인공지능
2차시: Intelligent Agents

서울대학교 컴퓨터공학부
담당 교수: 장병탁
Outline (Lecture 2)

» 2.1 Agents and Environments ...................................................... 3
» 2.2 Good Behavior: Rationality .................................................... 5
» 2.3 The Nature of Environments ................................................... 6
» 2.4 The Structure of Agents .......................................................... 11
» Summary ....................................................................................... 19
» Homework ..................................................................................... 20
1) Agent

- An agent is anything that can be viewed as perceiving its environment through sensors and acting upon that environment through actuators.
- Agents include humans, robots, softbots, thermostats, etc.
- The agent function maps from percept histories to actions:

\[ f : \mathcal{P}^* \rightarrow \mathcal{A} \]

- The agent program runs on the physical architecture to produce \( f \).
2) Example: vacuum-cleaner world

A vacuum-cleaner world with just two locations

Partial tabulation of a simple agent function

<table>
<thead>
<tr>
<th>Percept sequence</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>[A, clean]</td>
<td>Right</td>
</tr>
<tr>
<td>[A, dirty]</td>
<td>Suck</td>
</tr>
<tr>
<td>[B, clean]</td>
<td>Left</td>
</tr>
<tr>
<td>[B, dirty]</td>
<td>Suck</td>
</tr>
<tr>
<td>[A, clean], [A, clean]</td>
<td>Right</td>
</tr>
<tr>
<td>[A, clean], [A, dirty]</td>
<td>Suck</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>[A, clean], [A, clean], [A, clean]</td>
<td>Right</td>
</tr>
<tr>
<td>[A, clean], [A, clean], [A, dirty]</td>
<td>Suck</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>
### 2.2 Good Behavior: Rationality

**The concept of rationality**

- **A rational agent** chooses whichever action **maximizes the expected value** of the performance measure given the percept sequence to date.
- Rationality does not require omniscience.
  - An omniscient agent knows the actual outcome of its actions and can act accordingly.
  - Rational choice depends only on the percept sequence to date.
- **A rational agent should be able to explore**
  - Doing actions in order to modify future percepts (i.e. information gathering) is an important part of rationality. Exploration.
- **A rational agent should learn** as much as possible from what it perceives.
- **A rational agent should be autonomous**
  - It should learn what it can to compensate for partial or incorrect prior knowledge.
1) PEAS

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

- Performance
- Environment
- Actuators
- Sensors
PEAS of automated taxi

» Performance
  » Safety, destination, profits, legality, comfort, ...

» Environment
  » US streets/freeways, traffic, pedestrians,

» Actuators
  » Steering, accelerator, brake, horn, speaker/display,

» Sensors
  » Video, accelerometers, gauges, engine sensors, keyboard, GPS, …
2.3 The Nature of Environments (3/5)

PEAS of internet shopping agent

» **Performance**
  - 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)
2.3 The Nature of Environments (4/5)

2) Properties of task environments

» Fully Observable vs. Partially Observable
» Deterministic vs. Stochastic
» Episodic vs. Sequential
» Static vs. Dynamic
» Discrete vs. Continuous
» Single-agent vs. Multi-agent

### 3) Examples of environment types

<table>
<thead>
<tr>
<th></th>
<th>Solitaire</th>
<th>Backgammon</th>
<th>Internet Shopping</th>
<th>Taxi</th>
</tr>
</thead>
<tbody>
<tr>
<td>Observable?</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Deterministic?</td>
<td>Yes</td>
<td>No</td>
<td>Partly</td>
<td>No</td>
</tr>
<tr>
<td>Episodic?</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Static?</td>
<td>Yes</td>
<td>Semi</td>
<td>Semi</td>
<td>No</td>
</tr>
<tr>
<td>Discrete?</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes (except auctions)</td>
<td>No</td>
</tr>
<tr>
<td>Single-agent?</td>
<td>Yes</td>
<td>No</td>
<td>Yes (except auctions)</td>
<td>No</td>
</tr>
</tbody>
</table>

The environment type largely determines the agent design.
Agent types

Four basic types in order of increasing generality

- Simple reflex agents
- Model-based reflex agents
- Goal-based agents
- Utility-based agents

All these can be turned into learning agents
2.4 The Structure of Agents (2/8)

1) Simple reflex agents

![Diagram of a simple reflex agent](image)

```
function SIMPLE-REFLEX-AGENT(percept) returns an action
  persistent: rules, a set of condition-action rules
  state ← INTERPRET-INPUT(percept)
  rule ← RULE-MATCH(state, rules)
  action ← rule.ACTION
  return action
```

**Figure 2.10** A simple reflex agent. It acts according to a rule whose condition matches the current state, as defined by the percept.

*Citation*

2) Model-based reflex agents

![Diagram showing the structure of a model-based reflex agent]

- **State**: Represents the current state of the world.
- **Sensors**: Collect information about the world.
- **Condition-action rules**: Determine the actions based on the current state.
- **Actuators**: Execute actions.
- **Environment**: Interaction with the world.
- **What the world is like now**: The current state.
- **What action I should do now**: Based on the current state and condition-action rules.

2.4 The Structure of Agents (4/8)

function MODEL-BASED-REFLEX-AGENT(percept) returns an action

persistent: state, the agent’s current conception of the world state
model, a description of how the next state depends on current state and action
rules, a set of condition–action rules
action, the most recent action, initially none

state ← UPDATE-STATE(state, action, percept, model)
rule ← RULE-MATCH(state, rules)
action ← rule.ACTION
return action

Figure 2.12 A model-based reflex agent. It 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.
3) Goal-based agents

4) Utility-based agents

5) Learning agents

Three ways to represent states and transitions

(a) Atomic

(b) Factored

(b) Structured

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:

Several basic agent architectures exist:
- Reflex, model-based, goal-based, utility-based, learning agents.
Homework

Exercises

- 2.1
- 2.4
- 2.5
- 2.8
출처

사진


