Theoretical Analysis of Efficiency and Robustness of Softmax and Gap-Increasing Operators in Reinforcement Learning

Tadashi Kozuno[1] **Eiji Uchibe**[2] **Kenji Doya**[1]
[1] Neural Computation Unit, Okinawa Institute of Science and Technology
[2] Department of Brain Robot Interface, ATR Computational Neuroscience Laboratories

Abstract

In this paper, we propose and analyze *conservative value iteration*, which unifies value iteration, soft value iteration, advantage learning, and dynamic policy programming. Our analysis shows that algorithms using a combination of gap-increasing and max operators are resilient to stochastic errors, but not to non-stochastic errors. In contrast, algorithms using a softmax operator without a gap-increasing operator are less susceptible to all types of errors, but may display poor asymptotic performance. Algorithms using a combination of gap-increasing and softmax operators are much more effective and may asymptotically outperform algorithms with the max operator. Not only do these theoretical results provide a deep understanding of various reinforcement learning algorithms, but they also highlight the effectiveness of gap-increasing operators, as well as the limitations of traditional greedy value updates by the max operator.

1 INTRODUCTION

The use of neural networks for value function approximation has enabled human-level performance of reinforcement learning (RL) in challenging tasks (Mnih et al. (2015)). This success owes to stable learning realized by combining experience replay and a target network that is periodically updated to a main neural network. This scheme is an approximation of a dynamic programming (DP) algorithm called value iteration (VI) (Bertsekas and Tsitsiklis (1996)).

Proceedings of the 22[nd] International Conference on Artificial Intelligence and Statistics (AISTATS) 2019, Naha, Okinawa, Japan. PMLR: Volume 89.

While VI works well when updates are exact, theoretical analysis shows that VI works poorly when they are inexact (Bertsekas and Tsitsiklis (1996); Munos (2005); Scherrer and Lesner (2012)). To improve VI, various DP algorithms have been proposed. For instance, soft value iteration (SVI) uses a Boltzmann policy and a *softmax operator* with inverse temperature β for value updates in place of the traditional "hard" max operator in VI (Fox et al. (2016); Haarnoja et al. (2017)). In contrast, advantage learning (AL) uses a *hard gap-increasing operator*, the hyper-parameter α of which controls the degree of its gap-increasing-ness (Baird III (1999); Bellemare et al. (2016)). Dynamic policy programming (DPP) uses a Boltzmann policy together with a *soft gap-increasing operator* with $\alpha = 1$ (Azar et al. (2012); Rawlik (2013)). These algorithms demonstrated experimental performance superior to that of VI.

However, *(i) there is currently almost no theoretical explanation for why they perform better than VI.* For example, there is no performance bound for SVI and AL, and while there is a performance bound for DPP, it contradicts experimental results that a finite β is optimal (Azar et al. (2012)). Due to the lack of theories, *(ii) roles of their hyper-parameters are unknown too.*

In order to address these questions, we propose and analyze conservative value iteration (CVI), which unifies the previous algorithms as summarized in Figure 1. By the analysis of CVI, we show the following:

1. (Theorem 1) Novel performance bounds for the previous algorithms.
2. (Theorem 1) Algorithms with a gap-increasing operator are noise-tolerant. α controls the trade-off between noise-tolerance and convergence rate.
3. (Theorem 2) Algorithms with a hard gap-increasing operator have almost the same error dependency as does VI.
4. (Theorem 4) Algorithms with a softmax operator are error-tolerant, but asymptotic performance

may be poor. β controls the quality of asymptotic performance.

5. (Theorem 4) Algorithms with a soft gap-increasing operator enjoy both noise-tolerance and error-tolerance, while avoiding poor asymptotic performance.

(Error-tolerance refers to the tolerance of algorithms to errors such as function approximation error, while noise-tolerance refers to the tolerance of algorithms to stochastic errors that may cancel if averaged.) To better understand why DP algorithms with a softmax operator have greater error-tolerance, we note CVI's connection to natural actor-critic (NAC), which implies that β corresponds to a learning rate of a policy in NAC. Thus, greater error-tolerance is naturally understood as a consequence of stable learning by a low learning rate and inhibited policy oscillation. These theoretical results not only provide a deep understanding of various RL algorithms, but also highlight the effectiveness of gap-increasing operators, as well as the limitations of traditional greedy value updates by the max operator.

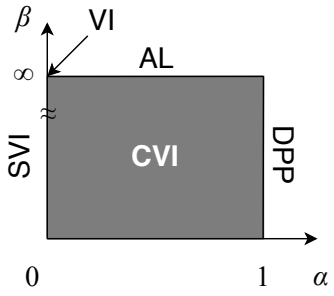

Figure 1: A summary of related DP algorithms. VI, SVI, AL and DPP correspond to the top left corner, left edge, top edge, and right edge, respectively. CVI unifies these algorithms.

2 REINFORCEMENT LEARNING

We consider infinite-horizon discounted Markov Decision Processes (MDPs) (Bertsekas and Tsitsiklis (1996)). An MDP is a tuple of $(\mathcal{S}, \mathcal{A}, P, r, \gamma)$, where $\mathcal{S}$ is the finite state space, $\mathcal{A}$ is the finite action space, $P : \mathcal{S}\times\mathcal{A}\times\mathcal{S} \to [0, 1]$ is the state transition probability, $r : \mathcal{S} \times \mathcal{A} \to [-r_{max}, r_{max}]$ is the reward function, and $\gamma \in [0, 1)$ is the discount factor.[1] We use s and a to denote a state and action, respectively. In particular, s_t and a_t (or S_t and A_t if they are random variables) are a state and action at time step t, respectively.

A policy is a probability distribution over actions conditioned by a state. We denote the Q-value function for a policy π by $Q^\pi(s, a) := \mathbb{E}^\pi[\sum_{t=0}^\infty \gamma^t r(S_t, A_t)|S_0 = s, A_0 = a]$, where the superscript π indicates that actions are selected according to π. The objective in RL is to find an optimal policy π^* that satisfies $Q^*(s, a) := Q^{\pi*}(s, a) = \sup_\pi Q^\pi(s, a)$ for any state-action pair $(s, a) \in \mathcal{S}\times\mathcal{A}$. This Q^* is called the optimal Q-value function. It is convenient to define the value and advantage function by $V^\pi(s) := \mathbb{E}^\pi[Q^\pi(s, A)]$ and $A^\pi(s, a) := Q^\pi(s, a) - V^\pi(s)$, respectively. The value and advantage function for an optimal policy are similarly denoted as $V^*(s) := V^{\pi^*}(s)$ and $A^*(s, a) := A^{\pi^*}(s, a)$. Note that Q^π and V^π are bounded by $V_{max} := r_{max}/(1 - \gamma)$.

We let $\mathcal{Q}$ and $\mathcal{V}$ denote sets of bounded functions over $\mathcal{S} \times \mathcal{A}$ and $\mathcal{S}$, respectively. Because both $\mathcal{S}$ and $\mathcal{A}$ are finite, they can be regarded as vector spaces over a field $\mathbb{R}$. A sum of $Q \in \mathcal{Q}$ and $V \in \mathcal{V}$ is defined by $(Q + V)(s, a) = Q(s, a) + V(s)$.

Expectation is a left-multiplication of a matrix with a vector. For example, an expected subsequent value of $V \in \mathcal{V}$ after taking an action a at state s is given by $(\boldsymbol{P}V)(s, a) := \sum_{s'\in\mathcal{S}} P(s'|s, a)V(s')$, where $\boldsymbol{P}$ is a $|\mathcal{S}||\mathcal{A}| \times \mathcal{S}$ matrix. Another example is $(\boldsymbol{\pi}Q)(s) := \sum_{a\in\mathcal{A}} \pi(a|s)Q(s, a)$, where we call $\boldsymbol{\pi}$ as a policy operator. When two operators, say $\boldsymbol{O}_1$ and $\boldsymbol{O}_2$, are applied to a function f consecutively, we omit parenthesis and write $\boldsymbol{O}_1\boldsymbol{O}_2 f$ instead of $\boldsymbol{O}_1(\boldsymbol{O}_2 f)$ for brevity.

We frequently use the following operators ($\mapsto$ means "maps to"): max operator $\boldsymbol{m} : Q \in \mathcal{Q} \mapsto \boldsymbol{m}Q \in \mathcal{V}$ such that $(\boldsymbol{m}Q)(s) := \max_{a\in\mathcal{A}} Q(s, a)$; the Bellman optimality operator $\boldsymbol{T} : Q \in \mathcal{Q} \mapsto r + \gamma\boldsymbol{P}\boldsymbol{m}Q \in \mathcal{Q}$; the softmax operator also known as the mellowmax operator $\boldsymbol{m}_\beta : Q \in \mathcal{Q} \mapsto \boldsymbol{m}_\beta Q \in \mathcal{V}$ such that

$$(\boldsymbol{m}_\beta Q)(s) := \frac{1}{\beta}\log\left(\frac{1}{|\mathcal{A}|}\sum_{a\in\mathcal{A}}\exp(\beta Q(s, a))\right),$$

where $\beta \in (0, \infty)$ is the inverse temperature and $|\mathcal{A}|$ is the number of actions (Asadi and Littman (2017)); a softened version of the Bellman optimality operator $\boldsymbol{T}_\beta : Q \in \mathcal{Q} \mapsto r + \gamma\boldsymbol{P}\boldsymbol{m}_\beta Q \in \mathcal{Q}$.

2.1 Dynamic Programming

One of the simplest DP algorithms for solving RL problems is value iteration (VI), which computes Q^* recursively by the following update rule (Bertsekas and Tsitsiklis (1996)): $Q_{k+1} := \boldsymbol{T}Q_k$, where the update is point-wise. As mentioned in Section 1, theoretical analysis implies that VI works poorly when updates

[1] Although the state space is assumed to be finite, we emphasize that our theoretical results can be extended to MDPs with a continuous state space.

are approximated. (See also Section 2.2.) There are several alternatives with empirically demonstrated superiority to VI under inexact update settings.

A softened version of VI called soft value iteration (SVI) is proposed in Fox et al. (2016) and Haarnoja et al. (2017). Its update rule is the following: $Q_{k+1} := \boldsymbol{T}_\beta Q_k$. Note that $\boldsymbol{m}_\beta$ is used in place of $\boldsymbol{m}$.

Advantage learning (AL) is proposed by Baird III (1999) and modified by Bellemare et al. (2016) for discrete time MDPs. Its update rule is the following:

$$Q_{k+1} := \boldsymbol{T}_{AL,\alpha} Q_k := \boldsymbol{T} Q_k + \alpha(Q_k - \boldsymbol{m} Q_k),$$

where $\alpha \in [0, 1]$. AL is VI with an additional advantage term $Q_k - \boldsymbol{m}Q_k$ that tries enhancing Q-value differences. We call $\boldsymbol{T}_{AL}$ as a *hard gap-increasing operator*.

Dynamic policy programming (DPP also known as Ψ-learning) is concurrently proposed in Azar et al. (2012) and Rawlik (2013). Its update rule is the following:[2]

$$\Psi_{k+1} := \boldsymbol{T}_{DPP,\beta} \Psi_k := \boldsymbol{T}_\beta \Psi_k + \Psi_k - \boldsymbol{m}_\beta \Psi_k,$$

where $\Psi_k - \boldsymbol{m_\beta} \Psi_k$ is analogous to $Q_k - \boldsymbol{m} Q_k$ except that it uses the softmax operator. We call this type of operator, in which the max operator of $\boldsymbol{T}_{AL}$ is replaced with the softmax operator, as *soft gap-increasing operators*. $\boldsymbol{T}_{DPP,\beta}$ is an instance with $\alpha = 1$. When we do not distinguish the hardness, we just call them gap-increasing operators.

2.2 Approximate Dynamic Programming

VI is a simple and efficient algorithm when $\mathcal{S} \times \mathcal{A}$ is small so that Q_k can be expressed by a table, and the true model of an environment (r and P) is given. However, those assumptions are rarely satisfied at the same time. Thus, the update is typically approximated as follows: suppose there are N tuples (s_i, a_i, r_i, s_i') of samples, where r_i and s_i' are samples of immediate reward and a subsequent state after taking an action a_i at state s_i. The approximated parameter update is given by $\theta_{k+1} = \arg\min_\theta \mathcal{L}_p(\theta|\theta_k)$, where θ_{k+1} stands for parameters of a function approximator after $k+1$ iterations, and $\mathcal{L}_p(\theta|\theta_k)$ is a loss function defined by

$$\sum_{i=1}^{N} \left| r_i + \gamma \max_{a' \in \mathcal{A}} Q(s_i', a'; \theta_k) - Q(s_i, a_i; \theta) \right|^p.$$

Typically $p = 1$ or 2 is used although any $p \in [1, \infty)$ is a reasonable choice. In deep RL, $Q(\cdot, \cdot; \theta_k)$ is called a target neural network.

[2]Precisely speaking, DPP by Azar et al. (2012) uses the Boltzmann-softmax explained in Appendix B, whereas Ψ-learning by Rawlik (2013) uses the mellowmax. Theorem 1 shows that any policy operator greedier than the mellowmax works.

Approximated updates inevitably involve errors. To analyze how they affect learning, error functions that abstractly express errors are frequently used (Munos (2005, 2007); Farahmand (2011); Scherrer et al. (2012); Scherrer and Lesner (2012)). Let us take VI as an example. Under a non-exact setting, VI's update rule is abstractly written as $Q_{k+1} := \boldsymbol{T} Q_k + \varepsilon_k$, where $\varepsilon_k \in \mathcal{Q}$ is the error function at the k-th iteration. It includes, but is not limited to sample estimation error of $\boldsymbol{T}Q_k$ and function approximation error.

A typical way to analyze DP algorithms is showing an upper bound (expressed with $\varepsilon_k, k = 0, 1, \ldots, K$) of

$$\|Q^* - Q^{g_K}\|_{\rho,p} := \left(\mathbb{E}_\rho \left|Q^*(S, A) - Q^{g_K}(S, A)\right|^p\right)^{1/p},$$

where g_K is a greedy policy with respect to Q_K, $p \in [1, \infty) \cup \{\infty\}$ (when $p = \infty$, l_∞-norm $\|Q\|_\infty := \max_{s,a} |Q(s, a)|$ is used), and $(S, A) \in \mathcal{S} \times \mathcal{A}$ is sampled from ρ that specifies the importance of states and actions. One of the natural choices for ρ is an initial state-action probability distribution. We call such an upper bound as l_p-norm performance bound.

For VI, the following simple performance bound is known (Scherrer and Lesner (2012)):

$$\|Q^* - Q^{g_K}\|_\infty \leq 2\gamma^{K+1} V_{max} + \frac{2\gamma}{1-\gamma} \mathcal{E}_K, \quad (1)$$

where $\mathcal{E}_K := \sum_{k=0}^{K-1} \gamma^k \|\varepsilon_{K-k-1}\|_\infty$. It shows two reasons why VI is prone to update errors. First, a sum of error terms $\mathcal{E}_K$ is divided by $1 - \gamma$, which is typically close to 0. As a result, VI shows a high error dependency. Second, because $\mathcal{E}_K$ is a sum of $\|\varepsilon_k\|_\infty$, it may be large even if $\varepsilon_l(s, a), l = 1, 2, \ldots$ are random variables such that $\sum_l \varepsilon_l(s, a) = 0$. Scherrer and Lesner (2012) proved that the error term of the performance bound is not improvable.

3 CONSERVATIVE VI

DP algorithms explained in Section 2.1 perform better than VI under a non-exact update setting. However, their superiority is shown mostly by experiments rather than theoretical analysis. Although a performance bound for DPP is given in Azar et al. (2012), it implies that $\beta = \infty$ is optimal. Nonetheless, Azar et al. (2012) noted that the best experimental results are obtained with a finite β. Thus, we currently lack a theoretical explanation of why those algorithms perform better than VI and on the roles of α and β. In this section, we propose CVI, which unifies the previous algorithms. We also provide its connection to other algorithms, including natural actor-critic to gain insight for theoretical analysis of CVI in Section 4, through which we will address those questions.

3.1 Derivation of CVI

We first derive CVI, whose update rule is the following:

$$\Psi_{k+1} = \boldsymbol{T}_\beta \Psi_k + \alpha \left(\Psi_k - \boldsymbol{m}_{\boldsymbol{\beta}} \Psi_k\right), \tag{2}$$

$$\pi_{k+1}(a|s) = \frac{\exp\left(\beta \Psi_{k+1}(s,a)\right)}{\sum_{b\in\mathcal{A}} \exp\left(\beta \Psi_{k+1}(s,b)\right)}, \tag{3}$$

where $\alpha \in [0,1]$ and $\beta \in (0,\infty) \cup \{\infty\}$. (When $\beta = \infty$, the mellowmax operator becomes the max operator Asadi and Littman (2017). Moreover, the Boltzmann policy reduces to a greedy policy.)

In policy search algorithms, Kullback–Leibler (KL) divergence and entropy are frequently used as policy update constraints (Williams and Peng (1991); Kakade and Langford (2002); Peters et al. (2010); Mnih et al. (2016)). A simple implementation of such algorithms could update a current policy π_k to a new policy π_{k+1} that maximizes for each state

$$\sum_a \pi_{k+1}(a|s)[r(s,a) + \gamma(\boldsymbol{P} W_{\pi_k}^{\pi_{k+1}})(s,a) + i_{\pi_k}^{\pi_{k+1}}(s)],$$

where $W_{\widetilde{\pi}}^{\pi}(s) := \sum_{t=0}^{\infty} \gamma^t \mathbb{E}^{\widetilde{\pi}} \left[r(S_t, A_t) + i_{\widetilde{\pi}}^{\pi}(S_t) | S_0 = s\right]$,

$$i_{\widetilde{\pi}}^{\pi}(s) := \sum_{a\in\mathcal{A}} \pi(a|s) \left[-\sigma \log \pi(a|s) - \tau \log \frac{\pi(a|s)}{\widetilde{\pi}(a|s)}\right],$$

$\tau \in [0,\infty)$ and $\sigma \in [0,\infty)$. This $W_{\widetilde{\pi}}^{\pi}$ can be understood as the state value function of π augmented with an entropy bonus and a KL divergence penalty. Note that when a policy is optimal, the update stops, i.e., KL penalty is 0. Thus, an optimal policy maximizes

$$\sum_{t=0}^{\infty} \gamma^t \mathbb{E}^{\pi} \left[r(S_t, A_t) - \sigma \log \pi(S_t, A_t) | S_0 = s\right],$$

which is the maximum entropy RL objective. (For more references and details, see Haarnoja et al. (2017, 2018) and their section on related work). Therefore, an optimal policy is a policy that collects many rewards while keeping its entropy not too low.

It is a conservative analogue to policy iteration, while an analogue to VI is repeatedly updating a value function W_{k+1} to

$$W_{k+1} := \boldsymbol{\pi}_{k+1} \left(r + \gamma \boldsymbol{P} W_k\right) + i_{\pi_k}^{\pi_{k+1}}, \tag{4}$$

where a policy π_{k+1} is a solution of the following optimization problem:

$$\underset{\pi}{\text{maximize}}\ \mathbb{E}^{\pi} \left[r(s,A) + \gamma \left(\boldsymbol{P} W_k\right)(s,A) + i_{\pi_k}^{\pi}(s)\right]$$

subject to $\sum_a \pi_{k+1}(a|s) = 1$ and $1 \geq \pi_{k+1}(a|s) \geq 0$, where variables are $\{\pi(a|s) | (s,a) \in \mathcal{S} \times \mathcal{A}\}$.

π_{k+1} can be analytically obtained as in Azar et al. (2012). Indeed, the objective function is differentiable and concave. Furthermore, both equality and inequality constraints are linear. Thus, the following solution is reached by solving simultaneous equations obtained from the Karush-Kuhn-Tucker condition:

$$\pi_{k+1}(a|s) = \frac{\pi_k(a|s)^\alpha \exp\left(\beta \left(r + \gamma \boldsymbol{P} W_k\right)(s,a)\right)}{Z(s)}, \tag{5}$$

where $\alpha := \tau/(\tau+\sigma)$, $\beta := 1/(\tau+\sigma)$, and $Z(s) := \sum_{a\in\mathcal{A}} \pi_k(a|s)^\alpha \exp\left(\beta \left(r + \gamma \boldsymbol{P} W_k\right)(s,a)\right)$ is a partition function. Using this expression, we obtain

$$W_{k+1}(s) = \frac{1}{\beta} \log Z(s). \tag{6}$$

Accordingly, the analogue to VI can be implemented with update rules (5) and (6). This naive implementation requires memory-size of $|\mathcal{S}||\mathcal{A}| + |\mathcal{S}|$. However, it is possible to retrieve both π_{k+1} and W_{k+1} by storing

$$\Psi_{k+1}(s,a) := \left(r + \gamma \boldsymbol{P} W_k\right)(s,a) + \frac{\alpha}{\beta} \log \pi_k(a|s) \tag{7}$$

because we have $\pi_{k+1}(a|s) \propto \exp\left(\beta \Psi_{k+1}(s,a)\right)$ and $W_{k+1}(s) = \beta^{-1} \log \sum_{a\in\mathcal{A}} \exp\left(\beta \Psi_{k+1}(s,a)\right)$. Plugging back these expressions in (7), we obtain an update rule $\Psi_{k+1} = \boldsymbol{T}_\beta \Psi_k + \alpha \left(\Psi_k - \boldsymbol{m}_{\boldsymbol{\beta}} \Psi_k\right)$, where we added a constant to the update rule to use the mellowmax operator. (The addition has no effect on the policies.) Therefore, the analogue can be implemented with update rules (2) and (3) efficiently.

According to the definition, $\alpha \in [0,1]$ and $\beta \in (0,\infty)$, respectively. However, we allow $\beta = \infty$ in CVI.

3.2 Connection to Other Algorithms

CVI is equivalent to the previous algorithms explained in Section 2.1 for specific choices of α and β (also see Figure 1 for a pictorial summary):

- When $\tau = 0$ (no KL penalty), $\alpha = 0$ and $\beta = \sigma$. CVI becomes SVI.
- When $\sigma = 0$ (no entropy bonus), $\alpha = 1$ and $\beta = \tau$. CVI becomes DPP.
- When $\tau + \sigma \to 0$ while keeping $\tau/(\tau+\sigma)$ constant, $\alpha = \tau/(\tau+\sigma)$ and $\beta \to \infty$. A limit case of CVI becomes AL.

Moreover, CVI can be understood as a variant of the natural actor-critic (NAC) algorithm. Although we later provide a performance bound, which shows that a small β may be preferable, this understanding gives an intuitive explanation of why it is so.

To begin, we explain NAC. Suppose a policy π parameterized by θ and normalized state visitation frequency

$d^\pi(s) = (1-\gamma)\sum_{t=0}^{\infty}\gamma^t \Pr(S_t = s|\pi)$ under the policy. An objective function is given by $J(\theta) := \mathbb{E}^\pi[r(S,A)]$, where the expectation is over actions $A \sim \pi(\cdot|S;\theta)$ and states $S \sim d^\pi$. Natural gradient of $J(\theta)$ with respect to θ is given by

$$G(\theta)\sum_{s,a\in\mathcal{S}\times\mathcal{A}} d^\pi(s)\left(Q^\pi(s,a) - b(s)\right)\nabla_\theta \pi(a|s;\theta), \quad (8)$$

where $G(\theta)$ is the inverse Fisher information matrix and b is a baseline. The natural policy gradient algorithm updates a policy by using the natural gradient computed with an estimate of Q^π (Kakade (2001)).

When the natural policy gradient is combined with actor-critic, a critic may introduce bias in gradient estimates. The compatibility condition (Konda and Tsitsiklis (1999); Sutton et al. (2000)) states that a critic $A(s,a;w) \approx Q^\pi(s,a) - b(s)$ introduces no bias when $\nabla_w A(s,a;w) = \nabla_\theta \log\pi(a|s;\theta)$ is satisfied. When a linear function apporximator $w^T\phi(s,a)$ is employed for the critic, it can be rephrased as $\phi(s,a) = \nabla_\theta \log\pi(a|s;\theta)$. Peters and Schaal (2008) showed that for a linear critic satisfying the compatibility condition, the natural gradient (8) is just w.

Now, let us derive a variant of NAC that corresponds to CVI. Suppose an actor π and critic given by

$$\pi(a|s;\theta) \propto \exp\left(\theta^T\phi(s,a)\right) \quad (9)$$

and $A(s,a;w) := w^T\phi(s,a) - \sum_{b\in\mathcal{A}}\pi(b|s;\theta)w^T\phi(s,b)$, respectively. Note that the compatibility condition is met. Its $k+1$-th policy parameter update with forgetting is given by $\theta_{k+1} := \theta_k + \eta w_k - \xi\theta_k$, where η is a learning rate and ξ is a forgetting rate. To see the correspondence, let us associate these parameters with quantities in CVI's update. From (3) and (9), it is natural to associate $\theta_k^T\phi$ with $\beta\Psi_k$ and $w_k^T\phi$ with $\boldsymbol{T}_\beta\Psi_k - \beta\eta^{-1}(1-\xi)\boldsymbol{m}_{\boldsymbol{\beta}}\Psi_k$. (We explain its meaning later.) Then, the update of θ is equivalent to function updates $\beta\Psi_{k+1} := \beta\Psi_k + \eta\boldsymbol{T}_\beta\Psi_k - \beta(1-\xi)\boldsymbol{m}_{\boldsymbol{\beta}}\Psi_k - \xi\beta\Psi_k$, i.e.,

$$\Psi_{k+1} = \frac{\eta}{\beta}\boldsymbol{T}_\beta\Psi_k + (1-\xi)\left(\Psi_k - \boldsymbol{m}_{\boldsymbol{\beta}}\Psi_k\right).$$

Clearly, it coincides with CVI's value update (2) when $1-\alpha = \xi$ and $\beta = \eta$. Therefore, CVI corresponds to this variant of NAC with a learning rate β and a forgetting rate $1-\alpha$. This result suggests that a small β leads to stable learning due to a small learning rate.

Finally, let us consider the meaning of the association of $w_k^T\phi(s,a)$ to $\boldsymbol{T}_\beta\Psi_k - \beta\eta^{-1}(1-\xi)\boldsymbol{m}_{\boldsymbol{\beta}}\Psi_k$. In the derivation of CVI, we have seen that (ignoring a constant) $\boldsymbol{T}_\beta\Psi_k = r + \gamma\boldsymbol{P}W_k$, where W_k is a crude estimate of the state value function augmented with an entropy bonus and a KL divergence penalty. Thus, $w_k^T\phi(s,a)$ is a crude estimate of the Q-value function with the augmentation. (Augmenting the current reward and $\beta\eta^{-1}(1-\xi)\boldsymbol{m}_{\boldsymbol{\beta}}\Psi_k$ are negligible because they will cancel out in $A(s,a;w)$.)

4 THEORETICAL ANALYSIS

Questions about CVI understandably center on its performance guarantee and roles of hyper-parameters. Here, we provide CVI's performance bounds, which also serve as the performance bounds for the algorithms in Section 2.1 unified by CVI. All proofs are deferred to the Appendix due to page limitations.

We use error functions explained in Section 2.2. Concretely, (approximate) CVI's update is given by

$$\Psi_{k+1} := \boldsymbol{T}_\beta\Psi_k + \alpha\left(\Psi_k - \boldsymbol{m}_{\boldsymbol{\beta}}\Psi_k\right) + \varepsilon_k, \quad (10)$$

where ε_k is an error function at iteration k. A policy is given in the same way as (3), that is,

$$\pi_{k+1}(a|s) \propto \exp\left(\beta\Psi_{k+1}(s,a)\right). \quad (11)$$

Just for simplicity, we assume that $\Psi_0(s,a) = 0$ throughout this section. However, it is not essential.

We will show l_p-norm performance bounds for CVI. They involve concentrability coefficients we now define (Munos (2005, 2007); Farahmand (2011)). Suppose a sequence of policies $\pi_0,\dots,\pi_t$ and probability distributions ρ,ν over $\mathcal{S}\times\mathcal{A}$. Let ρP^{π_0} be a probability distribution

$$\rho P^{\pi_0}(s_1,a_1) := \mathbb{E}_{(S_0,A_0)\sim\rho}\left[\pi_0(a_1|s_1)P(s_1|S_0,A_0)\right].$$

In other words, $\rho P^{\pi_0}(s_1,a_1)$ is an expected probability of a state-action pair (s_1,a_1) at time step 1. We recursively define a probability distribution $\rho P^{\pi_0}\cdots P^{\pi_t} := (\rho P^{\pi_0}P^{\pi_1}\cdots P^{\pi_{t-1}})P^{\pi_t}$. A concentrability coefficient is defined as

$$c(\rho,\nu;\pi_0,\dots,\pi_t) := \left\|\frac{\rho P^{\pi_0}\cdots P^{\pi_t}}{\nu}\right\|_{\nu,2},$$

where $\rho P^{\pi_0}\cdots P^{\pi_t}/\nu$ is a function of importance sampling ratio $\rho P^{\pi_0}\cdots P^{\pi_t}(s,a)/\nu(s,a)$. When $\nu(s,a) = 0$ while $\rho P^{\pi_0}\cdots P^{\pi_t}(s,a) \neq 0$ for some state-action pair, $c(\rho,\nu;\pi_0,\dots,\pi_t)$ is defined to be ∞.

4.1 α Controls a Trade-Off between Noise-Tolerance and Convergence Rate

We derive a performance bound for CVI that explains the role of α. It can be derived for an algorithm more general than CVI. To succinctly present the theorem, we need some notations. A policy π is said to be greedier than a policy π' (or an operator $\boldsymbol{\pi'}$) with respect to

a function $Q \in \mathcal{Q}$ if $(\boldsymbol{\pi} Q)(s) \geq (\boldsymbol{\pi'} Q)(s)$ for each state s. If so, we write $\boldsymbol{\pi} \geq_Q \boldsymbol{\pi'}$. The general algorithm has the following update:

$$\Psi_{k+1} := \boldsymbol{T}^{\nu_k} \Psi_k + \alpha (\Psi_k - \boldsymbol{\nu}_k \Psi_k) + \varepsilon_k, \tag{12}$$

where ν_k is a policy satisfying $\boldsymbol{\nu}_k \geq_{\Psi_k} \boldsymbol{m}_\beta$. Let ρ_k be a policy such that $\boldsymbol{\rho}_k \geq_{\Psi_k} \boldsymbol{\nu}_k$. We define short-hand notations for the following concentrability coefficients:

$$c^*_k := c(\rho, \nu; \overbrace{\pi^*, \ldots, \pi^*}^{k}),$$
$$c_{l,k} := c(\rho, \nu; \overbrace{\rho_K, \ldots, \rho_K}^{l}, \rho_K, \rho_{K-1}, \ldots, \rho_k).$$

We have the following theorem.

Theorem 1. *The general algorithm* (12) *has the following l_p-norm performance bound ($p \in [1, \infty)$):*

$$\|Q^* - Q^{\rho_K}\|_{\rho,p} \leq 2\gamma V_{max} \Gamma_K + B_K + 2\gamma\omega \mathcal{E}_{p,K}, \tag{13}$$

where $A_{K+1} := \sum_{k=0}^{K} \alpha^k$, $E_k := \sum_{l=0}^{k} \alpha^l \varepsilon_{k-l}$, $\omega := \dfrac{1}{1-\gamma}$

$$\Gamma_K := \frac{1}{A_{K+1}} \sum_{k=0}^{K} \gamma^k \alpha^{K-k},$$
$$B_K := \frac{\gamma(1-\gamma^K)\omega^2}{\beta A_{K+1}} \log|\mathcal{A}|,$$
$$C_k := \frac{\left((c^*_{k+1})^{1/p} + \sum_{l=0}^{\infty} \gamma^l \left(c^{1/p}_{l,k+1} + \gamma c^{1/p}_{l,k} \right) \right)}{2\omega},$$
$$\mathcal{E}_{p,K} := \sup_{\pi_K, \ldots, \pi_0} \sum_{k=0}^{K-1} \gamma^k C_k \left\| \frac{E_{K-k-1}}{A_{K+1}} \right\|_{\nu, 2p}.$$

Under the same assumptions and notations,

$$\|Q^* - Q^{\rho_K}\|_\infty \leq 2\gamma V_{max} \Gamma_K + B_K + 2\gamma\omega \mathcal{E}_{\infty,K}. \tag{14}$$

where $\mathcal{E}_{\infty,K} := \sum_{k=0}^{K-1} \gamma^k \left\| \dfrac{E_{K-k-1}}{A_{K+1}} \right\|_\infty$.

Remark 1. *Setting $\alpha = 1$, $\boldsymbol{\nu_k}$ to the mellowmax operator and ρ_k to the Boltzmann policy, yields a performance bound for DPP. In this case, $\Gamma_k = 2\gamma V_{max}(1-\gamma^{K+1})(1-\gamma)^{-1}K^{-1}$ is $1-\gamma$ times smaller than the corresponding term in the previous performance bound (Theorem 5 in Azar et al. (2012)) thanks to a new proof technique. (Their bound contains a mistake: $\|E_j\|_\infty$ in their bound must be multiplied by two.)*

Γ_K in Theorem 1 corresponds to γ^K in (1). Because Γ_K is the order of $O(\max\{\alpha, \gamma\}^K)$, the convergence rate is approximately the same as that of VI as long as $\alpha \leq \gamma$. However, when $\alpha = 1$ (i.e., DPP is used), Γ_K is the order of $O(K^{-1})$. Accordingly, its convergence is much slower.

B_K is an inevitable loss due to the use of softmax. It converges to $\gamma(1-\alpha)\omega^2\beta^{-1}\log|\mathcal{A}|$. Therefore, unless $\alpha = 1$ or $\beta = \infty$, B_K is not negligible. However, in Section 4.2, we show that a small β may be preferable despite this inevitable loss.

Theorem 1 shows that *a higher α leads to a greater noise-tolerance*. It states that $\|A^{-1}_{K+1} \sum_{l=0}^{k} \alpha^l \varepsilon_{k-l}\|_\infty$ essentially determines the loss $\|Q^* - Q^{\rho_K}\|_\infty$. Now, suppose that $\varepsilon_l(s,a)$ is sampled independently from a distribution with a mean of 0 and a standard deviation of 1 for any l, state s and action a. Then, a standard deviation of $A^{-1}_{K+1} \sum_{l=0}^{k} \alpha^l \varepsilon_{k-l}$ is given by $A^{-1}_{K+1}\sqrt{1+\alpha^2+\cdots+\alpha^{2k}}$. When $\alpha = 0.9$, it converges to approximately 0.23, which is four times smaller than 1. Therefore, an algorithm becomes tolerant to noise. Although $\varepsilon_k(s,a)$ is unlikely to satisfy the assumptions in reality, a similar result is expected in a model-free setting where errors contain noise stemming from the stochasticity of MDPs.

The greatest robustness can be attained when $\alpha = 1$, where the weighted average becomes a simple average. However, as argued above, the convergence is much slower in this case. Moreover, the effect of non-stochastic errors in the beginning of iterations may be sustained because of the form of E_{K-k-1}. For example, let us assume that ε_k are non-stochastic, and averaging is completely worthless. Then, when $\alpha = 1$,

$$\mathcal{E}_{\infty,K} \leq \frac{1}{1-\gamma} \sum_{k=0}^{K-1} \frac{\|\varepsilon_k\|}{K}.$$

Thus, the effect of errors in the beginning, e.g., $\|\varepsilon_0\|$ lingers. In contrast, when $\alpha = 0$, the effect of $\|\varepsilon_0\|$ decays with the rate γ^K.

The performance bounds are not meaningful if they are loose. The following theorem states that (14) is essentially not improvable when $\beta = \infty$.

Theorem 2. *When $\beta = \infty$, there exists an MDP and a sequence of ε_k satisfying the following: for any real value $\delta \in (0, \infty)$, there is a positive integer L such that*

$$2\gamma V_{max}\Gamma_K + 2\gamma\omega\mathcal{E}_{\infty,K} \leq \|Q^* - Q^{\rho_K}\|_\infty + \delta \tag{15}$$

holds for any $K \geq L$.

Suppose that there is a performance bound b_K that is smaller than $2\gamma V_{max}\Gamma_K + 2\gamma\omega\mathcal{E}_{\infty,K}$. Then, (15) states that $b_K < 2\gamma V_{max}\Gamma_K + 2\gamma\omega\mathcal{E}_{\infty,K} \leq b_K + \delta$ for a large enough K. In other words, the difference between b_K and our performance bound (14) is within $(0, \delta]$, and our bound is arbitrarily close to b_K.

From (15), when $\beta = \infty$, there exists an MDP in which the following holds:

$$\limsup_{K\to\infty} \|Q^* - Q^{\rho_K}\|_\infty = \frac{2\gamma}{1-\gamma}\bar{\varepsilon}, \tag{16}$$

where $\bar{\varepsilon} := \limsup_{K\to\infty} \mathcal{E}_{\infty,K}$. Note that it generalizes a simple performance bound (1) for VI in a sense that ε_{K-k-1} is replaced with a moving average of errors E_{K-k-1}/A_{K+1}. The important implication is that the error dependency of algorithms with a hard gap-increasing operator is almost the same as that of VI.

4.2 β controls the asymptotic performance

Theorem 1 states that $\beta = \infty$, i.e., algorithms with a hard gap-increasing operator are the optimal choice. However, there is experimental evidence that a finite β leads to better results (Azar et al. (2012); Fox et al. (2016); Haarnoja et al. (2017)). Furthermore, the connection of CVI and NAC implies that a small β leads to a stable learning while $\beta = \infty$ causes instability. In this section, we provide performance bounds for CVI that show benefits of setting β to a finite value. In particular, the performance bounds show that algorithms with the softmax operator may overcome the limitation of the hard gap-increasing operator (or greedy value updates) explained in the end of Section 4.1.

The following proposition provides a bound of KL divergence between π_k and π_{k-1}. It is utilized in a novel form of performance bounds following the proposition.

Proposition 3. *If $\|\varepsilon_k\| \le \varepsilon$ holds for any integer $k \in \{1, 2, \ldots\}$, a sequence of CVI's policies $\pi_0, \ldots, \pi_K$ in (11) satisfies $\max_s D_{KL}(\pi_K(\cdot|s)|\pi_{K-1}(\cdot|s)) \le \delta_K$, where δ_K is*

$$\delta_K := 4\beta\left(\frac{1-\gamma^K}{1-\gamma}\varepsilon + r_{max}\sum_{k=0}^{K-1}\alpha^k\gamma^{K-k-1}\right).$$

To succinctly state the theorem, we need the following short-hand notation for concentrability coefficients:

$$d_{l,k} := c(\rho, \nu; \overbrace{\pi_K, \ldots, \pi_K}^{l}, \pi_K, \pi_{K-1}, \ldots, \pi_k)$$

With this notation, we have the following theorem.

Theorem 4. *If $\|\varepsilon_k\| \le \varepsilon$ holds for any integer $k \in \{1, 2, \ldots\}$, CVI's policy π_K in (11) has the following l_p-norm performance bound ($p \in [1, \infty)$):*

$$\|Q^* - Q^{\pi_K}\|_{\rho,p} \le 2\gamma V_{max}\Gamma_K + (1-\gamma)B_K + 2\gamma\mathcal{E}'_{p,K} + \frac{\sqrt{2}\gamma^2 V_{max}}{1-\gamma}\sum_{k=0}^{K-1}\gamma^k\delta_{K-k}^{1/2}, \tag{17}$$

where

$$D_k := \frac{\left(c_{k+1}^*\right)^{1/p} + d_{0,k}^{1/p}}{2},$$
$$\mathcal{E}'_{p,K} := \sup_{\pi_K,\ldots,\pi_0}\sum_{k=0}^{K-1}\gamma^k D_k\left\|\frac{E_{K-k-1}}{A_{K+1}}\right\|_{\nu,2p}.$$

Under the same assumptions and notations,

$$\|Q^* - Q^{\pi_K}\|_\infty \le 2\gamma V_{max}\Gamma_K + (1-\gamma)B_K + 2\gamma\mathcal{E}_{\infty,K} + \frac{\sqrt{2}\gamma^2 V_{max}}{1-\gamma}\sum_{k=0}^{K-1}\gamma^k\delta_{K-k}^{1/2}. \tag{18}$$

Remark 2. *By taking the minimum of the bounds (13) and (17), we obtain a bound that is clearly no worse than both bounds.*

To understand Theorem 4, let us compare (14) with (18). Their major differences are the following: (i) $\mathcal{E}_{\infty,K}$ is multiplied by 2γ in (18), whereas it is multiplied by $2\gamma\omega$ in (14). (ii) There is an additional term $const.\sum_{k=0}^{K-1}\gamma^k\delta_{K-k}^{1/2}$ in (18). (iii) B_K in (18) is multiplied by $1-\gamma$.

The first difference indicates that *algorithms using the softmax operator are error-tolerant*. In Section 4.1, we explained that gap-increasing operators make algorithms noise-tolerant. However, if errors are not noise, the argument is nullified. In contrast, algorithms using the softmax operator have great tolerance to any type of error. The price to pay for this tolerance is the second difference, which decreases monotonically in β. Thus, a small β leads to better performance. Note that a small β results in the increase of B_K. To compensate for it, α must be large enough. Therefore, the use of the softmax operator alone is not sufficient.

To further understand Theorem 4, let us consider a simple case where $\alpha = 1$. Then, (18) yields

$$\limsup_{K\to\infty} \|Q^* - Q^{\pi_K}\|_\infty \le 2\gamma\bar{\varepsilon} + const.\beta^{1/2}, \tag{19}$$

where $\bar{\varepsilon} := \limsup_{K\to\infty}\mathcal{E}_{\infty,K}$. On the other hand, (14) yields the same bound as (16). Note that $\bar{\varepsilon}$ is multiplied by $2\gamma\omega$ in (16), while it is multiplied by 2γ in (19). Typically, $1-\gamma$ is close to 0; hence, (19) shows that DPP actually has much less error dependency when β is finite. β controls how closely the asymptotic performance approaches the optimal one.

This result can be understood from two perspectives. First, as explained in Section 3, $\tau = \alpha/\beta$ is the coefficient of the KL divergence constraint. Therefore, as β decreases, τ increases, which results in a smaller KL divergence, as shown in Proposition 3. A small β results in an increase of $\sigma = (1-\alpha)/\beta$. Therefore,

the entropy of a policy obtained with CVI increases, leading to a loss of performance expressed by B_K.

Second, a small learning rate in NAC results in robust learning. As we explained, CVI can be understood as a variant of NAC in which β is a learning rate and $1-\alpha$ is a forgetting rate. Therefore, a small β is expected to lead to a better asymptotic performance. On the other hand, a small α results in a reduced performance.

In addition, Theorem 4 shows another benefit of a finite β: concentrability coefficients D_k is better than C_k (when π_k is used in (14)). To see this, note that C_k contains $\sum_{l=0}^{\infty} \gamma^l c_{l,k}^{1/p} = \sum_{l=0}^{\infty} \gamma^l d_{l,k}^{1/p}$, which clearly satisfies

$$\sum_{l=0}^{\infty} \gamma^l d_{l,k}^{\frac{1}{p}} \geq d_{0,k}^{\frac{1}{p}}$$

As a consequence, $D_k = \infty$ implies $C_k = \infty$. Furthermore, it is possible to construct an example in which D_k is finite, but C_k is infinite. In this sense, D_k in (17) is better than C_k.

Finally, we note that $\alpha \in [0, 1)$ together with a finite β forces a policy π_k to be stochastic. As a result, concentrability coefficients of CVI with such α and β before taking $\sup_{\pi_K,\ldots,\pi_0}$ are expected to be smaller compared to algorithms with either $\alpha = 1$ or $\beta = \infty$. However, our analysis fails in capturing it.

5 Related research

A special case of CVI with $\alpha = 0$ is SVI (Fox et al. (2016); Haarnoja et al. (2017)). It is argued that an appropriately set β avoids overestimation of Q-values. It explains why a finite β works well from the perspective of sample estimation error. Our theoretical results add another explanation for why a finite β works well from the perspective of RL.

A special case of CVI with $\beta = \infty$ is AL (Baird III (1999); Bellemare et al. (2016)). AL lacks a theoretical guarantee under inexact update settings. We have provided performance bounds and explained how α affects the noise-tolerance of AL.

A special case of CVI with $\alpha = 1$ is DPP (Azar et al. (2012); Rawlik (2013)). A performance bound for DPP was provided by Azar et al. (2012). However, it states that $\beta = \infty$ is optimal despite experimental evidences that a finite β is optimal. Our work is the first paper that explains why DPP with a finite β is optimal.

CVI can be derived as a variant of NAC (Bhatnagar et al. (2009); Peters and Schaal (2008)). For policy iteration, a similar connection is shown in Wagner (2014), while for AL and DPP, this discussion seems to be the first.

6 Conclusion

SVI, AL, and DPP, all of which employ value-iteration-like, single-stage lookahead updates using the softmax operator and/or gap-increasing operator, demonstrated their superiority to VI (Baird III (1999); Azar et al. (2012); Rawlik (2013); Bellemare et al. (2016); Fox et al. (2016); Haarnoja et al. (2017)). However, they are not theoretically well understood. In this paper, we proposed and analyzed CVI that unifies them to explain their theoretical properties, such as performance guarantees under non-exact update settings and roles of their hyper-parameters.

Our analysis not only revealed the connection of those algorithms to NAC, but also provided two types of performance bounds: one without KL divergence and one with it.

The performance bounds without KL divergence improve the existing performance bound for DPP and comprise the first performance bound for SVI and AL. They also clarify the role of a hyper-parameter α in gap-increasing operators: α controls the trade-off between tolerance to stochastic error and convergence rate.

We also found that performance bounds without KL divergence are essentially tight as long as greedy value updates and a greedy policy are used. Furthermore, they imply that as long as greedy value updates and a greedy policy are used, tolerance of algorithms to non-stochastic errors are almost the same as that of VI.

Performance bounds with KL divergence show that the limitation by greedy value updates and a greedy policy can be overcome when the softmax operator is used. However, the softmax operator alone may lead to poor asymptotic performance, which is controlled by β. Algorithms with a soft gap-increasing operator enjoy both noise-tolerance and error-tolerance, while avoiding poor asymptotic performance.

The present paper is an important step toward understanding algorithms using the softmax operator and/or a gap-increasing operator.

Acknowledgement

This work was supported by JSPS KAKENHI Grant Numbers 16H06563 and 17H06042. We thank Dr. Steven D. Aird at Okinawa Institute of Science and Technology for editing and proofreading the paper. We are also grateful to reviewers for valuable comments and suggestions.

References

Asadi, K. and Littman, M. L. (2017). An alternative softmax operator for reinforcement learning. In *Proceedings of the Thirty-Fourth International Conference on Machine Learning*, pages 243–252.

Azar, M. G., Gómez, V., and Kappen, H. J. (2012). Dynamic policy programming. *Journal of Machine Learning Research*, 13(1):3207–3245.

Baird III, L. C. (1999). *Reinforcement Learning Through Gradient Descent*. PhD thesis, Carnegie Mellon University, Pittsburgh, PA, US.

Bellemare, M. G., Ostrovski, G., Guez, A., Thomas, P. S., and Munos, R. (2016). Increasing the action gap: New operators for reinforcement learning. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*.

Bertsekas, D. P. and Tsitsiklis, J. N. (1996). *Neuro-Dynamic Programming*. Athena Scientific, Nashua, NH, USA, 1st edition.

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

Farahmand, A.-m. (2011). *Regularization in reinforcement learning*. PhD thesis, University of Alberta, Edmonton, AB, Canada.

Fox, R., Pakman, A., and Tishby, N. (2016). Taming the noise in reinforcement learning via soft updates. In *Proceedings of the Thirty-Second Conference on Uncertainty in Artificial Intelligence*, pages 202–211.

Haarnoja, T., Tang, H., Abbeel, P., and Levine, S. (2017). Reinforcement learning with deep energy-based policies. In *Proceedings of the Thirty-Fourth International Conference on Machine Learning*, pages 1352–1361.

Haarnoja, T., Zhou, A., Abbeel, P., and Levine, S. (2018). Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *Proceedings of the Thirty-Fourth International Conference on Machine Learning*, pages 1856–1865.

Kakade, S. (2001). A natural policy gradient. In *Fourteenth Advances in Neural Information Processing Systems*, pages 1531–1538. MIT Press.

Kakade, S. and Langford, J. (2002). Approximately optimal approximate reinforcement learning. In *Proceedings of the Nineteenth AAAI Conference on Artificial Intelligence*, pages 267–274.

Konda, V. R. and Tsitsiklis, J. N. (1999). Actor-critic algorithms. In *Advances in Neural Information Processing Systems 12*, pages 1008–1014.

Mnih, V., Badia, A. P., Mirza, M., Graves, A., Lillicrap, T., Harley, T., Silver, D., and Kavukcuoglu, K. (2016). Asynchronous methods for deep reinforcement learning. In *Proceedings of The Thirty-Third International Conference on Machine Learning*, pages 1928–1937.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., Petersen, S., Beattie, C., Sadik, A., Antonoglou, I., King, H., Kumaran, D., Wierstra, D., Legg, S., and Hassabis, D. (2015). Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533.

Munos, R. (2005). Error bounds for approximate value iteration. In *Proceedings of the Twenty-Second AAAI Conference on Artificial Intelligence*, pages 1006–1011.

Munos, R. (2007). Performance Bounds in Lp norm for Approximate Value Iteration. *SIAM Journal on Control and Optimization*.

Peters, J., Mulling, K., and Altun, Y. (2010). Relative entropy policy search. In *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence*.

Peters, J. and Schaal, S. (2008). Natural actor-critic. *Neurocomput.*, 71(7-9):1180–1190.

Rawlik, K. C. (2013). *On probabilistic inference approaches to stochastic optimal control*. PhD thesis, The University of Edinburgh, Edinburgh, Scotland.

Scherrer, B., Ghavamzadeh, M., Gabillon, V., and Geist, M. (2012). Approximate modified policy iteration. In *Proceedings of the Twenty-Ninth International Conference on Machine Learning*.

Scherrer, B. and Lesner, B. (2012). On the use of non-stationary policies for stationary infinite-horizon markov decision processes. In *Advances in Neural Information Processing Systems 25*, pages 1826–1834.

Sutton, R. S., McAllester, D. A., Singh, S. P., and Mansour, Y. (2000). Policy gradient methods for reinforcement learning with function approximation. In *Advances in Neural Information Processing Systems 12*, pages 1057–1063.

Wagner, P. (2014). Policy oscillation is overshooting. *Neural Networks*, 52:43–61.

Williams, R. J. and Peng, J. (1991). Function optimization using connectionist reinforcement learning algorithms. *Connection Science*, 3(3):241–268.