Automatica 46 (2010) 804–814

Contents lists available at ScienceDirect

Automatica

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

Approximate dynamic programming with a fuzzy parameterization[☆]

Lucian Buşoniu[a,*], Damien Ernst[c], Bart De Schutter[a,b], Robert Babuška[a]

[a] *Delft Center for Systems & Control, Delft University of Technology, Mekelweg 2, 2628 CD Delft, The Netherlands*

[b] *Marine & Transport Technology, Delft University of Technology, The Netherlands*

[c] *FNRS; Institut Montefiore, Univ. Liège, Sart-Tilman, Bldg. B28, Parking P32, B-4000 Liège, Belgium*

ARTICLE INFO

Article history:
Received 15 December 2008
Received in revised form
25 January 2010
Accepted 1 February 2010
Available online 11 March 2010

Keywords:
Approximate dynamic programming
Fuzzy approximation
Value iteration
Convergence analysis

ABSTRACT

Dynamic programming (DP) is a powerful paradigm for general, nonlinear optimal control. Computing exact DP solutions is in general only possible when the process states and the control actions take values in a small discrete set. In practice, it is necessary to approximate the solutions. Therefore, we propose an algorithm for approximate DP that relies on a fuzzy partition of the state space, and on a discretization of the action space. This *fuzzy Q-iteration* algorithm works for deterministic processes, under the discounted return criterion. We prove that fuzzy Q-iteration asymptotically converges to a solution that lies within a bound of the optimal solution. A bound on the suboptimality of the solution obtained in a finite number of iterations is also derived. Under continuity assumptions on the dynamics and on the reward function, we show that fuzzy Q-iteration is consistent, i.e., that it asymptotically obtains the optimal solution as the approximation accuracy increases. These properties hold both when the parameters of the approximator are updated in a synchronous fashion, and when they are updated asynchronously. The asynchronous algorithm is proven to converge at least as fast as the synchronous one. The performance of fuzzy Q-iteration is illustrated in a two-link manipulator control problem.

1. Introduction

Dynamic programming (DP) is a powerful paradigm for solving optimal control problems, thanks to its mild assumptions on the controlled process, which can be nonlinear or stochastic (Bertsekas, 2007; Bertsekas & Tsitsiklis, 1996). In the DP framework, a model of the process is assumed to be available, and the immediate performance is measured by a scalar reward signal. The controller then maximizes the long-term performance, measured by the cumulative reward. DP algorithms can be extended to work without requiring a model of the process, in which case they are usually called reinforcement learning (RL) algorithms (Sutton & Barto, 1998). Most DP and RL algorithms work by estimating an optimal value function, i.e., the maximal cumulative reward as a function of the process state and possibly also of the control action. Representing value functions exactly is only possible when the state-action space contains a relatively small number of discrete elements. In large discrete spaces and in continuous spaces, the value function generally has to be approximated. This is especially the case in automatic control, where the state and action variables are usually continuous.

Therefore, this paper proposes an algorithm for approximate DP that represents state-action value functions (called Q-functions in the sequel) using a fuzzy rule base with singleton consequents (Kruse, Gebhardt, & Klowon, 1994, Section 4.2). This algorithm works for deterministic problems, under the discounted return criterion. It is called *fuzzy Q-iteration*, because it combines the classical Q-iteration algorithm with a fuzzy approximator. The fuzzy rule base receives the state as input, and produces the Q-values of the discrete actions as outputs. The set of discrete actions is selected beforehand from the (possibly continuous) original action space. The membership functions of the fuzzy antecedents can also be seen as state-dependent basis functions or features (Bertsekas & Tsitsiklis, 1996).

We show that fuzzy Q-iteration asymptotically converges to an approximate Q-function that lies within a bounded distance from the optimal Q-function. The suboptimality of the Q-function obtained after a finite number of iterations is also bounded. Both of these Q-functions lead to policies with a bounded suboptimality.

[☆] This research was financially supported by the BSIK-ICIS project (grant no. BSIK03024) and by the STW-VIDI project DWV.6188. The material in this paper was partially presented at IEEE International Conference on Fuzzy Systems 2007, Imperial College, London, UK, 23–26 July 2007; IEEE International Conference on Fuzzy Systems 2008 (IEEE World Congress on Computational Intelligence 2008), Hong Kong, 1–6 June 2008. This paper was recommended for publication in revised form by Associate Editor Zongli Lin under the direction of Editor Andrew R. Teel. We thank Pierre Geurts for allowing us the use of his code for building ensembles of regression trees.

[*] Corresponding author. Tel.: +31 152788573; fax: +31 152786679.
E-mail addresses: i.l.busoniu@tudelft.nl (L. Buşoniu), dernst@ulg.ac.be (D. Ernst), b.deschutter@tudelft.nl (B. De Schutter), r.babuska@tudelft.nl (R. Babuška).

0005-1098/$ – see front matter © 2010 Elsevier Ltd. All rights reserved.
doi:10.1016/j.automatica.2010.02.006

We also show that fuzzy Q-iteration is consistent: under appropriate continuity assumptions on the process dynamics and on the reward function, the approximate Q-function converges to the optimal one as the approximation accuracy increases. These properties hold both when the parameters of the approximator are updated in a synchronous fashion, and when they are updated asynchronously. Additionally, the asynchronous algorithm is proven to converge at least as fast as the synchronous one. In a simulation example, fuzzy Q-iteration is used to control a two-link manipulator, and compared with the state-of-the-art fitted Q-iteration algorithm (Ernst, Geurts, & Wehenkel, 2005).

The remainder of this paper is structured as follows. Section 2 gives a brief overview of the literature related to our results. Section 3 describes Markov decision processes and the Q-iteration algorithm. Section 4 introduces fuzzy Q-iteration. This novel algorithm is analyzed in Section 5, and applied to a two-link manipulator example in Section 6. Section 7 concludes the paper and outlines some ideas for future work.

2. Related work

A rich body of literature concerns the analysis of approximate value iteration, both in the DP (model-based) setting (Chow & Tsitsiklis, 1991; Gordon, 1995; Munos & Szepesvári, 2008; Santos & Vigo-Aguiar, 1998; Tsitsiklis & Van Roy, 1996) and in the RL (model-free) setting (Antos, Munos, & Szepesvári, 2008; Farahmand, Ghavamzadeh, Szepesvári, & Mannor, 2009; Szepesvári & Smart, 2004). In many cases, convergence is ensured by using linearly parameterized approximators (Gordon, 1995; Szepesvári & Smart, 2004; Tsitsiklis & Van Roy, 1996). Our convergence analysis for synchronous fuzzy Q-iteration is related to the convergence analysis of approximate V-iteration for discrete state-action spaces in Gordon (1995) and Tsitsiklis and Van Roy (1996). We additionally consider continuous state-action spaces, introduce an explicit discretization procedure for the continuous actions, consider asynchronous fuzzy Q-iteration, and study the finite-time performance of the algorithm. While (exact) asynchronous value iteration is widely known (Bertsekas, 2007, Section 1.3.2), asynchronous algorithms for approximate DP are not often studied. Many consistency results for model-based (DP) algorithms are found for discretization-based approximators (Chow & Tsitsiklis, 1991; Santos & Vigo-Aguiar, 1998). Such discretizations sometimes use interpolation schemes similar to fuzzy approximation. A different class of results analyzes the performance of approximate value iteration for stochastic processes, when only a limited number of samples are available (Antos et al., 2008; Farahmand et al., 2009; Munos & Szepesvári, 2008).

Fuzzy approximators have typically been used in model-free (RL) techniques such as Q-learning (Glorennec, 2000; Horiuchi, Fujino, Katai, & Sawaragi, 1996; Jouffe, 1998) and actor-critic algorithms (Berenji & Vengerov, 2003; Lin, 2003). Most of these approaches are heuristic in nature, and their theoretical properties have not been investigated yet. In this paper, we use fuzzy approximation with a *model-based* (DP) algorithm, and provide a detailed analysis of its convergence and consistency properties.

The present paper integrates and significantly extends the authors' earlier work on fuzzy Q-iteration (Buşoniu, Ernst, De Schutter, & Babuška, 2007, 2008a,b) by removing some limiting assumptions: an originally discrete action space in Buşoniu et al. (2007), Buşoniu et al. (2008b), and a restrictive bound on the Lipschitz constant of the process dynamics in Buşoniu et al. (2008a). Additionally, the solution obtained in a finite time is analyzed, and the suboptimality of the approximate solution is explicitly related to accuracy of the fuzzy approximator.

3. Markov decision processes and Q-iteration

This section introduces deterministic Markov decision processes (MDPs) and characterizes their optimal solution (Bertsekas, 2007; Sutton & Barto, 1998). Afterwards, exact and approximate Q-iteration are presented.

A deterministic MDP consists of the state space X, the action space U, the transition function $f : X \times U \to X$, and the reward function $\rho : X \times U \to \mathbb{R}$. As a result of the control action u_k applied in the state x_k, the state changes to $x_{k+1} = f(x_k, u_k)$ and a scalar reward $r_{k+1} = \rho(x_k, u_k)$ is generated, which evaluates the immediate effect of action u_k (the transition from x_k to x_{k+1}). The state and action spaces can be continuous or discrete. We assume that $\|\rho\|_\infty = \sup_{x,u} |\rho(x, u)|$ is finite. Actions are chosen according to the policy $h : X \to U$, which is a discrete-time state feedback $u_k = h(x_k)$.

The goal is to find an optimal policy, i.e., one that maximizes, starting from the current moment in time ($k = 0$) and from any initial state x_0, the discounted return:

$$R^h(x_0) = \sum_{k=0}^{\infty} \gamma^k r_{k+1} = \sum_{k=0}^{\infty} \gamma^k \rho(x_k, h(x_k)) \tag{1}$$

where the discount factor $\gamma \in [0, 1)$ and $x_{k+1} = f(x_k, h(x_k))$ for $k \geq 0$. Because the rewards are bounded, the infinite sum in (1) exists and is bounded. The task is therefore to maximize the long-term performance (return), while only using feedback about the immediate, one-step performance (reward).

Optimal policies can be conveniently characterized by the optimal Q-function $Q^* : X \times U \to \mathbb{R}$. For every pair (x, u), the optimal Q-function gives the discounted return obtained by first applying u in x, and then selecting actions optimally:

$$Q^*(x, u) = \rho(x, u) + \gamma \sup_h R^h(f(x, u)). \tag{2}$$

An optimal policy h^* can be found from Q^*, by ensuring that $h^*(x) \in \arg\max_u Q^*(x, u)$. Under mild technical assumptions, such a policy exists. In general, for a given Q, a policy h that satisfies $h(x) \in \arg\max_u Q(x, u)$ is called *greedy* in Q.

Let the set of all the Q-functions be denoted by $\mathbb{Q}$. Define the Q-iteration mapping $T : \mathbb{Q} \to \mathbb{Q}$:

$$[T(Q)](x, u) = \rho(x, u) + \gamma \sup_{u'} Q(f(x, u), u'). \tag{3}$$

The optimal Q-function satisfies the Bellman optimality equation $Q^* = T(Q^*)$ (Bertsekas, 2007, Section 6.4). So, Q^* is a fixed point of T. The *Q-iteration* algorithm starts from an arbitrary Q-function Q_0 and in each iteration ℓ updates it by using:

$$Q_{\ell+1} = T(Q_\ell). \tag{4}$$

It is well-known that T is a contraction with factor $\gamma < 1$ in the infinity norm, i.e., for any pair of functions Q and Q', it is true that $\|T(Q) - T(Q')\|_\infty \leq \gamma \|Q - Q'\|_\infty$. This can be shown e.g., by extending the analogous result for V-functions given in Bertsekas and Tsitsiklis (1996, Section 2.3). Therefore, Q^* is the *unique* fixed point of T, and Q-iteration converges to it as $\ell \to \infty$. Note that to implement (4), a model of the MDP is required, in the form of the transition and reward functions f, ρ.

In general, Q-iteration requires to store and update distinct Q-values for every state-action pair. This is only possible when the states and actions take values in a finite, discrete set. When the state or action variables are continuous, there are infinitely many state-action pairs, and the Q-function has to be represented approximately. Even when the number of state-actions pairs is finite but very large, exact Q-iteration might be impractical and approximation may be useful.

In this paper, we consider algorithms for *approximate Q-iteration* that parameterize the Q-function using a vector $\theta \in \mathbb{R}^n$. In addition to the Q-iteration mapping T (3), two other mappings are needed to formalize approximate Q-iteration. The *approximation* mapping $F : \mathbb{R}^n \to \mathbb{Q}$ produces an approximate Q-function