Machine Learning for Time Series

(MLTS or MLTS-Deluxe Lectures)

Dr. Dario Zanca

Machine Learning and Data Analytics (MaD) Lab

Friedrich-Alexander-Universität Erlangen-Nürnberg

09.01.2023

Topics overview

- Time series fundamentals and definitions (2 lectures)
- Bayesian Inference (1 lecture)
- Gussian processes (2 lectures)
- State space models (2 lectures)
- Autoregressive models (1 lecture)
- Data mining on time series (1 lecture) ←
- Deep learning on time series (4 lectures)
- Domain adaptation (1 lecture)

In this lecture…

1. **Introduction to Data Mining**
2. **Frequency analsys**
3. **Dynamic time warping**
4. **Feature extraction techniques**

Data Mining with Time Series
Data Mining

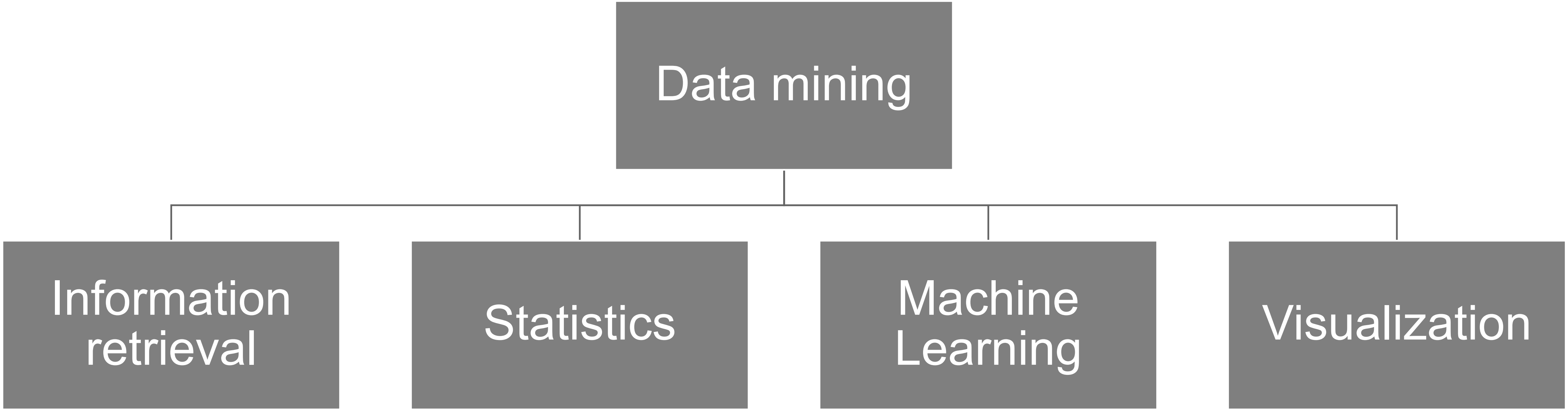
What is Data mining?

Definition 1. Extraction of non-simple, implicit, previously unknown, possibly useful data from database.

Definition 2. Automatic or semi-automatic search and analysis of a large amount of data with the goal of discovering significan patterns.

Data mining is useful when the information is hidden due to large amount of data, its complexity, heterogeneity, the speed at which it is collected and need for non-tradition queries.

What is Data mining?



Data mining

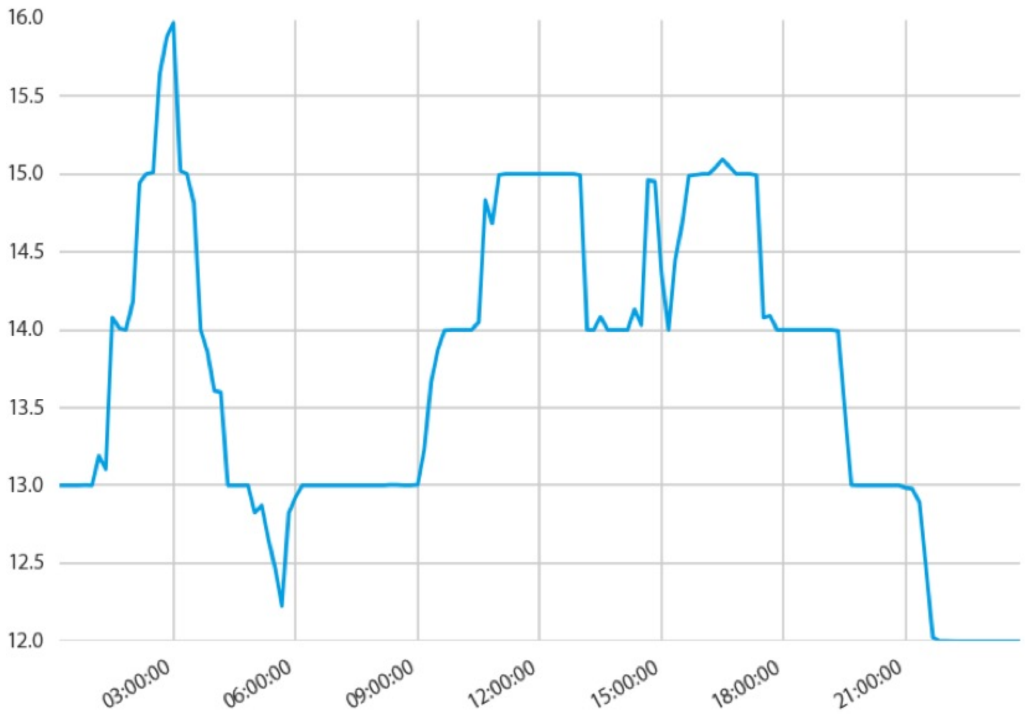
The most basic approach for data mining can include:

- **Compute data characteristics**
- **Data reduction**
- **Data transformation**

Compute data characteristics

Suppose we have a time series of ambient temperature recorded during one day.

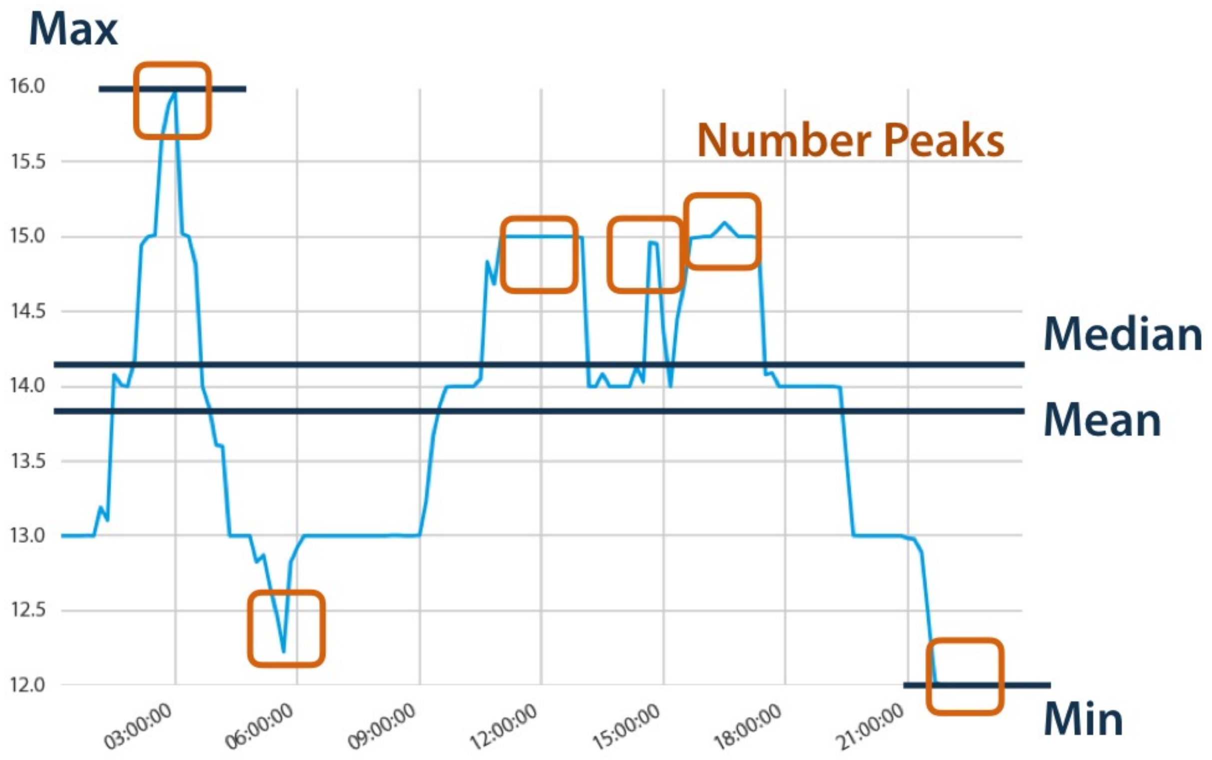
A characterization of this time series can be given by a set of **basic statistics**.



Compute data characteristics

Suppose we have a time series of ambient temperature recorded during one day.

A characterization of this time series can be given by a set of **basic statistics**.



Data reduction

When dealing with big dataset, the application of **data reduction techniques** is required in order to allow reasoning on smaller dimensional spaces.

→ Efficiency

→ Interpretability

→ Simplicity

There are three main ways to reduce data size:

- **Sampling.** Reducing the number of observations.
- **Selection and projection.** Reducing the number of features.
- **Discretization and aggregation.** Reduction of the number of possible values.

Data transformation

In most applications, it is necessary to **apply transformations to our data** to make it usable for further analysis.

→ Most machine learning algorithms perform better if data has a consistent scale distribution.

→ E.g., data is heterogenous because of different data sources (units of measure, sampling rates, data types).

Different data transformations:

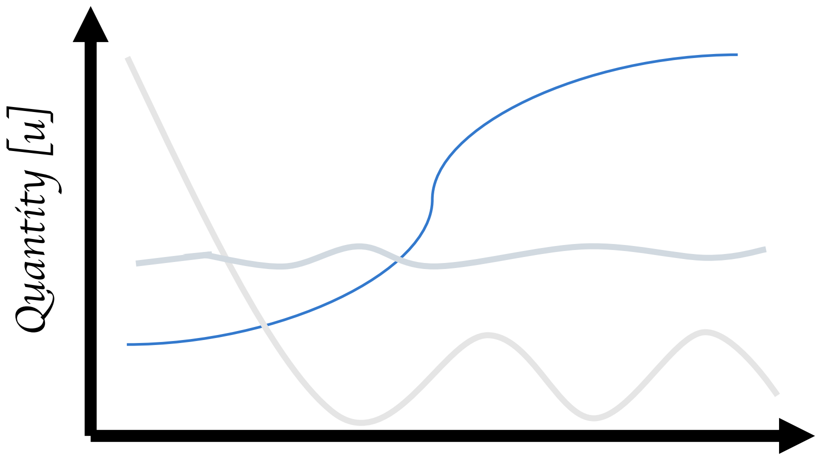
- Normalization
- Standardization

Data transformation: Normalization

Feature normalization is used to normalize the range of values of features.

Methods of data normalization:

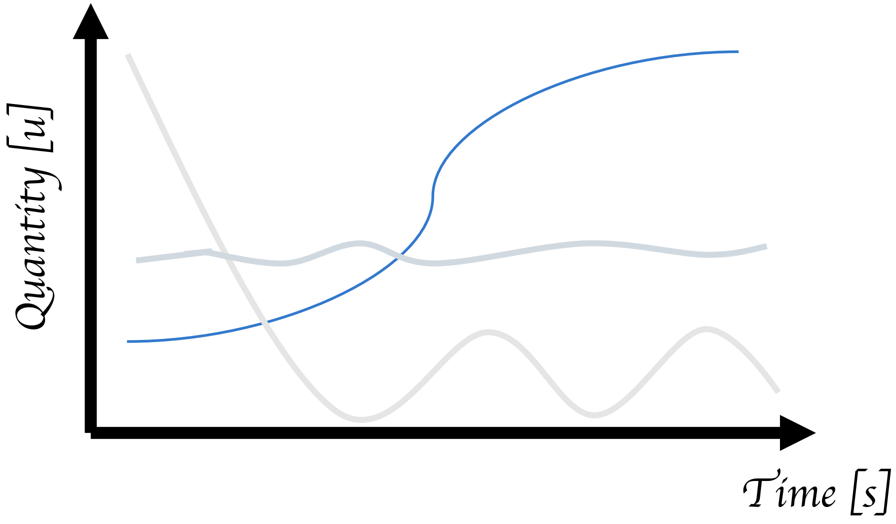
- Mean normalization
- Min-Max normalization



Data transformation: Mean Normalization

Let $S = (s_1, \dots, s_T)$ be a multivariate time series, $s_i \in \mathbb{R}^d$ is a d-dimenstional observation at time t_i.

We denote with $\boldsymbol{S_j} = (\boldsymbol{s_{1j}}, \dots, \boldsymbol{s_{Tj}})$ the j-th feature of the time series S, where $s_{ij} \in \mathbb{R}$ is an observation of the j-th feature at time t_i.



Mean normalization is defined by:

$$s'_{ij} = \frac{s_{ij} - \mu_j}{\mathrm{s}_{\max,j} - \mathrm{s}_{\min,j}}$$

where $\mathrm{s}_{\max,j}$ and $\mathrm{s}_{\min,j}$ are the max and min values of $\boldsymbol{S_j}$,

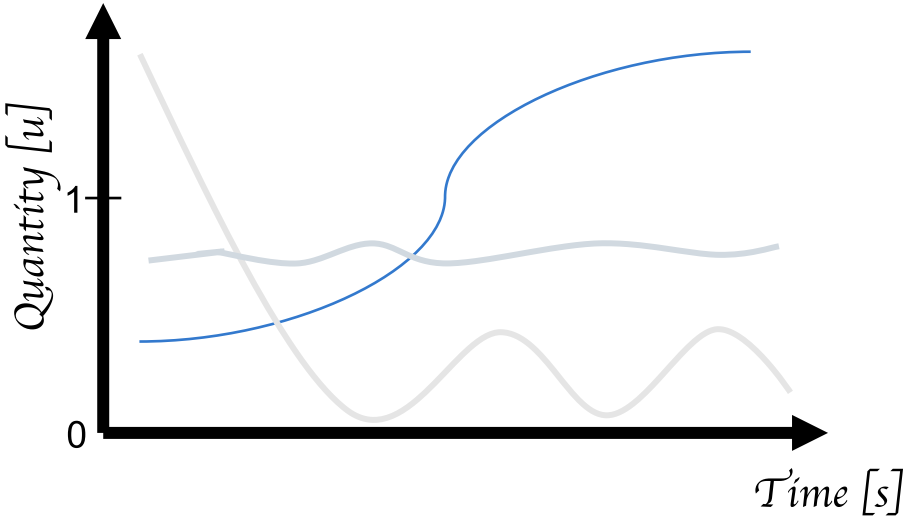
and $\mu_j = \frac{1}{T}\sum_{i=1}^{T} s_{ij}$.

Data transformation: Min-Max Normalization

Min-Max normalization is defined by:

$$s'_{ij} = \frac{s_{ij} - \mathrm{s}_{\mathrm{min,j}}}{\mathrm{s}_{\mathrm{max,j}} - \mathrm{s}_{\mathrm{min,j}}}$$

where $\mathrm{s}_{\mathrm{max,j}}$ and $\mathrm{s}_{\mathrm{min,j}}$ are the max and min values of $\boldsymbol{S_j}$.



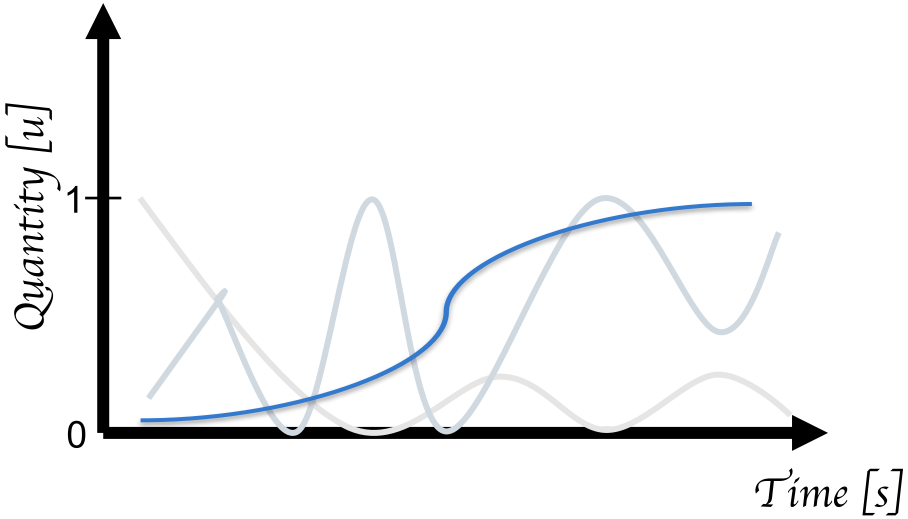
Data transformation: Min-Max Normalization

Min-Max normalization is defined by:

$$s'_{ij} = \frac{s_{ij} - \mathrm{s}_{\mathrm{min,j}}}{\mathrm{s}_{\mathrm{max,j}} - \mathrm{s}_{\mathrm{min,j}}}$$

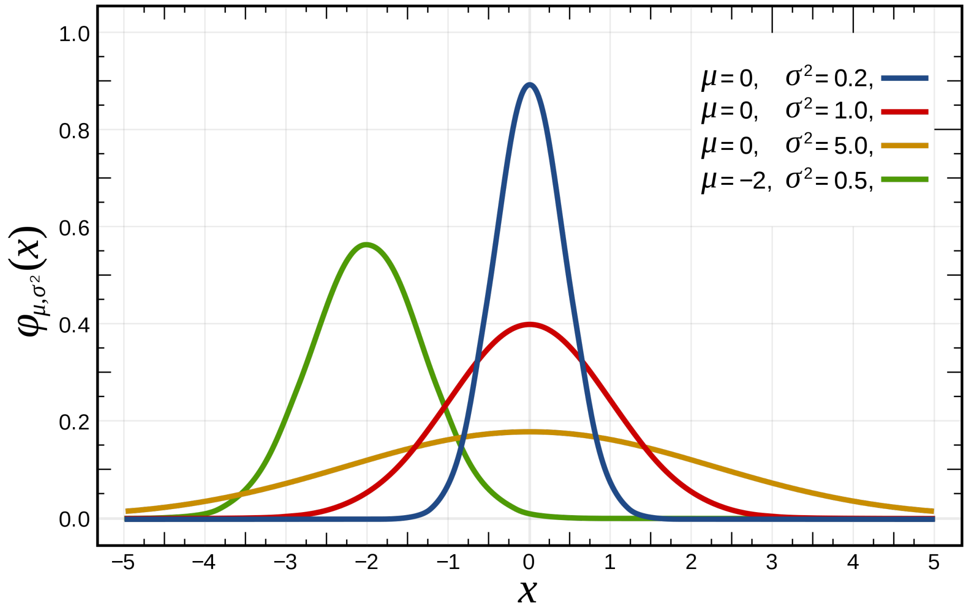
where $\mathrm{s}_{\mathrm{max,j}}$ and $\mathrm{s}_{\mathrm{min,j}}$ are the max and min values of $\boldsymbol{S_j}$.

This normalization establish two boundaries for the data between 0 and 1.



Data transformation: Standardization

Standardization is the process of converting a random variable with mean μ and standard deviation $\boldsymbol{\sigma}$ to a **“standard distribution”** (i.e., zero mean and unit standard deviation).



https://en.wikipedia.org/wiki/Normal_distribution

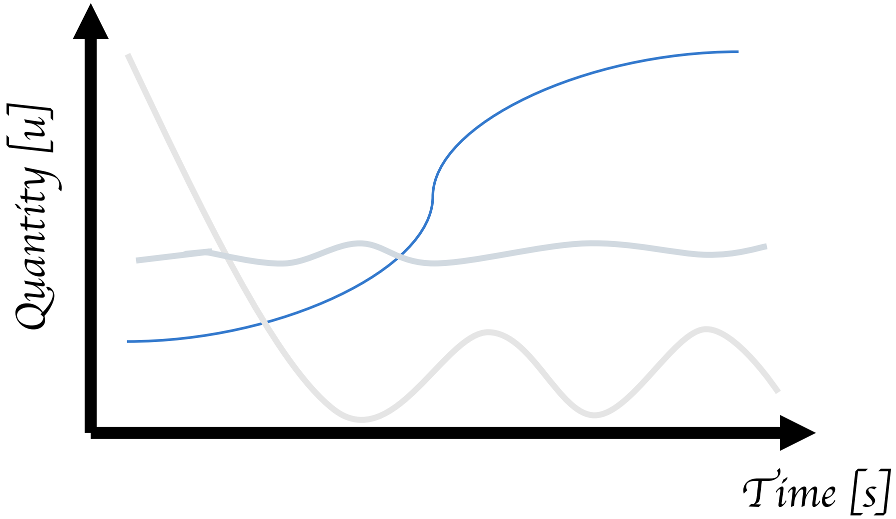
The **standard score** is the number of standard deviations by which the value of a raw score (i.e., an observed value or data point) is above or below the mean value of what is being observed or measured.

Data transformation: Z-score standardization

Z-score standardization is defined by:

$$s'_{ij} = \frac{s_{ij} - \mu_j}{\sigma_j}$$

where $\mathrm{s}_{\mathrm{max,j}}$ and $\mathrm{s}_{\mathrm{min,j}}$ are the max and min values of $\boldsymbol{S_j}$, and μ_j and σ_j are the feature mean and standard deviation.



- The transformed data will have zero mean and unit standard deviation.
- It is empirically shown that, if the original distribution is Gaussian, z-score based transformation generates values that are in the range $(-3, 3)$.

Data Mining with Time Series

Frequency analysis

Spectral analysis

Spectral analysis is a technique that allows us to discover underlying periodicities.

- Many time series show periodic behavior.
- This periodic behavior can be very complex.
- Additional tool to analyse time series (complementary to the time-domain analysis)

To perform spectral analysis, we first must transform data from time domain to frequency domain.

→ We use Fourier transform to convert the time series from the time-domain to the frequency domain

http://web.stanford.edu/class/earthsys214/notes/series.html#spectral-analysis

Fourier representation

Given a time series $S = (x_1, \ldots, x_N)$, the goal of spectral analysis is that of determining how to construct it using sines and cosines, i.e.,

$$S = \sum_k a_k \sin(2\pi \frac{k}{N} t) + b_k \cos\left(2\pi \frac{k}{N} t\right)$$

The above expression is called **Fourier representation** for a time series.

→ It allows us to re-express time series in a standard way

→ Different time series are characterized by different coeffients

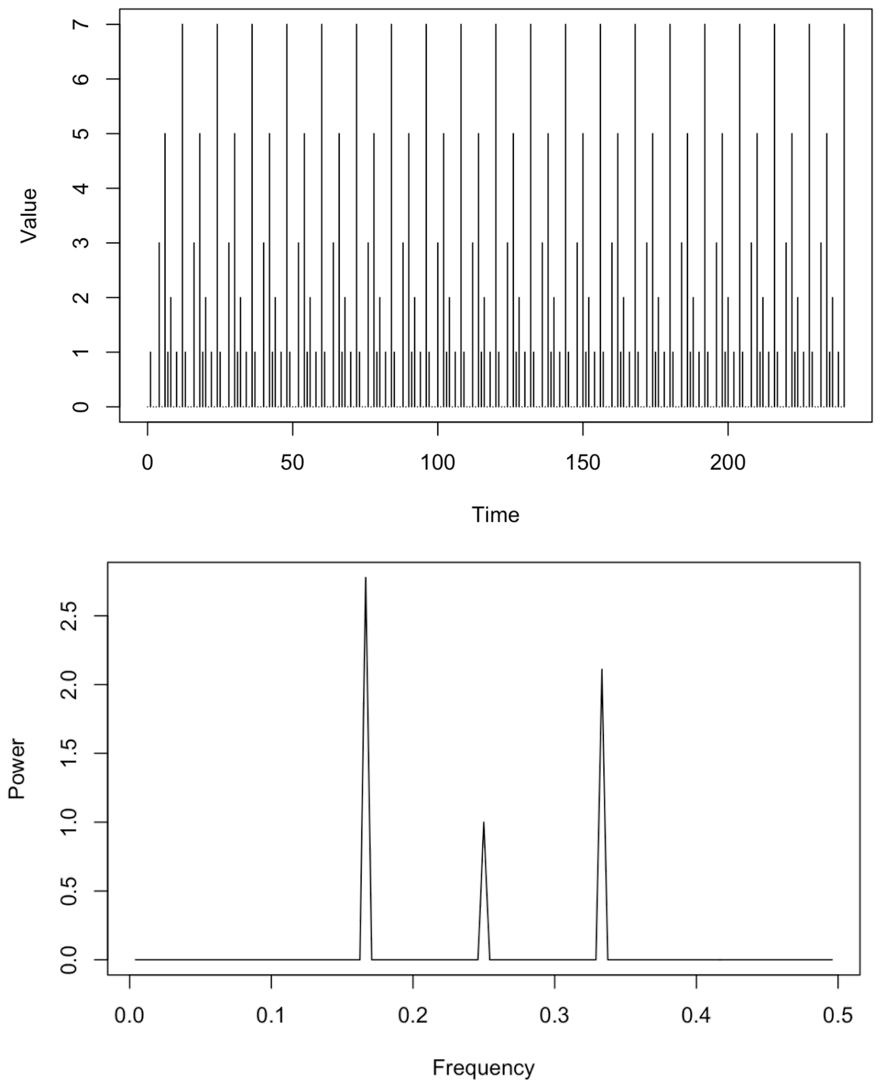
→ a_k's and b_k's can be determined in a closed form

→ We can compare time series by comparing their coefficients

Spectral density

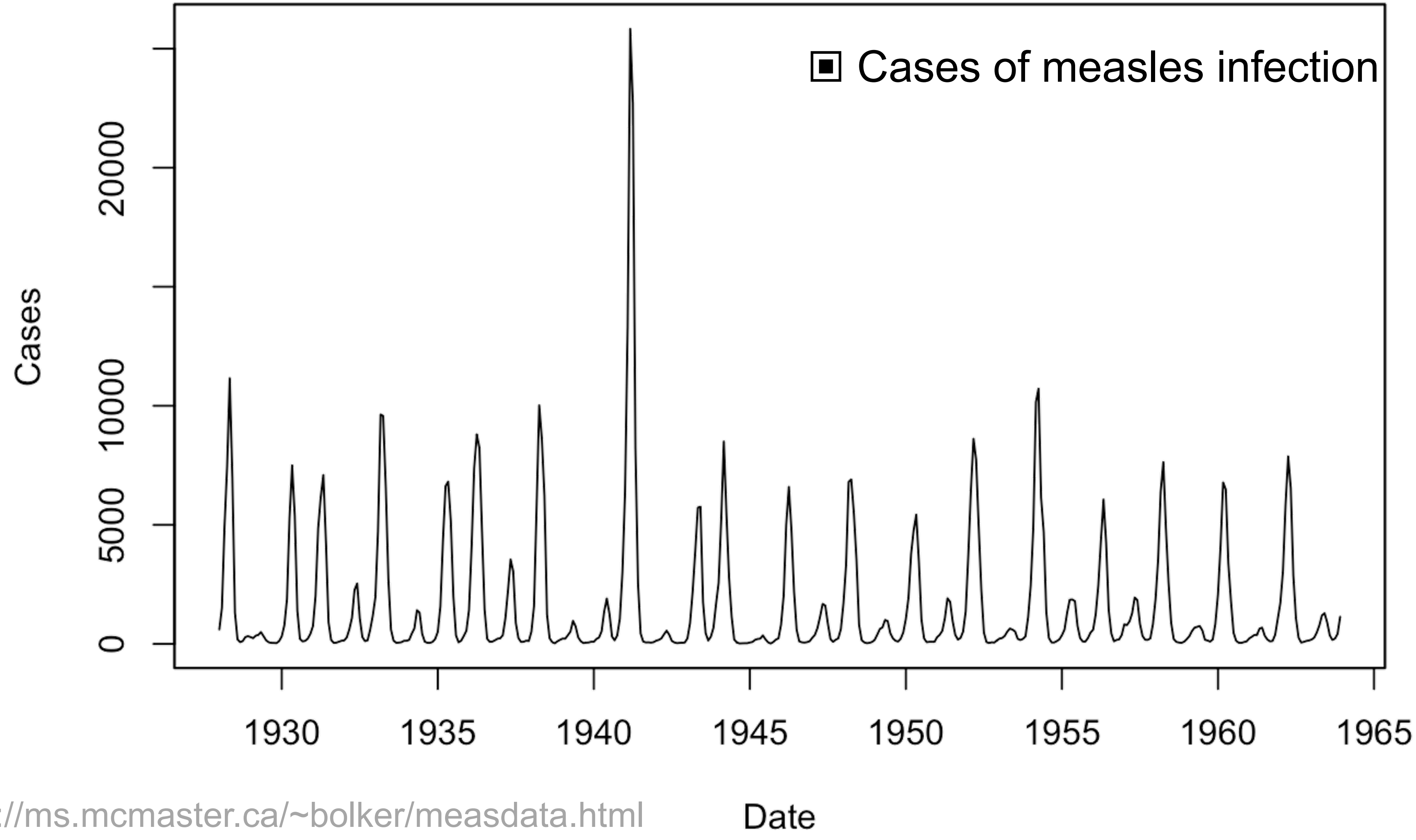
In brief, the covariance of the time series can be represented by a function known as the **spectral density**.

The spectral density can be estimated using on object known as a **periodogram**, which is the squared correlation between our time series and sine/cosine waves at the different frequencies spanned by the series.



http://web.stanford.edu/class/earthsys214/notes/series.html#spectral-analysis

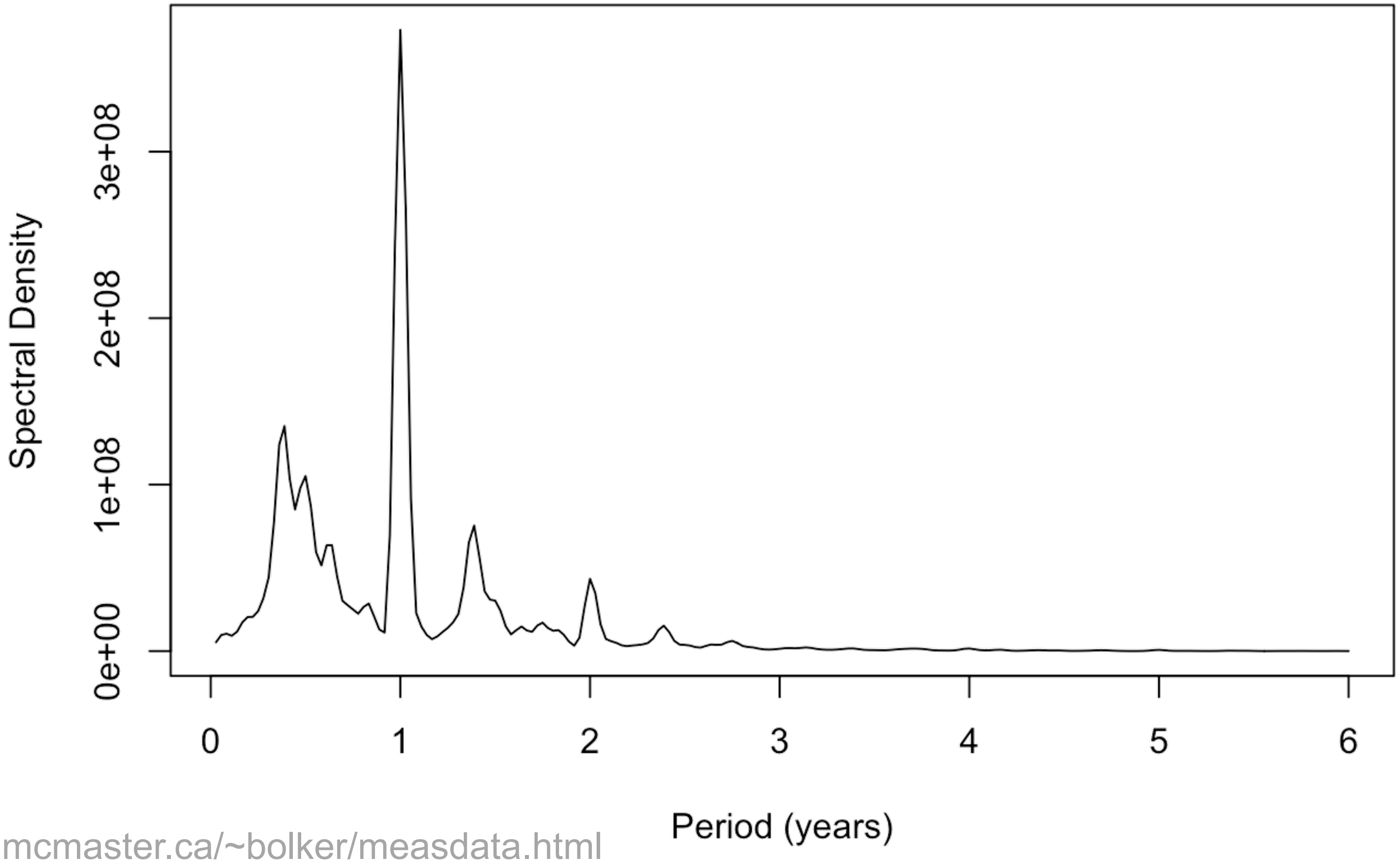
Example: spectral analysis



Data from: https://ms.mcmaster.ca/~bolker/measdata.html

Images from: http://web.stanford.edu/class/earthsys214/notes/series.html#spectral-analysis

Example: spectral analysis



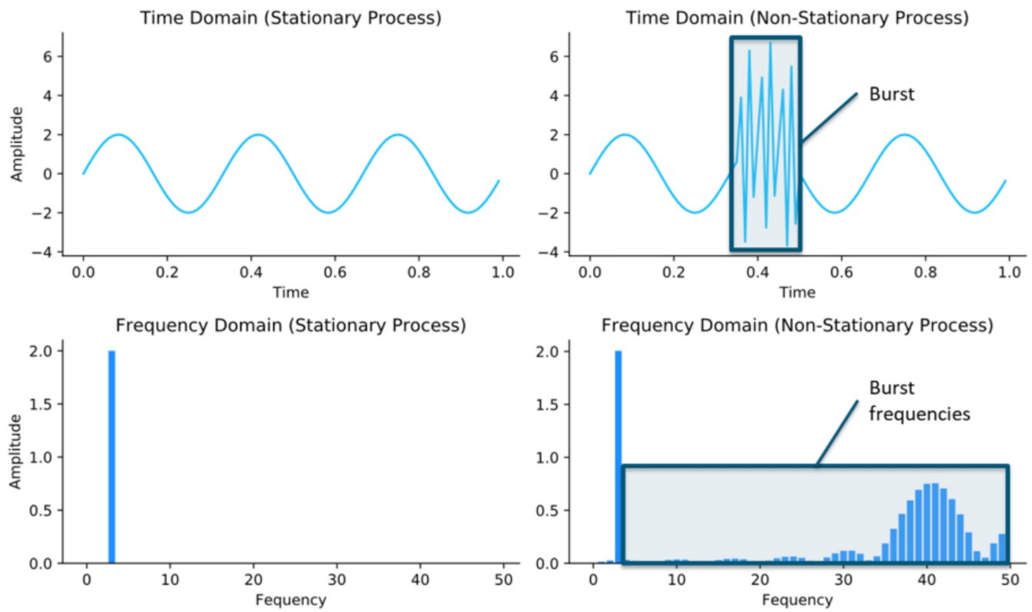
Data from: https://ms.mcmaster.ca/~bolker/measdata.html
Images from: http://web.stanford.edu/class/earthsys214/notes/series.html#spectral-analysis

Limitations of the Fourier transform

Fourier transform works well for sines/cosines waves which are generated by stationary signals.

For a non-stationary signal (e.g., containing an burst /anomaly) by performing a Fourier transform we obtain frequencies that construct the signal but we cannot identify which of them represents the burst/anomaly.

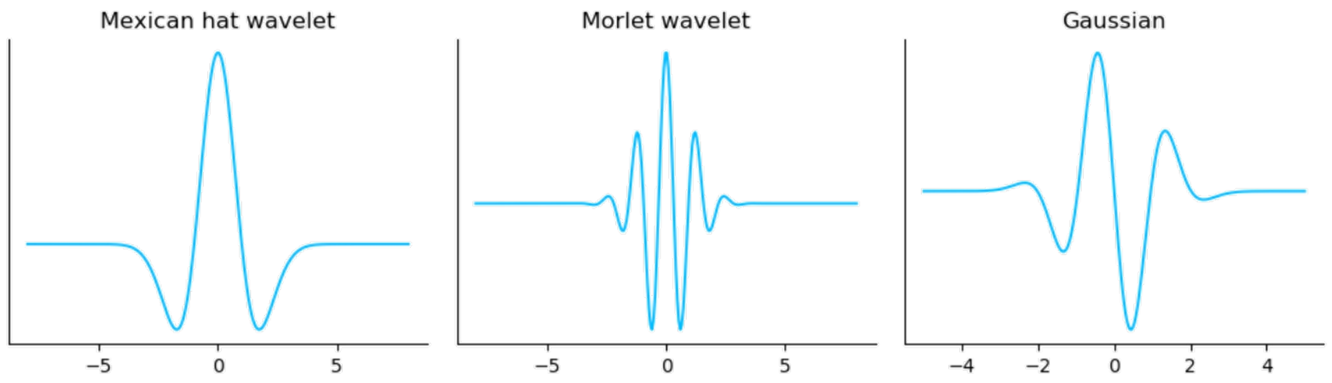
→ With non-stationary signlas we are confined to either time or frequency domain.



https://towardsdatascience.com/multiple-time-series-classification-by-using-continuous-wavelet-transformation-d29df97c0442

Continuous Wavelet Transform

Continuous Wavelet Transform (CWT) is based on the concept of **wavelets** (or mini wavelets).



- In contrast to sines/cosines used in Fourier transform, wavelets are limited (in time)
- Have zero mean.

Both this conditions allow a localization in time and frequency at the same time.

Continuous Wavelet Transform

Continuous Wavelet Transform (CWT) is defined as follows:

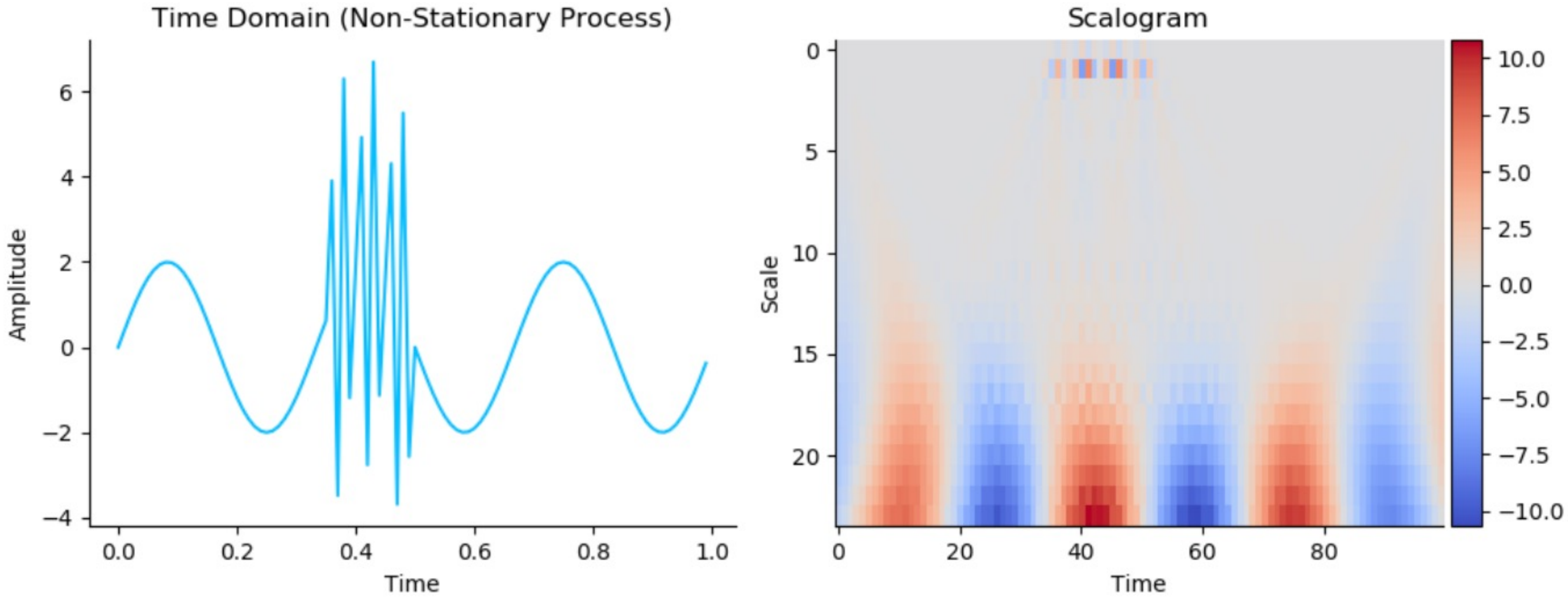
$$cwt(\tau, s) = \frac{1}{\sqrt{|s|}} \int_{-\infty}^{+\infty} x(t)\psi\left(\frac{t-\tau}{s}\right) dt$$

where τ is the translation, s is the scale, ψ is the mother wavelet.

- The CWT is based on the concepts of scaling and shifting.
- It transform the original 1-D time series to a N-D ”image”.

Example: Continuous Wavelet Transform

If we apply CWT to the previous non-stationary signal, we obtain:



https://towardsdatascience.com/multiple-time-series-classification-by-using-continuous-wavelet-transformation-d29df97c0442

Continuous Wavelet Transform

Continuous Wavelet Transform (CWT) are:

- Efficient in determining the damping ratio of oscillating signals (e.g. identification of damping in dynamic systems)
- Robust to the noise in the signal
- 2D scalogram can be used to improve the distinction between varying types of a signal. E.g.,
    - differentiate between different production processes in a machine
    - identifying components or tools faults

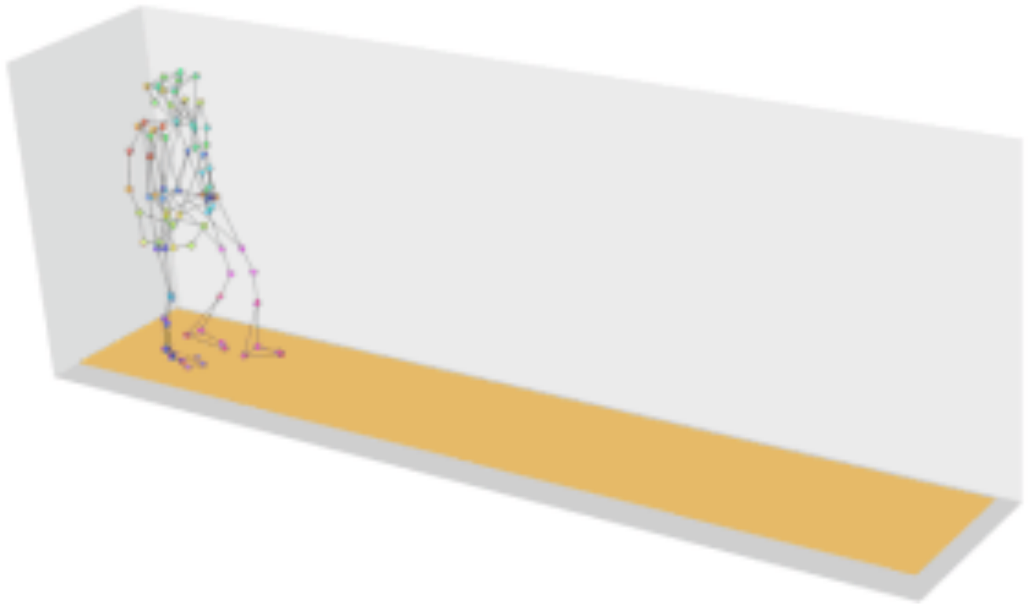
Data Mining with Time Series

Dynamic Time Warping

Dynamic time warping

In many application, there is the need to analyse multiple time series at the same time to find similarities between time series.

- E.g., speech recognition, signature recognition, similarity in walking, …

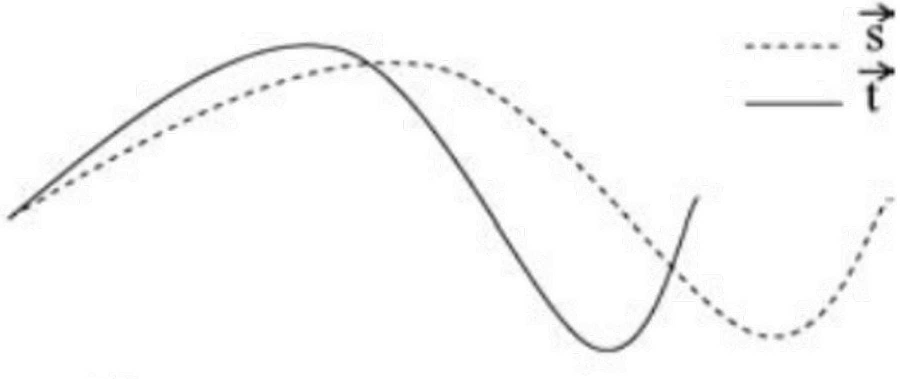


Euclidean distance does not work for time series that are not perfectly syncronized.

Dynamic time warping

Dynamic time warping (DTW) is an algorithm to measure similarity between two time-series that may **vary in speed and time**.

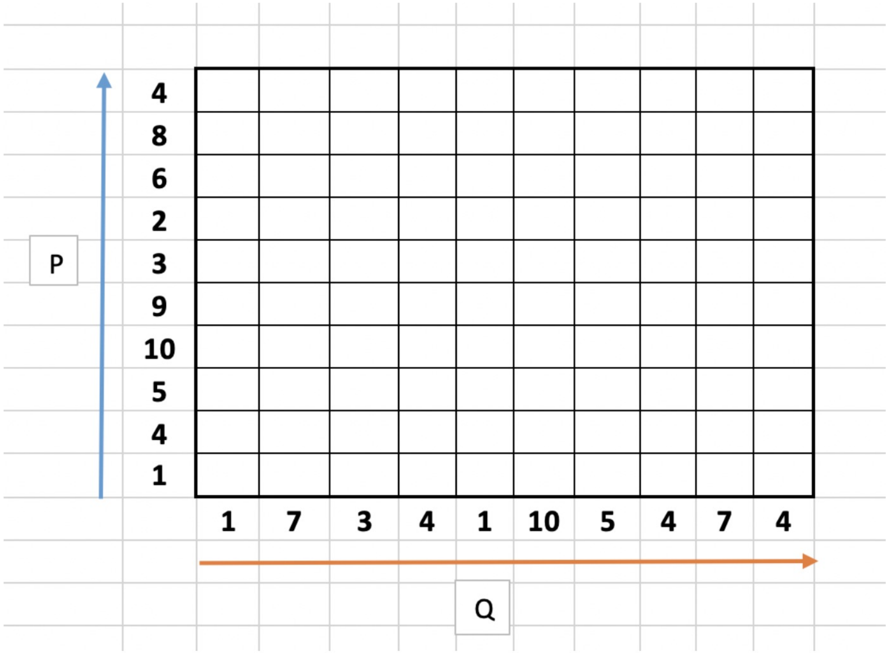
DTW determines the optimal global alignment between two time series.



Dynamic time warping: Algorithm

Given two time series x_t and y_t, we can compute the DTW distance as follows:

1. Initialize the distance matrix M



Dynamic time warping: Algorithm

Given two time series x_t and y_t, we can compute the DTW distance as follows:

1. Initialize the distance matrix M
2. Fill M from the bottom left corner, according to the formula:

$$M(i,j) = dist(x_i, y_j) + \min(M(i-1, j-1), M(i, j-1), M(i-1, j))$$

P										
4	42	24	-	-	-	-	-	-	-	-
8	39	21	-	-	-	-	-	-	-	-
6	32	20	-	-	-	-	-	-	-	-
2	27	19	-	-	-	-	-	-	-	-
3	26	14	-	-	-	-	-	-	-	-
9	24	10	-	-	-	-	-	-	-	-
10	16	8	12	**11**	-	-	-	-	-	-
5	7	5	5	5	-	-	-	-	-	-
4	**3**	3	4	4	7	13	14	14	17	17
1	0	6	**8**	11	11	20	24	27	33	36
	1	**7**	**3**	**4**	**1**	**10**	**5**	**4**	**7**	**4**

Q

Please notice:

- $M(i,j)$ denotes the cell of the i-th row and j-th column in the M matrix, starting from the bottom left.
- Near to the axis, $M(i-1, j-1)$, $M(i, j-1)$ and $M(i-1, j)$ fall "outside" the matrix M and are considered as "0" in the equation.

Dynamic time warping: Algorithm

Given two time series x_t and y_t, we can compute the DTW distance as follows:

1. Initialize the distance matrix M
2. Fill M from the bottom left corner, according to the formula:
   $M(i,j) = dist(x_i, y_j) + \min(M(i-1, j-1), M(i, j-1), M(i-1, j))$
3. Identify the **warping path** d, starting from the top right corner

P

4	42	24	20	17	20	25	19	18	18	**15**
8	39	21	19	17	20	19	18	19	**15**	18
6	32	20	14	13	17	18	15	15	**14**	16
2	27	19	11	12	12	20	14	**13**	17	18
3	26	14	10	11	13	16	**11**	12	16	17
9	24	10	14	16	19	**9**	12	17	18	21
10	16	8	12	11	14	**8**	13	18	16	21
5	7	5	5	5	**8**	12	12	13	15	16
4	3	**3**	**4**	**4**	7	13	14	14	17	17
1	**0**	6	8	11	11	20	24	27	33	36
	1	**7**	**3**	**4**	**1**	**10**	**5**	**4**	**7**	**4**

Q

Dynamic time warping: Algorithm

Given two time series x_t and y_t, we can compute the DTW distance as follows:

1. Initialize the distance matrix M
2. Fill M from the bottom left corner, according to the formula:
   $M(i,j) = dist(x_i, y_j) + \min(M(i-1, j-1), M(i, j-1), M(i-1, j))$
3. Identify the **warping path** d, starting from the top right corner
4. The overall path cost can be calculated as

$$D = \sum_{i=1}^{k} dist(i_k, j_k)$$

P

4	42	24	20	17	20	25	19	18	18	**15**
8	39	21	19	17	20	19	18	19	**15**	18
6	32	20	14	13	17	18	15	15	**14**	16
2	27	19	11	12	12	20	14	**13**	17	18
3	26	14	10	11	13	16	**11**	12	16	17
9	24	10	14	16	19	**9**	12	17	18	21
10	16	8	12	11	14	**8**	13	18	16	21
5	7	5	5	5	**8**	12	12	13	15	16
4	3	**3**	**4**	**4**	7	13	14	14	17	17
1	**0**	6	8	11	11	20	24	27	33	36
	1	**7**	**3**	**4**	**1**	**10**	**5**	**4**	**7**	**4**

Q

Dynamic time warping: Pros and Cons

Pros:

- Exploit a non-linear distortion (in time) to find non-trivial similarity

Cons:

- Hi**gh computational cost.** Alternatives for computing the alignment path more efficiently have been presented.
- It needs the preparation of reliable reference templates for the set of words to be recognized.

Data Mining with Time Series

Other feature extraction methods for time series

Similarity join

One method to find anomalies and trends in time series is to perform a **similarity join**.

→ Compare pair of snippets in a time series

→ Retrieve all data pairs whose distances are smaller than a predefined threshold ϵ.

→ **Very easy to implement**

→ **Computationally expensive for moderately large collection of data**

Matrix profiles

Matrix profile (MP) is a data structure for time series analysis.

Advantages:

- MP is domain agnostic
- Efficient
- Provides exact solution
- Only requires a single parameter

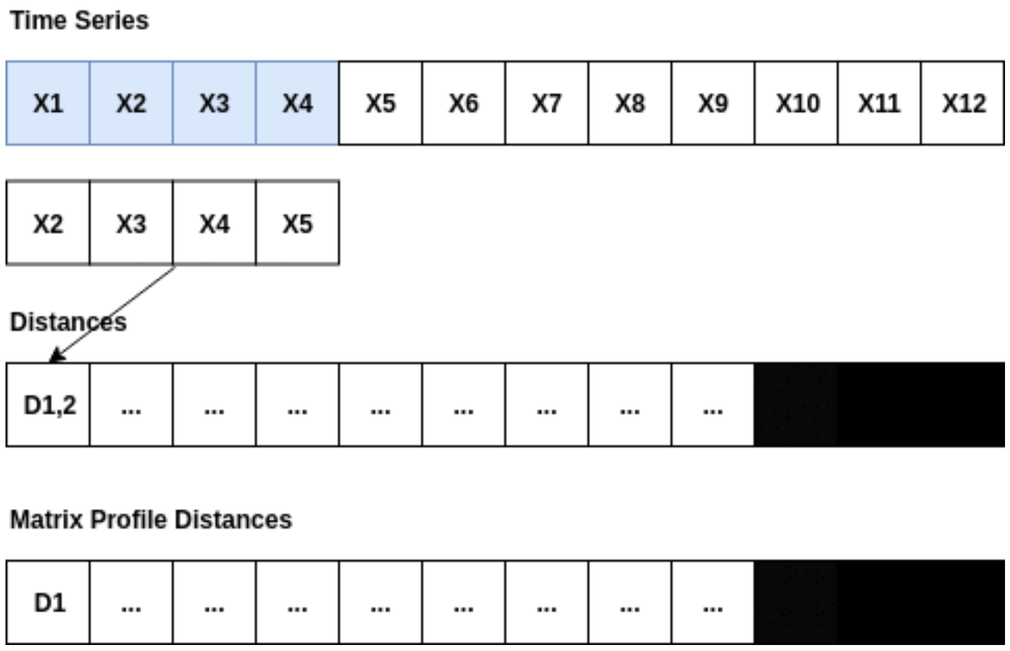
Matrix profiles

The Matrix Profile has two primary components:

- **distance profile**: a vector of minimum Z-Normalized Euclidean distances
- **profile index**: it contains the index of its first nearest-neighbor

The algorithms that compute the Matrix Profile use a **sliding window** approach.

Let m be the window size, and n be the time series lenght.

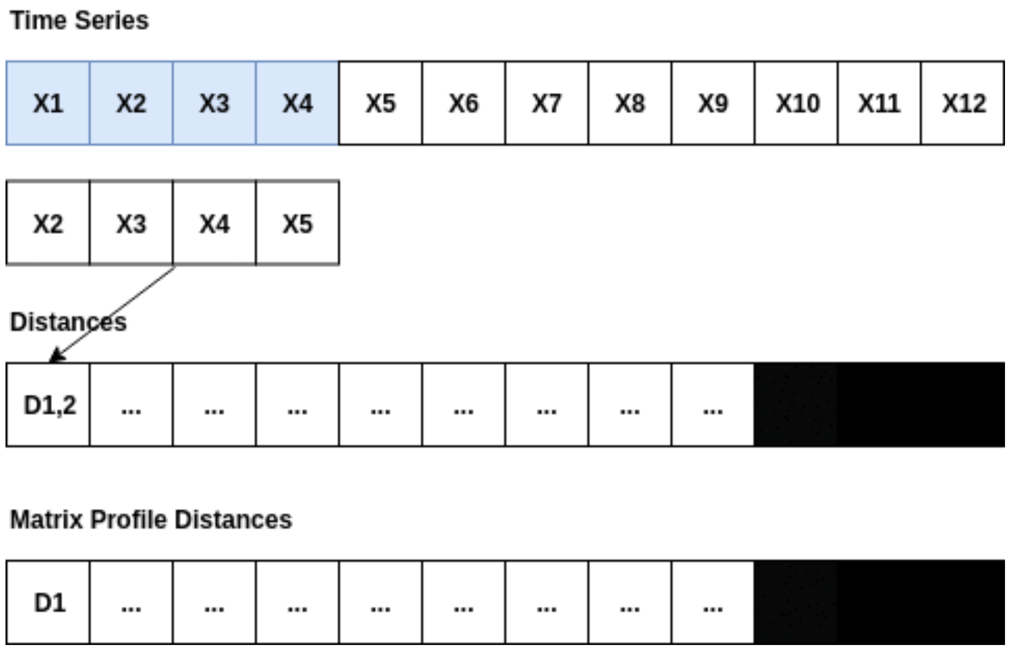


https://towardsdatascience.com/introduction-to-matrix-profiles-5568f3375d90
Original paper: All Pairs Similarity Joins for Time Series: A Unifying View that Includes Motifs, Discords and Shapelets. Chin-Chia Michael Yeh, Yan Zhu, Liudmila Ulanova, Nurjahan Begum, Yifei Ding, Hoang Anh Dau, Diego Furtado Silva, Abdullah Mueen, Eamonn Keogh (2016). IEEE ICDM 2016.

Matrix profiles: the algorithm

The general algorithm:

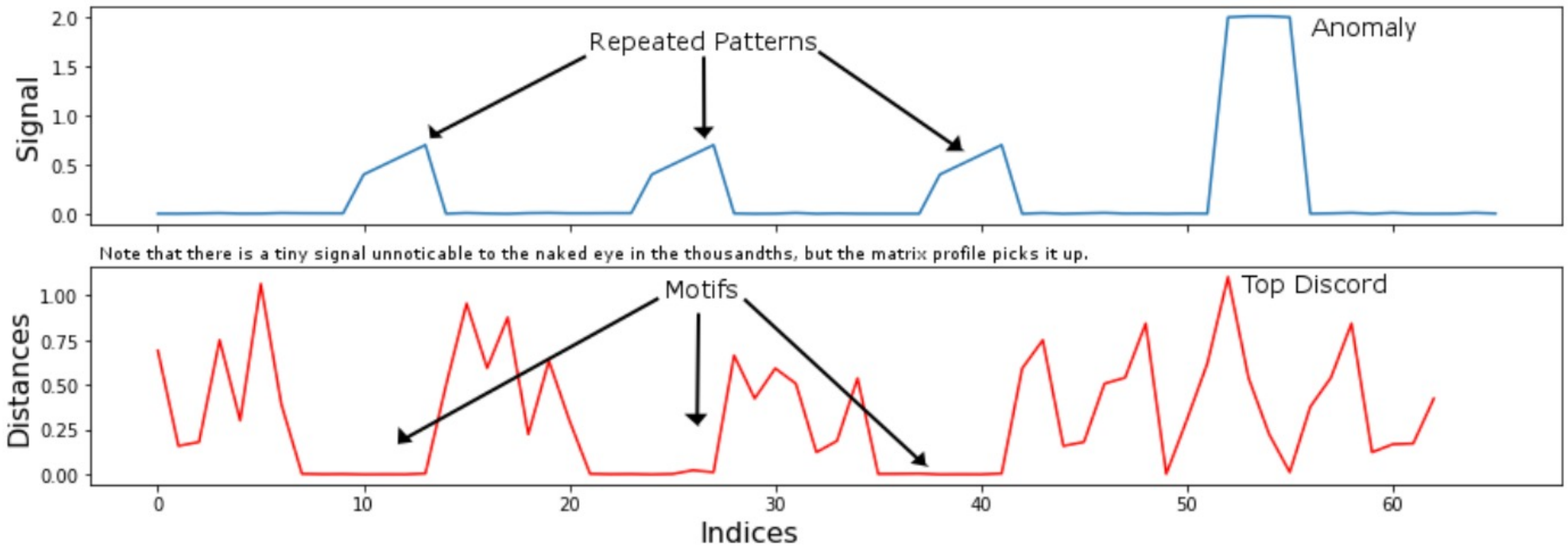
1. Compute the distances for the windowed sub-sequence against the entire time series
2. Set an exclusion zone to ignore trivial matches
3. Updates the distance profile with the minimal values
4. Set the first nearest-neighbor index



https://towardsdatascience.com/introduction-to-matrix-profiles-5568f3375d90
Original paper: All Pairs Similarity Joins for Time Series: A Unifying View that Includes Motifs, Discords and Shapelets. Chin-Chia Michael Yeh, Yan Zhu, Liudmila Ulanova, Nurjahan Begum, Yifei Ding, Hoang Anh Dau, Diego Furtado Silva, Abdullah Mueen, Eamonn Keogh (2016). IEEE ICDM 2016.

Matrix profiles: motifs and discords

When the matrix profile is computed, it is possible to identify **motifs (repeated patterns)** and **discords (anomalies)** in a time series.



https://towardsdatascience.com/introduction-to-matrix-profiles-5568f3375d90
Original paper: All Pairs Similarity Joins for Time Series: A Unifying View that Includes Motifs, Discords and Shapelets. Chin-Chia Michael Yeh, Yan Zhu, Liudmila Ulanova, Nurjahan Begum, Yifei Ding, Hoang Anh Dau, Diego Furtado Silva, Abdullah Mueen, Eamonn Keogh (2016). IEEE ICDM 2016.

Signature method

The **signature method** refers to a collection of feature extraction techniques for multivariate time series, derived from the theory of controlled differential equations.

→ Successfully applied in a wide range of ML tasks with sequential, e.g., chinese character recognition or feature extraction from financial data streams.

Given a multivariate time series $X: [a, b] \to \mathbb{R}^d$ (or more generally called **path** in this context), we define the increment of the i-th coordinate of the path as:

$$S(X)^i_{a,t} = \int_{a<s<t} dX^i_s = X^i_t - X^0_t$$

https://quantdare.com/an-introduction-to-time-series-signatures/
Original paper: Ilya Chevyrev and Andrey Kormilitzin. A primer on the signature method in machine learning. arXiv preprint arXiv:1603.03788, 2016.

Signature method

For every $(i,j) \in \{1,\ldots,d\}^2$ we define the double iterated integral:

$$S(X)^{i,j}_{a,t} = \int_{a<s<t} S(X)^{i}_{a,s}\, dX^{j}_{s} = \int_{a<r<s<t} dX^{i}_{r} dX^{j}_{s}$$

Similarly, we can define the **triple-iterated integral**:

$$S(X)^{i,j,k}_{a,t} = \int_{a<s<t} S(X)^{i,j}_{a,s}\, dX^{k}_{s}$$

And, recursively, we can construct **the k-fold iterated integral**:

$$S(X)^{i_1,\ldots,i_k}_{a,t} = \int_{a<s<t} S(X)^{i_1,\ldots,i_{k-1}}_{a,s}\, dX^{i_k}_{s}$$

https://quantdare.com/an-introduction-to-time-series-signatures/

Original paper: Ilya Chevyrev and Andrey Kormilitzin. A primer on the signature method in machine learning. arXiv preprint arXiv:1603.03788, 2016.

Signature method

We define signature of a time series (or, path) the infinite series of all the iterated integrals, defined by:

$$S(X)_{a,b} = (1, S(X)^1_{a,b}, \dots, S(X)^d_{a,b}, S(X)^{1,1}_{a,b}, S(X)^{1,2}_{a,b}, \dots)$$

where the superscripts vary within the set of all multi-indexes

$$W = \{(i_1, \dots, i_k) | k \geq 1; i_1, \dots, i_k \in \{1, \dots, d\}\}$$

https://quantdare.com/an-introduction-to-time-series-signatures/
Original paper: Ilya Chevyrev and Andrey Kormilitzin. A primer on the signature method in machine learning. arXiv preprint arXiv:1603.03788, 2016.

Data Mining with Time Series

Recap

In this lecture

- Introduction to Data Mining
- Frequency analsys
- Dynamic time warping
- Feature extraction techniques

Python resources

- **Tsfresh**: python library which is used to to extract characteristics from time series.
  https://tsfresh.readthedocs.io/en/latest/

Python resources

- **Tsfresh** is a Python library which is used to to extract characteristics from time series.
  https://tsfresh.readthedocs.io/en/latest/
- **TSFEL** is a Python package for feature extraction on time series data.
  https://tsfel.readthedocs.io/en/latest/

Python resources

- **Tsfresh** is a Python library which is used to to extract characteristics from time series.
  https://tsfresh.readthedocs.io/en/latest/
- **TSFEL** is a Python package for feature extraction on time series data.
  https://tsfel.readthedocs.io/en/latest/
- **A systematic review of Python packages for time series**, Siebert et al.
  Paper: https://arxiv.org/abs/2104.07406
  Website: https://siebert-julien.github.io/time-series-analysis-python/

	Tasks								Data Preparation			
Name	forecasting	classification	clustering	anomaly detection	segmentation	pattern recognition	change point detection	dimensionality reduction	missing values imputation	decomposition	preprocessing	simi mea
arch	true	false	false	false	false	false	false	false	false	false	false	false
atspy	true	false	false	false	false	false	false	false	true	true	true	false
banpei	false	false	false	true	false	false	true	false	false	false	false	false
cesium	false	false	false	false	false	false	false	false	false	false	true	false
darts	true	false	false	false	false	false	false	false	true	true	true	false
deeptime	true	false	true	false	false	false	false	true	false	true	true	false
deltapy	true	false	true	false	false	false	false	true	false	true	true	true
dtaidistance	false	false	true	false	false	false	false	false	false	false	false	true
EMD-signal	false	false	false	false	false	false	false	false	false	true	false	false
flood-forecast	true	false	false	false	false	false	false	false	true	false	true	false
name	forecasting	classification	clustering	anomaly detection	segmentation	pattern recognition	change point detection	dimensionality reduction	missing values imputation	decomposition	preprocessing	simi mea