1 Introduction to Artificial Intelligence

We can describe Artificial Intelligence (AI) as any task performed by a program or a machine that, if a human carried out the same activity, we would say the human had to apply intelligence to accomplish the task. The natural question comes to our mind is *what should we aim to do in AI?*

To answer this question, let us go back to the 120 years ago, when we were trying to create a object(airplane) that can fly like bird. At that time, we had two choices

1. Design objects that looks like birds.

2. Design objects that can fly without necessarily looking like birds.

The initial efforts were towards the first route, but later in time we settled for second route, and we created airplanes. Hence, we should aim to design the objects that can act rationally, and we call such objects as rational agent.

In this course, we will learn different techniques to design rational agent, but before going to details of it first address the question *how to determine whether or not these objects(rational agent) are capable of thinking like a human being.*

In 1950, Turing proposed that, let human and AI system communicate through some medium; and if AI system can convince human that he/she is talking to another human, not a AI system then we can say that AI system is capable of thinking like a human being. This test to determining whether AI system can fool human or not is commonly know as Turing test. Now a days before login into many website we have to prove that we are not robot by answering some sort of question or by entering some capcha, is an example of Turing test.

Now, let us discuss about Rational Agent Design(RAD) in detail.

1.1 Rational Agent Design

An agent is an entity that perceives its environment through its sensors. Figure 1[1] explains the setup, it has an environment sensors for perception, an *agent function* to map history of precepts to an action, and a actuators to preform the action.

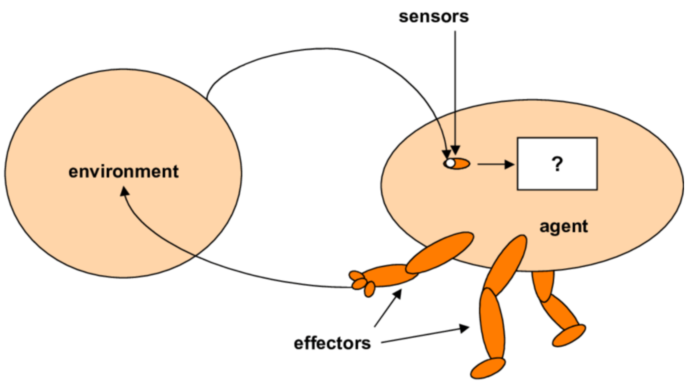

Figure 1: AI setup.

[1]The image is taken from https://www.tutorialspoint.com/artificial_intelligence/artificial_intelligence_agents_and_environments.htm

Agent Function: It is a functions that maps percepts histories/sequences to actions. Let P be the sequence of percepts and A be the the actions, then function $f : P^* \mapsto A$.

Agent Program: It run on physical architecture to perform agent function f.

Let us try to understand this setting through an example of mopping robot, we can call this as MopBot. Let us say there are two location A and B, and the percepts of MopBot allows it to figure out location and the state of the location, i.e, whether the location is dirty or clean.

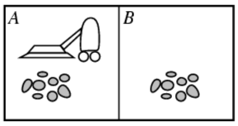

Figure 2: Mopping Bot Setup.

Enivorment - The location of MopBot and status of the each location.

Percepts - Current location and status of that location.

Actions - Left, Right, Clean, Idle.

Percepts Sequence	Action
[A, clean]	Right
[A, dirty]	Clean
[B, clean]	Left
[B, dirty]	Clean
[A, clean] [A, clean]	Right
[A, clean] [A, dirty]	Clean

Table 1: MopBot functions. Let us assume that location A is on left location B. Percept sequence represents [Location, Status].

Table 1 represents the actions functions for MopBot agent. But, the question arise here is *what is a rational agent*. Before formally answering this questions, let us introduce agent function for same MopBot agent. In Table 2, we can see that there are two set of actions are described for a percept sequence. Let say agent function f_1 maps percepts to action$_1$, and agent function f_2 maps percepts to action$_2$.

Now, the question is *What is the difference between a system with agent function f_1, and a system with agent function f_2*. As we can clearly see, with agent function f_2, the system is not MopBot. With the agent function f_2, the system is not doing the *right thing*, it is not behaving rational.

Rational Agent An agent should strive to *do the right thing* based on what it can perceive, and the actions it can perform. Note that rationality is not all-knowing with infinite knowledge.

Percepts Sequence	Action_1	Action_2
[A, clean]	Right	Mop
[A, dirty]	clean	Right
[B, clean]	Left	Mop
[B, dirty]	Clean	Left
[A, clean] [A, clean]	Right	Clean
[A, clean] [A, dirty]	Clean	Right

Table 2: MopBot functions. Let us assume that location A is on left location B. Percept sequence represents [Location, Status].

1.2 Task Environment

How to check if an agent is doing the right thing or not; for that, we should have objective criteria for measuring success of an agent's behavior, i.e, we should have a performance measure. For example in the case of vacuum-clear agent, we can check amount of dirt cleaned, time taken to clean, electricity consumed, noise generated as criteria for measuring performance.

To create a rational agent, we need to specify task environment. The task environment consists of following four things:

- Performance Measure
- Environment
- Actuators
- Sensors

Let us discuss different properties of task environment.

Fully observable vs Partially observable If sensors provide access to the complete state of the environment at each point in time, it is fully observable, else it is partially observable

Deterministic vs. Stochastic If the next state of the environment is completely determined by the current state and the action executed by the agent, it is deterministic, else stochastic.

Episodic vs. Sequential If the choice of current action does not depend on actions in past episodes, it is episodic, else sequential.

Static vs Dynamic If the environment is unchanged while an agent is deliberating, it is static, else it is dynamic.

Discrete vs. Continuous If it has a finite number of distinct states, percepts, and actions, it is discrete, else continuous.

Single agent vs. Multi-agent If an agent operating by itself in an environment, it is single agent, else multi-agent.

Properties of task environment largely determine agent design. World is partially observable, stochastic, sequential, dynamic, continuous,and multi-agent.

1.3 Agent Function

Let us again visit the definition of agent function: It is a functions that maps percepts histories/sequences to actions. Let P be the sequence of percepts and A be the the actions, then function $f : P^* \mapsto A$.

If we have a table-lookup definition for agent function, i.e a table where for every percept there is action.

Algorithm 1 Table-Driven-Agent(*percept*)

```
Static: percepts: a sequence
        table: a table of actions, indexed by percept sequences,
        fully specified.
percepts ← precepts ∪ percept.
action ← LOOKUP(percept,table)
return action
```

Algorithm 1 represents the table driven agent function definition. But, as we can see with algorithm 1,there are following drawbacks:

- Huge table to store.
- Take a long time to build the table.
- So many entries to get correct

There are four types of agent functions: Simple Reflex, Model-Based Reflex, Goal based, Utility based. Table 3 shows all four types of agent functions in detail.

Goal based agent function, tries to achieve respecified goal, and Utility based agent functions triers to maximize the utility. Performance measure is what others care about while utility is what the agent cares about. Ideally the two should match, but it is not always the case.

In addition to that, there are two more types of agents.

Learning agent The agent can learn about the performance/environment measure.

Autonamous agent An agent can update its agent program based on its experiences.

We will not discuss about Reflex/ Model-Based-Reflex agent, we will start our discussion with goal based agents.

Table 3: Four basic kinds of agent program

Agent Program	Description	Example
Simple Reflex	• Passive. • State depends only on percept • The agent follows simple action-condition rules.	Infant: • *Hungry* $\rightarrow$ *Cry* • *Happy* $\rightarrow$ *Sleep*
Model-Based Reflex	• Passive • Agent maintains some sort of internal state based on percept history that reflects some of the unobserved aspects of the world. • State depends on the model of the world, percept and current state	Kid: • Want Ice Cream & Didn't Receive $\rightarrow$ Ask for Ice Cream
Goal-Based	• Active • Plan actions depending on current state, percepts and model of the world to achieve a desirable goal.	Teenager: • Want to get into NUS $\rightarrow$ Plan Study and Playing based on the Goal.
Utility-Based	• Actions based on a weighted function of multiple goals • Goals may be inconsistent or conflicting • Useful when unclear which goal is most important and a weighted balance is needed	Adult: • αJob+βPartner+γHealth