A Unified View of Entropy-Regularized Markov Decision Processes

Gergely Neu
Dept. Information and Communication Technologies
Universitat Pompeu Fabra
gergely.neu@gmail.com

Vicenç Gómez
Dept. Information and Communication Technologies
Universitat Pompeu Fabra
vicen.gomez@upf.edu

Anders Jonsson
Dept. Information and Communication Technologies
Universitat Pompeu Fabra
anders.jonsson@upf.edu

Abstract

We propose a general framework for entropy-regularized average-reward reinforcement learning in Markov decision processes (MDPs). Our approach is based on extending the linear-programming formulation of policy optimization in MDPs to accommodate convex regularization functions. Our key result is showing that using the conditional entropy of the joint state-action distributions as regularization yields a dual optimization problem closely resembling the Bellman optimality equations. This result establishes a strong connection between recently successful entropy-regularized approximate dynamic programming and policy optimization methods, and in particular enables us to unify and extend several existing algorithms. We also show that several popular regularized reinforcement learning algorithms actually aim to optimize a non-stationary sequence of non-convex objectives, and thus they are not guaranteed to converge to a unique fixed point. Finally, we illustrate empirically the effects of using various regularization techniques on learning performance in a simple experimental setup.

1 Introduction

In recent years, the idea of *entropy regularization* has been used extensively in the reinforcement learning (RL) literature [37, 39]. Entropy-regularized variants of the classic Bellman equations and the entailing reinforcement-learning algorithms have been proposed to induce safe exploration [10] and risk-sensitive policies [12, 16, 33], or to model observed behavior of imperfect decision-makers [43, 42, 5], among others. Complementary to these approaches rooted in dynamic programming, another line of work proposes direct policy search methods attempting to optimize various entropy-regularized objectives [40, 27, 35, 19, 25], with the main goal of driving a safe online exploration procedure in an unknown Markov decision process. Notably, the state-of-the-art deep RL methods of Mnih et al. [19] and Schulman et al. [35] are both based on entropy-regularized policy search.

In this work, we connect these two seemingly disparate lines of work by showing a strong Lagrangian duality between the entropy-regularized Bellman equations and a certain regularized average-reward

31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA.

objective. Specifically, we extend the linear-programming formulation of the problem of optimization in MDPs to accommodate convex regularization functions, resulting in a convex program. We show that using the *conditional entropy* of the joint state-action distribution gives rise to a set of nonlinear equations resembling the Bellman optimality equations. Observing this duality enables us to establish a connection between regularized versions of value and policy iteration methods [30] and incremental convex optimization methods like Mirror Descent [21, 3] or Dual Averaging [41, 18, 11, 36]. This view enables us to formally analyze learning algorithms, to provide performance guarantees or point out design flaws. In particular, we show that the TRPO algorithm of Schulman et al. [35] actually converges to the optimal policy, while the A3C algorithm of Mnih et al. [19] may fail to converge to a unique fixed point even in very simple problems. To complement these results, we suggest an alternative objective that can be optimized consistently, avoiding the possibility of divergence.

Our theoretical framework significantly generalizes the previous work of Ziebart [42, Sec. 5.2] and Rawlik et al. [31], who show a similar Lagrangian duality between the Bellman equations and entropy maximization for a special class of *episodic* Markov decision processes where the time index within the episode is part of the state representation. Unlike these previous results, our theory also readily extends to discounted MDPs by replacing the stationary state-action distributions we consider by *discounted state-action occupancy measures*. For consistency, we will discuss each particular algorithm in their most natural average-reward version, noting that all conclusions remain valid in the simpler discounted and episodic settings.

2 Preliminaries on Markov decision processes

We consider a finite Markov decision process (MDP) $M = (\mathcal{X}, \mathcal{A}, P, r)$, where $\mathcal{X}$ is the finite state space, $\mathcal{A}$ is the finite action space, $P : \mathcal{X} \times \mathcal{A} \times \mathcal{X} \to [0, 1]$ is the transition function, with $P(y|x, a)$ denoting the probability of moving to state y from state x when taking action a, and $r : \mathcal{X} \times \mathcal{A} \to \mathbb{R}$ is the reward function mapping state-action pairs to rewards.

In each round t, the learner observes state $X_t \in \mathcal{X}$, selects action $A_t \in \mathcal{A}$, moves to the next state $X_{t+1} \sim P(\cdot|X_t, A_t)$, and obtains reward $r(X_t, A_t)$. The goal is to select actions as to maximize some notion of cumulative reward. In this paper we consider the *average-reward* criterion $\liminf_{T\to\infty} \mathbb{E}\left[\frac{1}{T}\sum_{t=1}^{T} r_t(X_t, A_t)\right]$. A *stationary state-feedback policy* (or *policy* for short) defines a probability distribution $\pi(\cdot|x)$ over the learner's actions in state x. MDP theory (see, e.g., Puterman [29]) stipulates that under mild conditions, the average-reward criterion can be maximized by stationary policies. Throughout the paper, we make the following mild assumption about the MDP:

Assumption 1. *The MDP M is* unichain*: All stationary policies π induce a unique stationary distribution ν_π over the state space satisfying $\nu_\pi(y) = \sum_{x,a} P(y|x, a)\pi(a|x)\nu_\pi(x)$ for all $y \in \mathcal{X}$.*

In particular, this assumption is satisfied if all policies induce an irreducible and aperiodic Markov chain [29]. For ease of exposition in this section, we also make the following simplifying assumption:

Assumption 2. *The MDP M admits a single recurrent class: All stationary policies π induce stationary distributions strictly supported on the same set $\mathcal{X}' \subseteq \mathcal{X}$.*

In general, this assumption is very restrictive in that it does not allow policies to cover different parts of the state space. We stress that our results in the later sections *do not* require this assumption to hold. With the above assumptions in mind, we can define the average reward of any policy π as

$$\rho(\pi) = \lim_{T\to\infty} \mathbb{E}\left[\frac{1}{T}\sum_{t=1}^{T} r_t(X_t, A_t)\right],$$

where $A_t \sim \pi(\cdot|X_t)$ in each round t and the existence of the limit is ensured by Assumption 1. Furthermore, the average reward of any policy π can be simply written as $\rho(\pi) = \sum_{x,a} \nu_\pi(x)\pi(a|x)r(x, a)$, which is a linear function of the stationary state-action distribution $\mu_\pi = \nu_\pi\pi$. This suggests that finding the optimal policy can be equivalently written as a linear program (LP) where the decision variable is the stationary state-action distribution. Defining the set of all feasible stationary distributions as

$$\Delta = \left\{\mu \in \Delta(\mathcal{X} \times \mathcal{A}) : \sum_b \mu(y, b) = \sum_{x,a} P(y|x, a)\mu(x, a) \quad (\forall y)\right\}, \tag{1}$$

the problem of maximizing the average reward can be written as

$$\mu^* = \arg\max_{\mu \in \Delta} \rho(\mu). \tag{2}$$

This linear program is well studied in the MDP literature (see, e.g., 29, Section 8.8), although most commonly as the dual of the linear program equivalent to the solution of the

$$\min_{\rho \in \mathbb{R}} \quad \rho \tag{3}$$

$$\text{subject to} \quad \rho + V(x) - \sum_y P(y|x,a)V(y) \geq r(x,a), \qquad \forall (x,a). \tag{4}$$

Here, the dual variables V are commonly referred to as the *value functions*. By strong LP duality and our Assumption 1, the solution to this LP equals the optimal average reward ρ^* and the dual variables V^* at the optimum are the solution to the *average-reward Bellman optimality equations*

$$V^*(x) = \max_a \left(r(x,a) - \rho^* + \sum_y P(y|x,a)V^*(x) \right), \qquad (\forall x)\,. \tag{5}$$

3 Regularized MDPs: A convex-optimization view

Inspired by the LP formulation of the average-reward optimization problem (2), we now define a regularized optimization objective—a framework that will lead us to our main results. Our results in this section only require the mild Assumption 1. Our regularized optimization problem takes the form

$$\max_{\mu \in \Delta} \ \widetilde{\rho}_\eta(\mu) = \max_{\mu \in \Delta} \left\{ \sum_{x,a} \mu(x,a) r(x,a) - \frac{1}{\eta} R(\mu) \right\}, \tag{6}$$

where $R(\mu) : \mathbb{R}^{\mathcal{X} \times \mathcal{A}} \to \mathbb{R}$ is a convex regularization function and $\eta > 0$ is a *learning rate* that trades off the original objective and regularization. Note that $\eta = \infty$ recovers the unregularized objective. Unlike previous work on LP formulations for MDPs, we find it useful to regard (6) as the *primal*.

We focus on two families of regularization functions: the *negative Shannon entropy* of $(X, A) \sim \mu$,

$$R_S(\mu) = \sum_{x,a} \mu(x,a) \log \mu(x,a), \tag{7}$$

and the *negative conditional entropy* of $(X, A) \sim \mu$,

$$R_C(\mu) = \sum_{x,a} \mu(x,a) \log \frac{\mu(x,a)}{\sum_b \mu(x,b)} = \sum_{x,a} \nu_\mu(x) \pi_\mu(a|x) \log \pi_\mu(a|x). \tag{8}$$

In what follows, we refer to these functions as the relative entropy and the conditional entropy. We also make use of the Bregman divergences induced by R_S and R_C which take the respective forms

$$D_S\left(\mu \| \mu'\right) = \sum_{x,a} \mu(x,a) \log \frac{\mu(x,a)}{\mu'(x,a)} \qquad \text{and} \qquad D_C\left(\mu \| \mu'\right) = \sum_{x,a} \mu(x,a) \log \frac{\pi_\mu(a|x)}{\pi_{\mu'}(a|x)}.$$

While the form of D_S is standard (it is the relative entropy between two state-action distributions), the fact that D_C is the Bregman divergence of R_C (or even that R_C is convex) is not immediately obvious[1]. The following proposition asserts this statement, which we prove in Appendix A.1. The only work we are aware of that establishes a comparable result is that of Neu and Gómez [22].

Proposition 1. *The Bregman divergence corresponding to the conditional entropy R_C is D_C. Furthermore, D_C is nonnegative on Δ, implying that R_C is convex and D_C is convex in its first argument.*

We proceed to derive the dual functions and optimal solutions to (6) for our two choices of regularization functions. Without loss of generality, we assume that the reference policy $\pi_{\mu'}$ has full support, which implies that the corresponding stationary distribution μ' is strictly positive on the recurrent set $\mathcal{X}'$. We only provide the derivations for the Bregman divergences; the calculations are analogous for R_S and R_C. Both of these solutions will be expressed with the help of dual variables $V : \mathbb{R}^{\mathcal{X}} \to \mathbb{R}$ which are useful to think about as *value functions*, as in the case of the LP formulation (2). We also define the corresponding *advantage functions* $A(x,a) = r(x,a) + \sum_y P(y|x,a)V(y) - V(x)$.

[1] In the special case of loop-free episodic environments, the convexity of R_C is straightforward [14, 42, 31].

3.1 Relative entropy

The choice $R = D_S(\cdot\|\mu')$ has been studied before by Peters et al. [27] and Zimin and Neu [44]; we defer the proofs to Appendix A.3. The optimal state-action distribution for a given value of η is

$$\mu^*_\eta(x,a) \propto \mu'(x,a)e^{\eta A^*_\eta(x,a)}, \tag{9}$$

where A^*_η is the advantage function for the optimal dual variables V^*_η. The dual function is

$$g(V) = \frac{1}{\eta}\log\sum_{x,a}\mu'(x,a)e^{\eta A(x,a)}, \tag{10}$$

that now needs to be minimized on $\mathbb{R}^{\mathcal{X}}$ with no constraints in order to obtain V^*_η. By strong duality, g is convex in V and takes the value $\widetilde{\rho}^*_\eta = \max_{\mu\in\Delta}\widetilde{\rho}_\eta(\mu)$ at its optimum.

3.2 Conditional entropy

The choice $R = D_C(\cdot\|\mu')$ leads to our main contributions. Similar to above, the optimal policy is

$$\pi^*_\eta(a|x) \propto \pi_{\mu'}(a|x)e^{\eta A^*_\eta(x,a)}. \tag{11}$$

In this case, the dual problem closely resembles the average-reward Bellman optimality equations (5):

Proposition 2. *The dual of the optimization problem* (6) *when* $R = D_C(\cdot\|\mu')$ *is given by*

$$\begin{aligned} &\min_{\lambda\in\mathbb{R}} && \lambda \\ &\textit{subject to} && V(x) = \frac{1}{\eta}\log\sum_a \pi_{\mu'}(a|x)\exp\left(\eta\left(r(x,a) - \lambda + \sum_y P(y|x,a)V(y)\right)\right),\ (\forall x)\,. \end{aligned}$$

We defer the proofs to Appendix A.4. Using strong duality, the optimum of the above problem is $\widetilde{\rho}^*_\eta$, which implies that the optimal dual variables V^*_η are given as a solution to the system of equations

$$V^*_\eta(x) = \frac{1}{\eta}\log\sum_a \pi_{\mu'}(a|x)\exp\left(\eta\left(r(x,a) - \widetilde{\rho}^*_\eta + \sum_y P(y|x,a)V^*_\eta(y)\right)\right), \quad (\forall x)\,. \tag{12}$$

By analogy with the Bellman optimality equations (5), we call this the *regularized average-reward Bellman optimality equations*. Since $\widetilde{\rho}^*_\eta$ is guaranteed to be finite (because it is the maximum of a bounded function on a compact domain), the solution to the above optimization problem is well-defined, bounded, and unique up to a constant shift (as in the case of the LP dual variables). Again, we can make the solution unique by imposing the constraint that the expected value should equal 0.

The notion of regularized Bellman optimality equations naturally leads to a definition of a regularized counterpart of the Bellman equations defined for arbitrary policies, which enables the derivation of regularized counterparts to several well-studied reinforcement learning algorithms. We provide the formal definitions to the resulting Bellman equations and Dynamic Programming operators in Appendix B. In this section, we highlight one important algorithmic tool resulting from this formalism: a regularized counterpart of the *policy gradient theorem* [38]. This statement provides a closed-form expression for the gradient of the regularized objective $\widetilde{\rho}_\eta$ when the policy π_θ is parameterized by a vector $\theta \in \mathbb{R}^d$:

Lemma 1. *Define* A^π_η *as the* regularized advantage function *defined for each policy* π *as*

$$A^\pi_\eta(x,a) = r(x,a) - \frac{1}{\eta}\log\frac{\pi(a|x)}{\pi'(a|x)} - \widetilde{\rho}_\eta(\pi) + \sum_y P(y|x,a)V^\pi_\eta(y) - V^\pi_\eta(x),$$

where V^π_η *is the regularized value function corresponding to policy* π *with baseline* π'*. Assuming that* $\frac{\partial\pi_\theta(a|x)}{\partial\theta_i}/\pi_\theta(a|x) > 0$ *for all* θ_i, x, a*, the gradient of* $\widetilde{\rho}_\eta$ *exists and satisfies*

$$\nabla\widetilde{\rho}_\eta(\pi_\theta) = \sum_{x,a}\mu_{\pi_\theta}(x,a)\nabla\log\pi_\theta(a|x)A^{\pi_\theta}_\eta(x,a).$$

4 Algorithms

In this section we provide a formal framework for deriving and analyzing reinforcement learning algorithms based on the insights of the previous section. For the derivation of the algorithms, we will assume that the MDP M is fully known. While this assumption may seem restrictive at first, we will see that these derivations will often result in closed-form update rules that can be approximated by standard RL methods (e.g., estimating value functions). We will study a generic sequential optimization framework where a sequence of policies π_k are computed iteratively. Inspired by the online convex optimization literature (see, e.g., 36, 11) and by our convex-optimization formulation, we study two families of algorithms: Mirror Descent and Dual Averaging (also known as Follow-the-Regularized-Leader).

4.1 Iterative policy optimization by Mirror Descent

A direct application of the Mirror Descent algorithm [21, 3, 17, 32] to our case is defined as

$$\mu_{k+1} = \operatorname*{arg\,max}_{\mu \in \Delta} \left\{ \rho(\mu) - \frac{1}{\eta} D_R\left(\mu \| \mu_k\right) \right\}, \tag{13}$$

where D_R is the Bregman divergence associated with the convex regularization function R. We now proceed to show how various learning algorithms can be recovered from this formulation.

4.1.1 Mirror Descent with the relative entropy

We first remark that the Relative Entropy Policy Search (REPS) algorithm of Peters et al. [27] can be formulated as an instance of Mirror Descent with the Bregman divergence D_S. This is easily seen by comparing the form of the update rule (13) with the problem formulation of Peters et al. [27, pp. 2], with the slight difference that our regularization is additive and theirs is enforced as a constraint. It is easy to see that this only amounts to a change in learning rate. This connection is not new: it has been first shown by Zimin and Neu [44][2], and has been recently rediscovered by Montgomery and Levine [20]. Independently of each other, Zimin and Neu [44] and Dick et al. [8] both show that Mirror Descent achieves near-optimal regret guarantees in an online learning setup where the transition function is known, but the reward function is allowed to change arbitrarily between decision rounds. This implies that REPS duly converges to the optimal policy in our setup.

4.1.2 Mirror Descent with the conditional entropy

We next show that the Dynamic Policy Programming (DPP) algorithm of Azar et al. [2] and the Trust-Region Policy Optimization (TRPO) algorithm of Schulman et al. [35] are both approximate variants of Mirror Descent with the Bregman divergence D_C. To see this, note that a full Mirror Descent update requires computing the optimal value function V_η^* for the baseline μ_k, e.g. by regularized value iteration or regularized policy iteration (see Appendix B). Since a full update for V_η^* is expensive, DPP and TRPO provide two ways to approximate it. We remark that the algorithm of Rawlik et al. [31] can also be viewed as an instance of Mirror Descent for the finite-horizon episodic setting, in which the exact update can be computed efficiently by dynamic programming.

Dynamic Policy Programming. We first claim that each iteration of DPP is a *single regularized value iteration step*: Starting from the previous value function V_k, it extracts the greedy policy $\pi_{k+1} = G_\eta^{\pi_k}[V_k]$ and applies the Bellman optimality operator $T_\eta^{*|\pi_k}$ to obtain $V_{k+1} = T_\eta^{*|\pi_k}[V_k]$. This follows from comparing the form of DPP presented in Appendix A of Azar et al. [2]: their update rules (19) and (20) precisely match the discounted analogue of our expressions (27) in Appendix B with $\pi' = \pi_k$. The convergence guarantees proved by Azar et al. [2] demonstrate the soundness of this approximate update.

Trust-Region Policy Optimization. Second, we claim that each iteration of TRPO is a *single policy iteration step*: TRPO first fully evaluates the policy π_k to compute its *unregularized* value function $V_k = V_\infty^{\pi_k}$ and then extracts the regularized greedy policy $\pi_{k+1} = G_\eta^{\pi_k}[V_k]$ with π_k as a

[2] Although they primarily referred to Mirror Descent as the "Proximal Point Algorithm" following [32, 17].

baseline. This can be seen by inspecting the TRPO update[3] that takes the form

$$\pi_{k+1} = \arg\max_{\pi} \left\{ \sum_x \nu_{\pi_k}(x) \sum_a \pi(a|x) \left(A_\infty^{\pi_k}(x,a) - \frac{1}{\eta} \log \frac{\pi(a|x)}{\pi_k(a|x)} \right) \right\}.$$

This objective approximates Mirror Descent by ignoring the effect of changing the policy on the state distribution, and in particular uses the divergence

$$D_{\text{TRPO}}(\mu \| \mu_k) = \sum_x \nu_{\mu_k}(x) \sum_a \pi_\mu(a|x) \log \frac{\pi_\mu(a|x)}{\pi_{\mu_k}(a|x)}$$

instead of D_C, the main difference being that the new state distribution ν_μ is replaced here by the old state distribution ν_{μ_k}. Surprisingly, using our formalism, this update can be expressed in closed form as

$$\pi_{k+1}(a|x) \propto \pi_k(a|x) e^{\eta A_\infty^{\pi_k}(x,a)}.$$

We present the detailed derivations in Appendix B.3. A particularly interesting consequence of this result is that TRPO is *completely equivalent* to the MDP-E algorithm of Even-Dar et al. [9] (see also [23, 24]), which is known to minimize regret in an online setting, thus implying that TRPO also converges to the optimal policy in the stationary setting. This guarantee is much stronger than the ones provided by Schulman et al. [35], who only claim that TRPO produces a monotonically improving sequence of policies (which may still converge to a suboptimal policy).

4.2 Iterative policy optimization by Dual Averaging

We next study algorithms arising from the Dual Averaging scheme [41, 18], commonly known as Follow-the-Regularized-Leader in online learning [36, 11]. This algorithm is defined by the iteration

$$\mu_{k+1} = \arg\max_{\mu \in \Delta} \left\{ \rho(\mu) - \frac{1}{\eta_k} R(\mu) \right\}, \tag{14}$$

where η_k is usually an increasing sequence to ensure convergence in the limit. We are unaware of any pure instance of dual averaging using relative entropy, and only discuss conditional entropy below.

4.2.1 Dual Averaging with the conditional entropy

Just as for Mirror Descent, a full update (14) requires computing the optimal value function V_η^*. Various approximations of this update have been long studied in the RL literature—see, e.g., [15] (with additional discussion by [1]), [26, 33, 28, 10]. In this section, we focus on the state-of-the-art algorithms of Mnih et al. [19] and O'Donoghue et al. [25] that were originally derived from an optimization formulation resembling our Equation (6). Our main insight is that this algorithm can be adjusted to have a dynamic-programming interpretation and a convergence guarantee.

Entropy-regularized policy gradients. The A3C algorithm of Mnih et al. [19] aims to maximize

$$\rho(\pi) - \frac{1}{\eta_k} \sum_x \nu_{\pi_k}(x) \sum_a \pi(a|x) \log \pi(a|x) \tag{15}$$

by taking policy gradient steps. Similarly to TRPO, A3C uses a regularization term ignoring the effect of changing the policy on the state distribution:

$$R_{\text{A3C}}(\mu) = \sum_x \nu_{\mu_k}(x) \sum_a \pi_\mu(a|x) \log \pi_\mu(a|x), \tag{16}$$

where the difference from the conditional entropy regularizer R_C is ν_μ is replaced by the previous state distribution ν_{μ_k}. Thus, A3C is attempting to optimize a non-stationary sequence of objective functions by gradient descent. Unlike TRPO, however, this change leads to major obstacles in analyzing the algorithm: by combining the recent results of O'Donoghue et al. [25] and Asadi and Littman [1], we can formally show that A3C does not converge to a unique optimum, and as a result

[3] As in the case of REPS, we discuss here the additive-regularization version of the algorithm. The entropy-constrained update actually implemented by Schulman et al. [35] only differs in the learning rate.

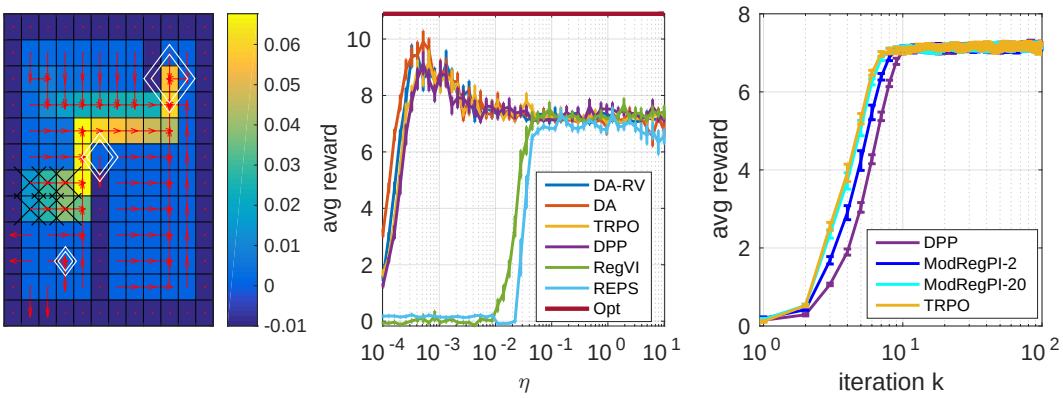

Figure 1: **(left)**: the MDP used for evaluation. Reward is -0.1 at the walls and $5, 50, 200$ at the diamonds. The optimal policy is indicated by red arrows. The cell colors correspond to the stationary state distribution for open locations. **(center)**: Average reward as a function of the learning rate η for all algorithms (see text for details). Number of iterations N and samples per iteration S are $N = S = 500$. Results are taken over 20 random runs per value of η. **(right)**: Performance of DPP, TRPO and two versions of modified regularized Policy Iteration for a fixed $\eta \approx 0.1$.

fails to retrieve the optimal policy. In particular, O'Donoghue et al. [25] show that the fixed points of the A3C objective correspond to the fixed points of a particular softmax value iteration scheme defined as

$$\pi_{k+1}(a|x) \propto e^{\eta_k A_\infty^{\pi_k}}, \tag{17}$$

which in turn is shown to have multiple fixed points by Asadi and Littman [1]. In Section 5.2, we empirically confirm that, already on a very simple example, A3C may converge to various fixed points depending on the initialization and random noise in the updates. We note that such problems are avoided by TRPO since the sum of the TRPO objectives is a sensible optimization objective [9, Theorem 4.1], whereas there is no such clear interpretation for the objective optimized by A3C.

To overcome these issues, we advocate for directly optimizing the objective (14) instead of (16) via gradient descent. Due to the fact that (14) is convex in μ and to standard results regarding dual averaging [18], this scheme is guaranteed to converge to the optimal policy. Estimating the gradients can be done analogously as for the unregularized objective, by our Lemma 1.

5 Experiments

In this section we illustrate the behavior of the algorithms from Section 4 in two sets of experiments.

5.1 Model-based reinforcement learning

In the first set of experiments we illustrate the interplay of regularization and model-estimation error in a simple grid MDP (Fig. 1, left). The agent has four actions (up, down, left and right) that succeed with probability 0.9; in case of failure, the agent stays put or moves to a random adjacent location. Hitting a wall results in negative reward, while reaching a white diamond location results in positive reward proportional to the distance from the starting locations (marked with 'X' in the figure). In either case the agent returns to a starting location. The optimal policy is to reach the top-right reward, while ignoring intermediate rewards (sometimes it is even better to hit a wall, e.g. at the bottom-left).

We consider an iterative setup where in each episode $k = 1, 2, \ldots, N$, we execute a policy π_k, observe the sample transitions and update the estimated model via maximum likelihood. We fix the number of iterations N and samples per iteration S and analyze the average reward of the final policy as a function of η. We compare the following algorithms: regularized Value Iteration with a fixed reference uniform policy and fixed η (RegVI); several variants of approximate Mirror Descent, including DPP and TRPO (Section 4.1); and two Dual Averaging methods (DA and DA-RV). DA corresponds to the update rule (17), which is not guaranteed to lead to an optimal policy, and DA-RV

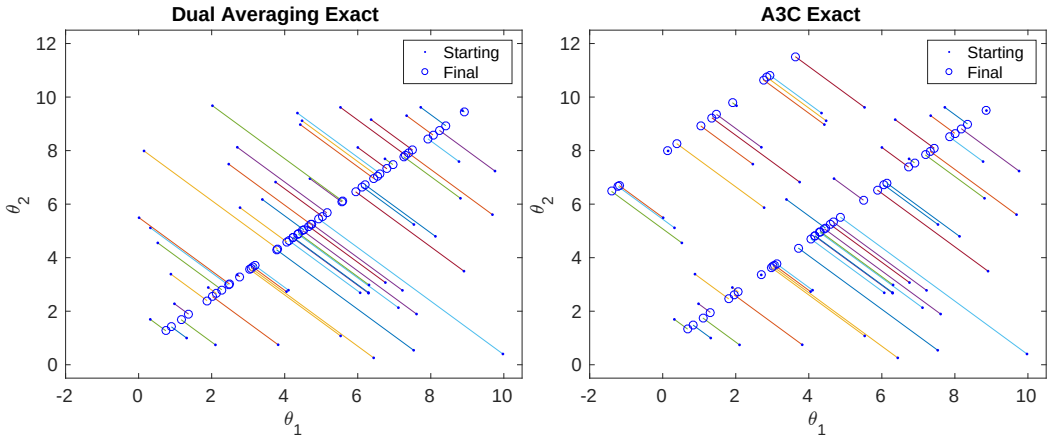

Figure 2: Comparison between Dual Averaging and A3C in parameter space of $\pi_\theta(a)$ for different random initializations and full policy evaluation (exact). Dots indicate starting condition whereas circles indicate converged values. **(left)** The Dual Averaging update always converges to the same policy, which is represented by the single diagonal ridge. **(right)** The A3C update converges to two different minima, corresponding to the policies of Figure 3 (right). We use $\alpha = 0.1$ and $\eta = 8$.

corresponds to the iteration (14), which has convergence guarantees (Section 4.2). For both variants, we use a linear annealing schedule $\eta_k = \eta \cdot k$.

Fig. 1 (center) shows results as a function of η. The maximum reward is depicted in brown at the top. For very small η (strong regularization), all algorithms perform poorly and do not even reach the intermediate reward. In contrast, for very large η, they converge prematurely to the greedy policy that exploits the intermediate reward. Typically, for an intermediate value of η, the algorithms occasionally discover the optimal path and exploit it. Note that this is not the case for RegVI, which never obtains the optimal policy. This shows that using both a fixed value of η and a fixed reference policy is a bad choice. In this MDP, we observe that the performance of both Dual Averaging methods (DA and DA-RV) is very similar, and in general slightly better than the approximate Mirror Descent variants.

We also show an interesting relationship between the Mirror Descent approximations. Our analysis in Section 4.1.2 suggests an entire array of algorithms lying between DPP and TRPO, just as Modified Policy Iteration lies between Value Iteration and Policy Iteration [30, 34]. Fig. 1 (right) illustrates this idea, showing the convergence of DPP and TRPO for a fixed value of η. TRPO tends to converge faster than DPP to a locally optimal policy, since DPP uses a single value update per iteration. Using more value updates leads to a modified regularized Policy Iteration algorithm (we call it ModRegPI-2 and ModRegPI-20, for 2 and 20 updates, respectively) that interpolates between DPP and TRPO.

5.2 Convergence of regularized policy gradient methods

In a second set of experiments we show empirically that gradient descent applied to our proposed objective, Equation (14), does not lead to problematic behavior, whereas the A3C algorithm of Mnih et al. [19], which follows the gradients of Equation (16), may converge to different solutions even without the presence of approximation errors. Directly inspired by the counterexample given by Asadi and Littman [1], we study here a simple MDP with one non-terminal state and two actions, and show that just like softmax value iteration, A3C also converges to one of multiple fixed points. Due to space constraints, the particular MDP is described in Appendix C.

We compare Dual Averaging and A3C on this MDP using a parametrized Boltzmann policy $\pi_\theta(a) \propto \exp(-\theta^\top f(a))$, where $f(a)$ is a two-element vector of indicator features corresponding to the two actions in state s_1, i.e., $f_i(a_j) = 1$ if $i = j$ and zero otherwise. For Dual Averaging we use the regularizer $R = D_C(\cdot\|\mu')$, where μ' is the fixed uniform policy that selects a_1 and a_2 with equal probability. We run both algorithms using the same settings for different initial random conditions. Figure 2 shows that Dual Averaging always converges to the same policy, represented by the single diagonal ridge. In contrast, the A3C update converges to two different minima, one being a deterministic policy. More details about the experiments appear in Appendix C.

We also analyze the behavior of both algorithms in the presence of approximation errors. We use *the same initial condition* and replace the exact policy evaluation step by a Monte Carlo approximation that records the cumulative reward of each episode (until reaching the terminal state s_2). In this case, while Dual Averaging converges closer to the previous optimal policy, with small differences caused by the noisy samples, A3C may still converge to either of the two fixed points. Therefore, for A3C, only the approximation error from the Monte Carlo samples can entirely determine which is the optimal policy found by the algorithm. The details of these experiments also appear in Appendix C.

References

[1] K. Asadi and M. L. Littman. A new softmax operator for reinforcement learning. *CoRR*, abs/1612.05628, 2016.

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

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

[4] D. P. Bertsekas. *Dynamic Programming and Optimal Control*, volume 2. Athena Scientific, Belmont, MA, 3 edition, 2007.

[5] D. A. Braun, P. A. Ortega, E. Theodorou, and S. Schaal. Path integral control and bounded rationality. In *Adaptive Dynamic Programming And Reinforcement Learning (ADPRL), 2011 IEEE Symposium on*, pages 202–209. IEEE, 2011.

[6] A. N. Burnetas and M. N. Katehakis. Optimal adaptive policies for Markov Decision Processes. 22(1):222–255, 1997.

[7] X.-R. Cao. *Stochastic Learning and Optimization: A Sensitivity-Based Approach*. Springer, New York, 2007.

[8] T. Dick, A. György, and Cs. Szepesvári. Online learning in markov decision processes with changing cost sequences. In *ICML 2014*, 2014.

[9] E. Even-Dar, S. M. Kakade, and Y. Mansour. Online Markov decision processes. *Mathematics of Operations Research*, 34(3):726–736, 2009.

[10] R. Fox, A. Pakman, and N. Tishby. Taming the noise in reinforcement learning via soft updates. In *Proceedings of the Thirty-Second Conference on Uncertainty in Artificial Intelligence*, 2016.

[11] E. Hazan. Introduction to online convex optimization. *Foundations and Trends® in Optimization*, 2(3-4):157–325, 2016.

[12] R. A. Howard and J. E. Matheson. Risk-sensitive Markov decision processes. *Management science*, 18(7):356–369, 1972.

[13] S. M. Kakade and J. Langford. Approximately optimal approximate reinforcement learning. In *Proceedings of the Nineteenth International Conference on Machine Learning (ICML 2002)*, pages 267–274, 2002.

[14] J. Lafferty, A. McCallum, and F. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *ICML 2001*, pages 282–289, 2001. URL `citeseer.ist.psu.edu/lafferty01conditional.html`.

[15] M. Littman and Cs. Szepesvári. A Generalized Reinforcement Learning Model: Convergence and applications. In *Int. Conf. on Machine Learning*, pages 310–318, 1996.

[16] S. I. Marcus, E. Fernández-Gaucherand, D. Hernández-Hernandez, S. Coraluppi, and P. Fard. Risk sensitive Markov decision processes. In *Systems and control in the twenty-first century*, pages 263–279. Springer, 1997.

[17] B. Martinet. Perturbation des méthodes d'optimisation. applications. *ESAIM: Mathematical Modelling and Numerical Analysis - Modélisation Mathématique et Analyse Numérique*, 12(2): 153–171, 1978. URL `http://eudml.org/doc/193317`.

[18] H. B. McMahan. A survey of algorithms and analysis for adaptive online learning. *arXiv preprint arXiv:1403.3465*, 2014.

[19] V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. Lillicrap, T. Harley, D. Silver, and K. Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In *International Conference on Machine Learning*, pages 1928–1937, 2016.

[20] W. H. Montgomery and S. Levine. Guided policy search via approximate mirror descent. In *NIPS-29*, pages 4008–4016, 2016.

[21] A. Nemirovski and D. Yudin. *Problem Complexity and Method Efficiency in Optimization*. Wiley Interscience, 1983.

[22] G. Neu and V. Gómez. Fast rates for online learning in Linearly Solvable Markov Decision Processes. In *Proceedings of the 2017 Conference on Learning Theory (COLT)*, volume 65, pages 1567–1588, 2017.

[23] G. Neu, A. György, and Cs. Szepesvári. The online loop-free stochastic shortest-path problem. In *Proceedings of the 23rd Annual Conference on Learning Theory (COLT)*, pages 231–243, 2010.

[24] G. Neu, A. György, Cs. Szepesvári, and A. Antos. Online Markov decision processes under bandit feedback. *IEEE Transactions on Automatic Control*, 59:676–691, 2014.

[25] B. O'Donoghue, R. Munos, K. Kavukcuoglu, and V. Mnih. PGQ: Combining policy gradient and Q-learning. In *5th International Conference on Learning Representations*, 2017.

[26] T. J. Perkins and D. Precup. A convergent form of approximate policy iteration. In *Advances in Neural Information Processing Systems*, pages 1595–1602, 2002.

[27] J. Peters, K. Mülling, and Y. Altun. Relative entropy policy search. In *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence*, pages 1607–1612, 2010.

[28] M. Petrik and D. Subramanian. An approximate solution method for large risk-averse Markov decision processes. In *Proceedings of the Twenty-Eighth Conference on Uncertainty in Artificial Intelligence*, pages 805–814, 2012.

[29] M. L. Puterman. *Markov Decision Processes*. Wiley, 1994. ISBN 978-0471727828.

[30] M. L. Puterman and M. C. Shin. Modified policy iteration algorithms for discounted Markov decision processes. *Management Science*, 24:1127–1137, 1978.

[31] K. Rawlik, M. Toussaint, and S. Vijayakumar. On stochastic optimal control and reinforcement learning by approximate inference. In *Proceedings of Robotics: Science and Systems VIII*, 2012.

[32] R. T. Rockafellar. Monotone Operators and the Proximal Point Algorithm. *SIAM Journal on Control and Optimization*, 14(5):877–898, 1976.

[33] A. Ruszczyński. Risk-averse dynamic programming for Markov decision processes. *Mathematical programming*, 125(2):235–261, 2010.

[34] B. Scherrer, V. Gabillon, M. Ghavamzadeh, and M. Geist. Approximate modified policy iteration. In *ICML 2012*, pages 1207–1214, 2012.

[35] J. Schulman, S. Levine, P. Abbeel, M. Jordan, and P. Moritz. Trust region policy optimization. In *Proceedings of the 32nd International Conference on Machine Learning (ICML-15)*, pages 1889–1897, 2015.

[36] S. Shalev-Shwartz. Online learning and online convex optimization. *Foundations and Trends in Machine Learning*, 4(2):107–194, 2012.

[37] R. Sutton and A. Barto. *Reinforcement Learning: An Introduction*. MIT Press, 1998.

[38] R. S. Sutton, D. A. McAllester, S. P. Singh, and Y. Mansour. Policy gradient methods for reinforcement learning with function approximation. In *NIPS-12*, pages 1057–1063, 1999.

[39] Cs. Szepesvári. *Algorithms for Reinforcement Learning*. Synthesis Lectures on Artificial Intelligence and Machine Learning. Morgan & Claypool Publishers, 2010.

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

[41] L. Xiao. Dual averaging methods for regularized stochastic learning and online optimization. *Journal of Machine Learning Research*, 11(Oct):2543–2596, 2010.

[42] B. D. Ziebart. *Modeling Purposeful Adaptive Behavior with the Principle of Maximum Causal Entropy*. PhD thesis, Carnegie Mellon University, 2010.

[43] B. D. Ziebart, J. A. Bagnell, and A. K. Dey. Modeling interaction via the principle of maximum causal entropy. In *International Conference on Machine Learning (ICML)*, pages 1247–1254, 2010.

[44] A. Zimin and G. Neu. Online learning in episodic Markovian decision processes by relative entropy policy search. In *NIPS-26*, pages 1583–1591, 2013.

A Complementary Technical Results

A.1 Convexity of the negative conditional entropy

Let us consider the joint state-action distribution μ on the finite set $\mathcal{X} \times \mathcal{A}$. We denote $\nu_\mu(x) = \sum_a \mu(x,a)$ and $\pi_\mu(a|x) = \mu(x,a)/\nu_\mu(x)$ for all x, a. We study the negative conditional entropy of $(X, A) \sim \mu$ as a function of μ:

$$R_C(\mu) = \sum_{x,a} \mu(x,a) \log \frac{\mu(x,a)}{\sum_b \mu(x,b)} = \sum_{x,a} \mu(x,a) \log \frac{\mu(x,a)}{\nu_\mu(x)}.$$

We will study the Bregman divergence D_{R_C} corresponding to R_C:

$$D_{R_C}(\mu \| \mu') = R_C(\mu) - R_C(\mu') - \nabla R_C(\mu')^\top (\mu - \mu'),$$

where the inner product between two vectors $v, w \in \mathbb{R}^{\mathcal{X}\times\mathcal{A}}$ is $w^\top v = \sum_{x,a} v(x,a) w(x,a)$. Our aim is to show that D_{R_C} is nonnegative, which will imply the convexity of R_C.

We begin by computing the partial derivative of $R_C(\mu)$ with respect to $\mu(x,a)$:

$$\frac{\partial R_C(\mu)}{\partial \mu(x,a)} = \log \frac{\mu(x,a)}{\nu_\mu(x)} + 1 - \sum_b \frac{\mu(x,b)}{\nu_\mu(x)} = \log \frac{\mu(x,a)}{\nu_\mu(x)},$$

where we used the fact that $\partial \nu_\mu(x)/\partial \mu(x,a) = 1$ for all a. With this expression, we have

$$\begin{aligned} R_C(\mu') + \nabla R_C(\mu')^\top (\mu - \mu') &= \sum_{x,a} \mu'(x,a) \log \frac{\mu'(x,a)}{\nu_{\mu'}(x)} + \sum_{x,a} (\mu(x,a) - \mu'(x,a)) \log \frac{\mu'(x,a)}{\nu_{\mu'}(x)} \\ &= \sum_{x,a} \mu(x,a) \log \frac{\mu'(x,a)}{\nu_{\mu'}(x)}. \end{aligned}$$

Thus, the Bregman divergence takes the form

$$\begin{aligned} D_{R_C}(\mu \| \mu') &= \sum_{x,a} \mu(x,a) \left(\log \frac{\mu(x,a)}{\nu_\mu(x)} - \log \frac{\mu'(x,a)}{\nu_{\mu'}(x)} \right) \\ &= \sum_{x,a} \mu(x,a) \log \frac{\pi_\mu(a|x)}{\pi_{\mu'}(a|x)} = \sum_x \nu(x) \sum_a \pi_\mu(a|x) \log \frac{\pi_\mu(a|x)}{\pi_{\mu'}(a|x)}. \end{aligned}$$

This proves that the Bregman divergence corresponding to R_C coincides with D_C, as claimed. To conclude the proof, note that D_C is the average relative entropy between the distributions π_μ and $\pi_{\mu'}$—that is, a sum a positive terms. Indeed, this shows that D_C is nonnegative on the set of state-action distributions $\Delta(\mathcal{X} \times \mathcal{A})$, proving that $R_C(\mu)$ is convex.

A.2 Derivation of optimal policies

Here we prove the results stated in Equations (9)-(11) and Proposition 2, which give the expressions for the dual optimization problems and the optimal solutions corresponding to the primal optimization problem (6), for the two choices of regularization function $D_S(\cdot\|\mu')$ and $D_C(\cdot\|\mu')$. We start with generic derivations that will be helpful for analyzing both cases and then turn to studying the individual regularizers.

Recall that the primal optimization objective in (6) is given by

$$\max_{\mu\in\Delta} \ \widetilde{\rho}_\eta(\mu) = \max_{\mu\in\Delta} \left\{ \sum_{x,a} \mu(x,a) r(x,a) - \frac{1}{\eta} R(\mu) \right\},$$

where Δ, the feasible set of stationary distributions, is defined by the following constraints:

$$\sum_b \mu(y,b) = \sum_{x,a} \mu(x,a) P(y|x,a), \quad \forall y \in \mathcal{X}, \tag{18}$$

$$\sum_{x,a} \mu(x,a) = 1, \tag{19}$$

$$\mu(x,a) \geq 0, \quad \forall (x,a) \in \mathcal{X} \times \mathcal{A}. \tag{20}$$

We begin by noting that for all state-action pairs where $\mu'(x,a) = 0$, the optimal solution $\mu^*_\eta(x,a)$ will also be zero, thanks to the form of our regularized objective. Thus, without loss of generality, we will assume that all states are recurrent under μ': $\mu'(x,a) > 0$ holds for all state-action pairs.

For any choice of regularizer R, the Lagrangian of the primal (6) is given by

$$\begin{aligned}
\mathcal{L}(\mu; V, \lambda, \varphi) &= \sum_{x,a} \mu(x,a) r(x,a) - \frac{1}{\eta} R(\mu) + \sum_y V(y) \left(\sum_{x,a} \mu(x,a) P(y|x,a) - \sum_b \mu(y,b) \right) \\
&\quad + \lambda \left(1 - \sum_{x,a} \mu(x,a) \right) + \sum_{x,a} \varphi(x,a) \mu(x,a) \\
&= \sum_{x,a} \mu(x,a) \left(r(x,a) + \sum_y P(y|x,a) V(y) - V(x) - \lambda + \varphi(x,a) \right) - \frac{1}{\eta} R(\mu) + \lambda \\
&= \sum_{x,a} \mu(x,a) \left(A(x,a) - \lambda + \varphi(x,a) \right) - \frac{1}{\eta} R(\mu) + \lambda,
\end{aligned}$$

where V, λ and φ are the Lagrange multipliers[4], and A is the advantage function for V. Setting the gradient of the Lagrangian with respect to μ to 0 yields the system of equations

$$\begin{aligned}
0 = \frac{\partial \mathcal{L}}{\partial \mu(x,a)} &= \left(A(x,a) - \lambda + \varphi(x,a) \right) - \frac{1}{\eta} \frac{\partial R(\mu)}{\partial \mu(x,a)}, \\
\Leftrightarrow \quad \frac{\partial R(\mu)}{\partial \mu(x,a)} &= \eta \left(A(x,a) - \lambda + \varphi(x,a) \right),
\end{aligned} \tag{21}$$

for all x, a. By the first-order stationary condition, the unique optimal solution μ^*_η satisfies this system of equations. To obtain the final solution we have to compute the optimal values V^*_η, λ^*_η and φ^*_η of the Lagrange multipliers by optimizing the dual optimization objective $g(V, \lambda, \varphi) = \mathcal{L}(\mu^*_\eta; V, \lambda, \varphi)$, and insert into the expression for μ^*_η. V and λ are unconstrained in the dual, while φ satisfies $\varphi(x,a) \geq 0$ for each $(x,a) \in \mathcal{X} \times \mathcal{A}$. We give the derivations for each regularizer below.

A.3 The relative entropy

Here we prove the results for $R(\mu) = D_S(\mu\|\mu') = \sum_{x,a} \mu(x,a) \log \frac{\mu(x,a)}{\mu'(x,a)}$. The gradient of R is

$$\frac{\partial R(\mu)}{\partial \mu(x,a)} = \log \frac{\mu(x,a)}{\mu'(x,a)} + 1.$$

[4]Technically, these are KKT multipliers as we also have inequality constraints. However, these will be eliminated by means of complementary slackness in the next sections.

The optimal state-action distribution μ_η^* is now directly given by Equation (21):

$$\mu_\eta^*(x,a) = \mu'(x,a)\exp\left(\eta\left(A(x,a) - \lambda + \varphi(x,a)\right) - 1\right). \tag{22}$$

For μ_η^* to belong to Δ, it has to satisfy Constraints (19) and (20). Because of the exponent in (22), $\mu_\eta^*(x,a) \geq 0$ trivially holds for any choice of $\varphi(x,a)$, and complementary slackness implies $\varphi_\eta^*(x,a) = 0$ for each (x,a). Eliminating φ and inserting μ_η^* into Constraint (19) gives us

$$\begin{aligned} 1 &= \sum_{x,a} \mu'(x,a)\exp\left(\eta A(x,a)\right) e^{-\eta\lambda - 1}, \\ \Leftrightarrow \quad \lambda &= \frac{1}{\eta}\left(\log \sum_{x,a} \mu'(x,a)\exp\left(\eta A(x,a)\right) - 1\right). \end{aligned} \tag{23}$$

Since the value of λ is uniquely determined by (23), we can optimize the dual over V only. The dual function is given by

$$\begin{aligned} g(V) = \mathcal{L}(\mu_\eta^*; V, \lambda) &= \sum_{x,a} \mu_\eta^*(x,a)\left(A(x,a) - \lambda - \frac{1}{\eta}\log\frac{\mu_\eta^*(x,a)}{\mu'(x,a)}\right) + \lambda = \frac{1}{\eta} + \lambda \\ &= \frac{1}{\eta}\log\sum_{x,a}\mu'(x,a)\exp\left(\eta A(x,a)\right). \end{aligned}$$

This is precisely the dual given in Equation (10). Note that this dual function has no associated constraints. The expression for the optimal state-action distribution in Equation (9) is obtained by inserting the advantage function A_η^* corresponding to the optimal value function V_η^* into (22).

A.4 The conditional entropy

We next prove the results for $R(\mu) = D_C\left(\mu\|\mu'\right) = \sum_{x,a}\mu(x,a)\log\frac{\pi_\mu(a|x)}{\pi_{\mu'}(a|x)}$. The gradient of R is

$$\frac{\partial R(\mu)}{\partial\mu(x,a)} = \log\frac{\pi_\mu(a|x)}{\pi_{\mu'}(a|x)} + \sum_b \frac{\mu(x,b)}{\pi_\mu(b|x)}\cdot\frac{\partial\pi_\mu(b|x)}{\partial\mu(x,a)}.$$

Since the policy is defined as $\pi_\mu(a|x) = \frac{\mu(x,a)}{\nu_\mu(x)}$, its gradient with respect to μ is

$$\frac{\partial\pi_\mu(b|x)}{\partial\mu(x,a)} = \frac{\mathbb{I}_{\{a=b\}}}{\nu_\mu(x)} - \frac{\mu(x,b)}{\nu_\mu(x)^2} = \frac{1}{\nu_\mu(x)}\left(\mathbb{I}_{\{a=b\}} - \pi_\mu(b|x)\right), \quad \forall x\in\mathcal{X}, a, b\in\mathcal{A}.$$

Inserting into the expression for the gradient of R yields

$$\begin{aligned} \frac{\partial R(\mu)}{\partial\mu(x,a)} &= \log\frac{\pi_\mu(a|x)}{\pi_{\mu'}(a|x)} + \sum_b \frac{\pi_\mu(b|x)}{\pi_\mu(b|x)}\left(\mathbb{I}_{\{a=b\}} - \pi_\mu(b|x)\right) \\ &= \log\frac{\pi_\mu(a|x)}{\pi_{\mu'}(a|x)} + 1 - \sum_b \pi_\mu(b|x) = \log\frac{\pi_\mu(a|x)}{\pi_{\mu'}(a|x)}. \end{aligned}$$

The optimal policy π_μ^* is now directly given by Equation (21):

$$\pi_\eta^*(a|x) = \pi_{\mu'}(a|x)\exp\left(\eta\left(A(x,a) - \lambda + \varphi(x,a)\right)\right). \tag{24}$$

For μ_η^* to belong to Δ, it has to satisfy Constraint (20). Because of the exponent in (24) and the fact that $\mu_\eta^*(x,a) \propto \pi_\eta^*(a|x)$, $\mu_\eta^*(x,a) \geq 0$ trivially holds for any choice of $\varphi(x,a)$, implying that $\varphi_\eta^*(x,a) = 0$ for each (x,a) by complementary slackness. Since $\pi_\eta^*(a|x) = \frac{\mu_\eta^*(x,a)}{\nu_\eta^*(x)}$, we also obtain the following set of constraints:

$$\sum_a \pi_\eta^*(a|x) = \sum_a \frac{\mu_\eta^*(x,a)}{\nu_\eta^*(x)} = \frac{\nu_\eta^*(x)}{\nu_\eta^*(x)} = 1, \quad \forall x\in\mathcal{X}.$$

Inserting the expression for π_η^* yields

$$1 = \sum_a \pi_{\mu'}(a|x) \exp\left(\eta A(x,a)\right) e^{-\eta\lambda}, \quad \forall x \in \mathcal{X}.$$

If we expand the expression for $A(x,a)$ and rearrange the terms we obtain

$$V(x) = \frac{1}{\eta} \log \sum_a \pi_{\mu'}(a|x) \exp\left(\eta \left(r(x,a) - \lambda + \sum_y P(y|x,a)V(y)\right)\right), \quad \forall x \in \mathcal{X}. \tag{25}$$

The dual function is obtained by inserting the expression for μ^* into the Lagrangian:

$$g(V,\lambda) = \mathcal{L}(\mu_\eta^*; V, \lambda) = \sum_{x,a} \mu_\eta^*(a|x) \left(A(x,a) - \lambda - \frac{1}{\eta} \log \frac{\pi_\eta^*(a|x)}{\pi_{\mu'}(a|x)}\right) + \lambda = \lambda. \tag{26}$$

Together, Equations (25) and (26) define the dual optimization problem in Proposition 2. The expression for the optimal policy in Equation (11) is obtained by inserting the optimal advantage function A_η^* into (24).

We remark that to recover the optimal stationary state-action distribution μ_η^*, we would have to insert the expression for the optimal policy π_η^* into Constraints (18) and (19), and solve for the stationary state distribution ν_η^*. However, this is not necessary since μ_η^* and ν_η^* are not required to solve the dual function or to compute the optimal policy.

B The regularized Bellman operators

In this section, we define the *regularized Bellman operator* $T_{\pi|\pi'}$ corresponding to the policy π and regularized with respect to baseline π' as

$$T_\eta^{\pi|\pi'}[V](x) = \sum_a \pi(a|x) \left(r(x,a) - \log \frac{\pi(a|x)}{\pi'(a|x)} + \sum_{x'} P(x'|x,a)V(x')\right) \qquad (\forall x)\,.$$

Similarly, we define the *regularized Bellman optimality operator* $T_{*|\pi'}$ with respect to baseline π' as

$$T_\eta^{*|\pi'}[V](x) = \frac{1}{\eta} \log \sum_a \pi'(a|x) \exp\left(\eta \left(r(x,a) + \sum_y P(y|x,a)V(y)\right)\right) \qquad (\forall x)\,,$$

and the *regularized greedy policy* with respect to the baseline π' as

$$G_\eta^{\pi'}[V](a|x) \propto \pi'(a|x) \exp\left(\eta \left(r(x,a) + \sum_y P(y|x,a)V(y) - V(x)\right)\right).$$

With these notations, we can define the *regularized relative value iteration* algorithm with respect to π' by the iteration

$$\pi_{k+1} = G_\eta^{\pi'}[V_k] \qquad V_{k+1}(x) = T_\eta^{*|\pi'}[V_k](x) - \delta_{k+1} \tag{27}$$

for some δ_{k+1} lying between the minimal and maximal values of $T_\eta^{*|\pi'}[V_k]$. A common technique is to fix a reference state x' and choose $\delta_{k+1} = T_\eta^{*|\pi'}[V_k](x')$.

Similarly, we can define the *regularized policy iteration* algorithm by the iteration

$$\pi_{k+1} = G_\eta^{\pi'}[V_k] \qquad V_{k+1}(x) = \left(T_\eta^{\pi_{k+1}|\pi'}\right)^\infty [V_k](x) - \delta_{k+1}, \tag{28}$$

with δ_{k+1} defined analogously.

For establishing the convergence of the above procedures, it is crucial to ensure that the operator $T_\eta^{*|\pi'}$ is a *non-expansion*: For any value functions V_1 and V_2, we need to ensure

$$\left\| T_\eta^{*|\pi'}[V_1] - T_\eta^{*|\pi'}[V_2] \right\| \leq \|V_1 - V_2\|$$

for some norm. We state the following result claiming that the above requirement indeed holds and present the simple proof below. We note that analogous results have been proven several times in the literature, see, e.g., [10, 1].

Proposition 3. *$T_\eta^{*|\pi'}$ is a non-expansion for the supremum norm $\|f\|_\infty = \max_x |f(x)|$.*

Proof. For simplicity, let us introduce the notation $Q_1(x,a) = r(x,a) + \sum_y P(y|x,a)V_1(y)$, with Q_2 defined analogously, and $\Delta = Q_1 - Q_2$. We have

$$\begin{aligned}
T_\eta^{*|\pi'}[V_1](x) - T_\eta^{*|\pi'}[V_2](x) &= \frac{1}{\eta}\left(\log\sum_a \pi'(a|x)\exp\left(\eta Q_1(x,y)\right) - \log\sum_a \pi'(a|x)\exp\left(\eta Q_2(x,y)\right)\right) \\
&= \frac{1}{\eta}\left(\log\frac{\sum_a \pi'(a|x)\exp\left(\eta Q_1(x,y)\right)}{\sum_a \pi'(a|x)\exp\left(\eta Q_2(x,y)\right)}\right) \\
&= \frac{1}{\eta}\left(\log\frac{\sum_a \pi'(a|x)\exp\left(\eta\left(Q_2(x,y)+\Delta(x,y)\right)\right)}{\sum_a \pi'(a|x)\exp\left(\eta Q_2(x,y)\right)}\right) \\
&= \frac{1}{\eta}\log\sum_a p(x,a)\exp\left(\eta\Delta(x,a)\right) \qquad \text{(with an appropriately defined } p\text{)} \\
&\leq \frac{1}{\eta}\log\max_a \exp\left(\eta\Delta(x,a)\right) \\
&= \max_a \Delta(x,a) = \max_a \sum_y P(y|x,a)\left(V_1(y) - V_2(y)\right) \\
&\leq \max_y |V_1(y) - V_2(y)|\,.
\end{aligned}$$

With an analogous technique, we can also show the complementary inequality

$$T_\eta^{*|\pi'}[V_2](x) - T_\eta^{*|\pi'}[V_1](x) \leq \max_y |V_2(y) - V_1(y)|\,,$$

which concludes the proof. □

Together with the easily-seen fact that $T_\eta^{*|\pi'}$ is continuous, this result immediately implies that $T_\eta^{*|\pi'}$ has a fixed point by Brouwer's fixed-point theorem. Furthermore, this insight allows us to treat the value iteration method (27) as an instance of *generalized value iteration*, as defined by Littman and Szepesvári [15].

We now argue that regularized value iteration converges to the fixed point of $T_\eta^{*|\pi'}$. If the initial value function V_0 is bounded, then so is V_k for each k since the operator $T_\eta^{*|\pi'}$ is a non-expansion. Similar to Section 3, we assume without loss of generality that the initial reference policy π_0 has full support, i.e. $\pi_0(a|x) > 0$ for each recurrent state x and each action a. Inspecting the greedy policy operator $G_\eta^{\pi'}$, it is easy to show by induction that π_k has full support for each k. In particular, $\pi_{k+1}(a|x)$ only equals 0 if either $\pi_k(a|x)$ equals 0 or if the exponent $A_k(x,a)$ equals $-\infty$, which is only possible if V_k is unbounded.

Now, since π_k has full support for each k, any trajectory always has a small probability of reaching a given recurrent state. We can now use a similar argument as Bertsekas [4, Prop. 4.3.2] to show that regularized value iteration converges to the fixed point for $T_\eta^{*|\pi'}$.

B.1 Regularized performance difference

In this section we provide a regularized counterpart to the *performance-difference lemma* (6, Prop. 1, 13, Lemma 6.1, 7).

Lemma 2. *For any pair of policies π, π^*, we have*

$$\widetilde{\rho}_\eta(\pi^*) - \widetilde{\rho}_\eta(\pi) = \sum_{x,a} \mu_{\pi^*}(x,a) A_\eta^\pi(x,a) - \frac{1}{\eta} D_C\left(\mu_{\pi^*} \| \mu_\pi\right).$$

Let μ and μ^* be the respective stationary distributions of π and π^*. The statement follows easily from using the definition of A^π_η:

$$\begin{aligned}\sum_{x,a}\mu^*(x,a)A^\pi_\eta(x,a) &= \sum_{x,a}\mu^*(x,a)\left(r(x,a)-\frac{1}{\eta}\log\frac{\pi(a|x)}{\pi'(a|x)}-\widetilde{\rho}_\eta(\mu)+\sum_y P(y|x,a)V^\pi_\eta(y)-V^\pi_\eta(x)\right)\\ &= \widetilde{\rho}_\eta(\mu^*)+\frac{1}{\eta}\sum_{x,a}\mu^*(x,a)\log\frac{\pi^*(a|x)}{\pi(a|x)}-\widetilde{\rho}_\eta(\mu)\\ &\qquad+\sum_{x,a}\mu^*(x,a)\left(\sum_y P(y|x,a)V^\pi_\eta(y)-V^\pi_\eta(x)\right)\\ &= \widetilde{\rho}_\eta(\mu^*)-\widetilde{\rho}_\eta(\mu)+\frac{1}{\eta}D_C\left(\mu^*\|\mu\right),\end{aligned}$$

where the last step follows from the stationarity of μ' and the definition of D_C.

B.2 Regularized policy gradient

Here we prove Lemma 1 which gives the gradient of the regularized average reward $\widetilde{\rho}_\eta(\theta)$ when the policy π_θ is parameterized on θ. Following Sutton et al. [38], we first compute the gradient of $V^{\pi_\theta}_\eta$:

$$\begin{aligned}\frac{\partial V^{\pi_\theta}_\eta(x)}{\partial\theta_i} &= \sum_a\frac{\partial\pi_\theta(a|x)}{\partial\theta_i}\left(r(x,a)-\frac{1}{\eta}\log\frac{\pi_\theta(a|x)}{\pi'(a|x)}-\widetilde{\rho}_\eta(\theta)+\sum_y P(y|x,a)V^{\pi_\theta}_\eta(y)\right)\\ &\quad+\sum_a\pi_\theta(a|x)\left(-\frac{1}{\eta\pi_\theta(a|x)}\frac{\partial\pi_\theta(a|x)}{\partial\theta_i}-\frac{\partial\widetilde{\rho}_\eta}{\partial\theta_i}+\sum_y P(y|x,a)\frac{\partial V^{\pi_\theta}_\eta(y)}{\partial\theta_i}\right).\end{aligned}$$

Rearranging the terms gives us

$$\begin{aligned}\frac{\partial\widetilde{\rho}_\eta}{\partial\theta_i} &= \sum_a\frac{\partial\pi_\theta(a|x)}{\partial\theta_i}\left(r(x,a)-\frac{1}{\eta}\log\frac{\pi_\theta(a|x)}{\pi'(a|x)}-\widetilde{\rho}_\eta(\theta)+\sum_y P(y|x,a)V^{\pi_\theta}_\eta(y)-\frac{1}{\eta}\right)\\ &\quad+\sum_a\pi_\theta(a|x)\sum_y P(y|x,a)\frac{\partial V^{\pi_\theta}_\eta(y)}{\partial\theta_i}-\frac{\partial V^{\pi_\theta}_\eta(x)}{\partial\theta_i}\\ &= \sum_a\frac{\partial\pi_\theta(a|x)}{\partial\theta_i}A^{\pi_\theta}_\eta(x,a)+\sum_a\pi_\theta(a|x)\sum_y P(y|x,a)\frac{\partial V^{\pi_\theta}_\eta(y)}{\partial\theta_i}-\frac{\partial V^{\pi_\theta}_\eta(x)}{\partial\theta_i}.\end{aligned}$$

The last equality follows from the fact that since $\sum_a\partial\pi_\theta(a|x)/\partial\theta_i=0$ for each x, we can add any state-dependent constant to the multiplier of $\partial\pi_\theta(a|x)/\partial\theta_i$; adding the term $\frac{1}{\eta}-V^{\pi_\theta}_\eta(x)$ results in the given expression. Summing both sides over the stationary state distribution ν_{π_θ} yields

$$\begin{aligned}\frac{\partial\widetilde{\rho}_\eta}{\partial\theta_i} &= \sum_x\nu_{\pi_\theta}(x)\frac{\partial\widetilde{\rho}_\eta}{\partial\theta_i}=\sum_{x,a}\nu_{\pi_\theta}(x)\frac{\partial\pi_\theta(a|x)}{\partial\theta_i}A^{\pi_\theta}_\eta(x,a)\\ &\qquad+\sum_y\sum_{x,a}\nu_{\pi_\theta}(x)\pi_\theta(a|x)P(y|x,a)\frac{\partial V^{\pi_\theta}_\eta(y)}{\partial\theta_i}-\sum_x\nu_{\pi_\theta}(x)\frac{\partial V^{\pi_\theta}_\eta(x)}{\partial\theta_i}\\ &= \sum_{x,a}\nu_{\pi_\theta}(x)\frac{\partial\pi_\theta(a|x)}{\partial\theta_i}A^{\pi_\theta}_\eta(x,a)+\sum_y\nu_{\pi_\theta}(y)\frac{\partial V^{\pi_\theta}_\eta(y)}{\partial\theta_i}-\sum_x\nu_{\pi_\theta}(x)\frac{\partial V^{\pi_\theta}_\eta(x)}{\partial\theta_i}\\ &= \sum_{x,a}\nu_{\pi_\theta}(x)\frac{\partial\pi_\theta(a|x)}{\partial\theta_i}A^{\pi_\theta}_\eta(x,a).\end{aligned}$$

To conclude the proof it is sufficient to note that

$$\mu_{\pi_\theta}(x,a)\frac{\partial\log\pi_\theta(a|x)}{\partial\theta_i}=\frac{\mu_{\pi_\theta}(x,a)}{\pi_\theta(a|x)}\frac{\partial\pi_\theta(a|x)}{\partial\theta_i}=\nu_{\pi_\theta}(x)\frac{\partial\pi_\theta(a|x)}{\partial\theta_i}.$$

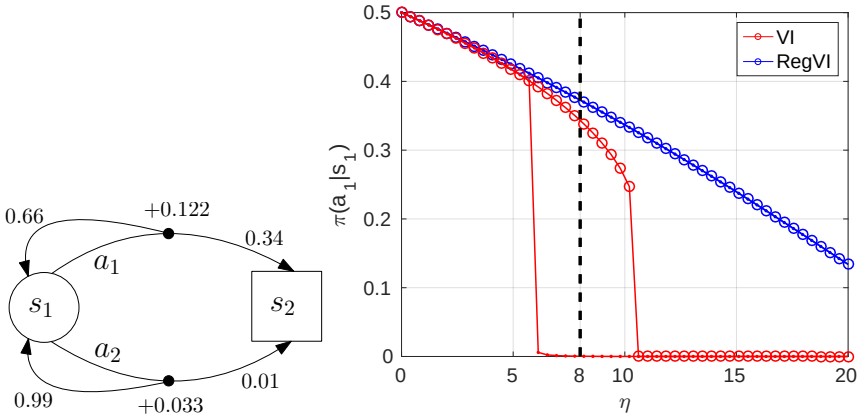

Figure 3: **(left)** The two-state MDP considered in our experiment (from Asadi and Littman [1]). State s_2 is a terminal state. Signed numbers correspond to rewards and unsigned ones to probabilities. We use $\gamma = 1$. **(right)** Standard Value Iteration using softmax policy updates (in red) suffers a hysteresis effect and two solutions coexist for a range of values of η. In contrast Regularized Value Iteration (in blue) has a single optimum. Circles (dots) denote increasing (decreasing) values of η, respectively.

B.3 The closed form of the TRPO update

Here we derive the closed-form solution of the TRPO update. To do so, we first briefly summarize the mechanism of the algorithm. The main idea of Schulman et al. [35] is replacing $\rho(\mu')$ by the surrogate

$$L^{\pi}(\pi') = \rho(\pi) + \sum_x \nu_\pi(x) \sum_a \pi'(a|x) A^{\pi}_{\infty}(x,a),$$

where A^{π}_{∞} is the unregularized advantage function corresponding to policy π. [5] Furthermore, TRPO uses the regularization term

$$D_{\text{TRPO}}(\mu\|\mu') = \sum_x \nu_{\mu'}(x) \sum_a \pi_\mu(a|x) \log \frac{\pi_\mu(a|x)}{\pi_{\mu'}(a|x)}.$$

The difference between $L + D_{\text{TRPO}}$ and $\rho + D_C$ is that the approximate version ignores the impact of changing the policy π on the stationary distribution. Given this surrogate objective, TRPO approximately computes the distribution

$$\mu_{k+1} = \operatorname*{arg\,max}_{\mu \in \Delta} \left\{ L_{\mu_k}(\mu) - \frac{1}{\eta} D_{\text{TRPO}}(\mu\|\mu_k) \right\}. \tag{29}$$

Observing that the TRPO policy update can be expressed equivalently as

$$\pi_{k+1} = \operatorname*{arg\,max}_{\pi} \left\{ \sum_x \nu_{\mu_k}(x) \sum_a \pi(a|x) \left(A^{\pi_k}_{\infty}(x,a) - \frac{1}{\eta} \log \frac{\pi(a|x)}{\pi_k(a|x)} \right) \right\},$$

we can see that the policy update can be expressed in closed form as

$$\pi_{k+1}(a|x) \propto \pi_k(a|x) e^{\eta A^{\pi_k}_{\infty}(x,a)}.$$

This update then can be seen as a regularized greedy step with respect to the value function of the previous policy π_k.

C Complementary experimental results

This section provides further details about the experimental results of Section 5.2. We first highlight the MDP structure shown on Figure 3 (left), and describe the fixed points of softmax value iteration

[5] This form is inspired by the well-known identity we state as Lemma 2.

in this MDP. Figure 3 (right) shows the policies obtained after convergence of VI for different values of η, in two sweeps: first increasing and then decreasing η. For each value of η, VI is run until convergence using as starting condition the solution obtained for the previous step. In the case of standard VI with softmax updates, we observe a hysteresis effect: there is a region of values of η for which two fixed points coexist, one corresponding to $\pi(a_1|s_1) = 0$ and another one corresponding to $\pi(a_1|s_1) > 0$. [6] In contrast, Regularized Value Iteration (in blue), Equation (12), has a single optimum. For the experiments, we set $\eta = 8$, which corresponds to the vertical, dashed line in the figure.

We now turn to describing the details of the gradient-descent methods we use. Gradient descent on the Dual Averaging objective performs the policy update

$$\theta_{k+1} = \theta_k + \alpha \sum_a \mu_{\pi_\theta}(a) \nabla_\theta \log \pi_\theta(a) A_\eta^{\pi_\theta}(a), \tag{30}$$

where $A_\eta^{\pi_\theta}(a) = Q_\eta^{\pi_\theta}(a) - V_\eta^{\pi_\theta}$ is the regularized advantage function. In our case, we use the uniform policy as the reference policy. The full policy evaluation step iterates the following equations until convergence:

$$Q_\eta^{\pi_\theta}(a) = r_{s_1,a} - \frac{1}{\eta} \log(2\pi(a)) + P(s_1|a, s_1) \sum_{a'} \pi_\theta(a') Q_\eta^{\pi_\theta}(a') \tag{31}$$

In contrast, A3C performs updates according to the following regularized gradient

$$\theta_{k+1} = \theta_k + \alpha \left(\sum_a \mu_{\pi_\theta}(a) \left(\nabla_\theta \log \pi_\theta(a) A_\eta^{\pi_\theta}(a) + \frac{1}{\eta} \nabla_\theta H(\pi_\theta(a)) \right) \right), \tag{32}$$

where $H(\cdot)$ is the entropy function. Unlike Dual Averaging, $A_\eta^{\pi_\theta}(a)$ evaluates $\pi_\theta(a)$ using the non-regularized Bellman equation, i.e., Equation (31) without the term $-\frac{1}{\eta} \log(2\pi(a))$.

We run both algorithms using the same settings for different initial random conditions. Figure 2 shows that Dual Averaging always converges to the same policy, represented by the single diagonal ridge. In contrast, the A3C update converges to two different minima, which correspond to the policies in red of Figure 3 (right).

It is also interesting to analyze the behavior of both algorithms in the presence of approximation errors. Figure 4 shows histograms of the values of the first component of θ after convergence for different runs using *the same initial condition*, but replacing the exact policy evaluation step by a Monte Carlo approximation. We observe that, while Dual Averaging converges closer to the previous optimal policy, with small differences caused by the noisy samples, A3C may still converge to either of the different solutions. Therefore, for A3C, only the approximation error from the Monte Carlo samples can entirely determine what is the optimal policy found by the algorithm.

[6] The region differs from that reported by Asadi and Littman [1] because we use $\gamma = 1$.

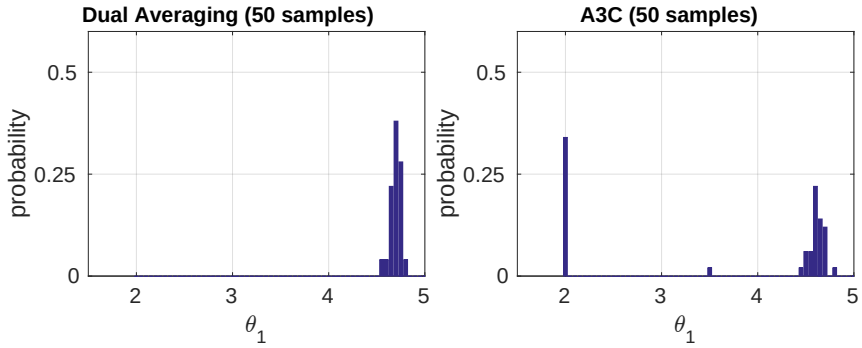

Figure 4: Comparison between Dual Averaging and A3C for *the same initialization* and using an approximate policy evaluation step based on 50 samples. **(left)** Dual Averaging converges to the previous policy up to a difference due to the noisy samples. **(right)** In the presence of approximate evaluation step, A3C converges to different policies, even using the same starting condition.