# 第二章 智能体

## 智能体和环境

任何通过传感器（sensor）感知环境（environment）并通过执行器（actuator）作用于该环境的事物都可以被视为**智能体（agent）**。

图表, 箱线图

描述已自动生成

智能体的**感知序列（percept sequence）**是智能体所感知的一切的完整历史。一般而言，一个智能体在任何给定时刻的动作选择可能取决于其内置知识和迄今为止观察到的整个感知序列，而不是它未感知到的事物。

从数学上讲，智能体的行为由**智能体函数（agent function**）描述，该函数将任意给定

的感知序列映射到一个动作。

在内部，人工智能体的智能体函数将由**智能体程序（agent program）**实现。

智能体函数是一种抽象的数学描述，而智能体程序是一个具体的实现，可以在某些物理系统中运行。

## 良好行为：理性的概念

A **rational agent** is one that does the right thing.  
性能度量

结果主义（consequentialism）：We evaluate an agent’s behavior by its consequences.

If the sequence of environment state is desirable, then the agent has performed well. This notion of desirability is captured by a **performance measure(性能度量)** that evaluates any given sequence of environment states.

作为一般规则，更好的做法是根据一个人在环境中真正想要实现的目标，而不是根据一个人认为智能体应该如何表现来设计性能度量。

### 理性

**理性**取决于以下四方面：

1. 定义成功标准的性能度量；
2. 智能体对环境的先验知识；
3. 智能体可以执行的动作；
4. 智能体到目前为止的感知序列。

**理性智能体**：

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.

### 全知、学习和自主

我们需要仔细区分理性和全知（omniscience）。全知的智能体能预知其行动的实际结果，并能据此采取行动，但在现实中，全知是不可能的。

Doing actions in order to modify future percepts— sometimes called **information gathering（信息收集）**—is an important part of rationality.

我们的定义要求理性智能体不仅要收集信息，还要尽可能多地从它所感知到的东西中学习（learn）。

一个理性的智能体应该具备自主性（autonomy），它应该学习如何弥补部分或不正确的先验知识。

## 环境的本质

We must think about **task environments**, which are essentially the “problems” to which rational agents are the “solutions.”

### 指定任务环境

We group all these under the heading of the task environment. we call this the **PEAS (Performance, Environment, Actuators, Sensors)** description. In designing an agent, the first step must always be to specify the task environment as fully as possible.

### 任务环境的属性

我们可以确定相当少的维度，并根据这些维度对任务环境进行分类：

1. 完全可观测的（fully observable）与部分可观测的（partially observable）以及不可观测的（unobservable）
2. 单智能体的（single-agent）与多智能体的（multiagent）
3. 竞争性（competitive）的多智能体环境和合作的cooperative）多智能体环境
4. 确定性的（deterministic）与非确定性的（nondeterministic）：如果环境的下一个状态完全由当前状态和智能体执行的动作决定，那么我们说环境是确定性的，否则是非确定性的。
5. 回合式的（episodic）与序贯的（sequential）：在回合式任务环境中，智能体的经验被划分为原子式的回合。在每一回合中，智能体接收一个感知，然后执行单个动作。至关重要的是，下一回合并不依赖于前几回合采取的动作。在序贯环境中，当前决策可能会影响未来所有决策。
6. 静态的（static）与动态的（dynamic）：如果环境在智能体思考时发生了变化，我们就说该智能体的环境是动态的，否则是静态的。如果环境本身不会随着时间的推移而改变，但智能体的性能分数会改变，我们就说环境是半动态的（semidynamic）。
7. 离散的（discrete）与连续的（continuous）
8. 已知的（known）与未知的（unknown）：这种区别不是指环境本身，而是指智能体（或设计者）对环境“物理定律”的认知状态。已知和未知环境之间的区别与完全可观测和部分可观测环境之间的区别不同。一个已知的环境很可能是部分可观测的，一个未知环境可以是完全可观测的。

### 智能体的结构

智能体 = 架构 + 程序

### 简单反射型智能体（simple reflex agent）

智能体根据当前感知选择动作，忽略感知历史的其余部分。只有在环境完全可观测的情况下才可行。

条件-动作规则（condition-action rule）：IF Then

图示

描述已自动生成

### 基于模型的反射型智能体

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. That is, 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.

随着时间的推移，更新这些内部状态信息需要在智能体程序中以某种形式编码两种知识。首先，需要一些关于世界如何随时间变化的信息，这些信息大致可以分为两部分：智能体行为的影响和世界如何独立于智能体而发展。

这种关于“世界如何运转”的知识被称为世界的**转移模型（transition model）**；

世界状态如何反映在智能体感知中的信息称为**传感器模型（sensor model）**；

转移模型和传感器模型结合在一起让智能体能够在传感器受限的情况下尽可能地跟踪世界的状态。使用此类模型的智能体称为**基于模型的智能体（model-based agent）**。

图示

描述已自动生成

### 基于目标的智能体

除了当前状态的描述之外，智能体还需要某种描述理想情况的目标信息。智能体程序可以将其与模型与基于模型的反射型智能体中使用的信息相同）相结合，并选择实现目标的动作。

图示

描述已自动生成

### 基于效用的智能体

智能体的效用函数（utility function）本质上是性能度量的内部化。如果内部效用函数和外部性能度量一致，那么根据外部性能度量选择动作，以使其效用最大化的智能体是理性的。

基于效用的理性智能体会选择能够最大化其动作结果期望效用（expected utility）的动作。

图示

描述已自动生成

### 学习型智能体

**性能元素**是我们之前认为的整个智能体：它接受感知并决定动作。

**学习元素**使用来自评估者（critic）对智能体表现的反馈，并以此确定应该如何修改性能元素以在未来做得更好。

**评估者**告诉学习元素：智能体在固定性能标准方面的表现如何。

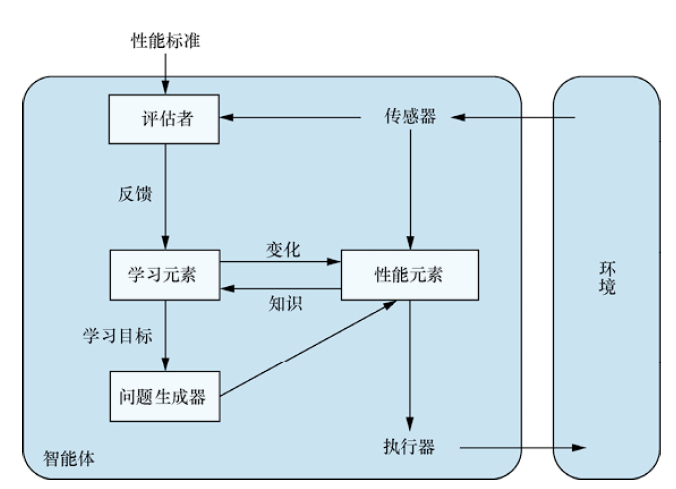
问题生成器（problem generator）负责建议动作，这些动作将获得全新和信息丰富的经验。

### 智能体程序的组件如何工作

在**原子表示（atomic representation**）中，世界的每一个状态都是不可分割的，它没有内部结构。

**因子化表示（factored representation）**将每个状态拆分为一组固定的变量或属性，每个变量或属性都可以有一个值

**结构化表示**（structured representation）



图示

中度可信度描述已自动生成