Engineering Applications of Artificial Intelligence 69 (2018) 178–187

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

journal homepage: www.elsevier.com/locate/engappai

Policy derivation methods for critic-only reinforcement learning in continuous spaces

Eduard Alibekov [a], Jiří Kubalík [a], Robert Babuška [b,a,*]

[a] Czech Institute of Informatics, Robotics, and Cybernetics, Czech Technical University in Prague, Prague, Czech Republic
[b] Cognitive Robotics, Faculty 3mE, Delft University of Technology, Delft, The Netherlands

ARTICLE INFO

Keywords:
Reinforcement learning
Continuous actions
Multi-variable systems
Optimal control
Policy derivation
Optimization

ABSTRACT

This paper addresses the problem of deriving a policy from the value function in the context of critic-only reinforcement learning (RL) in continuous state and action spaces. With continuous-valued states, RL algorithms have to rely on a numerical approximator to represent the value function. Numerical approximation due to its nature virtually always exhibits artifacts which damage the overall performance of the controlled system. In addition, when continuous-valued action is used, the most common approach is to discretize the action space and exhaustively search for the action that maximizes the right-hand side of the Bellman equation. Such a policy derivation procedure is computationally involved and results in steady-state error due to the lack of continuity. In this work, we propose policy derivation methods which alleviate the above problems by means of action space refinement, continuous approximation, and post-processing of the V-function by using symbolic regression. The proposed methods are tested on nonlinear control problems: 1-DOF and 2-DOF pendulum swing-up problems, and on magnetic manipulation. The results show significantly improved performance in terms of cumulative return and computational complexity.

1. Introduction

Reinforcement Learning (RL) algorithms provide a way to solve dynamic decision-making and control problems (Sutton and Barto, 1998; Polydoros and Nalpantidis, 2017; Kuvayev and Sutton, 1996). An RL agent interacts with the system to be controlled by measuring its states and applying actions according to a certain policy. After applying an action, the agent receives a scalar reward related to the immediate performance. The goal is to find an optimal policy, which maximizes the cumulative reward.

The available RL algorithms can be broadly classified into critic-only, actor-only, and actor-critic methods (Konda and Tsitsiklis, 2000). Critic-only methods first find the value function (V-function) and then derive an optimal policy from this value function. In contrast, actor-only methods search directly in the policy space. The two approaches can be combined into actor-critic architectures, where the actor and critic are both represented explicitly and trained simultaneously. Each class can be further divided into model-based and model-free algorithms. In the model-based scenario, a system model is used during learning or policy derivation. The system model may be stochastic or deterministic. In this paper, we consider the critic-only, model-based and deterministic variant of RL in continuous state and action spaces. For the methods developed, it is irrelevant whether the system model is available a priori or learnt online.

We address the policy derivation step, assuming that an approximation of the true unknown V-function has already been computed. Policy derivation can be understood as a hill climbing process: at each time step, the agent searches for the control input that leads to a state with a highest value given by the right-hand side (RHS) of the Bellman equation. An advantage of this control law is its inherent stability—the value function is analogous to the control Lyapunov function (Lewis et al., 2012; Primbs et al., 1999). However, direct policy derivation from the V-function suffers from several problems:

- **Computational inefficiency**. The most common approach to dealing with a continuous action space is to discretize it into a small number of actions, compute the value of the Bellman equation RHS for all of them, and select the one that corresponds to the largest value (Sutton and Barto, 1998; Bertsekas, 2011). The number of possible discrete actions grows exponentially with the dimension of the action space and so does the computational complexity of this method.

* Corresponding author at: Cognitive Robotics, Faculty 3mE, Delft University of Technology, Delft, The Netherlands.
E-mail addresses: eduard.alibekov@cvut.cz (E. Alibekov), jiri.kubalik@cvut.cz (J. Kubalík), r.babuska@tudelft.nl (R. Babuška).

https://doi.org/10.1016/j.engappai.2017.12.004
Received 15 December 2016; Received in revised form 16 October 2017; Accepted 5 December 2017
Available online 8 January 2018
0952-1976/

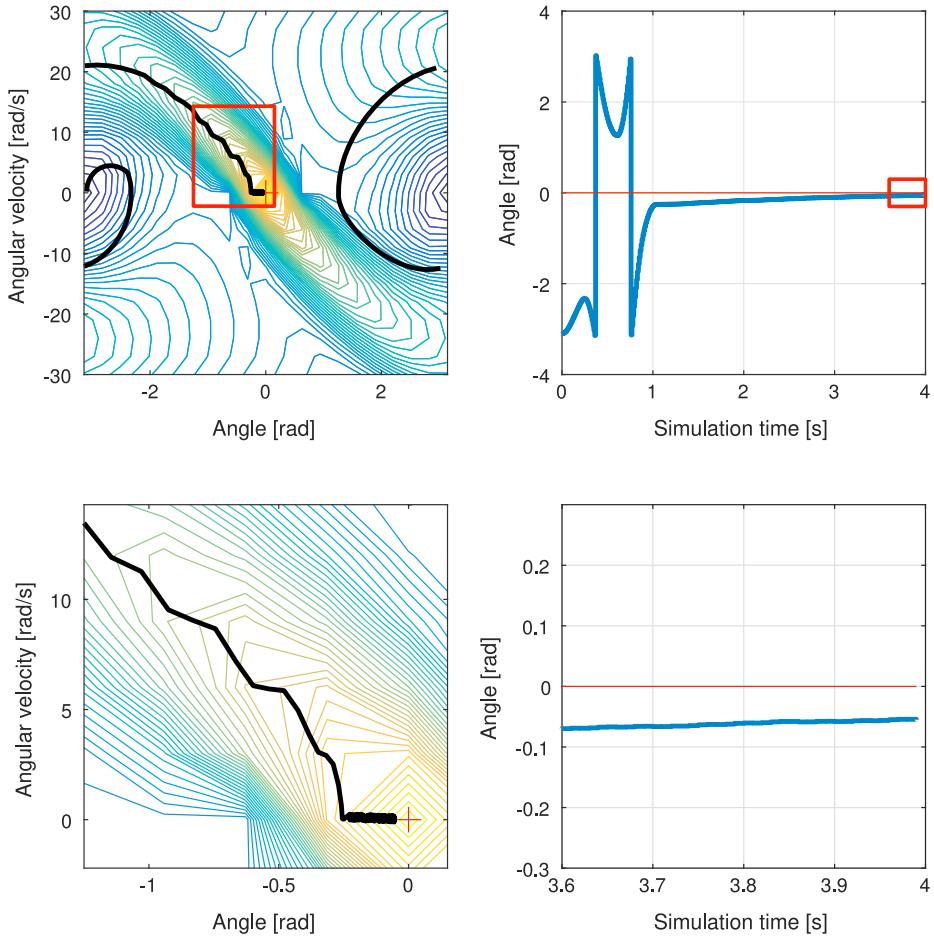

Fig. 1. A sample state trajectory obtained by simulating the pendulum swing-up task (see Section 5 for details). The bottom plots show an enlarged view of the areas indicated in the upper plots. The wiggly state trajectory (superimposed on the contours of the Bellman equation RHS) and the extremely slow convergence to the desired position result from the insufficient smoothness of the V-function approximation in combination with the use of discrete actions.

- **Insufficient smoothness of the V-function**. The above hill-climbing process is adversely affected by the approximate nature of the V-function, which has been observed e.g. in (Alibekov et al., 2016b). A typical approximation by means of basis functions exhibits artifacts which can lead to oscillations, as illustrated in the left column of Fig. 1. We refer to this problem by the term "insufficient smoothness", without relying on the exact mathematical definition of smoothness.
- **Discrete-valued control input**. The use of discrete actions leads in combination with insufficient smoothness to steady-state errors, as shown in the right column of Fig. 1. In the long run, the steady-state error can induce big losses in terms of the overall performance.

The aim of this paper is to alleviate the above problems. We extend our earlier work on policy derivation methods (Alibekov et al., 2016a). In addition to the methods originally proposed in this paper we introduce symbolic regression to address the V-function smoothness problem. Additionally, to enable the use of continuous actions, we propose a method based on a computationally efficient optimization technique.

The paper is organized as follows. Related work is discussed in Section 2 and Section 3 reviews the necessary background of reinforcement learning. The proposed policy derivation methods are introduced in Section 4. The results obtained on several benchmark problems are presented in Section 5 and discussed in detail in Section 6. Finally, Section 7 concludes the paper.

2. Related work

The problem of deriving policies for continuous state and action spaces in critic-only methods has not been sufficiently addressed in the literature. The most common approach is to discretize the action space, compute the RHS of the Bellman equation for all the discrete actions, and select the action that corresponds to the largest value. One of the earliest references to this approach is (Santamaria et al., 1996). The drawbacks of this method were discussed in the previous section.

Another similar approach is based on sampling (Sallans and Hinton, 2004; Kimura, 2007). Using Monte Carlo estimation, this approach can find a near-optimal action without resorting to exhaustive search over the discretized action space. However, for a good performance, this method requires a large number of samples and therefore it is computationally inefficient.

An alternative method would be *policy interpolation* (Busoniu et al., 2010), which is based on computing the control actions off-line for a pre-selected set of states and then interpolating these actions online. While computationally less involved, this method does not give any closed-loop stability guarantees and can suffer from severe interpolation errors, especially in constrained problems. Therefore, we do not consider policy interpolation in this paper.

A different approach relies on translating the continuous action selection step into a sequence of binary decisions (Pazis and Lagoudakis, 2009, 2011). Each decision whether to decrease or increase the control action eliminates a part of the action space. This process stops once a predefined precision is reached. There are two main drawbacks of this approach: it requires a binary code representation of each action, which