The Thirty-Third AAAI Conference on Artificial Intelligence (AAAI-19)

Natural Option Critic

Saket Tiwari
College of Information and Computer Sciences
University of Massachusetts Amherst
Amherst, MA 01003
sakettiwari@umass.edu

Philip S. Thomas
College of Information and Computer Sciences
University of Massachusetts Amherst
Amherst, MA 01003
pthomas@cs.umass.edu

Abstract

The recently proposed *option-critic* architecture (Bacon, Harb, and Precup 2017) provides a stochastic policy gradient approach to hierarchical reinforcement learning. Specifically, it provides a way to estimate the gradient of the expected discounted return with respect to parameters that define a finite number of temporally extended actions, called *options*. In this paper we show how the option-critic architecture can be extended to estimate the *natural* gradient (Amari 1998) of the expected discounted return. To this end, the central questions that we consider in this paper are: **1)** what is the definition of the natural gradient in this context, **2)** what is the Fisher information matrix associated with an option's parameterized policy, **3)** what is the Fisher information matrix associated with an option's parameterized termination function, and **4)** how can a *compatible function approximation* approach be leveraged to obtain natural gradient estimates for both the parameterized policy and parameterized termination functions of an option with per-time-step time and space complexity linear in the total number of parameters. Based on answers to these questions we introduce the natural option critic algorithm. Experimental results showcase improvement over the *vanilla gradient* approach.

Introduction

Hierarchical reinforcement learning methods enable agents to tackle challenging problems by identifying reusable *skills*—temporally extended actions—that simplify the task. For example, a robot agent that tries to learn to play chess by reasoning solely at the level of how much current to give to its actuators every 20ms will struggle to correlate obtained rewards with their true underlying cause. However, if this same agent first learns skills to move its arm, grasp a chess piece, and move a chess piece, then the task of learning to play chess (leveraging these skills) becomes tractable. Several mathematical frameworks for hierarchical reinforcement learning have been proposed, including *hierarchies of machines* (Parr and Russell 1998), MAXQ (Dietterich 2000), and the options framework (Sutton, Precup, and Singh 1999). However, none of these frameworks provides a practical mechanism for *skill discovery*: determining what skills will be useful for an agent to learn. Although skill discovery methods have been proposed, they tend to be *heuristic* in that they find skills that have a property that intuitively might make for good skills for some problems, but which do not follow directly from the primary objective of optimizing the expected discounted return (Machado, Bellemare, and Bowling 2017; Simsek and Barto 2008; Thrun and Schwartz 1995; Konidaris and Barto 2009).

The *option-critic* architecture (Bacon, Harb, and Precup 2017), stands out from other attempts at developing a general framework for skill discovery in that it searches for the skills that directly optimize the expected discounted return. Specifically, the option critic uses the aforementioned options framework, wherein a skill is called an *option*, and it proposes parameterizing all aspects of the option and then performing stochastic gradient descent on the expected discounted return with respect to these parameters. The key insight that enables the option-critic architecture is a set of theorems that give expressions for the gradient of the expected discounted return with respect to the different parameters of an option.

One limitation of the option critic is that it uses ordinary (stochastic) gradient descent. In this paper we show how the option critic can be extended to use *natural gradient descent* (Amari 1998), which exploits the underlying structure of the option-parameter space to produce a more informed update direction. The primary contributions of this work are theoretical: we define the natural gradients associated with the option critic, derive the *Fisher information matrices* associated with an option's parameterized policy and termination function, and show how the natural gradients can be estimated with per-time-step time and space complexity linear in the total number of parameters. This is achieved by means of *compatible function approximations*. We also analyze the performance of natural gradient descent based approach on various learning tasks.

Preliminaries and Notation

A *reinforcement learning* (RL) agent interacts with an environment, modeled as a *Markov decision process* (MDP), over a sequence of time steps $t \in \mathbb{N}_{\geq 0}$. A finite MDP is a tuple $(\mathcal{S}, \mathcal{A}, P, R, d_0, \gamma)$. $\mathcal{S}$ is the finite set of possible states of the environment. S_t is the state of the environment at time t. $\mathcal{A}$ is the finite set of possible actions the agent can take. A_t is the action taken by the agent at time t. $P : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$ is the transition function: $P(s, a, s') = \Pr(S_{t+1}{=}s' | S_t{=}s, A_t{=}a)$, for all t. Meaning,

$P(s, a, s')$ the probability of transitioning to state s' given the agent takes action a in state s. R_t denotes the reward at time t. R is the *reward function*, $R : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$, where $R(s, a) = \mathbb{E}[R_t|S_t{=}a, A_t{=}a]$, i.e., the expected reward the agent receives given it took action a in state s. We say that a process has ended when the environment enters a *terminal state*, meaning for a terminal state s, $P(s, a, s') = 0$ and $R(s, a) = 0$ for all $s' \in \mathcal{S} \backslash \{s\}$ and $a \in \mathcal{A}$. The process ends after T steps and we call T the *horizon*. We say the process is *infinite horizon* when there does not exist a finite T. d_0 is the initial state distribution, i.e., $d_0(s) = \Pr(S_0{=}s)$. The parameter $\gamma \in [0, 1]$ scales how the rewards are discounted over time. When a terminal state is reached, time is reset to $t = 0$ and consequently a new initial state is sampled using d_0.

A policy, $\pi : \mathcal{S} \times \mathcal{A} \rightarrow [0, 1]$, represents the agent's decision making system: $\pi(s, a) = \Pr(A_t{=}a|S_t{=}s)$. Given a policy, π, and an MDP, $(\mathcal{S}, \mathcal{A}, P, R, d_0, \gamma)$, an episode, H is a sequence of states of the environment, actions taken by the agent, and the rewards observed from the initial state, S_0, to the terminal state, S_T, i.e., $H = (S_0, A_0, R_0, S_1, A_1, R_1, ..., S_T, A_T, R_T)$. We also define the path that an agent takes to be a sequence of states and actions, i.e., a history without rewards, $X = (S_0, A_0, S_1, A_1, ..., S_T, A_T)$. Path X is a random variable from the set of all possible paths, $\mathcal{X}$. The return of an episode H is the discounted sum of all rewards, $g(H) = \sum_{t=0}^{T} \gamma^t R_t$. We call v_π the value function for the policy π, $v_\pi : \mathcal{S} \rightarrow \mathbb{R}$, where $v_\pi(s) = \mathbb{E}[\sum_{t=0}^{T} \gamma^t R_t|, S_0{=}s, \pi]$. We call q_π the action-value function associated with policy π, $q_\pi : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$, where $q_\pi(s, a) = \mathbb{E}[\sum_{t=0}^{T} \gamma^t R_t|S_0{=}s, A_0{=}a, \pi]$.

Policy Gradient Framework

The *policy gradient framework* (Sutton et al. 1999; Konda and Tsitsiklis 2000) assumes the policy π, parametrized by θ, is differentiable. The objective function, ρ, is defined with respect to a start state s_0, $\rho(\theta) = \mathbb{E}[\sum_{t=0}^{T} \gamma^t R_t|d_0, \theta]$. The agent learns by updating the parameters θ approximately proportional to the gradient $\partial\rho/\partial\theta$, i.e., $\theta \leftarrow \alpha\partial\rho/\partial\theta$ where α is the *learning rate* (LR): a scalar hyper-parameter.

Option Critic framework

The *options framework* (Sutton, Precup, and Singh 1999) formalizes the notion of temporal abstractions by introducing options. An option, o, from a set of options, $\mathcal{O}$, is a generalization of primitive actions. The intra-option policy $\pi_o : \mathcal{S} \times \mathcal{A} \rightarrow [0, 1]$ represents the agent's decision making while executing an option o: $\pi_o(s, a) = \Pr(A_t{=}a|S_t{=}s, O_t{=}o)$. Like primitive actions the agent executes an option at a state S_t and the option terminates at another $S_{t+\tau}$, where τ is the duration for which the agent is executing the option: o_t. While in the option o, from state S_t to $S_{t+\tau}$, the agent follows the policy π_o. Option o terminates stochastically in state s according to a distribution β. The framework puts restrictions on where an option can be initiated by defining an initiation state set, $\mathcal{I}_o$, for option o. The option o is initiated in state $s \in \mathcal{I}_o$ based on $\pi_\mathcal{O}(s)$, which is a policy over options defined as $\pi_\mathcal{O} : \mathcal{S} \times \mathcal{O} \rightarrow [0, 1]$. An initiation state set $\mathcal{I}_o$, an intra-option policy π_o and a termination function $\beta_o : \mathcal{S} \rightarrow [0, 1]$ comprise an option o. It is commonly assumed that all options are available everywhere and thereby we dispense with the notion of an initiation set.

The *option critic framework* makes all the options available everywhere, and introduces policy-gradient theorems within the options framework. The option active at time step t is O_t. The intra-option policies (π_o) and termination functions (β_o) are represented using differentiable functions parametrized by θ and ϑ, respectively. The goal is to optimize the expected discounted return starting at state s_0 and option o_0. We redefine the objective function, ρ, for the option critic setting: $\rho(\mathcal{O}, \theta, \vartheta, s, o) = \mathbb{E}[\sum_{t=0}^{\infty} \gamma^t R_t|\mathcal{O}, \theta, \vartheta, S_0 = s, O_0 = o]$.

Equations similar to those in the policy gradient framework (Sutton et al. 1999) are manipulated to derive gradients of the objective with respect to θ and ϑ in the option-critic framework. The analogous state value function is $v_{\pi_\mathcal{O}} : \mathcal{S} \rightarrow \mathbb{R}$, where $v_{\pi_\mathcal{O}}(s) = \mathbb{E}[\sum_t \gamma^t R_t|S_0{=}s]$. $v_{\pi_\mathcal{O}}(s)$ is the value of a state s, within the options framework, with the option set $\mathcal{O}$ and the policy over options $\pi_\mathcal{O}$. The option-value function is $q_{\pi_\mathcal{O}} : \mathcal{S} \times \mathcal{O} \rightarrow \mathbb{R}$, where $q_{\pi_\mathcal{O}}(s, o) = \mathbb{E}[\sum_t \gamma^t R_t|S_0{=}s, O_0{=}o]$. Here, $q_{\pi_\mathcal{O}}(s, o)$ is the value of state s when option o is active with the option set $\mathcal{O}$. The state-option-action value function is $q_U : \mathcal{S} \times \mathcal{O} \times \mathcal{A} \rightarrow \mathbb{R}$, where $q_U(s, o, a) = \mathbb{E}[\sum_t \gamma^t R_t|S_0{=}s, O_0{=}o, A_0{=}a]$. Here, $q_U(s, o, a)$ is the value of executing action a in the context of state-option pair (s, o). The option-value function *upon arrival* is $u : \mathcal{O} \times \mathcal{S} \rightarrow \mathbb{R}$, where $u(o, s') = \mathbb{E}[\sum_t \gamma^t R_t|S_1{=}s', O_0{=}o]$. Here, $u(o, s')$ is the value of option o being active upon the agent entering state s'. Bacon, Harb, and Precup (2017) observe a consequence of the definitions:

$$u(o, s') = (1 - \beta_o(s'))q_{\pi_\mathcal{O}}(s', o) + \beta_o(s')v_{\pi_\mathcal{O}}(s').$$

The main results presented by Bacon, Harb, and Precup (2017) are the *intra-option policy gradient theorem* and the *termination gradient theorem*. The gradient of the expected discounted return with respect to θ and initial condition (s_0, o_0) is:

$$\frac{\partial q_{\pi_\mathcal{O}}(s_0, o_0)}{\partial\theta} = \sum_{s,o} \mu_\mathcal{O}(s, o) \sum_a \frac{\partial\pi_o(s, a, \theta)}{\partial\theta} q_U(s, o, a),$$

where $\mu_\mathcal{O}(s, o)$ is the discounted weighting of state-option pair (s, o) along trajectories starting from (s_0, o_0) defined by : $\mu_\mathcal{O}(s, o) = \sum_{t=0}^{\infty} \gamma^t \Pr(S_t{=}s, O_t{=}o|s_0, o_0)$. The gradient of the expected discounted return with respect to ϑ and initial condition (s_1, o_0) is:

$$\frac{\partial u(o_0, s_1)}{\partial\vartheta} = -\sum_{o,s'} \mu_\mathcal{O}(s', o) \frac{\partial\beta_o(s', \vartheta)}{\partial\vartheta} a_\mathcal{O}(s', o),$$

where $a_\mathcal{O} : \mathcal{S} \times \mathcal{O} \rightarrow \mathbb{R}$ is the advantage function over options such that $a_\mathcal{O}(s', o) = q_{\pi_\mathcal{O}}(s', o) - v_{\pi_\mathcal{O}}(s')$. Here, $\mu_\mathcal{O}(s', o)$ is the discounted weighting of state option pair (s', o) from (s_1, o_0), i.e., according to a Markov chain shifted by one time step, defined by: $\mu_\mathcal{O}(s', o) = \sum_{t=0}^{\infty} \gamma^t \Pr(S_{t+1}{=}s', O_t{=}o|s_1, o_0)$. The agent learns by updating parameters θ and ϑ in the direction approximately proportional to $\partial q_\mathcal{O}(s_0, o_0)/\partial\theta$ and $\partial u(o_0, s_1)/\partial\vartheta$, respectively.

Meaning, it learns by updating $\theta \leftarrow \alpha_\theta \partial q_{\pi_{\mathcal{O}}}(s_0, o_0)/\partial\theta$ and $\vartheta \leftarrow \alpha_\vartheta \partial u(o_0, s_1)/\partial\vartheta$, where α_θ and α_ϑ are the learning rates for θ and ϑ, respectively.

Natural Actor Critic

Natural gradient descent (Amari 1998) exploits the underlying structure of the parameter space when defining the direction of steepest descent. It does so by defining the inner product $\langle \mathbf{x}, \mathbf{y} \rangle_\theta$ in the parameter space as:

$$\langle \mathbf{x}, \mathbf{y} \rangle_\theta = \mathbf{x}^T G_\theta \mathbf{y}, \tag{1}$$

where G_θ is called the *metric tensor*. Although the choice of G_θ remains open under certain conditions (Thomas et al. 2016) we choose the Fisher information matrix, as is common practice. The fisher information matrix distribution over random variable X, parametrized by policy parameters θ, that lie on a Reimannian manifold (Rao 1945; Amari 1985):

$$(G_\theta)_{i,j} = \mathbb{E}\left[\frac{\partial \ln \Pr(X;\theta)}{\partial \theta_i}\frac{\partial \ln \Pr(X;\theta)}{\partial \theta_j}\right],$$

where the expectation is over the distribution $\Pr(X)$ and $(G_\theta)_{i,j}$ represents a matrix with its i, j^{th} element being the expression as defined on the right hand side — we use this notation to represent a matrix throughout the paper. Kakade (2001) makes the assumption that every policy, π, is ergodic and irreducible, therefore it has a well-defined *stationary distribution* for each state s. Under this assumption, Kakade (2001) introduces the use of natural gradient for optimizing the expected reward over the parameters θ of policy π, as defined by $\rho(\theta) = \sum_{s,a} d^\pi(s)\pi(s,a,\theta)R(s,a)$. The *natural gradient* for the objective function, ρ, is defined as:

$$\widetilde{\nabla}\rho(\theta) = G_\theta^{-1}\frac{\partial \rho(\theta)}{\partial \theta}. \tag{2}$$

The derivation of a closed form expression for G_θ for the parameter space of policy π, parametrized by θ, is non-trivial as demonstrated for the limiting matrix of the infinite horizon problem in reinforcement learning (Bagnell and Schneider 2003). For a weight vector w let $\hat{q}_w$ be an approximation of the state action value function $q(s,a)$, which has the form:

$$\hat{q}_w(s,a) = w^T \frac{\partial \ln \pi(s,a,\theta)}{\partial \theta}.$$

The mean squared error $\epsilon(w,\theta)$, for a weight vector w and a given policy parametrized by θ, is defined as:

$$\epsilon(w,\theta) = \sum_{s,a} d^\pi(s)\pi(s,a,\theta)(\hat{q}_w(s,a) - q_\pi(s,a))^2,$$

where $d^\pi(s) = \sum_{t=0}^{\infty}\gamma^t \Pr(S_t{=}s|\pi)$ is the discounted weighting of state s in the infinite horizon problem. The weights $d^\pi(s)$ normalize to the stationary distribution for state s under policy π in the undiscounted setting where the MDP terminates at every time step t with probability $1-\gamma$. Theorem 1 as introduced by Kakade (2001) states that $\tilde{w}$ which minimizes the mean squared error, $\epsilon(w,\theta)$, is equal to the natural gradient as defined in (2).

Kakade (2001) also demonstrates how natural policy gradient performs under the re-scaling of parameters. In addition to that, Kakade (2001) demonstrates how the natural gradient weights the components of $\widetilde{\nabla}\rho(\theta)$ uniformly, instead of using $d^\pi(s)$. We also point out that the natural gradient is independent to local re-parametrization of the model (Pascanu and Bengio 2013) and can be used in *online learning* (Degris, Pilarski, and Sutton 2012). Natural gradients for reinforcement learning (Peters and Schaal; Bhatnagar et al. 2008; 2009; Degris, Pilarski, and Sutton 2012), as well as more recent work in deep neural networks (Desjardins et al. 2015; Pascanu and Bengio 2013; Thomas, Dann, and Brunskill 2018; Sun and Nielsen 2017) have shown to be effective in learning.

The Option-Critic architecture uses vanilla gradient to learn temporal abstraction and internal policies, which can be less data efficient compared to the natural gradient (Amari 1998). The natural gradient also overcomes the difficulty posed by the *plateau phenomena* (Amari 2016). We derive the metric tensors for the parameters in the option-critic architecture. Computing the complete Fisher information matrix or is expensive. We use a block-diagonal estimate of the Fisher information matrix as has been applied in the past to reinforcement learning (Thomas 2011) and to neural networks (Roux, Manzagol, and Bengio 2008; Kurita 1992; Martens 2010; Pascanu and Bengio 2013; Martens and Grosse 2015). Specifically, we estimate G_θ and G_ϑ separately, where θ and ϑ are the parameters of of the intra-option policy and the option termination function. These are then combined into a $(|\theta| + |\vartheta|) \times (|\theta| + |\vartheta|)$ sized estimate of the complete Fisher information matrix of the parameter space, where $|\theta|, |\vartheta|$ represent the size of vectors.

We also provide theoretical justification for the resulting algorithm inspired from the incremental natural actor critic algorithm (Bhatnagar et al. 2007) (INAC) and its extension to include eligibility traces (Morimura, Uchibe, and Kenji 2005; Thomas 2014).

Start State Fisher Information Matrix Over Intra-Option Path Manifold

We define path X in the options framework for the infinite horizon problem as the sequence of state-option-action tuples: $X = (S_0, O_0, A_0, S_1, O_1, A_1, ...)$. We use $\mathcal{X}$ to denote the set of all paths. We introduce the function $g : \mathcal{X} \to \mathbb{R}$ called the *expected return over path*, where $g(x) = \mathbb{E}[\sum_{t=0}^{T}\gamma^t R_t|x]$ is the expected return given the path x. The goal in a reinforcement learning problem, in the context of the option-critic architecture, is to maximize the discounted return, $\rho(\mathcal{O}, \theta, \vartheta, s_0, o_0)$. The goal can be re-written as maximizing $J(\theta, s_0, o_0) = \sum \Pr(x;\theta)g(x)$. Where the summation is over all $x \in \mathcal{X}$ starting from (s_0, o_0) and the intra-option policies are parametrized by θ. To optimize the objective J, we define it over a Riemannian space Θ, with $\theta \in \Theta$. In the Riemannian space the inner product is defined as in (1). The direction of steepest ascent of $J(\theta)$ in the Riemannian space, Θ, is given by $G_\theta^{-1}\partial J(\theta)/\partial\theta$ (Amari 1998), (see equation (2)).

In this section we use ∂_i to denote $\partial/\partial\theta_i$ and use $\langle f(X)\rangle_{\Pr(X)}$ to indicate the expected value of f with respect to distribution $\Pr(X)$. We obtain an alternative form of the Fisher information matrix which is a well know result

(DeGroot 1970) (for details see appendix):

$$(G_\theta)_{i,j} = -\langle \partial_i \partial_j \ln \Pr(X;\theta)\rangle_{\Pr(X;\theta)}. \tag{3}$$

Fisher Information Matrix Over Intra-Option Path Manifold

In Theorem 1 we show that the Fisher information matrix over the paths, X, truncated to terminate at time step $\mathcal{T}$ converges as $\mathcal{T} \to \infty$ to the Fisher information matrix over the intra-option policies, π_o. This gives an expression for Fisher information matrix over the set of paths, $\mathcal{X}$, and simplifies computation of the natural gradient when maximizing the objective $J(\theta, s_0, o_0)$. We use $G_\theta^{\mathcal{T}}$ to indicate the $\mathcal{T}$-step finite horizon Fisher information matrix, meaning the Fisher information matrix if the problem were to be reduced to terminate at step $\mathcal{T}$. We normalize the metric by the total length of path $\mathcal{T}$ (Bagnell and Schneider 2003) to get a convergent metric.

Theorem 1 (Infinite Horizon Intra-Option Matrix)**.** *Let $G_\theta^{\mathcal{T}}$ be the $\mathcal{T}$-step finite horizon Fisher information matrix and $\langle G_\theta \rangle_{\mu_{\mathcal{O}}(s,o)}$ be the Fisher information matrix of intra-option policies under a stationary distribution of states, actions and options: $\pi_o(s,a,\theta)\mu_{\mathcal{O}}(s,o)$. Then:*

$$\lim_{\mathcal{T}\to\infty} \frac{1}{\mathcal{T}} G_\theta^{\mathcal{T}} = \langle G_\theta \rangle_{\mu_{\mathcal{O}}(s,o)}.$$

Proof. See the appendix (supplementary materials). □

Compatible Function Approximation For Intra-Option Path Manifold

We subtract the option-state value function, $q_{\pi_{\mathcal{O}}}$, from the state-option-action value function, q_U, and treat it as a baseline to reduce variance in the gradient estimate of the expected discounted return. The baseline can be a function of both state and action in special circumstances, but none of those apply here (Thomas and Brunskill 2017). So, we define the *state-option-action advantage function* $a_U : \mathcal{S}\times\mathcal{O}\times\mathcal{A} \to \mathbb{R}$. Where $a_U(s,o,a) = q_U(s,o,a) - q_{\mathcal{O}}(s,o)$ is the advantage of the agent taking action a in state s in the context of option o. Here, a_U is approximated by some compatible function approximator $f_\eta^{\pi_o}$. For vector η and parameters θ we define:

$$f_\eta^{\pi_o}(s,a) = \eta^T \left(\frac{\partial \ln(\pi_o(s,a,\theta))}{\partial \theta}\right). \tag{4}$$

The $\tilde{\eta}$ that is a local minima of the squared error $\epsilon(\eta,\theta)$:

$$\epsilon(\eta,\theta) = \sum_{s,o,a} \mu_{\mathcal{O}}(s,o)\pi_o(s,a,\theta)(f_\eta^{\pi_o}(s,a) - a_U(s,o,a))^2.$$

is equal to the natural gradient of the objective, ρ, with respect to ϑ (the complete derivation is in the appendix):

$$\widetilde{\nabla}_\theta q_{\pi_{\mathcal{O}}}(s_0,o_0) = G_\theta^{-1} \frac{\partial q_{\pi_{\mathcal{O}}}(s_0,o_0)}{\partial \theta} = \tilde{\eta}.$$

Thus, for a *sensible* (Kakade 2001) function approximation, as in (4), in the option-critic framework the natural gradient of the expected discounted return is the weights of linear function approximation.

Start State Fisher Information Matrix Over State-Option Transition Path Manifold

We derive the Fisher information matrix for the parameters ϑ over the state-option transitions path manifold. We define X' as a path for state-option transitions in the option-critic architecture. More specifically, we define $X' = (O_0, S_1, O_1, S_2, O_2, S_3, ...)$ to be path tuples of state option pairs shifted by one time step. We define $\mathcal{X}'$ to be the set of all state-option transition paths. Similar to the previous section, we define the *expected return over state-option transitions* $g' : \mathcal{X}' \to \mathbb{R}$, where $g'(x') = \mathbb{E}[\sum_{t=0}^{T} \gamma^t R_t | x']$ is the expected return given state-option transitions path x'. The goal can be re-written to maximize $J'(\vartheta, s_1, o_0) = \sum \Pr(x')g'(x')$. Where the summation is over all $x' \in \mathcal{X}'$ starting from (s_1, o_0) and terminations are parametrized by ϑ. To optimize J' we define it over a Reimannian space Θ' with $\vartheta \in \Theta'$ and the inner product defined as in (1), similar to previous section. The direction of steepest ascent in the Reimannian space, Θ', is the natural gradient.

In this section, we use ∂_i to denote $\partial/\partial\vartheta_i$ and use $\langle f(X')\rangle_{\Pr(X')}$ to indicate the expected value of $f(X')$ with respect to the distribution $\Pr(X')$. Equation (3) implies that the Fisher information matrix can be written as:

$$(G_\vartheta)_{i,j} = -\langle \partial_i \partial_j \ln \Pr(X';\vartheta)\rangle_{\Pr(X';\vartheta)}.$$

Fisher Information Matrix Over State-Option Transition Path Manifold

In Theorem 2 we show that the Fisher information matrix over the paths, X', truncated to terminate at time step $\mathcal{T}$ converges as $\mathcal{T} \to \infty$ to an expression in terms of the terminations and the policy over options over the stationary distribution of states and options. This gives an expression for Fisher information Matrix over set of paths, $\mathcal{X}'$, and simplifies computation of the natural gradient when maximizing the objective $J'(\vartheta, s_1, o_0)$.

Theorem 2 (Infinite Horizon State-Option Transition Matrix)**.** *Let $G_\vartheta^{\mathcal{T}}$ be the $\mathcal{T}$-step finite horizon Fisher information matrix and $\mu_{\mathcal{O}}(s',o)$ is the stationary distribution of state-option pairs s', o. Then:*

$$\left(\lim_{\mathcal{T}\to\infty} \frac{1}{\mathcal{T}} G_\vartheta^{\mathcal{T}}\right)_{i,j} = -\langle \partial_i \ln \beta_o(s',\vartheta)$$
$$\partial_j \ln(1 - \beta_o(s',\vartheta) + \beta_o(s',\vartheta)\pi_{\mathcal{O}}(s',o))\rangle_{\mu_{\mathcal{O}}(s',o)}.$$

Proof. See appendix (supplementary materials). □

Compatible Function Approximation For State-Option Transition Path Manifold

We define the *advantage function of continued option* as: $a'_{\mathcal{O}} : \mathcal{S}\times\mathcal{O} \to \mathbb{R}$. Where $a'_{\mathcal{O}}(s',o) = u(o,s') - q_{\pi_{\mathcal{O}}}(s',o)$ is the advantage of the option o being active while exiting s' given that option o is active when the agent enters s'. We consider terminations improvement when $a'_{\mathcal{O}}$ is approximated by some compatible function approximator $h_\varphi^{\beta_o}$. For vector φ and parameters ϑ we define:

$$h_\varphi^{\beta_o}(s') = \varphi^T \frac{\partial \ln(1 - \beta_o(s',\vartheta) + \pi_{\mathcal{O}}(s',o)\beta_o(s',\vartheta)))}{\partial \vartheta}. \tag{5}$$

We define the squared error $\epsilon(\varphi, \vartheta)$ associated with vector φ as:

$$\epsilon(\varphi, \vartheta) = \sum_{s',o} \mu_{\mathcal{O}}(s', o) L(O_{t+1}{=}o | O_t{=}o, S_{t+1}{=}s'; \vartheta)$$
$$(h_{\varphi}^{\beta_o}(s') - a'_{\mathcal{O}}(s', o))^2,$$

where $L(O_{t+1}{=}o | O_t{=}o, S_{t+1} = s'; \vartheta)$ is the likelihood ratio of option o being active while exiting s' given that option o is active when the agent enters s'. It is defined as follows:

$$\begin{aligned} L(O_{t+1}{=}o | O_t{=}o, S_{t+1} = s'; \vartheta) & \\ &= \frac{\Pr(O_{t+1}{=}o | O_t{=}o, S_{t+1} = s'; \vartheta)}{\Pr(O_{t+1}{\neq}o | O_t{=}o, S_{t+1} = s'; \vartheta)} \\ &= \frac{\beta'_o(s', \vartheta)}{1 - \beta'_o(s', \vartheta)}. \end{aligned}$$

We assume, throughout the paper, that the denominator is not 0. The $\tilde{\varphi}$ that is a local minima of $\epsilon(\varphi)$ satisfies (the complete derivation is in the appendix):

$$\widetilde{\nabla}_{\vartheta} u(o_0, s_1) = G_{\vartheta}^{-1} \frac{\partial u(o_0, s_1)}{\partial \vartheta} = -\tilde{\varphi}.$$

Therefore, for an approximation of the continued state-option value function, as in (5), the natural gradient of the expected discounted return is the negative weights of the linear function approximation.

Incremental Natural Option Critic Algorithm

We introduce algorithms inspired from the incremental natural actor critic introduced by Degris, Pilarski, and Sutton (2012), who in turn built on the theoretical work of Bhatnagar et al. (2007). The algorithm learns the parameters for approximations of state-option-action advantage function, a_U, and the advantage function of continued option, $a'_{\mathcal{O}}$, incrementally by taking steps in the direction of reducing the error $\epsilon(\eta, \theta)$ and $\epsilon(\varphi, \vartheta)$. It does stochastic gradient descent using the gradients $\partial\epsilon(\eta, \vartheta)/\partial\eta$ and $\partial\epsilon(\varphi, \vartheta)/\partial\varphi$. Learning the parameters η and φ leads to natural gradient based updates for θ and ϑ. We introduce hyper parameters $\alpha_\eta, \alpha_\varphi$ and λ, which are the learning rate for η, the learning rate for φ and the λ the eligibility trace parameter of both η and φ, respectively. The algorithm learns the policy over options, $\pi_{\mathcal{O}}$, using intra-option Q-learning (Sutton, Precup, and Singh 1999) as in previous work (Bacon, Harb, and Precup 2017).

The algorithm uses *TD-error* style updates to learn θ and ϑ. Analogous to the consistent estimates used by Bhatnagar et al. (2007), we state that a *consistent estimate* of the state-option value function, $\hat{q}_{\pi_{\mathcal{O}}}$, satisfies $\mathbb{E}[\hat{q}_{\pi_{\mathcal{O}}}(s_t, o_t) | s_t, o_t, \pi_{\mathcal{O}}, \pi_{o_t}, \beta_{o_t}] = q_{\pi_{\mathcal{O}}}$. Similarly, a consistent estimate of the value function upon arrival, $\hat{u}$, satisfies $\mathbb{E}[\hat{u}(o_t, s_{t+1}) | o_t, s_{t+1}, \pi_{\mathcal{O}}, \pi_{o_t}, \beta_{o_t}] = u(o_t, s_{t+1})$. We define the TD-error for the intra-option policies at time step t to be $\delta_t^U = r_t + \gamma \hat{u}(o_t, s_{t+1}) - \hat{q}_{\pi_{\mathcal{O}}}(s_t, o_t)$.

A consistent estimate of the state value function, $\hat{v}_{\pi_{\mathcal{O}}}$, satisfies $\mathbb{E}[\hat{v}_{\pi_{\mathcal{O}}}(s_t) | s_t, \pi_{\mathcal{O}}, \pi_{o_t}, \beta_{o_t}] = v_{\pi_{\mathcal{O}}}(s_t)$. We define the TD-error at time step t for the terminations to be $\delta_t^{\mathcal{O}} = r_t + \gamma \hat{v}_{\pi_{\mathcal{O}}}(s_{t+1}) - \hat{v}_{\pi_{\mathcal{O}}}(s_t)$. We provide Lemmas 1 and 2 to show that δ_t^U and $\delta_t^{\mathcal{O}}$ are consistent estimates of a_U and $a_{\mathcal{O}}$.

Lemma 1. *Given intra-option policies, π_o for all $o \in \mathcal{O}$, policy over options, $\pi_{\mathcal{O}}$, and terminations, β_o for all $o \in \mathcal{O}$, then:*

$$\mathbb{E}[\delta_t^U | s_t, a_t, o_t, \pi_{o_t}, \pi_{\mathcal{O}}, \beta_{o_t}] = a_U(s_t, o_t, a_t).$$

Lemma 2. *Under the precondition $o_t = o_{t-1}$ and given intra-option policies, π_o for all $o \in \mathcal{O}$, policy over options, $\pi_{\mathcal{O}}$, and terminations, β_o for all $o \in \mathcal{O}$, then:*

$$\mathbb{E}[\delta_t^{\mathcal{O}} | s_t, o_t, o_t{=}o_{t-1}, \pi_{o_t}, \pi_{\mathcal{O}}] = a_{\mathcal{O}}(s_t, o_{t-1}).$$

The proofs are in the appendix (supplementary materials). Using these lemmas and theorems we introduce algorithm 20 (INOC). We provide details on how we arrive at the updates to parameters η and φ in the appendix. The precondition $o_t = o_{t-1}$ might lead to fewer updates to the parameters of the terminations. The options evaluation part in the algorithm is the same as in previous work (Bacon, Harb, and Precup 2017).

Algorithm 1 Incremental Natural Option-Critic Algorithm (INOC)

```
$s_0 \leftarrow d_0$ and choose $o$ using $\pi_{\mathcal{O}}$.
while Not in terminal state do
    Select action $a_t$ as per $\pi_{o_t}$
    Take action $a_t$ observe $s_{t+1}, r_t$
    $e_\eta \leftarrow \lambda e_\eta + \frac{\partial \ln \pi_{o_t}(s_t, a_t, \theta)}{\partial \theta}$
    $\delta_t^U \leftarrow r_t + \gamma u(o_t, s_{t+1}) - q_{\pi_{\mathcal{O}}}(s_t, o_t)$
    $\texttt{temp} = \frac{\partial \ln \pi_{o_t}(s_t, a_t, \theta)}{\partial \theta}$
    $\eta \leftarrow \eta + \alpha_\eta \delta_t^U e_\eta - \alpha_\eta \texttt{temp} \times \texttt{temp}^T \times \eta$
    $\theta \leftarrow \theta + \alpha_\theta \frac{\eta}{||\eta||_2}$
    if $o_t$ is the same as $o_{t-1}$ then
        $e_\varphi \leftarrow \lambda e_\varphi + \frac{\partial \ln \beta_{o_{t-1}}(s_t, \vartheta)}{\partial \vartheta}$
        $\delta_t^{\mathcal{O}} \leftarrow r_t + \gamma v_{\pi_{\mathcal{O}}}(s_{t+1}) - \gamma v_{\pi_{\mathcal{O}}}(s_t)$
        $\texttt{temp} = \frac{\partial \ln \beta_{o_{t-1}}(s_t, \vartheta)}{\partial \vartheta}$
        $\varphi \leftarrow \varphi + \alpha_\varphi \beta_{o_{t-1}}(s_t, \vartheta) \delta_t^{\mathcal{O}} e_\varphi + \alpha_\varphi \texttt{temp} \times \texttt{temp}^T \times \varphi$
        $\vartheta \leftarrow \vartheta - \alpha_\vartheta \frac{\varphi}{||\varphi||_2}$
    end if
    if should terminate $o_t$ in $s_{t+1}$ according to $\beta_{o_t}$ then
        Choose $o_{t+1}$ (next option) according to $\pi_{\mathcal{O}}$ and reset $\eta, \varphi, e_\eta, e_\varphi$
    end if
end while
```

Experiments

We look at the performance of natural option critic in three different types of domains: a simple 2 state MDP, one with linear state representations and one with neural networks for state representations, and compare it to option critic. In all the cases we use sigmoid terminations and linear-softmax intra-option policies, as in previous work (Bacon, Harb, and Precup 2017).

MDP Setup: We design an MDP to demonstrate the uniform weighting of the components of the natural termination

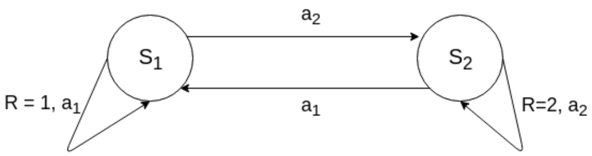


Figure 1: Simple deterministic MDP of two states and two actions

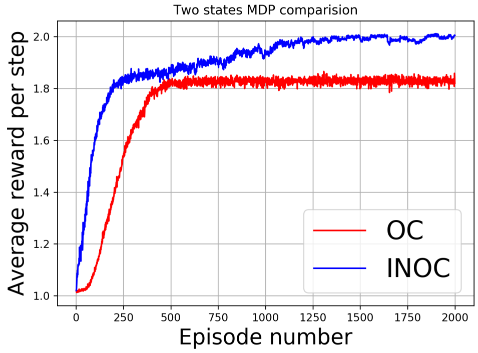


Figure 2: Average reward for INOC reaches the maxima while that of OC is stuck in a plateau. Results averaged over 200 runs of 2000 episodes.

gradient, $\widetilde{\nabla}_\theta q_{\pi_\mathcal{O}}(s_0, o_0)$, as opposed to using $\mu_\mathcal{O}(s, o)$. Note that the effectiveness of the natural policy gradient has been demonstrated sufficiently in past work (Kakade 2001; Bagnell and Schneider 2003; Degris, Pilarski, and Sutton 2012). We define a simple 2 state MDP as in Figure 1. The initial state distribution is $d_0(s_1) = 0.8$ and $d_0(s_2) = 0.2$. The transitions are deterministic. The reward for self loops into s_1 and s_2 are 1 and 2, respectively. The episode terminates after 30 steps. We use an ϵ-greedy policy over options, $\pi_\mathcal{O}$.

We consider a scenario with two options, o_1 and o_2, each of which has probability 0.9 for actions a_1 and a_2, respectively, regardless of the state. This gives us options as abstractions over individual actions. We initialize the terminations, β_o, and option value function, $q_{\pi_\mathcal{O}}(s, o)$ such that they are biased towards the greedy action, a_1, in state s_1 via the selection of option o_1. Specifically, we set $\beta_{o_1}(s_1) = 0.1$ and $\beta_{o_1}(s_2) = 0.1$, this way the setup is biased towards higher probability of $\mu_\mathcal{O}(s_1, o_1)$. This presents the agent with the challenge of learning the more optimal action of transitioning to state s_2, despite the higher probability $\mu(s_1, o_1)$ and the self loop reward of s_1. We set the learning rate for the intra-option policies, α_θ, to be negligible as our goal is to demonstrate the efficacy of the natural termination gradient.

As can be seen from Figure 2, the natural option critic converges to the optimal value, by overcoming the plateau, for average reward much faster than the option critic. The option critic is initially stuck in the greedy self-loop action, this is due to the weighting by $\mu_\mathcal{O}(s, o)$. Whereas the natural option critic begins learning early on and achieves the optimal average reward.

Four Rooms: The four rooms domain (Sutton, Precup, and Singh 1999) is a particularly favorable case for demonstrating the use of options. We use the same number of options, 4, as in previous work (Bacon, Harb, and Precup 2017). The result (Figure 3) indicates that natural option critic converges faster.

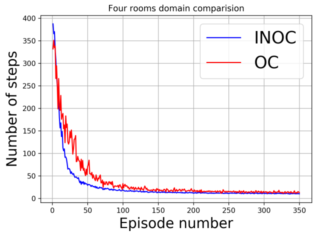


Figure 3: Four rooms with $\alpha_\theta = \alpha_\vartheta = 0.0025$, $\alpha_\eta = 0.5$, $\alpha_\varphi = 0.75$, $\lambda = 0.5$ and critic LR 0.5, averaged over 350 runs

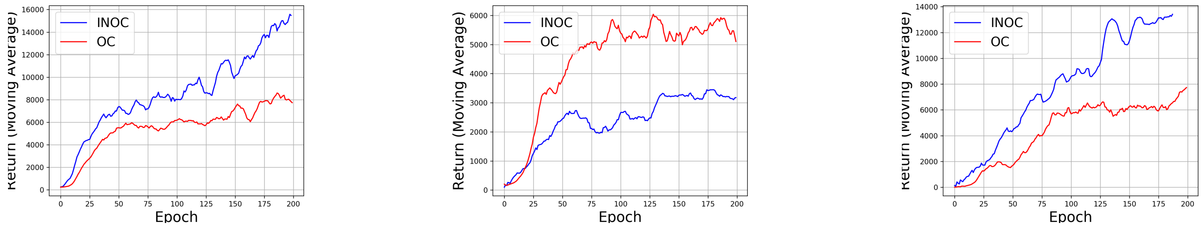
Arcade Learning Environment:

We compare natural option-critic with the option critic framework on the Arcade Learning Environment (Bellemare et al. 2013). To showcase the improvement over the option-critic architecture we use the same configuration for all the layers as in previous work (Bacon, Harb, and Precup 2017). Which in turn uses the same configuration for the first 3 convolutional layers of the network introduced by Mnih et al. (2013). The critic network was trained, similar to previous work (Bacon, Harb, and Precup 2017), using experience replay (Mnih et al. 2013) and RMSProp.

As in previous work (Bacon, Harb, and Precup 2017), we apply the regularizer prescribed by Mnih et al. (2016) to penalize low entropy policies. We use an on-policy estimate of the policy over options, $\pi_\mathcal{O}$, which is used in the computation of the natural gradient with respect to the termination parameters.

We compare the two approaches, option critic and natural option critic, by evaluating them for the games *Asterisk, Seaquest,* and *Zaxxon* (Bacon, Harb, and Precup 2017). For comparison we run training over same number of frames per epoch as done by Bacon, Harb, and Precup (2017), running the same number of trial and use the same number of options: 8. We demonstrate the results in Figure 4. More importantly, we use the same hyperparameters, for learning rates and entropy regularization, as in previous work to merit a fair comparison. We obtain improvements on the option-critic architecture (OC) for Asterisk and Zaxxon. We also note that we were unable to reproduce the results for Seaquest for option critic, but having given the same set of hyperparameters we observe that option critic performs better. We explain the issue with termination updates, and it's effect on the return, for Seaquest in the appendix.

For Zaxxon and Asterisk we see that NOC breaks the plateau much earlier than option critic. Note that the value network, for approximating Q_U, is learned using vanilla gradient.



(a) Asterisk

(b) Seaquest

(c) Zaxxon

Figure 4: Moving average of 10 returns for a single trial for Arcade learning Environment, with $\alpha_\theta = \alpha_\vartheta = 0.0025$, $\alpha_\eta = \alpha_\varphi = 0.75$, and $\lambda = 0.5$

Discussion

We have introduced a natural gradient based approach for learning intra-option policies and terminations, within the option-critic framework, which is linear in the number of parameters. More importantly, we have furnished instructive proofs on deriving the Fisher information matrix over path manifolds and corresponding function approximations based approach while reducing mean squared errors. We have also introduced an algorithm that uses consistent estimates of the advantage functions and learn the natural gradient by learning coefficients of the corresponding linear function approximators. The results showcase performance improvements on previous work. The proofs for finite horizon metrics are very similar to the ones provided by Bagnell and Schneider (2003). We also demonstrate the effectiveness of natural option critic in three distinct domains.

As discussed by Thomas (2014) we can obtain a truly unbiased estimate for our updates, but it may not be practical (Thomas 2014). The limitations that apply to the option-critic framework, except the use of vanilla gradient, apply. We use a block diagonal estimate of the Fisher information matrix. The complete Fisher information matrix for the option-critic framework over path manifolds is:

$$G_{\theta,\vartheta} = \begin{bmatrix} G_\theta & \left\langle \frac{\partial X}{\partial \theta} \frac{\partial X}{\partial \vartheta} \right\rangle \\ \left\langle \frac{\partial X}{\partial \vartheta} \frac{\partial X}{\partial \theta} \right\rangle & G_\vartheta \end{bmatrix},$$

where G_θ and G_ϑ are the Fisher information matrices for intra-option path manifold and state-option transition manifold, respectively. The random variable X is the path variable over state-option-action tuples. The computation of the complete Fisher information matrix suffers and its inverse is expensive and needs a compatible function approximation based approach to obtain a natural gradient estimate with space complexity linear in number of parameters.

Although our approach has added benefits it is limited by fewer updates of the termination policy. Work is required to develop better estimates of the advantage functions. More experimental work, e.g. applications to other domains, can further help understand the efficacy of natural gradients in the context of the option-critic framework.

References

Amari, S. 1985. Differential-geometrical methods in statistics. In *Lecture Notes in Statistics 28*. Springer-Verlag.

Amari, S.-I. 1998. Natural gradient works efficiently in learning. *Neural Comput.* 10(2):251–276.

Amari, S.-i. 2016. *Information Geometry and Its Applications*. Springer.

Bacon, P.-L.; Harb, J.; and Precup, D. 2017. The option-critic architecture. In *AAAI*.

Bagnell, J. A., and Schneider, J. 2003. Covariant policy search. IJCAI.

Bellemare, M. G.; Naddaf, Y.; Veness, J.; and Bowling, M. H. 2013. The arcade learning environment: An evaluation platform for general agents. *J. Artif. Intell. Res.* 47:253–279.

Bhatnagar, S.; Sutton, R. S.; Ghavamzadeh, M.; and Lee, M. 2007. Incremental natural actor-critic algorithms. In *Proceedings of the 20th International Conference on Neural Information Processing Systems*, NIPS'07, 105–112. USA: Curran Associates Inc.

Bhatnagar, S.; Sutton, R. S.; Ghavamzadeh, M.; and Lee, M. 2009. Natural actor-critic algorithms. *Automatica* 45(11):2471–2482.

Degris, T.; Pilarski, P. M.; and Sutton, R. S. 2012. Model-free reinforcement learning with continuous action in practice.

DeGroot, M. 1970. *Optimal Statistical Decisions*. Wiley Classics Library. Wiley.

Desjardins, G.; Simonyan, K.; Pascanu, R.; et al. 2015. Natural neural networks. In *Advances in Neural Information Processing Systems*, 2071–2079.

Dietterich, T. G. 2000. Hierarchical reinforcement learning with the maxq value function decomposition. *J. Artif. Intell. Res.(JAIR)* 13(1):227–303.

Kakade, S. 2001. A natural policy gradient. In Dietterich, T. G.; Becker, S.; and Ghahramani, Z., eds., *Advances in Neural Information Processing Systems 14 (NIPS 2001)*, 1531–1538. MIT Press.

Konda, V. R., and Tsitsiklis, J. N. 2000. Actor-critic algorithms. NIPS'2000, 1008–1014.

Konidaris, G., and Barto, A. G. 2009. Skill discovery in continuous reinforcement learning domains using skill chaining. In Bengio, Y.; Schuurmans, D.; Lafferty, J. D.; Williams, C. K. I.; and Culotta, A., eds., *Advances in Neural Information Processing Systems 22*. Curran Associates, Inc. 1015–1023.

Kurita, T. 1992. Iterative weighted least squares algorithms for neural networks classifiers. *New Generation Computing* 12:375–394.

Machado, M. C.; Bellemare, M. G.; and Bowling, M. 2017. A Laplacian framework for option discovery in reinforcement learning. In Precup, D., and Teh, Y. W., eds., *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, 2295–2304. International Convention Centre, Sydney, Australia: PMLR.

Martens, J., and Grosse, R. B. 2015. Optimizing neural networks with kronecker-factored approximate curvature. In *ICML*.

Martens, J. 2010. Deep learning via hessian-free optimization. In *Proceedings of the 27th International Conference on International Conference on Machine Learning*, ICML'10, 735–742. USA: Omnipress.

Mnih, V.; Kavukcuoglu, K.; Silver, D.; Graves, A.; Antonoglou, I.; Wierstra, D.; and Riedmiller, M. A. 2013. Playing atari with deep reinforcement learning. *CoRR* abs/1312.5602.

Mnih, V.; Badia, A. P.; Mirza, M.; Graves, A.; Lillicrap, T. P.; Harley, T.; Silver, D.; and Kavukcuoglu, K. 2016. Asynchronous methods for deep reinforcement learning. In *ICML*.

Morimura, T.; Uchibe, E.; and Kenji, D. 2005. Utilizing the natural gradient in temporal difference reinforcement learning with eligibility traces. 0–0.

Parr, R., and Russell, S. J. 1998. Reinforcement learning with hierarchies of machines. In *Advances in neural information processing systems*, 1043–1049.

Pascanu, R., and Bengio, Y. 2013. Revisiting natural gradient for deep networks.

Peters, J., and Schaal, S. 2008. Natural actor-critic. *Neurocomputing* 71:1180–1190.

Rao, C. R. 1945. Information and accuracy attainable in the estimation of statistical parameters. In *Bulletin of the Calcutta Mathematical Society*. 81–91.

Roux, N. L.; Manzagol, P.; and Bengio, Y. 2008. Topmoumoute online natural gradient algorithm. In Platt, J. C.; Koller, D.; Singer, Y.; and Roweis, S. T., eds., *Advances in Neural Information Processing Systems 20*. Curran Associates, Inc. 849–856.

Simsek, Ö., and Barto, A. G. 2008. Skill characterization based on betweenness. In *NIPS*.

Sun, K., and Nielsen, F. 2017. Relative fisher information and natural gradient for learning large modular models. In *ICML*.

Sutton, R. S.; McAllester, D.; Singh, S.; and Mansour, Y. 1999. Policy gradient methods for reinforcement learning with function approximation. In *Proceedings of the 12th International Conference on Neural Information Processing Systems*, NIPS'99, 1057–1063.

Sutton, R. S.; Precup, D.; and Singh, S. P. 1999. Between mdps and semi-mdps: A framework for temporal abstraction in reinforcement learning. *Artif. Intell.* 112:181–211.

Thomas, P. S., and Brunskill, E. 2017. Policy gradient methods for reinforcement learning with function approximation and action-dependent baselines. *CoRR* abs/1706.06643.

Thomas, P.; Silva, B. C.; Dann, C.; and Brunskill, E. 2016. Energetic natural gradient descent. In Balcan, M. F., and Weinberger, K. Q., eds., *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, 2887–2895. New York, New York, USA: PMLR.

Thomas, P. S.; Dann, C.; and Brunskill, E. 2018. Decoupling learning rules from representations. In *ICML*.

Thomas, P. S. 2011. Policy gradient coagent networks. In Shawe-Taylor, J.; Zemel, R. S.; Bartlett, P. L.; Pereira, F.; and Weinberger, K. Q., eds., *Advances in Neural Information Processing Systems 24*. Curran Associates, Inc. 1944–1952.

Thomas, P. 2014. Bias in natural actor-critic algorithms. In *ICML*.

Thrun, S., and Schwartz, A. 1995. Finding structure in reinforcement learning. In Tesauro, G.; Touretzky, D. S.; and Leen, T. K., eds., *Advances in Neural Information Processing Systems 7*. MIT Press. 385–392.