Model-free Prediction

Solving MDPs can mean two things:
Prediction: Given an MDP (S, A, P, R, γ) and a policy $\pi(a|s)$ find the state and action value functions (Q and V)
Control: Given an MDP (S, A, P, R, γ), find the optimal policy. Compare with the *learning* problem where rewards and dynamics are unknown.
Remember value functions estimate the goodness of a particular state or state-action pair: how good is for the agent to be in a particular state or execute a particular action at a particular state, for a given policy.
The value of a state is the sum of immediate reward received after transitioning to this state and discounted value of the next state, following a given policy.

Why value functions are useful

An optimal policy can be found by maximizing over $Q^*(s, a)$.
An optimal policy can be found from the dynamics model using one-step look ahead.
If we know $Q^*(s, a)$, we immediately have the optimal policy, we do not need the dynamics model.
If we know $V^*(s)$, we need the dynamics model to do one step lookahead, to choose the optimal action (because we don't know where we end up).

Summary so far

To estimate value functions we have been using DP with **known reward and dynamics** functions:

$$v_{k+1}(s) = \sum_a \pi(a|s)(r(s, a) + \gamma \sum_{s'} p(s'|s, a)v_k(s'))$$

for $\forall s$

$$v_{k+1}(s) = max_{a \in A}(r(s, a) + \gamma \sum_{s'} p(s'|s, a)v_k(s'))$$

for $\forall s$
Now, instead of probability distributions to compute expectations, use empirical expectations by averaging sampled returns.

Monte Carlo

Monte Carlo methods learn from complete sampled trajectories and their returns. Only defined for episodic tasks and all episodes must terminate.
Remember that the **return** is the total discounted reward:

$$G_t = R_{t+1} + \gamma R_{t+2} + \ldots + \gamma^{T-1} R_t$$

Remember that the **value function** is the expected return:

$$V_\pi(s) = E_\pi[G_t | S_t = s]$$

Monte Carlo policy evaluation uses **empirical mean return** instead of **expected return**.
Every-visit MC: average returns for every time s is visited in an episode. Imagine the same state is revisited by the agent multiple times.
First-visit MC: average returns only for first time s is visited in an episode.

Incremental Mean

The mean μ_k of a sequence $x_1, \ldots, x_n$ can be computed **incrementally**:

$$\mu_k = \mu_{k-1} + \frac{1}{k}(x_k - \mu_{k-1})$$

For each state s_t with return G_t:

$$N(s_t) = N(s_t) + 1$$

$$V(s_t) = V(s_t) + \frac{1}{N(s_t)}(G_t - V(s_t)$$

In non-stationary problems, better to track a **running mean**:

$$V(s_t) = V(s_t) + \alpha(G_t - V(s_t))$$

TD Learning

The simplest TD method **TD(0)**:

$$V(s_t) = V(s_t) + \alpha[R_{t+1} + \gamma V(s_{t+1}) - V(s_t)]$$

$R_{t+1} + \gamma V(s_{t+1})$ is the **TD target**.
$\delta_t = R_{t+1} + \gamma V(s_{t+1}) - V(s_t)$ is the **TD error**.

DP vs MC vs TD

DP:

$$V(s_t) = \sum_a \pi(a|s_t) \sum_{s',r'} p(s', r|s_t, a)[r + \gamma V(s')]$$

MC:

$$V(s_t) = V(s_t) + \alpha(G_t - V(s_t))$$

TD(0):

$$V(s_t) = V(s_t) + \alpha(R_{t+1} + \gamma V(s_{t+1}) - V(s_t))$$

Boostrapping:

- MC does not bootstrap
- DP bootstraps
- TD bootstraps

Sampling:

- MC samples
- DP doesn't sample
- TD samples

Bias-variance:

- MC has high variance, but zero bias
- TD has low variance, but some bias

Return depends on *many* random actions, transitions, rewards. On the other hand, TD target depends on *one* random action, transition, reward.

Convergence:

- MC: Good convergence, even with Function Approximation
- MC: insensitive to initialization (no bootstrapping)
- TD: converges, but be careful with FA
- TD: sensitive to initialization because of bootstrapping

TD is more efficient in Markovian environments because it exploits Markov Property
MC is more efficient in non-Markovian environments.