50.021 – Artificial Intelligence

Kwan Hui

Week 01: Agents and Environment

[The following notes are compiled from various sources such as textbooks, lecture materials, Web resources and are shared for academic purposes only, intended for use by students registered for a specific course. In the interest of brevity, every source is not cited. The compiler of these notes gratefully acknowledges all such sources. ]

Due: 2nd Feb, 11:59pm

Submission: via eDimension.

Name: Sharryl Seto

Cohort: Cl03

# PEAS Description

Based on the Agent Model, an agent can be described based on the PEAS description (Performance measure, Environment, Actuators, Sensors).

**TASK:** For the following agents, define the PEAS description based on these four characteristics and state any assumptions you may have.

1. A robotic platform for autonomous cleaning of the floor within a building.

Performance measure: Percentage of floor cleaned, safety

Environment: floor of building, objects, other people

Actuators: cleaning apparatus (like mop, broom) and movement platform (like wheels)

Sensors: camera, sensors to detect environment

1. A customer service chatbot for an online retailer.

Performance measure: Percentage of happy customer reviews/feedback (assuming all reviews are real and true)

Environment: online platform, customers

Actuators: software to process customers’ replies and reply customers

Sensors: chatroom where customer interacts with chatbot

1. An automated and unmanned convenience store, like Pick&Go in SUTD.

Performance measure: Percentage of accurate items sold

Environment: store with items, customers

Actuators: automated gate with cameras

Sensors: ceiling sensors and cameras

# Environment Types

There are six characteristics (Observable, Deterministic, Episodic, Static, Discrete, Single-agent) when describing a task environment.

**TASK:** For the following activities, describe the task environment using these six characteristics and state any assumptions you may have.

1. Playing an massively multiplayer online game, such as World of Warcraft

(https://en.wikipedia.org/wiki/World of Warcraft)

Partially observable, strategic (assuming the other agents are not NPCs – for example tutorial), sequential, dynamic, continuous, multiagent.

1. Enrolling for a course/subject for a new term at SUTD

Partially observable, deterministic (assuming no admin agents are interfering with enrolment), sequential, semidynamic (assuming the enrolment slots are limited), discrete, single agent.

1. Purchasing a drink from a vending machine

Partially observable, deterministic (assuming no malfunction), sequential, static (assuming no timeout if agent deliberates too long), discrete, single agent.

# Problem Formulation

Missionaries and cannibals is a classical formal problem, and is generally stated as follows. Three missionaries and three cannibals are on one side of the river. They all need to cross in a boat that only holds two people at once. There must never be a situation where there is a group of missionaries in one place who are outnumbered by cannibals.

**TASK:** Formalise the missionaries and cannibals problem in terms of its state space, initial state, goal test, actions, and path cost.

Assuming they are going from the left to the right side of the river, and boat must have 1 person on it at all times (referenced formal problem on Google).

Let M and C be number of missionaries and cannibals respectively, that are on the left side of the river. Let B be whether the boat is on the left side of the river.

* 1. State space: all possible combinations of missionaries/cannibals/boat, represented in a vector [M,C,B]
  2. Initial state: [M,C,B] = [3,3,1].
  3. Actions: vector addition/subtraction of vector permutations allowed
     1. [1,0,1], [2,0,1] (1/2 missionaries crossing from left to right, or vice versa)
     2. [0,1,1], [0,2,1] (1/2 cannibals crossing from left to right, or vice versa)
     3. [1,1,1] (1 missionary and 1 cannibal crossing from left to right, or vice versa)
  4. Transition model: Update vector state [M,C,B]
  5. Path cost: number of actions to reach goal
  6. Goal test: compare to goal state -> [M,C,B] = [0,0,0]