CSE6243: Advanced Machine Learning **Fall 2023**

Lecture 22: Learning with MDPs

Lecturer: Bo Dai *Scribes:*

Note: *LaTeX template courtesy of UC Berkeley EECS Department.*

22.1 Recap

Planning *v.s.* Learning. In planning, the agent has knowledge about how the environment works (the transition) and uses this to simulate and evaluate different action sequences to find the best strategy. In learning, the agent does not know the transition but can only sample from it. The agent learns a policy (a strategy for choosing actions) based on feedback from the environment.

In planning, one usually focuses on two types of evaluations, i.e.,

Policy Evaluation. In policy evaluation, the goal is to estimate the value of each state when following a particular policy. The value of a state under a policy is the expected return (rewards) starting from that state and following the policy thereafter.

Value Function Evaluation. Value function evaluation involves determining the value of each state (or state-action pair) without a specific policy. It's about understanding how good it is to be in a particular state (or to perform a specific action in a state), regardless of the policy being followed.

We are also interested in policy optimization.

Policy Optimization. Policy optimization is the process of improving the policy. The agent iteratively adjusts its policy to maximize the cumulative reward. This can be done through value iteration or policy iteration.

22.2 New Content

22.2.1 Learning in MDPs – An Overview

Perspective	Categorization
Problem setting	**Target:** policy evaluation *v.s.* policy optimization **Data collection:** passive (offline) *v.s.* active (online)
Algorithm Parametrization	**Modeling:** model-based (learn P, R) *v.s.* model-free (directly learn Q, V, π) **Exploration-exploitation:** on-policy *v.s.* off-policy **Representation:** tabular methods *v.s.* function approximation

Table 22.1: An overview

Note that the tabular methods is applicable for $|\mathcal{S}|$ is finite (can be combinatorially large).

22.2.2 Policy Evaluation

Objective. Given $\pi(a|s)$, we want to estimate $J(\pi) = \mathbb{E}_\pi[\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t)]$.

MC Approximation. In online setting, we can collect $\{(s_0, a_0, r_0, \ldots, s_H, a_H, r_H)\}_{i=1}^n$ and calculate

$$\hat{J}(\pi) = \frac{1}{n} \sum_{i=1}^{n} \sum_{t=1}^{H} \gamma^t r_t^i \tag{22.1}$$

Note that this approach is model-free and on-policy.

One might then be intrigued by the following question: *Can we do policy evaluation for one policy given some trajectories from the other policy?* This leads to:

Importance Sampling.

$$J(\pi') \approx \int \prod_{t=0}^{H} p(s_{t+1}|s_t, a_t) \frac{\pi'(a_t|s_t)}{\pi(a_t|s_t)} \left(\sum_{t=0}^{\infty} \gamma^t r_t \right) \left(\prod_{t=0}^{H} \pi(a_t|s_t) \right) \tag{22.2}$$

$$= \mathbb{E}_p i \left[\prod_{t=0}^{H} \frac{\pi'}{\pi} \left(\sum_{t=0}^{\infty} \gamma^t r_t \right) \right] \tag{22.3}$$

Note that the policy ratio term may tend to 0 as H tends to ∞, in which this estimation may fail.

Model-based Approach. In this approach, we sample $|\mathcal{S}||\mathcal{A}|$ trajectories and reward-state pairs. Then we have

$$\hat{P}(s'|s, a) = \frac{1}{n_{s,a}} \sum_{i=1}^{n_{s,a}} \mathbb{1}(s' = s'_{s,a}) \tag{22.4}$$

$$\hat{R}(s, a) = \frac{1}{n_{s,a}} \sum_{i=1}^{n_{s,a}} r_{s,a} \tag{22.5}$$

Note that this can be combinatorially large.

Temporal Difference (TD). Previously, we have

$$V_{t+1}^{\pi}(s) \Leftarrow R^{\pi}(s) + \gamma \mathbb{E}_{p,\pi(s'|s)}[V_t^{\pi}(s')] \tag{22.6}$$

With TD, we have

$$V_{t+1}^{\pi}(s) \Leftarrow (1 - \alpha) V_t^{\pi}(s) + \alpha \left(R^{\pi}(s) + \gamma V^{\pi}(s) \right) \tag{22.7}$$

where $\{s, a, s', R(s)\} \sim \pi(a|s)$.

Note that this method converge if $\sum_t \alpha \to \infty$, $\sum_t \alpha^2 < \infty$.

Deadly Triad refers to a combination of three components that, when used together, can lead to instability or divergence in the learning process. These components are:

- **Function approximation** can introduce bias and variance in the value estimates.

- **Bootstrapping** compounds these errors by updating estimates with other estimates, which may also be biased or inaccurate.
- **Off-policy learning** can exacerbate the situation because the data used for learning may not represent the policy being evaluated or optimized.

This combination can lead to unstable learning dynamics and divergence, where the value function fails to converge to the correct values. To mitigate the issues of the Deadly Triad, we may use the target networks. Specifically, we use algorithms like Deep Q-Networks (DQN) or double DQN, where a separate, slowly updated network is used for the bootstrap estimates.