A Bayesian Approach for Effective Reinforcement Learning through Expectation and Causal Entropy Maximization

Abstract

Policy Search (PS) techniques have received substantial attention in recent years for effective reinforcement learning (RL). Traditional PS algorithms rely mainly on Monte Carlo techniques in updating parametric policies. Recently, aimed at improving learning effectiveness, a complete Bayesian framework has been established, featuring the use of Gaussian Process (GP) modelling techniques for estimating policy gradients. Driven by this framework, this paper explores a wider use of Bayesian techniques by studying two promising possibilities of algorithmic improvements: (1) develop Bayesian techniques to approximate a new type of policy gradient derived from an Expectation-Maximization method; and (2) develop Bayesian techniques to maximize the Causal Entropy for the purpose of encouraging behavioral diversity during RL. Powered by the Bayesian learning techniques, our new RL algorithm is demonstrated to achieve noticeably higher effectiveness than several recently proposed RL algorithms.

1 Introduction

Many cutting-edge *Reinforcement Learning* (RL) algorithms are designed to use *Policy Search* (PS) techniques due to their wide applicability on various practically important problems, including continuous and partially observable problems [Sutton *et al.*, 2000; Kakade, 2001; Peters and Schaal, 2008; Bhatnagar *et al.*, 2009; Peters and Schaal, 2008; J. Kober *et al.*, 2013; Deisenroth and Rasmussen, 2011; Levine and Koltun, 2013]. Without building an *action-selection policy* indirectly from learned value functions, PS methods explicitly maintain parametric policies and are expected to achieve near optimal learning performance through direct update to the corresponding *policy parameters*.

PS is not new to the literature. For a comprehensive summary of popular PS algorithms, please refer to [Peters, 2010; J. Kober *et al.*, 2013; Peters and Schaal, 2008]. Many cutting-edge PS methods require accurate estimation of the *policy gradient* (PG), i.e. the gradient of the learning performance with respect to policy parameters. This can be achieved by introducing an artificial discount factor [Baxter and Bartlett, 2001], a carefully defined *reinforcement baseline* [Sutton *et al.*, 2000; Kakade, 2001], a linear transformation of PG to become *natural PG* [Peters and Schaal, 2008; Chen *et al.*, 2015], or an explicit representation for the value function of a parametric policy [Bhatnagar *et al.*, 2009].

It is noted in [Ghavamzadeh *et al.*, 2016] *et al.* that, rather than following the common practice of using *Monte Carlo* (MC) techniques to update policy parameters, a Bayesian approach [Engel and Ghavamzadeh, 2007; Ghanea-Hercock *et al.*, 2006] can be more effective at reducing the PG estimation variance by following strictly the *likelihood principle* [Berger *et al.*, 1988]. In fact, a complete Bayesian framework is established in [Ghavamzadeh *et al.*, 2016], featuring the use of *Gaussian Process* (GP) modelling techniques for accurate PG estimation from limited learning experiences.

Motivated by the prominent success of existing Bayesian RL algorithms, we aim to explore a wider use of Bayesian techniques for effective RL. For this purpose, we will specifically study two promising possibilities of algorithmic improvement: (1) using Bayesian techniques to estimate a new type of PG derived from an *Expectation-Maximization* (EM) method [Kober and Peters, 2009; Dayan and Hinton, 1997]; and (2) using Bayesian techniques to constantly maximize the *Causal Entropy* (CE) [Wissner-Gross and Freer, 2013] so as to encourage learner's behavioral diversity during RL. We decide to focus on the two possibilities since they contribute directly to the state-of-the-art research on RL and also have significant impact on learning effectiveness.

For possibility (1), we decide to explore a different type of PG derived from the lower bound of the expected performance. Several existing studies have already proved that maximizing this lower bound can often lead to more effective RL [J. Kober *et al.*, 2013; Kober and Peters, 2009; Kormushev *et al.*, 2010]. In this paper, a new approach will be pursued to accurately estimate the expected PG with the help of the *Bayesian Quadrature Theorem* (BQT) [O'Hagan, 1991]. Meanwhile, possibility (2) is closely associated with the *intrinsically-motivated* RL techniques [Mohamed and Rezende, 2015; Salge *et al.*, 2014]. Specifically our research is motivated by the key idea that a learning agent's preference is not forged first by any externally specified environmental rewards, but rather by the agent's internal measure of satisfaction that will naturally nurture sophisticated and desirable behaviors. One of such possible measure to be studied in

this paper is CE, which was originally proposed in [Wissner-Gross and Freer, 2013] to reveal a deep connection between intelligence and entropy maximization. By maximizing CE, the learning agent is expected to maintain its behavioral diversity, essential for effective RL.

Section 2 introduces the research background. In Section 3, we will develop two Bayesian learning techniques that can be utilized together to form a new Bayesian RL algorithm. The effectiveness of our algorithm is then experimentally evaluated in Section 4 with highly positive results. Finally concluding remarks are given in Section 5.

2 Background

Our discussion of the research background begins with a definition of the RL problem. We then present a general introduction to BQT for RL. Afterwards, a performance lower bound for RL will be derived from a simple EM method.

2.1 Reinforcement Learning Problems

The environment for RL is often described as a *Markov Decision Process* (MDP) [Sutton and Barto, 1998]. In this paper, we consider particularly MDPs with *continuous states* $\boldsymbol{s} \in \mathbb{S} \subseteq \mathbb{R}^n$ and *discrete actions* $a \in \{a^1, \ldots, a^m\}$. At each time step t, a learning agent observes its current state $\boldsymbol{s}_t$ and performs one of the m alternative actions a_t, resulting in a *state transition* to a new state $\boldsymbol{s}_{t+1}$. Meanwhile, a scalar reward $r(\boldsymbol{s}_t, a_t)$ will be produced as an environmental feedback. Starting from an initial state $\boldsymbol{s}_0$ at time $t = 0$, the typical performance measure for RL is presented as the long-term *cumulative reward* defined below

$$V(\boldsymbol{s}) = E\left\{\sum_{t=0}^{T} r(\boldsymbol{s}_t, a_t) \,|\boldsymbol{s}_0 = \boldsymbol{s}\right\} \tag{1}$$

where T is the total number of time steps in each learning episode. For the purpose of maximizing the learning performance in (1), the agent must identify a suitable policy π that will guide its selection of desirable actions. Specifically, $\pi(\boldsymbol{s}_t, a_t)$ is treated as a parametric function that specifies the probability of choosing any action a_t in arbitrary state $\boldsymbol{s}_t$. To facilitate learning, we assume that any parametric policy $\pi_{\boldsymbol{\theta}}$ and the performance obtainable upon using the policy are *differentiable* with respect to the corresponding policy parameters $\boldsymbol{\theta}$. This assumption enables us to achieve the RL goal of identifying the optimal policy parameters, i.e. $\boldsymbol{\theta}^*$:

$$\boldsymbol{\theta}^* = \arg\max_{\boldsymbol{\theta}} V^{\pi_{\boldsymbol{\theta}}}(\boldsymbol{s}_0) \tag{2}$$

2.2 Bayesian Quadrature Theorem for Reinforcement Learning

In practice the calculation of PG for RL often demands for the evaluation of an integral by using samples of its integrand. For example, PG can be determined through integration over the cumulative rewards of randomly sampled *trajectories* in [Williams, 1992; Engel and Ghavamzadeh, 2007]. Here a trajectory, i.e. $\xi_{\boldsymbol{\theta}} = \{\boldsymbol{s}_0, a_0, \ldots, \boldsymbol{s}_T, a_T\}$, refers to a series of states visited during a learning episode by following strictly a policy $\pi_{\boldsymbol{\theta}}$. Such integral is often of the general form

$$\rho = \int F(\xi) \cdot G(\xi) \mathrm{d}\xi \tag{3}$$

According to BQT, $F(\xi)$ holds random and unknown values. On the other hand $G(\xi)$ is considered to be known and well-defined. Due to our inherent uncertainty about $F(\xi)$, in BQT $F(\xi)$ is described through a GP model where the *prior mean* and *covariance* of the model are defined as

$$E[F(\xi)] = \tilde{F}(\xi) \text{ and } Cov[F(\xi), F(\xi')] = k(\xi, \xi') \tag{4}$$

respectively. Here $k(\cdot, \cdot)$ is a pre-determined *kernel function*. To obtain a good understanding of $F(\xi)$, suppose that a series of μ examples of the form $\boldsymbol{D}_\mu = \{(\xi_i, y_i \approx F(\xi_i))\}_{i=1}^{\mu}$ have been made available to the learning agent, where each y_i is a possibly noisy sample of $F(\xi_i)$. Using $\boldsymbol{D}_\mu$, the *posterior mean* and covariance of $F(\xi)$ can be subsequently determined as

$$\begin{aligned} &E[F(\xi)|\boldsymbol{D}_\mu] = \tilde{F}(\xi) + \boldsymbol{k}_*^T(\xi) \cdot \boldsymbol{C}(\boldsymbol{D}_\mu) \cdot (\boldsymbol{Y} - \boldsymbol{F}_0) \text{ and} \\ &Cov[F(\xi), F(\xi')] = k(\xi, \xi') - \boldsymbol{k}_*^T(\xi) \cdot \boldsymbol{C}(\boldsymbol{D}_\mu) \cdot \boldsymbol{k}_*(\xi') \end{aligned} \tag{5}$$

where

$$\boldsymbol{k}_*(\xi) = \{k(\xi, \xi_1), \ldots, k(\xi, \xi_\mu)\},\ \boldsymbol{Y} = \{y_1, \ldots, y_\mu\} \text{ and}$$
$$\boldsymbol{F}_0 = \{\tilde{F}(\xi_1), \ldots, \tilde{F}(\xi_\mu)\},\ \boldsymbol{C}(\boldsymbol{D}_\mu) = (\boldsymbol{K}(\boldsymbol{D}_\mu) + \sigma \cdot \boldsymbol{I})^{-1}$$

Here $\boldsymbol{K}(\boldsymbol{D}_\mu)$ is the Gram matrix with entries $[\boldsymbol{K}(\boldsymbol{D}_\mu)]_{i,j} = k(\xi_i, \xi_j)$ and σ is an arbitrarily small noise factor to be determined manually. Based on (5) and by following BQT, the posterior mean of ρ in (3) is further obtained as

$$E[\rho|\boldsymbol{D}_\mu] = \rho_0 + \boldsymbol{Z}^T \cdot \boldsymbol{C}(\boldsymbol{D}_\mu) \cdot (\boldsymbol{Y} - \boldsymbol{F}_0) \tag{6}$$

where

$$\rho_0 = \int \tilde{F}(\xi) G(\xi) \mathrm{d}\xi \text{ and } \boldsymbol{Z} = \int \boldsymbol{k}_*(\xi) \cdot G(\xi) \mathrm{d}\xi$$

It was shown in [Ghavamzadeh *et al.*, 2016] that a closed-form approximation of the PG defined in (7) can be obtained through BQT, giving rise to more effective RL performance.

$$\frac{\partial V^{\pi_{\boldsymbol{\theta}}}(\boldsymbol{s}_0)}{\partial \boldsymbol{\theta}} = \int_\xi R(\xi) \times \frac{\partial Pr_{\boldsymbol{\theta}}(\xi)}{\partial \boldsymbol{\theta}} \mathrm{d}\xi \tag{7}$$

Here $R(\xi)$ refers to the cumulative reward that can be obtained through trajectory ξ. Different from [Ghavamzadeh *et al.*, 2016], we will exploit BQT to compute a new type of PG identified in Subsection 3.1.

2.3 Determining Performance Lower Bound through Expectation-Maximization

While designing new machine learning algorithms, it is often desirable for us to focus on optimizing a lower bound of the predetermined performance measure. This idea fuels the development of many EM algorithms widely explored in the literature [Dayan and Hinton, 1997; Kober and Peters, 2009; Kormushev *et al.*, 2010]. In the field of RL, the Policy learning by Weighting Exploration with the Returns (PoWER) algorithm is a recent example that can attribute its success to EM [Kober and Peters, 2009]. According to [Kober and Peters, 2009], given any policy parameters $\boldsymbol{\theta}'$, the performance

of using the corresponding policy can be bounded from below as

$$\begin{aligned}\log V^{\pi_{\boldsymbol{\theta}'}}(\boldsymbol{s}_0) &= \log \int_{\xi} \frac{Pr_{\boldsymbol{\theta}}(\xi)}{Pr_{\boldsymbol{\theta}}(\xi)} Pr_{\boldsymbol{\theta}'}(\xi) \cdot R(\xi) \mathrm{d}\xi \\ &\geq \int_{\xi} Pr_{\boldsymbol{\theta}}(\xi) R(\xi) \cdot \log \frac{Pr_{\boldsymbol{\theta}'}(\xi)}{Pr_{\boldsymbol{\theta}}(\xi)} \mathrm{d}\xi + \mathbb{C} \\ &\propto \int_{\xi} Pr_{\boldsymbol{\theta}}(\xi) R(\xi) \log Pr_{\boldsymbol{\theta}'}(\xi) \mathrm{d}\xi = L_{\boldsymbol{\theta}}(\boldsymbol{\theta}')\end{aligned} \tag{8}$$

where $\boldsymbol{\theta}$ in (8) serves as another set of policy parameters, different from $\boldsymbol{\theta}'$. Guided by (8), it is straightforward to see that, rather than maximizing $V(\boldsymbol{s}_0)$ directly, we can optimize the lower bound $L_{\boldsymbol{\theta}}(\boldsymbol{\theta}')$ to indirectly achieve the learning goal in (2). Hence we identify a new type of PG below.

$$\frac{\partial L_{\boldsymbol{\theta}}(\boldsymbol{\theta}')}{\partial \boldsymbol{\theta}'} = \int_{\xi} Pr_{\boldsymbol{\theta}}(\xi) R(\xi) \frac{\partial \log Pr_{\boldsymbol{\theta}'}(\xi)}{\partial \boldsymbol{\theta}'} \mathrm{d}\xi \tag{9}$$

In [Kober and Peters, 2009], instead of using $\partial L_{\boldsymbol{\theta}}/\partial \boldsymbol{\theta}'$ to update policy parameters, the equation $\partial L_{\boldsymbol{\theta}}/\partial \boldsymbol{\theta}' = 0$ is directly solved to determine suitable $\boldsymbol{\theta}'$ for the next learning stage. Such $\boldsymbol{\theta}'$ is expected to directly maximize the lower bound $L_{\boldsymbol{\theta}}(\boldsymbol{\theta}')$ in (8). However, the solution to this equation requires two separate integrals to be evaluated. Since MC is applied to computing both integrals, the resulting $\boldsymbol{\theta}'$ can exhibit high level of variance. We therefore decide to follow a Bayesian approach to update policy parameters.

3 Using Bayesian Techniques for Effective Reinforcement Learning

We will develop two Bayesian techniques for effective RL. First BQT will be utilized to analytically approximate the expected PG. Then we will apply BQT again to develop a learning rule for maximizing CE. Finally we will present a complete Bayesian RL algorithm.

3.1 A Bayesian Approach to Estimating Policy Gradient

To accurately estimate PG in our RL algorithm, the integral in (9) can be processed through BQT. Specifically, a quick inspection of (9) suggests that the integrand can be divided into two parts, as shown below

$$F(\xi) = R(\xi) \text{ and } G(\xi) = Pr_{\boldsymbol{\theta}}(\xi) \cdot \frac{\partial \log Pr_{\boldsymbol{\theta}'}(\xi)}{\partial \boldsymbol{\theta}'} \tag{10}$$

Following the discussion in Subsection 2.2, $F(\xi)$ is treated as the unknown part to be further captured through a GP model. Meanwhile, $G(\xi)$ is considered as the known part and should be easily determined with respect to any trajectory ξ. For example, consider a specific trajectory $\xi = \{\boldsymbol{s}_0, a_0, \ldots, \boldsymbol{s}_T, a_T\}$, the probability of obtaining this trajectory upon following a policy $\pi_{\boldsymbol{\theta}}$ equals to

$$Pr_{\boldsymbol{\theta}}(\xi) = Pr(\boldsymbol{s}_0) \times Pr(\boldsymbol{s}_1|\boldsymbol{s}_0, a_0) \times \pi_{\boldsymbol{\theta}}(\boldsymbol{s}_0, a_0) \times \ldots \times \pi_{\boldsymbol{\theta}}(\boldsymbol{s}_T, a_T) \tag{11}$$

Provided that the state transition is deterministic in the learning environment, (11) can be re-written as

$$Pr_{\boldsymbol{\theta}}(\xi) = Pr(\boldsymbol{s}_0) \times \pi_{\boldsymbol{\theta}}(\boldsymbol{s}_0, a_0) \times \ldots \times \pi_{\boldsymbol{\theta}}(\boldsymbol{s}_T, a_T) \tag{12}$$

Similarly,

$$\frac{\partial \log Pr_{\boldsymbol{\theta}'}(\xi)}{\partial \boldsymbol{\theta}'} = \frac{\partial \log \pi_{\boldsymbol{\theta}'}(\boldsymbol{s}_0, a_0)}{\partial \boldsymbol{\theta}'} + \ldots + \frac{\partial \log \pi_{\boldsymbol{\theta}'}(\boldsymbol{s}_T, a_T)}{\partial \boldsymbol{\theta}'} \tag{13}$$

Both (12) and (13) can be calculated straightforwardly without any extra knowledge of the learning environment. Based on (10), BQT can be employed to compute the expected PG. However, different from [Ghavamzadeh *et al.*, 2016], in an attempt to reduce the modelling bias, the prior mean of $F(\xi)$ in (10) will be estimated through

$$\tilde{F}(\xi) = \boldsymbol{\omega}^T \cdot \frac{\partial \log Pr_{\boldsymbol{\theta}'}(\xi)}{\partial \boldsymbol{\theta}'} \tag{14}$$

in this paper. Here $\boldsymbol{\omega}$ is a vector of parameters for approximating $F(\xi)$ and can be easily determined by minimizing the *Mean Squared Error* (MSE) in between $\tilde{F}(\xi_i)$ and $F(\xi_i)$ over a collection of sampled trajectories $\{\xi_1, \ldots, \xi_\mu\}$. The details will be omitted. Moreover we can also build a training data set $\boldsymbol{D}_\mu = \{(\xi_i, R(\xi_i))\}_{i=1}^{\mu}$ to evaluate the expected PG. According to BQT and (6),

$$E\left[\frac{\partial L_{\boldsymbol{\theta}}(\boldsymbol{\theta}')}{\partial \boldsymbol{\theta}'} | \boldsymbol{D}_\mu\right] = \boldsymbol{\rho}_0^{\boldsymbol{\theta}'} + \boldsymbol{Z}_{\boldsymbol{\theta}'}^T \cdot \boldsymbol{C}_{\boldsymbol{\theta}'}(\boldsymbol{D}_\mu) \cdot (\boldsymbol{Y} - \boldsymbol{F}_0^{\boldsymbol{\theta}'}) \tag{15}$$

Similar to (6), $\boldsymbol{Y} = \{R(\xi_1), \ldots, R(\xi_\mu)\}$ and $\boldsymbol{F}_0^{\boldsymbol{\theta}'} = \{\tilde{F}(\xi_1), \ldots, \tilde{F}(\xi_\mu)\}$ in (15). We will show in the following that each of $\boldsymbol{\rho}_0^{\boldsymbol{\theta}'}$, $\boldsymbol{Z}_{\boldsymbol{\theta}'}$, and $\boldsymbol{C}_{\boldsymbol{\theta}'}(\boldsymbol{D}_\mu)$ can be calculated analytically. Subsequently, the expected PG in (15) can also be approximated in a Bayesian fashion. In line with (6), the derivation of $\boldsymbol{\rho}_0^{\boldsymbol{\theta}'}$ is given first

$$\begin{aligned}\boldsymbol{\rho}_0^{\boldsymbol{\theta}'} &= \int_{\xi} \tilde{F}(\xi) \cdot Pr_{\boldsymbol{\theta}}(\xi) \cdot \frac{\partial \log Pr_{\boldsymbol{\theta}'}(\xi)}{\partial \boldsymbol{\theta}'} \mathrm{d}\xi \\ &= \int_{\xi} \left(\boldsymbol{\omega}^T \cdot \frac{\partial \log Pr_{\boldsymbol{\theta}'}(\xi)}{\partial \boldsymbol{\theta}'}\right) \cdot Pr_{\boldsymbol{\theta}}(\xi) \cdot \frac{\partial \log Pr_{\boldsymbol{\theta}'}(\xi)}{\partial \boldsymbol{\theta}'} \mathrm{d}\xi \\ &= \boldsymbol{G}_{\boldsymbol{\theta}'} \cdot \boldsymbol{\omega}\end{aligned} \tag{16}$$

$\boldsymbol{G}_{\boldsymbol{\theta}'}$ in (16) is an important square matrix, which is utilized to define the kernel function below

$$k_{\boldsymbol{\theta}'}(\xi, \xi') = \left(\frac{\partial \log Pr_{\boldsymbol{\theta}'}(\xi)}{\partial \boldsymbol{\theta}'}\right)^T \cdot \boldsymbol{G}_{\boldsymbol{\theta}'}^{-1} \cdot \frac{\partial \log Pr_{\boldsymbol{\theta}'}(\xi)}{\partial \boldsymbol{\theta}'} \tag{17}$$

Using (17) and the same as in (5),

$$\begin{aligned}\boldsymbol{C}_{\boldsymbol{\theta}'}(\boldsymbol{D}_\mu) &= (\boldsymbol{K}_{\boldsymbol{\theta}'}(\boldsymbol{D}_\mu) + \sigma \cdot \boldsymbol{I})^{-1} \\ [\boldsymbol{K}_{\boldsymbol{\theta}'}(\boldsymbol{D}_\mu)]_{i,j} &= k_{\boldsymbol{\theta}'}(\xi_i, \xi_j), 1 \leq i, j \leq \mu\end{aligned} \tag{18}$$

Next, let $\boldsymbol{U}_{\boldsymbol{\theta}'}$ represent a matrix with μ columns. Its i-th column, $1 \leq i \leq \mu$, equals to $\partial \log Pr_{\boldsymbol{\theta}'}(\xi_i)/\partial \boldsymbol{\theta}'$. Based on $\boldsymbol{U}_{\boldsymbol{\theta}'}$ and (17), $\boldsymbol{Z}_{\boldsymbol{\theta}'}$ can be determined according to

$$\begin{aligned}\boldsymbol{Z}_{\boldsymbol{\theta}'} &= \int_{\xi} \boldsymbol{k}_*^{\boldsymbol{\theta}'}(\xi) \cdot Pr_{\boldsymbol{\theta}}(\xi) \cdot \left(\frac{\partial \log Pr_{\boldsymbol{\theta}'}(\xi)}{\partial \boldsymbol{\theta}'}\right)^T \mathrm{d}\xi \\ &= \int_{\xi} \left\{\left(\frac{\partial \log Pr_{\boldsymbol{\theta}'}(\xi)}{\partial \boldsymbol{\theta}'}\right)^T \cdot \boldsymbol{G}_{\boldsymbol{\theta}'}^{-1} \cdot \boldsymbol{U}_{\boldsymbol{\theta}'}\right\}^T \cdot Pr_{\boldsymbol{\theta}}(\xi) \cdot \left(\frac{\partial \log Pr_{\boldsymbol{\theta}'}(\xi)}{\partial \boldsymbol{\theta}'}\right)^T \mathrm{d}\xi \\ &= \boldsymbol{U}_{\boldsymbol{\theta}'}^T \cdot \boldsymbol{G}_{\boldsymbol{\theta}'}^{-1} \cdot \int_{\xi} \frac{\partial \log Pr_{\boldsymbol{\theta}'}(\xi)}{\partial \boldsymbol{\theta}'} \cdot \left(\frac{\partial \log Pr_{\boldsymbol{\theta}'}(\xi)}{\partial \boldsymbol{\theta}'}\right)^T \cdot Pr_{\boldsymbol{\theta}}(\xi) \mathrm{d}\xi \\ &= \boldsymbol{U}_{\boldsymbol{\theta}'}^T\end{aligned} \tag{19}$$

With the help of (16), (18), and (19), we have successfully established a Bayesian approach for approximating the expected PG in (15). Different from the Bayesian RL algorithms proposed in [Ghavamzadeh *et al.*, 2016], our method allows the policy parameters to be updated based on trajectories obtained by following a different policy. It naturally facilitates us to adopt a simple importance sampling technique for effective RL [J. Kober *et al.*, 2013], as described in Algorithm 1.

3.2 A Bayesian Approach to Maximizing Causal Entropy

CE will be studied in this subsection as an important self-satisfaction measure that helps a learning agent to improve its behavioral diversity during RL. Following [Wissner-Gross and Freer, 2013] and in consideration of the RL problem described in Subsection 2.1, CE can be defined straightforwardly as below

$$L_{\boldsymbol{\theta}}^{CE} = -\kappa \cdot \int_{\xi} Pr_{\boldsymbol{\theta}}(\xi) \cdot \log Pr_{\boldsymbol{\theta}}(\xi) \mathrm{d}\xi \tag{20}$$

where κ is a positive constant. Apparently from (20), $L_{\boldsymbol{\theta}}^{CE}$ will reach a high value whenever a wide variety of different trajectories can be generated with high probability upon following a policy $\pi_{\boldsymbol{\theta}}$. Hence, by updating policy parameters $\boldsymbol{\theta}$ in the direction of increasing $L_{\boldsymbol{\theta}}^{CE}$, behavioral diversity will be encouraged and it is more likely for the agent to successfully learn suitable ways to interact with its environment [Wissner-Gross and Freer, 2013]. Guided by this understanding, it is important to determine the gradient of $L_{\boldsymbol{\theta}}^{CE}$ with respect to $\boldsymbol{\theta}$, as shown below.

$$\begin{aligned}\frac{\partial L_{\boldsymbol{\theta}}^{CE}}{\partial \boldsymbol{\theta}} &= -\kappa \cdot \int_{\xi} \log Pr_{\boldsymbol{\theta}}(\xi) \frac{\partial Pr_{\boldsymbol{\theta}}(\xi)}{\partial \boldsymbol{\theta}} + \frac{\partial Pr_{\boldsymbol{\theta}}(\xi)}{\partial \boldsymbol{\theta}} \mathrm{d}\xi \\ &= -\kappa \int_{\xi} \left(1 + \log Pr_{\boldsymbol{\theta}}(\xi)\right) \frac{\partial Pr_{\boldsymbol{\theta}}(\xi)}{\partial \boldsymbol{\theta}} \mathrm{d}\xi\end{aligned} \tag{21}$$

For the integral in (21) above, similar to our analysis in Subsection 3.1, the corresponding integrand can be separated into two parts:

$$F_{CE}(\xi) = 1 + \log Pr_{\boldsymbol{\theta}}(\xi) \text{ and } G_{CE}(\xi) = \frac{\partial Pr_{\boldsymbol{\theta}}(\xi)}{\partial \boldsymbol{\theta}} \tag{22}$$

where $F_{CE}(\xi)$ and $G_{CE}(\xi)$ are treated as the unknown and known parts respectively. When $F_{CE}(\xi)$ is described through a GP model, the prior mean of the model can be obtained as

$$\tilde{F}_{CE} = \frac{1}{\mu} \sum_{i=1}^{\mu} (1 + \log Pr_{\boldsymbol{\theta}}(\xi_i)) \tag{23}$$

based on a collection of training examples $\boldsymbol{D}_{\mu} = \{(\xi_i, 1 + \log Pr_{\boldsymbol{\theta}}(\xi_i))\}_{i=1}^{\mu}$. Meanwhile, the prior covariance of $F_{CE}(\xi)$ is captured through the kernel function:

$$k_{CE}(\xi, \xi') = \left(\frac{\partial \log Pr_{\boldsymbol{\theta}}(\xi)}{\partial \boldsymbol{\theta}}\right)^T \cdot \boldsymbol{G}_{CE}^{-1} \cdot \frac{\partial \log Pr_{\boldsymbol{\theta}}(\xi')}{\partial \boldsymbol{\theta}} \tag{24}$$

where

$$\boldsymbol{G}_{CE} = \int_{\xi} \frac{\partial \log Pr_{\boldsymbol{\theta}}(\xi)}{\partial \boldsymbol{\theta}} \cdot \left(\frac{\partial \log Pr_{\boldsymbol{\theta}}(\xi)}{\partial \boldsymbol{\theta}}\right)^T \cdot Pr_{\boldsymbol{\theta}}(\xi) \mathrm{d}\xi$$

We can then proceed to evaluate the expectation of $\partial L_{\boldsymbol{\theta}}^{CE} / \partial \boldsymbol{\theta}$ through

$$E\left[\frac{\partial L_{\boldsymbol{\theta}}^{CE}}{\partial \boldsymbol{\theta}} | \boldsymbol{D}_{\mu}\right] = -\kappa \left(\boldsymbol{\rho}_0^{CE} + \boldsymbol{Z}_{CE}^T \cdot \boldsymbol{C}_{CE} \cdot (\boldsymbol{Y}_{CE} - \tilde{\boldsymbol{F}}_{CE})\right) \tag{25}$$

where $\boldsymbol{Y}_{CE} = \{1 + \log Pr_{\boldsymbol{\theta}}(\xi_1), \ldots, 1 + \log Pr_{\boldsymbol{\theta}}(\xi_{\mu})\}$. $\boldsymbol{C}_{CE}$ is defined similarly as in (18) and the details will be omitted. For practical use of (25), $\boldsymbol{\rho}_0^{CE}$ and $\boldsymbol{Z}_{CE}$ in (25) have to be further computed from $\boldsymbol{D}_{\mu}$. Particularly

$$\begin{aligned}\boldsymbol{\rho}_0^{CE} &= \int_{\xi} \tilde{F}_{CE} \cdot \frac{\partial Pr_{\boldsymbol{\theta}}(\xi)}{\partial \boldsymbol{\theta}} \mathrm{d}\xi \\ &= \int_{\xi} \tilde{F}_{CE} \cdot \frac{\partial \log Pr_{\boldsymbol{\theta}}(\xi)}{\partial \boldsymbol{\theta}} \cdot Pr_{\boldsymbol{\theta}}(\xi) \mathrm{d}\xi \\ &\approx \frac{\tilde{F}_{CE}}{\mu} \sum_{i=1}^{\mu} \frac{\partial \log Pr_{\boldsymbol{\theta}}(\xi_i)}{\partial \boldsymbol{\theta}}\end{aligned} \tag{26}$$

Meanwhile,

$$\begin{aligned}\boldsymbol{Z}_{CE} &= \int_{\xi} \boldsymbol{k}_*^{CE} \cdot \left(\frac{\partial Pr_{\boldsymbol{\theta}}(\xi)}{\partial \boldsymbol{\theta}}\right)^T \mathrm{d}\xi \\ &= \int_{\xi} \boldsymbol{k}_*^{CE} \cdot \left(\frac{\partial \log Pr_{\boldsymbol{\theta}}(\xi)}{\partial \boldsymbol{\theta}}\right)^T Pr_{\boldsymbol{\theta}}(\xi) \mathrm{d}\xi \\ &= \int_{\xi} \left\{\left(\frac{\partial \log Pr_{\boldsymbol{\theta}}(\xi)}{\partial \boldsymbol{\theta}}\right)^T \cdot \boldsymbol{G}_{CE}^{-1} \cdot \boldsymbol{U}_{CE}\right\}^T \cdot \left(\frac{\partial \log Pr_{\boldsymbol{\theta}}(\xi)}{\partial \boldsymbol{\theta}}\right)^T Pr_{\boldsymbol{\theta}}(\xi) \mathrm{d}\xi \\ &= \boldsymbol{U}_{CE}^T \cdot \boldsymbol{G}_{CE}^{-1} \cdot \int_{\xi} \frac{\partial \log Pr_{\boldsymbol{\theta}}(\xi)}{\partial \boldsymbol{\theta}} \cdot \left(\frac{\partial \log Pr_{\boldsymbol{\theta}}(\xi)}{\partial \boldsymbol{\theta}}\right)^T \cdot Pr_{\boldsymbol{\theta}}(\xi) \mathrm{d}\xi \\ &= \boldsymbol{U}_{CE}^T\end{aligned} \tag{27}$$

where $\boldsymbol{U}_{CE}$ is a matrix with μ columns. Its i-th column, $1 \leq i \leq \mu$, equals to $\partial \log Pr_{\boldsymbol{\theta}}(\xi_i) / \partial \boldsymbol{\theta}$. Up to now, by using (25), (26), and (27) together, a Bayesian method has been successfully established for evaluating $E[\partial L_{\boldsymbol{\theta}}^{CE} / \partial \boldsymbol{\theta}]$. Subsequently, through repeated updating of $\boldsymbol{\theta}$ along the direction of $E[\partial L_{\boldsymbol{\theta}}^{CE} / \partial \boldsymbol{\theta}]$, the agent's internal satisfaction and its behavioral diversity can be improved.

3.3 A Complete Bayesian Algorithm for Reinforcement Learning

In Subsection 3.1 and Subsection 3.2 we have developed two Bayesian learning methods, one for maximizing the performance lower bound and one for maximizing behavioral diversity. Both of the two methods can be jointly utilized in the same Bayesian RL algorithm, as we summarized in Algorithm 1.

4 Empirical Analysis

We perform experimental studies on two benchmark RL problems, i.e., the Cart Pole problem [Sutton and Barto, 1998; Duan *et al.*, 2016] and the Heating Coil problem [Anderson *et al.*, 1997; Hafner and Riedmiller, 2011]. The Cart Pole problem has been widely used as a popular RL testbed [Sutton

Algorithm 1 A Bayesian Algorithm for RL.

Require: an MDP $\langle \mathbb{S}, \mathbb{A}, \mathbb{P}, \mathbb{R} \rangle$,
α: the learning rate for performance maximization,
β: the learning rate for maximizing CE ($\beta = \alpha/2$ in experiments).

repeat for a given number of learning episodes:
  Use learned policy $\pi_{\boldsymbol{\theta}}$ to generate a new trajectory ξ.
  Evaluate the cumulative reward of ξ, i.e. $R(\xi)$.
  Importance Sampling:
    Compare $R(\xi)$ against rewards obtained by $\mu = 10$ existing trajectories in a repository.
    Replace the trajectory with the lowest reward in the repository by ξ if $R(\xi)$ is larger.
  Calculate $\frac{\partial L_{\boldsymbol{\theta}}(\boldsymbol{\theta}')}{\partial \boldsymbol{\theta}'}$ where $\boldsymbol{\theta}'$ equals to the policy parameters $\boldsymbol{\theta}$. ▷ See Subsection 3.1.
  Update policy parameters:

$$\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \alpha \cdot \frac{\partial L_{\boldsymbol{\theta}}(\boldsymbol{\theta}')}{\partial \boldsymbol{\theta}'}$$

  Calculate $\frac{\partial L_{\boldsymbol{\theta}}^{CE}}{\partial \boldsymbol{\theta}}$. ▷ See Subsection 3.2.
  Update policy parameters:

$$\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \beta \cdot \frac{\partial L_{\boldsymbol{\theta}}^{CE}}{\partial \boldsymbol{\theta}}$$

and Barto, 1998]. The problem considers the task of continuously balancing a pole mounted on a cart that is driven by a force (i.e., an action) $f \in [-10.0, 10.0]$ and moves horizontally from -10.0 to 10.0. The state of this problem is presented as a 4-dimensional vector, i.e., $\{x, \dot{x}, \beta, \dot{\beta}\}$. Each dimension represents respectively the cart's position, the cart's velocity $\dot{x}$, the pole's angle β (a.k.a, angle), and the pole's angular velocity $\dot{\beta}$.

Please refer to [Sutton and Barto, 1998] for detailed system motions in the Cart Pole problem which is also associated with three terminating conditions: (1) the pole fails to be balanced (i.e., $|\beta| > 0.2\pi$), (2) the cart moves beyond its boundary (i.e., $|x| > 10.0$), or (3) the maximum number of time steps is reached (200 for each learning episode and 2000 for each testing episode).

The Heating Coil is a challenging and recently proposed benchmark RL problem [Hafner and Riedmiller, 2011]. It simulates a complex Heating, Ventilation and Air Conditioning (HVAC) system with highly non-linear system dynamics. The objective of this problem is to control the output air temperature of an HVAC to be as close as possible to the reference temperature of 44° in our experiments. Without considering the environmental noises as originally modelled in [Hafner and Riedmiller, 2011] due to our requirement of deterministic state transitions, the state of the Heating Coil is represented as a 3-dimensional vector, i.e., $\{f_w, T_{wi}, T_{ao}\}$. They are the flow rate of boiler water $f_w \in [0.0, 1.0]$, the temperature of input boiler water $T_{wi} \in [73.0, 81.0]$, and the temperature of output air $T_{ao} \in [40.0, 50.0]$, respectively. Each learning/testing episode ends immediately after a certain number of continued adjustments to the control valve (i.e. the actions), which equals to 20 for learning and 100 for testing.

4.1 Experiment Setup

Several algorithms will be jointly examined in our experiments. To simplify our discussion, we will name Algorithm 1 as B-EMCE-PG, meaning that both the EM-based performance lower bound and the CE-based behavioral diversity are collectively maximized in a Bayesian style in the algorithm. Additionally a variation of Algorithm 1 that maximizes only the performance lower bound will be termed B-EM-PG. The two algorithms are also compared to another two commonly used algorithms, i.e. *Regular Actor Critic* (RAC) and *Natural Actor Critic with Advantage Parameters* (NACAP) proposed recently in [Bhatnagar *et al.*, 2009]. To understand the usefulness of Bayesian learning techniques, we have also implemented an algorithm called EM-PG that follows Algorithm 1 but relies purely on the MC sampling method (over a collection of μ sampled trajectories) shown below to estimate policy gradients.

$$\frac{\partial L_{\boldsymbol{\theta}}(\boldsymbol{\theta}')}{\partial \boldsymbol{\theta}'} \approx \frac{1}{\mu} \cdot \sum_{\xi \in \{\xi_1, \ldots, \xi_\mu\}} R(\xi) \frac{\partial \log Pr_{\boldsymbol{\theta}'}(\xi)}{\partial \boldsymbol{\theta}'} \tag{28}$$

All algorithms in our experimental study are designed to work with state features. In this paper we decide to implement simple triangular state features to avoid providing too much domain knowledge to any learning algorithm [Chen *et al.*, 2015]. For example, the complete range of the pole angle β in the Cart Pole problem is covered by 10 equal-spaced triangular functions in our experiments. Hence there are 10 separate state features associated with β. At any time, each triangular function takes the current pole angle as its input and subsequently produce a value from 0 to 1, which is deemed as the value of the corresponding state feature. The same technique applies to all dimensions of every state.

To draw a conclusion over the statistical significance of any observed performance differences, all experiment results in this section are averaged over 30 independent runs. During each run, after completing every 50 learning episodes, the learned policies will be further evaluated on 25 independent testing episodes. We set the total number of learning episodes to 1500, consistently for both the Cart Pole problem and the Heating Coil problem.

4.2 Experiment Results

Results on the Cart Pole problem

Figure 1 compares the learning performance of five algorithms on the Cart Pole problem in terms of the average number of steps that the pole can be balanced continuously. Noticeably B-EMCE-PG significantly outperformed other competing algorithms, confirming that Algorithm 1 is highly effective at solving this problem. Especially, our results indicate that maximizing behavior diversity measured through CE can help improve learning performance, in comparison to B-EM-PG. On the other hand, the perceived performance difference in between B-EM-PG and EM-PG cannot be verified as statistically significant (p-value equals to 0.37). This is due to the large variation in learning performance across different

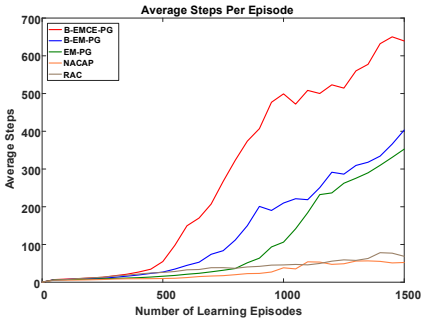

Figure 1: The average number of steps achieved per testing episode by B-EMCE-PG, B-EM-PG, EM-PG, RAC and NACAP on the Cart Pole problem.

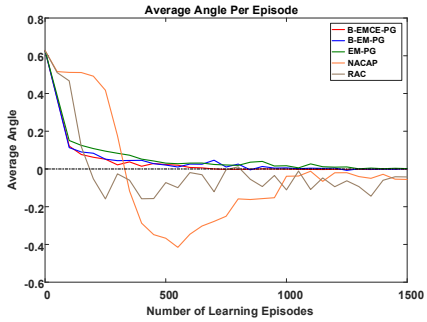

Figure 2: The average angle (radian) achieved per testing episode by B-EMCE-PG, B-EM-PG, EM-PG, RAC and NACAP on the Cart Pole problem.

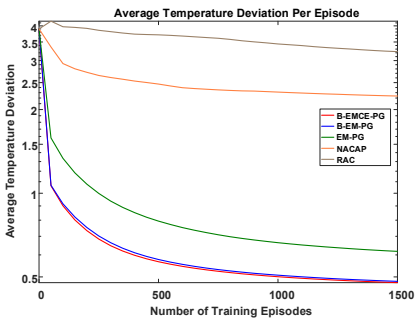

Figure 3: The average temperature deviation obtained per testing episode by B-EMCE-PG, B-EM-PG, EM-PG, RAC and NACAP on the Heating Coil problem.

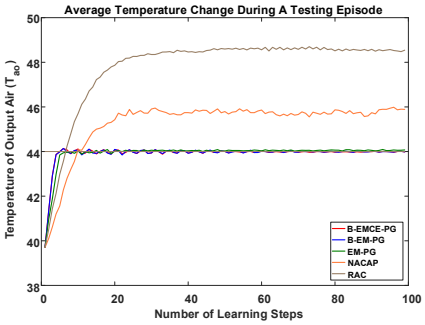

Figure 4: The average temperature change during a testing episode, after learning 1500 episodes by B-EMCE-PG, B-EM-PG, EM-PG, RAC and NACAP on the Heating Coil problem.

runs. Despite of that, both B-EM-PG and EM-PG performed consistently better than RAC and NACAP, suggesting that it may be more effective to update policy parameters according to EM-based PG defined in Subsection 2.3 than through the traditional PG derived directly from the cumulative rewards. The observed performance differences are also reflected in the average pole angles achieved by the five algorithms during a testing episode, as depicted in Figure 2. Apparently, when the average pole angle is closer to the up-right position, indicated by the black dotted line, the pole can be continuously balanced for longer time. In fact, B-EMCE-PG reached eventually an average pole angle of 4.58×10^{-4}. In this case, the only reason for a testing episode to stop is because the cart went beyond its boundaries.

Results on the Heating Coil Problem

The learning performance on the Heating Coil problem is evaluated through the average temperature deviation from 44° during a testing episode. As shown in Figure 3, B-EMCE-PG and B-EM-PG clearly outperformed EM-PG, RAC, and NACAP, agreeing with our expectation that Bayesian learning techniques are effective for RL. Meanwhile, EM-PG outperformed RAC and NACAP, indicating further that maximizing the performance lower bound in Subsection 2.3 is often a good choice for RL algorithms. Although B-EMCE-PG appears to be slightly better than B-EM-PG, the performance difference cannot be verified statistically. This observation suggests that the learner's behavioral diversity may not significantly affect its performance on the Heating Coil problem. Besides Figure 3, after 1500 learning episodes have been completed, the average temperature change during a testing episode is further presented in Figure 4. It is clear in this figure that B-EMCE-PG, B-EM-PG, and EM-PG can solve the problem highly effectively, whereas RAC and NACAP failed to control the target air temperature with reasonable accuracy [Peng *et al.*, 2016].

5 Conclusions

In this paper we focused on developing PS techniques for effective RL. Motivated by the fact that Bayesian learning techniques can be more effective than pure sampling-based methods, two possibilities of algorithmic improvement have been successfully explored in this paper. Particularly, we have developed a new Bayesian technique to approximate a different type of policy gradient derived from the lower bound of the learning performance. Another effective Bayesian technique has also been developed to guide the updating of policy parameters in the direction of maximizing the learner's behavioral diversity. Both Bayesian techniques have been jointly utilized to build a new and effective Bayesian RL algorithm. The experiment results showed that the benefits of encouraging behavioral diversity in our algorithm varies on different RL problems. This observation might correspond to the number of different solutions that these problems have and requires further efforts for verification.

Looking into the future, it is interesting to study more possible use of Bayesian techniques for effective RL, e.g. solving partially observable RL problems. It is also interesting to develop new algorithms that can more easily cope with non-deterministic state transitions. Meanwhile, the choice of kernel functions in GP models could play an important role for the overall learning performance and hence deserves further investigation.

References

[Anderson *et al.*, 1997] Charles W Anderson, Douglas C Hittle, Alon D Katz, and R Matt Kretchmar. Synthesis of reinforcement learning, neural networks and PI control applied to a simulated heating coil. *Artificial Intelligence in Engineering*, 11(4):421–429, 1997.

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

[Berger *et al.*, 1988] J. O. Berger, R. L. Wolpert, J. J. Bayarri M. H., DeGroot, B. M. Hill, D. A. Lane, and L. LeCam. The likelihood principle. *Lecture notes-Monograph series*, 6, 1988.

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

[Chen *et al.*, 2015] G. Chen, C. Douch, and M. Zhang. Using learning classifier systems to learn stochastic decision policies. *IEEE Transactions on Evolutionary Computation*, 19(6):885–902, 2015.

[Dayan and Hinton, 1997] P. Dayan and G. E. Hinton. Using expectation-maximization for reinforcement learning. *Neural Computation*, 9(2):271–278, 1997.

[Deisenroth and Rasmussen, 2011] M. Deisenroth and C. E. Rasmussen. Pilco: A model-based and data-efficient approach to policy search. In *Proceedings of the 28th International Conference on machine learning (ICML-11)*, pages 465–472, 2011.

[Duan *et al.*, 2016] Yan Duan, Xi Chen, John Schulman, and Pieter Abbeel. Benchmarking Deep Reinforcement Learning for Continuous Control. *arXiv*, 2016.

[Engel and Ghavamzadeh, 2007] Y. Engel and M. Ghavamzadeh. Bayesian policy gradient algorithms. In *Proceedings of the 2006 conference on advances in neural information processing systems*, volume 19, page 457, 2007.

[Ghanea-Hercock *et al.*, 2006] R. A. Ghanea-Hercock, F. Wang, and Y. Sun. Self-organizing and adaptive peer-to-peer network. *IEEE Transactions on Systems, Man, and Cybernetics–Part B*, 36(6):1230–1236, 2006.

[Ghavamzadeh *et al.*, 2016] M. Ghavamzadeh, Y. Engel, and M. Valko. Bayesian policy gradient and actor-critic algorithms. *Journal of Machine Learning Research*, 17(66):1–53, 2016.

[Hafner and Riedmiller, 2011] Roland Hafner and Martin Riedmiller. Reinforcement learning in feedback control. *Machine Learning*, 84(1-2):137–169, 2011.

[J. Kober *et al.*, 2013] Jens J. Kober, J. A. Bagnell, and J. Peters. Reinforcement learning in robotics: A survey. *The International Journal of Robotics Research*, 2013.

[Kakade, 2001] S. Kakade. A natural policy gradient, 2001.

[Kober and Peters, 2009] J. Kober and J. Peters. Policy search for motor primitives in robotics. In *Advances in neural information processing systems*, pages 849–856, 2009.

[Kormushev *et al.*, 2010] P. Kormushev, S. Calinon, and D. G. Caldwell. Robot motor skill coordination with em-based reinforcement learning. In *The 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3232–3237. IEEE, 2010.

[Levine and Koltun, 2013] Sergey Levine and Vladlen Koltun. Guided policy search. In *ICML (3)*, pages 1–9, 2013.

[Mohamed and Rezende, 2015] S. Mohamed and D. J. Rezende. Variational information maximisation for intrinsically motivated reinforcement learning. In *Advances in Neural Information Processing Systems*, pages 2125–2133, 2015.

[O'Hagan, 1991] A. O'Hagan. Bayes-hermite quadrature. *Journal of Statistical Planning and Inference*, 29(3):245–260, 1991.

[Peng *et al.*, 2016] Yiming Peng, Gang Chen, Mengjie Zhang, and Shaoning Pang. Generalized Compatible Function Approximation for Policy Gradient Search. pages 615–622. Springer International Publishing, Cham, 2016.

[Peters and Schaal, 2008] J. Peters and S. Schaal. Natural actor-critic. *Neurocomputing*, pages 1180–1190, 2008.

[Peters, 2010] J. Peters. Policy gradient methods. *Scholarpedia*, 11(5), 2010.

[Salge *et al.*, 2014] C. Salge, C. Glackin, and D. Polani. Changing the environment based on empowerment as intrinsic motivation. *Entropy*, 16(5):2789–2819, 2014.

[Sutton and Barto, 1998] R. S. Sutton and A. G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, 1998.

[Sutton *et al.*, 2000] R. S. Sutton, D. McAllester, S. Singh, and Y. Mansour. Policy gradient methods for reinforcement learning with function approximation. In *Advances in Neural Information Processing Systems 12 (NIPS 1999)*, pages 1057–1063. MIT Press, 2000.

[Williams, 1992] R. J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3-4):229–256, 1992.

[Wissner-Gross and Freer, 2013] A. D. Wissner-Gross and C. E. Freer. Causal entropic forces. *Physical review letters*, 110(16):168702, 2013.