Lecture 2 - Intelligent Agents

Dr. Brigitte Jaumard

April 2019

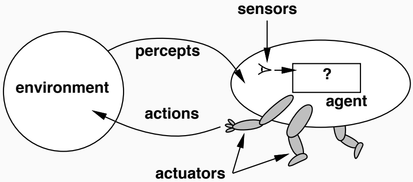
Computer Science & Software Engineering
bjaumard@cse.concordia.ca

Introduction

Outline

- Agents and environments
- Rationality
- PEAS (Performance measure, Environment, Actuators, Sensors)
- Environment types
- Agent types

Agents and environments



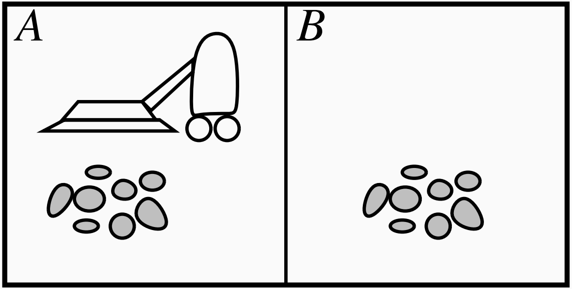
Agents include humans, robots, thermostats, etc.

The agent function maps from percept histories to actions:

$$f : \mathcal{P}^\star \to \mathcal{A}$$

The agent program runs on the physical architecture to produce f

Vacuum-cleaner world



Percepts: location and contents, e.g., $[A, \mathit{Dirty}]$

Actions: *Left*, *Right*, *Suck*, *NoOp*

A vacuum-cleaner agent

Percept sequence	Action
$[A, Clean]$	$Right$
$[A, Dirty]$	$Suck$
$[B, Clean]$	$Left$
$[B, Dirty]$	$Suck$
$[A, Clean]$, $[A, Clean]$	$Right$
$[A, Clean]$, $[A, Dirty]$	$Suck$
$\vdots$	$\vdots$

function Reflex-Vacuum-Agent[*location*, *status*]

if $status = Dirty$ **then** return $Suck$

else if $location = A$ then return $Right$

else if $location = B$ then return $Left$

What is the right way to fill out the table?

Can it be implemented in a small agent program?

Rationality

Fixed performance measure evaluates evaluates any given sequence of environment states.

- 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?

A rational agent chooses whichever action maximizes the expected value of the performance measure given the percept sequence to date

Rational $\rightsquigarrow$ exploration, learning, autonomy

Rationality (Cont'd)

A rational agent chooses whichever action maximizes the expected value of the performance measure given the percept sequence to date

An omniscient agent knows the actual outcome of its actions and can act accordingly; but omniscience is impossible in reality.

Rational $\neq$ omniscient
$\rightsquigarrow$ percepts may not supply all relevant information

Rational $\neq$ clairvoyant
$\rightsquigarrow$ action outcomes may not be as expected
Hence, rational $\neq$ successful

Rational $\rightsquigarrow$ exploration, learning, autonomy

PEAS - Performance measure, Environment, Actuators, Sensors

PEAS

To design a rational agent, we must specify the task environment

Consider, e.g., the task of designing an automated taxi:

- Performance measure ???
- Environment ???
- Actuators ???
- Sensors ???

PEAS

To design a rational agent, we must specify the task environment

Consider, e.g., the task of designing an automated taxi:

- Performance measure? safety, destination, profits, legality, comfort, ...
- Environment? Vietnam streets/freeways, traffic, pedestrians, weather, ...
- Actuators? steering, accelerator, brake, horn, speaker/display, ...
- Sensors? video, accelerometers, gauges, engine sensors, keyboard, GPS, ...

Internet shopping agent

- Performance measure ???
- Environment ???
- Actuators ???
- Sensors ???

Internet shopping agent

- Performance measure ? price, quality, appropriateness, efficiency
- Environment? current and future WWW sites, vendors, shippers
- Actuators? display to user, follow URL, fill in form
- Sensors? HTML pages (text, graphics, scripts)

Environment types

Fully observable vs. partially observable

A task environment is effectively fully observable if the sensors detect all aspects that are relevant to the choice of action; relevance, in turn, depends on the performance measure.

An environment might be partially observable because of noisy and inaccurate sensors or because parts of the state are simply missing from the sensor data

Example.

- a vacuum agent with only a local dirt sensor cannot tell whether there is dirt in other squares
- an automated taxi cannot see what other drivers are thinking

Deterministic vs. stochastic

A task environment is deterministic if the next state of the environment is completely determined by the current state and the action executed by the agent.

Otherwise, we say the environment is stochastic.

We say an environment is uncertain if it is not fully observable or not deterministic.

Example.

- Taxi driving is clearly stochastic, because one can never predict the behavior of traffic exactly
- An unreliable suction mechanism $\rightsquigarrow$ stochastic

Episodic vs. sequential

In an episodic task environment, the agent s experience is divided into atomic episodes. In each episode the agent receives a percept and then performs a single action. Crucially, the next episode does not depend on the actions taken in previous episodes. Many classification tasks are episodic.

In sequential environments, on the other hand, the current decision could affect all future decisions

Example.

- An agent that has to spot defective parts on an assembly line bases each decision on the current part, regardless of previous decisions $\leadsto$ episodic
- Chess and taxi driving are sequential

Static vs. dynamic

If the environment can change while an agent is deliberating, then we say the environment is dynamic for that agent; otherwise, it is static.

Static environments are easy to deal with because the agent need not keep looking at the world while it is deciding on an action.

If the environment itself does not change with the passage of time but the agent s performance score does, then we say the environment is semidynamic.

Example.

- Taxi driving is dynamic
- Chess, when played with a clock, is semidynamic
- Crossword puzzles are static

Discrete vs. continuous

The discrete/continuous distinction applies to the state of the environment, to the way time is handled, and to the percepts and actions of the agent.

Example.

- Chess environment has a finite number of distinct states $\leadsto$ a discrete set of percepts and actions
- Taxi driving is a continuous-state and continuous-time problem
- Input from digital cameras is discrete, strictly speaking, but is typically treated as representing continuously varying intensities and locations.

Single agent vs. multiagent

Simple?
An agent solving a crossword puzzle by itself is clearly in a single-agent environment, whereas an agent playing chess is in a two- agent environment

Example.

- Chess is a competitive multiagent environment
- Taxi driving environment: avoid collisions maximizes the performance measure of all agents $\rightsquigarrow$ it is a partially cooperative multiagent environment.

Environment types

A task environment is effectively fully observable if the sensors detect all aspects that are relevant to the choice of action; relevance, in turn, depends on the performance measure.

	Solitaire	Backgammon	Internet shopping	Taxi
Observable?				
Deterministic?				
Episodic?				
Static?				
Discrete?				
Single-agent?				

Environment types

A task environment is deterministic if the next state of the environment is completely determined by the current state and the action executed by the agent.

	Solitaire	Backgammon	Internet shopping	Taxi
Observable?	Yes	Yes	No	No
Deterministic?				
Episodic?				
Static?				
Discrete?				
Single-agent?				

Environment types

In an episodic task environment, the agent s experience is divided into atomic episodes. In each episode the agent receives a percept and then performs a single action. Crucially, the next episode does not depend on the actions taken in previous episodes. Many classification tasks are episodic.

	Solitaire	Backgammon	Internet shopping	Taxi
Observable?	Yes	Yes	No	No
Deterministic?	Yes	No	Partly	No
Episodic?				
Static?				
Discrete?				
Single-agent?				

Environment types

If the environment can change while an agent is deliberating, then we say the environment is dynamic for that agent; otherwise, it is static.

	Solitaire	Backgammon	Internet shopping	Taxi
Observable?	Yes	Yes	No	No
Deterministic?	Yes	No	Partly	No
Episodic?	No	No	No	No
Static?				
Discrete?				
Single-agent?				

Environment types

The discrete/continuous distinction applies to the state of the environment, to the way time is handled, and to the percepts and actions of the agent.

	Solitaire	Backgammon	Internet shopping	Taxi
Observable?	Yes	Yes	No	No
Deterministic?	Yes	No	Partly	No
Episodic?	No	No	No	No
Static?	Yes	Semi	Semi	No
Discrete?				
Single-agent?				

Environment types

	Solitaire	Backgammon	Internet shopping	Taxi
Observable?	Yes	Yes	No	No
Deterministic?	Yes	No	Partly	No
Episodic?	No	No	No	No
Static?	Yes	Semi	Semi	No
Discrete?	Yes	Yes	Yes	No
Single-agent?				

Environment types

	Solitaire	Backgammon	Internet shopping	Taxi
Observable?	Yes	Yes	No	No
Deterministic?	Yes	No	Partly	No
Episodic?	No	No	No	No
Static?	Yes	Semi	Semi	No
Discrete?	Yes	Yes	Yes	No
Single-agent?	Yes	No	Yes (except auctions)	No

The environment type largely determines the agent design

The real world is (of course) partially observable, stochastic, sequential, dynamic, continuous, multi-agent

Agent Types

Agent types

Four basic types in order of increasing generality:

- simple reflex agents
- reflex agents with state
- goal-based agents
- utility-based agents

All these can be turned into learning agents

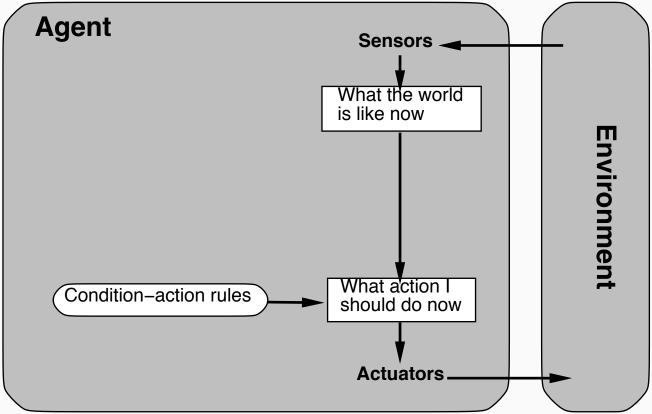
Simple reflex agents

The simplest kind of agent is the simple reflex agent. These agents select actions on the basis of the current percept, ignoring the rest of the percept history.

condition action rule
if car-in-front-is-braking then initiate-braking.

Simple reflex agents (Cont'd)

A simple reflex agent acts according to a rule whose condition matches the current state, as defined by the percept.



Example

function $\textsc{Reflex-Vacuum-Agent}[\textit{location}, \textit{status}]$

if *status* $=$ *Dirty* **then** return *Suck*

else if *location* $=$ *A* then return *Right*

else if *location* $=$ *B* then return *Left*

Reflex agents with state

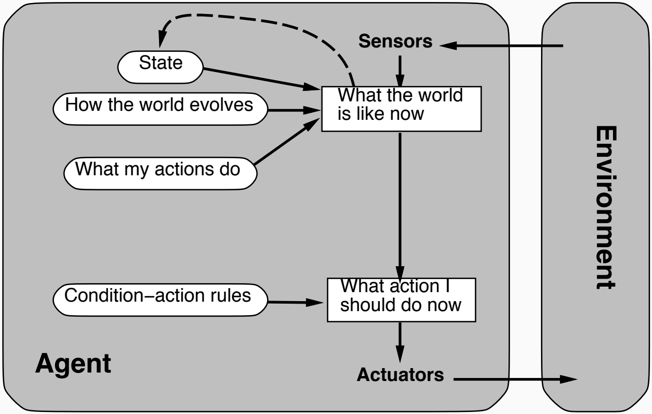
The most effective way to handle partial observability is for the agent to keep track of the part of the world it can t see now.

That is, the agent should maintain some sort of internal state that depends on the percept history and thereby reflects at least some of the unobserved aspects of the current state.

Braking problem. The internal state is not too extensive just the previous frame from the camera, allowing the agent to detect when two red lights at the edge of the vehicle go on or off simultaneously.

Reflex agents with state (Cont'd)

A model-based reflex agent keeps track of the current state of the world, using an internal model. It then chooses an action in the same way as the reflex agent.



Example

function $\textsc{Reflex-Vacuum-Agent}[\mathit{location}, \mathit{status}]$
static: *last_A*, *last_B*, numbers, initially ∞
if *status* = *Dirty* then ...

Goal-based agents

Knowing something about the current state of the environment is not always enough to decide what to do.

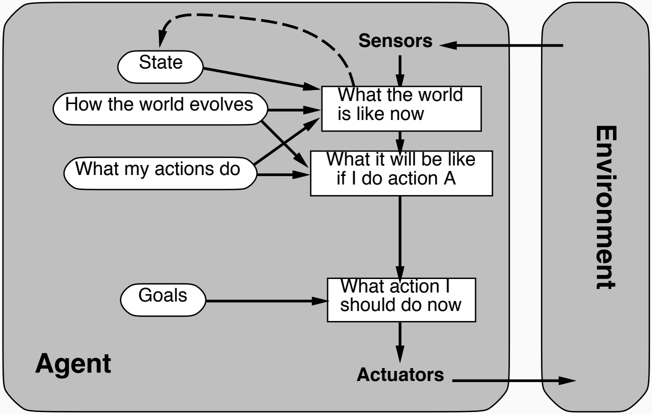
For example, at a road junction, the taxi can turn left, turn right, or go straight on.

The correct decision depends on where the taxi is trying to get to. In other words, as well as a current state description, the agent needs some sort of goal information that describes situations that are desirable for example, being at the passenger s destination.

Search and planning are the subfields of AI devoted to finding action sequences that achieve the agent s goals.

Goal-based agents (Cont'd)

A goal-based agent model keeps track of the world state as well as a set of goals it is trying to achieve, and chooses an action that will lead to the achievement of its goals.



Utility-based agents

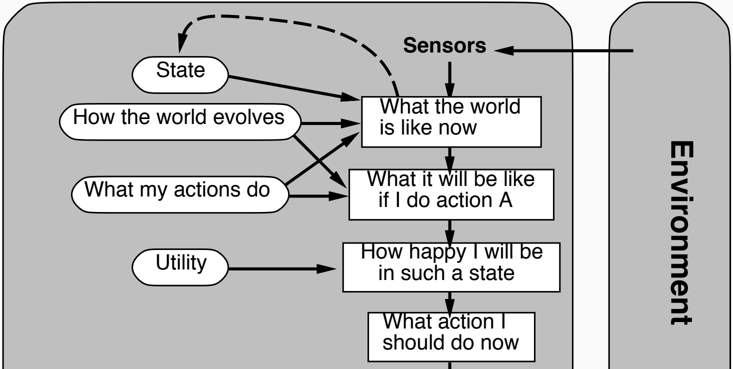
Goals alone are not enough to generate high-quality behavior in most environments.

many action sequences will get the taxi to its destination (thereby achieving the goal) but some are quicker, safer, more reliable, or cheaper than others

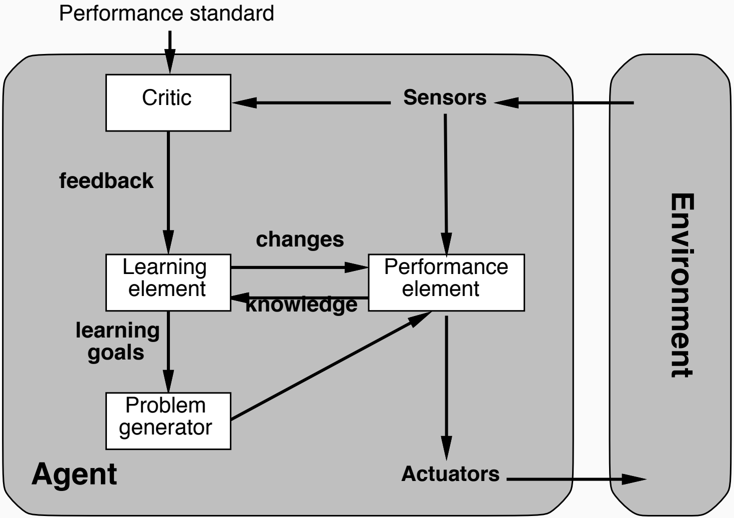
A more general performance measure should allow a comparison of different world states according to exactly how "happy" they would make the agent $\rightsquigarrow$ utility

Utility-based agents (Cont'd)

A model-based, utility-based agent. It uses a model of the world, along with a utility function that measures its preferences among states of the world. Then it chooses the action that leads to the best expected utility, where expected utility is computed by averaging over all possible outcome states, weighted by the probability of the outcome.



Learning agents



Summary

Agents interact with environments through actuators and sensors

The agent function describes what the agent does in all circumstances

The performance measure evaluates the environment sequence

A perfectly rational agent maximizes expected performance

Agent programs implement (some) agent functions

PEAS descriptions define task environments

Environments are categorized along several dimensions: observable? deterministic? episodic? static? discrete? single-agent?

Several basic agent architectures exist: reflex, reflex with state, goal-based, utility-based

Reading and Suggested Exercises

- Chapter 2
- Exercises 2.3, 2.4, 2.11

Exercises to be submitted

Exercise 4. An agent is placed in an environment containing two slot machines, each of which costs $1 to play. The expected payoffs are c and d and the agent can play each machine as many times as it likes. Describe, in qualitative terms, how a rational agent should behave in each of the following cases:

(i) The agent knows that $c = 2$ and $d = 0.75$.

(ii) The agent knows that $c = 2$ but d is unknown.

(iii) The agent knows that $c = 0.995$ but d is unknown.

(iv) c and d are unknown.

(*Hint*: Remember your PEAS! Not everything is fully specified in the question; you need to fill in the missing specifications.)

Exercises to be submitted (Cont'd)

Exercise 5

Give the initial state, goal test, successor function, and cost function for each of the following. Choose a formulation that is precise enough to be implemented. You are not required to supply the solutions!

- ***(i)*** You have to color a planar map using only three colors, with no two adjacent regions having the same color.
- ***(ii)*** A 3-foot-tall monkey is in a room where some bananas are suspended from the 8-foot ceiling. He would like to get the bananas. The room contains two stackable, movable, climbable 3-foot high crates.

Exercises to be submitted (Cont'd)

Exercise 6

For each of the following agents, develop a PEAS description of the task environment (including a characterization of the environment along the six dimensions given in lecture):

(i) Robot table-tennis player.

(ii) Medical diagnosis

(iii) Poker

(iii) Autonomous Mars rover.