Proceedings

Entropic Regularization of Markov Decision Processes

Boris Belousov [1,*] and Jan Peters [1,2]

[1] Department of Computer Science, TU Darmstadt, Hochschulstr. 10, 64289 Darmstadt, Germany
[2] Max Planck Institute for Intelligent Systems, Max-Planck-Ring 4, 72076 Tübingen, Germany
* Correspondence: boris@robot-learning.de; Tel.: +49-6151-16-25387

Version September 13, 2018 submitted to Proceedings

Abstract: The problem of synthesis of an optimal feedback controller for a given Markov decision process (MDP) can in principle be solved by value iteration or policy iteration. However, if system dynamics and the reward function are unknown, the only way for a learning agent to discover an optimal controller is through interaction with the MDP. During data gathering, it is crucial to account for the lack of information, because otherwise ignorance will push the agent towards dangerous areas of the state space. To prevent such behavior and smoothen learning dynamics, prior works proposed to bound the information loss measured by the Kullback-Leibler (KL) divergence at every policy improvement step. In this paper, we consider a broader family of f-divergences that preserve the beneficial property of the KL divergence of providing the policy improvement step in closed form accompanied by a compatible 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 fitting coupled with advantage-weighted maximum likelihood policy estimation is shown to correspond to the Pearson χ^2-divergence penalty. Other connections can be established by considering different choices of the penalty generator function f.

Keywords: reinforcement learning; actor-critic methods; entropic proximal mappings; policy search

---

1. Introduction

Top-performing reinforcement learning (RL) algorithms based on generalized policy iteration [1–4] are mindful of the covariate shift [5] problem so characteristic to RL—where data distribution changes after every policy update—and they actively try to alleviate it by limiting the loss of information between successive policies as measured by the KL divergence or approximations thereof [6]. Such approaches broadly fall into the category of proximal (or trust region) optimization algorithms [7].

It has been recently recognized, most prominently in the area of implicit generative modeling [8], that the choice of the distance measure on the space of probability distributions can have dramatic effects on the algorithm performance [9]. This insight, of course, is not entirely new, but it is surprising that just by choosing an appropriate metric one can significantly improve perceptual quality of generated data, as was exemplified in [10] among others, where f-divergence was employed as a measure of image dissimilarity.

In this paper, we carry over the idea of using generalized entropic proximal mappings [11] to reinforcement learning. We show that relative entropy policy search [2], understood as an instance of the mirror descent algorithm [12,13] (as pointed out by [6]), can be naturally extended to use any divergence measure from the family of f-divergences. The resulting algorithm provides deep insights into the compatibility of policy and value function update rules in actor-critic architectures, which we exemplify on several instantiations of the f-divergence from the sub-family of α-divergences [14].

2. Background

2.1. Policy gradient methods

Policy search algorithms [15] commonly use the gradient estimator [16]

$$\hat{g} = \hat{E}_t \left[\nabla_\theta \log \pi_\theta \hat{A}_t^w \right] \tag{1}$$

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. (Standard RL notation [17] is used throughout the paper.) The expectation $\hat{E}_t[\dots]$ indicates the empirical average over a finite batch of samples, in an algorithm that alternates between sampling and optimization. The advantage estimator $\hat{A}_t^w$ is usually fit by a form of least squares regression on the value function

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

followed by summing Bellman residuals $\hat{A}_t^w = \sum_{k=0}^\infty \gamma^k \delta_{t+k}^w$. Here, Monte Carlo estimate of the value function $\hat{V}_t = \sum_{k=0}^\infty \gamma^k R_{t+k}$ is used as the target in (2) and the Bellman residual, also known as the temporal difference (TD) error, is defined as $\delta_t^w = R_t + \gamma V^w(s_{t+1}) - V^w(s_t)$ [18,19]. Treating $\hat{A}_t^w$ as fixed for the purpose of policy update, we can view (1) as the gradient of an advantage-weighted log-likelihood; therefore, the optimal policy parameters θ solve the following optimization problem

$$\theta = \arg\max_{\tilde{\theta}} \ \hat{E}_t \left[\log \pi_{\tilde{\theta}} \hat{A}_t^w \right]. \tag{3}$$

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

2.2. Entropic penalties

As per definition [11], entropic penalties include f-divergences and Bregman divergences. In this paper, we will focus on f-divergences, leaving generalization to Bregman divergences to future work. The f-divergence [20] 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 [21]. Surprisingly, a variety of other commonly used smooth divergences lie on a curve of α-divergences [14,22] that is defined by a special choice of the generator function [23]

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

The α-divergence $D_\alpha = D_{f_\alpha}$ will be used as the primary example of the f-divergence throughout the paper. Noteworthy is the symmetry of the α-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 [24], where dynamics of an ergodic MDP are given by transition density $p(s'|s,a)$ which an intelligent agent can modulate by sampling parameters a from a stochastic policy $\pi(a|s)$ at every time step of the dynamical system evolution. The resulting modulated Markov chain $p_\pi(s'|s) = \int_A p(s'|s,a)\pi(a|s)da$ converges to a stationary state distribution $\mu_\pi(s)$ as time goes to infinity. This stationary state distribution, in its turn, induces a state-action distribution $\rho_\pi(s,a) = \mu_\pi(s)\pi(a|s)$, which corresponds to visitation frequencies of state-action pairs [25]. 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 $R\colon S \times A \to \mathbb{R}$ 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, in order to maximize (or even evaluate) the objective $J(\pi)$, the agent has to 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 policy parameters can be written as [26]

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

with a nice 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 policy parameters θ; therefore, such greedy optimization aims at a wrong objective [2]. To have the correct objective, the dataset must be sampled anew after every parameter update—doing otherwise will lead to catastrophic overfitting. This problem is known in statistics under the name covariate shift [5].

3.1. Fighting covariate shift

A principled way to accommodate the fact that sampling distribution is changing at every policy update step is to construct an auxiliary objective function that one can safely optimize till convergence being assured that negative effects of relying on a fixed dataset are bounded. Relative entropy policy search (REPS) algorithm [2] proposes a candidate for such an objective (in the original paper, a constraint instead of a penalty was used)

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

where π_0 is the current policy from which we collected data samples, policy π is an improved policy we would like to find, and $\eta > 0$ is a 'temperature' parameter that determines how much the next policy is allowed to deviate from the current one. As a measure of distance between probability distributions, the KL divergence D_1, also known as relative entropy, is used in REPS, hence the name.

Interestingly, objective function (5) can be optimized in closed form for policy π (i.e., treating the policy itself as a variable instead of its parameters, in contrast to standard policy gradients). To that end, several constraints on ρ_π need to be added to ensure that it is the stationary state-action distribution of the given MDP [2]. In a similar vein, we can solve Problem (5) for any f-divergence with 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 has to solve at every policy iteration step

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

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 system dynamics, and the latter two constraints ensure that π is a proper probability distribution. Note that π enters Problem (6) indirectly through ρ_π. Since the objective has the form of free energy [27] in ρ_π but with an f-divergence instead 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 [11],

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

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

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

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, as in the penalty-free linear programming formulation of policy improvement [25], which corresponds to the limit $\eta \to 0$. Also note that the dual objective in (8) is given by the expectation with respect to ρ_{π_0}, therefore can be easily estimated from rollouts. The last constraint in (8), despite looking unwieldy, is quite easy to evaluate for common α-divergences; the convex conjugate f^*_α 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 α-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 has to 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 $\eta \to \infty$, which corresponds to small policy update steps, is closely related to minimizing 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 π_0 and replace the expectation in the objective with a sample average. Note that in principle one also needs to estimate the expectation of future rewards $\int_S V(s')p(s'|s,a)ds'$, but since the probability of visiting the same state-action pair in continuous space is zero, one commonly estimates this integral from a single sample as $V(s')$, which is equivalent to assuming deterministic system dynamics [15]. Inequality constraints in (8) are linear and they have to 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 π_0, so 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 π directly because representation (7) is variational. A common approach [15] 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 $\pi(a|s)$ corresponding to (7), we minimize the KL divergence $D_1(\rho_\pi \| \rho_{\pi_\theta})$. Since only samples from ρ_π are known (obtained by weighting samples from ρ_{π_0} according to (7)), minimization of the 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. Neglecting the effect of policy parameters on the stationary state distribution [15], we arrive at the optimization problem for fitting policy parameters

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

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 at all and will be systematically explained later.

3.6. Temperature scheduling

The 'temperature' parameter η trades off reward vs divergence, as can be seen in the primal problem (6), in the objective function. In practice, tuning η may be hard, and simple decay schedules may fail because η is sensitive to reward scaling and policy parameterization. A more intuitive way to impose the f-divergence proximity condition may be to add it as a constraint $D_f(\rho_\pi \| \rho_{\pi_0}) \le \varepsilon$ with a fixed ε, and then treat $\eta \ge 0$ as an optimization variable. Such formulation is easy to incorporate into the dual (8) by adding a term $\eta\varepsilon$ to the objective and a constraint $\eta \ge 0$ to the list of constraints. Constraint-based formulation was successfully used before with a KL divergence constraint [2] and with its quadratic approximation [1,3]. For simplicity, we treat η as a fixed parameter since it also works well in practice if the reward function is well-conditioned.

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 (θ_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 π_{θ_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 θ;
end
Output: Optimal policy $\pi_\theta(a|s)$ and the corresponding value function $V^w(s)$

4. High- and low-temperature limits; α-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 χ^2-divergence penalty. Second, high- and low-temperature limits are studied, which pinpoint the exceptional property of the Pearson χ^2-divergence of being the high-temperature limit of all smooth f-divergences on one hand, and establish a link to the linear programming formulation of policy search as the low-temperature limit of our entropic penalty-based framework on the other hand.

4.1. KL divergence ($\alpha = 1$) and Pearson χ^2-divergence ($\alpha = 2$)

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

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

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

A generic primal-dual actor-critic algorithm with an α-divergence penalty performs two steps

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

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 has to be subtracted in the average reward setting [24]. More precisely, in all our derivations, the baseline corresponds to the dual variable λ, as in classical linear programming formulation of policy iteration [25].

4.1.1. Mean squared error minimization with advantage reweighting is equivalent to Pearson penalty

The baseline for $\alpha = 2$ is given by the average advantage $\lambda_2 = \hat{\mathbb{E}}_t\left[\hat{A}^w(s_t, a_t)\right]$, which also equals the average return in our setting [24,25]. 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

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

is proportional to the average squared advantage. Naive optimization of (11) leads to the family of residual gradient algorithms [28,29]. 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 $\eta = \hat{E}_t \left[\hat{A}^w(s_t, a_t) \right]$; that is, (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 least squares value function fitting coupled with advantage-weighted maximum likelihood policy parameters estimation (2)-(3) and the dual-primal pair of optimization problems (11)-(12) arising from our Algorithm 1 for the special choice of the Pearson χ^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 χ^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 χ^2-divergence

There are two ways to show that the asymptotic of the primal-dual solution (10)-(8) at high temperature is independent of the choice of the divergence function. The first way is to notice that big η leads to small policy update steps, therefore the divergence penalty in the primal problem (6) can be right away replaced by its quadratic approximation, which turns out to be the Pearson χ^2-divergence. After that, one may proceed to solve the problem with such a quadratic penalty, which is exactly equivalent to the natural policy gradient derivation [1].

The second way is to expand the solution (8)-(10) about $\eta \to \infty$. Taking this route, let us develop f_* from (8) into its Taylor series. For big η, we can drop dual variables $\kappa(s,a)$ if $\rho_{\pi_0}(s,a) > 0$. Then

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

By definition of the f-divergence, the generator function f satisfies the condition $f(1) = 0$. Without loss of generality [30], 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 can be checked directly for the α-divergence generator (4) and its convex conjugate (9). With this in mind, it is easy to see that substitution of (13) into (8) leads to $\hat{g}_2(w)$ from Table 1 up to the first order in $1/\eta$.

At the same time, to obtain the asymptotic policy update objective, one can expand (10) in the high-temperature limit $\eta \to \infty$ and observe that it equals $\hat{L}_2(\theta)$ from Table 1 also up to the first order in $1/\eta$. Thereby 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 mean squared Bellman error for value function fitting and advantage-weighted maximum likelihood for updating policy parameters.

4.2.2. Low temperatures: linear programming formulation in the limit

Setting η to a small number is equivalent to allowing large policy update steps because η 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 $\eta \to 0$ turns into a well-studied linear programming formulation [6,25] 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 $\eta \to 0$ because dual variables $\kappa(s,a)$ do not disappear this time, in contrast to the high-temperature limit (13). However, for the KL divergence penalty case (see Table 1), one can show that the policy evaluation objective $g_1(w)$ tends towards supremum of the advantage $g_1(w) \to \sup_{s,a} A^w(s,a)$; the optimal policy is deterministic $\pi(a|s) \to \delta(a - \arg\sup_b A^w(s,b))$, therefore $L(\theta) \to \log \pi_\theta(\bar{a}|\bar{s})$ with $(\bar{s},\bar{a}) = \arg\sup_{s',a'} A^w(s',a')$.

5. Related work

Entropic proximal mappings were introduced in [11] as a general framework for constructing approximation and smoothing schemes for optimization problem. Problem formulation (6) presented here can be considered an application of this general theory to policy optimization in Markov decision processes. Following the recent work [6], that establishes links between popular in reinforcement learning KL-divergence-regularized policy iteration algorithms [2,3] and the well-known in the optimization community mirror descent algorithm [12,13], one can view our Algorithm 1 as an instance of the mirror descent algorithm with an f-divergence penalty.

6. Discussion and conclusion

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 f-divergence. Several examples with α-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 χ^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 future work, it is interesting to consider f-divergence penalties with non-differentiable generator functions such as the absolute value $f(x) = 0.5|x-1|$, which corresponds to the total variation distance, or the absolute value with a dead-zone, which may provide a principled explanation for the empirical success of the proximal policy optimization algorithm [4], not accounted for by our smooth f-divergence framework. Another promising direction to explore is incorporation of Bregman divergences into our formulation; Bregman divergences introduce additional structure that can be exploited for improving sample efficiency of learning algorithms.

Acknowledgments: This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 640554.

Author Contributions: J.P. proposed the use of α-divergence penalties and perceived the significance of the $\alpha = 2$ case; B.B. conceived the general framework based on f-divergence, derived the practical Algorithm 1 together with implications thereof, and wrote the paper.

Conflicts of Interest: The authors declare no conflict of interest. The founding sponsors had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, and in the decision to publish the results.

References

1. Kakade, S.M. A Natural Policy Gradient. NIPS, 2001, pp. 1531–1538.
2. Peters, J.; Mülling, K.; Altun, Y. Relative Entropy Policy Search. AAAI, 2010, pp. 1607–1612.
3. Schulman, J.; Levine, S.; Jordan, M.; Abbeel, P. Trust Region Policy Optimization. ICML, 2015.
4. Schulman, J.; Wolski, F.; Dhariwal, P.; Radford, A.; Klimov, O. Proximal policy optimization algorithms. *arXiv:1707.06347* **2017**.

5. Shimodaira, H. Improving predictive inference under covariate shift by weighting the log-likelihood function. *Journal of Statistical Planning and Inference* **2000**, *90*, 227–244.
6. Neu, G.; Jonsson, A.; Gómez, V. A unified view of entropy-regularized Markov decision processes. *arXiv:1705.07798* **2017**.
7. Parikh, N. Proximal Algorithms. *Foundations and Trends® in Optimization* **2014**, *1*, 127–239.
8. Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative Adversarial Nets. NIPS, 2014.
9. Bottou, L.; Arjovsky, M.; Lopez-Paz, D.; Oquab, M. Geometrical Insights for Implicit Generative Modeling. *arXiv:1712.07822* **2017**.
10. Nowozin, S.; Cseke, B.; Tomioka, R. f-GAN: Training Generative Neural Samplers using Variational Divergence Minimization. NIPS, 2016, pp. 271–279.
11. Teboulle, M. Entropic Proximal Mappings with Applications to Nonlinear Programming. *Mathematics of Operations Research* **1992**, *17*, 670–690.
12. Nemirovski, A.; Yudin, D. *Problem complexity and method efficiency in optimization*; Wiley, 1983.
13. Beck, A.; Teboulle, M. Mirror descent and nonlinear projected subgradient methods for convex optimization. *Operations Research Letters* **2003**, *31*, 167–175.
14. Amari, S. *Differential-Geometrical Methods in Statistics*; Springer New York, 1985.
15. Deisenroth, M.P.; Neumann, G.; Peters, J.; Others. A survey on policy search for robotics. *Foundations and Trends®in Robotics* **2013**, *2*, 1–142.
16. Sutton, R.S.; Mcallester, D.; Singh, S.; Mansour, Y. Policy Gradient Methods for Reinforcement Learning with Function Approximation. NIPS, 1999, pp. 1057–1063.
17. Thomas, P.S.; Okal, B. A notation for Markov decision processes. *arXiv:1512.09075* **2015**.
18. Peters, J.; Schaal, S. Natural Actor-Critic. *Neurocomputing* **2008**, *71*, 1180–1190.
19. Schulman, J.; Moritz, P.; Levine, S.; Jordan, M.I.; Abbeel, P. High Dimensional Continuous Control Using Generalized Advantage Estimation. ICLR, 2016.
20. Csiszár, 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.
21. Zhu, H.; Rohwer, R. Information geometric measurements of generalisation. Technical report, Aston University, 1995.
22. Chernoff, H. A measure of asymptotic efficiency for tests of a hypothesis based on the sum of observations. *The Annals of Mathematical Statistics* **1952**, pp. 493–507.
23. Cichocki, A.; Amari, S. Families of alpha- beta- and gamma- divergences: Flexible and robust measures of Similarities. *Entropy* **2010**, *12*, 1532–1568.
24. Sutton, R.S.; Barto, A.G. *Reinforcement learning: An introduction*; MIT press Cambridge, 1998.
25. Puterman, M.L. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*; 1994.
26. Williams, R.J. Simple statistical gradient-following methods for connectionist reinforcement learning. *Machine Learning* **1992**, *8*, 229–256.
27. Wainwright, M.J.; Jordan, M.I. Graphical Models, Exponential Families, and Variational Inference. *Foundations and Trends in Machine Learning* **2007**, *1*, 1–305.
28. Baird, L. Residual Algorithms: Reinforcement Learning with Function Approximation. *Proceedings of the 12th International Conference on Machine Learning* **1995**, pp. 30–37.
29. Dann, C.; Neumann, G.; Peters, J. Policy Evaluation with Temporal Differences: A Survey and Comparison. *Journal of Machine Learning Research* **2014**, *15*, 809–883.
30. Sason, I.; Verdu, S. F-divergence inequalities. *IEEE Transactions on Information Theory* **2016**, *62*, 5973–6006.

© 2018 by the authors. Submitted to *Proceedings* for possible open access publication under the terms and conditions of the Creative Commons Attribution (CC BY) license (http://creativecommons.org/licenses/by/4.0/).