Bias in Natural Actor-Critic Algorithms

Philip S. Thomas PTHOMAS@CS.UMASS.EDU
Department of Computer Science, University of Massachusetts, Amherst, MA 01002 USA

Technical Report UM-CS-2012-018

Abstract

We show that two popular discounted reward natural actor-critics, NAC-LSTD and eNAC, follow biased estimates of the natural policy gradient. We derive the first unbiased discounted reward natural actor-critics using batch and iterative approaches to gradient estimation and prove their convergence to *globally* optimal policies for discrete problems and locally optimal policies for continuous problems. Finally, we argue that the bias makes the existing algorithms more appropriate for the average reward setting.

1. Introduction

We show that two popular discounted reward natural actor-critics, NAC-LSTD and eNAC (Peters & Schaal, 2008), do not produce unbiased estimates of the natural policy gradient as purported. 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, even though they use estimates of discounted reward value functions.

Another algorithm, INAC (Degris et al., 2012), which is a variant of the NTD algorithm (Morimura et al., 2005), was originally presented as a biased discounted reward algorithm. We suggest that it is more appropriate to think of it as an average reward algorithm.

We derive the unbiased discounted reward NAC-LSTD, eNAC, and NAC-S algorithms, where NAC-S is a linear-time algorithm similar to NTD and INAC. We prove that unbiased policy gradient and natural policy gradient algorithms, like those presented, are convergent to *globally* optimal policies for discrete problems. However, the unbiased discounted reward algorithms suffer from updates that rapidly decay to zero, which causes poor data efficiency.

2. Problem

We are interested in the problem of finding optimal decision rules or *policies* for sequential decision tasks formulated as Markov decision processes (MDPs). An MDP is a tuple, $M = (\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, d_0, \gamma)$. $\mathcal{S}$ 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 $\mathcal{P}^a_{ss'} = \Pr(s_{t+1}{=}s' | s_t{=}s, a_t{=}a)$, where $t \in \mathbb{N}^0$ denotes the time step, $s, s' \in \mathcal{S}$ and $a \in \mathcal{A}$. $\mathcal{R}$ is the *reward function*, where $\mathcal{R}^{a_t}_{s_t} = r_t$, where $s_t \in \mathcal{S}$, $a_t \in \mathcal{A}$, and $r_t \in [-r_{max}, r_{max}]$ for some uniformly bounding constant r_{max}. The initial state distribution is d_0, where $d_0(s) = \Pr(s_0{=}s)$, and γ is a discount factor.

A *policy* or *stochastic policy*, $\pi \in \Pi$, is a distribution over actions given a state: $\pi(s, a) = \Pr(a_t{=}a | s_t{=}s)$, where Π is the set of all possible policies. A *parameterized policy* μ with parameters $\theta \in \mathbb{R}^n$ is a function that maps its parameters to policies, i.e., $\mu : \mathbb{R}^n \to \Pi$ and $\mu(\theta)(s, a) = \Pr(a_t{=}a | s_t{=}s, \theta_t{=}\theta)$. For brevity, we write μ_θ for $\mu(\theta)$. We assume that, for all s, a, and θ, $\mu_\theta(s, a)$ is differentiable with respect to θ.

The *state value function*, V^π, for policy π, is a function mapping states to the expected sum of discounted reward (or expected *return*) that would be accrued therefrom if π were executed on M. That is, $V^\pi(s) = \mathrm{E}[\sum_{t=0}^\infty \gamma^t r_t | s_0{=}s, \pi, M]$.[2] Similarly, the *state-action value function* is $Q^\pi(s, a) = \mathrm{E}[\sum_{t=0}^\infty \gamma^t r_t | s_0{=}s, a_0{=}a, \pi, M]$. The *discounted state distribution*, d^π, gives the probability of each state under policy π, with a discount applied to states that occur at later times: $d^\pi(s) = (1-\gamma) \sum_{t=0}^\infty \gamma^t \Pr(s_t = s | s_0, \pi, M)$. The objective functional, J, gives the expected discounted return for running the provided policy on M for one episode: $J(\pi) = \mathrm{E}[\sum_{t=0}^\infty \gamma^t r_t | \pi, M]$, where an episode is one sequence of states, actions,

[1] We 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.

[2] To avoid clutter, we may suppress functions' dependencies on M. For example, V^π is a function of M.

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 a finite number of steps. To model episodic MDPs in a unified manner with non-episodic MDPs, we follow the formulation specified by Sutton & Barto (1998), in which only one action is admissible 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 $\gamma \in [0, 1]$, where $\gamma = 1$ only when the MDP is episodic.[3]

If $\mathcal{S}$ and $\mathcal{A}$ are countable, then the goal is to find an optimal policy, π^*, which maximizes the objective functional: $\pi^* \in \arg\max_{\pi \in \Pi} J(\pi)$. If $\mathcal{S}$ or $\mathcal{A}$ is continuous, we search for locally optimal policy parameters, θ^*, that is, parameters satisfying $\nabla \mathcal{J}(\theta^*) = 0$, where $\mathcal{J} = J \circ \mu$, and where we assume $\mathcal{J}$ is Lipschitz.

3. Policy Gradient

Gradient ascent algorithms for maximizing the objective functional are called *policy gradient* algorithms. Their basic update is $\theta_{t+1} \leftarrow \theta_t + \alpha_t \nabla \mathcal{J}(\theta_t)$, 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. Stochastic gradient ascent is guaranteed to converge to a local maximum if $\mathcal{J}$ is Lipschitz, $\sum_{t=0}^{\infty} \alpha_t = \infty$, and $\sum_{t=0}^{\infty} \alpha_t^2 < \infty$ (Bertsekas & Tsitsiklis, 2000). We assume that all step size schedules hereafter satisfy these constraints.

The policy gradient, $\nabla \mathcal{J}(\theta)$, is the direction $\Delta\theta$ that maximizes $\mathcal{J}(\theta + \Delta\theta)$ under the constraint that $\|\Delta\theta\|^2 = \epsilon^2$, for a sufficiently 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) then derived a proper Reimannian distance metric,[4] 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.

At this point, it was known that if

$$\sum_s d^\pi(s) \sum_a \mu_\theta(s, a) \times \left[Q^{\mu_\theta}(s, a) - f_\varpi(s, a) \right] \frac{\partial f_\varpi(s, a)}{\partial \varpi} = 0, \tag{1}$$

where $f_\varpi(s, a)$ is a linear function approximator with parameter vector $\varpi = [w^\top, v^\top]^\top$, $|w| = |\theta|$, feature vector $\psi_{sa} = [(\frac{\partial}{\partial \theta} \log \mu_\theta(s, a))^\top, \phi(s)^\top]^\top$, for arbitrary uniformly bounded ϕ, and $f_\varpi(s, a) = \varpi \cdot \psi_{sa}$, then the natural policy gradient is $\widetilde{\nabla} \mathcal{J}(\theta) = w$ (Sutton et al., 2000; Kakade, 2002).[5] The challenge was then to devise a method for finding w satisfying Equation 1.

4. Finding w

To satisfy Equation 1, Sutton et al. (2000), working in the $|v| = 0$ setting, suggest letting $f_\varpi : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ be an approximation to Q^{μ_θ} with parameter vector $\varpi = w$. They claim that learning f_ϖ by following μ_θ and updating ϖ by a rule such as $\Delta\varpi_t \propto \frac{\partial}{\partial \varpi} [\hat{Q}^{\mu_\theta}(s_t, a_t) - f_\varpi(s_t, a_t)]^2$, where $\hat{Q}^{\mu_\theta}(s, a)$ is some unbiased estimate of $Q^{\mu_\theta}(s, a)$, 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^π in Equation 1 is the *discounted* weighting of states encountered, whereas the states observed when merely following μ_θ 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^{μ_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 satisfy

[3] If $\gamma = 1$, every episode reaches a terminal state within some finite time, T, so $d^\pi(s)$ sums to T.

[4] Recent 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).

[5] Notice 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 $\sum_a \mu_\theta(s, a) v^\top \phi(s) \frac{\partial f_\varpi(s,a)}{\partial \varpi} = 0$ for all s, μ_θ, ϕ, ϖ and M. Also, for simplicity later, we assume that $\phi(s) = 0$ for the post-terminal absorbing state.

Equation 1, and thus bias in the natural policy gradient estimates.

One solution would be to convert the discounted MDP into an equivalent undiscounted MDP, as described in Section 2.3 of 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.[6] 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 functional, H, and prove that the local minima of this objective give w satisfying Equation 1. We then provide the stochastic gradient ascent updates for this objective.

When following μ_θ, the discounting from the discounted state distribution can be shifted into the objective functional in order to properly satisfy Equation 1. We select a w that is a component of a local minimum for the objective functional H:

$$\begin{aligned} H(\varpi) =& \sum_{t=0}^{\infty} \sum_{s} \Pr\left(s_t = s | M, \mu_\theta\right) \sum_{a} \mu_\theta(s,a) \times \\ & \left[\gamma^t \left(Q^{\mu_\theta}(s,a) - f_\varpi(s,a)\right)^2\right] \\ =& \sum_{t=0}^{\infty} \mathrm{E}_{s,a}\left[\gamma^t \left(\hat{Q}^{\mu_\theta}(s,a) - f_\varpi(s,a)\right)^2\right]. \end{aligned} \tag{2}$$

The objective functional 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 $Q^\pi(s,a) = 0$ for all π and the one admissible a, so $\sum_a \mu_\theta(s,a)\gamma^t \left(Q^{\mu_\theta}(s,a) - f_\varpi(s,a)\right)^2 = 0$ 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 $\frac{\partial}{\partial \varpi} H(\varpi) = 0$, so

$$\begin{aligned} \sum_{s} \sum_{t=0}^{\infty} & \gamma^t \Pr(s_t = s | M, \mu_\theta) \sum_{a} \mu_\theta(s,a) \times \\ & \left[Q^{\mu_\theta}(s,a) - f_\varpi(s,a)\right] \frac{\partial f_\varpi(s,a)}{\partial \varpi} = 0. \end{aligned} \tag{3}$$

[6] It is unclear whether eNAC would be unbiased in this situation, as described in Section 7.

By the definition of d^π, this is equivalent to Equation 1. Hence, when gradient descent on H has converged, the resulting w component of ϖ satisfies Equation 1.

Notice that the expectation in Equation 2 is over the observed probabilities of states and actions at time t if executing μ_θ on M. Hence, we can update ϖ via stochastic gradient descent:

$$\begin{aligned} \varpi \leftarrow & \varpi + \eta \times \\ & \sum_{t=0}^{\infty} \left[\gamma^t \left(\hat{Q}^{\mu_\theta}(s_t,a_t) - f_\varpi(s_t,a_t)\right)\right] \frac{\partial f_\varpi(s_t,a_t)}{\partial \varpi}, \end{aligned} \tag{4}$$

where $\hat{Q}^{\mu_\theta}$ is an unbiased estimate of Q^{μ_θ} and η is a step size satisfying the typical decay constraints. The substitution of $\hat{Q}^{\mu_\theta}$ for Q^{μ_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

A simple algorithm to find w would be to execute episodes and then perform the update in Equation 4 using the Monte Carlo return, $\hat{Q}^{\mu_\theta}(s_t,a_t) = \sum_{\tau=0}^{\infty} \gamma^\tau r_{t+\tau}$, as the unbiased estimate of $Q^{\mu_\theta}(s_t,a_t)$. 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^{μ_θ} for a fixed μ_θ is:

$$e_{t+1} = \gamma\lambda e_t + \gamma^t \frac{\partial f_\varpi(s_t,a_t)}{\partial \varpi} \tag{5}$$

$$\delta_t = r_t + \gamma f_\varpi(s_{t+1},a_{t+1}) - f_\varpi(s_t,a_t) \tag{6}$$

$$\varpi_{t+1} = \varpi_t + \eta_t \delta_t e_{t+1}, \tag{7}$$

where λ is a decay parameter for eligibility traces as in TD(λ) and s_t, a_t, and r_t come from running μ_θ on M. 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 Equation 1. The only difference between this algorithm and Sarsa(λ) is the γ^t in Equation 5. One can then reproduce the work of Bradtke & Barto (1996) to create

LSTD in this new setting, which approximates V^{μ_θ} in a least squares manner. This can be extended along the lines of Lagoudakis & Parr (2001) to create LSQ, which approximates Q^{μ_θ} in a least squares manner. The resulting LSQ algorithm in Peters and Schaal's NAC-LSTD changes only by the introduction of a γ^t term: $z_{t+1} = \lambda z_t + \gamma^t \hat{\phi}_t$. We omit the complete pseudocode for NAC-LSTD due to space constraints.

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

$$\sum_s \sum_{t=0}^{T} \Pr(s_t = s) \sum_a \mu_\theta(s,a)\gamma^t \times \left(Q^{\mu_\theta}(s,a) - \varpi \cdot \psi_{sa}\right)\psi_{sa} = 0. \tag{8}$$

By collecting the terms with ϖ on the left and the others on the right, we get

$$\sum_s \sum_{t=0}^{T} \Pr(s_t = s) \sum_a \mu_\theta(s,a)\gamma^t \psi_{sa}\psi_{sa}^\top \varpi = b, \tag{9}$$

where $b = \sum_{s,a} \sum_{t=0}^{T} \Pr(s_t{=}s)\mu_\theta(s,a)\gamma^t Q^{\mu_\theta}(s,a)\psi_{sa}$. If we let $\mathbf{A} = \sum_{s,a} \sum_{t=0}^{T} \Pr(s_t{=}s)\mu_\theta(s,a)\gamma^t \psi_{sa}\psi_{sa}^\top$, then we get the system of linear equations: $\mathbf{A}\varpi = b$, where $\mathbf{A}$ is a $|\psi_{sa}|$ by $|\psi_{sa}|$ square matrix. 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, eNAC2, is presented in Algorithm 1.

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 $\epsilon \to 0$ (Type1) or $k \to \infty$ (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. This complexity can be improved to linear by using the modified Sarsa(λ) algorithm in place of LSTD to find w satisfying Equation 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 Advantage 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.

Algorithm 1 episodic Natural Actor Critic 2—eNAC2

Input: MDP M, parameterized policy $\mu_\theta(s,a)$ with initial parameters θ, basis functions $\phi(s)$ for the state-value estimation, update frequency parameter k, discount parameter γ, decay constant β, learning rate schedule $\{\eta_t\}$, and maximum episode duration T.
$\mathbf{A} \leftarrow \mathbf{0}$; $b \leftarrow 0$; $\tau \leftarrow 0$
for $ep = 0, 1, 2, \ldots$ **do**
  Run an episode and remember the trajectory, $\{s_t, a_t, s_{t+1}, r_t\}$, $t \in [0, T-1]$.
  Update Statistics:
  $\mathbf{A} \leftarrow \mathbf{A} + \sum_{t=0}^{T} f(t)\psi_{s_t a_t}\psi_{s_t a_t}^\top$
  $b \leftarrow b + \sum_{t=0}^{T} f(t)\psi_{s_t a_t} \sum_{\hat{t}=t}^{T} \gamma^{\hat{t}-t} r_{\hat{t}}$
  $[w_{ep}^\top, v_{ep}^\top]^\top = \left(\mathbf{A}^\top\mathbf{A}\right)^{-1}\mathbf{A}^\top b$; // If Type2, this need only be done every k episodes.
  Update Actor (Natural Policy Gradient):
  if (Type1, $ep - k \geq 0$, and $\measuredangle(w_{ep}, w_{ep-k}) \leq \epsilon$)**or**
    (Type2 and $(ep+1) \mod k = 0$) **then**
    $\theta \leftarrow \theta + \eta_\tau \frac{w_{ep}}{||w_{ep}||_2}$
    $\tau = \tau + 1$; $\mathbf{A} \leftarrow \beta\mathbf{A}$; $b \leftarrow \beta b$

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 defined the natural gradient as only a direction and even discarded scaling constants in his derivation of a closed form for the natural gradient (Amari, 1998).

6. Convergence

The natural actor-critics compute and ascend the natural gradient of $\mathcal{J}$, and thus will converge to a locally optimal policy, at which point $\nabla\mathcal{J}(\theta) = 0$, assum-

[7] To 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 **N**atural **A**ctor **C**ritic using **S**arsa(λ)—NAC-S(λ)

Input: MDP M, parameterized policy $\mu_\theta(s,a)$ with initial parameters θ, basis functions $\phi(s)$ for the state-value estimation, update frequency parameter k, discount parameter γ, eligibility decay rate λ, and learning rate schedules $\{\alpha_t^w\}$, $\{\alpha_t^v\}$ and $\{\eta_t\}$.
$w_0 \leftarrow 0$; $v_0 \leftarrow 0$; $count \leftarrow 0$
for episode $= 0, 1, 2, \ldots$ **do**
  Draw initial state $s_0 \sim d_0(\cdot)$
  $e^w_{-1} = 0$; $e^v_{-1} = 0$; $\tau_1 = 0$; $\tau_2 = 0$
  for $t = 0, 1, 2, \ldots$ **do**
    $a_t \sim \mu_\theta(s_t, \cdot)$; $s_{t+1} \sim \mathcal{P}(s_t, a_t, \cdot)$; $r_t \leftarrow \mathcal{R}^{a_t}_{s_t}$;
    $count \leftarrow count + 1$
    Update Critic (Sarsa):
    $\delta_t = r_t + \gamma v_t \cdot \phi(s_{t+1}) - v_t \cdot \phi(s_t)$
    $e^w_t = \gamma\lambda e^w_{t-1} + f(t)[\frac{\partial}{\partial\theta} \log \mu_\theta(s_t, a_t)]$
    $e^v_t = \gamma\lambda e^v_{t-1} + f(t)\phi(s_t)$
    $w_{t+1} = w_t + \alpha^w_{t-\tau_1}[\delta_t - w_t \cdot [\frac{\partial}{\partial\theta} \log \mu_\theta(s_t, a_t)]]e^w_t$
    $v_{t+1} = v_t + \alpha^v_{t-\tau_1}\delta_t e^v_t$
    Update Actor (Natural Policy Gradient):
    if (TYPE1, $t - k \geq 0$, and $\measuredangle(w_t, w_{t-k}) \leq \epsilon$) **or**
      (TYPE2 and ($count \mod k = 0$)) **then**
      $\theta \leftarrow \theta + \eta_{\tau_2} \frac{w_{t+1}}{||w_{t+1}||_2}$; $\tau_1 = t$; $\tau_2 = \tau_2 + 1$
    if s_{t+1} terminal **then** break out of loop over t

ing the step size schedules are properly decayed and that the natural actor-critic's estimates of the natural gradient are unbiased (Amari, 1998). As stated previously, when $\lambda = 1$, $\beta = 0$ (if β is present), and $\epsilon \to 0$ (TYPE1) or $k \to \infty$ (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 & 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 sub-optimal 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 $\mathcal{J}$ (see Section 4.2 of 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 $\mathcal{J}$ at θ in the direction $d\theta$ is non-zero and therefore the gradient of $\mathcal{J}$ at θ must also be non-zero. Hence, θ cannot be a local optimum.

Policy gradient is typically applied to problems with continuous 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.

7. 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^{μ_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 functional similar to

$$\bar{\mathcal{J}}(\theta) = (1-\gamma)\sum_s \bar{d}^{\mu_\theta}(s) V^{\mu_\theta}(s), \qquad (10)$$

where $\bar{d}^\pi$ is the stationary distribution of the Markov chain induced by the policy π. More formally, we assume $\bar{d}^\pi(s) = \lim_{t\to\infty} \Pr(s_t = s | s_0, \pi, M)$ exists and is independent of s_0 for all policies. Notice that $\bar{\mathcal{J}}$ is not interesting for episodic MDPs since, for all policies, $\bar{d}^\pi(s)$ 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 $\mathcal{J}$ in the same form: $\mathcal{J}(\theta) = \sum_s d_0(s) V^{\mu_\theta}(s)$. The original objective functional, $\mathcal{J}$, gives the expected return from an episode. This means that for small γ,

[8] Notice that this applies to all algorithms that ascend the policy gradient or natural policy gradient.

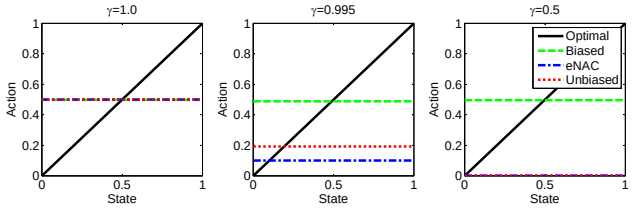

Figure 1. The optimal policy (optimal), the action selected by the biased NAC-LSTD, eNAC2, and INAC (biased), the action selected by the unbiased NAC-LSTD, eNAC2, NAC-S, as well as a random restart hill-climbing algorithm (unbiased), and the action selected by eNAC (eNAC).

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^{μ_θ}, is the typical average reward objective functional.

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 $\mathcal{R}^a_s = -(a - s)^2$. The optimal policy is to select $a_t = s_t$. We parameterize the policy with one parameter, such that μ_θ selects action $a_t \sim \mathcal{N}(\theta, \sigma^2)$ for all states, where $\mathcal{N}$ is a normal distribution with small constant variance, σ^2. If $\gamma = 1$, 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

$$\sum_s V^{\mu_\theta}(s) \frac{\partial \bar{d}^{\mu_\theta}(s)}{\partial \theta} = 0. \tag{11}$$

To derive Equation 11, we first review results concerning the average reward natural policy gradient. The typical objective for average reward learning is

$$\bar{J}(\theta) = \lim_{n\to\infty} \frac{1}{n} \sum_{t=0}^{\infty} \mathrm{E}[r_t | \mu_\theta, M]. \tag{12}$$

As mentioned previously, Kakade (2001) showed that this is equivalent to the definition in Equation 10. The state-action value function is defined as

$$\bar{Q}^{\mu_\theta}(s, a) = \sum_{t=0}^{\infty} \mathrm{E}[r_t - \bar{J}(\theta) | s_0 = s, a_0 = a, \mu_\theta, M]. \tag{13}$$

Kakade (2002) stated that if

$$\sum_s \bar{d}^{\pi}(s) \sum_a \mu_\theta(s, a) \times \left[\bar{Q}^{\mu_\theta}(s, a) - f_\varpi(s, a)\right] \frac{\partial f_\varpi(s, a)}{\partial \varpi} = 0 \tag{14}$$

then the natural gradient of $\bar{J}$ is

$$\widetilde{\nabla} \bar{J}(\theta) = w. \tag{15}$$

Thus, the unbiased average reward natural policy gradient is given by w satisfying Equation 14.

The biased algorithms perform stochastic gradient descent according to the scheme proposed by Sutton et al. (2000). They sample states, s, from $\bar{d}^{\mu_\theta}$ and actions, a, from μ_θ and perform gradient descent on the squared difference between $Q^{\mu_\theta}(s, a)$ and $f_\varpi(s, a)$. Thus, they select w satisfying

$$\sum_s \bar{d}^{\pi}(s) \sum_a \mu_\theta(s, a) \times \left[Q^{\mu_\theta}(s, a) - f_\varpi(s, a)\right] \frac{\partial f_\varpi(s, a)}{\partial \varpi} = 0. \tag{16}$$

Notice that Equation 16 uses the discounted state-action value function while Equation 14 uses the average reward state-action value funciton.

To determine if and when the biased algorithms compute $\widetilde{\nabla} \bar{J}(\theta)$, we must determine when a constant multiple of the solutions to Equation 16 satisfy Equation 14. To do this, we solve Equation 16 for w and substitute a constant, $k > 0$, times these w into Equation 14 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 must 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

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

1) to the derivative of $\bar{\mathcal{J}}$ without proper application of the chain rule:

Lemma 1

$$\sum_{s,a} \bar{d}^{\mu_\theta}(s) \frac{\partial \mu_\theta(s,a)}{\partial \theta} Q^{\mu_\theta}(s,a) = (1-\gamma) \sum_s \bar{d}^{\mu_\theta}(s) \frac{\partial V^{\mu_\theta}}{\partial \theta},$$

for all θ, μ, and M. For a proof of Lemma 1, see the appendix.

Solving Equation 16 for w, which gives the direction of the biased algorithms, we get

$$w = \left(\sum_s \bar{d}^{\mu_\theta}(s) \sum_a \mu_\theta(s,a) Q^{\mu_\theta}(s,a) \psi(sa) \right) \times \left(\sum_s \bar{d}^{\mu_\theta}(s) \sum_a \mu_\theta(s,a) \psi(sa) \psi(sa)^\top \right)^{-1}. \quad (17)$$

Notice that the second term is the inverse (average) Fisher information matrix (Bagnell & Schneider, 2003). Substituting k times this w into Equation 14 for w and canceling the product of the Fisher information matrix and its inverse gives

$$\begin{aligned} 0 =& \sum_s \bar{d}^{\mu_\theta} \sum_a \mu_\theta(s,a) \bar{Q}^{\mu_\theta}(s,a) \psi_{sa} - \\ & k \sum_s \bar{d}^{\mu_\theta} \sum_a \mu_\theta(s,a) Q^{\mu_\theta}(s,a) \psi_{sa} \\ =& \frac{\partial \bar{\mathcal{J}}(\theta)}{\partial \theta} - k(1-\gamma) \sum_s \bar{d}^{\mu_\theta}(s) \frac{\partial V^{\mu_\theta}(s)}{\partial \theta}, \end{aligned} \quad (18)$$

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

$$\frac{\partial \bar{\mathcal{J}}(\theta)}{\partial \theta} = k(1-\gamma) \sum_s \bar{d}^{\mu_\theta}(s) \frac{\partial V^{\mu_\theta}(s)}{\partial \theta}, \quad (19)$$

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 Equation 19 being satisfied). Setting $k = 1$ and substituting Equation 10 for $\bar{\mathcal{J}}(\theta)$, we get:

$$\begin{aligned} & \frac{\partial}{\partial \theta}(1-\gamma) \sum_s \bar{d}^{\mu_\theta}(s) V^{\mu_\theta}(s) = (1-\gamma) \sum_s \bar{d}^{\mu_\theta}(s) \frac{\partial V^{\mu_\theta}(s)}{\partial \theta} \\ & \sum_s \bar{d}^{\mu_\theta}(s) \frac{\partial V^{\mu_\theta}(s)}{\partial \theta} + \frac{\partial \bar{d}^{\mu_\theta}(s)}{\partial \theta} V^{\mu_\theta}(s) = \sum_s \bar{d}^{\mu_\theta}(s) \frac{\partial V^{\mu_\theta}(s)}{\partial \theta} \\ & \sum_s V^{\mu_\theta}(s) \frac{\partial \bar{d}^{\mu_\theta}(s)}{\partial \theta} = 0. \end{aligned} \quad (20)$$

We have shown that when Equation 11 holds, the biased algorithms compute the average reward natural policy gradient.

8. Discussion and Conclusion

We have shown that NAC-LSTD and eNAC produce biased estimates of the natural gradient. We argued that they, and INAC and NTD, act more like average reward natural actor-critics that do not properly account for how changes to θ change the expected return via d^{μ_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. For small γ, the updates will decay to zero rapidly, causing the unbiased algorithms to ignore data collected after a short burn-in period. 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.

Another interesting extension would be to determine how γ should be selected in the biased algorithms. Recall that Equation 10 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.

Degris, 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.

Appendix: Proof of Lemma 1

$$\frac{\partial V^{\mu_\theta}(s)}{\partial\theta} = \frac{\partial}{\partial\theta}\sum_a \mu_\theta(s,a)Q^{\mu_\theta}(s,a) \tag{21}$$

$$= \sum_a \left[\frac{\partial\mu_\theta(s,a)}{\partial\theta}Q^{\mu_\theta}(s,a) + \mu_\theta(s,a)\frac{\partial}{\partial\theta}Q^{\mu_\theta}(s,a)\right]$$

$$= \sum_a \left[\frac{\partial\mu_\theta(s,a)}{\partial\theta}Q^{\mu_\theta}(s,a) + \mu_\theta(s,a)\frac{\partial}{\partial\theta}\left(\mathcal{R}^a_s + \sum_{s'}\mathcal{P}^a_{ss'}\gamma V^{\mu_\theta}(s')\right)\right]$$

$$= \sum_a \left[\frac{\partial\mu_\theta(s,a)}{\partial\theta}Q^{\mu_\theta}(s,a) + \mu_\theta(s,a)\sum_{s'}\mathcal{P}^a_{ss'}\gamma\frac{\partial}{\partial\theta}V^{\mu_\theta}(s')\right].$$

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

$$\sum_a \frac{\partial\mu_\theta(s,a)}{\partial\theta}Q^{\mu_\theta}(s,a) = \tag{22}$$

$$\frac{\partial V^{\mu_\theta}(s)}{\partial\theta} - \gamma\sum_a \mu_\theta(s,a)\sum_{s'}\mathcal{P}^a_{ss'}\frac{\partial V^{\mu_\theta}(s')}{\partial\theta}.$$

Summing both sides over all states weighted by $\bar{d}^{\mu_\theta}$ gives

$$\sum_s \bar{d}^{\mu_\theta}(s)\sum_a \frac{\partial\mu_\theta(s,a)}{\partial\theta}Q^{\mu_\theta}(s,a) \tag{23}$$

$$= \left(\sum_s \bar{d}^{\mu_\theta}(s)\frac{\partial V^{\mu_\theta}(s)}{\partial\theta)}\right) - \gamma\sum_s \bar{d}^{\mu_\theta}(s)\sum_a \mu_\theta(s,a)\times \sum_{s'}\mathcal{P}^a_{ss'}\frac{\partial V^{\mu_\theta}(s')}{\partial\theta}$$

$$= \sum_s \bar{d}^{\mu_\theta}(s)\frac{V^{\mu_\theta}(s)}{\partial\theta} - \gamma\sum_s \bar{d}^{\mu_\theta}(s)\frac{\partial V^{\mu_\theta}(s)}{\partial\theta}$$

$$= (1-\gamma)\sum_s \bar{d}^{\mu_\theta}(s)\frac{\partial V^{\mu_\theta}(s)}{\partial\theta}.$$

□