

COMP210: Artificial Intelligence

Lecture 2. Typical AI problems and applications

Trevor Bench-Capon
Room 215, Ashton Building

<http://www.csc.liv.ac.uk/~tbc/COMP210>

- Last Time
 - General Course Information
 - Overview of What AI is
- Today
 - Overview of some common AI techniques we will study during the course
 - Typical AI applications
 - A brief history of AI

Example: "Expert" tasks

- Many tasks require a human expert
 - medical diagnosis;
 - equipment repair;
 - computer configuration;
 - Legal advice
 - financial planning.
- Everyday tasks (recognising faces, understanding speech) in general are much harder for a machine.
- Expert systems have been successfully developed in the areas of the expert tasks.

Typical AI Application Areas

- natural language processing - understanding, generating, translating;
- planning;
- vision - scene recognition, object recognition, face recognition;
- robotics; car driving
- theorem proving;
- speech recognition;
- game playing;
- problem solving;
- expert systems etc.

State of the Art (I)

- Autonomous planning and scheduling
 - Route planning
 - Automated scheduling of actions in spacecrafts
- Game playing
 - IBM's Deep Blue defeated G.Kasparov (the human world champion) (1997)
 - The program FRITZ running on an *ordinary PC* drew with V.Kramnik (the human world champion) (2002)
 - Poker (fixed limit) now a match for the world's best
- Autonomous control
 - Automated car steering
 - The Mars mission

State of the Art (II)

- Diagnosis
 - Evidence of human level performance
- Logistic planning
 - Defence Advanced Research Project Agency (US) stated that this single application more than paid back
 - DARPA's 30-year investment in AI
- Robotics
 - Microsurgery
 - RoboCup — "By the year 2050, develop a team of fully autonomous humanoid robots that can win against the human world soccer champion team"

In Twenty Years computers will be capable of doing any work that a man can do

Said in 1965 – Herb Simon (one of the founders of AI)

Basic AI Building Blocks

- Symbolic AI
 - Problem solving by *searching*
 - Ability to represent *knowledge*
 - Ability to *reason*
 - ...Many more...
 - AI programming languages
- Sub Symbolic AI
 - Artificial Neural Networks
 - Connectionism

Search

- Often no direct way to solve a problem.
- You may know what moves are allowed but not *how to* put the moves into a sequence to solve a problem.
- Can generate possibilities for next step and so on.
- Considering full search space often too expensive. Too many possibilities (even for computers), so heuristics are needed

Search Examples

- Planning
 - Route finders
 - Timetabling
- Games
 - Noughts and crosses
 - Chess
- Puzzles

Rubik's Cube

- 43,252,003,274,489,856,000 combinations
- Up to 481,229,803,398,374,426,442,198,455,156,736 brute-force solution attempts
- More than 15,259,696,962,150,381 years
- Need to look at heuristics or strategies, i.e. Selecting the best options to lead to a solution.

Knowledge Representation

- How do we represent the states of a Rubik cube, and the operations that can be performed to allow us to solve the problem?
- A good Knowledge representation should
 - make the processing easy
 - turn unfamiliar problems into familiar ones
- So it must fit
 - The domain - the knowledge to be represented
 - The task - what we want to do with the knowledge

Knowledge Representation

- Example: given the headline "Obama wins election." could a machine answer the question "Who is the president?"
 - Background knowledge is necessary
 - How is this knowledge written down, or encoded so a computer can use it?
 - How can it be written down efficiently? We can't write everything down.
 - What do the formal representations mean? Semantics.

Knowledge Representation Techniques

- Rules?
 - Production systems
 - Logic programming (Prolog)
- Relationships
 - Semantic networks
- Objects
 - Frames
 - Objects
 - Agents
- Idealised and formalised knowledge?
 - Mathematical equations
 - Various logics, first order, temporal, epistemic, description logics

Reasoning and Inference

- If we know that elephants are mammals with four legs and that Barbar is an elephant, can we conclude that Barbar is a mammal with four legs?
- If we use formal *logic as a knowledge representation language* logical proof can be used to allow us to infer new facts.
- But how do we deal with exceptions: Celeste is an elephant (and a mammal) but only has three legs?

AI Programming Languages

- Need a language good for symbolic manipulation (rather than numeric).
- LISP - short for List Processing. A functional programming language. Many AI systems have been written in LISP.
- Prolog - we'll use this language for the course. Short for PROgramming in LOGic. Based on a proof method in logic called resolution. Easy to get started and good for symbolic manipulation.
- But can be written in any general purpose language, eg C or Java. Particularly, as AI programs become components of other larger systems rather than stand alone systems.

Brief History of AI 1943–56:

- McCulloch & Pitts (1943)
 - artificial neural net — proved equivalent to Turing machine;
- Shannon, Turing (1950)
 - Information theory
 - Turing Test
 - chess playing programs
- Marvin Minsky (1951)
 - first neural net computer — SNARC
- Dartmouth College (1956)
 - term "AI" coined by John McCarthy
 - Newell & Simon presented LOGIC THEORIST program

Every aspect of learning or any other feature of intelligence can in principle be so precisely described that a machine can be made to simulate it' Dartmouth manifesto

1956-70

- Programs written that could:
 - plan, learn, play games, prove theorems, solve problems.
- Major centres established:
 - Minsky — MIT
 - McCarthy — Stanford
 - Newell & Simon — CMU
- Major feature of the period was *microworlds* — toy problem domains.
 - Example: blocks world.
 - "It'll scale, honest. . ."

1969 First International Joint Conference on Artificial Intelligence held
1970 First Issue of Artificial Intelligence Journal

1970s:

- 1970s period of recession for AI (especially in UK)
 - (Lighthill report in UK) "formed the basis for the decision by the British government to end support for AI research in all but two universities [from AIAMA]."
- Techniques developed on microworlds would not scale.
- Implications of complexity theory developed in late 1960s, early 1970s began to be appreciated:
- brute force techniques will not work.
- works in principle does not mean works in practice.
- In US — foundational work on expert and knowledge based systems

1980s:

- General purpose, brute force techniques don't work, so use knowledge rich solutions.
- Early 1980s saw emergence of expert systems as systems capable of exploiting knowledge about tightly focused domains to solve problems normally considered the domain of experts.
- Ed Feigenbaum's knowledge principle.
- In UK Alvey programme (1984-89) revived funding and interest

1990s:

- Many companies set up to commercialise expert systems technology went bust. (US AI winter)
- 1990s: emphasis on understanding the interaction between *agents and environments*.
- AI as component, rather than as end in itself.

2000s

- Agents developed as a key technology for symbolic AI
 - www as a delivery mechanism
- Revival of sub symbolic AI and probability networks
 - Advances in robotics, vision etc

- Today
 - Rapid look at some applications, techniques, history and state of the art
- Next Time
 - Introducing Prolog