UNIVERSITÄT OSNABRÜCK

Machine Learning Methodology

Kai-Uwe Kühnberger
Universität Osnabrück
08.01.2021

UNIVERSITÄT OSNABRÜCK

Machine Learning Methodology: Schedule

- Machine Learning:
  - What is it?
  - Concepts and Methodologies
- Example: Health Prediction
- Important Concepts
  - Similarity and Distance
  - Classification and Features
- Clustering methods
- Properties of Hypotheses

What is Machine Learning?

- Machine learning…
  - ... is any change of a system which enables the system to solve a similar task better next time. [Simon]
  - ... is the construction or change of representations of perceptions. [Michalski]
  - ... [are] algorithms [to] improve their performance with experience. [Langley, 1996]
  - ... means acquiring a program by any means other than explicit programming. [Valiant, 1984]
  - ... is an approximation problem
  - If the learner can't compress the data, he or she doesn't learn [Li and Vitanyi, 1997]

UNIVERSITÄT OSNABRÜCK

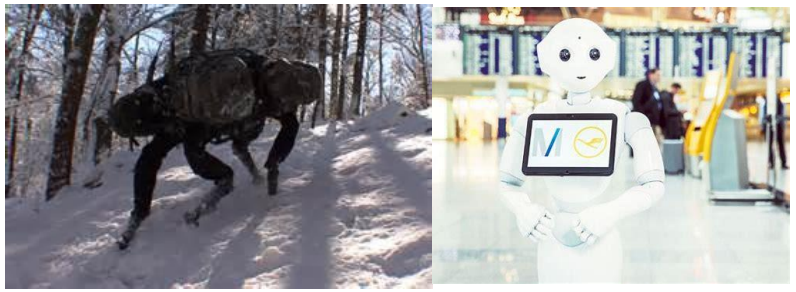
Machine Learning is Everywhere

Amazons Alexa

Smart home: heating, lightening, entertainment, shades etc.

Autonomous Google Car

Autonomous Daimler Truck



Boston Dynamics Big Dog, Softbank's Pepper at Munich Airport

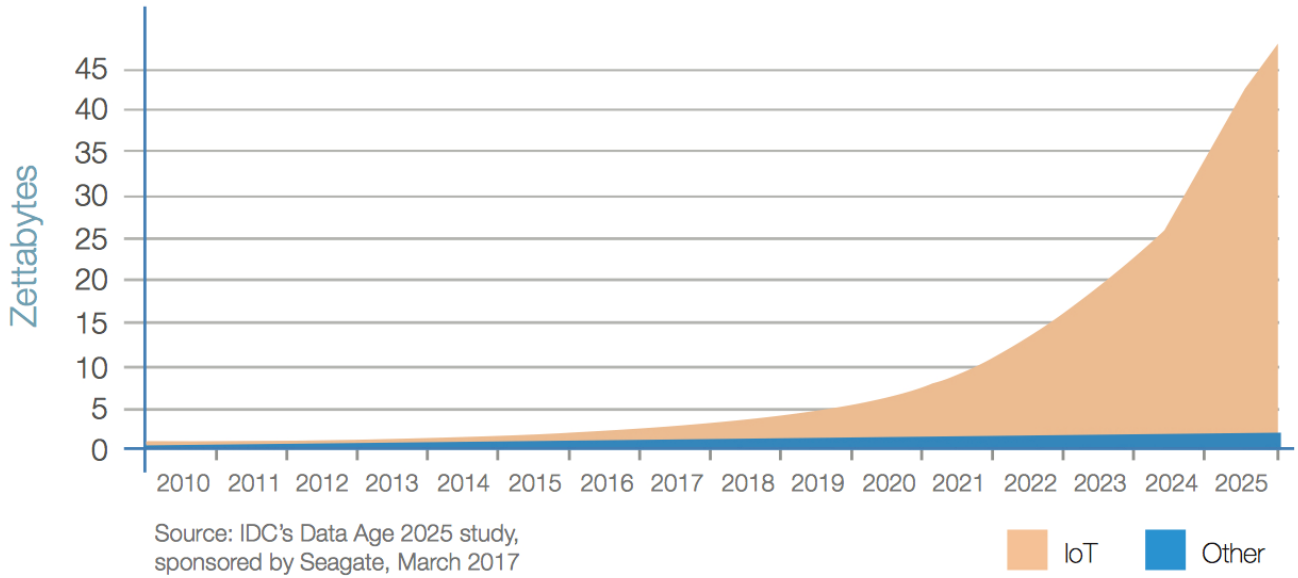
Machine Learning is Everywhere

- **Which tools and services are or will be available soon or significantly further improved?**
  - Predictive maintenance in industry
  - Autonomous mobility
  - Improved quality of products
  - Improved weather forecasts
  - Artificial assistants
  - Document management
  - Smart homes
  - Individual risk assessments in the health area
  - Automation of consultancy services
  - …

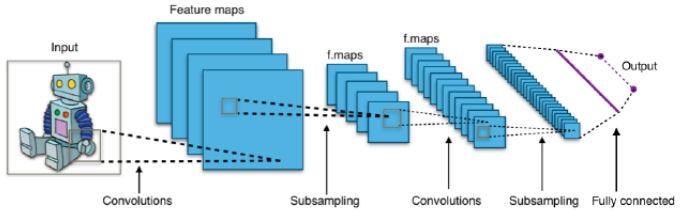
UNIVERSITÄT OSNABRÜCK

Machine Learning is Everywhere

- **What is the basis of future developments in AI?**
  - Human generated data and IoT generated data are growing exponentially.
  - BIG DATA + (Deep) Machine Learning = New Applications



Source: Storagenewsletter (2017)



Source: Garita (2018)

UNIVERSITÄT OSNABRÜCK

Examples of Learning

- Conceptualizations of learning
  - How can we learn? Here are the classical three learning types:
    - "Find patterns in the data"
      - **Unsupervised learning**
    - "By trial-and-error and feedback from the environment"
      - **Reinforcement learning**
    - "With an experienced teacher"
      - **Supervised learning**

- Perhaps one needs more learning types:
  - Learning step by step (until we run).
    - By example
  - Analogical learning
    - Learning guided by conceptualizations
  - Transfer learning
    - With networks or case-based reasoning
  - Etc.

UNIVERSITÄT OSNABRÜCK

Domains for Learning (Examples)

- **Classification learning:**
  E.g. learning to classify text into text genres, to recognize objects in an image, to interpret a gesture.
- **Reinforcement learning:**
  E.g. learning to grasp something, to hit the baseball with a bat, to find a policy to reach the next level in a computer game .
- **Skill acquisition:**
  E.g. learning how to solve problems, how to drive cars, how to play the piano.
- **Discovery:**
  E.g. learning laws from empirical data.
- **Meta-learning:**
  E.g. learning how to learn.

- Concrete examples for learning applications.
  - Credit risk management.
  - Speech recognition.
  - Text Mining.
  - Text Classification.
  - Face recognition.
  - Image recognition.
  - Bioinformatics.
  - Elevator group control.
  - Playing chess, go…
  - Etc.

Induction vs. Deduction

- Deductive reasoning:
  - The conclusion is entailed by the premises.
  - Entailment is defined by the semantics of the logical calculus.

- Inductive reasoning:
  - The conclusion is obtained by a generalization of the premises.
  - Different notions of generalization lead to different machine learning approaches.

- Traditional syllogism.
  - Premise 1:
    - *A lizard is a reptile.*
  - Premise 2:
    - *All reptiles have a scaly skin.*
  - Conclusion:
    - *Lizards have a scaly skin.*

- Inductive inference:
  - Premise 1:
    - *A lizard is a reptile.*
  - Premise 2:
    - *Lizards have a scaly skin.*
  - Conclusion:
    - *All reptiles have a scaly skin.*

Methodologies used for Machine Learning

- High-level descriptions (knowledge), symbolic computations.
  - Examples: inductive learning, inductive logic programming, decision/regression tree learning, version space.
- Sub-symbolic computations (biologically inspired).
  - Examples: Neural networks, connectionist systems, deep (reinforcement) learning, recurrent NNs.
- Survival of the fittest.
  - Examples: Genetic algorithms (biologically inspired by evolution).
- Learning by statistical algorithms.
  - Example: Markov decision processes, Markov logic networks, Bayesian learning,
- Further approaches: Support vector machines.

Example

Health Prediction in a Study Project

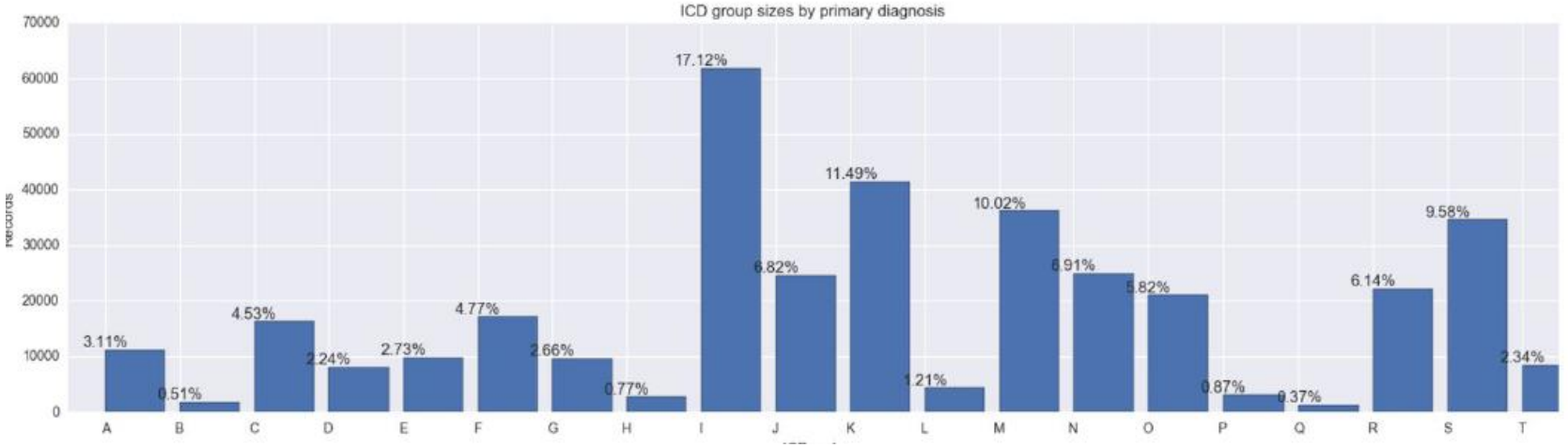
Participants: Kai Fritsch, Carolin Gaß, Alexander Höreth, Inga Ibs, Ann-Christin Meisener, Tim Patzelt, Justin Shenk, Geeske Sieckmann, Andrea Suckro

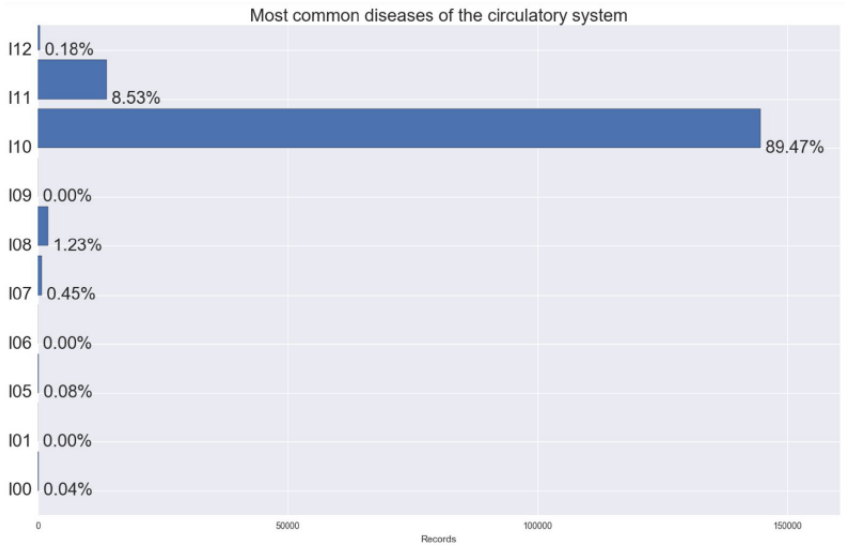
Predicting Health Issues

- One-year study project
- Data:
  - Data provided by a Berlin-based company
  - Anonymous Data of 360,000 patient entries from 8 different hospitals
  - Types of available and anonymous data: age, gender, date of entrance and discharge, diagnosis, treatments etc.
    - Diagnosis and treatment is coded in ICD codes and OPS codes.
    - Additionally, lab tests were available.
  - This results in significantly more than 50 millions feature-value pairs.
- Task:
  - Predictions of ICD (diagnosis), OPS (treatment), and DRG (payments) codes, length of stay of patients etc.
  - Prediction of missing lab values (e.g. in blood tests)
- Problems: Data was not consistent; no standard formats; different IDs, different units for blood tests etc.

UNIVERSITÄT OSNABRÜCK

Distribution of Data





Preprocessing & Feature Selection

- Normalization of lab values
- Balancing data sets is required because there are significantly more entries with normal values than entries with abnormal values
  - This can result sometimes in rather small sets, although the original set was rather large.
- Data set divided into training set (50%), test set (25%), and validation set (25%).
- Etc.
- Feature selection:
  - The idea is to find a maximal set of patients relative to a subset of test features of cardinality *n*.

Predicting ICD Codes: N17

- Task: Prediction of N17 (acute kidney failure)
- Methods used for ICD classification include
  - Logistic regression (LR)
  - Support Vector Machines (SVM)
  - Decision trees (DT)
  - Random forests (RF)
- Pre-processing: collect all data in the unified relational database, unify lab test identifiers and fix encoding problems.
- **Problem: find task tailored subsets of data such that**
  - Only records with common and discriminating features are used
  - Dropping erroneous records quite often results in small subsets. This leads to the challenge of limited and unbalanced datasets.

UNIVERSITÄT OSNABRÜCK

Predicting ICD Codes: N17

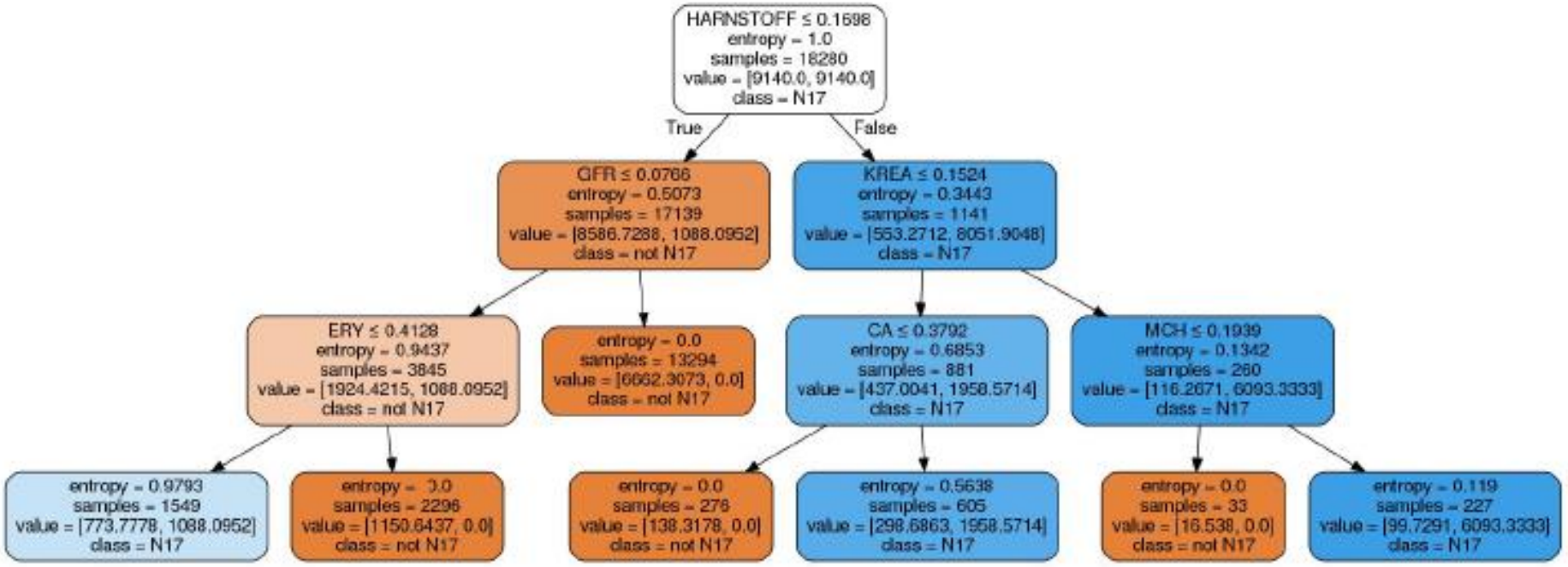


Figure 1. AKF Decision Tree

- Features that could be relevant for detecting N17 in a decision tree.

Predicting ICD Codes: N17

Scores	SVM	LR	DT	RF
True Positive	**0.866**	**0.881**	**0.851**	**0.805**
True Negative	0.925	0.927	0.882	0.957
False Positive	0.075	0.073	0.118	0.042
False Negative	**0.134**	**0.119**	**0.149**	**0.195**

- Results for predicting N17 (acute kidney failure) based on an appropriate feature set.
- Predictions of other ICD-Codes are sometimes more difficult, e.g. ICD code I10 (essential primary hypertension) shows around 0.24 false negatives for the best algorithm.

UNIVERSITÄT OSNABRÜCK

Clustering and Classification

Similarity Distances

Classification, Clustering & Reinforcement Learning

- Classification (supervised learning):
  - Classes are given, training data consists of data plus corresponding class
  - For a new, unseen example, decide to which class it belongs to.

- Clustering (unsupervised learning):
  - Given is a set of examples (without class membership).
  - Partition the set of examples into interesting subsets of similar elements.
  - Try to find descriptions / definitions of these subsets

- Reinforcement Learning
  - Given is a set of actions, sensor data, and a reward provided by the environment.
  - Find an optimal policy to reach a goal by maximizing the expected future reward.

Classification & Clustering

- Mathematical background for all approaches:
  - Similarity between examples / input
  - Distance between examples / input
  - Feature spaces
  - Classification functions

- These principles are independent from concrete learning methods.

Similarity <=> Distance

- Similarity measures
  - $sim : U \times U \rightarrow [0,1]$
  - $sim(x, x) = 1$
  - If $sim(x, y) = 1$, then x and y are indiscernible (indistinguishable)
  - $sim(x, y) = sim(y, x)$ [?]
  - Example "cosine similarity" in a (positive) vector space:
    $cos(\mathbf{x}, \mathbf{y}) = (\mathbf{x} \cdot \mathbf{y}) / |\mathbf{x}||\mathbf{y}|$

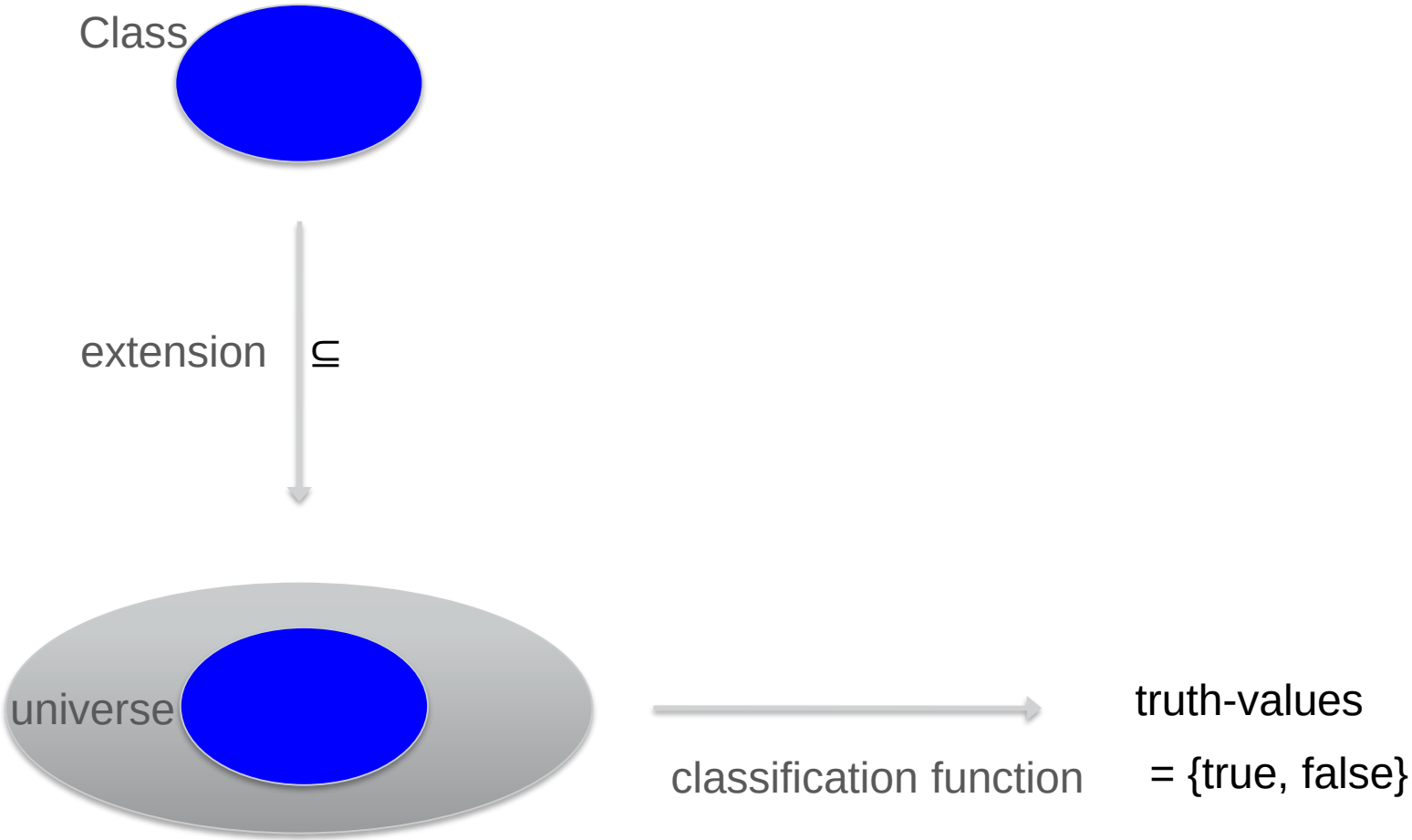
- Distance measures → metric spaces
  - $d : U \times U \rightarrow \Re$
  - $d(x, y) \geq 0$
  - $d(x, x) = 0$
  - $d(x, y) = d(y, x)$
  - $d(x, z) \leq d(x, y) + d(y, z)$ (triangle inequality)
  - $x \neq y \Rightarrow d(x, y) > 0$
  - Example Euclidian distance: $d(\mathbf{x},\mathbf{y}) = ((x_1 - y_1)^2 + (x_2 - y_2)^2 + \ldots + (x_n - y_n)^2)^{1/2}$

Similarity <=> Distance

- Translation between similarity measures and distance measures:
  - $sim(x, y) = 1 / (1 + d(y, x))$ or $sim(x, y) = e^{-d(y, x)}$
  - $d(x, y) = [1/sim(y, x)] - 1$ or $d(x, y) = log\ sim(y, x)$ (provided $sim(y, x) \neq 0$)

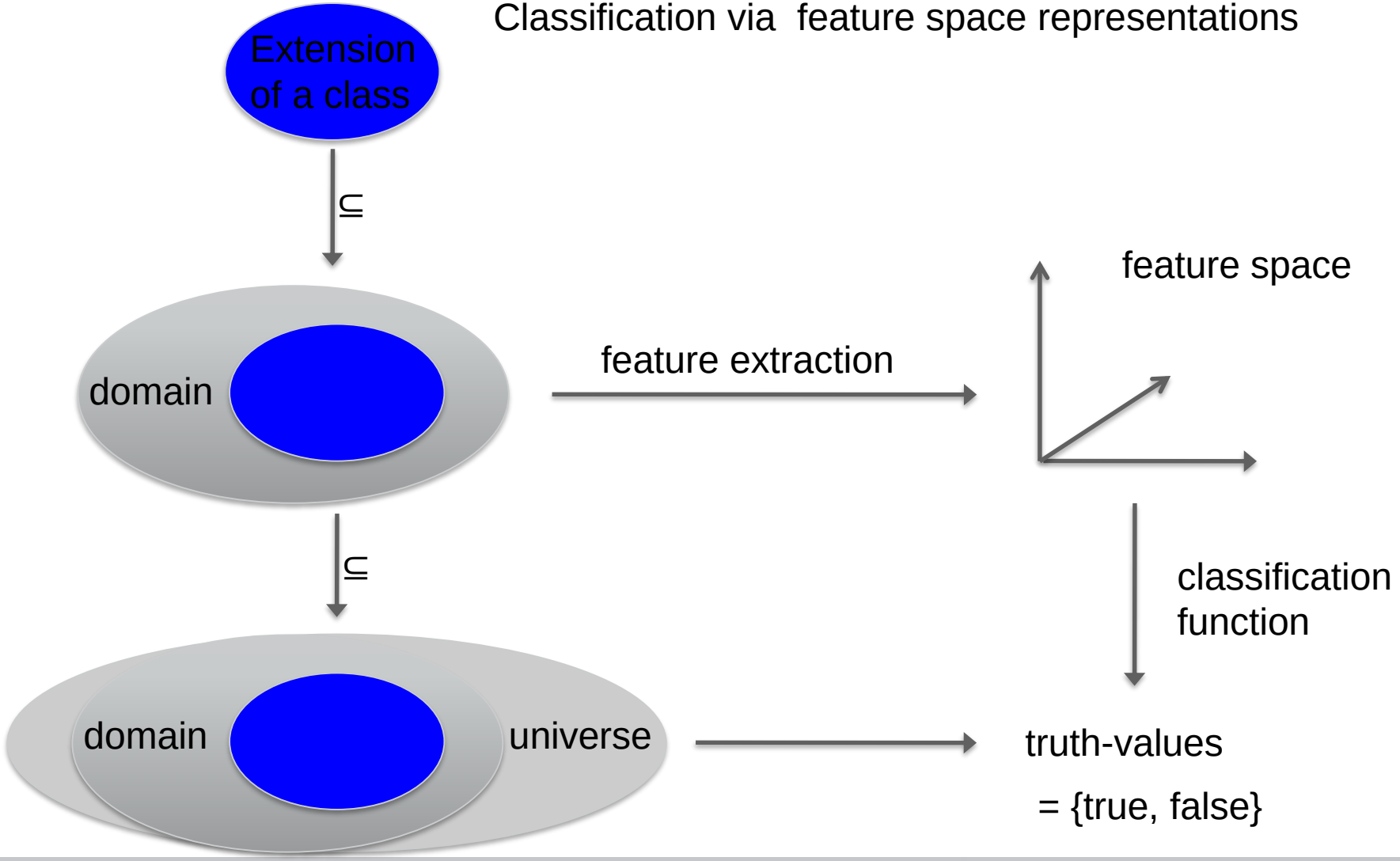
- Problems
  - Where do these measures come from, if *U* is not a metric space?
    - What does it mean that two chairs are similar?
  - Which properties do similarity / distance measures have?
    - Triangle inequality?
    - Is similarity transitive?
    - Is similarity symmetric?
  - Similarity judgments may have none of these properties! (Tversky, 1977)
  - For a discussion of these concepts with respect to human-inspired concepts of similarity and distance compare Quesada (2008).

Classification

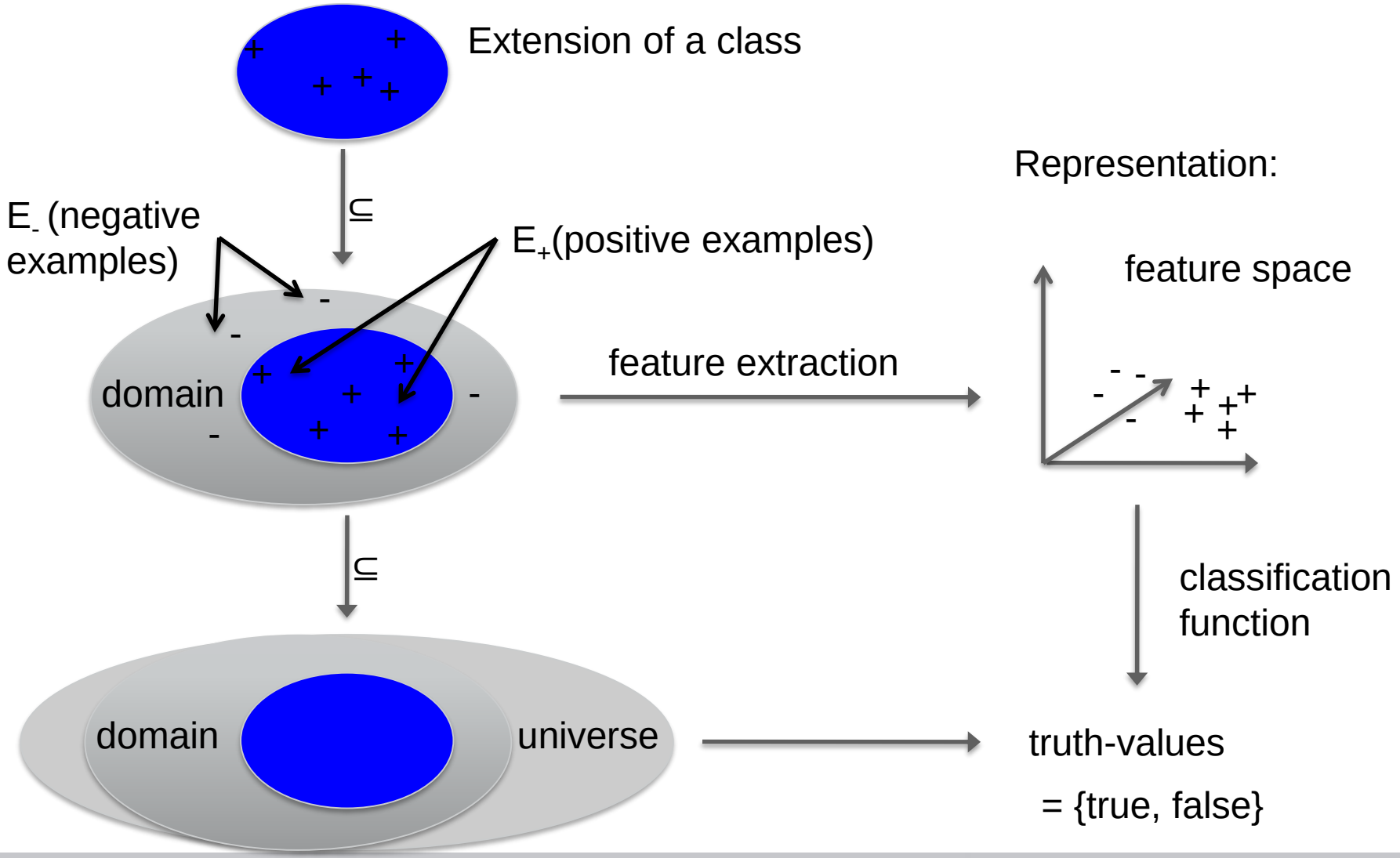


Classification

Classification via feature space representations



Classification



Feature spaces

- Vector space model
  - Embed D into $\Re^n$ to inherit the topology of $\Re^n$
  - $r : D \rightarrow \Re^n$
  - $r(x) = (f_1,..., f_n)$
    - The dimensions of $\Re^n$ may have an interpretation.
    - They are called *features*.

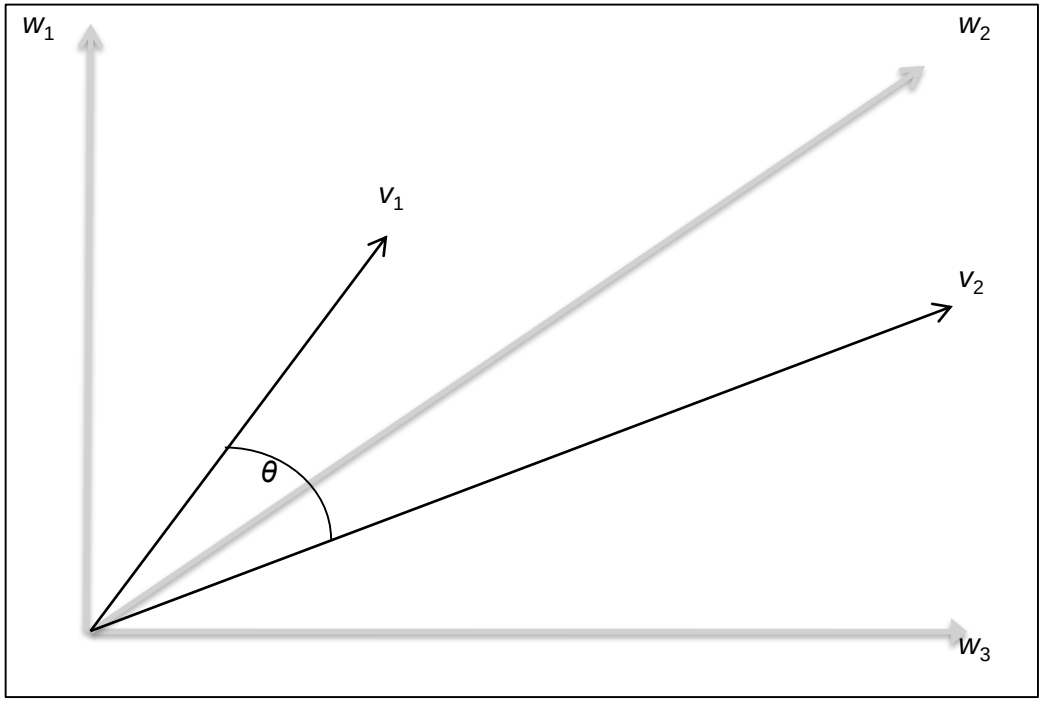
- Example: Documents
  - Features are (normalized) words
  - Consider function values dependent on the number of occurrences of the words in the documents, e.g. TF-IDF (term frequency times inverse document frequency)
  - Similarity measure: cosine
  - Application: clustering the results of a search engine

UNIVERSITÄT OSNABRÜCK

Feature spaces: Example

- Term frequency *tf* can be defined in many ways:
  - Boolean frequency
  - Raw count frequency

  f(*t*,*d*): number of times a word occurs in a document
  - Below: *tf* is logarithmically scaled term frequency
- *idf* is a measure how much information a word provides



$$sim(v_1, v_2) = \cos(\theta) = \frac{v_1 v_2}{\|v_1\| \|v_2\|}$$

$$tf(t_i, d_j) = 1 + \log f(t_i, d_j)$$

$$tf - idf(t_i, d_j, D) = tf(t_i, d_j) \log \frac{|D|}{|\{d \in D \mid t_i \in d\}|}$$

(Dis-)Similarity Matrix

	I_1	I_2	...	I_i	...	I_{n-1}	I_n
I_1	0	$d(I_1,I_2)$	...	$d(I_1,I_i)$	...	$d(I_1,I_{n-1})$	$d(I_1,I_n)$
I_2	$d(I_2,I_1)$	0	...	$d(I_2,I_i)$	...	$d(I_2,I_{n-1})$	$d(I_2,I_n)$
...	...	...	...	...	...	...	...
I_i	$d(I_i,I_1)$	$d(I_i,I_2)$	...	0	...	$d(I_i,I_{n-1})$	$d(I_i,I_n)$
...	...	...	...	...	...	...	...
I_{n-1}	$d(I_{n-1},I_1)$	$d(I_{n-1},I_2)$	...	$d(I_{n-1},I_i)$	...	0	$d(I_{n-1},I_n)$
I_n	$d(I_n,I_1)$	$d(I_n,I_2)$	...	$d(I_n,I_i)$	...	$d(I_n,I_{n-1})$	0

Clustering

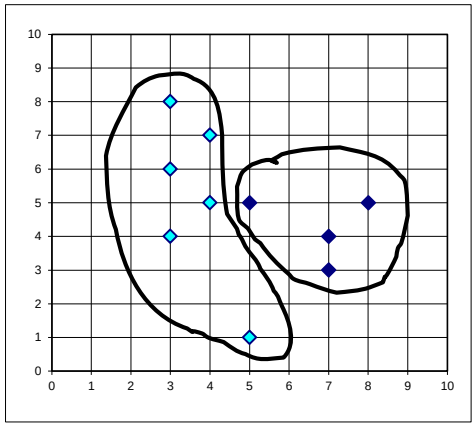
Putting things together which are similar

Clustering

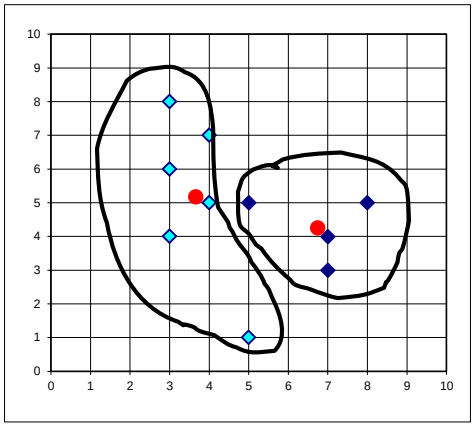
- Centroid models
  - Use the centroid (mean representation of all elements) to represent a cluster
  - Use minimal distance to the centroids for classification
- Prototype models
  - Use one (or more) elements as prototypes
  - Use minimal distance to the prototypes for classification
- *k*-means
  - Use *k* initial clusters (*k* initial centroids)
  - Classify each element and compute new centroids
  - Repeat until clusters don't change
- Hierarchical models
  - Single link
  - Complete link

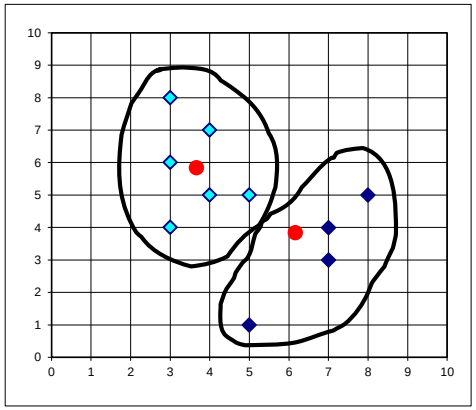
Clustering Using *k*-Means

initial classes

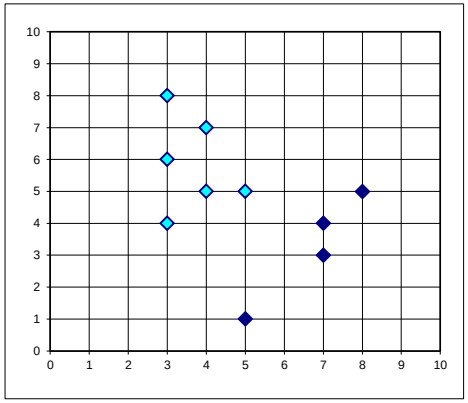


centroids



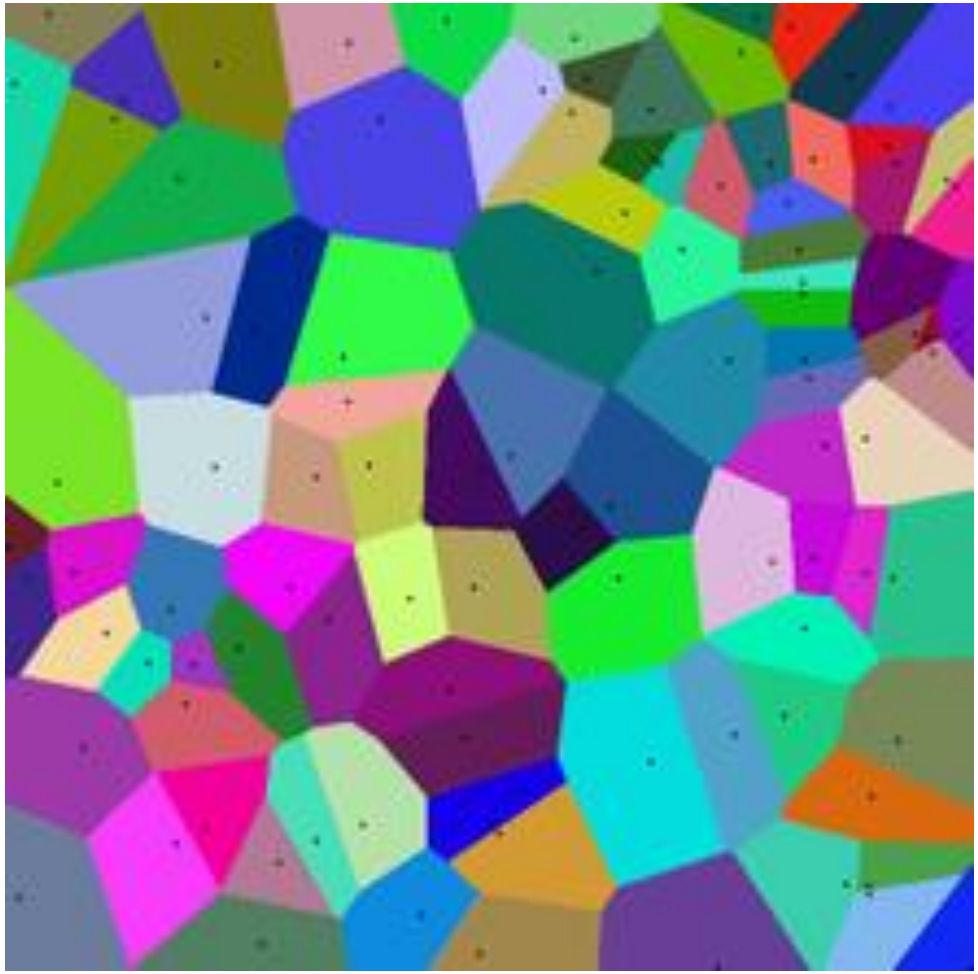


new centroids



new classification

Distance Based Clustering Method: Voronoi Diagrams

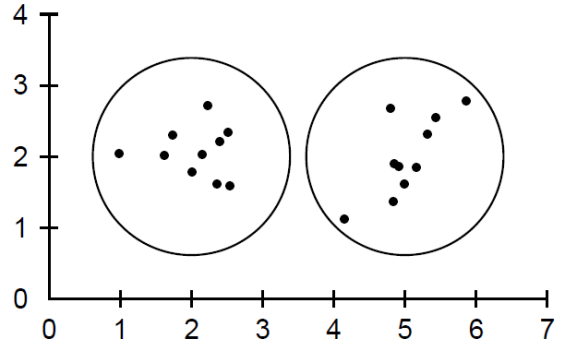


Hierarchical Clustering

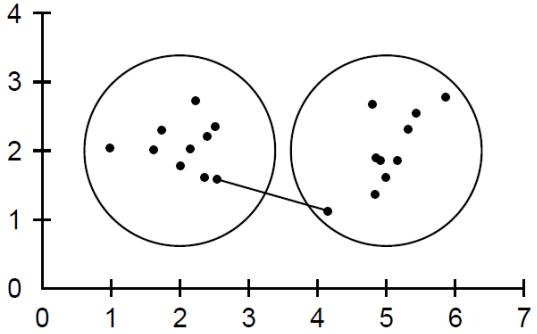
- Top down clustering:
  - Start with one cluster containing all elements.
  - Select a cluster and split it until the intended number of clusters is reached.

- Bottom up:
  - Start with one cluster per element.
  - Join the nearest clusters.
    - Different distance measures for clusters are possible.
      - Minimal distance between elements of each cluster is considered (single link clustering).
      - Maximal distance between elements of each cluster is considered (complete link clustering).
      - Mean distance between elements of each cluster.
- Results are trees of subsets (dendrograms).
- No pre-assumed number of clusters necessary.

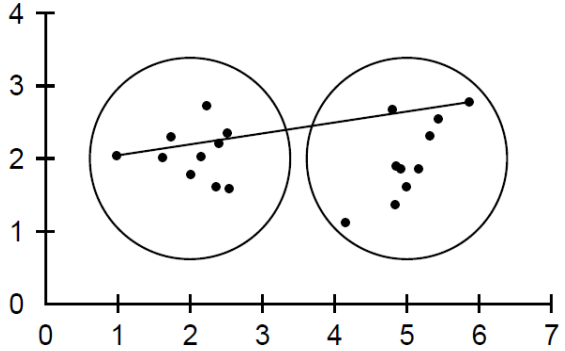
Hierarchical Clustering



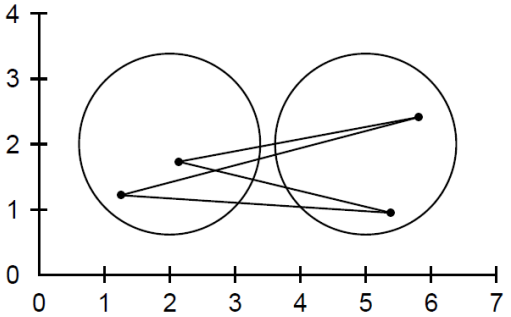
Classes with elements



Single-link distance



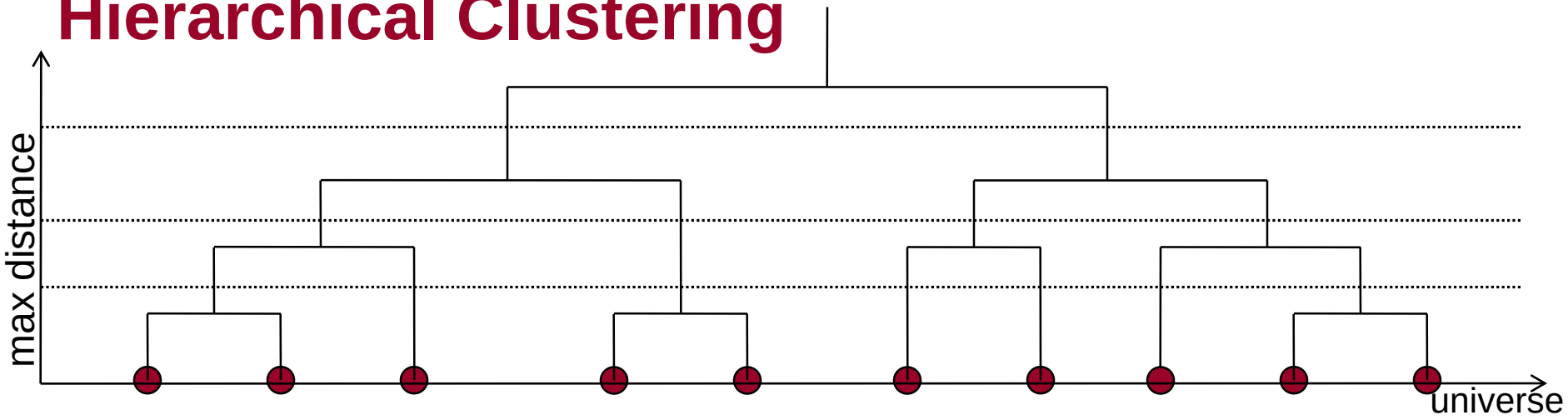
Complete link distance

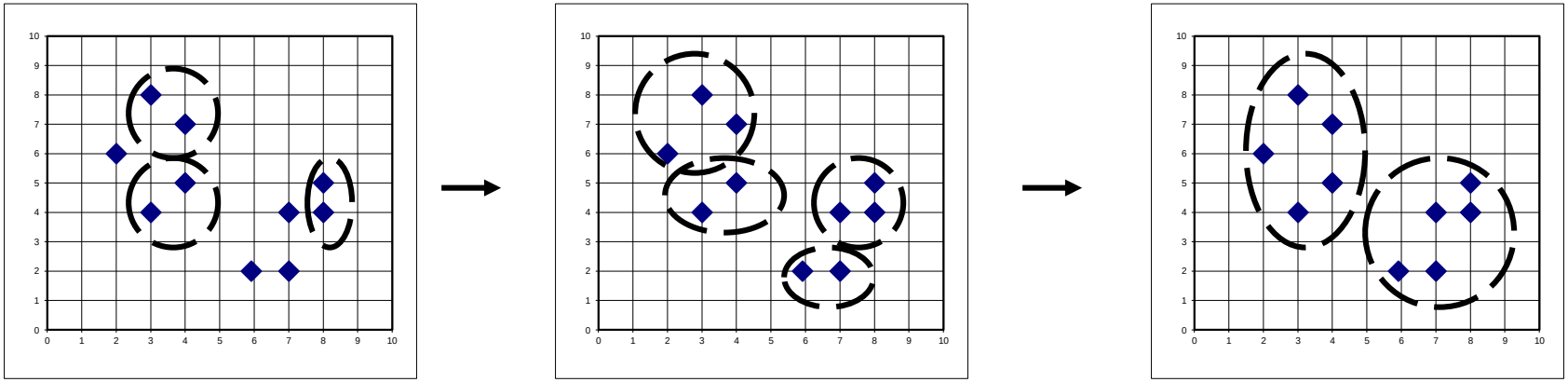


Mean / average distance

UNIVERSITÄT OSNABRÜCK

Hierarchical Clustering



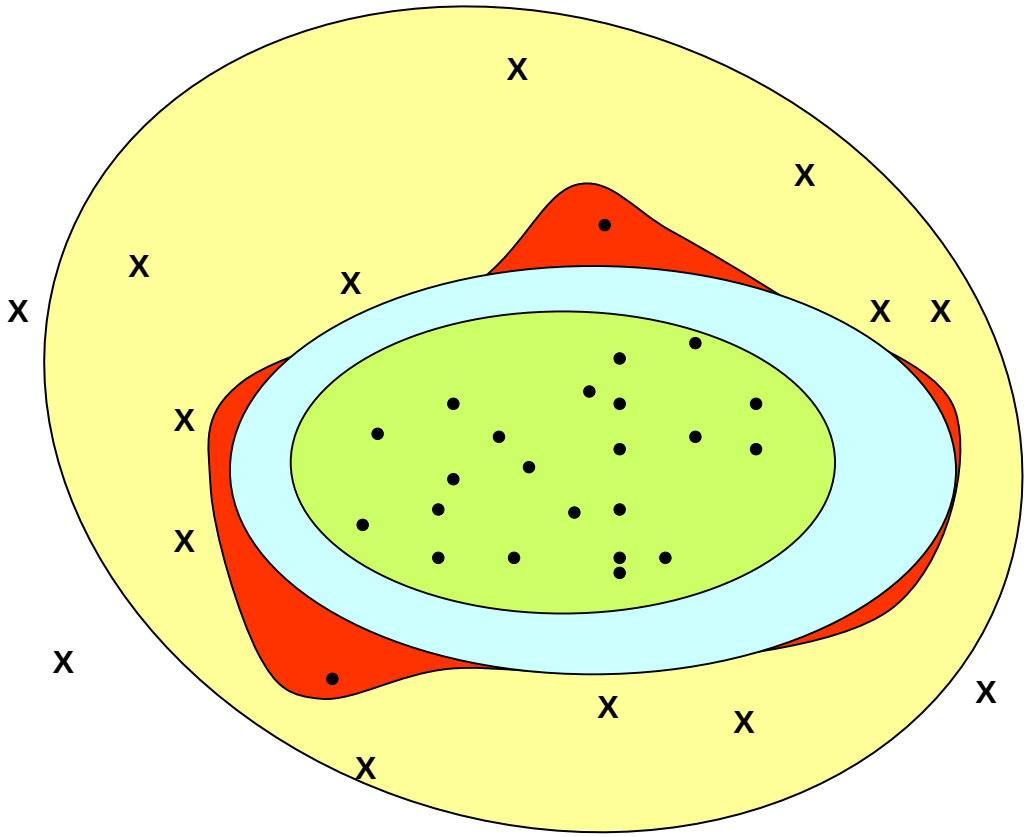


Clustering

- Quality of clustering
- Maximize: Intra-class similarity
  - Minimize mean distance of elements of a class.
- Maximize: Inter-class dissimilarity
  - Maximize mean distance of elements of different classes

Properties of Hypotheses

Examples and Hypotheses



Different Kinds of Hypotheses.

We could think about…

… Different hypotheses that only take correct examples into account (but miss perhaps some of the correct ones)

… Hypotheses that take into account definitely all correct examples (but include perhaps some false ones)

Properties of Hypotheses

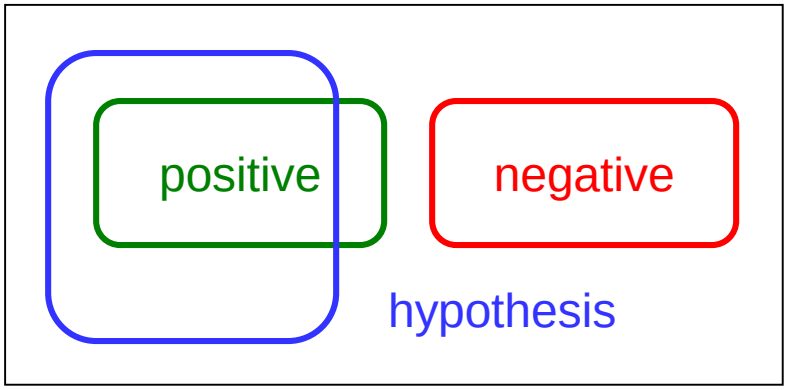
- Concepts we need for assessing hypotheses:
  - A machine learning algorithm *A* is given.
    - *A* receives a sample *s* of examples encoded in a language L_E.
      - $s = [<x_1,t(x_1)>, <x_2,t(x_2)>,\ldots,<x_n,t(x_n)>]$.
      - *t* is a target concept.
    - *A* has background knowledge Σ encoded in L_Σ.
    - *A* outputs a hypothesis *h* encoded in L_H.
      - $\Sigma \cup \{h\} \models E^+$ (completeness)
      - $\Sigma \cup \{h\} \not\models E^-$ (correctness)

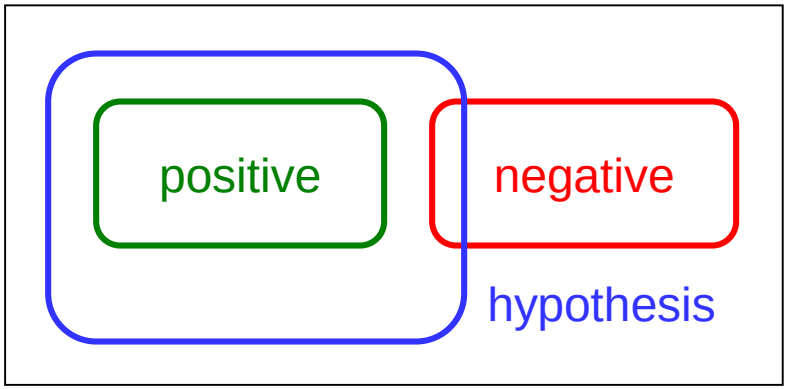
- Goal:
  - The goal is to approximate a target concept *t* by a hypothesis *h.*
  - We assume that the new theory shall explain more observations than the old theory (with less complexity).

Properties of Hypotheses

- The intuitive meaning of correctness and completeness:

- Correctness:
  - Hypothesis covers no negative example.
- Completeness:
  - Hypotheses covers all positive examples.
- General strategy:
  - Specialize incorrect and generalize incomplete hypotheses.
  - An algorithm doing precisely this is version space learning





Properties of Hypotheses

- In case where there is no complete and correct hypothesis *h*, precision and coverage are used to describe the quality of *h.*
  - Precision (measures the “degree of correctness”):
    - The number of positive examples supported by *h* divided by all examples supported by *h* (true positives TP divided by sum of true positives TP and false positives FP).

$$pre(h) = \frac{|\{e \in E^+ | \Sigma \cup \{h\} \models e\}|}{|\{x \, | \Sigma \cup \{h\} \models x\}|} = \frac{TP}{TP + FP}$$

  - Sensitivity / recall (measures the “degree of completeness”):
    - The number of positive examples supported by *h* divided by all positive examples (true positives TP divided by sum of true positives TP and false negatives FN).

$$rec(h) = \frac{|\{e \in E^+ | \Sigma \cup \{h\} \models e\}|}{|E^+|} = \frac{TP}{TP + FN}$$

  - In different fields different notions are used.

Properties of Hypotheses

- Further properties of hypotheses:
  - Accuracy (measures the "proportion of correct classifications"):
    - The sum of the number of true positives TP and true negatives FP divided by the sum of all data points).

$$acc(h) = \frac{TP + TN}{TP + TN + FP + FN}$$

  - F_1 score (measures the harmonic mean of precision and sensitivity/recall):

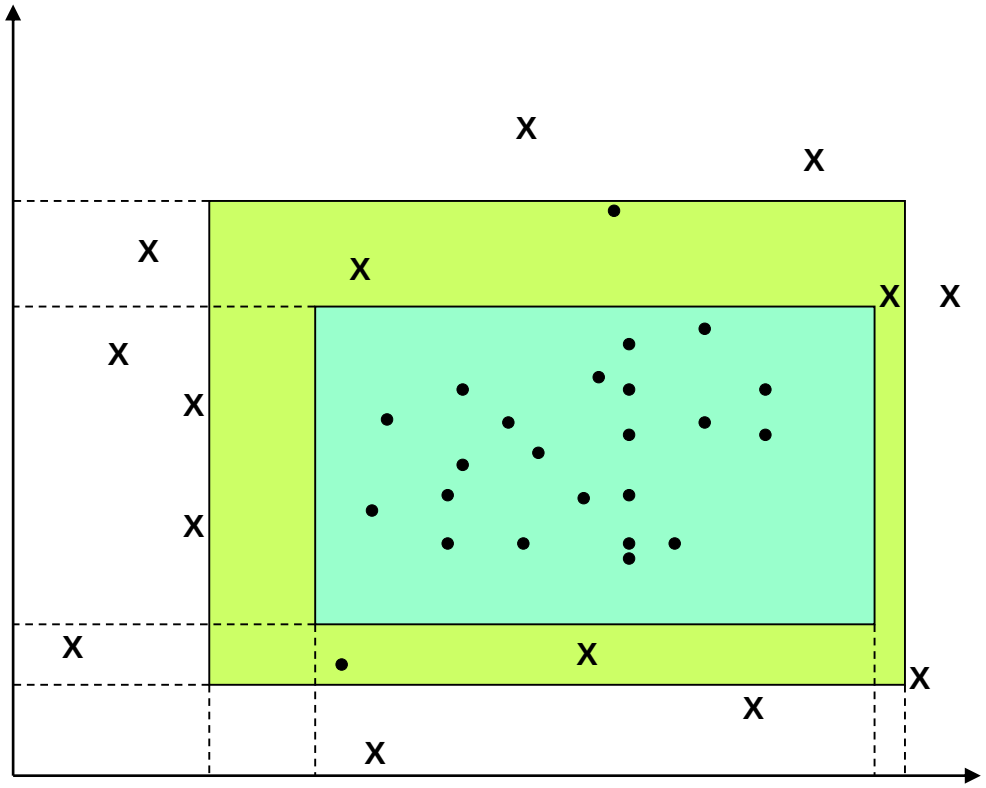
$$F_1(h) = \frac{2TP}{2TP + FP + FN}$$

  - A parameterized version of the F_1 score is the F_β score that allows to set a bias to recall and precision, e.g. that recall is β times more important than precision.

Remark concerning Hypothesis Spaces

- We saw that a machine learning algorithm *A* needs:
  - A language L_E to encode the sample.
  - A language L_Σ to encode the background knowledge.
  - A language L_H to encode the hypotheses.
  - Notice:
    - L_E must be suitable to encode the sample data, e.g. vectors of real numbers / integers / nominal values (in general: all kinds of scales).
    - L_Σ can, for example, be the language of predicate logic.
    - The quality of the modeling depends highly on the language L_H.
      - This language is usually quite restricted: often only conjunctions of equations can be expressed (i.e. *value*(*A*) = *x*).
    - On *H* we have a subsumption relation: $h_1 < h_2$ iff $h_1(x) \rightarrow h_2(x)$

Examples and Hypotheses

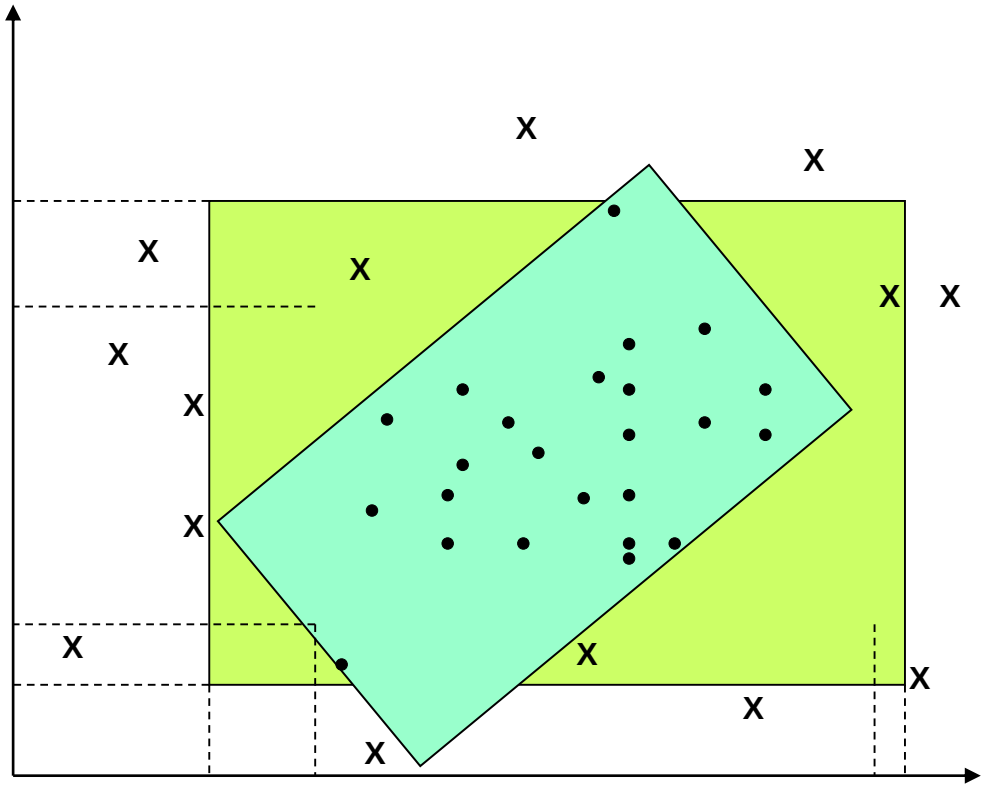


Assume the hypothesis space is restricted to rectangles in the plane (subranges for two attributes).

In the example, it is not possible to get a consistent hypothesis *h* which is complete and correct.

By changing the language – allowing more general shapes of areas – this problem can be potentially solved.

Examples and Hypotheses



Notice:

Even a change of the base of the coordinate system can solve the problem.

This corresponds to the possibility to combine features.

In machine learning particularly non-linear combinations of features can improve learning quality.

References

- General
  - Langley, P. (1996). Elements of Machine Learning, Morgan Kaufmann.
  - Li, M. & Vitanyi, P. (1997). An Introduction to Kolmogorov Complexity and Its Applications; Springer.
  - Valiant, L. (1984). A Theory of the Learnable, Communications of the ACM; pp. 1134-1142.
- Similarities
  - Quesada, J. (2008). Human Similarity Theories for the Semantic Web. In: Nature inspired Reasoning for the Semantic Web, 2008.
  - Tversky (1977). Features of Similarity; Psychological Review 84(4):327-352.