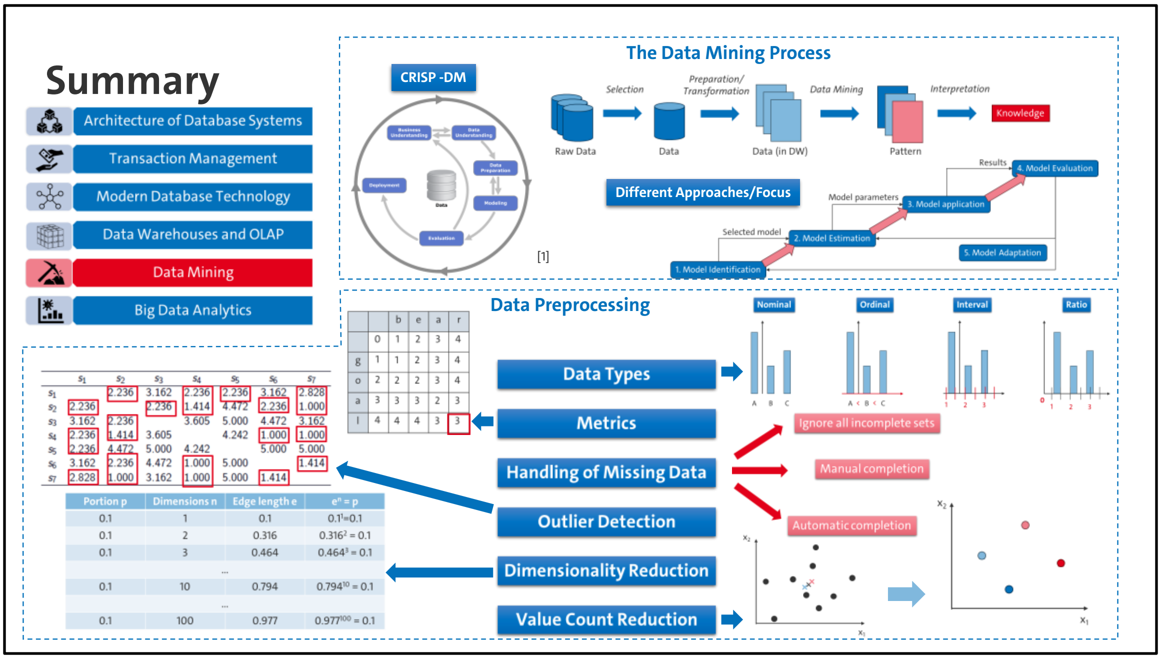


[1] source: Kenneth Jensen@Wikipedia based on
ftp://public.dhe.ibm.com/software/analytics/spss/documentation/modeler/18.0/en/ModelerCRISPDM.pdf

Time Series Model

Idea

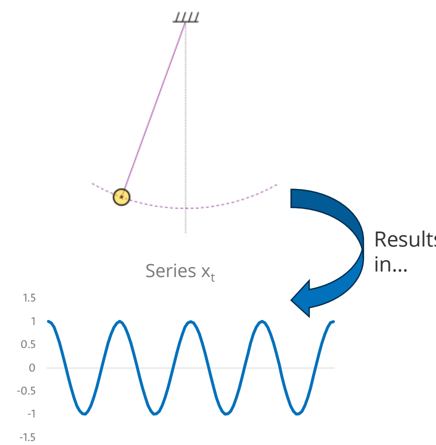
- Consider a time series as an unknown process combined with an unpredictable deviation
- It is measured at equidistant time instances and results in a sequence
- The unknown process is represented by a descriptive model P_t
- The unpredictable deviation res_t is assumed to be random, with
  - Expected value is 0
  - Variance is constant
  - Normally distributed
- Thus, the time series is represented by a descriptive model:

$$x_t = P_t + res_t$$

(P_t...process
res_t...residuals)

$$\underline{x}^T = (x_1, \dots, x_t, \dots, x_T)$$

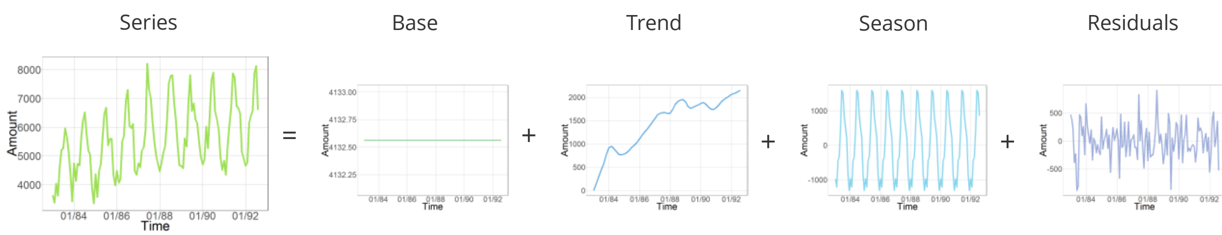
Example: Pendulum



Be aware of the sampling theorem!

Terminology

Additive composition $x_t = base_t + trend_t + season_t + res_t$

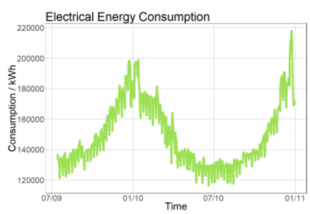


- Time series components
- Base: stationary part of the time series
- Trend: long-term change in the mean level
- Season: cyclical repeated behaviour
- Residuals: unstructured information assumed to be random

→ Often, base is part of the trend component!

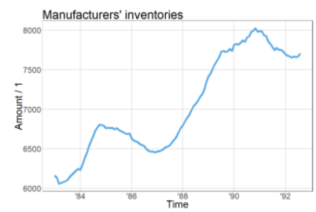
Time Series Occur in Many Domains

Economy



Utilities

Industry



Agriculture Animal Population Yield	**Chemistry** Chemical Concentration	**Computing** Online Advertisement Query Processing	**Crime** Robberies Drunkenness	**Demography** Population Rates Immigrants
Economic GDP Inflation	**Finance** Stock Development Price Development	**Health** Infections Suicide Rate	**Industry** Production Planning Inventory Planning	**Meteorology** Temperature Rainfall
Physics Earthquakes Sunspots	**Politics** Election Outcomes Sports	**Sports** Player Performance Team Performance	**Transport** Tourist Visits Airline Passengers	**Utilities** Energy Balancing Gas Production

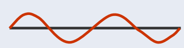
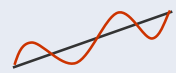
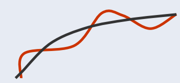
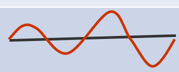
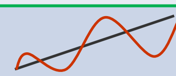
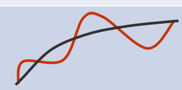
Data Mining

GDP – Gross domestic product

Time series analysis: Further reading

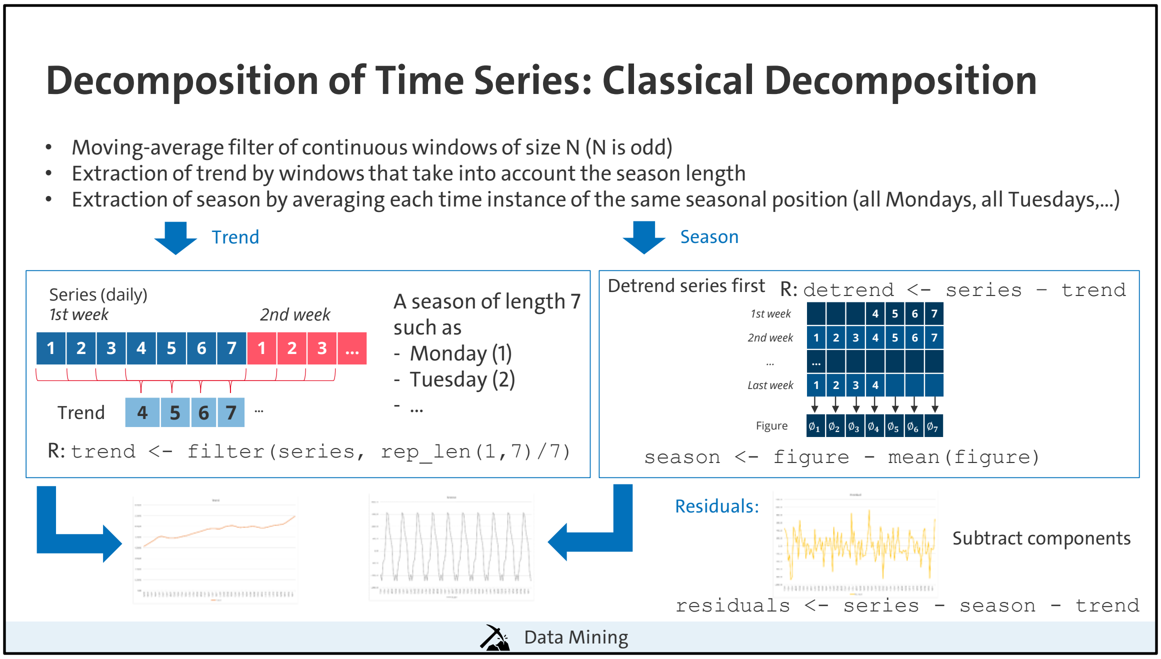
- Fulcher, B. D., Little, M. A., & Jones, N. S. (2013). Highly comparative time-series analysis: the empirical structure of time series and their methods.
- Wang, X., Smith, K., & Hyndman, R. J. (2006). Characteristic-based clustering for time series data. Data Mining and Knowledge Discovery.

Composition Types

	No Trend	Linear Trend	Non-linear Trend
No Season			
Additive Season			
Multiplicative Season			

The most common models

Pegels, C. C. (1969). Exponential Forecasting: Some New Variations. *Manage. Sci.*, *15*(5), 311–320.



Disadvantage: Does not decompose the endpoints

Average centering

- A technique needed if season length is even
- Take two moving averages and average their result

More techniques

<u>X11</u>

- Method published by the U.S. Bureau of the Census and used adopted by several statistical agencies
- Based on classical decomposition with several moving-average steps
- Advantages
    - Endpoints decomposed using predictions from ARIMA forecasting method
    - Extensions for holiday effects
    - Support slowly varying season representing changes in seasonal behavior
- Disadvantage: Only for quarterly and monthly time series

<u>STL</u>

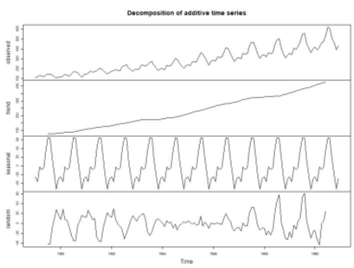
- Application of regression technique (*Loess smoothing*)

- Recursive application of smoothing and robustness checks
- Advantages
    - Support arbitrary season lengths
    - Versatile and robust decomposition technique
- Disadvantage: Parameters are not automatically retrieved

Examples in R

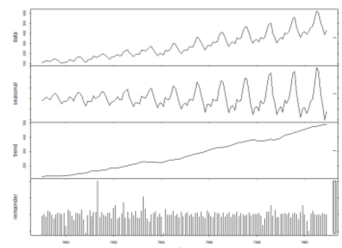
Classical decomposition

```
decomposed <-
decompose(AirPassengers)
```



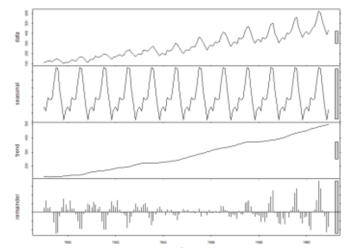
X11

```
library(seasonal)
decomposed <-
seas(AirPassengers, x11
= "")
```



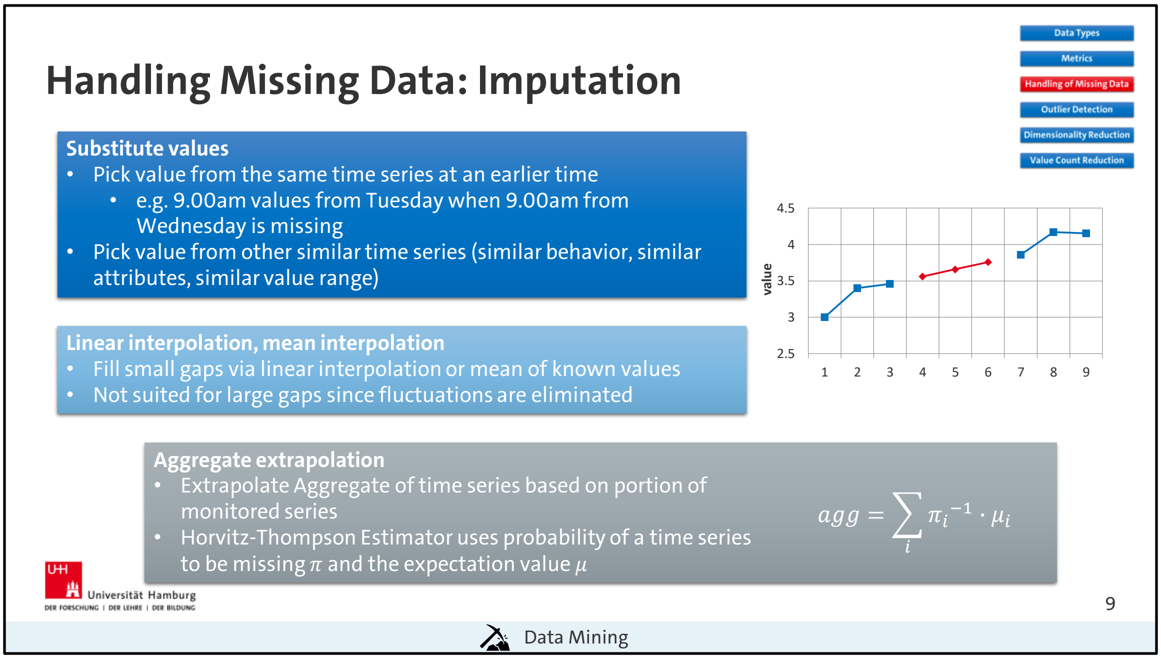
STL

```
decomposed <-
stl(AirPassengers,
s.window = “period”)
```



1st row: observed series
2nd row: trend (CD), season (X11/STL)
3rd row: seasonal (CD), trend (X11/STL)
4th row: residuals

Decomp. Features	DEC	X-11	STL
Arbitrary season length	✓	-	✓
Slowly varying season	-	✓	✓
Robustness	-	✓	✓
Endpoints	-	✓	✓



Ignoring/deleting all sets with incomplete cases does not work for time series

- Missing values have to be randomly distributed over the data set, otherwise deletion introduces bias
- Decrease reliability of analysis by decreasing sample size
- Aggregates would be too low
- Series where forecasts are requested are missing

Imputation

→ Replace missing values with substitute values

Base data

- Hot Deck – Use current data set for the replacement value calculation (most common)
- Cold Deck – Use another data set (e.g. older survey answers) for replacement value calculation

Singular imputation

Use only one technique to calculate replacement values

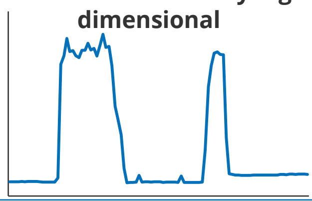
Multiple imputation (Ensemble)

Apply several single imputation techniques to impute the same value
Combine their replacement values in the final result

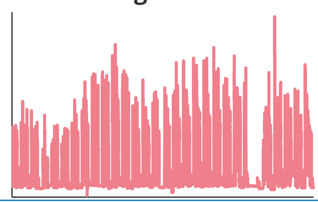
Data Types
Metrics
Handling of Missing Data
Outlier Detection
Dimensionality Reduction
Value Count Reduction

Curse of Dimensionality

Time series inherently high dimensional



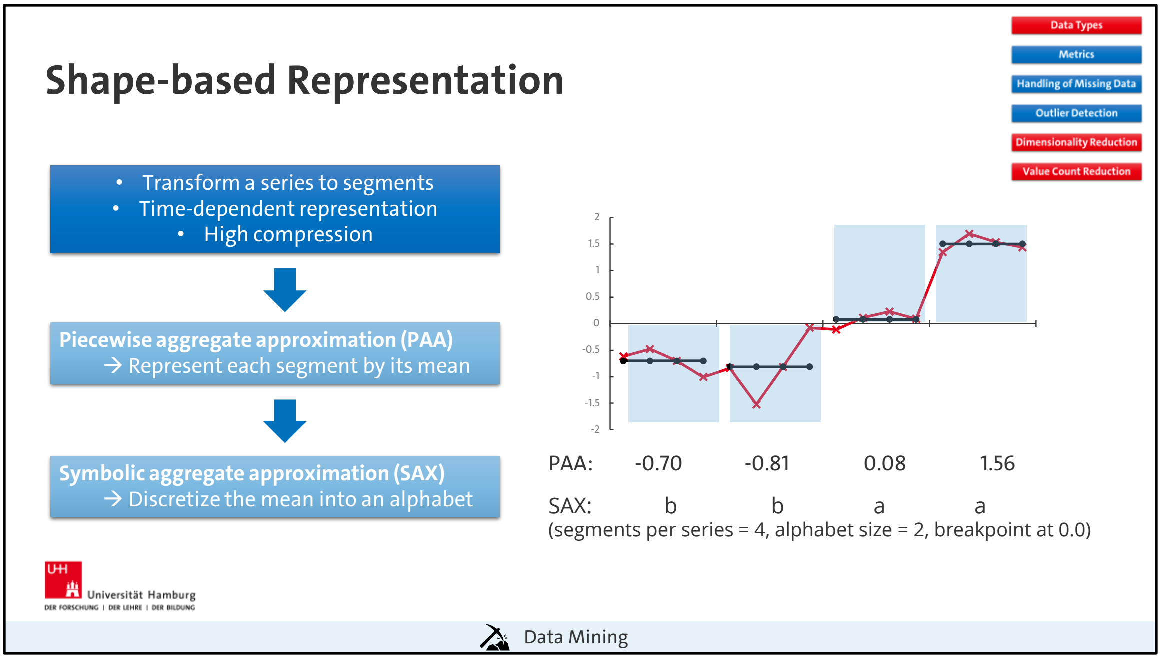
Increasing data volume



- How to capture the most meaningful information?
- How to prepare this information for data-mining tasks?

Time Series Engineering

Raw-value-based | **Shape-based** | **Feature-based** | **Model-based**



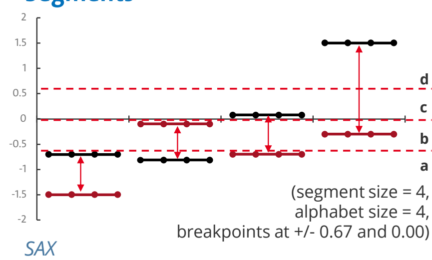
Different data types enable different distance measures

Further reading

Jessica Lin, Eamonn J. Keogh, Stefano Lonardi, and Bill Chiu. A Symbolic Representation of Time Series, with Implications for Streaming Al- gorithms. In Workshop Proc. of SIGMOD, 2003

The SAX Distance

Segments



SAX

$\hat{x}^1$	a	a	c	d
$\hat{x}^2$	a	b	a	b

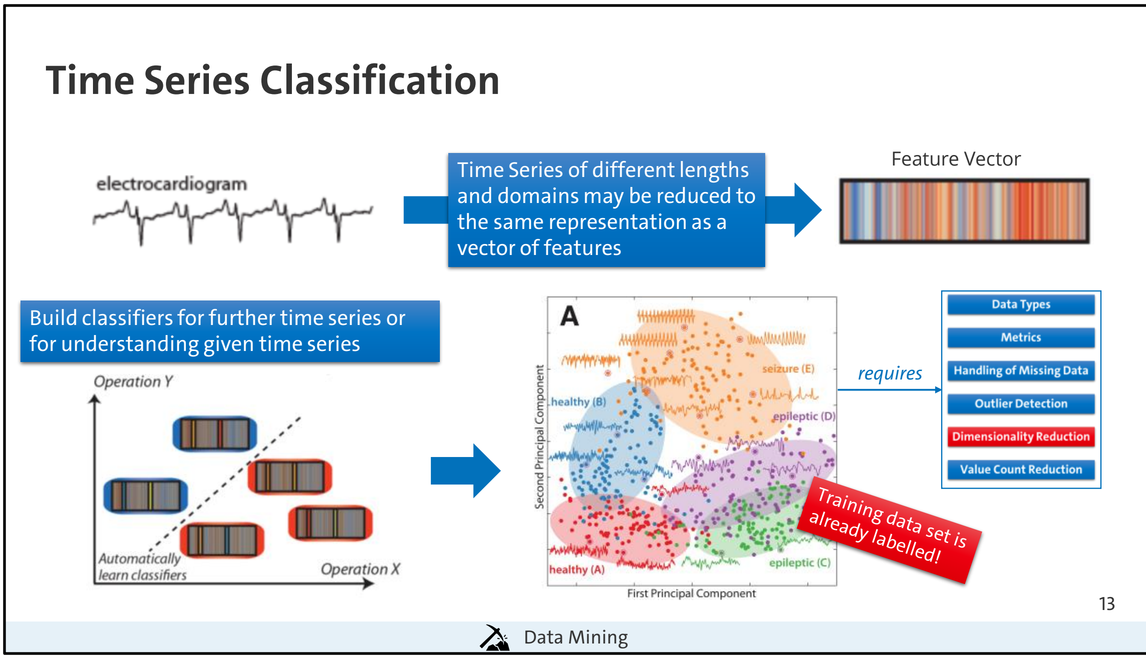
$$d_{SAX}(\hat{\underline{x}}^i, \hat{\underline{x}}^j) = \sqrt{T/W} \cdot \sqrt{\sum_{w=1}^{W} cell(\hat{x}_w^i, \hat{x}_w^j)^2}$$

cell returns the minimum distance of two symbols:

	a	b	c	d
a	0	0	0.67	1.34
b	0	0	0	0.67
c	0.67	0	0	0
d	1.34	0.67	0	0

- Same and neighbouring symbols = 0
- T – length of time series
- W – length of string/number of segments
- PAA distance measure

$$d_{PAA}(\underline{\bar{x}}^i, \underline{\bar{x}}^j) = \sqrt{T/W} \cdot \sqrt{\sum_{w=1}^{W} (\bar{x}_w^i, \bar{x}_w^j)^2}$$



Represent time series by their features

- Features cover large time series domains
- Time Series of different lengths and domains may be reduced to the same representation as a vector of features
- Fulcher reports ~9000 features from the literature

Classification

- Select interesting features with high classification performance
- Build classifiers for further time series or for understanding given time series

Advantages

- Gain insights into differences between classes of labeled time series datasets

Case Study: EEG Recordings

- EEG data (time series) from healthy patients (A, B), epileptic patients (C, D), and patients with epileptic seizure (E)
- Build classifier that differentiates between groups A and E

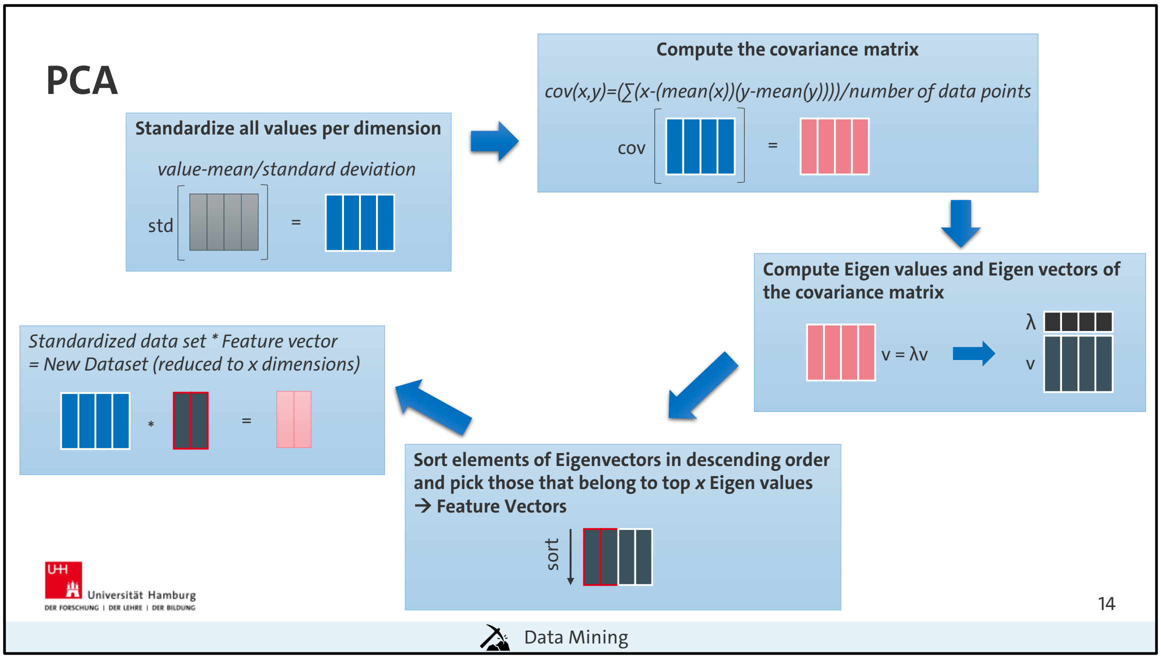
Results

- Features are reduced to 2 with Principal Component Analysis
- Groups A and E can be well discriminated
- Feature vectors are as performant as other analytical, but more complex time

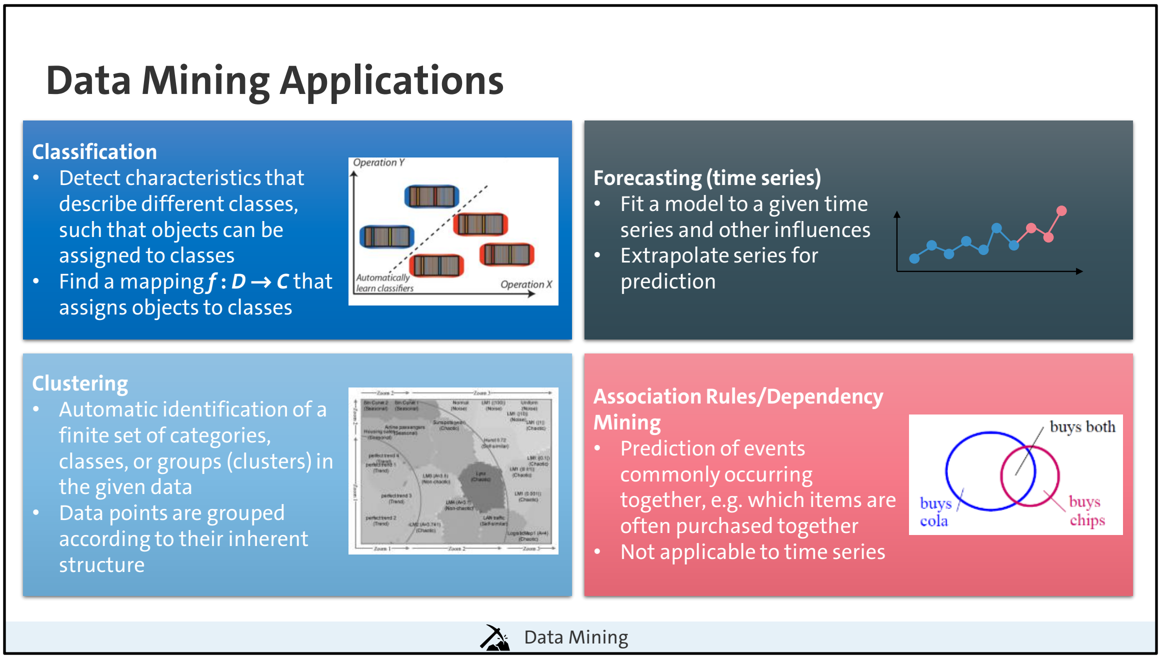
series transformations
- Support vector machine
- Discrete wavelet transform

Further Reading

Fulcher, B. D., Little, M. A., & Jones, N. S. (2013). Highly comparative time-series analysis: the empirical structure of time series and their methods.



PCA = Principal Component Analysis
Method for reducing dimensions by merging them



Classification → Find (and use) the classifiers
Classes are predefined → No unsupervised learning
No object belongs to several classes

<u>Given</u>

- Set of objects (database) $D = \{o_1, o_2, \ldots, o_m\}$ where each object o_i corresponds to a k-dimensional vector $<o_{i,1}, \ldots, o_{i,k}>$
- Set of classes $C = \{c_1, c_2, \ldots, c_n\}$ (usually: $|D| >> |C|$)
- Set of labelled training objects $T \subset D$

<u>Goal</u>

- Find mapping f that assigns objects to classes

Clustering → Find the classes/groups/categories

- Procedure: Grouping of data points according to their inherent structure
    - Points within the same cluster should be as similar as possible
    - Points in different clusters should be as dissimilar as possible
    - Learning without teacher
- Clustering approaches: partitioning, hierarchical, incremental, …
- Problem: What is the optimal clustering? What is the meaning of “optimal”?

Dependency Mining

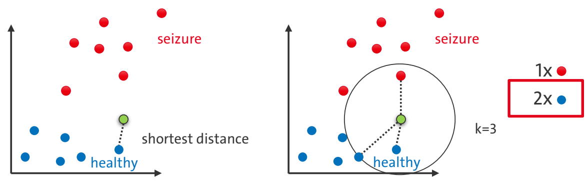
- Association rules: Rules of the form $a \wedge b \wedge \ldots \wedge c \rightarrow d \wedge e$
    - Example: *buys cola* → *buys chips*
- Challenge: Finding good combinations of premises and conclusions is a combinatorial problem

Classification Methods

kNN | Threshold based | Decision trees

k-Nearest Neighbors

→ Requires training data set and distance function, e.g. Euclidean distance



- Training data set is only stored → "lazy learning"
- No generalization of the available training data
- Classification effort grows linearly with amount of training data → One distance computation per training object
- Optimization possible, e.g. via filtering, pruning

Nearest Neighbor

Direct approach: training objects are
- directly stored in the classifier and
- used for classification

Given:
- Training data set $T = \{o_1, o_2, \dots, o_r\}$ with class labels c_t: $T \rightarrow C$
- Object distance function d

Nearest neighbor:
- The class of an object o is set to the class label of its nearest training object o, i.e.

$c(o) = c_t(o^*)$ where o = arg $\min_{o' \in T} d(o, o')$

(here we assume that the distance $d(o, o')$ is different for every $o' \in T$)

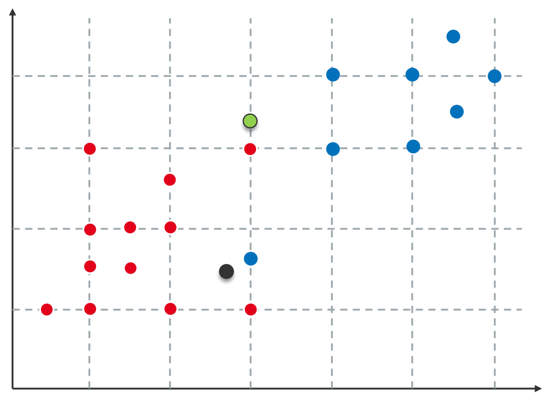
kNN

- Determine the set N of the k nearest neighbors of o in T
- Choose the class with the maximum number of training objects in N, i.e.

$c(o) = \arg\max_{c \in C} |\{o' | o' \; 2 \; N, c_t(o') = c\}|$

- More robust against singular data points
- But more expensive

Exercise kNN



- Which cluster(s) do the green and black objects belong to if k = 1, 2, 3, or 4?
- Which problems can occur?
- How can we solve them?

Classification Methods

kNN | **Threshold based** | Decision trees

Simple generalizing model for classification with two classes

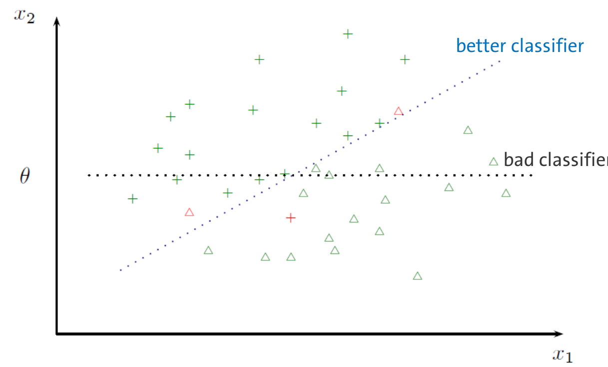
Threshold divides the data space into two subspaces along a single dimension

$$c(o_i) = \begin{cases} c_1, & o_{i,j} > \theta \\ c_2, & else \end{cases}$$

→ Analogue separation criteria for non-numeric data

Optimal threshold: Minimizes classification error for training objects
→ For numeric samples: minimal total distance (sum) of misclassified samples:

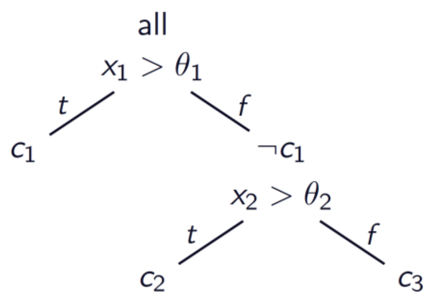
$$\theta = \text{arg min}_\theta \sum_{o_i \in T, c(o_i) \neq c_t(o_i)} |o_{i,j} - \theta|$$

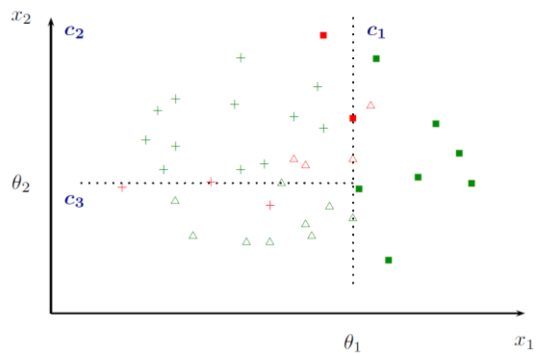


Classification Methods

kNN | Threshold based | **Decision trees**

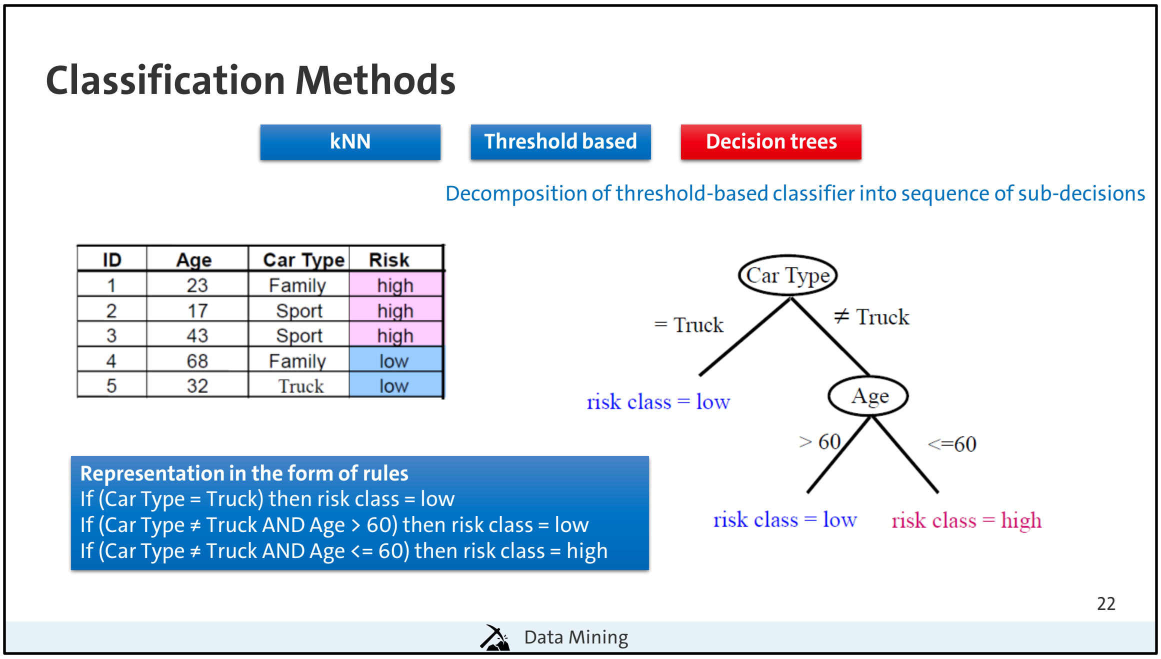
Decomposition of threshold-based classifier into sequence of sub-decisions





- Multi-branch splits possible
- Each leaf node represents a class $c \in C$
- Finding the optimal decision tree is NP complete

  → Deterministic (non-backtracking), greedy algorithms



- Usage of the decision tree to make predictions:
  → Top-down traversing of the tree from the root to one of the leaf nodes
  → Assignment of the object to the class of the resultant leaf node

Construction of a Decision Tree

Basic algorithm:

- Initially, all training objects belong to the root
- Selection of the next attribute (split strategy), e.g. by maximization of the information gain
- Partitioning of the training objects with the selected split attribute
- Algorithm is applied to each partition recursively

Stop criterion:

- No further split attributes
- All training objects of a node belong to the same class

Types of splits:

- Categorical attribute: Split condition of the form “attribute = a” or “attribute ∈ set” (many possible subsets)
- Numerical attribute: Split condition of the form “attribute < a” (many possible split points)

- Example: ID3 - split along a dimension as to maximize information gain
- Decision rules can be extracted from a decision tree
    - IF part: combine all tests on the path from the root node to the leaf node
    - THEN part: the final classification
- Enables a simple extraction of ‘real’ knowledge from the learned classifier

Classification Methods

kNN | Threshold based | **Decision trees**

Decomposition of threshold-based classifier into sequence of sub-decisions

The Problem of overfitting
→ Splitting until no object is misclassified usually means to adapt the classifier too much to the training data
→ Cut-off-criterion or post-pruning required