Christian-Albrechts-Universität zu Kiel

Lecture „Intelligent Systems“

Chapter 3: Pre-processing

Prof. Dr.-Ing. habil. Sven Tomforde / Intelligent Systems

Winter term 2020/2021

About this chapter

Christian-Albrechts-Universität zu Kiel

Contents

- Missing Values
- Scaling
- Outliers
- Data encoding
- Signal processing
- Conclusion and references

Goals

Students should be able to:

- understand the tasks of the “pre-processing" step
- explain approaches for handling missing values and noise and mechanisms for scaling, outlier detection and data coding.
- describe and compare simple forms of representation
- explain the basic idea of time-series representation

Why pre-processing?

- How can real data be "unclean"?
  - Incomplete: Missing values, missing attributes in case of different data sources
  - Noisy: Measurement error, outlier
  - Inconsistent: Contradictory measurements, different sensors, sometimes also different scaling or translation
- Pre-processing is almost always done as a basis for meaningful results

Pre-processing (2)

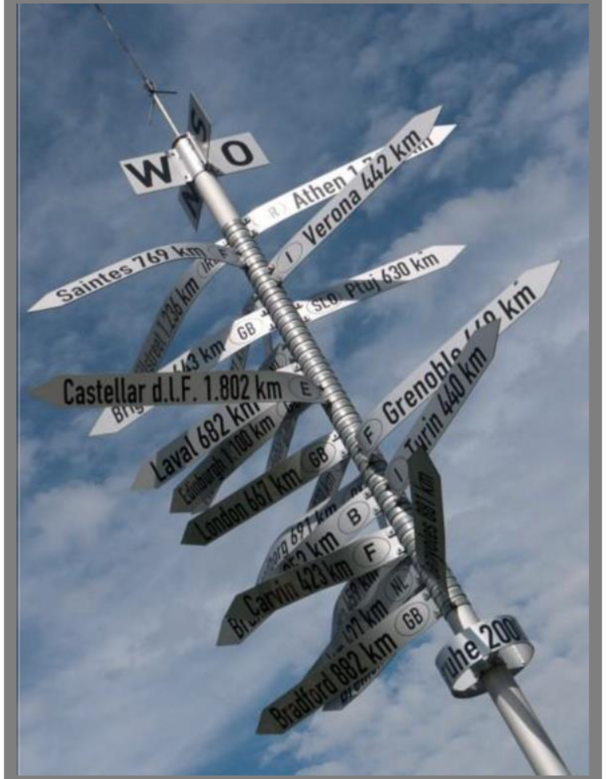
Main tasks of ‘Pre-processing”

- Cleanup:
  - handle missing values (e.g. replace)
  - Detect and treat outliers
  - Remove inconsistencies
- Integration: Combine information from multiple sources (also important: combine or split attributes, adjust time and value ranges)
- Transformation: normalisation, aggregation, conversion to another "basis”
- Reduction: as far as possible without (or with as little as possible) loss of information, e.g. via discretisation and aggregation

Agenda

Christian-Albrechts-Universität zu Kiel

- Missing Values
- Scaling
- Outliers
- Data encoding
- Signal processing
- Conclusion and references



Missing values

- For some samples, the values of individual attributes may be missing.
- Possible causes:
  - Failure of a sensor when measuring physical quantities
  - Reception or transmission problems (e.g. GPS in the underground car park)
  - Irrelevant attribute for the sample
  - Changes in a test setup
  - Combination of different data sets

Missing values (2)

Christian-Albrechts-Universität zu Kiel

The probability that the value is missing may or may not depend on the true value!

Examples:

- A temperature sensor does not provide values because its power supply has failed.
- A temperature sensor does not provide values below freezing.

Missing values (3)

Christian-Albrechts-Universität zu Kiel

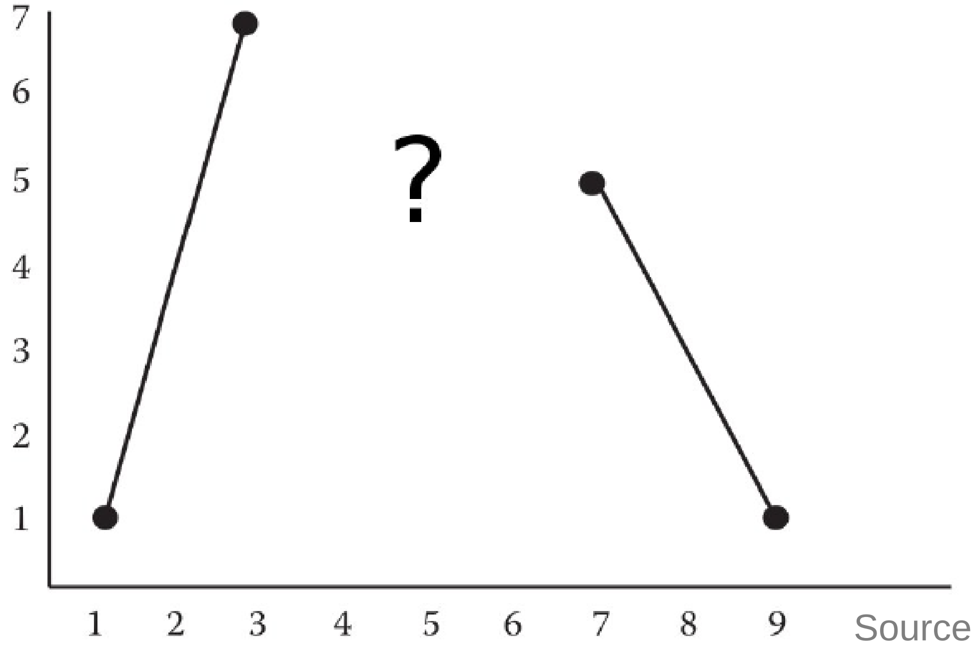
Possibilities for the treatment of missing values:

- Patterns with missing values are not used (only if a few patterns are affected, e.g. bad for time series).
- Missing values are taken into account by the subsequent processes themselves (process-dependent).
- Missing values are estimated, e.g. (see the process for data pre-processing!):
  - Use of the mean value
  - Use of the most common value
  - Estimation using the values of other attributes
  - Repetition of the last known valid value
  - Interpolation for time series
  - ...
- Important: Check whether the results of the subsequent processes can be falsified!

Missing values (4)

Especially with "few" missing values and "short" distances between measured values (e.g. time series from sensor data, GPS track):

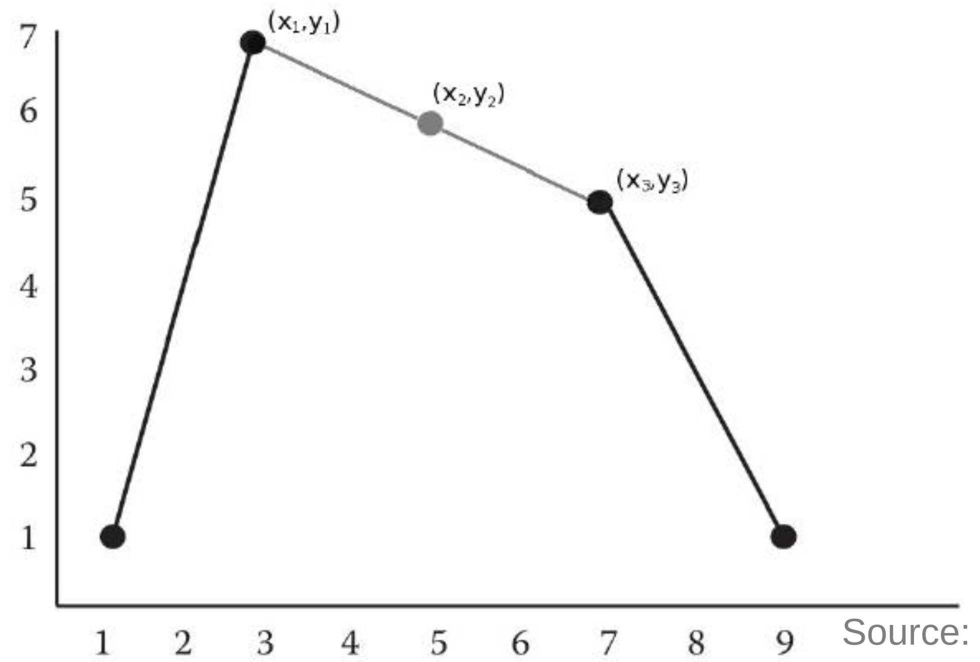
- Repetition of the last known value
- Linear (or quadratic, ...) interpolation



Source: [Mitsa 2010]

Linear interpolation

- $y_2 = y_1 + \frac{(y_3 - y_2)(x_2 - x_1)}{(x_3 - x_1)}$
- Example:



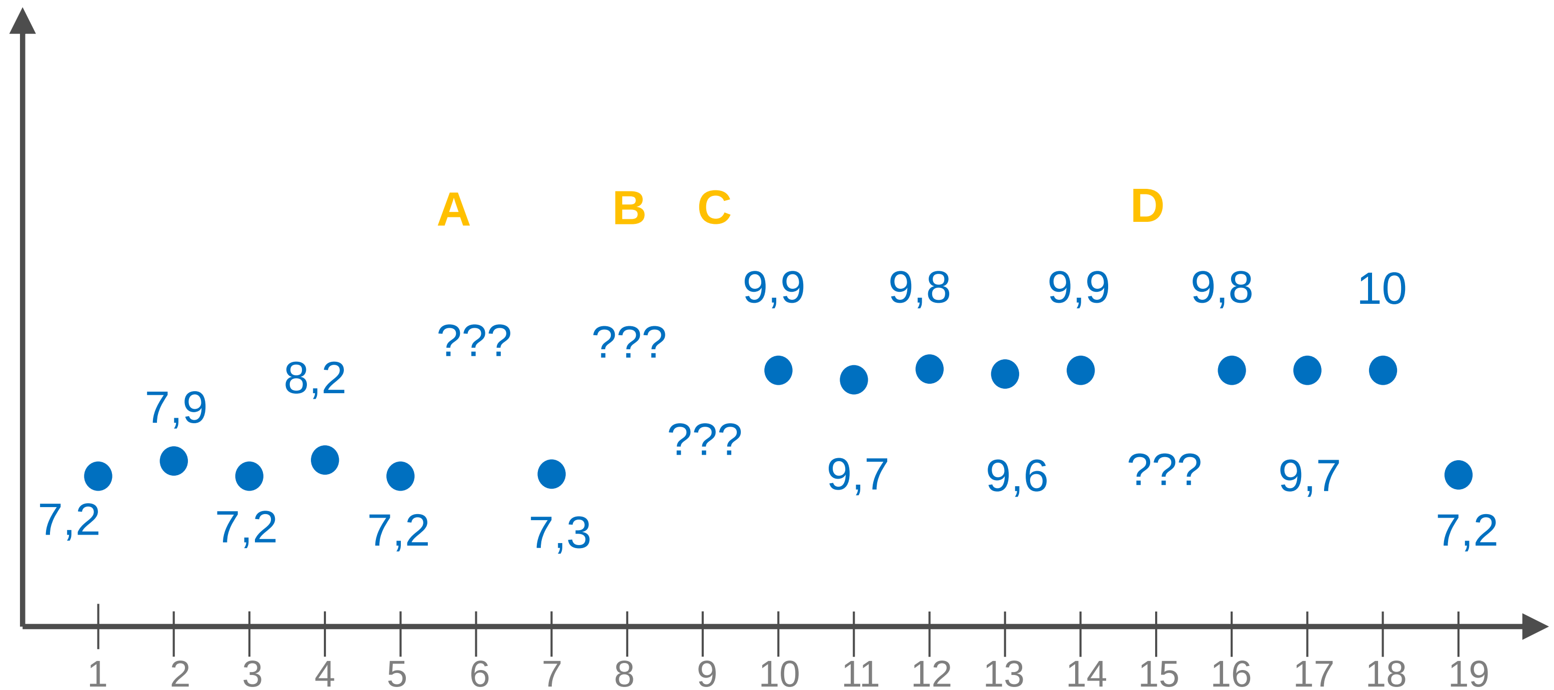
Source: [Mitsa 2010]

Missing values (6)

Christian-Albrechts-Universität zu Kiel

Question:

- Which values do you recommend for A, B, C, and D?



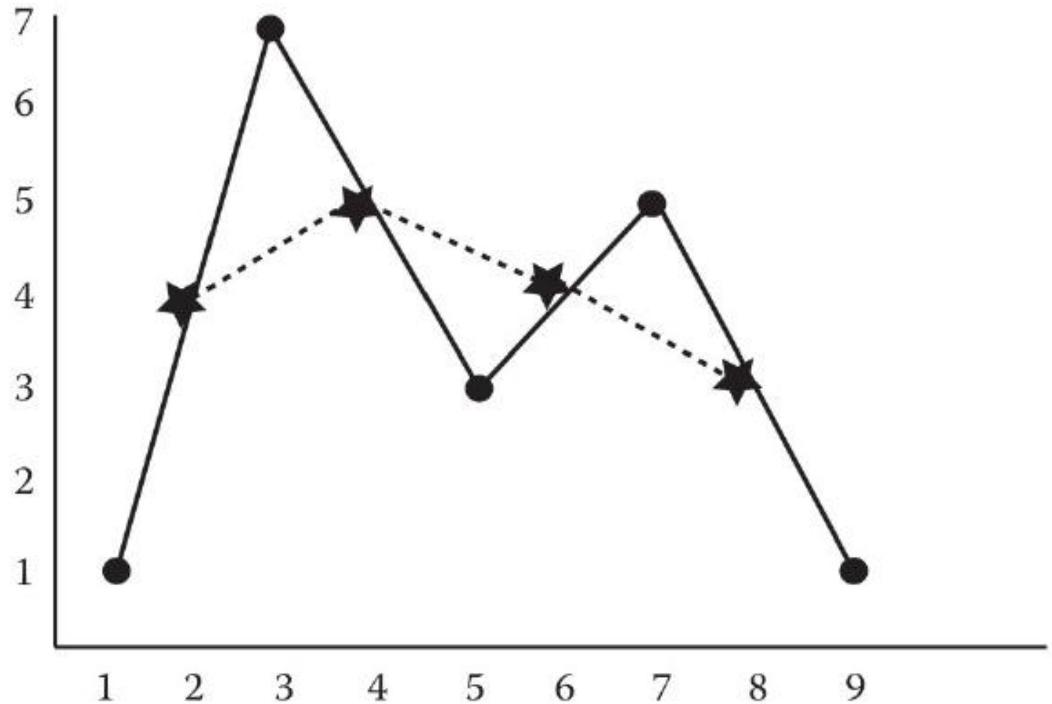
Causes of noise (sensor noise, inaccurate data, etc.)

- Poor sensors / insufficient resolution
- Recording error
- Interferences during transmission (interference etc.)

Solution approaches:

- Methods strongly dependent on the type of noise (e.g. normally distributed)
- Binning-Data is divided into equal bins and replaced by:
  - average
  - median or
  - border values

Moving average smoothing

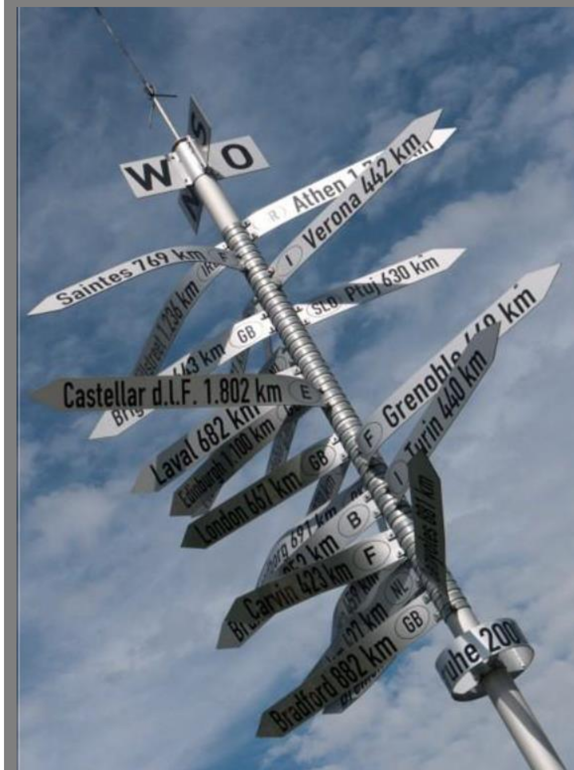


Source: [Mitsa 2010]

Agenda

Christian-Albrechts-Universität zu Kiel

- Missing Values
- **Scaling**
- Outliers
- Data encoding
- Signal processing
- Conclusion and references



Scaling

- Problem: Different value ranges of attributes
- Example 1: Temperature curves
  - Direct values from a sensor, such as the (temperature-dependent) resistance
  - Interpretable units such as Celsius, Kelvin, Fahrenheit or Rankine
  - Comparisons do not work if value ranges (reference system, basis, etc.) are different.
  - Even worse in reality: relations are unknown
- Example 2: Height and weight of a human being
  - If, for example, you measure the size in cm and weight in kg, the values that occur are approximately the same order of magnitude; it makes sense to calculate distances between patterns.
  - If, for example, you measure the size in m and weight in g, the values that occur are in different orders of magnitude; it makes no sense to calculate distances between patterns since the weight strongly dominates the size.
- Solution: Normalisation or standardisation of the values.

Christian-Albrechts-Universität zu Kiel

Normalisation

- If the values are in the interval $[a, b]$, they are transformed linearly so that the transformed values are in the unit interval $[0,1]$:

$$x' = \frac{x - a}{b - a}$$

- Here, x is the value to transform and x' is the transformed value.
- The values of a and b can be the minimum and the maximum value occurring in the data set for the attribute

Scaling (3)

The problem of normalisation:

- New data (e.g. in the application) may contain values outside the interval $[a, b]$.
- Individual outlier values can cause the available value range $[0,1]$ to be used very poorly.

Example: Monitoring the power consumption of a vehicle.

- Normally, the consumption fluctuates around 50 - 150 Watt for simple consumers, such as lights, windscreen wipers, seat heating or radio.
- When starting the vehicle, however, peaks of 5 kW and more occur, where by "normal" fluctuations are scaled into very small intervals.

Solution: Standardisation that avoids this outlier effect.

Standardisation

- Also known as ‘Mahalanobis Scaling‘
- Transforms the data to give a mean of 0 and a dispersion (empirical standard deviation) of 1:

$$x' = \frac{x - \mu}{\sigma}$$

- Here, μ is the mean and σ the empirical standard deviation.

Mean and variance

- Mean μ of the n samples y_k:

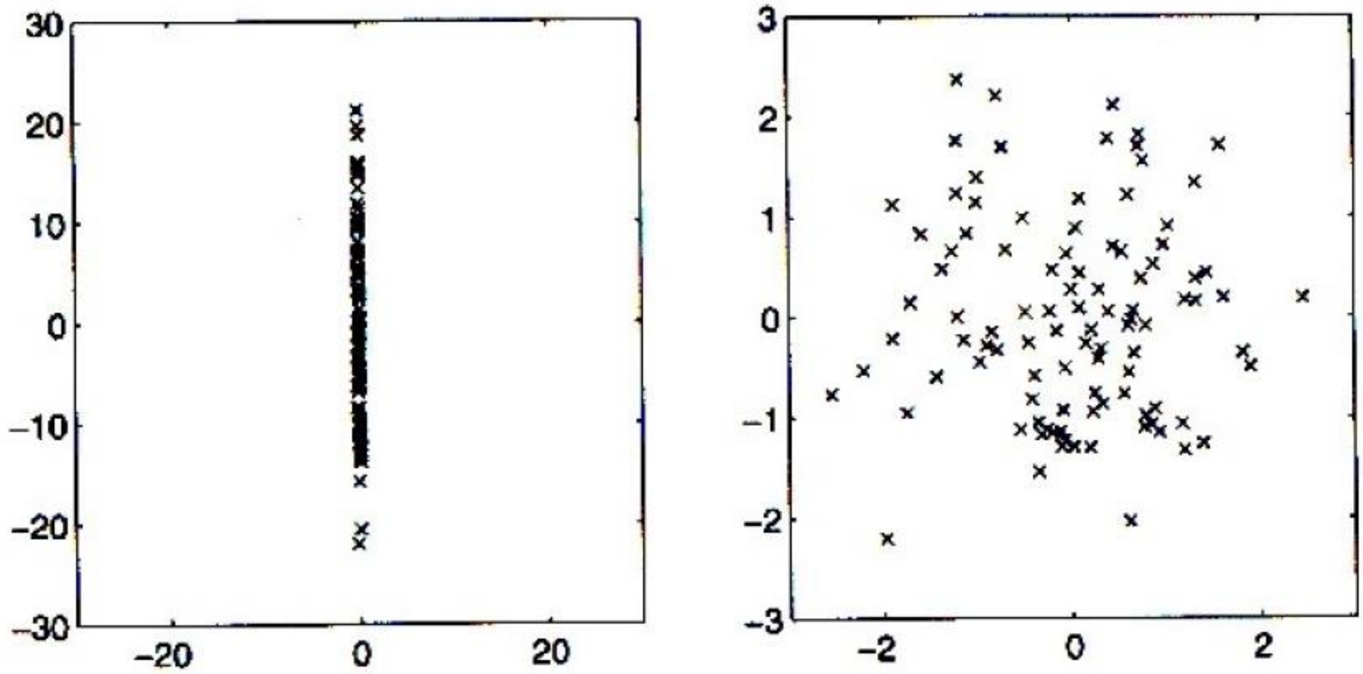
$$\mu = \frac{1}{n}\sum_{k=1}^{n} y_k$$

- Empirical variance σ^2 of n samples:

$$\sigma^2 = \frac{1}{n-1}\sum_{k=1}^{n} (y_k - \mu)^2$$

- The empirical standard deviation (or spread) is the square root of the empirical variance.

Standardisation (3)



Source: [Mitsa 2010]

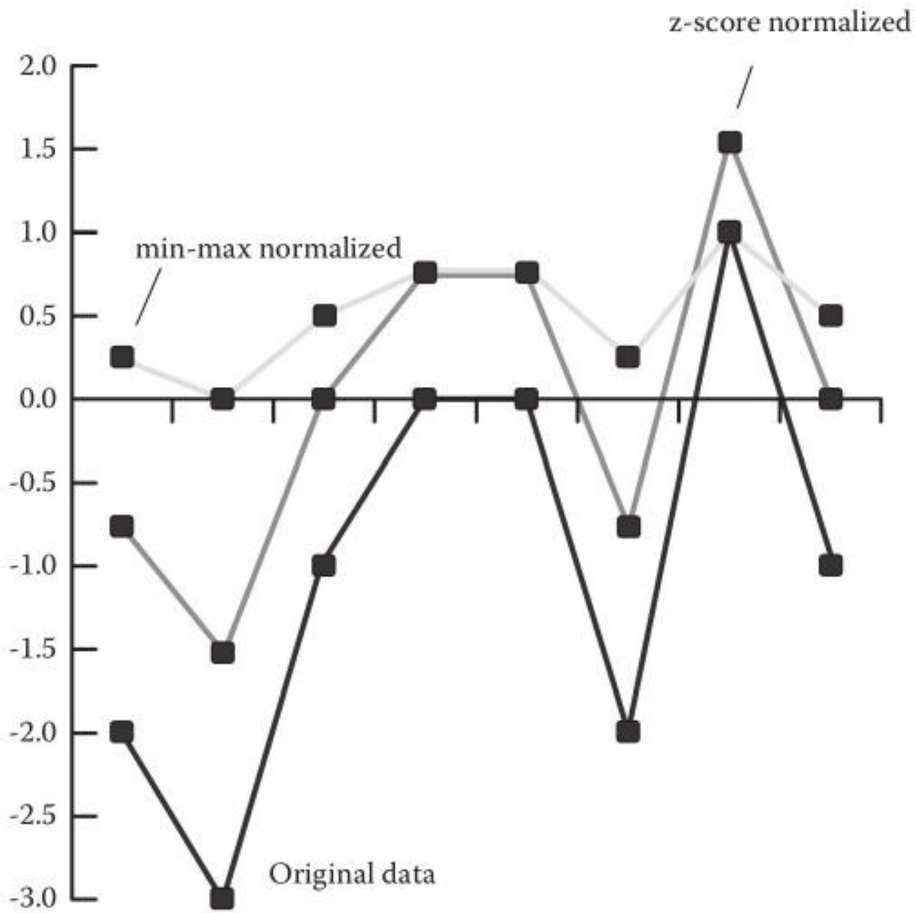
- Original data set (left): Gaussian random process with a mean (0,0) and a standard deviation (0.1,10).

Christian-Albrechts-Universität zu Kiel

Instructions for application:

- Normalisation or standardisation is performed separately for each attribute.
- Determination of scaling parameters from known data
- For time series: scaling of each data value with global parameters, not separately for each time series.

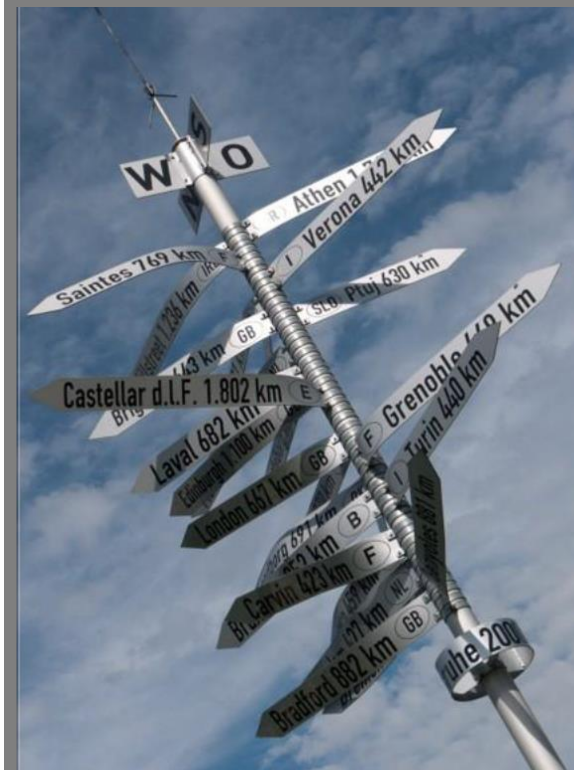
Standardisation (5)



Source: [Mitsa 2010]

Agenda

- Missing Values
- Scaling
- **Outliers**
- Data encoding
- Signal processing
- Conclusion and references



Outliers

- For some patterns, the values of attributes can be inaccurate, distorted, or falsified (see also missing values).
- Possible causes:
  - Sensor noise when measuring physical quantities
  - Transmission errors
  - False information during interviews (e.g. question about age or weight)
  - ...
- Such outliers should be recognised and treated appropriately.

Outliers (2)

Detection of outliers:

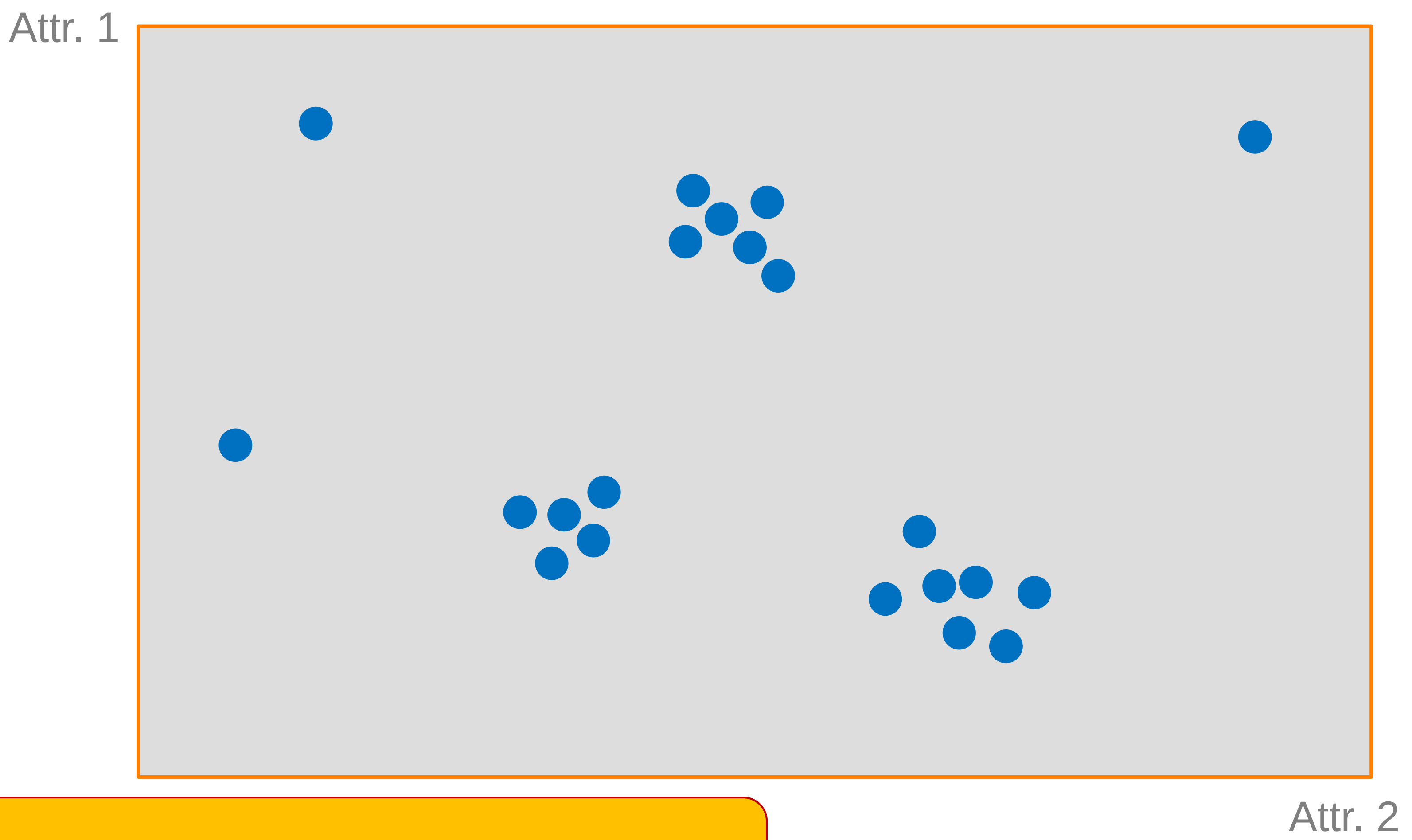
→ A pattern is identified as an outlier when:

- The value of at least one attribute is outside an allowed value range.
- The value of an attribute deviates from the mean by more than two or three times the standard deviation (statistical measure).
- The value of an attribute deviates from a value estimated with a suitable model by more than a specified amount.
- ...

Problem: Distinguishing outliers from exotics (correct but unusual data that carries valuable information).

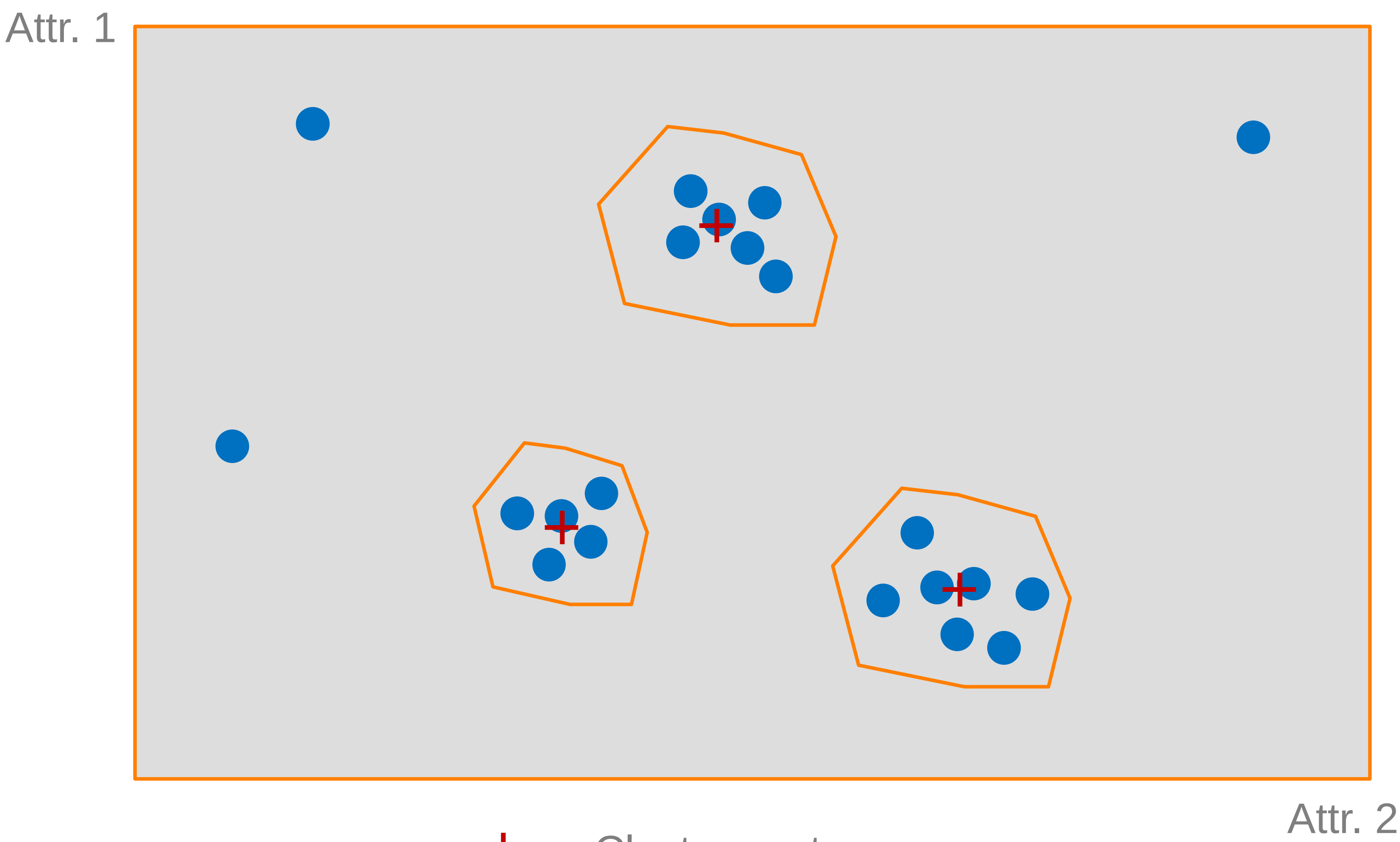
Outliers (3)

Christian-Albrechts-Universität zu Kiel



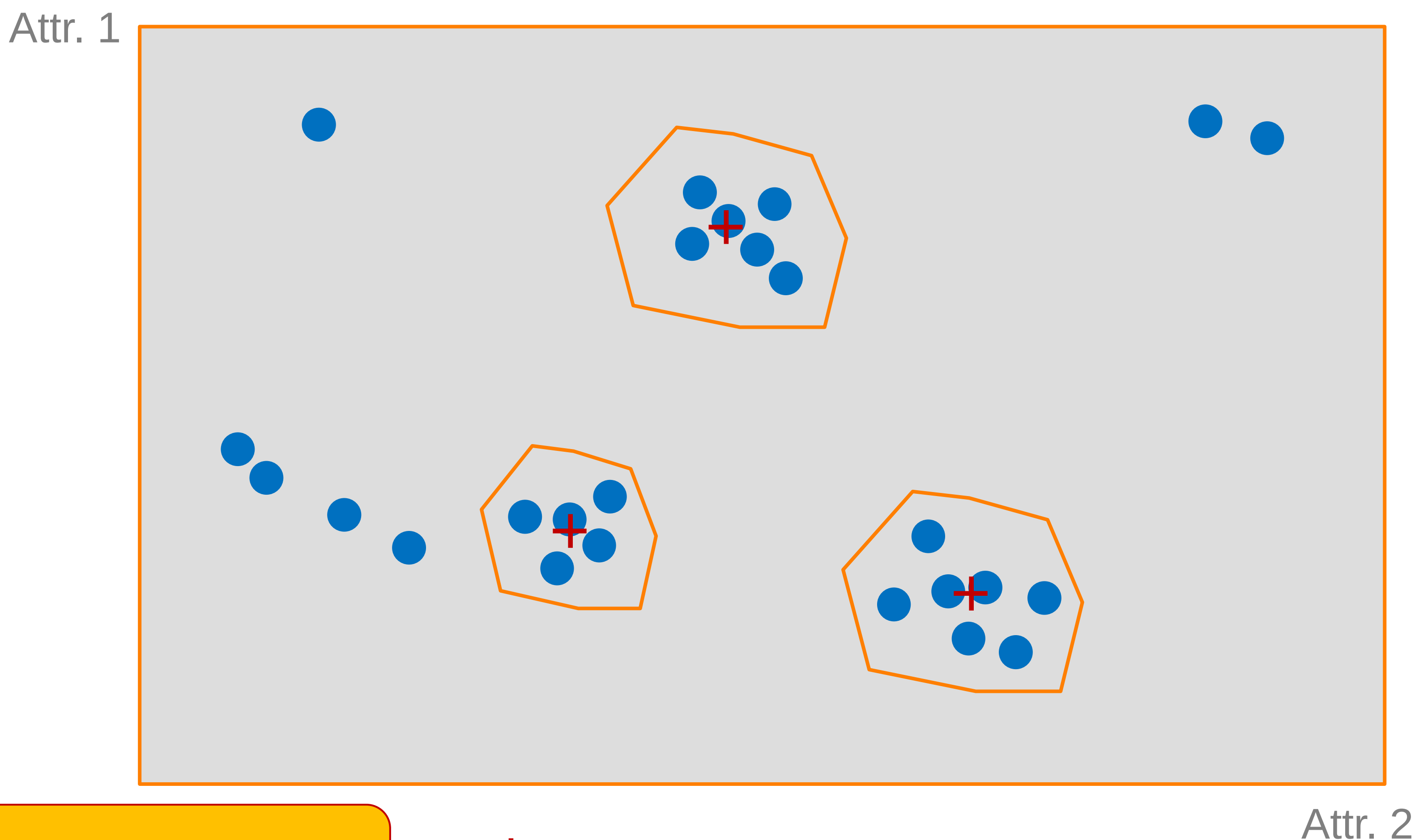
Which of the data points are outliers?

Outliers (4)



+ = Cluster centre

Outliers (5)



And now …?

+ = Cluster centre

Treatment of outliers:

→ Different options, depending on how much the data set is modified:

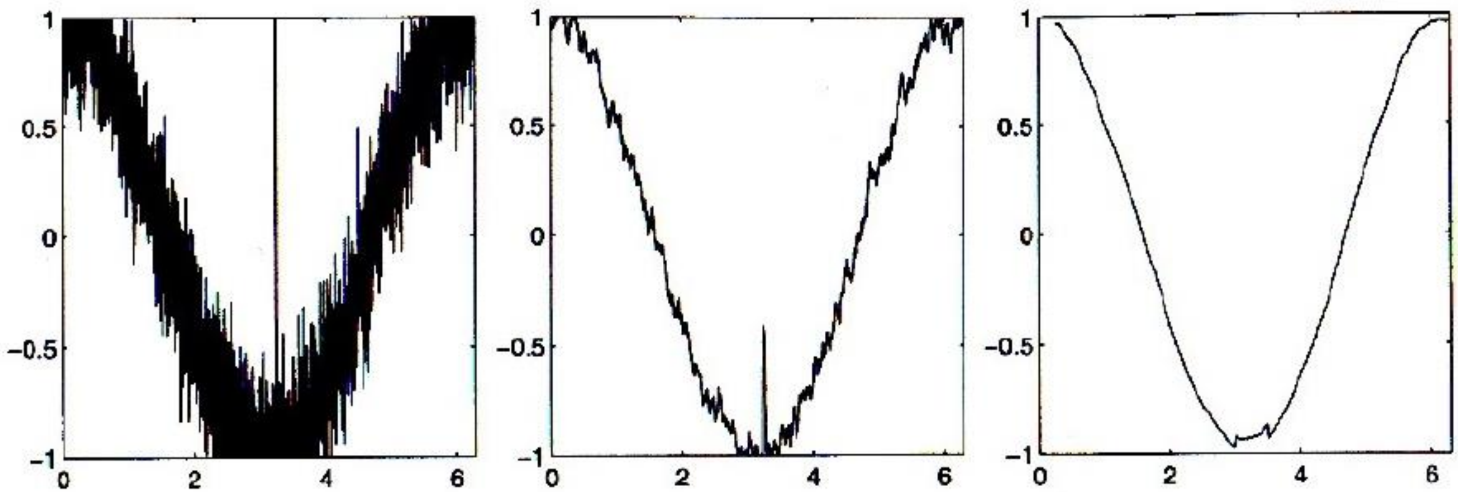
- Marking (only suitable for some subsequent techniques, see also missing values)
- Removal of the corresponding pattern or marking of the outlier as "invalid".
- Correction of the value

Techniques for correction:

- Replacement by the maximum or minimum value
- Replacement by the global mean value
- Linear or non-linear interpolation for time series
- Model-based addition using time series models, e.g. ARMA models etc.

→ Method strongly depends on the type of data or underlying process.

- Example: Elimination of outliers by moving average for a time series



[Runkler 2000]

- Original data record with outliers (left), a result of filtering by moving average with short time window (middle) and long time window (right).

Inconsistencies

- Goal: Detection and handling of inconsistencies
- Procedure similar to outlier detection
- E. g. Clustering the sample data and checking the homogeneity of the clusters about certain criteria
- A consistent set of examples can be very important, especially for later processing of the data (e.g. in the form of a model for several examples).

Scaling in the time domain

In addition to scaling in the value domain, scaling in the time domain may also be useful for time series (thus, sensor data)!

Examples:

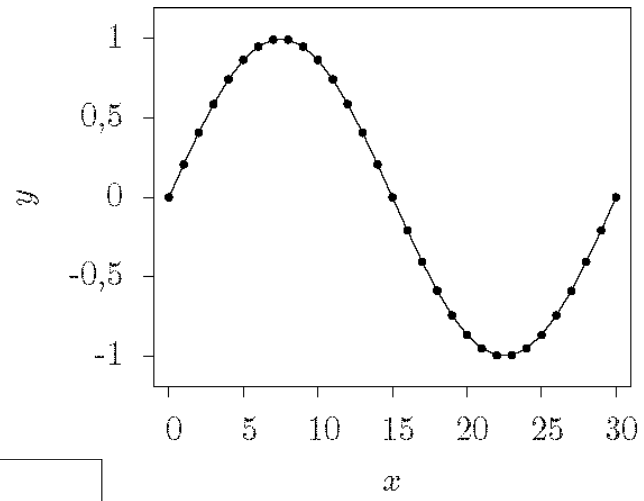
- Recording of temperature values at different intervals
- Use of different scales in the time domain (e.g. milliseconds and seconds)

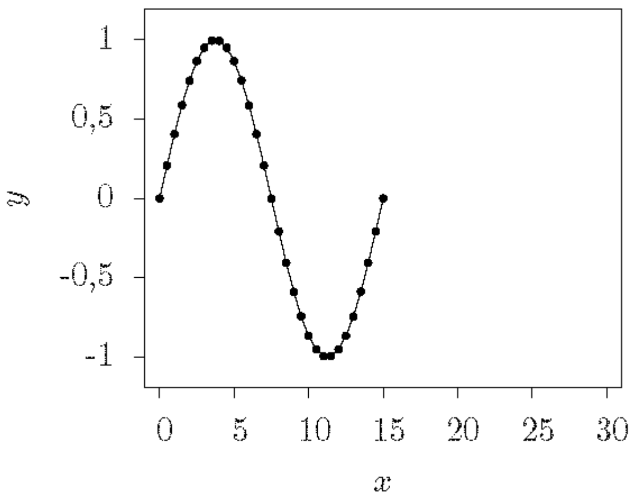
Problem: Behaviour is not directly comparable

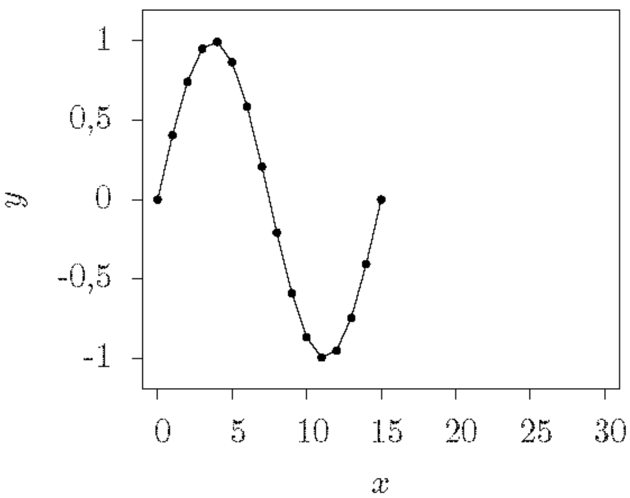
Solution:

- Scaling in the time domain or rescanning of the time series
- Additional application: Reduction of data volume

Scaling in the time domain (2)

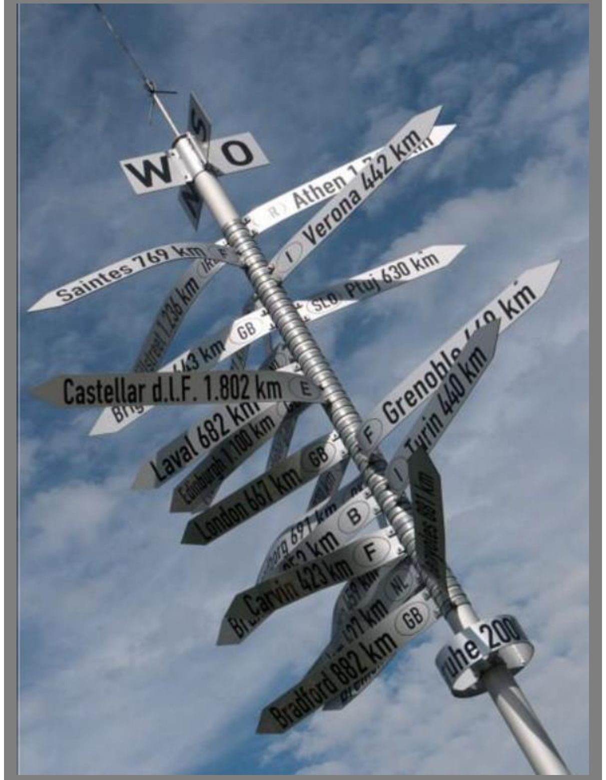






Agenda

- Missing Values
- Scaling
- Outliers
- **Data encoding**
- Signal processing
- Conclusion and references



Data encoding

- Problem: Some methods only work on numeric data.
- Non-numeric data must therefore be suitably coded.
  - Ordinal attributes: Rank-based Coding
  - Nominal attributes: orthogonal coding (e.g. 1-out-of-k coding: 00...010...00) if k is the number of possible expressions of the attribute.
- Sometimes when coding classes: orthogonal coding, where the length of the vector reflects the class strength (number of patterns available in the training data).

Data encoding (2)

Example of a rank-based coding

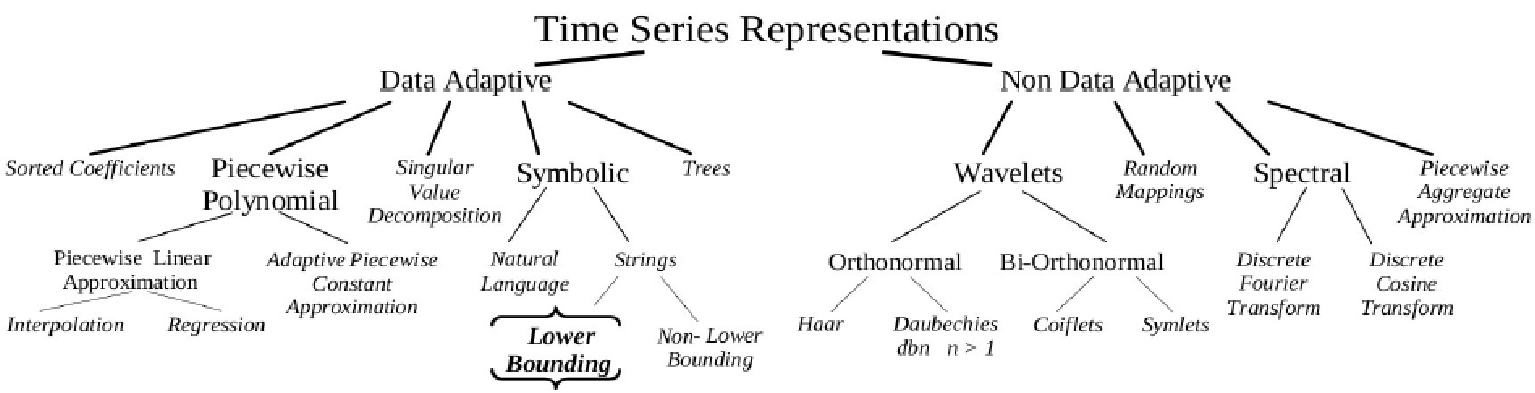
Ausbildung	Repräsentation
Hauptschulabschluss	1
Realschulabschluss	2
Abitur	3
Diplom	4
Promotion	5

Data encoding (3)

Example of orthogonal coding of classes with quadratic error as an error measure in the model building:

Class	Size	Representation
$\mathcal{A}$	$\lvert\mathcal{A}\rvert$	$\left(\frac{1}{\sqrt{\lvert\mathcal{A}\rvert}}, 0, 0, 0, 0\right)^{\mathrm{T}}$
$\mathcal{B}$	$\lvert\mathcal{B}\rvert$	$\left(0, \frac{1}{\sqrt{\lvert\mathcal{B}\rvert}}, 0, 0, 0\right)^{\mathrm{T}}$
$\mathcal{C}$	$\lvert\mathcal{C}\rvert$	$\left(0, 0, \frac{1}{\sqrt{\lvert\mathcal{C}\rvert}}, 0, 0\right)^{\mathrm{T}}$
$\mathcal{D}$	$\lvert\mathcal{D}\rvert$	$\left(0, 0, 0, \frac{1}{\sqrt{\lvert\mathcal{D}\rvert}}, 0\right)^{\mathrm{T}}$
$\mathcal{E}$	$\lvert\mathcal{E}\rvert$	$\left(0, 0, 0, 0, \frac{1}{\sqrt{\lvert\mathcal{E}\rvert}}\right)^{\mathrm{T}}$

- Many different forms of representation for time series

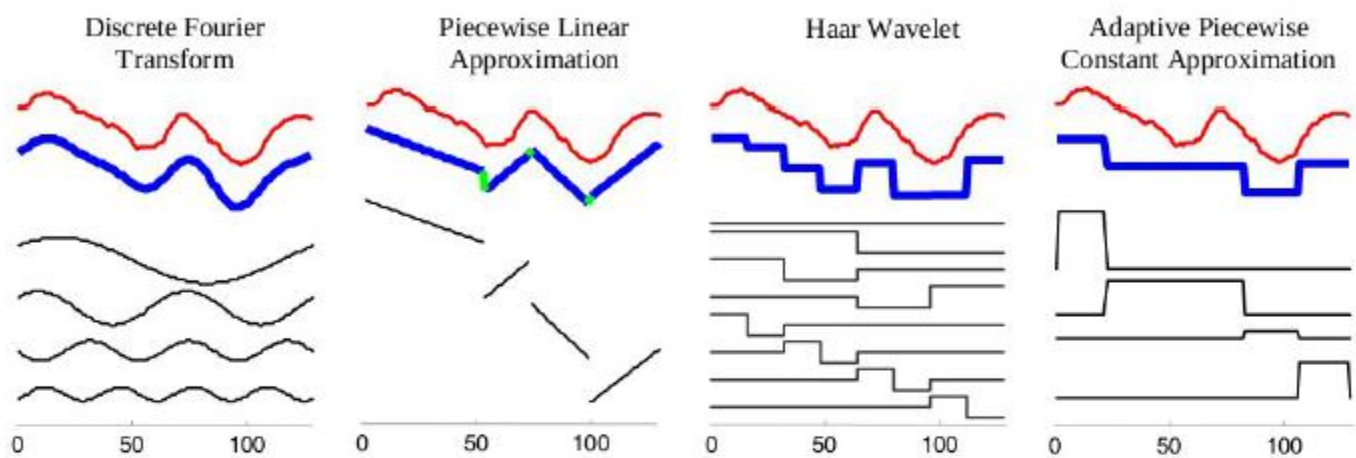


[Lin, Keogh, Wei und Lonardi, Experiencing SAX: a Novel Symbolic Representation of Time Series 2007]

- However, often just the "raw data" are used.

Possible differentiation criteria:

- "Basic" functions
- Adaptivity
- Representation of local or global processes



[Lin, Keogh, Wei und Lonardi, Experiencing SAX: a Novel Symbolic Representation of Time Series 2007]

Representation (3)

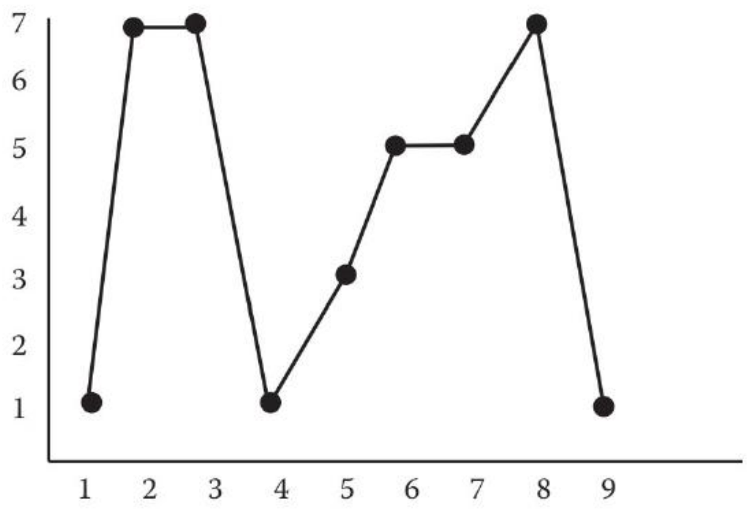
Christian-Albrechts-Universität zu Kiel

Statistical features

- Characteristics (attributes, features): the simplest form of representation
- Examples:
  - Average (see scaling)
  - Variance or standard deviation (see scaling)
  - Median
  - Mode
- Disadvantage: little or no recording of the time course
- Advantages:
  - All sequences are mapped to the same length
  - Insensitive to typical interference (noise, outliers, etc.)

Run-length based signature

- Process:
  - Values repeated several times are counted (directly consecutive repetitions)
  - Values and corresponding number result in signature
- Example:



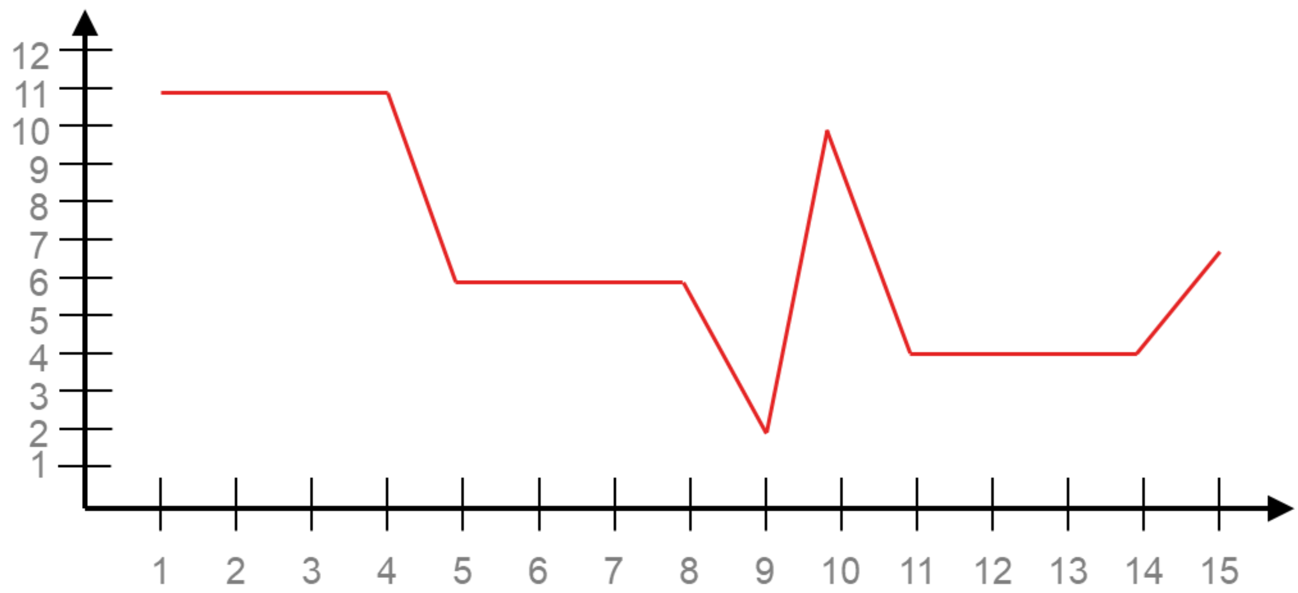
[Mitsa 2010]

- Run-length signature of the example time series: (5,2);(7,2)

Christian-Albrechts-Universität zu Kiel

Run-length based signature

- What is the signature in the following example?



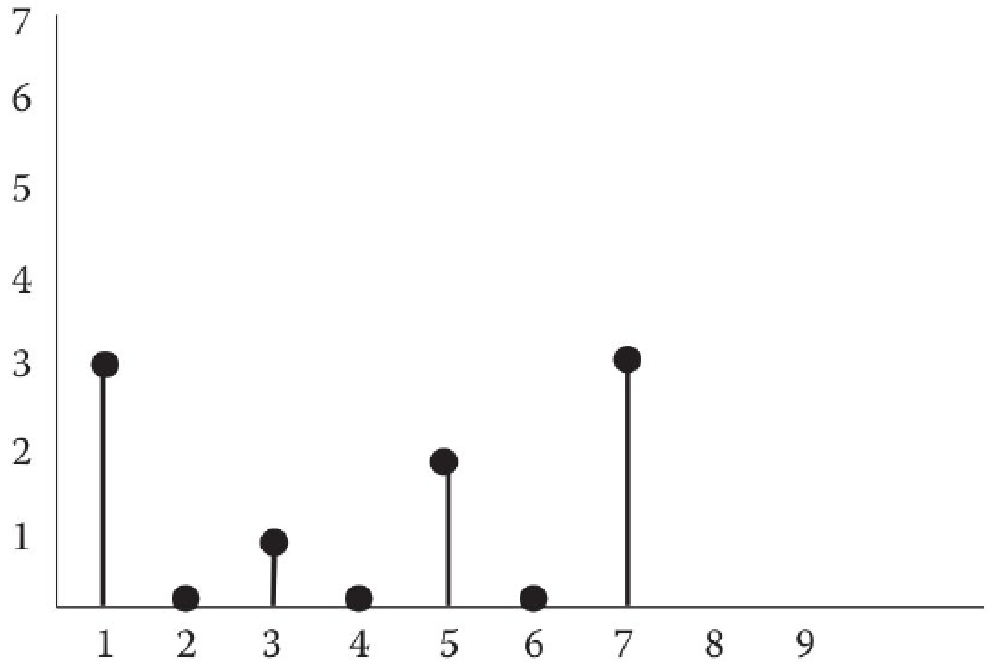
Solution

Christian-Albrechts-Universität zu Kiel

Histogram

- Process:
  - Number of all occurring values is determined

- Example:

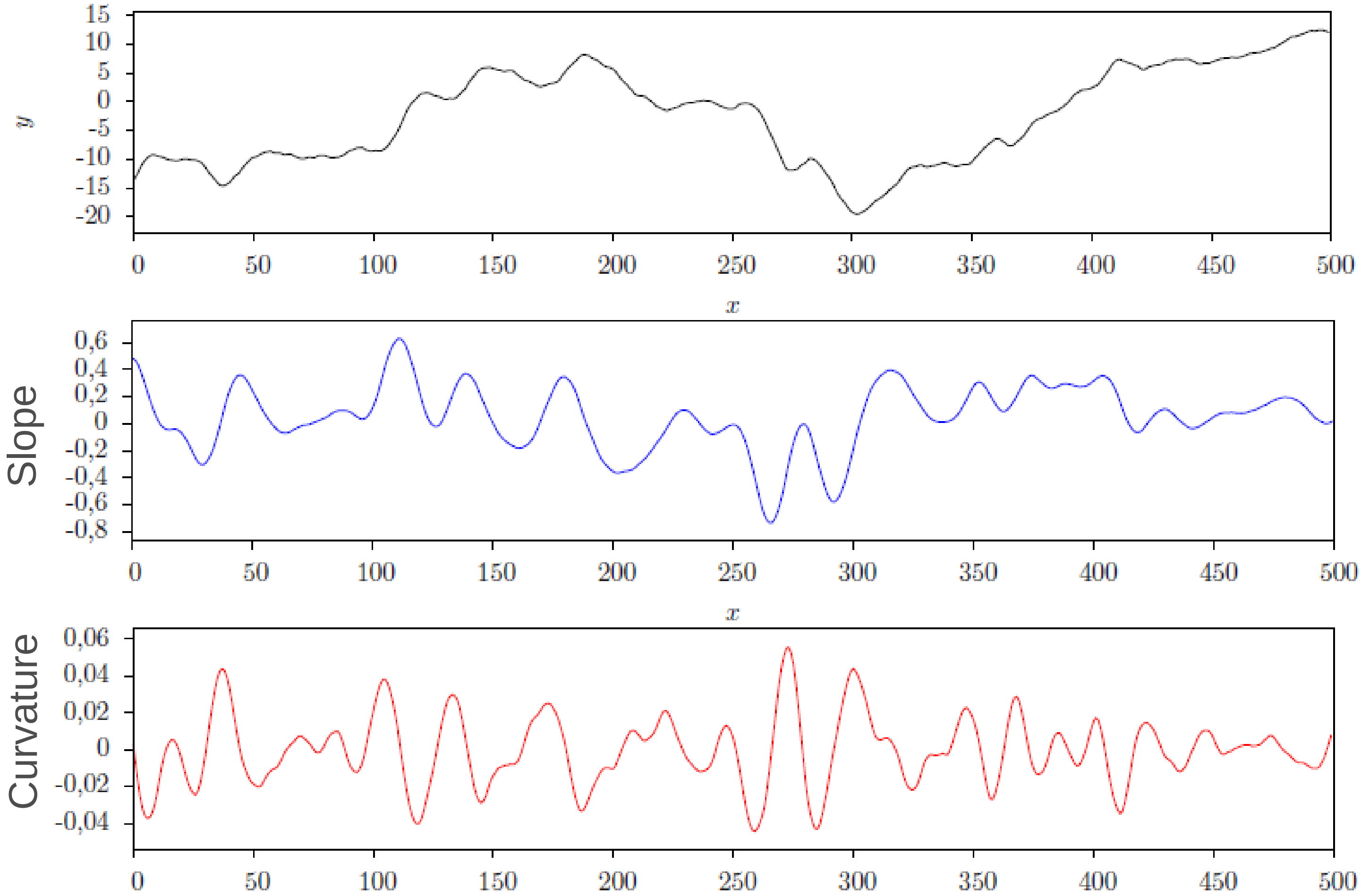


[Mitsa 2010]

Simple representations:

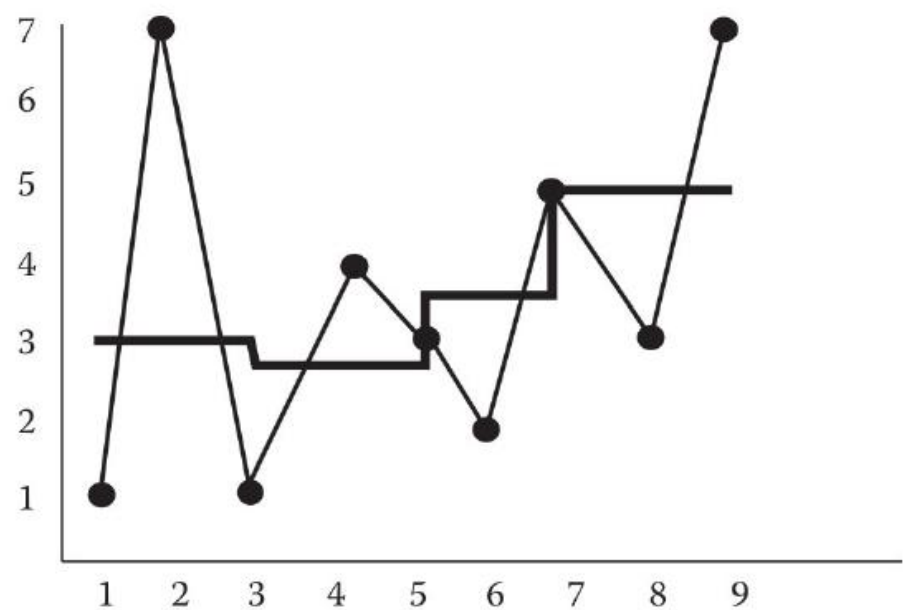
- Often only useful after discretisation / quantisation / symbolisation
- Many more features can be calculated from gradients
- Instead of a single value, it can also be useful to calculate characteristics for subsections of a time series.
- Example: Slope and curvature of a signal

Simple representation forms (2)



Piecewise Aggregate Approximation / Composition (PAA/PAC)

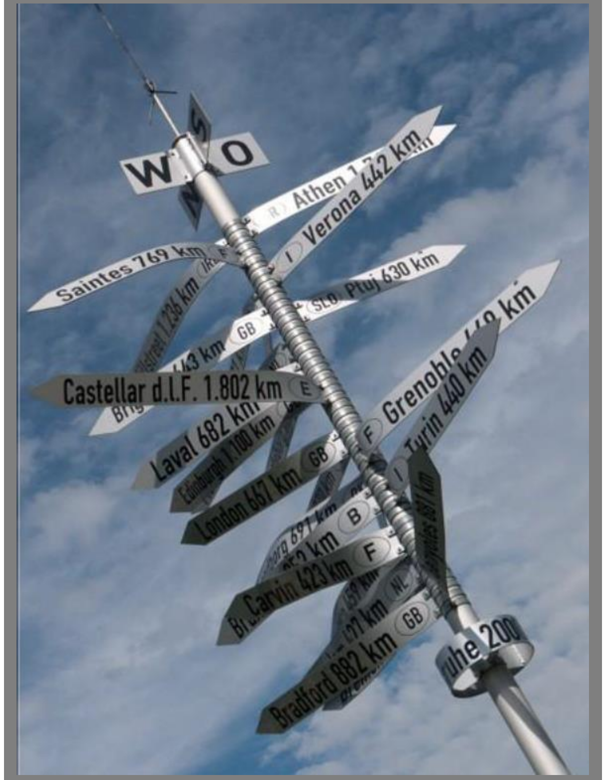
- Approach: Time series is divided into sections of equal length and each section is replaced by a constant value derived from the average of the values within each section.



[Mitsa 2010]

Agenda

- Missing Values
- Scaling
- Outliers
- Data encoding
- Signal processing
- Conclusion and references



Some basic on signals and signal processing

→ We‘ll use this afterwards to discuss digital filters as a final pre-processing step

Christian-Albrechts-Universität zu Kiel

Signals and systems

- Signals: Functions of time $x\colon \mathbb{T} \rightarrow \mathbb{W}$
  → Already known
- Now: Signal processing systems
  - As with sensors: System as Black Box

  - Maps input signal $x(t)$ to output signal $y(t)$

$$y(t) = \boldsymbol{S}\{x(t)\}$$

  - Depending on the type of the signal: analogue or digital

System properties

- Systems can have certain properties
- Allow for categorisation of systems
- Causality: A system is causal if the output signal at time t_0 depends only on values of the input signal $x(t)$ with $\mathrm{t} < t_0$. The system is also called *realisable* or *practicable*.
- Stability: A system is stable if it responds to a limited input signal with a limited output signal:

$$\forall t: |x| \leq A_1 < \infty \Rightarrow |y| \leq A_2 < \infty$$

- BIBO Property: Bounded Input – Bounded Output

System properties

- Linearity: A system is linear if $x_i(t)$ and associated constants $a_i \in \mathbb{R}$ apply to any input signal:
$$\boldsymbol{S}\left\{\sum_{i=1}^{I} a_i \cdot x_i(t)\right\} = \sum_{i=1}^{I} a_i \cdot \boldsymbol{S}\{x_i(t)\}$$
- Time-invariance: A system is time-variant if the relationship between the input signal and the output signal is not time-dependent, i.e. if the following applies to any time offset t_0:
$$\boldsymbol{S}\{x_i(t)\} = y(t) \Longrightarrow \boldsymbol{S}\{x_i(t-t_0)\} = y(t-t_0)$$
- Very important ‘class’ of systems:
Linear time-invariant (LTI) systems

Dirac pulse

- Also known as "Diracian Delta Function” or “Impulse Function”:

$$\delta(t) = \begin{cases} \infty & \text{for } t = 0 \\ 0 & \text{for } t \neq 0 \end{cases}$$

with

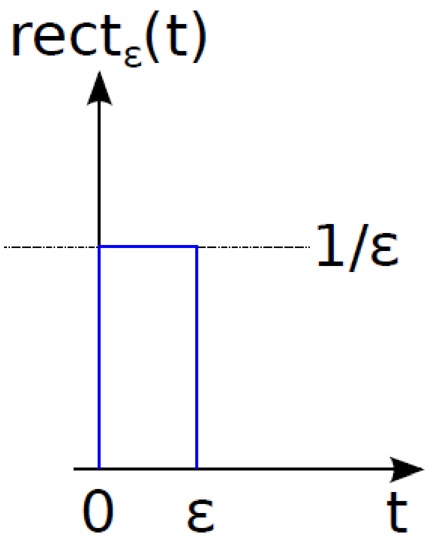
$$\int_{-\infty}^{+\infty} \delta(t) dt = 1$$

- No function in the “classic sense”
- Schematic representation:

Dirac pulse

- Derivation via rectangle function:

$$\mathrm{rect}_\varepsilon(t) = \begin{cases} \dfrac{1}{\varepsilon} & \text{for } 0 < t < \varepsilon \\ 0 & \text{otherwise} \end{cases}$$



- At the border crossing $\varepsilon \to 0$ applies:

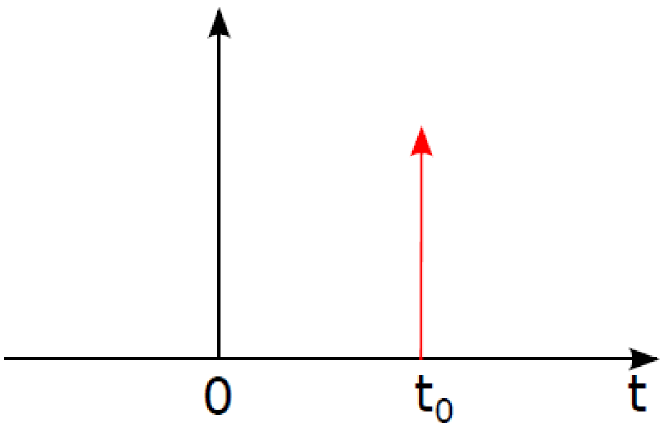
$$\lim_{\varepsilon \to 0} \mathrm{rect}_\varepsilon(t) = \delta(t)$$

- Alternative: Derivation via normal distribution function with vanishing variance

Dirac pulse

- The offset of the Dirac pulse:

$$\delta(t - t_0) = \begin{cases} \infty & \text{for } t - t_0 = 0 \\ 0 & \text{otherwise} \end{cases}$$

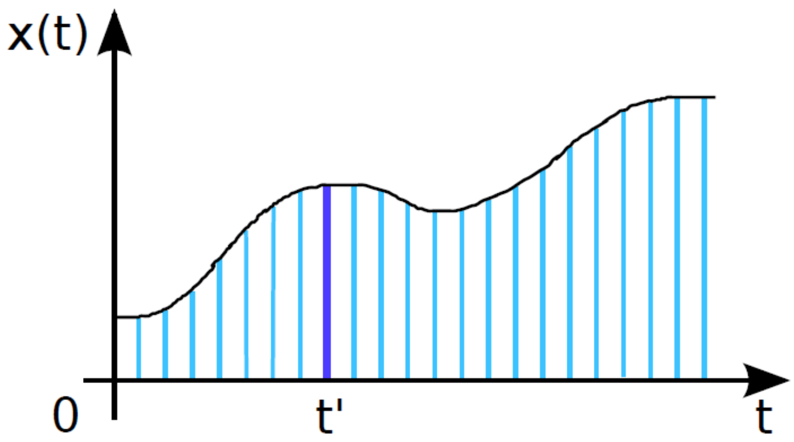


Dirac pulse: Representation of arbitrary functions

- Given is an input signal $x(t)$
- $x(t)$ can be composed of weighted Dirac pulses
  → Hide property of the delta function

$$x(t) = \int_{-\infty}^{+\infty} x(\tau)\, \delta(t-\tau) d\tau$$

- Example:



Calculation of the output of LTI systems

- Let $h(t) = \boldsymbol{S}\{\delta(t)\}$ be the output signal of an LTI system in case of a Dirac pulse as input (impulse response)
- For any input signal $x(t)$ the output signal $y(t)$ of the system applies:

$$y(t) = \boldsymbol{S}\{x(t)\}$$

$$= \boldsymbol{S}\left\{\int_{-\infty}^{+\infty} x(\tau) \cdot \delta(t-\tau) d\tau\right\}$$

$$= \int_{-\infty}^{+\infty} x(\tau) \cdot \delta(t-\tau) d\tau$$

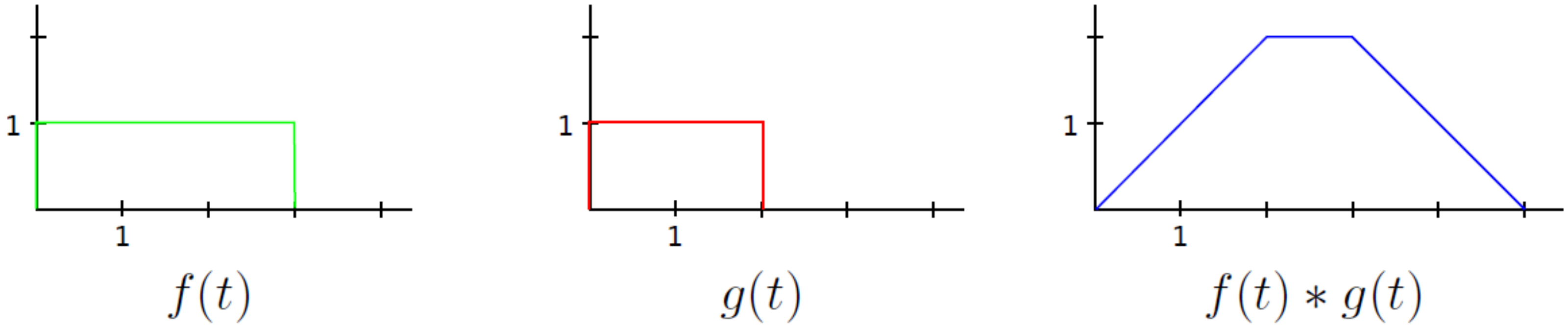
$$= \int_{-\infty}^{+\infty} x(\tau) \cdot h(t-\tau) d\tau$$

Convolution

- Let $f(t)$ and $g(t)$ be two functions, then their convolution is defined as:

$$f(t) * g(t) = \int_{-\infty}^{+\infty} f(\tau) \cdot g(t-\tau) d\tau$$

- Convolution operator: $*$
- Example:



- Convolution of the rectangle functions results in trapezoidal function

Summary: Folding and LTI systems

$$y(t) = \int_{-\infty}^{+\infty} x(\tau) \cdot h(t-\tau) d\tau$$

$$y(t) = x(t) * h(t)$$

- The output signal of an LTI system with the impulse response $h(t)$ corresponds to the convolution of the input signal with the impulse response.
- The impulse response completely describes the behaviour of an LTI system.

Digital Filters

- LTI systems can change the amplitudes and phases of the frequencies contained in an input signal (but not the frequencies themselves).
  - LTI systems are suitable for filtering sensor signals
- Goal: Suppress/amplify certain components (i.e. frequencies) of the input signal.
  - Reduction of interfering parts
  - Emphasis on informative or discriminatory elements
- Classification of digital filters
  - On the basis of their structure
    - Non-recursive filters
    - Recursive filters
  - Based on their impulse response
    - Finite impulse response (FIR)
    - Infinite impulse response (IIR)

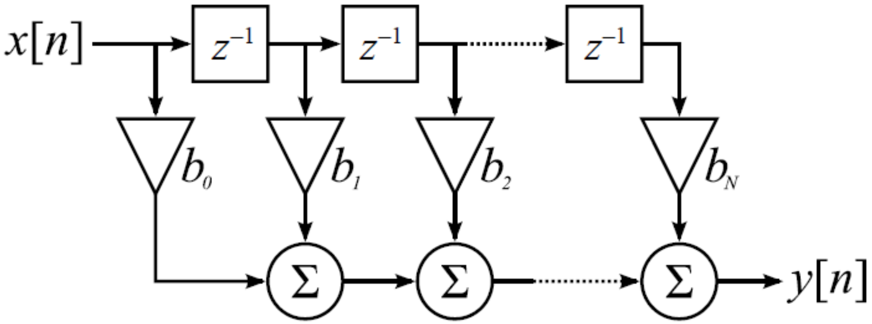
Digital filters (2)

Non-recursive filters

- They have no feedback:

$$y(t) = \sum_{k=0}^{N} b_k \cdot x(t-k)$$

- b_k are the filter coefficients
- Filter of order N
- Realises discrete convolution:



- Finite impulse response
  - Corresponds to the filter coefficients b_k
- Always stable

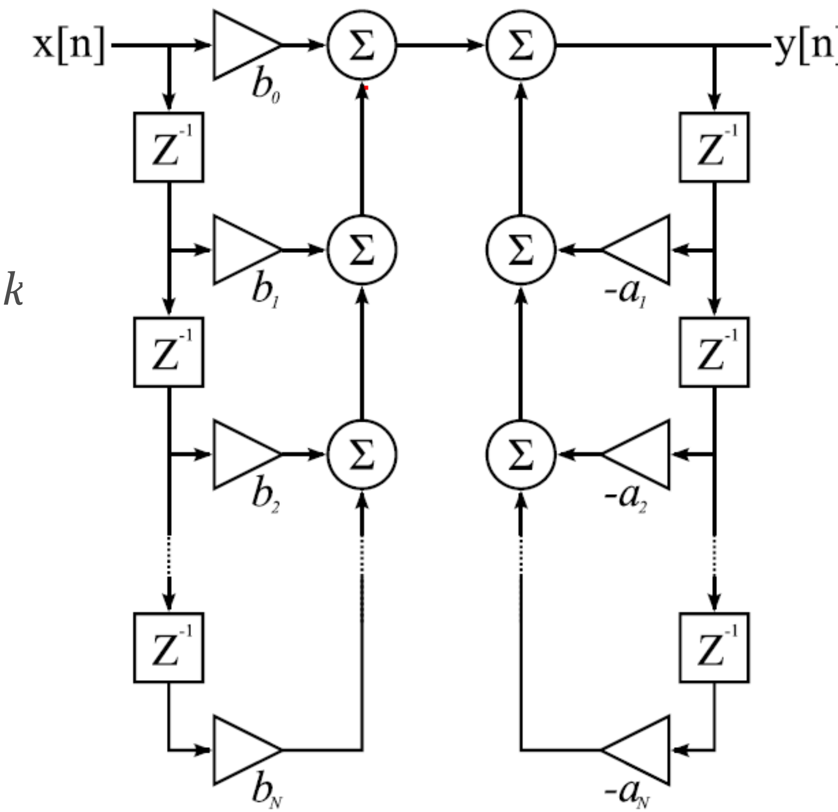
Digital filters (3)

Christian-Albrechts-Universität zu Kiel

Recursive filters

- Have at least one feedback

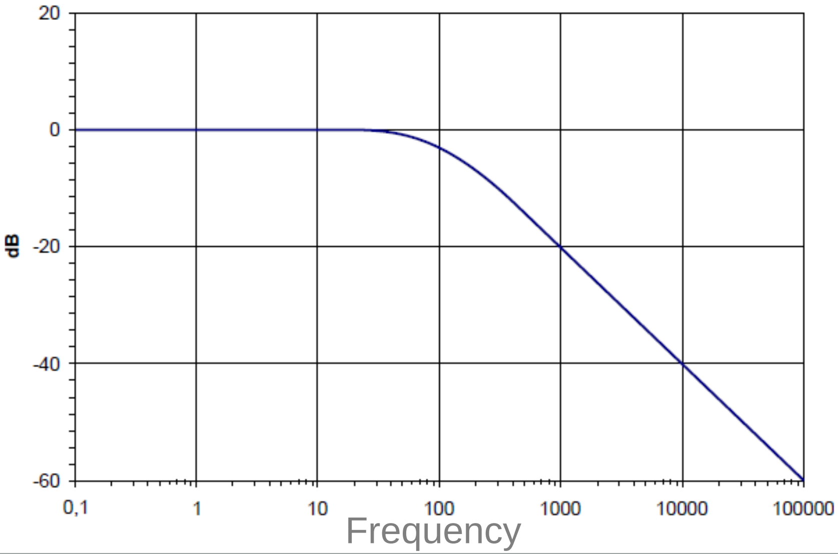
$$y(t) = \sum_{k=0}^{N} b_k \cdot x(t-k) - \sum_{k=1}^{M} a_k \cdot y(t-k$$

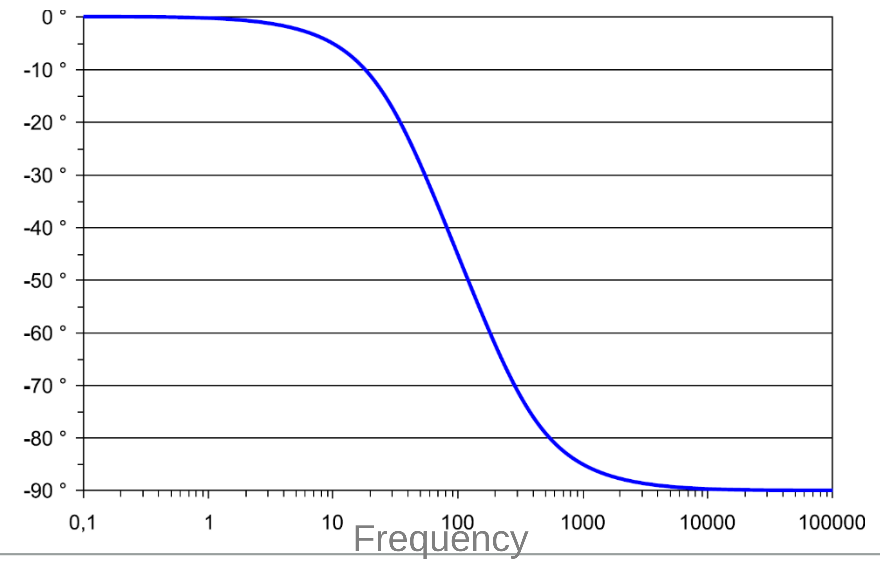


- Usually infinite impulse response
- ‘Danger’ of instability

Digital Filters: Characterisation via Frequency Response

- LTI filters change the amplitudes and phases of the frequencies contained in the input signal.
- Characterisation via frequency response (transfer function)
  - Amplitude response: Amplitude gain or amplitude damping as a function of frequency
  - Phase response: displacement of the phase position as a function of frequency

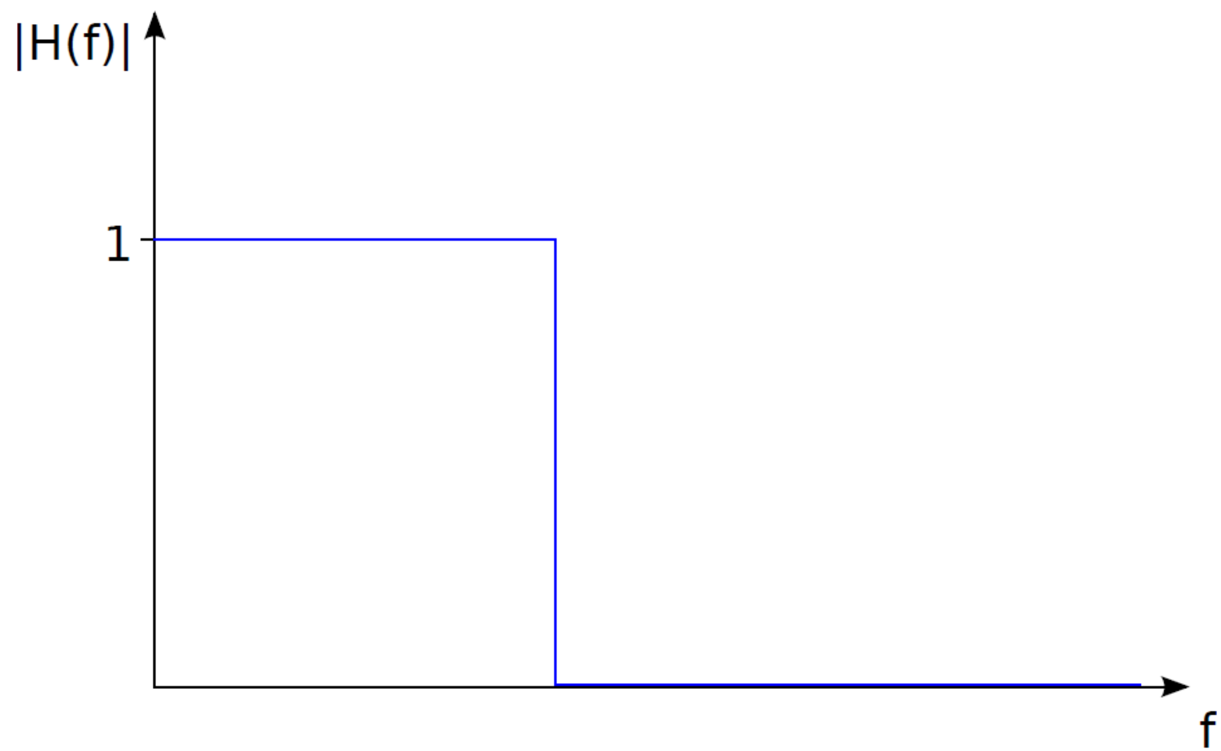




Christian-Albrechts-Universität zu Kiel

Filter types: Ideal low pass

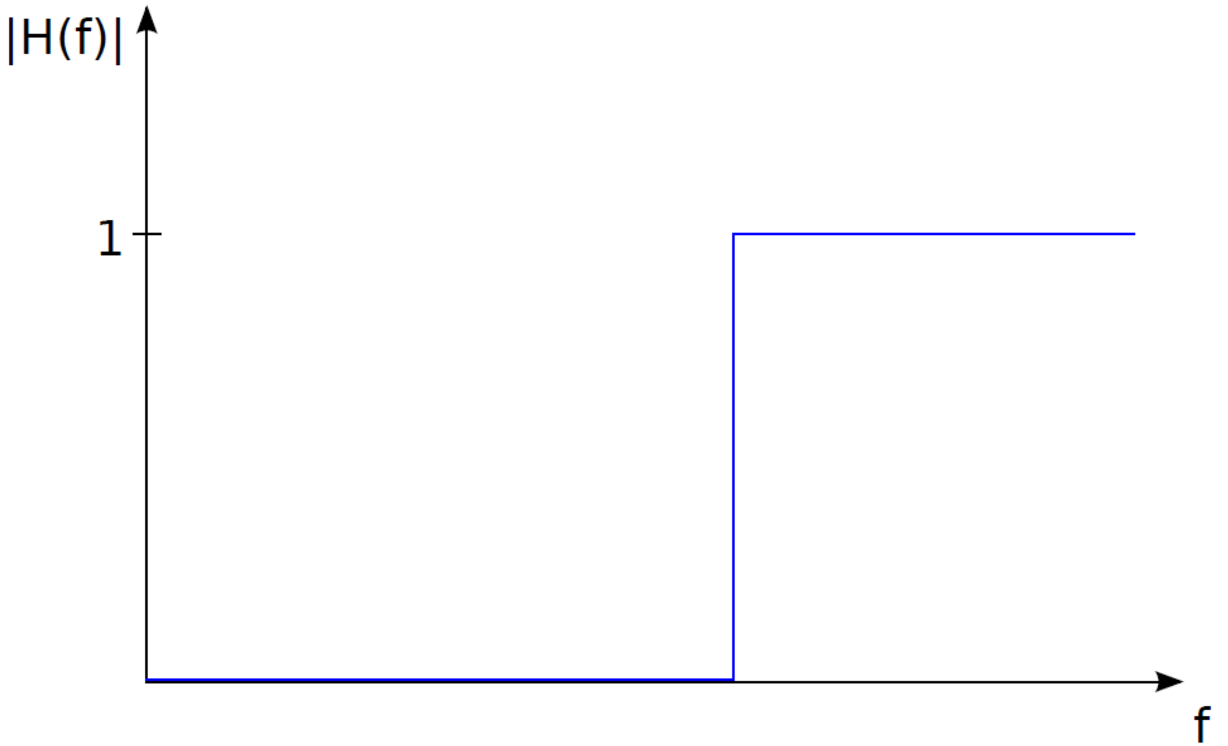
- Frequency response (amplitude as a function of frequency):



Christian-Albrechts-Universität zu Kiel

Filter types: Ideal high pass

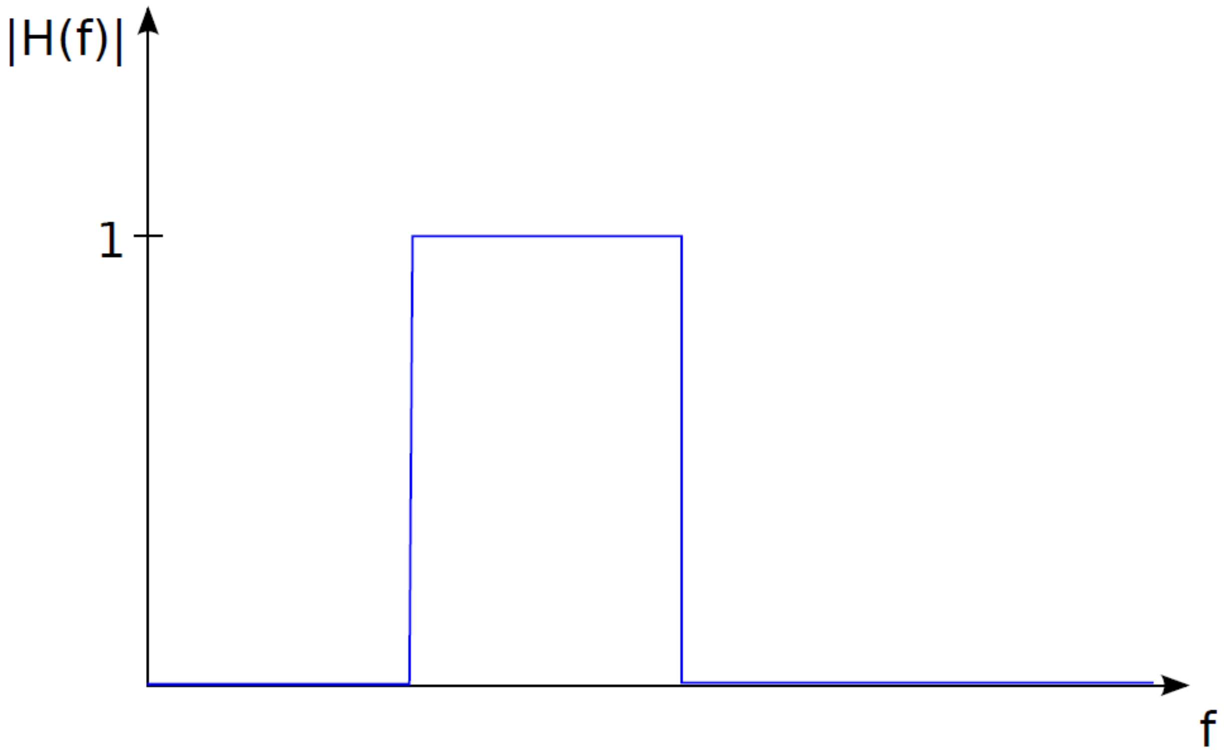
- Frequency response (amplitude as a function of frequency):



Christian-Albrechts-Universität zu Kiel

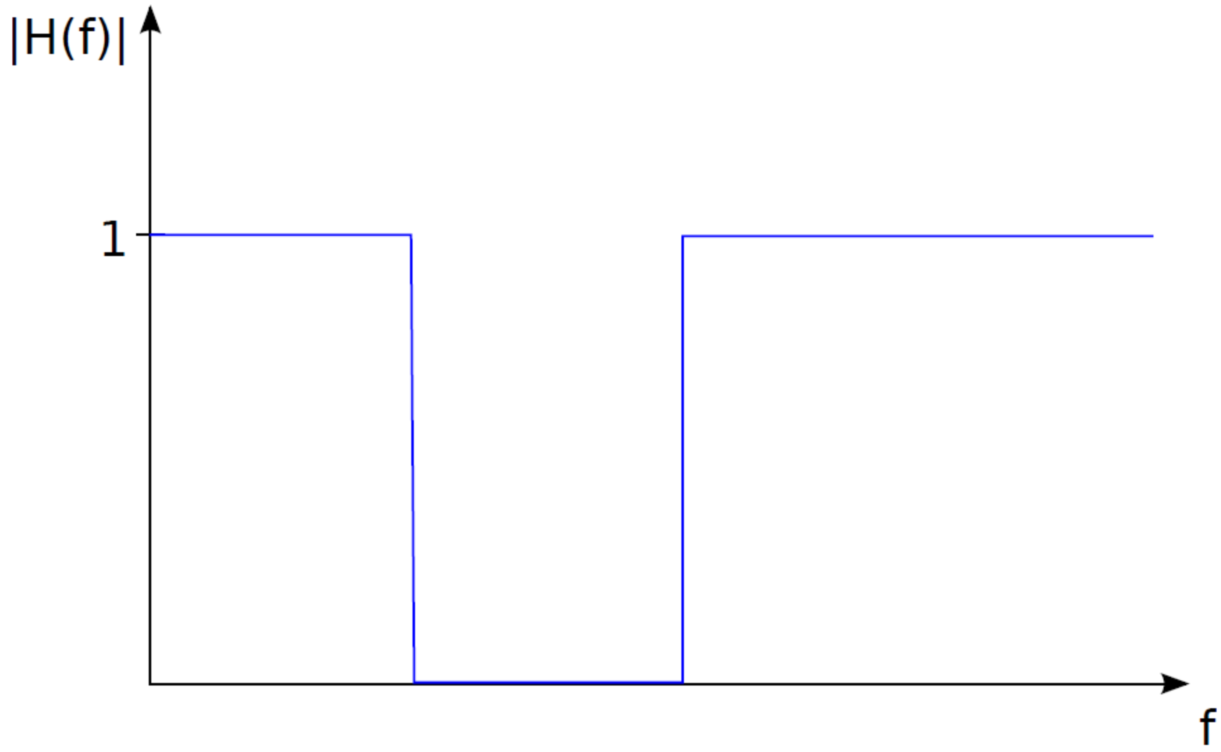
Filter types: Ideal pass stop

- Frequency response (amplitude as a function of frequency):



Filter types: Ideal band-stop

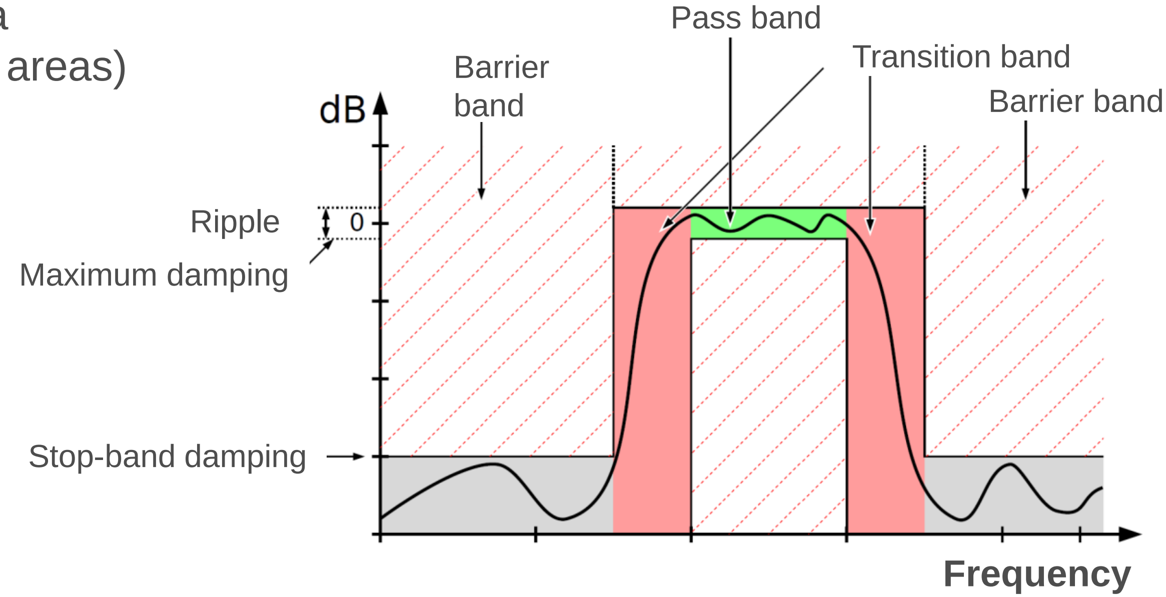
- Frequency response (amplitude as a function of frequency):



Christian-Albrechts-Universität zu Kiel

Ideal vs. realisable (practicable) filters

- Ideal filters (right-angled edges, constant barrier/passage) only achievable with filter order $N \to \infty$
- Means: Allow for tolerances
  - Passband (amplitude as unchanged as possible)
  - Blocking range (amplitude suppressed as far as possible)
  - Transition area (between both areas)



Christian-Albrechts-Universität zu Kiel

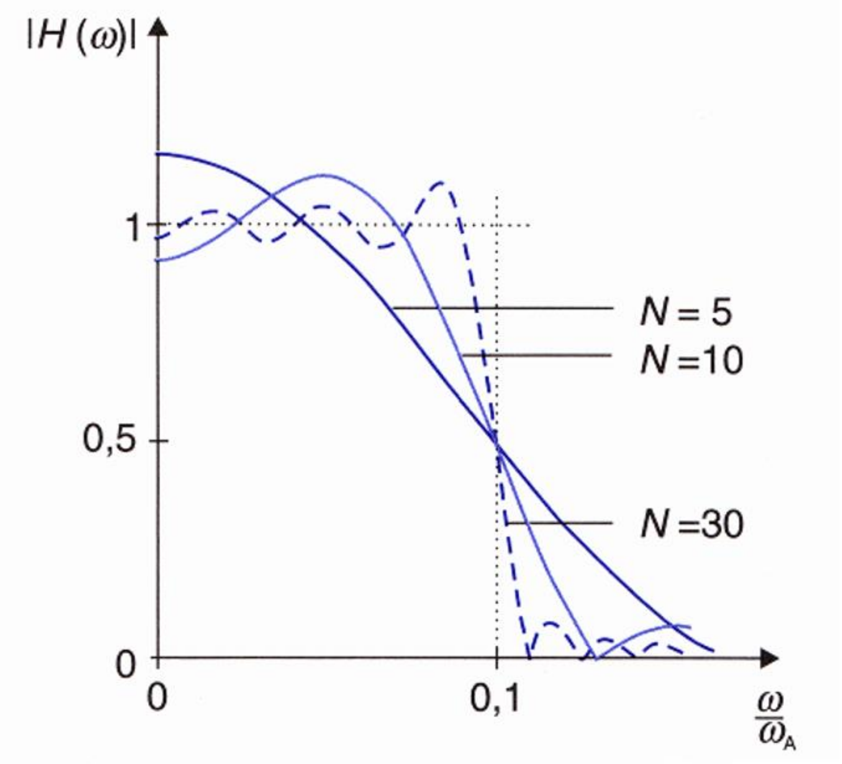
Influence of filter order

- Properties dependant on filter order N
  - Better filter properties
  - Higher expenses

- Presentation here:
  - Specification of the angular frequency

$$\omega = 2\pi f$$

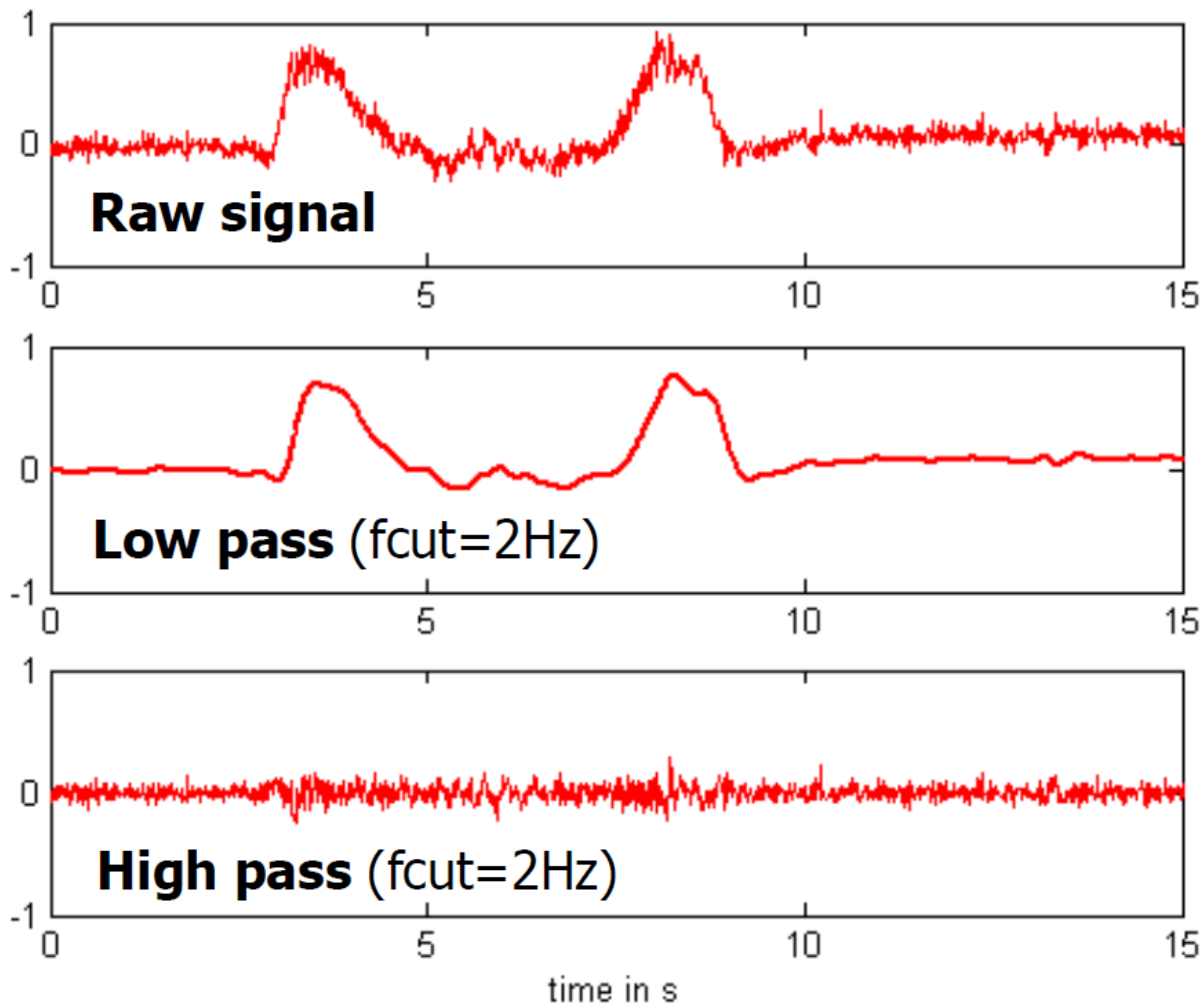
  - Relative to the sampling rate ω_a



Filter design

- Design of a filter → Determination of the filter coefficients
- For desired properties
  - Ripple (‘waviness’) in the passband and barrier band
  - Slope of the transition area
- Example for a low pass: A steep transition, a low ripple, and a blocking as complete as possible are to be aimed for.
- In general: At a given order N recursive filters achieve a better approximation to ideal conditions.
  - IIE more efficiently applicable
  - But: More difficult to design (instability)
- Manual filter design is not trivial
  - Software-supported design, e.g. in MATLAB or octave available

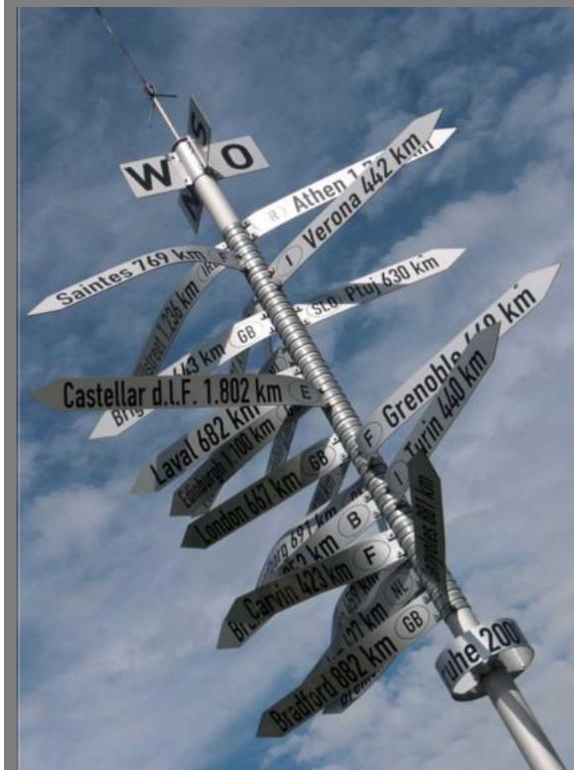
Example of filtering a signal:



Agenda

Christian-Albrechts-Universität zu Kiel

- Missing Values
- Scaling
- Outliers
- Data encoding
- Signal processing
- Conclusion and references



Christian-Albrechts-Universität zu Kiel

We discussed:

- Missing Values
- Scaling
- Outliers
- Data encoding
- Signal processing
- Conclusion
- Further readings

Students should now:

- be able to explain the tasks of the “pre-processing" step
- be able to introduce and compare approaches to handling missing values and noise and mechanisms for scaling, outlier detection and data coding.
- be able to apply simple forms of representation
- be able to explain filter types and their properties

Basic readings:

- Olaf Hochmuth, Beate Meffert
- “Werkzeuge der Signalverarbeitung: Grundlagen, Anwendungsbeispiele, Übungsaufgaben” (in German)
- Pearson Studium, 2004
- ISBN: 978-3827370655

Further readings

Christian-Albrechts-Universität zu Kiel

- [Mitsa 2010]: T. Mitsa: Temporal Data Mining, CRC Press, 2010.
- [Runkler 2010]: Runkler, Thomas A. Data Mining: Methoden und Algorithmen intelligenter Datenanalyse. Springer-Verlag, 2010.
- [Runkler 2000]: Runkler, Thomas A. "Information mining." Vieweg, Braunschweig/Wiesbaden (2000).
- [LKWL 2007]: Lin, J., Keogh, E., Wei, L., & Lonardi, S. (2007). Experiencing SAX: a novel symbolic representation of time series. Data Mining and knowledge discovery, 15(2), 107-144.

Christian-Albrechts-Universität zu Kiel

- Any questions…?