PAC Monte-Carlo Policy Evaluation with Restart Access

Shivaram Kalyanakrishnan

September 22, 2017

Abstract

We present a simple *Probably Approximately Correct* (PAC) algorithm for learning the value function of an MDP with which an agent interacts by sequentially taking actions. The MDP may be either *continuing* or *episodic*. To ease the burden of exploration, we assume that at any point, the agent may choose to terminate its current trajectory. If it does so, it is restarted from a state picked in a round-robin fashion. This sort of restart access is slightly different from the common assumption of exploring starts in episodic MDPs, under which the agent cannot choose to terminate an episode, but once an episode does terminate, it can decide from which state to restart.

In order to estimate the value of a state, our learning algorithm generates a large number, say N, of trajectories starting from that state. Each trajectory is run either until episode termination, or until M steps are completed, whichever happens first. The idea is to (1) make N large enough that the empirical average of the truncated returns from the trajectories will lie close to their true mean with high probability, and to (2) make M large enough that the contribution to the long-term reward beyond M transitions is small. The obtained estimate is the average M-step reward from the N trajectories. We provide a proof of correctness.

1 Problem Definition

We consider a Markov Decision Problem (MDP) $M = (S, A, R, T, \gamma)$, which has a set of states S in which an agent can be, and the set of actions A that the agent can execute. Upon taking action $a \in A$ from state $s \in S$, the agent is transported to a state s', selected from S at random with probability $T(s, a, s')$. The transition also yields the agent an immediate reward $R(s, a, s')$, which we shall assume lies in the range $[-R_{\max}, R_{\max}]$, with $R_{\max}$ being a *known* non-negative quantity (thus, $R_{\max}$ can be used by an algorithm). We also assume that S is finite, with $|S| = n \geqslant 1$.

Given a policy $\pi : S \to A$, the *value* of state $s \in S$ under π is defined as

$$V^{\pi}(s) \stackrel{\text{def}}{=} \mathbb{E}_{\pi}\left[\sum_{t=0}^{\infty} \gamma^t r^t | s^0 = s\right],$$

where $\gamma \in [0, 1)$ is a discount factor. Here $r^0, r^1, \ldots$ is the sequence of rewards obtained by the agent over time.

For a given tolerance $\epsilon > 0$, a function $X : S \to \mathbb{R}$ is said to be an ϵ-approximation of V^{π} if for every $s \in S$,

$$|X(s) - V^{\pi}(s)| \leqslant \epsilon.$$

Given $\delta \in (0, 1]$, our aim is to write down a procedure for the agent to interact with the MDP and produce an output $\hat{V}^{\pi} : S \to \mathbb{R}$ such that with probability at least $1 - \delta$, $\hat{V}^{\pi}$ is an ϵ-approximation of V^{π}.

Observe that there is no need for such a PAC algorithm in the *planning* setting: when T and R are known to the agent, it can simply solve Bellman's Equations to compute V^{π} *exactly*. On

the other hand, it is due to the unavailability of T and R—except indirectly through sampling—that there arises a need to reconstruct reality as accurately as desired, in a PAC sense, based on a finite number of samples.

2 Algorithm

Given π, we compute $\hat{V}^\pi$, a "close-enough" approximation of V^π, by generating for every state $s \in S$

$$N \stackrel{\text{def}}{=} \left\lceil \frac{8R_{max}^2}{\epsilon^2(1-\gamma)^2} \ln\left(\frac{2n}{\delta}\right) \right\rceil$$

trajectories, each with at most

$$M \stackrel{\text{def}}{=} \left\lceil \left(\frac{1}{1-\gamma}\right) \ln\left(\frac{2R_{max}}{\epsilon(1-\gamma)}\right) \right\rceil$$

actions taken, wherein the first state is s, and every action is taken according to π. Observe that the total number of transitions generated by our procedure is therefore at most nNM.

Let us consider the N trajectories obtained from a particular state s. For trajectory $1 \leqslant j \leqslant N$, let the rewards obtained be $r_j^0, r_j^1, \ldots, r_j^{M-1}$. If the episode has naturally terminated in $T < M$ steps, take $r_j^T = r_j^{T+1} = \cdots = r_j^{M-1} = 0$. Note that if M steps have been reached, or the episode has ended earlier, the agent starts a new trajectory. Let V_j^M be the M-step discounted reward for trajectory j:

$$V_j^M = \sum_{t=0}^{M-1} \gamma^t r_j^t.$$

After the N trajectories have been obtained from s, the agent computes its estimate $\hat{V}^\pi(s)$ as the average of these N M-step discounted rewards:

$$\hat{V}^\pi(s) = \frac{1}{N} \sum_{j=1}^{N} V_j^M.$$

We have set M and N such that our PAC analysis will go through. M is large enough that the M-step discounted return is close enough to the corresponding infinite-discounted return. N is large enough to ensure that our estimate of the expected M-step discounted return is sufficiently accurate.

3 Analysis

The following lemma shows that our algorithm enjoys the desired PAC guarantee.

Lemma 1. *With probability at least $1-\delta$, for every $s \in S$:*

$$|\hat{V}^\pi(s) - V^\pi(s)| < \epsilon.$$

Proof. Our proof is in three steps: the first and second steps apply to every given state s; the third step aggregates the mistake probabilities over the set of states.

Step 1. Consider a fixed state s from which we have applied the estimation procedure. First we consider $V^M(s)$, a random variable that denotes the M-step *expected* discounted reward:

$$V^M(s) = \mathbb{E}\left[\sum_{t=0}^{M-1} \gamma^t r^t | s^0 = s\right].$$

Thus, $V_1^M, V_2^M, \ldots, V_N^M$ are i.i.d. samples of $V^M(s)$. By Hoeffding's Inequality, we get that the probability of their empirical average—which is our estimate $\hat{V}^\pi(s)$—deviating from the true mean $V^M(s)$ is small. Note that $V^M(s)$ must lie in the interval $\left[-\frac{R_{max}}{1-\gamma}, \frac{R_{max}}{1-\gamma}\right]$.

$$\mathbb{P}\left\{\hat{V}^\pi(s) > V^M(s) + \frac{\epsilon}{2}\right\} \leqslant e^{-2N\left(\frac{\epsilon}{2}\right)^2 \frac{(1-\gamma)^2}{4R_{max}^2}} \leqslant \frac{\delta}{2n}.$$

Similarly, we get

$$\mathbb{P}\left\{\hat{V}^\pi(s) < V^M(s) - \frac{\epsilon}{2}\right\} \leqslant e^{-2N\left(\frac{\epsilon}{2}\right)^2 \frac{(1-\gamma)^2}{4R_{max}^2}} \leqslant \frac{\delta}{2n}.$$

Putting these results together, we obtain that with probability at least $1 - \frac{\delta}{n}$,

$$V^M(s) - \frac{\epsilon}{2} \leqslant \hat{V}^\pi(s) \leqslant V^M(s) + \frac{\epsilon}{2}. \tag{1}$$

Step 2. Our second step is to show that for every state $s \in S$, $V^M(s)$ is itself within $\frac{\epsilon}{2}$ of the (true) value $V^\pi(s)$.

$$\begin{aligned}
\left|V^\pi(s) - V^M(s)\right| &= \left|\mathbb{E}_\pi\left[\sum_{t=0}^{\infty} \gamma^t r^t | s^0 = s\right] - \mathbb{E}_\pi\left[\sum_{t=0}^{M-1} \gamma^t r^t | s^0 = s\right]\right| \\
&= \left|\mathbb{E}_\pi\left[\sum_{t=M}^{\infty} \gamma^t r^t | s^0 = s\right]\right| \\
&\leqslant \sum_{t=M}^{\infty} \gamma^t R_{max} \\
&= \frac{\gamma^M R_{max}}{1-\gamma} \\
&\leqslant \gamma^M \frac{\epsilon}{2} e^{(1-\gamma)M} \\
&\leqslant \frac{\epsilon}{2}
\end{aligned}$$

since $\gamma e^{1-\gamma}$ is at most 1 for $\gamma \in [0, 1)$. Thus:

$$V^\pi(s) - \frac{\epsilon}{2} \leqslant V^M(s) \leqslant V^\pi(s) + \frac{\epsilon}{2}. \tag{2}$$

Step 3. We put together (1) and (2) to conclude that for a given state $s \in S$, the probability that $|\hat{V}^\pi(s) - V^\pi(s)| \geqslant \epsilon$ is at most $\frac{\delta}{n}$. Hence, by way of a union bound, with probability at least $1 - \delta$: $\forall s \in S : |\hat{V}^\pi(s) - V^\pi(s)| < \epsilon$. In other words, with probability at least $1 - \delta$, $\hat{V}^\pi$ is an ϵ-approximation of V^π. □

The sample complexity incurred by our algorithm is upper-bounded by

$$nNM = O\left(\frac{nR_{max}^2}{\epsilon^2(1-\gamma)^3} \ln\left(\frac{n}{\delta}\right) \ln\left(\frac{R_{max}}{\epsilon(1-\gamma)}\right)\right).$$

Does the dependence on n, ϵ, δ, γ, and R_{max} agree with your intuition?