CSCE A405/A605 (Adv) Artificial Intelligence

Intelligent Agents

Ref: Artificial Intelligence: A Modern Approach, 4th ed by Stuart Russell and Peter Norvig, chapter 2

Instructor: Masoumeh Heidari (mheidari2@Alaska.edu)

UAA College of Engineering
University of Alaska Anchorage

Outline

- What is an intelligent agent?
- The nature of environments
- The structure of agents

What is an intelligent agent?

Learning objectives

- What does an intelligent agent do?
- What are important components of an intelligent agent?
- How to evaluate the behavior of an agent?
- What is a rational agent?

Agent?

- AI can be defined as the study of intelligent agents that receive percepts from the environment and perform actions.

Agent?

Agent?

Eyes, Ears, Other organs
Agent
Percepts
?
Environment
Actions
Hands, Legs, Vocal tract, …

Agent?

Agent
Cameras, Infrared range finders, …
Percepts
?
Environment
Actions
Various motors

Agent?

File contents, network packets, human input, etc
Agent
Percepts
?
Environment
Actions
Sound generator, Display screen, …

Agent?

- The term percept refers to the content an agent's sensors are perceiving.
- Percept sequence is the complete history of everything the agent has ever perceived.

- Agent's function maps any given percept sequence to an action.

- The entire universe!
- In practice, the part of the universe our agent cares about.
- The part of the universe that affect what the agent perceives and that is affected by the agent's actions.

A simple example

A simple example

A simple example

- Which square it is in
- Whether there is dirt in the square

A world consisting of two squares that can be dirty or clean.

- Move to the right
- Move to the left
- Suck up the dirt
- Do nothing!

A simple example

Percept sequence	Action

What does it mean to do the right thing?

- What makes an agent good or bad, intelligent or stupid?
- We evaluate an agent's behavior by its consequences.
- An agent generates a sequence of actions that causes the environment to go through a sequence of states. *If the sequence is **desirable**, then the agent has performed well.*
- The notion of desirability is captured by a **performance measure** that evaluates any given sequence of environment states.

Definition of a 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.

- Is the vacuum-cleaner agent rational?

Is the vacuum-cleaner agent rational?

- **Environment**: 1) geography of the environment is known a priory but the dirt distribution and the initial location of the agent are not, 2) Clean squares stay clean and sucking cleans the current square, 3) The *Right* and *Left* actions move the agent one square except when this would take the agent outside the environment, in which case the agent remains where it is.
- The only available **actions** are *Right*, *Left*, and *Suck*
- The agent's **sensors** give it the ability to correctly perceives its location and whether that location contains dirt.

Is the vacuum-cleaner agent rational?

Case 1

Performance measure: awards one point for each clean square at each time step, over a lifetime of 1000 time steps.

Its expected performance is at least as good as any other agent's.

Case 2

Performance measure: award one point for each clean square at each time step, over a lifetime of 1000 time steps. It includes a penalty of one point for each movement.

Once all the dirt is cleaned up, the agent will oscillate needlessly back and forth.

Recap

✓What is an intelligent agent?

✓What are important components of an intelligent agent?

✓How to evaluate the behavior of an agent?

Performance measure

✓What is a rational agent?

Considering the performance measure,
its actions lead to the desirable sequence of states.

The nature of environments

Learning objectives

- What are elements of task environment?
- How to specify task environment?
- Properties of task environment

The nature of environments

- Task environment: the “problems” to which rational agents are “solutions”.

- **First step in designing an agent**: Specifying the *task environment.*

- **PEAS**: Performance, Environment, Actuators, and Sensors

Task environment for an automated taxi driver

Agent Type	Performance Measure	Environment	Actuators	Sensors
Taxi driver				

Examples of agent types and their PEAS descriptions

Medical Diagnosis Agent

- **P:** Healthy patient, reduced cost
- **E**: patient, hospital, staff
- **A**: display of questions, diagnosis, treatments
- **S**: Touchscreen/voice entry of signals and findings

Examples of agent types and their PEAS descriptions

Online Shopping Agent

- **P:** Cost, Quality, efficiency
- **E:** Websites, vendors, shippers
- **A:** Display, payment form submission
- **S:** Human input, HTML webpages

Properties of task environments

- Identify a fairly small number of dimensions along which task environments can be categorized.
  - Fully observable Vs. Partially observable
  - Single-agent Vs. Multiagent
  - Deterministic Vs. Nondeterministic
  - Episodic Vs. Sequential
  - Static Vs. Dynamic
  - Discrete Vs. Continuous
  - Known Vs. Unknown

Fully observable Vs. Partially observable

- Fully observable: an agent's sensors give it access to the <u>complete state of the environment at each point in time</u>.

Interactive English tutor

Medical diagnosis

Fully observable Vs. Partially observable

- Fully observable: an agent's sensors give it access to the <u>complete state of the environment at each point in time</u>.

Interactive English tutor	Environment: set of students, testing agency Sensors: keyboard entry, voice
Medical diagnosis	Environment: Patient, hospital, staff Sensors: Touchscreen, voice of entry

Fully observable Vs. Partially observable

- Fully observable: an agent's sensors give it access to the <u>complete state of the environment at each point in time</u>.

Interactive English tutor	Partially observable
Medical diagnosis	Partially observable

Single-agent VS. Multiagent

- Does an agent A (the taxi driver for example) have to treat an object B (another vehicle) as an agent, or can it be treated merely as an object behaving according to the laws of physics, analogous to waves at the beach or leaves blowing in the wind?

- The key distinction is whether B's behavior is best described as maximizing a performance measure whose value depends on agent A's behavior.

Single-agent VS. Multiagent

- The key distinction is whether B's behavior is best described as maximizing a performance measure whose value depends on agent A's behavior.

Interactive English tutor

Medical diagnosis

Single-agent VS. Multiagent

- The key distinction is whether B's behavior is best described as maximizing a performance measure whose value depends on agent A's behavior.

Interactive English tutor	Multiagent
Medical diagnosis	Single-agent What if the patient does not cooperate?

Deterministic Vs. Nondeterministic

- If the next state of the environment is completely determined by the current state and the action executed by the agent(s), then we say the environment is deterministic; otherwise, it is nondeterministic.

Chess with a clock	Deterministic
Medical diagnosis	Nondeterministic

Episodic VS. Sequential

- In an episodic task environment, in each episode the agent receives a percept and then performs a single action.
- 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.
- Chess and taxi driving are sequential: in both cases, short-term actions can have long-term consequences.

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.

- 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.

Medical diagnosis

Chess with a clock

Crossword puzzle

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.

- 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.

Medical diagnosis	Chess with a clock	Crossword puzzle
Dynamic	Semidynamic	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.
- Chess environment has a finite number of distinct states (excluding the clock). Chess also has a discrete set of percepts and actions.
- Taxi driving is a continuous-state and continuous-time problem: the speed and location of the taxi and of the other vehicles sweep through a range of continuous values and do so smoothly over time. Taxi-driving actions are also continuous (steering angles, etc.).

Known Vs. Unknown

- This distinction refers not to the environment itself but to the agent's (or designer's) state of knowledge about the "laws of physics" of the environment. In a known environment, the outcomes (or outcome probabilities if the environment is nondeterministic) for all actions are given.
- Known/unknown ≢ Observable/partially observable

Video game: the screen may show the entire game state but I still don't know what the buttons do until I try them.

Card game: I know the rules but am still unable *to see* the cards that have not yet been turned over

The hardest case is **partially observable**, **multiagent**, **nondeterministic**, **sequential**, **dynamic**, **continuous**, and **unknown**.

- Taxi driving: Taxi driving is hard in all these senses, except that the driver's environment is mostly known.

- But imagine driving a rented car in a new country with unfamiliar geography, different traffic laws, and nervous passengers!

This Photo by Unknown Author is licensed under CC BY

Recap

✓What are elements of task environment?

Performance, Environment, Actuators, and Sensors

✓How to specify task environment?

✓Properties of task environment

Properties	
Fully observable	Partially observable
Single-agent	Multiagent
Deterministic	Nondeterministic
Episodic	Sequential
Static	Dynamic
Discrete	Continues
Known	Unknown

Spam classifier

- P: Correct classification
- E: Text
- A: Screen display
- S: Keyboard

The structure of agents

Learning objectives

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

Some sort of computing device with physical sensors and actuators.

Take the current percept as input from the sensors and return an action to the actuators.

Agent = Architecture + Program

Agent’s program

Agent’s function

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*
$\vdots$	$\vdots$
$[A, Clean], [A, Clean], [A, Clean]$	*Right*
$[A, Clean], [A, Clean], [A, Dirty]$	*Suck*
$\vdots$	$\vdots$

The TABLE-DRIVEN-AGENT program

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*]	
⋮	

```
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.

The TABLE-DRIVEN-AGENT program

Drawbacks:

- Too big to store (e.g. look up table for chess has at least 10^{150} entries.
- Take a long time to create such a table.
- No agent could ever learn all the right table entries from its experience.
- Not adaptive to changes in the environment (i.e. Entire table needs to be updated if changes occurs).

Simple reflex agent

- Selects actions on the basis of the current percept, ignoring the rest of the percept history.

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

Its decision is based only on the *current location* and on whether that location contains dirt.

Simple reflex agent

- Selects actions on the basis of the current percept, ignoring the rest of the percept history.

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*, (	
[*A*, *Clean*], [*A*, *Clean*], [*A*, *1*	
⋮	

```
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
```

Simple reflex agent

```
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
```

Generates an abstracted description of the current state from the percept.

Returns the first rule in the set of rules that matches the given state description.

Simple reflex agent

Drawbacks:

- It works only if the correct decision can be made on the basis of just the current percept.

Model-based reflex agent

Model-based reflex agent

Model-based reflex agent

- 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.

- 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.

Model-based reflex agent

Model-based reflex agent

[Did not move, clean] [Did not move, clean]

A/B B

Right

Model-based reflex agent

- Updating this internal state information as time goes by requires two kinds of knowledge to be encoded in the agent program in some form.
  1. We need some information about how the world changes over time, which can be divided roughly into two parts:
     - The effects of the agent's actions and how the world evolves independently of the agent.

Transition Model

  2. How the state of the world is reflected in the agent's percepts

Sensor Model

Model-based reflex agent

- Together, the transition model and sensor model allow an agent to keep track of the state of the world—to the extent possible given the limitations of the agent's sensors.

- An agent that uses such models is called a model-based agent.

- 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.

Model-based reflex agent

```
function MODEL-BASED-REFLEX-AGENT(percept) returns an action
  persistent: state, the agent's current conception of the world state
              transition_model, a description of how the next state depends on
                    the current state and action
              sensor_model, a description of how the current world state is reflected
                    in the agent's percepts
              rules, a set of condition–action rules
              action, the most recent action, initially none

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

Model-based reflex agent

Drawback:

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

Goal-based agent

- As well as a current state description, the agent needs some sort of goal information that describes situations that are desirable.

- The agent program can combine this with the model to choose actions that achieve the goal.

Goal-based agent

Drawback:

Sometimes goal-based action selection is straightforward—for example, when goal satisfaction results immediately from a single action. Sometimes it will be more tricky—for example, when the agent has to consider long sequences of twists and turns in order to find a way to achieve the goal.

Utility-based agent

- Goals alone are not enough to generate high-quality behavior in most environments because goals just provide a crude binary distinction between “happy” and “unhappy” states.
- A more general performance measure should allow a comparison of different world states according to exactly how happy they would make the agent.
- An agent’s utility function is essentially an internalization of the performance measure.

Utility-based agent

- Internal utility function and the external performance measure are in agreement, an agent that chooses actions to *maximize its utility* will be *rational* according to the external *performance measure*.

- When there are *conflicting goals*, only some of which can be achieved (for example, speed and safety), the utility function specifies the appropriate *tradeoff*.

- When there are several goals that the agent can aim for, none of which can be achieved with certainty, utility provides a way in which the *likelihood of success* can be weighed against the importance of the goals.

Utility-based agent

Utility-based agent

- “Is it that simple? We just build agents that maximize expected utility, and we’re done?”
  - A utility-based agent has to model and keep track of its environment, tasks that have involved a great deal of research on perception, representation, reasoning, and learning.
  - Choosing the utility-maximizing course of action is also a difficult task, requiring ingenious algorithms.
  - Even with these algorithms, perfect rationality is usually unachievable in practice because of computational complexity.
  - Finally, all of this assumes that the designer can specify the utility function correctly!

Recap

- What is an intelligent agent?
  - What does an intelligent agent do?
  - What are important components of an intelligent agent?
  - How to evaluate the behavior of an agent?
  - What is a rational agent?

- The nature of environments
  - What are elements of task environment?
  - How to specify task environment?
  - Properties of task environment

- The structure of agents
  - Simple reflex agents
  - Model-based reflex agents
  - Goal-based agents
  - Utility-based agents