European Workshop on Reinforcement Learning 14 (2018) October 2018, Lille, France.

TD-Regularized Actor-Critic Methods

Simone Parisi PARISI@IAS.TU-DARMSTADT.DE
Technical University of Darmstadt, Germany

Voot Tangkaratt VOOT.TANGKARATT@RIKEN.JP
RIKEN Centre for Advanced Intelligence Project, Tokyo, Japan

Jan Peters MAIL@JAN-PETERS.NET
Technical University of Darmstadt, Germany

Emtiyaz Khan EMTIYAZ.KHAN@RIKEN.JP
RIKEN Centre for Advanced Intelligence Project, Tokyo, Japan

Abstract

Actor-critic methods can achieve incredible performance on difficult reinforcement-learning problems, but they are also prone to instability due to the interplay between the actor and critic during learning. To improve their stability, we propose a novel TD-regularized actor-critic method. Our method regularizes the learning objective of the actor by penalizing the temporal difference error of the critic. This improves stability by avoiding overconfident steps in the actor update when the critic is highly inaccurate. We show that our TD-regularization can be easily applied to existing actor-critic methods, e.g., deterministic policy gradient and trust-region policy optimization, with only a slight increase in computation. Evaluations on standard benchmarks show that our method improves stability and exhibits better performance and data-efficiency than its non-regularized counterparts.

1. Introduction

Actor-critic methods have achieved incredible results, showing super-human skills in complex real tasks such as playing Atari games and the game of Go (Silver et al., 2016; Mnih et al., 2016). Their success is partly attributed to the critic, used to approximate the expected return. When compatible linear-functions are used to model the critic, the method shows good convergence behavior (Sutton et al., 1999; Peters and Schaal, 2008), but it can be unstable when nonlinear approximators are used, such as deep neural networks. The instability is partly due to the interplay between the actor and critic during learning; a bad step taken by one of them affects the other, giving rise to an unstable behavior. Figure 1 shows an example for a linear but incompatible function approximator, where we see that many learning-trajectories of the method fail to converge.

Improving the stability of the actor-critic methods is the main focus of this paper. Recent works have proposed methods to do so by improving the stability of the critic, e.g., the use of an additional critic (Lillicrap et al., 2016; Mnih et al., 2015; Hessel et al., 2018) or a low-variance critic (Munos et al., 2016; Gruslys et al., 2018). Others have taken a different route and proposed to stabilize the actor instead, e.g., by constraining its update using entropy or the Kullback-Leibler divergence (Peters et al., 2010; Schulman et al., 2015;

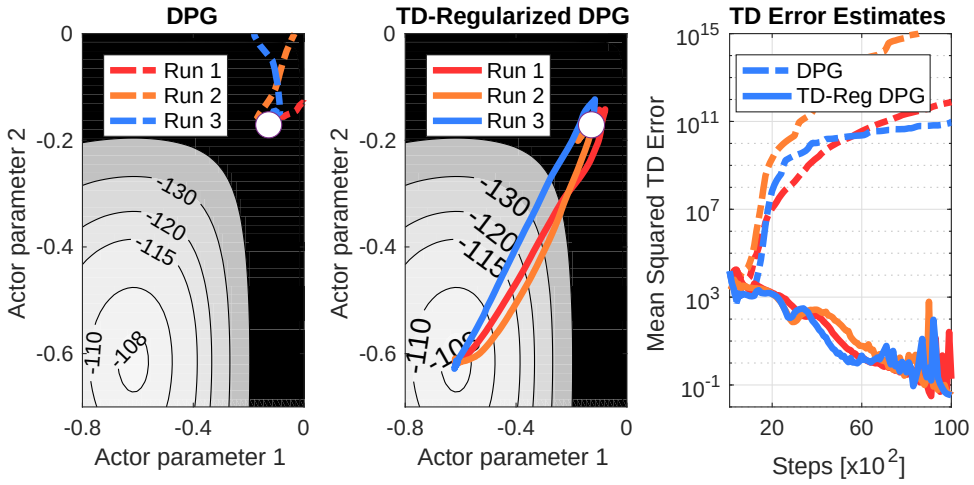


Figure 1: Left figure shows three runs that failed to converge out of ten runs for an actor-critic method called deterministic policy-gradient (DPG). The contours lines show the true expected return for the two parameters of the actor, while the white circle shows the starting parameters. For DPG, we approximate the value function by an incompatible linear function (details in Section 5). None of the three runs make it to the optimum which is located at the bottom left. In contrast, as shown in the middle figure, adding TD-regularization fixes the instability and all the runs converge. The rightmost figure shows the estimated TD error for the two methods. We clearly see that TD-regularization reduces the error over time and improves not only stability and convergence but also the overall performance.

Akrour et al., 2016; Achiam et al., 2017; Nachum et al., 2018; Haarnoja et al., 2018). In all these approaches, stabilizing either the actor or the critic improves the overall stability and achieves good performances. However, these approaches do not directly address the instability caused due to the interplay between the actor and critic.

In this paper, we propose a new method to stabilize the actor by penalizing the inaccuracy in the critic update. A highly inaccurate critic is the one that violates the Bellman equation severely, giving a large temporal-difference (TD) error. To inform the actor about the critic's inaccuracies, we incorporate the critic's TD-error to constraint the actor's learning-objective. By applying a penalty method, we can then update the actor using the usual gradient update, which gives us a simple yet powerful method. We call it the *TD-regularized actor-critic method*. Our method can be easily applied to existing methods such as stochastic and deterministic actor-critic methods (Sutton et al., 1999; Silver et al., 2014) and trust-region policy optimization (Schulman et al., 2015). The TD-regularization only slightly increases the computation cost, as it requires the additional gradient of the regularization term with respect to the actor parameters. Through evaluations on benchmark continuous control tasks, we show that our method improves stability and achieves better performance and data efficiency than its non-regularized counterparts.

2. Preliminaries

We consider Markov Decision Processes (MDPs) described by the tuple $\langle \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \mu_0 \rangle$, where $\mathcal{S} \subseteq \mathbb{R}^{d_{\mathrm{S}}}$ is the state space and $\mathcal{A} \subseteq \mathbb{R}^{d_{\mathrm{A}}}$ is the action space. $\mathcal{P}(s'|s,a)$ defines a Markovian transition probability density between the current s and the next state s' under action a. $\mathcal{R}(s,a)$ is the reward function and μ_0 is the initial state distribution such that $s_0 \sim \mu_0(\cdot)$. We call trajectory or episode a sequence of states, actions and rewards $(s_t, a_t, r_t)_{t=1\dots T}$, where r_t is the reward received according to $\mathcal{R}(s_t, a_t)$, the subscript t denotes the timestep $t \geq 1$, and T is the length of the episode. The goal of reinforcement learning is to learn a policy π maximizing the expected return

$$\max_{\pi} \mathbb{E}_{\mu_\pi(s),\pi(a|s)}[Q^\pi(s,a)], \quad Q^\pi(s_t,a_t) = \mathbb{E}_{s_{t+1:T},a_{t+1:T}}\left[\sum_{i=t}^{T} \gamma^{i-1} r_i \mid \mathcal{P}, \mathcal{R}, \pi\right], \tag{1}$$

where $\mu_\pi(s)$ is the state distribution under π, i.e., the probability of visiting state s when following policy π. Learning $\mu_\pi(s)$ is often very challenging, thus most of reinforcement learning algorithms learn only π, effectively fixing $\mu_\pi(s)$ to the stationary distribution given by previously collected samples. The policy can be deterministic $a = \pi(s)$ or stochastic $a \sim \pi(\cdot|s)$. $Q^\pi(s,a)$ is the Q-function of π and denotes the expected return of executing action a in state s and following π afterwards. The discount factor $\gamma \in [0,1)$ assigns weights to rewards observed at different timesteps. Another important quantity is the V-function of π, $V^\pi(s_t) = \mathbb{E}_{s_{t+1:T},a_{t:T}}\left[\sum_{i=t}^{T} \gamma^{i-1} r_i \mid \mathcal{P}, \mathcal{R}, \pi\right]$, denoting the expected return of being in state s_t and executing actions according to π afterwards.

In this paper, we consider *actor-critic* methods, a large class of methods learning two components, the actor and the critic. The *actor* is a parameterized policy $\pi(a|s;\boldsymbol{\theta})$ of parameters $\boldsymbol{\theta}$. The *critic* is a parameterized value function, either $Q(s,a;\boldsymbol{\omega})$ or $V(s;\boldsymbol{\omega})$, of parameters $\boldsymbol{\omega}$. In this section, we will focus on Q-function critics, but an equivalent discussion using the V-function can be made. The actor is learned by maximizing the expected value of the critic, i.e., by policy improvement

$$\max_{\boldsymbol{\theta}} \mathbb{E}_{\mu_\pi(s),\pi(a|s;\boldsymbol{\theta})}[Q(s,a;\boldsymbol{\omega})]. \tag{2}$$

The critic is learned to estimate the Q-function of the actor, i.e., $Q(s,a;\boldsymbol{\omega}) \approx Q^\pi(s,a)$. This is the so-called policy evaluation, a long-studied problem in reinforcement learning. Many methods learn the critic such that it minimizes the *temporal difference (TD) error*

$$\delta_Q(s,a,s';\boldsymbol{\theta},\boldsymbol{\omega}) = r + \gamma\, \mathbb{E}_{\pi(a'|s';\boldsymbol{\theta})}\big[Q(s',a';\boldsymbol{\omega})\big] - Q(s,a;\boldsymbol{\omega}). \tag{3}$$

The intuition behind TD methods is that if $Q(s,a;\boldsymbol{\omega})$ is the true value function of $\pi(a|s,\boldsymbol{\theta})$, then it satisfies the Bellman equation

$$Q(s,a;\boldsymbol{\omega}) = \mathbb{E}_{\mathcal{P}(s'|s,a),\mathcal{R}(s,a),\pi(a'|s';\boldsymbol{\theta})}\big[r + \gamma Q(s',a';\boldsymbol{\omega})\big], \qquad \forall s \in \mathcal{S}, a \in \mathcal{A}. \tag{4}$$

The TD error is the difference between the two sides of the equation evaluated over a single transition (Lagoudakis and Parr, 2003). *Direct* TD algorithms (Baird, 1995) treat the quantity $Q(s',a';\boldsymbol{\omega})$ as a constant. Their convergence is guaranteed only if linear function approximation is used (Tsitsiklis and Van Roy, 1997), but in practice have achieved impressive results (Mnih et al., 2015). TD(0), one of the most used, at each iteration i updates the critic parameters by

$$\boldsymbol{\omega}_{i+1} = \boldsymbol{\omega}_i + \alpha\, \mathbb{E}_{\mu_\pi(s),\pi(a|s;\boldsymbol{\theta})}\big[\delta_Q(s,a,s';\boldsymbol{\theta}_i,\boldsymbol{\omega}_i)\nabla_{\boldsymbol{\omega}} Q(s,a;\boldsymbol{\omega}_i)\big], \tag{5}$$

where α is the learning rate[1]. This update can be seen as doing gradient descent to solve

$$\min_{\boldsymbol{\omega}} \mathbb{E}_{\mu_\pi(s),\pi(a|s;\boldsymbol{\theta})}\left[\delta_Q(s,a,s';\boldsymbol{\theta},\boldsymbol{\omega})^2\right] \tag{6}$$

where δ_Q has a "fixed" TD target $Q(s',a';\boldsymbol{\omega})$ such that $\nabla_{\boldsymbol{\omega}} Q(s',a';\boldsymbol{\omega}) = 0$.

It is well-known that the actor is guaranteed to converge when the critic accurately estimates the true value function (Sutton et al., 1999). However, it is prohibitively expensive to learn the optimal critic each time the actor is updated. Furthermore, TD methods are guaranteed to converge, i.e., that $\lim_{i\to\infty} Q(s,a;\boldsymbol{\omega}_i) = Q^\pi(s,a)$, only when linear function approximation is used. Yet, a linear parameterization could be inappropriate to learn the optimal policy (Konda and Tsitsiklis, 2000). Nonetheless, critic learning with nonlinear function approximation becomes a non-convex optimization problem with many local solutions. Therefore, we cannot ensure to find the global optimum. For these reasons, actor-critic methods generally alternate between actor updates and critic updates using gradient descent. Despite their appeal, actor-critic methods often suffer from stability issues (Lillicrap et al., 2016; Dai et al., 2017). We argue that one major cause of these issues is that the actor update is based on an inaccurate critic. Using an inaccurate critic, the update directions of the actor can be arbitrarily wrong, causing instability and potentially preventing learning at all. In the next section, we introduce a novel regularization technique on the actor update to address this issue.

3. TD-Regularized Actor-Critic

Recall that the Bellman equation in Eq. (4) depends on both the actor and the critic. In traditional actor-critic, the Bellman equation is only considered during the critic learning. However, the actor update also changes the right hand-side of the Bellman equation, potentially increasing the squared TD error in Eq. (6) Consequently, at the next iteration, the critic will be updated to reduce the squared TD error. Issues arise when the actor updates take a step in the policy parameter space following an inaccurate critic, significantly increasing the squared TD error. In such case, the critic needs to take a large step in the value function parameters space to reduce the TD error. These large steps can cause the critic to change very abruptly, causing instability in learning.

To alleviate the issue, we propose a TD-regularized actor-critic approach that penalizes the updated actor for breaking the Bellman equation, i.e.,

$$\max_{\boldsymbol{\theta}} \mathbb{E}_{\mu_\pi(s),\pi(a|s;\boldsymbol{\theta})}[Q(s,a;\boldsymbol{\omega})] \tag{7}$$

$$\text{subject to } \delta_Q(s,a,s';\boldsymbol{\theta},\boldsymbol{\omega}) = 0, \quad \forall s \in \mathcal{S}, \forall a \in \mathcal{A}. \tag{8}$$

For the true Q^π, the constraint is always satisfied and we can ignore it. Using nonlinear function approximation $Q(s,a;\boldsymbol{\omega})$ for the critic we do not have guarantees of converging to Q^π. Even with linear function approximation, it may be prohibitively expensive to learn the optimal critic each time the actor is updated. Furthermore, even assuming convergence for the critic, the equation holds for the *current* policy π, but not for the policy *we are*

1. The expectation over π stays because it does not require to interact with the environment. It is also possible to use an off-policy approach and learn the Q-function with transitions generated from a different behavior policy, either deterministic (Lillicrap et al., 2016) or stochastic (Precup et al., 2001). For simplicity, in this paper we consider only on-policy learning.

optimizing. Thus, the constraint cannot be ignored. To solve the constrained problem, we introduce to the objective a penalty function replacing the constraint, consisting of a penalty parameter (Lagrangian) multiplied by a measure of violation of the constraints (Boyd and Vandenberghe, 2004). Since Eq. (8) is an equation constraint we use a simple squared penalty. However, Eq. (8) results in infinitely many constraints. Consequently, to keep the problem tractable, we require that the Bellman equation only matches its expected value, approximating the constraint. The resulting optimization problem is

$$\max_{\boldsymbol{\theta}} L(\boldsymbol{\theta}, \lambda) := \mathbb{E}_{\mu_\pi(s),\pi(a|s;\boldsymbol{\theta})}[Q(s,a;\boldsymbol{\omega})] - \lambda\, \mathbb{E}_{\mu_\pi(s),\pi(a|s;\boldsymbol{\theta})}\left[\delta_Q(s,a,s';\boldsymbol{\theta},\boldsymbol{\omega})^2\right], \tag{9}$$

where $L(\boldsymbol{\theta}, \lambda)$ is the Lagrangian function and λ is the Lagrangian multiplier. We name the regularization as *TD-regularization*, since each state/action pair is regularized by its squared TD error. We can also see that the penalty function is the same used to learn the critic (Eq. (6)). This has an important consequence for Eq. (9). Intuition suggests that the larger the penalty coefficient λ, the better the penalty approximates the original problem (Boyd and Vandenberghe, 2004). Depending on how λ is chosen, we have two classes of penalty methods. *Exterior* methods start at optimal but infeasible points and iterate to feasibility as $\lambda \to \infty$. On the contrary, *interior* methods start at feasible but sub-optimal points and iterate to optimality as $\lambda \to 0$. In actor-critic, we usually start at infeasible points, as the critic is not learned and δ_Q is very large. However, unlike classical constrained problems, the constraint changes at each iteration, because the critic is updated to minimize the same penalty function as well. Therefore, as the learning proceeds, δ_Q decreases and, thus, the violation of the constraint is smaller. For these reason, in the experiments, we will choose a small initial λ (as in exterior methods) and decrease at each iteration, as the critic is learned and $\delta_Q \to 0$ (as in interior methods). In the Appendix, we also show an analysis of different values of λ.

3.1 Analysis of the Approach

The resulting algorithm penalizes the actor update for increasing the TD error, thus for reducing the accuracy of the critic. However, we stress that the actor does not take the role of the critic. The minimization of the TD error is just a regularization and not the main objective, which remains the maximization of the value function, i.e., $\max \mathbb{E}[Q(s,a;\boldsymbol{\omega})]$. The TD-regularization rather *encourages* the actor to minimize the TD error. Consequently, the penalty function restrains the actor from introducing a large TD error for the next critic update, effectively preventing the need for large, possibly unstable, updates of critic.

Another important observation regards the effects of the TD-regularization when large TD error occurs. Because the penalty is proportional to the TD error, the higher the TD error is, the larger the regularization will be and, thus, the larger critic update will be. This may seem counterintuitive, as one may expect to update the actor less for higher prediction errors. However, we must distinguish between updates induced by the Q-function versus updates induced by the penalty function. In the case of inaccurate critic, the actor should obviously not trust the critic, i.e., the Q-function. Rather, the actor should wait for the critic to provide more reliable estimates of the value function to avoid bad updates and, potentially, divergence. Thus, it is reasonable that the penalty function affects the actor update more than the critic. Intuitively, at the beginning of the learning the critic provides

an incorrect estimate of the true Q-function to the actor, which can potentially diverge. An example of this behavior is depicted in Figure 1. As the learning proceeds, the TD error decreases and the critic provides reliable estimates. Subsequently, as the TD error decreases, the penalty does as well, and the Q-function affects the actor update more than the penalty function.

4. Applications of TD-Regularization to Existing Actor-Critic Methods

The TD-regularization can be applied to any actor-critic method. In this Section, we show how to apply it to classical policy gradient methods and to state-of-the-art TRPO. We recall that the dependency of μ_π on $\boldsymbol{\theta}$ is ignored in the gradient computation of the policy gradient. Learning μ_π is often very challenging, thus the algorithms presented below learn only π, effectively fixing μ_π to the stationary distribution given by previously collected samples.

4.1 TD-Regularized SGD

Stochastic gradient descent (SGD) solves Eq. (9) by gradient descent. The agent uses a stochastic policy to collect complete trajectories to compute an estimate of the gradient maximizing the expected return. Early policy gradient methods, like REINFORCE (Williams, 1992), used Monte Carlo estimates of Q-values to perform gradient ascent on Eq. (2), i.e., $\widehat{Q}^\pi(s,a) = \sum_{i=1}^{H} \gamma^{i-1} r_i$. Following the actor-critic paradigm it is possible to learn instead a function approximation representing the Q-function. Having a critic, we can then apply the proposed TD-regularization. By using the log-trick and recalling that $Q(s',a';\boldsymbol{\omega})$ depends on $\boldsymbol{\theta}$ because of the expectation over $\pi(a'|s';\boldsymbol{\theta})$, the gradient of the Lagrangian function is

$$\begin{aligned}\nabla_{\boldsymbol{\theta}} L(\boldsymbol{\theta},\lambda) = \mathbb{E}_{\mu_\pi(s),\pi(a|s;\boldsymbol{\theta}),\pi(a'|s';\boldsymbol{\theta})}\left[\nabla_{\boldsymbol{\theta}} \log \pi(a|s;\boldsymbol{\theta})(Q(s,a;\boldsymbol{\omega}) - \delta_Q(s,a,s';\boldsymbol{\theta},\boldsymbol{\omega})^2)\right] \\ + \mathbb{E}_{\mu_\pi(s),\pi(a|s;\boldsymbol{\theta}),\pi(a'|s';\boldsymbol{\theta})}\left[2\gamma\delta_Q(s,a,s';\boldsymbol{\theta},\boldsymbol{\omega})\nabla_{\boldsymbol{\theta}} \log \pi(a'|s';\boldsymbol{\theta}) Q(s',a';\boldsymbol{\omega})\right]. \quad (10)\end{aligned}$$

The first term is the REINFORCE gradient with the squared TD error as baseline. The second term is REINFORCE of the next state/action pair, scaled by the TD error.

4.2 TD-Regularized DPG

Deterministic policy gradient (DPG) (Silver et al., 2014) follows the same approach but with a deterministic policy. Because we do not have the expectation over π, the gradient of the Lagrangian function is

$$\begin{aligned}\nabla_{\boldsymbol{\theta}} L(\boldsymbol{\theta},\lambda) = \mathbb{E}_{\mu_\pi(s)}[\nabla_a Q(s,a;\boldsymbol{\omega})\nabla_{\boldsymbol{\theta}}\pi(s;\boldsymbol{\theta})] \quad (11)\\ + 2\,\mathbb{E}_{\mu_\pi(s)}\left[\gamma\delta_Q(s,a,s';\boldsymbol{\theta},\boldsymbol{\omega})\nabla_{a'} Q(s',a';\boldsymbol{\omega})\nabla_{\boldsymbol{\theta}}\pi(s';\boldsymbol{\theta})\right]. \quad (12)\end{aligned}$$

To collect samples, a behavior policy such as an ϵ-greedy is used. As shown by Silver et al. (2014), DPG can be more advantageous than SGD because deterministic policies have lower variance. On the other hand, the behavior policy has to be chosen appropriately.

4.3 TD-Regularized TRPO

Trust-region policy optimization (TRPO) (Schulman et al., 2015) maximizes an estimate of the advantage function learned using a V-function critic and limits the Kullback-Leibler di-

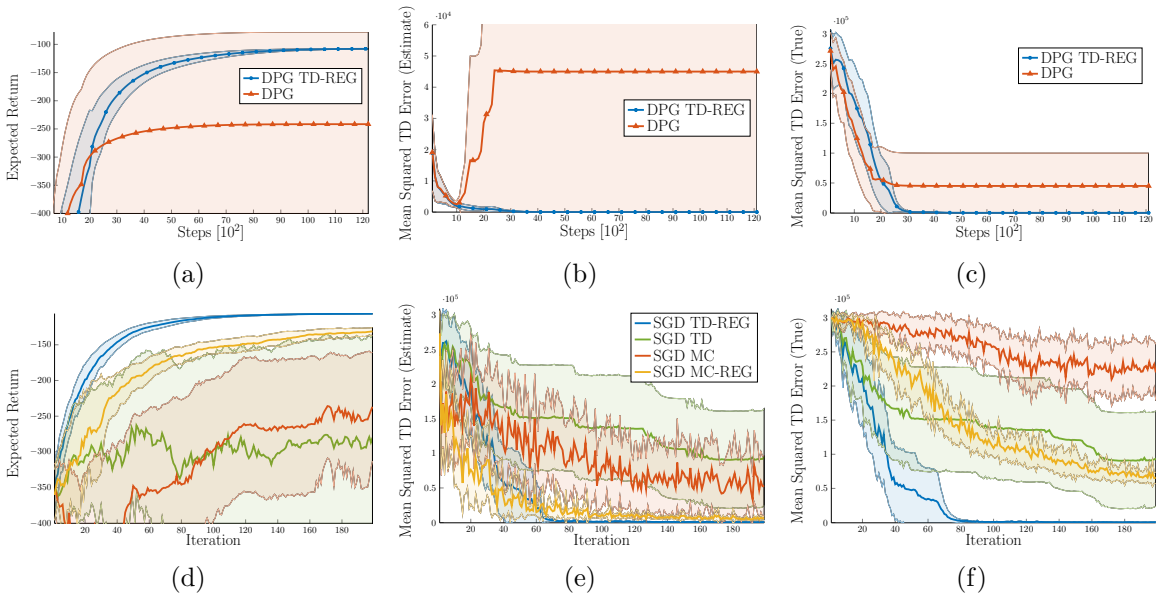


Figure 2: DPG and SGD comparison on the LQR. SGD MC algorithms use Monte Carlo estimates as Q-targets. SGD TD algorithms use TD(0).

vergence between two successive policies. We can extend it by adding the TD-regularization with respect to the V-function critic, resulting in

$$\max_{\boldsymbol{\theta}} \mathbb{E}_{\mu_\pi(s),\pi(a|s;\boldsymbol{\theta})}[A(s,a;\omega)] - \lambda\, \mathbb{E}_{\mu_\pi(s),\pi(a|s;\boldsymbol{\theta})}\left[\delta_V(s,s';\boldsymbol{\omega})^2\right] \tag{13}$$

$$\text{subject to } \mathbb{E}_{\mu_\pi(s)\pi(a|s;\boldsymbol{\theta})}\left[\log \frac{\pi(a|s;\boldsymbol{\theta})}{\pi(a|s;\boldsymbol{\theta}_{\text{old}})}\right] \leq \epsilon, \tag{14}$$

where $A(s,a;\omega) = \sum_{l=0}^{\infty}(\gamma\eta)^l \delta_V(s_t, s_{t+1};\boldsymbol{\omega})$ is the generalized advantage function (Schulman et al., 2016) and $\delta_V(s,s';\boldsymbol{\omega}) = r + \gamma V(s';\omega) - V(s;\omega)$ is the V-function TD error.

5. Evaluation

We evaluate SGD and DPG on the 2D linear-quadratic regulator (LQR). In this domain, we can compute the true Q-function, expected return, and TD error in closed form. Therefore, we show both the TD error estimated from samples (used for learning) and the true one. The Q-function approximation is $Q(s,a;\boldsymbol{\omega}) = \phi(s,a)^\mathsf{T}\boldsymbol{\omega}$, where $\phi(s,a)$ are polynomial features of degree three. However, quadratic features are sufficient to represent the true Q-function, thus the critic is prone to overfit. Results are averaged over 20 trials. TRPO is instead evaluated on OpenAI Gym (Brockman et al., 2016) continuous control tasks with MuJoCo (Todorov et al., 2012) physics. The Q-function is approximated by a two-layer neural network of 200 neurons each with hyperbolic tangent activation. Results are averaged over five trials. In all plots, shaded area denotes the standard error.

For the LQR, since the critic is prone to overfit, the initial TD error is very large, as shown in Figure 2c. Furthermore, the true TD error (Figure 2c) is much larger than the one estimated by the critic (Figure 2b), meaning that the critic underestimates the true TD

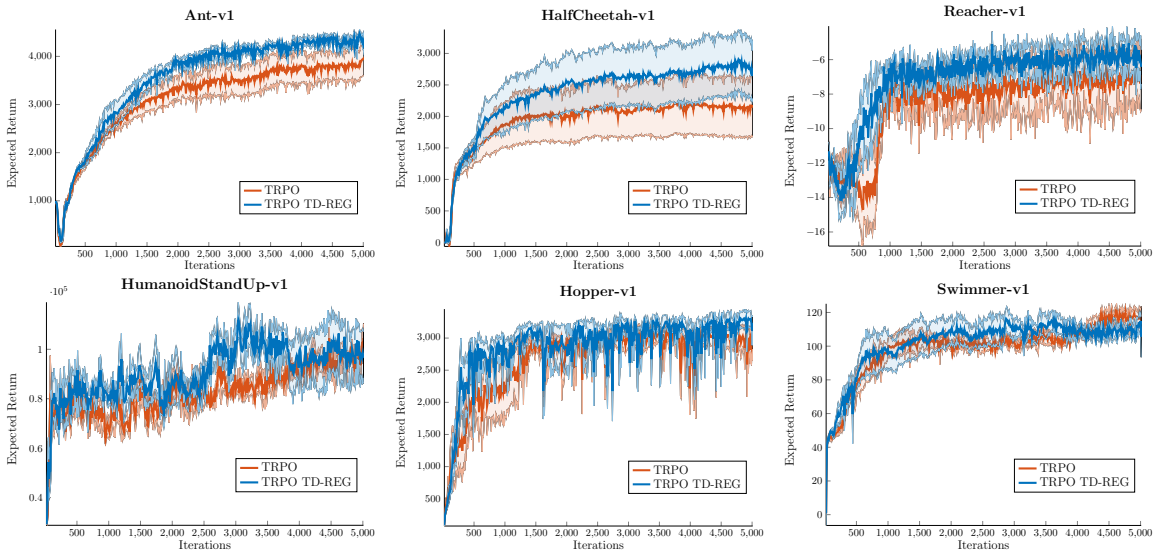


Figure 3: TRPO comparison on OpenAI Gym continuous control tasks.

error. Thus, initially the critic provides incorrect update steps to the actor. Consequently, non-regularized algorithms often diverge. On the contrary, TD-regularized never failed.

Figure 3 shows the results for TRPO. On Ant, HalfCheetah and Reacher, TRPO TD-REG clearly outperforms its non-regularized counterpart, as it converges faster to better policies. On HumanoidStandup, Swimmer and Hopper the performance is comparable.

6. Conclusion

Actor-critic methods often suffer from instability. A major cause is the function approximation error in the critic. In this paper, we addressed the stability issue taking into account the relationship between the critic and the actor. We presented a TD-regularized approach penalizing the actor for breaking the critic Bellman equation, in order to perform policy updates producing small changes in the critic. We presented practical implementations of our approach and showed that our TD-regularization allows for more stable updates, resulting in policy updates that are less likely to diverge.

Our method opens several avenues of research. In this paper, we only focused on direct TD methods. In future work, we will extend the regularization to residual methods, as they have stronger convergence guarantees even when nonlinear function approximation is used to learn the critic (Baird, 1995). We will also investigate different techniques to solve the constrained optimization problem with the Bellman equation constraint. We will focus on different penalty functions and methods for optimizing the Lagrangian λ, as presented for instance in relative entropy policy search (Peters et al., 2010). Furthermore, we will study equivalent formulations of the constrained problem with stronger guarantees. For instance, the approximation introduced by the expectation over the Bellman equation constraint could be addressed by using the representation theorem. Finally, it would be interesting to study the convergence of actor-critic methods with TD-regularization, including cases with tabular and linear function approximation, where convergence guarantees are available.

References

Joshua Achiam, David Held, Aviv Tamar, and Pieter Abbeel. Constrained policy optimization. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2017.

Riad Akrour, Abbas Abdolmaleki, Hany Abdulsamad, and Gerhard Neumann. Model-Free trajectory optimization for reinforcement learning. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2016.

Leemon Baird. Residual algorithms: Reinforcement learning with function approximation. In *Proceedings of the International Conference on Machine learning (ICML)*, 1995.

Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, New York, NY, USA, 2004. ISBN 0521833787.

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

Bo Dai, Albert Shaw, Niao He, Lihong Li, and Le Song. Boosting the actor with dual critic. *arXiv preprint arXiv:1712.10282*, 2017.

Audrunas Gruslys, Mohammad Gheshlaghi Azar, Marc G. Bellemare, and Remi Munos. The reactor: A fast and sample-efficient actor-critic agent for reinforcement learning. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2018.

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft Actor-Critic: Off-Policy maximum entropy deep reinforcement learning with a stochastic actor. *arXiv preprint arXiv:1801.01290*, 2018.

Matteo Hessel, Joseph Modayil, Hado Van Hasselt, Tom Schaul, Georg Ostrovski, Will Dabney, Dan Horgan, and David Silver. Rainbow: Combining improvements in deep reinforcement learning. In *Proceedings of the Conference on Artificial Intelligence (AAAI)*, 2018.

Vijay R Konda and John N Tsitsiklis. Actor-critic algorithms. In *Advances in Neural Information Processing Systems (NIPS)*, 2000.

Michail G Lagoudakis and Ronald Parr. Least-squares policy iteration. *Journal of Machine Learning Research (JMLR)*, 4:1107–1149, 2003.

Timothy P Lillicrap, Jonathan J Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2016.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015.

Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous methods for

deep reinforcement learning. In *International Conference on Machine Learning (ICML)*, 2016.

Rmi Munos, Tom Stepleton, Anna Harutyunyan, and Marc G. Bellemare. Safe and efficient off-policy reinforcement learning. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2016.

Ofir Nachum, Mohammad Norouzi, Kelvin Xu, and Dale Schuurmans. Trust-PCL: An Off-Policy trust region method for continuous control. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2018.

J. Peters, K. Muelling, and Y. Altun. Relative entropy policy search. In *Proceedings of the Conference on Artificial Intelligence (AAAI)*, 2010.

Jan Peters and Stefan Schaal. Natural actor-critic. *Neurocomputing*, 71(7):1180–1190, 2008.

Doina Precup, Richard S Sutton, and Sanjoy Dasgupta. Off-policy temporal-difference learning with function approximation. In *Proceedings of the International Conference on Machine learning (ICML)*, 2001.

John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2015.

John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. High-dimensional continuous control using generalized advantage estimation. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2016.

David Silver, Guy Lever, Nicolas Heess, Thomas Degris, Daan Wierstra, Martin Riedmiller, et al. Deterministic policy gradient algorithms. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2014.

David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, Sander Dieleman1, Dominik Grewe1, John Nham, Nal Kalchbrenner1, Ilya Sutskever, Timothy Lillicrap, Madeleine Leach, Koray Kavukcuoglu, Thore Graepel1, and Demis Hassabis. Mastering the game of go with deep neural networks and tree search. *Nature*, 529(7587):484–489, 2016.

Richard S. Sutton, David A. McAllester, Satinder P. Singh, and Yishay Mansour. Policy gradient methods for reinforcement learning with function approximation. In *Advances in Neural Information Processing Systems (NIPS)*, 1999.

Emanuel Todorov, Tom Erez, and Yuval Tassa. MuJoCo: A physics engine for model-based control. In *Proceedings of the International Conference on Intelligent Robots and Systems (IROS)*, 2012.

John N Tsitsiklis and Benjamin Van Roy. Analysis of temporal-difference learning with function approximation. In *Advances in Neural Information Processing Systems (NIPS)*, 1997.

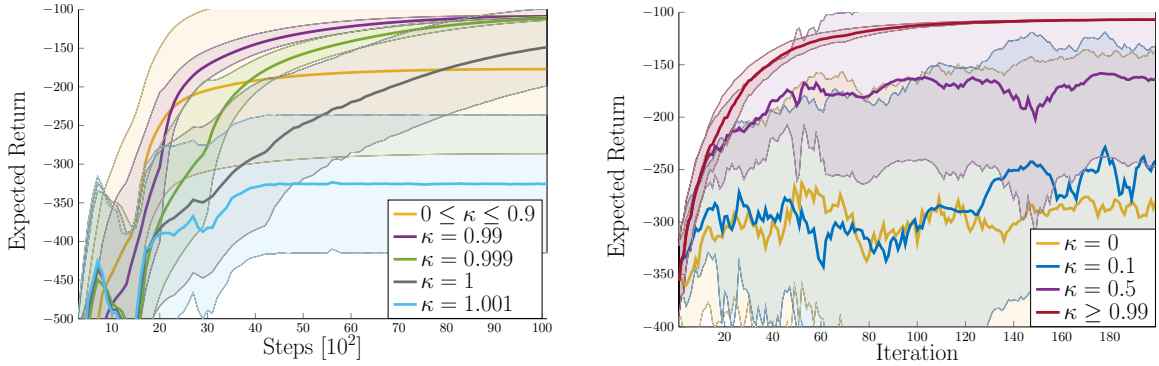
Ronald J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8(3-4):229–256, May 1992.

Appendix A. Analysis of the TD-Regularization Coefficient λ

In Section 3 we have discussed that Eq. (9) is the result of solving a constrained optimization problem with penalty function methods. In optimization, we can distinguish two approaches to apply penalty functions (Boyd and Vandenberghe, 2004). *Exterior* penalty methods start at optimal but infeasible points and iterate to feasibility as $\lambda \to \infty$. On the contrary, *interior* penalty methods start at feasible but sub-optimal points and iterate to optimality as $\lambda \to 0$. In actor-critic, we usually start at infeasible points, as the critic is not learned and the TD error is very large. However, unlike classical constrained problems, the constraint changes at each iteration, because the critic is updated to minimize the same penalty function as well. This trend emerged from the previous experiments, as shown by the mean squared TD error trends in Figure 2.

In this section, we provide a comparison of different values of κ on DPG TD-REG and SPG TD-REG on the LQR. We test also κ as growth factor, as in exterior penalty methods. In all experiments we start at $\lambda_0 = 0.1$ and we test the following κ: 0 (regularization applied only for the first update), 0.1, 0.5, 0.9, 0.99, 0.999, 1, 1.001.

As shown in Figure 4, the TD-regularization always succeeded only with $0.99 \leq \kappa < 1$, i.e., when κ is a decaying factor (as in interior penalty methods) and it is large enough such that the regularization is always in effect. On the contrary, smaller values of κ had λ decay too quickly. Consequently, the TD-regularization was not in effect and the learning still failed in some trials. Using κ as growth factor (as in exterior penalty methods) even harms the learning of DPG. This is expected, since by increasing λ we are also increasing the magnitude of the gradient, which then leads to large and unstable updates. On the contrary, a growing λ is not harmful for SPG because the algorithm normalizes the gradient.



(a) DPG (b) SPG

Figure 4: Comparison of different values of κ. Shaded area denotes the standard error. Only $0.99 \leq \kappa < 1$ allowed the TD-regularization to stably learn in all trials. Smaller values do not provide enough regularization. Larger values harm the learning for DPG, but not for SPG because the latter uses normalized gradients.