Artificial Intelligence Programming

Agents and Environments

Chris Brooks

Department of Computer Science
University of San Francisco

Overview

- What makes an agent?
- Defining an environment
- Types of agent programs

Overview

- What makes an agent?
- Defining an environment
- Types of agent programs

Intelligent Agents

- The idea of developing *intelligent agents* has been a unifying one for AI.
  - Previously, subdisciplines were fairly balkanized.
  - Learning, knowledge representation, search, vision, etc.
- Agent programs that may require several of these abilities.

What is an agent?

- There are lots of potential definitions ...
- R & N: An agent is anything that can be viewed as perceiving its environment through sensors and acting on that environment through actuators.
- Woolridge: An agent is a computer system that is *situated in an environment* and is capable of *autonomous action*.

Qualities of an agent

- Autonomy
- Adaptation
- Goal-directed behavior
- Has "beliefs" and "intentions"
- Proactive
- Situated within an environment

Autonomy

- Autonomy is a quality often attributed to agents.
- An autonomous agent is able to rely on its percepts and past experience to make decisions, rather than asking a human for help.
- This is a thorny area - most agents will not have complete autonomy.
  - When might we not want an agent to have complete autonomy?
- Challenge: Designing an agent that can reason about its own autonomy and know when to ask for help.

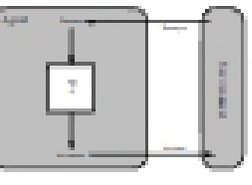
Agent-oriented Programming

- We can also think of agents as a programming paradigm.
  - The next logical step after objects.
  - "Objects do it for free, agents do it for money."
- Objects are **receivers** of actions, agents are actors.
- It's less useful to think of agent as an *objective* label than as a *subjective* description.
- Agency is a useful abstraction for us as programmers.
  - Allows us to think about a program at a higher level.
- Treat something as an agent if that helps to understand, predict, or explain its behavior.
  - Thermostats as agents

Usefulness of the agent metaphor

- Why bother with all this? We already know how to write programs.
- Agents tend to be *open-ended* programs
  - Difficult to specify all cases in advance.
  - Instead, write programs that can work with a wide set of cases.
  - Separate out the knowledge from the reasoning mechanism.
- It's helpful to talk about them *as if* they were intelligent.
  - "The robot wants to find the power supply."
  - "The server believes that the client has reset."
- This assigning of mental states to programs is called the *intentional stance*.

Agents and Environments



- One shift from other courses: we'll think explicitly about an agent's *environment* and how that affects execution.
- Percepts: Information delivered to an agent's sensors. (light, sound, EM waves, signals)
- Sensors: An agent's mechanisms for gathering data about its environment. (eyes, ears, photoelectric cells, ...)
- Actuators: An agent's mechanisms for affecting its environment. (Wheels, arms, radios, lights, etc)
- Actions: Actual changes to the environment. (turning, rolling)

Agent Programs

- We can describe our agent's behavior as a function F:
- Action = F(current-percept, percept-history).
- Maps a percept sequence to an action.
- Actually implementing this function is the work of an *agent program*.
  - That's what we'll spend most of our time on.

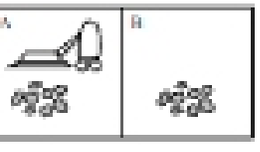
Example: Vacuum-cleaner World

- Robotic vacuum cleaners are actually on the market.

- $150 at Amazon
- A **reflex** agent (mostly)

Example: Vacuum-cleaner World

- Let's start with a very simple approximation.
- Two rooms, A and B. Each room can be either clean or dirty.
- This is the agent's **environment**.



- Sensors: Dirt sensor, location.
- Actuators: Vacuum, wheels.
- Percepts: Clean, Dirty
- Actions: Move left, move right, suck, do nothing.

Example: Vacuum-cleaner World

- In this simple world, we could list all the possible percept sequences and associated actions.
- This is known as a table-based or lookup agent.
- Question: How do we fill in the best action for each percept sequence?
- Great for simple worlds, but doesn't scale.
- We need a more compact representation for this table.

Rationality

- Roughly, rationality means "doing the right thing"
- More precision is needed - what is "the right thing"?
- We need a definition of success.
- Begin with a performance measure
  - This is a condition or state of the world we'd like the agent to achieve.
  - "Both rooms are clean." (perhaps more criteria, such as minimizing time, power consumed, or number of actions taken)
  - We might prefer a scalar measure or a boolean condition.

Rationality

- Notice that this is a specification of an *outcome*, rather than how an agent should behave.
- A rational action is one that tries to maximize an agent's performance measure, given its percepts and actions.
- R & N: Rational agents: 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.

Rationality

- "expected" vs. actual. We don't require that our agent be able to predict the future, or predict unlikely events.
- Information gathering might also be a rational action.
  - Crossing the street without looking is irrational
- Rational agents must be able to *learn* (except in very simple, well-understood environments).
  - Learning is defined as improving an agent's performance.
  - This could mean reducing uncertainty, or taking observations into account.

Overview

- What makes an agent?
- Defining an environment
- Types of agent programs