Lecture 7
Intelligent Agents as a Framework for AI: Part I

Rob Gaizauskas

Lecture Outline

- Introduction: Agents and Environments
- Rational Agent Behaviour
- Aspects of Environments

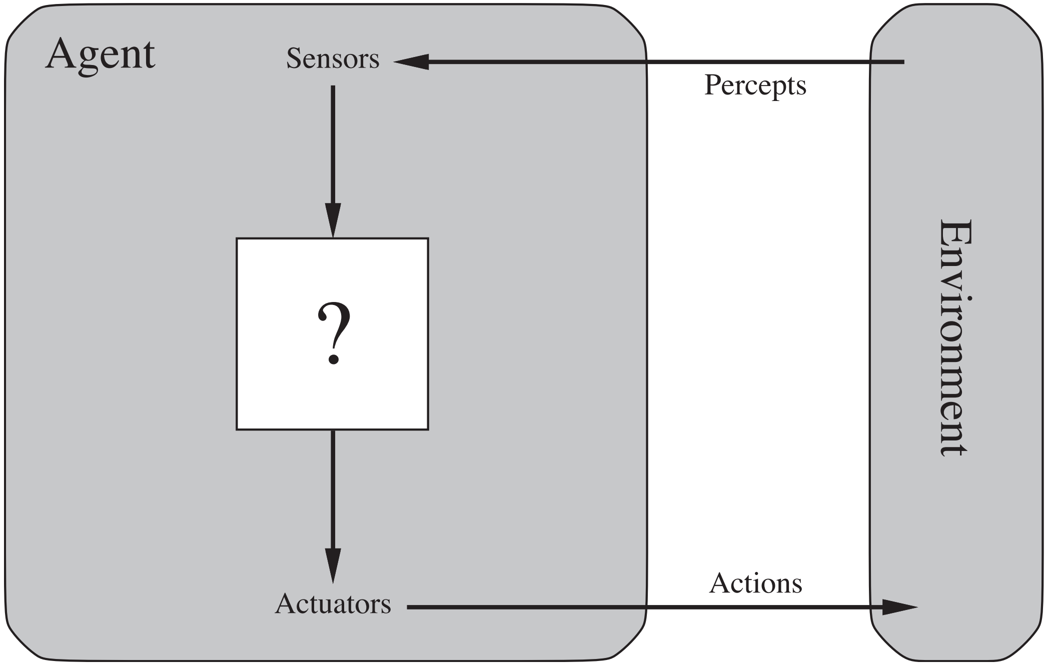
- Reading: *Russell & Norvig, Chapter 2: Intelligent Agents”

Introduction

- As we have seen, the "intelligent agents" approach arose as a general framework for studying AI in the 1990s
- Emphasises:
  - Agents that operate in an environment which they
    - Need to perceive/understand
    - Act upon to achieve goals
  - Intelligence – capability to act successfully in a complex environment
  - Interaction between multiple agents, each pursuing its own goals
- Russell and Norvig use this framework to structure their account of all of AI

Agents and Environments

- Definition: "an agent is anything that can be viewed as perceiving its environment through sensors and acting on that environment through actuators" (R&N, p. 34)



R&N Fig 2.1

Agents and Environments (cont)

- Examples:

Agent Type	Sensors	Actuators
Human	Eyes, ears, …	Hands, legs, vocal tract, …
Robot	Cameras, infrared range finder, …	Various motors
Software	Keystrokes, file contents, network packets, …	Screen display, writing to file, sending network packets
…	…	…

Percepts and Percept Sequences

- Definition: percept refers to an agent's perceptual inputs at any given instant
- Definition: an agent's percept(ual) sequence is the complete history of everything the agent has ever perceived

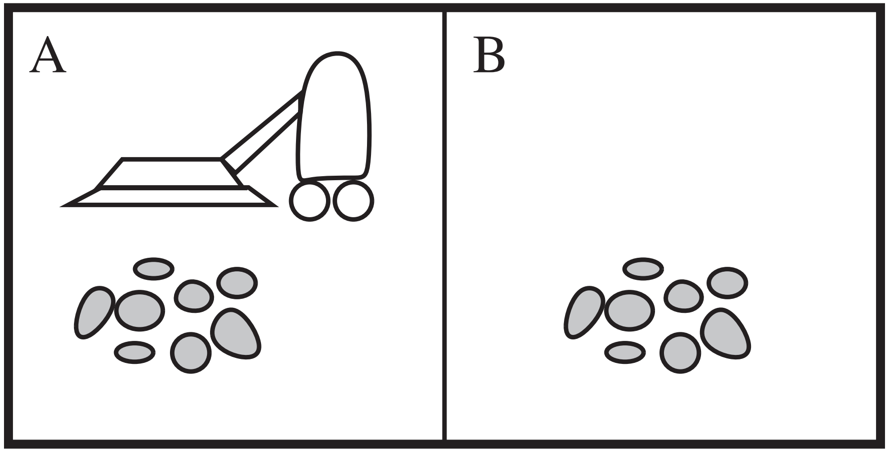
Agent Function

- An agent's choice of action at any time can depend on its entire percept sequence up to that time but not on anything it has not perceived
- If we specify an agent's choice of action for every possible percept sequence, we have completely described the agent
  - In mathematical terms an agent's behaviour is described by an agent function
    - maps any given percept sequence to an action
- Definition: agent function is a function from the set of percept sequences to the set of actions. It defines for any given percept sequence what action an agent will take when presented with that percept sequence

Agent Function vs Agent Program

- Could try to tabulate agent function for a given agent
  - Would be very large table – infinite unless some bound is placed on the length of percept sequences to be considered
- Given an agent we could experiment with it
  - Present possible percept sequences
  - Record agent's response
- Resulting table is an external characterisation of the agent
- Internally agent function implemented by an agent program
- Important to distinguish these two:
  - Agent function: abstract mathematical description of a mapping from percept sequences to actions
  - Agent program: concrete implementation running within a physical system

Example: Vacuum Cleaner World



- Two locations: A and B
- Vacuum agent can
  - Perceive
    - which square it is in
    - whether there is dirt in the square
  - Act by
    - Moving right; moving left
    - Sucking up dirt
    - Doing nothing
- One simple agent function: if current square dirty, then suck; else, move to other square (call this VCA-F1)

Example: Vacuum Cleaner World (Cont)

- Partial tabulation of the "suck or move" function:

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

Example: Vacuum Cleaner World (Cont)

- Filling in the "action" column differently leads to different agent functions
- What is the "right" way to fill in the column?
  - i.e. what makes an agent good/bad or intelligent/stupid?
- One answer:
  - An agent is good if it does the right thing, i.e. is rational
  - Can decide what is rational by looking at the consequences of an agent's behaviour

Good Behaviour + Rationality

- An agent, when placed in an environment, generates a sequence of actions depending on the percepts it receives
- This action sequence causes the environment to go through a sequence of states
- If the sequence of environment states is desirable then the agent has performed well
- To determine whether a sequence of environment states is desirable we need a performance measure that evaluates any sequence of environment states

Good Behaviour + Rationality (cont)

- Note that a performance measure must evaluate a sequence of environment states not agent states
  - Not sufficient for an agent to examine its sequence of actions and believe its performance is good
    - Could delude itself into thinking its performance was great
  - Must assess actual consequences of its actions in the environment

Good Behaviour + Rationality (cont)

- Different tasks require different performance measures
  - Need to be determined by agent designer
  - May not be straightforward.
- E.g. for vacuum cleaner agent
  - Measure 1: amount of dirt cleaned in 8 hour shift
    - Agent could maximise this measure by cleaning up dirt; dumping it on floor; cleaning it up again, etc.
  - Measure 2: reward for number of clean squares at each time step and average over time
- *In general better to design performance measures according to what is wanted in the environment, rather than according to how it seems the agent should behave*

Rational Agents

What is rational at a given time depends on:

- The performance measure that defines criterion of success
- The agent's prior knowledge of the environment
- The actions that the agent can perform
- The agent's percept sequence to date

Definition of rational agent:

> *For each possible percept sequence, a rational agent should select an action that is expected to maximize its performance measure, given the evidence provided by the percept sequence and whatever built-in knowledge the agent has.*

R&N, p. 37

Rational Agents (cont)

- Is the simple vacuum cleaner agent (VCA-F1) rational?
- Depends on
  - the performance measure
  - what's known about the environment
  - what sensors/actuators the agent has

Rational Agents (cont)

- Suppose:
  - Performance measure awards one point per clean square at each time step over a 1000 time step lifetime
  - Agent :
    - Knows "geography" of the environment (2 squares, A + B)
    - Does not know distribution of dirt or its initial position
    - Knows clean squares stay clean and sucking cleans current square
    - *Left/right* actions move left/right, except at boundary – then do nothing
  - Only available actions are: *left, right, suck*
  - Agent correctly perceives where it is + whether square contains dirt
- Then: the VCA-F1 agent is rational, i.e. expected performance is at least as high as any other agent's
  - Note: this claim can be proved

Rational Agents (cont)

Same agent would be irrational in different circumstances

- Different performance measure
  - Current agent shuffles needlessly back and forth between squares once they are clean
  - If performance measure penalises movement (e.g. because it consumes energy) then agent will not perform well
  - Better approach: do nothing once squares are clean; or, if squares can become dirty again then every so often the agent should check and clean any dirty squares
- Different knowledge of environment
  - If agent does not know "geography" of environment then will need to explore it rather than stick to squares A and B

Omniscience

- Need to distinguish rationality and omniscience
  - Omniscient (“all-knowing”) agents know the actual outcome of their actions and can act on this basis
- Omniscience impossible in reality
  - E.g. I plan to cross the street to see an old friend; I check traffic/other commitments, etc. and proceed to cross – but am obliterated by a door from an overflying airliner
  - Does this make me irrational? – no
- Rationality ≠ Perfection
  - Rataionality maximises expected performance; perfection maximises actual performance

Omniscience (cont)

- Definition of rationality does not require omniscience, because it relies only on percept sequence to date
- Does not mean an agent can act on the basis of under-informative percept sequences or be lazy about acquiring available perceptual info
  - E.g. needn't scan sky for bits of falling plane before crossing road; but do need to look both ways!
- Agents may need to engage in information gathering
  - perform actions in order to modify future percepts
  - exploration in order to map/understand an initially unknown environment

Information gathering is an important part of rationality

Learning

- Agent should not only gather information but also learn from what it perceives
- Agent may start with some prior knowledge of the environment, but as it gains experience this knowledge should may be modified and/or extended
- Agents that have a fixed/unmodifiable model of the environment are fragile /unable to adjust to change
- Example: dung beetle
  - drags ball of dung to nest and uses it to plug entrance (young will feed off it)
  - If ball is removed en route to nest will carry on as if nothing has happened
  - Example of "hard-wiring" in a simple agent – when it is violated agent cannot recover and unsuccessful (Irrational) behaviour results

Autonomy

- If an agent relies on prior knowledge of its designer and cannot adapt its behaviour based on its own percepts, we say it lacks autonomy
- An agent is more rational, i.e. is more likely to maximise its performance measure, if it is autonomous, i.e. can compensate for partial or incorrect prior knowledge
- Example: a vacuum cleaning agent that can learn to foresee where dirt will appear will do better than one that cannot
- Sensible to give artificial agents some knowledge at the outset – just as evolution give animals built-in reflexes to enable them to survive until they can learn for themselves
  - After sufficient experience agent can behave independently of its prior knowledge
  - Giving agents a learning capability allows them to succeed in a much larger set of environments

Aspects of Environments

- First step in designing an agent is to specify the task environment – involves specifying PEAS
  - Performance measure
  - Environment
  - Actuators
  - Sensors
- Keep in mind that agents can be robots, whose actuators/sensors interact with the physical world, or softbots, whose environment is the internet

Aspects of Environments

- First step in designing an agent is to specify the task environment – involves specifying PEAS
  - Performance measure
  - Environment
  - Actuators
  - Sensors

Beware! Potential Confusion: in R&N's terminology the task environment of an agent consists of 4 elements, one of which is the (external) environment of the agent …

- Keep in mind that agents can be robots, whose actuators/sensors interact with the physical world, or softbots, whose environment is the internet

Example Agent Types + PEAS Descriptions

Agent Type	Performance Measure	Environment	Actuators	Sensors
Taxi Driver	Safe, fast, legal, comfortable trip, maximise profits	Roads, other traffic, pedestrians, customers	Steering, accelerator, brake, signal, horn	Cameras, sonar, speedometer, GPS, odometer, accelerometer, etc.
Medical Diagnosis System	Healthy patient, reduced costs	Patient, hospital, staff	Display of questions, tests, diagnoses, treatments, etc.	Keyboard entry of symptoms, findings, patient's answers
Satellite image analysis system	Correct image categorization	Downlink from orbiting satellite	Display of scene categorization	Colour pixel arrays
Part-picking robot	% of parts in correct bins	Conveyor belt with parts; bins	Jointed arm and hand	Camera, joint angle sensors
Refinery controller	Purity, yield, safety	Refinery, operators	Valves, pumps , heaters, displays	Temperature, pressure, chemical sensors
Interactive English tutor	Student's score on test	Set of students, testing agency	Display of exercises, suggestions, corrections	Keyboard entry

Properties of Task Environments: Fully vs Partially Observable Environments

- A fully observable environment is one where an agent's sensors tell the agent the complete state of the environment at each point in time
  - Convenient because the agent need not maintain any internal state to keep track of the world
- A partially observable environment is one that is not fully observable
  - Sensors may be noisy or inaccurate
  - Part of the state may be missing from sensor data
    - E.g. vacuum cleaner agent can only sense dirt in the square it is in
    - E.g. automated taxi cannot see what other drivers are thinking

Single Agent vs Multiagent Environments

- Obvious distinction in some ways:
  - Agent solving a crossword by itself is in a single agent environment
  - Agent playing chess is in a multiagent environment
- Which entities which can be viewed as agents must be viewed as agents?
  - Must an automatic taxi agent A view another vehicle as an agent B? Or can it be treated just as a physical object?
- Key issue is whether B's behaviour is best described as trying to maximize a performance measure that depends on A's behaviour

Single Agent vs Multiagent Environments

- Example: in chess, opponent B is trying to maximize its performance measure, which minimizes A’s performance measure
- So, chess is a competitive multiagent environment
- Example: in the taxi-driving environment avoiding collisions maximizes performance measure of all drivers; however, only one car, e.g., can fit in a parking space
- So taxi-driving it is a partially co-operative and partially competitive multiagent environment
- Agent design for multiagent environments is quite different from single agent environments
  - Communication may emerge as a rational behaviour
  - Randomized behaviour may also be rational in some cases

Deterministic vs Stochastic Environments

- Definition: an environment is deterministic if the next state of the environment is completely determined by the current state and the agent's action; otherwise it is stochastic
- If the environment is only partially observable then it may appear to be stochastic
  - Most real environments are like this – cannot observe all potentially observable aspects
- Definition: An environment is said to be uncertain if it is either not fully observable or not deterministic

- Note: "stochastic" implies that uncertainty about outcomes is quantified as probabilities; in "nondeterministic" environments actions have associated possible outcomes, but no probabilities – here agents may be required to succeed for all outcomes

Episodic vs Sequential Environments

- Definition: in an episodic task environment the agent's experience is divided into atomic episodes, where:
  - in each episode the agent receives a percept and forms an action
  - the next episode does not depend on the actions taken in previous episodes
- Definition: in a sequential task environment the current decision could affect all future decisions
- Examples:
  - Classification tasks, e.g. agents spotting defective units on an assembly line, are often episodic – classification of previous units irrelevant to current decision
  - Chess and taxi-driving are sequential – current decision affects subsequent options
- Note: sequential task environments require the agent to think ahead (and hence are harder than episodic ones)

Static vs Dynamic Environments

- Definition: an environment is dynamic if the environment can change while the agent is deliberating; otherwise it is static
- Static environments are clearly simpler, as the agent does not need to check the world while working out what to do, or worry about time passing
- An environment is semi-dynamic if the environment does not change with time, but the agent's score does
- Examples:
  - Taxi-driving is dynamic
  - Chess, with a clock is semi-dynamic
  - Crossword puzzles are static

Discrete vs Continuous Environments

- Definition: a task environment is discrete is if it has a finite number of discrete states, time is handled discretely and if the agent has a finite number of distinct percepts and actions; otherwise it is continuous
- Examples:
  - Chess environment has a discrete number of states; it also has a discrete number of percepts and actions
  - Taxi-driving is
    - a continuous state/continuous time environment (speed/location take on a continuous range of values)
    - actions are continuous (steering angles/acceleration, etc.)
    - percepts are continuous (strictly speaking input from digital cameras is discrete, but treated as continuous)

Known vs Unknown

- Strictly, distinction refers not to environments, but to agent's state of knowledge of the "laws" that govern the environment
- In a known environment the outcomes (or outcome probabilities if stochastic) are given for all actions
- In an unknown environment the agent needs to learn how it works before good decisions can be made
- Note: known/unknown ≠ fully/partially observable
  - Known environment can be partially observable (e.g. solitaire)
  - Unknown environment can be fully observable (e.g. in a new video game the screen may show entire state, but what the buttons do may be unknown)

Examples of Task Environments and their Properties

Task Environment	Observable	Agents	Deterministic	Episodic	Static	Discrete
Crossword Puzzle	Fully	Single	Deterministic	Sequential	Static	Discrete
Chess with a Clock	Fully	Multi	Deterministic	Sequential	Semi	Discrete
Poker	Partially	Multi	Stochastic	Sequential	Static	Discrete
Backgammon	Fully	Multi	Stochastic	Sequential	Static	Discrete
Taxi Driving	Partially	Multi	Stochastic	Sequential	Dynamic	Continuous
Medical Diagnosis	Partially	Single	Stochastic	Sequential	Dynamic	Continuous
Image Analysis	Fully	Single	Deterministic	Episodic	Semi	Continuous
Part-picking Robot	Partially	Single	Stochastic	Episodic	Dynamic	Continuous
Refinery controller	Partially	Single	Stochastic	Sequential	Dynamic	Continuous
Interactive English Tutor	Partially	Multi	Stochastic	Sequential	Dynamic	Discrete

Summary

- AI can be conceived of as "the science of agent design" (R&N)
- An agent is something that perceives and acts in an environment
  - The agent function specifies the action an agent takes in response to a percept sequence
- A performance measure evaluates the behaviour of an agent in an environment
  - Rational agents act so as to maximise the expected value of the performance measure given the percept history

Summary (cont)

- A task environment includes PEAS: the performance measure, external environment, actuators and sensors
  - First step in agent design is to specify the task environment
- Task environments vary along several dimensions:
  - Fully vs partially observable; single vs multi-agent; deterministic vs stochastic; episodic vs sequential; discrete vs continuous; dynamic vs static; known vs unknown