Computer Vision group heidelberg

IWR Interdisciplinary Center for Scientific Computing

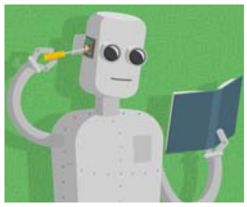
RUPRECHT-KARLS-UNIVERSITÄT HEIDELBERG

Artificial Intelligence

Prof. Björn Ommer
HCI & IWR

Outline – Learning

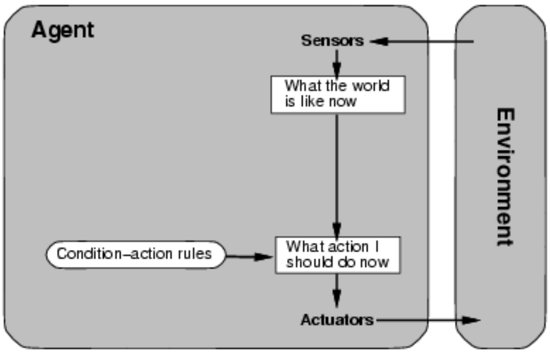
- **Learning agents**
- **Types of learning**
- **Ingredients to learning**
- Next chapters:
  - Artificial Neural Networks
  - CNNs
- Deep Learning of CNNs:
- Bayes Decision Theory
- Discriminant Functions
- Loss Functions & Classifiers
- Optimization ...



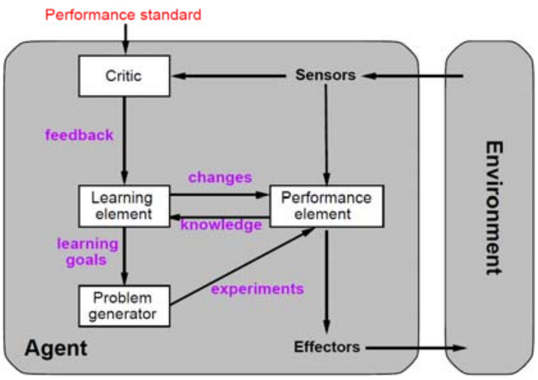
Learning

- **Machine Learning:**
  - **Principles, methods, and algorithms for learning and prediction on the basis of past evidence**
- **Goal: Machines that learn to perform a task from experience**
- **Learning is essential for unknown environments, i.e., when designer lacks omniscience**
- **Learning is useful as a system construction method, i.e., expose the agent to reality rather than trying to write it down**
- **Learning modifies the agent's decision mechanisms to improve performance**

Simple reflex agents



Learning Agents



Learning element

Design of learning element is dictated by
- ◇ what type of performance element is used
- ◇ which functional component is to be learned
- ◇ how that functional compoent is represented
- ◇ what kind of feedback is available

Example scenarios:

Performance element	Component	Representation	Feedback
Alpha–beta search	Eval. fn.	Weighted linear function	Win/loss
Logical agent	Transition model	Successor–state axioms	Outcome
Utility–based agent	Transition model	Dynamic Bayes net	Outcome
Simple reflex agent	Percept–action fn	Neural net	Correct action

Supervised learning: correct answers for each instance
Reinforcement learning: occasional rewards

Inductive learning (a.k.a. science)

Simplest form: learn a function from examples (tabula rasa)

f is the target function

An example is a pair x, $f(x)$, e.g., $\begin{array}{c|c|c} O & O & X \\ \hline & X & \\ \hline X & & \end{array}$, $+1$

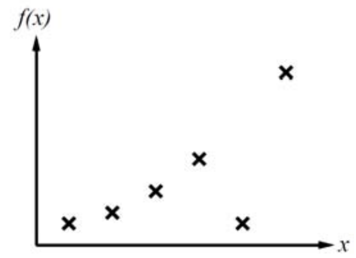
Problem: find a(n) hypothesis h
such that $h \approx f$
given a training set of examples

(This is a highly simplified model of real learning:
- Ignores prior knowledge
- Assumes a deterministic, observable "environment"
- Assumes examples are given
- Assumes that the agent wants to learn f—why?)

Inductive learning method

Construct/adjust h to agree with f on training set
(h is consistent if it agrees with f on all examples)

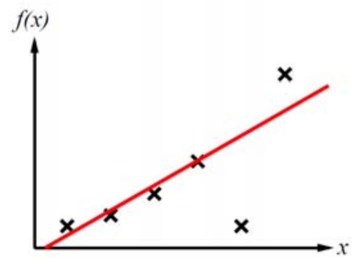
E.g., curve fitting:



Inductive learning method

Construct/adjust h to agree with f on training set
(h is consistent if it agrees with f on all examples)

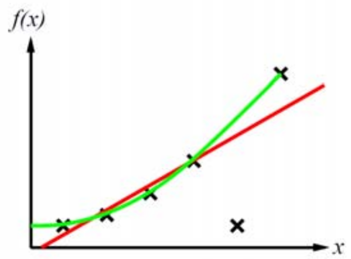
E.g., curve fitting:



Inductive learning method

Construct/adjust h to agree with f on training set
(h is consistent if it agrees with f on all examples)

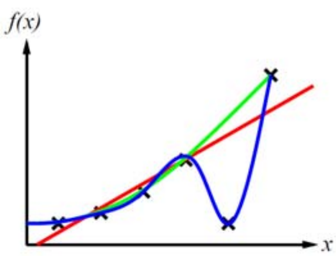
E.g., curve fitting:



Inductive learning method

Construct/adjust h to agree with f on training set
(h is consistent if it agrees with f on all examples)

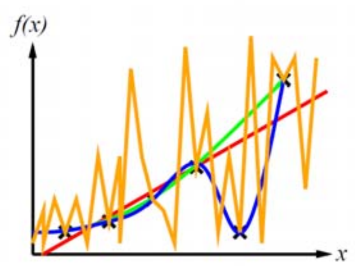
E.g., curve fitting:



Inductive learning method

Construct/adjust h to agree with f on training set
(h is consistent if it agrees with f on all examples)

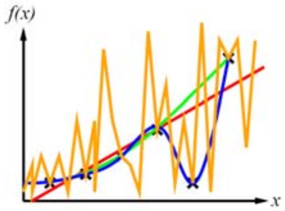
E.g., curve fitting:



Inductive learning method

Construct/adjust h to agree with f on training set
(h is consistent if it agrees with f on all examples)

E.g., curve fitting:



Ockham's razor: maximize a combination of consistency and simplicity

Core Questions

- *Learning to perform a task from experience*
- Task
  - Can often be expressed through a mathematical function

$$y = f(x; w)$$

  - x: Input
  - y: Output
  - w: Parameter (this is what is "learned")
- Classification vs. Regression
  - Regression: continuous y
  - Classification: discrete y
    - E.g. class membership, sometimes also posterior probability