

Intelligent Agents

Michael Rose

In which we discuss the nature of agents, perfect or otherwise, the diversity of environments, and the resulting menagerie of agent types.

2.1 | Agents and Environments

An **agent** is anything that can be viewed as perceiving its **environment** through **sensors** and acting upon that environment through **actuators**. We use the term **percept** to refer to the agent's perceptual inputs at any given instant. An agent's **percept sequence** is the complete history of everything the agent has ever perceived.

In general, an agent's choice of action at any given instant can depend on the entire percept sequence observed to date, but not on anything it hasn't perceived.

We can say that an agent's behavior is described by the **agent function** that maps any given percept sequence to an action.

2.2 | Good Behavior: The Concept of Rationality

A **rational agent** is one that does the right thing - conceptually speaking, every entry in its percept sequence table for the agent function is filled out correctly. In order to deduce if an agent has done the right thing, we need a performance measure upon which to base the consequences of its sequence of percepts in the environment.

As a general rule, it is better to design performance measures according to what one actually wants in the environment, rather than according to how one thinks the agent should behave.

2.2.1 | Rationality

What is rational at any given time depends on 4 things:

- The performance measure that defines the criterion of success
- The agent's prior knowledge of the environment
- The actions that the agent can perform
- The agent's percept sequence to date

This leads to the 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.

2.2.2 | Omniscience, learning, and autonomy

We must be careful to differentiate between rationality and **omniscience**. An omniscient agent knows the actual outcome of its actions and can act accordingly; but omniscience is impossible in reality. We must know that rationality is not the same as perfection. If we expect an agent to do what turns out to be the best action after the fact, it will be impossible to design an agent to fulfill this specification.

Doing actions in order to modify future percepts - **information gathering** - is an important part of rationality. A rational agent should be autonomous - it should learn to compensate for partial or incorrect prior knowledge.

2.3 | The Nature of Environments

We must consider **task environments**, which are essentially the problems to which rational agents are solutions.

2.3.1 | Specifying the Task Environment

In our discussion of the rationality of the simple vacuum-cleaner agent, we had to specify the performance measure, the environment, and the agent's actuators and sensors. We group these under the heading of the **task environment** and can call them **PEAS**(Performance, Environment, Actuators, Sensors). In defining a rational agent, the first step must always be to specify the task environment as fully as possible.

2.3.2 | Properties of Task Environments

The range of environments that may arise is vast, but we can identify a fairly small number of dimensions along which task environments can be categorized. These dimensions determine, to a large extent, the appropriate agent design and the applicability of each of the principal families of techniques for agent implementation.

Some dimensions:

- Fully Observable vs. Partially Observable
- Single Agent vs. Multiagent
- dealing with multiple agents denotes the need for competition or cooperation or both
- Deterministic vs. Stochastic
- The word stochastic implies that uncertainty about outcomes is quantified in terms of their probabilities. Nondeterminism implies that no probabilities are attached.
- Episodic vs. Sequential
- In an episodic task environment, the agent's experience is divided into episodes in which one percept action combo does not affect the next episode. Sequential environments have each decision affecting all future decisions.
- Static vs. Dynamic
- Is the environment changing? If the environment does not change with the passage of time, but the agent's performance score does, it is considered a semidynamic environment
- Discrete vs. Continuous
- This applies to the state of the environment, to the way time is handled, and to the percepts and actions of the agent.
- Known vs. Unknown
- Refers to the agent's state of knowledge about the 'laws of physics' of the environment

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

2.4 | The Structure of Agents

We must talk about how the insides of agents work. The job of AI is to design an agent program that implements the agent function - the mapping from percepts to actions. We assume it runs on some sort of computing device with physical sensors and actuators.

agent = architecture + program

All the agent programs in this book have the same skeleton: They take the current percept as input from the sensors and return an action to the actuators.

There are four basic kinds of agent programs that embody the principles underlying almost all intelligent systems:

- Simple Reflex Agents
- Model-Based Reflex Agents
- Goal-Based Agents
- Utility-Based Agents

2.4.2 | Simple Reflex Agents

These agents select actions on the basis of the current percept, ignoring the rest of the percept history. They act according to a rule whose condition matches the current state, as defined in a percept.

They are simple, but are of limited intelligence. they work only if the correct decision can be made on the basis of only the current percept – that is, only if the environment is fully observable. Even a little bit of unobservability can cause serious trouble. Infinite loops are often unavoidable for simple reflex agents operating in partially observable environments.

2.4.3 | Model-Based Reflex Agents

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. This knowledge of the environment around the agent is called a **model** of the world. An agent that uses such a model is called a Model-Based agent.

2.4.4 | Goal-Based Agents

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

Search and **Planning** are the subfields of AI devoted to finding action sequences that achieve the agents goals.

Decision making of this kind is fundamentally different from the condition-action rules described earlier, in that it involves consideration of the future - both “What will happen if I do such and such?” and “Will that make me happy?”

2.4.5 | Utility-Based Agents

Goals alone are not enough to generate high quality behavior in most environments. Goals simply provide a crude binary distinction between happy and not happy states. A more general performance measure should allow a comparison of different world states according to exactly how happy they would make the agent. Since happy does not sound scientific, economists and computer scientists use the term utility instead.

An agent's **utility function** is essentially an internalization of the performance measure. A rational utility-based agent chooses the action that maximizes the **expected utility** of the action outcomes.

2.4.6 | Learning Agents

A learning agent can be divided into four conceptual components

- learning element
- responsible for making improvements
- performance element
- responsible for selecting external actions. We previously considered the entire agent to be the performance element
- critic
- provides feedback to the learning element on how the agent is doing and determines how the performance element should be modified to do better in the future.
- problem generator
- responsible for suggesting actions that will lead to new and informative experiences.

Learning in intelligent agents can be summarized as the process of modification of each component of the agent to bring the components into closer agreement with the available feedback information, thereby improving the overall performance of the agent.

2.4.7 | How the Components of Agent Programs Work

We can place the representations of a system along an axis of increasing complexity and expressive power.

- atomic : a state is a black box with no internal structure
- factored : A state consists of a vector of attribute values. Values can be boolean, real, or one of a fixed set of symbols
- structured : A state includes objects, each of which may have attributes of its own as well as relationships to other objects

The algorithms underlying **search**, **game playing**, **hidden markov models** and **markov decision processes** all work with atomic representations, or at least, they treat representations as if they were atomic.

A **factored representation** splits up each state into a fixed set of variables or attributes, each of which contains value(s). With factored representations, we can also represent *uncertainty*. Many important areas of AI are based on factored representations, including **constraint satisfaction**, **propositional logic**, **planning**, **bayesian networks**, and **machine learning**.

A **structured representation** shows objects and their varying relationships explicitly. Structured representations underly **relational databases**, **first-order logic**, **first order probability models**, **knowledge-based learning**, and much of **natural language understanding**.