COSC343: Artificial Intelligence

Lecture 1: Introduction

Alistair Knott

Dept. of Computer Science, University of Otago

Your teachers

Lecturers:

Ali Knott

email	alik@cs.otago.ac.nz
phone	479 8299
office	Owheo room 254

Lech Szymanski

email	lechszym@cs.otago.ac.nz
phone	479 5691
office	Owheo room 249

Tutor:

Chris Gorman

email	343tutor@cs.otago.ac.nz

Textbook and webpage

The course textbook is *AI: A modern approach* by Stuart Russell and Peter Norvig (Third edition, 2010).

- The course follows the textbook quite closely.
- We'll be indicating readings in advance.

The course webpage has lots of useful resources:

- `http://www.cs.otago.ac.nz/cosc343/`

Internal assessment

Internal assessment counts for 40% of the course. There are two assignments, worth a total of 28%:

	Topic	Handed out	Due	Worth
1	LEGO robots	Wed Week 1	Tue Week 6	14%
2	Genetic algorithms	Wed Week 7	Tue Week 10	14%

In addition, each tutorial (except the last one) is worth 1%, which is given for a tutorial exercise. This makes up the remaining 12% of marks.

Course schedule

The course has five parts.

1	Autonomous agents	Introduction Robots	1 lecture 1 lecture
2	Machine learning	Introduction Probability theory Classification and regression Neural networks Kernel methods Genetic algorithms Unsupervised learning	1 lecture 2 lectures 3 lectures 6 lectures 1 lecture 1 lecture 1 lecture
3	Search methods	State-space search Adversarial search	2 lectures 1 lecture
4	Natural language	Syntactic models Statistical models	3 lectures 2 lectures
5	Review	Ethics of AI Revisions	1 lecture 1 lecture

In today's lecture

1. What is AI?
2. Brief history of AI
3. The state of the art
4. Definitions: agents and environments
5. Some different types of AI agent

What is AI?

Here's a fairly uncontroversial definition:

AI is the study and creation of machines that perform tasks normally associated with intelligence.

People are interested in AI for several very different reasons:

- Psychologists/cognitive scientists are interested in finding out how people work. Machine simulations can help with this task.
- Engineers are interested in building machines which can do useful things. These include things which require intelligence.

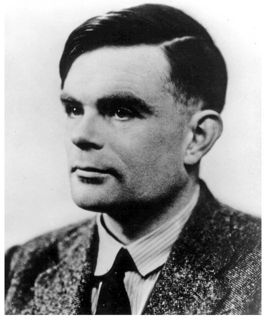
The 'tasks' to be performed could involve thinking, or acting, or some combination of these.

What is intelligence?

Alan Turing

Alan Turing is a founding father of AI, in two ways:

- he helped invent the computer;
- he initiated serious debate about whether computers could be made to think like humans.



The Turing test

Turing's main contribution to AI comes in a single article, "Computing machinery and intelligence" (Mind, 1950).

In this article,

- he tells people what computers are;
- he argues that since they can perform any computation, and human thinking just involves computation, they could be programmed to think like humans;
- he introduces a practical test for judging whether a computer can be said to be 'thinking like a human'.

The Turing test

Turing's argument is: 'If a computer can fool a judge into thinking it *is* human, we must acknowledge it is able to think like a human'.

In fact, there are many programs available now which *can* fool judges (at least some judges, for some length of time).

- The Loebner Prize: a yearly competition to see which program performs best in a Turing-test scenario.
  (Not everyone in AI thinks this is a good idea.)

Disciplines contributing to AI

AI has its origins in several disciplines, some old, some quite recent.

Philosophy:	'Reasoning': Aristotle, Boole, Frege 'The mind': Descartes, Locke, Berkeley, Hume
Linguistics	Chomsky's challenge to behaviourism
Experimental psychology:	Models of human information processing
Computer science:	Turing, Von Neumann...
Economics	Formal theory of rational decisions
Neuroscience	Bottom-up models of 'how the brain does it'
Cybernetics	Study of embodied agents
Mathematics	Notions of proof, algorithms, probability

Brief history of AI

1943-49	Early neural network models: McCulloch & Pitts, Hebb
1950	Turing's "Computing Machinery and Intelligence"
1950s	Early AI 'successes', including Samuel's checkers program,
	Newell & Simon's Logic Theorist
1956	Dartmouth meeting: "Artificial Intelligence" adopted
1966–74	AI discovers computational complexity
	Neural network research almost disappears
1970s–80s	Domain-specific AI: 'expert systems' developed
1980s–now	AI becomes an industry
1985–now	Neural networks return to popularity
1990s–now	Increased reliance on machine learning & stat. techniques Embodied AI and evolutionary approaches
2014	Neural networks adopted in industry

State of the art

Which of the following can be done at present?

- Quadruped robots...
- Hand/arm robots...
- Driverless cars...
- Visual object classifiers...
- Answering general knowledge questions...
- Machine translation...

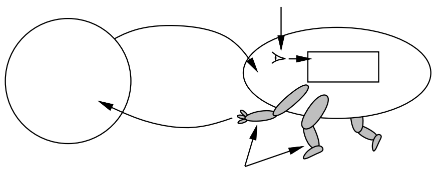
Agents and environments

Recall our general definition of AI:

AI is the study and creation of machines that perform tasks normally associated with intelligence.

In this course, we will use the term agent to refer to these machines.

- The term 'agent' is very general: it covers humans, robots, softbots, thermostats, etc.
- An agent has a set of sensors, a set of actuators, and operates in an environment.



The agent function

If we want, we can define an agent and its environment *formally*.

- We can define a set of actions $\mathcal{A}$ which the agent can perform.
- We can define a set of percepts $\mathcal{P}$ which the agent can receive. (Assume that there's one percept per time point.)

A simple agent function could simply map from percepts to actions:

$$f : \mathcal{P} \rightarrow \mathcal{A}$$

A more complex (and general) agent function maps from percept *histories* to actions:

$$f : \mathcal{P}^* \rightarrow \mathcal{A}$$

(This allows modelling of an agent with a *memory* for previous percepts.)

The agent program runs on the physical architecture to produce f.

An example: the vacuum-cleaner world

Here's a simple example: a formal definition of a robot vacuum cleaner, operating in a two-room environment.

- Its sensors tell it which room it's in, and whether it's clean or dirty.
- It can do four actions: move left, move right, suck, do-nothing.

Preliminaries for defining the agent function

To define the agent function, we need a syntax for percepts and actions.

- Assume percepts have the form [*location*, *state*]: e.g. [*A*, *Dirty*].
- Assume the following four actions: *Left*, *Right*, *Suck*, *NoOp*.

We also need a way of specifying the function itself.

- Assume a simple lookup table, which lists percept histories in the first column, and actions in the second column.

Percept history	Action
...	...
...	...

A vacuum-cleaner agent

Here's an example agent function for the vacuum cleaner agent:

Percept history	Action
[*A*, *Clean*]	*Right*
[*A*, *Dirty*]	*Suck*
[*B*, *Clean*]	*Left*
[*B*, *Dirty*]	*Suck*
[*A*, *Clean*], [*A*, *Clean*]	*Right*
[*A*, *Clean*], [*A*, *Dirty*]	*Suck*
⋮	⋮

And here's an agent program which implements this function:

function REFLEX-VACUUM-AGENT([*location*,*status*]) **returns** an action

if *status* = *Dirty* **then return** *Suck*
else if *location* = *A* **then return** *Right*
else if *location* = *B* **then return** *Left*

Evaluating the agent function

It is useful to *evaluate* the agent function, to see how well it performs. To do this, we need to specify a performance measure, which is defined as a function of the agent's environment over time.

Some example performance measures:

– one point per square cleaned up in time T?

– one point per clean square per time step, minus one per move?

– penalize for $> k$ dirty squares?

Formalising the agent’s environment

As well as a formal description of the agent, we can give a formal description of its environment.

Environments vary along several different dimensions:

- Fully observable vs partially observable
- Deterministic vs stochastic
- Episodic vs sequential
- Offline vs online
- Discrete vs continuous
- Single-agent vs multi-agent

The environment type largely determines the agent design.

Summary

In order to specify a scenario in which an agent performs a certain task, we need to define:

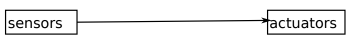
- a Performance measure;
- an Environment;
- a set of Actuators;
- a set of Sensors;

This is called the PEAS description of the agent/task.

Once the PEAS description has been given, we’re ready to define the Agent function.

Some different types of AI agent

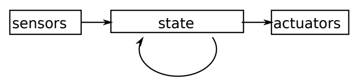
Reflex agents have a function that links sensors directly to actuators.



The function can be hardwired or learned.

Some different types of AI agent

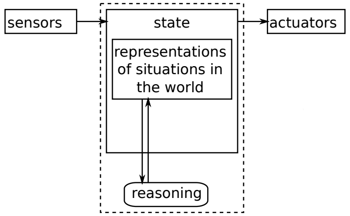
Agents with state have a function that can reference (and update) *internal variables*.



The function can be hardwired or learned.

Some different types of AI agent

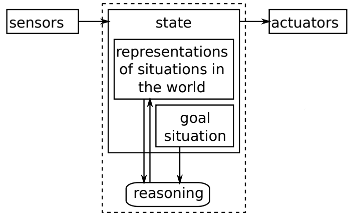
Reasoning agents can store (and manipulate) representations of *situations in the world* (real or imagined).



The functions can be hardwired or learned.

Some different types of AI agent

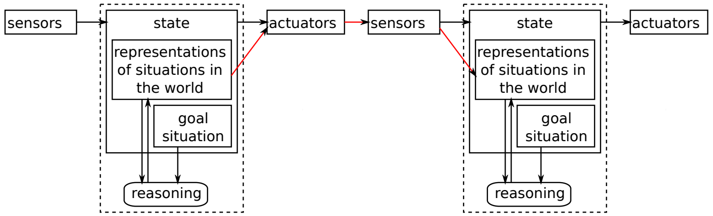
Goal-based agents can *search* for a way of achieving a desired situation.



The functions can be hardwired or learned.

Some different types of AI agent

Linguistic agents can *exchange* representations with one another.



The functions can be hardwired or learned.

Course schedule (recap)

1	Autonomous agents	Introduction	1 lecture
		Robots	1 lecture
2	Machine learning	Intro/probability	2 lectures
		Classification/regression	2 lectures
		Neural networks	6 lectures
		Kernel methods/GAs/unsupervised	3 lectures
3	Search methods	Heuristics, adversarial search	3 lectures
4	Natural language	Syntactic models	3 lectures
		Statistical models	2 lectures
4	Reasoning agents	Probability theory	1 lecture
		Bayesian methods	3 lectures
5	Review	Ethics of AI	1 lecture
		Revisions	1 lecture

Reading:

- For this lecture: AIMA Chs1 & 2
- For next lecture: no reading