Reinforcement Learning - Part 1

Alice Gao

Lecture 20

Readings: RN 21.2 - 21.3. PM 12.1, 12.5, 12.8.

Outline

Learning Goals

By the end of the lecture, you should be able to

- Trace and implement the passive adaptive dynamic programming algorithm.
- Explain the trade-off between exploration and exploitation.
- Trace and implement the active adaptive dynamic programming algorithm.

An Reinforcement Learning Agent

Let's consider fully observable, single-agent reinforcement learning. We will formalize this problem as a Markov decision process.

- Given the possible states and the set of actions.
- Observes the state and the rewards received.
- Carries out an action.
- Goal is to maximize its discounted reward.

Why is reinforcement learning challenging?

- Which action was responsible for this reward/punishment?
- How will this action impact my utility?
- Should I explore or exploit?

Passive Learning Agent

- Fix a policy π.
- Goal is to learn $V^\pi(s)$ (the expected value of policy π for state s).
- Similar to policy evaluation.
- Does not know the transition model $P(s'|s,a)$ nor the reward function $R(s)$.
- Solution: Adaptive Dynamic Programming
  - Learn $P(s'|s,a)$ and $R(s)$ using the observed transitions and rewards.
  - Learn $V^\pi(s)$ by solving Bellman equations (exactly or iteratively).

$$V^\pi(s) = R(s) + \gamma \sum_{s'} P(s'|s,\pi(s)) V^\pi(s').$$

- A model-based approach

Passive ADP Algorithm

1. Repeat steps 2 to 5.
2. Follow policy π and generate an experience $\langle s, a, s', r' \rangle$.
3. Update reward function: $R(s') \leftarrow r'$
4. Update the transition probability.

$$N(s, a) = N(s, a) + 1$$
$$N(s, a, s') = N(s, a, s') + 1$$
$$P(s'|s, a) = N(s, a, s')/N(s, a)$$

5. Derive $V^{\pi}(s)$ by using the Bellman equations.

$$V(s) = R(s) + \gamma \sum_{s'} P(s'|s, \pi(s))V(s')$$

Passive ADP Example

s_{11}	$+1$
s_{21}	-1

- $\pi(s_{11}) = down, \pi(s_{21}) = right$
- $\gamma = 0.9$
- $R(s_{11}) = -0.04, R(s_{21}) = -0.04, R(s_{12}) = 1, R(s_{22}) = -1$
- $N(s, a) = 5, \forall s, a.$
- $N(s, a, s') = 3$ for the intended direction.
- $N(s, a, s') = 1$ for a direction to the left or right of the intended direction.
- The current state is s_{11}.

Passive ADP Example continued

s_{11}	$+1$
s_{21}	-1

1. No need to update the reward function.
2. Update the counts.
   $N(s_{11}, down) = 6$ and $N(s_{11}, down, s_{21}) = 4$.
3. Solve the Bellman equations.

$$V(s_{11}) = -0.04 + 0.9(0.667V(s_{21}) + 0.167(1) + 0.167V(s_{11}))$$
$$V(s_{21}) = -0.04 + 0.9(0.6(-1) + 0.2V(s_{11}) + 0.2V(s_{21}))$$

The solutions are:

$$V(s_{11}) = -0.4378, V(s_{21}) = -0.8034$$

Active ADP

The passive ADP agent learns the expected value of a fixed policy.

What action should the agent take at each step?
Two things are useful for the agent to do:

1. exploit: take an action that maximizes $V(s)$.
2. explore: take an action that is different from the optimal one.

The greedy agent seldom converges to the optimal policy and sometimes converges to horrible policies because the learned model is not the same as the true environment.

There is a trade-off between exploitation and exploration.

Trade off Exploitation and Exploration

1. ϵ-greedy exploration strategy:
   - select random action with probability ϵ, and
   - select the best action with probability $1 - \epsilon$.
   - may decrease ϵ over time.

2. Softmax selection using Gibbs/Boltzmann distribution.
   - Choose action a with probability $\frac{Q(s,a)/T}{\sum_a Q(s,a)/T}$.
   - $T > 0$ is the temperature. When T is high, the distribution is close to uniform. When T is low, the higher-valued actions have higher probabilities.

3. Initialize the values optimistically to encourage exploration.

Optimistic Utility Values to Encourage Exploration

We will learn $V^+(s)$ (the optimistic estimates of $V(s)$).

$$V^+(s) \leftarrow R(s) + \gamma \max_a f\left(\sum_{s'} P(s'|s,a)V^+(s'), N(s,a)\right)$$

$$f(u,n) = \begin{cases} R^+, \text{ if } n < N_e \\ u, \text{otherwise} \end{cases}$$

$f(u,n)$ trade-offs exploitation and exploration.

- R^+ is the optimistic estimate of the best possible reward obtainable in any state.
- If we haven't visited (s,a) at least N_e times, assume its expected value is R^+.
- Otherwise, use the current $V^+(s)$ value.

Active ADP Algorithm

1. Initialize $R(s), V^+(s), N(s,a), N(s,a,s')$.
2. Repeat steps 3 to 7 until we have visited each (s,a) at least N_e times and the $V^+(s)$ values converged.
3. Determine the best action a for current state s using $V^+(s)$.

$$a = \arg\max_a f\left(\sum_{s'} P(s'|s,a)V^+(s'), N(s,a)\right), \; f(u,n) = \begin{cases} R^+, \text{ if } n < N_e \\ u, \text{otherwise} \end{cases}$$

4. Take action a and generate an experience $\langle s, a, s', r' \rangle$
5. Update reward function: $R(s') \leftarrow r'$
6. Update the transition probability.

$$N(s,a) = N(s,a) + 1, N(s,a,s') = N(s,a,s') + 1$$
$$P(s'|s,a) = N(s,a,s')/N(s,a)$$

7. Update $V^+(s)$ using the Bellman updates.

$$V^+(s) \leftarrow R(s) + \gamma \max_a f\left(\sum_{s'} P(s'|s,a)V^+(s'), N(s,a)\right)$$

An Active ADP Example

s_{11}	$+1$
s_{21}	-1

- $\pi(s_{11}) = down, \pi(s_{21}) = right$
- $\gamma = 0.9$
- $N_e = 10,\ R^+ = 5.$
- $R(s_{11}) = -0.04, R(s_{21}) = -0.04, R(s_{12}) = 1, R(s_{22}) = -1$
- $N_(s, a) = 5, \forall s, a.$
- $N(s, a, s') = 3$ for the intended direction.
- $N(s, a, s') = 1$ for any other direction with positive transition probability.

Revisiting the Learning Goals

By the end of the lecture, you should be able to

- Trace and implement the passive adaptive dynamic programming algorithm.
- Explain the trade-off between exploration and exploitation.
- Trace and implement the active adaptive dynamic programming algorithm.