Fourier Policy Gradients

Matthew Fellows [* 1]   Kamil Ciosek [* 1]   Shimon Whiteson [1]

Abstract

We propose a new way of deriving policy gradient updates for reinforcement learning. Our technique, based on Fourier analysis, recasts integrals that arise with *expected policy gradients* as convolutions and turns them into multiplications. The obtained analytical solutions allow us to capture the low variance benefits of EPG in a broad range of settings. For the critic, we treat trigonometric and radial basis functions, two function families with the universal approximation property. The choice of policy can be almost arbitrary, including mixtures or hybrid continuous-discrete probability distributions. Moreover, we derive a general family of sample-based estimators for stochastic policy gradients, which unifies existing results on sample-based approximation. We believe that this technique has the potential to shape the next generation of policy gradient approaches, powered by analytical results.

1. Introduction

Policy gradient methods, also known as actor-critic methods, are an effective way to perform reinforcement learning in large or continuous action spaces (Lillicrap et al., 2015; Schulman et al., 2015; 2017; Wu et al., 2017; Peters & Schaal, 2008; Sutton et al., 1999; Williams, 1992). Since they adjust the policy in small increments, they do not have to perform expensive optimisations over the action space, in contrast to methods based on value functions, such as Q-learning (Mnih et al., 2015; Silver et al., 2017; van Hasselt et al., 2015) or SARSA (van Seijen et al., 2009; Sutton & Barto, 1998; Sutton, 1996). Moreover, they are naturally suited to stochastic policies, which are useful for exploration and necessary to achieve optimality in some settings, e.g., competitive multi-agent systems.

Until recently, policy gradient methods were either restricted to deterministic policies (Silver et al., 2014) or suffered from high variance (Sutton et al., 1999). The latter problem is exacerbated in large state spaces, when the number of samples required to reduce the variance of the gradient estimate becomes infeasible for the simple score function estimators on which policy gradient methods typically rely. The problem also arises when training recurrent neural networks (RNNs) (Xu et al., 2015; Ba et al., 2015; Sukhbaatar et al., 2016) that have to be unrolled over several timesteps, each adding to the overall variance, and in multi-agent settings (Foerster et al., 2017), where the actions of other agents introduce a compounding source of variance.

Recently, a new approach called *expected policy gradients* (EPG) (Ciosek & Whiteson, 2018a;b) was proposed that eliminates the variance of a stochastic policy gradient by integrating over the actions analytically. However, this requires analytic solutions to the policy gradient integral and the original work addressed only polynomial critics.

In this paper, we employ techniques from Fourier analysis to derive analytic policy gradient updates for two important families of critics. The first, *radial basis functions* (RBFs), combines the benefits of shallow structure, which makes them tractable, with an impressive empirical track record (Buhmann, 2003; Carr et al., 2001). The second, trigonometric critics, is useful for modelling periodic phenomena. Similarly to polynomial critics (Ciosek & Whiteson, 2018b), these function classes are universal, i.e., they can approximate an arbitrary function on a bounded interval.

Furthermore, to address cases where analytical solutions are infeasible, we provide a general theorem for deriving Monte Carlo estimators that links existing methods using the first and second derivatives of the action-value function, relating it to existing sampling approaches.

Our technique also enables analytic solutions for new families of policies, extending EPG to any policy that has unbounded support, where it previously required the policy to be in an exponential family. We also develop results for mixture policies and hybrid discrete-continuous policies, which we posit can be useful in multi-agent settings, where having a rich class of policies is important not just for exploration but also for optimality (Nisan et al., 2007).

*Equal contribution [1]Department of Computer Science, University of Oxford, United Kingdom. Correspondence to: Matthew Fellows <matthew.fellows@cs.ox.ac.uk>.

Overall, we believe that the techniques developed in this paper can be used to shape the next generation of policy gradient methods suitable for any reasonable MDP and that, powered by analytical results, achieve zero or low variance. Moreover, our methods elucidate the way policy gradients work by explicitly stating the expected update. Finally, while the main contribution of this paper is theoretical, we also provide an empirical evaluation using a periodic critic on a simple turntable problem that demonstrates the practical benefit of using a trigonometric critic.

2. Background

Reinforcement learning (RL) aims to learn optimal behaviour policies for an agent (or many agents) acting in an environment with a scalar reward signal. Formally, we consider a Markov decision process, defined as a tuple $(S, A, R, p, p_0, \gamma)$. An agent has an environmental state $s \in S = \mathbb{R}^n$; takes a sequence of actions $a_1, a_2, ...$, where $a_t \in A$; transitions to the next state $s' \sim p(\cdot|s, a)$ under the state transition distribution $p(s'|s, a)$; and receives a scalar reward $r \in \mathbb{R}$. The agent's initial state s_0 is distributed as $s_0 \sim p_0(\cdot)$.

The agent samples from the policy β to generate actions $a \sim \beta(\cdot|s)$, giving a trajectory through the environment $\tau = (s_0, a_0, r_1, s_1, a_1, r_1, ...)$. The definition of the value function is $V^\beta(s) = \mathbb{E}_{\tau:s_0=s}[\sum_t \gamma^t r_t]$ and action-value function is $Q^\beta(s, a) = \mathbb{E}_{\tau:s_0=s,a_0=a}[\sum_t \gamma^t r_t]$, where $\gamma \in [0, 1)$ is a discount factor. An optimal policy $\beta^\star$ maximises the total return $J = \int_s V^{\beta^\star}(s)dp_o(s)$.

2.1. Policy Gradient Methods

Policy gradient methods seek a locally optimal policy by maintaining a *critic*, learned using a value-based method, and an *actor*, adjusted using a policy gradient update.

The critic $\hat{Q}$ is learned using variants of SARSA (van Seijen et al., 2009; Sutton & Barto, 1998; Sutton, 1996), with the goal of approximating the true action-value Q^β. Meanwhile, the actor adjusts the policy parameter vector θ of the policy β_θ with the aim of maximising J. For stochastic policies, this is done by following the gradient:

$$\nabla_\theta J = \int_s \rho(s) \underbrace{\int_a Q(s, a)\nabla_\theta \beta_\Theta(a|s)da}_{I_\theta(s)} ds, \quad (1)$$

where $\rho(s) \triangleq \sum_{t=0}^\infty \gamma^t p(s_t = s|s_0)$ is the discounted-ergodic occupancy measure. The outer integral can be approximated by following a trajectory of length T through the environment, yielding:

$$\nabla_\theta J = \mathbb{E}_{\rho(s)}[I_\theta(s)] \approx \sum_{t=0}^{T-1} \gamma^t \hat{I}_\theta(s_t), \quad (2)$$

Algorithm 1 Expected Policy Gradient

1: $s \leftarrow s_0, t \leftarrow 0$
2: initialise optimiser, initialise policy parameters θ
3: **while** not converged **do**
4:     $g_t \leftarrow \gamma^t \hat{I}_\theta(s)$
5:     $\theta \leftarrow \theta + \text{optimiser.UPDATE}(g_t)$
6:     $a \sim \beta(\cdot, s)$
7:     $s', r \leftarrow \text{environment.PERFORM-ACTION(a)}$
8:     $\hat{Q}.\text{UPDATE}(s, a, r, s')$
9:     $t \leftarrow t + 1$
10:    $s \leftarrow s'$
11: **end while**

where $\hat{I}_\theta$ is the integral of (1) but with the critic $\hat{Q}$ in place of the unknown true Q-function:

$$\hat{I}_\theta(s_t) = \int_a \hat{Q}(s_t, a)\nabla_\theta \beta_\Theta(a|s_t)da. \quad (3)$$

The subscript of $\hat{I}_\theta$ denotes the fact that we are differentiating with respect to θ. Now, since $\hat{Q}$, unlike Q, does *not* depend on the policy parameters, we can move the differentiation out of the inner integral as follows:

$$\hat{I}_\theta(s_t) = \nabla_\theta \underbrace{\int_a \hat{Q}(s_t, a)d\beta_\Theta(a|s_t)}_{E(s_t)=\mathbb{E}_\beta[\hat{Q}(s_t,\cdot)]}. \quad (4)$$

This transformation has two benefits: it allows for easier manipulation of the integral and it also holds for deterministic policies, where β is a Dirac-delta measure (Silver et al., 2014; Ciosek & Whiteson, 2018a;b).

Using (2) directly with an analytic value of $\hat{I}(s_t)$ yields *expected policy gradients* (EPG)[1] (Ciosek & Whiteson, 2018a;b), shown in Algorithm 1. If instead we add an additional Monte Carlo sampling step:

$$\hat{I}_\theta \approx \hat{Q}(s, a)\nabla_\theta \log \beta(a|s), \quad (5)$$

we get the original *stochastic policy gradients* (Williams, 1992; Sutton et al., 1999). In place of (5), alternative Monte Carlo schemes with better variance properties have also been proposed (Baxter & Bartlett, 2000; Baxter et al., 2001; Baxter & Bartlett, 2001; Gu et al., 2016; Kakade et al., 2003). If we can compute the integral in (3), then EPG is preferable since it avoids the variance introduced by the Monte Carlo step of (5).

This paper considers both methods for solving integrals of the form in (3) and Monte Carlo methods that improve on (5) for cases where analytical solutions are not possible.

---

[1]Called *all-action policy gradient* in an unpublished draft by Sutton et al. (2000).

Above, we used the symbol θ to denote a generic policy parameter. Often, the policy is described by its moments (for instance a Gaussian is fully defined by its mean and covariance). To achieve greater flexibility, these immediate parameters are obtained by a complex function approximator, such as a neural network, where the state vector is the input, and parameterised by $\boldsymbol{w}$. The total policy gradient for $\boldsymbol{w}$ is then obtained by using the chain rule. For example, for a Gaussian we have immediate parameters $\boldsymbol{\mu}, \boldsymbol{\Sigma}$ and the parameterisation is:

$$\beta(\boldsymbol{a}|\boldsymbol{s}) = \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}),$$
$$(\boldsymbol{\mu}, \boldsymbol{\Sigma}^{1/2}) = \text{net}_{\boldsymbol{w}}(\boldsymbol{s}),$$

where $\text{net}_{\boldsymbol{w}}$ is a neural network parameterised by the vector $\boldsymbol{w}$. The gradient for some $\boldsymbol{w}$ is then:

$$\nabla_{\boldsymbol{w}} I_\theta(\boldsymbol{s}) = \nabla_{\boldsymbol{w}} \boldsymbol{\mu} I_{\boldsymbol{\mu}} + \nabla_{\boldsymbol{w}} \boldsymbol{\Sigma}^{1/2} I_{\boldsymbol{\Sigma}^{1/2}}.$$

For clarity, we only give updates for the immediate parameters (in this case, $\hat{I}_{\boldsymbol{\mu}}$ and $\hat{I}_{\boldsymbol{\Sigma}^{1/2}}$) in the remainder of the paper, without explicitly mentioning $\boldsymbol{w}$.

2.2. Fourier Analysis

A *convolution* $f * g$ is an operation on two functions that returns another function, defined as:

$$(f * g)(\boldsymbol{x}) \triangleq \int_{x'} f(\boldsymbol{x}')g(\boldsymbol{x} - \boldsymbol{x}')d\boldsymbol{x}'.$$

Convolutions have convenient analytical properties that we use to derive our main result. To make convolutions easy to compute, we seek a transform $\mathcal{F}$ that, when applied to a convolution, yields a simple operation like multiplication, i.e., we want the property:

$$\mathcal{F}(f * g)(\boldsymbol{\omega}) = \Big(\mathcal{F}(f)\mathcal{F}(g)\Big)(\boldsymbol{\omega}). \tag{6}$$

We also need the dual property:

$$\mathcal{F}(fg) = \mathcal{F}(f) * \mathcal{F}(g),$$

to ensure symmetry between the space of functions and their transforms. It turns out that, up to scaling, there is only one transform that meets our needs (Alesker et al., 2008), the *Fourier transform*:

$$\mathcal{F}(f)(\boldsymbol{\omega}) \triangleq \int_x f(\boldsymbol{x})e^{-i\boldsymbol{\omega}^\top \boldsymbol{x}}d\boldsymbol{x}.$$

The two sets of parentheses on the lefthand side are required because the Fourier transform $\mathcal{F}(f)$ is a function, not a scalar, and $\mathcal{F}(f)(\boldsymbol{\omega})$ is the result of evaluating this function on $\boldsymbol{\omega}$. The Fourier transform of a probability density function is known as the *characteristic function*

of the corresponding distribution. An intuitive interpretation of the Fourier transforms is that it provides a mapping from the action-spatial domain to the frequency domain, $\mathcal{F}(f(\boldsymbol{x})) : \boldsymbol{x} \to \boldsymbol{\omega}$, decomposing the function f into its frequency components. Consider, as a simple example, a univariate sinusoidal function, $f(x) = \cos(x\Omega)$. The Fourier transform of f can be easily shown to be $\mathcal{F}(f) = \pi\delta(\omega - \Omega) + \pi\delta(\omega + \Omega)$ (Stein & Shakarchi, 2003); the Fourier transform has mapped a sinusoid of frequency Ω in the action domain to a double frequency spike at $\pm\Omega$ in the frequency domain.

The Fourier transform has another related intuitive interpretation as a change of basis in the space of functions. The Fourier basis functions $e^{-i\boldsymbol{\omega}^\top \boldsymbol{x}}$ make analytical operations convenient in much the same way as a choice of convenient basis in linear algebra makes certain matrix operations easier. Since the basis functions are periodic, the Fourier transform can also be viewed as a decomposition of the original function into cycles. Sometimes these cycles are written explicitly when the complex exponential $e^{-i\boldsymbol{\omega}^\top \boldsymbol{x}}$ is expressed in polar form, which includes sines and cosines. Indeed, the *Fourier series*, which we briefly discuss in Appendix A, can be used to prove that any function on a bounded interval can be approximated arbitrarily well with a sum of sufficiently many such trigonometric terms.

The *inverse Fourier transform* is defined as:

$$\mathcal{F}^{-1}(g)(\boldsymbol{x}) \triangleq \frac{1}{(2\pi)^n} \int_\omega g(\boldsymbol{\omega})e^{i\boldsymbol{\omega}^\top \boldsymbol{x}}d\boldsymbol{\omega},$$

which has the property that, for any function f,

$$\mathcal{F}^{-1}(\mathcal{F}(f)) = f. \tag{7}$$

Thus, we can recover the original function by applying the Fourier and inverse Fourier transforms. Just as the Fourier transform maps from the action domain to the frequency domain, the inverse Fourier transforms provides a mapping from the frequency domain back to the action-spatial domain $\mathcal{F}^{-1}(f(\boldsymbol{\omega})) : \boldsymbol{\omega} \to \boldsymbol{x}$. The Fourier transform also turns differentiation into multiplication:

$$i\boldsymbol{\omega}\mathcal{F}(f) = \mathcal{F}(\nabla_{\boldsymbol{x}} f(\boldsymbol{x})), \tag{8}$$
$$(i\boldsymbol{\omega})(i\boldsymbol{\omega})^\top \mathcal{F}(f) = \mathcal{F}\left(\nabla_{\boldsymbol{x}}^{(2)} f(\boldsymbol{x})\right), \tag{9}$$

where $\nabla_{\boldsymbol{x}}^{(m)} f$ denotes the mth order derivative of f w.r.t. $\boldsymbol{x} \ \forall \ m \geq 0$.

We formalise the n-dimensional Fourier transform in Appendix B, and provide definitions for Fourier transforms of matrix and vector quantities. We also derive the differentiation/multiplication property in n-dimensional space.

3. Main Result

In this section, we prove our main result. The motivating factor behind these derivations is that by viewing the inner integral $\hat{I}_\theta$ as a *convolution*, we can analyse our policy gradient in the frequency domain. This affords powerful analytical results that enable us to exploit the multiplication/derivative property of Fourier transforms, namely manipulation of expressions involving the derivatives of our critic $\hat{Q}$ in the action-spatial domain are represented simply by factors of $(i\boldsymbol{\omega})$ in the frequency domain. We apply this elegant property in Section 4.1 to demonstrate the relative ease of manipulation of the inner integral $\hat{I}_\theta$. In Section 4.2, we show that existing Monte Carlo policy gradient estimators arise from our theorem as a single family of cases using different factors $(i\boldsymbol{\omega})$ multiplied with the critic $\hat{Q}$.

Moreover, our theorems rely only on the characteristic function $\mathcal{F}(\tilde{\beta})$. While the original technique developed for EPG (Ciosek & Whiteson, 2018b) relies on the moment generating function to obtain $\hat{I}$ for a policy from an exponential family and a polynomial family of critics, we require only that both the policy PDF and the critic have a closed form Fourier transform (Karr, 1993). For policies, this condition is easy to satisfy since almost all common distributions have a closed form characteristic function.

Theorem 1 (Fourier Policy Gradients). *Let $\hat{I}_\theta(\boldsymbol{s}_t) = \nabla_\theta \int_a \hat{Q}(\boldsymbol{s}_t, \boldsymbol{a})\beta_\theta(\boldsymbol{a}|\boldsymbol{s}_t)d\boldsymbol{a}$ be the inner integral of the policy gradient for a critic $\hat{Q}(\boldsymbol{a})$ and policy $\beta(\boldsymbol{a})$ with auxiliary policy $\tilde{\beta}(\boldsymbol{\mu} - \boldsymbol{a}) = \beta(\boldsymbol{a})$. We may write $\hat{I}_\theta(\boldsymbol{s}_t)$ as:*

$$\hat{I}_\theta(\boldsymbol{s}_t) = \nabla_\theta \mathcal{F}^{-1}\Big(\mathcal{F}(\hat{Q})\mathcal{F}(\tilde{\beta})\Big)(\boldsymbol{\mu}). \quad (10)$$

Proof. Recall the definition of $\hat{I}_\theta(\boldsymbol{s}_t)$ from (4):

$$\hat{I}_\theta(\boldsymbol{s}_t) = \nabla_\theta \int_a \hat{Q}(\boldsymbol{s}_t, \boldsymbol{a})\beta d\boldsymbol{a}.$$

To exploit the convolution property of Fourier transforms given by (6), the first step is to introduce an auxiliary policy $\tilde{\beta}$, so that the above integral becomes a convolution. If the mean of the policy β is $\boldsymbol{\mu}$, the new auxiliary policy $\tilde{\beta}$ is:

$$\tilde{\beta}(\boldsymbol{\mu} - \boldsymbol{a}) = \beta(\boldsymbol{a}).$$

We start by rewriting $\hat{I}_\theta$ as:

$$\hat{I}_\theta = \nabla_\theta \int_a \hat{Q}(\boldsymbol{a})\tilde{\beta}(\boldsymbol{\mu} - \boldsymbol{a})d\boldsymbol{a} = \nabla_\theta(\hat{Q} * \tilde{\beta})(\boldsymbol{\mu}). \quad (11)$$

Now, we apply the Fourier transform to the convolution and use (6) to reduce it to a multiplication:

$$\mathcal{F}(\hat{Q} * \tilde{\beta})(\boldsymbol{\omega}) = \mathcal{F}\bigg(\int_a \hat{Q}(\boldsymbol{a})\tilde{\beta}(\boldsymbol{\mu} - \boldsymbol{a}))d\boldsymbol{a}\bigg)(\boldsymbol{\omega}),$$
$$= \Big(\mathcal{F}(\hat{Q})\mathcal{F}(\tilde{\beta})\Big)(\boldsymbol{\omega}).$$

Taking the inverse Fourier transform gives:

$$(\hat{Q} * \tilde{\beta})(\boldsymbol{\mu}) = \mathcal{F}^{-1}\Big(\mathcal{F}(\hat{Q})\mathcal{F}(\tilde{\beta})\Big)(\boldsymbol{\mu}).$$

Substituting this into (11) yields our main result:

$$\hat{I}_\theta(\boldsymbol{s}) = \nabla_\theta \mathcal{F}^{-1}\Big(\mathcal{F}(\hat{Q})\mathcal{F}(\tilde{\beta})\Big)(\boldsymbol{\mu}).$$

$\square$

We now derive a variant of our main theorem for the special case of $\boldsymbol{\mu}$.

Theorem 2 (Fourier Policy Gradients for $\boldsymbol{\mu}$). *Let $\hat{I}_{\boldsymbol{\mu}}(\boldsymbol{s}_t) = \nabla_{\boldsymbol{\mu}} \int_a \hat{Q}(\boldsymbol{s}_t, \boldsymbol{a})\beta_\theta(\boldsymbol{a}|\boldsymbol{s}_t)d\boldsymbol{a}$ be the inner integral of the policy gradient for $\boldsymbol{\mu}$ with a critic $\hat{Q}(\boldsymbol{a})$ and policy $\beta(\boldsymbol{a})$ with auxiliary policy $\tilde{\beta}(\boldsymbol{\mu} - \boldsymbol{a}) = \beta(\boldsymbol{a})$. We may write $\hat{I}_{\boldsymbol{\mu}}(\boldsymbol{s}_t)$ as:*

$$\hat{I}_{\boldsymbol{\mu}}(\boldsymbol{s}_t) = \mathcal{F}^{-1}\Big(\mathcal{F}(\hat{Q})i\boldsymbol{\omega}\mathcal{F}(\nabla\tilde{\beta})\Big)(\boldsymbol{\mu}). \quad (12)$$

Proof. We return to (11), retaining the derivative inside the integral:

$$\hat{I}_{\boldsymbol{\mu}}(\boldsymbol{s}) = \int_a \hat{Q}(\boldsymbol{a})\nabla_{\boldsymbol{\mu}}\tilde{\beta}(\boldsymbol{\mu} - \boldsymbol{a})d\boldsymbol{a}.$$

From the chain rule, we substitute $\nabla_{\boldsymbol{\mu}}\Big(\tilde{\beta}(\boldsymbol{\mu} - \boldsymbol{a})\Big) = \Big(\nabla\tilde{\beta}\Big)(\boldsymbol{\mu} - \boldsymbol{a})$, yielding:

$$\hat{I}_{\boldsymbol{\mu}}(\boldsymbol{s}) = \int_a \hat{Q}(\boldsymbol{a})\Big(\nabla\tilde{\beta}\Big)(\boldsymbol{\mu} - \boldsymbol{a})d\boldsymbol{a} = (\hat{Q} * \nabla\tilde{B})(\boldsymbol{\mu}). \quad (13)$$

Now, we take the Fourier transforms of the convolution $(\hat{Q} * \nabla\tilde{B})(\boldsymbol{\mu})$ and exploit the multiplication property of (6):

$$\mathcal{F}((\hat{Q} * \nabla\tilde{B}))(\boldsymbol{\omega}) = \Big(\mathcal{F}(\hat{Q})\mathcal{F}(\nabla\tilde{\beta})\Big)(\boldsymbol{\omega}).$$

Using the multiplication/derivative property from (8), we substitute for $\mathcal{F}(\nabla\tilde{\beta}) = i\boldsymbol{\omega}\mathcal{F}(\tilde{\beta})$:

$$\mathcal{F}((\hat{Q} * \nabla\tilde{B}))(\boldsymbol{\omega}) = \Big(\mathcal{F}(\hat{Q})i\boldsymbol{\omega}\mathcal{F}(\nabla\tilde{\beta})\Big)(\boldsymbol{\omega}).$$

Finally, taking inverse Fourier transforms and substituting into (13) yields our result:

$$\hat{I}_\theta(\boldsymbol{s}) = \mathcal{F}^{-1}\Big(\mathcal{F}(\hat{Q})\mathcal{F}(\nabla\tilde{\beta})\Big)(\boldsymbol{\mu}).$$

$\square$

We use Theorem 2 to derive the following corollary, valid for all parameters ψ s.t. $\boldsymbol{\mu}$ does not depend upon them.

Corollary 2.1. *Let ψ be a parameter that does not depend upon $\boldsymbol{\mu}$. We can write $\hat{I}_\psi(\boldsymbol{s}_t) = \nabla_\psi \int_a \hat{Q}(\boldsymbol{s}_t, \boldsymbol{a})\beta_\theta(\boldsymbol{a}|\boldsymbol{s}_t)d\boldsymbol{a}$ as:*

$$\hat{I}_\psi(\boldsymbol{s}) = \mathcal{F}^{-1}\left(\mathcal{F}(\hat{Q})\nabla_\psi\mathcal{F}(\tilde{\beta})\right)(\boldsymbol{\mu}). \quad (14)$$

The required auxiliary policy $\tilde{\beta}(\boldsymbol{a}) = \beta(\boldsymbol{\mu} - \boldsymbol{a})$ exists for all distributions β with unbounded support. For symmetric distributions, $\tilde{\beta}$ often has a convenient form, e.g., for a Gaussian policy $\beta = \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, $\tilde{\beta} = \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma})$. This transformation is similar to reparameterisation (Heess et al., 2015). For critics, we discuss tractable critic families in the remainder of the paper.

4. Applications

We now discuss a number of specialisations of (10) and (12), linking them to several established policy gradient approaches.

4.1. Frequency Domain Analysis

We now motivate the remainder of this section by considering the Gaussian policy $\tilde{\beta} = \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma})$. We need to calculate the gradient w.r.t. $\boldsymbol{\Sigma}^{\frac{1}{2}}$ where $(\boldsymbol{\Sigma}^{\frac{1}{2}})^\top\boldsymbol{\Sigma}^{\frac{1}{2}} = \boldsymbol{\Sigma}$. From the characteristic function for a multivariate Gaussian, $\mathcal{F}(\mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma})) = e^{-\frac{1}{2}\boldsymbol{\omega}^\top\boldsymbol{\Sigma}\boldsymbol{\omega}}$, we find derivatives as:

$$\nabla_{\boldsymbol{\Sigma}^{\frac{1}{2}}}\mathcal{F}(\tilde{\beta}) = \nabla_{\boldsymbol{\Sigma}^{\frac{1}{2}}}e^{-\frac{1}{2}\boldsymbol{\omega}^\top\boldsymbol{\Sigma}\boldsymbol{\omega}},$$
$$= -\frac{1}{2}\boldsymbol{\Sigma}^{\frac{1}{2}}\left(2\boldsymbol{\omega}\boldsymbol{\omega}^\top e^{-\frac{1}{2}\boldsymbol{\omega}^\top\boldsymbol{\Sigma}\boldsymbol{\omega}}\right),$$
$$= \boldsymbol{\Sigma}^{\frac{1}{2}}(i\boldsymbol{\omega})(i\boldsymbol{\omega})^\top\mathcal{F}(\tilde{\beta}).$$

Substituting for $\nabla_{\psi=\boldsymbol{\Sigma}^{\frac{1}{2}}}\mathcal{F}(\tilde{\beta})$ in (14) gives the gradient for $\boldsymbol{\Sigma}^{\frac{1}{2}}$. For completeness, we also include the update for $\boldsymbol{\mu}$ which, recall from (12), is the same for all policies with auxiliary function $\tilde{\beta}(\boldsymbol{a}) = \beta(\boldsymbol{\mu} - \boldsymbol{a})$.

$$\hat{I}_{\boldsymbol{\mu}} = \mathcal{F}^{-1}\left(\mathcal{F}(\hat{Q})(i\boldsymbol{\omega})\mathcal{F}(\tilde{\beta})\right)(\boldsymbol{\mu}), \quad (15)$$

$$\hat{I}_{\boldsymbol{\Sigma}^{\frac{1}{2}}} = \mathcal{F}^{-1}\left(\boldsymbol{\Sigma}^{\frac{1}{2}}\mathcal{F}(\hat{Q})(i\boldsymbol{\omega})(i\boldsymbol{\omega})^\top\mathcal{F}(\tilde{\beta})\right). \quad (16)$$

We see from (8) and (9) that the terms $i\boldsymbol{\omega}$, once pulled into the Fourier transform, become differentiation operators. However, (15) and (16) afford us a choice – we can pull them into the critic term or the policy term. This gives rise to a number of different expressions for the gradient. To differentiate between methods, we define the order of the method, denoted by M, the order of the derivative with respect to the critic.

We continue our example of Gaussian policies, using (15) to compute an update for $\boldsymbol{\mu}$ for $M \in \{0, 1\}$ and (16) to compute an update for $\Sigma^{1/2}$ for $M \in \{0, 1, 2\}$. Full derivations with Gaussian derivatives can be found in Appendix E.

Zeroth Order Method ($M = 0$) Using (15) and (16) in their current form gives an analytic expression for a zeroth order critic, as we do not multiply $\mathcal{F}(\hat{Q})$ by any factor of $i\boldsymbol{\omega}$. Using results for multidimensional Fourier transforms from (8) and (9) when taking these inverse transforms, we obtain:

$$\hat{I}_{\boldsymbol{\mu}} = \mathcal{F}^{-1}(\mathcal{F}(\hat{Q})\mathcal{F}(\nabla\tilde{\beta})) = \int_a \hat{Q}\nabla\tilde{\beta}d\boldsymbol{a}$$
$$= -\int_a \hat{Q}\nabla\beta d\boldsymbol{a}, \quad (17)$$

$$\hat{I}_{\boldsymbol{\Sigma}^{\frac{1}{2}}} = \mathcal{F}^{-1}(\boldsymbol{\Sigma}^{\frac{1}{2}}\mathcal{F}(\hat{Q})\mathcal{F}(\nabla^{(2)}\tilde{\beta})) = \boldsymbol{\Sigma}^{\frac{1}{2}}\int_a \hat{Q}\nabla^{(2)}\tilde{\beta}d\boldsymbol{a}$$
$$= \boldsymbol{\Sigma}^{\frac{1}{2}}\int_a \hat{Q}\nabla^{(2)}\beta d\boldsymbol{a}. \quad (18)$$

Here, we use the identities $\nabla\tilde{\beta} = -\nabla\beta$ and $\nabla^{(2)}\tilde{\beta} = \nabla^{(2)}\beta$ from Lemma 3.

First Order Method ($M = 1$) To obtain an analytic expression in terms of $\nabla_{\boldsymbol{a}}\hat{Q}$, we must manipulate the factors of $(i\boldsymbol{\omega})$ in (15) and (16) to obtain a factor of $(i\boldsymbol{\omega})\hat{Q}$. We then exploit the multidimensional Fourier transform result for vectors from (8) as before:

$$\hat{I}_{\boldsymbol{\mu}} = \mathcal{F}^{-1}\left((i\boldsymbol{\omega})\mathcal{F}(\hat{Q})\mathcal{F}(\tilde{\beta})\right) = \mathcal{F}^{-1}\left(\mathcal{F}(\nabla\hat{Q})\mathcal{F}(\tilde{\beta})\right)$$
$$= \int_a \beta\nabla\hat{Q}d\boldsymbol{a}, \quad (19)$$

$$\hat{I}_{\boldsymbol{\Sigma}^{\frac{1}{2}}} = \mathcal{F}^{-1}\left(\boldsymbol{\Sigma}^{\frac{1}{2}}(i\boldsymbol{\omega})\mathcal{F}(\tilde{\beta})(i\boldsymbol{\omega})^\top\mathcal{F}(\hat{Q})\right),$$
$$= \mathcal{F}^{-1}\left(\boldsymbol{\Sigma}^{\frac{1}{2}}\mathcal{F}(\nabla\tilde{\beta})\mathcal{F}(\nabla Q)^\top\right),$$
$$= -\boldsymbol{\Sigma}^{\frac{1}{2}}\int_a \nabla\beta(\nabla\hat{Q})^\top d\boldsymbol{a}. \quad (20)$$

Second Order Method ($M = 2$) We repeat the process, this time taking the derivative of $\hat{Q}$ twice:

$$\hat{I}_{\boldsymbol{\Sigma}^{\frac{1}{2}}} = \mathcal{F}^{-1}\left(\boldsymbol{\Sigma}^{\frac{1}{2}}(i\boldsymbol{\omega})(i\boldsymbol{\omega})^\top\mathcal{F}(\hat{Q})\mathcal{F}(\tilde{\beta})\right),$$
$$= \mathcal{F}^{-1}\left(\boldsymbol{\Sigma}^{\frac{1}{2}}\mathcal{F}(\nabla^{(2)}\hat{Q})\mathcal{F}(\tilde{\beta})\right) = \boldsymbol{\Sigma}^{\frac{1}{2}}\int_a \nabla_{\boldsymbol{a}}^{(2)}\hat{Q}\beta d\boldsymbol{a}. \quad (21)$$

Here, we exploit the multidimensional Fourier transform result for matrices from (9) in deriving the second line.

4.2. Family of SPG Estimators

We are going to revisit certain integrals from Section 4.1 using the following rule for deriving Monte Carlo estimators:

$$\int_a f(a)da \approx \frac{1}{\beta(a)}f(a) \quad \text{where} \quad a \sim \beta.$$

Here, the quantity on the right is a sample-based approximation. We have the following approximations for the integrals given by equations (17,18,19,20,21), which recall were defined for a Gaussian policy $\beta = \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$:

$$
\left.
\begin{aligned}
\hat{I}_{\boldsymbol{\mu}} &= -\int_a \hat{Q}\nabla_{\boldsymbol{a}}\beta d\boldsymbol{a} \approx \boldsymbol{\Sigma}^{-1}(\boldsymbol{a}_t - \boldsymbol{\mu})\hat{Q}, \\
\hat{I}_{\boldsymbol{\Sigma}^{\frac{1}{2}}} &= \boldsymbol{\Sigma}^{\frac{1}{2}}\int_a \hat{Q}\nabla_{\boldsymbol{a}}^{(2)}\beta d\boldsymbol{a}, \\
&\approx \Big((\boldsymbol{\Sigma}^{\frac{1}{2}})^{-\top}(\boldsymbol{a}_t - \boldsymbol{\mu})(\boldsymbol{a}_t - \boldsymbol{\mu})^{\top}\boldsymbol{\Sigma}^{-1} \\
&\quad -(\boldsymbol{\Sigma}^{\frac{1}{2}})^{-\top}\Big)\hat{Q},
\end{aligned}
\right\} M = 0
$$

$$
\left.
\begin{aligned}
\hat{I}_{\boldsymbol{\mu}} &= \int_a \nabla_{\boldsymbol{a}}\hat{Q}\beta d\boldsymbol{a} \approx \nabla_{\boldsymbol{a}}\hat{Q}, \\
\hat{I}_{\boldsymbol{\Sigma}^{\frac{1}{2}}} &= -\boldsymbol{\Sigma}^{\frac{1}{2}}\int_a \nabla_{\boldsymbol{a}}\beta(\nabla_{\boldsymbol{a}}\hat{Q})^{\top}d\boldsymbol{a}, \\
&\approx (\boldsymbol{\Sigma}^{\frac{1}{2}})^{-\top}(\boldsymbol{a}_t - \boldsymbol{\mu})(\nabla_{\boldsymbol{a}}\hat{Q})^{\top},
\end{aligned}
\right\} M = 1
$$

$$
\left.
\begin{aligned}
\hat{I}_{\boldsymbol{\Sigma}^{\frac{1}{2}}} &= \boldsymbol{\Sigma}^{\frac{1}{2}}\int_a \nabla_{\boldsymbol{a}}^{(2)}\hat{Q}\beta d\boldsymbol{a} \\
&\approx \boldsymbol{\Sigma}^{\frac{1}{2}}\nabla_{\boldsymbol{a}}^{(2)}\hat{Q}.
\end{aligned}
\right\} M = 2
$$

The above equations summarise existing results for stochastic policy gradients estimators, which are applicable for *any* policy and critic. The zeroth-order results ($M = 0$) correspond to standard policy gradient methods (Sutton et al., 1999; Williams, 1992; Glynn, 1990); the first-order ones ($M = 1$) correspond to reparameterisation-based methods (Heess et al., 2015; Kingma & Welling, 2013); and, when applied to a Gaussian policy, the second-order ($M = 2$) of the update for Σ is a sample-based version of *Gaussian policy gradients* (Ciosek & Whiteson, 2018a), a special case of EPG. Note that interpolations between different estimators can also be used (Gu et al., 2016) as a method of reducing variance further. The full derivations for of the derivatives for the multivariate Gaussian are given in Appendix E.

4.3. Periodic Action Spaces

In some settings, the action space of an MDP is naturally periodic, e.g., when actions specify angles. By using a trigonometric function in the critic, we encode the insight that rotating by $-\pi$ and by π leads to similar results, despite the fact that the two points lie on the opposite ends of the action range.

Consider the case where the policy is Gaussian, i.e., $\beta = \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ and the critic $\hat{Q}$ is a trigonometric function of the form

$$\hat{Q}(a) = \cos(\boldsymbol{f}^{\top}\boldsymbol{a} - h),$$

where $\boldsymbol{f} \in \mathbb{R}^n, h \in \mathbb{R}$, and n is the dimension of the action space.

While a policy gradient method involving a critic $\hat{Q}$ of this form superficially resembles approximating the value function with the Fourier basis (Konidaris et al., 2011) for the state space, it is in fact completely different. Indeed, our method uses a Fourier basis to approximate a function of the *action space*, not the *state space*, which often has different structure. The dependence of $\hat{Q}$ on the state can still be completely arbitrary (for example a neural network).

We seek to find the policy gradient update for this combination of critic and policy. First, we write out their Fourier transforms:

$$\mathcal{F}(\hat{Q}) = (2\pi)^n \left[\frac{e^{-ih}\delta(\boldsymbol{\omega} - \boldsymbol{f}) + e^{ih}\delta(\boldsymbol{\omega} + \boldsymbol{f})}{2} \right],$$

$$\mathcal{F}(\tilde{\beta}) = e^{-\frac{1}{2}\boldsymbol{\omega}^{\top}\Sigma\boldsymbol{\omega}}.$$

Computing the inverse Fourier transform yields:

$$\mathcal{F}^{-1}(\mathcal{F}(\hat{Q})\mathcal{F}(\tilde{\beta}))(\boldsymbol{a}) = e^{-\frac{1}{2}\boldsymbol{f}^{\top}\Sigma\boldsymbol{f}}\cos(\boldsymbol{f}^{\top}\boldsymbol{a} - h). \quad (22)$$

A more detailed derivation of (22) can be found in appendix G. We now use (10) to obtain the policy gradients for the mean and the covariance.

$$
\begin{aligned}
\hat{I}_{\boldsymbol{\mu}} &= \nabla_{\boldsymbol{\mu}}\mathcal{F}^{-1}(\mathcal{F}(\hat{Q})\mathcal{F}(\tilde{\beta}))(\boldsymbol{\mu}), \\
&= -e^{-\frac{1}{2}\boldsymbol{f}^{\top}\Sigma\boldsymbol{f}}\sin(\boldsymbol{f}^{\top}\boldsymbol{\mu} - h)\boldsymbol{f}, \quad (23) \\
\hat{I}_{\boldsymbol{\Sigma}} &= \nabla_{\boldsymbol{\Sigma}}\mathcal{F}^{-1}(\mathcal{F}(\hat{Q})\mathcal{F}(\tilde{\beta}))(\boldsymbol{\mu}), \\
&= -e^{-\frac{1}{2}\boldsymbol{f}^{\top}\Sigma\boldsymbol{f}}\frac{1}{2}\boldsymbol{f}\boldsymbol{f}^{\top}\cos(\boldsymbol{f}^{\top}\boldsymbol{\mu} - h).
\end{aligned}
$$

Intuitively, the mean update contains a frequency damping component $e^{-\frac{1}{2}\boldsymbol{f}^{\top}\Sigma\boldsymbol{f}}$, which is small for large $\boldsymbol{f}$, ensuring that the optimisation slows down when the signal is repeating frequently. The covariance update uses the same damping, while also making sure that exploration increases in the minima of the critic and decreases near the maxima, in a way slightly similar, but mathematically different, from Gaussian policy gradients (Ciosek & Whiteson, 2018a;b).

We evaluated a periodic critic of this form on a toy *turntable domain* where the goal is to rotate a flat record to the desired position by rotating it (see Appendix D for details). We compared it to the DPG baseline from OpenAI (Dhariwal et al., 2017), which uses a neural network based critic capable of addressing complex control tasks. As expected, the learning curves in Figure 1 show that using a periodic critic (F-EPG) leads to faster learning, because it encodes more information about the action space than a generic neural network. Our method efficiently uses this information in the policy gradient framework by deriving an exact policy gradient update.
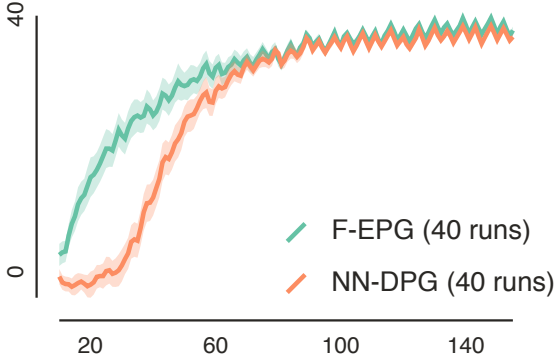
Figure 1. Learning curves for Turntable. EPG with periodic critic (F-EPG) vs. DPG with a neural network critic (NN-DPG).

4.4. Policy Gradients with Radial Basis Functions

Radial basis functions (RBFs) (Buhmann, 2003) have a long tradition as function approximators in machine learning and combine a simple, tractable structure with the universal approximation property (Park & Sandberg, 1991). In this section, we analyse the elementary RBF building block – a *single* RBF node. Results on combining many such blocks are deferred to Section 4.6.

Consider the setting where the policy is Gaussian, i.e., $\beta = \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ and the critic is an RBF $\hat{Q} = \mathcal{N}(\boldsymbol{l}, \boldsymbol{S})$. Although the critic $\hat{Q}$ has the shape of a Gaussian PDF, it is not a random variable but simply an ordinary function parametrised by the location vector $\boldsymbol{l}$ and the positive-definite scale matrix $\boldsymbol{S}$, which occupy the place of the mean and the covariance. We want to find the policy gradient updates for the mean and the covariance. We begin the derivation by writing out the Fourier transforms for the policy and the critic:

$$\mathcal{F}(\hat{Q}) = e^{i\boldsymbol{l}^\top \boldsymbol{\omega} - \frac{1}{2}\boldsymbol{\omega}^\top \boldsymbol{S}\boldsymbol{\omega}},$$
$$\mathcal{F}(\tilde{\beta}) = e^{-\frac{1}{2}\boldsymbol{\omega}^\top \boldsymbol{\Sigma}\boldsymbol{\omega}}.$$

The inverse Fourier transform has the following form:

$$\mathcal{F}^{-1}(\mathcal{F}(\hat{Q})\mathcal{F}(\tilde{\beta}))(\boldsymbol{a}) = \mathcal{N}(\boldsymbol{l}, \boldsymbol{\Sigma} + \boldsymbol{S})(\boldsymbol{a}).$$

Now, we substitute $\boldsymbol{a} = \boldsymbol{\mu}$ and introduce the notation:

$$E = \mathcal{N}(\boldsymbol{l}, \boldsymbol{\Sigma} + \boldsymbol{S})(\boldsymbol{\mu}).$$

We now derive the policy gradients using (10) and properties

of the derivative of the logarithm.

$$\hat{I}_{\boldsymbol{\mu}} = \nabla_{\boldsymbol{\mu}} E = E\nabla_{\boldsymbol{\mu}} \log E,$$
$$= -\frac{1}{2}E\nabla_{\boldsymbol{\mu}}\|\boldsymbol{\mu} - \boldsymbol{l}\|^2_{(\boldsymbol{\Sigma}+\boldsymbol{S})^{-1}}, \qquad (24)$$
$$\hat{I}_{\boldsymbol{\Sigma}} = \nabla_{\boldsymbol{\Sigma}} E = E\nabla_{\boldsymbol{\Sigma}} \log E,$$
$$= -\frac{1}{2}E\left(\nabla_{\boldsymbol{\Sigma}}\|\boldsymbol{\mu} - \boldsymbol{l}\|^2_{(\boldsymbol{\Sigma}+\boldsymbol{S})^{-1}} + \nabla_{\boldsymbol{\Sigma}} \log \det(2\pi\boldsymbol{\Sigma})\right).$$

The RBF Policy gradient simply minimises the Mahalanobis distance with the weight matrix $(\boldsymbol{\Sigma} + \boldsymbol{S})^{-1}$. Also, since E is a positive scalar, for multi-dimensional action spaces, the multiplication by E in the gradients does not change the gradient direction, only the magnitude.

For the mean update $\nabla_{\boldsymbol{\mu}}$, this result is intuitive – if we want our policy to reach the maximum of the RBF node (i.e., a bump) we simply minimise the distance between the current policy mean and the top of the bump. We now provide an additional variant of this result, based on natural policy gradients. The Fisher matrix for the Gaussian distribution parameterised by $\boldsymbol{\mu}$ (with the covariance kept constant) is simply $\boldsymbol{\Sigma}$, yielding the following update:

$$g_{\text{natural}} = -\frac{1}{2}(\boldsymbol{\Sigma})(\boldsymbol{\Sigma} + \boldsymbol{S})^{-1}(\boldsymbol{\mu} - \boldsymbol{l}),$$
$$= -\frac{1}{2}(\boldsymbol{S}\boldsymbol{\Sigma}^{-1} + \text{Id})^{-1}(\boldsymbol{\mu} - \boldsymbol{l}).$$

Here, the symbol Id denotes the identity matrix. The update given by g_{natural} can be used in place of $I_{\boldsymbol{\mu}}$ to obtain a natural policy gradient method. Moving from a standard first order policy gradient to the natural policy gradient is simply a change in the weighting matrix of the Mahalanobis distance from $(\boldsymbol{\Sigma} + \boldsymbol{S})^{-1}$ to $(\boldsymbol{S}\boldsymbol{\Sigma}^{-1} + \text{Id})^{-1}$. Furthermore, the Mahalanobis distance reduces to the unweighted L_2 distance when $\boldsymbol{S} = \boldsymbol{\Sigma}$. Intuitively, since the natural policy gradient takes the geometry of the space of distributions into account, a simpler update is obtained if this geometry is the same as the geometry of the RBF (as given by $\boldsymbol{S}$).

4.5. Revisiting Gaussian Policy Gradients

In this section, we revisit Gaussian policy gradients (Ciosek & Whiteson, 2018a) with the aim of contrasting it with the RBF derivation presented above. Gaussian policy gradients assume that the policy is Gaussian, i.e., $\beta = \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, and the critic is quadric, i.e.,

$$\hat{Q}(a) = \boldsymbol{a}^\top \boldsymbol{H}\boldsymbol{a} + \boldsymbol{a}^\top \boldsymbol{b} + \text{const} = (\boldsymbol{a} - \boldsymbol{l})^\top \boldsymbol{H}(\boldsymbol{a} - \boldsymbol{l}).$$

Here we denote by 'const' some constant which always exists so that the above equality holds for $\boldsymbol{l} = -\frac{1}{2}\boldsymbol{H}^{-1}\boldsymbol{b}$.

In this setting we have that,

$$E = \mathbb{E}_{\beta}\left[(\boldsymbol{a} - \boldsymbol{l})^\top \boldsymbol{H}(\boldsymbol{a} - \boldsymbol{l})\right],$$
$$= \text{trace}(\boldsymbol{H}\boldsymbol{\Sigma}) + (\boldsymbol{\mu} - \boldsymbol{l})^\top \boldsymbol{H}(\boldsymbol{\mu} - \boldsymbol{l}).$$

Now, we compute the policy gradient for the mean:

$$\hat{I}_{\boldsymbol{\mu}} = \nabla_{\boldsymbol{\mu}} E = \nabla_{\boldsymbol{\mu}} (\boldsymbol{\mu} - \boldsymbol{l})^{\top} \boldsymbol{H} (\boldsymbol{\mu} - \boldsymbol{l}).$$

This is almost the same as (24), except for a positive scaling factor and the fact that, $\boldsymbol{H}$ need not be positive-definite (unlike the matrix $\boldsymbol{S}$ from the definition of the RBF.). This illustrates both the similarities and differences between Gaussian policy gradients, which uses quadrics, and the updates given by (24), which is based on RBFs. The similarity is that we are minimising a quadratic form, while the difference is that the quadratic form used by RBF comes from a more restrictive family (i.e., it has to be positive definite). However, RBFs also have some advantages over quadrics in that they are bounded both from above and below.

4.6. Hybrid Critics

We now consider the case when the critic $\hat{Q}$ is a linear combination, i.e., $\hat{Q}(\boldsymbol{a}) = \sum_i c_i \hat{Q}_i(\boldsymbol{a})$ for some c_i. The main observation is that the integral $\hat{I}$ is linear in the critic, i.e., for any parameter θ we have that,

$$\hat{I}_{\theta} = \sum_i c_i \{\hat{I}_{\theta}\}_i + (\nabla_{\theta} c_i) E_i, \quad \text{where}$$

$$E_i = \int_{\boldsymbol{a}} \hat{Q}_i \beta(\boldsymbol{a}) d\boldsymbol{a} \quad \text{and} \quad \{\hat{I}_{\theta}\}_i = \nabla_{\theta} E_i.$$

Thus, we can compute the policy gradient update for each component of the critic separately, and then use a linear combination of the updates. If each of these components is in a tractable family, such as a trigonometric function (Section 4.3), an RBF (Section 4.4) or a polynomial (Ciosek & Whiteson, 2018b), the whole update is also tractable. In this way, we can use critics consisting not just of of a superposition of functions from a single family (like a Fourier basis, which consists of different trigonometric functions), but also hybrid ones, combining functions from many families.

All these three categories of critics have their corresponding universal approximation result, implying that a linear combination of a sufficient number of functions from that class alone is rich enough to approximate any reasonable function on a bounded interval to arbitrary accuracy. Indeed, we have the Weierstrass theorem about linear combinations of monomials (Weierstrass, 1885; Stone, 1948), the result by Park & Sandberg (1991) for linear combinations of RBFs, and the Fourier series approximation for linear combinations of trigonometric functions (see Appendix A).

These results show that, in principle, we can have analytic updates for a critic matching *any* Q-function, and hence any MDP, without the need for Monte Carlo sampling schemes similar to (5), with no sampling noise (given the state) and with virtually no computational overhead relative to stochastic policy gradients. However, there remain two obstacles.

First, for a finite number of basis functions, the approximation may introduce spurious local minima that are harmful to any local optimisation method. Second, even when local minima are not a problem, there is a case for using a degree of sampling in case we believe that our critic $\hat{Q}$ is biased – some sampling methods allow the use of direct reward rollouts to address bias. We believe that the practical impact of our analytic results and the question of which critic combination to use is yet to be determined.

4.7. Mixture Policies and Other Nonstandard Policies

We now consider mixture policies of the form $\beta(\boldsymbol{a}) = \sum_i b_i' \beta_i(\boldsymbol{a})$, where $b_i' \geq 0$ and $\sum_i b_i' = 1$. Similarly to the previous section, we use the linearity of the integral:

$$\hat{I}_{\theta} = \sum_i \{b_i' \hat{I}_{\theta}\}_{\beta_i} + (\nabla_{\theta} b_i') E_{\beta_i}, \quad \text{where} \quad (25)$$

$$E_{\beta_i} = \int_{\boldsymbol{a}} \hat{Q}(\boldsymbol{a}) \beta_i(\boldsymbol{a}) d\boldsymbol{a} \quad \text{and} \quad \{\hat{I}_{\theta}\}_{\beta_i} = \nabla_{\theta} E_{\beta_i}.$$

The two most common types of components for the policy are Gaussian and the deterministic policy (i.e., a Dirac-delta measure). Hence, using (25), we can obtain a policy gradient method for policies that have several modes (modelled with Gaussians) as well as several focussed (discrete) points. Of course, we can also use any other distribution with a characteristic function by substituting into (10). We believe that such policies can be particularly useful in multi-agent settings, where the concept of finding a maximum of the total expected return generalises to finding a Nash equilibrium and it is known that some Nash equilibria admit only stochastic policies (Nisan et al., 2007). It is also possible to have *both* a mixture policy and a hybrid critic. We do not give the formula, since it is straightforward to derive.

5. Conclusions

This paper developed new theoretical tools for deriving policy gradient updates, showing that expected policy gradients are tractable in three important classes of critics and for almost all policies. We also discussed a framework for deriving estimators for stochastic policy gradients, which generalises existing approaches. Moreover, we addressed the setting of MDPs with periodic action spaces and described an experiment demonstrating the benefits of explicitly modelling periodicity in a policy gradient method.

Acknowledgements

References

Alesker, Semyon, Artstein-Avidan, Shiri, and Milman, Vitali. A characterization of the fourier transform and related topics. *Comptes Rendus Mathematique*, 346(11-12): 625–628, 2008.

Ba, Jimmy Lei, Mnih, Volodymyr, and Kavukcuoglu, Koray. Multiple Object Recognition With Visual Attention. *Iclr*, pp. 1–10, 2015.

Baxter, Jonathan and Bartlett, Peter L. Direct gradient-based reinforcement learning. In *Circuits and Systems, 2000. Proceedings. ISCAS 2000 Geneva. The 2000 IEEE International Symposium on*, volume 3, pp. 271–274. IEEE, 2000.

Baxter, Jonathan and Bartlett, Peter L. Infinite-horizon policy-gradient estimation. *Journal of Artificial Intelligence Research*, 15:319–350, 2001.

Baxter, Jonathan, Bartlett, Peter L, and Weaver, Lex. Experiments with infinite-horizon, policy-gradient estimation. *Journal of Artificial Intelligence Research*, 15:351–381, 2001.

Buhmann, Martin D. *Radial basis functions: theory and implementations*, volume 12. Cambridge university press, 2003.

Carr, Jonathan C, Beatson, Richard K, Cherrie, Jon B, Mitchell, Tim J, Fright, W Richard, McCallum, Bruce C, and Evans, Tim R. Reconstruction and representation of 3d objects with radial basis functions. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pp. 67–76. ACM, 2001.

Ciosek, Kamil and Whiteson, Shimon. Expected Policy Gradients. *The Thirty-Second AAAI Conference on Artificial Intelligence (AAAI-18)*, 2018a.

Ciosek, Kamil and Whiteson, Shimon. Expected Policy Gradients for Reinforcement Learning. *Journal submission, arXiv preprint arXiv:1801.03326*, 2018b.

Dhariwal, Prafulla, Hesse, Christopher, Klimov, Oleg, Nichol, Alex, Plappert, Matthias, Radford, Alec, Schulman, John, Sidor, Szymon, and Wu, Yuhuai. Openai baselines. https://github.com/openai/baselines, 2017.

Foerster, Jakob, Farquhar, Gregory, Afouras, Triantafyllos, Nardelli, Nantas, and Whiteson, Shimon. Counterfactual Multi-Agent Policy Gradients. pp. 1–12, 2017. URL http://arxiv.org/abs/1705.08926.

Glynn, Peter W. Likelihood ratio gradient estimation for stochastic systems. *Communications of the ACM (Association of Computing Machinery)*, 33(10):75–84, 1990. ISSN 00010782. doi: 10.1145/84537.84552. URL http://portal.acm.org/citation.cfm?id=84552{%}5Cnhttp://portal.acm.org/ft{_}gateway.cfm?id=84552{&}type=pdf{&}coll=GUIDE{&}dl=GUIDE{&}CFID=92520986{&}CFTOKEN=53025364.

Gu, Shixiang, Lillicrap, Timothy, Ghahramani, Zoubin, Turner, Richard E., and Levine, Sergey. Q-Prop: Sample-Efficient Policy Gradient with An Off-Policy Critic. pp. 1–13, 2016. URL http://arxiv.org/abs/1611.02247.

Heess, Nicolas, Wayne, Greg, Silver, David, Lillicrap, Timothy, Tassa, Yuval, and Erez, Tom. Learning Continuous Control Policies by Stochastic Value Gradients. pp. 1–13, 2015. ISSN 10495258. URL http://arxiv.org/abs/1510.09142.

Kakade, Sham Machandranath et al. *On the sample complexity of reinforcement learning*. PhD thesis, University of London London, England, 2003.

Karr, Alan F. *Characteristic Functions*, pp. 163–182. Springer New York, New York, NY, 1993. ISBN 978-1-4612-0891-4. doi: 10.1007/978-1-4612-0891-4_7.

Kingma, Diederik P and Welling, Max. Auto-Encoding Variational Bayes PPT. *Ppt*, 2013. ISSN 1312.6114v10. URL http://arxiv.org/abs/1312.6114.

Konidaris, George, Osentoski, Sarah, and Thomas, Philip. Value Function Approximation in Reinforcement Learning using the Fourier Basis. *Proceedings of the Twenty-Fifth Conference on Artificial Intelligence*, pp. 380–385, 2011.

Lillicrap, Timothy P, Hunt, Jonathan J, Pritzel, Alexander, Heess, Nicolas, Erez, Tom, Tassa, Yuval, Silver, David, and Wierstra, Daan. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*, 2015.

Mnih, Volodymyr, Kavukcuoglu, Koray, Silver, David, Rusu, Andrei A., Veness, Joel, Bellemare, Marc G., Graves, Alex, Riedmiller, Martin, Fidjeland, Andreas K., Ostrovski, Georg, Petersen, Stig, Beattie, Charles, Sadik, Amir, Antonoglou, Ioannis, King, Helen, Kumaran, Dharshan, Wierstra, Daan, Legg, Shane, and Hassabis, Demis. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015. ISSN 14764687. doi: 10.1038/nature14236.

Nisan, Noam, Roughgarden, Tim, Tardos, Eva, and Vazirani, Vijay V. *Algorithmic game theory*, volume 1. Cambridge University Press Cambridge, 2007.

Park, Jooyoung and Sandberg, Irwin W. Universal approximation using radial-basis-function networks. *Neural computation*, 3(2):246–257, 1991.

Peters, Jan and Schaal, Stefan. Reinforcement learning of motor skills with policy gradients. *Neural networks*, 21 (4):682–697, 2008.

Petersen, K. B. and Pedersen, M. S. The matrix cookbook, nov 2012. Version 20121115.

Schulman, John, Levine, Sergey, Abbeel, Pieter, Jordan, Michael, and Moritz, Philipp. Trust region policy optimization. In *International Conference on Machine Learning*, pp. 1889–1897, 2015.

Schulman, John, Wolski, Filip, Dhariwal, Prafulla, Radford, Alec, and Klimov, Oleg. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Silver, David, Lever, Guy, Heess, Nicolas, Degris, Thomas, Wierstra, Daan, and Riedmiller, Martin. Deterministic Policy Gradient Algorithms. *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, pp. 387–395, 2014. ISSN 1938-7228.

Silver, David, Schrittwieser, Julian, Simonyan, Karen, Antonoglou, Ioannis, Huang, Aja, Guez, Arthur, Hubert, Thomas, Baker, Lucas, Lai, Matthew, Bolton, Adrian, Chen, Yutian, Lillicrap, Timothy, Hui, Fan, Sifre, Laurent, Van Den Driessche, George, Graepel, Thore, and Hassabis, Demis. Mastering the game of Go without human knowledge. *Nature*, 550(7676):354–359, 2017. ISSN 14764687. doi: 10.1038/nature24270.

Stein, Elias M. and Shakarchi, Rami. *Fourier Analysis: An Introduction*. 2003. ISBN 9780691113845.

Stone, Marshall H. The generalized weierstrass approximation theorem. *Mathematics Magazine*, 21(5):237–254, 1948.

Sukhbaatar, Sainbayar, Szlam, Arthur, and Fergus, Rob. Learning Multiagent Communication with Backpropagation. (Nips), 2016. ISSN 10495258. URL http://arxiv.org/abs/1605.07736.

Sutton, Richard S. Generalization in reinforcement learning: Successful examples using sparse coarse coding. In *Advances in neural information processing systems*, pp. 1038–1044, 1996.

Sutton, Richard S. and Barto, Andrew G. Sutton & Barto Book: Reinforcement Learning: An Introduction. *MIT Press, Cambridge, MA, A Bradford Book*, 1998. ISSN 10459227. doi: 10.1109/TNN.1998.712192.

Sutton, Richard S., Mcallester, David, Singh, Satinder, and Mansour, Yishay. Policy Gradient Methods for Reinforcement Learning with Function Approximation. *Advances in Neural Information Processing Systems 12*, pp. 1057–1063, 1999. ISSN 0047-2875. doi: 10.1.1.37.9714.

Sutton, Richard S, Singh, SP, and McAllester, DA. Comparing policy-gradient algorithms. *URL http://citeseerx. ist. psu. edu/viewdoc/download*, 2000.

van Hasselt, Hado, Guez, Arthur, and Silver, David. Deep Reinforcement Learning with Double Q-learning. 2015. ISSN 00043702. doi: 10.1016/j.artint.2015.09.002. URL http://arxiv.org/abs/1509.06461.

van Seijen, Harm, van Hasselt, Hado, Whiteson, Shimon, and Wiering, Marco. A theoretical and empirical analysis of expected sarsa. In *Adaptive Dynamic Programming and Reinforcement Learning, 2009. ADPRL'09. IEEE Symposium on*, pp. 177–184. IEEE, 2009.

Weierstrass, Karl. Über die analytische Darstellbarkeit sogenannter willkürlicher Functionen einer reellen Veränderlichen. *Sitzungsberichte der Königlich Preußischen Akademie der Wissenschaften zu Berlin*, 2:633–639, 1885.

Williams, Ronald J. Simple Statistical Gradient-Following Algorithms for Connectionist Reinforcement Learning. *Machine Learning*, 8(3):229–256, 1992. ISSN 15730565. doi: 10.1023/A:1022672621406.

Wu, Yuhuai, Mansimov, Elman, Grosse, Roger B, Liao, Shun, and Ba, Jimmy. Scalable trust-region method for deep reinforcement learning using kronecker-factored approximation. In *Advances in neural information processing systems*, pp. 5285–5294, 2017.

Xu, Kelvin, Ba, Jimmy, Kiros, Ryan, Cho, Kyunghyun, Courville, Aaron, Salakhutdinov, Ruslan, Zemel, Richard, and Bengio, Yoshua. Show, Attend and Tell: Neural Image Caption Generation with Visual Attention. 2015. ISSN 19410093. doi: 10.1109/72.279181. URL http://arxiv.org/abs/1502.03044.