Planning With Uncertainty: Processes and Values

Arnav Gupta

November 27, 2024

Contents

1 Worlds and Planning

Agents carry out actions according to the **horizon**:

- infinite horizon: forever
- indefinite horizon: until stopping criteria is met
- finite horizon: finite and fixed number of steps

It is helpful to know, what an agent should do when (for all planning horizons):

- it gets rewards/punishments and tries to maximize rewards

- actions can be noisy, so the outcome can't be fully predicted
- there is a model that specifies the probabilistic outcome of actions
- the world is fully observable: current state is always fully in evidence

1.1 World State

The information such that if the **world state** is known, no info about the past is relevant to the future.

The **Markovian Assumption** is: Let S_i, A_i be the state and action at time i, then

$$P(S_{t+1} \mid S_0, A_0, \ldots, S_t, A_t) = P(S_{t+1} \mid S_t, A_t)$$

where $P(s' \mid s, a)$ is the probability that the agent will be in state s' immediately after doing action a in state s.

The dynamics is **stationary** is the distribution is the same for each time point.

If the process never halts, this has an infinite horizon.

If the process stays in a state getting no reward, these are **absorbing states**, which has an indefinite horizon.

2 Markov Decision Processes

Augments a Markov chain with actions and values.

For an MDP, specify the:

- set S of states
- set A of actions
- $P(S_{t+1} \mid S_t, A_t)$ specifies the dynamics
- $R(S_t, A_t, S_{t+1})$ specifies the **reward**, which the agent gets at each time step, specifically when it ends up in S_{t+1} after doing A_t from S_t

Fully-observable MDP: agent gets to observe S_t when deciding on action A_t

Partially-observable MDP (POMDP): agent has some noisy sensor of the state, so it must remember sensing and acting history by maintaining a sufficiently complex **belief state**

2.1 Rewards and Values

Suppose the agent receives the sequence of rewards $r_1, r_2, \ldots$. The value that should be assigned could be:

- total reward

$$V = \sum_{i=1}^{\infty} r_i$$

- average reward

$$V = \lim_{n \to \infty} (r_1 + \cdots + r_n)/n$$

- discounted reward

$$V = r_1 + \gamma r_2 + \gamma^2 r_3 + \cdots$$

  where $0 \leq \gamma \leq 1$ is the **discount factor**

2.2 Policies

A **stationary policy** is a function $\pi : S \to A$ where for a given state s, $\pi(s)$ specifies what action the agent who is following π will do.

Optimal policy: one with maximum expected discounted reward

For a fully-observable MDP with stationary dynamics and rewards with infinite or indefinite horizon, there is always an optimal stationary policy.

2.2.1 Policy Value

$Q^\pi(s, a)$ for some action a and state s, is the expected value of doing a in state s, then following policy π.

$V^\pi(s)$ for some state s, is the expected value of following policy π in state s.

Q^π and V^π are **mutually recursive**:

$$Q^\pi(s, a) = \sum_{s'} P(s' \mid a, s)(r(s, a, s') + \gamma V^\pi(s'))$$

$$V^\pi(s) = Q^\pi(s, \pi(s))$$

2.2.2 Optimal Policy Value

$Q^*(s, a)$ for some action a and state s, is the expected value of doing a in state s, then following the optimal policy.

$\pi^*(s)$ is the optimal action to take in state s.

$V^*(s)$ for some state s, is the expected value of following the optimal policy in state s.

Q and V are **mutually recursive**. Further:

$$\pi(s) = \arg\max_a Q(s, a)$$

3 Value Iteration

t-step lookahead value function V^t: expected value with t steps to go

The goal, given an estimate of the t step lookahead value function, is to determine the $t + 1$ step lookahead value function.

3.1 Steps

Set V^0 arbitrarily and $t = 1$. Compute Q^t and V^t from V^{t-1}:

$$Q^t(s, a) = \left[R(s) + \gamma \sum_{s'} \Pr(s' \mid s, a) V^{t-1}(s')\right]$$

$$V^t(s) = \max_a Q^t(s, a)$$

The policy with t stages to go is simply the action that maximizes the following

$$\pi^t(s) = \arg\max_a [R(s) + \gamma \sum_{s'} \Pr(s' \mid s, a) V^{t-1}(s')]$$

This converges exponentially fast over t to the optimal value function.

Let $\|X\| = \max\{|x|, x \in X\}$. Convergence when $\|V^t(s) - V^{t-1}(s)\| < \epsilon \frac{(1-\gamma)}{\gamma}$ ensures V^t is within ϵ of the optimal.

3.2 Asynchronous Value Iteration

Can update value function for each state individually rather than sweeping through all states. This converges to the optimal value function if each state and action are visited infinitely often in the limit. Either $V[s]$ or $Q[s,a]$ can be stored.

To store $V[s]$, repeat the following forever:

1. select state s
2. $V[s]$ becomes

$$\max_a \sum_{s'} P(s' \mid s,a)(R(s,a,s') + \gamma V[s'])$$

3. select action a (using an exploration policy)

To store $Q[s,a]$, repeat the following forever:

1. select state s and action a
2. $Q[s,a]$ becomes

$$\sum_{s'} P(s' \mid s,a) \left(R(s,a,s') + \gamma \max_{a'} Q[s',a'] \right)$$

4 Markov Decision Processes and State

Represent $S = \{X_1, \dots, X_n\}$ where X_i are random variables. For each X_i and each action $a \in A$, there is $P(X_i' \mid S, A)$.

The reward may be additive:

$$R(X_1, \dots, X_N) = \sum_i R(X_i)$$

Value iteration proceeds as usual but can do one variable at a time, like variable elimination.

A **Partially Observable Markov Decision Process (POMDP)** is like an MDP but some variables are not observed. It is a tuple $\langle S, A, T, R, O, \Omega \rangle$:

- S is a finite set of unobservable states
- A is a finite set of agent actions

- $T : S \times A \rightarrow S$ is a transition function
- $R : S \times A \rightarrow \mathcal{R}$ is a reward function
- O is a set of observations
- $\Omega : S \times A \rightarrow O$ is an observation function