

Article

# Entropic Regularization of Markov Decision Processes

Boris Belousov <sup>1,\*</sup>  and Jan Peters <sup>1,2</sup> 

<sup>1</sup> Department of Computer Science, Technische Universität Darmstadt, 64289 Darmstadt, Germany

<sup>2</sup> Max Planck Institute for Intelligent Systems, 72076 Tübingen, Germany

\* Correspondence: belousov@ias.tu-darmstadt.de; Tel.: +49-6151-16-25387

Received: 14 June 2019; Accepted: 8 July 2019; Published: 10 July 2019

**Abstract:** An optimal feedback controller for a given Markov decision process (MDP) can in principle be synthesized by value or policy iteration. However, if the system dynamics and the reward function are unknown, a learning agent must discover an optimal controller via direct interaction with the environment. Such interactive data gathering commonly leads to divergence towards dangerous or uninformative regions of the state space unless additional regularization measures are taken. Prior works proposed bounding the information loss measured by the Kullback–Leibler (KL) divergence at every policy improvement step to eliminate instability in the learning dynamics. In this paper, we consider a broader family of  $f$ -divergences, and more concretely  $\alpha$ -divergences, which inherit the beneficial property of providing the policy improvement step in closed form at the same time yielding a corresponding dual objective for policy evaluation. Such entropic proximal policy optimization view gives a unified perspective on compatible actor-critic architectures. In particular, common least-squares value function estimation coupled with advantage-weighted maximum likelihood policy improvement is shown to correspond to the Pearson  $\chi^2$ -divergence penalty. Other actor-critic pairs arise for various choices of the penalty-generating function  $f$ . On a concrete instantiation of our framework with the  $\alpha$ -divergence, we carry out asymptotic analysis of the solutions for different values of  $\alpha$  and demonstrate the effects of the divergence function choice on common standard reinforcement learning problems.

**Keywords:** maximum entropy reinforcement learning; actor-critic methods;  $f$ -divergence; KL control

## 1. Introduction

Sequential decision-making problems under uncertainty are described by the mathematical framework of Markov decision processes (MDPs) [1]. The core problem in MDPs is to find an optimal policy—a mapping from states to actions which maximizes the expected cumulative reward collected by an agent over its lifetime. In reinforcement learning (RL), the agent is additionally assumed to have no prior knowledge about the environment dynamics and the reward function [2]. Therefore, direct policy optimization in the RL setting can be seen as a form of stochastic black-box optimization: the agent proposes a query point in the form of a policy, the environment evaluates this point by computing the expected return, after that the agent updates the proposal and the process repeats [3]. There are two conceptual steps in this scheme known as policy evaluation and policy improvement [4]. Both steps require function approximation in high-dimensional and continuous state-action spaces due to the curse of dimensionality [4]. Therefore, statistical learning approaches are employed to approximate the value function of a policy and to perform policy improvement based on the data collected from the environment.

In contrast to traditional supervised learning, in reinforcement learning, the data distribution changes with every policy update. State-of-the-art generalized policy iteration algorithms [5–8] are

mindful of this covariate shift problem [9], taking active measures to account for it. To smoothen the learning dynamics, these algorithms limit the information loss between successive policy updates as measured by the KL divergence or approximations thereof [10]. In the optimization literature, such approaches are categorized as proximal (or trust region) algorithms [11].

The choice of the divergence function determines the geometry of the information manifold [12]. Recently, in particular in the area of implicit generative modeling [13], the choice of the divergence function was shown to have a dramatic effect both on the optimization performance [14] and the perceptual quality of the generated data when various  $f$ -divergences were employed [15]. In this paper, we carry over the idea of using generalized entropic proximal mappings [16] given by an  $f$ -divergence to reinforcement learning. We show that relative entropy policy search [6], framed as an instance of stochastic mirror descent [17,18] as suggested by [10], can be extended to use any divergence measure from the family of  $f$ -divergences. The resulting algorithm provides insights into the compatibility of policy and value function update rules in actor-critic architectures, which we exemplify on several instantiations of the generic  $f$ -divergence with representatives from the parametric family of  $\alpha$ -divergences [19–21].

## 2. Background

This section provides the necessary background on policy gradients [3] and entropic penalties [16] for later derivations and analysis. Standard RL notation [22] is used throughout.

### 2.1. Policy Gradient Methods

Policy search algorithms [3] commonly use the gradient estimator of the following form [23]

$$\hat{g} = \hat{E}_t [\nabla_{\theta} \log \pi_{\theta} \hat{A}_t^w] \quad (1)$$

where  $\pi_{\theta}(a|s)$  is a stochastic policy and  $\hat{A}_t^w(s_t, a_t)$  is an estimator of the advantage function at timestep  $t$ . Expectation  $\hat{E}_t[\dots]$  indicates an empirical average over a finite batch of samples, in an algorithm that alternates between sampling and optimization. The advantage estimate  $\hat{A}_t^w$  in (1) can be obtained from an estimate of the value function [24,25], which in its turn is found by least-squares estimation. Specifically, if  $V^w(s)$  denotes a parametric value function, and if  $\hat{V}_t = \sum_{k=0}^{\infty} \gamma^k R_{t+k}$  is taken as its rollout-based estimate, then the parameters  $w$  can be found as

$$w = \arg \min_{\tilde{w}} \hat{E}_t [\|V^{\tilde{w}}(s_t) - \hat{V}_t\|^2]. \quad (2)$$

The advantage estimate  $\hat{A}_t^w = \sum_{k=0}^{\infty} \gamma^k \delta_{t+k}^w$  is then obtained by summing the temporal difference errors  $\delta_t^w = R_t + \gamma V^w(s_{t+1}) - V^w(s_t)$ , also known as the Bellman residuals. Treating  $\hat{A}_t^w$  as fixed for the purpose of policy improvement, we can view (1) as the gradient of an advantage-weighted log-likelihood; therefore, the policy parameters  $\theta$  can be found as

$$\theta = \arg \max_{\tilde{\theta}} \hat{E}_t [\log \pi_{\tilde{\theta}} \hat{A}_t^w]. \quad (3)$$

Thus, actor-critic algorithms that use the gradient estimator (1) to update the policy can be viewed as instances of the generalized policy iteration scheme, alternating between policy evaluation (2) and policy improvement (3). In the following, we will see that the actor-critic pair (2) and (3), that combines least-squares value function fitting with linear-in-the-advantage-weighted maximum likelihood policy improvement, is just one representative from a family of such actor-critic pairs arising for different choices of the  $f$ -divergence penalty within our entropic proximal policy optimization framework.

## 2.2. Entropic Penalties

The term entropic penalties [16] refers to both  $f$ -divergences and Bregman divergences. In this paper, we will focus on  $f$ -divergences, leaving generalization to Bregman divergences for future work. The  $f$ -divergence [26] between two distributions  $P$  and  $Q$  with densities  $p$  and  $q$  is defined as

$$D_f(p\|q) = E_q \left[ f \left( \frac{p}{q} \right) \right]$$

where  $f$  is a convex function on  $(0, \infty)$  with  $f(1) = 0$  and  $P$  is assumed to be absolutely continuous with respect to  $Q$ . For example, the KL divergence corresponds to  $f_1(x) = x \log x - (x - 1)$ , with the formula also applicable to unnormalized distributions [27]. Many common divergences lie on the curve of  $\alpha$ -divergences [19,20] defined by a special choice of the generator function [21]

$$f_\alpha(x) = \frac{(x^\alpha - 1) - \alpha(x - 1)}{\alpha(\alpha - 1)}, \quad \alpha \in \mathbb{R}. \quad (4)$$

The  $\alpha$ -divergence  $D_\alpha = D_{f_\alpha}$  will be used as the primary example of the  $f$ -divergence throughout the paper. For more details on the  $\alpha$ -divergence and its properties, see Appendix A. Noteworthy is the symmetry of the  $\alpha$ -divergence with respect to  $\alpha = 0.5$ , which relates reverse divergences as  $D_{0.5+\beta}(p\|q) = D_{0.5-\beta}(q\|p)$ .

## 3. Entropic Proximal Policy Optimization

Consider the average-reward RL setting [2], where the dynamics of an ergodic MDP are given by the transition density  $p(s'|s, a)$ . An intelligent agent can modulate the system dynamics by sampling actions  $a$  from a stochastic policy  $\pi(a|s)$  at every time step of the evolution of the dynamical system. The resulting modulated Markov chain with transition kernel  $p_\pi(s'|s) = \int_A p(s'|s, a)\pi(a|s)da$  converges to a stationary state distribution  $\mu_\pi(s)$  as time goes to infinity. This stationary state distribution induces a state-action distribution  $\rho_\pi(s, a) = \mu_\pi(s)\pi(a|s)$ , which corresponds to visitation frequencies of state-action pairs [1]. The goal of the agent is to steer the system dynamics to desirable states. Such objective is commonly encoded by the expectation of a random variable  $R: S \times A \rightarrow \mathbb{R}$  called reward in this context. Thus, the agent seeks a policy that maximizes the expected reward  $J(\pi) = E_{\rho_\pi(s, a)}[R(s, a)]$ .

In reinforcement learning, neither the reward function  $R$  nor the system dynamics  $p(s'|s, a)$  are assumed to be known. Therefore, to maximize (or even evaluate) the objective  $J(\pi)$ , the agent must sample a batch of experiences in the form of tuples  $(s, a, r, s')$  from the dynamics and use an empirical estimate  $\hat{J} = \hat{E}_t[R(s_t, a_t)]$  as a surrogate for the original objective. Since the gradient of the expected reward with respect to the policy parameters can be written as [28]

$$\nabla_\theta J(\pi_\theta) = E_{\rho_{\pi_\theta}(s, a)}[\nabla_\theta \log \pi_\theta(a|s)R(s, a)]$$

with a corresponding sample-based counterpart

$$\nabla_\theta \hat{J} = \hat{E}_t[\nabla_\theta \log \pi_\theta(a_t|s_t)R(s_t, a_t)],$$

one may be tempted to optimize a sample-based objective

$$\hat{E}_t[\log \pi_\theta(a_t|s_t)R(s_t, a_t)]$$

on a fixed batch of data  $\{(s, a, r, s')_t\}_{t=1}^N$  till convergence. However, such an approach ignores the fact that sampling distribution  $\rho_{\pi_\theta}(s, a)$  itself depends on the policy parameters  $\theta$ ; therefore, such greedy optimization aims at a wrong objective [6]. To have the correct objective, the dataset must be sampled anew after every parameter update—doing otherwise will lead to overfitting and divergence. This problem is known in statistics as the covariate shift problem [9].

### 3.1. Fighting Covariate Shift via Trust Regions

A principled way to account for the change in the sampling distribution at every policy update step is to construct an auxiliary local objective function that can be safely optimized till convergence. Relative entropy policy search (REPS) algorithm [6] proposes a candidate for such an objective

$$J_\eta(\pi) = E_{\rho_\pi}[R] - \eta D_1(\rho_\pi \| \rho_{\pi_0}) \quad (5)$$

with  $\pi_0$  being the current policy under which the data samples were collected, policy  $\pi$  being the improvement policy that needs to be found, and  $\eta > 0$  being a ‘temperature’ parameter that determines how much the next policy can deviate from the current one. The original formulation employs a relative entropy trust region constraint  $D_1$  with radius  $\varepsilon$  instead of a penalty, which allows for finding the optimal temperature  $\eta$  as a function of the trust region radius  $\varepsilon$ .

Importantly, the objective function (5) can be optimized in closed form for policy  $\pi$  (i.e., treating the policy itself as a variable and not its parameters, in contrast to standard policy gradients). To that end, several constraints on  $\rho_\pi$  are added to ensure stationarity with respect to the given MDP [6]. In a similar vein, we can solve Problem (5) with respect to  $\pi$  for any  $f$ -divergence with a twice differentiable generator function  $f$ .

### 3.2. Policy Optimization with Entropic Penalties

Following the intuition of REPS, we introduce an  $f$ -divergence penalized optimization problem that the learning agent must solve at every policy iteration step

$$\begin{aligned} & \underset{\pi}{\text{maximize}} && J_\eta(\pi) = E_{\rho_\pi}[R] - \eta D_f(\rho_\pi \| \rho_{\pi_0}) \\ & \text{subject to} && \int_A \rho_\pi(s', a') da' = \int_{S \times A} \rho_\pi(s, a) p(s' | s, a) ds da, \quad \forall s' \in S, \\ & && \int_{S \times A} \rho_\pi(s, a) ds da = 1, \\ & && \rho_\pi(s, a) \geq 0, \quad \forall (s, a) \in S \times A. \end{aligned} \quad (6)$$

The agent seeks a policy that maximizes the expected reward and does not deviate from the current policy too much. The first constraint in (6) ensures that the policy is compatible with the system dynamics, and the latter two constraints ensure that  $\pi$  is a proper probability distribution. Please note that  $\pi$  enters Problem (6) indirectly through  $\rho_\pi$ . Since the objective has the form of free energy [29] in  $\rho_\pi$  with an  $f$ -divergence playing the role of the usual KL, the solution can be expressed through the derivative of the convex conjugate function  $f'_*$ , as shown for general nonlinear problems in [16],

$$\rho_\pi(s, a) = \rho_{\pi_0}(s, a) f'_* \left( \frac{R(s, a) + \int_S V(s') p(s' | s, a) ds' - V(s) - \lambda + \kappa(s, a)}{\eta} \right). \quad (7)$$

Here,  $\{V(s), \lambda, \kappa(s, a)\}$  are the Lagrange dual variables corresponding to the three constraints in (6), respectively. Although we get a closed-form solution for  $\rho_\pi$ , we still need to solve the dual optimization problem to get the optimal dual variables

$$\begin{aligned} & \underset{V, \lambda, \kappa}{\text{minimize}} && g(V, \lambda, \kappa) = \eta E_{\rho_{\pi_0}} \left[ f'_* \left( \frac{A^V(s, a) - \lambda + \kappa(s, a)}{\eta} \right) \right] + \lambda \\ & \text{subject to} && \kappa(s, a) \geq 0, \quad \forall (s, a) \in S \times A, \\ & && \arg f'_* \in \text{range}_{x \geq 0} f'(x), \quad \forall (s, a) \in S \times A. \end{aligned} \quad (8)$$

Remarkably, the advantage function  $A^V(s, a) = R(s, a) + \int_S V(s') p(s' | s, a) ds' - V(s)$  emerges automatically in the dual objective. The advantage function also appears in the penalty-free linear

programming formulation of policy improvement [1], which corresponds to the zero-temperature limit  $\eta \rightarrow 0$  of our formulation. Thanks to the fact that the dual objective in (8) is given as an expectation with respect to  $\rho_{\pi_0}$ , it can be straightforwardly estimated from rollouts. The last constraint in (8) on the argument of  $f_*$  is easy to evaluate for common  $\alpha$ -divergences. Indeed, the convex conjugate  $f_\alpha^*$  of the generator function (4) is given by

$$f_\alpha^*(y) = \frac{1}{\alpha}(1 + (\alpha - 1)y)^{\frac{\alpha}{\alpha-1}} - \frac{1}{\alpha}, \quad \text{for } y(1 - \alpha) < 1. \tag{9}$$

Thus, the constraint on  $\arg f_*$  in (4) is just a linear inequality  $y(1 - \alpha) < 1$  for any  $\alpha$ -divergence.

### 3.3. Value Function Approximation

For small grid-world problems, one can solve Problem (8) exactly for  $V(s)$ . However, for larger problems or if the state space is continuous, one must resort to function approximation. Assume we plug an expressive function approximator  $V^w(s)$  in (8), then vector  $w$  becomes a new vector of parameters in the dual objective. Later, it will be shown that minimizing the dual when  $\eta \rightarrow \infty$  is closely related to minimizing the mean squared Bellman error.

### 3.4. Sample-Based Algorithm for Dual Optimization

To solve Problem (8) in practice, we gather a batch of samples from policy  $\pi_0$  and replace the expectation in the objective with a sample average. Please note that in principle one also needs to estimate the expectation of the future rewards  $\int_S V(s')p(s'|s, a)ds'$ . However, since the probability of visiting the same state-action pair in continuous space is zero, one commonly estimates this integral from a single sample [3], which is equivalent to assuming deterministic system dynamics. Inequality constraints in (8) are linear and they must be imposed for every  $(s, a)$  pair in the dataset.

### 3.5. Parametric Policy Fitting

Assume Problem (8) is solved on a current batch of data sampled from  $\pi_0$  and thus the optimal dual variables  $\{V(s), \lambda, \kappa(s, a)\}$  are given. Equation (7) allows one to evaluate the new density  $\rho_\pi(s, a)$  on any pair  $(s, a)$  from the dataset. However, it does not yield the new policy  $\pi$  directly because representation (7) is variational. A common approach [3] is to assume that the policy is represented by a parameterized conditional density  $\pi_\theta(a|s)$  and fit this density to the data using maximum likelihood.

To fit a parametric density  $\pi_\theta(a|s)$  to the true solution  $\pi(a|s)$  given by (7), we minimize the KL divergence  $D_1(\rho_\pi \parallel \rho_{\pi_\theta})$ . Minimization of this KL is equivalent to maximization of the weighted maximum likelihood  $\hat{E}[f'_*(\dots) \log \rho_{\pi_\theta}]$ . Unfortunately, distribution  $\rho_{\pi_\theta}(s, a) = \mu_{\pi_\theta}(s)\pi_\theta(a|s)$  is in general not known because  $\mu_{\pi_\theta}(s)$  does not only depend on the policy but also on the system dynamics. Assuming the effect of policy parameters on the stationary state distribution is small [3], we arrive at the following optimization problem for fitting the policy parameters

$$\theta = \arg \max_{\hat{\theta}} \hat{E}_t \left[ \log \pi_{\hat{\theta}}(a_t|s_t) f'_* \left( \frac{\hat{A}^w(s_t, a_t) - \lambda + \kappa(s_t, a_t)}{\eta} \right) \right]. \tag{10}$$

Compare our policy improvement step (10) to the commonly used advantage-weighted maximum likelihood (ML) objective (3). They look surprisingly similar (especially if  $f'_*(y) = y$  is a linear function), which is not a coincidence and will be systematically explained in the next sections.

### 3.6. Temperature Scheduling

The ‘temperature’ parameter  $\eta$  trades off reward vs divergence, as can be seen in the objective function in Problem (6). In practice, devising a schedule for  $\eta$  may be hard because  $\eta$  is sensitive to reward scaling and policy parameterization. A more intuitive way to impose the  $f$ -divergence proximity condition is by adding it as a constraint  $D_f(\rho_\pi \parallel \rho_{\pi_0}) \leq \epsilon$  with a fixed  $\epsilon$  and then treating the

temperature  $\eta \geq 0$  as an optimization variable. Such formulation is easy to incorporate into the dual (8) by adding a term  $\eta\epsilon$  to the objective and a constraint  $\eta \geq 0$  to the list of constraints. Constraint-based formulation was successfully used before with a KL divergence constraint [6] and with its quadratic approximation [5,7].

### 3.7. Practical Algorithm for Continuous State-Action Spaces

Our proposed approach for entropic proximal policy optimization is summarized in Algorithm 1. Following the generalized policy iteration scheme, we (i) collect data under a given policy, (ii) evaluate the policy by solving (8), and (iii) improve the policy by solving (10). In the following section, several instantiations of Algorithm 1 with different choices of function  $f$  will be presented and studied.

---

**Algorithm 1:** Primal-dual entropic proximal policy optimization with function approximation

---

**Input:** Initial actor-critic parameters  $(\theta_0, w_0)$ , divergence function  $f$ , temperature  $\eta > 0$   
**while** *not converged* **do**  
    sample one-step transitions  $\{(s, a, r, s')_t\}_{t=1}^N$  under current policy  $\pi_{\theta_0}$ ;  
    policy evaluation: optimize dual (8) with  $V(s) = V^w(s)$  to obtain critic parameters  $w$ ;  
    policy improvement: perform weighted ML update (10) to obtain actor parameters  $\theta$ ;  
**end**  
**Output:** Optimal policy  $\pi_{\theta}(a|s)$  and the corresponding value function  $V^w(s)$

---

## 4. High- and Low-Temperature Limits; $\alpha$ -Divergences; Analytic Solutions and Asymptotics

How does the  $f$ -divergence penalty influence policy optimization? How should one choose the generator function  $f$ ? What role does the step size play in optimization? This section will try to answer these and related questions. First, two special choices of the penalty function  $f$  are presented, which reveal that the common practice of using mean squared Bellman error minimization coupled with advantage reweighted policy update is equivalent to imposing a Pearson  $\chi^2$ -divergence penalty. Second, high- and low-temperature limits are studied, on one hand revealing the special role the Pearson  $\chi^2$ -divergence plays, being the high-temperature limit of all smooth  $f$ -divergences, and on the other hand establishing a link to the linear programming formulation of policy search as the low-temperature limit of our entropic penalty-based framework.

### 4.1. KL Divergence ( $\alpha = 1$ ) and Pearson $\chi^2$ -Divergence ( $\alpha = 2$ )

As can be deduced from the form of (10), great simplifications occur when  $f'_*(y)$  is a linear function ( $\alpha = 2$ , see (9)) or an exponential function ( $\alpha = 1$ ). The fundamental reason for such simplifications lies in the fact that linear and exponential functions are homomorphisms with respect to addition. This allows, in particular, discovery of a closed-form solution for the dual variable  $\lambda$  and thus eliminate it from the optimization. Moreover, in these two special cases, the dual variables  $\kappa(s, a)$  can also be eliminated. They are responsible for non-negativity of probabilities: when  $\alpha = 1$  (KL),  $\kappa(s, a) = 0$  uniformly for all  $\eta \geq 0$ , when  $\alpha = 2$  (Pearson),  $\kappa(s, a) = 0$  for sufficiently big  $\eta$ . Table 1 gives the corresponding empirical actor-critic optimization objective pairs. A generic primal-dual actor-critic algorithm with an  $\alpha$ -divergence penalty performs two steps

$$\begin{array}{ll} \text{(step 1: policy evaluation)} & \underset{w}{\text{minimize}} \quad \hat{g}_\alpha(w) \\ \text{(step 2: policy improvement)} & \underset{\theta}{\text{maximize}} \quad \hat{L}_\alpha(\theta) \end{array}$$

inside a policy iteration loop. It is worth comparing the explicit formulas in Table 1 to the customarily used objectives (2) and (3). To make the comparison fair, notice that (2) and (3) correspond to discounted infinite horizon formulation with discount factor  $\gamma \in (0, 1)$  whereas formulas in Table 1 are derived for the average-reward setting. In general, the difference between these two settings can be ascribed to

an additional baseline that must be subtracted in the average reward setting [2]. In our derivations, the baseline corresponds to the dual variable  $\lambda$ , as in classical linear programming formulation of policy iteration [1], and it is automatically gets subtracted from the advantage (see (8)).

**Table 1.** Empirical policy evaluation and policy improvement objectives for  $\alpha \in \{1, 2\}$ .

KL Divergence ( $\alpha = 1$ )	Pearson $\chi^2$ -Divergence ( $\alpha = 2$ )
$\hat{g}_1(w) = \eta \log \left( \hat{E}_t \left[ \exp \left( \frac{\hat{A}^w(s_t, a_t)}{\eta} \right) \right] \right)$	$\hat{g}_2(w) = \frac{1}{2\eta} \hat{E}_t \left[ (\hat{A}^w(s_t, a_t) - \hat{E}_t [\hat{A}^w])^2 \right]$
$\hat{L}_1(\theta) = \hat{E}_t \left[ \log \pi_\theta(a_t   s_t) \exp \left( \frac{\hat{A}^w(s_t, a_t) - \hat{g}_1(w)}{\eta} \right) \right]$	$\hat{L}_2(\theta) = \frac{1}{\eta} \hat{E}_t \left[ \log \pi_\theta(a_t   s_t) (\hat{A}^w(s_t, a_t) - \hat{E}_t [\hat{A}^w] + \eta) \right]$

Mean Squared Error Minimization with Advantage Reweighting is Equivalent to Pearson Penalty

The baseline for  $\alpha = 2$  is given by the average advantage  $\lambda_2 = \hat{E}_t [\hat{A}^w(s_t, a_t)]$ , which also equals the average return in our setting [1,2]. Therefore, to translate the formulas from Table 1 to the discounted infinite horizon form (2) and (3), we need to remove the baseline and add discounting to the advantage; that is, set  $A^w(s, a) = R(s, a) + \gamma \int_S V^w(s') p(s' | s, a) ds' - V^w(s)$ . Then the dual objective

$$\hat{g}_2(w) \propto \hat{E}_t \left[ (\hat{A}^w(s_t, a_t))^2 \right] \tag{11}$$

is proportional to the average squared advantage. Naive optimization of (11) leads to the family of residual gradient algorithms [30,31]. However, if the same Monte Carlo estimate of the value function is used as in (2), then (11) and (2) are exactly equivalent. The same holds for the Pearson actor

$$\hat{L}_2(\theta) \propto \hat{E}_t \left[ \log \pi_\theta(a_t | s_t) \hat{A}^w(s_t, a_t) \right] \tag{12}$$

and the standard policy improvement (3) provided that  $\eta = \hat{E}_t [\hat{A}^w(s_t, a_t)]$ . That means (12) is equivalent to (3) if the weight of the divergence penalty is equal to the expected return.

4.2. High- and Low-Temperature Limits

In the previous subsection, we established a direct correspondence between the least-squares value function fitting coupled with the advantage-weighted maximum likelihood policy parameters estimation (2) and (3) and the dual-primal pair of optimization problems (11) and (12) arising from our Algorithm 1 for the special choice of the Pearson  $\chi^2$ -divergence penalty. In this subsection, we will show that this is not a coincidence but a manifestation of the fundamental fact that the Pearson  $\chi^2$ -divergence is the quadratic approximation of any smooth  $f$ -divergence about unity.

4.2.1. High Temperatures: All Smooth  $f$ -Divergences Tend Towards Pearson  $\chi^2$ -Divergence

There are two ways to show the independence of the primal-dual solution (8)–(10) on the choice of the divergence penalty: either exactly solve an approximate problem or approximate the exact solution of the original problem. In the first case, the penalty is replaced with its Taylor expansion at  $\eta \rightarrow \infty$ , which turns out to be the Pearson  $\chi^2$ -divergence, and then the derivation becomes equivalent to the natural policy gradient derivation [5]. In the second case, the exact solution (8)–(10) is expanded by Taylor: for big  $\eta$ , dual variables  $\kappa(s, a)$  can be dropped if  $\rho_{\pi_0}(s, a) > 0$ , which yields

$$f_* \left( \frac{A^w(s, a) - \lambda}{\eta} \right) = f_*(0) + \frac{A^w(s, a) - \lambda}{\eta} f'_*(0) + \frac{1}{2} \left( \frac{A^w(s, a) - \lambda}{\eta} \right)^2 f''_*(0) + o \left( \frac{1}{\eta^2} \right). \tag{13}$$

By definition of the  $f$ -divergence, the generator function  $f$  satisfies the condition  $f(1) = 0$ . Without loss of generality [32], one can impose an additional constraint  $f'(1) = 0$  for convenience. Such constraint ensures that the graph of the function  $f(x)$  lies entirely in the upper half-plane, touching the  $x$ -axis at a single point  $x = 1$ . From the definition of the convex conjugate  $f'_* = (f')^{-1}$ , we can

deduce that  $f'_*(0) = 1$  and  $f_*(0) = 0$ ; by rescaling, it is moreover possible to set  $f''(1) = f''_*(0) = 1$ . These properties are automatically satisfied by the  $\alpha$ -divergence, which can be verified by a direct computation. With this in mind, it is straightforward to see that substitution of (13) into (8) yields precisely the quadratic objective  $\hat{g}_2(w)$  from Table 1, the difference being of the second order in  $1/\eta$ .

To obtain the asymptotic policy update objective, one can expand (10) in the high-temperature limit  $\eta \rightarrow \infty$  and observe that it equals  $\hat{L}_2(\theta)$  from Table 1 with the difference being of the second order in  $1/\eta$ . Therefore, it is established that the choice of the divergence function plays a minor role for big temperatures (small policy update steps). Since this is the mode in which the majority of iterative algorithms operate, our entropic proximal policy optimization point of view provides a rigorous justification for the common practice of using the mean squared Bellman error objective for value function fitting and the advantage-weighted maximum likelihood objective for policy improvement.

#### 4.2.2. Low Temperatures: Linear Programming Formulation Emerges in the Limit

Setting  $\eta$  to a small number is equivalent to allowing large policy update steps because  $\eta$  is the weight of the divergence penalty in the objective function (6). Such regime is rather undesirable in reinforcement learning because of the covariate shift problem mentioned in the introduction. Problem (6) for  $\eta \rightarrow 0$  turns into a well-studied linear programming formulation [1,10] that can be readily applied if the model  $\{p(s'|s, a), R(s, a)\}$  is known.

It is not straightforward to derive the asymptotics of policy evaluation (8) and policy improvement (10) for a general smooth  $f$ -divergence in the low-temperature limit  $\eta \rightarrow 0$  because the dual variables  $\kappa(s, a)$  do not disappear, in contrast to the high-temperature limit (13). However, for the KL divergence penalty (see Table 1), one can show that the policy evaluation objective  $g_1(w)$  tends towards the supremum of the advantage  $g_1(w) \rightarrow \sup_{s,a} A^w(s, a)$ ; the optimal policy is deterministic,  $\pi(a|s) \rightarrow \delta(a - \arg \sup_b A^w(s, b))$ , therefore  $L(\theta) \rightarrow \log \pi_\theta(\bar{a}|\bar{s})$  with  $(\bar{s}, \bar{a}) = \arg \sup_{s',a'} A^w(s', a')$ .

## 5. Empirical Evaluations

To develop an intuition regarding the influence of the entropic penalties on policy improvement, we first consider a simplified version of the reinforcement learning problem—namely the stochastic multi-armed bandit problem [33]. In this setting, our algorithm is closely related to the family of Exp3 algorithms [34], originally motivated by the adversarial bandit problem. Subsequently, we evaluate our approach in the standard reinforcement learning setting.

### 5.1. Illustrative Experiments on Stochastic Multi-Armed Bandit Problems

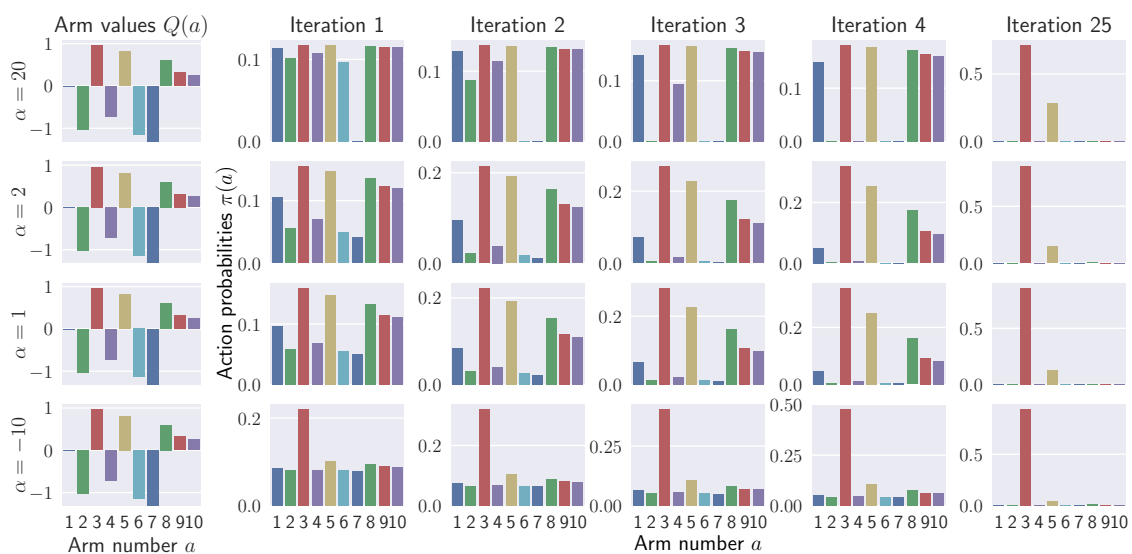
In the stochastic multi-armed bandit problem [33], at every time step  $t \in \{1, \dots, T\}$ , an agent chooses among  $K$  actions  $a \in \mathcal{A}$ . After every choice  $a_t = a$ , it receives a noisy reward  $R_t = R(a_t)$  drawn from a distribution with mean  $Q(a)$ . The goal of the agent is to maximize the expected total reward  $J = E[\sum_{t=1}^T R_t]$ . Given the true values  $Q(a)$ , the optimal strategy is to always choose the best action,  $a_t^* = \arg \max_a Q(a)$ . However, due to the lack of knowledge, the agent faces the exploration-exploitation dilemma. A generic way to encode the exploration-exploitation trade-off is by introducing a policy  $\pi_t$ , i.e., a distribution from which the agent draws actions  $a_t \sim \pi_t$ . Thus, the question becomes: given the current policy  $\pi_t$  and the current estimate of action values  $\hat{Q}_t$ , what should the policy  $\pi_{t+1}$  at the next time step be? Unlike the choice of the best action under perfect information, such sampling policies are hard to derive from first principles [35].

We apply our generic Algorithm 1 to the stochastic multi-armed bandit problem to illustrate the effects of the divergence choice. The value function disappears because there is no state and no system dynamics in this problem. Therefore, the estimate  $\hat{Q}_t$  plays the role of the advantage, and the dual optimization (8) is performed only with respect to the remaining Lagrange multipliers.



### 5.1.1. Effects of $\alpha$ on Policy Improvement

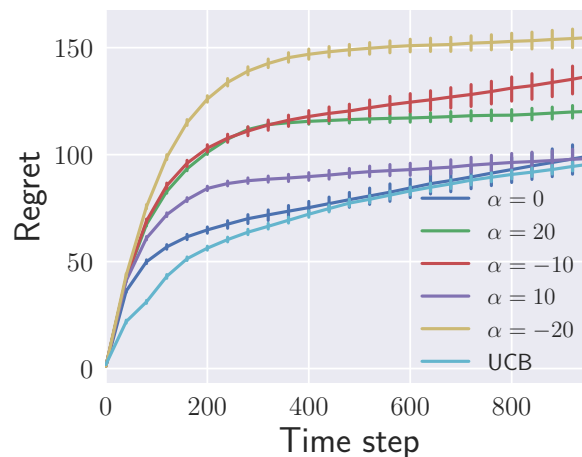
Figure 1 shows the effects of the  $\alpha$ -divergence choice on policy updates. We consider a 10-armed bandit problem with arm values  $Q(a) \sim \mathcal{N}(0, 1)$  and keep the temperature fixed at  $\eta = 2$  for all values of  $\alpha$ . Several iterations starting from an initial uniform policy are shown in the figure for comparison. Extremely large positive and negative values of  $\alpha$  result in  $\epsilon$ -elimination and  $\epsilon$ -greedy policies, respectively. Small values of  $\alpha$ , in contrast, weigh actions according to their values. Policies for  $\alpha < 1$  are peaked and heavy-tailed, eventually turning into  $\epsilon$ -greedy policies when  $\alpha \rightarrow -\infty$ . Policies for  $\alpha \geq 1$  are more uniform, but they put zero mass on bad actions, eventually turning into  $\epsilon$ -elimination policies when  $\alpha \rightarrow \infty$ . For  $\alpha \geq 1$ , policy iteration may spend a lot of time in the end deciding between two best actions, whereas for  $\alpha < 1$  the final convergence is faster.



**Figure 1.** Effects of  $\alpha$  on policy improvement. Each row corresponds to a fixed  $\alpha$ . First four iterations of policy improvement together with a later iteration are shown in each row. Large positive  $\alpha$ 's eliminate bad actions one by one, keeping the exploration level equal among the rest. Small  $\alpha$ 's weigh actions according to their values; actions with low value get zero probability for  $\alpha > 1$ , but remain possible with small probability for  $\alpha \leq 1$ . Large negative  $\alpha$ 's focus on the best action, exploring the remaining actions with equal probability.

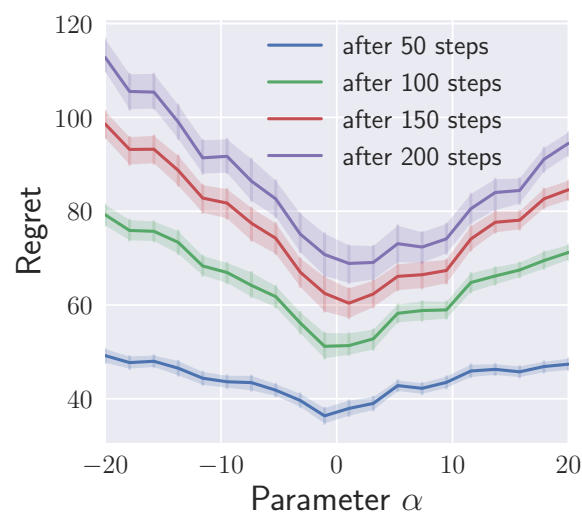
### 5.1.2. Effects of $\alpha$ on Regret

The average regret  $C_n = nQ_{\max} - \mathbb{E}[\sum_{t=0}^{n-1} R_t]$  is shown in Figure 2 for different values of  $\alpha$  as a function of the time step  $n$  with 95% confidence error bars. The performance of the UCB algorithm [33] is also shown for comparison. The presented results are obtained in a 20-armed bandit environment where rewards have Gaussian distribution  $R(a) \sim \mathcal{N}(Q(a), 0.5)$ . Arm values are estimated from observed rewards and the policy is updated every 20 time steps. The temperature parameter  $\eta$  is decreased starting from  $\eta = 1$  after every policy update according to the schedule  $\eta^+ = \beta\eta$  with  $\beta = 0.8$ . Results are averaged over 400 runs. In general, extreme  $\alpha$ 's accumulate more regret. However, they eventually focus on a single action and flatten out. Small  $\alpha$ 's accumulate less regret, but they may keep exploring sub-optimal actions longer. Values of  $\alpha \in [0, 2]$  perform comparably with UCB after around 400 steps, once reliable estimates of values have been obtained.



**Figure 2.** Average regret for various values of  $\alpha$ .

Figure 3 shows the average regret after a given number of time steps as a function of the divergence type  $\alpha$ . As can be seen from the figure, smaller values of  $\alpha$  result in lower regret. Large negative  $\alpha$ 's correspond to  $\epsilon$ -greedy policies, which oftentimes prematurely converge to a sub-optimal action, failing to discover the optimal action for a long time if the exploration probability  $\epsilon$  is small. Large positive  $\alpha$ 's correspond to  $\epsilon$ -elimination policies, which may by mistake completely eliminate the best action or spend a lot of time deciding between two options in the end of learning, accumulating more regret. The optimal value of the parameter  $\alpha$  depends on the time horizon for which the policy is being optimized. Depending on the horizon, the minimum of the curves shifts from slightly negative  $\alpha$ 's towards the range  $\alpha \in [0, 2]$  with increasing time horizon.

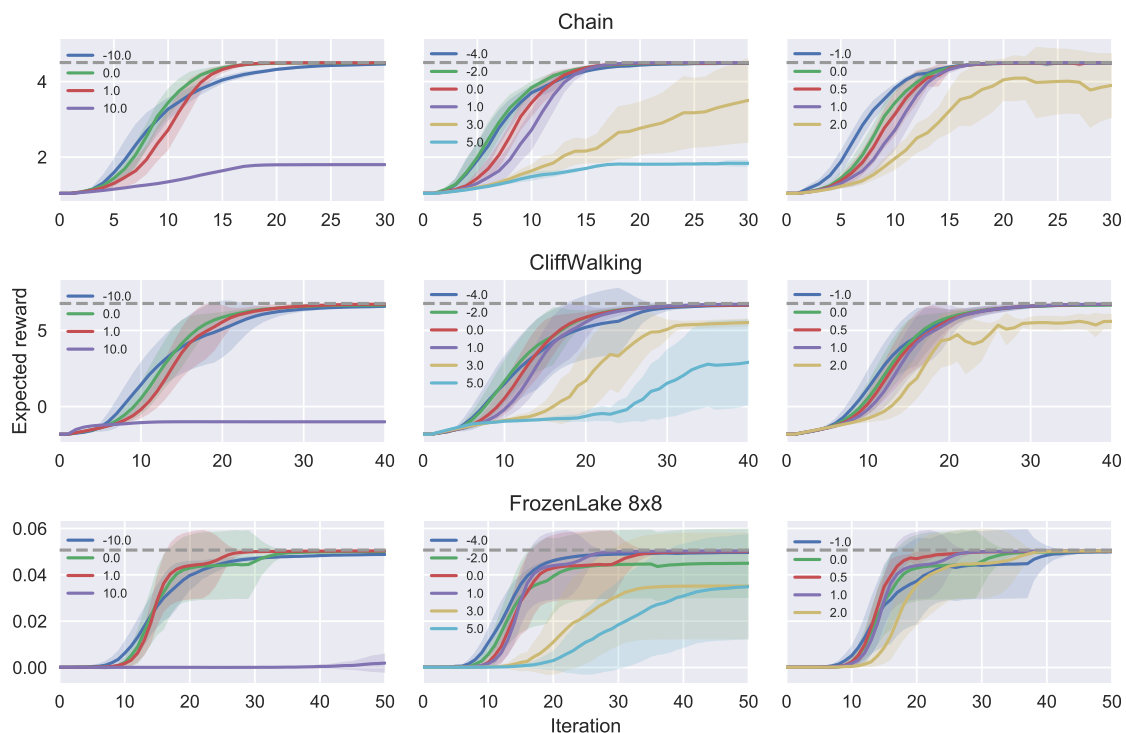


**Figure 3.** Regret after a fixed time as a function of  $\alpha$ .

## 5.2. Empirical Evaluations on Ergodic MDPs

We evaluate our policy iteration algorithm with  $f$ -divergence on standard grid-world reinforcement learning problems from OpenAI Gym [36]. The environments that terminate or have absorbing states are restarted during data collection to ensure ergodicity. Figure 4 demonstrates the learning dynamics on different environments for various choices of the divergence function. Parameter settings and other implementation details can be found in Appendix B. In summary, one can either promote risk averse behavior by choosing  $\alpha < 0$ , which may, however, result in sub-optimal

exploration, or one can promote risk seeking behavior with  $\alpha > 1$ , which may lead to overly aggressive elimination of options. Our experiments suggest that the optimal balance should be found in the range  $\alpha \in [0, 1]$ . It should be noted that the effect of the  $\alpha$ -divergence on policy iteration is not linear and not symmetric with respect to  $\alpha = 0.5$ , contrary to what one could have expected given the symmetry of the  $\alpha$ -divergence as a function of  $\alpha$ . For example, switching from  $\alpha = -3$  to  $\alpha = -2$  may have little effect on policy iteration, whereas switching from  $\alpha = 3$  to  $\alpha = 4$  may have a much more pronounced influence on the learning dynamics.



**Figure 4.** Effects of  $\alpha$ -divergence on policy iteration. Each row corresponds to a given environment. Results for different values of  $\alpha$  are split into three subplots within each row, from the more extreme  $\alpha$ 's on the left to the more refined values on the right. In all cases, more negative values  $\alpha < 0$  initially show faster improvement because they immediately jump to the mode and keep the exploration level low; however, after a certain number of iterations they get overtaken by moderate values  $\alpha \in [0, 1]$  that weigh advantage estimates more evenly. Positive  $\alpha > 1$  demonstrate high variance in the learning dynamics because they clamp the probability of good actions to zero if the advantage estimates are overly pessimistic, never being able to recover from such a mistake. Large positive  $\alpha$ 's may even fail to reach the optimum altogether, as exemplified by  $\alpha = 10$  in the plots. The most stable and reliable  $\alpha$ -divergences lie between the reverse KL ( $\alpha = 0$ ) and the KL ( $\alpha = 1$ ), with the Hellinger distance ( $\alpha = 0.5$ ) outperforming both on the FrozenLake environment.

## 6. Related Work

Apart from computational advantages, information-theoretic approaches provide a solid framework for describing and studying aspects of intelligent behavior [37], from autonomy [38] and curiosity [39] to bounded rationality [40] and game theory [41].

Entropic proximal mappings were introduced in [16] as a general framework for constructing approximation and smoothing schemes for optimization problem. Problem formulation (6) presented here can be considered as an application of this general theory to policy optimization in Markov decision processes. Following the recent work [10] that establishes links between popular in reinforcement learning KL-divergence-regularized policy iteration algorithms [6,7] and a well-known

in optimization stochastic mirror descent algorithm [17,18], one can view our Algorithm 1 as an analog of the mirror descent with an  $f$ -divergence penalty.

Concurrent works [42,43] consider similar regularized formulations, although in the policy space instead of the state-action distribution space and in the infinite horizon discounted setting instead of the average-reward setting. The  $\alpha$ -divergence in its entropic form, i.e., when the base measure is a uniform distribution, was used in several papers under the name Tsallis entropy [44–47], where its sparsifying effect was exploited in large discrete action spaces.

An alternative proximal reinforcement learning scheme was introduced in [48] based on the extragradient method for solving variational inequalities and leveraging operator splitting techniques. Although the idea of exploiting proximal maps and updates in the primal and dual spaces is similar to ours, regularization in [48] is applied in the value function space to smoothen generalized TD learning algorithms, whereas we study regularization in the primal space.

## 7. Conclusions

We presented a framework for deriving actor-critic algorithms as pairs of primal-dual optimization problems resulting from regularization of the standard expected return objective with so-called entropic penalties in the form of an  $f$ -divergence. Several examples with  $\alpha$ -divergence penalties have been worked out in detail. In the limit of small policy update steps, all  $f$ -divergences with twice differentiable generator function  $f$  are approximated by the Pearson  $\chi^2$ -divergence, which was shown to yield the most commonly used in reinforcement learning pair of actor-critic updates. Thus, our framework provides a sound justification for the common practice of minimizing mean squared Bellman error in the policy evaluation step and fitting policy parameters by advantage-weighted maximum likelihood in the policy improvement step.

In the future work, incorporating non-differentiable generator functions, such as the absolute value that corresponds to the total variation distance, may provide a principled explanation for the empirical success of the algorithms not accounted for by our current smooth  $f$ -divergence framework, such as the proximal policy optimization algorithm [8]. Establishing a tighter connection between online convex optimization that employs Bregman divergences and reinforcement learning will likely yield both a deeper understanding of the optimization dynamics in RL and allow for improved practical algorithms building on the firm fundament of optimization theory.

**Author Contributions:** Conceptualization, B.B. and J.P.; investigation, B.B. and J.P.; software, B.B.; supervision, J.P.; writing, B.B. and J.P.

**Funding:** This project has received funding from the European Union’s Horizon 2020 research and innovation program under grant agreement No. 640554.

**Acknowledgments:** We thank Hany Abdulsamad for many insightful discussions.

**Conflicts of Interest:** The authors declare no conflict of interest.

## Appendix A

This section provides the background on the  $f$ -divergence, the  $\alpha$ -divergence, and the convex conjugate function, highlighting the key properties required for our derivations.

The  $f$ -divergence [26,49,50] generalizes many similarity measures between probability distributions [32]. For two distributions  $\pi$  and  $q$  on a finite set  $\mathcal{A}$ , the  $f$ -divergence is defined as

$$D_f(\pi||q) = \sum_{a \in \mathcal{A}} q(a) f\left(\frac{\pi(a)}{q(a)}\right),$$

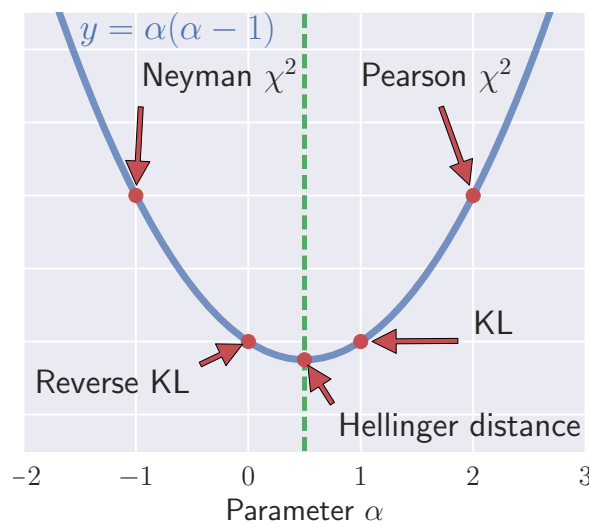
where  $f$  is a convex function on  $(0, \infty)$  such that  $f(1) = 0$ . For example, the KL divergence corresponds to  $f_{KL}(x) = x \log x$ . Please note that  $\pi$  must be absolutely continuous with respect to  $q$  to avoid division by zero, i.e.,  $q(a) = 0$  implies  $\pi(a) = 0$  for all  $a \in \mathcal{A}$ . We additionally assume  $f$  to be continuously differentiable, which includes all cases of interest for us. The  $f$ -divergence can be

generalized to *unnormalized distributions*. For example, the generalized KL divergence [27] corresponds to  $f_1(x) = x \log x - (x - 1)$ . The derivations in this paper benefit from employing unnormalized distributions and subsequently imposing the normalization condition as a constraint.

The  $\alpha$ -divergence [19,20] is a one-parameter family of  $f$ -divergences generated by the  $\alpha$ -function  $f_\alpha(x)$  with  $\alpha \in \mathbb{R}$ . The particular choice of the family of functions  $f_\alpha$  is motivated by generalization of the natural logarithm [21]. The  $\alpha$ -logarithm  $\log_\alpha(x) = (x^{\alpha-1} - 1)/(\alpha - 1)$  is a power function for  $\alpha \neq 1$  that turns into the natural logarithm for  $\alpha \rightarrow 1$ . Replacing the natural logarithm in the derivative of the KL divergence  $f'_1 = \log x$  by the  $\alpha$ -logarithm and integrating  $f'_\alpha$  under the condition that  $f_\alpha(1) = 0$  yields the  $\alpha$ -function

$$f_\alpha(x) = \frac{(x^\alpha - 1) - \alpha(x - 1)}{\alpha(\alpha - 1)}. \tag{A1}$$

The  $\alpha$ -divergence generalizes the KL divergence, reverse KL divergence, Hellinger distance, Pearson  $\chi^2$ -divergence, and Neyman (reverse Pearson)  $\chi^2$ -divergence. Figure A1 displays well-known  $\alpha$ -divergences as points on the parabola  $y = \alpha(\alpha - 1)$ . For every divergence, there is a reverse divergence symmetric with respect to the point  $\alpha = 0.5$ , corresponding to the Hellinger distance.



**Figure A1.** The  $\alpha$ -divergence smoothly connects several prominent divergences.

The convex conjugate of  $f(x)$  is defined as  $f^*(y) = \sup_{x \in \text{dom } f} \{ \langle y, x \rangle - f(x) \}$ , where the angle brackets  $\langle y, x \rangle$  denote the dot product [51]. The key property  $(f^*)' = (f')^{-1}$  relating the derivatives of  $f^*$  and  $f$  yields Table A1, which lists common functions  $f_\alpha$  together with their convex conjugates and derivatives. In the general case (A1), the convex conjugate and its derivative are given by

$$f_\alpha^*(y) = \frac{1}{\alpha} (1 + (\alpha - 1)y)^{\frac{\alpha}{\alpha-1}} - \frac{1}{\alpha},$$

$$(f_\alpha^*)'(y) = \sqrt[\alpha-1]{1 + (\alpha - 1)y}, \quad \text{for } y(1 - \alpha) < 1. \tag{A2}$$

Function  $f_\alpha$  is convex, non-negative, and attains minimum at  $x = 1$  with  $f_\alpha(1) = 0$ . Function  $(f_\alpha^*)'$  is positive on its domain with  $(f_\alpha^*)'(0) = 1$ . Function  $f_\alpha^*$  has the property  $f_\alpha^*(0) = 0$ . The linear inequality constraint (A2) on the dom  $f_\alpha^*$  follows from the requirement  $\text{dom } f_\alpha = (0, \infty)$ . Another result from convex analysis crucial to our derivations is Fenchel's equality

$$f^*(y) + f(x^*(y)) = \langle y, x^*(y) \rangle, \tag{A3}$$

where  $x^*(y) = \arg \sup_{x \in \text{dom } f} \{ \langle y, x \rangle - f(x) \}$ . We will occasionally put the conjugation symbol at the bottom, especially for the derivative of the conjugate function  $f'_* = (f^*)'$ .

**Table A1.** Function  $f_\alpha$ , its convex conjugate  $f_\alpha^*$ , and their derivatives for some values of  $\alpha$ .

Divergence	$\alpha$	$f(x)$	$f'(x)$	$(f^*)'(y)$	$f^*(y)$	dom $f^*$
KL	1	$x \log x - (x - 1)$	$\log x$	$e^y$	$e^y - 1$	$\mathbb{R}$
Reverse KL	0	$-\log x + (x - 1)$	$-\frac{1}{x} + 1$	$\frac{1}{1-y}$	$-\log(1 - y)$	$y < 1$
Pearson $\chi^2$	2	$\frac{1}{2}(x - 1)^2$	$x - 1$	$y + 1$	$\frac{1}{2}(y + 1)^2 - \frac{1}{2}$	$y > -1$
Neyman $\chi^2$	-1	$\frac{(x-1)^2}{2x}$	$-\frac{1}{2x^2} + \frac{1}{2}$	$\frac{1}{\sqrt{1-2y}}$	$-\sqrt{1-2y} + 1$	$y < \frac{1}{2}$
Hellinger	$\frac{1}{2}$	$2(\sqrt{x} - 1)^2$	$2 - \frac{2}{\sqrt{x}}$	$\frac{4}{(2-y)^2}$	$\frac{2y}{2-y}$	$y < 2$

### Appendix B

In all experiments, the temperature parameter  $\eta$  is exponentially decayed  $\eta_{i+1} = \eta_0 a^i$  in each iteration  $i = 0, 1, \dots$ . The choice of  $\eta_0$  and  $a$  depends on the scale of the rewards and the number of samples collected per policy update. Tables for each environment list these parameters along with the number of samples per policy update, the number of policy iteration steps, and the number of runs for averaging the results. Where applicable, environment-specific settings are also listed. (see the Tables A2–A4)

**Table A2.** Chain environment.

Parameter	Value
Number of states	8
Action success probability	0.9
Small and large rewards	(2.0, 10.0)
Number of runs	10
Number of iterations	30
Number of samples	800
Temperature parameters $(\eta_0, a)$	(15.0, 0.9)

**Table A3.** CliffWalking environment.

Parameter	Value
Punishment for falling from the cliff	-10.0
Reward for reaching the goal	100
Number of runs	10
Number of iterations	40
Number of samples	1500
Temperature parameters $(\eta_0, a)$	(50.0, 0.9)

**Table A4.** FrozenLake environment.

Parameter	Value
Action success probability	0.8
Number of runs	10
Number of iterations	50
Number of samples	2000
Temperature parameters $(\eta_0, a)$	(1.0, 0.8)

### References

- Puterman, M.L. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*; John Wiley & Sons: Hoboken, NJ, USA, 1994. [CrossRef]
- Sutton, R.S.; Barto, A.G. *Reinforcement Learning: An Introduction*; MIT Press: Cambridge, MA, USA, 1998.
- Deisenroth, M.P.; Neumann, G.; Peters, J. A survey on policy search for robotics. *Found. Trends® Robot.* **2013**, *2*, 1–142. [CrossRef]
- Bellman, R. Dynamic Programming. *Science* **1957**, *70*, 342. [CrossRef]

5. Kakade, S.M. A Natural Policy Gradient. In Proceedings of the 14th International Conference on Neural Information Processing Systems: Natural and Synthetic, Vancouver, BC, Canada, 3–8 December 2001; pp. 1531–1538. [[CrossRef](#)]
6. Peters, J.; Mülling, K.; Altun, Y. Relative Entropy Policy Search. In Proceedings of the 24th AAAI Conference on Artificial Intelligence, Atlanta, GA, USA, 11–15 July 2010; pp. 1607–1612.
7. Schulman, J.; Levine, S.; Moritz, P.; Jordan, M.; Abbeel, P. Trust Region Policy Optimization. In Proceedings of the 32nd International Conference on International Conference on Machine Learning, Lille, France, 6–11 July 2015.
8. Schulman, J.; Wolski, F.; Dhariwal, P.; Radford, A.; Klimov, O. Proximal policy optimization algorithms. *arXiv* **2017**, arXiv:1707.06347.
9. Shimodaira, H. Improving predictive inference under covariate shift by weighting the log-likelihood function. *J. Stat. Plann. Inference*. **2000**, 227–244. [[CrossRef](#)]
10. Neu, G.; Jonsson, A.; Gómez, V. A unified view of entropy-regularized Markov decision processes. *arXiv* **2017**, arXiv:1705.07798.
11. Parikh, N. Proximal Algorithms. *Found. Trends<sup>®</sup> Optim.* **2014**, 1, 127–239. [[CrossRef](#)]
12. Nielsen, F. An elementary introduction to information geometry. *arXiv* **2018**, arXiv:1808.08271.
13. Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative Adversarial Nets. In Proceedings of the 27th International Conference on Neural Information Processing Systems, Montreal, QC, Canada, 8–13 December 2014.
14. Bottou, L.; Arjovsky, M.; Lopez-Paz, D.; Oquab, M. Geometrical Insights for Implicit Generative Modeling. *Braverman Read. Mach. Learn.* **2018**, 11100, 229–268.
15. Nowozin, S.; Cseke, B.; Tomioka, R. f-GAN: Training Generative Neural Samplers using Variational Divergence Minimization. In Proceedings of the 30th International Conference on Neural Information Processing Systems, Barcelona, Spain, 5–10 December 2016; pp. 271–279.
16. Teboulle, M. Entropic Proximal Mappings with Applications to Nonlinear Programming. *Math. Operations Res.* **1992**, 17, 670–690. [[CrossRef](#)]
17. Nemirovski, A.; Yudin, D. Problem complexity and method efficiency in optimization. *J. Operational Res. Soc.* **1984**, 35, 455.
18. Beck, A.; Teboulle, M. Mirror descent and nonlinear projected subgradient methods for convex optimization. *Operations Res. Lett.* **2003**, 31, 167–175. [[CrossRef](#)]
19. Chernoff, H. A measure of asymptotic efficiency for tests of a hypothesis based on the sum of observations. *Ann. Math. Stat.* **1952**, 23, 493–507. [[CrossRef](#)]
20. Amari, S. *Differential-Geometrical Methods in Statistics*; Springer: New York, NY, USA, 1985. [[CrossRef](#)]
21. Cichocki, A.; Amari, S. Families of alpha- beta- and gamma- divergences: Flexible and robust measures of Similarities. *Entropy* **2010**, 12, 1532–1568. [[CrossRef](#)]
22. Thomas, P.S.; Okal, B. A notation for Markov decision processes. *arXiv* **2015**, arXiv:1512.09075.
23. Sutton, R.S.; Mcallester, D.; Singh, S.; Mansour, Y. Policy Gradient Methods for Reinforcement Learning with Function Approximation. In Proceedings of the 12th International Conference on Neural Information Processing Systems, Denver, CO, USA, 29 November–4 December 1999; pp. 1057–1063. [[CrossRef](#)]
24. Peters, J.; Schaal, S. Natural Actor-Critic. *Neurocomputing* **2008**, 71, 1180–1190. [[CrossRef](#)]
25. Schulman, J.; Moritz, P.; Levine, S.; Jordan, M.I.; Abbeel, P. High Dimensional Continuous Control Using Generalized Advantage Estimation. *arXiv* **2015**, arXiv:1506.02438.
26. Csizsár, I. Eine informationstheoretische Ungleichung und ihre Anwendung auf den Beweis der Ergodizität von Markoffschen Ketten. *Publ. Math. Inst. Hungar. Acad. Sci.* **1963**, 8, 85–108.
27. Zhu, H.; Rohwer, R. Information Geometric Measurements of Generalisation; Technical Report; Aston University: Birmingham, UK, 1995.
28. Williams, R.J. Simple statistical gradient-following methods for connectionist reinforcement learning. *Mach. Learn.* **1992**, 8, 229–256. [[CrossRef](#)]
29. Wainwright, M.J.; Jordan, M.I. Graphical Models, Exponential Families, and Variational Inference. *Found. Trends Mach. Learn.* **2007**, 1, 1–305. [[CrossRef](#)]
30. Baird, L. Residual Algorithms: Reinforcement Learning with Function Approximation. In Proceedings of the 12th International Conference on Machine Learning, Tahoe City, CA, USA, 9–12 July 1995; pp. 30–37. [[CrossRef](#)]

31. Dann, C.; Neumann, G.; Peters, J. Policy Evaluation with Temporal Differences: A Survey and Comparison. *J. Mach. Learn. Res.* **2014**, *15*, 809–883.
32. Sason, I.; Verdu, S. F-divergence inequalities. *IEEE Trans. Inf. Theory* **2016**, *62*, 5973–6006. [[CrossRef](#)]
33. Bubeck, S.; Cesa-Bianchi, N. Regret Analysis of Stochastic and Nonstochastic Multi-armed Bandit Problems. *Found. Trends Mach. Learn.* **2012**, *5*, 1–122. [[CrossRef](#)]
34. Auer, P.; Cesa-Bianchi, N.; Freund, Y.; Schapire, R. The Non-Stochastic Multi-Armed Bandit Problem. *SIAM J. Comput.* **2003**, *32*, 48–77. [[CrossRef](#)]
35. Ghavamzadeh, M.; Mannor, S.; Pineau, J.; Tamar, A. Bayesian Reinforcement Learning: A Survey. *Found. Trends Mach. Learn.* **2015**, *8*, 359–483. [[CrossRef](#)]
36. Brockman, G.; Cheung, V.; Pettersson, L.; Schneider, J.; Schulman, J.; Tang, J.; Zaremba, W. OpenAI Gym. *arXiv* **2016**, arXiv:1606.01540.
37. Tishby, N.; Polani, D. Information theory of decisions and actions. In *Perception-Action Cycle*; Cutsuridis, V., Hussain, A., Taylor, J., Eds.; Springer: New York, NY, USA, 2011; pp. 601–636.
38. Bertschinger, N.; Olbrich, E.; Ay, N.; Jost, J. Autonomy: An information theoretic perspective. *Biosystems* **2008**, *91*, 331–345. [[CrossRef](#)] [[PubMed](#)]
39. Still, S.; Precup, D. An information-theoretic approach to curiosity-driven reinforcement learning. *Theory Biosci.* **2012**, *131*, 139–148. [[CrossRef](#)]
40. Genewein, T.; Leibfried, F.; Grau-Moya, J.; Braun, D.A. Bounded rationality, abstraction, and hierarchical decision-making: An information-theoretic optimality principle. *Front. Rob. AI* **2015**, *2*, 27. [[CrossRef](#)]
41. Wolpert, D.H. Information theory—the bridge connecting bounded rational game theory and statistical physics. In *Complex Engineered Systems*; Braha, D., Minai, A., Bar-Yam, Y., Eds.; Springer: Berlin, Germany, 2006; pp. 262–290.
42. Geist, M.; Scherrer, B.; Pietquin, O. A Theory of Regularized Markov Decision Processes. *arXiv* **2019**, arXiv:1901.11275.
43. Li, X.; Yang, W.; Zhang, Z. A Unified Framework for Regularized Reinforcement Learning. *arXiv* **2019**, arXiv:1903.00725.
44. Nachum, O.; Chow, Y.; Ghavamzadeh, M. Path consistency learning in Tsallis entropy regularized MDPs. *arXiv* **2018**, arXiv:1802.03501.
45. Lee, K.; Kim, S.; Lim, S.; Choi, S.; Oh, S. Tsallis Reinforcement Learning: A Unified Framework for Maximum Entropy Reinforcement Learning. *arXiv* **2019**, arXiv:1902.00137.
46. Lee, K.; Choi, S.; Oh, S. Sparse Markov decision processes with causal sparse Tsallis entropy regularization for reinforcement learning. *IEEE Rob. Autom. Lett.* **2018**, *3*, 1466–1473. [[CrossRef](#)]
47. Lee, K.; Choi, S.; Oh, S. Maximum Causal Tsallis Entropy Imitation Learning. In Proceedings of the 32nd International Conference on Neural Information Processing Systems, Montreal, QC, Canada, 3–8 December 2018; pp. 4408–4418.
48. Mahadevan, S.; Liu, B.; Thomas, P.; Dabney, W.; Giguere, S.; Jacek, N.; Gemp, I.; Liu, J. Proximal reinforcement learning: A new theory of sequential decision making in primal-dual spaces. *arXiv* **2014**, arXiv:1405.6757.
49. Morimoto, T. Markov processes and the H-theorem. *J. Phys. Soc. Jpn.* **1963**, *18*, 328–331. [[CrossRef](#)]
50. Ali, S.M.; Silvey, S.D. A General Class of Coefficients of Divergence of One Distribution from Another. *J. R. Stat. Soc. Ser. B (Methodol.)* **1966**, *28*, 131–142. [[CrossRef](#)]
51. Boyd, S.; Vandenberghe, L. *Convex Optimization*; Cambridge University Press: Cambridge, UK, 2004; 487p. [[CrossRef](#)]

