1 Introduction to Artificial Intelligence

1.1 What is Artificial Intelligence?

- What intelligent human capability/job/skill that is currently unautomated do you think is potentially automatable?
- What would be an interesting thing for an AI system to be able to do?

Artificial intelligence is about automating intelligent capabilities.

1.2 Some well-known existing applications

- Expert systems (organic chemistry, medicine, geology, configuring computers)
- Speech recognition (Ticketmaster, Speechworks, Bell)
- Handwriting recognition (US postal service zip code readers)
- Game playing (chess, backgammon, checkers — world champion)
- Robots (automated cars, ping pong player, Honda robot)
- Automated theorem proving (Robbins problem — 60 year old open problem in finite algebra, 4-color theorem, circuit testing)
- Web search engines
- Natural language understanding (Eliza, Ask Jeeves, spell checkers, grammar checkers, translation: meteorological reports, Google)
- NASA probes (takes light $\sim$1min to Mars, $\sim$7mins to Jupiter)
- Logistics scheduling (military — people, cargo, vehicles)
- Cruise missiles
- Microsoft Answer Wizard

1.3 Main intelligent capabilities

Reasoning

- mathematical reasoning
- think logically
- solve problems
- figure out puzzles, games
- reason from first principles

E.g. Writing programs, designing software

Hard to automate because:

* large search space
* hard to decompose (modularity essential)
* hard to abstract (choosing data structures/objects critical)

Knowing

- embody knowledge about the world
- be an expert in some specialized domain
- have common sense about the way things work/are

E.g. Medicine, automechanics

Hard to automate because:

* vast knowledge
* implicit non-expert knowledge

Interpreting

- interpret inputs/signals
- perceive, understand
- draw plausible conclusions
- reason from incomplete/ambiguous/conflicting evidence
- understand speech/language/vision

E.g. Vision

Hard to automate because:

* image is just an array of numbers (pixel intensities)
* no introspection
* ambiguous, sketchy, conflicting evidence

E.g. Understanding language

Hard to automate because:

* no introspection
* ambiguous, sketchy, conflicting evidence

Behaving

- achieve goals
- act efficiently
- demonstrate skill, coordination
- make good decisions
- trade off short term vs. long term reward
- survive

E.g. Chess

Hard to automate because:

* uncertainty
* opponent modelling
* short term vs. long term tradeoffs

E.g. Office politics

E.g. Walking

Hard to automate because:

* coordinate many variables
* uncertainty
* system is inherently unstable
* real time (i.e. limited control computation time)

Learning

- adapt behavior

- acquire knowledge
- acquire skills
- improve performance with experience

E.g. Learning language

Hard to automate because:

* hidden prior constraints
* no introspection

E.g. Learning to walk

Hard to automate because:

* hidden prior constraints
* no introspection

1.4 What is this course about?

Fundamental principles and techniques for

- Automated reasoning, problem solving
- Knowledge representation *(briefly)*
- Probability models and probabilistic inference for interpretation *(sketch of applications to vision, speech, language interpretation)*
- Optimal behavior *(decisions, games, sketch of robotics)*
- Learning

We will focus on algorithms and underlying mathematics

- *not* applications (this is primarily for the course project)

1.5 AI is still primarily about *research*

- experimental computer science
- frontiers of computer science

1.6 What this course is *not* about

- Magic
- Hype

This course will focus on an engineering approach to AI, where we view AI simply as another application area of CS. That is, the specific problems studied are motivated by the desire to achieve the intelligent capabilities outlined above, but once identified, we take a standard CS approach to solving them: mathematical modeling, algorithm design and analysis, performance evaluation. The criterion for success is simply to achieve the desired capability, and nowhere in this endeavor do we invoke biological or psychological plausibility.

This course is not about:

Cognitive/neurological/psychological modeling. In fact, strictly speaking, this is not even a course about *science*, in the sense that we will not postulate models about nature (people, birds, insects) which we then go out and verify or refute by conducting physical experiments. Our goal is simply to achieve the desired capabilities in a computer, however we can achieve them (for now, let's assume a non-quantum computer).

Complementary to this viewpoint would be a course on "cognitive science" where one would investigate various models of human (and animal) intelligence, which would presumably then be testable in the world. We will not do that here. Disciplines usually represented in a cognitive science program are: psychology, linguistics, neuroscience, philosophy and computer science. (Waterloo, in fact, has a very interesting undergraduate program in cognitive science.) Clearly, cognitive science informs AI and vice versa, but I will not have time to teach much about cognitive science in this course. (Some excellent background references on cognitive science are listed below.)

Another thing this course is *not* about is:

Novel software engineering paradigms. From time to time, AI is presented to computer scientists as being primarily about creating new methodologies for developing complex software systems. Clearly, the resulting systems are often complex and push the limits of contemporary software engineering practice (in whatever time period). In fact, useful new programming languages and engineering strategies have indeed emerged along the way (*e.g.*, LISP and PROLOG, rule-based and knowledge-based systems). However, *every* area of CS pushes software engineering practice, all the time, and it is hard to argue that AI has been specially prolific in this regard. So rather than try to sell "new-age general-purpose programming remedies" I will instead focus on basic problems and how we can go about solving them using conventional CS methods.

A perspective we will adopt throughout this course is that:

- There is no magic bullet
- There is no free lunch
- There is no universal anything

  (learning, intelligence, problem solver, you name it)

  (and, yes, this includes human beings

  [as much as it includes chimpanzees, dolphins, elephants, ants ...])
- General purpose X generally means generally weak

In reality, when we study AI, we are faced with hard problems, hard engineering, and few easy answers. But with that in mind, perhaps we can start to make progress.

1.7 Textbook readings

Russell and Norvig 2nd Ed., Chapters 1 and 2.
Dean, Allen and Aloimonos, Chapter 1.

Background readings on cognitive science

Damasio (99) *The Feeling of What Happens: Body and Emotion in the Making of Consciousness.*
Pinker (97) *How the Mind Works.*
Gopnik, Meltzoff, Kuhl (99) *The Scientist in the Crib: What Early Learning Tells Us About the Mind.*
Pinker (94) *The Language Instinct: How the Mind Creates Language.*
Damasio (94) *Descarte's Error: Emotion, Reason and the Human Brain.*
Dehaene (97) *The Number Sense: How the Mind Creates Mathematics.*
Lakoff, Nuñez (00) *Where Mathematics Comes From: How the Embodied Mind Brings Mathematics into Being.*

1.8 Course outline

	Introduction	
	Lecture 1	Introduction to AI
Part 1	**Reasoning**	
	Lecture 2	Automating reasoning: formal inference
	Lecture 3	Correct and exhaustive reasoning
	Lecture 4	Constraint satisfaction search
	Lecture 5	Problem solving search
	Lecture 6	Automated planning
	Lecture 7	Planning algorithms
	Lecture 8	General first order representation
	Lecture 9	Planning in logic, First order inference
Part 2	**Knowing**	
	Reading	*Russell & Norvig 2nd Ed., Chap. 10, Knowledge Representation*
Part 3	**Interpreting**	
	Lecture 10	Automating interpretation systems
	Lecture 11	Probability modelling
	Lecture 12	Structured probability models
	Lecture 13	Efficient probabilistic inference
	Lecture 14	Inference in complex models
	Lecture 15	Interpreting senses (perception)
	Lecture 16	Parsing and disambiguation
Part 4	**Behaving**	
	Lecture 17	Optimal behavior: decision theory
	Lecture 18	Optimal sequential decision making
	Lecture 19	Optimal behavior: game theory
	Lecture 20	Scaling up: Partial observability
	Reading	*Russell & Norvig 2nd Ed., Chap. 25, Robotics and Control*
Part 5	**Learning**	
	Lecture 21	Types of learning problems
	Lecture 22	Function learning algorithms
	Lecture 23	Generalization theory / Overfitting
	Conclusion	
	Lecture 24	Course review