ESAIM: Control, Optimisation and Calculus of Variations Will be set by the publisher
URL: http://www.emath.fr/cocv/

VARIATIONAL ACTOR-CRITIC ALGORITHMS [*], [**]

Yuhua Zhu[1] and Lexing Ying[2]

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.

2020 Mathematics Subject Classification. 90C40,93E20.

The dates will be set by the publisher.

1. Introduction

Consider a discounted Markov Decision Process (MDP) $\mathcal{M} = (\mathbb{S}, \mathbb{A}, P, r, \gamma)$. Here $\mathbb{S}$ is the state space and $\mathbb{A}$ is the action space. $\Delta(\mathbb{S})$ and $\Delta(\mathbb{A})$ denote the set of probability distributions over $\mathbb{S}$ and $\mathbb{A}$, respectively. $P : \mathbb{S} \times \mathbb{A} \to \Delta(\mathbb{S})$ is the transition kernel, $r : \mathbb{S} \times \mathbb{A} \to \mathbb{R}$ 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 $\pi : \mathbb{S} \to \Delta(\mathbb{A})$ 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 π with step $t = 0, 1, \ldots$:

$$V^\pi(s) = \mathop{\mathbb{E}}_{\substack{a_t\sim\pi(\cdot|s_t)\\ s_{t+1}\sim P(\cdot|s_t,a_t)}} \left[\sum_{t\geq 0} \gamma^t r(s_t, a_t) | s_0 = s \right]. \tag{1.1}$$

Keywords and phrases: Markov Decision Process, Reinforcement learning, Policy gradient, Optimal control

[*] *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.*

[1] Department of Mathematics and Halicioğlu Data Science Institute, University of California, San Diego, La Jolla, California, U.S.A; e-mail: yuz244@ucsd.edu

[2] Department of Mathematics, Stanford University, Stanford, California, U.S.A; e-mail: lexing@stanford.edu

The value function $V^{\pi}(s)$ also satisfies the Bellman equation [24],

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

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: $Q^{\pi}(s,a) = \mathbb{E}[\sum_{t\geq 0}\gamma^t r(s_t,a_t)|s_0 = s, a_0 = a]$. The two functions V^{π} and Q^{π} 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 π^* 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 π with gradient updates

$$\begin{aligned} \max_{\pi} \quad & \mathbb{E}_{s\sim\rho}V^{\pi}(s) \\ \text{s.t.} \quad & V^{\pi}(s) = \mathop{\mathbb{E}}_{\substack{a_t\sim\pi(\cdot|s_t)\\ s_{t+1}\sim P(\cdot|s_t,a_t)}} [r(s_t,a_t) + \gamma V^{\pi}(s_{t+1})|s_t = s], \end{aligned} \tag{1.2}$$

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:

$$\begin{aligned} \max_{\pi} \quad & \mathbb{E}_{s\sim\rho}V_{\lambda}^{\pi}(s) \\ \text{s.t.} \quad & V_{\lambda}^{\pi}(s) = \mathop{\mathbb{E}}_{\substack{a_t\sim\pi(\cdot|s_t)\\ s_{t+1}\sim P(\cdot|s_t,a_t)}} [r(s_t,a_t) - \lambda\log\pi(a_t|s_t) + \gamma V_{\lambda}^{\pi}(s_{t+1})|s_t = s]. \end{aligned} \tag{1.3}$$

Let π_{λ}^* be the regularized optimal policy for (1.3). Note that $\pi_0^* = \pi^*$ (the non-regularized optimal policy) when $\lambda = 0$ but π_{λ}^* is different from π^* when $\lambda > 0$. In what follows, we shall abbreviate the optimal value functions $V_{\lambda}^{\pi_{\lambda}^*}$ and V^{π^*} as V_{λ}^* and V^*, respectively.

The most direct way of solving the optimization problem (1.3) is to update the policy π according to the gradient $\nabla_{\pi}\mathbb{E}_{s\sim\rho}V_{\lambda}^{\pi}(s)$. The calculation of $\nabla_{\pi}\mathbb{E}_{s\sim\rho}V_{\lambda}^{\pi}(s)$ however involves computing the exact value function $V_{\lambda}^{\pi}(s)$ or $Q_{\lambda}^{\pi}(s,a)$ under the current policy π. With an accurate approximation of the value function $V_{\lambda}^{\pi}(s)$, this gradient-based method can achieve a linear convergence rate [1, 4, 13]. However, the calculation of value function $V_{\lambda}^{\pi}(s)$ 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 $V_{\lambda}^{\pi}(s)$, 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,

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

where $\rho \in \mathbb{R}^{|\mathbb{S}|}$ 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 π updates.

Besides, we pointed out that the vanilla gradient descent will lead to a direction increases V_λ^π 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 *flipping* method further accelerates the convergence by continuously maximizing the value function in the right direction.

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

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 (Section 2.1) and then in the model-free setting (Section 2.2). In Section 3, we study the fixed point of the algorithm for both non-regularized (Section 3.1) and the regularized (Section 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:

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

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

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

Hereafter, we omit the subscript θ of π_θ for simplicity. The vector $r^\pi = (r_s^\pi)_{s\in\mathbb{S}} \in \mathbb{R}^{|\mathbb{S}|}$ in (2.1) is the reward under policy π with the component $r_s^\pi = \sum_a r_{sa}\pi_{sa}$, where r_{sa} is the immediate reward at (s,a). The matrix $P^\pi \in \mathbb{R}^{|\mathbb{S}|\times|\mathbb{S}|}$ in (2.1) is the transition matrix under policy π with the entry $P_{st}^\pi = \sum_a \pi_{sa}P_{st}^a$, where for each a, the matrix $P^a \in \mathbb{R}^{|\mathbb{S}|\times|\mathbb{S}|}$ is the state transition matrix under action a. The vector $\mathcal{H}(\pi) = (\mathcal{H}(\pi_s))_{s\in\mathbb{S}} \in \mathbb{R}^{|\mathbb{S}|}$ 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_λ^π 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_λ^π. Thus, combining the two terms of the RHS of (2.1) yields that (2.1) maximizes the true value function V_λ^π as in (1.3).

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

$$\begin{aligned} \partial_{V_s} E \quad &= -\rho_s + \beta(\ell_s\rho_s - \gamma \sum_t P_{ts}^\pi \ell_t \rho_t)) &&:= G_{V_s}, \\ (F^+ \nabla_\theta E)_{sa} \quad &= \rho_s \beta \ell_s \left[-\gamma \sum_t P_{st}^a V_t - r_{sa} + \lambda \log \pi_{sa} \right] + c_s &&:= G_{\theta_{sa}}^{(0)}, \end{aligned} \tag{2.2}$$

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

$$\ell(V, \pi) = (I - \gamma P^\pi)V - r^\pi + \lambda \mathcal{H}(\pi). \tag{2.3}$$

Here the natural gradient is used for the policy updates in (2.2) with the Fisher information matrix $F = \mathbb{E}_{s\sim\rho, a\sim\pi}\left[(\nabla_\theta \log \pi)(\nabla_\theta \log \pi)^\top\right]$. 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 $F^+ \nabla_{\theta_{sa}} E$). The vector $c = (c_s)_{s\in\mathbb{S}} \in \mathbb{R}^{|\mathbb{S}|}$ 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

$$V_s^{k+1} = V_s^k - \eta_V G_{V_s}(V^k, \pi^k), \quad \pi_{sa}^{k+1} \propto \pi_{sa}^k e^{-\eta_\pi G_{\theta_{sa}}^{(0)}(V^k, \pi^k)}, \tag{2.4}$$

where η_V and η_π 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 π^*, the trajectory towards the minimizer is often not optimal. When $\ell_s < 0$, the path from π towards π^* may detour if the algorithm is directed according to $G_{\theta_{sa}}^{(0)}$. 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 π tries to minimize the residual norm $\|\ell\|_\rho$ in the policy space. When $\ell < 0$, V underestimates the true value function V_λ^π. 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_λ^π, which is undesirable. On the other hand, when ℓ is non-negative, V overestimates V_λ^π. Hence, $\nabla_\pi \|\ell\|_\rho^2$ will lead to a direction that increases V_λ^π 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 $\partial_{\pi_{sa}} E$ of the objective function $E(V, \pi)$ in π has the same form as $G_{\theta_{sa}}^{(0)}$ in (2.2). Therefore,

$$\begin{aligned} -\partial_{\pi_{sa}} E(V, \pi) = &-\beta(\ell \odot \rho)^\top \partial_{\pi_{sa}} \left[(I - \gamma P^\pi)(V - (I - \gamma P^\pi)^{-1}(r^\pi - \lambda \mathcal{H}(\pi)))\right] \\ = &-\beta(\ell \odot \rho)^\top \partial_{\pi_{sa}} \left[(I - \gamma P^\pi)(V - V_\lambda^\pi)\right] \\ = &\beta(\ell \odot \rho)^\top (I - \gamma P^\pi) \partial_{\pi_{sa}}(V_\lambda^\pi) + \beta\gamma(\ell \odot \rho)^\top \partial_{\pi_{sa}}(P^\pi)(V - V_\lambda^\pi), \end{aligned} \tag{2.5}$$

where $\ell \odot \rho$ is the entry-wise product of ℓ and ρ, 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_λ^π for sufficiently large β. In fact, for any fixed π, the local fixed point of the V updates satisfies $G_{V_s} = 0$, where G_{V_s} is defined in (2.2). This implies that

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

where $\tilde{\rho}_s = 1/\rho_s$. Hence, the second term of the RHS of (2.5) is of order $O(1/\beta)$. When β is large, the gradient $-\partial_{\pi_{sa}} E(V, \pi)$ is dominated by the first term $\beta(\ell \odot \rho)^\top (I - \gamma P^\pi) \partial_{\pi_{sa}}(V_\lambda^\pi)$, which is equivalent to $\beta \ell \rho (1 - \gamma) \partial_\pi (V_\lambda^\pi)$

in the one-dimensional case. Note that $\partial_\pi(V_\lambda^\pi)$ is the steepest ascent direction for maximizing V_λ^π. Therefore, when $\ell < 0$, the term $\beta\ell\rho(1-\gamma)\partial_\pi(V_\lambda^\pi)$ is the opposite direction of the steepest ascent, which implies that the gradient descent algorithm based on $-\partial_\pi E$ does not move towards maximizing V_λ^π. This illustrates why the algorithm (2.4) can take a detour to the optimal policy π^*.

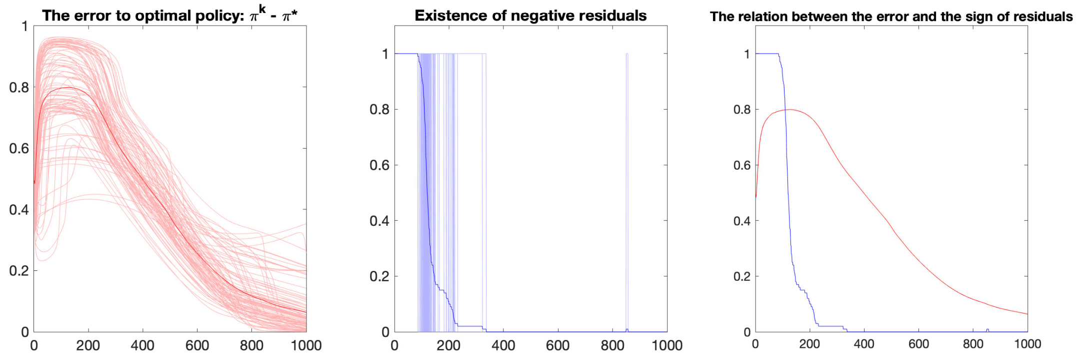

FIGURE 1. The left plot shows the error $\pi^k - \pi^*$ from the vanilla gradient descent method (2.2), i.e., Algorithm 1 with $h^{(0)}(\ell_s) = \ell_s$. We test the algorithm for an MDP with 5 states and 2 actions, and set $\beta = 10, \lambda = 0$ and the learning rate $\eta_V = \eta_\pi = 1/(4\beta)$. 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.

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

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

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

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

$$G_{\theta_{sa}}^{(2)} = \rho_s\beta|\ell_s|\left[-\gamma\sum_t P_{st}^a V_t - r_{sa} + \lambda\log\pi_{sa}\right].$$

From the analysis of (2.5), we see that the vanilla gradient descent with $G_{\theta_{sa}}^{(0)}$ in (2.2) would make the policy worse locally. Intuitively, the clipping method with $G_{\theta_{sa}}^{(1)}$ stops updating the policy when $\ell_s < 0$, while the flipping method further improves the policy π because $G_{\theta_{sa}}^{(2)}$ 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:

$$V^{k+1} = V^k - \eta_V G_V(V^k, \pi^k), \quad \pi_{sa}^{k+1} \propto \pi_{sa}^k \exp(-\eta_\pi G_{\theta_{sa}}^{(i)}(V^k, \pi^k)), \quad i = 0, 1, 2, \tag{2.6}$$

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

$$\begin{aligned} G_{V_s}(V, \pi) =& -\rho_s + \beta \left(\ell_s \rho_s - \gamma \sum_t P_{ts}^\pi \ell_t \rho_t \right), \\ G_{\theta_{sa}}^{(i)}(V, \pi) =& \beta \rho_s h^{(i)}(\ell_s) \left[-\gamma \sum_t P_{st}^a V_t - r_{sa} + \lambda \log \pi_{sa} \right] + c_s, \quad i = 0, 1, 2, \end{aligned} \tag{2.7}$$

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

$$h^{(0)} = x, \quad h^{(1)}(x) = x \mathbb{1}_{x>0}, \quad h^{(2)}(x) = |x|. \tag{2.8}$$

Here $\ell = (\ell_s)_{s \in \mathbb{S}}$ 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);
Random initialization of $V_0, \pi_0$
while $V, \theta$ do not converge do
    $\ell \leftarrow (I - \gamma P^\pi) V - r^\pi + \lambda \mathcal{H}(\pi)$;
    $V_s \leftarrow V_s - \eta_V(-\rho_s + \beta(\ell_s \rho_s - \gamma \sum_t P_{ts}^\pi \ell_t \rho_t))$;
    $\pi_{sa} \leftarrow \pi_{sa} \exp\left[-\eta_\pi \beta \rho_s h^{(i)}(\ell_s)(-\gamma \sum_t P_{st}^a V_t - r_{sa} + \lambda \log \pi_{sa})\right]$, where $h^{(i)}$ is defined in (2.8);
    $\pi_{sa} \leftarrow \frac{1}{\sum_b \pi_{sb}} \pi_{sa}$;
end while
```

We would like to point out that the three variants $G_{\theta_{sa}}^{(i)}$ for $i = 0, 1, 2$ coincide when $\ell_s \geq 0$. Lemma 3.2 demonstrates that ℓ_s is larger than 0 when (V, π) achieves the fixed point of the algorithm. Therefore, the three variants $G_{\theta_{sa}}^{(i)}$ 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_λ^π; 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_λ^π. 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^π 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, π) 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 π_b and the target policy π when approximating the term $P^\pi V$. 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:

$$\mathop{\mathbb{E}}_{(s,a)\sim\rho}\left[-Q(s,a)+\frac{\beta}{2}\left(Q(s,a)-\mathop{\mathbb{E}}_{s'\sim P^a(\cdot|s,a)}\left[\gamma\sum_a(Q(s',a)-\lambda\log\pi(s',a))\pi(s',a)|s,a\right]-r(s,a)\right)^2\right], \quad (2.9)$$

where $\rho>0$ is the positive stationary distribution from behavior policy π_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 ω and $\pi(s,a,\theta)$ is parametrized by θ, the updates of (Q,π) are according to the following unbiased estimates of the gradients:

$$\begin{aligned}(G_Q)_t &= -\nabla_\omega Q^k(s_t,a_t)+\beta L_t\left(\nabla_\omega Q^k(s_t,a_t)-\gamma\sum_a\nabla_\omega Q^k(s'_{t+1},a)\pi^k(s'_{t+1},a)\right),\\ (G_\pi^{(i)})_t &= \beta\hat{h}^{(i)}(L_t)\left(-\gamma\sum_a(Q^k(s'_{t+1},a)-\lambda\log\pi^k(s'_{t+1},a)-\lambda)\nabla_\theta\pi^k(s'_{t+1},a)\right),\end{aligned} \quad (2.10)$$

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,

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

Here the next state s'_{t+1} in G_Q and G_π 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,

$$\hat{h}^{(1)}(L_t)=L_t\mathbb{1}_{\hat{\ell}_{s_t}>0},\quad \hat{h}^{(2)}(L_t)=\begin{cases}L_t, & \hat{\ell}_{s_t}>0\\ -L_t, & \hat{\ell}_{s_t}<0\end{cases},\quad \text{where } \hat{\ell}_s=\frac{1}{|\{s_t=s\}|}\sum_{s_t=s}L_t. \quad (2.11)$$

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 $\mathbb{E}[h^{(i)}(L_t)|s_t=s]\neq h^{(i)}(\ell_s)$. Instead, $h^{(i)}(\ell_s)$ is estimated in two steps: first, one approximates the Bellman residual ℓ_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* $\mathcal{B}_\pi\nu(s,a)=r(s,a)+\gamma\mathbb{E}_{s'\sim P^a(\cdot|s,a)}[\sum_a\nu(s',a)\pi(s',a)|s,a]$ *and view* $\nu(s,a)$ *as* $Q(s,a)$*, one finds that* (2.9) *is equivalent to equation (14) in [16] up to a constant by setting* $f_*(x)=x^2/2$*. 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)

```
Require: $\eta_V, \eta_\pi$: learning rates; $\beta$: prefactor; $M$: batch size;
    $Q(s,a,\omega), \pi(s,a,\theta)$: parametrized approximation of $Q(s,a), \pi(s,a)$;
    $\{s_t, a_t, r_t\}_{t=0}^T$: trajectory generated from off-policy $\pi_b$;
Random initialization of $\theta_0, \omega_0$, $k = 0$
while $\omega, \theta$ do not converge do
    $j \leftarrow 1$, $k \leftarrow k+1$
    for $t = (k-1)M+1, \cdots, kM$ do
        $s_j = s_t$
        $L_j = Q(s_t, a_t, \omega) - r_t - \gamma(V(s_{t+1}) - \lambda\mathcal{H}(s_{t+1}))$
        $s'_{t+1} \leftarrow s_t + (s_{t+2} - s_{t+1})$
        $G_Q^j = -\nabla_\omega Q(s_t, a_t, \omega) + \beta L_j(\nabla_\omega Q(s_t, a_t, \omega) - \gamma \sum_a \nabla_\omega Q(s'_{t+1}, a_t, \omega)\pi(s'_{t+1}, a, \theta))$
        $G_\pi^j = \beta\left(-\gamma \sum_a (Q(s'_{t+1}, a_t, \omega) - \lambda \log \pi(s'_{t+1}, a, \theta) - \lambda)\nabla_\theta \pi(s'_{t+1}, a, \theta)\right)$
        $j \leftarrow j+1$
    end for
    $G_Q \leftarrow \frac{1}{M}\sum_{j=1}^M G_Q^j$; $\omega \leftarrow \omega - \eta_Q G_Q$
    $\hat{\ell}_s \leftarrow \sum_{s_j = s} L_j$
    $G_\pi^{(i)} \leftarrow \frac{1}{M}\sum_{j=1}^M \hat{h}^{(i)}(L_j) G_\pi^j$; $\theta \leftarrow \theta - \eta_\pi G_\pi^{(i)}$, where $\hat{h}^{(i)}$ is defined in (2.11)
    $V(s) \leftarrow \sum_a Q(s,a,\omega)\pi(s,a,\theta)$; $\mathcal{H}(s) \leftarrow \sum_a \pi(s,a,\theta)\log\pi(s,a,\theta)$
end while
```

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

$$(f_{sa}^i)_t = \gamma\pi^k(s,a)\beta\hat{h}^{(i)}(L_t)\left[V^k(s) - Q^k(s,a) + \lambda\log\pi^k(s,a) - \lambda\mathcal{H}(\pi_s^k)\right]\mathbb{1}_{s = s'_{t+1}},$$

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

3. Fixed Point Estimates

We define (V^∞, π^∞) as the *fixed point* of Algorithm 1 if

$$V^\infty = V^\infty - \eta_V G_V(V^\infty, \pi^\infty), \quad \pi_{sa}^\infty \propto \pi_{sa}^\infty \exp\left(-\eta_\pi G_{\theta_{sa}}^{(i)}(V^\infty, \pi^\infty)\right), \tag{3.1}$$

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 π^∞ is exactly the optimal policy π^* when β is sufficiently large; for the regularized MDP, i.e., $\lambda > 0$, the fixed point π^∞ is close to π^* for large β and small λ.

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

Before analyzing the fixed point of the algorithm, we state some basic properties of the matrix $(I - \gamma P^\pi)$ 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^∞, π^∞). This implies that $G^{(i)}_{\theta_{sa}}$ takes the same form at the fixed point (V^∞, π^∞). Hereafter, we shall omit the index i of $G^{(i)}_{\theta_{sa}}$ for notational simplicity.

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

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

For any constant c,

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

See Appendix C for the proof.

Lemma 3.2. *The fixed point (V^∞, π^∞) 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,

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

This leads to,

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

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

Since the three variants $G^{(i)}_{\theta_{sa}}$ with $i = 0, 1, 2$ defined in (2.7) are the same when $\ell_s > 0$ and ℓ_s is positive at the fixed point (V^∞, π^∞), 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 β_0, such that for all $\beta > \beta_0$, the fixed point π^∞ of the policy updates is the optimal policy $\pi^* = \operatorname{argmax}_\pi V^\pi = \operatorname{argmax}_\pi (I - \gamma P^\pi)^{-1} r$. For simplicity, we assume that the distribution in (2.1) is the uniform distribution, i.e., $\rho_s = 1/|\mathbb{S}|$ in this section. The results can be extended to general distribution ρ (see Remark 3.7 for details). Besides, we always assume that the action gap is strictly positive, i.e., let $a^*_s = \max_a(r_{sa} + \gamma \sum_t P^a_{st} V^*_t)$, then

$$\max_{a \ne a^*_s}(r_{sa} + \gamma \sum_t P^a_{st} V^*_t) < r_{sa^*_s} + \gamma \sum_t P^{a^*_s}_{st} V^*_t, \quad \forall s \in \mathbb{S}.$$

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

$$V(\pi) = (I - \gamma P^\pi)^{-1} r + \frac{1}{\beta}(I - \gamma P^\pi)^{-1}(I - \gamma P^\pi)^{-\top}\mathbf{1}.$$

When β 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 α, such that $|V^*_s - V^\infty_s| \le \alpha$ for $\forall s \in \mathbb{S}$, then $\pi^\infty = \pi^*$. Combining the above lemmas, one concludes in Theorem 3.6 that the fixed point π^∞ is the optimal policy π^* as long as $\beta > \frac{|\mathbb{S}|}{(1-\gamma)^2\alpha}$, where α is a constant related to the optimal solution (V^*, π^*). Note that the lower bound for the prefactor β is not sharp, and we shall see in Section 4 that the algorithm converges numerically to the optimal policy π^* with much smaller β.

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

$$\begin{cases} G_V(V^\infty, \pi^\infty) = 0, \\ \pi^\infty_{sa} = \begin{cases} 1, & a = a_s; \\ 0, & a \neq a_s, \end{cases} \quad \textit{where } a_s = \operatorname*{argmax}_a \left(\gamma \sum_t P^a_{st} V^\infty_t + r_{sa} \right). \end{cases} \tag{3.4}$$

Proof. Since V is updated as follows

$$V_{k+1} = V_k - \eta_V G_V(V_k, \pi_k),$$

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

$$(\pi_{k+1})_{sa} \propto (\pi_k)_{sa} \exp\left(-\eta_\pi G_{\theta_{sa}}(V_k, \pi_k)\right),$$

then $\lim_{k\to\infty} \pi_k = \pi^\infty$ with π^∞ stated in the lemma.

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

$$(\pi_{k+1})_{sa} \propto (\pi_k)_{sa} \exp\left(\eta_\pi \left(\gamma \sum_t P^a_{st} (V_k)_t + r_{sa} \right) \right).$$

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

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

where a_s is defined in (3.4). Then one has

$$\begin{cases} (\pi_{k+1})_{sa} \propto (\pi_k)_{sa}, & \text{for } a = a_s; \\ (\pi_{k+1})_{sa} \propto (\pi_k)_{sa} \exp(f_{sa}(V^\infty)), & \text{for } a \neq a_s, \end{cases}$$

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

$$\begin{cases} (\pi_k)_{sa} \propto (\pi_0)_{sa}, & \text{for } a = a_s; \\ (\pi_k)_{sa} \propto (\pi_0)_{sa} \exp(k f_{sa}(V^\infty)), & \text{for } a \neq a_s. \end{cases}$$

Notice that as $f_{sa}(V^\infty) < 0$, $\lim_{k\to\infty} (\pi_0)_{sa} \exp(k f_{sa}(V^\infty)) = 0$. 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. □

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

$$\begin{cases} V^* = r^{\pi^*} + \gamma P^{\pi^*} V^*, \\ \pi^*_{sa} = \begin{cases} 1, & a = a^*_s; \\ 0, & a \neq a^*_s, \end{cases} \quad \textit{where } a^*_s = \operatorname*{argmax}_a \left(r_{sa} + \gamma \sum_t P^a_{st} V^*_t \right). \end{cases} \tag{3.5}$$

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

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

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

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

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

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

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

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^∞ is close to V^*, then a_s and a^*_s defined in Lemmas 3.4 and 3.3 are the same.

Proof. If $|V^\infty_t - V^*_t| \le \epsilon$, then

$$V^*_t - \epsilon \le V^\infty_t, \quad -(V^* + \epsilon) \le -V^\infty_t, \quad \text{for } \forall t \in \mathbb{S},$$

which further leads to,

$$r_{sa^*_s} + \gamma \sum_t P^{a^*_s}_{st} (V^*_t - \epsilon) \le r_{sa^*_s} + \gamma \sum_t P^{a^*_s}_{st} V^\infty_t,$$
$$-r_{sa'} - \gamma \sum_t P^{a'}_{st} (V^* + \epsilon) \le -r_{sa'} - \gamma \sum_t P^{a'}_{st} V^\infty_t, \quad \text{for } a' \notin a^*_s.$$

Summing the two inequality together gives,

$$r_{sa^*_s} + \gamma \sum_t P^{a^*_s}_{st} V^*_t - \left(r_{sa'} + \gamma \sum_t P^{a'}_{st} V^* \right) - 2\gamma\epsilon \le r_{sa^*_s} + \gamma \sum_t P^{a^*_s}_{st} V^\infty_t - \left(r_{sa'} + \gamma \sum_t P^{a'}_{st} V^\infty_t \right).$$

Since the LHS $\ge \epsilon' - 2\gamma\epsilon = \epsilon' - \frac{2\gamma}{3}\epsilon' > 0$, one has

$$r_{sa^*_s} + \gamma \sum_t P^{a^*_s}_{st} V^\infty_t - \left(r_{sa'} + \gamma \sum_t P^{a'}_{st} V^\infty_t \right) > 0, \quad \text{for } \forall a' \neq a^*_s.$$

The above inequality implies that $a^*_s = \operatorname{argmax}_a \left(r_{sa} + \gamma \sum_t P^a_{st} V^\infty_t \right) = a_s$, which completes the proof.

□

Theorem 3.6. *Let (V^*, π^*) be the maximum and maximizer of (1.2), and α is a positive constant s.t. $\alpha < \frac{1}{3} g(V^*, \pi^*)$. There exists a constant $\beta_0 = \frac{|\mathbb{S}|}{(1-\gamma)^2 \alpha}$, such that $\forall \beta > \beta_0$, the fixed point (V^∞, π^∞) of Algorithm 1 is close to (V^*, π^*) in the sense that*

$$|V^\infty_s - V^*_s| \le \alpha, \text{ for } \forall s \in \mathbb{S}, \quad \pi^\infty = \pi^*,$$

where

$$g(V^*,\pi^*) = \min_{s\in\mathbb{S}}\left(r_{sa_s^*} + \gamma\sum_t P_{st}^{a_s^*} V_t^* - \max_{a\neq a_s^*}(r_{sa} + \gamma\sum_t P_{st}^a V_t^*)\right)$$

with $a_s^* = \mathrm{argmax}\,(r_{sa} + \gamma\sum_t P_{st}^a V_t^*)$.

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

$$(I-\gamma P^{\pi^\infty})V^\infty = r^{\pi^\infty} + \frac{1}{\beta}(I-\gamma P^{\pi^\infty})^{-\top}\mathbf{1}. \tag{3.7}$$

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

$$(I-\gamma P^{\pi^\infty})V^\infty - r^{\pi^\infty} - (I-\gamma P^{\pi^*})V^* + r^{\pi^*} = \frac{1}{\beta}(I-\gamma P^{\pi^\infty})^{-\top}\mathbf{1}. \tag{3.8}$$

By the definition of π^∞ and π^* 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

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

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

$$(I-\gamma P^{\pi^\infty})(V^\infty - V^*) \leq \frac{|\mathbb{S}|}{\beta(1-\gamma)}\mathbf{1}, \quad (I-\gamma P^{\pi^*})(V^\infty - V^*) > \mathbf{0}.$$

Applying (3.3) of Proposition 3.1 to the above two inequalities gives $\mathbf{0} \leq \mathbf{V}^\infty - \mathbf{V}^* \leq \frac{|\mathbb{S}|}{\beta(\mathbf{1}-\gamma)^\mathbf{2}}\mathbf{1}$. By Lemma 3.5, when $\frac{|\mathbb{S}|}{\beta(1-\gamma)^2} \leq \alpha = \frac{1}{3}g(V^*,\pi^*)$, then $a_s = a_s^*$, which implies $\pi^\infty = \pi^*$. □

Remark 3.7. *For general* ρ*, one has*

$$(I-\gamma P^{\pi^\infty})V^\infty = r^{\pi^\infty} + \frac{1}{\beta}\tilde{\rho}\odot\left[(I-\gamma P^{\pi^\infty})^{-\top}\rho\right].$$

where $(\tilde{\rho})_s = 1/\rho_s$*. The proof for Theorem 3.6 can be easily extended to general* ρ*. The main difference is the bound for the second term of the RHS of the above equation. By applying Propsition 3.1, one can bound*

$$\mathbf{0} \leq \frac{1}{\beta}(I-\gamma P^\pi)^{-1}\left[\tilde{\rho}\odot\left[(I-\gamma P^\pi)^{-\top}\rho\right]\right] \leq \frac{1/\min_s \rho_s}{\beta(1-\gamma)^2}\mathbf{1},$$

for $\forall\pi$*. Hence, Theorem 3.6 still holds for general* ρ *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

$$(\pi_\lambda^*)_s = \underset{\pi_s\in\Delta(\mathbb{A})}{\mathrm{argmax}}\left(r_s^\pi + \gamma\sum_t P_{st}^\pi (V_\lambda^\pi)_t - \lambda\mathcal{H}(\pi_s)\right), \quad (\pi_\lambda^*)_{sa} \propto \exp\left(\frac{1}{\lambda}\left(r_{sa} + \gamma\sum_t P_{st}^a (V_\lambda^*)_t\right)\right).$$

In this section, we prove that the fixed point V^∞ converges to the regularized optimal value function $\lim_{\beta\to\infty} V^\infty = V^*_\lambda$ as the prefactor β converges to infinity. On the other hand, for sufficiently large prefactor $\beta > \beta_0$, the fixed point π^∞ will be close to the non-regularized optimal policy π^* if the entropy constant λ is small. However, when λ is relatively large, the fixed point π^∞ will be close to the regularized optimal policy π^*_λ. For simplicity, we assume the distribution in (2.1) is the uniform distribution in this section. The results can be extended to general distribution ρ.

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^*_λ. Both lemmas will be useful in the proof of Theorem 3.10. In this section, we always assume that the learning rate $\eta_\pi > 0$ for the policy updates is sufficiently small, so that $\eta_\pi \beta \ell_s \lambda$ is always less than 1 and larger than 0.

Lemma 3.8. *If $\pi =$ soft-max(θ) and $\mu =$ soft-max(ω) with $\theta, \omega \in \mathbb{R}^d$, then the KL divergence between the probability distribution π and μ is $D_{KL}(\pi|\omega) \le 2\max_a |\theta_a - \omega_a|$.*

See Appendix D for the proof.

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

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

and

$$[(I - \gamma P^{\pi^*_\lambda})(V - V^*_\lambda)]_s \ge \sum_a (\pi_{sa} - (\pi^*_\lambda)_{sa})(\gamma \sum_t P^a_{st} V_t + r_{sa} - \lambda \log \pi_{sa}), \tag{3.9}$$

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 $(I - \gamma P^{\pi^*_\lambda})V^* + \lambda\mathcal{H}(\pi^*) = r^{\pi^*_\lambda}$, subtracting it from $G_V(V, \pi) = 0$ gives

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

Note that the regularized optimal policy π^*_λ can also be represented by

$$(\pi^*_\lambda)_s = \underset{\pi_s \in \Delta(\mathbb{A})}{\operatorname{argmax}}(r^\pi_s + \gamma \sum_t P^\pi_{st}(V^*_\lambda)_t - \lambda\mathcal{H}(\pi_s)).$$

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

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

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

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

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

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

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

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

which is equivalent to,

$$\begin{aligned}&[(I-\gamma P^{\pi^*_\lambda})(V-V^*_\lambda)]_s+\sum_a(\pi_{sa}-(\pi^*_\lambda)_{sa})(-\gamma\sum_t P^a_{st}V_t-r_{sa}+\lambda\log\pi_{sa})\\=&\lambda\sum_a(\pi^*_\lambda)_{sa}(\log(\pi^*_\lambda)_{sa}-\log\pi_{sa})+\frac{1}{\beta}\left[(I-\gamma P^\pi)^{-\top}\mathbf{1}\right]_s.\end{aligned}$$

Note that the first term of the RHS is the KL divergence of π^*_λ from π, 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 $\beta>\frac{2|\mathbb{S}|}{\epsilon(1-\gamma)^2}$, then the distance between the fixed point (V^∞,π^∞) of Algorithm 1 with $\lambda>0$ and the maximum and maximizer $(V^*_\lambda,\pi^*_\lambda)$ of (1.3) can be bounded by*

$$|V^\infty_s-(V^*_\lambda)_s|<\frac{\epsilon}{2},\ \textit{for}\ \forall s\in\mathbb{S},\quad D_{KL}(\pi^*_\lambda||\pi^\infty)\le\frac{\epsilon\gamma}{\lambda}.$$

If one further has $\epsilon<\frac{1}{3}g(V^*,\pi^*)$ and $\lambda<\frac{\epsilon(1-\gamma)}{2\log(|\mathbb{A}|)}$, then the fixed point π^∞ 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

$$g_{sa}=r_{sa_s}+\gamma\sum_t P^{a_s}_{st}V^\infty_t-\left(r_{sa}+\gamma\sum_t P^a_{st}V^\infty_t\right)\begin{cases}=0, & a=a_s;\\>0, & a\neq a_s,\end{cases}\qquad a_s=\operatorname*{argmax}_a P^a_{st}V^\infty_t.$$

Remark 3.11. *From the above theorem, one can see that as β approaches infinity, the fixed point V^∞ approaches the regularized optimal value function V^*_λ. However, when λ is small, the difference between the fixed point π^∞ and the regularized optimal policy π^*_λ could be amplified by $\frac{1}{\lambda}$. On the other hand, by Taylor expansion, the difference between π^∞ and the non-regularized optimal policy π^* 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 λ 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,

$$\beta\rho_s\ell_s\left[-\gamma\sum_t P^a_{st}V_t-r_{sa}+\lambda\log\pi_{sa}\right]+c_s=0.\tag{3.11}$$

It is equivalent to

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

Let

$$a_s = \operatorname*{argmax}_a \left(r_{sa} + \gamma \sum_t P_{st}^a V_t^\infty \right), \tag{3.12}$$

then π^∞ can be written as

$$\pi_{sa}^\infty \propto \exp\left(-\frac{\gamma}{\lambda} g_{sa} \right), \tag{3.13}$$

where

$$g_{sa} = r_{sa_s} + \gamma \sum_t P_{st}^{a_s} V_t^\infty - \left(r_{sa} + \gamma \sum_t P_{st}^a V_t^\infty \right) \begin{cases} = 0, & a = a_s; \\ > 0, & a \neq a_s. \end{cases}$$

On the other hand, by the equality (3.11), one has $\gamma \sum_t P_{st}^a V_t^\infty + r_{sa} - \lambda \log \pi_{sa}^\infty = f_s$, where f_s is a value independent of a. Inserting the above π^∞ into the π in (3.9) gives,

$$[(I - \gamma P^{\pi_\lambda^*})(V^\infty - V_\lambda^*)]_s \geq f_s \sum_a (\pi_{sa}^\infty - (\pi_\lambda^*)_{sa}) = 0,$$

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

$$\mathbf{0} < V^\infty - V_\lambda^* < \frac{|\mathbb{S}|}{\beta(1-\gamma)^2} \mathbf{1}. \tag{3.14}$$

On the other hand, $(\pi_\lambda^*)_{sa}$ can be represented by

$$(\pi_\lambda^*)_{sa} \propto \exp\left(\frac{1}{\lambda} \left(r_{sa} + \gamma \sum_t P_{st}^a (V_\lambda^*)_t \right) \right).$$

Since for $\forall s \in \mathbb{S}$,

$$\begin{aligned} &\max_a \left| \frac{1}{\lambda} \left(r_{sa} + \gamma \sum_t P_{st}^a (V_\lambda^*)_t \right) - \frac{1}{\lambda} \left(r_{sa} + \gamma \sum_t P_{st}^a V_t^\infty \right) \right| \\ =& \max_a \left| \frac{\gamma}{\lambda} \sum_t P_{st}^a ((V_\lambda^*)_t - V_t^\infty) \right| < \frac{\gamma |\mathbb{S}|}{\lambda \beta (1-\gamma)^2}, \end{aligned}$$

by Lemma 3.8, one has

$$D_{\mathrm{KL}}((\pi_\lambda^*)_s | \pi_s^\infty) \leq \frac{2\gamma|\mathbb{S}|}{\lambda\beta(1-\gamma)^2}. \tag{3.15}$$

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

$$\mathbf{0} < V^\infty - V_\lambda^* < \frac{\epsilon}{2} \mathbf{1}, \quad D_{\mathrm{KL}}(\pi_\lambda^* | \pi^\infty) < \frac{\epsilon\gamma}{\lambda},$$

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

For the second part, note that

$$V_\lambda^* = V_\lambda^{\pi_\lambda^*} \geq V_\lambda^{\pi^*} = (I - \gamma P^{\pi^*})^{-1} r^\pi + (I - \gamma P^{\pi^*})^{-1}(-\lambda H(\pi^*)) \geq V^{\pi^*} = V^*;$$

$$V_\lambda^* = V^{\pi_\lambda^*} + (I - \gamma P^{\pi_\lambda^*})^{-1}(-\lambda H(\pi_\lambda^*)) \leq V^{\pi^*} + \frac{1}{1-\gamma} \max_s (-\lambda \mathcal{H}(\pi_\lambda^*)_s) \leq V^* + \frac{\lambda}{1-\gamma} \log(|\mathbb{A}|),$$

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

$$\mathbf{0} < V^*_\lambda - V^* \leq \frac{\lambda}{1-\gamma}\log(|\mathbb{A}|)\mathbf{1}.$$

Combining it with the inequality (3.14), one has

$$|V^\infty_t - V^*_t| \leq |V^\infty_t - (V^*_\lambda)_t| + |(V^*_\lambda)_t - V^*_t| < \frac{|\mathbb{S}|}{\beta(1-\gamma)^2} + \frac{\lambda}{1-\gamma}\log(|\mathbb{A}|).$$

Therefore, when $\beta > \frac{2|\mathbb{S}|}{\epsilon(1-\gamma)^2}$ and $\lambda < \frac{\epsilon(1-\gamma)}{2\log(|\mathbb{A}|)}$, then $|V^\infty_t - V^*_t| < \epsilon$ for all $t \in \mathbb{S}$. As proved in Lemma 3.5, when $|V^\infty_t - V^*_t| < \epsilon = \frac{1}{3}g(V^*, \pi^*)$ for all $t \in \mathbb{S}$, then $a_s = a^*_s$ with a^*_s defined in Lemma 3.4 and a_s defined in (3.12). By the definition of π^* in Lemma 3.4 and π^∞ in (3.13), one has

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

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

□

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 π^*. 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 $\mathbb{S} = \left\{s_k = \frac{2\pi k}{n}\right\}_{k=0}^{n-1}$. The transition dynamics is given by

$$\tilde{s}_{t+1} = \mathrm{mod}\left(s_t + \frac{2\pi}{n}a_t + \sigma Z_t,\, n\right), \quad s_{t+1} = \begin{cases} \underset{i\in Z, i\in[0,n-1]}{\mathrm{argmin}} \ |\tilde{s}_{t+1} - i|, & \text{if } \tilde{s}_{t+1} \in [0, n-1/2), \\ 0, \quad \text{if } \tilde{s}_{t+1} \in [n-1/2, n), \end{cases} \tag{4.1}$$

where $a_t \in \mathbb{A} = \{\pm 1\}$ 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 η_V and η_π are both set to be $1/(4\beta)$ for all cases.

First, for both the regularized and non-regularized method, the difference between π_k and the true optimal policy π^* 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.

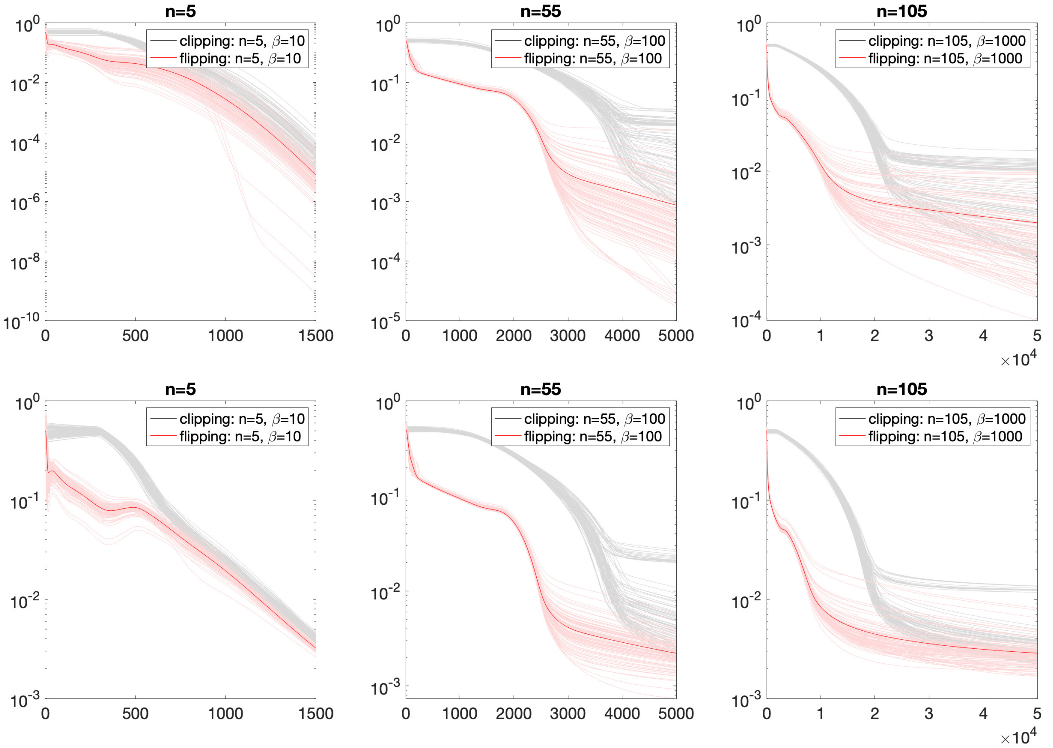

Figure 2. The plots show the error $\|\pi_k - \pi^*\|_{L^1}$ from Algorithm 1 for the size of the state space $n = 5, 55, 105$ from left to right. The first row is for the non-regularized method, while the second row is for the regularized method with $\lambda = 0.1$. We run 100 simulations for each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

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. The off-policy $\pi_b(a|s) = 1/2$ for $\forall s \in \mathbb{S}$ 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 π^* becomes larger as the prefactor β 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 β to be convergent compared with clipping.

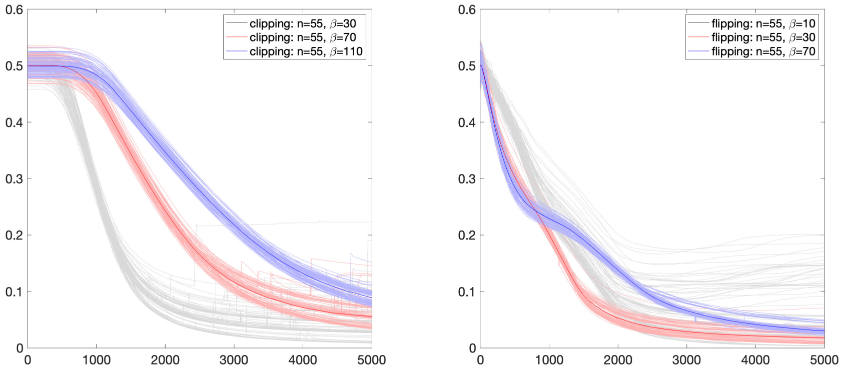

FIGURE 3. The plots show the error $\|\pi_k - \pi^*\|_{L^1}$ from Algorithm 2 with $\lambda = 0.1$ for different prefactors β. The left one is the results of the clipping method , while the right one is the results of the flipping method. The grey, pink and blue lines represent 100 simulations with β 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 π^*.

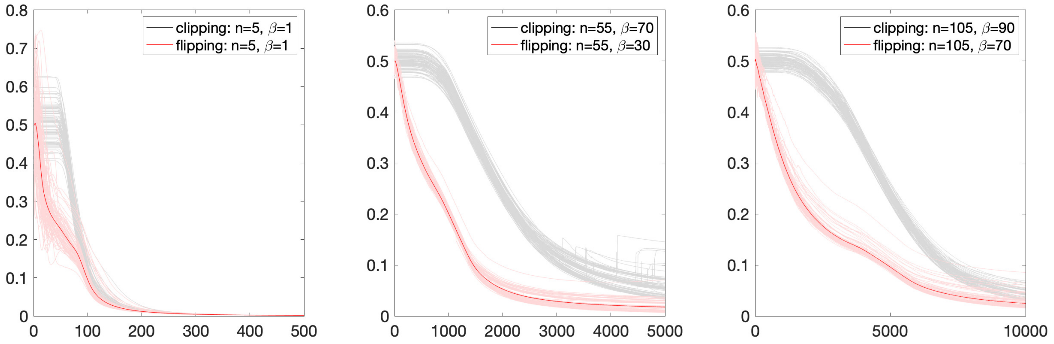

FIGURE 4. The plots show the error $\|\pi_k - \pi^*\|_{L^1}$ from Algorithm 2 with $\lambda = 0.1$ for the size of the state space $n = 5, 55, 105$ from left to right. The grey and pink lines represent 100 simulations for clipping and flipping methods, respectively. The mean of each case is plotted in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

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 β smaller and the learning rate larger to encourage stochasticity.

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

$$\pi_{sa}^{k+1} = \left(\pi_{sa}^{k}\right)^{1-\frac{\lambda\eta_\pi}{1-\gamma}} \exp\left(\frac{\eta Q^{\pi^k}(s,a)}{1-\gamma}\right), \tag{4.2}$$

where $Q^{\pi^k}(s,a)$ is estimated by solving the residual Bellman minimization problem

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

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

$$\begin{aligned} &w_t = Q^j(s_t,a_t) - r_t - \gamma\left(V^j(s_{t+1}) - \lambda H^k(s_{t+1})\right), \\ &G_t(s,a) = w_t \mathbb{1}_{s=s_t,a=a_t} - \gamma\pi(s_{t+1},a) w_t \mathbb{1}_{s=s'_{t+1}}, \\ &Q^{j+1} = Q^j - \eta_Q \sum_{t=(j-1)M+1}^{jM} G_t; \quad V^{j+1}(s) = \sum_a Q^{j+1}(s,a)\pi^k(s,a), \end{aligned} \tag{4.3}$$

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 $\beta = 80, (\eta_Q)_k = (\eta_\pi)_k = \min(0.999^k \frac{30}{C}, \frac{20}{C})$ for $n = 55$.

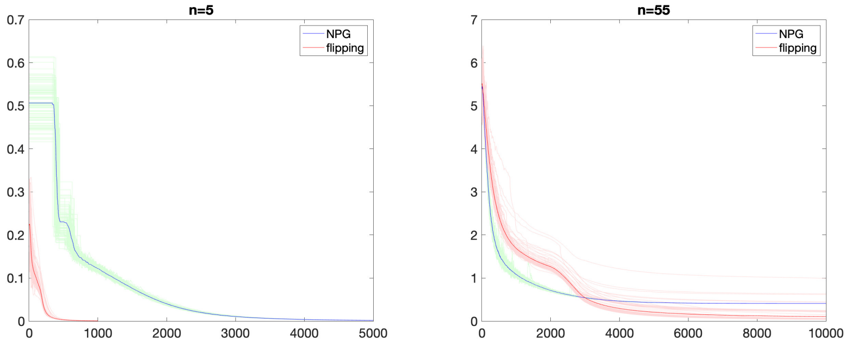

FIGURE 5. The plots show the comparison of the error $\|\pi_k - \pi^*\|_{L^1}$ between the flipping method and the NPG method (4.2) - (4.3) for the size of the state space $n = 5$ and $n = 55$. Both use the regularized objective function, i.e., $\lambda = 0.1$. The green and pink lines represent 100 simulations for the NPG method and the flipping method, respectively. The mean of each case is plotted in blue and red.

For $n = 5$, though both methods converge to the optimal policy π^*, the flipping method converges faster than the NPG method. For $n = 55$, NPG converges to the regularized optimal policy π^*_λ, while our method converges to a policy close to the true optimal policy π^* 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 Figure 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.

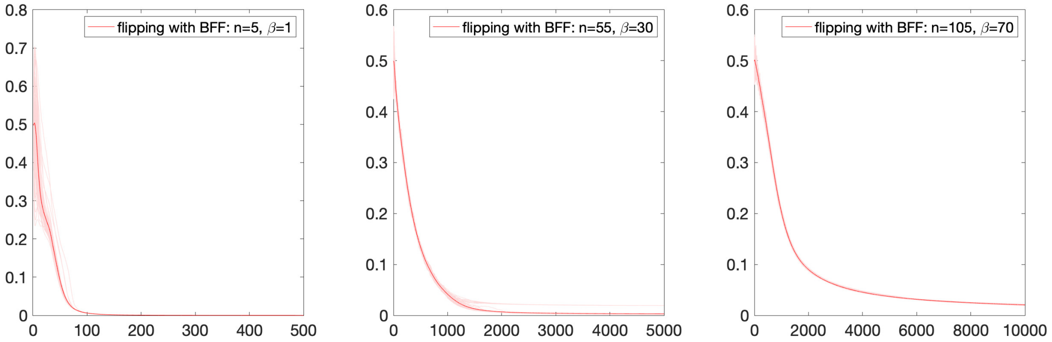

Figure 6. The plots show the error $\|\pi_k - \pi^*\|_{L^1}$ from the flipping method of Algorithm 2 with BFF for the size of the state space $n = 5, 55, 105$ from left to right. The pink lines represent 100 simulations, and their mean is plotted in red.

4.2. Example 2

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

$$
\begin{aligned}
&\tilde{s}_{t+1} \leftarrow s_t + (1 + \sigma Z_t) a_t, \\
&(\tilde{s}_{t+1})_k \leftarrow \operatorname{mod}\left((\tilde{s}_{t+1})_k, n_k\right), \quad k = 1, 2, \\
&(s_{t+1})_k = \begin{cases} \operatorname*{argmin}_{i \in Z, i \in [0, n_k - 1]} |(\tilde{s}_{t+1})_k - i|, & \text{if } (\tilde{s}_{t+1})_k \in [0, n_k - 1/2), \\ 0, \quad \text{if } (\tilde{s}_{t+1})_k \in [n_k - 1/2, n_k), \end{cases}
\end{aligned}
$$

where $a_t \in \mathbb{A} = \{(\pm 1, 0), (0, \pm 1)\}$ and $Z_t \sim N(0, 1)$. $n_1 = n_2 = 7$, the reward is set to be $r(s_{ij}) = 2 + \sin\left(\frac{2\pi i}{n_1}\right) + \cos\left(\frac{2\pi j}{n_2}\right)$, and the noise σ 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 $\eta_Q = \eta_\pi = \frac{30}{\beta}$. The error is plotted out in the L_1 norm. 98% of the simulations converge to the true optimal policy π^*. Note that the value function in the right plot of Figure 7 is the value function V^{π^k} under the policy π^k, which is different from the V^k in the algorithm. It shows that the policy indeed consistently maximizes the value function V^π.

References

[1] Alekh Agarwal, Sham M Kakade, Jason D Lee, and Gaurav Mahajan. Optimality and approximation with policy gradient methods in markov decision processes. In *Conference on Learning Theory*, pages 64–66. PMLR, 2020.

[2] Jing An, Lexing Ying, and Yuhua Zhu. Why resampling outperforms reweighting for correcting sampling bias. *International Conference on Learning Representations (ICLR)*, 2021.

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

[4] Shicong Cen, Chen Cheng, Yuxin Chen, Yuting Wei, and Yuejie Chi. Fast global convergence of natural policy gradient methods with entropy regularization. *arXiv preprint arXiv:2007.06558*, 2020.

[5] Thomas Degris, Patrick M Pilarski, and Richard S Sutton. Model-free reinforcement learning with continuous action in practice. In *2012 American Control Conference (ACC)*, pages 2177–2182. IEEE, 2012.

[6] Scott Fujimoto, Herke Hoof, and David Meger. Addressing function approximation error in actor-critic methods. In *International Conference on Machine Learning*, pages 1587–1596. PMLR, 2018.

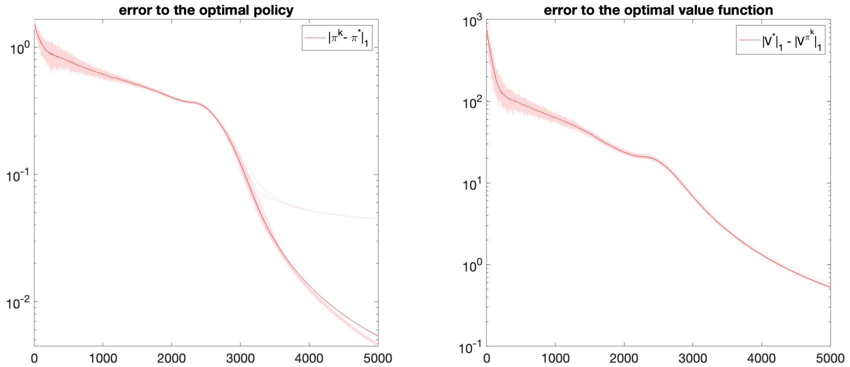

FIGURE 7. The above plots show the convergence of the flipping method to the optimal policy π^* and optimal value function V^*. The pink lines represent 100 simulations that correspond to different off-policy trajectories and initializations of the parameters. The mean is plotted in red.

[7] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International conference on machine learning*, pages 1861–1870. PMLR, 2018.

[8] Riashat Islam, Peter Henderson, Maziar Gomrokchi, and Doina Precup. Reproducibility of benchmarked deep reinforcement learning tasks for continuous control. *arXiv preprint arXiv:1708.04133*, 2017.

[9] Sham M Kakade. A natural policy gradient. *Advances in neural information processing systems*, 14, 2001.

[10] Vijay R Konda and John N Tsitsiklis. Actor-critic algorithms. In *Advances in neural information processing systems*, pages 1008–1014. Citeseer, 2000.

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

[12] Bo Liu, Ji Liu, Mohammad Ghavamzadeh, Sridhar Mahadevan, and Marek Petrik. Finite-sample analysis of proximal gradient td algorithms. *arXiv preprint arXiv:2006.14364*, 2020.

[13] Jincheng Mei, Chenjun Xiao, Csaba Szepesvari, and Dale Schuurmans. On the global convergence rates of softmax policy gradient methods. In *International Conference on Machine Learning*, pages 6820–6829. PMLR, 2020.

[14] Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In *International conference on machine learning*, pages 1928–1937. PMLR, 2016.

[15] Rémi Munos and Csaba Szepesvári. Finite-time bounds for fitted value iteration. *Journal of Machine Learning Research*, 9(5), 2008.

[16] Ofir Nachum, Bo Dai, Ilya Kostrikov, Yinlam Chow, Lihong Li, and Dale Schuurmans. Algaedice: Policy gradient from arbitrary experience. *arXiv preprint arXiv:1912.02074*, 2019.

[17] Jan Peters, Katharina Mulling, and Yasemin Altun. Relative entropy policy search. In *Twenty-Fourth AAAI Conference on Artificial Intelligence*, 2010.

[18] Jan Peters and Stefan Schaal. Policy gradient methods for robotics. In *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2219–2225. IEEE, 2006.

[19] Jan Peters and Stefan Schaal. Natural actor-critic. *Neurocomputing*, 71(7-9):1180–1190, 2008.

[20] Matthew Schlegel, Wesley Chung, Daniel Graves, Jian Qian, and Martha White. Importance resampling for off-policy prediction. *arXiv preprint arXiv:1906.04328*, 2019.

[21] John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. In *International conference on machine learning*, pages 1889–1897. PMLR, 2015.

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

[23] David Silver, Guy Lever, Nicolas Heess, Thomas Degris, Daan Wierstra, and Martin Riedmiller. Deterministic policy gradient algorithms. In *International conference on machine learning*, pages 387–395. PMLR, 2014.

[24] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction.* MIT press, 2018.

[25] Richard S Sutton, David A McAllester, Satinder P Singh, Yishay Mansour, et al. Policy gradient methods for reinforcement learning with function approximation. In *Advances in Neural Information Processing Systems*, volume 99, pages 1057–1063. Citeseer, 1999.

[26] Ziyu Wang, Victor Bapst, Nicolas Heess, Volodymyr Mnih, Remi Munos, Koray Kavukcuoglu, and Nando de Freitas. Sample efficient actor-critic with experience replay. *arXiv preprint arXiv:1611.01224*, 2016.

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

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

[29] Yuhuai Wu, Elman Mansimov, Roger B Grosse, Shun Liao, and Jimmy Ba. Scalable trust-region method for deep reinforcement learning using kronecker-factored approximation. *Advances in neural information processing systems*, 30:5279–5288, 2017.

[30] Zhuoran Yang, Yongxin Chen, Mingyi Hong, and Zhaoran Wang. Provably global convergence of actor-critic: A case for linear quadratic regulator with ergodic cost. *Advances in Neural Information Processing Systems*, 2019.

[31] Yuhua Zhu, Zach Izzo, and Lexing Ying. Borrowing from the future: Addressing double sampling in model-free control. *Mathematical and Scientific Machine Learning*, pages 1099–1136, 2022.

Appendices

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

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

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

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,

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

a_t' is a sample from $\pi_b(s_t, a)$ that is uncorrelated with a_t, and s_{t+1}' is a sample for the next state when action a_t' is taken at state s_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.

Algorithm 3 V-formulation

```
Require: $\eta_V, \eta_\pi$: prefactor; $\beta$: penalty constant; $M$: batch size;
    $V(s, \omega), \pi(s, a, \theta)$: parametrized approximation of $V(s), \pi(s, a)$;
    $\{s_t, a_t, r_t\}_{t=0}^T$: trajectory generated from off-policy $\pi_b$;
Random initialization of $\theta_0, \omega_0$, $k = 0$
while $\omega, \theta$ do not converge do
    $j \leftarrow 0$, $k \leftarrow k + 1$
    for $t = (k-1)M + 1, \cdots, kM$ do
        $s_j = s_t$
        $L_j = V(s_t, \omega) - r_t - \gamma \tau(s_t, a_t) V(s_{t+1}, \omega) + \lambda \mathcal{H}(s_t)$
        $s_{t+1}' \leftarrow s_t + (s_{t+2} - s_{t+1})$;    $a_t' \leftarrow a_{t+1}$
        $G_V^j = -\nabla_\omega V(s_t, \omega) + \beta L_t (\nabla_\omega V(s_t, \omega) - \gamma \tau(s_t, a_t') \nabla_\omega V(s_{t+1}', \omega))$
        $G_\pi^j = \beta \left( -\gamma \tau(s_t, a_t') V(s_{t+1}', \omega) \nabla_\theta \log \pi(s_t, a_t, \theta) + \lambda \sum_a (\log \pi(s_t, a, \theta) + 1) \nabla_\theta \pi(s_t, a, \theta) \right)$
        $j \leftarrow j + 1$
    end for
    $G_V \leftarrow \frac{1}{M} \sum_{j=1}^M G_V^j$;    $\omega \leftarrow \omega - \eta_V G_V$
    $\hat{\ell}_s \leftarrow \sum_{s_j = s} L_j$
    $G_\pi^{(i)} \leftarrow \frac{1}{M} \sum_{j=1}^M \hat{h}^{(i)}(L_j) G_\pi^j$;    $\theta \leftarrow \theta - \eta_\pi G_\pi^{(i)}$,    where $\hat{h}^{(i)}$ is defined in (2.11)
    $\tau(s, a) \leftarrow \frac{\pi(s,a,\theta)}{\pi_b(s,a,\theta)}$; $\mathcal{H}(s) \leftarrow \sum_a \pi(s, a, \theta) \log \pi(s, a, \theta)$
end while
```

B. **Fixed point of Algorithm 1 with $\lambda = 0$ is not stochastic policy**

Lemma B.1. *Assume $|\mathbb{A}| > 2$, the null space of $P^a - P^{a'}$ is the linear space spanned by $\mathbf{1}$ for all $a \neq a'$, and the reward is not a constant, i.e., $r_{sa} \not\equiv r$. When β 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 $r^a = (r_{sa})_{s\in\mathbb{S}}$ is an $|\mathbb{S}|$-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 $r^a - r^{a'} = r^a - r^{a''} = c \neq \mathbf{0}$, then $r^a = r^{a''} = \mathbf{0}$. Therefore, when $|\mathbb{A}| \geq 3$, the value of $r^{a'} - r^a$ can be separated into two different cases. The first case is that

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

The second case is that

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

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 $\mathbf{0}$. Let $\mathcal{N}$ be the null space of $(P^a - P^{a'})$ for $\forall a \neq a'$, which is a linear space spanned by $\mathbf{1}$. If the projection of $r^a - r^{a'}$ onto $\mathcal{N}$ is not equal to $\mathbf{0}$, 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 $\mathcal{N}$ are both equal to $\mathbf{0}$, 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 $V = c_1\mathbf{1}$ is the only solution to (B.1). Given that P^π is transition matrix, $P^\pi \mathbf{1} = \mathbf{1}$. Plugging it into $G_V = \mathbf{0}$ yields,

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

Multiplying $\mathbf{1}^\top$ 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

$$\bar{r} - r + \frac{1}{\beta(1-\gamma)} = \frac{1}{\beta}(I - \gamma P^\pi)^{-\top}\mathbf{1}.$$

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 Propsition 3.1, which gives contradiction. Therefore, $(G_V, G_\theta) \neq (0,0)$ for the second case (B.3). □

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 $\forall s$. 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

$$\begin{aligned}\sum_s c_s \mathbb{1}_{x_s \leq 0} =& \sum_s x_s \mathbb{1}_{x_s \leq 0} - \gamma \sum_{s,t} P_{ts} x_t \mathbb{1}_{x_s \leq 0} \\ =& \sum_s x_s \mathbb{1}_{x_s \leq 0} - \gamma \sum_{s,t} P_{ts} x_t \mathbb{1}_{x_t \leq 0} \mathbb{1}_{x_s \leq 0} - \gamma \sum_{s,t} P_{ts} x_t \mathbb{1}_{x_t > 0} \mathbb{1}_{x_s \leq 0} \\ \leq& \sum_s x_s \mathbb{1}_{x_s \leq 0} + \gamma \sum_t \left(\sum_s P_{ts} \mathbb{1}_{x_s \leq 0} \right) (-x_t \mathbb{1}_{x_t \leq 0}) \\ \leq& (1-\gamma) \sum_s x_s \mathbb{1}_{x_s \leq 0} \leq 0.\end{aligned}$$

The first inequality holds because the last term on the second line is always ≤ 0. The second inequality is due to $\sum_s P_{ts} \mathbb{1}_{x_s \leq 0} \leq \sum_s P_{ts} = 1$ for $\forall t$. 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

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

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

For $(I - \gamma P)x \leq c\mathbf{1}$, let $x_s = \max_i x_i$, then the s-th component of $(I - \gamma P)x \leq c\mathbf{1}$ 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, $x \leq x_s \leq \frac{c}{1-\gamma}\mathbf{1}$. □

D. Proof of Lemma 3.8

Proof. First note that

$$\log(\pi_a) - \log(\mu_a) = \theta_a - \omega_a - \left(\log\left(\sum_b \exp(\theta_b) \right) - \log\left(\sum_b \exp(\omega_b) \right) \right). \tag{D.1}$$

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 $\forall \theta, \omega \in \mathbb{R}^d$,

$$\begin{aligned}&\left|\log\left(\sum_b \exp(\theta_b)\right) - \log\left(\sum_b \exp(\omega_b)\right)\right| = |f(\theta) - f(\omega)| = |\langle \theta - \omega, \nabla_x f(x)\rangle| \\ &\leq \max_a |\theta_a - \omega_a| \, \|\nabla_x f(x)\|_1 = \max_a |\theta_a - \omega_a|,\end{aligned}$$

where x is a convex combination of θ and ω. Applying the above inequality into (D.1) yields

$$\log(\pi_a) - \log(\mu_a) \leq 2 \max_a |\theta_a - \omega_a|.$$

Therefore,

$$D_{\mathrm{KL}}(\pi|\omega) = \sum_a \pi_a(\log(\pi_a) - \log(\omega_a)) \leq 2b \sum_a \pi_a = 2b.$$

□