CSE 4705 - Artificial Intelligence

Introduction: What is AI? and Types of Intelligent Agents

Joe Johnson M.S., Ph.D., A.S.A.

Department of Computer Science and Engineering
University of Connecticut

UCONN | UNIVERSITY OF CONNECTICUT

Table of Contents

Table of Contents

Background

- Consider for a moment, a tree.
  - In particular, acquiring a ring on its trunk each year of its life.
  - If we cut or chop a tree down, we can count the rings to see how old the tree was.
  - Taken further, we can cut down a tree nearby and then reliably determine the difference in the ages of the trees by taking the difference between their numbers of rings.
  - Would we call this reliable ring-age-acquisition behavior intelligent?
    - One argument in favor of this being intelligent behavior: the reliability of this data that is, the tree is virtually never wrong (for the purpose of this discussion) - it never fails to create a ring on its trunk each and every season.
    - So is this tree intelligent, then? And if so, is it sufficient to say that any computation performed reliably constitutes intelligent behavior?

Background

- Now, consider a calculator.
    - Certainly, under normal circumstances, a calculator's computations are reliable.
    - Can we say, then, that the calculator is intelligent?
    - We know on some intuitive level that the answer to both of these questions - for both the tree and the calculator - that the answer is: 'No - these are not examples of intelligent behavior.' Why?
    - Maybe it's because the entities in question don't actually "know" anything while they do these computations.

Background

- Now, then, let's consider a database.
  - Suppose we have a database storing grades and course information for all students at UConn.
  - We might say this database "knows stuff".
  - Is it intelligent?
  - Here again, our answer should be no.
    - So, knowing stuff by itself does not constitute intelligence any more than reliable computations do.

Background

- Now, let's consider a squirrel building itself a nest in a tree.
  - Accomplishing this goal entails the following:
    - Seek out and find materials for the nest. (branches of the proper size, leaves, etc.)
    - Assemble these components into a nest that provides sufficient support, shelter, and protection.
  - This sounds more impressive than the other two situations we've looked at, but the squirrel does these things by instinct. It's not self-aware of the tasks it's completing. .
  - Would we call this behavior intelligent? Yes.
    - The squirrel builds a nest to achieve a *goal*.

Defintion - Artificial Intelligence (AI)

Definition

Artificial Intelligence or AI is the ability to pursue goals in the face of obstacles by making decisions according to rational truth-obeying rules.

Stephen Pinker - How the Mind Works 1997, paraphrased

Areas of AI

- Areas of AI:
  - Perception (vision, speech recognition)
  - Learning
  - Knowledge Representation
  - Reasoning (inference)
  - Planning
  - Natural Language Processing

Strong vs. Weak AI

- Strong vs. Weak AI:
  - Weak AI: emphasizes performance, without prioritizing the emulation of human cognition.
    - Think Watson, Deep Blue, Alexa, Self-Driving Car
  - Strong AI: attempts to emulate human cognition.

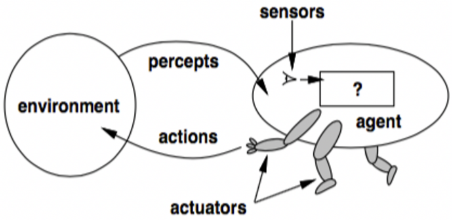
Rational Behavior

- Rational Behavior
  - is behavior that is expected to optimize the likelihood of achieving some goal, given the available information.
  - "Doing the right thing"
  - Does not necessarily involve *thinking* (remember the squirrel?) but if there is any thinking involved, it should be in the service of rational action.

Table of Contents

Intelligent Agent

- Intelligent Agent



- Features:
  - Sensors - detect, take in percepts from the environment.
  - Actuators - carry out actions to change the agent's state in its environment.
  - Knowledge - mental map of its problem domain upon which it can base decisions.
  - Reasoning - make inferences about its environment, make decisions, and develop a plan of action toward a goal.

Intelligent Agent

- Goal of this course:
  - Learn the techniques for building intelligent agents.

Agent Function

Agent function:

- Mapping of percept histories to actions.
  - $f : P* \rightarrow A$
- For any given class of environments and tasks, there is a large (perhaps infinite) set of agent functions.
- We seek the agent function which maximizes the likelihood of achieving the goal. This function defines rational behavior for the problem domain.
- Caveat: Computational limitations make perfect rationality unachievable.
  - Thus, we design the best program for the given machine resources.

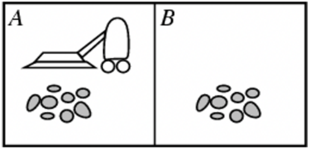
Agent Program

Agent Program:

- implementation of the agent function
- Agent = Architecture + Agent Program

Agent Program - Example: Vacuum Cleaner Agent

Agent Program - Example: Vacuum Cleaner Agent



- Percepts: location and contents, e.g., [A, Dirty]
- Actions: Left, Right, Suck, NoOp

Agent Program - Example: Vacuum Cleaner Agent

Vacuum-Cleaner Agent Function: Table-Lookup Algorithm:

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*
...	

Agent Program - Example: Vacuum Cleaner Agent

- Suppose P is the number of possible percepts.
- Suppose T is the number of percepts the agent will receive.
- Size of table $= P^T$
  - Grow exponentially with the lifetime of the agent.
  - Computationally intractable.

Agent Program - Example: Vacuum Cleaner Agent

Table-Lookup Algorithm:

- Drawbacks:
  - Huge table
  - Take a long time to build the table
  - No autonomy
  - Even with learning, need a long time to learn the table entries
- Advantages:
  - This agent does actually accomplish what we want, albeit at the expense of a large table.
- Need to be able to achieve the same goal with a much smaller program than this approach requires.

Agent Program - Example: Vacuum Cleaner Agent

Vacuum-Cleaner Agent Function: Table-Lookup Algorithm:

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*
...	

Agent Program - Example: Vacuum Cleaner Agent

Vacuum-Cleaner Agent Function: Table-Lookup Algorithm:

function TABLE-DRIVEN-AGENT(*percept*) **returns** an action
  persistent: *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 ← LOOKUP(*percepts*, *table*)
  return *action*

Figure 2.7 The TABLE-DRIVEN-AGENT program is invoked for each new percept and returns an action each time. It retains the complete percept sequence in memory.

Agent Program - Example: Vacuum Cleaner Agent

Vacuum-Cleaner Agent Function - *Improved*:

```
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
```

Table of Contents

PEAS

- PEAS: Performance measure, Environment, Actuators, Sensors
- Must first specify the setting for intelligent agent design.

Table of Contents

PEAS Example 1 - Automated Taxi Driver

PEAS Example 1 - Automated Taxi Driver

- Performance measure: Safe, fast, legal, comfortable trip, maximize profits
- Environment: Roads, other traffic, pedestrians, customers
- Actuators: Steering wheel, accelerator, brake, signal, horn
- Sensors: Cameras, sonar, speedometer, GPS, odometer, engine sensors, keyboard

PEAS Example 2 - Part Picking Robot

PEAS Example 2 - Part Picking Robot

- Performance measure: Percentage of parts in correct bins
- Environment: Conveyor belt with parts, bins
- Actuators: Jointed arm and hand
- Sensors: Camera, joint angle sensors

Table of Contents

Environment Types - Fully Observable

Fully Observable: (vs. Partially Observable:)

- Agent's sensors give it access to the complete state of the environment at each point in time.

Environment Types - Deterministic

Deterministic: (vs. Stochastic:)

- The next state of the environment is completely determined by the current state and the action executed by the agent.
- If the environment is deterministic except for the actions of other agents, then the environment is *strategic*.

Environment Types - Episodic

Episodic: (vs. Sequential:)

- The agent's experience is divided into atomic "episodes" (each episode consists of the agent perceiving and then performing a single action), and the choice of action in each episode depends only on the episode itself.

Environment Types - Static

Static (vs. dynamic):

- The environment is unchanged while an agent is deliberating.
- The environment is semi-dynamic if the environment itself does not change with the passage of time but the agent's performance score does.

Environment Types - Discrete

Discrete (vs. Continuous):

- The environment is characterized by a limited number of distinct, clearly defined percepts and actions.

Environment Types - Single Agent

Single Agent (vs. Multi-Agent):

- An agent operating by itself in the environment.

PEAS Examples - Comparisons

	Chess with a clock	Chess without a clock	Taxi driving
Fully observable	Yes	Yes	No
Deterministic	Strategic	Strategic	No
Episodic	No	No	No
Static	Semi	Yes	No
Discrete	Yes	Yes	No
Single agent	No	No	No

- The environment type largely determines the agent design
- The real world is: partially observable, stochastic, sequential, dynamic, continuous, multi-agent.

Table of Contents

Agent Types

Agent Types:

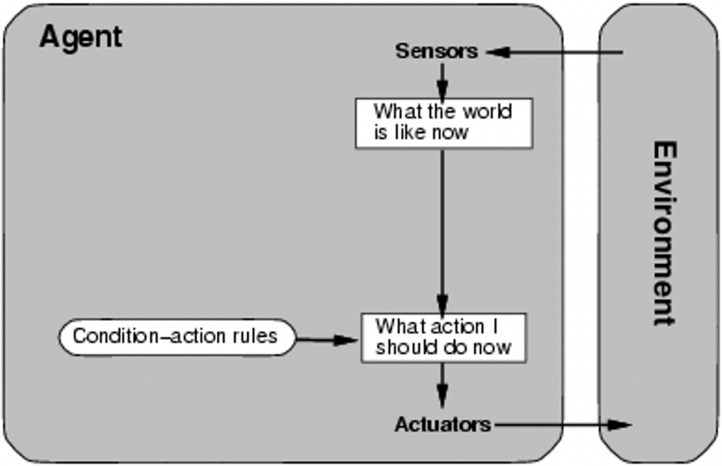
- Simple Reflex Agent
- Model-Based Reflex Agent
- Model-Based Goal-Based Agent
- Model-Based Utility-Based Agent
- Learning Agent

Simple Reflex Agent

Simple Reflex Agent:

- Works only if the correct decision can be made on the basis of only the current percept (ignores the prior percepts).
- Requires the environment to be fully observable.
- Example: Vacuum cleaner agent

Simple Reflex Agent



Simple Reflex Agent

function SIMPLE-REFLEX-AGENT(*percept*) **returns** an action
  persistent: *rules*, a set of condition–action rules

  $state \leftarrow$ INTERPRET-INPUT($percept$)
  $rule \leftarrow$ RULE-MATCH($state, rules$)
  $action \leftarrow rule$.ACTION
  return *action*

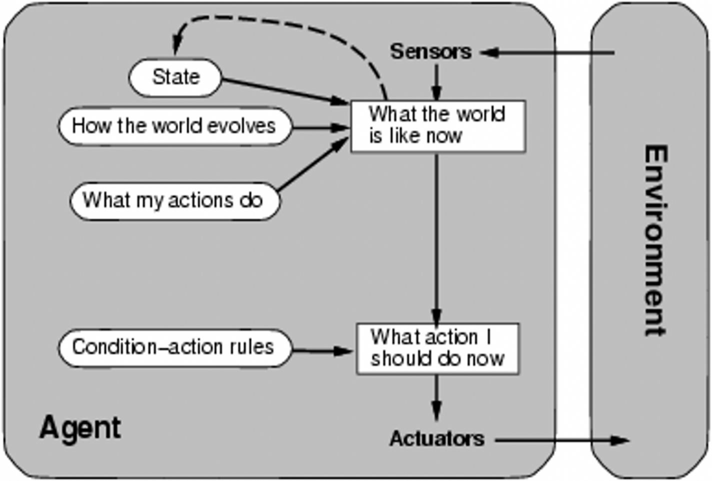
Figure 2.10 A simple reflex agent. It acts according to a rule whose condition matches the current state, as defined by the percept.

Model-Based Reflex Agent

Model-Based Reflex Agent:

- Consider a domain in which we have only partial observability.
- Agent must keep track of the world it cannot see right now.
- Agent maintains an internal state based on percept history (and thus reflects unobserved aspects of current state).
- Example – tail lights on a car in front going from off to on indicates brakes of front car have just been applied.

Model-Based Reflex Agent



Model-Based Reflex Agent

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.

Model-Based Reflex Agent

Model-Based Reflex Agent:

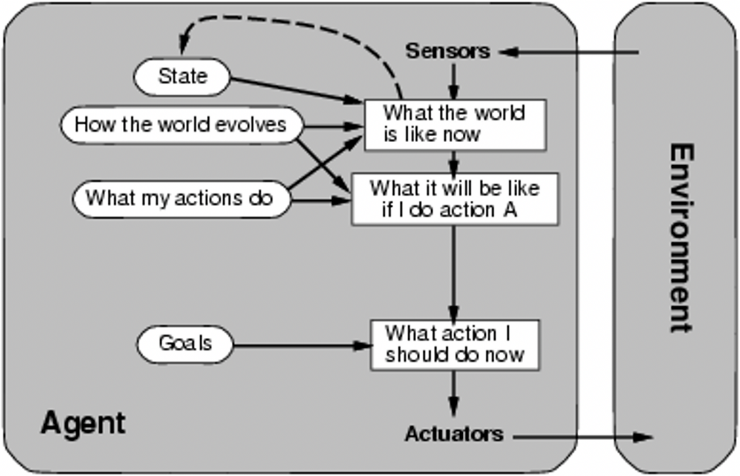
- Internals of the agent must also understand two important things:
  - How the world works independent of the agent's actions
    - Example – car in front applies brakes make the car slow down and all things being equal, causes the car to get closer
  - How the world changes as a result of the agent's actions.
    - Example - turn steering wheel to the right make the car go right

Model-Based, Goal-Based Agent

Model-Based, Goal-Based Agent:

- Sometimes having an understanding of state isn't sufficient to know what to do next.
  - Example: at a road intersection, a taxi could turn left, right, or go straight. Which choice should it make?
- Keeping track of goals can provide the necessary information about what action to take.
  - Example: The passenger's desired destination.

Model-Based, Goal-Based Agent

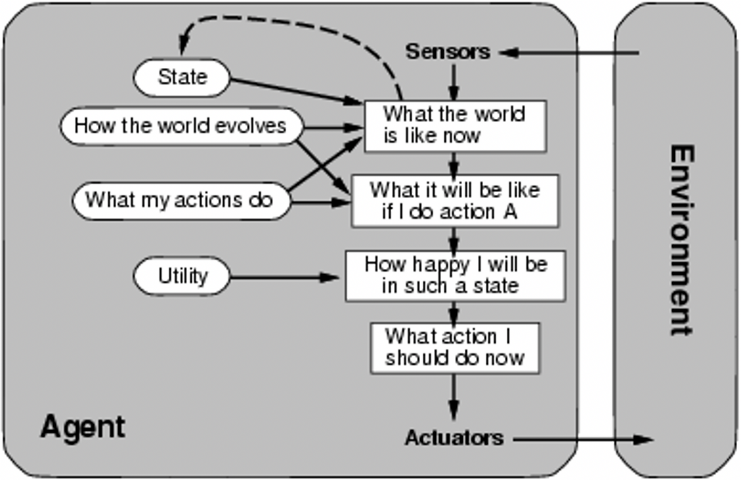


Model-Based, Utility-Based Agent

Model-Based, Utility-Based Agent:

- Sometimes goals are not sufficient if the possible subsequent states due to actions cannot be evaluated relative to a longer term goal.
    - Example: Many action sequences will get a taxi to its destination, but some are quicker, safer, more reliable, or cheapter than others.
- Assess the utility of each subsequent state relative to the goal.
- This requires the evaluation of a utility function for each subsequent state.

Model-Based, Utility-Based Agent



Learning Agent

Learning Agent:

- How do agent programs come into being?
  - Hand code all rules... or..
  - Learn rules through experience
- Learning agent
  - Performance component – translates percept sequences to actions
  - Learning component – evaluates the resulting state of the percept, action according to a performance measure.

Learning Agent