.

Machine Learning

Introduction

Prof. Dr. Martin Riedmiller
AG Maschinelles Lernen und Natürlichsprachliche Systeme
Institut für Informatik
Technische Fakultät
Albert-Ludwigs-Universität Freiburg

riedmiller@informatik.uni-freiburg.de

Acknowledgment Slides were adapted from slides provided by
Tom Mitchell, Carnegie-Mellon-University
and Peter Geibel, University of Osnabrück

Organisational issues

Prof. Dr. Martin Riedmiller
Martin.Riedmiller@informatik.uni-freiburg.de
Office hours: Tuesday, 14.00 - 15.00 Uhr

Slides are available online

Course language changes every other semester

SoSe 2012 Lecture in German (slides in english)

Overview of Today's Lecture: Introduction

1. Motivation
2. Types of machine learning tasks
3. Representing the knowledge: ML models
4. Adapting the knowledge: learning algorithms
5. A definition of machine learning
6. Overview of lecture

Motivation

- Learning characterizes the process of growing up (and the whole life)

- Baby starts to say "'Mummy"'
- Baby learns to grasp blocks.
- Baby stops trying to eat everything.
- Baby learns to build towers from blocks.

- Learnig is a characteristic and prerequisite for intelligent behaviour

Motivation

- Articificial intelligence (AI) aims at builing machines that behave intelligently ...
- ... and tries to understand the processes behind it.
- Machine learning (ML) is one of its major subfields.

(Human) types of learning:

- Learning from a teacher ML: Supervised Learning
- Learning from experience ML: Reinforcement learning
- Discovery (learn structures in empirical data) ML: Unsupervised Learning
- Meta-learning: learning to learn

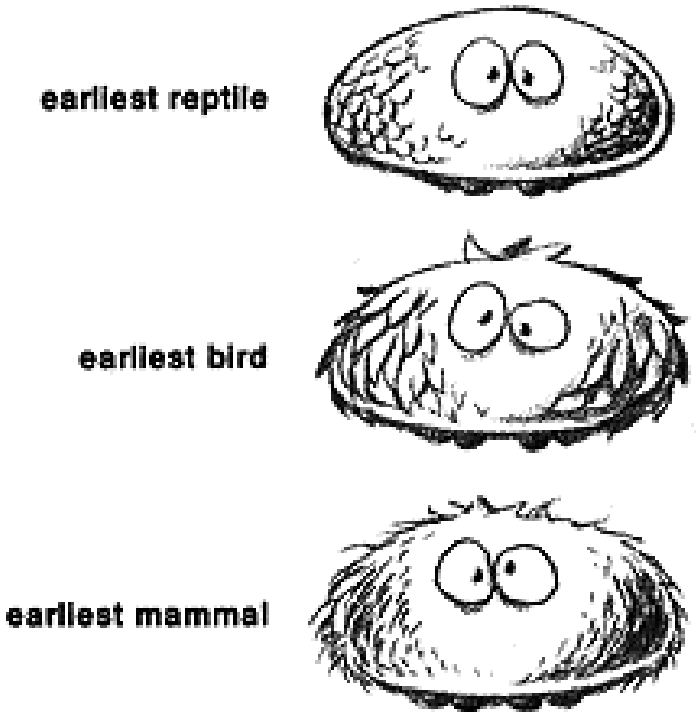
Citations

Changes in the system that are adaptive in the sense that they enable the system to do the same task or tasks drawn from the same population more efficiently next time. -Herb Simon

If an expert system–brilliantly designed, engineered and implemented–cannot learn not to repeat its mistakes, it is not as intelligent as a worm or a sea anemone or a kitten.
-Oliver G. Selfridge, from The Gardens of Learning.

"Find a bug in a program, and fix it, and the program will work today. Show the program how to find and fix a bug, and the program will work forever."
- Oliver G. Selfridge, in AI's Greatest Trends and Controversies

Aspects



- Learning does not only take place at the individual level
- Whole species and societes can evolve and adapt
- $\rightarrow$ different paradigms

Why Machine Learning

- Recent progress in algorithms and theory
- Growing flood of online data
- Computational power is available
- Budding industry

Three niches for machine learning:

- Data mining : using historical data to improve decisions
  - medical records $\rightarrow$ medical knowledge
- Software applications we can't program by hand
  - autonomous driving
  - speech recognition
- Self customizing programs
  - Newsreader that learns user interests

Relevant Disciplines

- Artificial intelligence
- Computational complexity theory
- Control theory
- Information theory
- Statistics
- Bayesian methods
- Philosophy
- Psychology and neurobiology
- ...

Applications

- Credit risk management
- Speech recognition
- Text Mining
- Face recognition
- Image recognition
- Bioinformatics
- Elevator group control
- Playing backgammon
- Robots playing soccer
- Autonomous cars

Credit Risk

- A private customer comes to a bank and applies for a loan.
- Should the credit be given to him?
- Decision support for the bank employee is needed!
- He enters description of customer and credit into system:
  - Income, age etc.
  - Credit amount etc.
- System judges the credit risk of the customer
  - Classification: good or bad customer
  - Regression: prediction of gain/loss for bank

Basic Idea

1. Use the past experience of the bank (or of other banks) to automatically construct a decision support program.
2. Machine learning terminology:
   - Past experience = training set
   - Automatically construct = learn, induce
   - Decision support program = hypothesis, classifier
3. Task: find function, i.e. learn classifier,

$$f : X_1 \times \ldots \times X_n \rightarrow \{good, bad\}$$

   from I/O examples that has low error
4. X_i is domain of feature/attribute/variable x_i.
5. This type of learning is supervised learning: classification

German Credit Data Set

Number of Instances: 1000

Number of Attributes: 20 (7 numerical, 13 categorical)

Attribute x_3:
Credit history

0 : no credits taken/
all credits paid back duly
1 : all credits at this bank paid back duly
2 : existing credits paid back duly till now
3 : delay in paying off in the past
4 : critical account/
other credits existing (not at this bank)

I.e. $X_3 = \{0, \ldots, 4\}$

German Credit Data Set

1. Other attributes:
   - Duration:real number
   - Purpose (e.g. car, house, holiday)
   - Credit amount
   - Present employment
   - Personal status and sex
   - Age in years

2. Type of attributes
   - Categorical (e.g. color $\in$ {red, green, blue})
   - Ordinal (e.g. size$\in$ {small, middle, large})
   - Continuous (e.g. size $\in \mathbb{R}$)

3. + discrete, numerical, real-valued, nominal, structured, complex

Training Set

- Every case is encoded by a sequence of numbers (= row)
- The classes are given with the examples (last column):
  - 1: good customer
  - 2: bad customer
- $\rightarrow$ Training set:

```
11 6 34 43 1169 65 75 4 93 101 4 121 67 143 152 2 173 1 192 201 1
12 48 32 43 5951 61 73 2 92 101 2 121 22 143 152 1 173 1 191 201 2
14 12 34 46 2096 61 74 2 93 101 3 121 49 143 152 1 172 2 191 201 1
11 42 32 42 7882 61 74 2 93 103 4 122 45 143 153 1 173 2 191 201 1
11 24 33 40 4870 61 73 3 93 101 4 124 53 143 153 2 173 2 191 201 2
14 36 32 46 9055 65 73 2 93 101 4 124 35 143 153 1 172 2 192 201 1
14 24 32 42 2835 63 75 3 93 101 4 122 53 143 152 1 173 1 191 201 1
12 36 32 41 6948 61 73 2 93 101 2 123 35 143 151 1 174 1 192 201 1
14 12 32 43 3059 64 74 2 91 101 4 121 61 143 152 1 172 1 191 201 1
12 30 34 40 5234 61 71 4 94 101 2 123 28 143 152 2 174 1 191 201 2
12 12 32 40 1295 61 72 3 92 101 1 123 25 143 151 1 173 1 191 201 2
```

Aspects of the Problem

- Up until now, we've spoken about the data
- How shall we represent the knowledge? Which kind of ML model should we use?
  - Symbolic: decision trees and rules
  - Subsymbolic: neural networks
  - Prototypes and similarities
  - Statistical decision rules
  - etc.
- And how shall we construct the data from the training set?

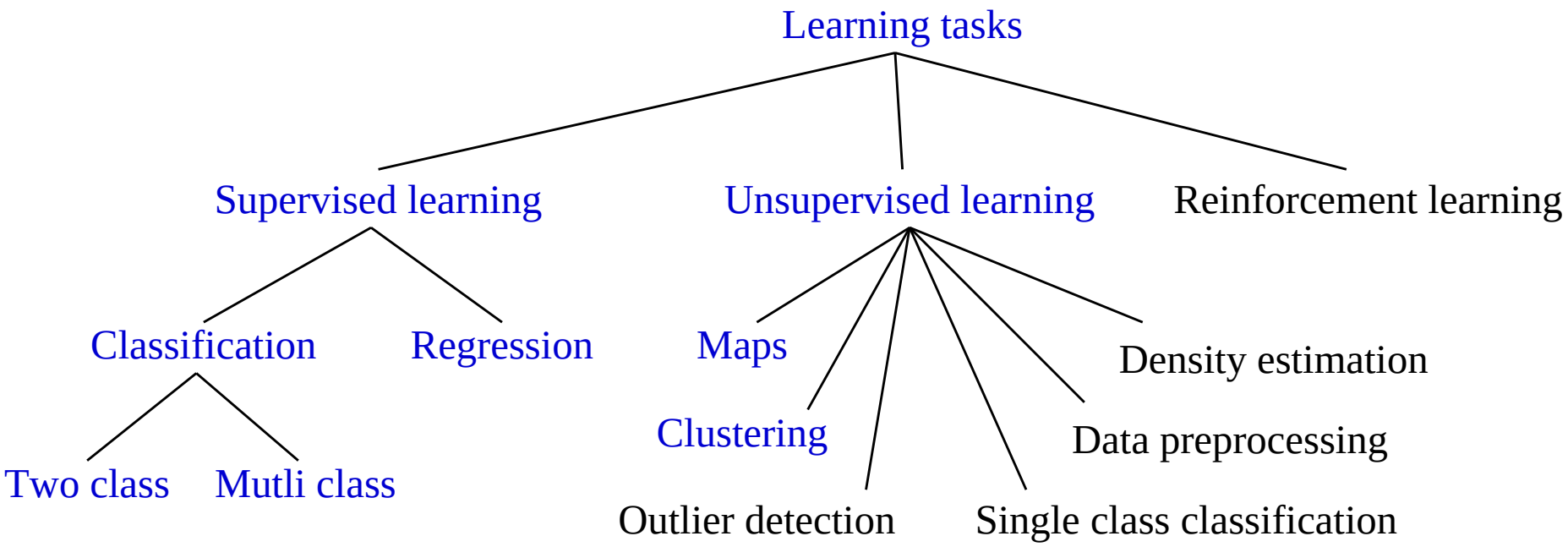
Characterizing the Field

Throughout the lecture, we will try to distinguish between the following 3 different levels of machine learning problems:

- 3 main types of ML tasks: supervised, unsupervised and reinforcement learning
- many differnet types of models to represent the knowledge: decision trees, neural networks, statistical models, parametrized functions, adaptive rules, ...
- ML methods and algorithms: from 'tailored to the model' to 'based on general principles' (like e.g. minimizing the training error by mathematical optimisation techniques)

Remark:
Of course, the levels are highly interleaved: sometimes, a ML task type directly implies the model to be used; some models can only be trained with specialized training algorithms, some solution methods for a certain ML task may use other types of learning tasks as subtasks (e.g. RL uses supervised learning).

Types of learning tasks



Types of learning tasks: Unsupervised Learning

- training data consist of a description of situations/ objects/ ...
- learn what is typical/ similar/ interesting/ strange within the data
- no teacher
- example: clustering of things that 'somehow belong together': clustering
  e.g. distinguish plants from animals
- example: detetect objects, that are different/ surprising from others (novelty detection)
  e.g. find suspicious outliers in radioactivity measurements

Types of learning tasks: Supervised Learning

- teacher provides examples of a situation and a desired ouput ('teacher' in an abstract sense; might also be past observations)
- learn functional relation between situations and desired output
- two subtypes: classification and regression

Classification:

- desired ouput is taken from a small set of possibilities
- patterns are partinioned into classes, partinitioning given by desired output
- example: digit recognition

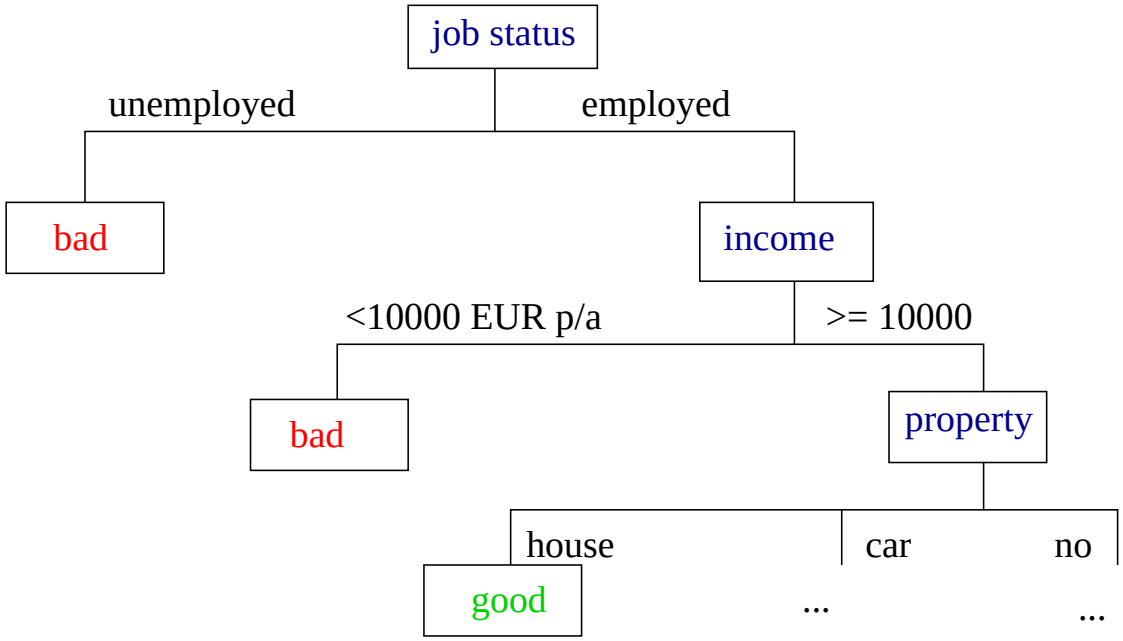
Regression:

- desired ouput is a real value
- a mathematical function is learned
- example: weather forcast (maximal temperatures)

Types of learning tasks: Reinforcement learning

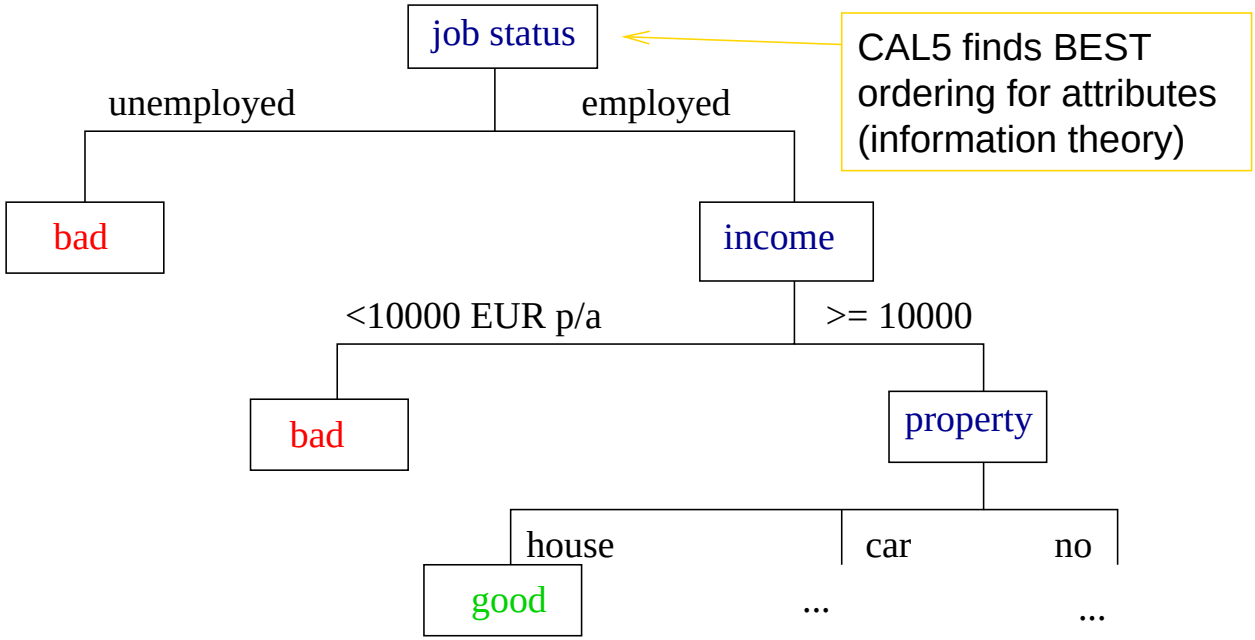
- training data consist of sequences of situations, actions and reward
- task is: learn a control strategy to maximize the reward
- no teacher
- (famous) example: learn to balance a pole

ML Models (1): Decision Trees



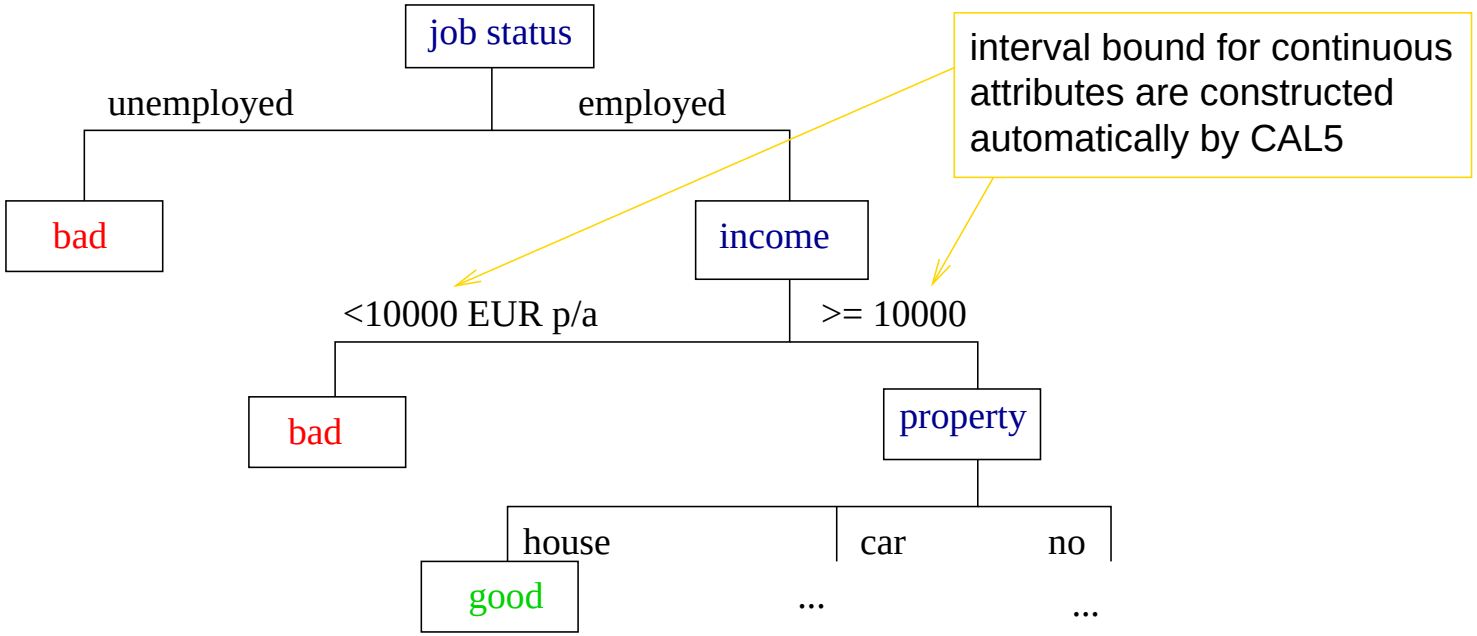
- Inner nodes labeled with attributes (tests)
- Branches labeled with conditions or outcomes
- Leaves labeled with classes

ML Models(1): Decision Trees



- Which attributes are to be used in what order?

ML Models (1): Decision Trees

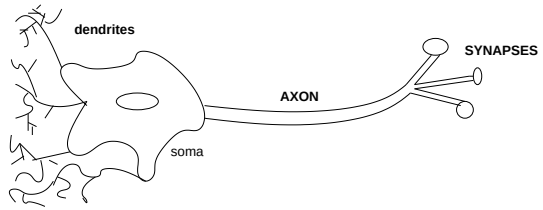


- How to deal with numerical attributes?

ML Models (1): Decision Trees

- Decision trees are a symbolic representation of the classifier
- Problems to be solved:
  - Tree construction
  - Sequence of the attributes
  - Intervals for continuous attributes

ML Models (2): Biological Neurons

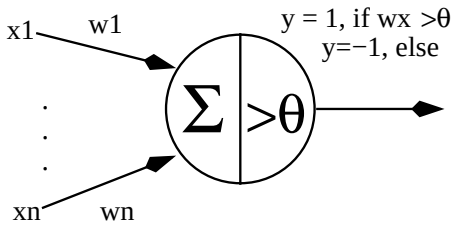


- Dentrites deliver input information to the cell
- Neuron fires (has action potential) if a certain threshold of activation is exceeded
- Output of information by axon
- The axon is connected to dentrites of other cells via synapses
- Learning corresponds to adaptation of the efficiency of synapse, of the synaptical weight

ML Models (2): Biological Neurons

- The human brain has approximately 10^{11} neurons
- Connections per neuron: $10^4 - 10^5$
- Switching time $0.001s$ (computer $\approx 10^{-10}s$)
- $0.1s$ for face recognition
- I.e. at most 100 computation steps
- $\rightarrow$ parallelism
- Additionally: robustness

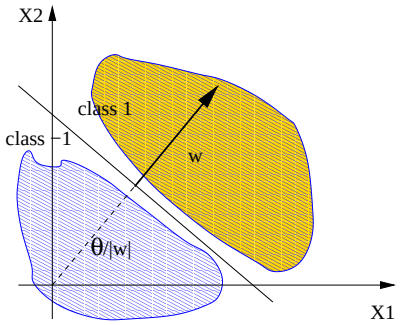
ML Models (2): Simplified Neuron Model



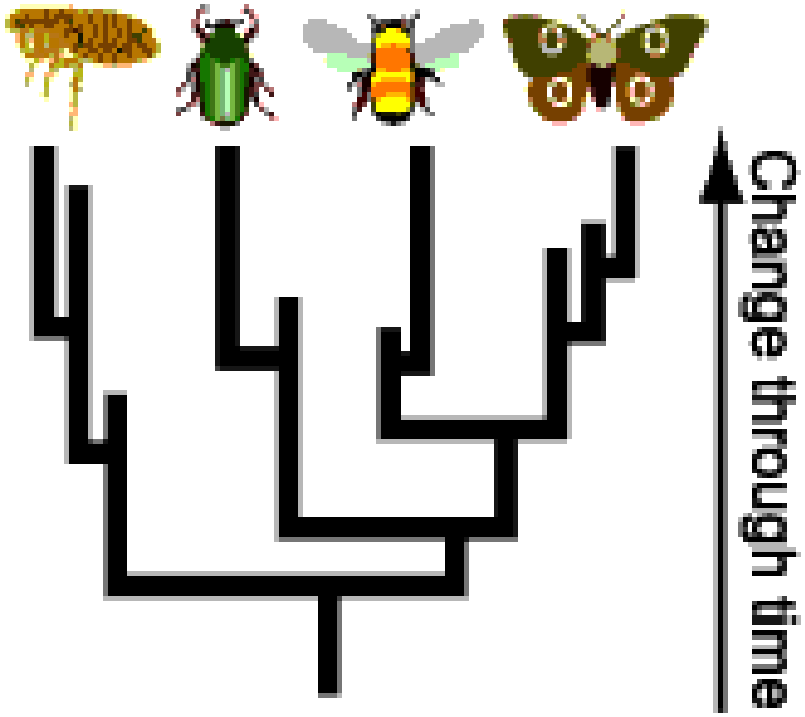
- Compute weighted input

$$w_1 x_1 + \ldots + w_n x_n$$

- Compare it to threshold $\theta \rightarrow$ Neuron fires or not
- Learning means finding the right weights!
- Can be interpreted as hyperplane in feature space!



ML Methods: Biological Evolution



- Species "transmute" over time
- Consistent, heritable variation among individuals in population
- Natural selection of the fittest

ML Methods: Evoluationary Computation

1. Computational procedures patterned after biological evolution

2. Hypotheses (individuals) are encoded by bit strings (genetic algorithms) or programs (genetic programming)

   - A population of hypotheses is maintained
   - Random mutation and crossover
   - fitness function

3. Method of stochastic search:

   - Evolution is known to be a robust and efficient method for adaptation
   - Fitness function can define complex behaviour
   - Can easily be parallelized

Example use in ML: Modify a behaviour, until a given task can be accomplished (e.g. using a teacher or a reward signal).

Important to ML: the idea of generalization

difference between $learning\ approaches$ and a $database$:

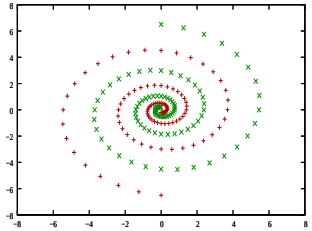
- a database can store data perfectly, a ML model need not.
- a database can recall stored data perfectly, a ML model need not.
- a database cannot make a guess for a situation that has not been stored, a ML model is expected to make a 'reasonable' guess
- a database is not tolerant with respect to noisy data, a ML model is expected to interpret noise as noise and not as information
- a database cannot deal with probabilistic features, a ML model is expected to

Generalization:

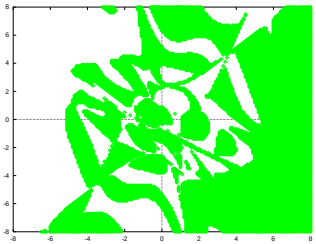
the ability to make reasonable guesses for situations that are unknown

The spiral problem: example of generalization

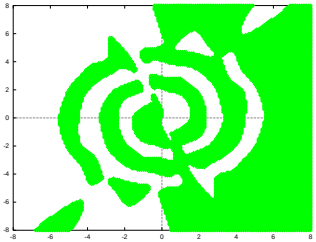
- 2-dim input patterns are arranged in two classes, representing two interwining spirals



- naive approach: training examples are perfectly met, but low generalization



- improved training method: training examples are perfectly met and generalization is good



What is the Learning Problem?

Learning = Improving with experience at some task

- Improve over task T,
- With respect to performance measure P,
- Based on experience E.

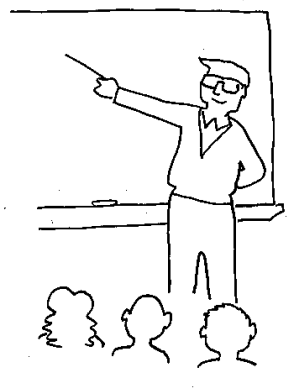
E.g., Learn to play Backgammon

- T: play Backgammon
- P: % of games won in world tournament
- E: opportunity to play against itself

Supervised Learning

Classification learning:

- Task T: predict class k for vector of features
- Performance measure P: number of errors
- Experience E: a training set with class labeled examples



Supervised learning or learning with teacher

ML example: Embedding learning components in large software systems
The Brainstormers in RoboCup

The Brainstormers (est. 1998)

RoboCup: 'In 2050 a team of robots shall win against the human world champion in soccer'

Annual competitions in different leagues

Brainstormers' longterm goal: Create a software agent that learns to play soccer by success or failure (Reinforcement Learning, RL)

- development of new RL methods: efficient, robust, scalable, multi-agent
- integration into a software system
- be competitive!

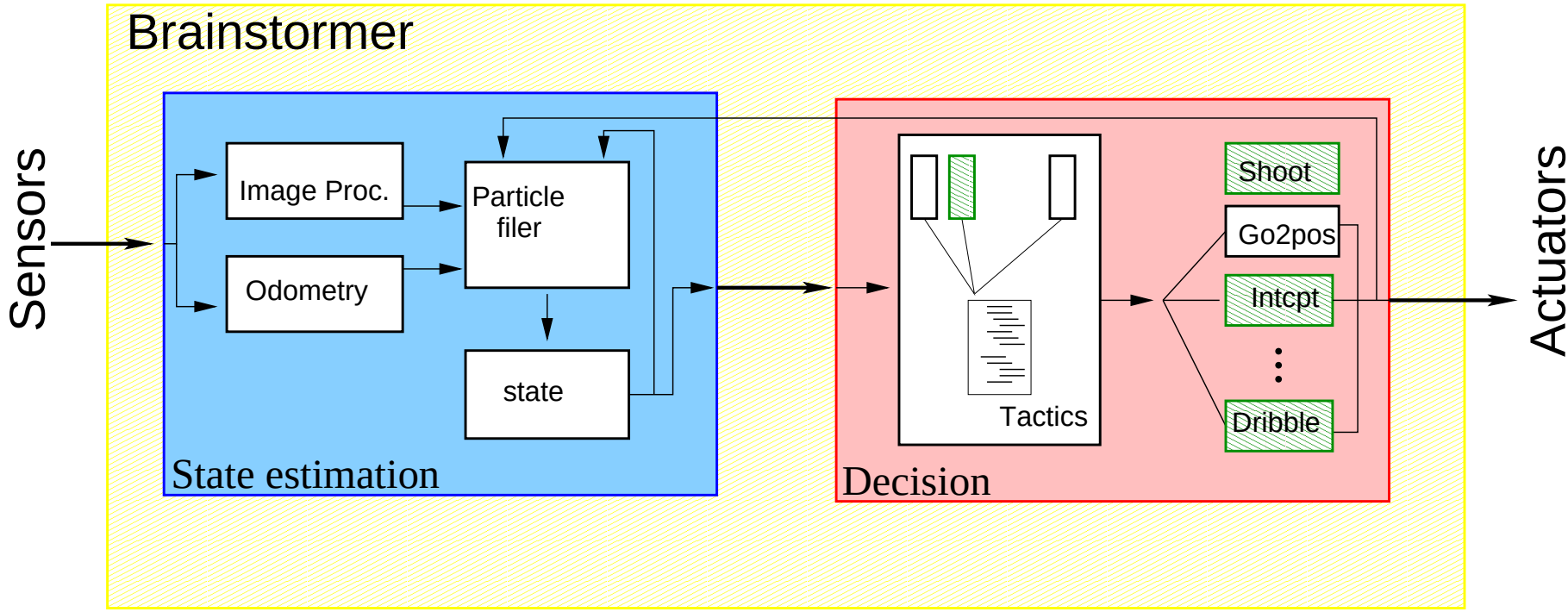
Software Design Decisions

1. formulate decision task as a Markov Decision Process
2. highly modular decision making unit to allow learning of partial capabilities

Requirement (1) is realised in a 'world modelling module' that integrates sensors, history and action information to estimate state. In the MidSize robot, also a prediction module is implemented, in order to cope with sensor and actuator delays.

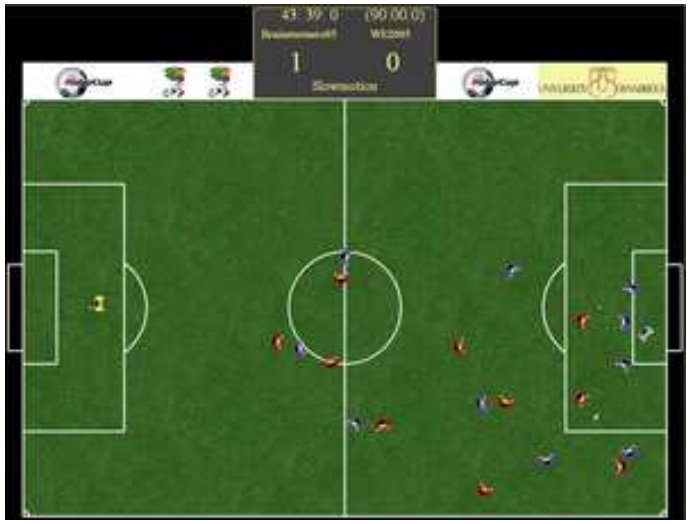
Requirement (2) is realized by building up the decision unit of 'behaviour' modules, that all have the same interface. Implementation of a module can be either done by learning or hand-coding (or both). Behaviour modules might have internal states to preserve their intentions over multiple time steps.

Decisions are made on a regular time step (30Hz Midsize and 10 Hz Simulation)


Brainstormer
Sensors
Image Proc.
Odometry
Particle filer
state
State estimation
Tactics
Shoot
Go2pos
Intcpt
Dribble
Decision
Actuators

Learning of basic skills (simulation league)

- many actions (100-500), continuous state spaces (3 - 7 dimensions)
- RL algorithm: fitted value iteration, greedy sampling
- value function: neural MLP, trained by Rprop
- learned behaviours able to beat 'state-of-the-art' hand-coded solutions are applied in the competition team: neural kicking, intercepting, positioning (demo)



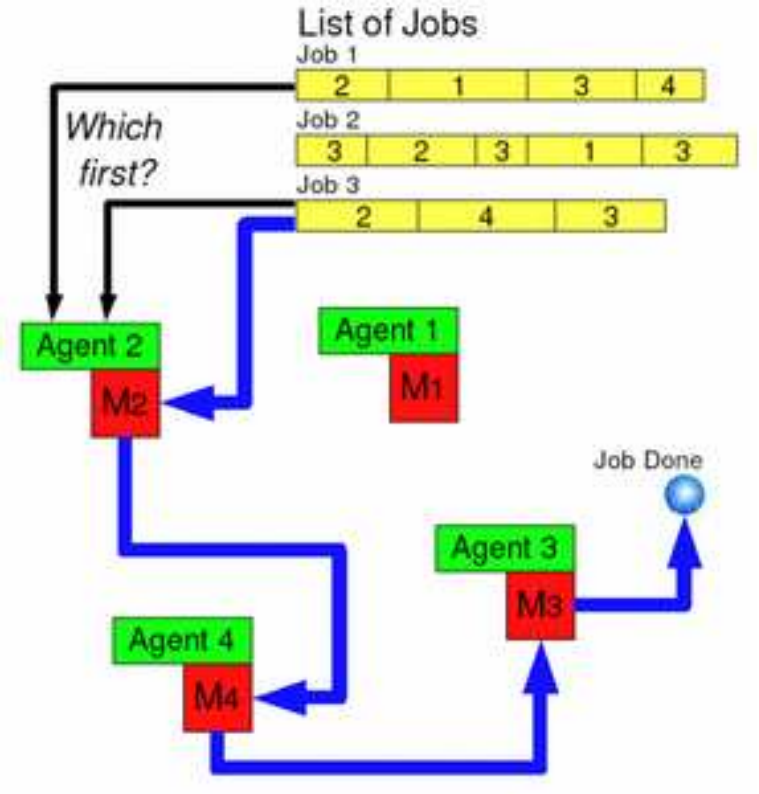
Cooperative RL for Attack Cooperation (simulation league)

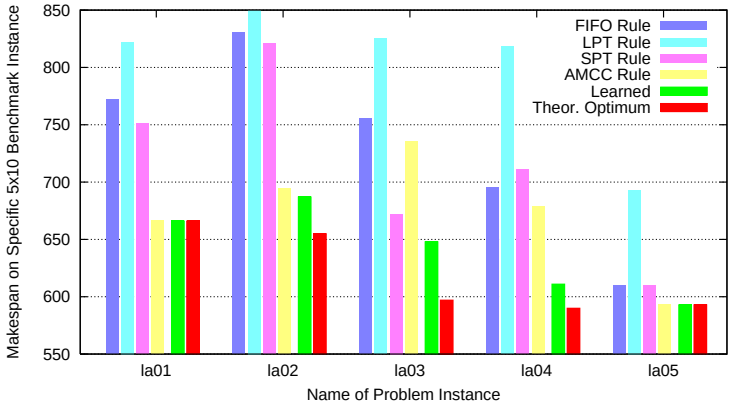
- forced to cooperate by shared training signal $r(\cdot)$: scored a goal/ lost ball/ else
- actions are basic 'skills': intercept, go to a position, pass, ...
- RL algorithm: batch learning of joint value function based on roll-outs
- use of partial and approximate models
- pre-selection of actions

demo

Spinoff: RL in Multi-Agent Systems

Scheduling (DFG-Project; 2005 - 2009)





- theoretical Background (Lauer, Riedmiller, ICML 01, AAMAS 04, ESANN 06)
- Benchmarking to classical algorithms - close to optimum

Learning on the Real MidSize Robot

- fully autonomous
- omnidirectional drive
- 3 independent DC motors
- noise, nonlinearities

Learning to Dribble

- RL algorithm: Neural fitted Q iteration (NFQ), random sampling
- needed less than 100 episodes to train behaviour directly on real robot (no model, no prior policy)
- learned behaviour beats hand-coded policy

demo

More than a proof of concept...

Learned modules used in our competition team (simulation league)

	2000	2001	2002	2003	2004	2005	2006	2007
NKick	●	●	●	●	●	●	●	●
NeuroIntercept	●	●	●	●		○		
NeuroGo2Pos	●	●	●	●	●			
NeuroDribble	○							
NeuroHoldBall	●	●						
NeuroHassle								●
NeuroAttack2vs2	○							
NeuroPos7vs8		●						
NeuroAttack3vs4			○					
NeuroAttack7vs8			●	●	●		●	●
NeuroPenalty1vs1				●	●	●	●	
NeurobScore					●			

Are learning agents competitive?

- from 2000 to 2009 always among the best 3 in international competitions
- Winning many European Championships in Simulationleague and MidSize from 2004 to 2009
- World Champion Simulationleague 2005, 2007, 2008; Middle Size 2006 and 2007
- Technical Challenge Awards 2006, 2007 and 2008

Goal of Lecture

Given a real world task, you should be able to

- select the attack points for applying ML methods
- decide for each subproblem to which types of ML tasks it can be mapped
- select a suitable model
- select a learning algorithm
- validate the trained model

Overview of Lecture

A selection of ...

- Introduction
- Version Spaces
- Decision Tree Learning
- Artificial Neural Networks
- SVM and Kernel Methods
- Basic Probability and Statistics
- Bayesian Approaches
- Reinforcement Learning
- Evolutionary Algorithms
- Case Based Reasoning
- Unsupervised Learning/Clustering

Literature

- Tom M. Mitchell (1997): Machine Learning, McGraw Hill, 1997.
- Ethem Alpaydin (2004): Introduction to Machine Learning, MIT Press.
- C. M. Bishop (2006): Pattern Recognition and Machine Learning.

Additional references:

- C. M. Bishop: Neural Networks for Pattern Recognition, Oxford University Press.
- R. S. Sutton, A. G. Barto: Reinforcement Learning – An Introduction, MIT Press, 1998, (HTML version Online)
- N. Lavrac, S. Dzeroski: Inductive Logic Programming, Ellis Horwood, 1994 (available online).
- B. Schölkopf, A. Smola: Learning with Kernels. MIT Press, 2001.