TRUNCATING VS. DISCOUNTING MONTE CARLO ESTIMATES OF RETURNS

David Krueger

ABSTRACT

Given the equivalence of discounting to a state-action independent probability of termination, Monte Carlo estimates of returns can be produced by replacing discounting with stochastic termination. Doing so increases the number of episodes which can be completed in a given period of time, at the expense of increasing the variance of the estimated returns. We empirically evaluate and extend this approach by expressing the discount factor as a product of a termination probability and a pseudo-discount parameter: $\gamma = \tilde{\gamma}\tilde{p}_\perp$.

1 INTRODUCTION

The disadvantages of truncating rewards are significant:

1. Stochastic termination adds another source of randomness, increasing variance.
2. For every-visit MC, truncation does not increase the number of unbiased updates (since the total number of visits in a fixed time-budget is approximately constant), but *does* decrease the expected number of reward observations which are used to generate the estimate for each update.

On the other hand, there are several reasons truncating might be expected to be advantageous. Firstly, for environments without terminal states, episodes have infinite length, and there is no alternative to truncation. However, we're interested in potential benefits for environments that do terminate (with probability 1) after a finite number of steps.

We identify two potential settings which may yield advantages for truncation:

1. Returns may be highly correlated across time.
2. In a highly "local" environment, where it may take many time-steps to transition between some areas of state-space, having more episodes (via truncation) may decrease the expected number of state-actions which the agent never visits, allowing more state-actions to be updated. For Gaussian distributed returns, the error of $Q(s, a)$ would decrease as the square root of the number of updates, so maximizing the entropy of the visit distribution would bring the greatest reduction in the MSE error of the Q estimate[1]. An extreme case of this is when the Markov Chain induced by the agent's policy is non-ergodic.
3. For first-visit MC, truncation could increase the number of updates an algorithm can make when states are revisited.

In this work, we restrict our experiments to the every-visit case, and demonstrate benefits of truncation in non-ergodic environments.

2 INTERPOLATING BETWEEN TRUNCATING AND DISCOUNTING

Our proposed interpolation simply writes $\gamma = \tilde{\gamma}\tilde{p}_\perp$, and we restrict $\tilde{\gamma}, \tilde{p}_\perp \in (\gamma, 1]$.

Proposition 5.3.1 of **?** demonstrates the equivalence of discounting with geometric random variable stopping time (i.e. truncation). We don't replicate the proof here, but explain why it generalizes to

[1] ...although not necessarily the greatest reduction in expected regret of the induced policy, since learning about more commonly visited areas might be more useful

our proposed algorithms in a straightforward way. We first define $\tilde{r}_t \doteq r_t \tilde{\gamma}^{t-1}$, that is, the pseudo-discounted reward, and replace $r_t \doteq r(X_t, y_t)$ in the proof with $\tilde{r}$ and replace γ (called λ in **?**) with $\tilde{p}_\perp$. By assumption, $\tilde{\gamma} < 1$ (else there is nothing to prove), and thus we may exchange the order of the sums, since $|\tilde{r}_t| < |r_t|$ and thus the series is dominated and also converges. Now we're left with:

$$v^\pi_{\tilde{\gamma}}(s) = \mathbb{E}^\pi_s[\sum_{t=1}^{\infty} \tilde{p}_\perp^{t-1} \tilde{r}_t] \tag{1}$$

$$= \mathbb{E}^\pi_s[\sum_{t=1}^{\infty} (\tilde{p}_\perp \tilde{\gamma})^{t-1} r_t] \tag{2}$$

$$= \mathbb{E}^\pi_s[\sum_{t=1}^{\infty} \gamma^{t-1} r_t] \tag{3}$$

as desired.

3 "Lanes" environments

Our original experiments used grid-world and random-walk environments, but we never observed any benefits to truncation in these environments, and we do not report the results here.

Instead we compare these algorithms in four environments which are designed to favor truncating rewards. In all four environments, we can view the agent's first action as selecting one of several vertically stacked "lanes" (i.e. states in $\mathcal{S} \setminus \{s_0, s_\perp\} = \{s_1, ..., s_9\}$). The selection is deterministic: $P(s_i|a_i, s_o) = 1$.

In the first environment ("lanes1"), $\mathcal{R}(s_0, a_k) = k$, and rewards are 0 after the first time-step, so subsequently terminating doesn't change the estimate of the returns.

In the second environment("lanes2"), the agent cannot leave the lane it initially selects, but accrues $\mathcal{R}(s_k, a) = k$ at every subsequent time-step, so returns are perfectly correlated with the initial reward.

In the third environment("lanes3"), subsequent actions can only move the agent to neighboring states within the column, so returns are strongly correlated with the initial choice of action.

In the fourth environment("lanes4"), subsequent actions teleport the agent to any state within the column, so returns are only weakly correlated with the initial choice of action.

All these environments are non-ergodic, since the start-state is never revisited. Lanes1-3 are highly local environments, since transitioning between the other states is difficult to impossible.

4 Experiments

We compare algorithms' performance as a function of wall-clock time. In all experiments, environments have a fixed start state, s_0, discount factor of $\gamma = 0.9$ and an *innate* state-action independent termination probability of $p_\perp = 0.001$. We consider both control and on-policy evaluation; the behavior policy, μ always chooses actions at uniform random.

We run experiments with 10 and 20 (non-terminal) states. We expect and show that, the non-ergodicity of the environments is more of an issue when there are more states; even in lanes4 the agent will need at least $|\mathcal{S}| - 1$ episodes to visit all of the actions from the start-state [2].

4.1 Results

First, For each environment, we plot the mean of 30 experiments along with standard error bars.

For environments with 10 states, we compare evaluation (Figure 1) and control (Figure 2); results are similar in these two cases. For lanes1, all methods achieve 0 error, and more truncation speeds this process. For lanes2-4, we find that some degree of truncation can be helpful, but mixing truncation

[2] Being an example of the coupon collector's problem, it is expected to take $\Theta(n\log(n))$ episodes.

and discounting ($\tilde{p}_\perp = .5\gamma$) consistently outperforms replacing discounting with truncation entirely ($\tilde{p}_\perp = 1\gamma$). For lanes2, truncation is superior up to a time-budget of 0.3 seconds. For lanes3-4, the pattern is more complex: truncation begins to outperform only after 0.001 seconds, but is again overtaken around 0.3 seconds.

For the larger (20 state) environment (Figure 3), the truncation methods advantage grows, as expected.

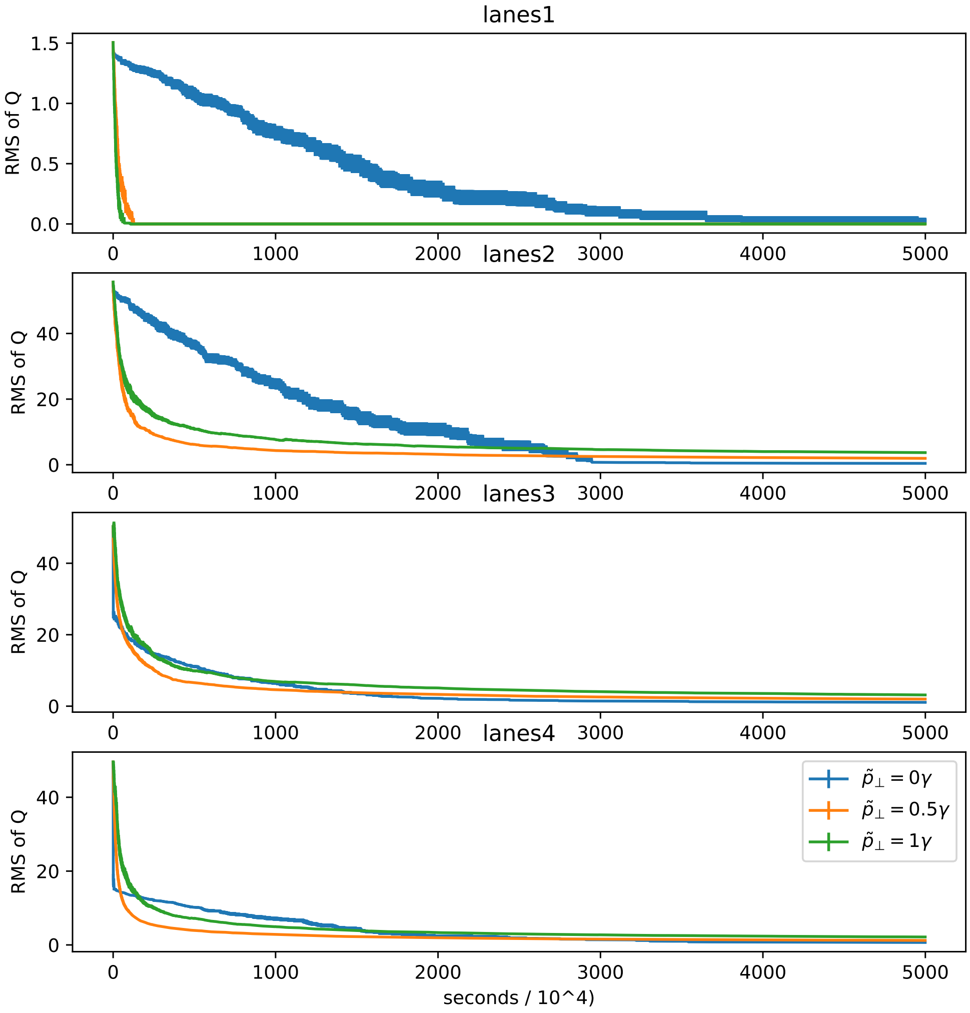

Figure 1: Monte Carlo evaluation in the "lanes" environments with 10 states.

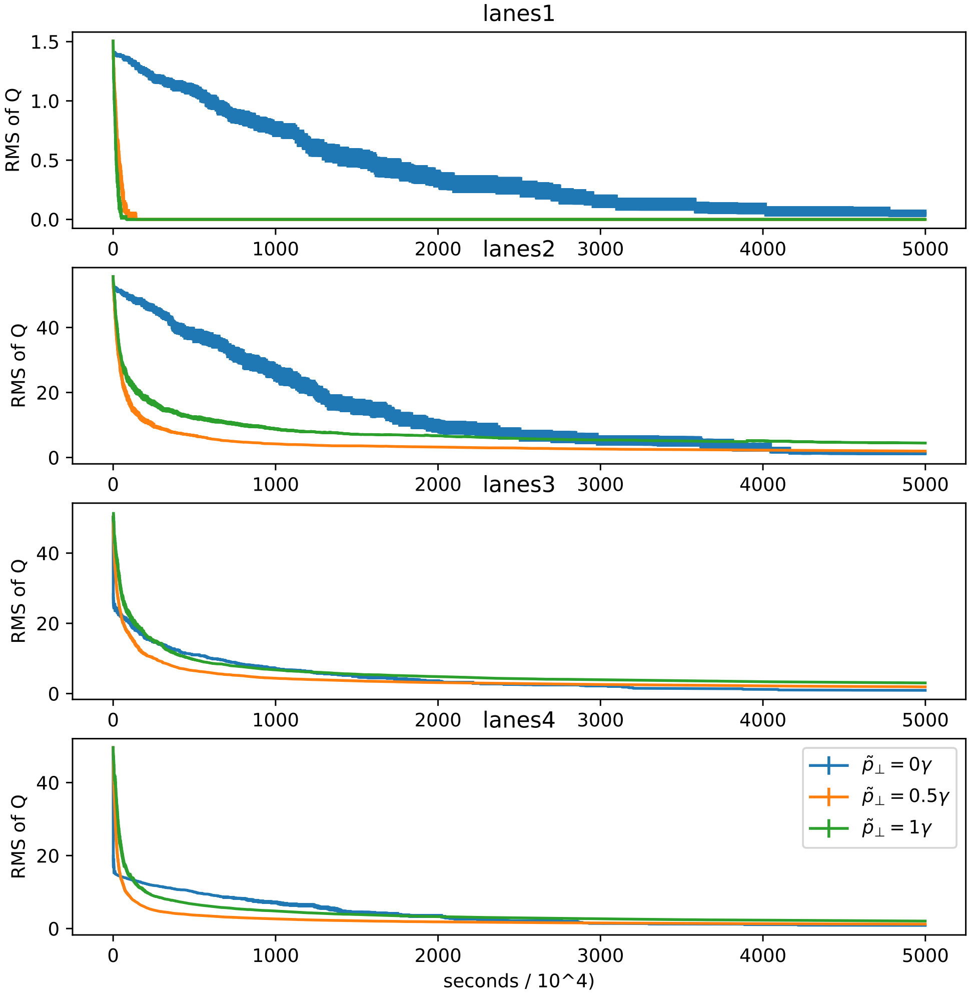

Figure 2: Monte Carlo control in the "lanes" environments with 10 states.

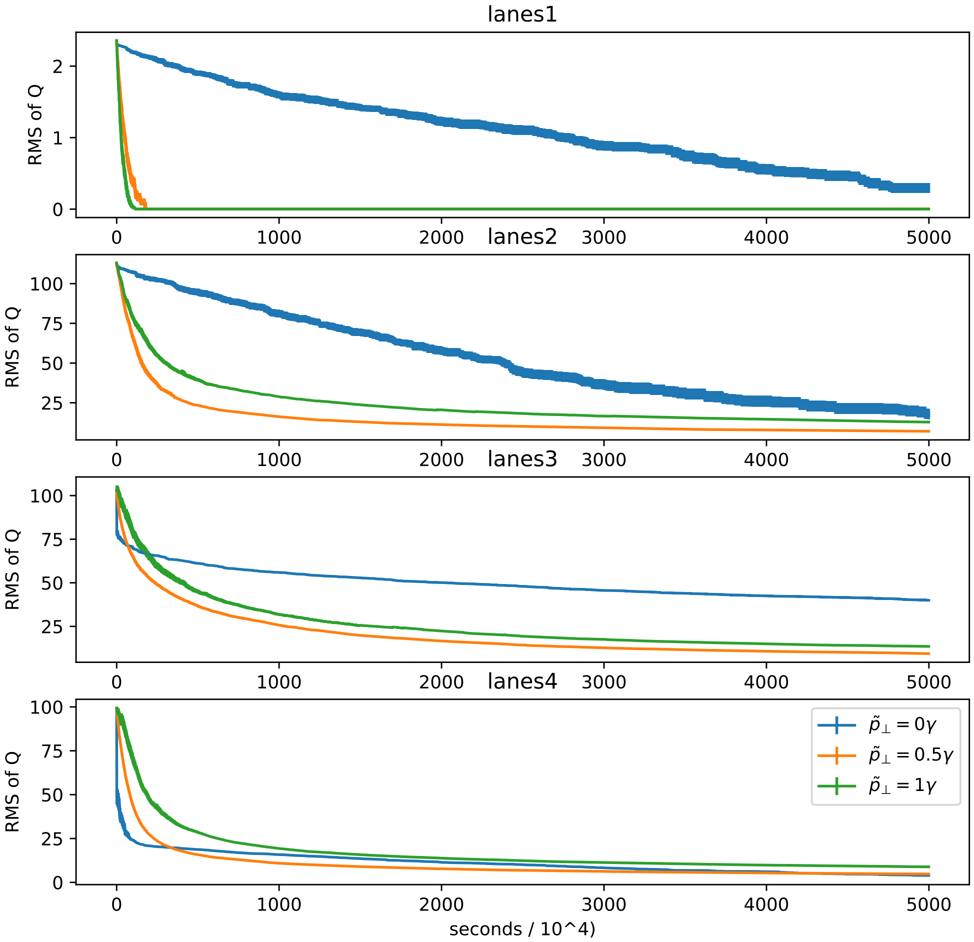

Figure 3: Monte Carlo evaluation in the "lanes" environments with 20 states.