Assignment-6 : Intelligent Agents

Harshvardhan Patidar
Department of Artificial Intelligence
Indian Institute of Technology Hyderabad
`ai24btech11015@iith.ac.in`

What is an Agent?

Agent is something which takes a percept(or simply input), by understanding or perceiving its environment, i.e. its surroundings, through sensors, and then accordingly acts upon the environment using actuators. Now, to act according to a specific percept, or percept sequence, there needs to be an agent function, which maps every single percept to an action. This agent function is just an abstract mathematical description, but making and storing it is a difficult task as the number of percept sequences is very big. So, the actual implementation is done in the form of an agent program. Think of an agent program as a black box, you put in some percept sequences in it, and it will throw out some series of actions to perform on the environment. All these three things combined, namely sensors, the black box, and the actuator form the agent.

Good Behaviour

For producing good behavior, we need to design rational agents. But how to know if an agent is rational or not?

Performance Measures are mostly done by the method of consequentialism : evaluating the agent's action based upon its consequences. Fitting in human goals in the agent will not always produce the desired output. There's a difference in the point of view of a human and a machine. An agent might resort to some inefficient or unethical ways to just achieve its target, rather than being efficient and ethical. As a general rule, it is better to design performance measures according to what one actually wants to achieve in the environment, rather than according to how one thinks the agent should behave.

Now we can define rational agents as the agents which try to maximize its performance measure for every percept sequence it encounters. But rationality doesn't guarantee perfection. For perfection we need to have something called Omniscience, or knowing everything. There are always chances of something very unpredictable to take place. Even humans can't always consider all such factors. By the notion of information gathering, an agent can do some actions to modify the future percepts. For information gathering, it might also do some exploration, for example in the case of vacuum-cleaning agents.

Task Environments

Now we know about almost everything that we need to know for making our rational agent. We now need to specify what will be the performance measure, environment, actuators and sensors for

our agent. All these are grouped under the heading of Task Environment. It is also referred to as **PEAS** (**P**erformance, **E**nvironment, **A**ctuators, **S**ensors) description. Listing the **PEAS** gives us an outline of what the problem actually is and what our agent should achieve.

The task environments can be categorized in several categories

- **Fully Observable** vs **Partially Observable** : In fully observable environment, an agent program has access to all the percepts which it require to make a choice for its action, where as in the partially observable environment, some data or percept is not reachable either due to lack of sensors, or inaccurate sensors.
- **Single-agent** vs **Multi-agent** : There might be some cases where multiple agents might be working together, which might be a co-operative environment or competitive environment depending on the goal of the agent.
- **Deterministic** vs **non-deterministic** : If the next state of the environment is completely determined by the current state and the action which is going to be performed, then the environment is deterministic. In the cases where this is not possible, if we have quantified probabilistic certainty, we call it stochastic, and if there is no quantitative probability at all, then it's non deterministic.
- **Episodic** vs **Sequential** : Episodic simply means the next action of the agent does not depend upon its previous action, whereas in the sequential environments, the current action might affect one or more future actions.
- **Static** vs **Dynamic** : static environments remain static, i.e. they dont change while the agent is thinking, but dynamic environments might change while the agent is thinking. It's important to consider the computation time that the agent takes to decide its actions.
- **Known** vs **Unknown** : If an agent knows the basics, like laws of physics of an environment, then that environment is said to be a known environment, else it is an unknown environment and the agent needs to learn the basics before actually making good decisions.

Designing our agent for a single environment isn't enough. We might need to make the same agent work in different environments, depending upon our needs. For that purpose, we don't test our agent on a single environment, rather we test it on an environment class. We consider the average performance of the agent over all the environments in the particular environment class we tested it upon.