Renewal Monte Carlo: A Renewal Theory Based Reinforcement Learning Algorithm

Paper ID: 2241

Abstract

In this paper, a new Monte Carlo based online reinforcement learning algorithm, called Renewal Monte Carlo (RMC), for infinite horizon models with a designated start state is introduced. RMC retains the advantages of the Monte Carlo approach including low bias, simplicity and ease of implementation, while circumventing its key drawbacks of high variance and delayed (end of episode) updates. RMC works for both discounted and average cost setups as well as for models with either discrete or continuous state and action spaces.

The key idea of RMC is as follows. Under any reasonable policy, the closed loop system is positive recurrent and there are states that are visited infinitely often. One of these recurrent states is picked as a reference state. Successive visits to the reference state is viewed as a regenerative process. When the reference state is same as the start state, the expected discounted reward for infinite horizon is equal to the ratio of the expected discounted first passage reward and the expected discounted first passage time. Using this idea, sample path based estimates of performance and performance derivatives are derived.

Detailed numerical studies are presented to compare the performance of RMC with temporal difference (SARSA lambda) methods. The numerical experiments demonstrate that RMC has lower bias compared to temporal difference methods while achieving almost the same variance. Hence, RMC is an attractive alternative to temporal difference methods for infinite horizon models with designated start state.

1 Introduction

Policy iteration based methods for reinforcement learning have two critical components: policy evaluation and policy improvement. In the policy evaluation step, the performance of a parameterized policy is evaluated while in the policy improvement step, the policy parameters are updated using stochastic gradient ascent.

Policy evaluation methods may be broadly classified as Monte Carlo (MC) methods and temporal difference (TD) methods. In MC methods, performance of a policy is estimated using the discounted return of a single sample path; in TD methods, the value(-action) function is guessed and this guess is iteratively improved using temporal differences. MC methods are attractive because they have zero bias, are simple and easy to implement, and work for both discounted and average reward setups as well as for models with continuous state and action spaces. However, they suffer from various drawbacks. First, they have a high variance because a single sample path is used to estimate performance. Second, they are not asymptotically optimal for infinite horizon models because it is effectively assumed that the model is episodic; in infinite horizon models, the trajectory is arbitrarily truncated to treat the model as an episodic model. Third, the policy improvement step cannot be carried out in tandem with policy evaluation. One must wait until the end of the episode to estimate the performance and only then can the policy parameters be updated. It is for these reasons that MC methods are largely ignored in the literature, which almost exclusively focuses on TD methods.

1.1 Brief summary of the results

In this paper, we propose a MC method—which we call *Renewal Monte Carlo* (RMC)—for infinite horizon models with a designated start state. Like MC, RMC has low bias, is simple and easy to implement, and works for models with continuous state and action spaces. However, it does not suffer from the drawbacks of MC. RMC is a low-variance online algorithm that works for infinite horizon discounted and average reward setups. One doesn't have to wait until the end of the episode to carry out the policy improvement step; it can be carried out whenever the system visits a designated reference state.

Suppose the policies are parameterized by θ taking values in some set Θ. Given a θ, our key idea for policy evaluation is that one need not wait until the end of the episode to estimate the performance. Instead, under any reasonable policy, the reward process is ergodic and visits the recurrent states infinitely often. Let's pick one of these states as a reference state. For simplicity, we'll assume that the reference state is the start state. (We show later that the argument also works for arbitrary reference states). Let τ_1, τ_2, etc. denote the stopping times for successive visits to the reference state and let R_n and T_n denote the discounted reward and discounted time, respectively, from τ_{n-1} to τ_n, with discount rate γ.

By the strong Markov property, $\{\mathsf{R}_n\}_{n\geq 1}$ and $\{\mathsf{T}_n\}_{n\geq 1}$ are i.i.d. processes. Let R_θ and T_θ denote $\mathbb{E}[\mathsf{R}_n]$ and $\mathbb{E}[\mathsf{T}_n]$, respectively. Sample path based unbiased estimators of R_θ

and T_θ are given by

$$\widehat{\mathsf{R}}_N = \frac{1}{N}\sum_{n=1}^{N} \mathsf{R}_n \quad \text{and} \quad \widehat{\mathsf{T}}_N = \frac{1}{N}\sum_{n=1}^{N} \mathsf{T}_n,$$

respectively, where N is a large number. These estimates have low variance because they are obtained by averaging multiple i.i.d. random variables.

At each visit to the reference state, the controlled Markov process *regenerates*. Thus, from renewal theory (Feller 1966), we get that the performance J_θ of the policy θ equals $\mathsf{R}_\theta/(1-\gamma)\mathsf{T}_\theta$, which, may be approximated as $\widehat{\mathsf{R}}_N/(1-\gamma)\widehat{\mathsf{T}}_N$. Although, $\widehat{\mathsf{R}}_N/(1-\gamma)\widehat{T}_N$ is a biased estimator of J_θ, we show that it satisfies a concentration inequality. Therefore, the bias goes to zero exponentially fast.

For policy improvement, we carry out a stochastic approximation update at any visit to the recurrence state. However, instead of using a stochastic approximation update to solve $\nabla_\theta J_\theta = 0$, we use a stochastic approximation update to solve

$$\mathsf{T}_\theta \nabla_\theta \mathsf{R}_\theta - \mathsf{R}_\theta \nabla_\theta \mathsf{T}_\theta = 0.$$

Both these equations have the same roots, but the latter has the advantage that we only need to estimate the gradients of R_θ and T_θ rather than that of J_θ.

We present two methods for estimating the gradients of R_θ and T_θ. The first is a likelihood ratio (or score function) based gradient estimator which works when the policy is differentiable with respect to the policy parameters. The second is a simultaneous perturbation based gradient estimator that uses finite differences, which is useful when the policy is not differentiable with respect to the policy parameters.

We present detailed numerical studies to compare the performance of RMC with temporal difference (SARSA-λ) methods. We first present a toy model with discrete state space and compare the bias-variance trade off of RMC with that of SARSA-λ. Then, we compare the performance of RMC and SARSA-λ for medium sized randomly generated MDPs. Finally, we present a continuous state model motivated by a practical remote estimation problem and show that RMC identifies optimal policies without the use of function approximation. The numerical experiments demonstrate that RMC has lower bias compared to temporal difference methods while achieving almost the same variance. Hence, RMC is an attractive alternative to temporal difference methods for infinite horizon models with designated start state.

1.2 Existing literature

Renewal theory is commonly used to estimate performance of stochastic systems in the simulation optimization community. See, for example (Glynn 1986; 1990). However, these methods assume that the probability law of the primitive random variables and its weak derivate are known. These methods cannot be used in reinforcement learning because one does not have access to the system model.

Renewal theory is also commonly used for Markov decision processes (MDPs) in the engineering literature on queuing theory and systems and control. However, most of these typically assume that the system model is known.

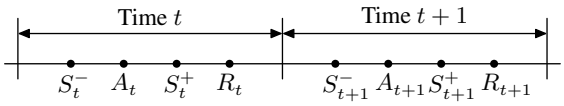


Figure 1: A timing diagram showing the system variables.

Perhaps the closest result to those presented in this paper is the simulation based algorithm for average reward Markov decision processes (MDPs) proposed in (Marbach and Tsitsiklis 2001; 2003) that uses the regenerative property of Markov processes. However, their algorithm uses a different renewal relation than ours and has an initial bias that asymptotically goes to zero. A renewal theory based stochastic approximation algorithm for identifying the optimal policy for a specific model was proposed in (Chakravorty, Subramanian, and Mahajan 2017). However, their algorithm was tuned to the specific model and is not applicable to general MDPs.

2 Model

For ease of exposition, we describe the model when the state and action spaces are finite. Similar description holds (under standard measurability conditions) when the state and action spaces are continuous. We consider a generalized Markov decision process (MDP) $(\mathcal{S}^-, \mathcal{S}^+, \mathcal{A}, P^-, P^+, r)$ where

- $\mathcal{S}^-$ is the pre-decision state space;
- $\mathcal{S}^+$ is the post-decision state space;
- $\mathcal{A}$ is the action space;
- P^- is a Markov transition kernel from $\mathcal{S}^+$ to $\mathcal{S}^-$;
- P^+ is a controlled Markov transition kernel from $\mathcal{S}^- \times \mathcal{A}$ to $\mathcal{S}^+$;
- $r\colon \mathcal{S}^- \times \mathcal{A} \times \mathcal{S}^+ \to \mathbb{R}$ is the per-step reward function.

At time t, $S_t^- \in \mathcal{S}^-$ denotes the pre-decision state, $A_t \in \mathcal{A}$ denotes the action, and $S_t^+ \in \mathcal{S}^+$ denotes the post-decision state. The sequential order in which the system variables are generated is shown in Fig. 1.

The system starts at an initial state $s_0^+ \in \mathcal{S}^+$. For every time t, the following events take place:

1. there is a controlled transition from S_t^- to S_t^+, where
$$\mathbb{P}(S_t^+ = s_t^+ | S_t^- = s_t^-, A_t = a_t) = P^+(s_t^+ | s_t^-, a_t)$$
2. there is an uncontrolled transition from S_t^+ to S_{t+1}^-, where
$$\mathbb{P}(S_{t+1}^- = s_{t+1}^- | S_t^+ = s_t^+) = P^-(s_{t+1}^- | s_t^+)$$
3. a per-step reward $R_t = r(S_t^-, A_t, S_t^+)$ is received.

Note that when $\mathcal{S}^+ = \mathcal{S}^-$ and P^- is identity, then the above model reduces to the standard MDP model. When P^+ is a deterministic transition, the model reduces to a standard MDP model with post decision states.

The actions $\{A_t\}_{t\geq 1}$ are chosen according to a time-homogeneous stochastic Markov policy $\pi\colon \mathcal{S}^- \to \Delta(\mathcal{A})$ (where $\Delta(\mathcal{A})$ denotes the space of probability distributions over $\mathcal{A}$), i.e., $\mathbb{P}[A_t = a_t | S_t^- = s_t^-] = \pi(a_t \,|\, s_t^-)$, which we also denote using $A_t \sim \pi(S_t^-)$.

The performance when the system starts in post-decision state $s^+ \in \mathcal{S}^+$ and follows policy π is given by

$$V_\pi^+(s^+) = \mathbb{E}_{A_t \sim \pi(S_t^-)} \left[\sum_{t=1}^{\infty} \gamma^{t-1} R_t \;\middle|\; S_0^+ = s^+ \right]. \quad (1)$$

From the standard dynamic programming argument, we know that the value function is given by the solution of the following fixed point equations:

$$V_\pi^+(s^+) = \sum_{s^- \in \mathcal{S}^-} P^-(s^-|s^+) V_\pi^-(s^-), \quad (2)$$

$$V_\pi^-(s^-) = \sum_{a \in \mathcal{A}} \pi(a|s^-) Q_\pi(s^-, a), \quad (3)$$

where

$$Q_\pi(s^-, a) = \sum_{s^+ \in \mathcal{S}^+} P^-(s^+|s^-, a) \big[r(s^-, a, s^+) + \gamma V_\pi^+(s^+) \big].$$

We call V^- and V^+ as the pre- and post-decision state value functions, respectively. We assume that the initial state $s_0^+ \in \mathcal{S}^+$ of the system is known. The objective is to choose a policy π to maximize $V_\pi^+(s_0^+)$, when the system model (P^-, P^+, r) is not known.

We present a new online reinforcement learning algorithm that we call Renewal Monte Carlo (RMC). In Sec. 3, we present RMC variant that uses the sample path $(S_0^+, S_1^-, A_1, S_1^+, R_1, S_2^-, \dots)$ to evaluate the performance of a particular policy. In Sec. 4 we present online reinforcement learning variant of RMC that identifies a locally optimal policy when the policies are parameterized by a closed and convex subset Θ of the Euclidean space.[1]

3 RMC Algorithm for Policy Evaluation

In this section, we present the RMC algorithm for evaluating the performance of an arbitrary policy π when the system starts at a designated start state s_0^+. We assume that the Markov chain induced on the post-decision states by the policy π has a single ergodic class and that class is positive recurrent. We pick one of the states from this ergodic class as a *reference state* and denote it by s_*^+. Since s_*^+ is positive recurrent, it is visited infinitely often along any sample path. For simplicity of exposition, *we assume that the reference state is the same as the start state.*

Let $\{\tau_n\}_{n \geq 1}$ denote a sequence of stopping times, where τ_n is the stopping time when the system returns to the post-decision state s_*^+ for the n-th time. Thus,

$$\tau_1 = \inf\{t > 1 : s_t^+ = s_*^+\}, \quad \tau_2 = \inf\{t > \tau_1 : s_t^+ = s_*^+\},$$

and so on. For ease of notation, we define $\tau_0 = 0$. Consider the sequence of (S_t^-, A_t, S_t^+, R_t) from $\tau_{n-1} + 1$ to τ_n. We call this the *n-th regenerative cycle*. Let R_n and T_n denote the total discounted reward and total discounted time of the n-th regenerative cycle, i.e.,

$$\mathsf{R}_n = \gamma^{-\tau_{n-1}} \sum_{t=\tau_{n-1}+1}^{\tau_n} \gamma^{t-1} R_t \quad \text{and} \quad \mathsf{T}_n = \gamma^{-\tau_{n-1}} \sum_{t=\tau_{n-1}+1}^{\tau_n} \gamma^{t-1}.$$

[1]For example, Θ could be the temperature vector in a Gibbs soft-max policy, or the weights of a deep neural network, or the thresholds in a control limit policy, and so on.

Algorithm 1: RMC Algorithm for policy evaluation

input : π : Policy
γ : Discount factor
s_*^+ : Reference state (same as initial state)
N : Number of regenerative cycles
output : $\hat{J}_N$: Estimate for performance
initialize: $\overline{\mathsf{R}} = 0$; $\overline{\mathsf{T}} = 0$; $t = 1$
for *regenerative cycle* $n = 1$ *up to* N **do**
  Set $\widehat{\mathsf{R}}_n = 0$; $\widehat{\mathsf{T}}_n = 0$; $\texttt{scale} = 1$
  do
    Observe (S_t^-, A_t, S_t^+, R_t) where $A_t \sim \pi(S_t^-)$
    $\widehat{\mathsf{R}}_n \leftarrow \widehat{R}_n + \texttt{scale} * R_t$
    $\widehat{\mathsf{T}}_n \leftarrow \widehat{T}_n + \texttt{scale}$
    $\texttt{scale} \leftarrow \texttt{scale} * \gamma$
    $t \leftarrow t + 1$
  while $S_t^+ \neq s_*^+$
  $\overline{\mathsf{R}} \leftarrow \overline{\mathsf{R}} + \frac{1}{n}(\widehat{\mathsf{R}}_n - \overline{\mathsf{R}})$
  $\overline{\mathsf{T}} \leftarrow \overline{\mathsf{T}} + \frac{1}{n}(\widehat{\mathsf{T}}_n - \overline{\mathsf{T}})$
return $\overline{\mathsf{R}}/(1-\gamma)\overline{\mathsf{T}}$

By the strong Markov property, $\{\mathsf{R}_n\}_{n \geq 1}$ and $\{\mathsf{T}_n\}_{n \geq 1}$ are i.i.d. sequences. Let R_π and T_π denote $\mathbb{E}[\mathsf{R}_n]$ and $\mathbb{E}[\mathsf{T}_n]$, respectively. Define

$$\widehat{\mathsf{R}}_N = \frac{1}{N} \sum_{n=1}^{N} \mathsf{R}_n \quad \text{and} \quad \widehat{\mathsf{T}}_N = \frac{1}{N} \sum_{n=1}^{N} \mathsf{T}_n.$$

From the strong law of large numbers, we have that

$$\lim_{N \to \infty} \widehat{\mathsf{R}}_N = \mathsf{R}_\pi, \text{ a.s.}^2 \quad \text{and} \quad \lim_{N \to \infty} \widehat{\mathsf{T}}_N = \mathsf{T}_\pi, \text{ a.s.}$$

Thus, $\widehat{\mathsf{R}}_N$ and $\widehat{\mathsf{T}}_N$ are unbiased estimators of R_π and T_π.

The following result relates the expected discounted reward and time of a regenerative cycle with the value function of the reference state.

Proposition 1 (Renewal relationship) *The post-decision value function at the reference state s_*^+ is given by*

$$V_\pi^+(s_*^+) = \frac{\mathsf{R}_\pi}{(1-\gamma)\mathsf{T}_\pi}.$$

PROOF For the ease of notation, define

$$\overline{\mathsf{T}}_\pi = \mathbb{E}_{A_t \sim \pi(S_t^-)}[\gamma^{(\tau_n - \tau_{n-1})}]$$

Using the formula for geometric series, we get that $\mathsf{T}_\pi = (1 - \overline{\mathsf{T}}_\pi)/(1-\gamma)$. Hence,

$$\overline{\mathsf{T}}_\pi = 1 - (1-\gamma)\mathsf{T}_\pi. \quad (4)$$

Now, consider the post-decision state value function:

$$V_\pi^+(s_*^+) = \mathbb{E}_{A_t \sim \pi(S_t^-)} \Bigg[\sum_{t=1}^{\tau_1} \gamma^{t-1} R_t + \gamma^{\tau_1} \sum_{t=\tau_1+1}^{\infty} \gamma^{t-\tau_1-1} R_t \;\Bigg|\; S_0^+ + s_*^+ \Bigg]$$

[2]The abbreviation a.s. means "almost surely".

$$\stackrel{(a)}{=} \mathsf{R}_\pi + \mathbb{E}_{A_t\sim\pi(S_t^-)}[\gamma^{\tau_1}]\, V_\pi^+(s_*^+)$$
$$= \mathsf{R}_\pi + \overline{T}_\pi V_\pi^+(s_*^+), \tag{5}$$

where the second expression in (a) uses the fact that the random variables from 1 to τ_1 are conditionally independent of those from $\tau_1 + 1$ onwards due to the strong Markov property. Substituting (4) in (5) and rearranging terms, we get the result of the proposition. ■

The RMC algorithm for policy evaluation uses

$$\hat{J}_N := \widehat{\mathsf{R}}_N/(1-\gamma)\widehat{\mathsf{T}}_N$$

as an estimator for the performance $V_\pi^+(s_*^+)$. The details are shown in Algorithm 1.

Recall that $\widehat{\mathsf{R}}_N$ and $\widehat{\mathsf{T}}_N$ are unbiased estimators of R_π and T_π, respectively. However, $\hat{J}_N$ is not an unbiased estimator of $V_\pi^+(s_*^+)$. In theory, it is possible to obtain unbiased estimator of $V_\pi^+(s_*^+)$ using $\widehat{\mathsf{R}}_N$ and $\widehat{\mathsf{T}}_N$ (see, e.g., (Blanchet and Glynn 2015)). But, such unbiased estimators tend to be complicated by using them we loose the simplicity of RMC, which is one of its attractive features. In practice, the bias of $\hat{J}_N$ is small because of the following concentration inequality.

Proposition 2 *Suppose there exist rate functions C_R and C_T such that for any $\varepsilon > 0$, there exists an N such that for all $M \geq N$, we have*

$$\mathbb{P}\big(|\mathsf{R}_\pi - \widehat{R}_M| > \varepsilon\big) \leq \exp\big(-MC_R(\varepsilon)\big),$$

and

$$\mathbb{P}\big(|\mathsf{T}_\pi - \widehat{T}_M| > \varepsilon\big) \leq \exp\big(-MC_T(\varepsilon)\big).$$

Then, there exists a rate function C such that for any $\varepsilon > 0$, there exists an N such that for all $M \geq N$, we have

$$\mathbb{P}\big(|V_\pi^+(s_*^+) - \hat{J}_M| > \varepsilon\big) \leq \exp\big(-MC(\varepsilon)\big).$$

See Appendix for proof.

Remark 1 The rate functions C_R and C_T exist under mild technical conditions. For example, if R_n and T_n are uniformly bounded, then Hoeffding inequality (Hoeffding 1963) implies the existence of such rate functions. □

4 RMC Algorithm for Policy Improvement

In this section, we present the RMC algorithm to identify a locally optimal policy when policies are parameterized by a closed and convex subset Θ of the Euclidean space. Given a $\theta \in \Theta$, we use π_θ to denote the policy parametrized by θ and J_θ to denote $V_{\pi_\theta}^+(s_0^+)$.

We assume that there is a reference state s_*^+ that is positive recurrent under all policy parameters $\theta \in \Theta$. Since s_*^+ is positive recurrent, it is visited infinitely often along any sample path. As before, for simplicity of exposition, *we assume that the reference state is the same as the start state*.

The typical approach for policy improvement is to start with an initial guess $\theta_0 \in \Theta$ and iteratively update it using stochastic gradient ascent. In particular, let $\hat{J}'_{\theta_m}$ be an unbiased estimator of $\nabla_\theta J_\theta\big|_{\theta=\theta_m}$, then update

$$\theta_{m+1} = \big[\theta_m + \alpha_m \hat{J}'_{\theta_m}\big]_\Theta \tag{6}$$

where $[\theta]_\Theta$ denotes the projection of θ onto Θ and $\{\alpha_m\}_{m\geq 1}$ is the sequence of learning rates that satisfies the standard assumptions of $\sum_{m=1}^\infty \alpha_m = \infty$ and $\sum_{m=1}^\infty \alpha_m^2 < \infty$. Under mild technical conditions, the above iteration converges to a θ^* that is locally optimal, i.e., $\nabla_\theta J_\theta\big|_{\theta=\theta^*} = 0$.

Using Proposition 1, we get that

$$\nabla_\theta J_\theta = H_\theta/T_\theta^2, \quad \text{where } H_\theta = T_\theta \nabla_\theta R_\theta - R_\theta \nabla_\theta T_\theta.$$

In RMC, instead of using stochastic gradient descent to find the minimum of $\nabla_\theta J_\theta$, we use stochastic approximation to find the root of H_θ. In particular, let $\widehat{H}_m$ be an unbiased estimator of H_{θ_m}. We then use the update

$$\theta_{m+1} = \big[\theta_m + \alpha_m \widehat{H}_m\big]_\Theta \tag{7}$$

where $\{\alpha_m\}_{m\geq 1}$ satisfies the standard condition on learning rates. Under mild technical conditions, the above iteration converges to a locally optimal policy. Specifically, we have the following.

Theorem 1 *Suppose $\widehat{H}_m$ is an unbiased estimator of H_θ and the map $\theta \mapsto H_\theta$ is Lipschitz then the sequence $\{\theta_m\}_{m\geq 1}$ generated by* (7) *converges and*

$$\lim_{m\to\infty} \nabla_\theta J_\theta\big|_{\theta_m} = 0.$$

PROOF Under the above assumptions, conditions (A1)–(A4) of (Borkar 2008, pg 10-11). Therefore, the result follows from (Borkar 2008, Theorem 2.2). ■

Remark 2 A sufficient condition for the map $\theta \to H_\theta$ to be Lipschitz that the rewards are uniformly bounded, which, is always the case when the state and action spaces are finite. □

In the remainder of this section, we present two methods for estimating the gradients of R_θ and T_θ. The first is a likelihood ratio (or score function) based gradient estimator which works when the policy is differentiable with respect to the policy parameters. The second is a simultaneous perturbation based gradient estimator that uses finite differences, which is useful when the policy is not differentiable with respect to the policy parameters.

4.1 Likelihood ratio (or score function) gradient based estimator

In the likelihood ratio (or score function) based estimator, we assume that $\pi_\theta(a \mid s^-)$ is differentiable with respect to θ. For any time t, define the score function

$$L_t = \nabla_\theta \log[\pi_\theta(A_t \mid S_t^-)]$$

and for the n-th regenerative cycle, define

$$\mathsf{L}_n = \sum_{t=\tau_{n-1}+1}^{\tau_n} L_t$$

Then, define the following estimators for $\nabla_\theta \mathsf{R}_\theta$ and $\nabla_\theta \mathsf{T}_\theta$:

$$\widehat{\mathsf{R}}'_N = \frac{1}{N}\sum_{n=1}^N \mathsf{R}_n \mathsf{L}_n \quad \text{and} \quad \widehat{\mathsf{T}}'_N = \frac{1}{N}\sum_{n=1}^N \mathsf{T}_n \mathsf{L}_n.$$

Algorithm 2: RMC Algorithm for RL: Likelihood ratio (or score function) based variant

input : θ_0 : Initial policy
γ : Discount factor
s_*^+ : Reference state (same as initial state)
N : Number of regenerative cycles
M : Number of iterations
output : θ_M: Estimate for best policy
initialize : $t = 1$
for *iteration* $m = 0$ *up to* $M-1$ **do**
  Set $\overline{R} = 0$; $\overline{R}' = 0$; $\overline{\mathsf{T}} = 0$; $\overline{\mathsf{T}}' = 0$
  for *regenerative cycle* $n = 1$ *up to* N **do**
    Set $\widehat{\mathsf{R}}_n = 0$; $\widehat{\mathsf{T}}_n = 0$; $\mathsf{L}_n = 0$; $\texttt{scale} = 1$
    do
      Observe (S_t^-, A_t, S_t^+, R_t) where $A_t \sim \theta_m(S_t^-)$
      $\widehat{\mathsf{R}}_n \leftarrow \widehat{R}_n + \texttt{scale} * R_t$
      $\widehat{\mathsf{T}}_n \leftarrow \widehat{T}_n + \texttt{scale}$
      $\mathsf{L}_n \leftarrow \mathsf{L}_n + \nabla_\theta \log\left[\pi_\theta(A_t \mid S_t^-)\right]$
      $\texttt{scale} \leftarrow \texttt{scale} * \gamma$
      $t \leftarrow t+1$
    while $S_t^+ \neq s_*^+$
    $\overline{\mathsf{R}} \leftarrow \overline{\mathsf{R}} + \frac{1}{n}(\widehat{\mathsf{R}}_n - \overline{\mathsf{R}})$; $\overline{\mathsf{R}}' \leftarrow \overline{\mathsf{R}}' + \frac{1}{n}(\widehat{\mathsf{R}}_n \mathsf{L}_n - \overline{\mathsf{R}}')$
    $\overline{\mathsf{T}} \leftarrow \overline{\mathsf{T}} + \frac{1}{n}(\widehat{\mathsf{T}}_n - \overline{\mathsf{T}})$; $\overline{\mathsf{T}}' \leftarrow \overline{\mathsf{T}}' + \frac{1}{n}(\widehat{\mathsf{T}}_n \mathsf{L}_n - \overline{\mathsf{T}}')$
  $\widehat{H}'_m = \overline{\mathsf{T}}\,\overline{\mathsf{R}}' - \overline{\mathsf{R}}\,\overline{\mathsf{T}}'$
  $\theta_{m+1} = \left[\theta_m + \alpha_m \widehat{H}'_m\right]_\Theta$
return θ_M

Proposition 3 *$\widehat{\mathsf{R}}'_N$ and $\widehat{\mathsf{T}}'_N$ defined above are unbiased estimators of $\nabla_\theta \mathsf{R}_\theta$ and $\nabla_\theta \mathsf{T}_\theta$.*

See Appendix for proof.

Proposition 3 suggests that we can use

$$\widehat{H}_N = \widehat{\mathsf{T}}_N \widehat{\mathsf{R}}'_N - \widehat{\mathsf{R}}_N \widehat{\mathsf{T}}'_N$$

as the estimator for H_θ. The complete algorithm for updating the policy parameters is shown in Algorithm 2.

4.2 Simultaneous perturbation based gradient estimator

When the policy π_θ is not differentiable with respect to θ, we can estimate the gradient of R_θ and T_θ using simultaneous perturbation finite differences. Two such commonly used methods are SPSA (Simultaneous Perturbation Stochastic Approximation) and SF (Smooth Functional) algorithms (Bhatnagar, Prasad, and Prashanth 2013).

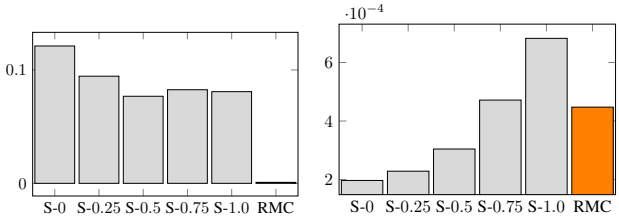
In SPSA, estimate of the gradient is

$$\widehat{\mathsf{R}}'_\theta = (\mathsf{R}_{\theta+\beta\delta} - \mathsf{R}_\theta)/\beta\delta, \tag{8}$$

where δ is a Bernoulli/Binomial random variable with the same dimensions as the parameter, θ, with each element of δ taking values in the set $\{-1, 1\}$ and β is a small positive value. The expression for $\widehat{\mathsf{T}}'_N$ is similar.

In SF, estimate of the gradient is

$$\widehat{\mathsf{R}}'_\theta = \eta(\mathsf{R}_{\theta+\beta\eta} - \mathsf{R}_\theta)/\beta, \tag{9}$$



(a) Bias (b) Variance

Figure 2: Bias variance trade-off in policy evaluation for different algorithms for Howard Taxi example. The label "S-λ" denotes SARSA-λ.

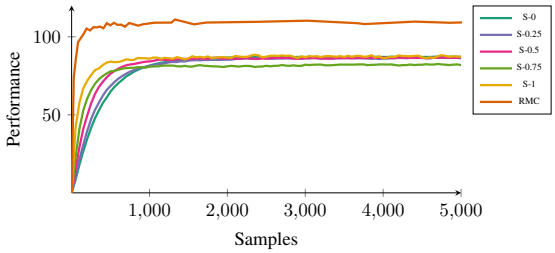


Figure 3: Comparison of policy improvement using SARSA-λ and RMC in Howard Taxi example.

where η is a Gaussian (zero mean, unit variance) random variable with the same dimensions as the parameter, θ and β is a small positive value. The expression for $\widehat{\mathsf{T}}'_N$ is similar.

We can use these *one sided* estimates to estimate H_θ as

$$\widehat{H}_\theta = \mathsf{T}_\theta \widehat{\mathsf{R}}'_\theta - \mathsf{R}_\theta \widehat{\mathsf{T}}'_\theta.$$

Substituting (8) or (9) in the above expression and simplifying, we get that for SPSA the estimate of H_θ becomes

$$\widehat{H}_\theta = (\mathsf{T}_\theta \mathsf{R}_{\theta+\beta\delta} - \mathsf{R}_\theta \mathsf{T}_{\theta+\beta\delta})/\beta\delta$$

and for SF, the estimate of H_θ becomes

$$\widehat{H}_\theta = \eta(\mathsf{T}_\theta \mathsf{R}_{\theta+\beta\eta} - \mathsf{R}_\theta \mathsf{T}_{\theta+\beta\eta})/\beta.$$

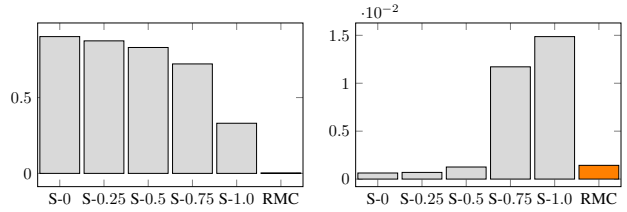
In either case, we can generate an estimate of H_θ by evaluating two policies π_θ and $\pi_{\theta+\beta\delta}$ or $\pi_{\theta+\beta\eta}$ using (a slight variation of) Algorithm 1.

5 Numerical Experiments

5.1 Howard Taxi

In this section, we compare the performance of RMC with that of SARSA-λ for Howard Taxi example (Howard 1960). This is a toy model for a taxi that plies between three cities. At each time, the taxi-driver has the option to: (i) cruise in the city and hope that a passenger hails the taxi; (ii) go to the taxi stand and wait his turn; or (iii) pull over and wait for a call from a radio dispatcher. We choose the discount factor $\gamma = 0.9$ and the rest of the parameters as as specified in (Howard 1960, pg 45).

We run two experiments with this model. In the first experiment, we pick a specific policy: select 'going to the taxi



(a) Bias (b) Variance

Figure 4: Bias variance trade-off in policy evaluation for different algorithms for Randomly Generated MDP example. The label "S-λ" denotes SARSA-λ.

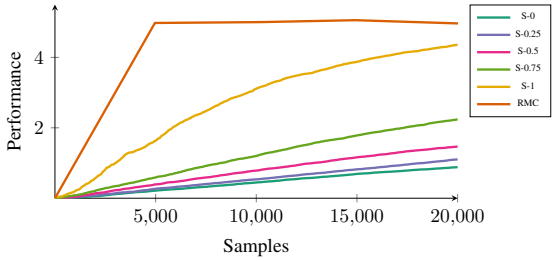


Figure 5: Comparison of policy improvement using SARSA-λ and RMC in Randomly Generated MDP example.

stand' in all states. This is the optimal policy for the chosen parameters. We evaluate the performance of this policy using SARSA-λ for $\lambda \in \{0, 0.25, 0.5, 0.75, 1\}$ and RMC. For SARSA-λ, we evaluate the performance of a trajectory of length 1000 and repeat the experiment 100 times. For RMC, we evaluate the performance of 67 renewals[3] and repeat the experiment 100 times. We compare the performance of these algorithms with that of exact value iteration (for policy evaluation). The resulting mean and bias (normalized to 1 using the solution obtained by value iteration) is shown in Fig. 2. As can be seen from the figure, RMC has negligible bias and has variance that is comparable to SARSA-0.75.

In the second experiment, we do reinforcement learning. We restrict attention to policies which randomize using the Gibbs soft-max function. Such policies are parameterized by 9 parameters (number of states times number of actions). We assume that each parameter belongs to the interval $[-30, 30]$. For both SARSA-λ and RMC, we use likelihood ratio (or score function) method to estimate the gradients. We pick the learning rates for policy improvement using ADAM (Kingma and Ba 2014) with the default values of the hyper-parameters. We run policy improvement for 5000 samples and repeat the experiment 100 times. The performance of all six algorithms is shown in Fig. 3. As can be seen from the figure, RMC converges significantly faster than SARSA-λ and converges to a better estimate.

[3]This number was chosen so that the average trajectory length for RMC (which was 989.12 in this case) is close to the trajectory length for SARSA-λ.

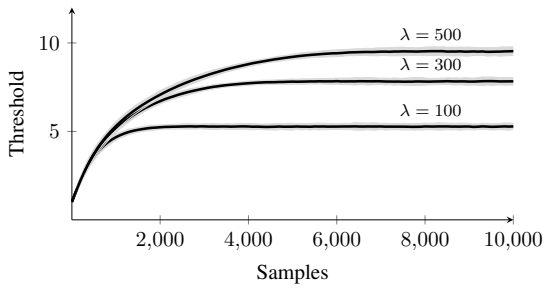


Figure 6: Policy parameters versus number of samples when using RMC for the remote estimation problem for different values of communication cost λ. The experiment was repeated 100 times. The solid lines show the mean and the shaded area shows the 2σ deviation (across the 100 runs).

5.2 Randomly Generated MDP

In this section, we repeat the experiment given in Section 5.1, but for a larger MDP with 1000 states and 20 actions.The transition matrix for each of the 20 actions is generated randomly by first selecting each element using Unif$[0, 1]$ and then normalizing the row such that the row sum is 1. Similar, for each action, the reward matrix is generated randomly by selecting each element using Unif$[0, 1]$. As in Sec. 5.1, we run two experiments: first on policy evaluation on the second on policy improvement.

For the first experiment, we first identify the optimal policy using value iteration and then evaluate its performance using SARSA-λ and RMC. For SARSA-λ, we evaluate the performance of a trajectory of length 10,000 and repeat the experiment 100 times. For RMC, we evaluate the performance of 10 renewals[4] and repeat the experiment 100 times. We compare the performance of these algorithms with that of exact value iteration (for policy evaluation). The resulting mean and bias (normalized to 1 using the solution obtained by value iteration) is shown in Fig. 4. As can be seen from the figure, RMC has negligible bias and has variance that is comparable to SARSA-0.5, whereas the SARSA methods are yet to converge (which can be inferred from their large normalized bias).

The second experiment is similar to the second experiment for Howard Taxi. The only difference in this case is that we run policy improvement for 20,000 samples. Fig. 5 shows that, as in the previous example, RMC converges significantly faster than SARSA-λ and converges to a better estimate.

5.3 Remote Estimation

In this experiment, we study remote estimation problem that arises in networked control systems (Lipsa and Martins 2011). A transmitter observes a first-order autoregressive process $\{X_t\}_{t\geq 1}$, i.e., $X_{t+1} = \alpha X_t + W_t$, where $\alpha, X_t, W_t \in \mathbb{R}$, and $\{W_t\}_{t\geq 1}$ is an i.i.d. process. At each time, the transmitter uses an event-triggered policy (explained below) to

[4]As before, this number was chosen so that the average trajectory length for RMC (which was 9694.81 in this case) is close to the trajectory length for SARSA-λ.

determine whether to transmit or not (denoted by $A_t = 1$ and $A_t = 0$, respectively). Transmission takes place over an i.i.d. erasure channel with erasure probability p_d. Let S_t^- and S_t^+ denote the "error" between the source realization and it's reconstruction at a receiver. It can be shown that S_t^- and S_t^+ evolve as follows (Lipsa and Martins 2011): when $A_t = 0$, $S_t^+ = S_t^-$; when $A_t = 1$, $S_t^+ = 0$ if the transmission is successful (w.p. $(1 - p_d)$) and $S_t^+ = S_t^-$ if the transmission is not successful (w.p. p_d); and $S_{t+1}^- = \alpha S_t^+ + W_t$. Note that the post-decision state resets to zero after every successful transmission.[5]

The objective is to minimize the expected discounted cost, where the per-step cost is $\lambda A_t + (S_t^+)^2$, where λ corresponds to the cost of communicating a packet and $(S_t^+)^2$ is the estimation error.

An event-triggered policy is a threshold policy that chooses $A_t = 1$ whenever $|S_t^-| \geq \theta$, where θ is a design choice. Under certain conditions, such an event-triggered policy is known to be optimal (Lipsa and Martins 2011). When the system model is known, algorithms to various algorithms to compute the optimal θ are presented in (Xu and Hespanha 2004; Chakravorty and Mahajan 2017; Chakravorty, Subramanian, and Mahajan 2017). In this section, we use RMC to identify the optimal policy when the model parameters are not known. An event-triggered policy is a parametric policy but $\pi_\theta(a|s^-)$ is not differentiable in θ. Therefore, the likelihood ratio (or the score function) methods cannot be used to estimate performance gradient.

In our experiments, we use RMC and estimate the gradient using simultaneous perturbation variant called one-sided smooth functional approximation (Bhatnagar, Prasad, and Prashanth 2013) with 1000 renewals in each step to estimate the gradient. We simulate the system using $\alpha = 1$, $p_d = 0.1$, and $W_t \sim \mathcal{N}(0, 1)$ (the parameters are not known to the RMC algorithm). The optimal parameters for different choices of communication cost λ are shown in Fig. 6.

6 Conclusions

We presented a new RL algorithm, called Renewal Monte Carlo (RMC), for infinite-horizon discounted reward problems with a designated state state. Our experimental study suggests that RMC has significantly low bias with variance that is similar to SARSA-λ. RMC does not suffer from the high variance of MC methods because the estimates are obtained by averaging large number of independent performance evaluations. RMC is an Actor only method and works for models with continuous state and action spaces without the need to approximate the value function.

Although we restricted attention to discounted reward model, all the results immediately extend to the average reward model as well. To simplify the discussion, we assumed that the reference state is the same as the start state. Even if that is not the case, the arguments presented in this paper go through with slight modification. For example, if $\mathsf{R}_\pi(s^+)$ and $\mathsf{T}_\pi(s^+)$ denote the expected reward and time to hit the reference state s_*^+ when starting from state s_0^+. Then, similar to Proposition 1, we can show that $V_\pi^+(s_0^+) = \mathsf{R}_\pi(s_0^+) + \overline{T}_\pi(s_0^+) V_\pi^+(s_*^+)$ where $\mathsf{R}_\pi(s_0^+)$ and $T_\pi(s_0^+)$ can be estimated as in Sec. 3.

Finally, we only presented the simplest form of the RMC algorithm. It is possible to obtain an "every step" variant of RMC that can be used to to estimate the entire value function (or its approximation).

[5] Had we used the standard MDP model instead of the model of Sec. 2, this restart would not have always resulted in a renewal.

Appendices

A. Proof of Proposition 2

For any $\varepsilon > 0$, identify $\varepsilon_R, \varepsilon_T \in \mathbb{R}$ such that

$$\frac{\mathsf{R}_\pi}{T_\pi} - \frac{\mathsf{R}_\pi - \varepsilon_R}{T_\pi + \varepsilon_T} = \frac{\mathsf{R}_\pi + \varepsilon_R}{T_\pi - \varepsilon_T} - \frac{\mathsf{R}_\pi}{T_\pi} \tag{10}$$

and

$$C_R(\varepsilon_R) = C_T(\varepsilon_T). \tag{11}$$

Define $C(\varepsilon) = C_R(\varepsilon_R) = C_T(\varepsilon_T)$.

Define $B_\varepsilon(x)$ as $\{y \in \mathbb{R} : |x - y| \leq \varepsilon\}$. (10) implies that if $\widehat{\mathsf{R}}_M \in B_{\varepsilon_R}(\mathsf{R}_\pi)$ and $\widehat{\mathsf{T}}_M \in B_{\varepsilon_T}(\mathsf{T}_\pi)$, then $\widehat{\mathsf{R}}_M / \widehat{\mathsf{T}}_M \in B_\varepsilon\big(\mathsf{R}_\pi / \mathsf{T}_\pi\big)$. Thus, for any $M \geq N$, we have that

$$\begin{aligned}
&\mathbb{P}\left(\frac{\widehat{\mathsf{R}}_M}{\widehat{\mathsf{T}}_M} \notin B_\varepsilon\left(\frac{\mathsf{R}_\pi}{\mathsf{T}_\pi}\right)\right) \\
&\quad \overset{(a)}{\leq} \mathbb{P}\big(\{\widehat{\mathsf{R}}_M \notin B_{\varepsilon_R}(\mathsf{R}_\pi)\}\big) + \mathbb{P}\big(\{\widehat{\mathsf{T}}_M \notin B_{\varepsilon_T} \mathsf{T}_\pi\}\big) \\
&\quad \overset{(b)}{\leq} \exp(-MC_R(\varepsilon_R)) + \exp(-MC_T(\varepsilon_T)) \\
&\quad \overset{(c)}{=} \exp(-MC(\varepsilon))
\end{aligned}$$

where (a) follows from the union bound on probability, (b) follows from the assumptions in the proposition, and (c) follows from (11).

B. Proof of Proposition 3

Let P_θ denote the probability induced on the sample paths when the system is following policy θ. Let $D_n = (S_t^-, A_t, S_t^+)_{t=\tau_{n-1}+1}^{\tau_n}$ denote the sample path for the n-th regenerative cycle. Then,

$$P_\theta(D_n) = \prod_{t=\tau_{n-1}+1}^{\tau_n} P^-(S_t^- | S_{t-1}^+) \pi_\theta(A_t \mid S_t^-) P^+(S_t^+ | S_t^-, A_t)$$

Therefore,

$$\nabla_\theta \log P_\theta(D_n) = \sum_{t=\tau_{n-1}+1}^{\tau_n} \nabla_\theta \log \pi_\theta(A_t | S_t^-) = \mathsf{L}_n. \tag{12}$$

Now, recall that $\mathsf{R}_\theta = \mathbb{E}_{A_t \sim \pi_\theta(S_t^-)}[\mathsf{R}_n]$. Therefore, from the log-derivative trick, we get

$$\begin{aligned}
\nabla_\theta \mathsf{R}_\theta &= \mathbb{E}_{A_t \sim \pi_\theta(S_t^-)}[\mathsf{R}_n \nabla_\theta \log P_\theta(D_n)] \\
&\overset{(a)}{=} \mathbb{E}_{A_t \sim \pi_\theta(S_t^-)}[\mathsf{R}_n \mathsf{L}_n]
\end{aligned} \tag{13}$$

where (a) follows from (12). Note that $\widehat{\mathsf{R}}'_N$ is an unbiased estimator of the right hand side of (13). The result for $\widehat{\mathsf{T}}'_N$ follows from a similar argument.

References

Bhatnagar, S.; Prasad, H.; and Prashanth, L. 2013. *Stochastic recursive algorithms for optimization: simultaneous perturbation methods*, volume 434. Springer.

Blanchet, J. H., and Glynn, P. W. 2015. Unbiased monte carlo for optimization and functions of expectations via multi-level randomization. In *Proceedings of the 2015 Winter Simulation Conference*, 3656–3667. IEEE Press.

Borkar, V. 2008. *Stochastic Approximation: A Dynamical Systems Viewpoint*. Cambridge University Press.

Chakravorty, J., and Mahajan, A. 2017. Fundamental limits of remote estimation of Markov processes under communication constraints. *IEEE Transactions on Automatic Control* 62(3):1109–1124.

Chakravorty, J.; Subramanian, J.; and Mahajan, A. 2017. Stochastic approximation based methods for computing the optimal thresholds in remote-state estimation with packet drops. In *Proc. American Control Conference*, 462–467.

Feller, W. 1966. *An Introduction to Probability Theory and its Applications*, volume 1. John Wiley and Sons.

Glynn, P. 1986. Optimization of stochastic systems. In *Proc. Winter Simulation Conference*, 52–59.

Glynn, P. 1990. Likelihood ratio gradient estimation for stochastic systems. *Communications of the ACM* 33:75–84.

Hoeffding, W. 1963. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association* 58(301):13–30.

Howard, R. 1960. *Dynamic Programming and Markov Processes*. Published jointly by the Technology Press of the Massachusetts Institute of Technology and John Wiley & Sons, Inc.

Kingma, D., and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Lipsa, G. M., and Martins, N. 2011. Remote state estimation with communication costs for first-order LTI systems. *IEEE Transactions on Automatic Control* 56(9):2013–2025.

Marbach, P., and Tsitsiklis, J. N. 2001. Simulation-based optimization of markov reward processes. *IEEE Transactions on Automatic Control* 46(2):191–209.

Marbach, P., and Tsitsiklis, J. N. 2003. Approximate gradient methods in policy-space optimization of markov reward processes,. *Discrete Event Dynamical Systems* 13(2):111–148.

Xu, Y., and Hespanha, J. P. 2004. Optimal communication logics in networked control systems. In *43rd IEEE Conference on Decision and Control*, 3527–3532.