SARIMA)
- Fractional ARIMA (FARIMA)

Forecasting using frequency domain representation

- Transform the signal to the frequency domain (e.g. using Fourier transform)
- Remove insignificant high-frequency components
- Forecast for each remaining component
- Transform the signal back to the time domain

Time-series classification

Problem: Assign class labels to $x_i \ldots x_{i+n}$

Figure: Classification of time-series data [LBKLT16]

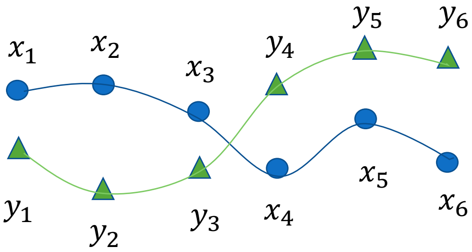
Time-series classification

- Represent time-series in a suitable domain
- Select a similarity measure
- Classification method (K-nearest neighbor is very popular)

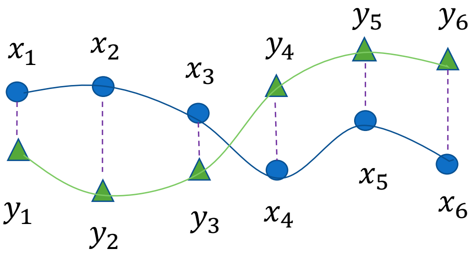
Representation and similarity measure go hand-in-hand and should be matched!

Similarity measure

How to measure similarity of two time-series to each other?

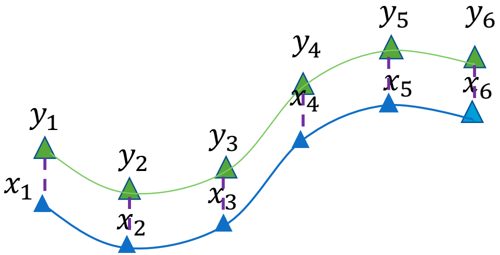


Euclidean distance



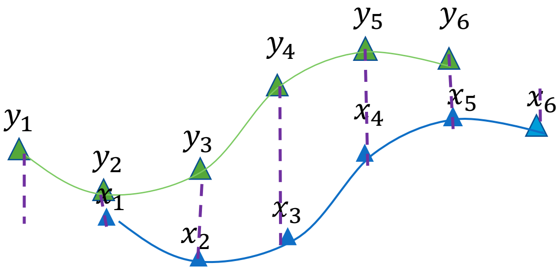
Euclidean distance

Very similar time-series



Euclidean distance

Very similar time-series (?)



What do we miss?

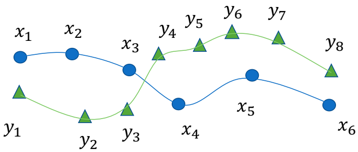
Euclidean distance:

- Sensitive to shifting, time or amplitude scaling

Dynamic time warping (DTW)

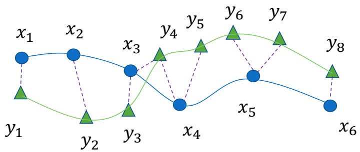
- DTW-algorithm is able to compare two curves in a way that makes sense to human. It maintains the importance of spots in curves that are important for humans when comparing curves.
- Elastic similarity measure
- The most used measure of similarity between time-series
- Works by finding the optimal **alignment** between two time-series
- Based on pair-wise distance matrix of time-series

DTW [CB17]

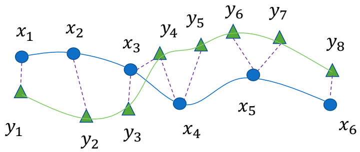


DTW

Intuition: finding the best matching pair of points on two time-series



DTW



	y_1	y_2	y_3	y_4	y_5	y_6	y_7	y_8
x_1	1	0	0	0	0	0	0	0
x_2	0	1	0	0	0	0	0	0
x_3	0	0	1	1	0	0	0	0
x_4	0	0	0	1	1	0	0	0
x_5	0	0	0	0	0	1	1	0
x_6	0	0	0	0	0	0	0	1

The goal of DTW is finding the best alignment path

Pair-wise distance matrix

- The matrix can be initialized from data, through recursion we find the optimal alignment
- $\Delta_{(i,j)}$ is $|x_i - y_j|$

$\Delta_{(1,1)}$	$\Delta_{(1,2)}$	$\Delta_{(1,3)}$	$\Delta_{(1,4)}$	$\Delta_{(1,5)}$	$\Delta_{(1,6)}$	$\Delta_{(1,7)}$	$\Delta_{(1,8)}$
$\Delta_{(2,1)}$	$\Delta_{(2,2)}$	$\Delta_{(2,3)}$	$\Delta_{(2,4)}$	$\Delta_{(2,5)}$	$\Delta_{(2,6)}$	$\Delta_{(2,7)}$	$\Delta_{(2,8)}$
$\Delta_{(3,1)}$	$\Delta_{(3,2)}$	$\Delta_{(3,3)}$	$\Delta_{(3,4)}$	$\Delta_{(3,5)}$	$\Delta_{(3,6)}$	$\Delta_{(3,7)}$	$\Delta_{(3,8)}$
$\Delta_{(4,1)}$	$\Delta_{(4,2)}$	$\Delta_{(4,3)}$	$\Delta_{(4,4)}$	$\Delta_{(4,5)}$	$\Delta_{(4,6)}$	$\Delta_{(4,7)}$	$\Delta_{(4,8)}$
$\Delta_{(5,1)}$	$\Delta_{(5,2)}$	$\Delta_{(5,3)}$	$\Delta_{(5,4)}$	$\Delta_{(5,5)}$	$\Delta_{(5,6)}$	$\Delta_{(5,7)}$	$\Delta_{(5,8)}$
$\Delta_{(6,1)}$	$\Delta_{(6,2)}$	$\Delta_{(6,3)}$	$\Delta_{(6,4)}$	$\Delta_{(6,5)}$	$\Delta_{(6,6)}$	$\Delta_{(6,7)}$	$\Delta_{(6,8)}$

$$dtw(i,j) =$$
$$\Delta_{i,j} + min(dtw(i-1,j-1), dtw(i-1,j), dtw(i,j-1))$$

A recursive process

Finding the best alignment path is achieved through recursion using the pairwise distance matrix

$dtw(i, j) =$
$\Delta_{i,j} + min(dtw(i-1, j-1), dtw(i-1, j), dtw(i, j-1))$

Other similarity measures

- Least Common Subsequence (LCSS)
- Edit Distance on Real sequence (EDR)
- ...

End of theory!

Part 3: Assignment

References I

George EP Box, Gwilym M Jenkins, Gregory C Reinsel, and Greta M Ljung, *Time series analysis: forecasting and control*, John Wiley & Sons, 2015.

Mitra Baratchi, Nirvana Meratnia, and Paul J. M. Havinga, *Recognition of periodic behavioral patterns from streaming mobility data*, Mobile and Ubiquitous Systems: Computing, Networking, and Services (Cham) (Ivan Stojmenovic, Zixue Cheng, and Song Guo, eds.), Springer International Publishing, 2014, pp. 102–115.

Marco Cuturi and Mathieu Blondel, *Soft-dtw: a differentiable loss function for time-series*, arXiv preprint arXiv:1703.01541 (2017).

References II

- Daoyuan Li, Tegawendé F Bissyandé, Jacques Klein, and Yves Le Traon, *Dsco-ng: A practical language modeling approach for time series classification*, International Symposium on Intelligent Data Analysis, Springer, 2016, pp. 1–13.
- Zhenhui Li, Bolin Ding, Jiawei Han, Roland Kays, and Peter Nye, *Mining periodic behaviors for moving objects*, Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining, ACM, 2010, pp. 1099–1108.
- Douglas C Montgomery, Cheryl L Jennings, and Murat Kulahci, *Introduction to time series analysis and forecasting*, John Wiley & Sons, 2015.

References III

- Andreea-Cristina Petre, Cristian Chilipirea, Mitra Baratchi, Ciprian Dobre, and Maarten van Steen, *Chapter 14 - wifi tracking of pedestrian behavior*, Smart Sensors Networks, Intelligent Data-Centric Systems, 2017, pp. 309 – 337.
- Xiaoyue Wang, Abdullah Mueen, Hui Ding, Goce Trajcevski, Peter Scheuermann, and Eamonn Keogh, *Experimental comparison of representation methods and distance measures for time series data*, Data Mining and Knowledge Discovery **26** (2013), no. 2, 275–309.