On the Linear Convergence of Policy Gradient Methods for Finite MDPs

Jalaj Bhandari
Columbia University
Simons Institute, UC Berkeley

Daniel Russo
Columbia University

Abstract

We revisit the finite time analysis of policy gradient methods in the one of the simplest settings: finite state and action MDPs with a policy class consisting of all stochastic policies and with exact gradient evaluations. There has been some recent work viewing this setting as an instance of smooth non-linear optimization problems and showing sub-linear convergence rates with small step-sizes. Here, we take a different perspective based on connections with policy iteration and show that many variants of policy gradient methods succeed with large step-sizes and attain a linear rate of convergence.

1 Introduction

Policy gradient methods, dating back to the works of [Williams, 1992, Baxter and Bartlett, 1999, Sutton et al., 2000, Marbach and Tsitsiklis, 2001], along with their modern variants [Kakade, 2002, Silver et al., 2014], have emerged as one of the most effective classes of algorithms for solving challenging reinforcement learning problems with impressive empirical success [Schulman et al., 2015, 2017]. Despite this, little was known about their global convergence properties, as these methods search over a parameterized class of policies by performing (stochastic) gradient descent on a scalar loss function that is typically non-convex.

This has changed recently with several recent papers analysing the global convergence properties of policy gradient methods. Our earlier work identifies properties which guarantee that (despite non-convexity) the optimization landscape does not suffer from spurious local optima, thereby implying convergence of policy gradient methods to globally optimal solutions [Bhandari and Russo, 2019]. Though that work does not consider specific algorithms, some convergence rates follow easily from the framework (e.g. a sub-linear convergence rate for projected gradient descent with natural parameterization.) The most comprehensive analysis of convergence rates appears in Agarwal et al. [2020], showing results for different combinations of policy parametrization (natural and softmax policies), algorithms (projected and natural gradient descent) as well as entropy regularization[1]. Shani et al. [2020] focus on analyzing trust region optimization methods [Schulman et al., 2015, 2017] based on mirror descent [Beck and Teboulle, 2003], giving rates for both unregularized and regularized tabular MDPs. Essentially all of these papers view policy optimization as instances of general smooth nonlinear optimization problems. The analyses suggest small step-sizes to control for the error due to local linearization and show convergence to an ϵ–optimal policy within either $O\left(\frac{1}{\epsilon}\right)$ or $O\left(\frac{1}{\epsilon^2}\right)$ iterations, depending on the precise algorithm used.

In this work, we revisit the finite time analysis of policy gradient methods in the simplest setting: finite state and action MDPs with a policy class consisting of all stochastic policies and with exact gradient evaluations. This setting was covered in the aforementioned works of Bhandari and Russo [2019], Agarwal et al. [2020], Shani et al. [2020]. Instead of viewing the problem through the lens of nonlinear optimization, we take a policy iteration perspective. We highlight that many forms of policy gradient can work with extremely large stepsizes and attain a *linear* rate of convergence, meaning they require only $O(\log(1/\epsilon))$ iterations to reach an ϵ–optimal policy. At the core of our ideas is a deep connection between policy gradients and policy iteration, which underlies the analysis in Bhandari and Russo [2019].

For finite MDPs, we show that this leads to an extremely simple analysis covering many different first-order methods applied to the policy gradient objective, including projected gradient descent, Frank-Wolfe, mirror descent, and natural gradient descent. In an idealized setting where step-sizes are set by line search, a one paragraph proof applies to all algorithms. For natural gradient algorithms, a slightly longer calculation studies a specific step-size sequence. In

Proceedings of the 24th International Conference on Artificial Intelligence and Statistics (AISTATS) 2021, San Diego, California, USA. PMLR: Volume 130. Copyright 2021 by the author(s).

[1] Agarwal et al. [2020] also go beyond tabular MDPs to give results for a *compatible* function approximation setting

the final section of this paper, we also discuss a setting of approximate line search as well as natural gradient methods entropy regularization.

Scope and purpose of this work: It is possible that readers might find our setting of tabular MDPs with access to exact gradients somewhat limited. It is worth noting that recent works of [Agarwal et al., 2020, Shani et al., 2020, Cen et al., 2020, Mei et al., 2020] have all compared the convergence rates of different policy gradient methods in this setting. Our work clarifies that with exact gradient evaluations, much faster convergence rates can be achieved with larger step-sizes. The results on line search based step-size selection are especially idealized, but show that classical non-linear optimization techniques would automatically select larger step sizes and attain linear convergence rates.

Small step-sizes may be critical for controlling approximation errors and stabilizing algorithms in practical settings. Studying such issues likely requires a model that focuses on approximation errors and incomplete policy classes. Our work instead offers a clear understanding of what to expect without these challenges.

On concurrent work: We remark on the concurrent works of [Cen et al., 2020, Mei et al., 2020] which also show linear convergence of exact policy gradient methods for entropy regularized tabular MDPs with softmax policies and exact gradients. The main motivation behind these works is to theoretically characterize the benefits of using entropy based regularizers to obtain faster convergence rates. While both analyze different variants (simple gradients vs natural gradients), using entropy regularization seems crucial to their results. Another key difference is that unlike [Cen et al., 2020, Mei et al., 2020], our proof techniques rely on a direct connection between policy gradients and policy iteration, leading to concise proofs that are applicable to a broad range of algorithms along with transparent bounds with a clear dependence on all relevant constants. Instead of leveraging sophisticated algebra, our focus is on giving readers a clear understanding.

2 Problem Formulation

Consider a Markov decision process (MDP), which is a six-tuple $(\mathcal{S}, \mathcal{A}, g, P, \gamma, \rho)$, consisting of a state space $\mathcal{S}$, action space $\mathcal{A}$, cost function g, transition kernel P, discount factor $\gamma \in (0,1)$ and initial distribution ρ. We assume the state space $\mathcal{S}$ to be finite and index the states as $\mathcal{S} = \{s_1, \cdots, s_n\}$. For each state $s \in \mathcal{S}$, we assume that there is a finite set of k arms to choose from and take the action space, $\mathcal{A} = \Delta^{k-1}$ to be the set of all probability distributions over those k arms. That is, any action $a \in \mathcal{A}$ is a probability vector where each component a_i denotes the probability of taking the i-th action. The transition kernel P specifies the probability $P(s'|s,a)$ of transitioning to a state s' upon choosing action a in state s. The cost function $g(s,a) \in \mathbb{R}$ denotes the instantaneous expected cost incurred when selecting action a in state s. Cost and transition functions can be naturally extended to functions on the probability simplex by defining:

$$g(s,a) = \sum_{i=1}^{k} g(s,e_i)\, a_i, \quad P(s'|s,a) \quad = \sum_{i=1}^{k} P(s'|s,e_i)\, a_i. \tag{1}$$

where e_i is the i-th standard basis vector, representing one of the k possible arms. We assume that costs are non-negative, meaning $g(s,e_i) \geq 0$ for all $s \in \mathcal{S}$ and $i \in \{1,\ldots,k\}$. The holds without loss of generality, as one can always add the same large constant to the cost of each state and action without changing the decision problem.

Cost-to-go functions and Bellman operators. A stationary policy $\pi : \mathcal{S} \to \mathcal{A}$ selects a distribution over the $k-1$ dimensional simplex, Δ^{k-1} for each state $s \in \mathcal{S}$. We use the notation $\pi(s,i)$ to denote the probability of selecting action i in state s under policy π. Let Π denote the set of all stationary policies over the simplex,

$$\Pi = \{\pi \in \mathbb{R}_{+}^{n \times k} : \sum_{i=1}^{k} \pi(s,i) = 1 \ \ \forall\, s \in \mathcal{S}\}.$$

For any policy $\pi \in \Pi$, $J_\pi : \mathcal{S} \to \mathbb{R}$ is defined as,

$$J_\pi(s) = \mathbb{E}_\pi \left[\sum_{t=0}^{\infty} \gamma^t g(s_t, \pi(s_t)) \;\middle|\; s_0 = s \right].$$

As the per-step costs are uniformly bounded, so are the cost-to-go functions. Define the Bellman operator $T_\pi : \mathbb{R}^n \to \mathbb{R}^n$ under policy π and the Bellman optimality operator $T : \mathbb{R}^n \to \mathbb{R}^n$ as,

$$\left(T_\pi J\right)(s) := g(s,\pi(s)) + \gamma \sum_{s' \in \mathcal{S}} P(s'|s,\pi(s)) J(s')$$

$$\left(TJ\right)(s) := \min_{a \in \mathcal{A}} \left[g(s,a) + \gamma \sum_{s' \in \mathcal{S}} P(s'|s,a) J(s') \right].$$

Note that the Bellman optimality operator can be equivalently defined as $(TJ)(s) = \min_{\pi \in \Pi} (T_\pi J)(s)$. The cost-to-go function under policy π is the unique solution to the Bellman equation, $J_\pi = T_\pi J_\pi$. Similarly, the optimal cost-to-go function, J^* which satisfies $J^*(s) = \min_\pi J_\pi(s)$ for all $s \in \mathcal{S}$, is the unique fixed point of T and that there is at least one optimal policy, $\pi^* \in \Pi$ that attains this minimum for every $s \in \mathcal{S}$. From the above definitions, it is simple to check that: $J_\pi = T_\pi J_\pi \succeq T J_\pi$ for any $\pi \in \Pi$. We will use this inequality repeatedly throughout our analysis.

Our analysis uses a few basic properties of Bellman operators, see Bertsekas [1995] or Puterman [2014] for proofs. Under the assumption that per-period costs are bounded, T

and T_π are monotone, meaning the element-wise inequality $J \preceq J'$ implies $TJ \preceq TJ'$ and $T_\pi J \preceq T_\pi J'$. They are also contraction operators with respect to the maximum norm. That is, $\|TJ - TJ'\|_\infty \leq \gamma \|J - J'\|_\infty$ and $\|T_\pi J - T_\pi J'\|_\infty \leq \gamma \|J - J'\|_\infty$ hold for for any $J, J' \in \mathbb{R}^n$. The state-action cost-to-go function under policy $\pi \in \Pi$,

$$Q_\pi(s,a) = g(s,a) + \gamma \sum_{s' \in \mathcal{S}} P(s' \mid s,a) J_\pi(s'),$$

measures the cumulative expected cost of taking action a in state s and applying π thereafter. For any polices $\pi, \pi' \in \Pi$, we have the following relations:

$$\begin{aligned} Q_\pi(s, \pi(s)) &= J_\pi(s), \\ Q_\pi(s, \pi'(s)) &= (T_{\pi'} J_\pi)(s), \\ \min_{a \in \mathcal{A}} Q_\pi(s,a) &= (TJ_\pi)(s). \end{aligned}$$

Note that for any policy $\pi \in \Pi, s \in \mathcal{S}$ and $a \in \Delta^{k-1}$, linearity of the cost and transitions functions in (1) implies that the Q-function is linear in a.

$$Q_\pi(s,a) = \sum_{i=1}^{k} Q_\pi(s, e_i) a_i = \langle Q_\pi(s, \cdot), a \rangle$$

Loss function and initial distribution. Policy gradient methods seek to minimize the scalar loss function

$$\ell(\pi) = (1-\gamma) \sum_{s \in \mathcal{S}} J_\pi(s)\, \rho(s),$$

in which the states are weighted by their initial probabilities under ρ and we have normalized costs by $(1-\gamma)$ for convenience. We assume throughout that ρ is supported on $\mathcal{S}$, meaning that $\rho(s) > 0$ for all $s \in \mathcal{S}$ which implies that $\pi \in \arg\min_{\bar{\pi}} \ell(\bar{\pi})$ if and only if $\pi \in \arg\min_{\bar{\pi}} J_{\bar{\pi}}(s) \;\; \forall\, s \in \mathcal{S}$. Assuming an exploratory initial distribution is critical as it is well known that, in the absence of strong assumptions on the transition kernel, policy gradient methods can fail catastrophically if applied without some form of intelligent exploration. See [Thrun, 1992, Kakade and Langford, 2002] for a simple example and the discussions in [Agarwal et al., 2020, Bhandari and Russo, 2019].

State distributions. We define the discounted state occupancy measure under any policy π and initial state distribution ρ as:

$$\eta_\pi = (1-\gamma) \sum_{t=0}^{\infty} \gamma^t \rho P_\pi^t = (1-\gamma)\rho(I - \gamma P_\pi)^{-1}.$$

where η_π and ρ are both row vectors, $P_\pi \in \mathbb{R}^{n \times n}$ denotes the Markov transition matrix under π, i.e. $P_\pi = (P(s'|s, \pi(s)))_{s,s' \in \mathcal{S}}$ and P_π^t denotes its t-step counterpart. Thus, η_π is essentially the discounted fraction of time the system spends in a given state. Note that we have $\eta_\pi(s) \geq (1-\gamma)\rho(s) > 0$ as we assumed $\rho(s) > 0$ for all $s \in \mathcal{S}$.

3 Linear convergence of policy iteration

We briefly revisit the classic policy iteration algorithm as our analysis of policy gradient methods is intricately tied to it. Starting from an initial policy π, policy iteration first evaluates the corresponding cost-to-go function Q_π, and then updates to a new policy π^+ such that

$$\pi^+(s) \in \arg\min_{a \in \mathcal{A}} Q_\pi(s,a) \quad \forall\, s \in \mathcal{S}.$$

In terms of the Bellman operators, this can be equivalently expressed as, $T_{\pi^+} J_\pi = TJ_\pi$. A simple analysis of policy iteration follows by using the monotonicity and contraction properties of the Bellman operators. Observe that

$$J_\pi = T_\pi J_\pi \succeq TJ_\pi = T_{\pi^+} J_\pi. \tag{2}$$

Inductively applying T_{π^+} to each side and using the monotonicity property yields a policy improvement property,

$$J_\pi \succeq T_{\pi^+} J_\pi \succeq T_{\pi^+}^2 J_\pi \succeq \cdots \succeq J_{\pi^+}. \tag{3}$$

Here we use the definition that $J_{\pi^+} = \lim_{k \to \infty} T_{\pi^+} J$ for any $J \in \mathbb{R}^n$. Since $J_\pi \succeq TJ_\pi \succeq J_{\pi^+} \succeq J^*$ we have,

$$\begin{aligned} \|J_{\pi^+} - J^*\|_\infty \leq \|TJ_\pi - J^*\|_\infty &= \|TJ_\pi - TJ^*\|_\infty \\ &\leq \gamma \|J_\pi - J^*\|_\infty, \end{aligned} \tag{4}$$

using the contraction property. From this, we conclude that policy iteration converges to the optimal policy at a linear rate. Let $\{\pi^t\}_{t \geq 0}$ be the set of policies produced by policy iteration. Then iterating over (4) shows

$$\|J_{\pi^t} - J^*\|_\infty \leq \gamma \|J_{\pi^{t-1}} - J^*\|_\infty \leq \cdots \leq \gamma^t \|J_{\pi^0} - J^*\|_\infty.$$

In fact, policy iteration can sometime also converge quadratically in the limit [Puterman, 2014].

4 A sharp connection between policy gradient and policy iteration

Recently, Bhandari and Russo [2019] analyze the optimization landscape of the policy gradient objective $\ell(\cdot)$ for general MDPs and policy classes. A starting point of that analysis is rewriting the policy gradient theorem in a form that emphasizes the illuminating connections between policy gradient and policy iteration. We specialize that presentation to the tabular setting and argue that several first-order methods applied to the policy gradient loss $\ell(\cdot)$ will essentially perform a *soft policy iteration* update and hence converge at a geometric rate, similar to policy iteration.

For any policy $\pi \in \Pi$, consider the weighed policy iteration or ”Bellman” objective, defined as

$$\mathcal{B}(\bar{\pi} | \eta_\pi, J_\pi) = \sum_{s \in \mathcal{S}} \eta_\pi(s) Q_\pi(s, e_i) \bar{\pi}(s,i) = \langle Q_\pi\,, \bar{\pi} \rangle_{\eta_\pi \times 1}$$

where e_i denotes the i-th standard basis vector, denoting one of the k arms, $\langle v, u\rangle_W = \sum_{s=1}^{n}\sum_{i=1}^{k} v(s,i)u(s,i)W(s,i)$ denotes the W-weighted inner product and $\eta_\pi \times 1$ denotes a weighting that places weight $\eta_\pi(s)\cdot 1$ on any state-action pair (s,i). Recall that since $\rho(s) > 0$ by assumption, $\eta_\pi(s) > 0$ for all $s \in \mathcal{S}$ and hence the policy iteration update can be equivalently written as optimizing the Bellman objective,

$$\pi^+ = \arg\min_{\bar\pi\in\Pi} \mathcal{B}(\bar\pi|\eta_\pi, J_\pi).$$

It is worth emphasizing that the Bellman cost function is a *single period* objective, considering the cost-to-go of following $\bar\pi$ for a single period and following π thereafter. A policy gradient theorem connects gradients of the infinite horizon cost function $\ell(\cdot)$ to gradients of the *single period* Bellman objective underlying policy iteration. In particular, we have the following lemma from Bhandari and Russo [2019], which is essentially a restatement of the classical version by [Sutton et al., 2000, Sutton and Barto, 2018].

Lemma 1 (Policy gradient theorem for tabular MDPs)**.** *Assuming per-period costs are uniformly bounded, $\ell(\pi)$ is continuously differentiable and*

$$\frac{\partial \ell(\pi)}{\partial \pi(s,i)} = \left.\frac{\partial \mathcal{B}(\bar\pi|\eta_\pi, J_\pi)}{\partial \bar\pi(s,i)}\right|_{\bar\pi=\pi} = \eta_\pi(s)Q_\pi(s,e_i)$$

Equivalently, we can write a first order Taylor expansion of $\ell(\cdot)$ as

$$\begin{aligned}\ell(\bar\pi) &= \ell(\pi) + \langle \nabla\ell(\pi)\,,\, \bar\pi - \pi\rangle + O(\|\bar\pi-\pi\|^2)\\ &= \ell(\pi) + \langle Q_\pi\,,\, \bar\pi - \pi\rangle_{\eta_\pi\times 1} + O(\|\bar\pi-\pi\|^2).\end{aligned}$$

Presentation of the policy gradient theorem in terms of the Bellman objective clarifies an important connection – we can interpret $\nabla\ell(\pi)$ as gradient of the weighted policy iteration objective. What is special about the tabular setting, relative to the general problems considered by Bhandari and Russo [2019], is that the weighted policy iteration objective is *linear*. In the following section, we use this connection to show that various first-order methods applied to $\ell(\cdot)$ can optimize the Bellman objective $\mathcal{B}(\cdot|\eta_\pi, J_\pi)$ to optimality in a single update with large (and possibly infinite) stepsizes; equivalent to a policy iteration update. For finitely large step-sizes, a simple argument establishes equivalence between a policy gradient step and a soft policy iteration update, again implying geometric convergence.

Note that for tabular MDPs, a policy iteration step is simple as it reduces to solving a linear optimization problem over the probability simplex, and the solution is to select the best action for each state.

5 Policy gradient methods for finite MDPs

We write all algorithms in terms of their evolution in the space of policies Π. Several of them could instead be viewed as operating in the space of parameters for some parameterized policy class. We discuss this in Remark 1, but keep our formulation and results focused on the space of policies Π. Note that $\Pi = \Delta^{k-1}\times\cdots\times\Delta^{k-1}$ is the n-fold product of the probability simplex. This form of the policy class will cause policy gradient updates to decouple across states.

Frank-Wolfe. Starting with some policy $\pi \in \Pi$, an iteration of the Frank-Wolfe algorithm computes

$$\pi^+ = \arg\min_{\bar\pi\in\Pi} \langle \nabla\ell(\pi), \bar\pi\rangle = \arg\min_{\bar\pi\in\Pi} \langle Q_\pi, \bar\pi\rangle_{\eta_\pi\times 1} \quad (5)$$

and then updates the policy to $\pi' = (1-\alpha)\pi + \alpha\pi^+$ for $\alpha\in[0,1]$. We use the notation π^+ in (5) as it is exactly the policy iteration update to π so *Frank-Wolfe mimics a soft-policy iteration step*, akin to the *conservative policy iteration* update[2] in Kakade and Langford [2002]. Note, the minimization problem in (5) decouples across states to optimize a linear objective over the probability simplex, so

$$\pi^+(s) \in \arg\min_{d\in\Delta^{k-1}} d^\top Q_\pi(s,\cdot)$$

is a point-mass that places all weight on $\arg\min_i Q_\pi(s,e_i)$.

Projected Gradient Descent. Starting with some policy $\pi\in\Pi$, an iteration of the projected gradient descent algorithm with constant stepsize α updates to the solution of a regularized problem

$$\begin{aligned}\pi' &= \arg\min_{\bar\pi\in\Pi}\langle\nabla\ell(\pi),\bar\pi\rangle + \frac{1}{2\alpha}\|\bar\pi-\pi\|_2^2\\ &= \arg\min_{\bar\pi\in\Pi}\langle Q_\pi,\bar\pi\rangle_{\eta_\pi\times 1} + \frac{1}{2\alpha}\|\bar\pi-\pi\|_2^2.\end{aligned}$$

As $\alpha\to\infty$ (the regularization term tends to zero), π' converges to the solution of (5), which is exactly the policy iteration update as noted above. For intermediate values of α, the projected gradient update decouples across states and takes the form:

$$\pi'_s = \mathrm{Proj}_{2,\Delta^{k-1}}(\pi_s - \alpha Q_\pi(s,\cdot))$$

which is a gradient step followed by a projection onto the probability simplex. Note that from an implementation perspective, projections onto the probability simplex involves a computationally efficient ($\mathcal{O}(k\log k)$) soft-thresholding operation Duchi et al. [2008].

Mirror-descent. The mirror descent method adapts to the geometry of the probability simplex by using a non-euclidean regularizer. We focus on using the Kullback Leibler (KL) divergence, a natural choice for the regularizer,

[2] A generalized version of Frank-Wolfe was studied in [Scherrer and Geist, 2014] under the name of "Boosted Policy Search" to show global optimality guarantees for any locally optimal policy.

under which an iteration of mirror descent updates policy π to π' as:

$$\begin{aligned}\pi' &= \arg\min_{\bar{\pi}\in\Pi}\langle \nabla\ell(\pi), \bar{\pi}\rangle + \frac{1}{\alpha}\sum_{s=1}^{n} D_{\mathrm{KL}}(\bar{\pi}_s \,||\, \pi_s) \\ &= \arg\min_{\bar{\pi}\in\Pi}\langle Q_\pi, \bar{\pi}\rangle_{\eta_\pi\times 1} + \frac{1}{\alpha}\sum_{s=1}^{n} D_{\mathrm{KL}}(\bar{\pi}_s \,||\, \pi_s),\end{aligned}$$

where $D_{\mathrm{KL}}(p||q) = \sum_{i=1}^{k} p_i \log(p_i/q_i)$ denotes the KL divergence. It is well know that the solution to this optimization problem is the exponentiated gradient update [Bubeck et al., 2015, Section 6.3],

$$\pi'_{s,i} = \frac{\pi_{s,i}\cdot\exp\{-\alpha\eta_\pi(s)Q_\pi(s,e_i)\}}{\sum_{j=1}^{k}\pi_{s,j}\cdot\exp\{-\alpha\eta_\pi(s)Q_\pi(s,e_j)\}}.$$

Again, we can see that π' converges to a policy iteration update as $\alpha\to\infty$.

Natural policy gradient and TRPO. We consider the natural policy gradient (NPG) algorithm of Kakade [2002] which is closely related to the widely used TRPO algorithm of Schulman et al. [2015]. We focus on NPG applied to the *softmax parameterization* for which it is actually an instance of mirror descent with a specific regularizer. In particular, beginning with some policy $\pi\in\Pi$, an iteration of NPG updates to π':

$$\begin{aligned}\pi' &= \arg\min_{\bar{\pi}\in\Pi}\ \langle \nabla\ell(\pi), \bar{\pi}\rangle + \frac{1}{\alpha}\sum_{s=1}^{n}\eta_\pi(s) D_{\mathrm{KL}}(\bar{\pi}_s \,||\, \pi_s) \\ &= \arg\min_{\bar{\pi}\in\Pi}\ \langle Q_\pi, \bar{\pi}\rangle_{\eta_\pi\times 1} + \frac{1}{\alpha}\sum_{s=1}^{n}\eta_\pi(s) D_{\mathrm{KL}}(\bar{\pi}_s \,||\, \pi_s)\end{aligned} \tag{6}$$

where we use a regularizer that penalizes changes to the the action distribution at states in proportion to their occupancy measure η_π. As discussed above, it is well known that this KL divergence regularized problem is solved by an exponential weights update for each state $s\in\{1,\ldots,n\}$,

$$\pi'(s,i) = \left(\frac{\pi_{s,i}\cdot\exp\{-\alpha Q_\pi(s,e_i)\}}{\sum_{j=1}^{k}\pi_{s,j}\cdot\exp\{-\alpha Q_\pi(s,e_j)\}}\right). \tag{7}$$

Note that this update rule is independent of the state occupancy measure η_π. A potential source of confusion is that natural policy gradient is usually described as steepest descent in a variable metric defined by the Fisher information matrix induced by the current policy[3],

$$\begin{aligned}\pi' &= \pi + \alpha F(\pi)^\dagger \nabla\ell(\pi) \\ F(\pi) &= \sum_{s,i}\eta_\pi(s)\pi(s,i)\left[\nabla\log\pi(s,i)\left(\nabla\log\pi(s,i)\right)^\top\right]\end{aligned}$$

where $M^\dagger$ denotes the pseudoinverse of matrix M. Readers can check that the exponentiated update in (7) matches the explicit formula for the NPG update with softmax policies as given in Kakade [2002] and Agarwal et al. [2020].

Step-size selection is an important issue for most first order methods. Each of the algorithms above can be applied with a sequence of stepsizes $\{\alpha_t\}_{t\geq 0}$ to produce a sequence of policies $\{\pi^t\}_{t\geq 0}$. We define one stepsize selection rule below.

Exact line search. At iteration t, the update rules for each of the algorithms described above actually specify a new policy π_α^{t+1} for a range of stepsizes, $\alpha\geq 0$. We consider an idealized stepsize rule using *exact line search*, which directly optimizes over this choice of stepsize at each iteration, selecting $\pi^{t+1} = \pi_{\alpha^*}^{t+1}$ where $\alpha^* = \arg\min_\alpha \ell(\pi_\alpha^{t+1})$ whenever this minimizer exists. More generally, we define

$$\pi^{t+1} = \arg\min_{\pi\in\Pi^{t+1}}\ell(\pi). \tag{8}$$

where $\Pi^{t+1} = \mathrm{Closure}(\{\pi_\alpha^{t+1}\})$ denotes the closed curve of policies traced out by varying α. For Frank-Wolfe, $\Pi^{t+1} = \{\alpha\pi^t + (1-\alpha)\pi_+^t : \alpha\in[0,1]\}$ is the line segment connecting the current policy π^t and its policy iteration update π_+^t. Under NPG, $\Pi^{t+1} = \{\pi_\alpha^{t+1}\}$ is a curve where $\pi_0^{t+1} = \pi^t$ and $\pi_\alpha^{t+1}\to\pi_+^t$ as $\alpha\to\infty$. Since π_+^t is not attainable under any fixed α, this curve is not closed. By taking the closure, and defining line search via (8), certain formulas become cleaner. Of course, it is also possible to nearly solve (8) without taking the closure and obtain essentially the same results. We elaborate on this in the discussion that follows our main result in Theorem 1.

Remark 1 (Policy parameterization and infima vs minima)**.** *We chose to work with the class of all stochastic policies* Π *(often termed as natural parameterization) as opposed to some parameterized policy classes, which are more commonly used in practice. For example, a policy gradient algorithm might search over the parameter* $\theta\in\mathbb{R}^{n\times k}$ *of a softmax policy* $\pi_\theta\in\Pi$*, defined by* $\pi_\theta(s,i)\propto e^{\theta_{s,i}}$*. For example, consider the TRPO algorithm proposed by [Schulman et al., 2015]. This forms a locally linearization of* $\ell(\pi)$*, forms the regularized minimization problem in (6), and then updates the parameter of a softmax policy* π_θ *by solving*

$$\arg\min_{\theta}\ \langle Q_{\pi_\theta}, \pi_{\bar{\theta}}\rangle_{\eta_{\pi_\theta}\times 1} + \frac{1}{\alpha}\sum_{s=1}^{n}\eta_{\pi_\theta}(s) D_{\mathrm{KL}}(\pi_{\bar{\theta}}(s)\,||\,\pi_\theta(s)).$$

We could define similar versions of projected gradient descent or Frank-Wolfe, which also linearize $\ell(\pi)$*, but then optimize the resulting local approximation only over parameterized policies. Since the class of softmax policies can approximate any stochastic policy to arbitrary precision, however, this is nearly the same as optimizing over the class* Π*. Studying* Π *directly makes mathematical analysis easier, because it is closed. For example, it contains*

[3] This is equivalent to mirror descent under some conditions Raskutti and Mukherjee [2015].

an optimal policy, whereas any softmax policy $\pi_\theta \in \Pi_\Theta$ can only come infinitesimally close to an optimal policy. In practice, optimization problems are never solved beyond machine precision, so we don't view the distinction between infimum and minimum to be relevant to the paper's main insights. We caution the reader that our results do not apply to more more naive gradient methods that directly linearize $\ell(\pi_\theta)$ with respect to θ. In that case, a gradient update to θ may not approximate a policy iteration update, no matter how large the stepsize is chosen to be. In fact, such methods may perform very poorly due to issues of poor conditioning [Kakade, 2002].

6 Main result: geometric convergence

So far, we have described different variants of policy gradient methods for tabular MDPs. For large step-sizes, all these algorithms essentially make a policy iteration update. Hence, intuitively, it is reasonable to expect that their convergence behavior closely resemble that of policy iteration rather than that of gradient descent for smooth objectives. We quantify this precisely in Theorem 1 below.

Our first result confirms that all of the algorithms we presented in the previous section converge geometrically when step-sizes are set by exact line search on $\ell(\cdot)$. Again, the idea is that *a policy gradient step is a policy iteration update* for an appropriate choice of stepsize. Our proof effectively uses that exact line search updates make at least as much progress in reducing $\ell(\cdot)$ as a policy iteration update. The mismatch between the policy gradient loss $\ell(\cdot)$, which governs the stepsize choice, and the maximum norm, which governs policy iteration convergence, is the source of the term $\min_{s\in\mathcal{S}} \rho(s)$ in the bound. We further elaborate on this issue in the discussion that follows Theorem 1.

Our second and third results show that dependence on the initial distribution in the bounds can be avoided by forcing the algorithm to use large stepsizes. A simple result in part (b) applies to the Frank-Wolfe algorithm with a constant stepsize, which gives performance improvement in max norm. This bound follows by essentially making a minor modification to the linear convergence result of policy iteration as reviewed in Section 3. Recall that we already showed a Frank-Wolfe update to be exactly equivalent to a soft policy iteration update,

$$\pi^{t+1}(s) = (1-\alpha)\pi^t(s) + \alpha\pi_+^t(s).$$

Given this close connection, a simple argument shows that an α-step Frank-Wolfe update offers at least a fraction of the performance improvement offered by a policy iteration update,

$$J_{\pi^{t+1}} \leq (1-\alpha)J_{\pi^t} + \alpha T J_{\pi^t}$$

which implies the result. A comparison between parts (a) and (b) suggest that for $\alpha \geq 1/|\mathcal{S}|$, Frank-Wolfe with exact line search might converge slowly in the worst case.

For softmax policies and exact gradient evaluations, we show in part (c) that NPG with an *adaptive stepsize sequence* converges to an ϵ optimal policy in $O(\log(1/\epsilon))$ iterations. The error term, ϵ, is inversely related to the stepsize and reflects the fact that NPG updates with finite stepsizes only approximately resemble the policy iteration updates[4]. As we take the step-size to infinity, we recover the same result as one would expect for policy iteration. Compared to the first result in part (a) which applies with exact line search, the result in part (c) is useful in the sense that it gives a precise quantification of how large the step-sizes need to be for linear convergence to hold.

Theorem 1 (Geometric convergence)**.** *Suppose one of the first-order algorithms in Section 5 is applied to minimize $\ell(\pi)$ over $\pi \in \Pi$ with step-size sequence $\{\alpha_t\}_{t\geq 0}$. Let π^0 denote the initial policy and $\{\pi^t\}_{t\geq 0}$ denote the sequence of iterates. The following bounds apply.*

(a) ***Exact line search.*** *If either Frank-Wolfe, projected gradient descent, mirror descent, or NPG is applied with stepsizes chosen by exact line search as in (8), then*

$$\|J_{\pi^t} - J^*\|_\infty \leq \left(1 - \min_{s\in\mathcal{S}} \rho(s)(1-\gamma)\right)^t \frac{\|J_{\pi^0} - J^*\|_\infty}{\min_{s\in} \rho(s)}.$$

(b) ***Constant stepsize Frank-Wolfe.*** *Under Frank-Wolfe with constant stepsize $\alpha \in (0,1]$,*

$$\|J_{\pi^t} - J^*\|_\infty \leq (1-\alpha(1-\gamma))^t \|J_{\pi^0} - J^*\|_\infty.$$

(c) ***Natural policy gradient with softmax policies and adaptive stepsize.*** *Fix any $\epsilon > 0$. Let $i_t^* = \arg\min_i Q_{\pi^t}(s,i)$. Suppose that NPG is performed with an adaptive step-size sequence,*

$$\alpha_t(s) \geq \frac{2}{(1-\gamma)\epsilon} \log\left(\frac{2}{\pi^t(s,i_t^*)}\right).$$

Then,

$$\|J_{\pi^t} - J^*\|_\infty \leq \left(\frac{1+\gamma}{2}\right)^t \|J_{\pi^0} - J^*\|_\infty + \epsilon.$$

Remark 2. *For the result in part (c), note that for the softmax parameterization, $\pi_\theta(s,i) > 0$ for any $\theta \in \mathbb{R}^{n\times k}$. So, $\pi^t(s,i_t^*) > 0$ for all t. A similar result can also be obtained without the need of adaptive step-sizes by considering entropy regularized MDPs. This is discussed below.*

Discussion of results: The following discussion is based primarily on feedback of the reviewers. We thank them for their valuable inputs.

[4] More precisely, our proof shows that in this case, the NPG update is equivalent to a soft policy iteration update upto some additive error.

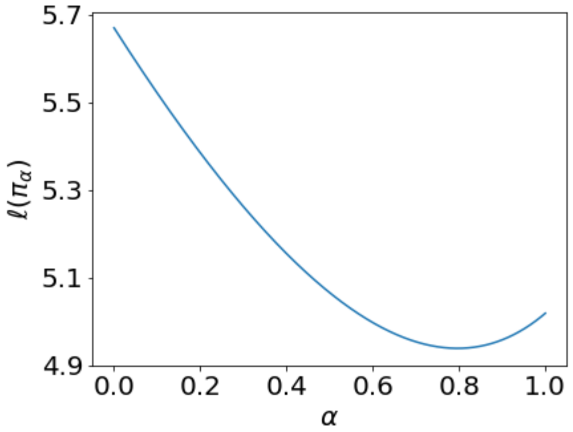

Figure 1: Line search objective for a Frank-Wolfe update for a two state three action MDP is non-monotonic with a minimum at $\alpha = 0.83$. Therefore, exact line search picks a smaller step-size than the greedy update, i.e. $\pi_{\alpha^*} \neq \pi_+$.

1. Dependence on $\rho_{\min}$ for exact line search result: Readers will note that proof of our result in part (a) of Theorem 1 also shows that,

$$\ell(\pi^{t+1}) - \ell(\pi^*) \leq (1 - \rho_{\min}(1-\gamma))^t \, \ell(\pi^0) - \ell(\pi^*).$$

A natural question to ask is whether the presence of the factor of $\rho_{\min}$ in the geometric rate is merely an artifact of our analysis technique and if in practice, line search always ends up picking the policy iteration update. In Figure 1 below, we plot the line search objective,

$$\{\ell(\pi_\alpha) : \pi_\alpha = \alpha\pi + (1-\alpha)\pi_+, \, \alpha \in [0,1]\},$$

for a Frank-Wolfe update for a randomly generated[5] MPD with two states and three actions. For a given (P, g, ρ) and policy π (see Appendix B for details), we observe that $\ell(\pi_\alpha)$ is non-monotonic and so exact line-search does often select smaller step-sizes as compared to the greedy update ($\alpha = 1$). Although we do not show a lower bound, this example suggests that a factor of $\rho_{\min}$ in the bounds might be unavoidable.

2. On inexact line search: Though our result in part (a) focuses on an idealized setting with exact line search, we do note that a similar result can also be obtained if we can ensure, say using inexact line search, that the improvement in total cost $\ell(\cdot)$ at every update is at least a fraction of the improvement offered by exact line search. For example, if we select a step-size sequence $\{\alpha_t\}_{t\geq 0}$ which offers half the possible improvement at every update, meaning $\ell(\pi^t) - \ell(\pi^{t+1}_{\alpha_t}) \geq (1/2)(\ell(\pi^t) - \inf_{\alpha'} \ell(\pi^{t+1}_{\alpha'}))$, then our result in part (a) follows with an extra factor of $\frac{1}{2}$ in the bound. One essentially needs to modify the first step in the proof (Equation (11)) and the rest is the same.

A linear convergence result can also be obtained if the sequence of policies, $\{\pi^t\}_{t\geq 0}$ obtained via inexact line search offer approximately the same improvement as a policy iteration update, i.e. $\ell(\pi^{t+1}) \leq \ell(\pi^t_+) + \delta$ holds uniformly for some $\delta > 0$. In this case, a bound similar to that in part (a) will hold with an additional scaled bias term of $\delta/(1-\gamma)$.

3. NPG with softmax policies for regularized MDPs: Recall that the result in part (c) uses an adaptive step-size sequence $\alpha_t(s)$ that depends on $\pi^t(s, i_t^*)$, the probability under the randomized policy at iteration t assigned to the action i_t^* prescribed by policy iteration. This dependence is a bit undesirable and can be removed by considering *entropy regularized* MDPs. Entropy regularization prevents policies from picking near deterministic actions and essentially lower bounds $\pi^t(s, i^*)$. Rather than presenting a lengthy re-derivation of the result in part (c), we sketch a simple argument essentially based on some past work on the theory of regularized MDPs [Neu et al., 2017, Geist et al., 2019], to show linear convergence with a particular choice of step-size. Although this result in (9) is almost identical to the one in Cen et al. [2020], our ideas, based on connections to policy iteration, considerably simplify the proof.

A common way to enforce regularization is by adding a a small penalty to the cost function,

$$g^\lambda(s,a) = \sum_{i=1}^{k} (g(s, e_i)a_i + \lambda \log(a_i))$$

for some parameter $\lambda > 0$. Let $J_\pi^\lambda(s)$ and $Q_\pi^\lambda(s,a)$ be the corresponding cost-to-go functions for any $\pi \in \Pi$,

$$J_\pi^\lambda(s) = \mathbb{E}_\pi\left[\sum_{t=0}^{\infty} \gamma^t g^\lambda(s_t, \pi(s_t)) \;\middle|\; s_0 = s\right],$$

$$Q_\pi^\lambda(s,a) = g(s,a) + \gamma \sum_{s' \in \mathcal{S}} P(s'|s,a) J_\pi^\lambda(s').$$

Similar to (6), a quick calculation using the policy gradient theorem reveals that an NPG update for a λ-regularized MDP solves the following problem,

$$\arg\min_{\bar\pi \in \Pi} \langle \nabla \ell^\lambda(\pi), \bar\pi \rangle_{\eta_\pi \times 1} + \frac{1}{\alpha} \sum_{s\in\mathcal{S}} \eta_\pi(s) D_{\mathrm{KL}}(\bar\pi_s || \pi_s)$$

for any $\alpha \leq 1/\lambda$ with $\nabla \ell^\lambda(\pi) = Q_\pi^\lambda + \lambda \log \pi$. For $\alpha = 1/\lambda$, these updates take a particularly simple form of $\pi'(s) = \text{Softmax}\left(\frac{-Q_\pi^\lambda(s,\cdot)}{\lambda}\right)$. This update can alternatively be viewed as a policy iteration update with

[5] We generated many random MDPs to compare updates of policy iteration with those of Frank-Wolfe using grid search and found many cases where these differ. Details of only one such example is given to illustrate our point.

respect to a regularized Bellman optimality operator, $T^\lambda(\cdot)$ defined by:

$$(T^\lambda J^\lambda_\pi)(s) = \min_{\bar{\pi}\in\Pi} \langle Q^\lambda_\pi(s,\cdot), \bar{\pi}(s)\rangle + \lambda\mathcal{H}(\pi(\bar{s})).$$

where $\mathcal{H}(\pi(s)) = \sum_{i=1}^{k} \pi(s,i)\log\pi(s,i)$ is the negative entropy. Importantly, $T^\lambda(\cdot)$ can also be shown to be a monotone and γ-contraction in the maximum norm with a unique fixed point, $J^{*,\lambda}$ such that $\|J^{*,\lambda} - J^*\|_\infty \leq \frac{\lambda \log k}{(1-\gamma)}$. See [Geist et al., 2019] for details. Therefore, similar to the proof of policy iteration in Section 3, we can obtain a geometric convergence result for NPG with softmax policies and a constant step-size of $\alpha = 1/\lambda$,

$$\|J_{\pi^t} - J^*\|_\infty \leq \gamma^t \|J_{\pi^0} - J^*\|_\infty + \frac{2\lambda \log k}{(1-\gamma)^2}. \tag{9}$$

6.1 Proof of Theorem 1

Proof. Throughout, we use some standard properties of the Bellman operator as described in Section 3. We denote π^t_+ to be the policy iteration update to any policy $\pi^t \in \Pi$ and $\|\cdot\|$ to be the ℓ_∞-norm.

Part (a): Exact line-search: Under each algorithm and at each iteration t, the policy iteration update π^t_+ is contained in the class Π^{t+1} introduced in (8). Therefore, for each algorithm,

$$\ell(\pi^{t+1}) = \min_{\pi\in\Pi^t} \ell(\pi) \leq \ell(\pi^t_+) \tag{10}$$

Recall policy improvement property in (3), which shows $J^* \preceq J_{\pi^t_+} \preceq TJ_{\pi^t} \preceq J_{\pi^t}$. Denote $\rho_{\min} := \min_{s\in\mathcal{S}} \rho(s)$. We have,

$$\begin{aligned}
\ell(\pi^t) - \ell(\pi^{t+1}) &\geq \ell(\pi^t) - \ell(\pi^t_+) \qquad (11)\\
&= \sum_{s\in\mathcal{S}} \rho(s) \left(J_{\pi^t}(s) - J_{\pi^t_+}(s) \right)\\
&\geq \rho_{\min} \left(\sum_{s\in\mathcal{S}} J_{\pi^t}(s) - J_{\pi^t_+}(s) \right)\\
&\geq \rho_{\min} \|J_{\pi^t} - J_{\pi^t_+}\|\\
&\geq \rho_{\min} \|J_{\pi^t} - TJ_{\pi^t}\|\\
&= \rho_{\min} \|J_{\pi^t} - J^* - (TJ_{\pi^t} - J^*)\|\\
&\geq \rho_{\min} \left(\|J_{\pi^t} - J^*\| - \|TJ_{\pi^t} - J^*\|\right)\\
&= \rho_{\min} \left(\|J_{\pi^t} - J^*\| - \|TJ_{\pi^t} - TJ^*\|\right)\\
&\geq \rho_{\min} (1-\gamma) \|J_{\pi^t} - J^*\|\\
&\geq \rho_{\min} (1-\gamma) \left(\ell(\pi^t) - \ell(\pi^*)\right).
\end{aligned}$$

where the penultimate inequality follows by contractivity property of the Bellman operator, $\|TJ_{\pi^t} - TJ^*\| \leq \gamma\|J_{\pi^t} - J^*\|$ Rearranging terms gives,

$$\begin{aligned}
\ell(\pi^{t+1}) - \ell(\pi^*) &\leq (1 - \rho_{\min}(1-\gamma)) \left(\ell(\pi^t) - \ell(\pi^*)\right)\\
&\leq (1 - \rho_{\min}(1-\gamma))^t \left(\ell(\pi^0) - \ell(\pi^*)\right),
\end{aligned}$$

where the final inequality follows by inductively applying the first one. We immediately have the looser bound $\ell(\pi^{t+1}) - \ell(\pi^*) \leq (1 - \rho_{\min}(1-\gamma))^t \|J_{\pi^0} - J^*\|$. The final result follows from observing that

$$\|J_{\pi^t} - J^*\|_\infty \leq \left(\ell(\pi^{t+1}) - \ell(\pi^*)\right) / \rho_{\min}$$

Part (b): Constant stepsize Frank-Wolfe: The proof follows the policy iteration analysis reviewed in Section 3. Recall from Section 5 that a Frank-Wolfe update is equivalent to a soft policy iteration update:

$$\pi^{t+1}(s) = (1-\alpha)\pi^t(s) + \alpha\pi^t_+(s)$$

where π^t_+ is the policy iteration update to π^t. Thus, starting from a feasible policy $\pi^0 \in \Pi$, we always maintain feasibility for $\alpha \in (0,1]$. By linearity of the cost and transition functions as shown in (1), we have that for any state s,

$$\begin{aligned}
T_{\pi^{t+1}} J_{\pi^t}(s) &= (1-\alpha) J_{\pi^t}(s) + \alpha T_{\pi^t_+} J_{\pi^t}(s)\\
&= (1-\alpha) J_{\pi^t}(s) + \alpha T J_{\pi^t}(s)
\end{aligned}$$

Using $TJ_{\pi^t} \preceq J_{\pi^t}$ as in (2), we get

$$T_{\pi^{t+1}} J_{\pi^t} = (1-\alpha) J_{\pi^t} + \alpha T J_{\pi^t} \preceq J_{\pi^t}. \tag{12}$$

Using monotonicity of $T_{\pi^{t+1}}$, along with the fact that $J_{\pi^{t+1}} = \lim_{n\to\infty} T^n_{\pi^{t+1}} J_{\pi^t}$ implies,

$$J_{\pi^t} \succeq T_{\pi^{t+1}} J_{\pi^t} \succeq T^2_{\pi^{t+1}} J_{\pi^t} \succeq \ldots \succeq J_{\pi^{t+1}}$$

Therefore, from (12), we get

$$J_{\pi^{t+1}} \preceq T_{\pi^{t+1}} J_{\pi^t} = (1-\alpha) J_{\pi^t} + \alpha T J_{\pi^t}.$$

Subtracting J^* from both sides shows

$$J_{\pi^{t+1}} - J^* \preceq (1-\alpha)\left(J_{\pi^t} - J^*\right) + \alpha\left(TJ_{\pi^t} - J^*\right).$$

Since the above inequality holds element wise,

$$\begin{aligned}
\|J_{\pi^{t+1}} - J^*\| &\leq (1-\alpha)\|J_{\pi^t} - J^*\| + \alpha\|TJ_{\pi^t} - J^*\|\\
&\leq \left((1-\alpha) + \gamma\alpha\right)\|J_{\pi^t} - J^*\|,
\end{aligned}$$

where we use that $J^* = TJ^*$ and $\|TJ_{\pi^t} - TJ^*\| \leq \gamma\|J_{\pi^t} - J^*\|$ as $T(\cdot)$ is a γ-contraction. Iterating over the above equation gives us our final result:

$$\|J_{\pi^{t+1}} - J^*\|_\infty \leq (1 - \alpha(1-\gamma))^t \|J_{\pi^0} - J^*\|.$$

Part (c): Proof for natural policy gradient with softmax policies and adaptive step-sizes: Recall that in Section 5, the natural policy gradient (NPG) update with a step-size sequence $\{\alpha_t\}_{t\geq 0}$ take the form:

$$\pi^{t+1}(s,i) = \frac{\pi^t(s,i)\cdot e^{-\alpha_t(s) Q^t(s,i)}}{\sum_{j=1}^{k} \pi^t(s,j)\cdot e^{-\alpha_t(s) Q^t(s,j)}},$$

where we use the shorthand notation $\pi^t(\cdot)$ to denote $\pi_{\theta^t}(\cdot)$ and $Q^t(s,i)$ to denote $Q_{\pi_{\theta^t}}(s,i)$. For simplicity, we let $c := \frac{2}{(1-\gamma)}$ which implies, $\alpha_t(s) \geq \frac{c}{\epsilon}\log\left(\frac{2}{\pi^t(s,i^*)}\right)$.

Our proof strategy shows that for any state $s \in \mathcal{S}$, an NPG update with step-size $\alpha_t(s)$ decreases the probability of *sub-optimal* actions by a multiplicative factor. Informally, the set of sub-optimal actions per state can be understood to be the set of actions with action gap[6] larger than some threshold. Essentially, this shows the NPG update is equivalent to a soft policy iteration update upto a small additive error. We divide the proof into three steps.

Step 1: NPG update for *sub-optimal* actions: Fix some state $s \in \mathcal{S}$. Without loss of generality, we assume the following ordering on the Q-values: $Q^t(s,1) < Q^t(s,2) \ldots < Q^t(s,k)$ which implies that action 1 is optimal in state s under policy π^t. For error tolerance $\epsilon > 0$, define $O_t^-(s)$ and $O_t^+(s)$ as:

$$O_t^-(s) := \left\{ i \mid Q^t(s,i) - Q^t(s,1) \geq \frac{\epsilon}{c} \right\}$$
$$O_t^+(s) := \left\{ i \mid Q^t(s,i) - Q^t(s,1) < \frac{\epsilon}{c} \right\}$$

The set $O_t^-(s)$ can be interpreted as the set of *sub-optimal* actions with the *action gap*, $Q^t(s,i) - Q^t(s,1)$, larger than the threshold ϵ/c. Similarly, $O_t^+(s)$ can be interpreted to be the set of *nearly optimal* actions according to policy π^t. The following lemma (proved in Ashows that NPG updates decrease the probability of playing sub-optimal actions by a multiplicative factor.

Lemma 2. *For any state s, $\frac{\pi^{t+1}(s,i)}{\pi^t(s,i)} \leq \frac{1}{2} \;\; \forall i \in O_t^-(s)$.*

Step 2: NPG updates as soft policy iteration: The policy iteration update, $\pi_+^t(s) = \arg\min_{i\in\{1,2,\ldots,k\}} \; Q^t(s,i)$, puts entire mass on the best action (according to Q-values of the current policy) and zeros out the probability of playing other actions. On the other hand, Lemma 2 shows how an NPG update with appropriate stepsize decays the probabilities of *sub-optimal* actions (in the set $O_t^-(s)$) by a multiplicative factor instead of zeroing them out[7]. This resembles a *soft-policy iteration* update for the set of actions $O_t^-(s)$. We formalize this intuition in the following lemma which characterizes the progress made by an NPG update vis-a-vis a policy iteration update.

Lemma 3 (Progress quantification)**.** *Let $J_{\pi^t}(s)$ denote the cost-to-go function for policy π^t from any starting state $s \in \mathcal{S}$. Then,*

$$T_{\pi^{t+1}} J_{\pi^t}(s) - J_{\pi^t}(s) \leq \frac{1}{2} \cdot (TJ_{\pi^t}(s) - J_{\pi^t}(s)) + \frac{\epsilon}{c}$$

Step 3: Completing the proof: Lemma 3 clearly quantifies the relationship between an NPG update with step-size α_t and a soft policy iteration update with an additive error $\frac{\epsilon}{c}$. With this connection, we give a simple proof of geometric convergence for the natural policy gradient method. First, we claim that $J_{\pi^{t+1}}(s) \leq J_{\pi^t}(s)$. To see this, recall from Section 5, an NPG update with step-size $\alpha(s)$ can equivalently be written as,

$$\pi^{t+1}(s) = \arg\min_{a\in\Delta^{k-1}} \left[Q^t(s,a) + \frac{\eta_{\pi^t}(s)}{\alpha(s)} D_{\mathrm{KL}}(a\|\pi^t(s)) \right]$$

But staying at the current policy, i.e. taking $a = \pi^t(s)$. is feasible for the optimization problem above. Therefore,

$$T_{\pi^{t+1}} J_{\pi^t}(s) = Q^t(s, \pi^{t+1}(s)) \leq Q^t(s, \pi^t(s)) = J_{\pi^t}(s)$$

Using that $J_{\pi^{t+1}} = \lim_{n\to\infty} T^n_{\pi^{t+1}} J_{\pi^t}$ along with monotonicity of $T_{\pi^{t+1}}$ implies,

$$J_{\pi^t} \succeq T_{\pi^{t+1}} J_{\pi^t} \succeq T^2_{\pi^{t+1}} J_{\pi^t} \succeq \ldots \succeq J_{\pi^{t+1}}.$$

Thus, from Lemma 3, we get that

$$J_{\pi^{t+1}} - J_{\pi^t} \preceq T_{\pi^{t+1}} J_{\pi^t} - J_{\pi^t} \preceq \frac{1}{2} \cdot (TJ_{\pi^t} - J_{\pi^t}) + \frac{\epsilon}{c}.$$

Subtracting J^* from both sides and rearranging terms gives,

$$\begin{aligned} J_{\pi^{t+1}} - J^* &\preceq \frac{1}{2} J_{\pi^t} + \frac{1}{2} TJ_{\pi^t} - J^* + \frac{\epsilon}{c} \\ &= \frac{1}{2}(J_{\pi^t} - J^*) + \frac{1}{2}(TJ_{\pi^t} - J^*) + \frac{\epsilon}{c}. \end{aligned}$$

As the above inequality holds element wise, we use the contraction property of $T(\cdot)$ as shown in (4) to get

$$\|J_{\pi^{t+1}} - J^*\|_\infty \leq \left(\frac{1}{2} + \frac{\gamma}{2}\right) \|J_{\pi^t} - J^*\| + \frac{\epsilon}{c}.$$

Iterating over the above equation and rewriting $\left(\frac{1}{2} + \frac{\gamma}{2}\right) = \left(1 - \frac{1}{2}(1-\gamma)\right)$ gives us our desired result. □

7 Conclusion and Future Work

In this work, we use illuminating connections with policy iteration as shown in Bhandari and Russo [2019] to show how many variants of policy gradient algorithms with large step-sizes and true gradient evaluations converge geometrically fast for tabular MDPs. An interesting question for future work is whether these results can be extended to function approximation settings where the policy class might be restricted, for example in Agarwal et al. [2020]. Another interesting question is whether our results hold in settings where unbiased estimates of the value functions are obtained via sampling. Here some exciting progress has been recently made for the undiscounted (average cost setting) in [Abbasi-Yadkori et al., 2019, Hao et al., 2020] for ergodic MDPs, by leveraging connections to approximate policy iteration.

[6] The action gap of any action $i \in \{1,\ldots,k\}$ is the difference between Q-values when compared to the optimal action.

[7] This defintion of sub-optimal actions based on action gap threshold, ϵ/c, is essentially an artifact that we are taking gradient steps with finite step-sizes. As $\alpha_t(s) \to \infty \; \forall s \in \mathcal{S}$, an NPG update is exactly equal to a policy iteration update.

Acknowledgements

We thank anonymous reviewers for their valuable feedback. This work was done in part when JB was participating in the Theory of Reinforcement Learning program at the Simons Institute of the Theory of Computing. JB also thanks Garud Iyengar for his support throughout the PhD program at Columbia.

References

Yasin Abbasi-Yadkori, Peter Bartlett, Kush Bhatia, Nevena Lazic, Csaba Szepesvari, and Gellért Weisz. Politex: Regret bounds for policy iteration using expert prediction. In *International Conference on Machine Learning*, pages 3692–3702. PMLR, 2019.

Alekh Agarwal, Sham M Kakade, Jason D Lee, and Gaurav Mahajan. Optimality and approximation with policy gradient methods in markov decision processes. In Jacob Abernethy and Shivani Agarwal, editors, *Proceedings of Thirty Third Conference on Learning Theory*, volume 125, pages 64–66. PMLR, 2020.

Jonathan Baxter and Peter L Bartlett. Direct gradient-based reinforcement learning: I. gradient estimation algorithms. Technical report, Citeseer, 1999.

Amir Beck and Marc Teboulle. Mirror descent and nonlinear projected subgradient methods for convex optimization. *Operations Research Letters*, 31(3):167–175, 2003.

Dimitri P Bertsekas. *Dynamic programming and optimal control*. Athena scientific Belmont, MA, 1995.

Jalaj Bhandari and Daniel Russo. Global optimality guarantees for policy gradient methods. *arXiv preprint arXiv:1906.01786*, 2019.

Sébastien Bubeck et al. Convex optimization: Algorithms and complexity. *Foundations and Trends® in Machine Learning*, 8(3-4):231–357, 2015.

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.

John Duchi, Shai Shalev-Shwartz, Yoram Singer, and Tushar Chandra. Efficient projections onto the l 1-ball for learning in high dimensions. In *Proceedings of the 25th international conference on Machine learning*, pages 272–279, 2008.

Matthieu Geist, Bruno Scherrer, and Olivier Pietquin. A theory of regularized Markov decision processes. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97, pages 2160–2169. PMLR, 2019.

Botao Hao, Nevena Lazic, Yasin Abbasi-Yadkori, Pooria Joulani, and Csaba Szepesvari. Provably efficient adaptive approximate policy iteration. *arXiv preprint arXiv:2002.03069*, 2020.

Sham Kakade and John Langford. Approximately optimal approximate reinforcement learning. In *ICML*, volume 2, pages 267–274, 2002.

Sham M Kakade. A natural policy gradient. In *Advances in neural information processing systems*, pages 1531–1538, 2002.

Peter Marbach and John N Tsitsiklis. Simulation-based optimization of markov reward processes. *IEEE Transactions on Automatic Control*, 46(2):191–209, 2001.

Jincheng Mei, Chenjun Xiao, Csaba Szepesvari, and Dale Schuurmans. On the global convergence rates of softmax policy gradient methods. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119, pages 6820–6829. PMLR, 2020.

Gergely Neu, Anders Jonsson, and Vicenç Gómez. A unified view of entropy-regularized markov decision processes. *arXiv preprint arXiv:1705.07798*, 2017.

Martin L Puterman. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 2014.

Garvesh Raskutti and Sayan Mukherjee. The information geometry of mirror descent. *IEEE Transactions on Information Theory*, 61(3):1451–1457, 2015.

Bruno Scherrer and Matthieu Geist. Local policy search in a convex space and conservative policy iteration as boosted policy search. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 35–50. Springer, 2014.

John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. In *International Conference on Machine Learning*, pages 1889–1897, 2015.

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

Lior Shani, Yonathan Efroni, and Shie Mannor. Adaptive trust region policy optimization: Global convergence and faster rates for regularized mdps. In *Proceedings of the 37th International Conference on Machine Learning*, volume 34, pages 5668–5675. AAAI Press, 2020.

David Silver, Guy Lever, Nicolas Heess, Thomas Degris, Daan Wierstra, and Martin Riedmiller. Deterministic policy gradient algorithms. In *ICML*, 2014.

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

Richard S Sutton, David A McAllester, Satinder P Singh, and Yishay Mansour. Policy gradient methods for reinforcement learning with function approximation. In *Ad-*

vances in neural information processing systems, pages 1057–1063, 2000.

Sebastian B Thrun. Efficient exploration in reinforcement learning. Technical report, School of Computer Science, Carnegie Mellon University, 1992.

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

Appendix: On the Linear Convergence of Policy Gradient Methods for Finite MDPs

A Proof of supporting lemmas

We give proofs of Lemmas 2 and 3, which were excluded from the main text.

Lemma 2. *For any state s, $\frac{\pi^{t+1}(s,i)}{\pi^t(s,i)} \leq \frac{1}{2} \;\; \forall\, i \in O_t^-(s)$.*

Proof. The proof follows a simple argument. By definition, for any $i \in O_t^-(s)$:

$$\begin{aligned} \big(Q^t(s,i) - Q^t(s,1)\big) &\geq \frac{\epsilon}{c} \\ \Rightarrow\; \alpha_t(s)\,\big(Q^t(s,i) - Q^t(s,1)\big) &\geq \log\frac{2}{\pi^t(s,1)} \end{aligned}$$

which follows by the definition, $\alpha_t(s) \geq \frac{c}{\epsilon}\log\frac{2}{\pi^t(s,1)}$ which implies $\frac{\epsilon}{c} \geq \frac{1}{\alpha_t(s)}\log\frac{2}{\pi^t(s,1)}$. Rearranging, we get

$$\log\Big(\pi^t(s,1)e^{-\alpha_t(s)Q^t(s,1)}\Big) + \log\bigg(\frac{1}{2}\bigg) \geq -\alpha_t(s)Q^t(s,i)$$

Define, $Z_t = \Big(\sum_{j=1}^k \pi^t(s,j)e^{-\alpha_t(s)Q^t(s,j)}\Big)$. Then,

$$\log(Z_t) \geq \log\Big(\pi^t(s,1)e^{-\alpha_t(s)Q^t(s,1)}\Big)$$

which holds as all the terms in Z_t are positive, i.e. $\pi^t(s,j)e^{-\alpha_t(s)Q^t(s,j)} > 0 \;\; \forall\, j \in \{1,2,\ldots,k\}$, and $\log(\cdot)$ is a monotonic transformation. Rearranging, we get our desired result.

$$\begin{aligned} \log\left(\frac{Z_t}{2}\right) \geq \log\left(\frac{\pi^t(s,1)}{2}e^{-\alpha_t(s)Q^t(s,1)}\right) &\geq -\alpha_t(s)Q^t(s,i) \\ \Rightarrow\; \frac{\pi^{t+1}(s,i)}{\pi^t(s,i)} = \frac{1}{Z_t}e^{-\alpha_t(s)Q^t(s,i)} &\leq \frac{1}{2}. \end{aligned}$$

□

Lemma 3 (Progress quantification)**.** *Let $J_{\pi^t}(s)$ denote the cost-to-go function for policy π^t from any starting state $s \in \mathcal{S}$. Then,*

$$T_{\pi^{t+1}}J_{\pi^t}(s) - J_{\pi^t}(s) \leq \frac{1}{2}\cdot\big(TJ_{\pi^t}(s) - J_{\pi^t}(s)\big) + \frac{\epsilon}{c}$$

Proof. Fix any state $s \in \mathcal{S}$. Without loss of generality, we assume the following ordering on Q-values: $Q^t(s,1) < Q^t(s,2)\ldots < Q^t(s,k)$ which implies that the policy iteration update, π_t^+ puts the entire mass on action 1, which is the best action under the current policy π^t. That is, $\pi^t_+(s,1)=1$ and $\pi^t_+(s,i)=0\ \ \forall i\neq 1$. Consider,

$$
\begin{aligned}
T_{\pi^{t+1}}J_{\pi^t}(s)-TJ_{\pi^t}(s) &= \langle \pi^{t+1}(s,\cdot)-\pi^t_+(s,\cdot), Q^t(s,\cdot)\rangle \\
&= (\pi^{t+1}(s,1)-1)Q^t(s,1)+\sum_{j=2}^{k}\pi^{t+1}(s,j)Q^t(s,j) \\
&= -\sum_{j=2}^{k}\pi^{t+1}(s,j)Q^t(s,1)+\sum_{j=2}^{k}\pi^{t+1}(s,j)Q^t(s,j) \\
&= \sum_{j=2}^{k}\pi^{t+1}(s,j)\left(Q^t(s,j)-Q^t(s,1)\right) \\
&= \sum_{j\in\mathcal{O}_t^-}\pi^{t+1}(s,j)\left(Q^t(s,j)-Q^t(s,1)\right)+\sum_{j\in\mathcal{O}_t^+}\pi^{t+1}(s,j)\left(Q^t(s,j)-Q^t(s,1)\right) \\
&= \sum_{j\in\mathcal{O}_t^-}\frac{\pi^{t+1}(s,j)}{\pi^t(s,j)}\pi^t(s,j)\left(Q^t(s,j)-Q^t(s,1)\right)+\sum_{j\in\mathcal{O}_t^+}\pi^{t+1}(s,j)\underbrace{\left(Q^t(s,j)-Q^t(s,1)\right)}_{<\frac{\epsilon}{c}} \\
&\leq \frac{1}{2}\sum_{j\in\mathcal{O}_t^-}\pi^t(s,j)\left(Q^t(s,j)-Q^t(s,1)\right)+\frac{\epsilon}{c} \\
&\leq \frac{1}{2}\left(\sum_{j=2}^{k}\pi^t(s,j)(Q^t(s,j)-Q^t(s,1))\right)+\frac{\epsilon}{c} \\
&= \frac{1}{2}\left(\sum_{j=2}^{k}\pi^t(s,j)Q^t(s,j)-\sum_{j=2}^{k}\pi^t(s,j)Q^t(s,1)\right)+\frac{\epsilon}{c} \\
&= \frac{1}{2}\left(\left(\pi^t(s,1)-1\right)Q^t(s,1)+\sum_{j=2}^{k}\pi^t(s,j)Q^t(s,j)\right)+\frac{\epsilon}{c} \\
&= \frac{1}{2}\langle \pi^t(s,\cdot)-\pi^t_+(s,\cdot), Q^t(s,\cdot)\rangle+\frac{\epsilon}{c} \\
&= \frac{1}{2}\left(J_{\pi^t}(s)-TJ_{\pi^t}(s)\right)+\frac{\epsilon}{c}
\end{aligned}
\tag{13}
$$

where we used that $\frac{\pi^{t+1}(s,j)}{\pi^t(s,j)}\leq\frac{1}{2}\ \forall j\in\mathcal{O}_t^-(s)$ as shown above in Lemma 2 along with the fact that $(Q^t(s,j)-Q^t(s,1))\leq\frac{\epsilon}{c}\ \forall j\in\mathcal{O}_t^+(s)$, which follows by definition. Subtracting $J_{\pi^t}(s)$ from both sides in (13) and rearranging terms gives our desired result,

$$
T_{\pi^{t+1}}J_{\pi^t}(s)-J_{\pi^t}(s)\leq\frac{1}{2}\cdot\left(TJ_{\pi^t}(s)-J_{\pi^t}(s)\right)+\frac{\epsilon}{c}.
$$

□

B Details of MDP in Figure 1

We used the following two state three action MDP, $P\in\mathbb{R}^{|\mathcal{S}||\mathcal{A}|\times|\mathcal{S}|}$, $g\in\mathbb{R}^{|\mathcal{S}||\mathcal{A}|}$, $\gamma,\rho\in\mathbb{R}^{|\mathcal{S}|}$, to generate Figure 1.

$$
P=\begin{bmatrix}0.666066 & 0.333934\\ 0.662211 & 0.337789\\ 0.441947 & 0.558053\\ 0.391257 & 0.608743\\ 0.452186 & 0.547814\\ 0.035519 & 0.964481\end{bmatrix},\ g=\begin{bmatrix}0.079718\\ 0.629733\\ 0.717644\\ 0.673362\\ 0.762623\\ 0.541251\end{bmatrix},\ \gamma=0.9,\ \rho=\begin{bmatrix}0.168831\\ 0.831169\end{bmatrix}
$$

Policy π for the two states s_1 and s_2 was taken to be,

$$\pi(s_1) = \begin{bmatrix} 0.449416 \\ 0.251788 \\ 0.298796 \end{bmatrix}, \pi(s_2) = \begin{bmatrix} 0.318626 \\ 0.346284 \\ 0.335090 \end{bmatrix}.$$