Constraint-Space Projection Direct Policy Search

Riad Akrour[1] RIAD@ROBOT-LEARNING.DE
Jan Peters[1,2] JAN@ROBOT-LEARNING.DE
Gerhard Neumann[1,2] GERI@ROBOT-LEARNING.DE
[1] *CLAS/IAS, Technische Universitäte Darmstadt, Hochschulstr. 10, D-64289 Darmstadt, Germany*
[2] *Max Planck Institute for Intelligent Systems, Max-Planck-Ring 4, Tübingen, Germany*
[3] *L-CAS, University of Lincoln, Lincoln LN6 7TS, UK*

Abstract

Direct policy search usually frames the search distribution update as a constrained maximization of the expected return. The constraint bounds the information loss of the search distribution and is an ad hoc solution to the exploration-exploitation dilemma. In this paper we propose an alternative to the method of Lagrange multipliers to solve the constrained problem. We propose a projection that maps a parametric representation of the search distribution to a search distribution complying with the update constraints. This projection transforms the constrained optimization problem to an unconstrained one which is then solved using standard gradient ascent. We show on a toy optimization problem that the proposed approach finds better solutions and is more robust to small sample counts than two other state-of-the-art approaches that rely on the method of Lagrange multipliers. In a second phase we extend our approach to step-based reinforcement learning and show that one can seamlessly use the tools introduced in this paper to add hard entropy constraints to existing reinforcement learning algorithms.

1. Introduction

Policy search comprises a wide variety of approaches to tackle reinforcement learning (Deisenroth et al., 2013). Among these approaches, a distinguishing property of direct policy search is its reliance on parameter-space exploration as opposed to action space exploration. In parameter-space exploration, a search distribution is updated from sampled parameters of deterministic policies and an evaluation thereof. In contrast, action space exploration is used to optimize a stochastic policy that adds exploration noise to every time-step. We refer the reader to Deisenroth et al. (2013), Sec. 2.1, for a more in depth discussion on exploration strategies in policy search. In robotics, parameter-space exploration results in a less jerky exploration which reduces the amount of wear and tear caused to the robot. When used in conjunction to specialized low dimensional policies, direct policy search can solve complex tasks in a model-free fashion, running directly on robotic platforms (Parisi et al., 2015). It was also shown in simulation that parameter exploration can be used to train larger, neural network based, policies (Plappert et al., 2017).

We focus in this paper on a simple and well founded formulation of direct policy search that maximizes the expected policy return under Kullback-Leibler (KL) constraint between successive search distributions. The KL constraint is akin to specifying a learning rate, trading-off between exploration and exploitation and preventing the search distribution from collapsing to a point-mass after a single iteration. The considered baseline algorithms

are REPS (Peters et al., 2010) and MORE (Abdolmaleki et al., 2015) that both rely on the method of Lagrange multipliers to obtain a closed form solution to the constrained problem. Although a closed form solution is derived in both cases, the former algorithm does not restrict the search distribution to a particular class in the optimization problem and requires in practice an additional approximation step from samples; causing large violations of the KL constraint. The latter algorithm always complies with the KL constraint but requires to learn a quadratic model mapping policy parameters to policy returns and the policy parameters are usually high dimensional. Limitations of both approaches are especially apparent when the sample count is low compared to the dimensionality of the problem. A concise description of both algorithms is provided in Sec. 4 and their limitations are scrutinized in Sec. 5.

The main contribution of the paper is to propose an alternative approach to updating the search distribution that is robust to low sample counts. The core of our approach lies in a projection g that maps any search distribution to a search distribution that complies with the update constraints. As a result, the constrained maximization of some objective function f is transformed to the unconstrained maximization of $f \circ g$. We experimentally demonstrate on a toy task that maximizing $f \circ g$ by gradient ascent yields comparable results to REPS and MORE in high sample count regimes but significantly outperforms these algorithms when the sample count drops.

An important characteristic of our work is that the constraint projection g is independent of the objective f. We show in Sec. 6 how projections developed for parameter-space exploration algorithms can be seamlessly reused in action-space exploration algorithms. We notably show how to integrate a hard entropy constraint to two popular reinforcement learning algorithms for both discrete and continuous actions spaces, and analyze its empirical effect in Sec. 7.

2. Problem definition

Let f be a noisy function and let $\pi(\theta) = \mathcal{N}(\theta; \mu, \Sigma)$ be a Gaussian distribution of mean μ and covariance matrix Σ over $\mathbb{R}^d$. Typically, θ will be the parameters of a (deterministic) policy and $f(\theta)$ is a random realization of its cumulative rewards in the stochastic environment. Our goal is to find $\theta^* = \arg\max_\theta \mathbb{E}[f(\theta)]$. For this purpose we consider the iterative algorithm that samples and evaluates K parameters from the current Gaussian search distribution q and finds the next search distribution by solving the following constrained optimization problem

$$\begin{aligned} \arg\max_{\pi} \quad & \mathbb{E}_\pi[f(\theta)] && (1)\\ \text{subject to} \quad & \mathrm{KL}(\pi \parallel q) \leq \epsilon, && (2)\\ & \mathcal{H}(q) - \mathcal{H}(\pi) \leq \beta, && (3) \end{aligned}$$

where ϵ and β are two strictly positive constants and $\mathcal{H}$ denotes the entropy. This problem is identical to the one solved by MORE (Abdolmaleki et al., 2015) which adds to the parameter exploration version of REPS (Deisenroth et al. (2013), Sec. 2.4.3) an entropy constraint given by (3). The problem solved by MORE is important and has applications

in approximate policy iteration (Akrour et al., 2018), variational inference (Arenz et al., 2018) and Bayesian optimization (Akrour et al., 2017).

The objective is to find a new search distribution π that maximizes the expected cumulative reward (1) while staying close to the current search distribution (2). The expectation is taken w.r.t. distribution π and the stochastic environment. In our algorithm we approximate (1) by

$$L(\pi) = \frac{1}{K} \sum_{\theta^{[i]} \sim q} \frac{\pi(\theta^{[i]})}{q(\theta^{[i]})} f^{[i]}(\theta^{[i]}). \tag{4}$$

The use of importance sampling in Eq. (4) behaves well in practice because constraint (2) enforces π and q to be close to each other. Although the KL constraint already limits the loss in entropy of q, separating the entropy constraint from the KL constraint allows larger modifications to the mean and rotation of the covariance matrix of q while being more cautious in reducing its entropy (exploration).

3. Constraint projection update

To solve the optimization problem defined in Sec. 2 we will use a series of projections that will ensure that all parameterizations of a search distribution comply with constraints (2) and (3). To fix ideas let us first consider the maximization of some function $L(\pi)$ under a single entropy equality constraint $\mathcal{H}(\pi) = c$ for some $c \in \mathbb{R}$. In relation to Sec. 2, L is as in (4) and $c = \mathcal{H}(q) - \beta$.

Let $\pi = \mathcal{N}(\mu, \Sigma)$ be a Gaussian with diagonal Σ. We recall that the entropy of a Gaussian distribution only depends on its covariance matrix and the notation $\mathcal{H}(\Sigma)$ will be used interchangeably with $\mathcal{H}(\pi)$ which is given by $\mathcal{H}(\Sigma) = \frac{1}{2}\log(|2\pi e\Sigma|)$. Finally we define

$$h(\lambda, c) = \left(\frac{d}{2}\log(2\pi e) + \sum_i \lambda_i\right) - c \tag{5}$$

where the inner most term is the entropy of some diagonal covariance matrix having vector $\exp(2\lambda) \in \mathbb{R}^d$ in its diagonal and c is the target entropy. The first parameterization that transforms a constrained problem to an unconstrained one is given by the following property.

Proposition 1 *Optimizing any function $L(\pi)$ w.r.t. mean vector μ and diagonal matrix Σ of a Gaussian $\pi = \mathcal{N}(\mu, \Sigma)$ under entropy equality constraint $\mathcal{H}(\pi) = c$ is equivalent to the unconstrained optimization of $L(\pi)$ w.r.t. mean vector μ and the real valued parameter vector λ such that $\Sigma_{i,i} = \exp^2(\lambda_i - \frac{1}{d}h(\lambda, c))$ with h as defined in Eq. 5.*

Proof We will show that any value of parameter vector λ will yield a Gaussian distribution that satisfies the entropy equality constraint and that for any Gaussian distribution $\mathcal{N}(\mu, \Sigma)$ satisfying the entropy constraint there is a parameter vector λ representing Σ. First note that for any parameter vector λ the entropy of $\pi = \mathcal{N}(\mu, \Sigma)$ where $\Sigma_{i,i} = \exp^2(\lambda_i - \frac{1}{d}h(\lambda, c))$ is always c—which can be verified through direct computation given the expression of h in Eq. 5. Conversely for any $\pi = \mathcal{N}(\mu, \Sigma)$ such that $\mathcal{H}(\Sigma) = c$ there is a parameter vector λ

that yields the covariance Σ; which is $\lambda_i = \frac{1}{2}\log(\Sigma_{i,i})$ where in this case $h(\lambda, c) = 0$. Hence optimizing $L(\pi)$ w.r.t. Σ under constraint $\mathcal{H}(\pi) = c$ is equivalent to the unconstrained optimization of $L(\pi)$ w.r.t. λ with the given parameterization. ■

Prop. 1 defines a projection g that maps any diagonal covariance matrix to a diagonal covariance matrix having an entropy of exactly c. As this projection is differentiable, and assuming L is also differentiable (which is true for Eq. (4)), one can use gradient ascent for the unconstrained maximization of $L \circ g$. In the following, we will extend this principle to the inequality constraint $\mathcal{H}(\pi) \geq c$, to full covariance matrices and to the KL constraint.

Proposition 2 *Optimizing any function $L(\pi)$ w.r.t. mean vector μ and diagonal matrix Σ of a Gaussian $\pi = \mathcal{N}(\mu, \Sigma)$, under entropy inequality constraint $\mathcal{H}(\pi) \geq c$ is equivalent to the unconstrained optimization of $L(\pi)$ w.r.t. mean vector μ and the real valued parameter vector λ such that $\Sigma_{i,i} = \exp^2(\max(\lambda_i, \lambda_i - \frac{1}{d}h(\lambda, c)))$ with h as defined in Eq. 5.*

Proof As for the equality case, first note that for any vector λ, if $\Sigma_{i,i} = \exp^2(\max(\lambda_i, \lambda_i - \frac{1}{d}h(\lambda, c)))$ and $\Sigma'_{i,i} = \exp^2(\lambda_i - \frac{1}{d}h(\lambda, c))$ then $\mathcal{H}(\Sigma) \geq \mathcal{H}(\Sigma')$ and we have already shown that $\mathcal{H}(\Sigma') = c$. Now let a diagonal Gaussian distribution $\pi = \mathcal{N}(\mu, \Sigma)$ such that $\mathcal{H}(\Sigma) \geq c$ and let λ be the parameter vector such that $\lambda_i = \frac{1}{2}\log(\Sigma_{i,i})$, then $h(\lambda, c) \geq 0$ implying that $\max(\lambda_i, \lambda_i - \frac{1}{d}h(\lambda, c)) = \lambda_i$ and hence the parameter vector λ will yield Σ. As a result, optimizing $L(\pi)$ w.r.t. Σ under constraint $\mathcal{H}(\pi) \geq c$ is equivalent to the unconstrained optimization of $L(\pi)$ w.r.t. λ with the given parameterization. ■

This proposition extends to full covariance matrices Σ where A is its Cholesky decomposition, $\Sigma = AA^T$. By having $A_{i,i} = \exp(\max(\lambda_i, \lambda_i - \frac{1}{d}h(\lambda, c)))$ and real valued off-diagonal entries, all assertions used in the proof of Prop. 2 remain valid since the entropy only depends on the diagonal of A.

Let us now consider the KL constraint. The KL between two Gaussian distributions $\pi = \mathcal{N}(\mu, \Sigma)$ and $q = \mathcal{N}(\mu_q, \Sigma_q)$ is given by

$$\mathrm{KL}(\pi \parallel q) = \frac{1}{2}\left((\mu - \mu_q)^T\Sigma_q^{-1}(\mu - \mu_q) + \mathrm{tr}(\Sigma_q^{-1}\Sigma) - d + \log\frac{|\Sigma_q|}{|\Sigma|}\right).$$

Let $m_q(\mu) = \frac{1}{2}(\mu - \mu_q)^T\Sigma_q^{-1}(\mu - \mu_q)$ be the part of the KL that measures the change in the mean. Assume $m_q(\mu) \neq 0$, using a projected mean of the shape $\mu' = (1 - \eta_1)\mu_q + \eta_1\mu$ one can find η_1 such that the change of the mean in the KL part is equal to some positive target t_1, $\eta_1 = \sqrt{\frac{t_1}{m_q(\mu)}}$. Similarly for $r_q(\Sigma) = \frac{1}{2}(\mathrm{tr}(\Sigma_q^{-1}\Sigma) - d)$ the part of the KL that measures rotation of the covariance matrix, using a projected covariance of the form $\Sigma' = (1 - \eta_2)\Sigma_q + \eta_2\Sigma$ and assuming $r_q(\Sigma) \neq 0$, one can find η_2 such that the change of the rotation in the KL part is equal to some positive target t_2, $\eta_2 = \frac{t_2}{r_q(\Sigma)}$. The remaining term in the KL is related to the change in entropy which is already bounded by the entropy constraint, i.e., if $\mathcal{H}(q) - \mathcal{H}(\pi) \leq \beta$ then $e_q(\Sigma) = \frac{1}{2}\log\frac{|\Sigma_q|}{|\Sigma|} \leq \beta$.

The projection for both the entropy and KL constraints of some search distribution π of mean μ and covariance Σ_0 of Cholesky A_0 proceeds as follow. First we alter the diagonal of the Cholesky following Prop. 2 to ensure that the entropy constraint is respected. Let us denote the resulting covariance by Σ of Cholesky A. Afterwards, if the KL constraint is

respected we return the current μ and A. Otherwise, assuming $\beta < \epsilon$ (if it is not the case the entropy constraint would be superfluous), then we can achieve a target KL of ϵ by dampening $m_q(\mu)$ and $r_q(\Sigma)$. We first dampen $r_q(\Sigma)$ by setting a target $t_2 = r_q(\Sigma)\frac{\epsilon - \max(e_q(\Sigma),0)}{m_q(\mu)+r_q(\Sigma)}$ and compute the resulting η_2 and Σ', then if necessary we reduce $m_q(\mu)$ by setting the target $t_1 = \epsilon - e_q(\Sigma') - r_q(\Sigma')$ and compute μ', which completes the projection.

Note that this projection will not always return a search distribution that has KL equal to ϵ if the initial search distribution has KL higher than ϵ. The reason for this is that when altering $r_q(\Sigma)$ we also alter $e_q(\Sigma)$. For example, if the entropy is maximally reduced, i.e. $\mathcal{H}(\Sigma) = \mathcal{H}(\Sigma_q) - \beta$, then having Σ' interpolating between Σ and Σ_q will result in a smaller entropy reduction and hence the overall KL might be less than ϵ. However, solving $r_q(\Sigma') + e_q(\Sigma') = t_3$ directly is not feasible as it involves solving equations of the form $x + \log x = y$. One way to improve the projection is to use approximations of $x + \log x$ to yield a more accurate η_2. We are unsure however if this would benefit the policy search algorithm as with the current projection, gradient ascent returns solutions that have KL nearly always equal to ϵ. A further theoretical analysis is necessary to clarify this point.

4. State-of-the-art baselines

We consider two baseline algorithms that solve a similar problem to the one in Sec. 2. REPS (Peters et al., 2010; Deisenroth et al., 2013) has the same objective but only constraints the KL. The closed form solution of the update is given by

$$\pi(\theta) \propto q(\theta) \exp\left(\frac{\bar{f}(\theta)}{\eta^*}\right),$$

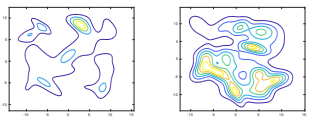
where $\bar{f}(\theta) = \mathbb{E}\left[f(\theta)\right]$ and η^* is a dual variable that is computed using gradient descent. However, π is not necessarily Gaussian and an additional weighted maximum likelihood step is necessary to obtain the next search distribution. This final step can cause large violations of the KL constraint.

MORE (Abdolmaleki et al., 2015) solves the same problem as in Sec. 2, but uses $\hat{f}$, a quadratic approximation of f learned by linear regression. The resulting policy is

$$\pi(\theta) \propto q(\theta)^{\eta^*/(\eta^*+\omega^*)} \exp\left(\frac{\hat{f}(\theta)}{\eta^* + \omega^*}\right).$$

As $\hat{f}$ is quadratic and q Gaussian the resulting search distribution remains Gaussian and the KL and entropy constraints are never violated.

5. Experiments



We compare our approach to the two baselines of Sec. 4 for the optimization of randomly generated and smooth two dimensional objective functions, illustrated in the opposing figure. The results are reported in Fig. 1 on 11 independent runs and varying number of samples per iteration. The 11 randomly generated functions are sampled once and kept fixed for all the algorithms and varying hyper-parameters. For each function,

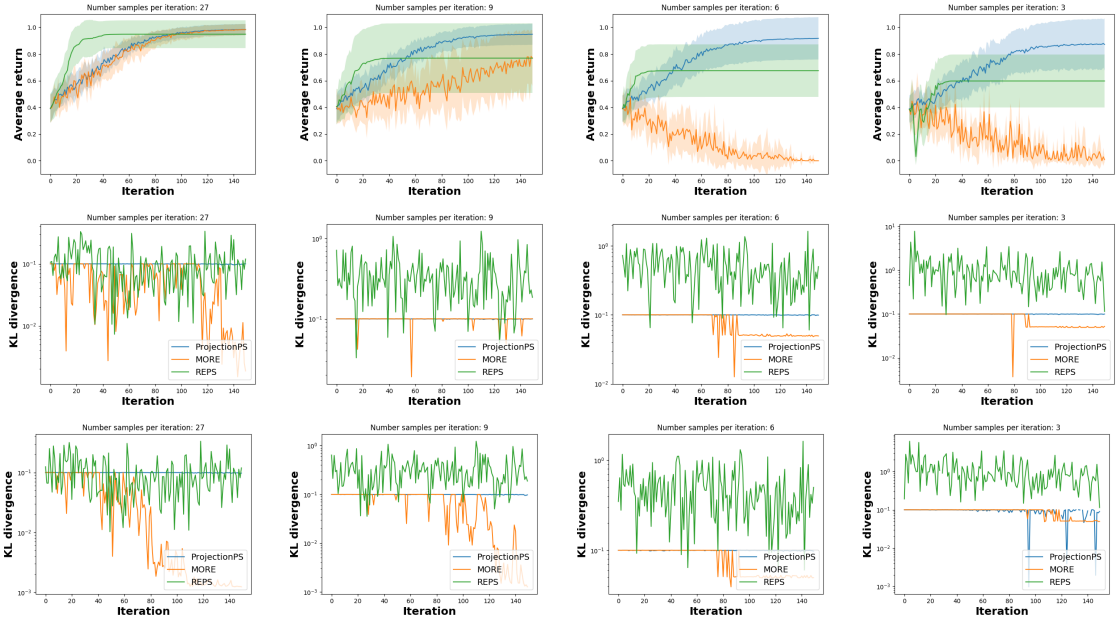


Figure 1: Optimization of smooth objective functions with varying number of samples per iteration, with values of 27, 9, 6, and 3 from left to right columns. First row shows the average return at each iteration averaged over 11 runs. Second and third row show the KL divergence between successive policies of two runs.

the reported results are mapped to $[0, 1]$ after computing the minimal and maximal values reached for this function across all algorithms and hyper-parameters.

First row of Fig. 1 shows the average return at each iteration for the three direct policy search algorithms. The number of samples per iteration takes values 27, 9, 6 and 3 from left to right column respectively while the dimensionality of the problem is $d = 2$. Our approach, termed 'ProjectionPS' is very robust to reduction in sample count and changes moderately across scenarios. While REPS exhibits signs of premature convergence as the sample count drops, caused by large KL constraint violations as seen in Fig. 1, second and third row. MORE never violates the KL constraint but the quadratic models are of poor quality using only 3 and 6 samples which deteriorates performance. Our algorithm nearly always returns a solution with maximum allowed KL constraint $\epsilon = .1$ apart from a single run with a sample count of 3 as seen in Fig. 1.

6. Extension to action-space exploration

We have introduced so far a set of projections to help solve optimization problems involving Gaussian distributions. The projections allow one to tackle the maximization of an objective function under entropy and (I-projection) KL constraint. While finding an appropriate projection for a constraint is not trivial, the advantage of our approach to constrained optimization is that the projection is independent of the objective and can be used to optimize any objective function. We show in this section how projections for entropy constraints can be used to add strict entropy constraints to two step-based RL algorithms.

We consider in this section two approximate policy iteration algorithms, TRPO (Schulman et al., 2015) and PPO (Schulman et al., 2017), as they are applicable to both discrete and continuous action spaces. These two algorithms share the same policy evaluation step but differ largely in their policy update. TRPO maximizes a loss given by

$$L(\pi) = \frac{1}{K} \sum_{(s_i,a_i)\sim q} \frac{\pi(a_i|s_i)}{q(a_i|s_i)} A(s_i, a_i). \tag{6}$$

In contrast to Eq. (4), the policies π and q now represent conditional distributions over an action space $\mathcal{A}$ for each s in state space $\mathcal{S}$, and the advantage function A is the objective function to maximize over. TRPO maximizes this loss under a KL constraint between π and q, while PPO clips the probability ratio $\frac{\pi(a_i|s_i)}{q(q_i|s_i)}$ in Eq. (6) as an alternative way to prevent large deviations between π and q.

Let $P(\pi)$ be the optimization problem solved by either TRPO or PPO. We investigate in this section how to introduce an additional entropy constraint to P of the following form

$$\begin{aligned} &\text{solve for } \pi \quad && P(\pi), \\ &\text{subject to} && \mathcal{H}\left(\pi(.|s)\right) \geq \beta, \text{for all } s. \end{aligned} \tag{7}$$

To solve this problem we will follow a similar approach to the direct policy search case. For continuous action space problems, it is usual for $\pi(.|s)$ to be Gaussian $\mathcal{N}(f_\omega(s), \Sigma)$, where f_ω is a parametric function such as a neural network and Σ is a covariance matrix independent of s. As such, to ensure satisfaction of constraint (7) for Gaussian policies, it suffices to apply the projection introduced in Prop. 2 to Σ.

For discrete action spaces, a usual choice is for π to be a soft-max distribution $\pi(a_i|s) \propto \exp(f^i_\omega(s))$ where f^i_ω is the i-th output of parameterized function f_ω. From here on we term f_ω the 'logits' of π, and let $\mathcal{H}(f_\omega(s))$ be the entropy of the associated soft-max distribution. For a given s, let r_i be the probability of action i according to f_ω, i.e. $r_i \propto \exp(f^i_\omega(s))$. To ensure satisfaction of constraint (7), we derive a projection g_β such that $\mathcal{H}(g_\beta \circ f_\omega(s)) \geq \beta$ for all s. The resulting policy π of logits $g_\beta \circ f_\omega$ is given by

$$\pi(a_i|s) = \begin{cases} r_i, & \text{if } \mathcal{H}(f_\omega) \geq \beta \\ \alpha r_i + (1-\alpha)\frac{1}{|\mathcal{A}|}, & \text{otherwise, where } \alpha = \frac{\log(|\mathcal{A}|)-\beta}{\log(|\mathcal{A}|)-\mathcal{H}(f_\omega)} \end{cases}$$

This policy will always comply with the constraint $\mathcal{H}(\pi(.|s)) \geq \beta$ for all s. It is true by definition for $\mathcal{H}(f_\omega) \geq \beta$ and can easily be verified when $\mathcal{H}(f_\omega) < \beta$ since

$$\begin{aligned} \mathcal{H}\left(\alpha r + (1-\alpha)\frac{1}{|\mathcal{A}|}\right) &\geq \alpha\mathcal{H}(f_\omega) + (1-\alpha)\log\left(|\mathcal{A}|\right), \\ &= \beta. \end{aligned}$$

The inequality follows from the fact that the entropy of a mixture is greater than the mixture of entropies (Cover and Thomas, 2006). The mixture being between the probability distribution defined by r which has entropy $\mathcal{H}(f_\omega)$ and the uniform distribution which has entropy $\log\left(|\mathcal{A}|\right)$. The equality follows from the definition of α. However, as with the KL projection, the projection of the logits is not on the constraint boundary if $\mathcal{H}(f_\omega) < \beta$. We show in Appx. A that it has limited to no effect on a practical case.

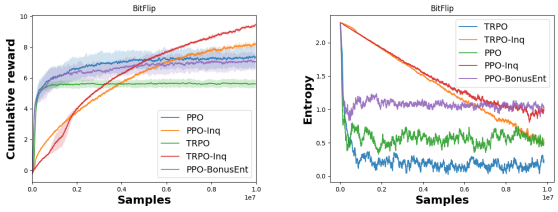


Figure 2: Impact of constraining the entropy of TRPO and PPO on the BitFlip task. Left, policy return averaged over 11 runs. Right, policy entropy of the first run.

7. Evaluation of action-space entropy constraint

We evaluate the impact of constraining the entropy of TRPO and PPO during policy update on 4 tasks from OpenAI's gym (Brockman et al., 2016) for continuous action tasks and on a toy discrete action task. The discrete action task, named 'BitFlip', is designed to require sustained exploration. In BitFlip, the state space is a vector of N bits and there are N actions, flipping the value of each bit. The reward is given by $r(s, a, s') = -\text{val}(s')$ if a flips a bit to 1 and $\text{val}(s)$ otherwise, where $\text{val}(s)$ is the numerical value of the bit vector s. All bits of s_0 are 0 and the optimal policy is to continuously flip the bits from right to left. This problem is challenging because the optimal policy has to chooses roughly half of the time the action that does not provide the highest immediate reward.

In the following, variants of TRPO and PPO with an inequality entropy constraint as defined in Eq. (7) have a β decreasing linearly every iteration. In addition to the base version of each algorithm, we add a variant of PPO with an entropy bonus (instead of a hard constraint). Fig. 2 shows the performance of all 5 algorithms on the BitFlip task averaged over 11 runs and the policy entropy of the first run. The plot shows that TRPO without an entropy constraint converges early to a sub-optimal policy and barely improves over it from there on. The entropy of PPO and PPO-BonusEnt plateaus at a higher level than TRPO. This allows both baselines to steadily improve, but because of the abrupt reduction of entropy in the initial iterations, both baselines improve slowly. In contrast, our linear entropy reduction scheme slows down TRPO-Inq and PPO-Inq initially but allows both algorithms to sustain a fast increase of the policy return, resulting in a significantly better final policy. Similar results are observed in Appx. B for continuous action spaces.

8. Conclusion

We have presented in this paper a new direct policy search algorithm that updates the search distribution under KL and entropy constraint without using the method of Lagrange multipliers. Instead, we introduced a projection that maps any search distribution to a distribution satisfying the update constraints, and solve the resulting unconstrained problem by gradient ascent. We have shown on a toy problem that our approach is more robust to small sample counts compared to related algorithms from the literature and that the same tools developped for policy search can be extended to action-space exploration RL. An interesting direction for future work is to combine the proposed parameter exploration method with a learned Q-Function in order to optimize higher dimensional policies in a similar setting to that of Plappert et al. (2017).

Acknowledgments

The research leading to these results has received funding from the DFG Project LearnRobotS under the SPP 1527 Autonomous Learning, from the Intel Corporation, and from the European Union's Horizon 2020 research and innovation programme under grant agreement No. 640554 (SKILLS4ROBOTS).

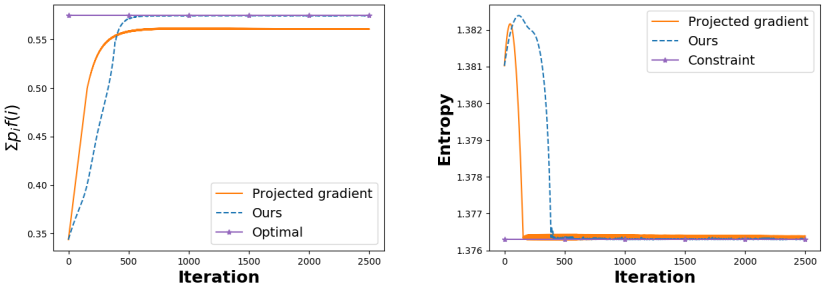


Figure 3: Comparison between the method of projected gradient and our approach using the same projection for the entropy constraint. Our method is able to find the optimum even though the projection is not on the constraint boundary.

Appendix A. Numerical evaluation of soft-max entropy projection

We assess the soft-max entropy constraint projection on a simple problem of maximizing over p the objective $\sum p_i f(i)$ for some arbitrary vector f and under entropy constraint $\mathcal{H}(p) \geq \beta$. Using the method of Lagrange multipliers, the optimal solution is of shape $p_i \propto \exp(f(i)/\eta)$ where η is a dual parameter that can be efficiently computed. We apply our method by optimizing over the 'logits' of p using the projection defined in Sec. 6 to ensure that $\mathcal{H}(p) \geq \beta$. Using the same projection, we compare our method to the projected gradient method that projects the 'logits' after each gradient update—instead of applying gradient ascent to the composition of both the objective and projection.

Fig. 3 shows an example run of such comparison. The projection for the entropy constraint is not on the constraint boundary, preventing the projected gradient method to find an optimum whereas our method is able to find the same solution found when using Lagrange multipliers. We ran 50 independent runs with randomly sampled f and initial 'logits' parameters; the average KL between the distributions found using our method and Lagrange multipliers is $< 10^{-7}$ with similarly small standard deviation. While the average KL between the distribution found using projected gradients and Lagrange multipliers is ≈ 0.016 with standard deviation ≈ 0.014. Despite the soft-max distribution entropy constraint not projecting distributions on the constraint boundary, our method is still able to find a solution that is close to optimal. In contrast, using the same projection but only projecting distributions after gradient direction update does not result in a similar accuracy.

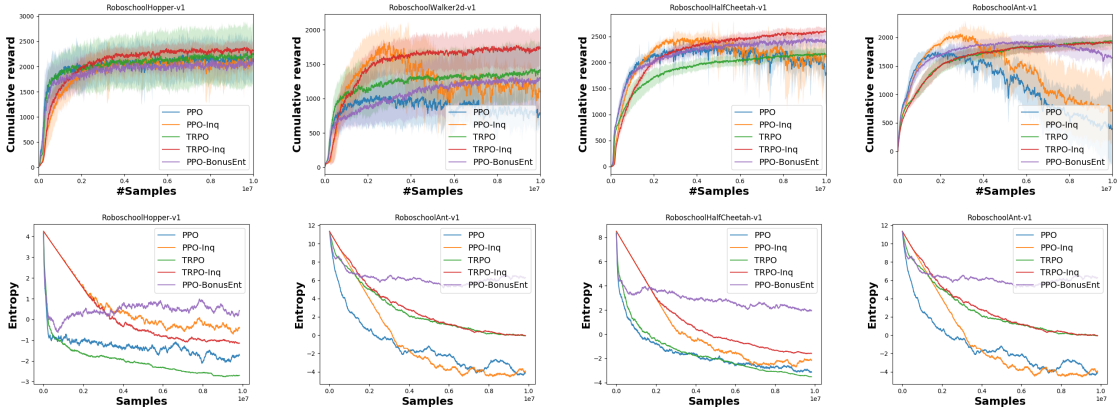


Figure 4: Impact of constraining the entropy of TRPO and PPO on four OpenAI gym tasks. First row shows the policy return averaged over 11 runs. Bottom row shows the policy entropy of the first run.

Appendix B. Evaluation of continuous action entropy constraint

We evaluate the effect of the policy entropy constraint on TRPO and PPO on four tasks from OpenAI's gym testbed comparing the same 5 algorithm variants described in Sec. 7. Fig. 4 shows similar behavior to what was observed with the discrete action task except that the improvements of the end policy are more marginal, if at all present as in the RoboschoolAnt task. This might indicate that rewards are sufficiently informative and exploration in these tasks is not especially challenging.

References

A. Abdolmaleki, R. Lioutikov, J. Peters, N. Lau, L. Pualo Reis, and G. Neumann. Model-based relative entropy stochastic search. In *Advances in Neural Information Processing Systems (NIPS)*. 2015.

R. Akrour, D. Sorokin, J. Peters, and G. Neumann. Local bayesian optimization of motor skills. In *International Conference on Machine Learning (ICML)*, 2017.

R. Akrour, A. Abdolmaleki, H. Abdulsamad, J. Peters, and G. Neumann. Model-free trajectory-based policy optimization with monotonic improvement. *Journal of Machine Learning Resource (JMLR)*, 2018.

O. Arenz, M. Zhong, and G. Neumann. Efficient gradient-free variational inference using policy search. In *International Conference on Machine Learning (ICML)*, 2018.

Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. Openai gym, 2016.

Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory*. Wiley-Interscience, 2006. ISBN 0471241954.

M. P. Deisenroth, G. Neumann, and J. Peters. A Survey on Policy Search for Robotics. *Foundations and Trends in Robotics*, pages 388–403, 2013.

Simone Parisi, Hany Abdulsamad, Alexandros Paraschos, Christian Daniel, and Jan Peters. Reinforcement learning vs human programming in tetherball robot games. In *International Conference on Intelligent Robots and Systems (IROS)*, 2015.

Jan Peters, Katharina Mülling, and Yasemin Altün. Relative entropy policy search. In *AAAI Conference on Artificial Intelligence*, 2010.

Matthias Plappert, Rein Houthooft, Prafulla Dhariwal, Szymon Sidor, Richard Y. Chen, Xi Chen, Tamim Asfour, Pieter Abbeel, and Marcin Andrychowicz. Parameter space noise for exploration. *CoRR*, 2017.

John Schulman, Sergey Levine, Michael Jordan, and Pieter Abbeel. Trust Region Policy Optimization. *International Conference on Machine Learning (ICML)*, page 16, 2015.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *CoRR*, abs/1707.06347, 2017.