Monte Carlo Methods

Monte Carlo Methods

- Do not require model of the environment, only experience
- Solve the RL problem by sampling and averaging sample returns
- Only works for episodic tasks
    - Value estimates and policies are only updated after an episode terminates

Monte Carlo Prediction

- Learns the state-value function for a given policy
- Expected value is calculated through experience
    - Average returns from experience
- First-Visit MC Method
    - Estimates $\mathbf{v_\pi(s)}$ as the average of the returns following first visits to **s**
    - Standard deviation of value estimate falls as $1/\sqrt{n}$, where n is the number of returns averaged
- Every-Visit MC Method
    - Estimates $\mathbf{v_\pi(s)}$ as the average of the returns following all visits to **s**
    - Converges quadratically to $\mathbf{v_\pi(s)}$

First-visit MC prediction, for estimating $V \approx v_\pi$

Initialize:
    $\pi \leftarrow$ policy to be evaluated
    $V \leftarrow$ an arbitrary state-value function
    $Returns(s) \leftarrow$ an empty list, for all $s \in \mathcal{S}$

Repeat forever:
    Generate an episode using π
    For each state s appearing in the episode:
        $G \leftarrow$ the return that follows the first occurrence of s
        Append G to $Returns(s)$
        $V(s) \leftarrow \text{average}(Returns(s))$

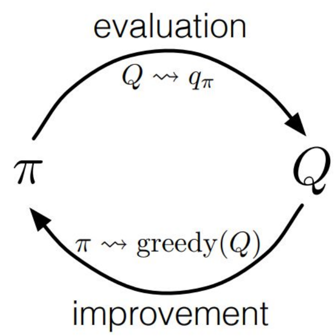
Monte Carlo Estimation of Action Values

- Without a model, state values alone are not sufficient for policy
- Estimates **q_π(s, a)**
  - Averages returns when in state **s** and taking action **a**
- Exploration
  - If π is a deterministic policy, then π will only observe returns for one action from each state
  - Maintaining Exploration with Exploring Starts
    - Start episode in a random state-action pair
    - Exploring starts are often not feasible
      - On-policy and off-policy methods are used to solve this issue

Monte Carlo ES (Exploring Starts), for estimating $\pi \approx \pi_*$

Initialize, for all $s \in \mathcal{S}$, $a \in \mathcal{A}(s)$:
    $Q(s,a) \leftarrow$ arbitrary
    $\pi(s) \leftarrow$ arbitrary
    $Returns(s,a) \leftarrow$ empty list

Repeat forever:
    Choose $S_0 \in \mathcal{S}$ and $A_0 \in \mathcal{A}(S_0)$ s.t. all pairs have probability > 0
    Generate an episode starting from S_0, A_0, following π
    For each pair s, a appearing in the episode:
        $G \leftarrow$ the return that follows the first occurrence of s, a
        Append G to $Returns(s,a)$
        $Q(s,a) \leftarrow \text{average}(Returns(s,a))$
    For each s in the episode:
        $\pi(s) \leftarrow \text{argmax}_a Q(s,a)$



MC Control Improvement Cycle

Monte Carlo Control

- Using Monte Carlo simulation to approximate optimal policies
- Cycle of MC Improvement
  - Policy Evaluation - Value function is altered to more closely approximate the value function for the current policy
    - Uses MC estimation of action values
  - Policy Improvement - Policy is improved with respect to the current value function
    - Make policy greedy with respect to the current function
  - Evaluation and Improvement are alternated on an episode-by-episode basis
- Approximation
  - Measurements and assumptions can be made to obtain the magnitude and probability of error in value and policy estimates
  - Don't complete policy evaluation before returning to policy improvement
    - Each evaluation step moves value function towards **q^*_π**

On-Policy Methods

- Attempt to evaluate or improve the policy that is used to make decisions
- Uses soft policies
    - Policy starts with a non-zero chance of selecting each state-action pair, and becomes closer to a deterministic policy over time
- Uses Epsilon-greedy policies
    - Most of the time, the policy is greedy, but with probability epsilon it selects an action at random
        - Minimal chance of a non-greedy action being selected is $\frac{\varepsilon}{|A(s)|}$
- Policy Improvement approximation
    - Each improvement step moves policy towards a greedy policy
    - If policy immediately became greedy, it wouldn't explore

On-policy first-visit MC control (for ε-soft policies), estimates $\pi \approx \pi_*$

Initialize, for all $s \in \mathcal{S}$, $a \in \mathcal{A}(s)$:
  $Q(s,a) \leftarrow$ arbitrary
  $Returns(s,a) \leftarrow$ empty list
  $\pi(a|s) \leftarrow$ an arbitrary ε-soft policy

Repeat forever:
  (a) Generate an episode using π
  (b) For each pair s, a appearing in the episode:
    $G \leftarrow$ the return that follows the first occurrence of s, a
    Append G to $Returns(s,a)$
    $Q(s,a) \leftarrow \text{average}(Returns(s,a))$
  (c) For each s in the episode:
    $A^* \leftarrow \arg\max_a Q(s,a)$ (with ties broken arbitrarily)
    For all $a \in \mathcal{A}(s)$:

$$\pi(a|s) \leftarrow \begin{cases} 1 - \varepsilon + \varepsilon/|\mathcal{A}(s)| & \text{if } a = A^* \\ \varepsilon/|\mathcal{A}(s)| & \text{if } a \neq A^* \end{cases}$$

Off-Policy Methods

- Learning control methods seek to learn action values conditional on subsequent optimal behaviour, but they need to behave non-optimally in order to explore all action
    - On-policy approach compromises by learning action values for a near-optimal policy that still explores
- Off-Policy
    - Slower to converge
    - Are of greater variance
    - More powerful and general
    - Can learn from a human expert
- Off-policy approach uses two policies
    - Target Policy
        - The policy being learned about - is improved over time
    - Behavior Policy
        - The policy used to generate behavior - is static
        - Is a soft policy - samples all actions in all states with nonzero probability

- Prediction Problem
  - Both target and behavior policies are fixed
    - Suppose we wish to estimate $\mathbf{v}_\pi$ or $\mathbf{q}_\pi$ but all we have are episodes following another policy **b**, where **b** ≠ $\boldsymbol{\pi}$
      - $\boldsymbol{\pi}$ is the target policy, **b** is the behavior policy
  - In order to use episodes from **b** to estimate values for $\boldsymbol{\pi}$
    - *Coverage* is needed
      - Every action take under $\boldsymbol{\pi}$ is also taken, at least occasionally, under **b**
      - **b** must be stochastic in states where it is not identical to $\boldsymbol{\pi}$
      - $\boldsymbol{\pi}$ is usually deterministic-greedy with respect to the current action-value function estimate
- Importance sampling
  - Technique for estimating expected values under one distribution given samples from another
  - Importance-Sampling Ratio
    - Returns are weighted (given importance) according to the relative probability of their trajectories (A_t, S_{t+1}, A_{t+1} … S_T) occurring under the target and behavior policies
    - Ratio is trajectory probability of each policy, Target:Behavior

Off-Policy MC Prediction

Off-policy MC prediction, for estimating $Q \approx q_\pi$

Input: an arbitrary target policy π

Initialize, for all $s \in \mathcal{S}$, $a \in \mathcal{A}(s)$:
- $Q(s,a) \leftarrow$ arbitrary
- $C(s,a) \leftarrow 0$

Repeat forever:
- $b \leftarrow$ any policy with coverage of π
- Generate an episode using b:
  - $S_0, A_0, R_1, \ldots, S_{T-1}, A_{T-1}, R_T, S_T$
- $G \leftarrow 0$
- $W \leftarrow 1$
- For $t = T-1, T-2, \ldots$ down to 0:
  - $G \leftarrow \gamma G + R_{t+1}$
  - $C(S_t, A_t) \leftarrow C(S_t, A_t) + W$
  - $Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \frac{W}{C(S_t,A_t)}\left[G - Q(S_t, A_t)\right]$
  - $W \leftarrow W \frac{\pi(A_t|S_t)}{b(A_t|S_t)}$
  - If $W = 0$ then exit For loop

Off-Policy MC Control

Off-policy MC control, for estimating $\pi \approx \pi_*$

Initialize, for all $s \in \mathcal{S}$, $a \in \mathcal{A}(s)$:
- $Q(s,a) \leftarrow$ arbitrary
- $C(s,a) \leftarrow 0$
- $\pi(s) \leftarrow \operatorname{argmax}_a Q(S_t, a)$ (with ties broken consistently)

Repeat forever:
- $b \leftarrow$ any soft policy
- Generate an episode using b:
  - $S_0, A_0, R_1, \ldots, S_{T-1}, A_{T-1}, R_T, S_T$
- $G \leftarrow 0$
- $W \leftarrow 1$
- For $t = T-1, T-2, \ldots$ down to 0:
  - $G \leftarrow \gamma G + R_{t+1}$
  - $C(S_t, A_t) \leftarrow C(S_t, A_t) + W$
  - $Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \frac{W}{C(S_t,A_t)}\left[G - Q(S_t, A_t)\right]$
  - $\pi(S_t) \leftarrow \operatorname{argmax}_a Q(S_t, a)$ (with ties broken consistently)
  - If $A_t \neq \pi(S_t)$ then exit For loop
  - $W \leftarrow W \frac{1}{b(A_t|S_t)}$

$$\prod_{k=t}^{T-1} \frac{\pi(A_k|S_k)}{b(A_k|S_k)}$$

Importance-Sampling Ratio