Fundamentals of Artificial Intelligence

Laboratory

Dr. Mauro Dragoni

Department of Information Engineering and Computer Science

Academic Year 2021/2022

Intro

- Schedule: **Wednesday 11:30-13:30**

- Website: http://www.maurodragoni.com/teaching/fai/ (under construction)

- Class exercises and homework

Intro

Being proactive is the best way to learn!!!

Exercise 2.1

- For each of the following assertions, say whether it is true or false and support the answer with examples or counterexamples where appropriate.
  - An agent that senses only partial information about the state cannot be perfectly rational.
  - There exist task environments in which no pure reflex agent can behave rationally.
  - There exists a task environment in which every agent is rational.
  - The input to an agent program is the same as the input to the agent function.
  - Every agent function is implementable by some program/machine combination.
  - Suppose an agent selects its action uniformly at random from the set of possible actions. There exists a deterministic task environment in which this agent is rational.
  - It is possible for a given agent to be perfectly rational in two distinct task environments.
  - Every agent is rational in an unobservable environment.
  - A perfectly rational poker-playing agent never loses.

Exercise 2.1.1

- For each of the following assertions, say whether it is true or false and support the answer with examples or counterexamples where appropriate.

  - **An agent that senses only partial information about the state cannot be perfectly rational.**

  False. Perfect rationality refers to the ability to make good decisions given the sensor information received.

Exercise 2.1.2

- For each of the following assertions, say whether it is true or false and support the answer with examples or counterexamples where appropriate.

  - **There exist task environments in which no pure reflex agent can behave rationally.**

  True. A pure reflex agent ignores previous percepts, so cannot obtain an optimal state estimate in a partially observable environment. For example, correspondence chess is played by sending moves; if the other player's move is the current percept, a reflex agent could not keep track of the board state and would have to respond to, say, "A4" in the same way regardless of the position in which it was played.

Exercise 2.1.3

- For each of the following assertions, say whether it is true or false and support the answer with examples or counterexamples where appropriate.

  - **There exists a task environment in which every agent is rational.**

  True. For example, in an environment with a single state, such that all actions have the same reward, it doesn’t matter which action is taken. More generally, any environment that is reward-invariant under permutation of the actions will satisfy this property.

Exercise 2.1.4

- For each of the following assertions, say whether it is true or false and support the answer with examples or counterexamples where appropriate.

  - **The input to an agent program is the same as the input to the agent function.**

  False. The agent function, notionally speaking, takes as input the entire percept sequence up to that point, whereas the agent program takes the current percept only.

Exercise 2.1.5

- For each of the following assertions, say whether it is true or false and support the answer with examples or counterexamples where appropriate.

  - **Every agent function is implementable by some program/machine combination.**

  False. For example, the environment may contain Turing machines and input tapes and the agent's job is to solve the halting problem; there is an agent function that specifies the right answers, but no agent program can implement it. Another example would be an agent function that requires solving intractable problem instances of arbitrary size in constant time.

Exercise 2.1.6

- For each of the following assertions, say whether it is true or false and support the answer with examples or counterexamples where appropriate.

  - **Suppose an agent selects its action uniformly at random from the set of possible actions. There exists a deterministic task environment in which this agent is rational.**

  True. This is a special case of 2.1.3; if it doesn't matter which action you take, selecting randomly is rational.

Exercise 2.1.7

- For each of the following assertions, say whether it is true or false and support the answer with examples or counterexamples where appropriate.

  - **It is possible for a given agent to be perfectly rational in two distinct task environments.**

  True. For example, we can arbitrarily modify the parts of the environment that are unreachable by any optimal policy as long as they stay unreachable.

Exercise 2.1.8

- For each of the following assertions, say whether it is true or false and support the answer with examples or counterexamples where appropriate.

  - **Every agent is rational in an unobservable environment.**

  False. Some actions are stupid, and the agent may know this if it has a model of the Environment, even if one cannot perceive the environment state.

Exercise 2.1.9

- For each of the following assertions, say whether it is true or false and support the answer with examples or counterexamples where appropriate.

  - **A perfectly rational poker-playing agent never loses.**

  False. Unless it draws the perfect hand, the agent can always lose if an opponent has better cards. This can happen for game after game. The correct statement is that the agent's expected winnings are nonnegative.

Exercise 2.2

- For each of the following activities, give a PEAS description of the task environment and characterize it in terms of their properties.
    - Playing soccer.
    - Exploring the subsurface oceans of Titan.
    - Shopping for used AI books on the Internet.
    - Playing a tennis match.
    - Practicing tennis against a wall.
    - Performing a high jump.
    - Knitting a sweater.

Exercise 2.2.1

- For each of the following activities, give a PEAS description of the task environment and characterize it in terms of their properties.
    - **Playing soccer.**

  Partially observable, stochastic, sequential, dynamic, continuous, multi-agent.

Exercise 2.2.2

- For each of the following activities, give a PEAS description of the task environment and characterize it in terms of their properties.

    - **Exploring the subsurface oceans of Titan.**

    Partially observable, stochastic, sequential, dynamic, continuous, single agent (unless there are alien life forms that are usefully modeled as agents).

Exercise 2.2.3

- For each of the following activities, give a PEAS description of the task environment and characterize it in terms of their properties.

  - **Shopping for used AI books on the Internet.**

  Partially observable, deterministic, sequential, static, discrete, single agent. This can be multi-agent and dynamic if we buy books via auction, or dynamic if we purchase on a long enough scale that book offers change.

Exercise 2.2.4

- For each of the following activities, give a PEAS description of the task environment and characterize it in terms of their properties.
  - **Playing a tennis match.**

  Fully observable, stochastic, episodic (every point is separate), dynamic, continuous, multi-agent.

Exercise 2.2.5

- For each of the following activities, give a PEAS description of the task environment and characterize it in terms of their properties.

  - **Practicing tennis against a wall.**

  Fully observable, stochastic, episodic, dynamic, continuous, single agent.

Exercise 2.2.6

- For each of the following activities, give a PEAS description of the task environment and characterize it in terms of their properties.
  - **Performing a high jump.**

Fully observable, stochastic, sequential, static, continuous, single agent.

Exercise 2.2.7

- For each of the following activities, give a PEAS description of the task environment and characterize it in terms of their properties.
  - **Knitting a sweater.**

Fully observable, deterministic, sequential, static, continuous, single agent.

Exercise 2.3

- Write pseudocode agent programs for the goal-based and utility-based agents.

Exercise 2.3

- Write pseudocode agent programs for the goal-based agents.

```
function GOAL-BASED-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
              goal, a description of the desired goal state
              plan, a sequence of actions to take, initially empty
              action, the most recent action, initially none

      state ← UPDATE-STATE (state, action, percept, model)
      if GOAL-ACHIEVED (state, goal) then return a null action
      if plan is empty then
              plan ← PLAN (state, goal, model)
              action ← FIRST (plan)
              plan ← REST (plan)
      return action
```

Exercise 2.3

- Write pseudocode agent programs for the utility-based agents.

```
function UTILITY-BASED-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
              utility-function, a description of the agent's utility function
              plan, a sequence of actions to take, initially empty
              action, the most recent action, initially none

        state ← UPDATE-STATE (state, action, percept, model)
        if plan is empty then
                plan ← PLAN (state, utility-function, model)
        action ← FIRST (plan)
        plan ← REST (plan)
        return action
```

Exercise 2.4 - Homework

- Let us examine the rationality of various vacuum-cleaner agent functions.
  - Show that the simple vacuum-cleaner agent function is rational.
  - Describe a rational agent function for the case in which each movement costs one point. Does the corresponding agent program require internal state?
  - Discuss possible agent designs for the cases in which clean squares can become dirty and the geography of the environment is unknown. Does it make sense for the agent to learn from its experience in these cases? If so, what should it learn? If not, why not?