ACCELERATING PRIMAL-DUAL METHODS FOR REGULARIZED MARKOV DECISION PROCESSES*

HAOYA LI†, HSANGFU YU‡, LEXING YING§, AND INDERJIT DHILLON¶

Abstract. Entropy regularized Markov decision processes have been widely used in reinforcement learning. This paper is concerned with the primal-dual formulation of the entropy regularized problems. Standard first-order methods suffer from slow convergence due to the lack of strict convexity and concavity. To address this issue, we first introduce a new quadratically convexified primal-dual formulation. The natural gradient ascent descent of the new formulation enjoys a global convergence guarantee and exponential convergence rate. We also propose a new interpolating metric that further accelerates the convergence significantly. Numerical results are provided to demonstrate the performance of the proposed methods under multiple settings.

Key words. Reinforcement learning, Markov decision process, primal-dual method, entropy regularization

AMS subject classifications. 49M29, 65B99, 65K10, 68T05, 90C40, 90C47, 93D30

1. Introduction.

1.1. Setup. Consider an infinite-horizon Markov decision process (MDP) \( \mathcal{M} = (S, A, P, r, \gamma) \), where \( S \) is a set of states of the Markov chain and \( A \) is a set of actions. \( P \) is a transition probability tensor with \( P_{ss'}^a \) being the probability of transitioning from state \( s \) to state \( s' \) when taking action \( a \), \( r \) is a reward matrix with \( r_{sa} \) being the reward obtained when taking action \( a \) at state \( s \), and \( \gamma \in (0, 1) \) is the discount factor. In this paper, we assume that the state space \( S \) and the action space \( A \) are finite.

A policy \( \pi \) is a randomized strategy over the actions at each state, i.e., for each state \( s \), \( \pi_{sa} \) is the probability of choosing action \( a \) at \( s \). For a given policy, the value function \( v_\pi \in \mathbb{R}^{\vert S \vert} \) is a vector defined as

\[
(v_\pi)_s := \mathbb{E} \sum_{k=0}^{\infty} \left( \gamma^k r_{s_k a_k} \mid s_0 = s \right),
\]

where the expectation is taken over all possible trajectories \( \{(s_k, a_k)\}_{k \geq 0} \) starting from \( s_0 = s \) following the policy \( \pi \). The value function \( v_\pi \) satisfies the well-known Bellman equation [4]

\[
(I - \gamma P_\pi)v_\pi = r_\pi,
\]

where \( (P_\pi)_{ss'} := \sum_{a \in A} \pi_{sa} P_{s a s'} \), \( (r_\pi)_s := \sum_{a \in A} \pi_{sa} r_{sa} \), and \( I \) is the identity operator. In a Markov decision problem, the goal is to find the optimal policy \( \pi^* \) such that

\[
v_{\pi^*}(s) \geq v_\pi(s), \quad \forall s \in S,
\]

for any other policy \( \pi \). The corresponding optimal value function \( v_{\pi^*} \) will also be referred to as \( v^* \) in this paper. The existence of \( v^* \) and \( \pi^* \) is guaranteed by the theory of MDP [29].

In recent studies, entropy regularization has been widely used in MDP problems to encourage exploration and enhance the robustness [27, 10, 12, 2, 1, 24, 7, 42]. With the entropy regularization, the value function is defined by

\[
(v_\pi)_s := \mathbb{E} \sum_{k=0}^{\infty} \left( \gamma^k (r_{s_k a_k} - \tau \log \pi_{s_k a_k}) \mid s_0 = s \right),
\]

where \( \tau > 0 \) is the regularization coefficient. \( v_\pi \) satisfies the regularized Bellman equation

\[
(I - \gamma P_\pi)v_\pi = r_\pi - \tau h_\pi,
\]

*Submitted to the editors DATE.
†Stanford University, Stanford, CA (lihaoya@stanford.edu).
‡Amazon Search (hsangfu@amazon.com).
§Stanford University, Stanford, CA (lexing@stanford.edu).
¶University of Texas at Austin and Google, work done while at Amazon Search (inderjit@cs.utexas.edu).
where \( h_\pi \) is a vector in \( \mathbb{R}^{|S|} \) with each entry \((h_\pi)_s\) given by the negative Shannon entropy of \((\pi_{sa})_{a \in A}\)

\[
(h_\pi)_s = \sum_{a \in A} \pi_{sa} \log \pi_{sa}.
\]

Here we overload the notation \( v_\pi \) for the regularized value function and for the rest of the paper \( v_\pi \) shall always denote the regularized value function \((1.3)\) unless otherwise specified. For the entropy regularized MDP (see [12]), there exists a unique optimal policy \( \pi^* \), such that

\[
(1.5) \quad v^*(s) := v_{\pi^*}(s) \geq v_\pi(s), \quad \forall s \in S,
\]

for any other policy \( \pi \).

Without loss of generality, the reward \( r_{sa} \) is assumed to be nonnegative throughout this paper. This can be guaranteed by adding to the rewards a sufficiently large constant \( C \). Note that such a uniform shift keeps the optimal policy \( \pi^* \) unchanged and shifts \( v^* \) by a constant \( \frac{C}{\gamma} \).

1.2. Primal-dual formulation. Entropy regularized MDPs enjoy regularized linear programming formulations, in the primal, dual, and primal-dual forms. In this paper, we are concerned with the primal-dual formulation (see, for example, [27, 41]):

\[
\min_{v \in \mathbb{R}^{|S|}} \max_{u \in \mathbb{R}^{|S| \times |A|}} \sum_{s \in S} e_s v_s + \sum_{s \in S, a \in A} u_{sa} (r_{sa} - ((I - \gamma P_a) v)_s) - \tau \sum_{s \in S, a \in A} u_{sa} \log (u_{sa}/\tilde{u}_s),
\]

where \( \tilde{u}_s := \sum_{a \in A} u_{sa} \). The policy \( \pi \) is related to \( u \) via the relationship

\[
\pi_{sa} = u_{sa}/\tilde{u}_s.
\]

The main advantage of working with the primal-dual formulation is that the transition matrix \( P_a \) appears linearly in the objective function of the primal-dual problem. This linearity brings an important benefit when a stochastic gradient method is used to solve the primal-dual formulation: an unbiased estimator of the transition matrix \( P_a \) guarantees an unbiased estimator for the gradient-based update rule. This avoids the famous double-sampling problem [35] that affects any formulation that performs a nonlinear operation to the transition matrix \( P_a \). Examples of these affected formulations include the primal formulation, where a nonlinear max or exponentiation operator is applied to \( P_a \), and the dual formulation, where the inverse of \( I - \gamma P_a \) is needed. From this perspective, the primal-dual formulation is convenient in the model-free setting, where the transition probability tensor can only be estimated from samples and is thus inherently noisy.

In what follows, we shall simplify the notation by denoting \( K_a = I - \gamma P_a \) and \( K_\pi = I - \gamma P_\pi \). With this simplification, the primal-dual problem can be rewritten more compactly as

\[
(1.6) \quad \min_{v} \max_{u} \sum_{s} e_s v_s + \sum_{sa} u_{sa} (r_{sa} - (K_a v)_s) - \tau \sum_{sa} u_{sa} \log (u_{sa}/\tilde{u}_s).
\]

Though theoretically appealing, the primal-dual formulation \((1.6)\) often poses computational challenges because it is a minimax optimization. Newton-type methods are often impractical to apply because either \( P_a \) is only accessible via samples or its size is too large for practical inversion. A close look at the objective function of \((1.6)\) suggests that it is linear with respect to both the value function \( v \) and the dual variable \( u \) in the radial direction. This lack of strict convexity/concavity makes it difficult for the first-order methods to converge.

1.3. Contributions. To overcome this difficulty, this paper proposes a quadratically convexified reformulation of \((1.6)\) that shares the same solution with \((1.6)\) and an interpolating natural gradient ascent descent method that significantly speeds up the convergence. More specifically, the main contributions of this paper are listed as follows:

- We propose a new quadratically convexified primal-dual formulation in which the linear weighted sum \( e^T v \) of \((1.6)\) is replaced with a quadratic term \( \frac{\alpha}{2} \| v \|^2 \). The surprising feature is that the solution \((v^*, u^*)\) is unchanged and is independent of the hyperparameter \( \alpha > 0 \).
We prove that the vanilla natural gradient ascent descent (NGAD) of this quadratically convexified problem enjoys a Lyapunov function [23] and converges linearly. To the best of our knowledge, this is the first quadratically convexified primal-dual formulation of Markov decision problems.

- We propose an interpolating natural gradient ascent descent (INGAD) by introducing a new interpolating metric for the value function. The corresponding Lyapunov function is constructed and the convergence of the new dynamics is proved. The acceleration is verified by numerical tests under multiple settings.

1.4. Related work. Regarding the primal-dual formulation, the first primal-dual learning algorithm is given in [39]. A follow-up work [38] leverages the binary-tree data structure and adaptive importance sampling techniques to reduce the complexity. The convergence result for these two papers is however only for the average of all the policies rather than the policy obtained in the last iteration. In these papers, no regularization is used in the formulation and no preconditioner is used in the iterative update scheme. As a comparison, the current paper proves a last-iteration convergence result with the help of the Lyapunov method and entropy regularization, and derives an interpolating metric that accelerates the algorithm. Various studies have been carried out following the primal-dual formulation in [39]. For example, a modified form with the Q-function is proposed in [20], and the corresponding primal-dual type algorithm is derived. An extension to the infinite-horizon average-reward setting is provided in [37], but only the average-case convergence result is given. A later work [8] further extended this method to the function approximation setting. A comprehensive review of the primal-dual methods in the average reward problem in [37] to address large-scale problems. However, it is worth noting that no (entropy) regularization is used in the formulation and no preconditioners, gradient-type methods can take exponential time to converge [21].

Besides the primal-dual formulations, the discussion below briefly touches on the primal and dual formulations. For the entropy regularized Markov decision process, the primal formulation [41] takes the form

\begin{equation}
\begin{align*}
\phi_s = \tau \log \left( \sum_a \exp \left( \frac{r_{sa} + \gamma \sum_{s'} P_{ass'} \phi_{s'}}{\tau} \right) \right), \tag{1.7}
\end{align*}
\end{equation}

which leads to a value iteration algorithm. Let \( \varphi(v) : \mathbb{R}^{[S]} \rightarrow \mathbb{R}^{[S]} \) be the fixed-point map such that

\[ \varphi(v)_s = \tau \log \left( \sum_{a \in A} \exp \left( \tau^{-1} (r_{sa} + \gamma \sum_{s' \in S} P_{ass'} v_{s'}) \right) \right). \]

By calculating the derivative matrix, we have

\[ \|D\varphi(v)\|_{\infty} = \max_s \sum_{s'} |(D\varphi(v))_{ss'}| = \max_s \sum_a \gamma \sum_{s'} \frac{\sum_{s''} P_{ass''} \exp \left( \tau^{-1} (r_{sa} + \gamma \sum_{s''} P_{ass''} v_{s''}) \right)}{\sum_a \exp \left( \tau^{-1} (r_{sa} + \gamma \sum_{s''} P_{ass''} v_{s''}) \right)} = \gamma. \]

Hence \( \varphi \) is a contraction map and converges to a fixed point, which is the solution to (1.7) at a linear rate \( O(\gamma^T) \), where \( T \) is the number of iterations. After obtaining the optimal value function \( v \), the corresponding policy \( \pi \) is given by [41]:

\begin{equation}
\begin{align*}
\pi_{sa} = \frac{\exp \left( \tau^{-1} (r_{sa} + \gamma \sum_{s' \in S} P_{ass'} v_{s'}) \right)}{\sum_a \exp \left( \tau^{-1} (r_{sa} + \gamma \sum_{s' \in S} P_{ass'} v_{s'}) \right)} = \exp \left( \tau^{-1} \left( r_{sa} - \sum_{s'} (I - \gamma P_a)_{ss'} v_{s'} \right) \right), \tag{1.8}
\end{align*}
\end{equation}
As a result of the aforementioned double-sampling problem, the value-iteration algorithm based on (1.7) is mainly used in the model-based setting, but due to the nice properties of \( \phi \), it appears as an important ingredient in various other algorithms. For example, in [3] and [30], the authors use the function \( \phi \) as an alternative softmax operator and form a Q-learning type algorithm, and in [20], the function \( \phi \) appears as a result of the inner optimization of an entropy regularized trust region-type formulation and is used to form the loss function. In [10], the mean squared regularized Bellman error is employed to establish the optimization problem.

An alternative way to solve a regularized Markov decision problem in the model-based setting is the dual formulation [41], in which one seeks a policy \( \pi \) that solves the following optimization problem:

\[
\max_{\pi} e^T v_{\pi} := e^T (I - \gamma P_{\pi})^{-1} (r_{\pi} - \tau h_{\pi}),
\]

where \( e \succ 0 \) is a weight vector. By the existence and uniqueness of the optimal value function and optimal policy and the optimality (1.5), it is clear that any choice of \( e \) leads to the optimal policy and the optimal value function. A variety of policy gradient algorithms can be used to solve the dual problem. Examples include [40, 36, 15, 32, 31, 33], to mention only a few. Recently, [22] proposes a quasi-Newton policy gradient algorithm, where an approximate Hessian of the objective function in (1.9) is used as a preconditioner for the gradient, resulting in a quadratic convergence rate by better fitting the problem geometry.

The word primal-dual also appears in other types of formulations where the dual variables do not represent the policy. For example, in [11], the authors apply the natural policy gradient method to constrained MDPs (CMDPs), where the dual variables are the multipliers of the constraints. Similarly, in [9], the dual variables come from the constraints in CMDPs. In this paper, the Lyapunov method is used to give a theoretical analysis of the natural gradient flow of the method we propose. The idea of Lyapunov methods has also been applied to discrete time control problems [17, 16] and to discrete Markovian systems [25]. Recently it has also been used to address the safety problem, where safety usually appears as additional constraints in the model [28, 9, 5], and the Lyapunov function is usually defined on the state space and is used explicitly in the policy iteration or in finding the controller.

### 1.5. Notations

For a vector \( x \in \mathbb{R}^d \), \( \text{diag}(x) \) denotes a diagonal matrix with size \( d \times d \) and the \( k \)-th diagonal element being \( x_k \), \( 1 \leq k \leq d \). For \( u \in \mathbb{R}^{(|S||A|)} \), we denote the \(( (s - 1)|A| + a ) \)-th element as \( u_{sa} \). While \( u_a \) denotes the vector in \( \mathbb{R}^{|A|} \) with the \( a \)-th element being \( u_{sa} \), \( u_a \) denotes the vector in \( \mathbb{R}^{|S|} \) with the \( s \)-th element being \( u_{sa} \). The states of the MDP are typically referred to with \( s, s', \) and \( s'' \) while the actions are referred to by \( a \) and \( a' \). The vector with length \( d \) and all elements equal to 1 is denoted by \( 1_d \), and the subscript \( d \) is often omitted when there is no ambiguity. The \( d \)-by-\( d \) identity matrix is denoted by \( I_d \), again with the subscript \( d \) often omitted when there is no ambiguity. For a matrix \( B \), \( B^H \) denotes its Hermitian transpose. If a scalar function is applied to a vector, then the result is defined element-wise unless otherwise specified, e.g., for \( x \in \mathbb{R}^d \), \( \exp(x) \in \mathbb{R}^d \) with \( \exp(x)_k = \exp(x_k) \) for \( 1 \leq k \leq d \).

### 1.6. Contents

The rest of the paper is organized as follows. Section 2 derives the quadratically convexified primal-dual formulation, proves its equivalence with (1.6), and shows that the vanilla NGAD of the new formulation converges linearly using a Lyapunov function method. Section 3 introduces an interpolating metric by leveraging the flexibility of the underlying metric described by the block diagonal part of the Hessian. The convergence rate of the INGAD based on this new interpolating metric is significantly improved. We also provide a Lyapunov-style proof for global convergence and an analysis of the exponential convergence rate in the last-iterate sense. Finally, Section 4 demonstrates the numerical performance of these proposed natural gradient methods.

2. Quadratically convexified primal-dual formulation.
2.1. Formulation. In what follows, we use $E_0(v, u)$ to denote the objective of the standard entropy regularized primal-dual formulation

$$\min_v \max_u E_0(v, u) := \sum_s c_s v_s + \sum_{s,a} u_{sa} (r_{sa} - (K_a v)_s) - \tau \sum_{s,a} u_{sa} \log \frac{u_{sa}}{u_s}.$$  

(2.1)

Since it is linear in $v$ and linear along the radial direction of $u$, first-order optimization methods typically experience slow convergence. To address the issue in the $v$ variable, we propose a quadratically convexified primal-dual formulation:

$$\min_v \max_u E(v, u) := \frac{\alpha}{2} \sum_s v_s^2 + \sum_{s,a} u_{sa} (r_{sa} - (K_a v)_s) - \tau \sum_{s,a} u_{sa} \log \frac{u_{sa}}{u_s}.$$  

(2.2)

Though these two formulations look quite different, they are indeed equivalent when $r_{sa} > 0$ in the following sense.

- They share the same optimal value function $v^*$.
- The optimal dual variable $u^*$ differs only by an $s$-dependent scaling factor. This implies that the optimal policy $\pi^*_{sa} = u^*_{sa}/\tilde{u}^*_{sa}$ are the same.

One geometric way to see this equivalence is to go through the associated primal formulations

$$\min_v e^T v, \ s.t. \ \forall s, v_s \geq \tau \log \left( \sum_{a \in A} \exp \left( \frac{r_{sa} + \gamma \sum_{s'} P_{ass'} v_{s'}}{\tau} \right) \right),$$  

(2.3)

and

$$\min_v \frac{\alpha}{2} \|v\|^2, \ s.t. \ \forall s, v_s \geq \tau \log \left( \sum_{a \in A} \exp \left( \frac{r_{sa} + \gamma \sum_{s'} P_{ass'} v_{s'}}{\tau} \right) \right).$$  

(2.4)

Figure 1 illustrates the primal formulations of a randomly generated MDP with $|S| = |A| = 2$, where the yellow region represents the feasible set and the red dot represents the optimal value $v^*$. Due to the key assumption $r_{sa} \geq 0$, the feasible set lies in the first quadrant. From the contour plots of the objective function $e^T v$ and $\|v\|^2$ shown by the dotted curves, it is clear that both of them are minimized at $v^*$ when constrained to the feasible set.

The following theorem states this equivalence formally, with its proof given in section 6.

**Theorem 2.1.** For an infinite-horizon discounted MDP with finite state space $S$, finite action space $A$ and nonnegative reward $r$, we have the following properties:

(a) There is a unique solution $(v^*, u^*)$ to the primal-dual problem:

$$\min_v \max_u E_0(v, u) = \sum_s c_s v_s + \sum_{s,a} u_{sa} \left( r_{sa} - \sum_{s'} K_{ass'} v_{s'} \right) - \tau \sum_{s,a} u_{sa} \log \frac{u_{sa}}{u_s},$$

where $v^*$ is the optimal value function defined by (1.5) and $\frac{u^*_{sa}}{u_s}$ gives the optimal policy $\pi^*_{sa}$.

(b) There is a unique solution $(v^*, u^*)$ to the quadratically convexified problem:

$$\min_v \max_u E(v, u) = \frac{\alpha}{2} \sum_s v_s^2 + \sum_{s,a} u_{sa} \left( r_{sa} - \sum_{s'} K_{ass'} v_{s'} \right) - \tau \sum_{s,a} u_{sa} \log \frac{u_{sa}}{u_s},$$

where $v^*$ is the optimal value function, and $\frac{u^*_{sa}}{u_s}$ coincides with the optimal policy $\pi^*_{sa}$.

**Remark 2.2.** With the same method as the one used for the proof of Theorem 2.1, one can show that the conclusions of Theorem 2.1 still hold if the term $\frac{\alpha}{2} \sum_s v_s^2$ in the formulation (2.2) is replaced with a strictly increasing convex function of $v$. The intuition provided in Figure 1 also applies.
2.2. Natural gradient ascent descent. As mentioned earlier, the gradient-based methods for the primal-dual formulation (2.1) suffer from slow convergence, partly due to the linearity of $E_0(v, u)$ in $v$. Since the quadratically convexified scheme (2.2) gives the same value function $v^*$ and policy $\pi^*$ as the original primal-dual problem (2.1), we work instead with (2.2) and propose an NGAD algorithm.

The first-order derivatives of the new objective function $E(v, u)$ are

$$\frac{\partial E}{\partial v_{s'}} = \alpha v_{s'} - \sum_{s, a} K_{a,s'} u_{sa}, \quad s' \in S,$$

$$\frac{\partial E}{\partial u_{sa}} = \left( r_{sa} - \sum_{s'} K_{a,s'} v_{s'} \right) - \tau \log \frac{u_{sa}}{u_s}, \quad (s, a) \in S \times A.$$

(2.5)

The diagonal blocks of the second-order derivatives $\frac{\partial^2 E}{\partial v_{s'}^2}$ and $\frac{\partial^2 E}{\partial u_{sa}^2}$ are

$$\frac{\partial^2 E}{\partial v_{s'} \partial v_{s'}} = \alpha \delta_{ss'}, \quad (s, s') \in S \times S,$$

$$\frac{\partial^2 E}{\partial u_{sa} \partial u_{sa'}} = -\tau \delta_{ss'} \left( \frac{\delta_{aa'}}{u_{sa}} - \frac{1}{u_s} \right), \quad (s, s', a, a') \in S^2 \times A^2.$$

(2.6)

Of the two diagonal blocks above, $\frac{\partial^2 E}{\partial v_{s'}^2}$ is easy to invert since it is diagonal with positive diagonal entries, whereas $\frac{\partial^2 E}{\partial u_{sa}^2}$ is the sum of a diagonal part and a low-rank part. In the natural gradient dynamics below, we only keep the first part of $\frac{\partial^2 E}{\partial u_{sa}^2}$, namely $-\tau \delta_{ss'} \delta_{aa'}/u_{sa}$ (or more compactly $-\tau \text{diag}(1/u)$ in the matrix form). The resulting NGAD flow is:

$$\frac{dv_{s'}}{dt} = \frac{1}{\alpha} \left( \alpha v_{s'} - \sum_{s, a} K_{a,s'} u_{sa} \right), \quad s' \in S,$$

$$\frac{du_{sa}}{dt} = -\frac{1}{\tau} u_{sa} \left( \tau \log \frac{u_{sa}}{u_s} - \left( r_{sa} - \sum_{s'} K_{a,s'} v_{s'} \right) \right), \quad (s, a) \in S \times A.$$
The next lemma states that
\[ L(s') = -\left( v_s' - \frac{1}{\alpha} \sum_{sa} K_{ass'} u_{sa} \right), \quad s' \in S, \]
\[ \frac{du_{sa}}{dt} = -u_{sa} \left( \log \frac{u_{sa}}{d_s} - \frac{1}{\tau} \left( r_{sa} - \sum_{s'} K_{ass'} v_{s'} \right) \right), \quad (s, a) \in S \times A. \]
(2.7)

To analyze its convergence, we start by identifying a Lyapunov function of this dynamics. By Theorem 2.1 there is a unique solution \((v^*, u^*)\) to problem (2.2). Based on the solution \((v^*, u^*)\), define

\[ L(v, u) = \frac{\alpha}{2} \sum_{s \in S} |v_s - v_s^*|^2 + \tau \sum_{s \in S, a \in A} \left( u_{sa}^* \log \frac{u_{sa}^*}{u_{sa}} + u_{sa} - u_{sa}^* \right). \]
(2.8)

The following lemma summarizes some key properties of \(L(v, u)\).

**Lemma 2.3.** \(L(v, u) \geq 0\) is strictly convex, and the unique minimum is \((v^*, u^*)\), which satisfies \(L(v^*, u^*) = 0\). In addition, any sublevel set of \(L\) is bounded.

The next lemma states that \(L(v, u)\) is a Lyapunov function of (2.7).

**Lemma 2.4.** \(L(v, u)\) is a Lyapunov function for the dynamics (2.7), i.e., \(\frac{dL}{dt} \leq 0\) when \(\frac{dv}{dt}\) and \(\frac{du}{dt}\) are defined in (2.7), and the only trajectory of the dynamics (2.7) satisfying \(\frac{dL}{dt} = 0\) is \((v, u) = (v^*, u^*)\).

The proofs of these two lemmas are given in section 6.

**Theorem 2.5.** The dynamics of (2.7) converges globally to \((v^*, u^*)\).

**Proof.** By Lemma 2.3, Lemma 2.4 and the Barbashin-Krasovskii-LaSalle theorem [14], the dynamics of (2.7) is globally asymptotically stable, which means the NGAD dynamics converges globally to \((v^*, u^*)\). \(\square\)

To show the exponential convergence of (2.7), we follow Lyapunov’s indirect method, i.e., analyzing the linearization of (2.7) at \((v^*, u^*)\) and demonstrating that the real part of the eigenvalues of the corresponding matrix is negative. This result is the content of Theorem 2.6, with the proof given in section 6.

**Theorem 2.6.** The dynamics of (2.7) converges at rate \(O(e^{-ct})\) to \((v^*, u^*)\) for some \(c > 0\).

Below we discuss the implementation of (2.7). By introducing \(u_{sa} = \exp(\theta_{sa})\), (2.7) can be rewritten as

\[ \frac{dv_{s'}}{dt} = -\left( v_{s'} - \frac{1}{\alpha} \sum_{sa} K_{ass'} \exp(\theta_{sa}) \right), \quad s' \in S, \]
\[ \frac{d\theta_{sa}}{dt} = -\left( \theta_{sa} - \log \left( \sum_a \exp(\theta_{sa}) \right) - \frac{1}{\tau} \left( r_{sa} - \sum_{s'} K_{ass'} v_{s'} \right) \right), \quad (s, a) \in S \times A. \]
(2.9)

With a learning rate \(\eta > 0\), this leads to the update rule

\[ v_{s'} \leftarrow (1 - \eta) v_{s'} + \frac{\eta}{\alpha} \sum_{sa} K_{ass'} \exp(\theta_{sa}), \quad s' \in S, \]
\[ \theta_{sa} \leftarrow (1 - \eta) \theta_{sa} + \eta \log \left( \sum_a \exp(\theta_{sa}) \right) + \frac{\eta}{\tau} \left( r_{sa} - \sum_{s'} K_{ass'} v_{s'} \right), \quad (s, a) \in S \times A. \]
(2.10)

The details of the algorithm are summarized in Algorithm 2.1.
Algorithm 2.1 Standard NGAD for quadratically convexified formulation

**Require:** the MDP model $\mathcal{M} = (S, A, P, r, \gamma)$, initialization $(v_{\text{init}}, \theta_{\text{init}})$, convergence threshold $\epsilon_{\text{tol}}$, coefficient $\alpha > 0$ for the quadratic term in (2.2), regularization coefficient $\tau$, learning rate $\eta$.

1: Initialize the value and parameters $v = v_{\text{init}}$, $\theta = \theta_{\text{init}}$.
2: Calculate $u_{sa} = \exp(\theta_{sa})$, $(s, a) \in S \times A$.
3: Set $q = 1 + \epsilon_{\text{tol}}$.
4: while $q > \epsilon_{\text{tol}}$ do
5: Calculate $(v_{\text{new}})_{s'} = (1 - \eta) v_{s'} + \frac{\eta}{\alpha} \sum_s K_{ass'} u_{sa}$, $s' \in S$.
6: Update $\theta$ by
\[
\theta_{sa} \leftarrow (1 - \eta) \theta_{sa} + \eta \log \frac{1}{u_{sa}} + \eta \tau \left( r_{sa} - \sum_{s'} K_{ass'} (v_{\text{new}})_{s'} \right), \ (s, a) \in S \times A.
\]
7: Calculate $(u_{\text{new}})_{sa} = \exp(\theta_{sa})$, $(s, a) \in S \times A$.
8: Calculate $q = \max\{|v_{\text{new}} - v|/\|v\|, |u_{\text{new}} - u|/\|u\|\}$.
9: Update $(v, u)$ by $v \leftarrow v_{\text{new}}$, $u \leftarrow u_{\text{new}}$.
end while

3. Interpolating natural gradient method. In subsection 2.2, NGAD is introduced using the diagonal part of $\frac{\partial^2 E}{\partial u^2}$. A natural question is whether the whole matrix $\frac{\partial^2 E}{\partial u^2}$ can be used. Under the matrix notation, $\frac{\partial^2 E}{\partial u^2}$ in (2.6) takes the form
\[
(3.1) \quad \frac{\partial^2 E}{\partial u^2} = \begin{bmatrix} H_1 & \cdots & \cdot \\ \cdot & \ddots & \cdot \\ \cdot & \cdot & H_{|S|} \end{bmatrix}, \quad H_s = \text{diag}((u_s .)^{-1}) - \frac{1}{u_s} 1_{|A|} 1_{|A|}^T, \ s \in S.
\]

Since the Hessian matrix describes the local geometry of the problem, the standard NGAD in Section subsection 2.2 can be viewed as approximating the Hessian diagonally
\[
\frac{\partial^2 E}{\partial u^2} \approx \begin{bmatrix} \text{diag}((u_1 .)^{-1}) \\ \cdot \\ \cdot \\ \text{diag}((u_{|S|} .)^{-1}) \end{bmatrix}
\]
and using its inverse
\[
\begin{bmatrix} \text{diag}(u_1 .) \\ \cdot \\ \cdot \\ \text{diag}(u_{|S|} .) \end{bmatrix} \equiv \begin{bmatrix} \tilde{u}_1(\text{diag}(\pi_1 .)) \\ \cdot \\ \cdot \\ \tilde{u}_{|S|}(\text{diag}(\pi_{|S|} .)) \end{bmatrix}
\]
to precondition the gradient. However, $H_s$ is in fact singular with $\text{Null}(H_s) = \text{Span}(u_s .)$ and its pseudoinverse reads
\[
\begin{bmatrix} \tilde{u}_1(\text{diag}(\pi_1 .) - \pi_1 \pi_1^T) \\ \cdot \\ \cdot \\ \tilde{u}_{|S|}(\text{diag}(\pi_{|S|} .) - \pi_{|S|} \pi_{|S|}^T) \end{bmatrix}.
\]

If we had constructed the natural gradient method with this pseudoinverse, the component in the $1_{|A|}$ direction would not have been updated in the dynamics.

The key idea is that one can interpolate between these two extreme cases, i.e., we propose to
use
\[
\begin{bmatrix}
\hat{u}_1(\text{diag}(\pi_1) - c\pi_1^T) \\
\vdots \\
\hat{u}_{|S|}(\text{diag}(\pi_{|S|}) - c\pi_{|S|}^T)
\end{bmatrix}
\tag{3.2}
\]
for \(0 < c < 1\) to precondition the gradient.

Under this interpolating metric (3.2), the new interpolating NGAD (INGAD) is given by
\[
\frac{dv_{s'}}{dt} = -\left(v_{s'} - \frac{1}{\alpha} \sum_{sa} K_{ass'} u_{sa}\right), \quad s' \in S,
\tag{3.3}
\]
\[
\frac{d\theta}{dt} = -\bar{u}_{s} (\text{diag}(\pi_{s}) - c\pi_{s}^T) \left( \log \frac{u_{s}}{\bar{u}_s} - \frac{1}{\tau} \left(r_{s} - \sum_{s'} K_{ss'} v_{s'} \right) \right), \quad s \in S,
\]
where \(u_{sa} \in \mathbb{R}^{|A|}\). When \(c = 0\), this dynamics reduces to (2.7).

A Lyapunov function of this dynamics can also be identified. Using the unique solution \((v^*, u^*)\) to (2.2), we define
\[
L_c(v, u) = \frac{\alpha}{2} \sum_s |v_s - v_s^*|^2 + \tau \left( \sum_{sa} \left( u_{sa}^* \log \frac{u_{sa}^*}{u_{sa}} + u_{sa} - u_{sa}^* \right) \right) + \frac{c}{1-c} \sum_s \left( \bar{u}_s^* \log \frac{\bar{u}_s}{\bar{u}_s^*} + \bar{u}_s - \bar{u}_s^* \right)
\tag{3.4}
\]
where the subscript \(c\) denotes the hyperparameter in the function. Some key properties of \(L_c(v, u)\) are summarized in the following lemma.

**Lemma 3.1.** \(L_c(v, u)\) is convex and the unique minimum is \(L_c(v^*, u^*) = 0\). The sublevel sets of \(L_c\) are bounded.

The next lemma states that \(L_c(v, u)\) is a Lyapunov function for (3.3).

**Lemma 3.2.** \(L_c(v, u)\) is a Lyapunov function for the dynamics (3.3), i.e., \(\frac{dv_{s'}}{dt} \leq 0\) when \(\frac{dv}{dt}\) and \(\frac{d\theta}{dt}\) are defined by (3.3), and the only trajectory of the dynamics (3.3) satisfying \(\frac{dL_c}{dt} = 0\) is \((v, u) = (v^*, u^*)\).

The proofs of these two lemmas can be found again in section 6.

**Theorem 3.3.** The dynamics of (3.3) converges globally to \((v^*, u^*)\).

**Proof.** Similar to Theorem 2.5, by Lemma 3.1, Lemma 3.2 and the Barbashin-Krasovskii-LaSalle theorem [14], the dynamics of (3.3) is globally asymptotically stable and hence converges globally to \((v^*, u^*)\).

The local exponential convergence of (3.3) can also be shown with Lyapunov’s indirect method. This result is stated in Theorem 3.4.

**Theorem 3.4.** The dynamics of (3.3) converges at rate \(O(e^{-ct})\) to \((v^*, u^*)\) for some \(c > 0\).

Finally, we discuss the implementation of (2.7). By letting \(u_{sa} = \exp(\theta_{sa})\), (3.3) can be written as
\[
\frac{dv_{s'}}{dt} = -\left(v_{s'} - \frac{1}{\alpha} \sum_{sa} K_{ass'} \exp(\theta_{sa}) \right), \quad s' \in S,
\tag{3.5}
\]
\[
\frac{d\theta_{s}}{dt} = -\left( I - c \frac{1}{|T|} \exp(\theta_{s}) \right) \left( \theta_{s} - \log \sum_{a} \exp(\theta_{sa}) 1 - \frac{1}{\tau} \left(r_{s} - \sum_{s'} K_{ss'} v_{s'} \right) \right), \quad s \in S.
\]
With a learning rate $\eta > 0$, this becomes

$$v_{s'} \leftarrow (1 - \eta)v_{s'} + \frac{\eta}{\alpha} \sum_{sa} K_{ass'} \exp(\theta_{sa}), \quad s' \in S,$$

$$\theta_s \leftarrow \theta_s - \eta \left( I - \frac{c \mathbf{1} \exp(\theta_s)^\top}{\mathbf{1}^\top \exp(\theta_s)} \right) \left( \theta_s - \log \sum_a \exp(\theta_{sa}) \mathbf{1} - \frac{1}{\tau} \left( r_s - \sum_{s'} K_{ss'}v_{s'} \right) \right), \quad s \in S.$$  

The details of the algorithm can be found in Algorithm 3.1 below.

**Algorithm 3.1** INGAD for quadratically convexified formulation

**Require:** the MDP model $\mathcal{M} = (S, A, P, r, \gamma)$, initialization $(v_{\text{init}}, \theta_{\text{init}})$, convergence threshold $\epsilon_{\text{tol}}$, coefficient $\alpha > 0$ for the quadratic term in (2.2), regularization coefficient $\tau$, metric coefficient $0 < \epsilon < 1$, learning rate $\eta$.

1. Initialize the value and parameters $v = v_{\text{init}}, \theta = \theta_{\text{init}}$.
2. Calculate $u_{sa} = \exp(\theta_{sa}), \ (s, a) \in S \times A$.
3. Set $q = 1 + \epsilon_{\text{tol}}$.
4. **while** $q > \epsilon_{\text{tol}}$ **do**
   5. Calculate $(v_{\text{new}})_{s'} = (1 - \eta)v_{s'} + \frac{\eta}{\alpha} \sum_{sa} K_{ass'}u_{sa}, \quad s' \in S$.
   6. Update $\theta$ by
      $$\theta_s \leftarrow \theta_s - \eta \left( I - \frac{c \mathbf{1} \exp(\theta_s)^\top}{\mathbf{1}^\top \exp(\theta_s)} \right) \left( \theta_s - \log \sum_a u_{sa} \mathbf{1} - \frac{1}{\tau} \left( r_s - \sum_{s'} K_{ss'}(v_{\text{new}})_{s'} \right) \right), \quad s \in S.$$
   7. Calculate $(u_{\text{new}})_{sa} = \exp(\theta_{sa}), \ (s, a) \in S \times A$.
   8. Calculate $q = \max\{\|v_{\text{new}} - v\|/\|v\|, \|u_{\text{new}} - u\|/\|u\|\}$.
   9. Update $(v, u)$ by $v \leftarrow v_{\text{new}}, \ u \leftarrow u_{\text{new}}$.
10. **end while**

4. Numerical results. In this section, we examine the performance of Algorithm 2.1 and Algorithm 3.1 with several different examples. Subsection 4.1 compares Algorithm 2.1 and Algorithm 3.1 in a complete-information case where the transition probabilities and the rewards are known exactly. A comparison with an existing method in [42] is showcased in this setting as well. The sample-based setting is investigated in subsection 4.2, where we give an adapted version of INGAD with sample access, and test its performance on two different MDPs.

4.1. Experiments with complete information. Here we test the numerical performance of the standard natural gradient in Algorithm 2.1 and the interpolating natural gradient in Algorithm 3.1 in a complete information situation. The MDP used is from [42], where $|S| = 200$, $|A| = 50$, and the transition probabilities and rewards are randomly generated. More specifically, the transition probabilities are set as $P_{ass'} = 1/20$ for any $s' \in S_{sa}$, where $S_{sa}$ is a uniformly randomly chosen subset of $S$ such that $|S_{sa}| = 20$, and the reward $r_{sa} = U_{sa}U_s$ for $(s, a) \in S \times A$, where $U_{sa}$ and $U_s$ are independently uniformly sampled from $[0, 1]$.

A comparison of Algorithm 2.1 and Algorithm 3.1 is carried out using the same discount rate $\gamma = 0.99$ and hyperparameters $(\epsilon_{\text{tol}}, \alpha, \tau) = (1 \times 10^{-5}, 0.1, 0.01)$. Since both algorithms are explicit discretizations of the corresponding flow, a sufficiently small learning rate is needed to ensure convergence. In the tests, the learning rates are set as $\eta = 3 \times 10^{-4}$ for Algorithm 2.1 and $\eta = 8 \times 10^{-3}$ for Algorithm 3.1, which are both manually tuned to be close to the largest learning rates such that convergence is achieved. For Algorithm 3.1, we set $c = 0.98$.

As a result, Algorithm 2.1 takes 59296 iterations to converge while Algorithm 3.1 takes 2213 iterations, demonstrating that the interpolating metric introduced in section 3 gives rise to an acceleration of more than 1 magnitude. Plotted in Figure 2(a) and Figure 2(c) are the errors of the value and policy with respect to the ground truth in the training process, which verifies
ACCELERATING PRIMAL-DUAL FOR MDPS

that Algorithm 3.1 achieves the same precision more than a magnitude faster than Algorithm 2.1. Moreover, it can be observed from Figure 2(b) and Figure 2(d) that the Lyapunov function decreases monotonically in both cases, confirming the theoretical analyses in section 2 and section 3.

Comparison with PMD [42]. Next, we compare the performance of Algorithm 3.1 (INGAD) with an existing method, namely the policy mirror descent (PMD) method used in [42]. The underlying MDP of the problem is the same as in subsection 4.1. For the hyperparameters of INGAD, we take \((N_{	ext{iter}}, \alpha, c) = (2000, 0.1, 0.98)\). In order to make a fair comparison, the learning rate is set as \(\eta = 8 \times 10^{-3}\), and the regularization coefficient is set as \(\tau = 0.01\) for both methods. For the PMD method, we take the first 20000 iterations.

Fig. 3. Comparison of Algorithm 3.1 with PMD [42]. (a): Convergence of the value function; (b): Convergence of the policy. Blue curves: The convergence of \(\|\pi - \pi^*\|_F/\|\pi^*\|_F\) in the training process. Orange curves: The convergence of \(\|v - v^*\|_2/\|v^*\|_2\) in the training process. A logarithmic scale is used for all vertical axes.
It can be seen from Figure 3 that Algorithm 3.1 admits a faster convergence than PMD. For both the value function and the policy, Algorithm 3.1 achieves a higher precision in 2000 iterations than PMD with 20000 iterations. The final errors in the value function and policy are approximately (0.0034, 0.0025) for INGAD and (0.39, 0.0049) for PMD.

Algorithm 4.1 INGAD for quadratically convexified formulation (sample version)

Require: the discount rate $\gamma$, initialization $(v_{\text{init}}, \theta_{\text{init}})$, convergence threshold $\epsilon_{\text{tol}}$, maximum number of iterations $N_{\text{iter}}$, coefficient $\alpha > 0$ for the quadratic term in (2.2), regularization coefficient $\tau$, metric coefficient $0 \leq c < 1$, the initial and the final learning rate $(\eta_{\text{init}}, \eta_{\text{end}})$.

1: Initialize the value and parameters $v = v_{\text{init}}$, $\theta = \theta_{\text{init}}$.
2: Calculate $u_{sa} = \exp(\theta_{sa})$, $(s, a) \in S \times A$.
3: Set $q = 1 + \epsilon_{\text{tol}}$ and $i = 0$.
4: Initialize a buffer $\mathcal{B}$ with $N$ transition samples $(s, a, s', r)$.
5: while $q > \epsilon_{\text{tol}}$ and $i < N_{\text{iter}}$ do
6: Calculate $\eta_i = (1 + i(N_{\text{iter}}\eta_{\text{end}})^{-1}(\eta_{\text{init}} - \eta_{\text{end}}))^{-1}\eta_{\text{init}}$.
7: Randomly sample a batch of samples from $\mathcal{B}$ with size $N_b$.
8: Estimate $\hat{K}^{(i)}$ from the samples.
9: Calculate $(v_{\text{new}})_{sa} = (1 - \eta_i)u_{sa} + \frac{\epsilon}{\alpha} \sum_{s' \in S} \hat{K}^{(i)}_{ss'a} u_{sa}$, $s' \in S$.
10: Update $\theta$ by

$$
\theta_s \leftarrow \theta_s - \eta_i \left( I - \frac{c1_u^{T}1_{u_s}}{1_{u_s}^{T}1_{u_s}} \right) \left( \theta_s - \left( \log \sum_{a} u_{sa} \right) \right) \left( 1 - \frac{1}{\tau} \right) \left( r_s - \sum_{s'} \hat{K}^{(i)}_{ss'} (v_{\text{new}})_{s'} \right), \quad s \in S.
$$

11: Calculate $(u_{\text{new}})_{sa} = \exp((\theta_{\text{new}}))_{sa}$, $(s, a) \in S \times A$.
12: Calculate $q = \max\{||v_{\text{new}} - v||/||v||, ||u_{\text{new}} - u||/||u||\}$.
13: Update $(v, u)$ by $v \leftarrow v_{\text{new}}$, $u \leftarrow u_{\text{new}}$.
14: $i \leftarrow i + 1$.
15: end while

4.2. Experiments with random samples. Finally, we test the INGAD algorithm on the case where the transition probabilities are unknown. In each iteration, a size-$N_b$ batch of samples is used to estimate the transition probabilities and used for the INGAD update, as presented in Algorithm 4.1. In order to stabilize the training dynamics, we use a decaying learning rate starting with $\eta_{\text{init}}$ and ending with $\eta_{\text{end}}$. If $\eta_{\text{init}} = \eta_{\text{end}}$, then the algorithm reduces to the constant learning rate case. We first use the MDP introduced in subsection 4.1.

In this experiment, we adopt $(\gamma, N_{\text{iter}}, N_b, \alpha, \tau, c) = (0.9, 12000, 1 \times 10^5, 0.1, 0.1, 0.9)$ and $(\eta_{\text{init}}, \eta_{\text{end}}) = (0.001, 0.001)$. Altogether $1 \times 10^8$ samples are used in the training process. It can be seen from Figure 4 that the approximate value function and policy given by Algorithm 4.1 converge to the ground truth and oscillate around it at the final stage. The final errors in the value function and policy are approximately 0.015 and 0.030, respectively. It can also be seen from Figure 4(b) that the Lyapunov function mostly decreases in the training process even though the transition probabilities used are just unbiased estimators of the ground truth.

Experiment with the FrozenLake environment. In this part, the MDP we consider is from the FrozenLake environment (see [6]). The environment describes the problem where the player aims to walk on a frozen lake from one corner to another without falling into the holes. In the example we use below, the map is an 8 $\times$ 8 square grid with 10 randomly generated holes. Therefore, the size of the state space is 64, and there are 4 actions, corresponding to the 4 directions one can choose at each position. In order to model the low-friction property of ice, the transition is not deterministic. More specifically, the agent has a 1/3 probability of moving in the intended direction or the two perpendicular directions. An illustration of the lake map is given in Figure 5.

In the numerical experiment, we set $(\gamma, N_{\text{iter}}, N_b, \alpha, \tau, c) = (0.9, 80000, 2000, 0.1, 0.1, 0.9)$, and $(\eta_{\text{init}}, \eta_{\text{end}}) = (0.002, 0.0002)$. The buffer size $N$ and the batch-size $N_b$ are chosen as $2 \times 10^6$ and
ACCELERATING PRIMAL-DUAL FOR MDPS

(a) Error of the value function and policy. (b) The Lyapunov function.

Fig. 4. Performance of Algorithm 4.1 for the MDP problem described in Subsection 4.1. (a): Convergence of the value and policy during training of Algorithm 4.1; Blue curve: the convergence of $\|\pi - \pi^*\|_F/\|\pi^*\|_F$ in the training process; Orange curve: the convergence of $\|v - v^*\|_2/\|v^*\|_2$ in the training process. (b): Lyapunov function (3.4). A logarithmic scale is used for all vertical axes.

Fig. 5. Map of the FrozenLake environment with size $8 \times 8$ and 10 randomly generated holes. The green and the orange boxes represent the starting position and the target position, respectively. The blue area represents the positions with ice, while the grey spots indicate the positions of holes.

2000, respectively.

(a) Error of the value function and policy. (b) The Lyapunov function.

Fig. 6. Performance of Algorithm 4.1 for the $8 \times 8$ FrozenLake problem. (a): Convergence of the value and policy during training of Algorithm 4.1; Blue curve: the convergence of $\|\pi - \pi^*\|_F/\|\pi^*\|_F$ in the training process; Orange curve: the convergence of $\|v - v^*\|_2/\|v^*\|_2$ in the training process. (b): Lyapunov function (3.4). A logarithmic scale is used for all vertical axes.

Similar to the previous example, both the error of the value function and the error of the policy function reduce in the training process, indicating the effectiveness of Algorithm 4.1 given sample access to the MDP. The oscillations represent the randomness in the samples gathered in each batch. The final errors for the value and policy are 0.012 and 0.026, respectively. The Lyapunov function also shows a clear decreasing trend along the training process.

5. Conclusion and discussion. In this paper, we focused on the primal-dual formulation of entropy regularized Markov decision problems. We proposed a quadratically convexified primal-dual formulation that makes the landscape of the objective function smoother and enables faster numerical algorithms. We proved the equivalence of the quadratically convexified primal-dual
that the solution (v)
Combining (6.4) and (6.5), we conclude that (6.5)
Also from [41] we know that any solution to the primal-dual formulation (2.1) satisfies
As a result, (6.3)
π
and policy (1.5), or equivalently, (1.7) and (1.8) hold. From [41], we know that this optimal value function
K
14
H. LI, H. YU, L. YING, AND I. DHILLON
formulation with the original primal-dual formulation. Leveraging the enhanced convexity of the
objective function, we proposed an NGAD method and proved its convergence properties using
the Lyapunov methods. We further introduced an INGAD algorithm that accelerates convergence
significantly. The efficiency and robustness of the proposed algorithms are demonstrated through
multiple numerical experiments.
For future directions, one can potentially extend the convergence analysis here to the finite
sample case with standard statistical methods. Another interesting direction to explore is the
application of other optimization techniques to the convexified formulation proposed here.

6. Proofs.

6.1. Proof of Theorem 2.1.
Proof of Theorem 2.1. First, we show that there exists a unique solution to (2.1). By [12],
there exists a unique optimal policy π∗ and a unique optimal value function v∗ = vπ,∗ such that
(1.5), or equivalently, (1.7) and (1.8) hold. From [41], we know that this optimal value function
and policy (v∗, π∗) also yields a solution (v, u) to the primal-dual problem by u∗ sa = π∗ sa(K−T π e)s.
Also from [41] we know that any solution to the primal-dual formulation (2.1) satisfies v = v∗, u∗ sa/˜u = π∗ sa(s, a) ∈ S × A and ˜u = K−T π e, which combined with the uniqueness of (v∗, π∗) shows
that the solution (v∗, u∗) to (2.1) is unique.
Next, we show that (v∗, u∗) satisfies the first-order condition of (2.2), where
(6.1)
u∗ sa := w♯ sa u∗ sa , u := αK−T π v∗.
The first-order condition of (2.1) gives
e∗ sa′ − s K′ as a u∗ sa = 0, ∀sa ∈ S,
(6.2)
r∗ sa = s K′ a s a v∗ sa − τ log(u∗ sa/˜u∗ sa) = 0, ∀(s, a) ∈ S × A.
Since e∗ sa = s K′ a s a u∗ sa = s a K′ a s a π∗ sa s a K−1 π e = ∑∞ k=1 γ k (Pπ)ke, and thus ˜u∗ sa ≥ e∗ σ > 0 for all s ∈ S. Similarly, w∗ sa ≥ απ∗ sa for all s ∈ S since
w = αK−T π v∗. By (1.8), it is also known that π∗ sa > 0 for all (s, a) ∈ S × A, so (r∗ sa − τhπ,∗)s > 0
for all s ∈ S since r is nonnegative. Again by an expansion of K−1 π , one can show that v∗ sa = K−1 π (r∗ sa − τhπ,∗) ≥ (r∗ sa − τhπ,∗)s > 0. Hence u∗ sa = w♯ sa u∗ sa > 0 is well-defined. In addition,
˜u∗ = w and
(6.3)
u∗ sa u∗ sa = w♯ sa u∗ sa = π∗ sa.
As a result,
r∗ sa − s K′ a s a v∗ sa = τ log(u∗ sa/˜u∗ sa) = τ log u∗ sa u∗ sa , ∀(s, a) ∈ S × A.
(6.4)
Moreover, one can show that
(6.5)
απ∗ sa = K−1 π w = s Kπ,∗ s a u∗ sa π∗ sa = s (s K′ a s a π∗ sa )u∗ sa π∗ sa = s K′ a s a u∗ sa , ∀sa ∈ S.
Combining (6.4) and (6.5), we conclude that (v∗, u∗) is a solution to
(6.6)
απ∗ sa − s K′ a s a u∗ sa = 0, ∀sa ∈ S,
r∗ sa − s K′ a s a v∗ sa − τ log u∗ sa u∗ sa = 0, ∀(s, a) ∈ S × A.
This is the first-order stationary condition for the problem (2.2).

Finally, we show that $(v^*, u^*)$ is the unique solution to (6.6). Assume that $(v^1, u^1)$ and $(v^2, u^2)$ are both solutions to (6.6). If $v^1 \neq v^2$, then $E(v^1, u^1) < E(v^2, u^1)$ and $E(v^2, u^2) < E(v^1, u^2)$ since for any $v$, $E(v, u)$ is strictly convex in $v$. On the other hand, for any $v$, $E(v, u)$ is concave in $u$ (see for example [27] or [41]). So $E(v^1, u^1) \geq E(v^1, u^2)$ and $E(v^2, u^2) \geq E(v^2, u^1)$ and

$$E(v^1, u^1) \geq E(v^1, u^2) > E(v^2, u^2) \geq E(v^2, u^1) > E(v^1, u^1),$$

which is a contradiction, so we must have $v^1 = v^2$ instead. By the second equation in (6.6),

$$\frac{u^1_{sa}}{u^1_s} = \exp \left( \tau^{-1}(r_{sa} - \sum_{s'} K_{ass'} v^1_{s'}) \right) = \exp \left( \tau^{-1} \left( r_{sa} - \sum_{s'} K_{ass'} v^2_{s'} \right) \right) = \frac{u^2_{sa}}{u^2_s},$$

thus $\pi^1 = \pi^2$, where $\pi^1_{sa} = \frac{u^1_{sa}}{u^1_s}$, $\pi^2_{sa} = \frac{u^2_{sa}}{u^2_s}$. Since $(\pi^1, v^1) = (\pi^2, v^2)$, by the first equation in (6.6),

$$\tilde{u}^1 = \alpha K_{\pi^1}^{-1} v^1 = \alpha K_{\pi^2}^{-1} v^2 = \tilde{u}^2,$$

As a result,

$$u^1_{sa} = \tilde{u}^1_s \frac{u^1_{sa}}{u^1_s} = \tilde{u}^2_s \frac{u^2_{sa}}{u^2_s} = u^2_{sa}, \quad \forall (s, a) \in S \times A,$$

and $(v^1, u^1) = (v^2, u^2)$. Hence the solution to (6.6) is unique. Therefore, $(v^*, u^*)$ is the unique solution to (6.6). By equation (6.3), the policy yielded by $u^*$ coincides with the optimal policy $\pi^*$, which finishes the proof.

6.2. Proof of Lemma 2.3.

Proof of Lemma 2.3. From the definition of $L$ we know that $\frac{\partial^2 L}{\partial u_{sa} \partial v_{s'}} = 0$. Moreover,

$$\frac{\partial^2 L}{\partial u_s \partial v_s} = \alpha \delta_{ss'}, \quad \frac{\partial^2 L}{\partial u_{sa} \partial u_{s'a'}} = \tau \delta_{(s, a), (s', a')} \frac{u^*_{sa}}{u^2_{sa}}, \quad (s, s', a, a') \in S^2 \times A^2,$$

which means that the Hessian matrix of $L$ is a diagonal matrix with positive diagonal elements on the domain $\mathbb{R}^{\left| S \right|} \times \mathbb{R}^{\left| S \right| \times \left| A \right|}$. Hence $L$ is strictly convex. Since the first-order condition:

$$\frac{\partial L}{\partial v_s} = \alpha (v_s - v^*_s) = 0, \quad \frac{\partial L}{\partial u_{sa}} = \tau \left( 1 - \frac{u^*_{sa}}{u_{sa}} \right) = 0, \quad (s, a) \in S \times A,$$

has a unique solution $(v, u) = (v^*, u^*)$, it is also the unique global minimum of $L$. Let

$$\varphi_s(x) = \frac{1}{2} \alpha |x - v^*_s|^2, \quad \psi_{sa}(x) = \tau \left( u^*_{sa} \log \frac{u^*_{sa}}{u_{sa}} / x + x - u^*_{sa} \right).$$

By the calculation above, one can also show that $\varphi_s$ and $\psi_{sa}$ are strictly convex and non-negative. Moreover, since $\lim_{x \to +\infty} \psi_{sa}(x) = +\infty$, we have $M(C) = \max \sup_{sa} \{ x > 0 \mid \psi_{sa}(x) \leq C \} < +\infty$. As a result, the sublevel set

$$\{ (v, u) \mid L(v, u) \leq C \} \subset \{ (v, u) \mid |v_s - v^*_s| \leq \sqrt{2C/\alpha}, 0 < u_{sa} < M(C), s' \in S, (s, a) \in S \times A \}$$

is bounded.

6.3. Proof of Lemma 2.4. We first prove the following lemma.

Lemma 6.1. Define $H : \mathbb{R}^{|A|} \to \mathbb{R}$ by $H(z) = \sum_{a} z_a \log z_a - \bar{z} \log \bar{z}$, where $\bar{z} = \sum_{a} z_a$, then $H$ is convex. Moreover, $(z_2 - z_1) \cdot (\nabla H(z_1) - \nabla H(z_2)) \geq 0$, and the equality is achieved if and only if $z_2 = cz_1$ for some $c > 0$. 

This manuscript is for review purposes only.
Proof. The second-order derivatives of $H$ read $\frac{\partial^2 H}{\partial z_a \partial z_a'} = \frac{\delta a a'}{z_a} - \frac{1}{2}$. By the Cauchy-Schwarz inequality, for any $x \in \mathbb{R}^{[A]}$

$$\sum_{aa'} x_a \frac{\partial^2 H}{\partial z_a \partial z_a'} x_{a'} = \sum_{aa'} x_a \left( \frac{\delta a a'}{z_a} - \frac{1}{2} \right) x_{a'} = \sum_{a} \frac{x_a^2}{z_a} - \frac{1}{2} \left( \sum_{a} x_a \right)^2 \geq 0.$$ 

Hence the Hessian matrix of $H$ is positive semi-definite and $H$ is convex. By convexity $(z_1 - z_2) \cdot (\nabla H(z_1) - \nabla H(z_2)) \geq 0$. Suppose now that equality holds. If $z_1 = z_2$, then clearly $z_2 = c z_1$ for $c = 1$. If $z_1 \neq z_2$, let $h(t) = H(z_1 + t(z_2 - z_1))$, then $h$ is also convex and $h'(0) = (z_2 - z_1) \cdot \nabla H(z_1) = (z_2 - z_1) \cdot \nabla H(z_2) = h'(1)$, so $h'(t) = h'(0)$ for any $t \in [0, 1]$, thus

$$0 = h''(0) = (z_2 - z_1)^T \nabla^2 H(z_1)(z_2 - z_1).$$

Hence from the equality condition of the Cauchy-Schwarz inequality, we conclude $z_2 - z_1 = c z_1$ and thus $z_2 = c z_1$ for some $c$, and we have $c > 0$ since $z_1, z_2 \in \mathbb{R}^{[A]}$.

Proof of Lemma 2.4. By Theorem 2.1, $(v^*, u^*)$ is also the unique solution to (6.6), so

$$\alpha v_{s'}^* - \sum_{sa} K_{ass'} u_{sa}^* = 0, \quad s' \in S,$$

$$\left( r_{sa} - \sum_{s'} K_{ass'} v_{s'}^* \right) - \tau \log \frac{u_{sa}^*}{u_s} = 0, \quad (s, a) \in S \times A.$$ 

Subtracting this from the dynamics (2.7) leads to

$$\frac{dv_{s'}}{dt} = - \left( v_{s'} - v_{s'}^* \right) - \frac{1}{\alpha} \sum_{sa} K_{ass'}(u_{sa} - u_{sa}^*), \quad s' \in S,$$

$$\frac{du_{sa}}{dt} = -u_{sa} \left( \log \frac{u_{sa}}{u_s} - \log \frac{u_{sa}^*}{u_s} \right) + \frac{1}{\tau} \sum_{s'} K_{ass'}(v_{s'} - v_{s'}^*), \quad (s, a) \in S \times A.$$ 

Taking the derivative of $L$ gives

$$\frac{dL}{dt} = - \alpha \sum_{s'} (v_{s'} - v_{s'}^*) \left( v_{s'} - v_{s'}^* \right) - \frac{1}{\alpha} \sum_{sa} K_{ass'}(u_{sa} - u_{sa}^*)$$

$$- \tau \sum_{sa} \frac{u_{sa} - u_{sa}^*}{u_{sa}} u_{sa} \left( \log \frac{u_{sa}}{u_s} - \log \frac{u_{sa}^*}{u_s} \right) + \frac{1}{\tau} \sum_{s'} K_{ass'}(v_{s'} - v_{s'}^*)$$

$$= - \alpha \sum_{s'} (v_{s'} - v_{s'}^*)^2 - \tau \sum_{sa} (u_{sa} - u_{sa}^*) \left( \log \frac{u_{sa}}{u_s} - \log \frac{u_{sa}^*}{u_s} \right),$$

where we have used (6.7). By Lemma 6.1, 

$$\sum_{sa} (u_{sa} - u_{sa}^*)(\log u_{sa}/u_s - \log u_{sa}^*/u_s^*) = \sum_{s} (u_s - u_s^*) \cdot (\nabla H(u_s) - \nabla H(u_s^*)) \geq 0,$$

where $H$ is defined in Lemma 6.1. Therefore,

$$- \alpha \sum_{s'} (v_{s'} - v_{s'}^*)^2 - \tau \sum_{sa} (u_{sa} - u_{sa}^*) \left( \log \frac{u_{sa}}{u_s} - \log \frac{u_{sa}^*}{u_s^*} \right) \leq 0.$$ 

By Lemma 6.1 the equality holds only when $v = v^*$ and $u_{sa} = c_s u_{sa}^*$ for $c_s > 0$, $(s, a) \in S \times A$. Let

$$\mathcal{R} = \{(v, u) | - \alpha \sum_{s'} (v_{s'} - v_{s'}^*)^2 - \tau \sum_{sa} (u_{sa} - u_{sa}^*) (\log u_{sa}/u_s - \log u_{sa}^*/u_s^*) = 0\},$$
then $\mathcal{R} = \{(v, u) \mid v = v^*, u_{sa} = c_s u^*_{sa}, c_s \in \mathbb{R}_+, s \in S\}$. We proceed to prove that the only trajectory of (6.9) in $\mathcal{R}$ is $(v, u) = (v^*, u^*)$. Since $v = v^*$ for any $(v, u) \in \mathcal{R}$, $\frac{dv}{dt} = 0$ for any $s' \in S$. The following equality

$$0 = \sum_{sa} K_{ass'}(u_{sa} - u^*_{sa}) = \sum_{sa} K_{ass'}(c_s - 1)u^*_{sa}$$

(6.11)

$$= \sum_{sa} K_{ass'}(c_s - 1)\tilde{u}^*_s u^*_{sa} = \sum_s K_{ss'}(c_s - 1)\tilde{u}^*_s$$

means that, for any point $(v, u)$ on the trajectory of (6.9) in $\mathcal{R}$, $K^T_{ss'}((c-1)\tilde{u}^*) = 0$. Here $(c-1)\tilde{u}^*$ is the vector with length $|S|$ whose $s$-th element is $(c_s - 1)\tilde{u}^*_s$. Thus $c_s = 1$ for any $s \in S$, and the trajectory is a single point $(v, u) = (v^*, u^*)$.

6.4. Proof of Theorem 2.6.

Proof of Theorem 2.6. The linearized dynamics of the standard natural gradient (2.7) is

$$\frac{dv}{dt} = - \left( (v_{s'} - v^*_{s'}) - \frac{1}{\alpha} \sum_{sa} K_{ass'}(u_{sa} - u^*_{sa}) \right), \quad s' \in S,$$

(6.12)

$$\frac{du_{sa}}{dt} = -u^*_{sa} \left( \frac{1}{\tau} \sum_{s'} K_{ass'}(v_{s'} - v^*_{s'}) + \frac{u_{sa} - u^*_{sa}}{u^*_{sa}} - \frac{\tilde{u}_s - \tilde{u}^*_s}{\tilde{u}^*_s} \right), \quad (s, a) \in S \times A.$$

Define matrix $\tilde{K}$ by $\tilde{K}_{(s-1)|A|+a, s'} = K_{ass'}$, and let $\delta v = v - v^*$, $\delta u = u - u^*$. Then (6.12) becomes

$$\frac{d}{dt} \begin{bmatrix} \delta v \\ \delta u \end{bmatrix} = - \begin{bmatrix} I_{|S|} & -\frac{1}{\alpha} \tilde{K}^T \\ \frac{1}{\tau} \text{diag}(u^*) \tilde{K} & \text{diag}(u^*) M \end{bmatrix} \begin{bmatrix} \delta v \\ \delta u \end{bmatrix},$$

(6.13)

where $\text{diag}(u^*)$ is a diagonal matrix whose $((s-1)|A|+a)$-th diagonal element is $u^*_{sa}$. Here $M$ is a block-diagonal matrix defined as:

$$M = \begin{bmatrix} M_1 \\ \vdots \\ M_{|S|} \end{bmatrix}, \quad M_s = \text{diag}((u^*_s)^{-1}) - \frac{1}{u^*_s} I_{|A|} I^T_{|A|}, \quad s \in S,$$

(6.14)

where $\text{diag}((u^*_s)^{-1})$ is a diagonal $|A| \times |A|$ matrix with the $a$-th diagonal element equal to $1/u^*_{sa}$. Notice that $M$ is symmetric and by the Cauchy-Schwarz inequality, for any $x \in \mathbb{R}^{|A|}$

$$\sum_{aa'} x_a M_{sa} x_{a'} = \sum_a x_a^2 \frac{1}{u^*_s} \left( \sum_a x_a \right)^2 \geq 0, \quad s \in S.$$

Hence $M_s$ is positive semi-definite for all $s$ and thus $M$ is also positive semi-definite. Define invertible matrix $P$ as

$$P = \begin{bmatrix} \sqrt{\tau} I_{|S|} \\ \sqrt{\alpha} \text{diag}(\sqrt{u^*}) \end{bmatrix},$$

where $\text{diag}(\sqrt{u^*})$ is a diagonal $|S||A| \times |S||A|$ matrix with the $((s-1)|A|+a)$-th diagonal element equal to $\sqrt{u^*_{sa}}$. Denote the matrix in the linearized dynamics (6.13) by $-J$, i.e.,

$$J = \begin{bmatrix} I_{|S|} & -\frac{1}{\alpha} \tilde{K}^T \\ \frac{1}{\tau} \text{diag}(u^*) \tilde{K} & \text{diag}(u^*) M \end{bmatrix}.$$

Then

$$P^{-1}JP = \begin{bmatrix} \frac{1}{\sqrt{\tau}} I_{|S|} \\ \frac{1}{\sqrt{\alpha}} \text{diag}(\sqrt{u^*})^{-1} \end{bmatrix} \begin{bmatrix} I_{|S|} & -\frac{1}{\alpha} \tilde{K}^T \\ \frac{1}{\tau} \text{diag}(u^*) \tilde{K} & \text{diag}(u^*) M \end{bmatrix} \begin{bmatrix} \sqrt{\tau} I_{|S|} \\ \sqrt{\alpha} \text{diag}(\sqrt{u^*}) \end{bmatrix}.$$
It suffices to show that the real part of the eigenvalues of $P^{-1}JP$ is positive. Denote $P^{-1}JP$ by $\tilde{J}$. Using the positive semi-definiteness of $M$, for any eigenpair $(\lambda, x)$ of $\tilde{J}$ we can deduce

$$\Re(\lambda) = \frac{1}{2} \left( \frac{x^H \tilde{J} x}{x^H x} + \frac{x^H \tilde{J} x}{x^H x} \right) = \frac{1}{2} x^H x \left[ \begin{bmatrix} I_{|S|} \\ \sqrt{1/\alpha^T} \text{diag}(\sqrt{u^T}) \tilde{K} \text{diag}(\sqrt{u^T}) M \text{diag}(\sqrt{u^T}) \end{bmatrix} + \begin{bmatrix} -1/\sqrt{\alpha^T} \tilde{K} \text{diag}(\sqrt{u^T}) M \text{diag}(\sqrt{u^T}) \end{bmatrix} \right] x$$

(6.16)

where the superscript $H$ denotes the Hermitian transpose. Now we proceed to show $\Re(\lambda) \neq 0$.

Let $x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$, where $x_1 \in \mathbb{R}^{|S|}$, $x_2 \in \mathbb{R}^{|S|}$. If $\Re(\lambda) = 0$, then

$$0 = x^H \begin{bmatrix} I_{|S|} \\ \sqrt{1/\alpha^T} \text{diag}(\sqrt{u^T}) \tilde{K} \text{diag}(\sqrt{u^T}) M \text{diag}(\sqrt{u^T}) \end{bmatrix} x = x_1^T x_1 + (\text{diag}(\sqrt{u^T}) x_2)^H M (\text{diag}(\sqrt{u^T}) x_2) \geq 0,$n

thus $x_1 = 0$ and the equality condition of the Cauchy-Schwarz inequality (6.15) must hold. Hence $(x_2)_{sa} = c_s \sqrt{u_{sa}}$ for some $c_s \in \mathbb{R}$. We also know that $c_s$ is not all zero for $s \in S$; otherwise, $x_2 = 0$ and $x = 0$ is not an eigenvector. Thus

$$\tilde{J} x = \begin{bmatrix} I_{|S|} \\ \sqrt{1/\alpha^T} \text{diag}(\sqrt{u^T}) \tilde{K} \text{diag}(\sqrt{u^T}) M \text{diag}(\sqrt{u^T}) \end{bmatrix} x = x_1^T x_1 + (\text{diag}(\sqrt{u^T}) x_2)^H M (\text{diag}(\sqrt{u^T}) x_2) = \begin{bmatrix} 0 \\ \tilde{K} \text{diag}(\sqrt{u^T}) x_2 \end{bmatrix},$$

which is not a scalar multiple of $x$ unless $\tilde{K} \text{diag}(\sqrt{u^T}) x_2 = 0$. However, as

$$\tilde{K} \text{diag}(\sqrt{u^T}) x_2 = K_{ss'} c_u x_2$$

$\tilde{K} \text{diag}(\sqrt{u^T}) x_2 = K_{ss'}^T c_u x_2$ where $c_u x_2$ denotes the elementwise product. Thus $K_{ss'}^T c_u x_2 = 0$ and then $c_u x_2 = 0$, contradicting with the fact that $c_s$ is not all zero. The contradiction means that $\Re(\lambda) \neq 0$. Together with the inequality (6.16) we have $\Re(\lambda) > 0$ for any eigenvalue $\lambda$ of $J$. Hence $\Re(\lambda) < 0$ for any eigenvalue $\lambda$ of $-J$, the matrix in the linearized dynamics (6.13). By Lyapunov's indirect theorem [14], (2.7) has locally exponential convergence. □

6.5. Proof of Lemma 3.1.

Proof of Lemma 3.1. Similar to Lemma 2.3, we first note that $\frac{\partial^2 L_c}{\partial u_{sa} \partial v_{s'}} = 0$. Moreover,

$$\frac{\partial^2 L_c}{\partial v_{s} \partial v_{s'}} = \alpha \delta_{ss'}, \quad \frac{\partial^2 L_c}{\partial u_{sa} \partial u_{s'}} = \tau \delta_{ss'} \left( \delta_{aa} \frac{u_{sa}^* u_{sa}^*}{u_{sa}^2} + \frac{c_u^*}{(1-c)u_{sa}^2} \right), \quad (s, s', a, a') \in S^2 \times A^2.$$n

Hence the Hessian matrix of $L_c$ is

$$\begin{bmatrix} \alpha I_{|S|} & 0 \\ 0 & \text{diag}(u^*/u^2) + \frac{c^*}{u^2} B \end{bmatrix},$$

where $(u^*/u^2)_{sa} = u_{sa}^*/u_{sa}^2$ and $B$ is a positive definite block-diagonal matrix:

(6.17)

$$B := \begin{bmatrix} \frac{\sqrt{u}}{u_{|S|}^T} 1_{|A|} 1_{|A|}^T \\ \vdots \\ \frac{\sqrt{u}}{u_{|S|}^T} 1_{|A|} 1_{|A|}^T \end{bmatrix}.$$
Thus the Hessian of $L_c$ is positive definite and $L_c$ is strictly convex. The derivatives of $L_c$ are

$$\frac{\partial L_c}{\partial v_s} = \alpha(v_s - v_s^*), \quad s \in S,$$

$$\frac{\partial L_c}{\partial u_{sa}} = \tau \left( \frac{u_{sa} - u_{sa}^*}{u_{sa}} + \frac{c}{1 - c} \frac{\tilde{u}_s - \tilde{u}_s^*}{\tilde{u}_s} \right), \quad (s, a) \in S \times A,$$

(6.18)

from which we can see that $(v^*, u^*)$ is a solution to the first-order condition $\frac{\partial L_c}{\partial v} = 0$, $\frac{\partial L_c}{\partial u} = 0$. Since $L_c$ is strictly convex, it is also the unique minimizer of $L_c$. Now we prove that $L_c$ has bounded sublevel sets. Let $\ell(u) = \sum_s (\tilde{u}_s^* \log \tilde{u}_s^*/\tilde{u}_s + \tilde{u}_s - \tilde{u}_s^*)$. Then $L_c(v, u) = L_0(v, u) + \frac{\tau}{c} \ell(u)$. Since $\frac{\partial \ell}{\partial u_s} = B$ is positive definite, $\ell$ is strictly convex. Moreover, $\frac{\partial \ell}{\partial u_{sa}} = (\tilde{u}_s - \tilde{u}_s^*)/\tilde{u}_s$ equals to 0 when $u = u^*$, so by the strict convexity of $\ell$, $u^*$ is the unique minimizer of $\ell$, and thus $\ell(u) \geq \ell(u^*) = 0$. Hence the sublevel set $\{(v, u) \mid L_c(v, u) \leq C\} \subset \{(v, u) \mid L_0(v, u) \leq C\}$. Since the latter is bounded according to Lemma 2.3, the sublevel set of $L_c$ is also bounded.

6.6. Proof of Lemma 3.2.

Proof of Lemma 3.2. Plugging the first-order condition (6.8) for the exact solution $(v^*, u^*)$ into the interpolating natural gradient (3.3) results in

$$\frac{dv_{s'}}{dt} = - \left( v_{s'} - v_{s'}^* \right) - \frac{1}{\alpha} \sum_s K_{sss'}(u_{sa} - u_{sa}^*), \quad s' \in S,$$

$$\frac{du_s}{dt} = -\tilde{u}_s \left( \text{diag}(\pi_s) - c\pi_s, \pi_s^T \right) \left( \log \frac{u_s}{u_s^*} - \log \frac{u_s^*}{\tilde{u}_s} \right) + \frac{c}{1 - c} \sum_{s'} K_{ss's'}(v_{s'} - v_{s'}^*), \quad s \in S,$$

(6.20)

where $\pi_{sa}$ is defined as $u_{sa}/\tilde{u}_s$. A direct calculation shows that

$$\frac{dL}{dt} = -\alpha \sum_s (v_{s'} - v_{s'}^*) \left( v_{s'} - v_{s'}^* \right) - \frac{1}{\alpha} \sum_s K_{sss'}(u_{sa} - u_{sa}^*)$$

$$- \tau \sum_s \left[ \left( \frac{u_s - u_s^*}{\tilde{u}_s} \right) \left( \text{diag}(1/\pi_s) + \frac{c}{1 - c} 1_{|A|} 1_{|A|}^T \right) \tilde{u}_s \left( \text{diag}(\pi_s) - c\pi_s, \pi_s^T \right) \right]$$

$$\left( \log \frac{u_s}{u_s^*} - \log \frac{u_s^*}{\tilde{u}_s} \right) + \frac{1}{\tau} \sum_{s'} K_{ss's'}(v_{s'} - v_{s'}^*)$$

$$= -\alpha \sum_s (v_{s'} - v_{s'}^*)^2 - \tau \sum_s (u_{sa} - u_{sa}^*) \left( \log \frac{u_{as}}{u_s} - \log \frac{u_{as}^*}{u_s^*} \right),$$

(6.21)

where we have used the fact that

$$\left( \text{diag}(1/\pi_s) + \frac{c}{1 - c} 1_{|A|} 1_{|A|}^T \right) \left( \text{diag}(\pi_s) - c\pi_s, \pi_s^T \right)$$

$$= \text{diag}(1/\pi_s) \text{diag}(\pi_s) + \frac{c}{1 - c} 1_{|A|} 1_{|A|}^T \text{diag}(\pi_s) - \frac{c^2}{1 - c} 1_{|A|} 1_{|A|}^T \pi_s, \pi_s^T - c \text{diag}(1/\pi_s) \pi_s, \pi_s^T$$

$$= I + \left( \frac{c}{1 - c} - \frac{c^2}{1 - c} - c \right) 1_{|A|}\pi_s^T = I$$

Therefore,

$$\frac{dL_c}{dt} = -\alpha \sum_{s'} (v_{s'} - v_{s'}^*)^2 - \tau \sum_{sa} (u_{sa} - u_{sa}^*) \left( \log \frac{u_{as}}{u_s} - \log \frac{u_{as}^*}{u_s^*} \right),$$
where the right-hand side coincides with that of (6.10). Hence \( \frac{dv_s}{dt} = \frac{du_s}{dt} \leq 0 \) by the proof of Lemma 2.4. Let
\[
\mathcal{R} = \{(v, u) \mid -\alpha \sum_{s'} (v_{s'} - v_{s}^*)^2 - \tau \sum_{s} (u_{sa} - u_{sa}^*) (\log u_{sa}/\tilde{u}_s - \log u_{sa}^*/\tilde{u}_s) = 0\}.
\]
Then by the proof of Lemma 2.4, \( \mathcal{R} = \{(v, u) \mid v = v^*, u_{sa} = c_s u_{sa}^*, c_s \in \mathbb{R}_+, s \in S\} \). We proceed to prove that the only trajectory of (6.19) in \( \mathcal{R} \) is \((v, u) = (v^*, u^*)\). Since \( v = v^* \) for any \((v, u) \in \mathcal{R}\), \( \frac{dv_s'}{dt} = 0 \) for \( s' \in S \). In addition, for any \( s' \in S \) we have
\[
0 = \sum_{sa} K_{ass'}(u_{sa} - u_{sa}^*) = \sum_{s} K_{s'ss'}(c_s - 1)\tilde{u}_s^*,
\]
by the same calculation as (6.11). This means that for point \((v, u)\) on the trajectory of (6.19) in \( \mathcal{R} \), \( K_{ss'}((c - 1)\tilde{u}_s^*) = 0 \), thus \( (c - 1)\tilde{u}_s^* = 0 \) and \( c_s = 1 \) for any \( s \in S \). Since this is true for any \((v, u)\) on the trajectory, the trajectory is a single point \((v, u) = (v^*, u^*)\).

### 6.7. Proof of Theorem 3.4.

**Proof of Theorem 3.4.** The linearized dynamic of the interpolating natural gradient (3.3) is

\[
\frac{dv_{s'}}{dt} = -\left((v_{s'} - v_{s'}^*) - \frac{1}{\alpha} \sum_{sa} K_{ass'}(u_{sa} - u_{sa}^*)\right), \quad s' \in S,
\]
\[
\frac{du_s}{dt} = -\left(\text{diag}(u_s^*) - \frac{c}{\tilde{u}_s} u_s^* (u_s^*)^T\right) \left(\frac{1}{\tau} \sum_{s'} K_{ss'}(v_{s'} - v_{s'}^*) + \frac{u_s - u_s^*}{u_s^*} - \frac{\tilde{u}_s - \tilde{u}_s^*}{\tilde{u}_s^*} \mathbf{1}\right), \quad s \in S.
\]

Define \( \tilde{K} \) by \( \tilde{K}_{(s-1)||A||, s'} = K_{ass'} \) and let \( \delta v = v - v^* \), \( \delta u = u - u^* \). Then (6.22) becomes

\[
\frac{d}{dt} \begin{bmatrix} \delta v \\ \delta u \end{bmatrix} = -\begin{bmatrix} I_{|S|} \\ \frac{1}{\tau} G \tilde{K} \end{bmatrix} \begin{bmatrix} \delta v \\ \delta u \end{bmatrix},
\]
where \( M \) is a block-diagonal matrix defined as in (6.14) and \( G \) is a block-diagonal matrix

\[
G = \begin{bmatrix} G_1 \\ \vdots \\ G_{|S|} \end{bmatrix},
\]
with \( G_s = \text{diag}(u_s^*) - \frac{c}{\tilde{u}_s} u_s^* (u_s^*)^T \). Notice that \( G_s \) is symmetric. By the Cauchy-Schwarz inequality
\[
x^T G_s x = \sum_a u_{sa}^2 x_a^2 - \frac{c}{\tilde{u}_s} \left(\sum_a u_{sa}^2 x_a^2\right)^2 \geq \frac{1}{\tilde{u}_s^2} \left(\sum_a u_{sa}^2 x_a^2\right)^2 - \frac{c}{\tilde{u}_s^2} \left(\sum_a u_{sa}^2 x_a^2\right)^2
\]
\[
= \frac{1 - c}{\tilde{u}_s^2} \left(\sum_a u_{sa}^2 x_a^2\right) > 0, \quad \forall x \in \mathbb{R}^{|A|}, x \neq 0, \quad \forall s \in S.
\]

Thus \( G \) is positive definite, and we can define the positive definite square root \( F \) of \( G \), i.e., \( F^2 = G \). Define an invertible matrix \( Q \)
\[
Q = \begin{bmatrix} \sqrt{T} I_{|S|} \\ \sqrt{a} F \end{bmatrix}
\]
and denote the matrix in the linearized dynamics (6.23) as \(-J\), i.e.,
\[
J = \begin{bmatrix} I_{|S|} \\ \frac{1}{\tau} G \tilde{K} \end{bmatrix} \begin{bmatrix} -\frac{1}{\alpha} \tilde{K}^T \\ \frac{1}{\tau} G M \end{bmatrix}.
\]
Thus it suffices to show that the real part of the eigenvalues of $Q^{-1}JQ$ is positive. Denote $Q^{-1}JQ$ by $J$. Using the positive semi-definiteness of $FMF$, for any eigenpair $(\lambda, x)$ of $J$ we can deduce

$$\text{Re}(\lambda) = \frac{1}{2} \left( x^H Jx \right) = \frac{1}{2x^H x} \left( x^H \tilde{F} x \right).$$

(6.25)

It remains to show $\text{Re}(\lambda) \neq 0$. Let $x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$, where $x_1 \in \mathbb{R}^{|S|}$, $x_2 \in \mathbb{R}^{|S||A|}$. Then if $\text{Re}(\lambda) = 0$,

$$0 = x^H \left[ \begin{bmatrix} I_{|S|} \\ F \end{bmatrix} \right] x = x_1^H x_1 + (Fx_2)^H M(Fx_2) \geq 0,$$

thus $x_1 = 0$, and the equality condition of the Cauchy-Schwarz inequality (6.15) must hold. Hence $(Fx_2)_{ss} = c_s u_{ss}^*$ for some $c_s \in \mathbb{R}$, $s \in S$. We also know that $c_s$ is not all zero for $s \in S$; otherwise, $x_2 = 0$, so $x = 0$ is not an eigenvector. Thus

$$\tilde{J}x = \begin{bmatrix} I_{|S|} \\ F \end{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{\alpha}} \tilde{F}K \\ FMF \end{bmatrix} x = \begin{bmatrix} \frac{1}{\sqrt{\alpha}} \tilde{F}K \\ FMF \end{bmatrix} x = \begin{bmatrix} \frac{1}{\sqrt{\alpha}} \tilde{F}K \\ FMF \end{bmatrix} x = \begin{bmatrix} -1 \\ 0 \end{bmatrix},$$

which is not a scalar multiple of $x$ unless $\tilde{K}^T F x_2 = 0$. Since

$$\begin{align*}
(\tilde{K}^T F x_2)_{s'} & = \sum_{ss'} K_{ass'} c_s u_{ss'}^* = \sum_{s} K_{\pi ss'} c_s u_{ss'}^*, \quad s' \in S,
\end{align*}$$

$\tilde{K}^T F x_2 = K_{\pi}^T c\tilde{u}_*^*$. Thus $c\tilde{u}_*^* = 0$, contradicting the fact that $c_s$ is not all zero. This contradiction means that $\text{Re}(\lambda) \neq 0$. Together with the inequality (6.16), $\text{Re}(\lambda) > 0$ for any eigenvalue $\lambda$ of $J$. Hence $\text{Re}(\lambda) < 0$ for any eigenvalue $\lambda$ of $-J$, the matrix in the linearized dynamics (6.23).

Finally, by Lyapunov’s indirect theorem [14], (3.3) has locally exponential convergence.}

\[\square\]

REFERENCES

[1] A. Agarwal, S. M. Kakade, J. D. Lee, and G. Mahajan, Optimal model approximation with policy gradient methods in Markov decision processes, in Conference on Learning Theory, PMLR, 2020.

[2] Z. Ahmed, N. Le Roux, M. Norouzi, and D. Schuurmans, Understanding the impact of entropy on policy optimization, in International Conference on Machine Learning, PMLR, 2019.

[3] K. Asadi and M. L. Littman, An alternative softmax operator for reinforcement learning, in International Conference on Machine Learning, PMLR, 2017.

[4] R. E. Bellman and S. E. Dreyfus, Applied dynamic programming, Princeton university press, 2015.

[5] F. Berkenkamp, M. Turchetta, A. P. Schoellig, and A. Krause, Safe model-based reinforcement learning with stability guarantees, (2017), https://arxiv.org/abs/1705.08551.

[6] G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, and W. Zaremba, Openai gym, arXiv preprint arXiv:1606.01540, (2016).

[7] S. Cen, C. Cheng, Y. Chen, Y. Wei, and Y. Chi, Fast global convergence of natural policy gradient methods with entropy regularization, July 2020, https://arxiv.org/abs/2007.06558.

[8] W. S. Cho and M. Wang, Deep primal-dual reinforcement learning: Accelerating actor-critic using bellman duality, Dec. 2017, https://arxiv.org/abs/1712.02467.

[9] Y. Chow, O. Nachum, E. Duenez-Guzman, and M. Ghavamzadeh, A lyapunov-based approach to safe reinforcement learning, May 2018, https://arxiv.org/abs/1805.07708.
[10] B. Dai, A. Shaw, L. Li, L. Xiao, N. He, Z. Liu, J. Chen, and L. Song, *Sbeed: Convergent reinforcement learning with nonlinear function approximation*, in International Conference on Machine Learning, PMLR, 2018.

[11] D. Ding, K. Zhang, T. Basar, and M. Jovanovic, *Natural policy gradient primal-dual method for constrained markov decision processes*, in Advances in Neural Information Processing Systems, 2020.

[12] M. Geist, B. Scherrer, and O. Pietquin, *A theory of regularized Markov decision processes*, in International Conference on Machine Learning, PMLR, 2019.

[13] H. Gong, *Primal-Dual Method for Reinforcement Learning and Markov Decision Processes*, PhD thesis, Princeton University, 2021.

[14] W. M. Haddad and V. Chellaboina, *Nonlinear dynamical systems and control*, Princeton university press, 2011.

[15] S. M. Kakade, *A natural policy gradient*, in Advances in Neural Information Processing Systems, 2001.

[16] R. Kalman and J. Bertram, *Control system analysis and design via the second method of Lyapunov: (i) continuous-time systems (ii) discrete time systems*, IRE Transactions on Automatic Control, 4 (1959), pp. 112–112.

[17] R. E. Kalman and J. E. Bertram, *Control system analysis and design via the “second method” of Lyapunov: I—continuous-time systems*, Journal of Basic Engineering, 82 (1960), pp. 371–393.

[18] S. Khodadadian, P. R. Jhunjhunwala, S. M. Varma, and S. T. Maguluri, *On the linear convergence of natural policy gradient algorithm*, in 2021 69th IEEE Conference on Decision and Control (CDC), IEEE, 2021, pp. 3794–3799.

[19] G. Lan, *Policy mirror descent for reinforcement learning: Linear convergence, new sampling complexity, and generalized problem classes*, Mathematical programming, 198 (2023), pp. 1059–1106.

[20] D. Lee and N. He, *Stochastic primal-dual Q-learning*, Oct. 2018, https://arxiv.org/abs/1810.08298.

[21] G. Li, Y. Wei, Y. Chi, Y. Gu, and Y. Chen, *Softmax policy gradient methods can take exponential time to converge*, Feb. 2021, https://arxiv.org/abs/2102.11379.

[22] H. Li, S. Gupta, H. Yu, L. Ying, and I. Dhillon, *Quasi-newton policy gradient algorithms*, Oct. 2021, https://arxiv.org/abs/2110.02398.

[23] A. M. Lyapunov, *The general problem of the stability of motion*, International journal of control, 55 (1992), pp. 531–534.

[24] 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.

[25] S. P. Meyn and R. L. Tweedie, *Markov chains and stochastic stability*, Springer Science & Business Media, 2012.

[26] O. Nachum, M. Norouzi, K. Xu, and D. Schuurmans, *Trust-pcl: An off-policy trust region method for continuous control*, July 2017, https://arxiv.org/abs/1705.07798.

[27] G. Neu, A. Jonsson, and V. Gómez, *A unified view of entropy-regularized Markov decision processes*, May 2017, https://arxiv.org/abs/1705.07798.

[28] T. J. Perkins and A. G. Barto, *Lyapunov design for safe reinforcement learning*, Journal of Machine Learning Research, 3 (2002), pp. 803–832.

[29] M. L. Puterman, *Markov decision processes: discrete stochastic dynamic programming*, John Wiley & Sons, 2014.

[30] K. Rawlik, M. Toussaint, and S. Vijayakumar, *On stochastic optimal control and reinforcement learning by approximate inference*, in Twenty-third International Joint Conference on Artificial Intelligence, AAAI Press, 2013.

[31] J. Schulman, S. Levine, P. Abbeel, M. Jordan, and P. Moritz, *Trust region policy optimization*, in International Conference on Machine Learning, PMLR, 2015.

[32] J. Schulman, P. Moritz, S. Levine, M. Jordan, and P. Abbeel, *High-dimensional continuous control using generalized advantage estimation*, June 2015, https://arxiv.org/abs/1506.02438.

[33] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, *Proximal policy optimization algorithms*, July 2017, https://arxiv.org/abs/1707.06347.

[34] J. B. Serrano and G. Neu, *Faster saddle-point optimization for solving large-scale Markov decision processes*, in Learning for Dynamics and Control, PMLR, 2020.

[35] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*, MIT press, 2018.

[36] R. S. Sutton, D. A. McAllester, S. P. Singh, and Y. Mansour, *Policy gradient methods for reinforcement learning with function approximation*, in Advances in Neural Information Processing Systems, 2000.

[37] M. Wang, *Primal-dual π learning: Sample complexity and sublinear run time for ergodic Markov decision problems*, Oct. 2017, https://arxiv.org/abs/1710.06100.

[38] M. Wang, *Randomized linear programming solves the Markov decision problem in nearly linear (sometimes sublinear) time*, Mathematics of Operations Research, 45 (2020), pp. 517–546.

[39] M. Wang, *An online primal-dual method for discounted Markov decision processes*, in IEEE 55th Conference on Decision and Control, IEEE, 2016.

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

[41] L. Ying and Y. Zhu, *A note on optimization formulations of Markov decision processes*, Communications in Mathematical Sciences, (2021).

[42] W. Zhan, S. Cen, B. Huang, Y. Chen, J. D. Lee, and Y. Chi, *Policy mirror descent for regularized reinforcement learning: A generalized framework with linear convergence*, May 2021, https://arxiv.org/
[43] J. Zhang, A. S. Bedi, M. Wang, and A. Koppel, *Cautious reinforcement learning via distributional risk in the dual domain*, IEEE Journal on Selected Areas in Information Theory, 2 (2021), pp. 611–626.