Mitigating the Curse of Horizon in Monte-Carlo Returns

Gersi Doko

Department of Computer Science
University of New Hampshire

MDP

- A Markov Decision Process (MDP) is a tuple $(S, A, f, R, \gamma, \eta)$

- Here we consider continuous MDPs, with deterministic transitions and rewards for simplicity

- $\frac{ds(t)}{dt} = f(s(t), a(t))$

Monte-Carlo Returns Discrete Case

Very common to estimate the value of a policy by sampling returns over M trajectories each of length T $[(s_t^m, a_t^m, r_t^m)_{t=0}^T]_{m=0}^M$

$$\hat{G}_m^\pi = \sum_{t=0}^{T} \gamma^t \tilde{R}_t^m$$

$$\hat{V}_M^\pi = \frac{1}{M} \sum_{m=0}^{M} \hat{G}_m^\pi$$

What is the relationship between M, T and $||\tilde{V}_\pi - V_\pi||_1$?

Monte-Carlo Returns Continuous Case

To investigate this question the paper considers the continuous time case,

$$G_T^\pi = \int_0^T \gamma^t r(s_t, \pi(s_t))\, dt$$

$$V_T^\pi = \mathbb{E}[G_T^\pi \mid s_0 \sim \eta]$$

approximate the above integral over T using discretization $N = [n_0, n_1, \dots]$

$$\hat{G}_m^\pi = \sum_{n=0}^{N} \gamma^t \bar{r}_n^m$$

$$\hat{V}_M^\pi = \frac{1}{M} \sum_{m=0}^{M} \hat{G}_m^\pi$$

Monte-Carlo Returns Continuous Case

$$\bar{r}_m(n) = \frac{\gamma^{t_n} r_m(t_n) + \gamma^{t_{n-1}} r_m(t_{n-1})}{2}(t_n - t_{n-1})$$

$$\hat{G}^{\pi}_m = \sum_{n=0}^{N} \gamma^t \bar{r}_m(n)$$

$$\hat{V}^{\pi}_M = \frac{1}{M} \sum_{m=0}^{M} \tilde{G}^{\pi}_m$$

What is the relationship between M, N and $||\hat{V}^{\pi}_M - V_{\pi}||_1$?

Goal

- We have a fixed computation budget $B = M \cdot N$
- Want to minimize $||\hat{V}_M^\pi - V_\pi||_1$

- **How should we allocate M and N?**

- *Approach*: Allocate N first, them $M = B/N$

Adaptive

Algorithm 1 ADAPTIVE

To approximate $\int_{\tau_1}^{\tau_2} r(t)dt$ within tolerance ε.

Input: The rewards r, the limits of integration τ_1 and τ_2, and the tolerance ε

$\tau_3 = \frac{\tau_1+\tau_2}{2}$

$Q_{\tau_i,\tau_j} = \frac{\gamma^{\tau_i} r(\tau_i)+\gamma^{\tau_j} r(\tau_j)}{2}(\tau_j - \tau_i)$ for $(i,j) = \{(1,2),(1,3),(3,2)\}$.

if $|Q_{\tau_1,\tau_2} - Q_{\tau_1,\tau_3} - Q_{\tau_3,\tau_2}| > \varepsilon$ **then**

  $Q = \text{ADAPTIVE}(r,\tau_1,\tau_3,\varepsilon/2) + \text{ADAPTIVE}(r,\tau_3,\tau_2,\varepsilon/2)$

else

  $Q = Q_{\tau_1,\tau_2}$

end if

return Q

Figure: Adaptive choice of discretization.

Uniform

Algorithm 2 UNIFORM

To approximate $\int_a^b r(t)dt$ with uniformly spaced points.
Input: The rewards r, the number of points N.
$h = \frac{b-a}{N-1}$
$Q = h \cdot \frac{\gamma^{t_1} r(t_1) + \gamma^{t_2} r(t_2)}{2}$
for $i = 0, \ldots, N-1$ **do**
    $t_i = a + ih$
    $Q = Q + h \cdot \gamma^{t_i} r(t_i)$
end for
return Q

Figure: Uniform choice of discretization.

Experiments

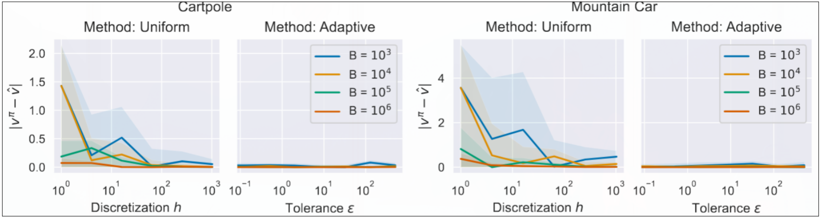

Figure: Experiments comparing Adaptive and Uniform discretization.

Experiments Cont.

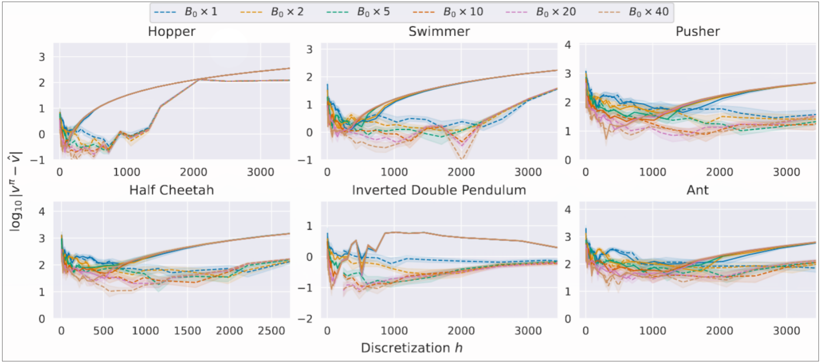

Figure: Dashed line is adaptive, solid line is uniform.

Conclusion

- Don't always use all samples from Monte-Carlo rollouts
- Adaptive discretization is better than Uniform generally
- Better to use more samples with smaller discretization in uncertain regions of the state space