Christian-Albrechts-Universität zu Kiel

Lecture
„Intelligent Systems“

Chapter 4: Time-Series Features

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

Winter term 2020/2021

About this chapter

Christian-Albrechts-Universität zu Kiel

Contents

- Feature extraction
- Time-independent features
- Time-dependent features
- Feature selection
- Taxonomy of feature selection
- Feature selection algorithms
- Feature transformation
- Principle Component Analysis
- Conclusion and further readings

Goals

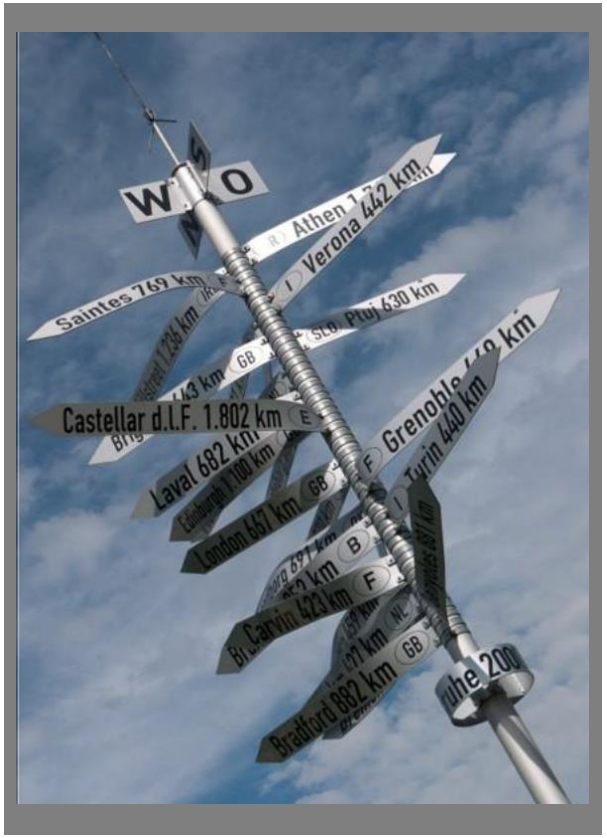
Students should be able to:

- motivate and define the three steps feature extraction, feature selection, and feature transformation.
- introduce different kinds of features.
- compare different algorithms and their applicability.
- distinguish filters and wrappers.
- apply accuracy, information, distance, dependency, and consistency measures.

Agenda

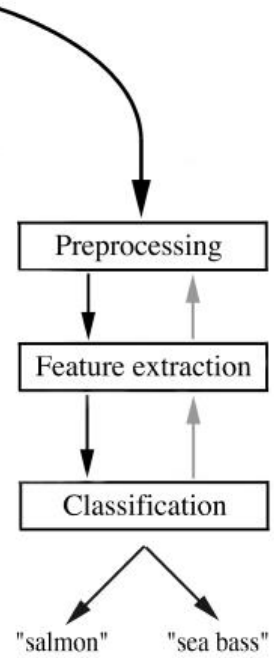
Christian-Albrechts-Universität zu Kiel

- Feature extraction
- Time-independent features
- Time-dependent features
- Feature selection
- Taxonomy of feature selection
- Feature selection algorithms
- Feature transformation
- Principle Component Analysis
- Conclusion and further readings



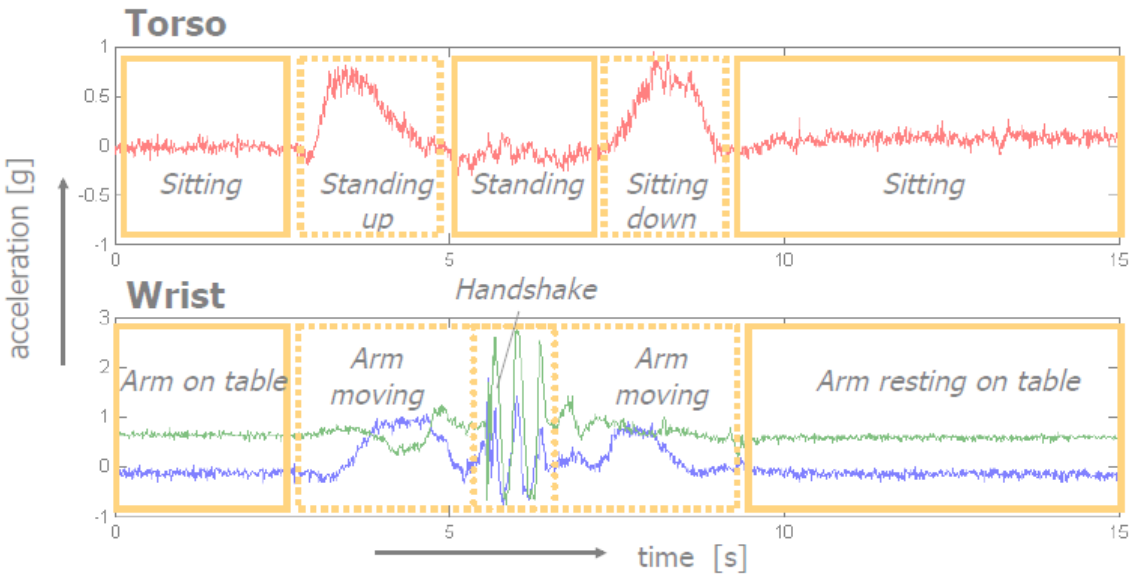
Feature extraction

- Example:
  - Classification of fish
  - Sensor: Camera
- Output signal: grey scale images
- Problems with raw data
  - High-dimensional (camera resolution, e.g. 640x480, Vector of length 307,200)
  - Contains a lot of unnecessary (i.e. not problem-relevant) information



Feature extraction

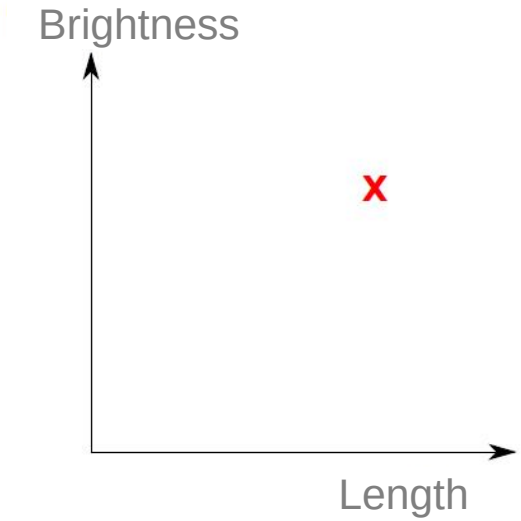
- Example: Activity detection
  - Sensors: Accelerometers
  - Output signals: Time series
- Problems with raw data
  - High-dimensional → Vector length = length of activities
  - Variable length



Feature Extraction: Goals

- Abstract representation of input patterns by features
- Transformation of input data into the feature space (problem-specific representation)
  - Reduction of dimensionality to essential information (effort, robustness)
  - Elimination of "misleading" or irrelevant information
  - Constant dimensionality
- "Simplification" of the decision space

⇒



Features: Requirements

- Continue to describe input patterns in such a way that the actual task can still be performed (distinguishability).
  - For classification: patterns of the same class very similar, for different classes very dissimilar
  - For clustering: analogous for "natural" groups within data
- Invariance against problem-specific irrelevant transformations
  - Translation
  - Rotation
  - Scaling
  - ...
- E.g. characteristics describing properties such as shape, colour, texture, edges, curvature

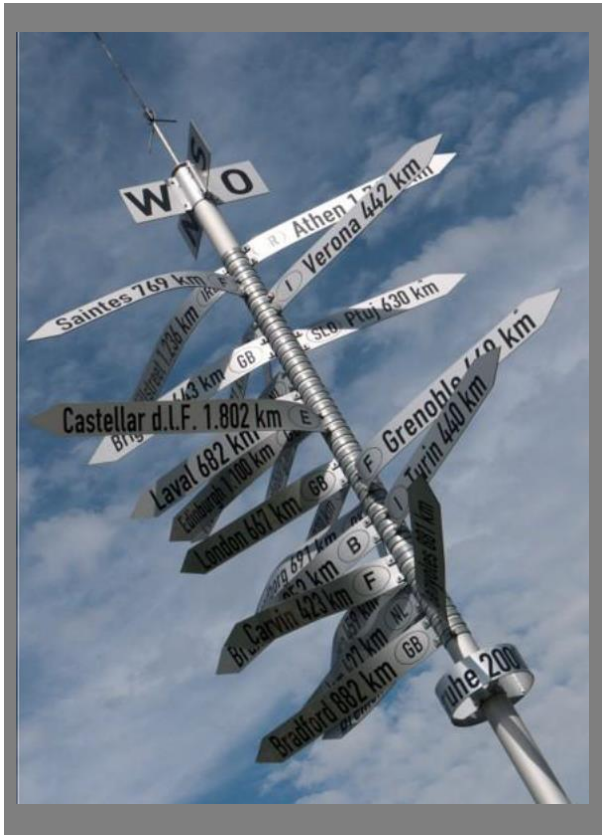
Which features for which task?

- Features are derived from the input signals based on an analysis of the task.
  - Understanding the problem is critical!
  - Decision often based on experience and intuition
- Highly application-specific
- Possible features result from the type of input signals
  - Time- and space-independent data
    - The input signal was recorded at a certain time and is only relevant for classification at this time.
  - Time- and space-dependent data
    - Continuously measured data where development over time is essential
    - Images (space-dependent)

Agenda

Christian-Albrechts-Universität zu Kiel

- Feature extraction
- Time-independent features
- Time-dependent features
- Feature selection
- Taxonomy of feature selection
- Feature selection algorithms
- Feature transformation
- Principle Component Analysis
- Conclusion and further readings



Time-independent features

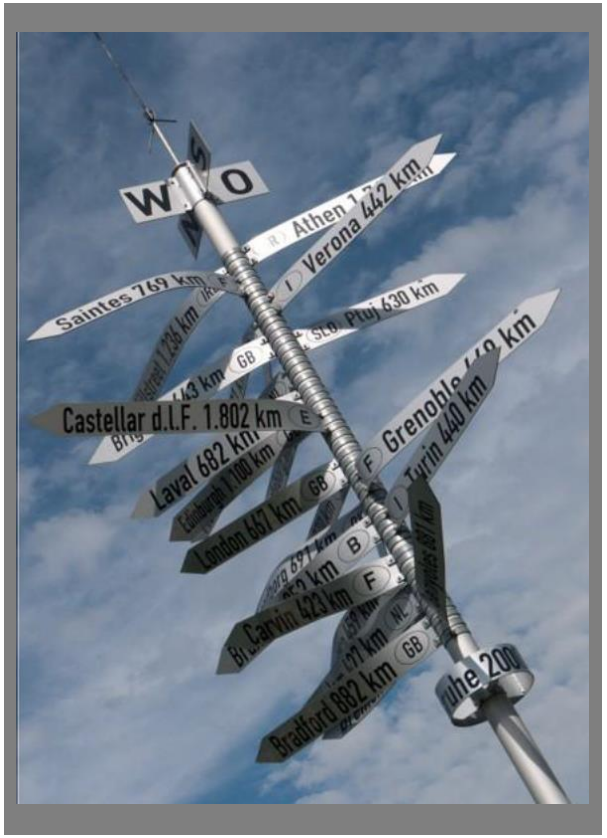
- Example: Absolute value of a vector as a feature
- Acceleration sensor: Acceleration(a_x, a_y, a_z) depends on the orientation of the smartphone on the body.
  → It is not possible to separate the motion part from the gravity part.
- Also, the change of the absolute value $|a| = \sqrt{a_x^2 + a_y^2 + a_z^2}$ contains characteristics of most movements.
  → The absolute value is invariant to the orientation of the smartphone!
- Means: Use the absolute value as a feature instead of the raw data.

Time-independent data

- Further possibilities to extract features from raw data:
- Coordinate transformation
  - Selection of the reference system (e.g. absolute vs. relative velocity)
  - Use of polar coordinates
- General functions of subsets of the sensor signals used
  - Component ratios
  - Linear combinations of the components
  - Logical statements about the components

Agenda

- Feature extraction
- Time-independent features
- **Time-dependent features**
- Feature selection
- Taxonomy of feature selection
- Feature selection algorithms
- Feature transformation
- Principle Component Analysis
- Conclusion and further readings

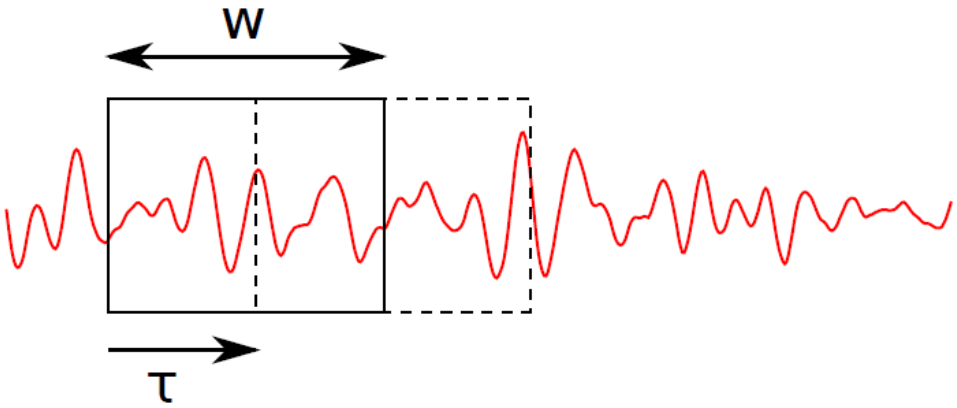


Processing time-series data

- Various possibilities:
  - Consider the entire time series
    - Bad for "online" processing
    - Problematic if individual subsections (e.g. activities) are to be classified
  - Processing parts of the time series with window method
    - Sliding Window
    - Growing Window

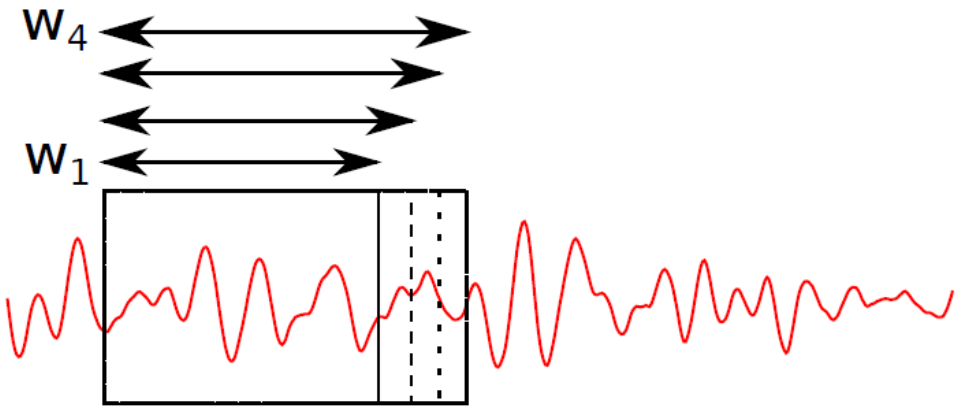
Sliding window

- Features are calculated in a fixed-length window that moves across the input signal, each of which affects the entire window.
  - Hiding the remaining information
  - Reduction of dimensionality
- Two degrees of freedom:
  - Window size ω: Must be adapted to the typical duration of the phenomena to be found
  - Time offset τ: Update cycle of one or more time steps



Growing window

- In a window of increasing length, characteristics are calculated that affect the entire window.
  - Hiding the remaining information
  - Reduction of dimensionality
  - Length of the window grows: $w_1 < w_2 < w_3 < w_4$
  - The starting point is fixed
- After detection: Window is shifted (time steps) and reduced, the procedure repeats itself.
- Phenomena of variable length can be detected.



Statistical features of time-series

→ Determined based on the entire time-series or in windows

- Means:

$$\mu = \frac{1}{N}\sum_{i=1}^{N} x_i$$

- Root mean square (RMS) / “effective value“:

$$rms = \sqrt{\frac{1}{N}\sum_{i=1}^{N} x_i}$$

- Variance:

$$\sigma^2 = \frac{1}{N-1}\sum_{i=1}^{N} (x_i - \mu)^2$$

- Standard deviation:

$$\sigma = \sqrt{\sigma^2}$$

Statistical features in time-series

→ Determined based on the entire time-series or in windows

- Minimum and maximum values
  - Difference Max-Min (span)
- Median (“middle value”) or Mode (most often occurring value)
- Threshold values (e.g. zero crossings)
  - Over/underflow yes/no?
  - How often?

Statistical features in time-series

→ Determined based on the entire time-series or in windows

- Average of the absolute values of the first differences:

$$\gamma = \frac{1}{N-1}\sum_{i=1}^{N-1}|x_{i+1} - x_i|$$

- Average of the absolute values of the second differences:

$$\gamma = \frac{1}{N-2}\sum_{i=1}^{N-2}|x_{i+2} - x_i|$$

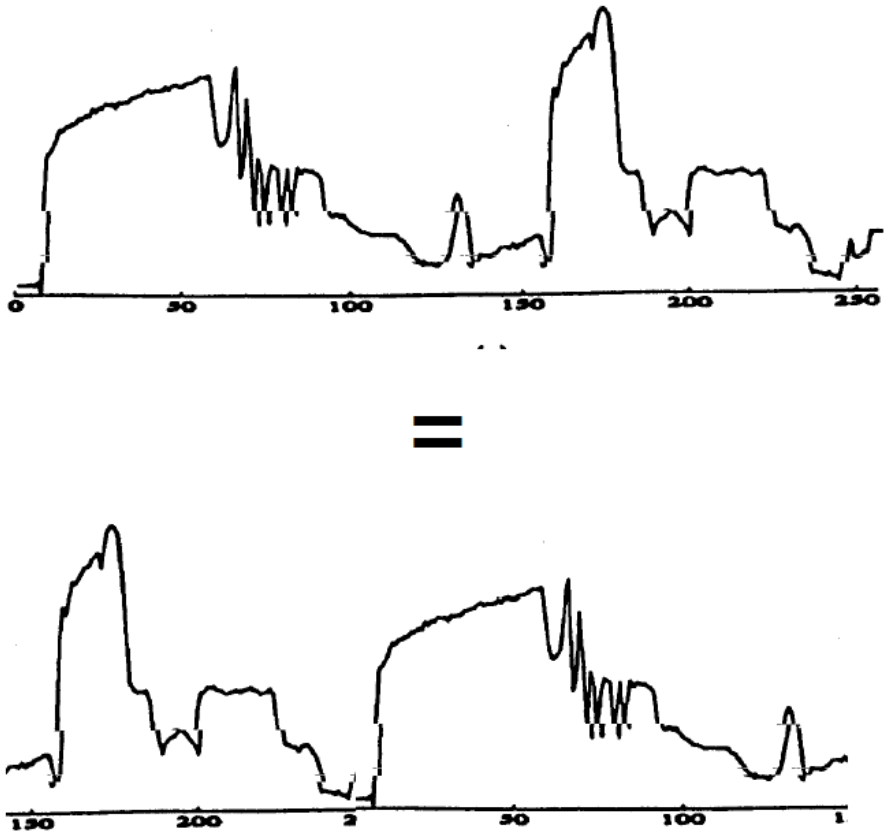
- If necessary normalised with the standard deviation σ

Statistical features: Assessment

- Advantages:
  - All-time series are mapped to vectors of the same length
  - (Mostly) Insensitive to typical disturbances (outliers, noise, ...)
- Disadvantages:
  - Sliding window: Choice of window size critical and problem-specific
    - Too small: Relevant phenomena are not recorded
    - Too big: class changes are not detected
  - Growing window:
    - The longer the window becomes, the more phenomena are smoothed out and possibly no longer detected.

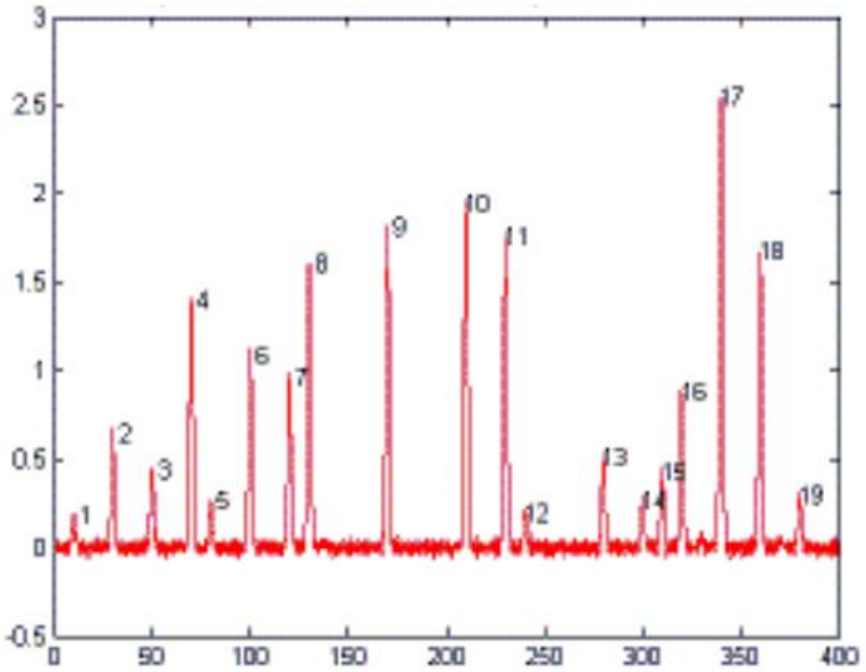
Further disadvantage:

- Features are cumulated over the entire window length
- Temporal structure / behaviour is not or only insufficiently covered



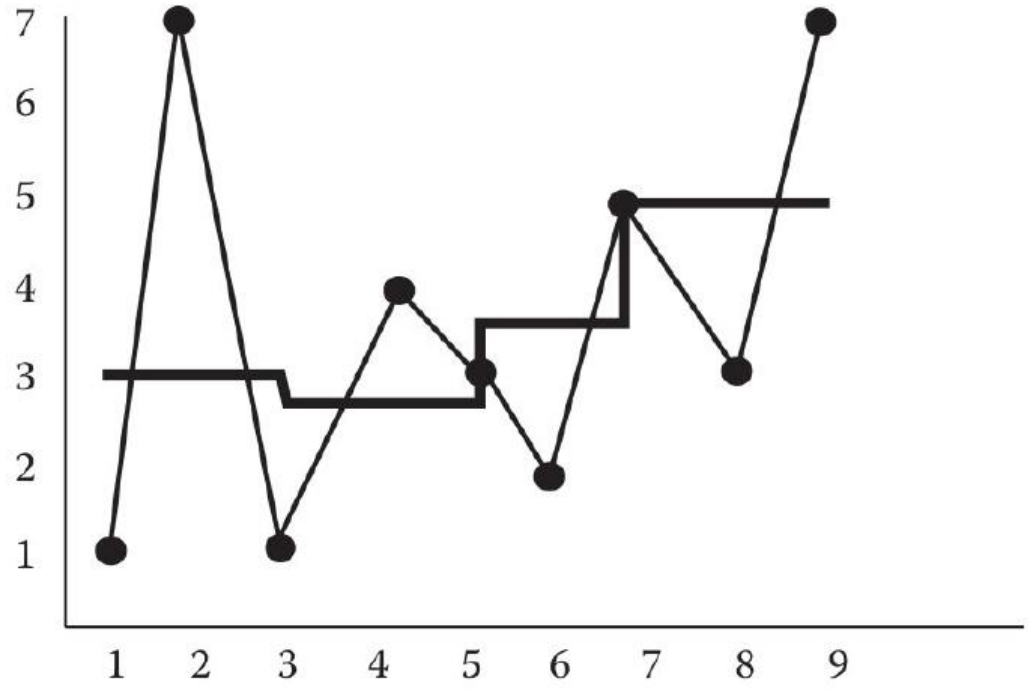
Description by peaks

- Possible parameters
  - Position of the peaks
  - Height of the peaks
  - Width of the peaks
- For a constant number of dimensions:
  - n highest peaks of the window



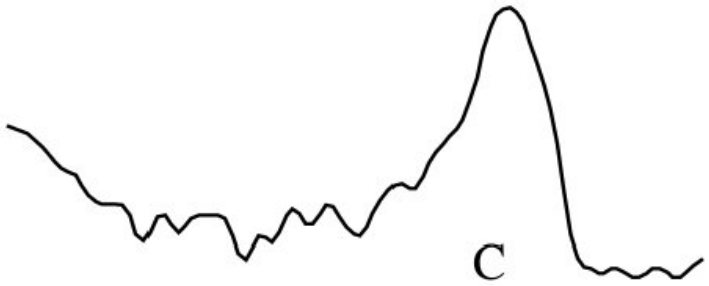
Piecewise Aggregate Approximation (PAA)

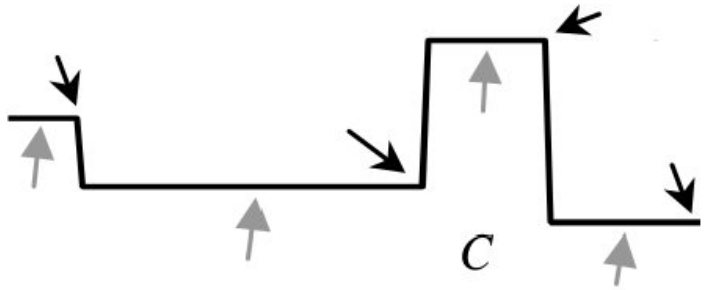
- Time series is divided into sections of equal length and these are replaced by a constant value (average) resulting from the sections.



Adaptive Piecewise Aggregate Approximation (APAA)

- Main difference to PAA: variable length of sections
- Basic idea: Adaptive to local details of a time-series, i.e. phases with more “dynamics" are divided into smaller sections, phases without significant information into longer sections.

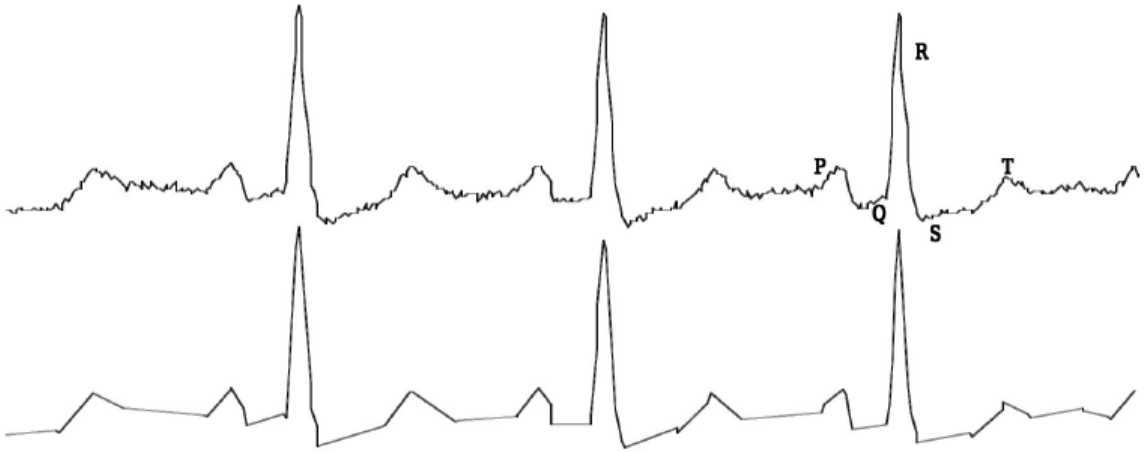




- Finding the optimal sections not trivial
  - Optimal concerning an error measure that evaluates the difference between the approximation and the true signal
- Solution e.g. utilising heuristics or dynamic programming

Piecewise Linear Approximation (PLA)

- Basic idea:
  - Again divided into sections
  - Each section is represented by a straight line
  - Vector: length and slope



Polynomial approximation

- Idea: Represent time-series / intervals of time-series by a polynomial of a given degree
- Degree n refers to highest exponent

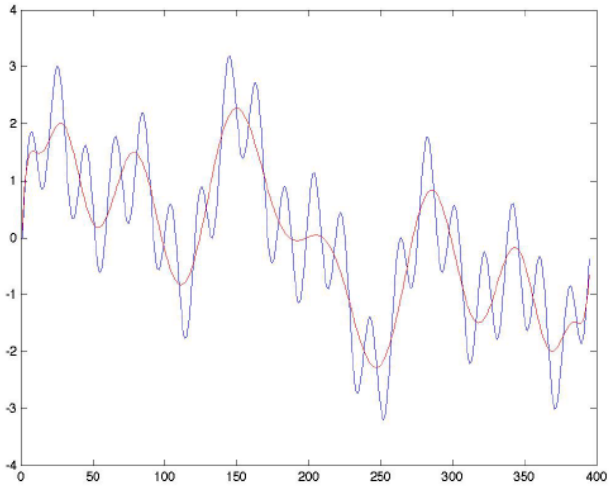
$$p(x) = a_0 + a_1 x^1 + a_2 x^2 + \cdots + + a_n x^n$$

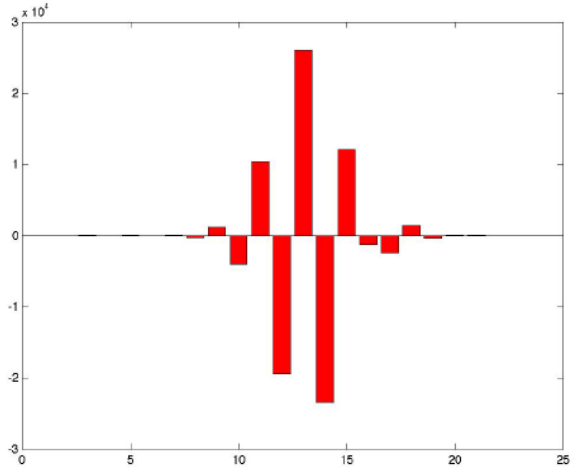
- Signal is reduced to the coefficients $a_0, a_1, a_2, \ldots, , a_n$
  - Considers temporal behaviour
  - But: Still able to represent a long signal with only a few values

Please note: The features are just the coefficients!

Polynomial approximation

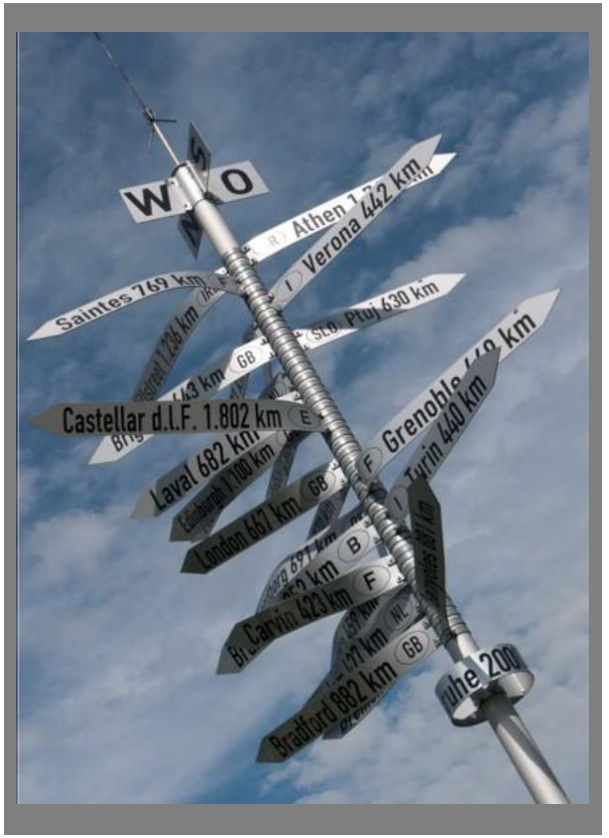
- Advantage:
  - Very compact description of time-series
  - Perfect representation for prediction
- Disadvantage
  - Which is the appropriate degree?





Agenda

- Feature extraction
- Time-independent features
- Time-dependent features
- **Feature selection**
- Taxonomy of feature selection
- Feature selection algorithms
- Feature transformation
- Principle Component Analysis
- Conclusion and further readings



The task of feature selection:

- The task of feature selection is to select an optimal subset from the set of available features in a data set.
- Depending on the task, the set of "important" features (to be selected) should
  - offer the highest accuracy (classification, regression, ...),
  - be minimum with a given minimum accuracy,
  - have low costs (monetary costs or calculation costs),
  - be better understandable,
  - ...
- Here: Focus on accuracy aspect for classification tasks

Feature selection: Task

- Important (good) characteristics need to be distinguished from unimportant (bad) characteristics.
- Other and related tasks are:
  - Reduction of the sample number after feature selection
  - Establishment of a ranking of features (feature ranking)
  - Improvement of the visualisation of data
  - Construction of (meta-)features, e.g. by main axis transformation and selection of the most important main components (feature construction)

What is the benefit of feature selection?

- Time savings due to smaller amounts of data (fewer features, possibly fewer patterns); the calculation effort for the subsequent ML algorithms (e.g. for classification) is reduced.
- Better data quality through the elimination of unimportant (e.g. redundant or heavily noisy) information
- Improved generalisation performance due to a smaller number of features
- Cost savings e.g. through the well-directed use of relevant features (no need for unnecessary sensor technology) or in the case of time-consuming calculation of features (e.g. signal processing)

Definition (for classification tasks)

Definition: Feature Selection

Let F be a full set of features and $G \sqsubseteq F$. In general, the goal of feature selection can be formalised as selecting a minimum subset G such that $P(C|G)$ is equal or as close as possible to $P(C|F)$, where $P(C|G)$ is the probability distribution of different classes given the feature values in G and $P(C|F)$ is the original distribution given the feature values in F. We call such a minimum subset an optimal subset.

[Yu, Liu 2004]

Conditional properties

- $P(C|G)$: Probability for class C assuming (precondition) that G occurred.
- $P(C, G)$: Probability for the joint occurrence of C and G (compound probability).
- It applies:

$$P(C, G) = P(C|G) \cdot P(G) = P(G|C) \cdot P(C)$$

Informal clarification of terms

- Target of the feature selection:

  "Its objective is to select a minimal subset of features according to some reasonable criteria so that the original task can be achieved equally well, if not better." [Liu, Motoda 1998]

- Why better?
  - If the number of features is lower, a better generalisation performance may be possible!
  - Cf. overfitting of a classifier (later)

Relevance of features

- Definition (highly relevant)

> A feature F_i is highly relevant if the following applies:
> $$P(C|F_i, S_i) \neq P(C|S_i)$$

- Feature F_i is necessary for an optimal subset. It cannot be removed without changing the original distribution
- Whereby S_i set of al features without F_i

Relevance of features

- Definition (weakly relevant)

A feature F_i is highly relevant if the following applies:

$$P(C|F_i, S_i) = P(C|S_i)$$

and: $\exists S_i' \subset S_i : P(C|F_i, S_i') = P(C|S_i')$

- Feature F_i is not always necessary for an optimal subset.
- But it may become necessary under certain conditions.

Relevance of features

- Definition (irrelevant)

A feature F_i is irrelevant if the following applies:

$$\forall S_i' \subseteq S_i : P(C|F_i, S_i') = P(C|S_i')$$

- Feature F_i is never necessary for an optimal subset.

Relevance of features

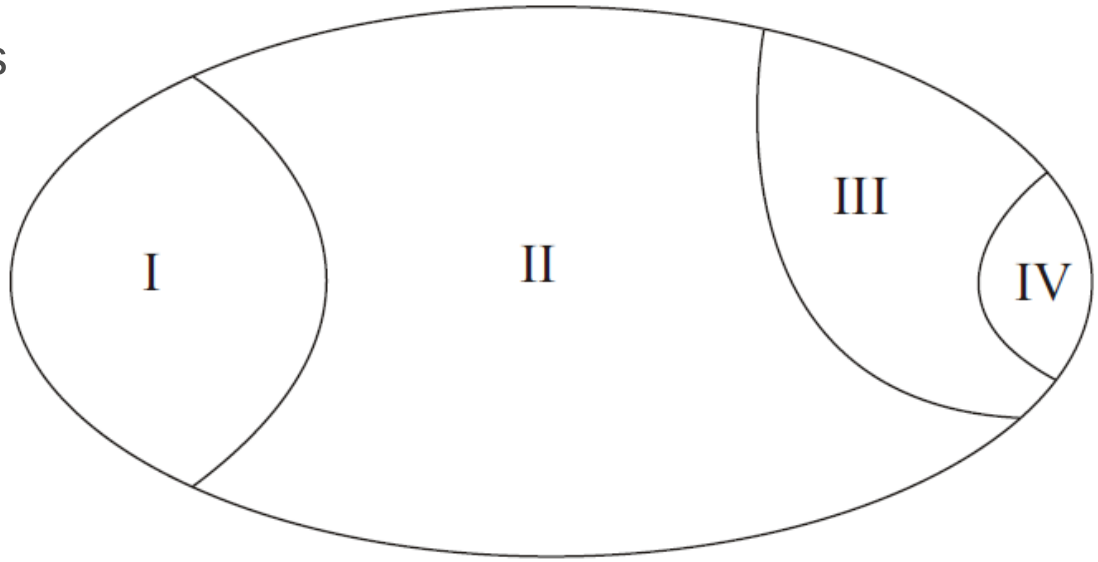
- Definition (redundant)

> Two features F_i and F_k are redundant if they are fully correlated.

- Examples
  - $F_i = F_k$
  - $F_i = \overline{F_k}$
  - $F_i = F_k + 1$
  - …

Relevance and redundancy

- I: irrelevant features
- II: weakly relevant features
- III: weakly relevant but non-redundant features
- IV: strongly relevant features
- III+IV: optimal subset



- An optimal subset contains all strongly relevant features, no irrelevant features and a subset of weakly relevant features!

Challenge

- In practice ...
  - there can be non-linear relationships between features,
  - groups of features can be interchangeable, not just individual features,
  - accuracy is difficult to measure when it comes to generalisation performance,
  - ...
- So how can "important" features be identified?

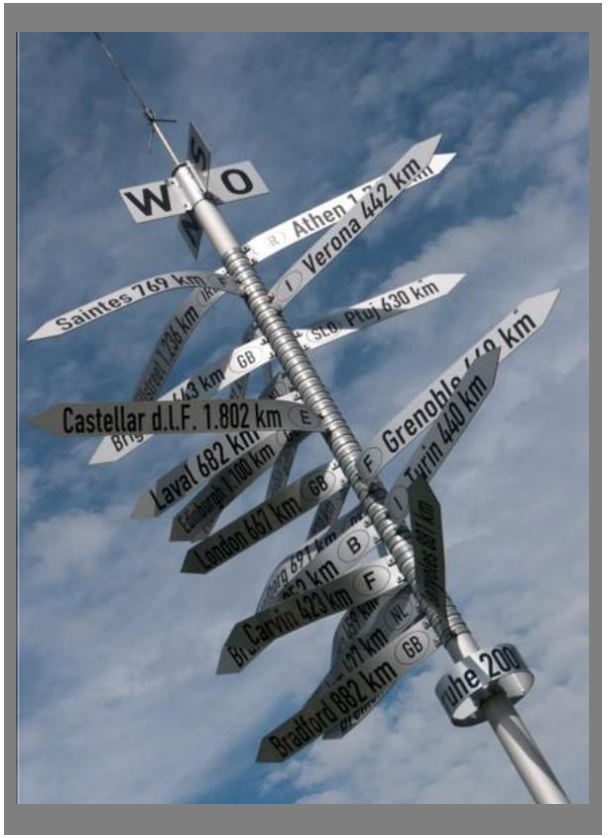
Approaches to feature selection

- Manual feature selection: Selection of relevant features by hand, i.e. expert knowledge is required.
- Automatic feature selection: Selection of relevant features automatically using suitable procedures.

- Here: automatic feature selection!

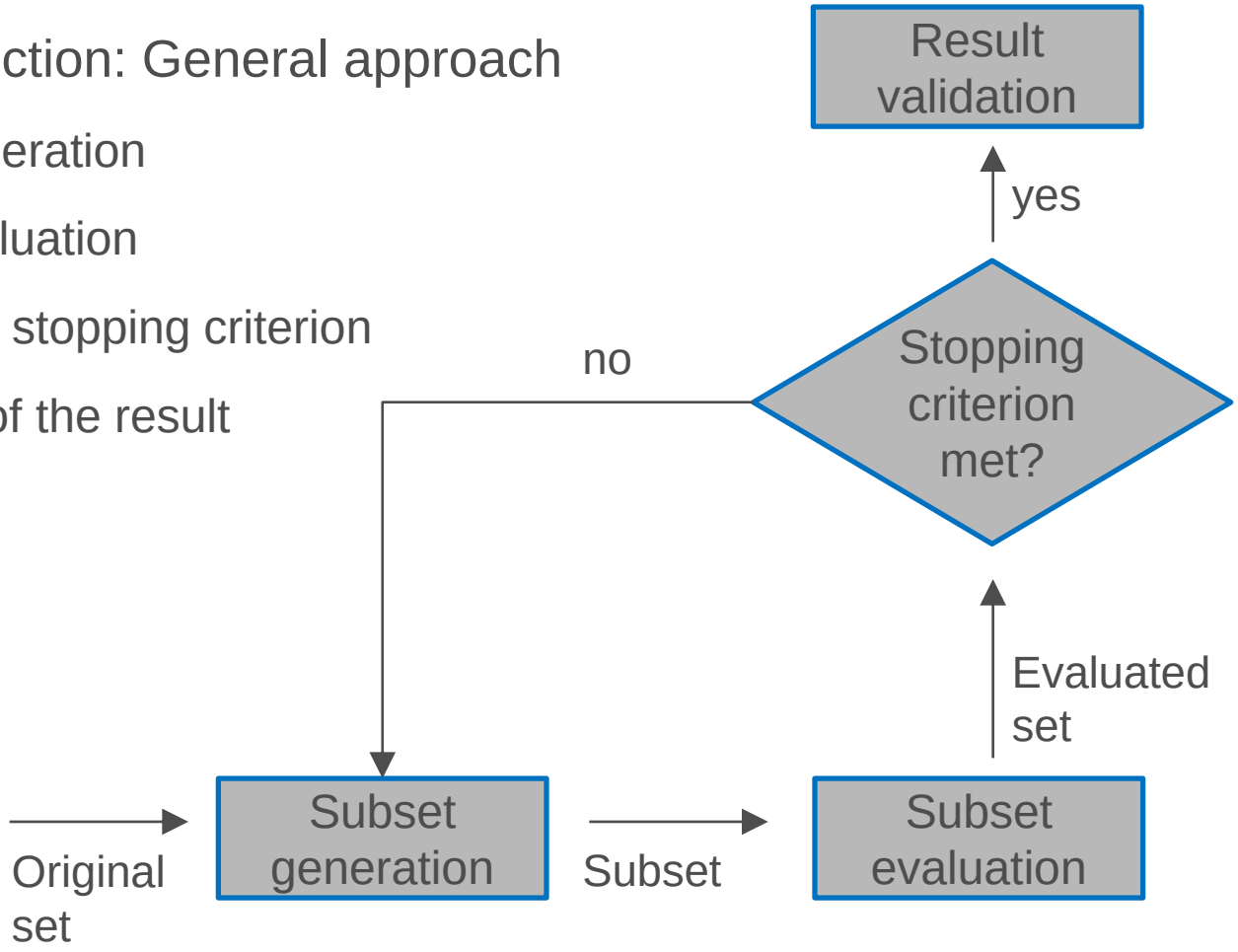
Agenda

- Feature extraction
- Time-independent features
- Time-dependent features
- Feature selection
- Taxonomy of feature selection
- Feature selection algorithms
- Feature transformation
- Principle Component Analysis
- Conclusion and further readings



Feature selection: General approach

1. Subset generation
2. Subset evaluation
3. Checking a stopping criterion
4. Validation of the result



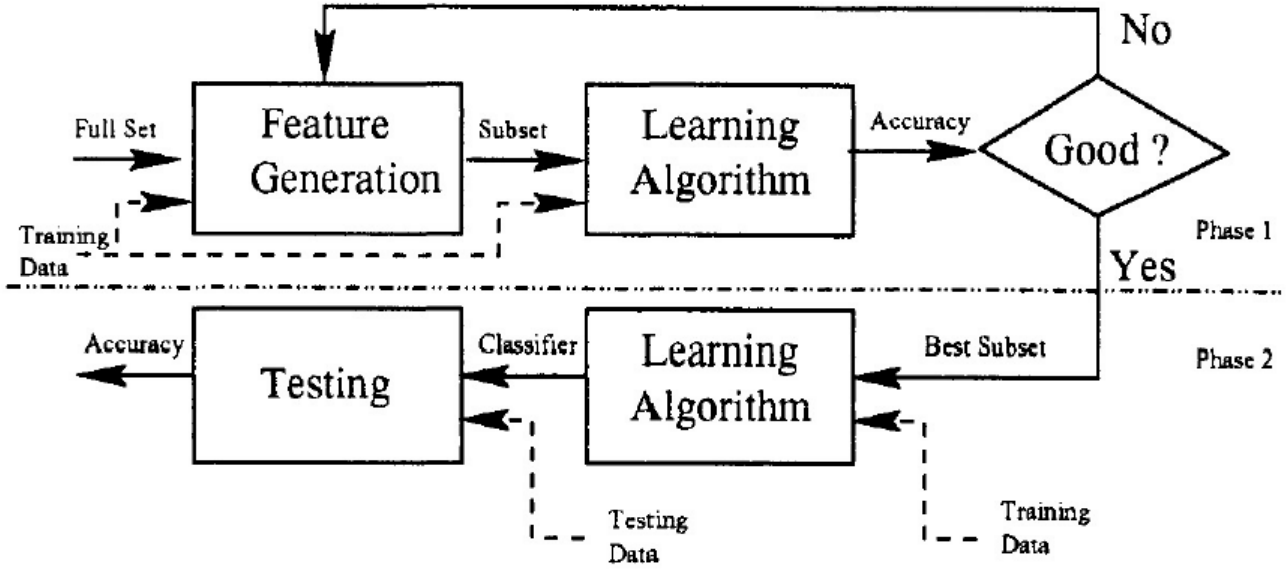
Taxonomy

- Feature selection procedures can be classified using the following criteria:
  - Filters, wrappers and hybrid solutions
  - Monitored or unmonitored approaches (supervised/unsupervised)
  - Generation of subsets (search directions and search strategies in the feature space)
  - Evaluation measures for selection algorithms
  - Stopping criteria

Wrappers and Filters

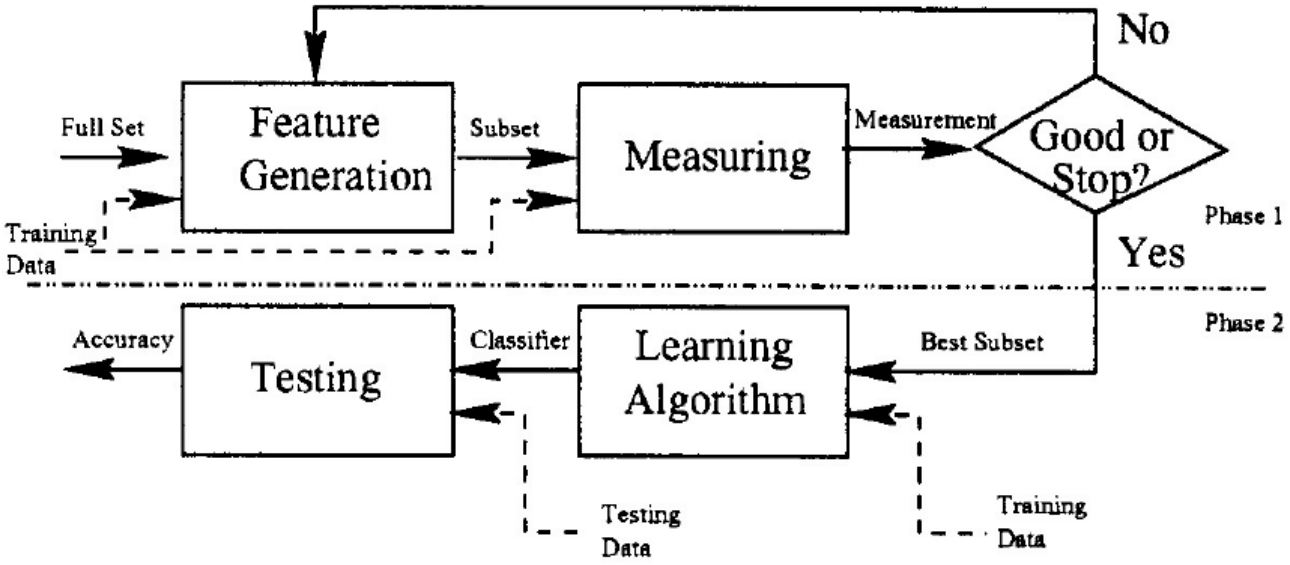
- Filters evaluate the quality of features based on an analysis of the data in the feature space.
- Wrappers evaluate the quality of a model instance found with selected features and a suitable ML algorithm (e.g. classifier).
- In principle, wrappers can deliver better results because the actual task is taken into account.
- However, the time expenditure is considerably higher, possibly too high in some tasks.
- Hybrid approaches are possible.

Wrapper



[Liu, Motoda 1998]

Filter



[Liu, Motoda 1998]

- We are primarily interested in filters…

Taxonomy: Feature selection (6)

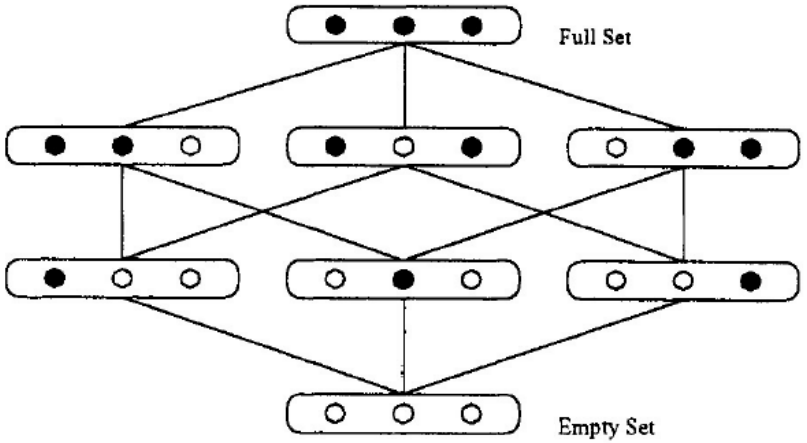
Supervised vs. unsupervised approaches

- Supervised procedures for feature selection (here: classification!) work with patterns with predefined class assignments (that is, data with labels).
- Unsupervised procedures for feature selection work with patterns without class assignment.
  (Techniques: cluster analysis, entropy determination, etc.)

→ We are primarily interested in supervised methods!

Generation of subsets

- The solution space for feature selection consists of all subsets of the set of available features:



[Liu, Motoda 1998]

- The number N of features is sometimes $\gg 1000$,…

Generation of subsets

- Questions:
  - Where is the search started?
  - In which direction is the search conducted?
- According to which criterion is the next possible solution selected?
- To generate solutions ("generating" subsets of the feature set),
  - the search direction and
  - the search strategy
- must be defined.

Generation of subsets

- Search directions:
  - Sequential Forward Generation (SFG)
  - Sequential Backward Generation (SBG)
  - Bidirectional Generation (BG)
  - Random Generation (RG)
- Search strategies:
  - Exhaustive / Complete Search
  - Heuristic Search
  - Nondeterministic search

Generation of subsets

- Search direction Sequential Forward Generation (SFG)

1. Starting from an empty set of features, exactly one feature (the “best“ in each case) is added in each run.
2. The search ends when either all features are selected or a certain threshold (i.e. stopping criterion) is reached regarding the quality of the selected features.
3. The result is a ranking of features; if required, a fixed number (the “best“ k features) can be selected.
4. In the first step, the forward generation variant considers all one-element-subsets, then all two-element-subsets, and so on.

Generation of subsets

- Search direction Sequential Backward Generation (SBG)

1. Starting from the total set of features, exactly one feature (the “most unimportant“ in each case) is removed in each run.
2. The search ends when there is only one feature left or when the evaluation criterion falls below a certain threshold.
3. The result can also be turned into a ranking of features.
4. In the first step, the backward generation variant considers all N-element-subsets, then all $(N-1)$-element-subsets, and so on.

Generation of subsets

- Search direction Bi-directional Generation (BG)

1. Start with both directions simultaneously, i.e. with SFG and SBG.
2. Goal of BG: Acceleration of the search, since the search in one of the two directions often ends early.
3. The search ends if either a given number of k features in one of the two directions has been reached or if both methods performed a run with $\frac{N}{2}$ elementary-subsets.

Generation of subsets

- Search direction Random Generation (BG)

1. Problem with SFG, SBG, BG: Accepting the best feature in the respective step or removing the feature least important in the respective step does not have to lead to the globally optimal solution!
2. Goal of RG: Avoidance of local optima
3. Here you start with a random selection of features.
4. In each run a feature is either removed or added (more or less) randomly.
5. Termination criterion here is the time or a given number of runs.

Generation of subsets

- Search strategy: Exhaustive Search

- All possible feature combinations are examined
  - E.g. forward generation: $\binom{N}{0} + \binom{N}{1} + \binom{N}{2} + \cdots + \binom{N}{m}$
  - Cf. Width search
- Runtime complexity: $O(2^N)$
- Global optimum is always found
- Hardly feasible in practice, only for small N
- Possibility to find the global optimum without explicitly evaluating all solutions (i.e. Complete Search). A special property of the evaluation criterion for selection algorithms (see below) is assumed: monotony.

Generation of subsets

- Search Strategy: Heuristic Search

- Heuristics (e.g. based on expert knowledge) are used.
- Usually searches a specific path from the full set of characteristics to the empty subset or vice versa; then the runtime is O(N).
  - Cf. depth search
- Finding the global optimum cannot be guaranteed; however, very good optima are often found.

Generation of subsets

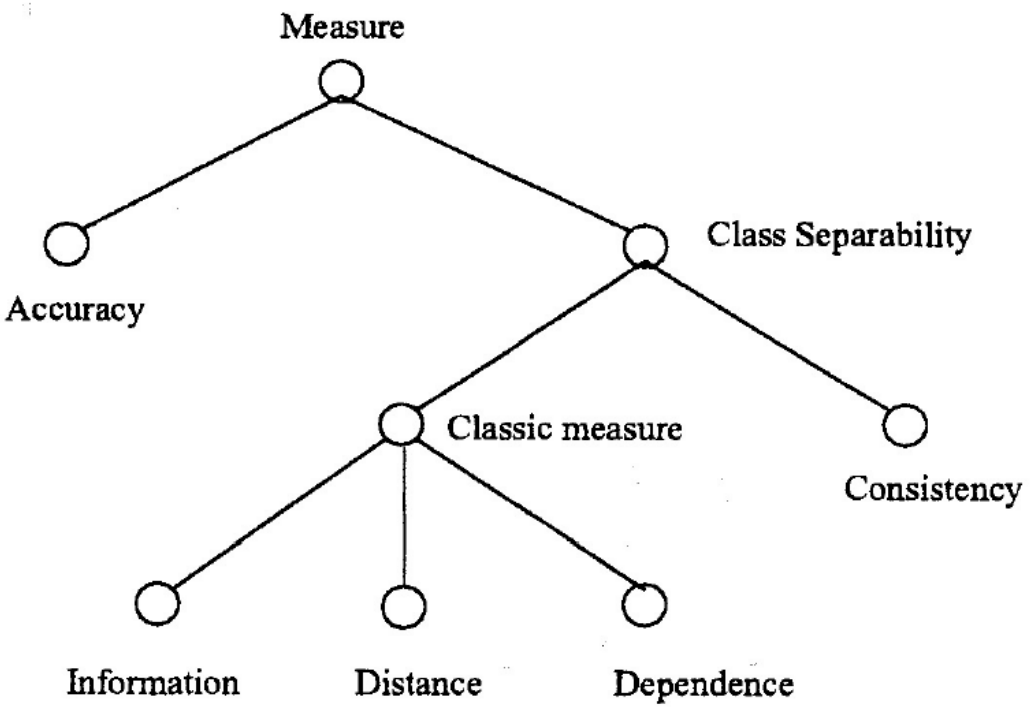
- Search Strategy: Non-deterministic Search

- Selection of the next set of features not according to a fixed rule, but randomly.
  - I. e., the next subset is not created by deterministic addition/removal of features.
- Global optimum cannot be guaranteed either.
- The stopping criterion is typically a fixed number of runs.

Evaluation measures

- Evaluation measures (selection criteria) are required to evaluate the quality of a generated subset of characteristics.
- Classes of measures:
  - Accuracy Measures
  - Classic measures
    - Information Measures
    - Distance measures
    - Dependence Measures
- Consistency Measures

Evaluation measures



[Liu, Motoda 1998]

Accuracy measures ...

- ... evaluate the quality of the model instance created with selected features (e.g. by the classification rate).
- ... are therefore generally used for wrapper approaches, i.e. the following problems occur:
  - Selection of an appropriate ML paradigm (e.g. neural net, SVM, decision tree, ...)
  - Time required for model creation in each run
  - Avoid over-fitting in each run
  - Evaluation of the generalisation performance in each run

Information measures ...

- ... evaluate (originally) the uncertainty of a recipient when receiving a message:
  - If the recipient knows which message he will receive, his surprise (uncertainty) is small.
  - If the recipient does not know which message is coming (assumption: all messages are equally likely to come), his uncertainty is high.
  - For classification tasks: Messages are classes.
- .. are often based e.g. on Shannon's entropy measure.

Evaluation measures

- Definition: Shannon‘s entropy E

$$E(D) = -\sum_{i=1}^{d} p_i \log_2 p_i$$

- D: Data volume
- d: Number of classes
- p_i: Probability for class i

- Measures the information content (impurity, disorder) of a statement or data set.
- The greater the entropy, the greater the information content!

Shannon‘s Entropy

- Example: Coin toss
  - $p_{head} = 0.5, p_{tail} = 0.5, d = 2$

SOLUTION

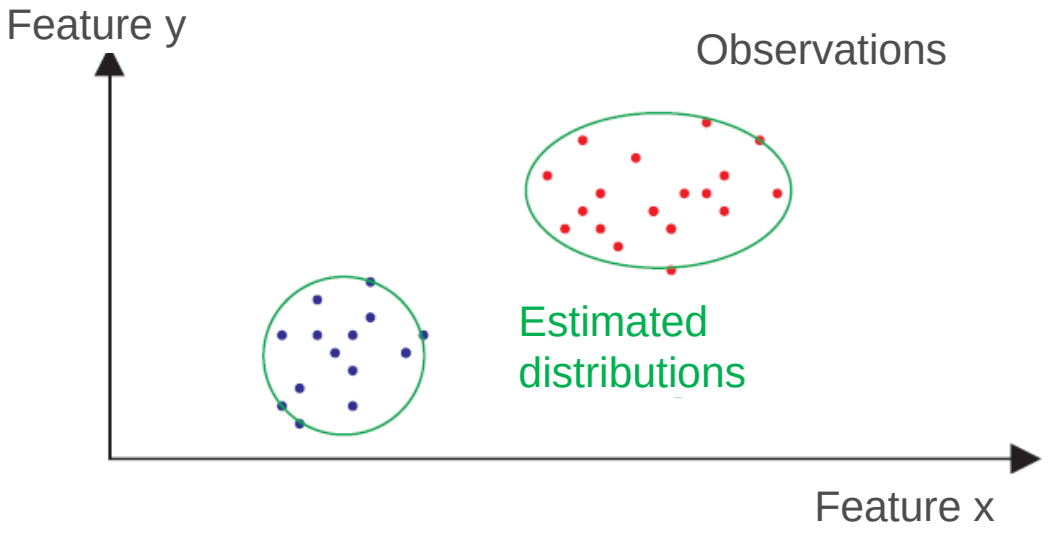
- Example: Faked coin toss
  - $p_{head} = 0.75, p_{tail} = 0.25, d = 2$

SOLUTION

- The more ‘predictable‘ the result, the smaller the information content!

Distance measures

- Evaluate the separability of classes, e.g. based on estimated distribution functions.
- Are also known as separability measures, divergences measures, or discrimination measures.



- Feature x is probably better suited for separation than feature y.

Dependency measures

- Evaluate (originally) the correlation of features with the correlation coefficient ϱ
- Means: They answer the question of how predictable the value of a feature is from the value of another feature.

$$\varrho = \frac{Cov(X,Y)}{\sigma(X)\sigma(Y)}$$

- Investigates the correlation of features with classes.
- Also known as association measures or correlation measures.

Consistency measures

- Avoid a problem of classical measures: They cannot recognise redundant features, i.e. they cannot distinguish between equally good features.
- Examine contradictory patterns (i.e., the same input features), but different class assignments.
- Are also able to identify irrelevant features.

Evaluation measures

- Specific examples for different evaluation measures will be presented in the following using a simple application example.
- The question is whether the fact that a person gets a sunburn depends:
  - on his hair colour,
  - on his size,
  - on his weight,
  - on the use of sun lotion.

Example

[Liu, Motoda 1998]

- Sunburn, depending on hair colour, size, weight, use of sun lotion:

Hair colour	Size	Weight	Use of sun lotion	Result
Blonde	Average	Light	No	Sunburned
Blonde	Large	Average	Yes	None
Brown	Small	Average	Yes	None
Blonde	Small	Average	No	Sunburned
Red	Average	Heavy	No	Sunburned
Brown	Large	Heavy	No	None
Brown	Average	Heavy	No	None
Blonde	Small	Light	Yes	None

Example

- Coding based on probabilities

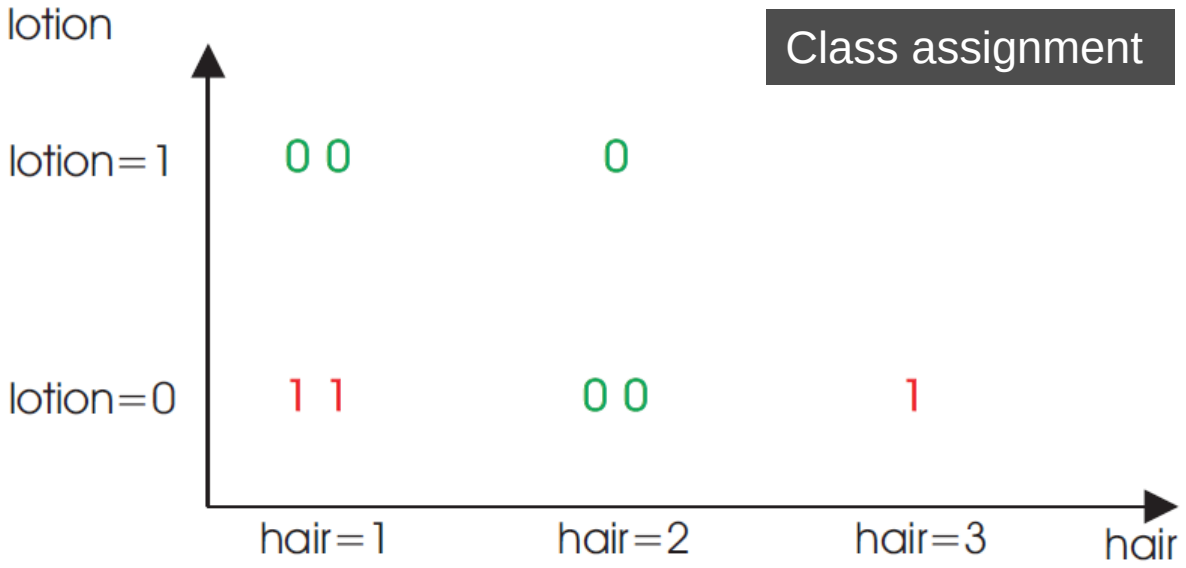
[Liu, Motoda 1998]

	Hair	Height	Weight	Lotion	Result
i_1	1	2	1	0	1
i_2	1	3	2	1	0
i_3	2	1	2	1	0
i_4	1	1	2	0	1
i_5	3	2	3	0	1
i_6	2	3	3	0	0
i_7	2	2	3	0	0
i_8	1	1	1	1	0

	Result (Sunburn)	
	No	Yes
P(Result)	5/8	3/8
P(Hair=1|Result)	2/5	2/3
P(Hair=2|Result)	3/5	0
P(Hair=3|Result)	0	1/3
P(Height=1|Result)	2/5	1/3
P(Height=2|Result)	1/5	2/3
P(Height=3|Result)	2/5	0
P(Weight=1|Result)	1/5	1/3
P(Weight=2|Result)	2/5	1/3
P(Weight=3|Result)	2/5	1/3
P(Lotion=0|Result)	2/5	3/3
P(Lotion=1|Result)	3/5	0

Example

- The data set is small, the relationships simple, therefore a simple analysis is possible here:



- The features
  'lotion' and 'hair' are sufficient for an unambiguous class allocation
- However: Both features are required!

Information measure "Information Gain" (IG)

- Partitioning a data set X by a feature d in L subsets X_d with $l = 1, \dots, L$.
- There are C classes corresponding to the manifestations (possible instances) of the feature.
- Information Gain (IG) of a feature d:

$$IG(d) := I(X) - \sum_{l=1}^{L} \frac{|x_{d_l}|}{|X|} I(x_{d_l})$$

$$I(x_{d_l}) := -\sum_{c=1}^{C} px_{d_l}(c) \cdot \log_2 px_{d_l}(c)$$

$$I(X) := -\sum_{c=1}^{C} px\,(c) \cdot \log_2 px\,(c)$$

Information measure "Information Gain" (IG)

- Example:
  - $I(X)$

$$I(X) = -\sum_{c=1}^{C} px(c) \cdot \log_2 px(c)$$
$$= -\frac{5}{8}\log_2\frac{5}{8} - \frac{3}{8}\log_2\frac{3}{8}$$
$$= 0.954$$

  - $I(X_{hair_1})$

$$I(X_{hair_1}) = -\sum_{c=1}^{2} px_{hair_1}(c) \cdot \log_2 px_{hair_1}(c)$$
$$= -\frac{2}{4}\log_2\frac{2}{4} - \frac{2}{4}\log_2\frac{2}{4}$$
$$= 1$$

Information measure "Information Gain" (IG)

- Example:
  - $I(X_{hair_2})$

$$I(X_{hair_2}) = -\sum_{c=1}^{2} px_{hair_2}(c) \cdot \log_2 px_{hair_2}(c)$$
$$= -1 \log_2 1 - 0 \log_2 0$$
$$= 0$$

  - $I(X_{hair_3})$

$$I(X_{hair_3}) = -\sum_{c=1}^{2} px_{hair_3}(c) \cdot \log_2 px_{hair_3}(c)$$
$$= -0 \log_2 0 - 1 \log_2 1$$
$$= 0$$

→ No unpredictable assignment!

Information measure "Information Gain" (IG)

- Example:
  - $IG(hair)$

$$IG(hair) = I(X) - \sum_{c=1}^{3} \frac{|x_{hair_c}|}{|X|} I(hair_c)$$

$$= 0.954 - \frac{4}{8} \cdot 1$$

$$= 0.454$$

- Since $IG(hair) = 0.454$, $IG(lotion) = 0.348$, $IG(height) = 0.266$ and $IG(weight) = 0.016$, the feature $hair$ is the first in the rank list!

Consistency measure "inconsistency rate" (IR)

- Two patterns are inconsistent if they are identical except for the class assignment.
- The inconsistency number IZ of a pattern is the number of occurrences of this pattern minus the largest number with a particular class assignment.
- The inconsistency rate IR of a data set is the sum of all inconsistency numbers divided by the total number of samples.

Please note:

- IR is calculated during feature selection for various subsets that are created from the original data set by omitting features.

Consistency measure "inconsistency rate" (IR)

- Example:

Sample	Class A	Class B	Class C	
00	1	1	2	$\rightarrow IZ(00) = 4 - 2 = 2$
01	2	3	0	$\rightarrow IZ(01) = 5 - 3 = 2$
10	0	0	1	$\rightarrow IZ(10) = 1 - 1 = 0$
11	2	8	0	$\rightarrow IZ(11) = 10 - 8 = 2$

$$\Rightarrow IR = \frac{2+2+0+2}{20} = 0.3$$

Remarks:

- IR is calculable in $O(N)$ (N: number of samples)
- IR is a monotonous measure

Consistency measure "inconsistency rate" (IR)

- Selection criterion for subset based on IR:
  - Be D_i and D_j two (sub)sets of features with inconsistency rates $IR(D_i)$ and $IR(D_j)$.
  - Then applies:
    - D_i and D_j are indistinguishable if $IR(D_i) = IR(D_j)$ and $|D_i| = |D_j|$
  - D_i will be preferred over D_j if:
    - $IR(D_i) = IR(D_j)$ and $|D_i| < |D_j|$
    - $IR(D_i) < IR(D_j)$ and $|D_i| \leq |D_j|$

Consistency measure "inconsistency rate" (IR)

- Example sunburn: The inconsistency rate is 0 for
  - the two features of hair and lotion
  - the three characteristics of hair, height and weight
  - any supersets of these two quantities
- Since the smallest quantity of features is sought, the two features hair and lotion are chosen.

There are many other dimensions, for example:

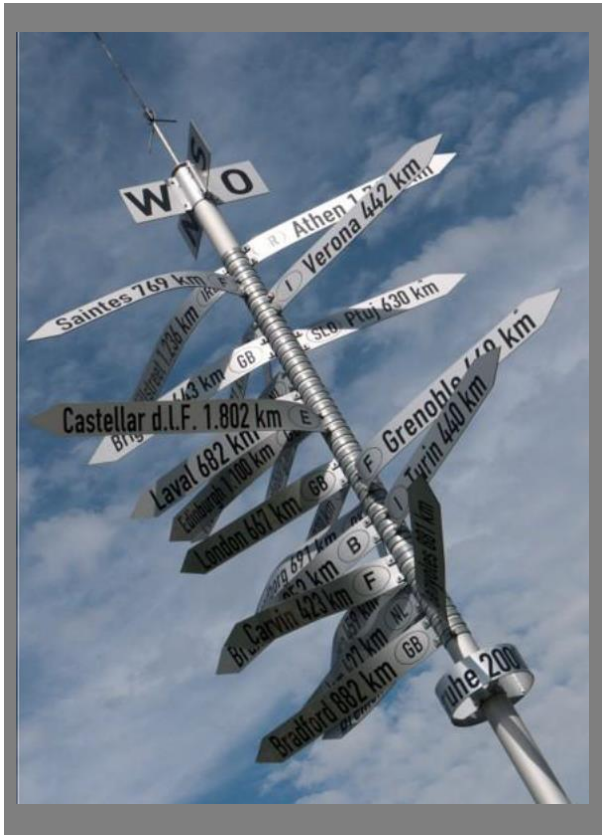
- Minimum Description Length
- Distance measure Directed Divergence
- Distance dimension Variance
- Bhattacharyya dependence measure
- etc.

Different stopping criteria are possible:

- Time: Termination when a time limit is reached
- Number of features: Termination if a predefined number of features is selected (forwards and backwards possible)
- Iterations: Termination if a given number of iterations have been passed through in the algorithm
- Algorithm-specific stopping criteria
- Application-dependent stopping criteria

Agenda

- Feature extraction
- Time-independent features
- Time-dependent features
- Feature selection
- Taxonomy of feature selection
- **Feature selection algorithms**
- Feature transformation
- Principle Component Analysis
- Conclusion and further readings



Selected Algorithms

- Focus
- Automated Branch and Bound (ABB)
- Relief

FOCUS

- Idea:
  - Start with an empty feature set.
  - Consider all $\binom{D}{1}$, $\binom{D}{2}$, $\binom{D}{3}$,… combinations of features of size $1, 2, 3, \ldots$ (D: total number of features)
  - Stop if the inconsistency rate equals
    → Alternatively: predefined inconsistency rate
  - Subset must be examined $\sum_{d=1}^{D'} \binom{D}{d}$ to find a subset of the size $D' \leq D$.

FOCUS

- Algorithm:

Focus
Input: F - all features x in data D
U - inconsistency rate as evaluation measure

initialize: $S = \{\}$

for $i = 1$ to N
for each subset S of size i
if $\mathrm{Cal}U(S, D) = 0$ /* $\mathrm{Cal}U(S, D)$ returns *inconsistency**/
return S
Output: S - a minimum subset that satisfies U

Focus – classification according to taxonomy:

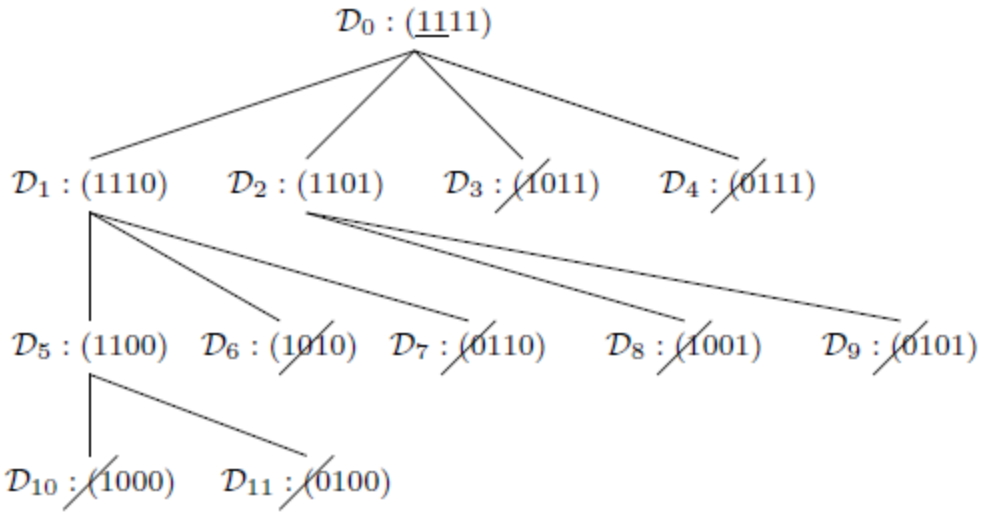
- Type: Filter
- Supervised: yes
- Search direction: SFG (sequential forward generation)
- Search strategy: comprehensive search (exhaustive search)
- Measure: Consistency measure (inconsistency rate)
- Termination criterion: all feature subset examined or subset found with IR = 0

Automated Branch and Bound (ABB)

- Idea:
  - Start with the complete feature set.
  - If IR of the subset is larger, stop BS (breadth search) at this point and mark the subset as invalid.
  - Accordingly, all successors of this node (subsets) are also invalid.
  - Otherwise, continue with BS.
  - When BS is finished, return the smallest valid feature set.

Automated Branch and Bound (ABB) :

- Example:
  - Set of 4 features
  - Assumption: the first two features are important, the other two are not (omitting increases inconsistency rate)



Automated Branch and Bound (ABB) :

- Comments on the method
  - Coding of the selected features:
    - Binary string (1: Feature) selected; 0: Feature not selected).
  - Generate subsets:
    - Set first "0" from left as a marker.
    - Generate subsets by changing a "1" to a "0" from the marker to the left.
  - Find "non-legitimate" subsets:
    - Generated subsets are "non-legitimate" (no need to be evaluated) if their Hamming distance to a subset already marked as "invalid" is "1".
  - Find "invalid" subsets:
    - Generated subsets are "invalid" if their inconsistency rate is higher (Pruning: trimming the BS tree at this point).

Automated Branch and Bound (ABB) :

- Characteristics of the approach
  - Complexity depends on the number of subsets considered, which is relatively small on average due to Pruning.
  - The complexity of the legitimacy check depends on the number of features, the complexity of the inconsistency check on the number of samples.
  - Here, too, a threshold value can be used for the inconsistency rate.

Classification of ABB:

- Type: Filter
- Supervised: yes
- Search direction: SBG (sequential backward generation)
- Search strategy: complete search
- Measure: Consistency measure (inconsistency rate)
- Stopping criterion: algorithm-specific

RELIEF - idea:

- Creates ranking of features by assigning weights to the features (Feature Ranking)
- Finds features that are statistically relevant by calculating distances on samples (!).
- Basic idea for two-class problems:
  - For samples, the next neighbour (sample) of the same class (nearest-hit) and the other class (nearest-miss) is searched with a next neighbour algorithm.
  - Features are regarded as relevant if the respective values are similar in the considered sample and nearest-hit or different in the considered sample and nearest-miss.
- Weights are calculated iteratively.

RELIEF

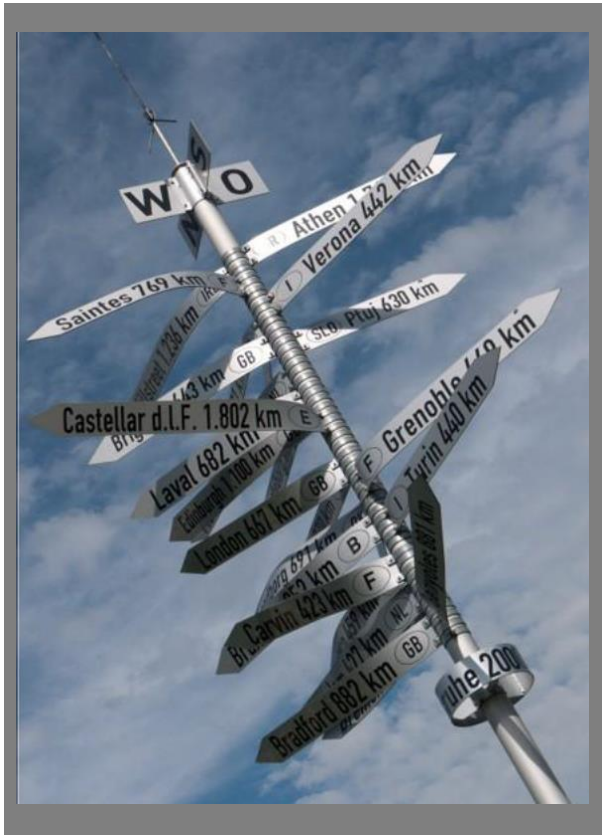
- Be $x = (x_1, x_2, x_3, \ldots, x_D)$ as well as $y = (y_1, y_2, y_3, \ldots, y_D)$ samples of the data set (with D as number of features). Then, the following information is required:
  - Nearest-hit(x): Pattern $x^{(H)}$with the smallest (typically for continuous, standardised features: Euclidean) distance to x among all samples with the same class assignment.
  - Nearest-miss(x): Pattern $x^{(M)}$with the smallest (typically for continuous, standardised features: Euclidean) distance to x among all samples with a different class assignment.
  - $\|x_d - y_d\|$: Distance between the two values x_d and y_d (e.g., 0/1 for discrete features or difference for continuous, standardised features)
- Determine w sample-based as the vector of feature weights.

Classification of RELIEF:

- Type: Filter
- Supervised: yes
- Search direction: difficult to classify (looks at samples)
- Search strategy: difficult to classify (looks at samples)
- Dimension: distance dimension (e.g. Euclidean distance on samples)
- Stopping criterion: a specified number of iterations

Agenda

- Feature extraction
- Time-independent features
- Time-dependent features
- Feature selection
- Taxonomy of feature selection
- Feature selection algorithms
- **Feature transformation**
- Principle Component Analysis
- Conclusion and further readings

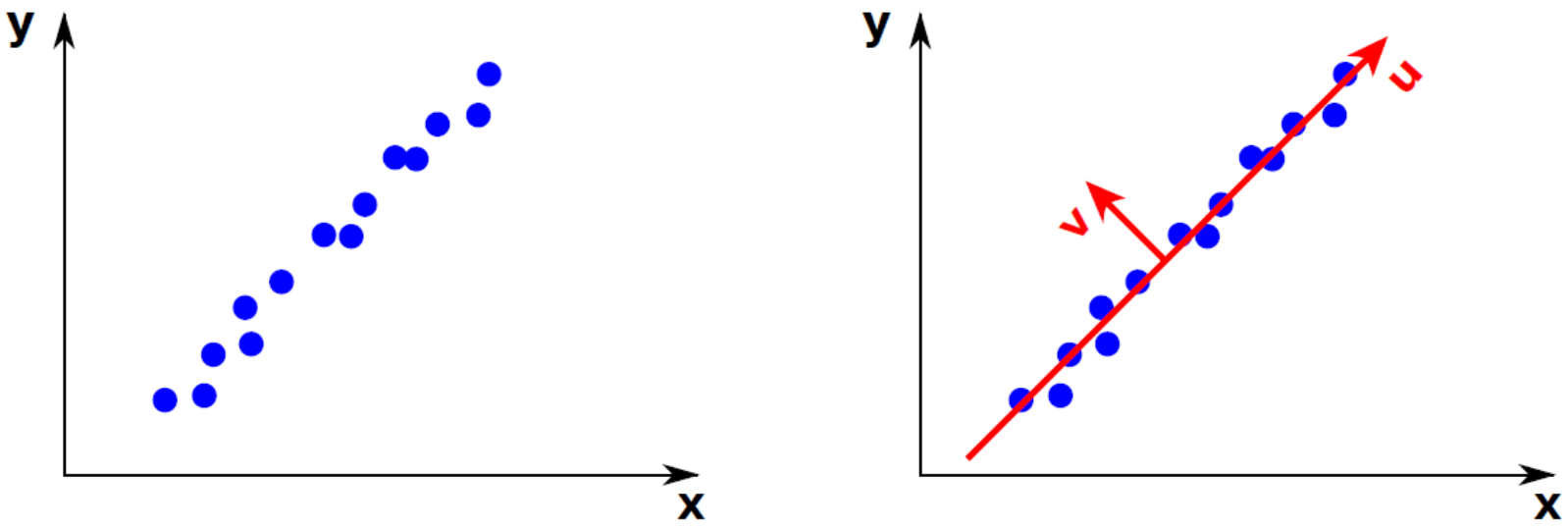


Feature transformation: reduction of dimensions

- Previously: Reduction of existing information by selecting a subset of "important/relevant" features (attributes, dimensions).
  - Start with an empty, complete or random subset of all features
  - Successive addition or omission of certain features
  - Until certain stopping criterion (quality, number of iterations, ...) fulfilled
- Now: Transformation of data
  - Projection into a space of smaller dimensionality
  - Objective: Summarisation of relevant information in a smaller number of features

    → Maintain data structure as well as possible
  - Transformed features: Linear combination of the original features (main components)

Feature transformation

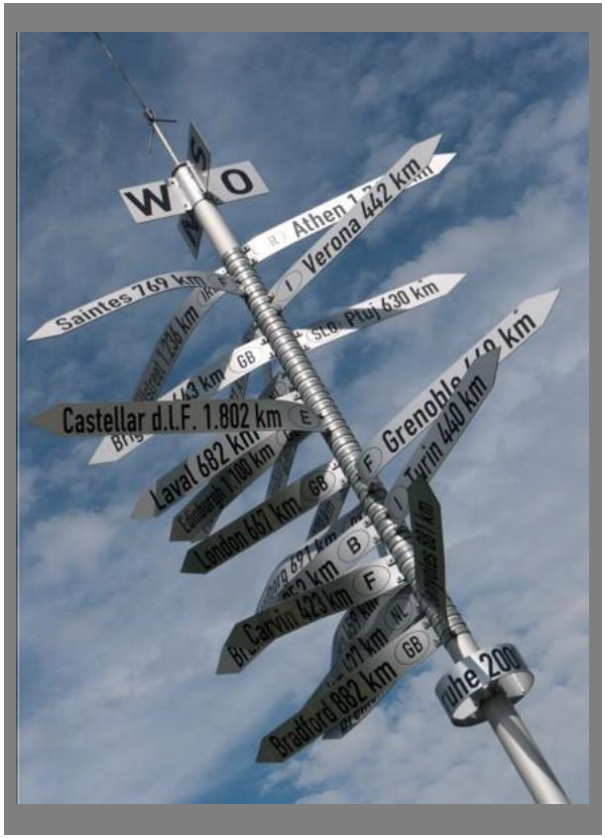


Motivation

- Given:
  - A data set X with samples $x_1, x_2, x_3, \dots, x_N$
  - The samples are D-dimensional, i.e. there are D features
- Wanted:
  - A data set Y with samples $y_1, y_2, y_3, \dots, y_M$
  - The samples are also D-dimensional, i.e. there are D features
  - The following applies: $|X| = |Y|$ or $N = M$
  - The information content is the same as in data set X
  - But: The information content of the data set Y is stored in the first, few features (dimensions)
  - This is also referred to as meta-features.
- Principal Component Analysis:
  - A method to find such a data set

Agenda

- Feature extraction
- Time-independent features
- Time-dependent features
- Feature selection
- Taxonomy of feature selection
- Feature selection algorithms
- Feature transformation
- Principle Component Analysis
- Conclusion and further readings



What does information content mean?

- Assumption: high information content corresponds to high variance!

- Main objectives of the Principal Component Analysis (PCA)
  - Primary goal: Ordered dimensions
  - Secondary goal: Dimension reduction
    - Done based on the ordering
    - Less important dimensions can be omitted
    - I.e. the number D' if the features in the transformed data set is $D' \ll D$.

Benefits of Principal Component Analysis:

- Time saving: by using ML algorithms on reduced information (i.e., data sets or features).
- Feature selection: very simple by selecting the most important meta-features (i.e., the first x).
- Understanding: better recognition of structures in data, e.g. by visualising the data set in the space of the two or three most important meta-features.

Other names for Principal Component Analysis:

- Hotelling Transformation
- Karhunen-Loeve Transformation
- ...

Basics

- To transform the data set, the arithmetic mean of each sample i is calculated as a first step:

$$\mu_i = \frac{1}{N}\sum_{n=1}^{N} x_{i,n}$$

- Instead of the original samples, the average-cleaned samples are then used again:

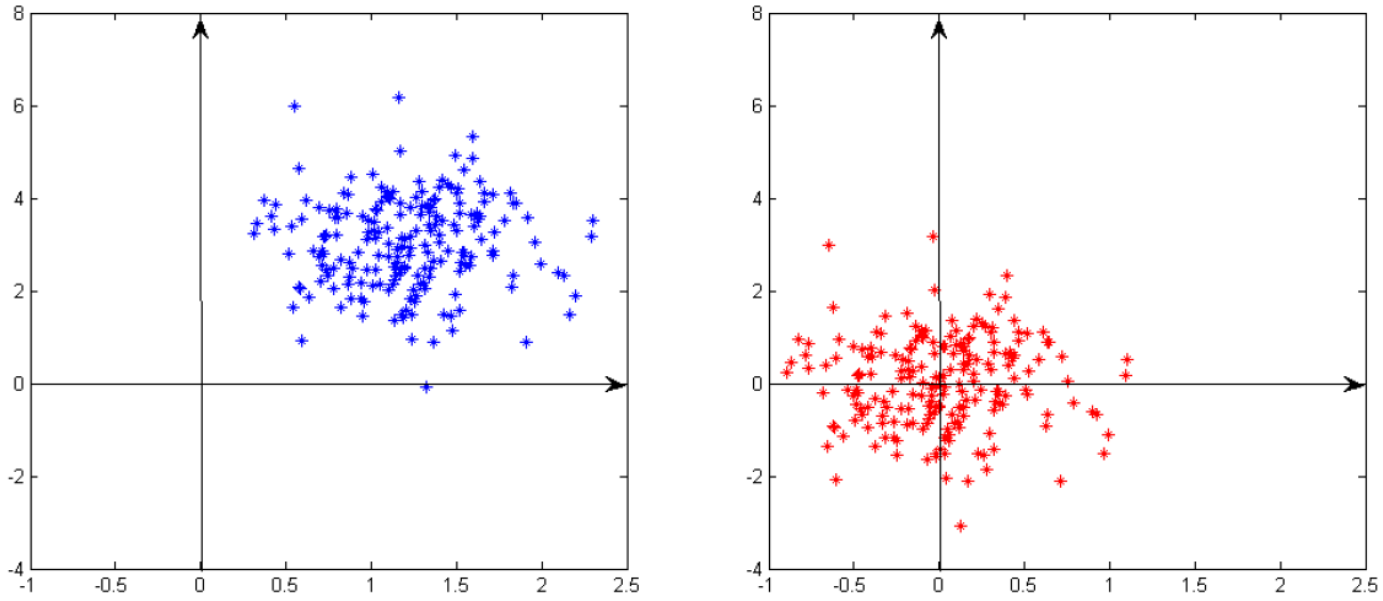
$$\forall_{n=1,\dots,N}\forall_{i=1,\dots,D}: x'_{i,n} = x_{i,n} - \mu_i$$

- Alternatively:

$$\mu := \begin{pmatrix} \mu_1 \\ \mu_2 \\ \dots \\ \mu_D \end{pmatrix}$$

- $\mu - x_n$ for $n = 1,2,\dots,N$.
- This corresponds geometrically to a translation (displacement) of the data.

Example



- The blue dots represent the original data set, with the arithmetic mean $\mu_1 = 1.2$ and $\mu_2 = 13$.
- The red dots represent the original data set after translation.

Basics (continued)

- The empirical variance of a feature i is then:

$$\sigma_i^2 := \frac{1}{N-1}\sum_{n=1}^{N} x_{i,n}'^2$$

- Thus, the empirical standard deviation of the feature is:

$$\sigma_i = \sqrt{\sigma_i^2}$$

Basics (continued)

- The covariance of two features i and j is also required:

$$s_{i,j} := \frac{1}{N-1} \sum_{n=1}^{N} x'_{i,n} \cdot x'_{j,n}$$

- A covariance $s_{i,i}$ (i.e., a feature with itself) is, of course, the variance again.
- In addition applies: $s_{i,j} = s_{j,i}$

Covariance: Interpretation

- Covariance is a measure of the linear relationship between two features.
- $s_{i,j} > 0$: Large values of feature i are associated with large values of j (analogously for small values).
- $s_{i,j} < 0$: Large values of feature i are associated with small values of j (and vice versa).
- $s_{i,j} = 0$: No linear relationship between the two features.

Covariance

- Covariance is always calculated in pairs, i.e. for two features
- For D-dimensional data there are $\frac{D!}{(D-2)!2!}$ many covariances.
- If we write the covariances in a matrix:

$$\begin{pmatrix} S_{1,1} & \cdots & S_{1,D} \\ \vdots & \ddots & \vdots \\ S_{D,1} & \cdots & S_{D,D} \end{pmatrix}$$

  - This matrix is symmetrical.
  - The diagonal shows the variances of the features.

Basics (continued)

- An eigenvector v of such a matrix C is a D-dimensional vector, for which applies:

$$C \cdot v = \lambda \cdot v$$

- $\lambda \in \mathbb{R}$ means eigenvalue to the eigenvector v.

It applies:

- C has D eigenvectors $v_1, v_2, v_3, \ldots, v_D$ with the corresponding eigenvalues $\lambda_1, \lambda_2, \lambda_3, \ldots, \lambda_D$.
- Without limitation of the generality, we assume for $v_1, v_2, v_3, \ldots, v_D$:

$$\lambda_1 \geq \lambda_2 \geq \lambda_3 \geq \cdots \geq \lambda_D$$

- The eigenvectors are perpendicular to each other, i.e. they are orthogonal to each other.
- Multiples of an eigenvector are also eigenvectors, we use those that are normalised to length 1

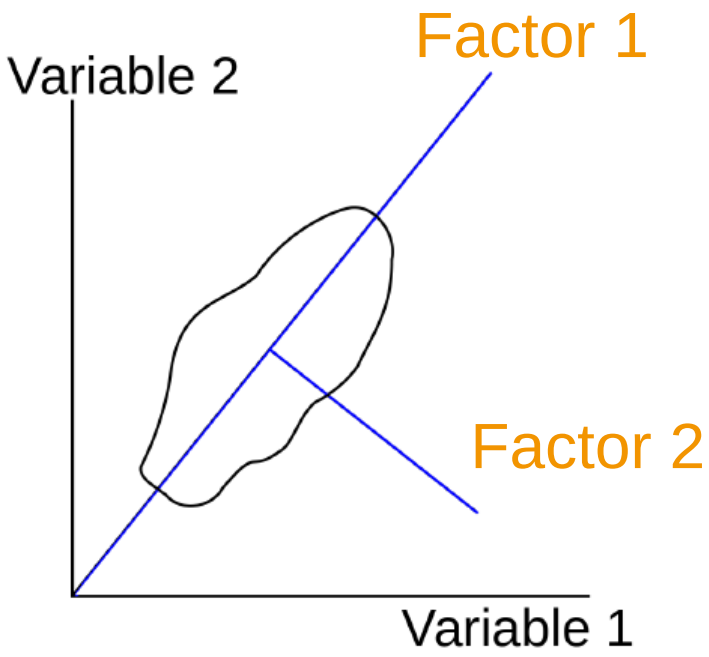
A most important property of the eigenvectors:

- The eigenvector with the highest eigenvalue specifies the direction in which the data set has the highest variance.
- The eigenvector with the second-highest eigenvalue specifies an orthogonal direction in which the data set has the second-highest variance.
- etc.

The variances are described by the respective eigenvalues!

- Variance → Information content!

Example:



Example 2: Three farmers

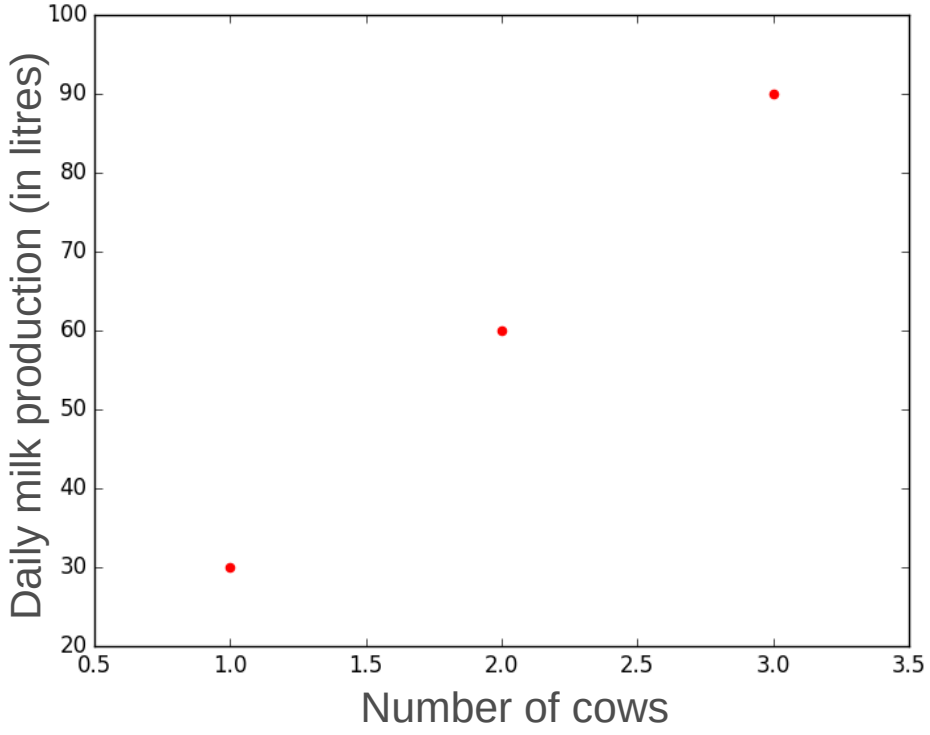
	Anton	Ben	Carl
Number of cows	1	2	3
Daily milk production (litre)	30	60	90
Deviation of #cows from average	-1	0	1
Deviation of #litres from average	-30	0	30
Product of deviations	30	0	30

- Question (to be answered by covariance calculation): Is there a causal relationship between the number of cows and milk production?

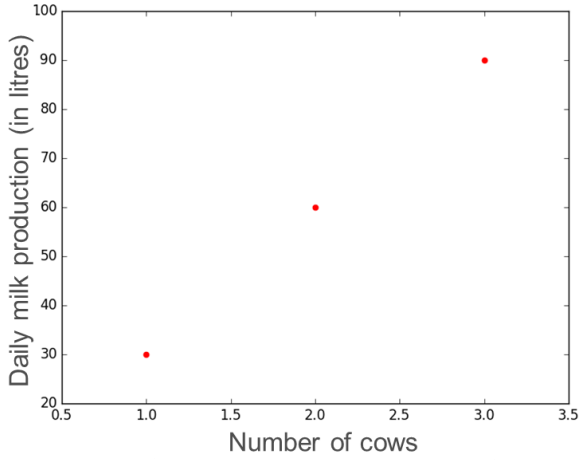
Example 2: Three farmers

- Intuition suggest: Yes!

- See scatter plot



Example:



- The three deviation products are added together: $30 + 0 + 30 = 60$
- Then divided by the number of features carriers (3 dairy farmers): $60/3 = 20$
- In the case of the three dairy farmers, we have to consider the total population.
- If (alternatively) a sample set is considered, the so-called empirical covariance is involved
  - Then: Divide by the number of feature carriers minus 1 (i.e., $3 - 1 = 2$)
  - Instead of dividing by the number of feature carriers

Example (continued):

- The product of the deviations for Anton results from the fact that the negative deviation of -1 is multiplied by the negative deviation of -30 litres.
- This results in +30
- As a complete formula:

$$Cov = \frac{[(1-2)\cdot(30-60)] + [(2-2)\cdot(60-60)] + [(3-2)\cdot(90-60)]}{n}$$

$$= \frac{30+0+30}{3} = 20$$

- Positive values of the covariance mean that high values of one feature (here: milk production) are accompanied by high values of the other feature (here: number of cows) and vice versa.

Back to PCA: Basics (continued)

- Next, a specific number $D' \leq D$ is selected from the eigenvectors to transform the data
- All eigenvectors (i.e., $D' = D$) are selected if the goal of the PCA is, e.g., a principal axis transformation to decorrelate the data
- In this case, the mean-adjusted samples are transformed as follows:

$$y_k = \begin{pmatrix} v_1^T \\ v_2^T \\ \vdots \\ v_D^T \end{pmatrix} x'_k$$

- This corresponds to a rotation of the data!

Basics (continued)

- A smaller number of eigenvectors (usually $D' \ll D$) is selected if the goal of the PCA is data reduction.
- In this case, the mean-adjusted samples are transformed as follows:

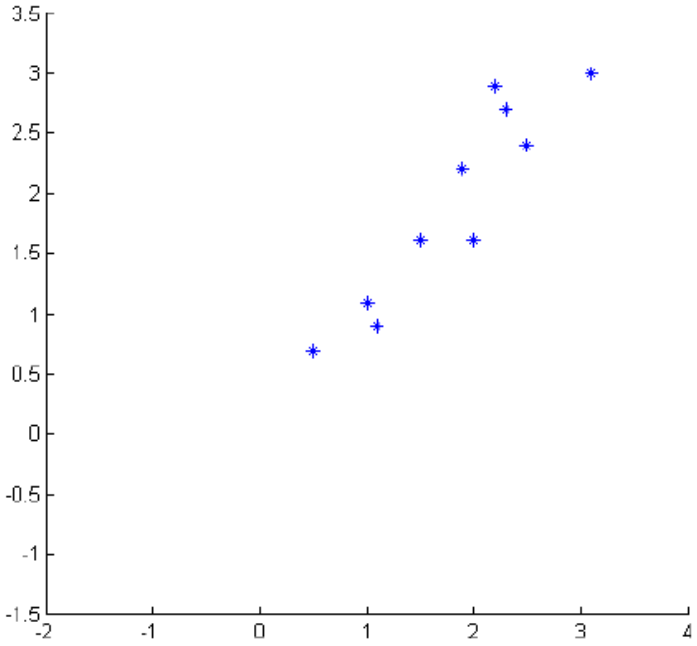
$$y_k = \begin{pmatrix} v_1^T \\ v_2^T \\ \vdots \\ v_{D'}^T \end{pmatrix} x'_k$$

A retransformation is possible - but at D' dimensions only with the loss of information

- So the transformed data y_k has only D' dimensions!

Example of a 2-dimensional data set:

Sample	Feature 1	Feature 2
x_1	2.5	2.4
x_2	0.5	0.7
x_3	2.2	2.9
x_4	1.9	2.2
x_5	3.1	3.0
x_6	2.3	2.7
x_7	2.0	1.6
x_8	1.0	1.1
x_9	1.5	1.6
x_{10}	1.1	0.9



Mean A: 1,81
Mean B: 1,91

PCA: Approach

- In each dimension, the mean value is subtracted from the data.

- The mean value of the transformed data is then 0.

$\mathbf{x}'_1$	0.69	0.49
$\mathbf{x}'_2$	-1.31	-1.21
$\mathbf{x}'_3$	0.39	0.99
$\mathbf{x}'_4$	0.09	0.29
$\mathbf{x}'_5$	1.29	1.09
$\mathbf{x}'_6$	0.49	0.79
$\mathbf{x}'_7$	0.19	-0.31
$\mathbf{x}'_8$	-0.81	-0.81
$\mathbf{x}'_9$	-0.31	-0.31
$\mathbf{x}'_{10}$	-0.71	-1.01

- The covariance matrix is then calculated:

$$\mathbf{C} = \begin{pmatrix} 0.617 & 0.615 \\ 0.615 & 0.717 \end{pmatrix}$$

- Since the elements apart from the diagonals are positive, there is a positive correlation between the two features (cf. correlation coefficient).

- The eigenvalues and eigenvectors of the matrix C are:

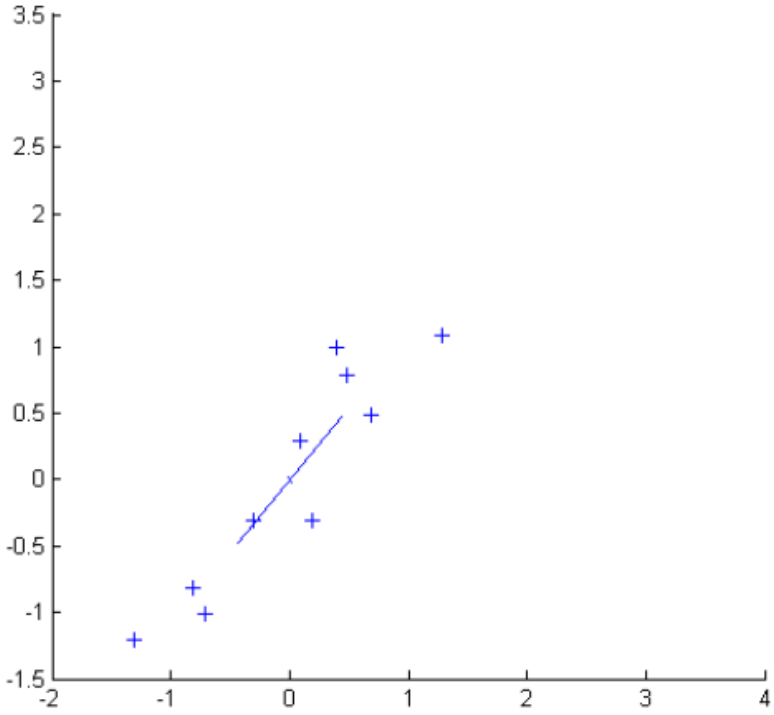
$$\mathbf{v}_1 = \begin{pmatrix} -0.678 \\ -0.735 \end{pmatrix} \text{ mit } \lambda_1 = 1.284$$

$$\mathbf{v}_2 = \begin{pmatrix} -0.735 \\ 0.678 \end{pmatrix} \text{ mit } \lambda_2 = 0.049$$

- The eigenvectors have length one and are orthogonal to each other
- $\mathbf{v}_1$ (higher eigenvalue) describes the first main component, $\mathbf{v}_2$ the second.

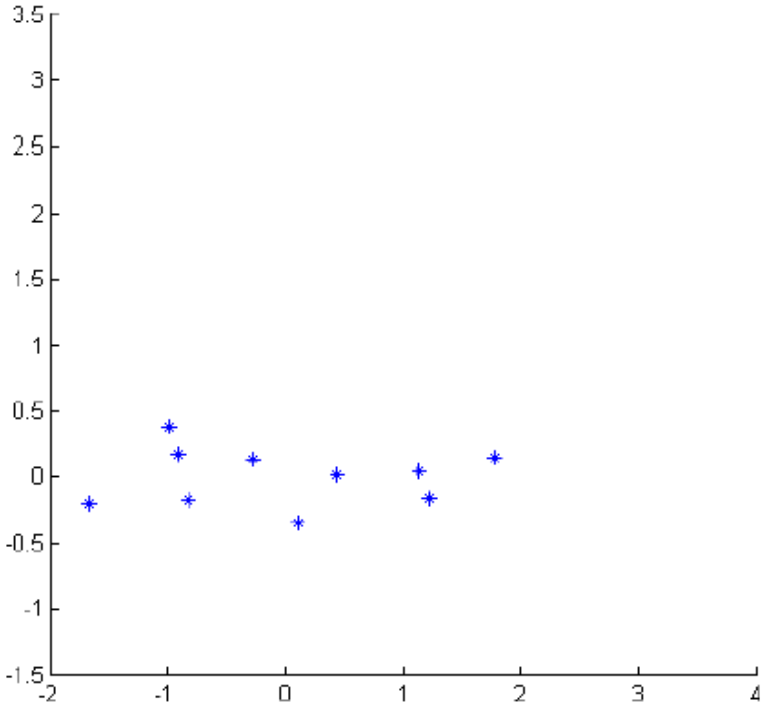
Illustration

- Starting from the mean value, each eigenvector is drawn in both directions; length corresponds to the eigenvalue.



- Transformation of the data using both eigenvectors:

$\mathbf{y}_1$	-0.828	-0.175
$\mathbf{y}_2$	1.778	0.143
$\mathbf{y}_3$	-0.992	0.384
$\mathbf{y}_4$	-0.274	0.130
$\mathbf{y}_5$	-1.676	-0.209
$\mathbf{y}_6$	-0.913	0.175
$\mathbf{y}_7$	0.099	-0.350
$\mathbf{y}_8$	1.145	0.046
$\mathbf{y}_9$	0.438	0.018
$\mathbf{y}_{10}$	1.224	-0.163

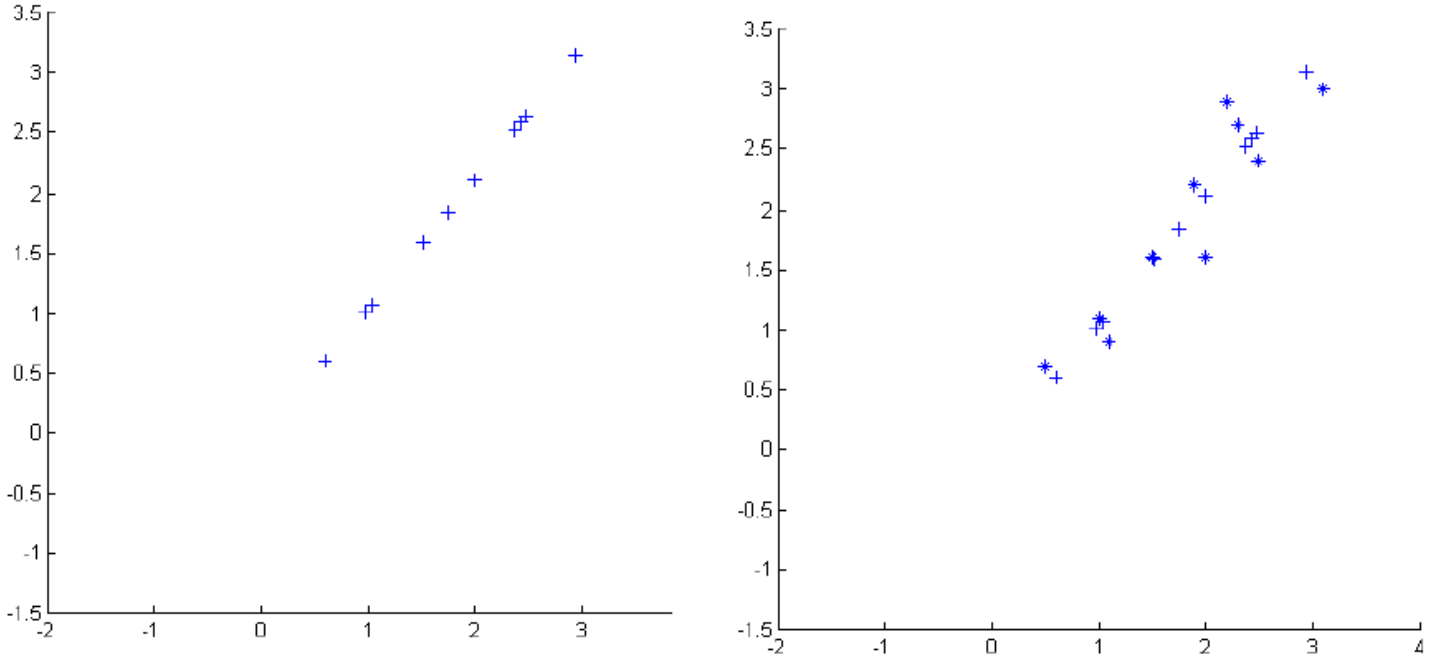


- Transformation of the data using the eigenvector with the higher eigenvalue:

$\mathbf{y}_1$	-0.828
$\mathbf{y}_2$	1.778
$\mathbf{y}_3$	-0.992
$\mathbf{y}_4$	-0.274
$\mathbf{y}_5$	-1.676
$\mathbf{y}_6$	-0.913
$\mathbf{y}_7$	0.099
$\mathbf{y}_8$	1.145
$\mathbf{y}_9$	0.438
$\mathbf{y}_{10}$	1.224

This, of course, corresponds to the first column in the table of the previous slide!

- Back-transformation of this data shows the loss of information!
- (+: back-transformed data, *: original data record)
- (Corresponds to the projection of the data onto the axis described by the first principal component.)

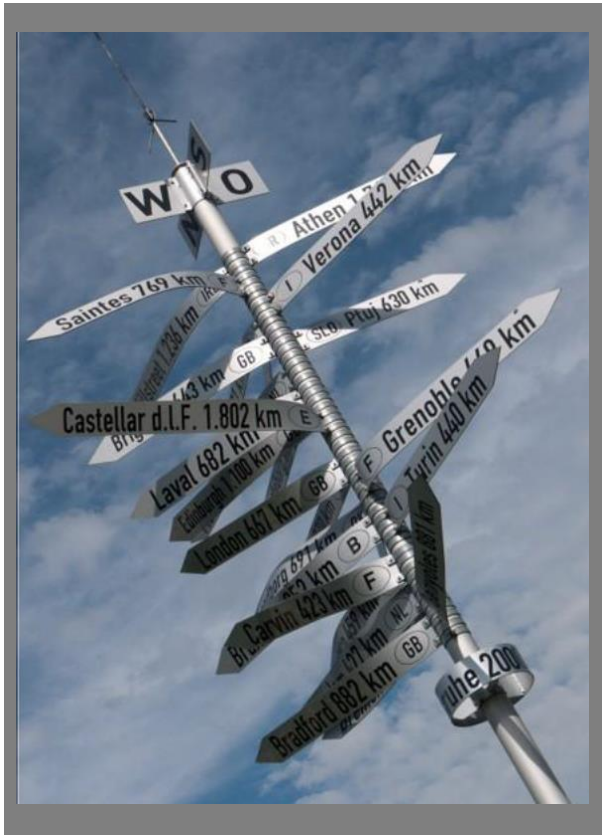


Some final remarks to PCA

- Which criteria are used to determine a suitable number D' of main components for data reduction?
- The sum of the eigenvalues of the most important D' eigenvectors should account for a certain proportion (e.g. at least 0.75) of the sum of all D eigenvalues.
- Dimensions are omitted if the eigenvalues of the corresponding eigenvectors are lower than the average of all eigenvalues.
- The eigenvalues are represented according to the descending importance of the eigenvectors. If this curve becomes significantly flattered at one point, the corresponding dimensions are omitted (so-called elbow or knee criterion).
- ...

Agenda

- Feature extraction
- Time-independent features
- Time-dependent features
- Feature selection
- Taxonomy of feature selection
- Feature selection algorithms
- Feature transformation
- Principle Component Analysis
- Conclusion and further readings

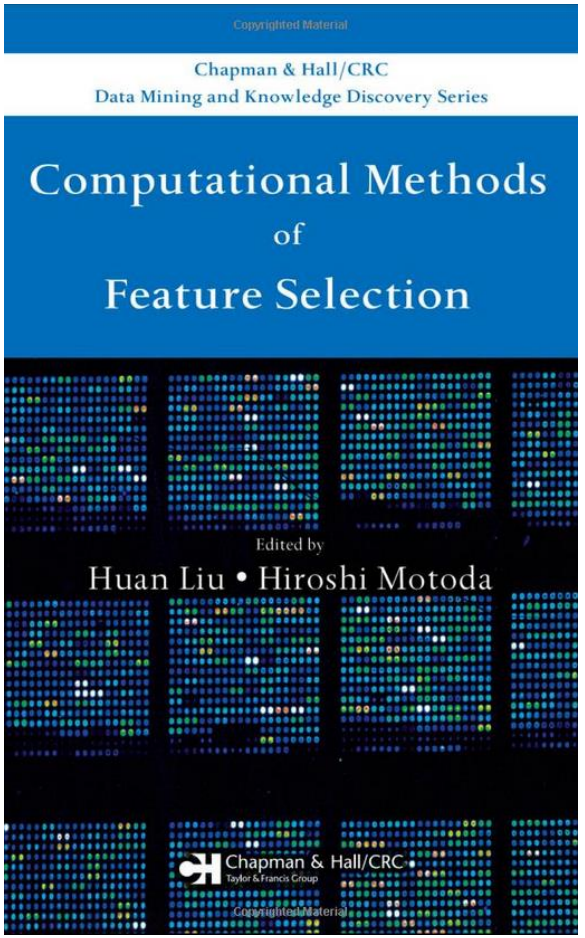


- Representations can be adjusted adaptively to the conditions of the time series.
- The Principal Component Analysis serves to transform the data, whereby information content is determined by variance.
- Piecewise forms of representation usually try a segmented approximation, e.g. via polynomials.
- Feature selection is used, among other things, to reduce the search space.
- A basic distinction is made between filters and wrappers
- This requires evaluation measures: Information, accuracy and consistency measures are the most important representatives here.

Basic readings:

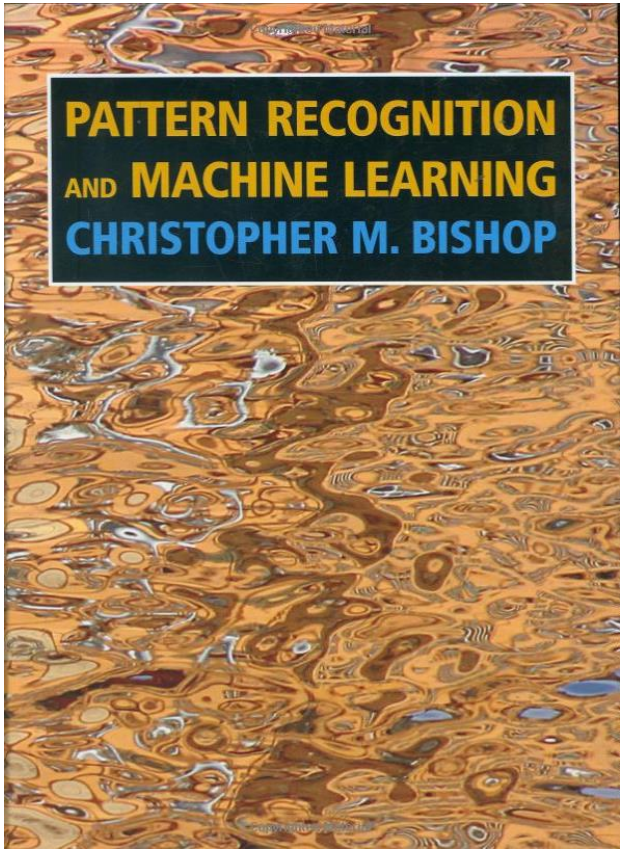
- Huan Liu, Hiroshi Motada
- Computational Methods of Feature Selection
- Chapman & Hall/CRC Data Mining and
- Knowledge Discovery Series
- 2007
- ISBN: 978-1584888789

Contents and illustrations partly based on lecture “Intelligent Technical Systems” by Prof. Paul Lukowicz (DFKI Kaiserslautern)



Basic readings in general:

- Christopher M. Bishop
- Pattern Recognition and Machine Learning (Information Science and Statistics)
- Springer Verlag
- 2011, 2nd edition
- ISBN: 978-0387310732



Further readings (3)

- [Lin et al., 2003]: Lin, Keogh, Lonardi, Chiu, A Symbolic Representation of Time Series, with Implications for Streaming Algorithms 2003
- [Mitsa 2010]: Mitsa, Theophano. Temporal data mining. CRC Press, 2010.
- [Chakravarty 2000]: Chakravarty, Shahar, CAPSUL: A constraint-based specification of repeating patterns in time-oriented data, 2000
- [Fu 2008]: Fu, Chung, Luk, Ng, Representing financial time series based on data point importance 2008
- [Keogh et al., 2001]: Keogh, Chakrabarti, Mehrotra, Pazzani, Locally Adaptive Dimensionality Reduction for Indexing Large Time Series Databases 2001
- [Keogh 1998]: Keogh, Pazzani, An enhanced representation of time series which allows fast and accurate classification, clustering and relevance feedback 1998

- Any questions…?