ROB311 Quiz 3

Hanhee Lee

March 1, 2025

Contents

Partially Observable Probabilistic Decision Problems

1 Reinforcement Learning

Summary: In a RL problem, $p(\cdot \mid \cdot, \cdot)$ and/or $r(\cdot\cdot, \cdot)$ unknown.

1.1 Estimating Q-Star Empirically

Summary:

γ	**Equation**
0	$q^*(s,a) = \lim_{K\to\infty} \bar{R}_K$ • $\bar{R}_K = \frac{1}{K}\sum_{k=1}^{K} r_k$: empirical average reward. • r_k: reward obtained in the k^{th} simulation. • K: # of times action a taken in state s (# of simulations)
0	$q^*(s,a) \leftarrow q^*(s,a) + \frac{1}{N(s,a)}\left(r(s,a,s') - q^*(s,a)\right)$ • $N(s,a)$: # of times action a taken in state s.
$\neq 0$	$q^*(s,a) \leftarrow q^*(s,a) + \frac{1}{N(s,a)}\left(\left[r(s,a,s') + \gamma \max_{a'} q^*(s',a')\right] - q^*(s,a)\right)$ • Using old q^* values to estimate.

1.1.1 Running Average Update Rule

Definition:

$$\bar{x} \leftarrow \bar{x} + \alpha(x_{\text{new}} - \bar{x}).$$

- α: learning rate

1.2 Q-Learning Algorithm

Algorithm:

```
procedure Q_LEARNING():
    for each episode do
        set initial state s ← s_0
        while s ∉ 𝒯 do # 𝒯: terminal states
            randomly choose an action in 𝒜(s)
            get next state, s', and reward r
            update N(s,a) and q*(s,a) as follows:

            q*(s,a) ← q*(s,a) + 1/N(s,a) (r(s,a,s') + γ max_{a'} q*(s',a') − q*(s,a))

            N(s,a) ← N(s,a) + 1

            s ← s'
        end while
    end for
```

- **Note:** Possible infinite while loop if $\mathcal{T}$ is not reached.

1.3 Modified Q-Learning Algorithm

Algorithm:

```
procedure Q_LEARNING():
    for each episode do
        l ← 0
        set initial state s ← s_0
        while s ∉ 𝒯 and l < l_max do
            randomly choose an action in 𝒜(s)
            get next state, s', and reward r
            update N(s,a) and q*(s,a) as follows:

            q*(s,a) ← q*(s,a) + 1/N(s,a) (r(s,a,s') + γ max_{a'} q*(s',a') − q*(s,a))

            N(s,a) ← N(s,a) + 1

            l ← l + 1
            s ← s'
        end while
    end for
```

Notes: Choice of γ and $l_{\max}$ are coupled:
- $\gamma \approx 1$ requires large $l_{\max}$
- $\gamma \approx 0$ requires small $l_{\max}$

1.4 Training vs. Testing

Notes: Episodes are classified as either:
- training (sim): reward accumulated during episode does not count
- testing (test): reward accumulated during episode counts

1.4.1 K Sims, 1 Test

Notes:
1. select actions randomly during K simulations
2. extract optimal policy, π^*
3. use π^* during test

1.4.2 K Tests

Notes:
- maximize average reward over K tests
- must balance between exploration and exploitation
- Common ways to balance exploration and exploitation: ε-greedy strategy, UCB algorithm

Strategy	Description
ε-greedy	choose optimal action with probability $\varepsilon(k)$

- In episode k, choose the optimal action with probability $\varepsilon(k)$, where:
  - $\varepsilon(0) \approx 0$
  - $\varepsilon(k)$ is increasing
  - $\varepsilon(k) \to 1$ as $k \to \infty$
- Common choice for $\varepsilon(k)$ is $1 - \frac{1}{k}$.

Strategy	Description
UCB algorithm	choose action that maximizes $\text{UCB}(\cdot)$

$$\text{UCB}(s, a) = \begin{cases} q^*(s, a) + C\sqrt{\dfrac{\log k}{N(s, a)}}, & \text{if } N(s, a) > 0 \\ \infty, & \text{otherwise} \end{cases}$$

- In episode k, choose the action that maximizes $\text{UCB}(\cdot)$.
- C: exploration parameter
- $N(s, a)$: # of times a taken from s.

2 Partially Observable MDPs (POMDPs)

Summary: In a **POMDPs**, we assume that:

- environment modelled using state space, $\mathcal{S}$
- single agent
- S_t = state after transition t
- A_t = action inducing transition t
- stochastic state transitions with memoryless property:

$$S_T \perp S_0, A_1, \ldots, A_{T-1}, S_{T-2} \mid S_{T-1}, A_T$$

- R_t = reward for transition t, i.e., (S_{T-1}, A_T, S_T)
- O_t = observation of S_t

Name	**Function:**
Initial state distribution	$p_0(s) := \mathbb{P}[S_0 = s]$
Transition distribution	$p(s'\|s, a) := \mathbb{P}[S_t = s'\|A_t = a, S_{t-1} = s]$
Reward function	$r(s, a, s') :=$ reward for transition (s, a, s')

- Since actual state is unknown, so are legal actions.
- Can fix by assuming $\mathcal{A}(s) = \mathcal{A}(s') := \mathcal{A}\ \forall s, s'$:
  - if $a \notin \mathcal{A}(s)$, then $p(s'|s, a) = 0$ for all $s' \neq s$
  - if $a \notin \mathcal{A}(s)$, then $r(s, a, s') = 0$ for all s'

Policy for choosing actions	$\pi_t(a\|o_0, \ldots, o_t) := \mathbb{P}[A_t = a\|O_0 = o_0, \ldots, O_t = o_t]$
Measurement model	$m(o\|s) := \mathbb{P}[O_t = o\|S_t = s]$

- Observe that policy is now time-dependent.
- **Special Case:** If we assume the agent cannot use past observations, $A_t \perp O_0, \ldots, O_{t-1} \mid O_t$, policy becomes time-independent,

$$\pi_t(a|o_0, \ldots, o_t) = \pi_0(a|o_t).$$

  - Only need to specify π_0.

Belief after t observations

$$b_t(s_t|a_{1:t}, o_{0:t}) = \mathbb{P}[S_t = s_t|A_t = a_t, O_{0:t} = o_{0:t}]$$
$$b_t(s_t|a_{1:t}, o_{0:t}) = m(o_t|s_t) \sum_{s_{t-1}} p(s_t|s_{t-1}, a_t) b_{t-1}(s_{t-1}|a_{1:t-1}, o_{1:t-1})$$

- b_t: Probability distribution
- $b_0(s_0) = \mathbb{P}[S_0 = s_0]$: Initial belief distribution
- Only holds for $t \geq 1$.
- For $t = 0$ (assuming uniform prior): $b_0(s_0|o_0) = \dfrac{m(o_0|s_0)}{\sum_s m(o_0|s)}$.

2.1 Bayesian Network

Notes: $S_0, O_0, A_1, R_1, S_1, O_1, A_2, R_2, S_2, O_2, \ldots$ form a Bayesian network:

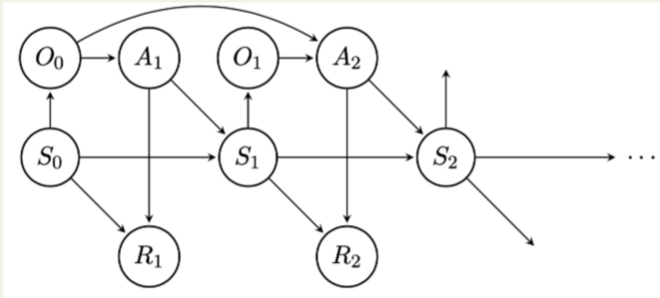

Figure 1

- Assuming $A_t \perp O_0, \ldots, O_{t-1} \mid O_t$. WHERE DOES THIS COME INTO PLAY.

Example: