Neural Networks 21 (2008) 682–697

Contents lists available at ScienceDirect

Neural Networks

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

2008 Special Issue

Reinforcement learning of motor skills with policy gradients

Jan Peters[a,b,*], Stefan Schaal[b,c]

[a] Max Planck Institute for Biological Cybernetics, Spemannstr. 38, 72076 Tübingen, Germany
[b] University of Southern California, 3710 S. McClintoch Ave – RTH401, Los Angeles, CA 90089-2905, USA
[c] ATR Computational Neuroscience Laboratory, 2-2-2 Hikaridai, Seika-cho, Soraku-gun Kyoto 619-0288, Japan

ARTICLE INFO

Article history:
Received 11 September 2007
Received in revised form
24 February 2008
Accepted 24 February 2008

Keywords:
Reinforcement learning
Policy gradient methods
Natural gradients
Natural Actor-Critic
Motor skills
Motor primitives

ABSTRACT

Autonomous learning is one of the hallmarks of human and animal behavior, and understanding the principles of learning will be crucial in order to achieve true autonomy in advanced machines like humanoid robots. In this paper, we examine learning of complex motor skills with human-like limbs. While supervised learning can offer useful tools for bootstrapping behavior, e.g., by learning from demonstration, it is only reinforcement learning that offers a general approach to the final trial-and-error improvement that is needed by each individual acquiring a skill. Neither neurobiological nor machine learning studies have, so far, offered compelling results on how reinforcement learning can be scaled to the high-dimensional continuous state and action spaces of humans or humanoids. Here, we combine two recent research developments on learning motor control in order to achieve this scaling. First, we interpret the idea of modular motor control by means of motor primitives as a suitable way to generate parameterized control policies for reinforcement learning. Second, we combine motor primitives with the theory of stochastic policy gradient learning, which currently seems to be the only feasible framework for reinforcement learning for humanoids. We evaluate different policy gradient methods with a focus on their applicability to parameterized motor primitives. We compare these algorithms in the context of motor primitive learning, and show that our most modern algorithm, the Episodic Natural Actor-Critic outperforms previous algorithms by at least an order of magnitude. We demonstrate the efficiency of this reinforcement learning method in the application of learning to hit a baseball with an anthropomorphic robot arm.

1. Introduction

In order to ever leave the well-structured environments of factory floors and research labs, future robots will require the ability to acquire novel behaviors and motor skills as well as to improve existing ones based on rewards and costs. Similarly, the understanding of human motor control would benefit significantly if we can synthesize simulated human behavior and its underlying cost functions based on insight from machine learning and biological inspirations. Reinforcement learning is probably the most general framework in which such learning problems of computational motor control can be phrased. However, in order to bring reinforcement learning into the domain of human movement learning, two deciding components need to be added to the standard framework of reinforcement learning: first, we need a domain-specific policy representation for motor skills, and, second, we need reinforcement learning algorithms which work efficiently with this representation while scaling into the domain of high-dimensional mechanical systems such as humanoid robots.

Traditional representations of motor behaviors in robotics are mostly based on desired trajectories generated from spline interpolations between points, i.e., spline nodes, which are part of a longer sequence of intermediate target points on the way to a final movement goal. While such a representation is easy to understand, the resulting control policies, generated from a tracking controller of the spline trajectories, have a variety of significant disadvantages, including that they are time indexed and thus not robust towards unforeseen disturbances, that they do not easily generalize to new behavioral situations without complete recomputation of the spline, and that they cannot easily be coordinated with other events in the environment, e.g., synchronized with other sensory variables like visual perception during catching a ball. In the literature, a variety of other approaches for parameterizing movement have been suggested to overcome these problems, see Ijspeert, Nakanishi, and Schaal (2002, 2003) for more information. One of these approaches proposed using parameterized nonlinear dynamical systems as motor primitives, where the attractor properties of these dynamical systems defined the

[*] Corresponding author at: Max Planck Institute for Biological Cybernetics, Spemannstr. 38, 72076 Tübingen, Germany.
E-mail address: jan.peters@tuebingen.mpg.de (J. Peters).

0893-6080/$ – see front matter © 2008 Elsevier Ltd. All rights reserved.
doi:10.1016/j.neunet.2008.02.003

desired behavior (Ijspeert et al., 2002, 2003). The resulting framework was particularly well suited for supervised imitation learning in robotics, exemplified by examples from humanoid robotics where a full-body humanoid learned tennis swings or complex polyrhythmic drumming patterns. One goal of this paper is the application of reinforcement learning to both traditional spline-based representations as well as the more novel dynamic system based approach.

However, despite the fact that reinforcement learning is the most general framework for discussing the learning of movement in general, and motor primitives for robotics in particular, most of the methods proposed in the reinforcement learning community are not applicable to high-dimensional systems such as humanoid robots. Among the main problems are that these methods do not scale beyond systems with more than three or four degrees of freedom and/or cannot deal with parameterized policies. Policy gradient methods are a notable exception to this statement. Starting with the pioneering work[1] of Gullapali and colleagues (Benbrahim & Franklin, 1997; Gullapalli, Franklin, & Benbrahim, 1994) in the early 1990s, these methods have been applied to a variety of robot learning problems ranging from simple control tasks (e.g., balancing a ball on a beam (Benbrahim, Doleac, Franklin, & Selfridge, 1992), and pole balancing (Kimura & Kobayashi, 1998)) to complex learning tasks involving many degrees of freedom such as learning of complex motor skills (Gullapalli et al., 1994; Mitsunaga, Smith, Kanda, Ishiguro, & Hagita, 2005; Miyamoto et al., 1995, 1996; Peters & Schaal, 2006; Peters, Vijayakumar, & Schaal, 2005a) and locomotion (Endo, Morimoto, Matsubara, Nakanishi, & Cheng, 2005; Kimura & Kobayashi, 1997; Kohl & Stone, 2004; Mori, Nakamura, aki Sato, & Ishii, 2004; Nakamura, Mori, & Ishii, 2004; Sato, Nakamura, & Ishii, 2002; Tedrake, Zhang, & Seung, 2005).

The advantages of policy gradient methods for parameterized motor primitives are numerous. Among the most important ones are that the policy representation can be chosen such that it is meaningful for the task, i.e., we can use a suitable motor primitive representation, and that domain knowledge can be incorporated, which often leads to fewer parameters in the learning process in comparison to traditional value function based approaches. Moreover, there exist a variety of different algorithms for policy gradient estimation in the literature, most with rather strong theoretical foundations. Additionally, policy gradient methods can be used model-free and therefore also be applied to problems without analytically known task and reward models.

Nevertheless, many recent publications on applications of policy gradient methods in robotics overlooked the newest developments in policy gradient theory and their original roots in the literature. Thus, a large number of heuristic applications of policy gradients can be found, where the success of the projects mainly relied on ingenious initializations and manual parameter tuning of algorithms. A closer inspection often reveals that the chosen methods might be statistically biased, or even generate infeasible policies under less fortunate parameter settings, which could lead to unsafe operation of a robot. The main goal of this paper is to discuss which policy gradient methods are applicable to robotics and which issues matter, while also introducing some new policy gradient learning algorithms that seem to have superior performance over previously suggested methods. The remainder of this paper will proceed as follows: firstly, we will introduce the general assumptions of reinforcement learning, discuss motor primitives in this framework and pose the problem statement of this paper. Secondly, we will analyze the different approaches to policy gradient estimation and discuss their applicability to reinforcement learning of motor primitives. We focus on the most useful methods and examine several algorithms in depth. The presented algorithms in this paper are highly optimized versions of both novel and previously published policy gradient algorithms. Thirdly, we show how these methods can be applied to motor skill learning in humanoid robotics and show learning results with a seven degree of freedom, anthropomorphic SARCOS Master Arm.

1.1. General assumptions and problem statement

Most robotics domains require the state-space and the action spaces to be continuous and high dimensional such that learning methods based on discretizations are not applicable for higher-dimensional systems. However, as the policy is usually implemented on a digital computer, we assume that we can model the control system in a discrete-time manner and we will denote the current time step[2] by k. In order to take possible stochasticity of the plant into account, we denote it using a probability distribution

$$\boldsymbol{x}_{k+1} \sim p\left(\boldsymbol{x}_{k+1} \,|\boldsymbol{x}_k, \boldsymbol{u}_k\right) \tag{1}$$

where $\boldsymbol{u}_k \in \mathbb{R}^M$ denotes the current action, and $\boldsymbol{x}_k, \boldsymbol{x}_{k+1} \in \mathbb{R}^N$ denote the current and the next state respectively. We furthermore assume that actions are generated by a policy

$$\boldsymbol{u}_k \sim \pi_{\boldsymbol{\theta}}\left(\boldsymbol{u}_k \,|\boldsymbol{x}_k\right) \tag{2}$$

which is modeled as a probability distribution in order to incorporate exploratory actions; for some special problems, the optimal solution to a control problem is actually a stochastic controller, see e.g., Sutton, McAllester, Singh, and Mansour (2000). The policy is parameterized by some policy parameters $\boldsymbol{\theta} \in \mathbb{R}^K$ and assumed to be continuously differentiable with respect to its parameters $\boldsymbol{\theta}$. The sequence of states and actions forms a trajectory (also called history or roll-out) denoted by $\boldsymbol{\tau} = [\boldsymbol{x}_{0:H}, \boldsymbol{u}_{0:H}]$ where H denotes the horizon, which can be infinite. At each instant of time, the learning system receives a reward denoted by $r\left(\boldsymbol{x}_k, \boldsymbol{u}_k\right) \in \mathbb{R}$.

The general goal of policy gradient reinforcement learning is to optimize the policy parameters $\boldsymbol{\theta} \in \mathbb{R}^K$ so that the expected return

$$J(\boldsymbol{\theta}) = \frac{1}{a_\Sigma} E\left\{\sum_{k=0}^{H} a_k r_k\right\} \tag{3}$$

is optimized where a_k denote time-step-dependent weighting factors and a_Σ is a normalization factor in order to ensure that the normalized weights a_k/a_Σ sum up to one. We require that the weighting factors fulfill $a_{l+k} = a_l a_k$ in order to be able to connect to the previous policy gradient literature; examples are the weights $a_k = \gamma^k$ for discounted reinforcement learning (where γ is in $[0, 1]$) where $a_\Sigma = 1/(1-\gamma)$; alternatively, they are set to $a_k = 1$ for the average reward case where $a_\Sigma = H$. In these cases, we can rewrite *a normalized expected return* in the form

$$J(\boldsymbol{\theta}) = \int_{\mathbb{X}} d^\pi(\boldsymbol{x}) \int_{\mathbb{U}} \pi(\boldsymbol{u}|\boldsymbol{x}) r(\boldsymbol{x}, \boldsymbol{u}) \mathrm{d}\boldsymbol{x} \mathrm{d}\boldsymbol{u} \tag{4}$$

[1] Note that there has been earlier work by the control community, see e.g., Dyer and McReynolds (1970), Hasdorff (1976) and Jacobson and Mayne (1970), which is based on exact analytical models. Extensions based on learned, approximate models originated in the literature on optimizing government decision policies, see Werbos (1979), and have also been applied in control (Atkeson, 1994; Morimoto & Atkeson, 2003). In this paper, we limit ourselves to model-free approaches as the most general framework, while future work will address specialized extensions to model-based learning.

[2] Note, that throughout this paper, we will use k and l for denoting discrete steps, m for update steps and h for the current vector element, e.g., $\boldsymbol{\theta}_h$ denotes the hth element of $\boldsymbol{\theta}$.