Computers & Operations Research 43 (2014) 108–115

Contents lists available at ScienceDirect

Computers & Operations Research

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

Low-discrepancy sampling for approximate dynamic programming with local approximators

C. Cervellera *, M. Gaggero, D. Macciò

Institute of Intelligent Systems for Automation, National Research Council, Via De Marini 6, 16149 Genova, Italy

ARTICLE INFO

Available online 17 September 2013

Keywords:
Approximate dynamic programming
Low-discrepancy sampling
Local approximation
Nadaraya–Watson models
Inventory forecasting

ABSTRACT

Approximate dynamic programming (ADP) relies, in the continuous-state case, on both a flexible class of models for the approximation of the value functions and a smart sampling of the state space for the numerical solution of the recursive Bellman equations. In this paper, low-discrepancy sequences, commonly employed for number-theoretic methods, are investigated as a sampling scheme in the ADP context when local models, such as the Nadaraya–Watson (NW) ones, are employed for the approximation of the value function. The analysis is carried out both from a theoretical and a practical point of view. In particular, it is shown that the combined use of low-discrepancy sequences and NW models enables the convergence of the ADP procedure. Then, the regular structure of the low-discrepancy sampling is exploited to derive a method for automatic selection of the bandwidth of NW models, which yields a significant saving in the computational effort with respect to the standard cross validation approach. Simulation results concerning an inventory management problem are presented to show the effectiveness of the proposed techniques.

1. Introduction

Consider a Markovian Decision Problem (MDP) characterized by a continuous-state discrete-time dynamic system evolving according to the stochastic state equation

$$\boldsymbol{x}_{t+1} = \boldsymbol{f}(\boldsymbol{x}_t, \boldsymbol{u}_t, \boldsymbol{\theta}_t),$$

where $t = 0, \ldots, T-1$, $\boldsymbol{f}$ is a smooth vectorial field, $\boldsymbol{x}_t \in X_t \subset \mathbb{R}^n$ is the state vector, $\boldsymbol{u}_t \in U_t \subset \mathbb{R}^m$ is a decision (or control) vector and $\boldsymbol{\theta}_t \in \Theta_t \subset \mathbb{R}^q$ is a random vector affecting the system. The vectors $\boldsymbol{\theta}_t$ are characterized by a probability measure $P(\boldsymbol{\theta}_t)$ and constitute an independent chain of vectors over the T stages.

The problem consists in finding optimal decision vectors that minimize a cost function depending on the evolution of the state, which has an additive form over T stages:

$$J = \sum_{t=0}^{T-1} h(\boldsymbol{x}_t, \boldsymbol{u}_t, \boldsymbol{\theta}_t) + h_T(\boldsymbol{x}_T),$$

where h and h_T are suitable single-stage cost functions.

Since the system is affected by random quantities, we look for closed-loop decision vectors: $\boldsymbol{u}_t$ must be a function, commonly called policy, of the current state, i.e., $\boldsymbol{u}_t = \boldsymbol{\mu}_t(\boldsymbol{x}_t)$.

Many optimization problems coming from different fields such as economics, engineering, environment management, logistics, artificial intelligence can be formalized within this framework. Dynamic Programming (DP) is the standard mathematical tool to solve this kind of problems [1,2]. In general, the DP equations have to be solved numerically, leading to approximations of the value functions and, possibly, optimal policies. This popular procedure usually takes the name of Approximate Dynamic Programming (ADP) and is based, in the continuous-state case, on two main ingredients: (i) a class of models to be used in order to approximate the value functions and (ii) a proper sampling of the state space for the numerical solution of the DP equations (for an introduction see, e.g., [3] and the references therein).

Concerning (i), many different architectures have been proposed in the literature, among which we can cite polynomial approximators [4], splines [5], multivariate adaptive regression splines [6] and neural networks [7,8]. In this paper we focus on a choice based on local approximation, specifically on Nadaraya–Watson (NW) models (see, e.g., [9,10]). Local approximators based on kernel functions are routinely employed in the ADP framework [3,11] and in the reinforcement learning context [12], closely related to the ADP one. The main advantage, with respect to other popular models, lies on the small computational effort required to obtain the value function approximations, basically corresponding to the minimization of a one-dimensional cost. Concerning the other fundamental ingredient of ADP mentioned above at point (ii), i.e., sampling, the classic choice of a regular grid of points

* Corresponding author. Tel.: +39 0106475654; fax: +39 0106475600.
E-mail addresses: cervellera@ge.issia.cnr.it (C. Cervellera), mauro.gaggero@ge.issia.cnr.it (M. Gaggero), ddmach@ge.issia.cnr.it (D. Macciò).

0305-0548/$ - see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.cor.2013.09.006

coming from a uniform sampling of all the state components is not feasible in high-dimensional contexts, due to the well-known curse of dimensionality phenomenon, i.e., the exponential growth of the number of grid points as the state dimension increases. Then, efficient alternatives have been proposed in the literature, like sampling techniques commonly used in statistics for design of experiments, such as orthogonal arrays [13] and Latin hypercubes [14]. Another kind of sequences that have been successfully employed in the general context of ADP are low-discrepancy sequences (see, e.g., [15–18]) commonly used for numerical integration and number-theoretic algorithms [19]. Here, the use of such sequences is investigated in the context of ADP in combination with NW models. The main motivation behind the use of low-discrepancy sampling is that the resulting sets of points provide an efficient uniform covering of the state space as the number of points grows. More formally, it is proved in the paper that the choice of low-discrepancy sampling endows the NW structure with the ability to approximate the value functions arbitrarily well, which is a fundamental assumption needed to guarantee the convergence of the whole ADP procedure.

From a practical point of view, another motivation to employ low-discrepancy sampling with NW models in the ADP context is that the very regular structure and uniformity of the former can be exploited to derive a bandwidth selection method that can be used as an alternative to the cross validation procedure, which is the standard way of optimizing the kernel widths. In fact, rules of thumb available in the literature for the choice of the bandwidth of NW models employed in density estimation problems in the one-dimensional case, such as Silverman's rule or plugin rules [20], cannot be applied in a straightforward way to the multidimensional approximation case considered here. Thus, in this paper we present a method, directly based on the notion of discrepancy, that allows one to find reasonable values for the bandwidth parameter without recurring to any optimization procedure. Having a quick bandwidth selection method leads to a saving in the overall computational requirements of the entire ADP procedure, which are then basically reduced, at each stage, only to the minimization efforts needed to compute the value function estimates.

To sum up, the main contributions of the paper are:

- the introduction and analysis of the use of low-discrepancy sampling in the context of continuous-state ADP when local models are used for value function estimation, as an efficient alternative to random sampling in high-dimensional settings;
- the definition of a new method for automatic selection of the kernel bandwidths for value function approximation, exploiting the regularity of the low-discrepancy sampling.

Then, the paper provides the ADP practitioner with tools to improve the accuracy and computational burden of the procedure when local models are employed. The proposed approaches have been tested in a 15-dimensional inventory forecasting problem, comparing them with random sampling and the standard cross validation for bandwidth selection.

The paper is organized as follows. In Section 2, a review of the ADP algorithm is reported. Section 3 describes the considered NW models and low-discrepancy sequences. The use of NW models and low-discrepancy sequences within the ADP algorithm is discussed in Section 4. Numerical results are reported in Section 5, and conclusions are given in Section 6.

2. Review of the approximate dynamic programming scheme

The general scheme of the ADP procedure is briefly recalled in this section, together with some theoretical results on its convergence.

The DP method relies on the backward computation of a value function that represents at each stage the optimal cost to be paid from that stage on. This leads to the recursive solution of the following well-known Bellman equations:

$$J_T^\circ(\boldsymbol{x}_T) = h_T(\boldsymbol{x}_T),$$
$$J_t^\circ(\boldsymbol{x}_t) = \min_{\boldsymbol{u}_t \in U_t} \mathrm{E}_{\boldsymbol{\theta}_t}[h(\boldsymbol{x}_t, \boldsymbol{u}_t, \boldsymbol{\theta}_t) + J_{t+1}^\circ(\boldsymbol{f}(\boldsymbol{x}_t, \boldsymbol{u}_t, \boldsymbol{\theta}_t))], \quad t = T-1, \ldots, 0, \quad (1)$$

where the value function is represented by $J_t^\circ(\boldsymbol{x}_t)$ and $\mathrm{E}(\cdot)$ is the expectation operator. It is possible to prove (see, e.g., [2]) that $J_0^\circ(\boldsymbol{x}_0)$ corresponds to the optimal cost of the original MDP problem.

Since the value function is generally not available in the analytical form, in ADP an approximation, denoted by $\hat{J}_t$, has to be chosen inside a suitable class of models Γ at each stage t. To this purpose, a set Σ_t^L of L sampling points is chosen in X_t:

$$\Sigma_t^L = \{\boldsymbol{x}_{t,j} \in X_t : j = 1, \ldots, L\}, \quad t = 1, \ldots, T-1.$$

Then, using the approximate value function $\hat{J}_{t+1}(x_{t+1})$ defined at the previous stage, the basic ADP equation is written as

$$\tilde{J}_t^\circ(\boldsymbol{x}_{t,j}) = \min_{\boldsymbol{u}_t \in U_t} \frac{1}{S} \sum_{s=1}^{S} [h(\boldsymbol{x}_{t,j}, \boldsymbol{u}_t, \boldsymbol{\theta}_{t,s}) + \hat{J}_{t+1}(\boldsymbol{f}(\boldsymbol{x}_{t,j}, \boldsymbol{u}_t, \boldsymbol{\theta}_{t,s}))] \quad (2)$$

for each $\boldsymbol{x}_{t,j} \in \Sigma_t^L$. In Eq. (2), $\tilde{J}_t^\circ$ represents the estimated value of J_t° obtained by using $\hat{J}_{t+1}$, while the expected value with respect to $\boldsymbol{\theta}_t$ is estimated through an empirical mean computed over S realizations $\{\boldsymbol{\theta}_{t,1}, \ldots, \boldsymbol{\theta}_{t,S}\}$ of the random variables drawn from their probability distribution.

Once the L values $\tilde{J}_t^\circ(\boldsymbol{x}_{t,j})$, $j = 1, \ldots, L$, have been computed, the approximate value function $\hat{J}_t$ is obtained by exploiting the available observations. In general, we define for each stage t a class of models $\Gamma = \{\psi(\cdot, \boldsymbol{\alpha}_t) : \boldsymbol{\alpha}_t \in \Lambda \subset \mathbb{R}^k\}$ and set $\hat{J}_t(\boldsymbol{x}_t) = \psi(\boldsymbol{x}_t, \boldsymbol{\alpha}_t^*)$, where $\boldsymbol{\alpha}_t^*$ is obtained by minimizing an error criterion. A popular choice is the Mean Squared Error (MSE), which leads to the following optimization problem:

$$\boldsymbol{\alpha}_t^* = \arg\min_{\boldsymbol{\alpha} \in \Lambda} \frac{1}{L} \sum_{j=1}^{L} [\tilde{J}_t^\circ(\boldsymbol{x}_{t,j}) - \psi(\boldsymbol{x}_{t,j}, \boldsymbol{\alpha})]^2. \quad (3)$$

After this step, the value function can be estimated at each point of X_t and employed at stage $t-1$ for the computation of $\tilde{J}_{t-1}^\circ$.

Notice that the above described procedure to obtain the value function approximations is typically performed off line. Then, the approximations can be employed on line to compute the optimal decision vector in a given state, forward from $t=0$ to $t = T-1$. Specifically, the optimal vectors $\tilde{\boldsymbol{u}}_0^\circ, \ldots, \tilde{\boldsymbol{u}}_{T-1}^\circ$ are computed on line, starting from the initial state $\tilde{\boldsymbol{x}}_0$, using at each stage t the ADP equations in the following way:

$$\tilde{\boldsymbol{u}}_t^\circ = \arg\min_{\boldsymbol{u}_t \in U_t} \frac{1}{S} \sum_{s=1}^{S} [h(\tilde{\boldsymbol{x}}_t, \boldsymbol{u}_t, \boldsymbol{\theta}_{t,s}) + \hat{J}_{t+1}(\boldsymbol{f}(\tilde{\boldsymbol{x}}_t, \boldsymbol{u}_t, \boldsymbol{\theta}_{t,s}))], \quad t = 0, \ldots, T-1,$$

where $\tilde{\boldsymbol{x}}_t = \boldsymbol{f}(\tilde{\boldsymbol{x}}_{t-1}, \tilde{\boldsymbol{u}}_{t-1}^\circ, \tilde{\boldsymbol{\theta}}_{t-1})$, being $\tilde{\boldsymbol{\theta}}_{t-1}$ the realization of the random vector that affects the system in the on line phase. Eventually, the resulting states and decision vectors lead to the total cost actually paid in the on line phase from $\tilde{\boldsymbol{x}}_0$ to $\tilde{\boldsymbol{x}}_T$.

A key role for the convergence of the ADP algorithm is played by the approximation capabilities of the models in Γ. It is known that the convergence of the ADP solution, in terms of convergence of the total cost to the optimal one, is directly affected by the accuracy of the approximation of $\hat{J}_t$ in the various state points at stage t. To illustrate this, for a generic function $g : Z \to \mathbb{R}$, denote the infinite norm of g by $\|g\|_\infty = \sup_{\boldsymbol{z} \in Z} |g(\boldsymbol{z})|$. Then, we introduce the following assumption, which formalizes the universal approximation capability of the functions in the class Γ where we look for the approximation of the value functions.

Assumption 1. At each stage t, the value function approximation $\hat{J}_t$ is such that $\|\hat{J}_t - \tilde{J}_t^\circ\|_\infty \leq \varepsilon$ for every $\varepsilon > 0$.