January 10, 2024 Notes

Intelligent (Autonomous) Agents

Agent

An agent is just something that acts (from the Latin agere, to do).

Of course, we would prefer "acting" to be:

- autonomous
- situated in some environment (that could be really complex)
- adaptive
- create and goal-oriented

Rational Agent

A rational agent is one that acts to achieve the best outcome, or when there is uncertainty, the best expected outcome[1].

AI: Constructing Agents

You can say that: AI is focused on the study and construction of agents that do the right thing.

Percepts and Percept Sequences

Percept – content/information that agent's sensors are perceiving / capturing currently

Percept Sequence – a complete history of everything that agent has ever perceived

- any practical issues that you can see here?
- what can a percept sequence be used for?

[1] no worries, we will make it a little less vague soon

Percepts, Knowledge, Actions, States

- Agent's choice of action / decision at any given moment:
  - CAN depend on:
    - built-in knowledge
    - entire percept sequence
  - CANNOT depend anything it hasn't perceived
- Agent's action CAN change the environment state

Knowledge is power, right?

Agent Function/Program

Specifying an action choice for every possible percept sequence would define an agent

- Action <-> percept sequence mapping IS the agent function.
- Agent function describes agent behavior.
- Agent function is an abstract concept.
- Agent program implements agent function.

Vacuum Cleaner Agent Example

Algorithm 1.1 Vacuum Cleaner Agent Example

```
function TABLE-DRIVEN-AGENT(percept) returns an action
    percepts: a sequence, initially empty
    table, a table of actions, indexed by percept sequences, initially fully specified
    append percept to the end of percepts
    $action \leftarrow$ LOOKUP(percepts, table)
    return action
end 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
    end if
end function
```

Actions Have Consequences

- An agent can act upon its environment, but how do we know if the end result is “right”?
- After all, actions have consequences: either good or bad.
- Recall that agent actions change environment state!
- If state changes are desirable, and agent performs well.
- Performance measure evaluates state changes.

Performance Measure

A Tip

It is better to design performance measures according to what one actually wants to be achieved in the environment, rather than according to how one thinks the agent should behave.

A Warning

If it is difficult to specify the performance measure, agents may end up optimizing a wrong objective. Handle uncertainty well in such cases.

Rationality

Rational decisions at the moment depend on:

- The performance measure that defines success criteria
- The agent’s prior knowledge of the environment
- The actions that the agent can perform
- The agent’s percept sequence so far

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.

Rationality in Reality

- An omniscient agent will ALWAYS know the final outcome of its action. Impossible in reality. That would be perfection.
- Rationality maximizes what is EXPECTED to happen.
- Perfection maximizes what WILL happen.
- Performance can be improved by information gathering and learning.

Designing the Agent for the Task

Analyze the Problem

Task Environment — PEAS

In order to start the agent design process, we need to specify/define:

- The Performance measure
- The Environment in which the agent will operate
- The Actuators that the agent will use to affect the environment
- The Sensors

Task Environment Properties

Key dimensions by which task environments can be categorized:

- Fully vs partially observable (can be unobservable too)
- Single agent vs multi-agent
  - multi-agent: competitive vs. co-operative
- Deterministic vs. non-deterministic (stochastic)
- Episodic vs. sequential
  - Sequential requires planning ahead
- Static vs. dynamic
- Discrete vs. continuous
- Known vs. unknown (to the agent)

Select Agent Architecture

Agent = Architecture + Program

Typical Agent Architectures

- Simple reflex agent.
- Model-based reflex agent.
- Goal-based reflex agent.
- Utility-based reflex agent.

Select Internal Representations