Computers & Operations Research 40 (2013) 1076–1084

Contents lists available at SciVerse ScienceDirect

Computers & Operations Research

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

Adaptive value function approximation for continuous-state stochastic dynamic programming

Huiyuan Fan [a,*], Prashant K. Tarun [b], Victoria C.P. Chen [c]

[a] Rolls-Royce Energy Systems Inc, Mount Vernon, OH 43050, USA
[b] Steven L. Craig School of Business, Missouri Western State University, St. Joseph, MO 64507, USA
[c] Industrial and Manufacturing Systems Engineering, University of Texas at Arlington, Arlington, TX 76019, USA

ARTICLE INFO

Available online 23 November 2012

Keywords:
Approximate dynamic programming
Sequential design of experiments
Statistical modeling
Neural network
Number theoretic methods
Inventory forecasting

ABSTRACT

Approximate dynamic programming (ADP) commonly employs value function approximation to numerically solve complex dynamic programming problems. A statistical perspective of value function approximation employs a design and analysis of computer experiments (DACE) approach, where the "computer experiment" yields points on the value function curve. The DACE approach has been used to numerically solve high-dimensional, continuous-state stochastic dynamic programming, and performs two tasks primarily: (1) design of experiments and (2) statistical modeling. The use of design of experiments enables more efficient discretization. However, identifying the appropriate sample size is not straightforward. Furthermore, identifying the appropriate model structure is a well-known problem in the field of statistics. In this paper, we present a sequential method that can adaptively determine both sample size and model structure. Number-theoretic methods (NTM) are used to sequentially grow the experimental design because of their ability to fill the design space. Feed-forward neural networks (NNs) are used for statistical modeling because of their adjustability in structure-complexity . This adaptive value function approximation (AVFA) method must be automated to enable efficient implementation within ADP. An AVFA algorithm is introduced, that increments the size of the state space training data in each sequential step, and for each sample size a successive model search process is performed to find an optimal NN model. The new algorithm is tested on a nine-dimensional inventory forecasting problem.

1. Introduction

Dynamic programming (DP) was introduced by Bellman [3] as a mathematical programming method for solving multistage decision-making problems. Despite the advantages DP offers, it can only solve small problems under strict restrictions such as linear dynamics, quadratic cost, Gaussian random variable, etc. There are two principal obstacles to the use of DP. One, it assumes a fully observed system. In other words, the solution methodology DP employs is based on the complete knowledge of the state space. Unfortunately, the accurate and comprehensive process knowledge of a complex nonlinear control system in the real-world is seldom known *a priori*. Two, DP's solution methodology becomes progressively limited due to "curse of dimensionality" implying that the implementation of DP can be prohibitive in terms of cost, time, or both with an exponential increase in the computational effort due to an increase in the number of state and decision variables across the stages. These issues become insurmountable for the classical DP methods when the decision problems exhibit high-dimensionality and include state and decision variables that are both continuous and stochastic. To address these issues, there is a need to develop solution methodologies that can adaptively find approximate solutions to high-dimensional, continuous-state, stochastic problems. In the wake of recent advances in computational power, numerous methods have been developed by researchers to numerically solve DP problems, which led to a new class of dynamic programming methods called "approximate dynamic programming (ADP)."

ADP tackles the issue of continuous state space through discretization and approximates the future value (or *cost-to-go*) function using statistical modeling techniques. One of the earliest strategies for discretization was to form a finite grid of discrete points in the state space and use multilinear or spline interpolation subsequently (see [24]). When the number of state variables is small, the grid size can be small. If this is the case, then the corresponding DP problem can be solved comprehensively. On the other hand, the DP problem corresponding a high-dimensional,

* Corresponding author. Tel.: +1 740 393 8519; fax: +1 740 393 8336.
E-mail addresses: huiyuan.fan@rolls-royce.com, huiyuanfan@yahoo.com (H. Fan).

0305-0548/$ - see front matter © 2012 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.cor.2012.11.016

continuous-state, stochastic decision scenario would be practically intractable due to the issue of "curse of dimensionality". In other words, high-dimensional, continuous-state, stochastic problems would lead to an exponential increase in the number of discrete points in the grid with an increase in the number of state variables thereby making the DP problem practically infeasible. To circumvent the practical limitations of DP in solving complex multistage problems, research efforts in the area of continuous-state DP have been mainly focused on developing solution methods that address the "curse of dimensionality" issue effectively. There have been significant improvements over the first set of uniform grid discretization methods introduced by Bellman [3]. Specifically, Foufoula-Georgiou and Kitanidis [18] proposed an algorithm, referred to as gradient DP, which employed cubic Hermite polynomials to approximate the future value function. Johnson et al. [24] compared numerical solution methods using multilinear, Hermite gradient DP, and tensor-product cubic spline interpolation for a four-reservoir problem. The study by Johnson et al. [24] showed that the cubic splines required fewer grid levels in each dimension, which led to a reduction in the computational time. Unfortunately, these solution methods were based on a full grid of discrete points (i.e., a full factorial experimental design), which grows exponentially with an increase in the number of dimensions.

In the past decade, there have been significant contributions in developing practical solution methodologies for complex ADP problems. For instance, Chen [9] and Chen et al. [10] developed methodologies to generate practical numerical solutions for high-dimensional ADPs. These methodologies incorporated orthogonal array (OA) based experimental designs and multivariate adaptive regression splines (MARS), where OAs were special subsets of full factorial experimental designs that grew polynomially (NOT exponentially) with the number of dimensions. These approaches helped reduce the computational efforts required to solve high-dimensional problems. Furthermore, the approaches described in Chen [9] and Chen et al. [10] produced better solutions than the methods proposed in the study by Johnson et al. [24] that used a full factorial design with tensor-product cubic spline interpolation. More recently, the study by Cervellera et al. [6] used Latin hypercube based experimental designs and neural network approximation. The results obtained were comparable to the results obtained using OAs and MARS for a nine-dimensional inventory forecasting problem and an eight-dimensional water reservoir problem. Additionally, Cervellera et al. [6] and Wen [40] studied experimental designs based on number-theoretic methods (NTM) and successfully solved a 30-dimensional water reservoir problem. Baglietto et al. [1] recently presented two methods for approximately solving T-stage stochastic optimal control (SOC) problems based on the use of "one-hidden-layer (OHL) networks" consisting of linear combinations of simple basis functions containing the parameters to be optimized, and revealed the efficacy of these methods for high-dimensional water resource systems. Cervellera and Maccio [8] and Cervellera et al. [7] also proposed a semi-local approximation method and applied it to the inventory forecasting and water reservoir problems mentioned previously, demonstrating that the semi-local approximation results in smaller computational times with respect to the neural network approximation. The methods proposed in this paper stem from the success of the aforementioned methodologies.

Researchers in the machine learning area of the artificial intelligence have also shown strong interests in solving ADP. In 1980s, they began exploring the possibility of applying the theories in psychology and animal learning to solving such a problem [28,43,27]. Their research contributions led to an entire new class of ADP methods with "machine-learning perspective". Contrary to the "statistical perspective" described previously, the methods with machine-learning perspective are primarily based on the concept of "reinforcement learning (RL)". RL was initially inspired by the studies on animal learning in the field of experimental psychology. In a general RL model [39,25], an agent interacts with its environment through sensors (perception) and actors (actions). Each interaction typically goes as follows: the agent receives inputs representative of a state in the environment; the agent then performs an action at that state; the agent then receives a reward indicating the value of the state-action transition. The amount of reward acts as a "reinforcement" to the agent in its interaction with the environment. The agent's objective is to learn a policy that maximizes the expected cumulative reward. The RL family of methods also includes "active critic", "neuro-dynamic programming", etc., which, in essence, share the same methodological philosophy [42]. The research community in machine learning had lofty expectations from RL-based methods in solving approximate dynamic programming (ADP) problems as large and complex as the ones a normal mammal brain could learn to handle [43]. Today's reality, however, is far from those high expectations [44].

There have been two predominant ways of solving ADP problems: classical methods with statistical perspective (based on backward solution algorithms); and newer methods with machine learning perspective (based on reinforcement learning (RL) concept). Evidently, classes of methods based on either statistical or machine learning perspective have their pros and cons in solving real-world ADP problems. The classical methods are limited in its application and work effectively only for dynamic programming problems with finite horizons. Other disadvantages of classical methods include its inability to be implemented online, its dependence on traditional optimization algorithms usually under the strict assumption of convexity, and its ineffectiveness in solving non-convex problems. In contrast, RL-based methods use forward solution algorithms to seek a time invariant steady-state solution and, hence, are appropriate for infinite-horizon problems. They are also deemed good at dealing with problems with extremely high complexity, and can be implemented online by trial-and-error . A critical drawback of RL methods, however, is their costly online learning (or training).

This paper focuses on developing a unique methodology for solving ADP problems that integrates some of the advantages of the RL-based methods into the methods with statistical perspective. It is well-known that the key to solving an ADP problem is the future value (or cost-to-go) function approximation. However, the vast majority of current approaches for approximating the future value function fall short due to their underlying assumption that the approximating model is fixed in structure. In practice, the approximating model structure must be tuned carefully, usually by trial-and-error, to get a good solution. To address this issue, our paper proposes a sequential concept to help determine the approximating model structure.

To implement the sequential concept, a modeling technique is first required, within which the structure-complexity of a model can be incrementally adjustable. Feed-forward neural networks (NNs) are selected because of their adjustability in structure-complexity . In addition, NNs are with a capability to handle nonlinear decision-making situations. Contrary to the existing statistical perspective that uses batch sampling strategy and sets up the sample size of the batch in advance, the sequential concept uses low-discrepancy sequential sampling techniques, such as NTM, to sample the state space. The sequential sampling techniques are promising because of their ability to fill the space in a sequential manner. As a result, the sequential concept incorporates adaptability in both identification of the approximating model structure and determination of the sample size. Furthermore, the influence of