Classification-based Policy Iteration with a Critic

Victor Gabillon VICTOR.GABILLON@INRIA.FR
Alessandro Lazaric ALESSANDRO.LAZARIC@INRIA.FR
Mohammad Ghavamzadeh MOHAMMAD.GHAVAMZADEH@INRIA.FR
INRIA Lille - Nord Europe, Team SequeL, FRANCE

Bruno Scherrer BRUNO.SCHERRER@INRIA.FR
INRIA Nancy - Grand Est, Team Maia, FRANCE

Abstract

In this paper, we study the effect of adding a value function approximation component (critic) to rollout classification-based policy iteration (RCPI) algorithms. The idea is to use a critic to approximate the return after we truncate the rollout trajectories. This allows us to control the bias and variance of the rollout estimates of the action-value function. Therefore, the introduction of a critic can improve the accuracy of the rollout estimates, and as a result, enhance the performance of the RCPI algorithm. We present a new RCPI algorithm, called *direct policy iteration with critic* (DPI-Critic), and provide its finite-sample analysis when the critic is based on the LSTD method. We empirically evaluate the performance of DPI-Critic and compare it with DPI and LSPI in two benchmark reinforcement learning problems.

1. Introduction

Policy iteration is a method of computing an optimal policy for any given Markov deci- sion process (MDP). It is an iterative procedure that discovers a deterministic optimal policy by generating a sequence of monotonically improving policies. Each iteration k of this algorithm consists of two phases: *policy evaluation* in which the action-value function Q^{π_k} of the current policy π_k is computed, and *policy improvement* in which the new (improved) policy π_{k+1} is generated as the greedy policy w.r.t. Q^{π_k}, i.e., $\pi_{k+1}(x) = \arg\max_{a\in\mathcal{A}} Q^{\pi_k}(x,a)$. Unfortunately, in MDPs with large (or continuous) state and/or action spaces, the policy evaluation problem cannot be solved exactly and approximation techniques are required. There have been two main approaches to deal with this issue in the literature. The most common approach is to find a good approximation of the action-value function of π_k in a real-valued function space (see e.g., Lagoudakis & Parr 2003a). The second approach **1)** replaces the policy evaluation step (approximating the action-value function over the entire state-action space) with computing rollout estimates of Q^π over a finite number of states $\mathcal{D} = \{x_i\}_{i=1}^N$, called the *rollout set*, and the entire action space, and **2)** casts the policy improvement step as a *classification* problem to find a policy in a given hypothesis space that best predicts the greedy action at every state (see e.g., Lagoudakis & Parr 2003b; Fern et al. 2004; Lazaric et al. 2010a). Although whether selecting a suitable policy space is any easier than a value function space is highly debatable, it may be argued that *classification-based API* methods can be advantageous in problems where good policies are easier to represent and learn than their value functions.

As it is suggested by both theoretical and empirical analysis, the performance of the classification-based API algorithms is closely related to the accuracy in estimating the greedy action at each state of the rollout set, which itself depends on the accuracy of the rollout estimates of the action-values. Thus, it is quite important to balance the bias and variance of the rollout estimates, $\widehat{Q}^\pi$'s, that both depend on the length H of the rollout trajectories. While the bias in $\widehat{Q}^\pi$, i.e., the difference between $\widehat{Q}^\pi$ and the actual Q^π, decreases as H becomes larger, its variance (due to stochastic MDP transitions and rewards) increases with the value of H. Although the bias and variance of $\widehat{Q}^\pi$ estimates may be optimized by the value of H, when the *budget*, i.e., the number of calls to the generative model, is limited, it may not be possible to find an H that guarantees an accurate enough training set.

A possible approach to address this problem is to introduce a *critic* that provides an approximation of the

Appearing in *Proceedings of the 28^{th} International Conference on Machine Learning*, Bellevue, WA, USA, 2011.

value function. In this approach, we define each $\widehat{Q}^\pi$ estimate as the average of the values returned by H-horizon rollouts plus the critic's prediction of the return from the time step H on. This allows us to use small values of H, thus having a small estimation variance, and at the same time, to rely on the value function approximation provided by the critic to control the bias. The idea is similar to actor-critic methods (Barto et al., 1983) in which the variance of the gradient estimates in the actor is reduced using the critic's prediction of the value function.

In this paper, we introduce a new classification-based API algorithm, called *DPI-Critic*, obtained by adding a critic to the *direct policy iteration* (DPI) algorithm (Lazaric et al., 2010a). We provide finite-sample analysis for DPI-Critic when the critic approximates the value function using least-squares temporal-difference (LSTD) learning (Bradtke & Barto, 1996).[1] We empirically evaluate the performance of DPI-Critic and compare it with DPI and LSPI (Lagoudakis & Parr, 2003a) on two benchmark reinforcement learning (RL) problems: mountain car and inverted pendulum. The results indicate that DPI-Critic can take advantage of both its components and improve over DPI and LSPI.

2. Preliminaries

In this section we set the notation used throughout the paper. For a measurable space with domain $\mathcal{X}$, we let $\mathcal{S}(\mathcal{X})$ and $\mathcal{B}(\mathcal{X}; L)$ denote the set of probability measures over $\mathcal{X}$, and the space of bounded measurable functions with domain $\mathcal{X}$ and bound $0 < L < \infty$, respectively. For a measure $\rho \in \mathcal{S}(\mathcal{X})$ and a measurable function $f : \mathcal{X} \to \mathbb{R}$, we define the $\ell_p(\rho)$-norm of f as $||f||_{p,\rho}^p = \int |f(x)|^p \rho(dx)$. We consider the standard RL framework (Sutton & Barto, 1998) in which a learning agent interacts with a stochastic environment and this interaction is modeled as a discrete-time MDP. A discounted MDP is a tuple $\mathcal{M} = \langle \mathcal{X}, \mathcal{A}, r, p, \gamma \rangle$, where the state space $\mathcal{X}$ is a subset of a Euclidean space $\mathbb{R}^d$, the set of actions $\mathcal{A}$ is finite ($|\mathcal{A}| < \infty$), the reward function $r : \mathcal{X} \times \mathcal{A} \to \mathbb{R}$ is uniformly bounded by $R_{\max}$, the transition model $p(\cdot|x, a)$ is a distribution over $\mathcal{X}$, and $\gamma \in (0, 1)$ is a discount factor. We define deterministic policies as the mapping $\pi : \mathcal{X} \to \mathcal{A}$. The value function of a policy π, V^π, is the unique fixed-point of the Bellman operator $\mathcal{T}^\pi : \mathcal{B}(\mathcal{X}; V_{\max}) \to \mathcal{B}(\mathcal{X}; V_{\max})$ defined by

$$(\mathcal{T}^\pi V)(x) = r\big(x, \pi(x)\big) + \gamma \int_{\mathcal{X}} p\big(dy|x, \pi(x)\big) V(y) ,$$

while the action-value function Q^π is defined as

$$Q^\pi(x, a) = r(x, a) + \gamma \int_{\mathcal{X}} p(dy|x, a) V^\pi(y) .$$

Since the rewards are bounded by $R_{\max}$, all values and action-values are bounded by $q = \frac{R_{\max}}{1-\gamma}$. A policy π is greedy w.r.t. an action-value function Q, if $\pi(x) \in \arg\max_{a\in\mathcal{A}} Q(x, a), \ \forall x \in \mathcal{X}$.

```
Input: policy space $\Pi$, state distribution $\rho$
Initialize: Let $\pi_0 \in \Pi$ be an arbitrary policy
for $k = 0, 1, 2, \ldots$ do
  Construct the rollout set $\mathcal{D}_k = \{x_i\}_{i=1}^N,\ x_i \overset{\text{iid}}{\sim} \rho$
  • Critic:
  Construct the set $S_k$ of $n$ samples (e.g., by following
  a trajectory or by using the generative model)
  $\widehat{V}^{\pi_k} \leftarrow$ VF-APPROX($S_k$)                    (critic)
  • Rollout:
  for all states $x_i \in \mathcal{D}_k$ and actions $a \in \mathcal{A}$ do
    for $j = 1$ to $M$ do
      Perform a rollout and return $R_j(x_i, a)$
    end for
    $\widehat{Q}^{\pi_k}(x_i, a) = \frac{1}{M}\sum_{j=1}^M R_j(x_i, a)$
  end for
  $\pi_{k+1} = \arg\min_{\pi\in\Pi} \widehat{\mathcal{L}}_{\pi_k}(\widehat{\rho}; \pi)$          (classifier)
end for
```

Figure 1. The pseudo-code of the DPI-Critic algorithm.

To approximate value functions, we use a linear approximation architecture with parameters $\alpha \in \mathbb{R}^d$ and basis functions $\varphi_j \in \mathcal{B}(\mathcal{X}; L),\ j = 1, \ldots, d$. We denote by $\phi : \mathcal{X} \to \mathbb{R}^d,\ \phi(\cdot) = \big(\varphi_1(\cdot), \ldots, \varphi_d(\cdot)\big)^\top$ the feature vector, and by $\mathcal{F}$ the linear function space spanned by the features φ_j, i.e., $\mathcal{F} = \{f_\alpha(\cdot) = \phi(\cdot)^\top \alpha : \alpha \in \mathbb{R}^d\}$. Finally, we define the Gram matrix $G \in \mathbb{R}^{d\times d}$ w.r.t. a distribution $\rho \in \mathcal{S}(\mathcal{X})$ as

$$G_{ij} = \int \varphi_i(x)\varphi_j(x)\rho(dx), \qquad i, j = 1, \ldots, d .$$

3. The DPI-Critic Algorithm

In this section, we outline the algorithm we propose in this paper, called Direct Policy Iteration with Critic (DPI-Critic), which is an extension of the DPI algorithm (Lazaric et al., 2010a) by adding a critic. As illustrated in Fig. 1, DPI-Critic starts with an arbitrary initial policy $\pi_0 \in \Pi$. At each iteration k, we build a set of n samples S_k, called the *critic training set*. The critic uses S_k in order to compute $\widehat{V}^{\pi_k}$, an approximation of the value function of the current policy π_k. Then, a new policy π_{k+1} is computed from π_k, as the best approximation of the greedy policy w.r.t. Q^{π_k}, by solving a cost-sensitive classification problem. Similar to DPI, DPI-Critic is based on the following *loss function* and *expected error* :

$$\ell_{\pi_k}(x; \pi) = \max_{a\in\mathcal{A}} Q^{\pi_k}(x, a) - Q^{\pi_k}\big(x, \pi(x)\big), \qquad \forall x \in \mathcal{X},$$

$$\mathcal{L}_{\pi_k}(\rho; \pi) = \int_{\mathcal{X}} \Big[\max_{a\in\mathcal{A}} Q^{\pi_k}(x, a) - Q^{\pi_k}\big(x, \pi(x)\big)\Big] \rho(dx) .$$

[1] The finite-sample analysis of DPI-Critic with Bellman residual minimization is available at Gabillon et al. (2011).

In order to minimize this loss, a *rollout set* $\mathcal{D}_k$ is built by sampling N states i.i.d. from a distribution ρ. For each state $x_i \in \mathcal{D}_k$ and each action $a \in \mathcal{A}$, M independent estimates $\{R_j^{\pi_k}(x_i, a)\}_{j=1}^M$ are computed, where

$$R_j^{\pi_k}(x_i,a) = R_j^{\pi_k,H}(x_i,a) + \gamma^H \widehat{V}^{\pi_k}(x_{i,j}^H) \;, \qquad (1)$$

in which $R_j^{\pi_k,H}(x_i,a)$ is the outcome of an H-horizon rollout, i.e.,

$$R_j^{\pi_k,H}(x_i,a) = r(x_i,a) + \sum_{t=1}^{H-1} \gamma^t r(x_{i,j}^t, \pi_k(x_{i,j}^t)) \;, \qquad (2)$$

and $\widehat{V}^{\pi_k}(x_{i,j}^H)$ is the critic's estimate of the value function at state $x_{i,j}^H$. In Eq. 2, $(x_i, x_{i,j}^1, x_{i,j}^2, \ldots, x_{i,j}^H)$ is the trajectory induced by taking action a at state x_i and following the policy π_k afterwards, i.e., $x_{i,j}^1 \sim p(\cdot|x_i,a)$ and $x_{i,j}^t \sim p\big(\cdot\,|x_{i,j}^{t-1}, \pi_k(x_{i,j}^{t-1})\big)$ for $t \geq 2$. An estimate of the action-value function of the policy π_k is then obtained by averaging the M estimates as

$$\widehat{Q}^{\pi_k}(x_i,a) = \frac{1}{M}\sum_{j=1}^{M} R_j^{\pi_k}(x_i,a) \;. \qquad (3)$$

Given the action-value function estimates, the *empirical loss* and *empirical error* are defined as

$$\widehat{\ell}_{\pi_k}(x;\pi) = \max_{a\in\mathcal{A}} \widehat{Q}^{\pi_k}(x,a) - \widehat{Q}^{\pi_k}\big(x,\pi(x)\big), \quad \forall x \in \mathcal{X},$$

$$\widehat{\mathcal{L}}_{\pi_k}(\widehat{\rho};\pi) = \frac{1}{N}\sum_{i=1}^{N}\Big[\max_{a\in\mathcal{A}} \widehat{Q}^{\pi_k}(x_i,a) - \widehat{Q}^{\pi_k}(x_i,\pi(x_i))\Big]. \quad (4)$$

Finally, DPI-Critic makes use of a classifier which solves a multi-class cost-sensitive classification problem and returns a policy that minimizes the empirical error $\widehat{\mathcal{L}}_{\pi_k}(\widehat{\rho};\pi)$ over the policy space Π.

As it can be seen from Eq. 1, the main difference between DPI-Critic and DPI is that after H steps DPI rollouts are truncated and the return thereafter is implicitly set to 0, while in DPI-Critic an approximation of the value function learned by the critic is used to predict this return. Hence, with a fixed horizon H, even if the critic is a rough approximation of the value function, whenever its accuracy is higher than the implicit prediction of 0 in DPI, the rollouts in DPI-Critic are expected to be more accurate than those in DPI. Similarly, we expect DPI-Critic to obtain the same accuracy as DPI with a shorter horizon, and as a result, a smaller number of interactions with the generative model. In fact, while in DPI decreasing H leads to a smaller variance and a larger bias, in DPI-Critic the increase in the bias is controlled by the critic. Finally, it is worth noting that DPI-Critic still benefits from the advantages of the classification-based approach to policy iteration compared to value-function-based API algorithms such as LSPI. This is due to the fact that DPI-Critic still relies on approximating the policy improvement step, and thus similar to DPI, whenever approximating good policies is easier than their value functions, DPI-Critic is expected to perform better than its value-function-based counterparts. Furthermore, while DPI-Critic only needs a rough approximation of the value function at certain states, value-function-based API methods, like LSPI, need an accurate approximation of the action-value function over the entire state-action space, and thus they usually require more samples than the critic in DPI-Critic.

4. Theoretical analysis

In this section, we provide a finite-sample analysis of the error incurred at each iteration of DPI-Critic. The full analysis of the propagation is reported in Gabillon et al. (2011).

In order to use the existing finite-sample bounds for pathwise-LSTD (Lazaric et al., 2010b), we introduce the following assumptions.

Assumption 1. *At each iteration k of DPI-Critic, the critic uses a linear function space $\mathcal{F}$ spanned by d bounded basis functions (see Section 2). A data-set $S_k = \{(X_i, R_i)\}_{i=1}^n$ is built, where X_i's are obtained by following a single trajectory generated by a stationary β-mixing process with parameters $\hat{\beta}, b, \kappa$, and a stationary distribution σ_k equal to the stationary distribution of the Markov chain induced by policy π_k, and $R_i = r\big(X_i, \pi_k(X_i)\big)$.*

Assumption 2. *The rollout set sampling distribution ρ is such that for any policy $\pi \in \Pi$ and any action $a \in \mathcal{A}$, $\mu = \rho P^a (P^\pi)^{H-1} \leq C\sigma$, where $C < \infty$ is a constant and σ is the stationary distribution of π. The distribution μ is the distribution induced by starting at a state sampled from ρ, taking action a, and then following policy π for $H-1$ steps.*

Before stating the main results of this section, Lemma 1 and Theorem 1, we report the performance bound for pathwise-LSTD as in Lazaric et al. (2010b). Since all the following statements are true for any iteration k, in order to simplify the notation, we drop the dependency of all the variables on k.

Proposition 1 (Thm. 5 in Lazaric et al. 2010b)**.** *Let n be the number of samples collected as in Assumption 1 and $\widehat{V}^\pi$ be the approximation of the value function of policy π returned by pathwise-LSTD truncated in the range $[-q, q]$. Then for any $\delta > 0$, we have*

$$||V^\pi - \widehat{V}^\pi||_{2,\sigma} \leq \epsilon_{LSTD} =$$

$$\Bigg[\frac{2}{\sqrt{1-\gamma^2}}\big(2\sqrt{2}\inf_{f\in\mathcal{F}}||V^\pi - f||_{2,\sigma} + \mathcal{E}_2\big) + \frac{2}{1-\gamma}\Big(\gamma q L\sqrt{\frac{8d}{\omega}}\big(\sqrt{\frac{8\log(32|\mathcal{A}|d/\delta)}{n}} + \frac{1}{n}\big)\Big) + \mathcal{E}_1\Bigg]$$

with probability $1-\delta$ *(w.r.t. the samples in* S*), where*

(1) $\mathcal{E}_1 = 24q\sqrt{\frac{2\Lambda_1(n,d,\delta/4)}{n}\max\{\frac{\Lambda_1(n,d,\delta/4)}{b},1\}^{1/\kappa}}$, *in which* $\Lambda_1(n,d,\delta) = 2(d+1)\log n + \log\frac{e}{\delta} + \log^{+}(\max\{18(6e)^{2(d+1)},\hat{\beta}\})$,
(2) $\mathcal{E}_2 = 12(q+L||\alpha^*||)\sqrt{\frac{2\Lambda_2(n,\delta/4)}{n}\max\{\frac{\Lambda_2(n,\delta/4)}{b},1\}^{1/\kappa}}$, *in which* $\Lambda_2(n,\delta) = \log\frac{e}{\delta} + \log(\max\{6,n\hat{\beta}\})$ *and* $\alpha^* = \arg\min_{\alpha\in\mathbb{R}^d}||V^{\pi_k} - f_{\alpha^*}||_{2,\sigma}$,
(3) $\omega > 0$ *is the smallest strictly positive eigenvalue of the Gram matrix w.r.t. the distribution* σ*.*

In the following lemma, we derive a bound for the difference between the actual action-value function of policy π and its estimate computed by DPI-Critic.

Lemma 1. *Let Assumptions 1 and 2 hold and* $\mathcal{D} = \{x_i\}_{i=1}^N$ *be the rollout set with* $x_i \overset{iid}{\sim} \rho$*. Let* Q^π *be the true action-value function of policy* π *and* $\widehat{Q}^\pi$ *be its estimate computed by DPI-Critic using* M *rollouts with horizon* H *(Eqs. 1–3). Then for any* $\delta > 0$

$$\max_{a\in\mathcal{A}}\Big|\frac{1}{N}\sum_{i=1}^{N}\big[Q^\pi(x_i,a) - \widehat{Q}^\pi(x_i,a)\big]\Big| \le \epsilon_1+\epsilon_2+\epsilon_3+\epsilon_4,$$

with probability $1-\delta$ *(w.r.t. the rollout estimates and the samples in the critic training set* S*), where*

$$\epsilon_1 = (1-\gamma^H)q\sqrt{\frac{2\log(4|\mathcal{A}|/\delta)}{MN}},\quad \epsilon_2 = \gamma^H q\sqrt{\frac{2\log(4|\mathcal{A}|/\delta)}{MN}},$$

$$\epsilon_3 = 24\gamma^H q\sqrt{\frac{2\Lambda(N,d,\frac{\delta}{4|\mathcal{A}|M})}{N}},\quad \epsilon_4 = 2\gamma^H\sqrt{C}\,\epsilon_{LSTD}\,,$$

with $\Lambda(N,d,\delta) = \log\big(\frac{9e}{\delta}(12Ne)^{2(d+1)}\big)$.

Proof. We prove the following series of inequalities:

$$\Big|\frac{1}{N}\sum_{i=1}^{N}\big[Q^\pi(x_i,a) - \widehat{Q}^\pi(x_i,a)\big]\Big|$$

$$\overset{(a)}{=}\Big|\frac{1}{MN}\sum_{i=1}^{N}\sum_{j=1}^{M}\big[Q^\pi(x_i,a) - R_j^\pi(x_i,a)\big]\Big|$$

$$\overset{(b)}{\le}\Big|\frac{1}{MN}\sum_{i=1}^{N}\sum_{j=1}^{M}\big[Q_H^\pi(x_i,a) - R_j^{\pi,H}(x_i,a)\big]\Big|$$

$$+\Big|\frac{\gamma^H}{MN}\sum_{i=1}^{N}\sum_{j=1}^{M}\big[\widehat{V}^\pi(x_{i,j}^H) - \mathbf{E}_{x\sim\nu_i}[V^\pi(x)]\big]\Big|$$

$$\overset{(c)}{\le}\epsilon_1 + \Big|\frac{\gamma^H}{MN}\sum_{i=1}^{N}\sum_{j=1}^{M}\big[\widehat{V}^\pi(x_{i,j}^H) - V^\pi(x_{i,j}^H)\big]\Big|$$

$$+\Big|\frac{\gamma^H}{MN}\sum_{i=1}^{N}\sum_{j=1}^{M}\big[V^\pi(x_{i,j}^H) - \mathbf{E}_{x\sim\nu_i}[V^\pi(x)]\big]\Big| \qquad \text{w.p. } 1-\delta'$$

$$\overset{(d)}{\le}\epsilon_1+\epsilon_2+\frac{\gamma^H}{M}\sum_{j=1}^{M}||V^\pi - \widehat{V}^\pi||_{1,\widehat{\mu}_j} \qquad \text{w.p. } 1-2\delta'$$

$$\overset{(e)}{\le}\epsilon_1+\epsilon_2+\frac{\gamma^H}{M}\sum_{j=1}^{M}||V^\pi - \widehat{V}^\pi||_{2,\widehat{\mu}_j} \qquad \text{w.p. } 1-2\delta'$$

$$\overset{(f)}{\le}\epsilon_1+\epsilon_2+\epsilon_3+2\gamma^H||V^\pi - \widehat{V}^\pi||_{2,\mu} \qquad \text{w.p. } 1-3\delta'$$

$$\overset{(g)}{\le}\epsilon_1+\epsilon_2+\epsilon_3+2\gamma^H\sqrt{C}||V^\pi - \widehat{V}^\pi||_{2,\sigma}$$

$$\overset{(h)}{\le}\epsilon_1+\epsilon_2+\epsilon_3+2\gamma^H\sqrt{C}\,\epsilon_{LSTD} \qquad \text{w.p. } 1-4\delta'$$

The statement of the lemma is obtained by setting $\delta' = \delta/4$ and taking a union bound over actions.

(a) We use Eq. 3 to replace $\widehat{Q}^\pi(x_i,a)$.
(b) We replace $R_j^\pi(x_i,a)$ from Eq. 1 and use the fact that $Q^\pi(x_i,a) = Q_H^\pi(x_i,a) + \gamma^H\mathbf{E}_{x\sim\nu_i}[V^\pi(x)]$, where $Q_H^\pi(x_i,a) = \mathbf{E}\big[r(x_i,a) + \sum_{t=1}^{H-1}\gamma^t r\big(x_i^t,\pi(x_i^t)\big)\big]$ and $\nu_i = \delta(x_i)P^a(P^\pi)^{H-1}$ is the distribution over states induced by starting at state x_i, taking action a, and then following the policy π for $H-1$ steps. We split the sum using the triangle inequality.
(c) Using the Chernoff-Hoeffding inequality, with probability $1-\delta'$ (w.r.t. the samples used to build the rollout estimates), we have

$$\Big|\frac{1}{MN}\sum_{i=1}^{N}\sum_{j=1}^{M}\big[Q_H^\pi(x_i,a) - R_j^{\pi,H}(x_i,a)\big]\Big| \le \epsilon_1$$

$$= (1-\gamma^H)q\sqrt{\frac{2\log(1/\delta')}{MN}}\,.$$

(d) Using the Chernoff-Hoeffding inequality, with probability $1-\delta'$ (w.r.t. the last state reached by the rollout trajectories), we have

$$\Big|\frac{\gamma^H}{MN}\sum_{i=1}^{N}\sum_{j=1}^{M}\big[V^\pi(x_{i,j}^H) - \mathbf{E}_{x\sim\nu_i}[V^\pi(x)]\big]\Big| \le \epsilon_2$$

$$= \gamma^H q\sqrt{\frac{2\log(1/\delta')}{MN}}\,.$$

We also use the definition of empirical ℓ_1-norm and replace the second term with $||V^\pi - \widehat{V}^\pi||_{1,\widehat{\mu}_j}$, where $\widehat{\mu}_j$ is the empirical distribution corresponding to the distribution $\mu = \rho P^a(P^\pi)^{H-1}$. In fact for any $1 \le j \le M$, samples $x_{i,j}^H$ are i.i.d. from μ.
(e) We move from ℓ_1-norm to ℓ_2-norm using the Cauchy-Schwarz inequality.
(f) Note that $\widehat{V}$ is a random variable independent from the samples used to build the rollout estimates. Using Corollary 12 in (Lazaric et al., 2010b), we have

$$||V^\pi - \widehat{V}^\pi||_{2,\widehat{\mu}_j} \le 2||V^\pi - \widehat{V}^\pi||_{2,\mu} + \epsilon_3(\delta'')$$

with probability $1-\delta''$ (w.r.t. the samples in $\widehat{\mu}_j$) for any j, and $\epsilon_3(\delta'') = 24q\sqrt{\frac{2\Lambda(N,d,\delta'')}{N}}$. By taking a union bound over all j's and setting $\delta'' = \delta'/M$, we obtain the definition of ϵ_3 in the final statement.

(g) Using Assumption 2, we have $||V^\pi - \widehat{V}||_{2,\mu} \le \sqrt{C}||V^\pi - \widehat{V}||_{2,\sigma}$.
(h) We replace $||V^\pi - \widehat{V}||_{2,\sigma}$ using Proposition 1. □

Using the result of Lemma 1, we now prove a performance bound for a single iteration of DPI-Critic.

Theorem 1. *Let Π be a policy space with finite VC-dimension $h = VC(\Pi) < \infty$ and ρ be a distribution over the state space $\mathcal{X}$. Let N be the number of states in $\mathcal{D}_k$ drawn i.i.d. from ρ, H be the horizon of the rollouts, M be the number of rollouts per state-action pair, and $\widehat{V}^{\pi_k}$ be the estimation of the value function returned by the critic. Let Assumptions 1 and 2 hold and $\pi_{k+1} = \arg\min_{\pi\in\Pi} \widehat{\mathcal{L}}_{\pi_k}(\widehat{\rho};\pi)$ be the policy computed at the k'th iteration of DPI-Critic. Then, for any $\delta > 0$, we have*

$$\mathcal{L}_{\pi_k}(\rho;\pi_{k+1}) \le \inf_{\pi\in\Pi} \mathcal{L}_{\pi_k}(\rho;\pi) + 2(\epsilon_0+\epsilon_1+\epsilon_2+\epsilon_3+\epsilon_4), \quad (5)$$

with probability $1-\delta$, where

$$\epsilon_0 = 16q\sqrt{\frac{2}{N}\left(h\log\frac{eN}{h} + \log\frac{32}{\delta}\right)}\;.$$

The proof follows similar steps as in Lazaric et al. (2010a) and is reported in Gabillon et al. (2011).

Remark 1. The terms in the bound of Theorem 1 are related to the performance at each iteration of DPI-Critic. The first term, $\inf_{\pi\in\Pi}\mathcal{L}_{\pi_k}(\rho;\pi)$, is the approximation error of the policy space Π, i.e., the best approximation of the greedy policy in Π. Since the classifier relies on a finite number of samples in its training set, it is not able to recover the optimal approximation and incurs an additional estimation error ϵ_0 which decreases as $O(N^{-1/2})$. Furthermore, the training set of the classifier is built according to action-value estimates, whose accuracy is bounded by the remaining terms. The term ϵ_1 accounts for the variance of the rollout estimates due to the limited number of rollouts for each state in the rollout set. While it decreases as M and N increase, it increases with H, because longer rollouts have a larger variance due to the stochasticity in the MDP dynamics. The terms ϵ_2, ϵ_3, and ϵ_4 are related to the bias induced by truncating the rollouts. They all share a factor γ^H decaying exponentially with H and are strictly related to the critic's prediction of the return from H on. While ϵ_3 depends on the specific function approximation algorithm used by the critic (LSTD in our analysis) just through the dimension d of the function space $\mathcal{F}$, ϵ_4 is strictly related to LSTD's performance, which depends on the size n of its training set and the accuracy of its function space, i.e., the approximation error $\inf_{f\in\mathcal{F}}||V^\pi - f||_{2,\sigma}$.

Remark 2. We now compare the result of Theorem 1 with the corresponding result for DPI in Lazaric et al. (2010a), which bounds the performance as

$$\mathcal{L}_{\pi_k}(\rho;\pi_{k+1}) \le \inf_{\pi\in\Pi}\mathcal{L}_{\pi_k}(\rho;\pi) + 2(\epsilon_0+\epsilon_1+\gamma^H q). \quad (6)$$

While the approximation error $\inf_{\pi\in\Pi}\mathcal{L}_{\pi_k}(\rho;\pi)$ and the estimation errors ϵ_0 and ϵ_1 are the same in Eqs. 5 and 6, the difference in the way that these algorithms handle the rollouts after H steps leads to the term $\gamma^H q$ in DPI and the terms ϵ_2, ϵ_3, and ϵ_4 in DPI-Critic. The terms ϵ_2, ϵ_3, and ϵ_4 have the term $\gamma^H q$ multiplied by a factor which decreases with the number of rollout states N, the number of rollouts M, and the size of the critic training set n. For large enough values of N and n, this multiplicative factor is smaller than 1, thus making $\epsilon_2+\epsilon_3+\epsilon_4$ smaller than $\gamma^H q$ in DPI. Furthermore, since these ϵ values upper bound the difference between quantities bounded in $[-q, q]$, their values cannot exceed $\gamma^H q$. This comparison supports the idea that introducing a critic improves the accuracy of the truncated rollout estimates by reducing the bias with no increase in the variance.

Remark 3. Although Theorem 1 reveals the potential advantage of DPI-Critic w.r.t. DPI, the comparison in Remark 2 does not take into consideration that DPI-Critic uses n samples more than DPI, thus making the comparison potentially unfair. We now analyze the case when the total budget (number of calls to the generative model) of DPI-Critic is fixed to B. The total budget is split in two parts: **1)** $B_R = B(1-p)$ the budget available for the rollout estimates and **2)** $B_C = Bp = n$ the number of samples used by the critic, where $p \in (0,1)$ is the *critic ratio* of the total budget. By substituting B_R and B_C in the bound of Theorem 1 and setting $M = 1$, we note that for a fixed H, while increasing p increases the estimation error terms $\epsilon_0, \epsilon_1, \epsilon_2$, and ϵ_3 (the rollout set becomes smaller), it decreases the estimation error of LSTD ϵ_4 (the critic's training set becomes larger). This trade-off (later referred to as the *critic trade-off*) is optimized by a specific value $p = p^*$ which minimizes the expected error of DPI-Critic. By comparing the bounds of DPI and DPI-Critic, we first note that for any fixed p, DPI benefits from a larger number of samples to build the rollout estimates, thus has smaller estimation errors ϵ_0 and ϵ_1 w.r.t. DPI-Critic. However, as pointed out in Remark 2, the bias term $\gamma^H q$ in the DPI bound is always worse than the corresponding term in the DPI-Critic bound. As a result, whenever the advantage obtained by relying on the critic is larger than the loss in having a smaller number of rollouts, we expect DPI-Critic to outperform DPI. Whether this is the case depends on a number of factors such as the dimensionality and the approximation error of the space $\mathcal{F}$, the horizon H, and the size N of the rollout set.

Remark 4. According to Assumption 1 the samples in the critic's training set are completely independent from those used in building the rollout estimates. A more data-efficient version of the algorithm can be devised as follows: We first simulate all the trajectories used in the computation of the rollouts and use the last few transitions of each to build the critic's training set S_k. Then, after the critic (LSTD) computes an estimate of the value function using the samples in S_k, the action-values of the states in the rollout set $\mathcal{D}_k$ are estimated as in Eqs. 1–3. This way the function approximation step does not change the total budget. We call this version of the algorithm *Combined DPI-Critic* (CDPI-Critic). From a theoretical point of view, the main problem is that the samples in S_k are no longer drawn from the stationary distribution σ_k of the policy under evaluation π_k. However, the samples in S_k are collected at the end of the rollout trajectories of length H obtained by following π_k, and thus, they are drawn from the distribution $\mu = \rho P^a (P^{\pi_k})^{H-1}$ that approaches σ_k as H increases. Depending on the mixing rate of the Markov chain induced by π_k, the difference between μ and σ_k could be relatively small, thus supporting the conjecture that CDPI-Critic may achieve a similar performance to DPI-Critic without the overhead of n independent samples. While we leave a detailed theoretical analysis of CDPI-Critic as future work, we use it in the experiments of Section 5.

5. Experimental Results

In this section, we report the empirical evaluation of DPI-Critic with LSTD and compare it to DPI (built on truncated rollouts) and LSPI (built on value function approximation). In the experiments we show that DPI-Critic, by combining truncated rollouts and function approximation, can improve over DPI and LSPI.

5.1. Setting

We consider two standard goal-based RL problems: mountain car (MC) and inverted pendulum (IP). We use the formulation of MC in Dimitrakakis & Lagoudakis (2008) with the action noise bounded in $[-1, 1]$ and $\gamma = 0.99$. The value function is approximated using a linear space spanned by a set of radial basis functions (RBFs) evenly distributed over the state space. The critic training set is built using one-step transitions from states drawn from a uniform distribution over the state space, while LSPI is trained off-policy using samples from a random policy. In IP, we use the same implementation, features, and critic's training set as in Lagoudakis & Parr (2003a) with $\gamma = 0.95$. In both domains, the function space to approximate the action-value function in LSPI is obtained by replicating the state-features for each action as suggested in Lagoudakis & Parr (2003a). Similar to Dimitrakakis & Lagoudakis (2008), the policy space Π (classifier) is defined by a multi-layer perceptron with 10 hidden units, and is trained using stochastic gradient descent with a learning rate of 0.5 for 400 iterations. In the experiments, instead of directly solving the cost-sensitive multi-class classification step as in Fig. 1, we minimize the classification error. In fact, the classification error is an upper-bound on the empirical error defined by Eq. 4. Finally, the rollout set is sampled uniformly over the state space.

Each DPI-based algorithm is run with the same fixed budget B per iteration. As discussed in Remark 3, DPI-Critic splits the budget into a rollout budget $B_R = B(1-p)$ and a critic budget $B_C = Bp$, where $p \in (0, 1)$ is the critic ratio. The rollout budget is divided into M rollouts of length H for each action in $\mathcal{A}$ and each state in the rollout set $\mathcal{D}$, i.e., $B_R = HMN|\mathcal{A}|$. In CDPI-Critic the critic training set S_k is built using all transitions in the rollout trajectories except the first one. LSPI is run off-policy (i.e., samples are collected once and reused through the iterations) and in order to have a fair comparison, its total number of samples equal to B times the number of iterations (5 in the following experiments).

In Fig. 2 and 3, we report the performance of DPI, DPI-Critic, CDPI-Critic, and LSPI. In MC, the performance is evaluated as number of steps-to-go with a maximum of 300. In IP, the performance is the number of balancing steps with a maximum of 3000 steps. The performance of each run is computed as the best performance over 5 iterations of policy iteration. The results are averaged over 1000 runs. Although in the graphs we report the performance of DPI and LSPI at $p = 0$ and $p = 1$, respectively, DPI-Critic does not necessarily tend to the same performance as DPI and LSPI when p approaches 0 or 1. In fact, values of p close to 0 correspond to building a critic with very few samples (thus affecting the performance of the critic), while values of p close to 1 correspond to a very small rollout set (thus affecting the performance of the classifier). We tested the performance of DPI and DPI-Critic on a wide range of parameters (H, M, N) but we only report the performance of the best combination for DPI, and show the performance of DPI-Critic for the best choice of M ($M = 1$ was the best choice in all the experiments) and different values of H.

5.2. Experiments

In both MC and IP, the reward function is constant everywhere except at the terminal state. Thus, rollouts are *informative* only if their trajectories reach the terminal state. Although this would suggest to have large values for the horizon H, the size of the rollout set would correspondingly decrease as $N = O(B/H)$, thus decreasing the accuracy of the classifier (see ϵ_0 in Thm. 1). This leads to a trade-off (referred to as the *rollout trade-off*) between long rollouts (which increase the chance of observing informative rewards) and the

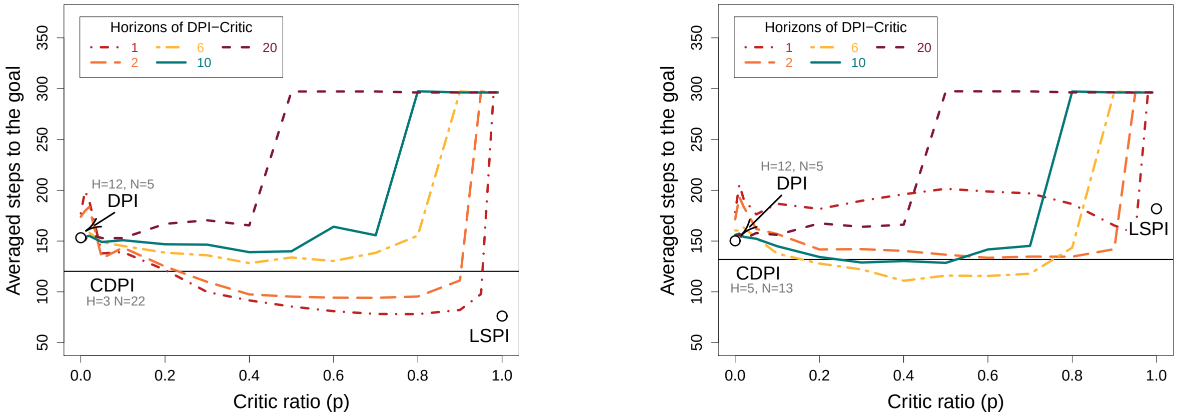

Figure 2. Performance of the learned policies in mountain car with a 3×3 RBF grid (left) and a 2×2 RBF grid (right). The total budget B is set to 200. The objective is to minimize the number of steps to the goal.

number of states in the rollout set. The solution to this trade-off strictly depends on the accuracy of the estimate of the return after a rollout is truncated. As discussed in Sec. 3, while in DPI this return is implicitly set to 0, in DPI-Critic it is set to the value returned by the critic. In this case, a very accurate critic would lead to solve the trade-off for small values of H, because the lack of informative rollouts is compensated by the critic. On the other hand, when the critic is inaccurate, H should be selected in a way to guarantee a sufficient number of informative rollouts, and at the same time, a large enough rollout set.

Fig. 2 shows the learning results in MC with budget $B = 200$. In the left panel, the function space for the critic consists of 9 RBFs distributed over a uniform grid. Such a space is rich enough for LSPI to learn nearly-optimal policies (about 80 steps to reach the goal). On the other hand, DPI achieves a poor performance of about 150 steps, which is obtained by solving the rollout trade-off at $H = 12$ and $N = 5$. We also report the performance of DPI-Critic for different values of H and p. We note that, as discussed in Remark 3, for a fixed H, there exists an optimal value p^* which optimizes the critic trade-off. For very small values of p, the critic has a very small training set and is likely to return a very poor approximation of the return. In this case, DPI-Critic performs similar to DPI and the rollout trade-off is achieved by $H = 12$, which limits the effect of potentially inaccurate predictions without reducing too much the size of the rollout set. On the other hand, as p increases the accuracy of the critic improves as well, and the best choice for H rapidly reduces to 1, which corresponds to rollouts built almost entirely on the basis of the values returned by the critic. For $H = 1$ and $p \approx 0.8$, DPI-Critic achieves a slightly better performance than LSPI. Finally, the horizontal line represents the performance of CDPI-Critic (for the best choice of H) which improves over DPI without matching the performance of LSPI.

Although this experiment shows that the introduction of a critic in DPI compensates for the truncation of the rollouts and improves their accuracy, most of this advantage is due to the quality of $\mathcal{F}$ in approximating value functions (LSPI itself is nearly-optimal). In this case, the results would suggest the use of LSPI rather than any DPI-based algorithm. In the next experiment, we show that DPI-Critic is able to improve over both DPI and LSPI even if $\mathcal{F}$ has a lower accuracy. We define a new space $\mathcal{F}$ spanned by 4 RBFs distributed over a uniform grid. The results are reported in the right panel of Fig. 2. The performance of LSPI now worsens to 180 steps. Since the quality of the critic returned by LSTD in DPI-Critic is worse than in the case of 9 RBFs, $H = 1$ is no longer the best choice for the rollout trade-off. However, as soon as $p > 0.1$, the accuracy of the critic is still higher than the 0 prediction used in DPI, thus leading to the best horizon at $H = 6$ (instead of 12 as in DPI), which guarantees a large enough number of informative rollouts. At the same time, other effects might influence the choice of the best horizon H. As it can be noticed, for $H = 6$ and $p \approx 0.5$, DPI-Critic successfully takes advantage of the critic to improve over DPI, and at the same time, it achieves a better performance than LSPI. Unlike LSPI, DPI-Critic computes its action-value estimates by combining informative rollouts and the critic value function, thus obtaining estimates which cannot be represented by the action-value function space used by LSPI. Additionally, similar to DPI, DPI-Critic performs a policy approximation step which could lead to better policies w.r.t. those obtained by LSPI.

Finally, Fig. 3 displays the results of similar experiments in IP with $B = 1000$. In this case, although the function space is not accurate enough for LSPI to learn good policies, it is helpful in improving the accu-

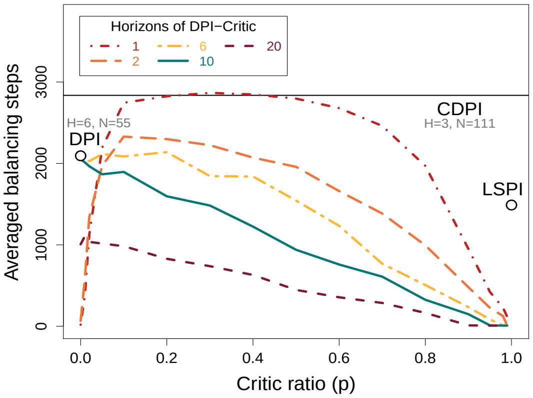

Figure 3. Performance of the learned policies in inverted pendulum. The budget is $B = 1000$. The goal is to keep the pendulum balanced with a maximum of 3000 steps.

racy of the rollouts w.r.t. DPI. When $p > 0.05$, $H = 1$ is the horizon which optimizes the rollout trade-off. In fact, since by following a random policy the pendulum falls after very few steps, rollouts of length one still allow to collect samples from the terminal state whenever the starting state is close enough to the horizontal line. Hence, with $H = 1$ action-values are estimated as a mix of both informative rollouts and the critic's prediction, and at the same time, the classifier is trained on a relatively large training set. Finally, it is interesting to note that in this case CDPI-Critic obtains the same nearly-optimal performance as DPI-Critic.

6. Conclusions

DPI-Critic adds value function approximation to the classification-based approach to policy iteration. The motivation behind DPI-Critic is two-fold. **1)** In some settings (e.g., those with delayed reward), DPI action-value estimates suffer from either high variance or high bias (depending on H). Introducing critic to the computation of the rollouts may significantly reduce the bias, which in turn allows for shorter horizon and thus lower variance. **2)** In value-based approaches (e.g., LSPI), it is often difficult to design a function space which accurately approximates action-value functions. In this case, integrating rough approximation of the value function returned by the critic with the rollouts obtained by direct simulation of the generative model may improve the accuracy of the function approximation and lead to better policies.

In Sec. 4, we theoretically analyzed the performance of DPI-Critic and showed that depending on several factors (notably the function approximation error), DPI-Critic may achieve a better performance than DPI. This analysis is also supported by the experimental results of Sec. 5, which confirm the capability of DPI-Critic to take advantage of both rollouts and critic, and improve over both DPI and LSPI. Although further investigation of the performance of DPI-Critic in more challenging domains is needed and in some settings either DPI or LSPI might still be the better choice, DPI-Critic seems to be a promising alternative that introduces additional flexibility in the design of the algorithm. Possible directions for future work include complete theoretical analysis of CDPI-Critic, a more detailed comparison of DPI-Critic and LSPI, and finding optimal or good rollout allocation strategies.

Acknowledgments Experiments presented in this paper were carried out using the Grid'5000 experimental testbed (https://www.grid5000.fr). This work was supported by Ministry of Higher Education and Research, Nord-Pas de Calais Regional Council and FEDER through the "contrat de projets état region 2007–2013", and by PASCAL2 Network of Excellence.

References

Barto, A., Sutton, R., and Anderson, C. Neuronlike elements that can solve difficult learning control problems. *IEEE Transaction on Systems, Man and Cybernetics*, 13:835–846, 1983.

Bradtke, S. and Barto, A. Linear least-squares algorithms for temporal difference learning. *Journal of Machine Learning*, 22:33–57, 1996.

Dimitrakakis, C. and Lagoudakis, M. Rollout sampling approximate policy iteration. *Machine Learning Journal*, 72(3):157–171, 2008.

Fern, A., Yoon, S., and Givan, R. Approximate policy iteration with a policy language bias. In *Proceedings of NIPS 16*, 2004.

Gabillon, V., Lazaric, A., Ghavamzadeh, M., and Scherrer, B. Classification-based policy iteration with a critic. Technical Report 00590972, INRIA, 2011.

Lagoudakis, M. and Parr, R. Least-squares policy iteration. *JMLR*, 4:1107–1149, 2003a.

Lagoudakis, M. and Parr, R. Reinforcement learning as classification: Leveraging modern classifiers. In *Proceedings of ICML*, pp. 424–431, 2003b.

Lazaric, A., Ghavamzadeh, M., and Munos, R. Analysis of a classification-based policy iteration algorithm. Technical Report 00482065, INRIA, 2010a.

Lazaric, A., Ghavamzadeh, M., and Munos, R. Finite-sample analysis of least-squares policy iteration. Technical Report inria-00528596, INRIA, 2010b.

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