Reinforcement Learning and Optimal Control

by

Dimitri P. Bertsekas

Massachusetts Institute of Technology

DRAFT TEXTBOOK

This is a draft of a textbook that is scheduled to be finalized in 2019, and to be published by Athena Scientific. It represents "work in progress," and it will be periodically updated. It more than likely contains errors (hopefully not serious ones). Furthermore, its references to the literature are incomplete. Your comments and suggestions to the author at dimitrib@mit.edu are welcome. The date of last revision is given below.

December 14, 2018

WWW site for book information and orders

http://www.athenasc.com

Athena Scientific, Belmont, Massachusetts

Athena Scientific
Post Office Box 805
Nashua, NH 03060
U.S.A.

Email: info@athenasc.com
WWW: http://www.athenasc.com

Publisher's Cataloging-in-Publication Data

Bertsekas, Dimitri P.
Reinforcement Learning and Optimal Control
Includes Bibliography and Index
1. Mathematical Optimization. 2. Dynamic Programming. I. Title.
QA402.5 .B465 2019 519.703 00-91281

ISBN-10: 1-886529-39-6, ISBN-13: 978-1-886529-39-7

ABOUT THE AUTHOR

Dimitri Bertsekas studied Mechanical and Electrical Engineering at the National Technical University of Athens, Greece, and obtained his Ph.D. in system science from the Massachusetts Institute of Technology. He has held faculty positions with the Engineering-Economic Systems Department, Stanford University, and the Electrical Engineering Department of the University of Illinois, Urbana. Since 1979 he has been teaching at the Electrical Engineering and Computer Science Department of the Massachusetts Institute of Technology (M.I.T.), where he is currently the McAfee Professor of Engineering.

His teaching and research spans several fields, including deterministic optimization, dynamic programming and stochastic control, large-scale and distributed computation, and data communication networks. He has authored or coauthored numerous research papers and seventeen books, several of which are currently used as textbooks in MIT classes, including "Dynamic Programming and Optimal Control," "Data Networks," "Introduction to Probability," and "Nonlinear Programming."

Professor Bertsekas was awarded the INFORMS 1997 Prize for Research Excellence in the Interface Between Operations Research and Computer Science for his book "Neuro-Dynamic Programming" (co-authored with John Tsitsiklis), the 2001 AACC John R. Ragazzini Education Award, the 2009 INFORMS Expository Writing Award, the 2014 AACC Richard Bellman Heritage Award, the 2014 Khachiyan Prize for Life-Time Accomplishments in Optimization, the 2015 George B. Dantzig Prize, and the 2018 John von Neumann Theory Prize. In 2001, he was elected to the United States National Academy of Engineering for "pioneering contributions to fundamental research, practice and education of optimization/control theory, and especially its application to data communication networks."

ATHENA SCIENTIFIC

OPTIMIZATION AND COMPUTATION SERIES

1. Abstract Dynamic Programming, 2nd Edition, by Dimitri P. Bertsekas, 2018, ISBN 978-1-886529-46-5, 360 pages
2. Dynamic Programming and Optimal Control, Two-Volume Set, by Dimitri P. Bertsekas, 2017, ISBN 1-886529-08-6, 1270 pages
3. Nonlinear Programming, 3rd Edition, by Dimitri P. Bertsekas, 2016, ISBN 1-886529-05-1, 880 pages
4. Convex Optimization Algorithms, by Dimitri P. Bertsekas, 2015, ISBN 978-1-886529-28-1, 576 pages
5. Convex Optimization Theory, by Dimitri P. Bertsekas, 2009, ISBN 978-1-886529-31-1, 256 pages
6. Introduction to Probability, 2nd Edition, by Dimitri P. Bertsekas and John N. Tsitsiklis, 2008, ISBN 978-1-886529-23-6, 544 pages
7. Convex Analysis and Optimization, by Dimitri P. Bertsekas, Angelia Nedić, and Asuman E. Ozdaglar, 2003, ISBN 1-886529-45-0, 560 pages
8. Network Optimization: Continuous and Discrete Models, by Dimitri P. Bertsekas, 1998, ISBN 1-886529-02-7, 608 pages
9. Network Flows and Monotropic Optimization, by R. Tyrrell Rockafellar, 1998, ISBN 1-886529-06-X, 634 pages
10. Introduction to Linear Optimization, by Dimitris Bertsimas and John N. Tsitsiklis, 1997, ISBN 1-886529-19-1, 608 pages
11. Parallel and Distributed Computation: Numerical Methods, by Dimitri P. Bertsekas and John N. Tsitsiklis, 1997, ISBN 1-886529-01-9, 718 pages
12. Neuro-Dynamic Programming, by Dimitri P. Bertsekas and John N. Tsitsiklis, 1996, ISBN 1-886529-10-8, 512 pages
13. Constrained Optimization and Lagrange Multiplier Methods, by Dimitri P. Bertsekas, 1996, ISBN 1-886529-04-3, 410 pages
14. Stochastic Optimal Control: The Discrete-Time Case, by Dimitri P. Bertsekas and Steven E. Shreve, 1996, ISBN 1-886529-03-5, 330 pages

Contents

Preface

In this book we consider large and challenging multistage decision problems, which can be solved in principle by dynamic programming (DP for short), but their exact solution is computationally intractable. We discuss solution methods that rely on approximations to produce suboptimal policies with adequate performance. These methods are collectively referred to as *reinforcement learning*, and also by alternative names such as *approximate dynamic programming*, and *neuro-dynamic programming*.

Our subject has benefited greatly from the interplay of ideas from optimal control and from artificial intelligence. One of the aims of the book is to explore the common boundary between these two fields and to form a bridge that is accessible by workers with background in either field.

Our primary focus will be on *approximation in value space*. Here, the control at each state is obtained by limited lookahead with cost function approximation, i.e., by optimization of the cost over a limited horizon, plus an approximation of the optimal future cost, starting from the end of this horizon. The latter cost, which we generally denote by $\tilde{J}$, is a function of the state where we may be at the end of the horizon. It may be computed by a variety of methods, possibly involving simulation and/or some given or separately derived heuristic/suboptimal policy. The use of simulation often allows for model-free implementations that do not require the availability of a mathematical model, a major idea that has allowed the use of dynamic programming beyond its classical boundaries.

We focus selectively on four types of methods for obtaining $\tilde{J}$:

(a) *Problem approximation*: Here $\tilde{J}$ is the optimal cost function of a related simpler problem, which is solved by exact DP. Certainty equivalent control and enforced decomposition schemes are discussed in some detail.

(b) *Rollout and model predictive control*: Here $\tilde{J}$ is the cost function of some known heuristic policy. The needed cost values to implement a rollout policy are often calculated by simulation. While this method applies to stochastic problems, the reliance on simulation favors deterministic problems, including challenging combinatorial problems for which heuristics may be readily implemented. Rollout may also be combined with adaptive simulation and Monte Carlo tree search, which have proved very effective in the context of games such as backgammon, chess, Go, and others.

Model predictive control was originally developed for continuous-space optimal control problems that involve some goal state, e.g., the origin in a classical control context. It can be viewed as a specialized rollout method that is based on an optimization algorithm for reaching a goal state.

(c) *Parametric cost approximation*: Here $\tilde{J}$ is chosen from within a parametric class of functions, including neural networks, with the parameters "optimized" or "trained" by using state-cost sample pairs and some type of incremental least squares/regression algorithm. Approximate policy iteration and its variants are covered in some detail, including several actor-critic schemes. These include policy evaluation that involves temporal difference-based training methods, and policy improvement that is based on approximation in policy space.

(d) *Aggregation*: Here the cost function $\tilde{J}$ is the optimal cost function of some approximation to the original problem, called aggregate problem, which has fewer states. The aggregate problem can be formulated in a variety of ways, and may be solved by using exact DP techniques. Its optimal cost function is then used as $\tilde{J}$ in a limited lookahead scheme. Aggregation may also be used to provide local improvements to parametric approximation schemes that involve neural networks or linear feature-based architectures.

We have adopted a gradual expository approach, which proceeds along three directions:

(1) *From exact DP to approximate DP*: We first discuss exact DP algorithms, explain why they may be difficult to implement, and then use them as the basis for approximations.

(2) *From finite horizon to infinite horizon problems*: We first discuss finite horizon exact and approximate DP methodologies, which are intuitive and mathematically simple in Chapters 1-3. We then progress to infinite horizon problems in Chapters 4 and 5.

(3) *From model-based to model-free approaches*: Reinforcement learning methods offer a major potential benefit over classical DP approaches, which were practiced exclusively up to the early 90s: they can be implemented by using a simulator/computer model rather than a mathematical model. In our presentation, we first discuss model-based methods, and then we identify those methods that can be appropriately modified to work with a simulator.

After the first chapter, each new class of methods is introduced as a more sophisticated or generalized version of a simpler method introduced earlier. Moreover, each type of method is illustrated by means of examples, which should be helpful in providing insight into its use, but may also be skipped selectively and without loss of continuity. Detailed solutions to some of the simpler examples are given, and may illustrate some of the implementation details.

The mathematical style of this book is somewhat different from the one of the author's dynamic programming books [Ber12], [Ber17a], [Ber18a], and the neuro-dynamic programming research monograph, written jointly with John Tsitsiklis [BeT96]. While we rigorously present the theory of finite and infinite horizon dynamic programming, and some fundamental approximation methods, we rely more on intuitive explanations and less on proof-based insights. Moreover, our mathematical requirements are modest: calculus, elementary probability, and a minimal use of matrix-vector algebra.

Furthermore, we present methods that are often successful in practice, but have less than solid performance properties. This is a reflection of the state of the art in the field: there are no methods that are guaranteed to work for all or even most problems, but there are enough methods to try on a given problem with a reasonable chance of success in the end. For this process to work, however, it is important to have proper intuition into the inner workings of each type of method, as well as an understanding of its analytical and computational properties. To quote a statement from the preface of the neuro-dynamic programming (NDP) monograph [BeT96]: "It is primarily through an understanding of the mathematical structure of the NDP methodology that we will be able to identify promising or solid algorithms from the bewildering array of speculative proposals and claims that can be found in the literature."

Dimitri P. Bertsekas

Winter 2018