A Model-Based Actor-Critic Algorithm in Continuous Time and Space

Rémi Coulom REMI.COULOM@LORIA.FR
CORTEX Group, LORIA, Nancy, France

Abstract

This paper presents a model-based actor-critic algorithm in continuous time and space. Two function approximators are used: one learns the policy (the actor) and the other learns the state-value function (the critic). The critic learns with the TD(λ) algorithm and the actor by gradient ascent on the Hamiltonian. A similar algorithm had been proposed by Doya, but this one is more general. This algorithm was applied successfully to teach simulated articulated robots to swim.

1. Introduction

Although the traditional theoretical framework of reinforcement learning is discrete, this method can still be applied to decision problems in continuous time and space. It can be done either by discretizing the problem, or by using continuous formulations of learning algorithms. The latter approach avoids approximation errors introduced by the discretization of states and actions, so it is usually more efficient (Doya, 2000).

Unlike the usual purely critic TD(λ) method, which consists in using a greedy policy with respect to the estimated value function, the actor-critic algorithm uses two function approximators. An actor provides an action as a function of state, and a critic estimates the value function. They adjust each other during the learning process.

This kind of learning algorithm may seem uselessly more complex than the purely critic algorithm, but it still has some advantages. First, once learning is over, the critic is not necessary anymore, and the actor alone is enough to control the system. This actor has often a much lower computational cost than finding the greedy action with respect to some value function. Besides, using a continuous actor solves all the problems related to the discontinuity of the greedy control (Coulom, 2002). Lastly, although there is no convergence proof for the algorithm presented in this paper, the optimization performed in the actor-critic algorithm seems more theoretically sound, and may provide better convergence in practice than the purely critic algorithm.

The rest of this paper presents the formulation of this continuous actor-critic algorithm, and experimental results obtained with this method.

2. Algorithm

2.1 Problem Definition

In general, we will suppose that we are to solve motor problems defined by:

- states $\vec{x} \in S \subset \mathbb{R}^p$,
- controls $\vec{u} \in U \subset \mathbb{R}^q$,
- system dynamics $f : S \times U \mapsto \mathbb{R}^p$,
- a reward function $r : S \times U \mapsto \mathbb{R}$,
- a shortness factor $s_\gamma \geq 0$ ($\gamma = e^{-s_\gamma \delta t}$).

A *strategy* or *policy* is a function $\pi : S \mapsto U$ that maps states to controls. Applying a policy from a starting state $\vec{x}_0$ at time t_0 produces a trajectory $\vec{x}(t)$ defined by the ordinary differential equation

$$\begin{aligned} \forall t \geq t_0 \quad \dot{\vec{x}} &= f\big(\vec{x}, \pi(\vec{x})\big) \ , \\ \vec{x}(t_0) &= \vec{x}_0 \ . \end{aligned}$$

The value function of π is defined by

$$V^\pi(\vec{x}_0) = \int_{t=t_0}^{\infty} e^{-s_\gamma (t-t_0)} r\Big(\vec{x}(t), \pi\big(\vec{x}(t)\big)\Big) dt \ .$$

The goal is to find a policy that maximizes the total amount of reward over time, whatever the starting state $\vec{x}_0$. More formally, the problem consists in finding π^* so that

$$\forall \vec{x}_0 \in S \quad V^{\pi^*}(\vec{x}_0) = \max_{\pi : S \mapsto U} V^\pi(\vec{x}_0) \ .$$

2.2 TD(λ) Policy Evaluation

The algorithm used in experiments reported in this paper is Doya's (Doya, 2000) continuous TD(λ). It is a continuous version of Sutton's discrete algorithm.

In order to approximate the optimal value function V^* with a parametric function approximator $V_{\vec{w}}$, where $\vec{w}$ is the vector of weights (parameters), the continuous TD(λ) algorithm consists in integrating an ordinary differential equation:

$$\begin{cases} \dot{\vec{w}} = \eta \mathcal{H} \vec{e} \ , \\ \dot{\vec{e}} = -(s_\gamma + s_\lambda)\vec{e} + \dfrac{\partial V_{\vec{w}}(\vec{x})}{\partial \vec{w}} \ , \\ \dot{\vec{x}} = f\big(\vec{x}, \pi(\vec{x})\big) \ , \end{cases}$$

with

$$\mathcal{H} = r\big(\vec{x}, \pi(\vec{x})\big) - s_\gamma V_{\vec{w}}(\vec{x}) + \frac{\partial V_{\vec{w}}}{\partial \vec{x}} \cdot f\big(\vec{x}, \pi(\vec{x})\big) \ .$$

$\mathcal{H}$ is the Hamiltonian and is a continuous equivalent of Bellman's residual. $\mathcal{H} > 0$ indicates a "good surprise" and causes and increase in the past values, whereas $\mathcal{H} < 0$ is a "bad surprise" and causes a decrease in the past values. The magnitude of this change is controlled by the learning rate η, and its time extent in the past is defined by the parameter s_λ. s_λ can be related to the traditional λ parameter in the discrete algorithm by $\lambda = e^{-s_\lambda \delta t}$. $\vec{e}$ is the vector of eligibility traces. Learning is decomposed into several episodes, each starting from a random initial state, thus insuring exploration of the whole state space.

2.3 Policy Improvement

The actor-critic algorithm consists in using a parametric function approximator for the policy. $\pi_{\vec{\theta}}$ depends on a vector of parameters $\vec{\theta}$. $\vec{\theta}$ varies according to:

$$\dot{\vec{\theta}} = \eta_\theta \frac{\partial \mathcal{H}}{\partial \vec{\theta}} \ .$$

Gradient ascent of the Hamiltonian is a classical technique for parametric optimization of policies in finite-horizon deterministic optimal control problems (White & Jordan, 1992). Those are purely critic methods that estimate the value gradient thanks to Pontryagin's maximum principle. The actor-critic approach is based on Bellman's maximum principle and allows to apply this method to infinite-horizon and stochastic problems.

This gradient-ascent equation can also be viewed as a continuous equivalent of a discrete theorem proved in (Marbach & Tsitsiklis, 2001). In fact, this gradient-ascent method is justified in the average-reward case, not the discounted case. So, the algorithm presented here should be re-formulated in the average-reward framework to be a little more sound.

This actor-critic algorithm is a generalization of what Doya proposed (Doya, 2000). The actor-critic algorithm he described is in fact similar to gradient ascent on the Hamiltonian, but it works only in the particular case of the pendulum swing-up task. The algorithm presented here can be applied to any continuous problem.

3. Experiments

Experiments were run with the swimmer problem described in (Coulom, 2002), using feedforward neural networks as actors and critics. The performance of the actor-critic algorithm looks similar to what was obtained in the purely critic case: swimmers made progress at roughly the same speed. There were also instabilities. But they are of a different kind, and swimming techniques obtained differ a little. It seems that the more neurons in the critic, the more stable the algorithm: a higher number of neurons provides a more accurate estimation of the value function, so it is likely to provide a better direction for gradient ascent for policy improvement. More experiments are required to test this further.

Acknowledgements

I thank Rémi Munos for his comments that helped to improve this paper.

References

Coulom, R. (2002). *Reinforcement learning using neural networks, with applications to motor control.* Doctoral dissertation, Institut National Polytechnique de Grenoble.

Doya, K. (2000). Reinforcement learning in continuous time and space. *Neural Computation*, *12*, 243–269.

Marbach, P., & Tsitsiklis, J. N. (2001). Simulation-based optimization of markov reward processes. *IEEE Transactions on Automatic Control*, *46*, 191–209.

White, D. A., & Jordan, M. I. (1992). Optimal control: A foundation for intelligent control. In D. A. White and D. A. Sofge (Eds.), *Handbook of intelligent control—neural, fuzzy, and adaptative approaches.* New York: Van Nostrand Reinhold.