
Bias in Natural Actor-Critic Algorithms

Philip S. Thomas

PTHOMAS@CS.UMASS.EDU

School of Computer Science, University of Massachusetts, Amherst, MA 01002 USA

Abstract

We show that several popular discounted reward natural actor-critics, including the popular NAC-LSTD and eNAC algorithms, do not generate unbiased estimates of the natural policy gradient as claimed. We derive the first unbiased discounted reward natural actor-critics using batch and iterative approaches to gradient estimation. We argue that the bias makes the existing algorithms more appropriate for the average reward setting. We also show that, when Sarsa(λ) is guaranteed to converge to an optimal policy, the objective function used by natural actor-critics has only global optima, so policy gradient methods are guaranteed to converge to *globally* optimal policies as well.

1. Introduction

Natural actor-critics are an increasingly popular class of algorithms for finding locally optimal policies for continuous-action Markov decision processes (MDPs). We show that the existing discounted natural actor-critic algorithms (Degris et al., 2012; Peters & Schaal, 2006; 2008) do not produce unbiased estimates of the natural policy gradient as sometimes purported (Peters & Schaal, 2006; 2008), since they are missing a γ^t term. Some algorithms do not claim to follow unbiased estimates of the natural policy gradient (Degris et al., 2012), however they are still missing the term, which results in additional bias.

Although the missing term is just a γ^t , we argue that it has significant ramifications, beyond voiding some convergence guarantees. We prove that, for a set of Markov decision processes, these biased *discounted* reward natural actor-critics are actually unbiased *average* reward natural actor-critics. We derive the first unbiased discounted-reward natural actor-critics, but we argue that our unbiased algorithms are *not* practical, and only support their

use when maximization of the discounted-reward objective is imperative. The goal of this paper is to raise awareness about what the algorithms that are in use are actually doing, not to present a superior method.

We also address a common misconception about policy gradient algorithms—that they are inferior to algorithms like Sarsa(λ) because they are local methods that can become stuck at arbitrarily bad locally optimal policies. We prove that, in all of the settings where Sarsa(λ) is guaranteed to reach a globally optimal policy, the average and discounted-reward objective functions have only global optima, and hence policy gradient methods also converge to *globally* optimal policies.

2. Problem

We are interested in the problem of finding locally optimal policies for Markov decision processes (MDPs). An MDP is a tuple, $(\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, d_0, \gamma)$. \mathcal{S} and \mathcal{A} denote the sets of possible states and actions, which may be countable (discrete), or uncountable (continuous).¹ \mathcal{P} is called the *transition function*, where $\mathcal{P}_{ss'}^a = \Pr(s_{t+1}=s'|s_t=s, a_t=a)$, where $t \in \mathbb{N}^0$ denotes the time step, $s, s' \in \mathcal{S}$ and $a \in \mathcal{A}$. \mathcal{R}_s^a is the expected value of the scalar reward, r_t , when action a is taken in state s , i.e., $\mathcal{R}_s^a = \mathbb{E}[r_t | s_t = s, a_t = a]$. We assume that $r_t \in [-r_{max}, r_{max}]$ for some uniformly bounding constant r_{max} . The initial state distribution is d_0 , where $d_0(s) = \Pr(s_0=s)$, and $\gamma \in [0, 1]$ is a discount factor. A *parameterized policy*, π , is a distribution over actions given a state and parameter vector $\theta \in \mathbb{R}^n$. That is, $\pi(s, a, \theta) = \Pr(a_t=a | s_t=s, \theta)$. We assume that, for all s, a , and θ , $\pi(s, a, \theta)$ is differentiable with respect to θ .

The *state value function*, V^θ , when using policy parameters θ , is a function that gives the expected sum of discounted reward (or expected *discounted return*) that would be accrued from a state, s , when following the policy with parameters θ . That is, $V^\theta(s) =$

¹We abuse notation by writing summations and probabilities over \mathcal{S} and \mathcal{A} . If these sets are continuous, the summations and probabilities should be replaced with integrals and probability densities. We also treat s_t and a_t as random variables or observed values depending on context.

$E[\sum_{t=0}^{\infty} \gamma^t r_t | s_0=s, \theta]$. Similarly, the *state-action value function* is $Q^\theta(s, a) = E[\sum_{t=0}^{\infty} \gamma^t r_t | s_0=s, a_0=a, \theta]$. The *discounted state distribution*, d^θ , gives the probability of each state when using policy parameters θ , with a discount applied to states that occur at later times: $d^\theta(s) = (1 - \gamma) \sum_{t=0}^{\infty} \gamma^t \Pr(s_t=s | s_0, \theta)$. The objective function, $J : \mathbb{R}^n \rightarrow \mathbb{R}$, gives the expected discounted return for using the provided policy parameters for one episode: $J(\theta) = E[\sum_{t=0}^{\infty} \gamma^t r_t | \theta]$, where an episode is one sequence of states, actions, and rewards, starting from a state sampled from d_0 and following the dynamics specified by \mathcal{P} and \mathcal{R} .

We call an MDP *episodic* if there is one or more state in which the process terminates, and, for all policies, every episode reaches a terminal state within some finite time, T . To model episodic MDPs in a unified manner with non-episodic MDPs, we assume that there is only one admissible action in terminal states, and it causes a transition to an absorbing state with zero reward, which we call a *post-terminal absorbing state*. This absorbing state also has only one admissible action, which causes a self-transition with zero reward. We allow $\gamma = 1$ only when the MDP is episodic.²

If \mathcal{S} and \mathcal{A} are discrete, then the goal is to find a policy parameter vector that maximizes J . If \mathcal{S} or \mathcal{A} is continuous, we search for locally optimal policy parameters, θ^* , that is, parameters satisfying $\nabla J(\theta^*) = 0$.

3. Policy Gradient

Gradient ascent algorithms for maximizing J are called *policy gradient* algorithms. Their basic update is $\theta_{t+1} = \theta_t + \alpha_t \nabla J(\theta_t)$, where $\{\alpha_t\}$ is a scalar step size schedule. Policy gradient methods may also use unbiased estimates of the gradient, making them *stochastic gradient ascent* algorithms. Gradient ascent is guaranteed to converge to a local maximum if J continuously differentiable, ∇J is Lipschitz, $\sum_{t=0}^{\infty} \alpha_t = \infty$, and $\sum_{t=0}^{\infty} \alpha_t^2 < \infty$ (Bertsekas & Tsitsiklis, 2000). We assume that these constraints are satisfied.

The policy gradient, $\nabla J(\theta)$, is the direction $\Delta\theta$ that maximizes $J(\theta + \Delta\theta)$ subject to $\|\Delta\theta\|^2 = \epsilon^2$, for infinitesimally small ϵ , where $\|\cdot\|$ denotes the Euclidean (L^2) norm. Amari (1998) suggested that Riemannian distance may be a more appropriate metric than Euclidean distance for parameter space. He calls the direction satisfying this modified constraint the *natural gradient*. Kakade (2002) suggested the application of natural gradients to policy gradients to get the *natural policy gradient*. Bagnell & Schneider (2003)

²If $\gamma = 1$, we modify the sum in the definition of d^θ to sum to T rather than ∞ .

then derived a proper Riemannian distance metric,³ based on Amari and Kakade’s work, and showed that the natural policy gradient is covariant. Bhatnagar et al. (2009) built on this foundation to create several provably convergent policy gradient and natural policy gradient algorithms for the *average reward* setting.

Let $f_\varpi(s, a) = \varpi^\top \psi_{sa}$ be a linear function approximator with weight vector $\varpi = [w^\top, v^\top]^\top$, where $|w| = |\theta|$, and feature vector $\psi_{sa} = [(\frac{\partial}{\partial \theta} \log \pi(s, a, \theta))^\top, \phi(s)^\top]^\top$, for arbitrary uniformly bounded ϕ . If

$$\sum_{s,a} d^\theta(s) \pi(s, a, \theta) [Q^\theta(s, a) - f_\varpi(s, a)] \frac{\partial f_\varpi(s, a)}{\partial \varpi} = 0, \quad (1)$$

then the natural policy gradient is $\tilde{\nabla} J(\theta) = w$ (Kakade, 2002).⁴

4. Finding w

To find w that satisfy (1), Sutton et al. (2000), working in the $|v| = 0$ setting, suggest letting $f_\varpi : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$ be an approximation to Q^θ with parameter vector $\varpi = w$. They claim that learning f_ϖ by following $\pi(s, \cdot, \theta)$ and updating ϖ by a rule such as $\Delta \varpi_t \propto \frac{\partial}{\partial \varpi} [\hat{Q}^\theta(s_t, a_t) - f_\varpi(s_t, a_t)]^2$, where $\hat{Q}^\theta(s, a)$ is some unbiased estimate of $Q^\theta(s, a)$, will result in satisfactory w . However, this is only true for the average reward setting or the discounted setting when $\gamma = 1$ because, in the discounted setting, d^θ in (1) is the *discounted* weighting of states encountered, whereas the states observed when merely following $\pi(s, \cdot, \theta)$ come from the *undiscounted* state distribution.

Peters & Schaal (2006; 2008) observed that the scheme proposed by Sutton et al. (2000) is a forward TD(1) algorithm. Because forward and backward TD(λ) are approximately equivalent, they suggest using least squares temporal difference (LSTD), a backwards TD(λ) method, to approximate Q^θ with f_ϖ , where $\lambda = 1$. They call the resulting algorithms the natural actor-critic using LSTD (NAC-LSTD) and the episodic natural actor-critic (eNAC). Because the scheme proposed by Sutton et al., and thus TD(1), does not incorporate the γ^t weighting in the discounted state distribution, this results in w that do not sat-

³Recent work has proposed the use of a different metric that accounts not only for how the distribution over actions (the policy) changes as the parameters change, but also for how the state distribution changes as the parameters change (Morimura et al., 2009).

⁴Notice that if $|\phi(s)| = 0$, we can drop v from Equation (1) to get the exact constraint specified by Sutton et al. (2000). Equation (1) follows immediately since $\sum_a \pi(s, a, \theta) v^\top \phi(s) \frac{\partial f_\varpi(s, a)}{\partial \varpi} = 0$ for all s, θ, ϕ , and ϖ . We include the *baseline*, $v^\top \phi(s)$, since it can reduce the variance of gradient estimates (Sutton et al., 2000). Also, for simplicity later, we assume that $\phi(s) = 0$ for the post-terminal absorbing state.

isfy (1), and thus bias in the natural policy gradient estimates. Natural actor-critic algorithms that do not include the γ^t term continue to be published (Degris et al., 2012).

One solution would be to convert the discounted MDP into an equivalent undiscounted MDP (Bertsekas & Tsitsiklis, 1996). To do this, each observed trajectory must be truncated after each transition with probability $1 - \gamma$. Notice that NAC-LSTD is not biased when $\gamma = 1$, because then the discounted and undiscounted state distributions are identical.⁵ So, after the trajectories are truncated, the existing NAC-LSTD algorithm could be used with $\gamma = 1$ to find a policy for the original MDP. However, this approach may discard significant amounts of data when truncating episodes. Instead, we propose the use of all of the observed data with proper discounting in order to produce unbiased gradient estimates.

We present a new objective function, H , and prove that the local minima of this objective give w satisfying (1). We then provide the stochastic gradient descent updates for this objective. When using policy parameters θ , the discounting from the discounted state distribution can be shifted into the objective function in order to properly satisfy (1). We select a w that is a component of a local minimum for the objective function H :

$$\begin{aligned} H(\varpi) &= \sum_{t=0}^{\infty} \sum_s \Pr(s_t = s | \theta) \sum_a \pi(s, a, \theta) \times \\ &\quad \left[\frac{\gamma^t}{2} (Q^\theta(s, a) - f_\varpi(s, a))^2 \right] \\ &= \sum_{t=0}^{\infty} \mathbb{E}_{s,a} \left[\frac{\gamma^t}{2} (\hat{Q}^\theta(s, a) - f_\varpi(s, a))^2 \right], \end{aligned} \quad (2)$$

where \times denotes scalar-scalar multiplication split across two lines. This objective function is always finite because either $\gamma < 1$ or the MDP is episodic. If the MDP is episodic, it must enter the post-terminal absorbing state within a finite number of steps. In this state, $\psi_{s,a} = 0$, and $Q^\theta(s, a) = 0$ for all π and the one admissible a , so $\sum_a \pi(s, a, \theta) \gamma^t (Q^\theta(s, a) - f_\varpi(s, a))^2 = 0$ for all ϖ . Hence, if the MDP is episodic, only a finite number of terms in the infinite sum will be non-zero.

We propose performing stochastic gradient descent on H to obtain a local minimum where $\frac{\partial}{\partial \varpi} H(\varpi) = 0$:

$$\begin{aligned} 0 &= \sum_s \sum_{t=0}^{\infty} \gamma^t \Pr(s_t = s | \theta) \sum_a \pi(s, a, \theta) \times \\ &\quad [Q^\theta(s, a) - f_\varpi(s, a)] \frac{\partial f_\varpi(s, a)}{\partial \varpi}. \end{aligned}$$

⁵It is unclear whether eNAC would be unbiased in this situation, as described in Section 6.

By the definition of d^θ , this is equivalent to (1). Hence, when gradient descent on H has converged, the resulting w component of ϖ satisfies (1). Notice that the expectation in (2) is over the observed probabilities of states and actions at time t when using θ . Hence, we can update ϖ via stochastic gradient descent:

$$\begin{aligned} \varpi &\leftarrow \varpi + \eta_t \times \\ &\quad \sum_{t=0}^{\infty} \left[\gamma^t (\hat{Q}^\theta(s_t, a_t) - f_\varpi(s_t, a_t)) \right] \frac{\partial f_\varpi(s_t, a_t)}{\partial \varpi}, \end{aligned} \quad (3)$$

where \hat{Q}^θ is an unbiased estimate of Q^θ and $\{\eta_t\}$ is a step size schedule that satisfies the typical decay constraints. The substitution of \hat{Q}^θ for Q^θ does not influence convergence (Bertsekas & Tsitsiklis, 2000). Because $\partial f_\varpi(s, a) / \partial \varpi$ is zero for terminal states and the post-terminal absorbing state, the above update need only be performed for the pre-terminal states. With $|v| = 0$, this differs from the method proposed by Sutton et al. (2000) only by the sum over time and the γ^t term.

5. Algorithms and Convergence

A simple algorithm to find w would be to execute episodes and then perform the update in (3) using the Monte Carlo return, $\hat{Q}^\theta(s_t, a_t) = \sum_{\tau=0}^{\infty} \gamma^\tau r_{t+\tau}$, as the unbiased estimate of $Q^\theta(s_t, a_t)$. This is a forward TD(1) algorithm, with an additional discount applied to updates based on the time at which they occur. However, this algorithm requires that entire trajectories be stored in memory. To overcome this, we can derive the equivalent *backwards* update by following Sutton and Barto’s derivation of backwards TD(λ) (Sutton & Barto, 1998).

The resulting on-policy backwards algorithm for estimating Q^θ for a fixed policy parameter vector θ is:

$$\begin{aligned} e_{t+1} &= \gamma \lambda e_t + \gamma^t \frac{\partial f_\varpi(s_t, a_t)}{\partial \varpi} \\ \delta_t &= r_t + \gamma f_\varpi(s_{t+1}, a_{t+1}) - f_\varpi(s_t, a_t) \\ \varpi_{t+1} &= \varpi_t + \eta_t \delta_t e_{t+1}, \end{aligned}$$

where λ is a decay parameter for eligibility traces as in TD(λ) and s_t, a_t , and r_t come from using policy parameter vector θ . Although the backwards and forward algorithms are only approximately equivalent (Sutton & Barto, 1998), their convergence guarantees are the same (Bertsekas & Tsitsiklis, 1996). Hence, if $\lambda = 1$ and η_t is decayed appropriately, the modified backwards TD(λ) algorithm above will produce w satisfying (1). The only difference between this algorithm and Sarsa(λ) is the γ^t in the equation for e_{t+1} .

One can then reproduce the work of Bradtke & Barto (1996) to create LSTD in this new setting, which approximates V^θ in a least squares manner. This can be extended

following the work of Lagoudakis & Parr (2001) to create LSQ, which approximates Q^θ in a least squares manner. The resulting LSQ algorithm in NAC-LSTD changes only by the introduction of a γ^t term: $z_{t+1} = \lambda z_t + \gamma^t \hat{\phi}_t$.⁶

To create an episodic algorithm, we convert (1) into a system of linear equations using the assumption that all episodes terminate within T steps. We rewrite (1) by replacing the infinite sum in d^θ with a finite one because $\partial f_\omega(s, a) / \partial \omega$ is zero for absorbing states:

$$0 = \sum_s \sum_{t=0}^T \Pr(s_t = s) \sum_a \pi(s, a, \theta) \gamma^t \times \\ (Q^\theta(s, a) - \varpi^\top \psi_{sa}) \psi_{sa} = 0.$$

This can be written as $\mathbf{A}\varpi = b$, where $b = \sum_{s,a} \sum_{t=0}^T \Pr(s_t=s) \pi(s, a, \theta) \gamma^t Q^\theta(s, a) \psi_{sa}$ and $\mathbf{A} = \sum_{s,a} \sum_{t=0}^T \Pr(s_t=s) \pi(s, a, \theta) \gamma^t \psi_{sa} \psi_{sa}^\top$. We can then generate unbiased estimates of \mathbf{A} and b from sample trajectories. As the number of observed trajectories grows, our estimates of \mathbf{A} and b converge to their true values, giving an unbiased estimate of the natural gradient. The resulting episodic natural actor-critic algorithm is presented in Algorithm 1. Notice that this is different from eNAC (Peters & Schaal, 2008), so we call it eNAC2.

For both algorithms presented, the user must select either TYPE1 or TYPE2 updates. In the former, which emulates the update scheme proposed by Peters & Schaal (2008), the policy is updated when the gradient estimate has converged, while in the latter, which emulates the two-timescale update scheme proposed by Bhatnagar et al. (2009), the policy is updated after a constant number of time steps. The user must also select $f(t) = \gamma^t$ to get the unbiased algorithm or $f(t) = 1$ to get the biased algorithm. The unbiased algorithms are only truly unbiased when $\lambda = 1$, $\beta = 0$ (if β is present), and $\epsilon \rightarrow 0$ (TYPE1) or $k \rightarrow \infty$ (TYPE2), in which case they compute and ascend the exact natural policy gradient.

NAC-LSTD and eNAC2 have computational complexity proportional to $|\varpi|^2$ per time step just to update statistics, and $|\varpi|^3$ to compute the natural policy gradient estimate for policy improvement steps. The complexity of policy improvement steps can be improved to $|\varpi|^2$ using the Sherman-Morrison formula to maintain estimates of \mathbf{A}^{-1} directly. It can be further improved to linear by using the modified Sarsa(λ) algorithm in place of LSTD to find w satisfying (1). We call the resulting algorithm the *Natural Actor-Critic using Sarsa(λ)*, or *NAC-S*. Notice that some mean zero terms can be removed from the Sarsa(λ) update and the resulting algorithm, provided in Algorithm 2, can be viewed as the discounted reward and eligibility trace extensions of the Natural-Gradient Actor-Critic with Advan-

⁶This equation uses the notation of Peters & Schaal (2008).

Algorithm 1 episodic Natural Actor Critic 2—eNAC2

- 1: **Input:** Parameterized policy $\pi(s, a, \theta)$ with initial parameters θ , basis functions $\phi(s)$ for the state-value estimation, update frequency parameter k , discount parameter γ , decay constant β , step size schedule $\{\eta_t\}$, and maximum episode duration T .
 - 2: $\mathbf{A} \leftarrow \mathbf{0}$;
 - 3: $b \leftarrow 0$; $\tau \leftarrow 0$
 - 4: **for** $ep = 0, 1, 2, \dots$ **do**
 - 5: Run an episode and remember the trajectory,
 - 6: $\{s_t, a_t, s_{t+1}, r_t\}$, where $t \in [0, T - 1]$.
 - 7: **Update Statistics:**
 - 8: $\mathbf{A} \leftarrow \mathbf{A} + \sum_{t=0}^T f(t) \psi_{s_t a_t} \psi_{s_t a_t}^\top$
 - 9: $b \leftarrow b + \sum_{t=0}^T f(t) \psi_{s_t a_t} \sum_{i=t}^T \gamma^{i-t} r_i$
 - 10: $[w_{ep}^\top, v_{ep}^\top]^\top = (\mathbf{A}^\top \mathbf{A})^{-1} \mathbf{A}^\top b$
 - 11: **Update Actor (Natural Policy Gradient):**
 - 12: **if** (TYPE1, $ep - k \geq 0$, and $\angle(w_{ep}, w_{ep-k}) \leq \epsilon$) **or**
 - 13: (TYPE2 and $(ep + 1) \bmod k = 0$) **then**
 - 14: $\theta \leftarrow \theta + \eta_\tau \frac{w_{ep}}{\|w_{ep}\|_2}$
 - 15: $\tau = \tau + 1$; $\mathbf{A} \leftarrow \beta \mathbf{A}$; $b \leftarrow \beta b$
-

tage Parameters (Bhatnagar et al., 2009).⁷ NAC-S can also be viewed as INAC (Degris et al., 2012) or NTD (Morimura et al., 2005) corrected to include the γ^t term and with the option of computing exact gradient estimates or using two-timescales.

Notice that in all algorithms presented in this paper, the natural gradient is normalized. This normalization is optional. It may void convergence guarantees and it often makes it difficult to achieve empirical convergence. However, in practice we find it easier to find fixed step sizes that work on difficult problems when using normalized updates to θ . Amari (1998) defined the natural gradient as only a direction and even discarded scaling constants in his derivation of a closed form for the natural gradient.

Peters & Schaal (2008) claim that the natural actor-critics compute and ascend the natural gradient of J , and thus will converge to a locally optimal policy, at which point $\nabla J(\theta) = 0$, assuming the step size schedules are properly decayed and that the natural actor-critic’s estimates of the natural gradient are unbiased. As stated previously, when $\lambda = 1$, $\beta = 0$ (if β is present), and $\epsilon \rightarrow 0$ (TYPE1) or $k \rightarrow \infty$ (TYPE2), the natural gradient estimates will be exact. In practice, large k or small ϵ and small fixed step sizes usually result in convergence.

Policy gradient approaches are typically purported to have one significant drawback: whereas Q -based methods converge to globally optimal policies for problems with discrete states and actions, policy gradient algorithms can become stuck in arbitrarily bad local optima (e.g., (Peters &

⁷To get Bhatnagar’s algorithm, select TYPE2 updates with $k = 1$, $f(t) = 1$, and replace the discounted TD error with the average reward TD error.

Algorithm 2 Natural Actor Critic using Sarsa(λ)—NAC-S(λ)

```

1: Input: Parameterized policy  $\pi(s, a, \theta)$  with initial parameters  $\theta$ , basis functions  $\phi(s)$  for the state-value estimation, update frequency parameter  $k$ , discount parameter  $\gamma$ , eligibility decay rate  $\lambda$ , and step size schedules  $\{\alpha_t^w\}$ ,  $\{\alpha_t^v\}$  and  $\{\eta_t\}$ .
2:  $w_0 \leftarrow 0$ ;  $v_0 \leftarrow 0$ ;  $count \leftarrow 0$ 
3: for episode = 0, 1, 2, ... do
4:   Draw initial state  $s_0 \sim d_0(\cdot)$ 
5:    $e_{-1}^w = 0$ ;  $e_{-1}^v = 0$ ;  $\tau_1 = 0$ ;  $\tau_2 = 0$ 
6:   for  $t = 0, 1, 2, \dots$  do
7:      $a_t \sim \pi(s_t, \cdot, \theta)$ ;  $s_{t+1} \sim \mathcal{P}(s_t, a_t, \cdot)$ ;
8:      $r_t \leftarrow \mathcal{R}_{s_t}^{a_t}$ ;  $count \leftarrow count + 1$ ;
9:     Update Critic (Sarsa):
10:     $\delta_t = r_t + \gamma v_t^\top \phi(s_{t+1}) - v_t^\top \phi(s_t)$ 
11:     $e_t^w = \gamma \lambda e_{t-1}^w + f(t) [\frac{\partial}{\partial \theta} \log \pi(s_t, a_t, \theta)]$ 
12:     $e_t^v = \gamma \lambda e_{t-1}^v + f(t) \phi(s_t)$ 
13:     $w_{t+1} = w_t + \alpha_{t-\tau_1}^w [\delta_t - w_t^\top [\frac{\partial}{\partial \theta} \log \pi(s_t, a_t, \theta)]] e_t^w$ 
14:     $v_{t+1} = v_t + \alpha_{t-\tau_1}^v \delta_t e_t^v$ 
15:    Update Actor (Natural Policy Gradient):
16:    if (TYPE1,  $t - k \geq 0$ , and  $\angle(w_t, w_{t-k}) \leq \epsilon$ ) or
17:      (TYPE2 and  $(count \bmod k = 0)$ ) then
18:         $\theta \leftarrow \theta + \eta_{\tau_2} \frac{w_{t+1}}{\|w_{t+1}\|_2}$ ;  $\tau_1 = t$ ;  $\tau_2 = \tau_2 + 1$ 
19:    if  $s_{t+1}$  terminal then break out of loop over  $t$ 
    
```

Bagnell, 2010; Peters, 2010)). We argue that with assumptions similar to those required by Q -learning and Sarsa, ascending the policy gradient results in convergence to a globally optimal policy as well.⁸

First, we assume that \mathcal{S} and \mathcal{A} are countable and that every state-action pair is observed infinitely often. Second, we assume that for all θ , all states s , and all actions a and \hat{a} , where $a \neq \hat{a}$, there is a direction $d\theta$ of change to θ that causes the probability of a in state s to increase while that of \hat{a} decreases, while all other action probabilities remain unchanged. These two assumptions are satisfied by policy parameterizations such as tabular Gibbs softmax action selection (Sutton & Barto, 1998).

We argue that at all suboptimal θ , the policy gradient will be non-zero. For any policy that is not globally optimal, there exists a reachable state for which increasing the probability of a specific action a while decreasing the probability of \hat{a} would increase J (see Section 4.2 of Sutton and Barto’s book (Sutton & Barto, 1998)). By our first assumption, this state-action pair is reached by the policy, and by our second assumption, there is a direction, $d\theta$, of change to θ that can make exactly this change. So, the directional derivative of J at θ in the direction $d\theta$ is non-zero and therefore the gradient of J at θ must also be non-zero. Hence, θ cannot be a local optimum.

Policy gradient is typically applied to problems with con-

⁸Notice that this applies to all algorithms that ascend the policy gradient or natural policy gradient with a well-behaved metric tensor.

tinuous state or action sets, in which case the assumptions above cannot be satisfied, so convergence to only a local optimum can be guaranteed. However, the above argument suggests that, in practice and on continuous problems, local optima can be avoided by increasing exploration and the representational power of the policy parameterization. However, if one desires a specific low-dimensional policy parameterization, such as a proportional-derivative controller with limited exploration, then increasing the exploration and representational power of the policy may not be an acceptable option, in which case local optima may be unavoidable.

6. Analysis of Biased Algorithms

In this section we analyze how the bias changes performance. Recall that, without the correct discounting, w are the weights that minimize the squared error in the Q^θ estimate, with states sampled from actual episodes. With the proper discounting, states that are visited at later times factor less into w . Because w will be the change to the policy parameters, this means that in the biased algorithms the change to the policy parameters considers states that are visited at later times just as much as states that are visited earlier. This suggests that the biased algorithms may be optimizing a different objective function similar to

$$\bar{J}(\theta) = (1 - \gamma) \sum_s \bar{d}^\theta(s) V^\theta(s), \quad (4)$$

where \bar{d}^θ is the stationary distribution of the Markov chain induced by the policy π . More formally, we assume $\bar{d}^\theta(s) = \lim_{t \rightarrow \infty} \Pr(s_t = s | s_0, \theta)$ exists and is independent of s_0 for all policies.

Notice that \bar{J} is not interesting for episodic MDPs since, for all policies, $\bar{d}^\theta(s)$ is non-zero for only the post-terminal absorbing state. So, henceforth, our discussion is limited to the non-episodic setting. For comparison, we can write J in the same form: $J(\theta) = \sum_s d_0(s) V^\theta(s)$. This original objective function gives the expected discounted return from an episode. This means that for small γ , it barely considers the quality of the policy at states that are visited late in a trajectory. On the other hand, \bar{J} considers states based on their visitation frequency, regardless of when they are visited. Kakade (2001) showed that \bar{J} , which includes discounting in V^θ , is the typical average reward objective function.

To see that the biased algorithms appear to optimize something closer to this average reward objective, consider an MDP with $\mathcal{S} = [0, 1]$, where $s_0 = 0$, $s = 1$ is terminal, $s_{t+1} = s_t + 0.01$, and $\mathcal{R}_s^a = -(a - s)^2$. The optimal policy is to select $a_t = s_t$. We parameterize the policy with one parameter, such that $a_t \sim \mathcal{N}(\theta, \sigma^2)$ for all states, where \mathcal{N} is a normal distribution with small constant variance, σ^2 . If

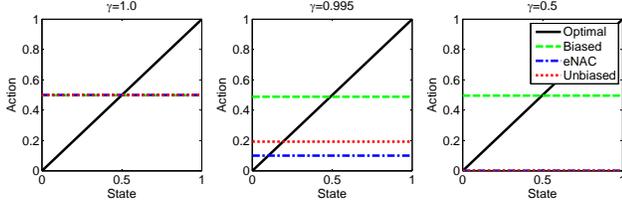


Figure 1. The optimal policy (optimal), the mean action selected by the biased NAC-LSTD, eNAC2, and INAC (biased), the mean action selected by the unbiased NAC-LSTD, eNAC2, NAC-S, as well as a random restart hill-climbing algorithm (unbiased), and the mean action selected by eNAC (eNAC). Note that the mean action is the policy parameter for each algorithm after training.

$\gamma = 1$, the optimal parameter, θ^* , is $\theta^* = 0.5$. Both the biased and unbiased algorithms converge to this θ^* . However, when $\gamma = 0.995$ or $\gamma = 0.5$, the optimal θ^* decreases in order to receive more reward initially. We found that the unbiased natural actor-critics properly converge to the new optimal θ^* , as does a simple hill-climbing algorithm that we implemented as a control. However, the biased algorithms still converge to $\theta^* \approx 0.5$.⁹ We found that eNAC converges to θ that differ from those of all other algorithms when $\gamma \neq 1$, which suggests that eNAC, but not eNAC2, may have additional bias. These results are presented in Figure 1.

This difference raises the question of whether the biased algorithms actually compute the natural policy gradient in the average reward setting. In the remainder of this section, we prove that they do whenever

$$\sum_s V^\theta(s) \frac{\partial \bar{d}^\theta(s)}{\partial \theta} = 0. \quad (5)$$

This can happen, for example, if the distribution over states does not depend on the policy, in which case $\frac{\partial \bar{d}^\theta(s)}{\partial \theta} = 0$.

The typical objective for average reward learning is $\bar{J}(\theta) = \lim_{n \rightarrow \infty} \frac{1}{n} \sum_{t=0}^{\infty} \mathbb{E}[r_t | \theta]$, which is equivalent to the definition in (4) (Kakade, 2001). The state-action value function is $\bar{Q}^\theta(s, a) = \sum_{t=0}^{\infty} \mathbb{E}[r_t - \bar{J}(\theta) | s_0 = s, a_0 = a, \theta]$. Kakade (2002) stated that the natural gradient of \bar{J} is $\tilde{\nabla} \bar{J}(\theta) = w$ if

$$\sum_{s,a} \bar{d}^\theta(s) \pi(s, a, \theta) [\bar{Q}^\theta(s, a) - f_\varpi(s, a)] \frac{\partial f_\varpi(s, a)}{\partial \varpi} = 0. \quad (6)$$

Thus, the unbiased average reward natural policy gradient is given by w satisfying (6). The biased algorithms sample states, s , from \bar{d}^θ and actions, a , from $\pi(s, \cdot, \theta)$ and perform gradient descent on the squared difference between

⁹We used random restarts for all methods and observed no local optima.

$Q^\theta(s, a)$ and $f_\varpi(s, a)$. So, they select w that satisfy

$$\sum_{s,a} \bar{d}^\theta(s) \pi(s, a, \theta) [Q^\theta(s, a) - f_\varpi(s, a)] \frac{\partial f_\varpi(s, a)}{\partial \varpi} = 0. \quad (7)$$

Notice that (7) uses the discounted state-action value function while (6) uses the average reward state-action value function.

To determine if and when the biased algorithms compute $\tilde{\nabla} \bar{J}(\theta)$, we must determine when a constant multiple of the solutions to (7) satisfy (6). To do this, we solve (7) for w and substitute a constant, $k > 0$, times these w into (6) to generate a constraint that, when satisfied, results in the biased algorithms producing the same direction (but not necessarily magnitude) as the average reward natural policy gradient. When doing so, we assume that $v = 0$, since it does not influence the solutions to either equation. First, we establish a lemma that relates the policy gradient theorem using the average reward state distribution but discounted reward state-action value function (left hand side of Lemma 1) to the derivative of \bar{J} without proper application of the chain rule:

Lemma 6.1. For all θ ,

$$\begin{aligned} \sum_{s,a} \bar{d}^\theta(s) \frac{\partial \pi(s, a, \theta)}{\partial \theta} Q^\theta(s, a) \\ = (1 - \gamma) \sum_s \bar{d}^\theta(s) \frac{\partial V^\theta}{\partial \theta}. \end{aligned}$$

Proof.

$$\begin{aligned} \frac{\partial V^\theta(s)}{\partial \theta} &= \frac{\partial}{\partial \theta} \sum_a \pi(s, a, \theta) Q^\theta(s, a) \\ &= \sum_a \left[\frac{\partial \pi(s, a, \theta)}{\partial \theta} Q^\theta(s, a) + \right. \\ &\quad \left. \pi(s, a, \theta) \frac{\partial}{\partial \theta} \left(\mathcal{R}_s^a + \sum_{s'} \mathcal{P}_{ss'}^a \gamma V^\theta(s') \right) \right] \\ &= \sum_a \left[\frac{\partial \pi(s, a, \theta)}{\partial \theta} Q^\theta(s, a) + \right. \\ &\quad \left. \pi(s, a, \theta) \sum_{s'} \mathcal{P}_{ss'}^a \gamma \frac{\partial}{\partial \theta} V^\theta(s') \right]. \end{aligned}$$

Solving for $\sum_a \frac{\partial \pi(s, a, \theta)}{\partial \theta} Q^\theta(s, a)$ yields

$$\begin{aligned} \sum_a \frac{\partial \pi(s, a, \theta)}{\partial \theta} Q^\theta(s, a) \\ = \frac{\partial V^\theta(s)}{\partial \theta} - \gamma \sum_a \pi(s, a, \theta) \sum_{s'} \mathcal{P}_{ss'}^a \frac{\partial V^\theta(s')}{\partial \theta}. \end{aligned}$$

Summing both sides over all states weighted by \bar{d}^θ gives

$$\begin{aligned}
 & \sum_{s,a} \bar{d}^\theta(s) \frac{\partial \pi(s,a,\theta)}{\partial \theta} Q^\theta(s,a) \\
 &= \left(\sum_s \bar{d}^\theta(s) \frac{\partial V^\theta(s)}{\partial \theta} \right) - \\
 & \quad \gamma \sum_s \bar{d}^\theta(s) \sum_a \pi(s,a,\theta) \sum_{s'} \mathcal{P}_{ss'}^a \frac{\partial V^\theta(s')}{\partial \theta} \\
 &= \sum_s \bar{d}^\theta(s) \frac{V^\theta(s)}{\partial \theta} - \\
 & \quad \gamma \sum_s \bar{d}^\theta(s) \frac{\partial V^\theta(s)}{\partial \theta} = (1-\gamma) \sum_s \bar{d}^\theta(s) \frac{\partial V^\theta(s)}{\partial \theta}.
 \end{aligned}$$

■

Solving (7) for w , which gives the direction of the biased algorithms, we get

$$\begin{aligned}
 w &= \left(\sum_s \bar{d}^\theta(s) \sum_a \pi(s,a,\theta) \psi_{sa} \psi_{sa}^\top \right)^{-1} \times \\
 & \quad \left(\sum_s \bar{d}^\theta(s) \sum_a \pi(s,a,\theta) Q^\theta(s,a) \psi_{sa} \right).
 \end{aligned}$$

Substituting k times this w into (6) for w and canceling the product of the Fisher information matrix and its inverse gives

$$\begin{aligned}
 0 &= \sum_s \bar{d}^\theta \sum_a \pi(s,a,\theta) \bar{Q}^\theta(s,a) \psi_{sa} - \\
 & \quad k \sum_s \bar{d}^\theta \sum_a \pi(s,a,\theta) Q^\theta(s,a) \psi_{sa} \\
 &= \frac{\partial \bar{J}(\theta)}{\partial \theta} - k(1-\gamma) \sum_s \bar{d}^\theta(s) \frac{\partial V^\theta(s)}{\partial \theta},
 \end{aligned}$$

by substitution of the policy gradient theorem (Sutton et al., 2000) and Lemma 1. Thus, when, for some k ,

$$\frac{\partial \bar{J}(\theta)}{\partial \theta} = k(1-\gamma) \sum_s \bar{d}^\theta(s) \frac{\partial V^\theta(s)}{\partial \theta}, \quad (8)$$

the biased algorithms produce the direction of the unbiased average reward natural policy gradient.

If we let $k = 1$, we will still get a constraint that results in the two directions being the same, although if the constraint is not satisfied, it does not mean the two are different (since a different k may result in (8) being satisfied). Setting $k = 1$ and substituting (4) for $\bar{J}(\theta)$, we get:

$$\begin{aligned}
 & \frac{\partial}{\partial \theta} (1-\gamma) \sum_s \bar{d}^\theta(s) V^\theta(s) = (1-\gamma) \sum_s \bar{d}^\theta(s) \frac{\partial V^\theta(s)}{\partial \theta} \\
 & \sum_s \bar{d}^\theta(s) \frac{\partial V^\theta(s)}{\partial \theta} + \frac{\partial \bar{d}^\theta(s)}{\partial \theta} V^\theta(s) = \sum_s \bar{d}^\theta(s) \frac{\partial V^\theta(s)}{\partial \theta} \\
 & \quad \sum_s V^\theta(s) \frac{\partial \bar{d}^\theta(s)}{\partial \theta} = 0.
 \end{aligned}$$

We have shown that when (5) holds, the biased algorithms compute the average reward natural policy gradient.

7. Discussion and Conclusion

We have shown that NAC-LSTD and eNAC produce biased estimates of the natural gradient. We argued that they, and INAC, act more like average reward natural actor-critics that do not properly account for how changes to θ change the expected return via d^θ . We proved that in certain situations the biased algorithms produce unbiased estimates of the natural policy gradient for the *average* reward setting. The bias stems from improper discounting when approximating the state-action value function using compatible function approximation. We derived the properly discounted algorithms to produce the unbiased NAC-LSTD and eNAC2, as well as the biased and unbiased NAC-S, a linear time complexity alternative to the squared to cubic time complexity NAC-LSTD and eNAC2. However, the unbiased algorithms have a critical drawback that limits their practicality.

The unbiased algorithms discount their updates by γ^t , which can result in poor data efficiency, particularly when γ is small.¹⁰ With small γ , the updates will decay to zero rapidly, causing the unbiased algorithms to ignore data collected after a short burn-in period. However, in some cases, this data inefficiency is unavoidable. Consider an MDP like the one presented earlier, where the set of states that occur early and those that occur later are disjoint. In this setting, the discounted reward objective mandates that data recorded late in trajectories must be ignored. In this situation, the rapid decay of updates is a curse of the choice of objective function. However, if the states that are visited early in a trajectory are also visited later in a trajectory, off-policy methods may be able to take advantage of data from late in an episode to provide meaningful updates even for the discounted reward setting. They may also be able to properly use data from previous policies to improve the estimates of the natural policy gradient in a principled manner. These are possible avenues for future research.

As stated in the introduction:

The goal of this paper is to raise awareness about what the algorithms that are in use are actually doing, not to present a superior method.

Until the data efficiency of these unbiased algorithms is improved, perhaps by leveraging off-policy techniques, they are of little practical value. We only recommend their use when optimization of the discounted-reward objective is

¹⁰Although the unbiased algorithms suffer from data efficiency problems, they are still more efficient than the naïve truncation approach discussed in Section 4, which discards data.

absolutely critical.

Another interesting extension would be to determine how γ should be selected in the *biased* algorithms. Recall that (4) is the average reward objective, for all γ . This suggests that in the biased algorithms, γ may be selected by the researcher. Smaller values of γ are known to result in faster convergence of value function estimates (Szepesvari, 1997), however larger γ typically result in smoother value functions that may be easier to approximate accurately with few features.

Lastly, we argued that, with certain policy parameterizations, policy gradient methods converge to globally optimal policies for discrete problems, and suggested that local optima may be avoided in continuous problems by increasing exploration and the policy's representational power. Future work may attempt to provide global convergence guarantees for a subset of the continuous-action setting by intelligently increasing the representational power of the policy when it becomes stuck in a local optimum.

References

- Amari, S. Natural gradient works efficiently in learning. *Neural Computation*, 10:251–276, 1998.
- Bagnell, J. A. and Schneider, J. Covariant policy search. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pp. 1019–1024, 2003.
- Bertsekas, D. P. and Tsitsiklis, J. N. *Neuro-Dynamic Programming*. Athena Scientific, Belmont, MA, 1996.
- Bertsekas, D. P. and Tsitsiklis, J. N. Gradient convergence in gradient methods. *SIAM J. Optim.*, 10:627–642, 2000.
- Bhatnagar, S., Sutton, R. S., Ghavamzadeh, M., and Lee, M. Natural actor-critic algorithms. *Automatica*, 45(11): 2471–2482, 2009.
- Bradtke, S. J. and Barto, A. G. Linear least-squares algorithms for temporal difference learning. *Machine Learning*, 22:33–57, 1996.
- Degrís, T., Pilarski, P. M., and Sutton, R. S. Model-free reinforcement learning with continuous action in practice. In *Proceedings of the 2012 American Control Conference*, 2012.
- Kakade, S. Optimizing average reward using discounted rewards. In *Proceedings of the 14th Annual Conference on Computational Learning Theory*, 2001.
- Kakade, S. A natural policy gradient. In *Advances in Neural Information Processing Systems*, volume 14, pp. 1531–1538, 2002.
- Lagoudakis, M. and Parr, R. Model-free least-squares policy iteration. In *Neural Information Processing Systems: Natural and Synthetic*, pp. 1547–1554, 2001.
- Morimura, T., Uchibe, E., and Doya, K. Utilizing the natural gradient in temporal difference reinforcement learning with eligibility traces. In *International Symposium on Information Geometry and its Application*, 2005.
- Morimura, T., Uchibe, E., Yoshimoto, J., and Doya, K. A generalized natural actor-critic algorithm. In *Neural Information Processing Systems: Natural and Synthetic*, 2009.
- Peters, J. Policy gradient methods. *Scholarpedia*, 5(11): 3698, 2010.
- Peters, J. and Bagnell, J. A. Policy gradient methods. *Encyclopedia of Machine Learning*, 2010.
- Peters, J. and Schaal, S. Policy gradient methods for robotics. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2006.
- Peters, J. and Schaal, S. Natural actor-critic. *Neurocomputing*, 71:1180–1190, 2008.
- Sutton, R. S. and Barto, A. G. *Reinforcement Learning: An Introduction*. MIT Press, Cambridge, MA, 1998.
- Sutton, R. S., McAllester, D., Singh, S., and Mansour, Y. Policy gradient methods for reinforcement learning with function approximation. In *Advances in Neural Information Processing Systems 12*, pp. 1057–1063, 2000.
- Szepesvari, C. S. The asymptotic convergence-rate of Q-learning. In *Advances in Neural Information Processing Systems*, volume 10, pp. 1064–1070, 1997.