VARIATIONAL ACTOR-CRITIC ALGORITHMS*,**, YUHUA ZHU1,*** AND LEXING YING2

Abstract. We introduce a class of variational actor-critic algorithms based on a variational formulation over both the value function and the policy. The objective function of the variational formulation consists of two parts: one for maximizing the value function and the other for minimizing the Bellman residual. Besides the vanilla gradient descent with both the value function and the policy updates, we propose two variants, the clipping method and the flipping method, in order to speed up the convergence. We also prove that, when the prefactor of the Bellman residual is sufficiently large, the fixed point of the algorithm is close to the optimal policy.

Mathematics Subject Classification. 90C40, 93E20.

Received August 2, 2021. Accepted January 13, 2023.

1. Introduction

Consider a discounted Markov Decision Process (MDP) $\mathcal{M} = (\mathcal{S}, \mathcal{A}, P, r, \gamma)$. Here $\mathcal{S}$ is the state space and $\mathcal{A}$ is the action space. $\Delta(\mathcal{S})$ and $\Delta(\mathcal{A})$ denote the set of probability distributions over $\mathcal{S}$ and $\mathcal{A}$, respectively. $P : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$ is the transition kernel, $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is the reward function, and $\gamma \in (0, 1)$ is the discounted factor. For each state-action pair $(s, a)$, we denote by $P(s'|s,a)$ the transition probability from state $s$ to state $s'$ given action $a$, $r(s,a)$ the immediate reward received at state $s$ with action $a$.

A policy $\pi : \mathcal{S} \to \Delta(\mathcal{A})$ represents an action selection rule, where $\pi(a|s)$ specifies the probability of taking action $a$ at state $s$. The state value function $V^\pi(s)$ is the expected discounted cumulative reward if one starts from an initial state $s$ and follows a policy $\pi$ with step $t = 0, 1, \ldots$:

$$V^\pi(s) = \mathbb{E}_{s_t \sim \pi(\cdot|s_t), a_t \sim \pi(\cdot|s_t), s_{t+1} \sim P(\cdot|s_t, a_t)} \left[ \sum_{t \geq 0} \gamma^t r(s_t, a_t) | s_0 = s \right].$$

1. Department of Mathematics and Halicioğlu Data Science Institute, University of California, San Diego, La Jolla, California, USA.
2. Department of Mathematics, Stanford University, Stanford, California, USA.

* The work of Y.Z. and L.Y. is partially supported by the U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing Research, Scientific Discovery through Advanced Computing (SciDAC) program.
** L.Y. is also supported by the National Science Foundation under award DMS-2011699.
*** Corresponding author: yuhuazhu@stanford.edu

© The authors. Published by EDP Sciences, SMAI 2023

This is an Open Access article distributed under the terms of the Creative Commons Attribution License (https://creativecommons.org/licenses/by/4.0), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.
The value function $V^\pi(s)$ also satisfies the Bellman equation [24],

$$
V^\pi(s) = \mathbb{E}_{\substack{a_t \sim \pi^t(\cdot|s_t) \\ s_{t+1} \sim P(\cdot|s_t,a_t)}} \left[ r(s_t, a_t) + \gamma V^\pi(s_{t+1}) | s_t = s \right].
$$

The state-action value function $Q^\pi(s, a)$, often referred as the $Q$-function, is the expected discounted cumulative reward if one takes action $a$ at initial state $s$: $Q^\pi(s, a) = \mathbb{E} \left[ \sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) | s_0 = s, a_0 = a \right]$. The two functions $V^\pi$ and $Q^\pi$ are related in the sense that $V^\pi(s) = \mathbb{E}_{a \sim \pi(\cdot|s)} Q^\pi(s, a)$.

A primary goal of reinforcement learning (RL) is to learn the optimal policy $\pi^*$ and its corresponding value function $V^*$. Among various approaches, the policy gradient methods have experienced significant advances recently, for example see [9, 10, 19, 21, 22, 27]. From the optimization perspective, a policy gradient method optimizes the following objective over policy $\pi$ with gradient updates

$$
\max_{\pi} \mathbb{E}_{s \sim \rho} V^\pi(s) \\
\text{s.t. } V^\pi(s) = \mathbb{E}_{\substack{a_t \sim \pi(\cdot|s_t) \\ s_{t+1} \sim P(\cdot|s_t,a_t)}} \left[ r(s_t, a_t) + \gamma V^\pi(s_{t+1}) | s_t = s \right], \\
(1.2)
$$

where $\rho(s)$ is a positive probability distribution. Policy gradient methods are often more convenient than the value based methods in the settings of continuous action space, high dimensional action space, and partially observed MDP [5, 18, 23]. It is also quite flexible to adopt various kinds of policy parameterizations in the policy gradient methods, which makes them powerful for both stochastic policies [3, 25] and deterministic policies [11, 23].

For policy gradient methods, entropy regularization is often included because it improves exploration by discouraging premature convergence to suboptimal deterministic policies [14, 17, 28]. More specifically, entropy regularization takes for example the following regularized maximization formulation:

$$
\max_{\pi} \mathbb{E}_{s \sim \rho} V^{\pi}_{\lambda}(s) \\
\text{s.t. } V^{\pi}_{\lambda}(s) = \mathbb{E}_{\substack{a_t \sim \pi(\cdot|s_t) \\ s_{t+1} \sim P(\cdot|s_t,a_t)}} \left[ r(s_t, a_t) - \lambda \log \pi(a_t|s_t) + \gamma V^{\pi}_{\lambda}(s_{t+1}) | s_t = s \right], \\
(1.3)
$$

Let $\pi^{\lambda}_* \equiv \pi^{\lambda}$ be the regularized optimal policy for (1.3). Note that $\pi^{0}_* = \pi^*$ (the non-regularized optimal policy) when $\lambda = 0$ but $\pi^{\lambda}_*$ is different from $\pi^*$ when $\lambda > 0$. In what follows, we shall abbreviate the optimal value functions $V^{\pi}_{\lambda}$ and $V^{\pi}$ as $V^{\lambda}_*$ and $V^*$, respectively.

The most direct way of solving the optimization problem (1.3) is to update the policy $\pi$ according to the gradient $\nabla_{\pi} \mathbb{E}_{s \sim \rho} V^{\pi}_{\lambda}(s)$. The calculation of $\nabla_{\pi} \mathbb{E}_{s \sim \rho} V^{\pi}_{\lambda}(s)$ however involves computing the exact value function $V^{\pi}_{\lambda}(s)$ or $Q^{\pi}_{\lambda}(s,a)$ under the current policy $\pi$. With an accurate approximation of the value function $V^{\pi}_{\lambda}(s)$, this gradient-based method can achieve a linear convergence rate [1, 4, 13]. However, the calculation of value function $V^{\pi}_{\lambda}(s)$ can be computationally intensive for large MDP problems. Especially in the model-free setting, a large data set is often needed in order to achieve a good approximation [12, 15].

To avoid the explicit computation of $V^{\pi}_{\lambda}(s)$, the actor-critic methods [10] have been widely studied in the literature [6, 7, 14, 26, 29] as a way to update the policy and value function at the same time. However, the convergence of the actor-critic algorithm is guaranteed only for two-timescale algorithms [30], where a smaller stepsize is used for the actor updates and a larger stepsize is used for the critic updates. The stabilities of the actor-critic algorithms are often sensitive to the choice of stepsizes [8].
Contributions. In this paper, we propose a new actor-critic method based on a variational formulation over the policy and the value function. Consider the optimization problem,

\[ \min_{V, \pi} E(V, \pi) = E_{s \sim \rho(s)} \left[ -V(s) + \frac{\beta}{2} (V(s) - E[r(s, a_t) + \gamma V(s_{t+1}) - \lambda \log(\pi(a_t|s_t)|s_t = s)]^2 \right], \quad (1.4) \]

where \( \rho \in \mathbb{R}^{|S|} \) can be any positive probability distribution and \( \beta > 0 \) is a positive constant. The objective function (1.4) consists two parts: the first is to maximize the value function, while the second is to minimize the Bellman residual. The variational structure ensures that the vanilla gradient descent almost surely converges to a local minimum without requirements on different stepsizes for \( V \) and \( \pi \) updates.

Besides, we pointed out that the vanilla gradient descent will lead to a direction increases \( V^*_\lambda \) at the initial stage because of the negative Bellman residuals. In order to improve the convergence speed of the vanilla gradient descent of (1.4), we further propose two variants. The first clipping method can be viewed as the gradient descent of the objective function with a non-Euclidean metric. The second clipping method further accelerates the convergence by continuously maximizing the value function in the right direction.

We prove that, when \( \lambda = 0 \) and the prefactor \( \beta \) is sufficiently large, the fixed point of the proposed algorithm is exactly the optimal policy \( \pi^* \). Furthermore, we prove that when \( \lambda > 0 \) i.e., in the regularized setting, the fixed point is close to the non-regularized optimal policy \( \pi^* \) for large \( \beta \) and small \( \lambda \).

Contents. The variational actor-critic algorithm is introduced in Section 2, where the clipping and the flipping methods are first presented in the model-based setting first (Sect. 2.1) and then in the model-free setting (Sect. 2.2). In Section 3, we study the fixed point of the algorithm for both non-regularized (Sect. 3.1) and the regularized (Sect. 3.2) objective functions. Several numerical experiments are reported in Section 4 to demonstrate the performance of the proposed algorithms.

2. Variational Actor-Critic

Section 2.1 presents the variational actor-critic algorithm in the model-based setting, where the value function \( V(s) \) is of the tabular form and the policy is parameterized with the soft-max function. In Section 2.2, we introduce the stochastic variational actor-critic algorithm in the model-free setting, which applies to the general case of nonlinear approximation to the policies and \( Q \)-functions.

2.1. Model-based setting

To simplify the discussions, we assume that both the state and action spaces are finite discrete sets. Consider the following minimization problem:

\[ \min_{V \in \mathbb{R}^{|S|}, \theta \in \mathbb{R}^{|S| \times |A|}} E(V, \pi) = -\rho^\top V + \frac{\beta}{2} \| (I - \gamma P^\pi) V - r^\pi + \lambda H(\pi_\theta) \|_\rho^2, \quad (2.1) \]

where \( \rho \in \mathbb{R}^{|S|} \) is a positive probability distribution, and \( \beta > 0 \) is a prefactor. The norm \( \| \cdot \|_\rho \) is defined by \( \|x\|_\rho^2 := \sum_{s \in S} x_s^2 \rho_s \). Here \( V = (V_s)_{s \in S} \) in (2.1) is an \(|S|\)-dimensional vector, and the policy \( \pi_\theta = \{ (\pi_\theta)_{sa} \}_{s \in S, a \in A} \) is an \(|S| \times |A|\) matrix. We assume the policy \( \pi_\theta \) is a soft-max function, i.e., for any pair \( (s, a) \in S \times A \),

\[ (\pi_\theta)_{sa} = \frac{e^{\theta_{sa}}}{\sum_{b \in A} e^{\theta_{sb}}}, \quad s \in S, a \in A. \]

Hereafter, we omit the subscript \( \theta \) of \( \pi_\theta \) for simplicity. The vector \( r^\pi = (r^\pi)_{s \in S} \in \mathbb{R}^{|S|} \) in (2.1) is the reward under policy \( \pi \) with the component \( r^\pi_s = \sum_a r_{sa} \pi_{sa} \), where \( r_{sa} \) is the immediate reward at \( (s, a) \). The matrix \( P^\pi \in \mathbb{R}^{|S| \times |S|} \) in (2.1) is the transition matrix under policy \( \pi \) with the entry \( P^\pi_{st} = \sum_a \pi_{sa} P^\pi_{st} \), where for each \( a \),

\[ P^\pi_{st} = \sum_a \pi_{sa} P^\pi_{st}. \]
the matrix $P^a \in \mathbb{R}^{[S] \times [S]}$ is the state transition matrix under action $a$. The vector $\mathcal{H}(\pi) = (\mathcal{H}(\pi_s))_{s \in S} \in \mathbb{R}^{[S]}$ in (2.1) is the entropy regularizer with the component $\mathcal{H}(\pi_s) = \sum_a \pi_{sa} \log \pi_{sa}$.

The minimization problem (2.1) is a relaxation of the maximization problem of $V_\lambda^\pi$ as in (1.3). Note that minimizing the first term $-\rho^\top V$ of the RHS of (2.1) has the same effect as maximizing $V$. On the other hand, the second term $\|I - \gamma P^\pi V - r^\pi + \lambda \mathcal{H}(\pi)\|_\rho^2$ of the RHS of (2.1) is the norm of the Bellman residual. As $V_\lambda^\pi = (I - \gamma P^\pi)^{-1}(r^\pi - \lambda \mathcal{H}(\pi_\theta))$, the minimization of the second term of the RHS of (2.1) leads $V$ to the value function $V_\lambda^\pi$. Thus, combining the two terms of the RHS of (2.1) yields that (2.1) maximizes the true value function $V_\lambda^\pi$ as in (1.3).

One approach for solving the minimization problem (2.1) is to update the $(V, \theta)$ pair following the gradients of the objective function. The gradients are given by

$$
\frac{\partial V}{\partial \theta} E = -\rho + \beta(\ell \rho - \gamma \sum_t P_{ts} \ell_t \rho_t)) \quad; \quad := G_{V_s},
$$

$$
(F^+ \nabla_\theta E)_{sa} = \rho_{sa} \beta \| \! - \gamma \sum_t P_{st}^a V_t - r_{sa} + \lambda \log \pi_{sa} \| + c_s \quad; \quad := G^{(0)}_{\theta,s},
$$

where $\ell(V, \pi)$ is an $|S|$-dimensional function denoting the Bellman residual

$$
\ell(V, \pi) = (I - \gamma P^\pi)V - r^\pi + \lambda \mathcal{H}(\pi).
$$

Here the natural gradient is used for the policy updates in (2.2) with the Fisher information matrix $F = \mathbb{E}_{s \sim \rho, a \sim \pi} [(\nabla_\theta \log \pi)(\nabla_\theta \log \pi)\top]$. The operator $F^+$ in (2.2) denotes the Moore-Penrose pseudoinverse of $F$ (see e.g. Appendix C.6 of [4] for the calculation of $F^+ \nabla_\theta E$). The vector $c = (c_s)_{s \in S} \in \mathbb{R}^{[S]}$ in (2.2) depends on state $s$ and is independent of action $a$. When the policy is represented by the soft-max function, the explicit form of $c$ does not influence the update of the policy. The vanilla gradient descent algorithm for the minimization problem (2.1) takes the form

$$
V_s^{k+1} = V_s^k - \eta_V G_{V_s}(V_s^k), \quad \pi_{sa}^{k+1} \propto \pi_{sa}^k e^{-\eta_\pi G_{\theta,s}^{(0)}(V_s^{k}, \pi^k)},
$$

(2.4)

where $\eta_V$ and $\eta_\pi$ are the learning rates.

Although we show in Section 3 that the above algorithm converges to a policy that is close to the optimal policy $\pi^*$, the trajectory towards the minimizer is often not optimal. When $\ell_s < 0$, the path from $\pi$ towards $\pi^*$ may detour if the algorithm is directed according to $G^{(0)}_{\theta,s}$. As shown in Figure 1, the error $\pi - \pi^*$ can increase at the initial stage of the algorithm. Intuitively, since $\nabla_\pi E = \nabla_\pi \| \ell \|_\rho^2$, the gradient $\nabla_\pi E$ of the objective function in $\pi$ serves to minimize the residual norm $\| \ell \|_\rho$ in the policy space. When $\ell < 0$, $V$ underestimates the true value function $V_\lambda^\pi$. Therefore, in order to reduce the residual $\| \ell \|_\rho$ in the policy space, $\nabla_\pi \| \ell \|_\rho^2$ will lead to a direction that reduces $V_\lambda^\pi$, which is undesirable. On the other hand, when $\ell$ is non-negative, $V$ overestimates $V_\lambda^\pi$. Hence, $\nabla_\pi \| \ell \|_\rho^2$ will lead to a direction that increases $V_\lambda^\pi$ and, therefore, reduces $\| \ell \|_\rho$, which is the desired direction.

Another way to understand this aforementioned detour is through the gradient of the objective function. Notice that the gradient $\partial_{\pi,s} E$ of the objective function $E(V, \pi)$ in $\pi$ has the same form as $G^{(0)}_{\theta,s}$ in (2.2). Therefore,

$$
-\partial_{\pi,s} E(V, \pi) = -\beta(\ell \odot \rho)^\top \partial_{\pi,s} [(I - \gamma P^\pi)(V - (I - \gamma P^\pi)^{-1}(r^\pi - \lambda \mathcal{H}(\pi)))]
$$

$$
= -\beta(\ell \odot \rho)^\top \partial_{\pi,s} [(I - \gamma P^\pi)(V - V_\lambda^\pi)]
$$

$$
= \beta(\ell \odot \rho)^\top (I - \gamma P^\pi) \partial_{\pi,s}(V_\lambda^\pi) + \beta \gamma(\ell \odot \rho)^\top \partial_{\pi,s}(P^\pi)(V - V_\lambda^\pi),
$$

(2.5)
Figure 1. The left plot shows the error $\pi^k - \pi^*$ from the vanilla gradient descent method (2.2), i.e., Algorithm 1 with $h^{(0)}(\ell_s) = \ell_s$. We test the algorithm for an MDP with 5 states and 2 actions, and set $\beta = 10, \lambda = 0$ and the learning rate $\eta_V = \eta_\pi = 1/(4\beta)$. The red line is the mean over the 100 simulations. The middle plot shows the existence of negative residuals. It plots 1 when there exist negative residuals at step $k$ and 0 when the residuals are non-negative at all states. The blue line is the mean value over the 100 simulations. The right plot collects the average lines of the left two plots in one figure. One can see from the left plot that the error increases at the initial stage and then decreases at the latter stage. From the middle plot, one can see that there always exists at least one negative residual among the 5 states at the initial stage and then the residuals become all non-negative at the latter stage. On the right plot, one can see that the error increase is closely related to the existence of negative residuals. When the residuals have fewer or no negative values, the error decreases rapidly as expected.

where $\ell \odot \rho$ is the entry-wise product of $\ell$ and $\rho$, i.e., $(\ell \odot \rho)_s = \rho_s \ell_s$. Note first that the second term of the RHS of (2.5) contains $(V - V_\lambda^\pi)$ and $(V - V_\lambda^\pi)$ is small because the objective function (2.1) pushes $V$ to the true value function $V_\lambda^\pi$ for sufficiently large $\beta$. In fact, for any fixed $\pi$, the local fixed point of the $V$ updates satisfies $G_{V_\pi} = 0$, where $G_{V_\pi}$ is defined in (2.2). This implies that

$$V - V_\lambda^\pi = V - (I - \gamma P_\pi)^{-1}(r - \lambda H(\pi)) = \frac{1}{\beta}(I - \gamma P_\pi)^{-1}[\hat{\rho} \odot [(I - \gamma P_\pi)^{-T}\rho]] \sim O\left(\frac{1}{\beta}\right),$$

where $\hat{\rho}_s = 1/\rho_s$. Hence, the second term of the RHS of (2.5) is of order $O(1/\beta)$. When $\beta$ is large, the gradient $-\partial_{\pi^k} E(V, \pi)$ is dominated by the first term $\beta(\ell \odot \rho)^T( I - \gamma P_\pi)\partial_{\pi^k}(V_\lambda^\pi)$, which is equivalent to $\beta\ell \rho(1 - \gamma)\partial_s(V_\lambda^\pi)$ in the one-dimensional case. Note that $\partial_s(V_\lambda^\pi)$ is the steepest ascent direction for maximizing $V_\lambda^\pi$. Therefore, when $\ell < 0$, the term $\beta\ell \rho(1 - \gamma)\partial_s(V_\lambda^\pi)$ is the opposite direction of the steepest ascent, which implies that the gradient descent algorithm based on $-\partial_s E$ does not move towards maximizing $V_\lambda^\pi$. This illustrates why the algorithm (2.4) can take a detour to the optimal policy $\pi^*$.

To address this issue, we propose two methods to improve the efficiency of the algorithm.

- The clipping method. The idea is to suppress $\ell_s$ when $\ell_s < 0$, i.e., the policy update is based on a clipping modification,

$$G_{\theta_{sa}}^{(1)} = \rho_s \beta \ell_s 1_{\ell_s > 0} \left[ -\gamma \sum_t P_{st}^a V_t - r_{sa} + \lambda \log \pi_{sa} \right].$$

This algorithm can be viewed as a gradient descent method for the optimization problem (2.2) with a metric $(id, 1_{\ell_s > 0} \cdot id)$ on $(V_s, \theta_{sa})$.
• The *flipping* method. The idea is to flip the sign of $\ell_s$ when $\ell_s < 0$, *i.e.*, the policy update is based on

$$G^{(2)}_{\theta_{sa}} = \rho_s \beta |\ell_s| \left[ -\gamma \sum_i P_{si}^\pi V_i - r_{sa} + \lambda \log \pi_{sa} \right].$$

From the analysis of (2.5), we see that the vanilla gradient descent with $G^{(0)}_{\theta_{sa}}$ in (2.2) would make the policy worse locally. Intuitively, the clipping method with $G^{(1)}_{\theta_{sa}}$ stops updating the policy when $\ell_s < 0$, while the flipping method further improves the policy $\pi$ because $G^{(2)}_{\theta_{sa}}$ always has the same direction as $\nabla_\pi V^\pi$.

Three different versions (vanilla, clipping, and flipping) of the variational actor-critic based on the objective function (2.1) can be summarized as follows:

$$V^{k+1} = V^K - \eta_V G_V(V^K, \pi^K), \quad \pi_{sa}^{k+1} \propto \pi_{sa}^k \exp(-\eta_\pi G^{(i)}_{\theta_{sa}}(V^K, \pi^K)), \quad i = 0, 1, 2,$$

where $G_V$ and $G^{(i)}_{\theta}$ represent the gradients with respect to $V$ and $\theta$,

$$G_V(V, \pi) = -\rho_s + \beta \left( \ell_s \rho_s - \gamma \sum_i P_{si}^\pi \ell_i t_i \right),$$

$$G^{(i)}_{\theta_{sa}} (V, \pi) = \beta \rho_s h^{(i)}(\ell_s) \left[ -\gamma \sum_i P_{si}^\pi V_i - r_{sa} + \lambda \log \pi_{sa} \right] + c_s, \quad i = 0, 1, 2,$$

and $h^{(i)}$ is defined as

$$h^{(0)} = x, \quad h^{(1)}(x) = x \mathbb{1}_{x > 0}, \quad h^{(2)}(x) = |x|.$$

Here $\ell = (\ell_s)_{s \in S}$ is the Bellman residual defined in (2.3). Under the model-based setting (*i.e.*, assuming that the transition dynamics is explicitly known), the algorithm is outlined in Algorithm 1.

**Algorithm 1** Variational actor-critic (model-based version)

- **Require:** $\eta_V, \eta_\pi$; learning rate; $\beta$: penalty constant;
- **Require:** $i$: $i = 0$ (vanilla gradient descent) or $i = 1$ (clipping) or $i = 2$ (flipping);
- 1: Random initialization of $V_0, \pi_0$
- 2: **while** $V, \theta$ do not converge **do**
- 3: $\ell \leftarrow (1 - \gamma P^\pi)V - r^\pi + \lambda H(\pi)$;
- 4: $V_s \leftarrow V_s - \eta_V (\rho_s + \beta (\ell_s \rho_s - \gamma \sum_i P_{si}^\pi \ell_i t_i))$;
- 5: $\pi_{sa} \leftarrow \pi_{sa} \exp \left[ -\eta_\pi \beta \rho_s h^{(i)}(\ell_s) (-\gamma \sum_i P_{si}^\pi V_i - r_{sa} + \lambda \log \pi_{sa}) \right]$, where $h^{(i)}$ is defined in (2.8);
- 6: $\pi_{sa} \leftarrow \frac{1}{\sum_b \pi_{sa}^b \pi_{sa}^b}$;
- 7: **end while**

We would like to point out that the three variants $G^{(i)}_{\theta_{sa}}$ for $i = 0, 1, 2$ coincide when $\ell_s \geq 0$. Lemma 3.2 demonstrates that $\ell_s$ is larger than 0 when $(V, \pi)$ achieves the fixed point of the algorithm. Therefore, the three variants $G^{(i)}_{\theta_{sa}}$ with $i = 0, 1, 2$ are different only at the initial stage of the optimization process and become the same at the latter stage with $\ell_s > 0$. The vanilla gradient descent might go to a worse policy first and then go to the direction that maximizes $V_0^\pi$; the clipping method might stop updating the policy until $V$ near its local fixed point with $\ell_s > 0$; the flipping method would go all the way along the direction maximizing $V_0^\pi$. Although the three variants converge to the fixed point with different dynamics, they eventually converge to the same fixed point.
2.2. Model-free setting

When the transition dynamics $P^s$ is unknown as in the model-free RL, one only has access to one (or multiple) off-policy trajectory $\{(s_t, a_t, r_t)\}_{t=1}^T$ generated by a behavior policy $\pi_b(s,a)$. Algorithm 1 can in principle be generalized to the model-free setting if one updates $(V, \pi)$ based on an unbiased estimate of the gradient (2.7) (see Appendix A for the stochastic algorithm in $V$-formulation). However, reweighting is necessary in order to correct the difference between the behavior policy $\pi_b$ and the target policy $\pi$ when approximating the term $P^s V$. The reweighting method, although unbiased, would cause instability in the process of SGD ([2, 20]).

It is instead preferred to use the $Q$-formulation as there is no need to correct the behavior policy. The stochastic algorithm in the $Q$-formulation is based on the following objective function:

$$
\mathbb{E}_{(s,a) \sim \rho} \left[ -Q(s,a) + \frac{\beta}{2} \left( Q(s,a) - \mathbb{E}_{s' \sim P^s(\cdot|s,a)} \left[ \gamma \sum_a (Q(s',a) - \lambda \log \pi(s',a) | s,a) - r(s,a) \right] \right)^2 \right],
$$

(2.9)

where $\rho > 0$ is the positive stationary distribution from behavior policy $\pi_b$. By using the $Q$-formulation, one can directly use the trajectory $\{(s_t, a_t, r_t)\}_{t=1}^T$ without reweighting.

When $Q(s,a,\omega)$ is parametrized by $\omega$ and $\pi(s,a,\theta)$ is parametrized by $\theta$, the updates of $(Q, \pi)$ are according to the following unbiased estimates of the gradients:

$$
(G_Q)_t = -\nabla_{\omega} Q^k(s_t, a_t) + \beta L_t \left( \nabla_{\omega} Q^k(s_t, a_t) - \gamma \sum_a \nabla_{\omega} Q^k(s_{t+1}, a) \pi^k(s_{t+1}, a) \right),
$$

$$
(G^i_\pi)_t = \beta \hat{h}^{(i)}(L_t) \left( -\gamma \sum_a (Q^k(s_{t+1}, a) - \lambda \log \pi^k(s_{t+1}, a) - \lambda \nabla_{\theta} \pi^k(s_{t+1}, a) \right),
$$

(2.10)

where $Q^k(s,a) = Q(s, a, \omega_k)$, $\pi^k(s,a) = \pi(s,a,\theta_k)$ and $L_t$ is the unbiased Bellman residual,

$$
L_t = Q^k(s_t, a_t) - r_t - \gamma \sum_a (Q^k(s_{t+1}, a) - \lambda \log \pi^k(s_{t+1}, a)) \pi^k(s_{t+1}, a).
$$

Here the next state $s_{t+1}'$ in $G_Q$ and $G_\pi$ needs to be uncorrelated with the next state $s_{t+1}$ in the trajectory. Since it is usually unrealistic to generate another independent sample at state $s_t$ with action $a_t$, the BFF algorithm is proposed in [31] to generate an approximate $s_{t+1}'$

$$
s_{t+1}' = s_t + (s_{t+2} - s_{t+1}).
$$

It is shown in [31] that when the underlying dynamics changes smoothly with respect to the actions and states, the BFF approximation is close to the independent sample in expectation. Furthermore, $\hat{h}^{(i)}(x)$ is defined as follows,

$$
\hat{h}^{(1)}(L_t) = L_t \mathbb{1}_{\hat{\ell}_{s_t} > 0}, \quad \hat{h}^{(2)}(L_t) = \left\{ \begin{array}{ll} L_t, & \hat{\ell}_{s_t} > 0 \\ -L_t, & \hat{\ell}_{s_t} < 0 \end{array} \right., \text{ where } \hat{\ell}_s = \frac{1}{|\{s_t = s\}|} \sum_{s_t = s} L_t.
$$

(2.11)

Note that one cannot directly apply the clipping or flipping function $h^{(i)}$ defined in (2.8) on the stochastic Bellman residual $L_t$ because $\mathbb{E}[\hat{h}^{(i)}(L_t)|s_t = s] \neq h^{(i)}(\ell_s)$. Instead, $h^{(i)}(\ell_s)$ is estimated in two steps: first, one approximates the Bellman residual $\ell_s$ with $\hat{\ell}_s$, and then $L_t$ is suppressed or flipped according to the value of $\hat{\ell}_s$. The stochastic algorithm for the $Q$-formulation is summarized in Algorithm 2.
Remark 2.1. A similar objective function has been used in [16]. Note that if one multiplies a negative constant to equation (14) of [16], then the maximum operators become minimum operators. Extend the operator $E_{s,a}v(s,a) = r(s,a) + \gamma \mathbb{E}_{s',a'}P_{s,a}(s',a)\left[\sum_a v(s',a)\pi(s',a)|s,a\right]$ and view $v(s,a)$ as $Q(s,a)$, one finds that (2.9) is equivalent to equation (14) in [16] up to a constant by setting $f_s(x) = x^2/2$. In other words, our formulation (2.9) is equivalent to the main formulation (8) in [16] when $f = x^2/2$ and $\alpha < 0$. The paper [16] also pointed out that the off-policy trajectory can be directly used for the policy gradient. Although [16] uses a similar trick as the clipping method for numerical experiments, it is however only mentioned in the Appendix. There are other two differences between the current paper and [16]. First, we propose another more efficient algorithm, flipping, to accelerate the convergence rate. One can see the comparison of the two methods in Section 4. Second, we use $\hat{h}^{(1)}(L_t)$ defined in (2.11), while [16] directly applied $h^{(1)}(x)$ defined in (2.8) to $L_t$. We note that $\hat{h}^{(1)}(L_t)$ is a better estimates to $h^{(1)}(\ell_t)$ than $h^{(1)}(L_t)$ as explained after (2.11).

Algorithm 2 Variational actor-critic (model-free version)

\begin{algorithm}
\textbf{Require:} $\eta_V, \eta_\pi$: learning rates; $\beta$: pre-factor; $M$: batch size;
\hspace*{1cm} $Q(s,a,\omega), \pi(s,a,\theta)$: parametrized approximation of $Q(s,a), \pi(s,a)$;
\hspace*{1cm} $\{s_t, a_t, r_t\}_{t=0}^T$: trajectory generated from off-policy $\pi^*_t$;
\begin{algorithmic}
\State Random initialization of $\theta_0, \omega_0, k = 0$
\While{$\omega, \theta$ do not converge}
\State $j \leftarrow 1, k \leftarrow k + 1$
\For{$t = (k - 1)M + 1, \ldots, kM$}
\State $s_j = s_t$
\State $L_j = Q(s_t, a_t, \omega) - r_t - \gamma(V(s_{t+1}) - \lambda \mathcal{H}(s_{t+1})))$
\State $s_{t+1}' = s_t + (s_{t+2} - s_{t+1})$
\State $G_j^Q = -\nabla_\omega Q(s_t, a_t, \omega) + \beta L_j (\nabla_\omega Q(s_t, a_t, \omega) - \gamma \sum_a \nabla_\omega Q(s_{t+1}', a_t, \omega)\pi(s_{t+1}', a, \theta))$
\State $G_j^\pi = \beta (-\gamma \sum_a (Q(s_{t+1}', a_t, \omega) - \lambda \log\pi(s_{t+1}', a, \theta) - \lambda)\nabla_\theta \pi(s_{t+1}', a, \theta))$
\State $j \leftarrow j + 1$
\EndFor
\State $G_Q \leftarrow \frac{1}{M} \sum_{j=1}^M G_j^Q, \omega \leftarrow \omega - \eta_Q G_Q$
\State $\hat{\ell}_s \leftarrow \sum_{s_j = s} L_j$
\State $G_\pi^{(i)} \leftarrow \frac{1}{M} \sum_{j=1}^M \hat{h}^{(1)}(L_j) G_j^\pi, \theta \leftarrow \theta - \eta_\pi G_\pi^{(i)}$, where $\hat{h}^{(1)}$ is defined in (2.11)
\State $V(s) \leftarrow \sum_a Q(s,a,\omega)\pi(s,a,\theta); \mathcal{H}(s) \leftarrow \sum_a \pi(s,a,\theta)\log\pi(s,a)$
\EndWhile
\end{algorithmic}
\end{algorithm}

Specifically, if $\pi$ is the soft-max function of $\theta$, then the $\pi$ updates based on $(G_\pi^{(i)})_t$ in (2.10) can be simplified to,

\[(f_{sa})_t = \gamma \pi^k(s,a) \hat{\beta}(L_t) \left[ V^k(s) - Q^k(s,a) + \lambda \log \pi^k(s,a) - \lambda \mathcal{H}(\pi^k) \right] \mathbb{I}_{s=s_{t+1}}, \]

\[\pi^{k+1}(s,a) \propto \pi^k(s,a) \exp \left(-\frac{\eta_\pi}{M} \sum_{t=(k-1)M+1}^{kM} (f_{sa})_t \right). \]

3. Fixed point estimates

We define $(V^\infty, \pi^\infty)$ as the fixed point of Algorithm 1 if

\[V^\infty = V^\infty - \eta_V G_V(V^\infty, \pi^\infty), \quad \pi^\infty \propto \pi^\infty \exp \left(-\eta_\pi G_\pi^{(i)}(V^\infty, \pi^\infty) \right), \quad (3.1)\]
where $G_V$ and $G_{\theta}^{(i)}$ are defined in (2.7). Specifically, for the non-regularized MDP, i.e., $\lambda = 0$, the fixed point $\pi^\infty$ is exactly the optimal policy $\pi^*$ when $\beta$ is sufficiently large; for the regularized MDP, i.e., $\lambda > 0$, the fixed point $\pi^\infty$ is close to $\pi^*$ for large $\beta$ and small $\lambda$.

We analyze the non-regularized MDP and regularized MDP in Sections 3.1 and 3.2, respectively. For $\lambda = 0$, we prove in Lemma B.1 that when $\beta$ is sufficiently large and $V$ achieves its fixed point $V^\infty$, the gradient of the policy $G_{\theta}^{(i)}(V^\infty, \pi)$ cannot be equal to 0 for any $\pi$. This implies that the fixed point $\pi^\infty$ of the policy updates is on the boundary of the probability simplex, i.e., $\pi^\infty$ is a deterministic policy. Since all deterministic policies form a discrete set and the optimal policy $\pi^*$ for a non-regularized MDP is also a deterministic policy, there exists $\beta_0 > 0$ such that for all $\beta > \beta_0$, the fixed point $\pi^\infty$ of the algorithm is the optimal policy $\pi^*$. On the other hand, for $\lambda > 0$, the fixed point $\pi^\infty$ is a stochastic policy. Therefore, one can only prove that for $\beta > \beta_0$ and $0 < \lambda < \lambda_0$, the fixed point is close to the non-regularized optimal policy $\pi^*$.

Before analyzing the fixed point of the algorithm, we state some basic properties of the matrix $(I - \gamma P^\infty)$ in Proposition 3.1. In Lemma 3.2, we prove that the Bellman residual $\ell(V, \pi)$ defined in (2.3) is always positive at the fixed point $(V^\infty, \pi^\infty)$. This implies that $G_{\theta}^{(i)}$ takes the same form at the fixed point $(V^\infty, \pi^\infty)$. Hereafter, we shall omit the index $i$ of $G_{\theta}^{(i)}$ for notational simplicity.

**Proposition 3.1.** For any transition matrix $P$ and positive vector $c$, the following inequalities hold,

$$0 < (I - \gamma P)^{-1}c < \frac{\max_i c_i}{1 - \gamma} \mathbf{1}, \quad 0 < (I - \gamma P)^{-\top}c < \frac{\sum_i c_i}{1 - \gamma} \mathbf{1}. \quad (3.2)$$

For any constant $c$,

$$\text{if } (I - \gamma P)x \leq c\mathbf{1}, \quad \text{then } x \leq \frac{c}{1 - \gamma} \mathbf{1}. \quad (3.3)$$

See Appendix C for the proof.

**Lemma 3.2.** The fixed point $(V^\infty, \pi^\infty)$ of Algorithm 1 satisfies $\ell(V^\infty, \pi^\infty) > 0$ with $\ell(V, \pi)$ defined in (2.3).

**Proof.** The $V$ update achieves its fixed point when $G_V(V^\infty, \pi^\infty) = 0$, which gives,

$$-\rho + \beta(I - \gamma P^\pi) \top \left[ (I - \gamma P^\pi)V^\infty - r^\pi + \lambda \mathcal{H}(\pi^\infty) \right] \circ \rho = 0.$$

This leads to,

$$\ell(V^\infty, \pi^\infty) = \left[ (I - \gamma P^\pi)V^\infty - r^\pi + \lambda \mathcal{H}(\pi^\infty) \right] \frac{1}{\beta} \circ \left[ (I - \gamma P^\pi)^{-\top} \rho \right],$$

where $\tilde{\rho}_s = 1/\rho_s > 0$. By (3.2) of Proposition 3.1, all elements of $\ell$ are positive, which completes the proof. \qed

Since the three variants $G_{\theta}^{(i)}$ with $i = 0, 1, 2$ defined in (2.7) are the same when $\ell_s > 0$ and $\ell_s$ is positive at the fixed point $(V^\infty, \pi^\infty)$, they share the same fixed point.

**3.1. Fixed point for the non-regularized MDP**

Recall that the non-regularized MDP refers to the case where $\lambda = 0$. Below we prove that there exists a threshold $\beta_0$, such that for all $\beta > \beta_0$, the fixed point $\pi^\infty$ of the policy updates is the optimal policy $\pi^* = \arg\max_\pi V^\pi = \arg\max_\pi (I - \gamma P^\pi)^{-1}r$. For simplicity, we assume that the distribution in (2.1) is the uniform distribution, i.e., $\rho_s = 1/|S|$ in this section. The results can be extended to general distribution $\rho$ (see Rem. 3.7 for
Plugging \( V \equiv \ell \equiv V_\infty \), the only fixed point for the above update satisfies \( G(V, \pi) = 0 \).

**Proof.** Since \( V \) is updated as follows around the fixed point,

\[
V_k+1 = V_k - \eta V G(V, \pi_k),
\]

the only fixed point for the above update satisfies \( G(V, \pi) = 0 \). Hence, it is equivalent to prove that if \( V_k \equiv V^\infty \) in the \( \pi \) updates,

\[
(\pi_{k+1})_{sa} \propto (\pi_k)_{sa} \exp \left( -\eta_{s} G_{\theta,sa}(V_k, \pi_k) \right),
\]

then \( \lim_{k \to \infty} \pi_k = \pi^\infty \) with \( \pi^\infty \) stated in the lemma.

We prove in Lemma 3.2 that the Bellman residual \( \ell_s > 0 \) always holds at the fixed point \( (V^\infty, \pi^\infty) \), so \( \pi_k \) is updated as follows around the fixed point,

\[
(\pi_{k+1})_{sa} \propto (\pi_k)_{sa} \exp \left( \eta \pi \left( \gamma \sum_a P^a_{st} V^\infty_t + r_{sa} \right) \right).
\]

Plugging \( V_k \equiv V^\infty \) into the above equation gives

\[
(\pi_{k+1})_{sa} \propto (\pi_k)_{sa} \exp \left[ \eta \pi \left( \gamma \sum_a P^a_{st} V^\infty_t + r_{sa} \right) - \left( \gamma \sum_a P^a_{st} V^\infty_t + r_{sa} \right) \right],
\]

Details). Besides, we always assume that the action gap is strictly positive, i.e., let \( a_s^* = \max_a (r_{sa} + \gamma \sum_t P^a_{st} V^*_{t+1}) \), then

\[
\max_{a \neq a_s^*} (r_{sa} + \gamma \sum_t P^a_{st} V^*_{t+1}) < r_{sa^*} + \gamma \sum_t P^a_{st} V^*_{t+1}, \quad \forall s \in S.
\]

The fixed point \((V^\infty, \pi^\infty)\) of the algorithm is stated in Lemma 3.3. Note that \((V^\infty, \pi^\infty)\) satisfies similar coupled equations as the optimal solution \((V^*, \pi^*)\) in Lemma 3.4. The only difference is that \((V^\infty, \pi^\infty)\) satisfies \( G(V^\infty, \pi^\infty) = 0 \) while \((V^*, \pi^*)\) satisfies the Bellman equation \( V^* = (I - \gamma P^\pi)^{-1} r \). Note that \( G = 0 \) can be written as

\[
V(\pi) = (I - \gamma P^\pi)^{-1} r + \frac{1}{\beta} (I - \gamma P^\pi)^{-1} (I - \gamma P^\pi)^{-1} 1.
\]

When \( \beta \) is sufficiently large, \( V^\infty(\pi) \) approaches the true value function \( V^\pi = (I - \gamma P^\pi)^{-1} r \). On the other hand, we prove in Lemma 3.5 that there exists a threshold \( \alpha \), such that \( |V^*_s - V^\infty_s| \leq \alpha \) for \( \forall s \in S \), then \( \pi^\infty = \pi^* \). Combining the above lemmas, one concludes in Theorem 3.6 that the fixed point \( \pi^\infty \) is the optimal policy \( \pi^* \) as long as \( \beta > \frac{1}{(1 - \gamma)^2} \). Note that the lower bound for the prefactor \( \beta \) is not sharp, and we shall see in Section 4 that the algorithm converges numerically to the optimal policy \( \pi^* \) with much smaller \( \beta \).

**Lemma 3.3.** The fixed point \((V^\infty, \pi^\infty)\) of Algorithm 1 satisfies the following coupled equations,

\[
\begin{cases}
G(V^\infty, \pi^\infty) = 0, \\
\pi^\infty_s = \begin{cases} 1, & a = a_s; \\
0, & a \neq a_s, \end{cases} \quad \text{where } a_s = \arg\max_a \left( \gamma \sum_t P^a_{st} V^\infty_t + r_{sa} \right).
\end{cases}
\]

**Proof.** Since \( V \) is updated as follows

\[
V_{k+1} = V_k - \eta V G(V_k, \pi_k),
\]

the only fixed point for the above update satisfies \( G(V^\infty, \pi^\infty) = 0 \). Hence, it is equivalent to prove that if \( V_k \equiv V^\infty \) in the \( \pi \) updates,

\[
\pi_{k+1} \propto \pi_k \exp \left( -\eta G_{\theta,sa}(V_k, \pi_k) \right),
\]

then \( \lim_{k \to \infty} \pi_k = \pi^\infty \) with \( \pi^\infty \) stated in the lemma.

We prove in Lemma 3.2 that the Bellman residual \( \ell_s > 0 \) always holds at the fixed point \((V^\infty, \pi^\infty)\), so \( \pi_k \) is updated as follows around the fixed point,

\[
\pi_{k+1} \propto \pi_k \exp \left( \eta \pi \left( \gamma \sum_t P^a_{st} V^\infty_t + r_{sa} \right) \right).
\]

Plugging \( V_k \equiv V^\infty \) into the above equation gives

\[
\pi_{k+1} \propto \pi_k \exp \left[ \eta \pi \left( \gamma \sum_t P^a_{st} V^\infty_t + r_{sa} \right) - \left( \gamma \sum_t P^a_{st} V^\infty_t + r_{sa} \right) \right],
\]

where

\[
\omega = \sum_{a \neq a_s^*} \frac{P^a_{st} V^\infty_t + r_{sa}}{P^a_{st} V^\infty_t + r_{sa} - \left( \gamma \sum_t P^a_{st} V^\infty_t + r_{sa} \right)}.
\]
where $a_s$ is defined in (3.4). Then one has

$$\begin{align*}
\left\{ \begin{array}{ll}
(\pi_{k+1})_{sa} \propto (\pi_k)_{sa}, & \text{for } a = a_s; \\
(\pi_{k+1})_{sa} \propto (\pi_k)_{sa} \exp(f_{sa}(V^\infty)), & \text{for } a \neq a_s,
\end{array} \right.
\end{align*}$$

where $f_{sa}(V^\infty) < 0$. Hence, the $\pi$ updates can be equivalently written as

$$\begin{align*}
\left\{ \begin{array}{ll}
(\pi_k)_{sa} \propto (\pi_0)_{sa}, & \text{for } a = a_s; \\
(\pi_k)_{sa} \propto (\pi_0)_{sa} \exp(kf_{sa}(V^\infty)), & \text{for } a \neq a_s.
\end{array} \right.
\end{align*}$$

Notice that as $f_{sa}(V^\infty) < 0$, $\lim_{k \to \infty}(\pi_k)_{sa} \exp(kf_{sa}(V^\infty)) = 0$. Therefore,

$$\lim_{k \to \infty}(\pi_k)_{sa} = \begin{cases} 1, & a = a_s; \\ 0, & a \neq a_s, \end{cases}$$

which completes the proof. \hfill \Box

**Lemma 3.4.** The maximum and maximizer $(V^*, \pi^*)$ of the optimization problem (1.2) satisfy the following coupled equations:

$$\begin{align*}
\begin{cases}
V^* = r^\pi^* + \gamma P^\pi^* V^*, \\
\pi^*_{sa} = \begin{cases} 1, & a = a^*; \\ 0, & a \neq a^*, \end{cases}
\end{cases}
\end{align*}$$

where $a^*_s = \arg\max_a \left( r_{sa} + \gamma \sum_t P_{st}^a V^*_t \right)$. \hfill (3.5)

**Proof.** The maximum $V^*$ also satisfies the optimal Bellman equation as follows [24],

$$V^*_s = \max_a \left( r_{sa} + \gamma \sum_t P_{st}^a V^*_t \right), \quad \text{for } \forall s \in S. \hfill (3.6)$$

For $a^*_s$ and $\pi^*$ defined in (3.5), the following equality holds

$$r^\pi^*_s + \gamma \sum_t P^\pi^*_t V^*_t = \sum_a \pi^*_{sa} \left( r_{sa} + \gamma \sum_t P_{st}^a V^*_t \right) = \max_a \left( r_{sa} + \gamma \sum_t P_{st}^a V^*_t \right) = V^*_s.$$ 

Hence, $V^*$ satisfies the Bellman equation $V^* = r^\pi^* + \gamma P^\pi^* V^*$, which completes the proof. \hfill \Box

**Lemma 3.5.** For any value functions $V^\infty, V^* \in \mathbb{R}^{|S|}$, let $a_s = \arg\max_a \left( r_{sa} + \gamma \sum_t P_{st}^a V^\infty_t \right)$ and $a^*_s = \arg\max_a \left( r_{sa} + \gamma \sum_t P_{st}^a V^*_t \right)$ be the maximizers, then there exists

$$\epsilon' = \min_s \left( r_{sa^*_s} + \gamma \sum_t P_{st}^a V^*_t - \max_{a \neq a^*_s} \left( r_{sa} + \gamma \sum_t P_{st}^a V^*_t \right) \right),$$

such that as long as $|V^\infty_s - V^*_s| < \epsilon = \epsilon'/3$ for $\forall s$, then $a_s = a^*_s$ for $\forall s$.

The above lemma tells us that when the fixed point $V^\infty$ is close to $V^*$, then $a_s$ and $a^*_s$ defined in Lemmas 3.4 and 3.3 are the same.
Proof. If \(|V_t^\infty - V_t^*| \leq \epsilon\), then

\[ V_t^* - \epsilon \leq V_t^\infty, \quad -(V^* + \epsilon) \leq -V_t^\infty, \quad \text{for } \forall t \in S, \]

which further leads to,

\[ r_{s_a^*} + \gamma \sum_t P^i_{st} (V_t^* - \epsilon) \leq r_{s_a^*} + \gamma \sum_t P^i_{st} V_t^\infty, \]

\[-r_{s_a'} - \gamma \sum_t P^{i'}_{st} (V^* + \epsilon) \leq -r_{s_a'} - \gamma \sum_t P^{i'}_{st} V_t^\infty, \quad \text{for } a' \notin a_s^*. \]

Summing the two inequality together gives,

\[ r_{s_a^*} + \gamma \sum_t P^i_{st} V_t^\infty - \left( r_{s_a'} + \gamma \sum_t P^{i'}_{st} V^* \right) - 2\gamma \epsilon \leq r_{s_a^*} + \gamma \sum_t P^i_{st} V_t^\infty - \left( r_{s_a'} + \gamma \sum_t P^{i'}_{st} V_t^\infty \right). \]

Since the LHS \(\geq \epsilon' - 2\gamma \epsilon = \epsilon' - \frac{2\gamma}{2} \epsilon' > 0\), one has

\[ r_{s_a^*} + \gamma \sum_t P^i_{st} V_t^\infty - \left( r_{s_a'} + \gamma \sum_t P^{i'}_{st} V_t^\infty \right) > 0, \quad \text{for } \forall a' \neq a_s^*. \]

The above inequality implies that \(a_s^* = \text{argmax}_a (r_{s_a} + \gamma \sum_t P^i_{st} V_t^\infty) = a_s\), which completes the proof. \(\square\)

**Theorem 3.6.** Let \((V^*, \pi^*)\) be the maximum and maximizer of (1.2), and \(\alpha\) is a positive constant s.t. \(\alpha < \frac{1}{\beta_0} g(V^*, \pi^*)\). There exists a constant \(\beta_0 = \frac{\|S\|}{(1-\gamma)\alpha}\), such that \(\forall \beta > \beta_0\), the fixed point \((V^\infty, \pi^\infty)\) of Algorithm 1 is close to \((V^*, \pi^*)\) in the sense that

\[ |V_s^\infty - V_s^*| \leq \alpha, \quad \text{for } \forall s \in S, \quad \pi^\infty = \pi^*, \]

where

\[ g(V^*, \pi^*) = \min_{a \in S} \left( r_{s_a^*} + \gamma \sum_t P^i_{st} V_t^* - \max_{a \neq a_s^*} (r_{s_a} + \gamma \sum_t P^i_{st} V_t^*) \right) \]

with \(a_s^* = \text{argmax} (r_{s_a} + \gamma \sum_t P^i_{st} V_t^*). \)

**Proof.** The fixed point \((V^\infty, \pi^\infty)\) satisfies \(G_V(V^\infty, \pi^\infty) = 0\), which gives

\[ (I - \gamma P^{\pi^\infty}) V^\infty = r^{\pi^\infty} + \frac{1}{\beta} (I - \gamma P^{\pi^\infty})^{-1} 1. \]  (3.7)

Subtracting the value function in (3.5) \((I - \gamma P^*) V^* = r^*\) from the one in (3.7) yields,

\[ (I - \gamma P^{\pi^\infty}) V^\infty - r^{\pi^\infty} - (I - \gamma P^*) V^* + r^* = \frac{1}{\beta} (I - \gamma P^{\pi^\infty})^{-1} 1. \]  (3.8)
By the definition of \(\pi^\infty\) and \(\pi^*\) in Lemmas 3.3 and 3.4, one has
\[
r^{\pi^\infty} + \gamma P^{\pi^\infty} V^{\infty} \geq r^{\pi^*} + \gamma P^{\pi^*} V^{\infty}, \quad r^{\pi^*} + \gamma P^{\pi^*} V^* \geq r^{\pi^\infty} + \gamma P^{\pi^\infty} V^*.
\]
Applying the above two inequalities to (3.8) yields
\[
(I - \gamma P^{\pi^\infty})(V^{\infty} - V^*) \leq \frac{1}{\beta}(I - \gamma P^{\pi^\infty})^{-\top} 1 \leq (I - \gamma P^{\pi^*})(V^{\infty} - V^*).
\]
By (3.2) of Proposition 3.1, one has \(0 < \frac{1}{\beta}(I - \gamma P^{\pi^\infty})^{-\top} 1 < \frac{|S|}{\beta(1 - \gamma)} 1\). Therefore,
\[
(I - \gamma P^{\pi^\infty})(V^{\infty} - V^*) \leq \frac{|S|}{\beta(1 - \gamma)} 1, \quad (I - \gamma P^{\pi^*})(V^{\infty} - V^*) > 0.
\]
Applying (3.3) of Proposition 3.1 to the above two inequalities gives \(0 \leq V^{\infty} - V^* \leq \frac{|S|}{\beta(1 - \gamma)^2} 1\). By Lemma 3.5, when \(\frac{|S|}{\beta(1 - \gamma)^2} \leq \alpha = \frac{1}{2} g(V^*, \pi^*)\), then \(a_s = a^*_s\), which implies \(\pi^\infty = \pi^*\).

**Remark 3.7.** For general \(\rho\), one has
\[
(I - \gamma P^{\pi^\infty}) V^{\infty} = r^{\pi^\infty} + \frac{1}{\beta} \hat{\rho} \odot [(I - \gamma P^{\pi^\infty})^{-\top} \rho].
\]
where \((\hat{\rho})_s = 1/\rho_s\). The proof for Theorem 3.6 can be easily extended to general \(\rho\). The main difference is the bound for the second term of the RHS of the above equation. By applying Proposition 3.1, one can bound
\[
0 \leq \frac{1}{\beta} (I - \gamma P^\pi)^{-1} [\hat{\rho} \odot (I - \gamma P^\pi)^{-\top} \rho] \leq \frac{1/\min_s \rho_s}{\beta(1 - \gamma)^2} 1,
\]
for \(\forall \pi\). Hence, Theorem 3.6 still holds for general \(\rho\) with \(\beta_0 = \frac{1/(\min_s \rho_s)}{\alpha(1 - \gamma)^2}\).

3.2. Fixed point for the regularized MDP

Recall that the regularized MDP refers to the case where \(\lambda > 0\). The regularized optimal policy can be written in the following two equivalent forms
\[
(\pi_\lambda^*)_s = \arg\max_{\pi_s \in \Delta(\mathcal{A})} \left( r^s_s + \gamma \sum_t P_{st}^\pi(V^\pi_t) - \lambda H(\pi_s) \right), \quad (\pi_\lambda^*)_{sa} \propto \exp \left( \frac{1}{\lambda} \left( r_{sa} + \gamma \sum_t P_{st}^\pi(V^\pi_t) \right) \right).
\]
In this section, we prove that the fixed point \(V^\infty\) converges to the regularized optimal value function \(\lim_{\beta \to \infty} V^\infty = V^\lambda_\lambda^*\) as the prefactor \(\beta\) converges to infinity. On the other hand, for sufficiently large prefactor \(\beta > \beta_0\), the fixed point \(\pi^\infty\) will be close to the non-regularized optimal policy \(\pi^*\) if the entropy constant \(\lambda\) is small. However, when \(\lambda\) is relatively large, the fixed point \(\pi^\infty\) will be close to the regularized optimal policy \(\pi^*_\lambda\). For simplicity, we assume the distribution in (2.1) is the uniform distribution in this section. The results can be extended to general distribution \(\rho\).

In order to prove Theorem 3.10, we first prove Lemmas 3.8 and 3.9. The first Lemma is about the KL-divergence of two soft-max functions, and the second one gives a lower bound and an upper bound for the difference between the local fixed point of the \(V\) updates and the regularized optimal value function \(V^\lambda_\lambda^*\). Both lemmas will be useful in the proof of Theorem 3.10. In this section, we always assume that the learning rate \(\eta_s > 0\) for the policy updates is sufficiently small, so that \(\eta_s \beta \ell_s \lambda\) is always less than 1 and larger than 0.
Lemma 3.8. If $\pi = \text{soft-max}(\theta)$ and $\mu = \text{soft-max}(\omega)$ with $\theta, \omega \in \mathbb{R}^d$, then the KL divergence between the probability distribution $\pi$ and $\mu$ is $D_{KL}(\pi|\omega) \leq 2 \max_a |\theta_a - \omega_a|$.

See Appendix D for the proof.

Lemma 3.9. For any $\pi$, the solution $V$ to $G_V(V, \pi) = 0$ with $\lambda > 0$ satisfies the following inequalities:

$$V - V_\lambda^* < \frac{|S|}{\beta(1 - \gamma)^2} \mathbf{1},$$

and

$$[(I - \gamma P^{\pi^*})(V - V_\lambda^*)]_s \geq \sum_a (\pi_{sa} - (\pi_\lambda^*)_a)(\gamma \sum_t P_{at} V_t + r_{sa} - \lambda \log \pi_{sa}),$$

where $G_V$ is defined in (2.7), and $(V_\lambda^*, \pi^*_\lambda)$ are the maximum and maximizer to (1.3).

Proof. Since $(V_\lambda^*, \pi^*_\lambda)$ satisfies the regularized Bellman equation $(I - \gamma P^{\pi^*_\lambda})V^* + \lambda H(\pi^*) = r^{\pi^*_\lambda}$, subtracting it from $G_V(V, \pi) = 0$ gives

$$(I - \gamma P^\pi) V + \lambda H(\pi) - (I - \gamma P^{\pi^*_\lambda}) V^* - \lambda H(\pi^*_\lambda) = r^\pi - r^{\pi^*_\lambda} + \frac{1}{\beta}(I - \gamma P^\pi)^{-\top} \mathbf{1}. \quad (3.10)$$

Note that the regularized optimal policy $\pi^*_\lambda$ can also be represented by

$$(\pi^*_\lambda)_s = \arg \max_{\pi_s \in \Delta(A)} (r^\pi_s + \gamma \sum_t P^\pi_{at} (V^*_\lambda)_t - \lambda \log \pi_s).$$

Therefore, $r^{\pi^*_\lambda} V^* + \lambda H(\pi^*_\lambda) \geq r^\pi + \gamma P^\pi V^* - \lambda H(\pi)$. Plugging it to (3.10) leads to

$$(I - \gamma P^\pi)(V - V^*_\lambda) \leq \frac{1}{\beta}(I - \gamma P^\pi)^{-\top} \mathbf{1}.$$ 

Further, By (3.2) in Proposition 3.1, one has $0 < \frac{1}{\beta}(I - \gamma P^\pi)^{-\top} \mathbf{1} < \frac{|S|}{\beta(1 - \gamma)} \mathbf{1}$ for $\forall \pi$. Therefore,

$$(I - \gamma P^\pi)(V^\infty - V^*_\lambda) < \frac{|S|}{\beta(1 - \gamma)} \mathbf{1}.$$ 

Applying (3.3) in Proposition 3.1 to the above inequality yields

$$V^\infty - V^*_\lambda < \frac{|S|}{C(1 - \gamma)^2} \mathbf{1}.$$ 

On the other hand, (3.10) can also be written as,

$$(I - \gamma P^{\pi^*_\lambda})(V - V^*_\lambda) - \gamma (P^\pi - P^{\pi^*_\lambda})V + \lambda (H(\pi) - H(\pi^*_\lambda)) - (r^\pi - r^{\pi^*_\lambda}) = \frac{1}{\beta}(I - \gamma P^\pi)^{-\top} \mathbf{1},$$
which is equivalent to,

\[
(I - \gamma P^\pi_s)(V - V^*_s) + \sum_a (\pi_{sa} - (\pi^*_\lambda)_{sa})(-\gamma \sum_t P^\pi_{st} V_t - r_{sa} + \lambda \log \pi_{sa})
\]

\[
= \lambda \sum_a (\pi^*_\lambda)_{sa}(\log(\pi^*_\lambda)_{sa} - \log \pi_{sa}) + \frac{1}{\beta} [(I - \gamma P^\pi)^\top 1]_s.
\]

Note that the first term of the RHS is the KL divergence of \(\pi^*_\lambda\) from \(\pi\), so it is always positive. The second term of the RHS is also positive by (3.2) in Proposition 3.1. Therefore, the RHS of the above equation is larger than 0, which completes the proof of (3.9).

Theorem 3.10. For any \(\epsilon > 0\), if \(\beta > \frac{2|S|}{\epsilon(1-\gamma)^2}\), then the distance between the fixed point \((V^\infty, \pi^\infty)\) of Algorithm 1 with \(\lambda > 0\) and the maximum and maximizer \((V^*_\lambda, \pi^*_\lambda)\) of (1.3) can be bounded by

\[
|V^\infty_s - (V^*_\lambda)_s| < \frac{\epsilon}{2}, \text{ for all } s \in S, \quad D_{KL}(\pi^*_\lambda || \pi^\infty) \leq \frac{\epsilon \gamma}{\lambda}.
\]

If one further has \(\epsilon < \frac{1}{3} g(V^*, \pi^*)\) and \(\lambda < \frac{\epsilon(1-\gamma)}{2\log(|A|)}\), then the fixed point \(\pi^\infty\) is close to the non-regularized optimal policy in the sense that

\[
D_{KL}(\pi^* || \pi^\infty) = \log \left( 1 + \sum_{a \neq a_s} \exp \left( -\frac{\gamma}{\lambda} g_{sa} \right) \right),
\]

where \(g(V^*, \pi^*)\) is the same value defined in Theorem 3.6, and

\[
g_{sa} = r_{sa} + \gamma \sum_t P^\pi_{st} V^\infty_t - \left( r_{sa} + \gamma \sum_t P^\pi_{st} V^\infty_t \right) \quad \begin{cases} 
= 0, & a = a_s; \\
> 0, & a \neq a_s,
\end{cases} \quad a_s = \arg\max_a P^\pi_{sa} V^\infty_t.
\]

Remark 3.11. From the above theorem, one can see that as \(\beta\) approaches infinity, the fixed point \(V^\infty\) approaches the regularized optimal value function \(V^*_\lambda\). However, when \(\lambda\) is small, the difference between the fixed point \(\pi^\infty\) and the regularized optimal policy \(\pi^*_\lambda\) could be amplified by \(\frac{1}{\lambda}\). On the other hand, by Taylor expansion, the difference between \(\pi^\infty\) and the non-regularized optimal policy \(\pi^*\) can be approximated by

\[
D_{KL}(\pi^* || \pi^\infty) \approx \sum_{a \neq a_s} \exp \left( -\frac{\gamma}{\lambda} g_{sa} \right),
\]

which is close to 0 when \(\lambda\) is small.

Proof. The fixed point of the policy updates satisfies \(G_{\theta_{sa}}(V^\infty, \pi^\infty) = 0\), where \(G_{\theta_{sa}}\) is defined in (2.7). That is,

\[
\beta \rho_s \ell_s \left[ -\gamma \sum_t P^\pi_{st} V_t - r_{sa} + \lambda \log \pi_{sa} \right] + c_s = 0.
\]

It is equivalent to

\[
\pi^\infty_{sa} \propto \exp \left( \frac{1}{\lambda} \left( r_{sa} + \gamma \sum_t P^\pi_{st} V^\infty_t \right) \right).
\]
Let 

\[ a_s = \arg\max_a \left( r_{sa} + \gamma \sum_t P_{at} V^\infty_t \right), \]  

(3.12)

then \( \pi^\infty \) can be written as 

\[ \pi^\infty_{sa} \propto \exp \left( -\frac{\gamma}{\lambda} g_{sa} \right), \]  

(3.13)

where 

\[ g_{sa} = r_{sa} + \gamma \sum_t P_{at} V^\infty_t - \left( r_{sa} + \gamma \sum_t P_{at} V^\infty_t \right) \]

\[ = 0, \quad a = a_s, \]  

\[ > 0, \quad a \neq a_s. \]

On the other hand, by the equality (3.11), one has 

\[ \gamma \sum_t P_{at} V^\infty_t + r_{sa} - \lambda \log \pi^\infty_{sa} = f_s, \]

where \( f_s \) is a value independent of \( a \). Inserting the above \( \pi^\infty \) into the \( \pi \) in (3.9) gives, 

\[ [(I - \gamma P^{\pi^\infty}) (V^\infty - V^{\star \lambda})]_s \geq f_s \sum_a (\pi^\infty_{sa} - (\pi^\star \lambda)_{sa}) = 0, \]

where the last equality is due to \( \sum_a \pi^\infty_{sa} = \sum_a (\pi^\star \lambda)_{sa} = 1 \). Therefore, by (3.3) in Proposition 3.1, one has \( V^\infty - V^{\star \lambda} > 0 \). Combining it with Lemma 3.9 implies 

\[ 0 < V^\infty - V^{\star \lambda} < \frac{|S|}{\beta(1 - \gamma)^2} 1. \]  

(3.14)

On the other hand, \( (\pi^\star \lambda)_{sa} \) can be represented by 

\[ (\pi^\star \lambda)_{sa} \propto \exp \left( \frac{1}{\lambda} \left( r_{sa} + \gamma \sum_t P_{at} (V^{\star \lambda}_t) \right) \right). \]

Since for \( \forall s \in S \), 

\[ \max_a \left| \frac{1}{\lambda} \left( r_{sa} + \gamma \sum_t P_{at} (V^{\star \lambda}_t) \right) - \frac{1}{\lambda} \left( r_{sa} + \gamma \sum_t P_{at} V^\infty_t \right) \right| \]

\[ = \max_a \left| N \sum_t P_{at} ((V^{\star \lambda}_t) - V^\infty_t) \right| \]

\[ < \frac{\gamma |S|}{\lambda \beta(1 - \gamma)^2}, \]

by Lemma 3.8, one has 

\[ D_{KL}((\pi^\star \lambda)_{s} | \pi^\infty_{s}) \leq \frac{2|S|}{\lambda \beta(1 - \gamma)^2}. \]  

(3.15)

To sum up, if \( \beta > \frac{2|S|}{\epsilon(1 - \gamma)^2} \), then by (3.14) and (3.15) 

\[ 0 < V^\infty - V^{\star \lambda} < \frac{\epsilon}{2} 1, \quad D_{KL}(\pi^\star \lambda | \pi^\infty) < \frac{\epsilon \gamma}{\lambda}, \]
which completes the proof for the first part of the lemma.

For the second part, note that
\[
V^* = V^\pi^* \geq V^\lambda = (I - \gamma P^\pi^*)^{-1} \pi^* + (I - \gamma P^\pi^*)^{-1}(-\lambda H(\pi^*)) \geq V^\pi^* \geq V^*;
\]
\[
V^\lambda = V^\pi^* + (I - \gamma P^\pi^*)^{-1}(-\lambda H(\pi^*)) \leq V^\pi^* + \frac{1}{1 - \gamma} \max_s (-\lambda H(\pi^*_s)) \leq V^* + \frac{\lambda}{1 - \gamma} \log(|A|),
\]

where one applies (3.2) in Proposition 3.1 to the second inequality on the first equation and the first inequality on the second equation. Hence, one has
\[
0 < V^*_\lambda - V^* \leq \frac{\lambda}{1 - \gamma} \log(|A|)1.
\]

Combining it with the inequality (3.14), one has
\[
|V^\infty_t - V^*_t| \leq |V^\infty_t - (V^*_\lambda)_t| + |(V^*_\lambda)_t - V^*_t| < \frac{|S|}{\beta(1 - \gamma)^2} + \frac{\lambda}{1 - \gamma} \log(|A|).
\]

Therefore, when \( \beta > \frac{2|S|}{e(1-\gamma)^2} \) and \( \lambda < \frac{e(1-\gamma)}{2 \log(|A|)} \), then \( |V^\infty_t - V^*_t| < \epsilon \) for all \( t \in S \). As proved in Lemma 3.5, when \( |V^\infty_t - V^*_t| < \epsilon = \frac{1}{2} g(V^*, \pi^*) \) for all \( t \in S \), then \( a_s = a^*_s \) with \( a^*_s \) defined in Lemma 3.4 and \( a_s \) defined in (3.12). By the definition of \( \pi^* \) in Lemma 3.4 and \( \pi^\infty \) in (3.13), one has
\[
D_{KL}(\pi^*||\pi^\infty) = \log \left( \frac{1}{\pi^\infty_{s^*}} \right) = \log \left( 1 + \sum_{a \neq a^*} \exp \left( -\frac{\gamma}{\lambda} g_{sa} \right) \right),
\]

which completes the proof for the second part of the lemma.

\[\square\]

4. Numerical experiments

This section studies the performance of the model-based and model-free algorithms numerically (Algorithms 1 and 2). Two different MDPs, one with states embedded in the 1D space and another with states in the 2D space, are used as testing examples. The numerical experiments demonstrate that both non-regularized (\( \lambda = 0 \)) and regularized (\( \lambda > 0 \)) versions of the proposed algorithm converge to policies close to the non-regularized optimal policy \( \pi^* \). In addition, the algorithm combined with the BFF idea solves the double sampling problem. A comparison between the flipping method and the natural policy gradient (NPG) method is also provided to demonstrate that the flipping method outperforms the NPG method.

4.1. Example 1

Consider an MDP with a discrete state space \( S = \left\{ s_k = \frac{2\pi k}{n} \right\}_{k=0}^{n-1} \). The transition dynamics is given by
\[
\hat{s}_{t+1} = \text{mod} \left( s_t + \frac{2\pi}{n} a_t + \sigma Z_t, n \right), \quad s_{t+1} = \begin{cases} 
\arg\min_{i \in Z, i \in [0, n-1]} |\hat{s}_{t+1} - i|, & \text{if } \hat{s}_{t+1} \in [0, n - 1/2), \\
0, & \text{if } \hat{s}_{t+1} \in [n - 1/2, n),
\end{cases}
\]

(4.1)
Figure 2. The plots show the error $\|\pi_k - \pi^*\|_{L_1}$ from Algorithm 1 for the size of the state space $n = 5, 55, 105$ from left to right. The first row is for the non-regularized method, while the second row is for the regularized method with $\lambda = 0.1$. We run 100 simulations for each case and plot the mean in black and red.

where $a_t \in \mathcal{A} = \{\pm 1\}$ and $Z_t \sim N(0, 1)$ follows the normal distribution. The reward function $r(s) = 1 + \sin(s)$.

In Figures 2–5, $\sigma = 0$, i.e. the dynamics is deterministic given the current state and action. In Figure 6, $\sigma \neq 0$ and hence given the current state and action the next state is stochastic.

Results of Algorithm 1. Here we assume that the transition dynamics is known. $V(s)$ is represented in the tabular form and $\pi(s, a)$ is parameterized with the soft-max function. Since it is shown in Figure 1 that the vanilla gradient descent results in increasing error in the initial stage, only the clipping and flipping methods are tested here. Both the non-regularized ($\lambda = 0$) and regularized ($\lambda = 0.1$) method are tested. The error $\pi_k - \pi^*$ in the $L_1$ norm is shown in Figure 2. In order to demonstrate the stability of the algorithm, 100 simulations with different initializations are run for each case and the mean of 100 simulations is plotted in a darker color. The learning rates $\eta_V$ and $\eta_\pi$ are both set to be $1/(4\beta)$ for all cases.

First, for both the regularized and non-regularized method, the difference between $\pi_k$ and the true optimal policy $\pi^*$ approaches to 0. Second, the prefactors that make the number of states $n = 5, 55, 105$ converge are $\beta = 10, 100, 1000$, respectively. As the number of states increases, the prefactor increases as expected, which is consistent with what we demonstrated in Theorems 3.6 and 3.10. In addition, one finds that the flipping method decays consistently, while the clipping method decays slowly at first and then matches the rate of the flipping method.

Results of Algorithm 2 with different prefactors. Here we assume that the transition dynamics is unknown. $Q(s, a)$ is represented by the tabular form and $\pi(s, a)$ is parameterized by the soft-max function. Note that in this example, given $s$ and $a$, the transition dynamics is deterministic. Therefore, one only needs to duplicate the first sample for the next state to the second sample, namely, letting $s'_{t+1} = s_{t+1}$ in Algorithm 2.
Figure 3. The plots show the error $\|\pi_k - \pi^*\|_{L^1}$ from Algorithm 2 with $\lambda = 0.1$ for different prefactors $\beta$. The left one is the results of the clipping method, while the right one is the results of the flipping method. The grey, pink and blue lines represent 100 simulations with $\beta$ from small to large. The mean of each case is plotted in black, red and dark blue. The error is for the non-regularized optimal policy $\pi^*$.

Figure 4. The plots show the error $\|\pi_k - \pi^*\|_{L^1}$ from Algorithm 2 with $\lambda = 0.1$ for the size of the state space $n = 5, 55, 105$ from left to right. The grey and pink lines represent 100 simulations for clipping and flipping methods, respectively. The mean of each case is plotted in black and red.

The off-policy $\pi_b(a|s) = 1/2$ for $\forall s \in S$ is used to generate the trajectory $\{s_t, a_t, r_t\}_{t=0}^T$. The error $\pi_k - \pi^*$ in the $L^1$ norm is shown in Figure 3. In order to show the stability of the algorithm, 100 simulations (with different off-policy trajectories and different parameter initializations) are run for each case and the mean of 100 simulations is plotted in a darker color. To encourage exploration, we set $\lambda = 0.1$. The learning rate $\eta_\pi = 4/C$, $\eta_Q = 30/C$, and the batch size $M = 1000$.

Figure 3 shows that, for both clipping and flipping methods, the probability of reaching the optimal policy $\pi^*$ becomes larger as the prefactor $\beta$ grows. Furthermore, clipping still has several simulations diverge with $\beta = 70$, while all the simulations for flipping converge with $\beta = 70$. Therefore, flipping requires a smaller $\beta$ to be convergent compared with clipping.

Figure 4 compares the convergence curves of clipping and flipping. Similar to Figure 2, the error for the clipping method decays slowly at first, while the error from flipping decays consistently. Comparing Figure 4 with Figure 2, one can see that the stochastic algorithm converges in fewer steps. The reason is that one can set the prefactor $\beta$ smaller and the learning rate larger to encourage stochasticity.
Figure 5. The plots show the comparison of the error $\|\pi_k - \pi^*\|_{L_1}$ between the flipping method and the NPG method (4.2) – (4.3) for the size of the state space $n = 5$ and $n = 55$. Both use the regularized objective function, i.e., $\lambda = 0.1$. The green and pink lines represent 100 simulations for the NPG method and the flipping method, respectively. The mean of each case is plotted in blue and red.

Figure 6. The plots show the error $\|\pi_k - \pi^*\|_{L_1}$ from the flipping method of Algorithm 2 with BFF for the size of the state space $n = 5, 55, 105$ from left to right. The pink lines represent 100 simulations, and their mean is plotted in red.

Comparison with other methods. Figure 5 compares the flipping method with the natural policy gradient method (NPG) given by,

$$
\pi_{sa}^{k+1} = (\pi_{sa}^k)^{1 - \frac{\lambda \eta}{1 - \gamma}} \exp \left( \frac{\eta Q_n^k(s, a)}{1 - \gamma} \right),
$$

(4.2)

where $Q_n^k(s, a)$ is estimated by solving the residual Bellman minimization problem

$$
\min_{Q} \mathbb{E}_{(s, a) \sim \rho} \left( Q(s, a) - \mathbb{E}_{s' \sim P_s^k(\cdot | s, a)} \left[ \gamma \sum_a (Q(s', a) - \lambda \log \pi(s', a) \pi(s', a) | s, a) - r(s, a) \right] \right)^2.
$$
The transition dynamics is given by

\[ \n(s_{t+1})_k \leftarrow \text{mod} \left( (\tilde{s}_{t+1})_k, n_k \right), \quad k = 1, 2, \]

\[ (s_{t+1})_k \left\{ \begin{array}{ll}
\arg\min_{i \in Z, i \in [0, n_k - 1]} & \left| (\tilde{s}_{t+1})_k - i \right|, & \text{if } (\tilde{s}_{t+1})_k \in [0, n_k - 1/2), \\
0, & \text{if } (\tilde{s}_{t+1})_k \in [n_k - 1/2, n_k), \end{array} \right. \]

The algorithm for \( Q^{\pi_k} \) updates \( Q^j \) with initialization \( Q^0 = Q^{\pi_{k-1}} \) and stops when \( \sum_{s,a} (Q^j(s,a) - Q^{j-1}(s,a))^2/n < \epsilon \):

\[
\begin{align*}
    w_t &= Q^j(s_t, a_t) - r_t - \gamma \left( V^j(s_{t+1}) - \lambda H^k(s_{t+1}) \right), \\
    G_t(s,a) &= w_t \mathbb{1}_{s_t = s, a_t = a} - \gamma \pi(s_{t+1}, a) w_t \mathbb{1}_{s_t = s', t+1}, \\
    Q^{j+1} &= Q^j - \eta Q \sum_{t = (j-1)M+1}^j G_t; \quad V^{j+1}(s) = \sum_a Q^{j+1}(s,a) \pi^k(s,a),
\end{align*}
\]

where \( H^k(s) = \sum_{a} \pi^k(s,a) \log(\pi^k(s,a)) \). The batch size \( M = 1000 \) and regularization constant \( \lambda = 0.1 \) are the same for both methods. For the NPG method, we set \( \epsilon = 2 \times 10^{-4}, \eta_Q = 4, \eta_\pi = 0.1 \) for \( n = 55 \) and \( \epsilon = 2 \times 10^{-2}, \eta_Q = 30, \eta_\pi = 0.1 \) for \( n = 55 \). For the flipping method, we set \( \beta = 1, \eta_Q = 2/C, \eta_\pi = 1/C \) for \( n = 5 \) and \( \beta = 80, (\eta_Q)_k = (\eta_\pi)_k = \min(0.999k^{10}, 0.9) \) for \( n = 55 \).

For \( n = 5 \), though both methods converge to the optimal policy \( \pi^* \), the flipping method converges faster than the NPG method. For \( n = 55 \), NPG converges to the regularized optimal policy \( \pi^*_k \), while our method converges to a policy close to the true optimal policy \( \pi^* \) with a high probability.

**Results of Algorithm 2 with BFF.** Here we assume the transition dynamics is stochastic given the current state and action. We set \( \sigma = 1 \) for \( n = 5 \), \( \sigma = 0.5 \) for \( n = 55 \) and \( \sigma = 0.1 \) for \( n = 105 \). Unlike Figures 4–5, BFF is used to approximate the second independent sample for the next state in Figure 6. Other than that, the setting remains the same as Figure 4. One can see that BFF provides a good approximation for the gradient. The approximation error of the flipping method decays quickly for all three different cases.

### 4.2. Example 2

Consider another MDP with a discrete state space \( S = \{s_{ij}\}_{i,j=0}^{n_1-1, j=n_2-1} \), where \( s_{ij} = (i, j) \) is a two-dimensional vector. The transition dynamics is given by

\[
\begin{align*}
    \tilde{s}_{t+1} &\leftarrow s_t + (1 + \sigma Z_t)a_t, \\
    (\tilde{s}_{t+1})_k &\left\{ \begin{array}{ll}
\arg\min_{i \in Z, i \in [0, n_k - 1]} & \left| (\tilde{s}_{t+1})_k - i \right|, & \text{if } (\tilde{s}_{t+1})_k \in [0, n_k - 1/2), \\
0, & \text{if } (\tilde{s}_{t+1})_k \in [n_k - 1/2, n_k), \end{array} \right. \\
    (s_{t+1})_k &\left\{ \begin{array}{ll}
\arg\min_{i \in Z, i \in [0, n_k - 1]} & \left| (\tilde{s}_{t+1})_k - i \right|, & \text{if } (\tilde{s}_{t+1})_k \in [0, n_k - 1/2), \\
0, & \text{if } (\tilde{s}_{t+1})_k \in [n_k - 1/2, n_k), \end{array} \right.
\end{align*}
\]
where \( a_t \in \mathcal{A} = \{(\pm 1,0), (0,\pm 1)\} \) and \( Z_t \sim N(0,1) \). \( n_1 = n_2 = 7 \), the reward is set to be \( r(s_{ij}) = 2 + \sin\left(\frac{2\pi t}{n_1}\right) + \cos\left(\frac{2\pi t}{n_2}\right) \), and the noise \( \sigma \) is set to be 0.1.

The result is plotted in Figure 7. We set \( \lambda = 0 \) for the non-regularized objective function and use BFF to approximate the second independent sampling for the next state. The prefactor and learning rate are set to be \( \beta = 30 \) and \( \eta_3 = \eta_\pi = \frac{30}{7} \). The error is plotted out in the \( L_1 \) norm. 98% of the simulations converge to the true optimal policy \( \pi^* \). Note that the value function in the right plot of Figure 7 is the value function \( V^{\pi_k} \), which is different from the \( V^k \) in the algorithm. It shows that the policy indeed consistently maximizes the value function \( V^\pi \).

### APPENDIX A. SGD ALGORITHM FOR V

Given a trajectory \( \{(s_t, a_t, r_t)\}_{t=0}^T \), the unbiased stochastic estimate for the gradient of (2.1) is

\[
(G_V)_t = -\nabla_\omega V^k(s_t) + \beta L_t \left( \nabla_\omega V^k(s_t) - \gamma \frac{\pi^k(s_t, a'_t)}{\pi_b(s_t, a'_t)} \nabla_\omega V^k(s_{t+1}) \right);
\]

\[
(G^i_\pi)_t = \beta \hat{h}^{(i)}(L_t) \left( -\gamma \frac{\pi^k(s_t, a'_t)}{\pi^b(s_t, a'_t)} V^k(s_{t+1}) \nabla_\theta \log \pi^k(s_t, a'_t) + \lambda \sum_a \nabla_\theta \pi^k(s_t, a) (\log \pi^k(s_t, a) + 1) \right);
\]

where \( V^k(s) = V(s, \omega_k), \pi^k(s, a) = \pi(s, a, \theta_k) \) and \( \hat{h}^{(i)} \) is defined in (2.11). Here \( L_t \) is the estimates for the Bellman residual,

\[
L_t = V^k(s_t) - r_t - \gamma \frac{\pi^k(s_t, a'_t)}{\pi^b(s_t, a'_t)} V^k(s_{t+1}) + \lambda \sum_a \pi^k(s_t, a) \log \pi^k(s_t, a) + \lambda \sum_a \pi^k(s_t, a) \log \pi^k(s_t, a).
\]

#### Algorithm 3 V-formulation

**Require:** \( \eta_V, \eta_\pi \): prefactor; \( \beta \): penalty constant; \( M \): batch size; 

\( \{s_t, a_t, r_t\}_{t=0}^T \): trajectory generated from off-policy \( \pi_b \); 

1. Random initialization of \( \theta_0, \omega_0 \), \( k = 0 \)
2. **while** \( \omega, \theta \) do not converge **do**
3. \( j \leftarrow 0, k \leftarrow k + 1 \)
4. **for** \( t = (k-1)M + 1, \ldots, kM \) **do**
5. \( s_j = s_t \)
6. \( L_j = V(s_t, \omega) - r_t - \gamma \tau(s_t, a_t) V(s_{t+1}, \omega) + \lambda \mathcal{H}(s_t) \)
7. \( s'_{t+1} \leftarrow s_t + (s_{t+2} - s_{t+1}); \quad a'_{t} \leftarrow a_{t+1} \)
8. \( G^V_V = -\nabla_\omega V(s_t, \omega) + \beta L_t (\nabla_\omega V(s_t, \omega) - \gamma \tau(s_t, a'_t) \nabla_\omega V(s_{t+1}, \omega)) \)
9. \( G^V_\pi = \beta \left( -\gamma \tau(s_t, a'_t) V(s_{t+1}, \omega) \nabla_\theta \log \pi(s_t, a_{t}, \theta) + \lambda \sum_a (\log \pi(s_t, a, \theta) + 1) \nabla_\theta \pi(s_t, a, \theta) \right) \)
10. \( j \leftarrow j + 1 \)
11. **end for**
12. \( G_V \leftarrow \frac{1}{M} \sum_{j=1}^M G^V_V; \quad \omega \leftarrow \omega - \eta_V G_V \)
13. \( \hat{h}_s \leftarrow \sum_{j=s}^{s+j} L_j \)
14. \( G^i_\pi \leftarrow \frac{1}{M} \sum_{j=1}^M \hat{h}^{(i)}(L_j) G^V_\pi; \quad \theta \leftarrow \theta - \eta_\pi G^i_\pi \), where \( \hat{h}^{(i)} \) is defined in (2.11)
15. \( \tau(s, a) \leftarrow \frac{\pi(s, a, \theta)}{\pi^b(s, a, \theta)}; \quad \mathcal{H}(s) \leftarrow \sum_a \pi(s, a, \theta) \log \pi(s, a, \theta) \)
16. **end while**
The reward is not a constant, i.e., where \( \bar{r} \) is taken at state \( t \). Here we use the BFF algorithm proposed in [31] to approximate this two samples.

\[
a_t^* = a_{t+1}, \quad s_{t+1} = s_t + (s_{t+2} - s_{t+1}).
\]

The stochastic algorithm for the V-formulation is summarized in Algorithm 3.

**APPENDIX B. FIXED POINT OF ALGORITHM 1 WITH \( \lambda = 0 \) IS NOT STOCHASTIC POLICY**

**Lemma B.1.** Assume \(|A| > 2\), the null space of \( P^a - P^{a'} \) is the linear space spanned by \( 1 \) for all \( a \neq a' \), and the reward is not a constant, i.e., \( r_{sa} \neq r \). When \( \beta \) is sufficiently large, then \((G_V, G_\theta) \neq (0, 0)\).

**Proof.** Assume that \( G_\theta = 0 \), then it gives

\[
\gamma P^a V + r^a = c, \quad \forall a.
\]

where \( r^a = (r_{sa})_{a \in S} \) is an \(|S|\)-dimensional vector and \( c \) is a constant vector. This is equivalent to,

\[
\gamma (P^a - P^{a'}) V = r^{a'} - r^a, \quad \forall a \neq a'. \tag{B.1}
\]

Note that if there exists three different actions \( a, a', a'' \), such that \( r^a - r^{a'} = r^a - r^{a''} = c \neq 0 \), then \( r^a = r^{a''} = 0 \). Therefore, when \(|A| \geq 3\), the value of \( r^{a'} - r^a \) can be separated into two different cases. The first case is that

there exists three different actions \( a, a', a'' \in A \) such that \( r^a - r^{a'} \neq r^a - r^{a''} \neq 0 \). \tag{B.2}

The second case is that

there exists two different actions \( a, a' \in A \) such that \( r^a = r^{a'} = r \), \tag{B.3}

where \( r \) is a constant vector.

Let us consider the first case where \( r^a - r^{a'} \neq r^a - r^{a''} \) and both \( r^a - r^{a'} \) and \( r^a - r^{a''} \) are not equal to \( 0 \). Let \( N \) be the null space of \((P^a - P^{a'})\) for \( \forall a \neq a' \), which is a linear space spanned by \( 1 \). If the projection of \( r^a - r^{a'} \) onto \( N \) is not equal to \( 0 \), then there is no solution for \( V \) in (B.1). If the projection of \( r^a - r^{a'} \) and \( r^a - r^{a''} \) onto the null space \( N \) are both equal to \( 0 \), then there does not exist a vector \( V \) such that \( \gamma (P^a - P^{a'}) V = r^{a'} - r^a \) and \( \gamma (P^a - P^{a''}) V = r^{a''} - r^a \). Therefore, there is no solution for (B.1). To sum up, \( G_\theta \neq 0 \) for the first case (B.2).

Next, let us consider the second case where \( r^a = r^{a'} = r \), then \( V = c_1 1 \) is the only solution to (B.1). Given that \( P^\pi \) is transition matrix, \( P^\pi 1 = 1 \). Plugging it into \( G_V = 0 \) yields,

\[
- \rho + \beta (I - \gamma P^\pi)^\top \left[ ((1-\gamma)c_1 1 - r) \odot \rho \right] = 0. \tag{B.4}
\]

Multiplying \( 1^\top \) to (B.4) gives

\[
(1-\gamma)c_1 = \bar{r} + \frac{1}{\beta(1-\gamma)},
\]

where \( \bar{r} = \sum_s r_s \rho_s \). Plugging it back to (B.4) leads to

\[
\bar{r} - r + \frac{1}{\beta(1-\gamma)} = \frac{1}{\beta} (I - \gamma P^\pi)^\top 1.
\]
When $\beta > \frac{1}{(1-\gamma)(\max s r_s - \bar{r})}$, then at least one element of the LHS is negative. However, the RHS is always positive by Proposition 3.1, which gives contradiction. Therefore, $(G_V, G_0) \neq (0, 0)$ for the second case (B.3).

**APPENDIX C. PROOF OF PROPOSITION 3.1**

**Proof.** Let $x = (I - \gamma P)^{-1}c$, and assume $x_s = \min_i x_i \leq 0$. The $s$-th component of $(I - \gamma P)x = c$ is

$$c_s = x_s - \gamma \sum_t P_{st}x_t \leq x_s - \gamma \sum_t P_{st}x_s = x_s - \gamma x_s \leq 0,$$

which contradicts with the assumption $c_s > 0$ for all $s$. On the other hand, by letting $x_{s'} = \max_i x_i$, the $s'$-th component of the $(I - \gamma P)x = c$ is

$$(1 - \gamma)x_{s'} \leq x_{s'} - \gamma \sum_t P_{s't}x_t = c_s \leq \max_i c_i.$$

Therefore,

$$x_{s'} \leq \frac{\max_i c_i}{1 - \gamma},$$

which completes the proof for the first part.

For $(I - \gamma P)^{\top}x = c$, summing over all the components that $x_s \leq 0$ yields

$$\sum_s c_s \mathbb{1}_{x_s \leq 0} = \sum_s x_s \mathbb{1}_{x_s \leq 0} - \gamma \sum_{s,t} P_{ts}x_t \mathbb{1}_{x_s \leq 0}$$

$$= \sum_s x_s \mathbb{1}_{x_s \leq 0} - \gamma \sum_{s,t} P_{ts}x_t \mathbb{1}_{x_s \leq 0} \leq \sum_{s,t} \left( \sum_s P_{ts} \mathbb{1}_{x_s \leq 0} \right) (-x_t \mathbb{1}_{x_t \leq 0})$$

$$\leq (1 - \gamma) \sum_s x_s \mathbb{1}_{x_s \leq 0} \leq 0.$$

The first inequality holds because the last term on the second line is always $\leq 0$. The second inequality is due to $\sum_s P_{ts} \mathbb{1}_{x_s \leq 0} \leq \sum_s P_{ts} = 1$ for all $t$. However, the LHS is always strictly larger than 0, which gives a contradiction. Therefore all components of $x$ are positive. On the other hand, note that

$$(1 - \gamma) \sum_s x_s = 1^\top (I - \gamma P)^\top x = 1^\top c = \sum_s c_s,$$

and $x_s > 0$, therefore, $x_s \leq \sum_s x_s = \frac{1}{1-\gamma}c$, which completes the proof for the second part.

For $(I - \gamma P)x \leq c1$, let $x_s = \max_i x_i$, then the $s$-th component of $(I - \gamma P)x \leq c1$ is

$$c \geq x_s - \gamma \sum_t P_{st}x_t \geq x_s - \gamma \sum t P_{st}x_s = (1 - \gamma)x_s,$$

which leads to $(1 - \gamma)x_s \leq c$. Therefore, $x \leq x_s \leq \frac{c}{1-\gamma}1.$
Appendix D. Proof of Lemma 3.8

Proof. First note that
\[
\log(\pi_a) - \log(\mu_a) = \theta_a - \omega_a - \left( \log \left( \sum_b \exp(\theta_b) \right) - \log \left( \sum_b \exp(\omega_b) \right) \right).
\] (D.1)

Let \( f(x) = \log \left( \sum_a \exp(x_a) \right) \) be a function mapping \( x \in \mathbb{R}^d \) to \( \mathbb{R} \), then
\[
\nabla_x f = \frac{\exp(x_a)}{\sum_b \exp(x_a)},
\]
which implies that \( \|\nabla_x f(x)\|_1 = 1 \) for \( \forall x \in \mathbb{R}^d \). By the mean value theorem, one has \( \forall \theta, \omega \in \mathbb{R}^d \),
\[
\left| \log \left( \sum_b \exp(\theta_b) \right) - \log \left( \sum_b \exp(\omega_b) \right) \right| = |f(\theta) - f(\omega)| = |\langle \theta - \omega, \nabla_x f(x) \rangle| \\
\leq \max_a |\theta_a - \omega_a| \|\nabla_x f(x)\|_1 = \max_a |\theta_a - \omega_a|,
\]
where \( x \) is a convex combination of \( \theta \) and \( \omega \). Applying the above inequality into (D.1) yields
\[
\log(\pi_a) - \log(\mu_a) \leq 2 \max_a |\theta_a - \omega_a|.
\]

Therefore,
\[
D_{KL}(\pi|\omega) = \sum_a \pi_a (\log(\pi_a) - \log(\omega_a)) \leq 2b \sum_a \pi_a = 2b.
\]

\[ \square \]

References

[1] A. Agarwal, S.M. Kakade, J.D. Lee and G. Mahajan, Optimality and approximation with policy gradient methods in Markov decision processes, in Conference on Learning Theory, PMLR (2020) 64–66.
[2] J. An, L. Ying and Y. Zhu, Why resampling outperforms reweighting for correcting sampling bias. International Conference on Learning Representations (ICLR) (2021).
[3] J. Baxter and P.L. Bartlett, Infinite-horizon policy-gradient estimation. J. Artific. Intell. Res. 15 (2001) 319–350.
[4] S. Cen, C. Cheng, Y. Chen, Y. Wei and Y. Chi, Fast global convergence of natural policy gradient methods with entropy regularization. Preprint arXiv:2007.06558 (2020).
[5] T. Degris, P.M. Pilarski and R.S. Sutton, Model-free reinforcement learning with continuous action in practice, in 2012 American Control Conference (ACC). IEEE (2012) 2177–2182.
[6] S. Fujimoto, H. Hoof and D. Meger, Addressing function approximation error in actor-critic methods, in International Conference on Machine Learning, PMLR (2018) 1587–1596.
[7] T. Haarnoja, A. Zhou, P. Abbeel and S. Levine, Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor, in International conference on machine learning, PMLR (2018) 1861–1870.
[8] R. Islam, P. Henderson, M. Gomrokchi and D. Precup, Reproductibility of benchmarked deep reinforcement learning tasks for continuous control. Preprint arXiv:1708.04133 (2017).
[9] S.M. Kakade, A natural policy gradient, Adv. Neural Inf. Process. Syst. 14 (2001).
[10] V.R. Konda and J.N. Tsitsiklis, Actor-critic algorithms, in Advances in neural information processing systems, Citeseer (2000) 1008–1014.
[11] T.P. Lillicrap, J.J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver and D. Wierstra, Continuous control with deep reinforcement learning. Preprint arXiv:1509.02971 (2015).
[12] B. Liu, J. Liu, M. Ghavamzadeh, S. Mahadevan and M. Petrik, Finite-sample analysis of proximal gradient td algorithms. Preprint arXiv:2006.14364 (2020).
Y. ZHU AND L. YING

[13] J. Mei, C. Xiao, C. Szepesvari and D. Schuurmans, On the global convergence rates of softmax policy gradient methods, in International Conference on Machine Learning, PMLR (2020) 6820–6829.

[14] V. Mnih, A.P. Badia, M. Mirza, A. Graves, T. Lillicrap, T. Harley, D. Silver and K. Kavukcuoglu, Asynchronous methods for deep reinforcement learning, in International conference on machine learning, PMLR (2016) 1928–1937.

[15] R. Munos and C. Szepesvári, Finite-time bounds for fitted value iteration. J. Mach. Learn. Res. 9 (2008).

[16] O. Nachum, B. Dai, I. Kostrikov, Y. Chow, L. Li and D. Schuurmans, Algaedice: Policy gradient from arbitrary experience. Preprint arXiv:1912.02074 (2019).

[17] J. Peters, K. Mulling and Y. Altun, Relative entropy policy search, in Twenty-Fourth AAAI Conference on Artificial Intelligence (2010).

[18] J. Peters and S. Schaal, Policy gradient methods for robotics, in 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems. IEEE (2006) 2219–2225.

[19] J. Peters and S. Schaal, Natural actor-critic. Neurocomputing 71 (2008) 1180–1190.

[20] M. Schlegel, W. Chung, D. Graves, J. Qian and M. White, Importance resampling for off-policy prediction. Preprint arXiv:1906.04328 (2019).

[21] J. Schulman, S. Levine, P. Abbeel, M. Jordan and P. Moritz, Trust region policy optimization, in International conference on machine learning. PMLR (2015) 1889–1897.

[22] J. Schulman, F. Wolski, P. Dhariwal, A. Radford and O. Klimov, Proximal policy optimization algorithms. Preprint arXiv:1707.06347 (2017).

[23] D. Silver, G. Lever, N. Heess, T. Degris, D. Wierstra and M. Riedmiller, Deterministic policy gradient algorithms, in International conference on machine learning, PMLR (2014) 387–395.

[24] R.S. Sutton, A.G. Barto, Reinforcement learning: An introduction. MIT Press (2018).

[25] Z. Wang, V. Bapst, N. Heess, V. Mnih, R. Munos, K. Kavukcuoglu and N. de Freitas, Sample efficient actor-critic with experience replay. Preprint arXiv:1611.01224 (2016).

[26] Y. Zhu, Z. Izzo and L. Ying, Borrowing from the future: addressing double sampling in model-free control. Math. Sci. Mach. Learn. (2022) 1099–1136.

Please help to maintain this journal in open access!

This journal is currently published in open access under the Subscribe to Open model (S2O). We are thankful to our subscribers and supporters for making it possible to publish this journal in open access in the current year, free of charge for authors and readers.

Check with your library that it subscribes to the journal, or consider making a personal donation to the S2O programme by contacting subscribers@edpsciences.org.

More information, including a list of supporters and financial transparency reports, is available at https://edpsciences.org/en/subscribe-to-open-s2o.