

KIIS: Kunstig Intelligens & Intelligente Systemer

Artificial Intelligence & Intelligent Systems

Henning Christiansen

Introduction and overview

Autumn Semester 2007

©2007 by the author

Program for today

- What do we mean by Artificial Intelligence
- Overview of the course, practical issues
- (Artificial) intelligence and knowledge
- Examples of AI technology and applications
- A detour into the philosophical issues
- Prolog: An AI language :) *workshop 2.5h at least*

Read MN, chap 1, historical overview, etc. by yourself

Two views of Artificial Intelligence

- 1. Philosophical:* Understanding "intelligence", how to define "intelligence", can "intelligence" be implemented by a machine designed by human? [paper by A. Turing, 1950]
- 2. Pragmatic:* Practically relevant technology, doing tasks that we usually associate with human intelligence [no matter how it is done]
- 3. Salesmanship:* Smart people sell intelligent solution, no one sells stupid solutions

Two views of Artificial Intelligence

1. *Philosophical*: Understanding "intelligence", how to define "intelligence", can "intelligence" be implemented by a machine designed by human? [paper by A. Turing, 1950]
- 2. *Pragmatic*: Practically relevant technology, doing tasks that we usually associate with human intelligence [no matter how it is done]
3. *Salesmanship*: Smart people sell intelligent solution, no one sells stupid solutions

This course

- An overview of the area:
 - Basic concepts
 - Important applications
 - ... and techniques
- In depth with specific topics
- Book: *M. Negnevitsky: Artificial Intelligence, A Guide to Intelligent Systems, 2nd ed., Addison-Wesley, 2004*
- Complemented with
 - Articles, presentations by you
 - Course notes
 - Practical exercises, practical exercises, etc.!

Overall goals

- You get an overview of AI and AI methods so that
 - you can participate in AI development projects (e.g., student projects)
 - you get an idea of present state of the art
- You will be presented for current research topics, reflection (or course) the teachers' research interests
 - ideas for MSc theses and new research
- You get some experiences developing small toy AI applications (Prolog, CHR, other high-level tools)

List of possible topics in the course

- Rule-based expert system
 - (forward, backward chaining, etc.)
- Reasoning in logic based systems:
 - Deduction, Induction & Abduction modelled in logic programming.
 - Applications for diagnosis, natural language understanding (discourse analysis ...)
- Uncertainty: Statistics, Bayesian reasoning, related machine learning techniques
- To level of "know-about":
 - Fuzzy logic, neural nets, data mining techniques
- Evolutionary computation, genetic programming

Experimental tools

Central tools:

- Prolog *NB: We use SICTUS Prolog... to have:*
- Constraint Handling Rules

Additional specialized tools, perhaps

- Neural Networks in Excel
- some genetic programming system
- PRISM (T.Sato & al) for parameter learning
- some ILP (Inductive logic programming) tool

Practical

- Everything announced at the course web
Ecampus -> KIIS (...)
- Reading etc. for a next Tue, latest Wed (Thu)
- We aim at:
 - Each course day is seen as a whole, so practical exercises reflects the day's theoretical material
- We combine
 - traditional lectures, presentation by you
 - practical & theoretical exercises
 - workshop: mixing it all together
- **Exam** (more info at course www soon)
 - You give in some assignment(s) during course
 - Oral exam in January

Intelligence and intelligent systems

Requires

- Knowledge and knowledge representation
- Knowledge processing capabilities

May or may not involve

- ability to learn and adapt

NB: distinction between "knowledge" and "data" may be quite subtle; example: a rescue robot

- knowing about buildings, 3D geom., etc.
- can learn a 3D map from its obs., share with other robots...
- learn strategies from its earlier rescue tasks

Knowledge representation can be

- Explicit

- An encyclopedia
- Rule based

IF rains AND go-out AND NOT umbrella THEN get-wet

- Implicit or encoded

- A 1000000 line assembler program that plays chess
- The weights in a trained neural network for recognizing my face

Knowledge representation can be

- Explicit

- An encyclopedia
- Rule based

IF rains AND go-out AND NOT umbrella THEN get-wet

- Implicit or encoded

- A 1000000 line assembler program that plays chess
- The weights in a trained neural network for recognizing my face

Processing is (often) "logical"

GOAL: NOT get-wet GIVEN rains

PROCESS: Apply logical law of contraposition to get

RESULT: NOT go-out OR umbrella

Knowledge interesting by itself
Process may generate explanation

Representation can be

Processing is (often) "logical"

- Explicit

- An encyclopedia
- Rule based

GOAL: NOT get-wet GIVEN rains

PROCESS: Apply logical law of contraposition to get

RESULT: NOT go-out OR umbrella

IF rains AND go-out AND NOT umbrella THEN get-wet

- Implicit or encoded

- A 1000000 line assembler program that plays chess
- The weights in a trained neural network for recognizing my face

Knowledge interesting by itself
Process may generate explanation

Representation can be

Processing is (often) "logical"

- Explicit

- An encyclopedia
- Rule based

GOAL: NOT get-wet GIVEN rains

PROCESS: Apply logical law of contraposition to get

RESULT: NOT go-out OR umbrella

IF rains AND go-out AND NOT umbrella THEN get-wet

- Implicit or encoded

- A 1000000 line assembler program that plays chess
- The weights in a trained neural network for recognizing my face

Processing is opaque

**KnowledgeRepr.+Process only recognizable by
sufficiently high percentage of successful results**

Representation can be

Knowledge interesting by itself
Process may generate explanation

Processing is (often) "logical"

- Explicit

- An encyclopedia
- Rule based

GOAL: NOT get-wet GIVEN rains

PROCESS: Apply logical law of contraposition to get

RESULT: NOT go-out OR umbrella

IF rains AND go-out AND NOT umbrella THEN get-wet

- Implicit or encoded

- A 1000000 line assembler program that plays chess
- The weights in a trained neural network for recognizing my face

In place for knowledge
difficult to formalize

Processing is opaque

KnowledgeRepr.+Process only recognizable by
sufficiently high percentage of successful results

Knowledge about real world may often be

- incomplete (for given problem...)
- imprecise (temp is 800 degrees \pm 50)
- stochastic (75% chance that Powerpoint crashes during my lecture)
- fuzzy (Peter is tall)
- wrong
- inconsistent

This calls for

- sophisticated knowledge repr. formalisms
- add-hoc tricks, rule-of-thumbs, etc.

- **sophisticated knowledge repr. formalisms**
 - logic with open and closed predicates ("null values"),
 - fuzzy logic ~ graduated truth, linguistic quantifiers
 - probability theory
 - integration of probability theory and logic
 - paraconsistent logic
 - 10^6 different "uncertainty logics"
- **add-hoc tricks, rule-of-thumbs, etc.**
 - meta-rules in rule-based systems
 - hacks in the code, shifting algorithm, ...
 - behaviour of implicit representations, e.g., neural nets capable of recognizing my face when I have cleaned the fire place

Machine learning

Extracting knowledge from sets of observations

- Supervised learning

 - Data annotated manually (typ.) with classification

 - Machine identifies relationships Data --> class

- Unsupervised learning

 - Machine identifies regularities in data

 - Perhaps helped by "bias"

Supervised learning

- Train neural net with different faces with info.
"this shows NN1", "this shows also NN1", "this shows NN2", ..
- Learn rules to classify words from annotated text, "tagging"
- "Give/V-IMP-XP help/N! Help/V-IMP-XP me/PN-1P-OBJ"
- Protein chemical formulas --> shape of molecule, or the other way round!

Unsupervised learning

- Datamining in data warehouses
"80% of those who bought gin bought also tonic"
"50% of those who bought gin and tonic bought also peanuts"
- Learning logic programs from facts
- Learning Integrity Constraints in databases

A detour into the philosophical issues

Alan Turing (1950):

Computing Machinery and Intelligence

~philosophical considerations by one of the greatest computer scientists

The paper is very interesting — read it if you have time!

We touch upon a single issue: ***The Turing test***

The Turing test: A computer is intelligent if it can fool you to believe it's human

Think of a convincing chat program,

To abstract irrelevant features away, assume all communication goes via keyboard and text windows

Turing's original version:

1: A: Man, B: Woman, C: "Interrogator"

2: A: **Machine**, B: Woman, C: "Interrogator"

Game: A does what he/it can to fool C to believe A is the woman.

If C is fooled equally many times by A="Man" and A="Machine", the Machine is Intelligent

Rest of today

- Prolog workshop
- short lecture with online demo
- you work with exercise
- ... and you have written your first intelligent programs by the end of the day ;-)

If time permits, another short lecture, and more exercises....