Overview on Machine Learning

Frederik Janssen

TECHNISCHE UNIVERSITÄT DARMSTADT

Outline

Acknowledgments and Goals

TECHNISCHE UNIVERSITÄT DARMSTADT

- sources used to prepare the slides
  - wikipedia (`http://en.wikipedia.org`) for figures and algorithm properties
  - VL “Maschinelles Lernen” by Prof. Johannes Fürnkranz (the lectures given in all the different semesters are available under `http://www.ke.tu-darmstadt.de/lehre`)
  - VL “Künstliche Intelligenz” by Prof. Johannes Fürnkranz (lectures available under same url as above)
  - various other internet sources
- note that I neither claim that this talk is complete nor that the given opinions are valid in general (it is my view on machine learning)
- Goals
  - give a brief introduction to machine learning
  - provide an overview over the “all” topics in machine learning
  - give insights in how the most important algorithms work (at least to some extent)
  - to ease researchers to apply machine learning for their tasks

Machine Learning

TECHNISCHE UNIVERSITÄT DARMSTADT

- Definition (Mitchell, 1997)
  - "A computer program is said to *learn* from experience *E* with respect to some class of tasks *T* and performance measure *P*, if its performance at tasks in *T*, as measured by *P*, improves with experience *E*."
- Given:
  - a task *T*
  - a performance measure *P*
  - some experience *E* with the task
- Goal:
  - generalize the experience in a way that allows to improve your performance on the task

Classification

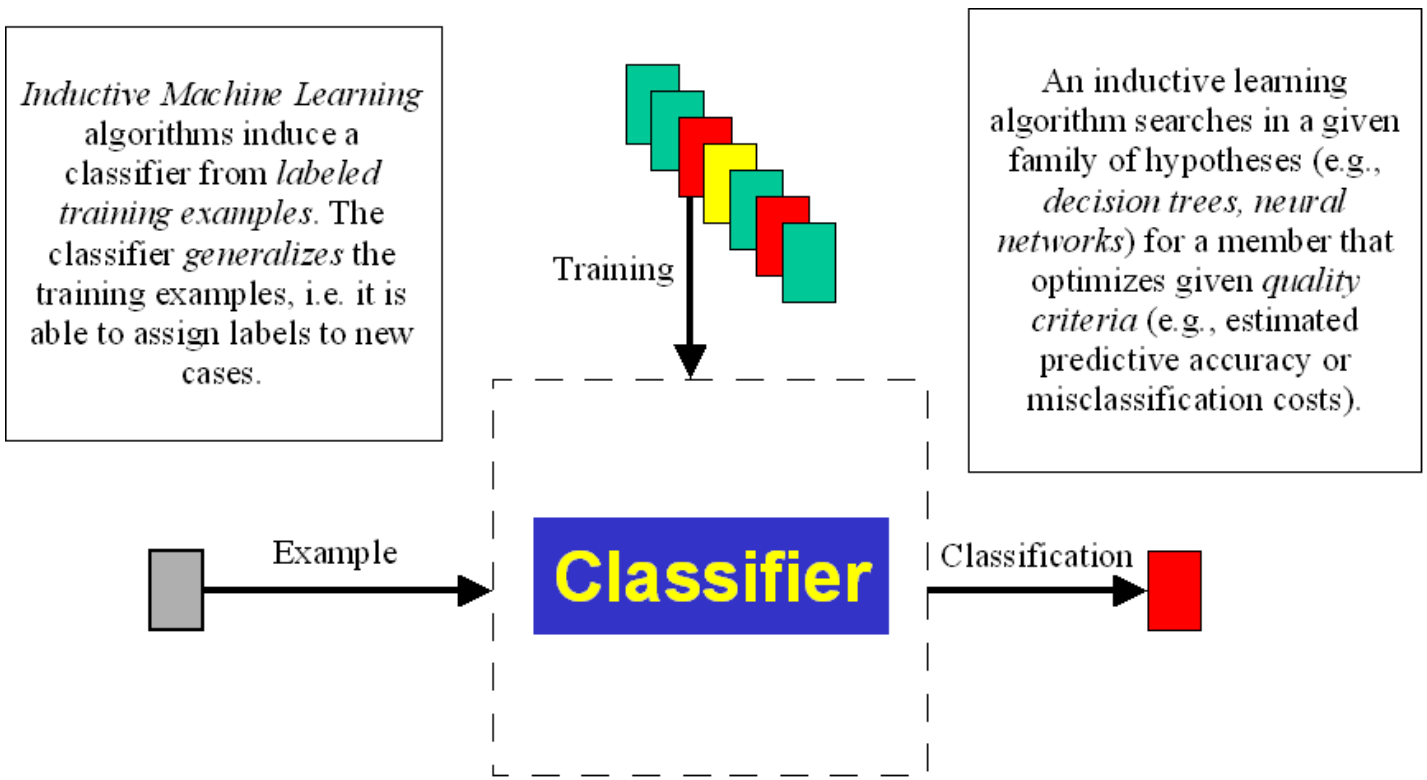
TECHNISCHE UNIVERSITÄT DARMSTADT

The most “popular” learning problem:

- Task:
  - learn a *model* that predicts the outcome of a dependent variable for a given instance
- Experience:
  - experience is given in the form of a data base of examples
  - an *example* describes a single previous observation
    - *instance:* a set of measurements that characterize a situation plus a label
    - *label:* the outcome that was observed in this situation (usually the number of labels (or classes) can be greater than 2, here, we most often assume binary classification
- Performance Measure:
  - compare the predicted outcome to the observed outcome
  - estimate the probability of predicting the right outcome in a new situation

Induction of Classifiers

TECHNISCHE UNIVERSITÄT DARMSTADT



Data Representation

Attribute-Value Data

TECHNISCHE UNIVERSITÄT DARMSTADT

- Each example (or instance) is described with values for a fixed number of *attributes* (also called *features*)
  - **Nominal Attributes:**
    - store an unordered list of symbols (e.g., *color*)
  - **Numeric Attributes:**
    - store a number (e.g., *income*)
  - **Other Types:**
    - ordered values
    - hierarchical attributes
    - set-valued attributes
- note that attribute values can also be missing (usually marked with a “?”)

A sample task

TECHNISCHE UNIVERSITÄT DARMSTADT

Day	Temperature	Outlook	Humidity	Windy	Play Golf?
07-05	26	sunny	high	false	no
07-06	28	sunny	high	true	no
07-07	29	overcast	high	false	yes
07-09	23	rain	normal	false	yes
07-10	20	overcast	normal	true	yes
07-12	12	sunny	high	false	no
07-14	8	sunny	normal	false	yes
07-15	25	rain	normal	false	yes
07-20	18	sunny	normal	true	yes
07-21	18	overcast	high	true	yes
07-22	20	overcast	normal	false	yes
07-23	19	rain	high	true	no
07-26	11	rain	normal	true	no
07-30	16	rain	high	false	yes

today	9	sunny	normal	false	?
tomorrow	13	sunny	normal	false	?

Concept Representation

TECHNISCHE UNIVERSITÄT DARMSTADT

- several different ways to represent the learned concept
  - interpretable models (i.e., decision trees or rule sets)
  - hyperplanes (SVMs or neural networks)
  - coefficients (weights) of a linear model
  - table of conditional probabilities (naive bayes)
  - none at all (lazy learning)
- 2 important criteria when searching for a good concept
  - Overfitting Avoidance
  - Occam's Razor
- two ways of representation:
  - learn a single model
  - learn many models and combine them (ensemble learning or meta learning)
    - *bagging*, *boosting*, *stacking*, ...
    - advantages: confidence in classification is higher due to the combination of many models
    - disadvantages: takes more time to learn and more storage space

Evaluation

Very brief introduction

TECHNISCHE UNIVERSITÄT DARMSTADT

- different ways to evaluate a learning algorithm
  - validation through expert
  - if test data is provided: learn model on training data, evaluate it on test data
  - if no test data is available
    - train-test split
    - $p \times k$ cross validation (p = number of passes, k = number of folds)

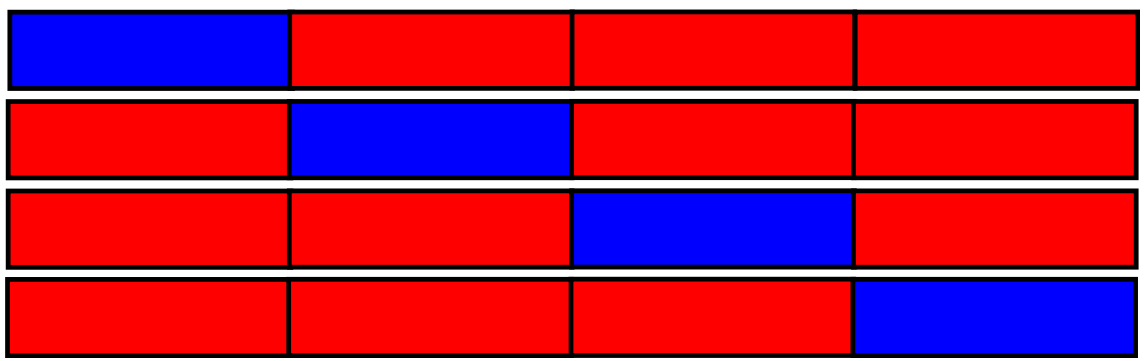
Evaluation

Very brief introduction

TECHNISCHE UNIVERSITÄT DARMSTADT

- different ways to evaluate a learning algorithm
  - validation through expert
  - if test data is provided: learn model on training data, evaluate it on test data
  - if no test data is available
    - train-test split
    - $p \times k$ cross validation (p = number of passes, k = number of folds)
    - example for a 1×4 cross validation (i.e., 4 folds)

testing folds training folds



Frameworks for Machine Learning

TECHNISCHE UNIVERSITÄT DARMSTADT

- many frameworks for machine learning are available, often open source
- still the most popular one: *weka* (`http://www.cs.waikato.ac.nz/ml/weka/`)
- nice alternative: *Rapid Miner* (`http://rapid-i.com/`)
- these 2 frameworks provide implementations of a variety of different algorithms
- but there are also many frameworks that are specialized for a certain task
  - *NTLK*: framework for natural language processing (`http://www.nltk.org/`)
  - *SVMlight*: implementation of a fast support vector machine (`http://svmlight.joachims.org/`)
  - *LibSVM* for *weka* (`http://www.csie.ntu.edu.tw/~cjlin/libsvm/`)
  - and many more...

Supervised Learning

Definition

TECHNISCHE UNIVERSITÄT DARMSTADT

- a teacher provides correct labels to **all** examples
- usually the last attribute encodes the class (or target) attribute
- different settings:
  - all examples are available (*batch learning*)
  - *iterative learning*: examples arrive one by one
- two ways to deal with labeled data:
  - *eager learning*: a model (concept) is learned on the labeled data
  - this model is used to classify unseen examples
  - *lazy learning*: no model is learned
  - when a new unseen training example should be classified, the classification is build directly on the training data

Decision Trees

Training and Classification

TECHNISCHE UNIVERSITÄT DARMSTADT

- a decision tree consists of
  - **nodes**: test for the value of a certain attribute
  - **edges**: the outcome of a test; connect to the next node or leaf
  - **leaves**: terminal nodes that are used to classify
- learned by a *divide-and-conquer* strategy
- often by using *entropy*
- $E(S) = -\frac{p}{p+n} \cdot log_2 \frac{p}{p+n} - \frac{n}{p+n} \cdot log_2 \frac{n}{p+n}$

where S is an attribute, p = number of positive examples, n = number of negative examples

Decision Trees

Training and Classification

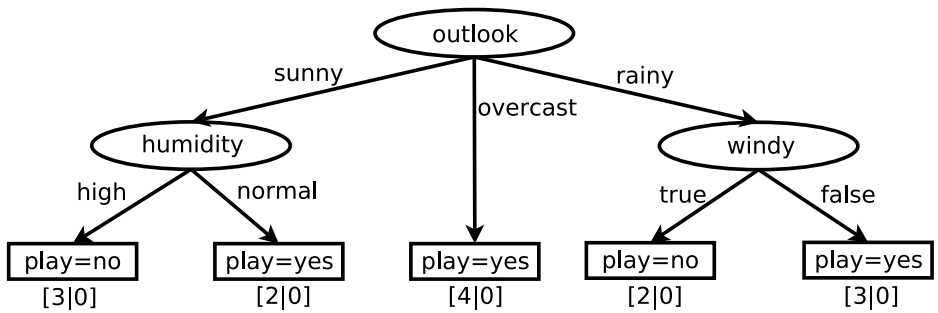
TECHNISCHE UNIVERSITÄT DARMSTADT

- a decision tree consists of
  - **nodes**: test for the value of a certain attribute
  - **edges**: the outcome of a test; connect to the next node or leaf
  - **leaves**: terminal nodes that are used to classify
- learned by a *divide-and-conquer* strategy
- often by using *entropy*
- $E(S) = -\frac{p}{p+n} \cdot log_2 \frac{p}{p+n} - \frac{n}{p+n} \cdot log_2 \frac{n}{p+n}$

where S is an attribute, p = number of positive examples, n = number of negative examples

Day	Temperature	Outlook	Humidity	Windy	Play Golf?
07-05	26	sunny	high	false	no
07-06	28	sunny	high	true	no
07-07	29	overcast	high	false	yes
07-09	23	rain	normal	false	yes
07-10	20	overcast	normal	true	yes
07-12	12	sunny	high	false	no
07-14	8	sunny	normal	false	yes
07-15	25	rain	normal	false	yes
07-20	18	sunny	normal	true	yes
07-21	18	overcast	high	true	yes
07-22	20	overcast	normal	false	yes
07-23	19	rain	high	true	no
07-26	11	rain	normal	true	no
07-30	16	rain	high	false	yes

today	9	sunny	normal	false	?
tomorrow	13	sunny	normal	false	?



Decision Trees

Advantages/Disadvantages/Algorithms

TECHNISCHE UNIVERSITÄT DARMSTADT

- Advantages
  - interpretable model
  - easy to understand
  - well-researched
- Disadvantages
  - limitations (e.g., `XOR` cannot be learned)
  - prone to overfitting (solution: *pruning* is necessary)
  - favor attributes with more levels (solution: use *intrinsic information*, i.e., how much information do we need to tell which branch an instance belongs to)
- Algorithms (excerpts)
  - ID3: early, solid implementation, **limitations**: cannot handle numeric attributes or missing values (Quinlan, 1983)
  - C4.5: state-of-the-art decision tree learner (also commercially used, see C5.0) (Quinlan, 1993)
  - RandomForests: ensemble technique (Breiman, 2001)

Rule Learning

Training and Heuristics

TECHNISCHE UNIVERSITÄT DARMSTADT

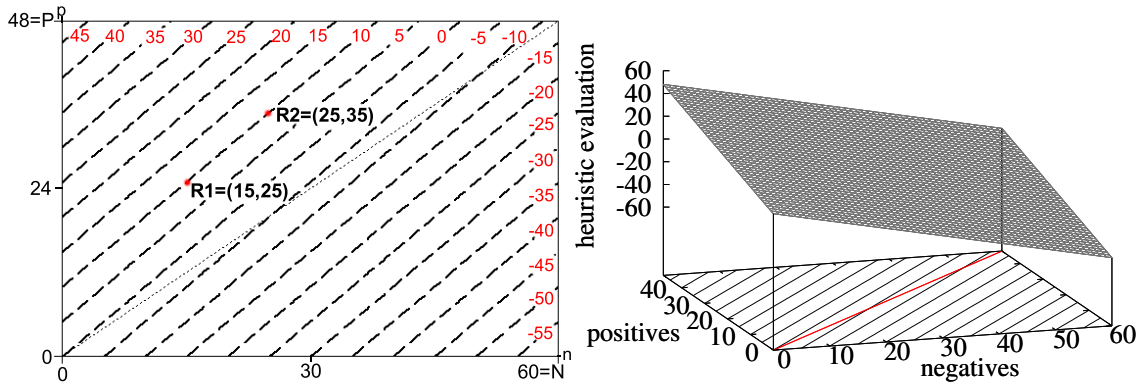
- most algorithms are based on the *separate-and-conquer* strategy:
  1. learn a “good” rule on the dataset (by iteratively adding conditions to it)
  2. add the rule to the ruleset and remove all the examples covered by the rule
  3. if (positive) examples are left `goto` 1.
- what is a “good” rule?
  - *heuristics* are used to evaluate the quality of the rule
  - there are many different heuristics that have different preferences, i.e., learn overfitting/underfitting rules, ...
  - the preference structure of a heuristic can be visualized in so-called *coverage spaces*

Rule Learning

Training and Heuristics

TECHNISCHE UNIVERSITÄT DARMSTADT

- most algorithms are based on the *separate-and-conquer* strategy:
  1. learn a “good” rule on the dataset (by iteratively adding conditions to it)
  2. add the rule to the ruleset and remove all the examples covered by the rule
  3. if (positive) examples are left `goto` 1.
- what is a “good” rule?
  - *heuristics* are used to evaluate the quality of the rule
  - there are many different heuristics that have different preferences, i.e., learn overfitting/underfitting rules, ...
  - the preference structure of a heuristic can be visualized in so-called *coverage spaces*



Rule Learning

Classification

TECHNISCHE UNIVERSITÄT DARMSTADT

- rule sets can be ordered (first rule that covers the example fires) or unordered (all covering rules are used for classification)
- such a rule set should be compact, but nevertheless with high accuracy

Rule Learning

Classification

TECHNISCHE UNIVERSITÄT DARMSTADT

- rule sets can be ordered (first rule that covers the example fires) or unordered (all covering rules are used for classification)
- such a rule set should be compact, but nevertheless with high accuracy

Day	Temperature	Outlook	Humidity	Windy	Play Golf?
07-05	26	sunny	high	false	no
07-06	28	sunny	high	true	no
07-07	29	overcast	high	false	yes
07-09	23	rain	normal	false	yes
07-10	20	overcast	normal	true	yes
07-12	12	sunny	high	false	no
07-14	8	sunny	normal	false	yes
07-15	25	rain	normal	false	yes
07-20	18	sunny	normal	true	yes
07-21	18	overcast	high	true	yes
07-22	20	overcast	normal	false	yes
07-23	19	rain	high	true	no
07-26	11	rain	normal	true	no
07-30	16	rain	high	false	yes

today	9	sunny	normal	false	?
tomorrow	13	sunny	normal	false	?

- sample rule set(s) (note that t = temperature):

r_1: play=no $\leftarrow$ t $\geq$ 25.5 $\wedge$ t $<$ 28.5
r_2: play=no $\leftarrow$ t $<$ 14 $\wedge$ t $\geq$ 9.5
r_3: play=no $\leftarrow$ outlook=rain $\wedge$ windy=true

Rule Learning

Classification

TECHNISCHE UNIVERSITÄT DARMSTADT

- rule sets can be ordered (first rule that covers the example fires) or unordered (all covering rules are used for classification)
- such a rule set should be compact, but nevertheless with high accuracy

Day	Temperature	Outlook	Humidity	Windy	Play Golf?
07-05	26	sunny	high	false	no
07-06	28	sunny	high	true	no
07-07	29	overcast	high	false	yes
07-09	23	rain	normal	false	yes
07-10	20	overcast	normal	true	yes
07-12	12	sunny	high	false	no
07-14	8	sunny	normal	false	yes
07-15	25	rain	normal	false	yes
07-20	18	sunny	normal	true	yes
07-21	18	overcast	high	true	yes
07-22	20	overcast	normal	false	yes
07-23	19	rain	high	true	no
07-26	11	rain	normal	true	no
07-30	16	rain	high	false	yes

today	9	sunny	normal	false	?
tomorrow	13	sunny	normal	false	?

- sample rule set(s) (note that t = temperature):

r_1: play=no $\leftarrow$ t $\geq$ 25.5 $\wedge$ t $<$ 28.5
r_2: play=no $\leftarrow$ t $<$ 14 $\wedge$ t $\geq$ 9.5
r_3: play=no $\leftarrow$ outlook=rain $\wedge$ windy=true

- but also:

r_1: play=no $\leftarrow$ t $<$ 26.5 $\wedge$ t $\geq$ 25.5 $\wedge$
outlook=sunny $\wedge$ humidity=high $\wedge$ windy=false
r_2: play=no $\leftarrow$ t $<$ 28.5 $\wedge$ t $\geq$ 27.5 $\wedge$
outlook=sunny $\wedge$ humidity=high $\wedge$ windy=true

Rule Learning

Classification

TECHNISCHE UNIVERSITÄT DARMSTADT

- rule sets can be ordered (first rule that covers the example fires) or unordered (all covering rules are used for classification)
- such a rule set should be compact, but nevertheless with high accuracy

Day	Temperature	Outlook	Humidity	Windy	Play Golf?
07-05	26	sunny	high	false	no
07-06	28	sunny	high	true	no
07-07	29	overcast	high	false	yes
07-09	23	rain	normal	false	yes
07-10	20	overcast	normal	true	yes
07-12	12	sunny	high	false	no
07-14	8	sunny	normal	false	yes
07-15	25	rain	normal	false	yes
07-20	18	sunny	normal	true	yes
07-21	18	overcast	high	true	yes
07-22	20	overcast	normal	false	yes
07-23	19	rain	high	true	no
07-26	11	rain	normal	true	no
07-30	16	rain	high	false	yes

today	9	sunny	normal	false	?
tomorrow	13	sunny	normal	false	?

- sample rule set(s) (note that t = temperature):

  r_1: play=no $\leftarrow$ t $\geq$ 25.5 $\wedge$ t $<$ 28.5
  r_2: play=no $\leftarrow$ t $<$ 14 $\wedge$ t $\geq$ 9.5
  r_3: play=no $\leftarrow$ outlook=rain $\wedge$ windy=true

- but also:

  r_1: play=no $\leftarrow$ t $<$ 26.5 $\wedge$ t $\geq$ 25.5 $\wedge$
  outlook=sunny $\wedge$ humidity=high $\wedge$ windy=false
  r_2: play=no $\leftarrow$ t $<$ 28.5 $\wedge$ t $\geq$ 27.5 $\wedge$
  outlook=sunny $\wedge$ humidity=high $\wedge$ windy=true

- or even:

  r_1: play=yes $\leftarrow$ `true`

Rule Learning

Advantages/Disadvantages/Algorithms

TECHNISCHE UNIVERSITÄT DARMSTADT

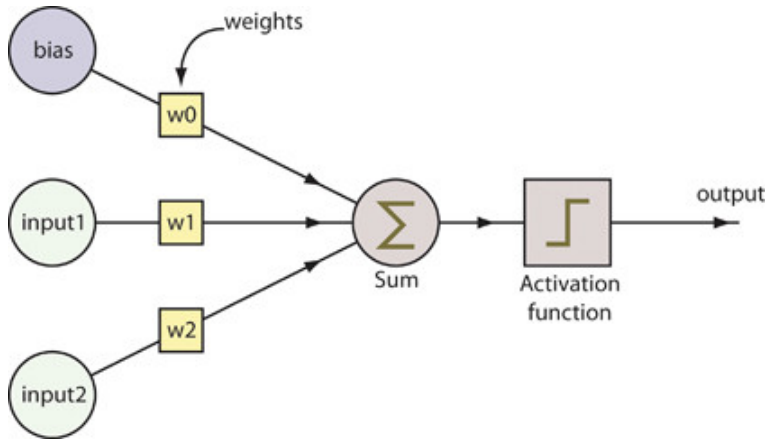
- Advantages
  - interpretable model
  - easy to understand
  - well-researched
  - higher expressiveness than decision trees (rules may overlap)
- Disadvantages
  - limitations (e.g., `XOR` cannot be learned)
  - prone to overfitting (solution: *pruning* is necessary)
- Algorithms
  - AQ: first implementation of a *separate-and-conquer* algorithm (Michalski, 1969)
  - CN2: solid implementation (Clark and Niblett, 1989)
  - Ripper: additional adaptions to dataset, state-of-the-art algorithm (Cohen, 1995)

Neural Networks

Definition

TECHNISCHE UNIVERSITÄT DARMSTADT

- neural networks are based on how the human brain works
- simplest version: perceptron



- input: arbitrary number of input signals
- output: $y = \begin{cases} 1 & \text{if } \sum_{i=1}^{n} w_i \cdot input_i + w_o \cdot bias > 0 \\ -1 & \text{otherwise} \end{cases}$

note: graphic was borrowed from `http://mos.futurenet.com/techradar/Review%20images/PC%20Plus/PCP%20288/PCP288.theory.figure2-420-90.jpg`

Neural Networks

Definition (continued)

TECHNISCHE UNIVERSITÄT DARMSTADT

- neural network (*multilayer perceptron*) is formed by connecting many perceptrons
- usually there is an input and an output layer (note that there can be more than one output nodes)
- in between is the hidden layer
- hidden layer is useful to enable the multilayer perceptron to learn more complex function, i.e.,
  - every continuous function can be learned with three layers (one hiden layer)
  - every function can be learned with four layers (two hidden layers)

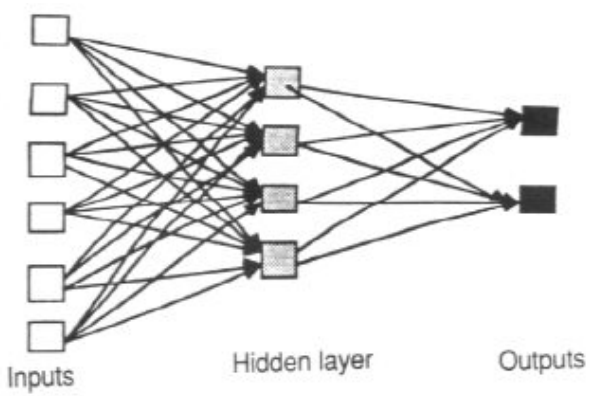


Figure from `http://www.doc.ic.ac.uk/~nd/surprise_96/journal/vol4/cs11/report.html`

Neural Networks

Training

TECHNISCHE UNIVERSITÄT DARMSTADT

- training is adjusting the weights w_i n times, where n is the number of training epochs
- algorithm: *backpropagation*
  - requires that the activation function is differentiable (due to gradient descent)

1. forward propagation of one training example through the network
2. computation of the error
3. backward propagation of the target information in order to adjust the weights of each neuron towards the correct value
4. **loop** as long as number of epochs is not reached (or as long as error is not low enough)

Advantages/Disadvantages

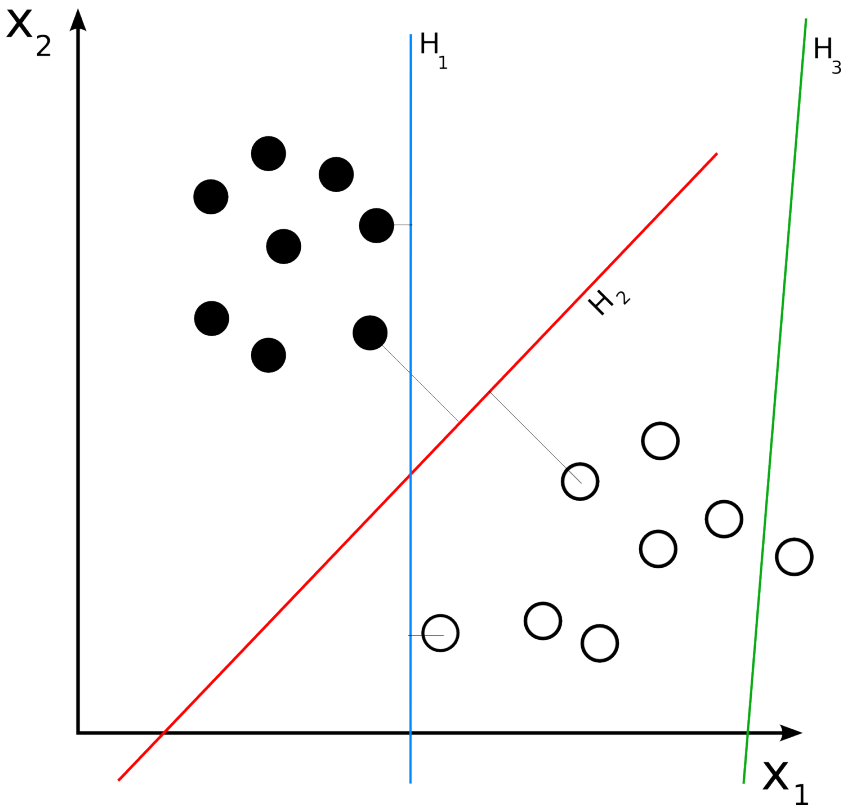
- Advantages
  - can also be used for unsupervised learning or reinforcement learning
  - not tailored to a specific problem and indeed applied successfully in many domains (e.g., flying an aircraft, learning to play backgammon, etc.)
- Disadvantages
  - usually require a large diversity of training data (at least in a real-world scenario)
  - most often complex networks require huge amount of storage space (due to all the connections and neurons)
  - hard to design, i.e., how many hidden layers, etc.
  - limited in the interpretability (only the weights can be interpreted)

Support Vector Machines

Idea

TECHNISCHE UNIVERSITÄT DARMSTADT

- idea: calculate a *hyperplane* that has the maximum margin to the positive and negative class



- H_3 does not separate the classes
- H_1 separates the classes but with a small margin
- H_2 separates the classes with maximum margin

Figure taken from wikipedia (`http://en.wikipedia.org/wiki/Support_vector_machine`)

Support Vector Machines

Properties

TECHNISCHE UNIVERSITÄT DARMSTADT

- types of *SVMs*
  - large margin svm
  - soft margin svm
  - multiclass svm
  - transductive svm (for semi-supervised learning)
  - regression svm (for solving regression)
- if the examples are not linear separable, the svm is able to use a so-called *kernel* to overcome this problem
  - a *kernel* projects the data into a high-dimensional space so that the data can be separated again
  - this is done implicitly
  - many kernels are available for different tasks

Support Vector Machines

Advantages/Disadvantages/Algorithms

TECHNISCHE UNIVERSITÄT DARMSTADT

- Advantages
  - one of the most accurate predictors (at least when the parameters are optimized)
  - kernels make them very flexible
  - unique optimal solution (convex optimization problem)
- Disadvantages
  - models are not interpretable
  - some older versions are very slow
- Algorithms
  - LibSVM: fast svm implementation for *weka* (Chang and Lin, 2011)
  - SVMlight: fast svm implementation in a standalone framework (from T. Joachims, `http://svmlight.joachims.org/`)
  - many implementations available (overview: `http://www.support-vector-machines.org/SVM_soft.html`)

Naive Bayes

Training and Classification

TECHNISCHE UNIVERSITÄT DARMSTADT

- idea: combining knowledge of the training data (*a priori* probabilities) with observed data
- Bayes' theorem: $P(h|D) = \frac{P(D|h) \cdot P(h)}{P(D)}$
  where $P(h)$ = *a priori* probability of hypothesis h, $P(D)$ = *a priori* probability of training data D
- since $P(h|D)$ is not known, but $P(D|h)$ can be estimated, the Bayes' theorem is the essential idea of NaiveBayes
- algorithm:
  1. for each class value v_j: $P(v_j) \leftarrow$ estimate $P(v_j)$
  2. for each attribute value a_i of each attribute a: $P(a_i|v_j) \leftarrow$ estimate $P(a_i|v_j)$
- output is a table of conditional probabilities
- naive bayes classification: $h_{NB} = \arg\max_{v_j \in V} P(v_j) \prod_i P(a_i|v_j)$

Naive Bayes

Training and Classification (continued)

TECHNISCHE UNIVERSITÄT DARMSTADT

Day	Temperature	Outlook	Humidity	Windy	Play Golf?
07-05	26	sunny	high	false	no
07-06	28	sunny	high	true	no
07-07	29	overcast	high	false	yes
07-09	23	rain	normal	false	yes
07-10	20	overcast	normal	true	yes
07-12	12	sunny	high	false	no
07-14	8	sunny	normal	false	yes
07-15	25	rain	normal	false	yes
07-20	18	sunny	normal	true	yes
07-21	18	overcast	high	true	yes
07-22	20	overcast	normal	false	yes
07-23	19	rain	high	true	no
07-26	11	rain	normal	true	no
07-30	16	rain	high	false	yes

today	9	sunny	normal	false	?
tomorrow	13	sunny	normal	false	?

- exemplary estimation (training) for attribute `outlook`

	outlook		
	sunny	overcast	rain
yes	2/9	4/9	3/9
no	3/5	0	2/5

- classification only with attribute `outlook`, e.g., *outlook=sunny*
- yes=$\frac{2}{9} \cdot \frac{9}{14} = \frac{2}{14} < \frac{3}{14} = \frac{3}{5} \cdot \frac{5}{14}$=no $\rightarrow$ no

Naive Bayes

Training and Classification (continued)

TECHNISCHE UNIVERSITÄT DARMSTADT

Day	Temperature	Outlook	Humidity	Windy	Play Golf?
07-05	26	sunny	high	false	no
07-06	28	sunny	high	true	no
07-07	29	overcast	high	false	yes
07-09	23	rain	normal	false	yes
07-10	20	overcast	normal	true	yes
07-12	12	sunny	high	false	no
07-14	8	sunny	normal	false	yes
07-15	25	rain	normal	false	yes
07-20	18	sunny	normal	true	yes
07-21	18	overcast	high	true	yes
07-22	20	overcast	normal	false	yes
07-23	19	rain	high	true	no
07-26	11	rain	normal	true	no
07-30	16	rain	high	false	yes

today	9	sunny	normal	false	?
tomorrow	13	sunny	normal	false	?

- exemplary estimation (training) for attribute `outlook`

	outlook		
	sunny	overcast	rain
yes	2/9	4/9	3/9
no	3/5	0	2/5

- classification only with attribute `outlook`, e.g., *outlook=sunny*
- yes=$\frac{2}{9} \cdot \frac{9}{14} = \frac{2}{14} < \frac{3}{14} = \frac{3}{5} \cdot \frac{5}{14}$=no $\rightarrow$ no

- Advantages: fast training, fast classification, surprisingly good performance
- Disadvantages: assumes that the features are conditionally independent (over-simplified assumption)

Nearest Neighbor

Classification

TECHNISCHE UNIVERSITÄT DARMSTADT

- no model is build, i.e., *lazy learning*
- define a distance measure on all attributes
- compute the distance between the new example and all other ones and use the k nearest neighbors for classification
- example for distance of numerical attributes: $dist(v_i, v_j) = \frac{|v_i - v_j|}{\text{max value in training data}}$
- example for distance of nominal attributes: $dist(v_i, v_j) = \begin{cases} 0 & \text{if } v_i = v_j \\ 1 & \text{otherwise} \end{cases}$
- global distance metric necessary
- choice of k is important:
  - too small: influence of close examples is reduced
  - too big: all examples influence the decision, also very distant ones

Unsupervised Classification

Definition

TECHNISCHE UNIVERSITÄT DARMSTADT

- no information of the target class of the examples is given
- algorithms try to find regularities in the data
- somewhat hard to evaluate
- related to density estimation in statistics

Clustering

Overview

TECHNISCHE UNIVERSITÄT DARMSTADT

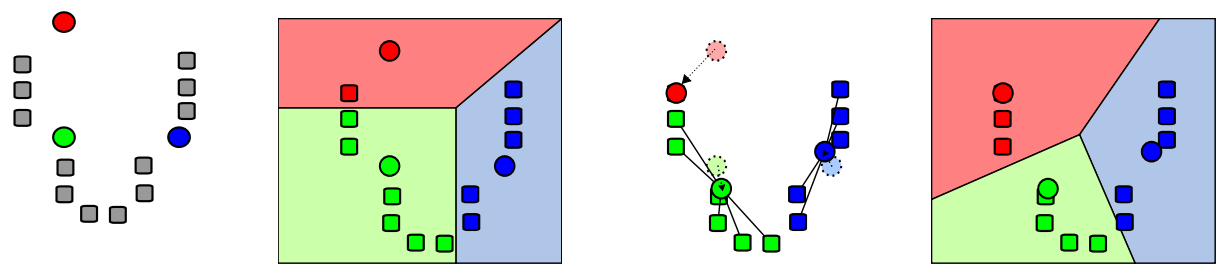
- different methods, among them are *hierarchical clustering*, *centroid-based clustering*, *distribution-based clustering*, *density-based clustering*
- task is to cluster similar examples together
- given the definition of similarity, one of the above methods should be chosen
- all of them have different objectives, i.e., different definitions of similarity
- example algorithm: k-means (k = number of clusters)
  1. randomly select k centroids
  2. associate every data point with the nearest mean
  3. compute new centroids (with new clusters)
  4. **repeat** steps 2 and 3 until convergence

Clustering

Overview

TECHNISCHE UNIVERSITÄT DARMSTADT

- different methods, among them are *hierarchical clustering*, *centroid-based clustering*, *distribution-based clustering*, *density-based clustering*
- task is to cluster similar examples together
- given the definition of similarity, one of the above methods should be chosen
- all of them have different objectives, i.e., different definitions of similarity
- example algorithm: k-means (k = number of clusters)
  1. randomly select k centroids
  2. associate every data point with the nearest mean
  3. compute new centroids (with new clusters)
  4. **repeat** steps 2 and 3 until convergence



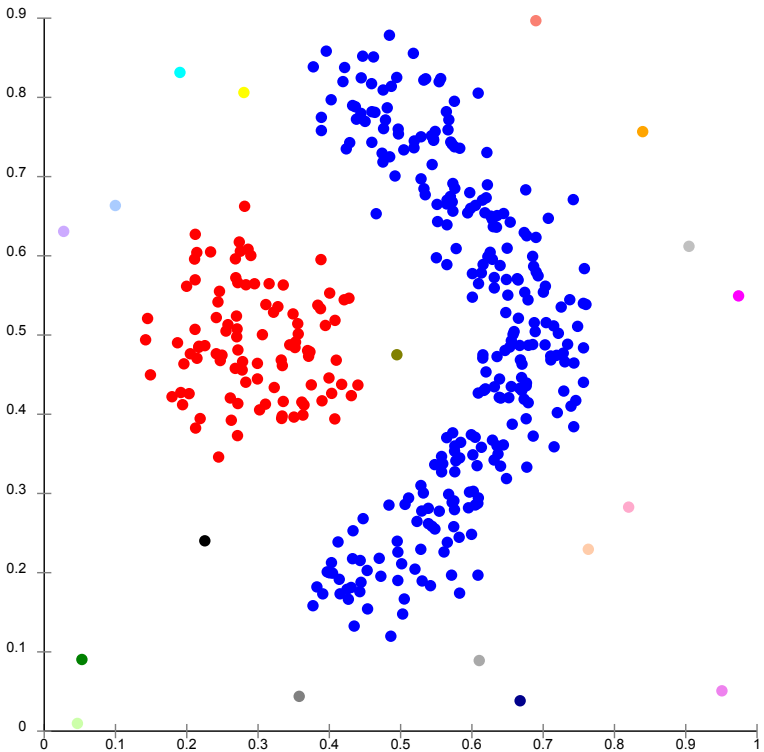
Figures are taken from wikipedi (`http://en.wikipedia.org/wiki/K-means`)

Clustering

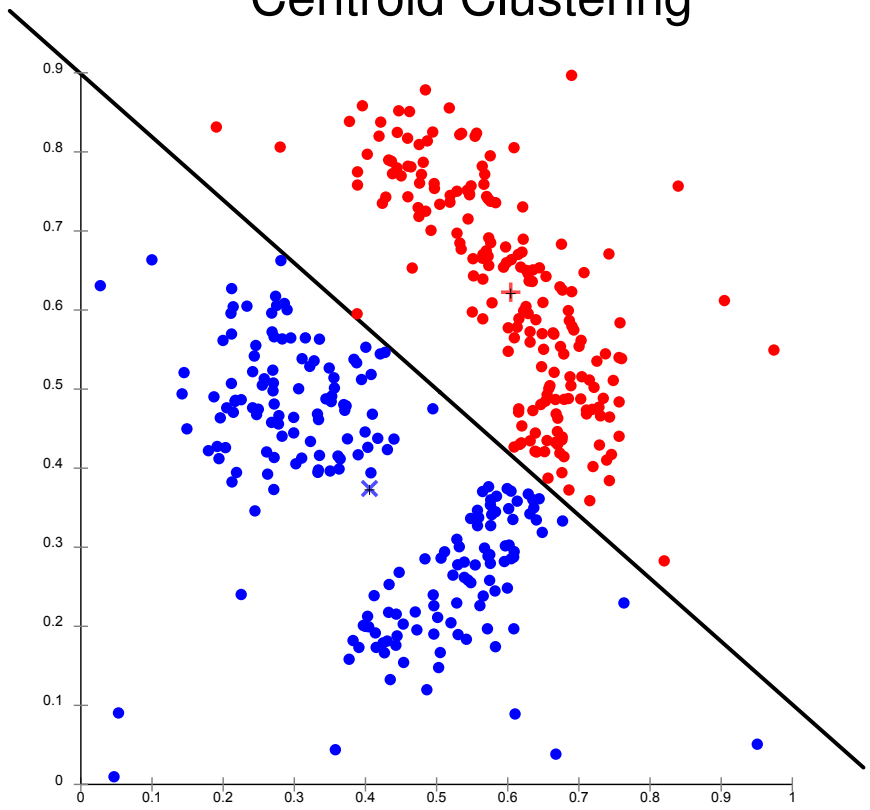
Examples

TECHNISCHE UNIVERSITÄT DARMSTADT

Hierarchical Clustering



Centroid Clustering



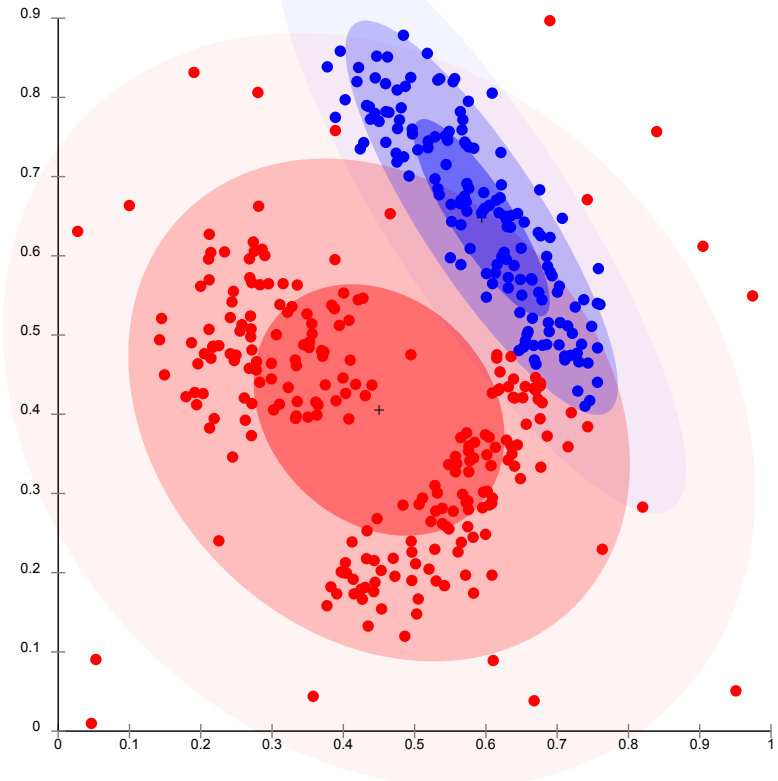
figures taken from wikipedia (`http://en.wikipedia.org/wiki/Data_clustering`)

Clustering

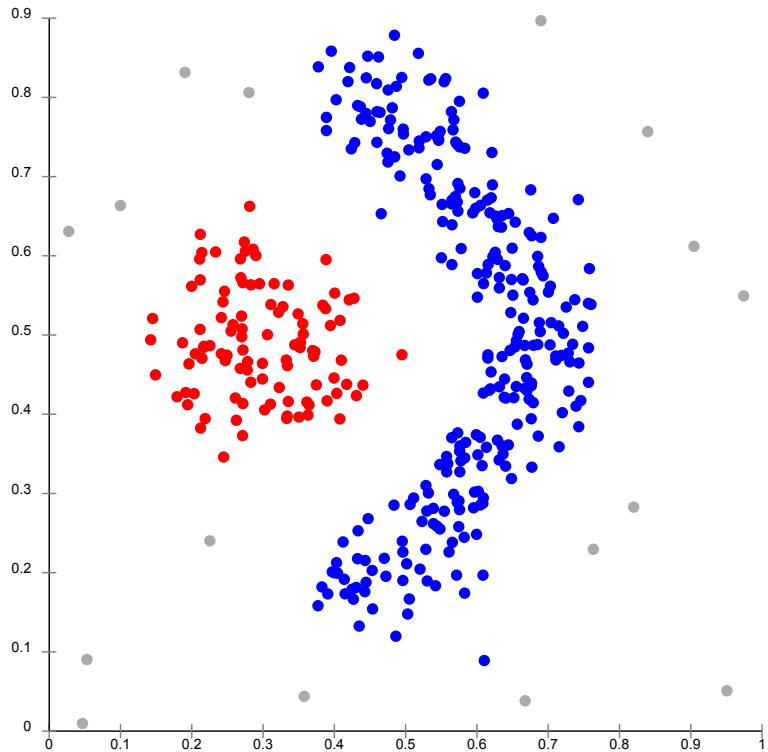
Examples (continued)

TECHNISCHE UNIVERSITÄT DARMSTADT

Distribution-based Clustering



Density-based Clustering



figures taken from wikipedia (`http://en.wikipedia.org/wiki/Data_clustering`)

Association Rule Learning

Overview

TECHNISCHE UNIVERSITÄT DARMSTADT

- association rule mining is used to find relations in huge databases
- origin is market basket analysis
- most famous finding: `diapers` $\rightarrow$ `beer`
- often used to recommend items
- but also to place certain items in a supermarket, i.e., such that items that are frequently bought together are also placed close together
- most famous algorithm: Apriori
- related to rule learning, but here the conclusion can have multiple items (not only one)

Association Rule Learning

Apriori

TECHNISCHE UNIVERSITÄT DARMSTADT

- quality measures
  - **support**: proportion of examples that are covered
  - **confidence**: how strong is the implication of the rule
  - example: `Bread`, `Cheese` $\rightarrow$ `RedWine` ($S = 0.01, C = 0.8$)
    - 80% of all customers that bought bread and cheese also bought red wine. 1% of all customers bought all three items together.
- algorithm:
  - create frequent itemsets, i.e., sets of items that have a minimum support
  - create association rules from the frequent itemsets that have a minimum confidence
- properties
  - efficient generation of frequent itemsets and association rules (alphabetical order and *anti-monotonicity* of confidence)
  - graphical interpretation by *Search space* and the *positive and negative border*
  - other measures than confidence possible: *lift* and *leverage* (interestingness measures)

Subgroup Discovery

TECHNISCHE UNIVERSITÄT DARMSTADT

- can also be supervised
- is also used to discover interesting relations in huge databases
- is expressed by rules, where the conclusion is a single item
- subgroups do not to be complete, i.e., a subgroup covering only a small part of the data is also possible

Semi-supervised Learning

TECHNISCHE UNIVERSITÄT DARMSTADT

- in semi-supervised learning only a part of the training data is labeled (typically only a small amount is labeled)
- the learner has access to all data
- different techniques:
  - self-training: the learning algorithms learns a model on the labeled data and uses this model to classify the unlabeled examples, then the model is re-learned
  - active learning: the learner suggests unlabeled examples that can then be labeled by an expert/a teacher, only those examples are requested to be labeled that are important for the algorithm, i.e., from which the learner is able to improve the model
  - co-training: two classifiers are learned on two different views on the data (ideally conditionally independent), the most confident predictions on the unlabeled data are then used to label more and more training data
    - note that this is a very strong technique: in the original publication 95% of 788 web pages were labeled correctly with the use of only 12 labeled pages (Blum and Mitchell, 1998)

Regression

TECHNISCHE UNIVERSITÄT DARMSTADT

- contrary to classification in regression the target variable is numeric (therefore also called learning of functions)
- different ways to deal with this situation:
  1. either by discretizing the target variable and use standard classification algorithms, or
  2. by working on the numeric targets directly
- main disadvantage of the first case: we do not know the right number of classes in advance
- main disadvantage of the latter: we have to adapt the algorithms

Regression

Algorithms

TECHNISCHE UNIVERSITÄT DARMSTADT

- most algorithms known from classification also have a regression version (SVMs, rule learning algorithms, decision trees, neural networks)
- the measures change: measures known from classification (e.g., accuracy) cannot be used any more
- statistical measures, i.e., loss or error functions (e.g., *mean squared error* or *mean absolute error*)
- often solved by handling it as an optimization problem (e.g., *least angle* or *gradient boosting*
- the most popular algorithm of the statistics community is *linear regression* which has some nice properties:
  - very fast in training
  - accurate prediction
  - at least to some extent interpretable, i.e., coefficients or weights of attributes

- also called *online learning*, i.e., examples arrive one by one
- the model is refined with each new example
- important observations:
  - when should old examples be excluded from the current model (*concept drift*)
  - when should all examples be skipped and a completely new model should be learned (*concept shift*)
- algorithms:
  - *windowing*: only keep a window of the examples
  - ID5R: decision tree learner for iterative learning, refines the decision tree with each example by rotating it (to keep it balanced)

Multi-Label

TECHNISCHE UNIVERSITÄT DARMSTADT

- the target can have many variables at the same time, but not all of them are true
- application:
  - categorization of newspaper articles, i.e., one article can have many categories (e.g., actor, politics, sports for an article about Arnold Schwarzenegger)
  - genres of a movie
- algorithms
  - ML-*k*-NN: a nearest neighbor classifier that is able to deal with multilabel data, i.e., also search the nearest neighbors but then use some procedure to assign the actual labels
  - *binary relevance*, i.e., decompose the target classes so that we have binary problems again

Multi-Target

TECHNISCHE UNIVERSITÄT DARMSTADT

- contrary to supervised learning and multi-label learning, in multi-target each example has many labels that all have to be predicted at the same time
- thus, dependencies between the different targets can be utilized
- can be also numeric targets, i.e., multi-target regression
- not many algorithms proposed so far
- notable exception:
  - (Ženko and Džeroski, 2008): a framework for learning classification rules for multiple target attributes is presented

Inductive Logic Programming

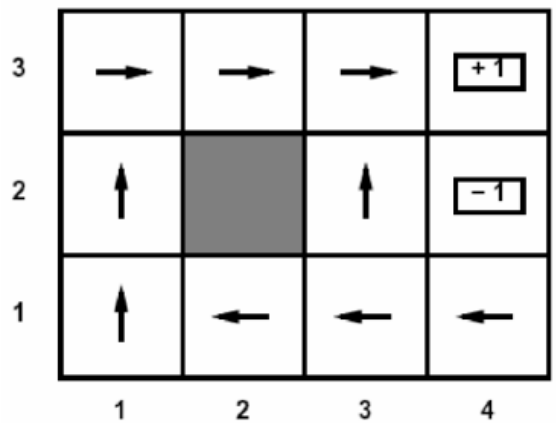
TECHNISCHE UNIVERSITÄT DARMSTADT

- rules in first order logic are learned
- the input is *background knowledge* and positive/negative examples
- rules predict a concept based on the background knowledge given the positive/negative examples of the concept
- well-researched technique
- algorithms:
  - Foil: first version of a first-order inductive learner (learns *datalog* relations) (Quinlan, 1990)
  - Progol: improved implementation (efficient search, etc.) (Muggleton, 1995)
- example of a first order rule:
  - `father(A,B) :- parent(A,B), male(A).`
  - background knowledge: `parent(christopher,arthur)`, `male(christopher)`
  - positive/negative examples: $\oplus$: `father(christopher,arthur)`, $\ominus$: `father(penelope,arthur)`

Reinforcement Learning

TECHNISCHE UNIVERSITÄT DARMSTADT

- given:
  - a learning agent
  - S: a set of possible states
  - A: a set of possible actions
  - a state transition function $\delta : S \times A \rightarrow S$
  - a reward function $r : S \times A \rightarrow \Re$
- goal: learn an optimal policy for an agent in a given environment
- successfully applied to learning tic-tac-toe and backgammon (strength of a world champion)



Discussion

TECHNISCHE UNIVERSITÄT DARMSTADT

- a very brief and general overview of machine learning
- introduction of various scenarios
- brief introduction of different algorithms
- (biased) evaluation of learning algorithms, i.e., advantages/disadvantages

References

TECHNISCHE UNIVERSITÄT DARMSTADT

- (Blum and Mitchell, 1998): A. Blum and T. Mitchell. Combining labeled and unlabeled data with co-training. *COLT: Proceedings of the Workshop on Computational Learning Theory*, Morgan Kaufmann, pages 92-100, 1998.
- (Ženko and Džeroski, 2008): B. Ženko and S. Džeroski. Learning Classification Rules for Multiple Target Attributes. *PAKDD: Advances in Knowledge Discovery and Data Mining, 12th Pacific-Asia Conference*, Osaka, Japan, May 20-23, pages 454-465, 2008.
- (Quinlan, 1983): J.R. Quinlan. Learning Efficient Classification Procedures and their Application to Chess End Games. *Machine Learning. An Artificial Intelligence Approach*, pages 463-482, 1983.
- (Quinlan, 1993): J.R. Quinlan. In R. S. Michalski, J. G. Carbonell, and T. M. Mitchell, editors, *Programs for Machine Learning*. Morgan Kaufmann Publishers, pages 463-482, 1993.
- (Breiman, 2001): L. Breiman. Random Forests. *Machine Learning* 45(1):5-32, 2001.
- (Michalski, 1969): R.S. Michalski. On the Quasi-Minimal Solution of the Covering Problem. *FCIP: Proceedings of the 5th International Symposium on Information Processing*, volume A3, pages 128-128, 1969.
- (Clark and Niblett, 1989): P. Clark and T. Niblett. The CN2 Induction Algorithm. *Machine Learning*, 3(4):261-283, 1989.
- (Cohen, 1995): W. W. Cohen. Fast Effective Rule Induction. In *ICML: roceedings of the 12th International Conference on Machine Learning*, pages 115-123, 1995.
- (Chang and Lin, 2011): C.-C. Chang and C.-J. Lin. LIBSVM : a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology* 2(3):1-27, 2011. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`
- (Quinlan, 1990): J. R. Quinlan. Learning Logical Definitions from Relations. *Machine Learning*, 5:239-266, 1990.
- (Muggleton, 1995): S. Muggleton. Inverse Entailment and Progol. *New Generation Computing Journal*, Vol. 13, pages 245-286, 1995.