Revisiting natural actor-critics with value function approximation

Matthieu Geist and Olivier Pietquin

IMS Research Group, Supélec, Metz, France

Abstract. Actor-critics architectures have become popular during the last decade in the field of reinforcement learning because of the introduction of the policy gradient with function approximation theorem. It allows combining rationally actor-critic architectures with value function approximation and therefore addressing large-scale problems. Recent researches led to the replacement of policy gradient by a natural policy gradient, improving the efficiency of the corresponding algorithms. However, a common drawback of these approaches is that they require the manipulation of the so-called advantage function which does not satisfy any Bellman equation. Consequently, derivation of actor-critic algorithms is not straightforward. In this paper, we re-derive theorems in a way that allows reasoning directly with the state-action value function (or Q-function) and thus relying on the Bellman equation again. Consequently, new forms of critics can easily be integrated in the actor-critic framework.

1 Introduction

Reinforcement learning (RL) is generally considered as the machine learning answer to the optimal control problem. In this paradigm, an agent learns to control optimally a dynamic system through interactions. At each time step i, the dynamic system is in a given state s_i and receives from the agent a command (or action) a_i. According to its own dynamics, the system transits to a new state s_{i+1}, and a reward r_i is given to the agent. The objective is to learn a control policy which maximizes the expected cumulative discounted reward.

Actor-critics approaches were among the first to be proposed for handling the RL problem [1]. In this setting, two structures are maintained, one for the actor (the control organ) and one for the critic (the value function which models the expected cumulative reward to be maximized). One advantage of such an approach is that it does not require knowledge about the system dynamics to learn an optimal policy. However, the introduction of the state-action value (or Q-) function [2] led to a focus of research community in pure critics methods, for which the control policy is derived from the Q-function and has no longer a specific representation. Actually, in contrast with value function, state-action value function allows deriving a greedy policy without knowing system dynamics, and function approximation (which is a way to handle large problems) is easier to combine with pure critics approaches. Pure critic algorithms therefore aim at learning this Q-function. However, actor-critics have numerous advantages over pure critics: a separate representation is maintained for the policy (in which we are

ultimately interested), they somehow implicitly solve a problem known as dilemma between exploration and exploitation, they handle well large action spaces (which is not the case of pure critics, as some maxima over actions have always to be computed), and above all errors in the Q-function estimation can lead to bad derived policies.

A major march for actor-critics is the policy gradient with function approximation theorem [3, 4]. This result allows combining actor-critics with value function approximation, which was a major lack of the field. Another important improvement is the natural policy gradient [5] which replaces the gradient ascent over policy parameters by a natural gradient ascent improving consequently the efficiency of resulting algorithms. These results share the drawback that they lead to work with the advantage function which does not satisfy a Bellman equation. Consequently, derivation of practical algorithms is not straightforward, as it requires estimating the advantage function which is unnatural in RL. In this paper, we reformulate (and re-prove) the theorems so as to work directly with the state-action value function. This allows a very straightforward derivation of new actor-critic algorithms, some of them being proposed here. All results are given for the discounted cumulative reward case, however they can be easily extended to the average reward case.

2 Policy gradient

The system to be controlled is standardly modeled as a Markov decision process (MDP) $\{S, A, P, R, \gamma\}$ [6]: S is the (finite) state space, A the (finite) action space, $P \in \mathcal{P}(S)^{S \times A}$ the set of transition probabilities, $R \in \mathbb{R}^{S \times A \times S}$ the deterministic reward function and γ the forgetting factor. Actions are selected according to a stochastic policy $\pi \in \mathcal{P}(A)^S$ which has to be optimized. The criteria to be maximized is the expected discounted cumulative reward starting in state s_0 and then following the policy π:

$$\rho(\pi) = E[\sum_{i=0}^{\infty} \gamma^i r_i | s_0, \pi] \tag{1}$$

State-action and state value functions are defined as:

$$Q^\pi(s,a) = E[\sum_{i=0}^{\infty} \gamma^i r_i | s_0 = s, a_0 = a, \pi] \tag{2}$$

$$V^\pi(s) = E_{a|s,\pi}[Q^\pi(s,a)] \tag{3}$$

Both state-action and state value functions satisfy a Bellman equation:

$$Q^\pi(s,a) = \sum_{s' \in S} p(s'|s,a)(r(s,a,s') + \gamma \sum_{a' \in A} \pi(a'|s') Q^\pi(s',a')) \tag{4}$$

$$V^\pi(s) = \sum_{s',a} p(s'|s,a)\pi(a|s)(r(s,a,s') + \gamma V^\pi(s')) \tag{5}$$

The advantage function is defined as their difference:

$$A^\pi(s,a) = Q^\pi(s,a) - V^\pi(s) \tag{6}$$

The advantage function does not satisfy any Bellman equation and it is obvious that $E_{a|s,\pi}[A^\pi(s,a)] = 0$. Let also d^π be the discounted weighting of states encountered starting at s_0 and then following π:

$$d^\pi(s) = \sum_{i=0}^{\infty} \gamma^i p(s_i = s|s_0, \pi) \tag{7}$$

Let the policy π_ω be parameterized by a parameter vector ω (and differentiable with respect to its parameters). We also assume that it is never zero ($\forall s, a, \omega$, $\pi_\omega(a|s) > 0$). The policy gradient approach consists in correcting parameters by following a gradient ascent:

$$\omega_i = \omega_{i-1} + \alpha_i \nabla_\omega \rho(\pi_{\omega_{i-1}}) \tag{8}$$

An important result is the policy gradient theorem (see [3] for a proof).

Theorem 1 (Policy gradient). *The gradient of the performance metric respectively to the policy parameters is:*

$$\nabla_\omega \rho(\pi_\omega) = \sum_{s \in S} d^\pi(s) \sum_{a \in A} Q^{\pi_\omega}(s,a) \nabla_\omega \pi_\omega(a|s) \tag{9}$$

An interesting thing would be to replace the true state-action value function $Q^{\pi_\omega}(s,a)$ by some approximation $\hat{Q}(s,a)$. Actually, it is possible thanks to the policy gradient with function approximation theorem [3, 4], if the approximation is "good enough" and if the representations for the actor and the critic are "compatible". The proposed approach differs from previous ones on this last point.

Definition 1 (Semi-compatible approximation). *Let the approximation of the state-action value function be parameterized by two parameter vectors θ and ξ such that:*

$$\hat{Q}_{\theta,\xi}(s,a) = f_\theta(s,a) + g_\xi(s) \tag{10}$$

This approximation is said to be semi-compatible if its state-action part f_θ is compatible with the policy parameterization in the sense that:

$$\nabla_\omega \ln \pi_\omega(a|s) = \nabla_\theta f_\theta(s,a) \tag{11}$$

This definition differs from previous works in the sense that it adds a state-dependent parameterization g_ξ to the approximation, whereas in previous approaches the Q-function is approximated by only f_θ. However, $E_{a|s,\pi_\omega}[f_\theta(s,a)] = E_{a|s,\pi_\omega}[\theta^T \nabla_\omega \ln \pi_\omega(a|s)] = \theta^T \nabla_\omega E_{a|s,\pi_\omega}[1] = 0$, thus f_θ is an approximation of the advantage function rather than of the state-action value function. On the other hand, $E_{a|s,\pi_\omega}[\hat{Q}_{\theta,\xi}(s,a)] = g_\xi(s)$, thus g_ξ is an approximation of the value function. We propose to rederive all classic results with this new parameterization. The interest of this approach is that it simplifies the design of the critic, as it implies to work directly with the state-action value function.

Theorem 2 (Policy gradient with state-action value function approximation). *Let $\hat{Q}_{\theta,\xi}$ be semi-compatible as defined before. Moreover, assume that it is a good approximation in the sense that it is a local optimum of the square error between the true*

state-action value function Q^{π_ω} and its approximation[1]*:*

$$\begin{aligned}&\nabla_{\theta,\xi} E_{s,a|d^{\pi_\omega},\pi_\omega}[(Q^{\pi_\omega}(s,a)-\hat{Q}_{\theta,\xi}(s,a))^2]=0\\ \Leftrightarrow\ &E_{s,a|d^{\pi_\omega},\pi_\omega}[(Q^{\pi_\omega}(s,a)-\hat{Q}_{\theta,\xi}(s,a))\nabla_{\theta,\xi}\hat{Q}_{\theta,\xi}(s,a)]=0\end{aligned} \tag{12}$$

Then the policy gradient satisfies:

$$\nabla_\omega \rho(\pi_\omega)=\sum_{s\in S} d^\pi(s)\sum_{a\in A}\hat{Q}_{\theta,\xi}(s,a)\nabla_\omega\pi_\omega(a|s) \tag{13}$$

Proof. The gradient of $\hat{Q}_{\theta,\xi}$ is:

$$\nabla_{\theta,\xi}\hat{Q}_{\theta,\xi}(s,a)=[\nabla_\theta^T f_\theta(s,a),\nabla_\xi^T g_\xi(s)]^T \tag{14}$$

Let also $\Delta Q(s,a)$ be defined as:

$$\Delta Q(s,a)=Q^{\pi_\omega}(s,a)-\hat{Q}_{\theta,\xi}(s,a) \tag{15}$$

Part of condition (12) corresponding to parameters θ can thus be extracted and expanded thanks to the semi-compatibility condition (11):

$$\begin{aligned}0&=E_{s,a|d^{\pi_\omega},\pi_\omega}[\Delta Q(s,a)\nabla_\theta f_\theta(s,a)]\\&=E_{s,a|d^{\pi_\omega},\pi_\omega}[\Delta Q(s,a)\nabla_\omega \ln\pi_\omega(a|s)]\\&=\sum_{s\in S} d^{\pi_\omega}(s)\sum_{a\in A}\pi_\omega(a|s)\Delta Q(s,a)\nabla_\omega\ln\pi_\omega(a|s)\end{aligned}$$

However $\pi_\omega(a|s)\nabla_\omega\ln\pi_\omega(a|s)=\nabla_\omega\pi_\omega(a|s)$ thus:

$$\sum_{s\in S} d^{\pi_\omega}(s)\sum_{a\in A}\Delta Q(s,a)\nabla_\omega\pi_\omega(a|s)=0 \tag{16}$$

Substracting Eq. (16) to Eq. (9) gives the result:

$$\begin{aligned}\nabla_\omega\rho(\omega)&=\sum_{s\in S} d^{\pi_\omega}(s)\sum_{a\in A}Q^{\pi_\omega}(s,a)\nabla_\omega\pi_\omega(a|s)-0\\&=E_{s|d^{\pi_\omega}}[\sum_{a\in A}(Q^{\pi_\omega}(s,a)-\Delta Q(s,a))\nabla_\omega\pi_\omega(a|s)]\\&=\sum_{s\in S} d^{\pi_\omega}(s)\sum_{a\in A}\hat{Q}_{\theta,\xi}(s,a)\nabla_\omega\pi_\omega(a|s)\end{aligned} \tag{17}$$

Notice that this results still holds by replacing $\hat{Q}_{\theta,\xi}(s,a)$ by $f_\theta(s,a)+b(s)$ where $b(s)$ is any baseline only depending on states, see [3]. Moreover, the minimum variance baseline for the state-action value function estimator is the value function $V^{\pi_\omega}(s)$ itself [7]. Recall that $g_\xi(s)$ is in fact an estimate of this value function, so using $\hat{Q}_{\theta,\xi}$ lets envision a low variance estimate.

[1] As $\sum_{s\in S} d^\pi(s)=\frac{1}{1-\gamma}$, d^π is not really a distribution and the notation $E_{s|d^\pi}[h(s)]=\sum_{s\in S} d^\pi(s)h(s)$ is slightly abusive.

Thanks to this result, new actor-critics algorithms can be naturally derived. The actor is the policy π_ω parameterized by ω, corrected by a gradient ascent using *e.g.* a sampled version of (13), and the critic is the approximated state-action value function $\hat{Q}_{\theta,\xi}$ parameterized by $[\theta^T, \xi^T]$ satisfying the semi-compatibility condition (11) and for which parameters are learnt such that condition (12) is satisfied.

However, another important progress for actor-critics is to correct policy parameters according to a natural gradient ascent, and we examine this point before proposing some practical algorithms.

3 Natural policy gradient

The idea of natural policy gradient is to correct the policy representation according to a natural gradient ascent rather than a gradient ascent. The natural gradient $\tilde{\nabla}$ is the gradient pre-multiplied by the inverse of the Fisher information matrix [8]:

$$\tilde{\nabla}\rho(\pi_\omega) = G^{-1}(\omega)\nabla\rho(\pi_\omega) \tag{18}$$

with the Fisher information matrix being equal to (see [5]):

$$G(\omega) = E_{s,a|d^{\pi_\omega},\pi_\omega}[\nabla_\omega \ln \pi_\omega(a|s) \nabla_\omega^T \ln \pi_\omega(a|s)] \tag{19}$$

The natural policy gradient was first introduced in [9] from a pure actor perspective. It was then used in [5] from an actor-critic perspective. They show an important result: the natural gradient is actually the advantage function parameter vector under the compatible approximation assumption. We show here that this result still holds for the proposed extended parameterization $\hat{Q}_{\theta,\xi}$.

Theorem 3 (Natural policy gradient with state-action value function approximation). *Let $\hat{Q}_{\theta,\xi}$ be semi-compatible as defined before and satisfying condition (12). Then the natural policy gradient satisfies:*

$$\tilde{\nabla}_\omega \rho(\pi_\omega) = \theta \tag{20}$$

Proof. Under these assumptions, theorem 2 applies:

$$\begin{aligned} \nabla_\omega \rho(\omega) &= \sum_{s \in S} d^{\pi_\omega}(s) \sum_{a \in A} \hat{Q}_{\theta,\xi}(s,a) \nabla_\omega \pi_\omega(a|s) \\ &= \sum_{s \in S} d^{\pi_\omega}(s) \sum_{a \in A} (f_\theta(s,a) + g_\xi(s)) \nabla_\omega \pi_\omega(a|s) \end{aligned} \tag{21}$$

As the term g_ξ does not depends on actions, it disappears from the above equation: $\sum_a g_\xi(s) \nabla_\omega \pi_\omega(a|s) = g_\xi(s) \nabla_\omega \sum_a \pi_\omega(a|s) = g_\xi(s) \nabla_\omega 1 = 0$. As $\nabla_\omega \pi_\omega(a|s) = \pi_\omega(a|s) \nabla_\omega \ln \pi_\omega(a|s)$ and as $f_\theta(s,a) = \theta^T \nabla_\omega \ln \pi_\omega(a|s)$ (semi-compatibility condition), Eq. (21) leads to:

$$\begin{aligned} \nabla_\omega \rho(\omega) &= \sum_{s \in S} d^{\pi_\omega}(s) \sum_{a \in A} f_\theta(s,a) \nabla_\omega \pi_\omega(a|s) \\ &= E_{s,a|d^{\pi_\omega},\pi_\omega}[\theta^T \nabla_\omega \ln \pi_\omega(a|s) \nabla_\omega \ln \pi_\omega(a|s)] \\ &= E_{s,a|d^{\pi_\omega},\pi_\omega}[\nabla_\omega \ln \pi_\omega(a|s) \nabla_\omega^T \ln \pi_\omega(a|s)] \theta \end{aligned} \tag{22}$$

Recall the definition (19) of $G(\omega)$, this leads to: $\nabla_\omega \rho(\omega) = G(\omega)\theta$. This last equation and the natural gradient definition (18) lead directly to the result:

$$\tilde{\nabla}\rho(\omega) = G^{-1}(\omega)\nabla\rho(\omega) = G^{-1}(\omega)G(\omega)\theta = \theta \tag{23}$$

Thanks to this result, other actor-critic algorithms can be derived. The principle is the same as previously, but the gradient ascent is replaced by a natural gradient ascent, which is actually straightforward thanks to the above result.

4 Position to previous works

In previous works [3, 5, 7], theorems are derived using a parameterization f_θ of the advantage function. In order to obtain practically a critic, the advantage function (which does not satisfy a Bellman equation) has to be estimated. This is done either by adding a value component to the advantage function [5] or by using a TD error (linked to a value estimate) as the target for the advantage function. In this paper, we consider directly a parameterization of the Q-function and rederive theorems consequently. From a technical point of view, the new proofs heavily rely on the fact that the policy gradient is invariant to a state-dependent bias, which is a long known result [3]. Consequently, we do not really propose new theoretical insights on actor-critic architectures. However, from a practical point of view, our approach allows working directly with the state-action value function, which makes easier critics derivations. Moreover, notice that resulting algorithms are different from previously published ones, as shown in the next section.

5 Deriving new actor-critic algorithms

Given these results, deriving practical actor-critic algorithms is quite direct. The actor is updated according to the natural-gradient, and the only remaining choice is the critic learner. We propose three critics here, the first one being based on TD with function approximation [6] and on a two-timescale approach [7], the two other ones being based on a Kalman-based Temporal Differences framework [10].

5.1 TD-NAC

The first algorithm, which we call TD-NAC (TD-based Natural Actor-Critic), is based on the classical TD with function approximation. A semi-compatible parameterization $\hat{Q}_{\theta,\xi}$ is adopted, and the critic is updated as follows:

$$\begin{pmatrix}\theta_i \\ \xi_i\end{pmatrix} = \begin{pmatrix}\theta_{i-1} \\ \xi_{i-1}\end{pmatrix} + \alpha_i \delta_i \nabla_{\theta,\xi} \hat{Q}_{\theta_{i-1},\xi_{i-1}}(s_i, a_i) \tag{24}$$

where α_i is the learning rate and δ_i the temporal difference error:

$$\delta_i = r_i + \gamma \hat{Q}_{\theta_{i-1},\xi_{i-1}}(s_{i+1}, a_{i+1}) - \hat{Q}_{\theta_{i-1},\xi_{i-1}}(s_i, a_i) \tag{25}$$

Deriving the critic update rule is thus a very direct application of the TD algorithm, much more direct than starting from the advantage function. A remaining problem is to

ensure condition (12). We follow [7] and use two different timescales for the actor and the critic. The actor is updated using another learning rate β_i such that:

$$\sum_i \alpha_i = \sum_i \beta_i = \infty, \ \sum_i \alpha_i^2 = \sum_i \beta_i^2 < \infty, \ \lim_{i \to \infty} \frac{\beta_i}{\alpha_i} = 0 \tag{26}$$

The idea behind this is that in order to ensure condition (12), the actor should remain stationary from the critic point of view. Conditions (26) ensure that the critic converges faster. The TD-NAC algorithm can be summarized as follows, the temporal difference error δ_i being defined in Eq. (25):

$$\theta_i = \theta_{i-1} + \alpha_i \delta_i \nabla_\omega \ln \pi_{\omega_{i-1}}(s_i|a_i) \tag{27}$$

$$\xi_i = \xi_{i-1} + \alpha_i \delta_i \nabla_\xi g_{\xi_{i-1}(\mathsf{s}_i)} \tag{28}$$

$$\omega_i = \omega_{i-1} + \beta_i \theta_i \tag{29}$$

This algorithm is very close[2] to algorithm 3 in [7], which was actually first proposed in [11][3] (who call it NTD for Natural policy gradient using the Temporal Differences). Their principle is to estimate the value function (ξ_i parameters) and to use the associated temporal difference error as a target for the advantage function, which is actually a form of bootstrapping. This is less direct than the proposed approach which considers directly the Q-function and is therefore less dependent to the learning algorithm used to estimate it. The critic update for these algorithms is as follows:

$$\delta'_i = r_i + g_{\xi_{i-1}}(s_{i+1}) - g_{\xi_{i-1}}(s_i) \tag{30}$$

$$\theta_i = \theta_{i-1} + \alpha_i \nabla_\theta (f_{\theta_{i-1}}(s_i, a_i))(\delta'_i - f_{\theta_{i-1}}(s_i, a_i)) \tag{31}$$

$$\xi_i = \xi_{i-1} + \alpha_i \nabla_\xi (g_{\xi_{i-1}}(s_i))\delta'_i \tag{32}$$

Recall that $g_{\xi_{i-1}}$ (resp. $f_{\theta_{i-1}}$) is an approximation of the value (resp. advantage) function. Therefore, the difference between TD-NAC and NTD is that $E_{\mathsf{a}_{i+1}|\mathsf{s}_{i+1}}[\delta_i]$ is used instead of δ_i for the θ update, and that $E_{\mathsf{a}_i|\mathsf{s}_i}[[E_{\mathsf{a}_{i+1}|\mathsf{s}_{i+1}}[\delta_i]]$ is used instead of δ_i for the ξ update. Roughly speaking, the value function estimate is sometimes used instead of state-action value function estimate. These slight variations about the TD error are not new in reinforcement learning, see for example [12] and references therein.

5.2 KNAC

TD-NAC is based on a first-order critic. Using a second-order critic should speed up learning, as such algorithms are more sample-efficient. Actually, [5] introduced a natural actor-critic based on the Least-Squares Temporal Differences (LSTD) algorithm of [13]. However, in order to satisfy condition (12), this actor-critic algorithm is usually considered in a batch setting. Contrary to TD-NAC for which policy is improved at each time-step, the policy is maintained, its state-action value function is evaluated using obtained trajectories, and then the policy is improved. The advantage of using a

[2] In [7] algorithms are derived in the average reward case. However, extension to the discounted cumulative reward case is quite direct, and what we say is based on this extension.

[3] This algorithm is really derived in the discounted reward case.

second-order algorithm is thus somehow lost, as the policy cannot be improved after each interaction.

Kalman Temporal Differences (KTD) is another second-order algorithm, introduced in [10]. It has some interesting aspects, such as nonlinear parameterization handling. However, the feature we are interested in is its ability to handle non-stationarities. In an (online) actor-critic, the policy is updated after each interaction and is therefore not stationary. Consequently, the associated state-action value function is non-stationary too. To handle this problem (linked to condition (12)), the two-timescale approach is used in TD-NAC (the actor is stationary from the critic point of view). We argue that using a critic which tracks the state-action value function rather than converging to it is another manner to handle this problem, and thus to satisfy condition (12). Actually, [14] show theoretically that this condition is at least satisfied for a deterministic MDP and a stationary policy, and they show empirically that it still holds for non-stationary policies. This idea of using an adaptive critic also appears in [15], where a recursive form of LSTD integrating a forgetting factor (less general than the evolution model of KTD to be presented) is considered[4].

A semi-compatible parameterization $\hat{Q}_{\theta,\xi}$ is adopted. KTD algorithm is derived from a so called *state-space* formulation[5], which in our case is given by:

$$\begin{cases} \begin{pmatrix} \theta_i \\ \xi_i \end{pmatrix} = \begin{pmatrix} \theta_{i-1} \\ \xi_{i-1} \end{pmatrix} + v_i \\ r_i = \hat{Q}_{\theta_i,\xi_i}(s_i,a_i) - \gamma \hat{Q}_{\theta_i,\xi_i}(s_{i+1},a_{i+1}) + n_i \end{cases} \tag{33}$$

The first equation is the evolution equation, it specifies that parameters (which are modeled as random variables) evolve according to a random walk driven by the evolution noise v_i (to be chosen by the practitioner). It allows handling non-stationarity and avoiding local minima. The second equation is the observation equation which links the reward to the estimated state-action value function through a sampled Bellman equation. The observation noise n_i (also to be chosen) is an inductive bias which arises from the fact that the true state-action value function does not necessarily exists in the hypothesis space spanned by parameters.

The critic practical update is obtained directly from state-space model (33) and using the KTD-SARSA algorithm described in [10]. It is not difficult, but it takes room, so it is not fully described here. It should be sufficient to know that the critic is updated according to:

$$\begin{pmatrix} \theta_i \\ \xi_i \end{pmatrix} = \begin{pmatrix} \theta_{i-1} \\ \xi_{i-1} \end{pmatrix} + K_i(r_i - \hat{r}_i) \tag{34}$$

where K_i is the Kalman gain, which computation is detailed in the aforementioned paper, and where $\hat{r}_i$ is the prediction of the reward according to past estimates of the parameters and using the observation equation. Actually, $r_i - \hat{r}_i$ is a temporal difference error which takes into account the statistical nature of parameters in this model.

[4] However, quite surprisingly, they also consider eligibility traces which induce a memory effect and therefore harm the non-stationary handling ability.

[5] The name state-space comes from the Kalman filtering literature and should no be confused with the state space of the MDP.

The actor is updated according to the natural gradient ascent (29), θ_i being estimated by KTD. Notice that here there is only one β_i learning rate (there is no learning rate for KTD). We call the resulting natural actor-critic algorithm KNAC (Kalman-based Natural Actor-Critic), which can be summarized:

$$\begin{pmatrix}\theta_i\\ \xi_i\end{pmatrix} = \begin{pmatrix}\theta_{i-1}\\ \xi_{i-1}\end{pmatrix} + K_i(r_i - \hat{r}_i) \tag{35}$$

$$\omega_i = \omega_{i-1} + \beta_i \theta_i \tag{36}$$

Recall that the main difference between NTD and TD-NAC is the replacement of the state-action value function by the value function in the temporal difference error. This can be easily adapted to KNAC: the term $\hat{Q}_{\xi_i}(s_{i+1}, a_{i+1})$ can be replaced by $g_{\xi_i}(s_{i+1})$ in the observation equation of state-space model (33):

$$\begin{cases}\begin{pmatrix}\theta_i\\ \xi_i\end{pmatrix} = \begin{pmatrix}\theta_{i-1}\\ \xi_{i-1}\end{pmatrix} + v_i\\ r_i = f_{\theta_i}(s_i, a_i) + g_{\xi_i}(s_i) - \gamma g_{\xi_i}(s_{i+1}) + n_i\end{cases} \tag{37}$$

We call this alternative algorithm aKNAC for averaged KNAC.

6 Experimental results

In this section, we compare the three proposed new actor-critics to the NTD algorithm of [11] (which we recall to be equivalent to what would have been algorithm 3 of [7] in the discounted cumulative reward case). The benchmark on which these algorithms are compared is the inverted pendulum as described for example in [16].

This task requires balancing a pendulum of unknown length and mass at the upright position by applying forces to the cart it is attached to. Three actions are allowed: left force (-1), right force (+1), or no force (0). The associated state space consists in vertical angle φ and angular velocity $\dot{\varphi}$ of the pendulum. Deterministic transitions are computed according to physical dynamics of the system, and depends on current action a:

$$\ddot{\varphi} = \frac{g\sin(\varphi) - \beta ml\dot{\varphi}^2\sin(2\varphi)/2 - 50\beta\cos(\varphi)a}{4l/3 - \beta ml\cos^2(\varphi)} \tag{38}$$

where g is the gravity constant, m and l the mass and the length of the pendulum, M the mass of the cart, and $\beta = \frac{1}{m+M}$. The reward is the cosine of the angular position, that is $r_i = \cos(\varphi_i)$, and the episode ends when $|\varphi_i| \geq \frac{\pi}{2}$. The discount factor γ is set to 0.95.

The policy is parameterized according to a Gibbs distribution. Let p be the size of the ω parameter vector (and thus of θ) and q the size of the ξ parameter vector. Let $vai\ \psi ramr$ $ddlli$

The semi-compatibility condition is therefore:

$$\nabla_\theta f_\theta(s,a) = \nabla_\omega \ln(\pi_\omega(a|s)) = \phi(s,a) - \sum_{b\in A} \pi_\omega(b|s)\phi(s,b) \tag{40}$$

Consequently, the f_θ function is given by:

$$f_\theta(s,a) = (\phi(s,a) - \sum_{b\in A} \pi_\omega(b|s)\phi(s,b))^T \theta \tag{41}$$

The parameterization is composed of a constant term and a set of 9 equispaced Gaussian kernels (centered in $\{-\frac{\pi}{4}, 0, \frac{\pi}{4}\} \times \{-1, 0, 1\}$ and with a standard deviation of 1) for each action. Thus there is a set of $p = 30$ basis functions. A parameterization has also to be chosen for the g_ξ part of the estimated state-action value function. Let $\psi(s) = (\psi_i(s))_{1\leq i\leq q}$ be a feature vector, we choose a linear parameterization:

$$g_\xi(s) = \psi(s)^T \xi \tag{42}$$

The parameterization is also composed of a constant term and a set of 9 equispaced Gaussian kernels (centered in $\{-\frac{\pi}{4}, 0, \frac{\pi}{4}\} \times \{-1, 0, 1\}$ and with a standard deviation of 1). Thus there is a set of $q = 10$ basis functions.

Some parameters have to be chosen for all algorithms. The ones we provide here allow obtaining good results. They are probably not optimal (better results could certainly have been obtained by testing more systematically all parameters), however orders of magnitude are valid. For all algorithms, the initial parameter vector $[\theta_0^T, \xi_0^T]$ is set to zero. For NTD and TD-NAC the critic learning rate is set to $\alpha_i = \alpha_0 \frac{\alpha_c}{\alpha_c + i^{\frac{2}{3}}}$ with $\alpha_0 = 10^{-2}$ and $\alpha_c = 10^4$. For all algorithms the actor learning rate is set to $\beta_i = \beta_0 \frac{\beta_c}{\beta_c + i}$ with $\beta_0 = 10^{-3}$ and $\beta_c = 10^4$. These learning rates satisfy the two-timescale condition (26) for NTD and TD-NAC. KNAC and aKNAC are based on KTD. This critic is a second-order algorithm for which parameters are modelled as random variables. A prior variance over theses parameters P_0 has to be chosen. Here we set $P_0 = I$ with I the identity matrix. Evolution and observation noises have also to be chosen. They are centered by assumption, and KTD only needs their second-order moments. In this experiment, the variance of the observation noise is set to $P_{n_i} = 10^{-1}$. An adaptive evolution noise is chosen, and its variance is set to $P_{v_i} = \eta P_{i-1}$ where $\eta = 10^{-5}$ is a forgetting factor and P_{i-1} is the estimate of parameters variance at time $i-1$ which is computed in the KTD algorithm. See [14] for a discussion on the choice of these parameters.

Algorithms are compared on their ability to learn the optimal policy. At the beginning of each episode the pendulum is initialized in a position close to the equilibrium $(\varphi, \dot{\varphi}) = (0, 0)$. The performance is measured as the number of steps the pendulum is maintained is the admissible zone (otherwise speaking, the length of the episode) in function of the number of episodes. A maximum of 3000 timesteps is allowed (the optimal policy would lead to an infinite episode). Results presented on Fig. 1 are averaged over 100 independent trials.

Results of NTD and TD-NAC are not significantly different. Both algorithms learn the optimal policy in about 1600 episodes. However TD-NAC critic is simpler to derive

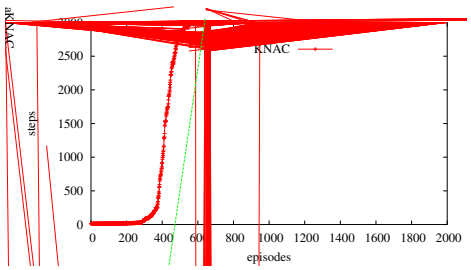

Fig. 1. Comparison of algorithms on the inverted pendulum task.

(or at least more direct). KNAC learns the optimal policy faster, in about 600 episodes. This was to be expected: NTD and TD-NAC are based on a first-order critic, which is less sample-efficient than KTD which is a second-order critic. The aKNAC algorithm is even more efficient than KNAC, it learns the optimal policy in about only 200 episodes. We explain this by the fact that the aKNAC critic takes into account the expectation over action (by replacing $\hat{Q}_{\theta,\xi}(s',a')$ by $g_\xi(s') = E_{a'|s'}[\hat{Q}_{\theta,\xi}(s',a')]$ in the observation equation). This provides a better state-action value function estimation, see Bellman Eq. (4). These results show empirically the validity of the proposed alternative actor-critic theorems, as well as the interest of using a second-order critic handling non-stationarities to speed up learning in an online setting.

7 Conclusion

In this paper we have presented alternative results concerning policy gradient with function approximation and natural policy gradient. They allow working directly with the state-action value function rather than with the advantage function. If these results do not change fundamentally the recent actor-critic theory, they allow deriving new critics in an easier way, as the state-action value function satisfies a Bellman equation, contrary to the advantage function. We have also illustrated the ease of critic design by introducing three new actor-critic algorithms, TD-NAC, KNAC and aKNAC. The first one is derived using TD with function approximation and a two-timescale approach, and it is close to the NTD algorithm [11] and to algorithm 3 of [7] as discussed before. KNAC and aKNAC are derived using the KTD framework of [10] which provides second-order function approximators able to handle non-stationarities. All these algorithms have been compared on the classic inverted pendulum task.

Actor-critic with function approximation is a very interesting paradigm. However, a number of questions remain open. An important problem is to design a critic which satisfies condition (12). So far, most of critics were designed in a batch setting [5] or using a two-timescale approach [7]. In this paper, we have proposed to use a critic which handles non-stationarities in order to satisfy this condition. Interesting perspectives would be to provide some guarantees for the proposed approach (that is choosing a provably appropriate evolution noise) and to discover new ways to ensure this condition. Another important point is the (semi-) compatibility condition and its implications. Actually, choosing a parameterization for the state-action value function or the policy

is a difficult and problem-dependent choice itself. This condition renders this choice even more difficult. An interesting perspective would be to propose some feature selection framework (that is learning the structure of the representation in addition to its parameters) for such actor-critic algorithms.

References

1. Barto, A.G., Sutton, R.S., Anderson, C.W.: Neuronlike adaptive elements that can solve difficult learning control problems. (1988) 535–549
2. Watkins, C.: Learning from Delayed Rewards. PhD thesis, Cambridge University, Cambridge, England (1989)
3. Sutton, R.S., McAllester, D.A., Singh, S.P., Mansour, Y.: Policy Gradient Methods for Reinforcement Learning with Function Approximation. In: Advances in Neural Information Processing Systems (NIPS 12). (2000) 1057–1063
4. Konda, V.R., Tsitsiklis, J.N.: Actor-Critic Algorithms. In: Advances in Neural Information Processing Systems (NIPS 12). (2000)
5. Peters, J., Vijayakumar, S., Schaal, S.: Reinforcement Learning for Humanoid Robotics. In: third IEEE-RAS International Conference on Humanoid Robots (Humanoids 2003). (2003)
6. Sutton, R.S., Barto, A.G.: Reinforcement Learning: An Introduction (Adaptive Computation and Machine Learning). 3rd edn. The MIT Press (March 1998)
7. Bhatnagar, S., Sutton, R.S., Ghavamzadeh, M., Lee, M.: Incremental Natural Actor-Critic Algorithms. In: Advances in Neural Information Processing Systems (NIPS 21), Vancouver, Canada (2007)
8. Amari, S.I.: Natural gradient works efficiently in learning. Neural Computation **10**(2) (1998) 251–276
9. Kakade, S.: A Natural Policy Gradient. In: Advances in Neural Information Processing Systems (NIPS 14). (2002) 1531–1538
10. Geist, M., Pietquin, O., Fricout, G.: Kalman Temporal Differences: the deterministic case. In: Proceedings of the IEEE International Symposium on Adaptive Dynamic Programming and Reinforcement Learning (ADPRL 2009), Nashville, TN, USA (April 2009)
11. Morimura, T., Uchibe, E., Doya, K.: Utilizing the Natural Gradient in Temporal Difference Reinforcement Learning with Eligibility Traces. In: 2nd Internatinal Symposium on Information Geometry and its Applications, Tokyo, Japan (2005) 256–263
12. Wiering, M., van Hasselt, H.: The QV Family Compared to Other Reinforcement Learning Algorithms. In: IEEE International Symposium on Adaptive Dynamic Programming and Reinforcement Learning (ADPRL 2009), Nashville, TN, USA (April 2009)
13. Bradtke, S.J., Barto, A.G.: Linear Least-Squares algorithms for temporal difference learning. Machine Learning **22**(1-3) (1996) 33–57
14. Geist, M., Pietquin, O., Fricout, G.: Tracking in reinforcement learning. In: Proceedings of the 16th International Conference on Neural Information Processing (ICONIP 2009). Volume 5863, Part I., Bangkok (Thailand), Springer LNCS (December 2009) 502–511
15. Park, J., Kim, J., Kang, D.: An RLS-Based Natural Actor-Critic Algorithm for Locomotion of a Two-Linked Robot Arm . In: International Conference on Computational Intelligence and Security (CIS 2005), Xi'an (China), Springer LNAI (2005) 65–72
16. Lagoudakis, M.G., Parr, R.: Least-Squares Policy Iteration. Journal of Machine Learning Research **4** (2003) 1107–1149