Rollout, Approximate Policy Iteration, and Distributed Reinforcement Learning

by

Dimitri P. Bertsekas

CSE 691 Course Notes
Spring 2020
Arizona State University

These notes represent “work in progress,” and will be periodically updated. It more than likely contains errors (hopefully not serious ones). Furthermore, the references to the literature are incomplete. Your comments and suggestions to the author at dbertsek@asu.edu are welcome.

February, 2020
Contents

1. Exact Dynamic Programming
   1.1. Deterministic Dynamic Programming . . . . . . . . . . . p. 2
      1.1.1. Basic Finite Horizon Problem Formulation . . . . . . p. 2
      1.1.2. The Dynamic Programming Algorithm . . . . . . . . . p. 5
      1.1.3. Approximation in Value Space . . . . . . . . . . . . p. 7
   1.2. Stochastic Dynamic Programming . . . . . . . . . . . . . p. 9
      1.2.1. Finite Horizon Problems . . . . . . . . . . . . . . . p. 10
      1.2.2. Infinite Horizon Problems - An Overview . . . . . . p. 13
   1.3. Examples, Variations, and Simplifications . . . . . . . . . p. 19
      1.3.1. Discrete Deterministic Optimization . . . . . . . . . p. 21
      1.3.2. Problems with a Termination State . . . . . . . . . p. 24
      1.3.3. Partial State Information and Belief States . . . . . . p. 28
   1.4. Reinforcement Learning and Optimal Control - Some . . . . .
      Terminology . . . . . . . . . . . . . . . . . . . . . . . . . . p. 30
   1.5. Notes and Sources . . . . . . . . . . . . . . . . . . . . . . . p. 32

2. Rollout and Policy Improvement
   2.1. Approximation in Value and Policy Space . . . . . . . . . p. 37
      2.1.1. Approximation in Value Space - One-Step and . . . . .
            Multistep Lookahead . . . . . . . . . . . . . . . . . . . p. 37
      2.1.2. Approximation in Policy Space . . . . . . . . . . . . . p. 41
      2.1.3. Combined Approximation in Value and Policy . . . . . .
            Space . . . . . . . . . . . . . . . . . . . . . . . . . . . . p. 43
   2.2. General Issues of Approximation in Value Space . . . . . . . . . p. 47
      2.2.1. Model-Based and Model-Free Implementations . . . . . . p. 47
      2.2.2. Off-Line and On-Line Implementations . . . . . . . . . p. 48
      2.2.3. Methods for Cost-to-Go Approximation . . . . . . . . . p. 49
      2.2.4. Methods for Simplification of the Lookahead . . . . . . .
            Minimization . . . . . . . . . . . . . . . . . . . . . . . p. 50
      2.2.5. Simplification of the Lookahead Minimization . . . . . .
            by Q-Factor Approximation . . . . . . . . . . . . . . . . p. 52
2.3. Rollout and the Policy Improvement Principle ........................................ p. 54
  2.3.1. On-Line Rollout for Deterministic Discrete ......................................
  2.3.2. The Fortified Rollout Algorithm ...................................................... p. 66
  2.3.3. Truncated Rollout Algorithms .......................................................... p. 69
2.4. Stochastic Rollout and Monte Carlo Tree Search ....................................... p. 71
  2.4.1. Simulation-Based Implementation of the Rollout Algorithm ..................... p. 74
  2.4.2. Variance Reduction in Rollout - Comparing Advantages ......................... p. 76
  2.4.3. Monte Carlo Tree Search .................................................................... p. 79
  2.4.4. Randomized Policy Improvement by Monte Carlo Tree Search ................. p. 82
  2.4.5. Rollout Paralellization ....................................................................... p. 83
2.5. On-Line Rollout for Deterministic Infinite-Spaces Problems ........................ p. 84
  2.5.1. Model Predictive Control ..................................................................... p. 85
  2.5.2. Target Tubes and the Constrained Controllability Condition ................. p. 92
  2.5.3. Variants of Model Predictive Control .................................................. p. 96
2.6. Multiagent Rollout .................................................................................... p. 98
  2.6.1. Multiagent Paraledlization .................................................................... p. 105
2.7. Constrained Rollout for Deterministic Optimization Heuristics .................. p. 108
  2.7.1. Constrained Multiagent Rollout ........................................................... p. 119
2.8. Discrete and Combinatorial Optimization Applications ............................... p. 122
  2.8.1. Multidimensional Assignment ............................................................. p. 127
2.9. Rollout for Minimax Control ...................................................................... p. 135
2.10. Notes and Sources .................................................................................. p. 137

3. Learning Values and Policies
  3.1. Approximation Architectures .................................................................... p. 144
    3.1.1. Feature-Based Architectures .............................................................. p. 145
    3.1.2. Training of Linear and Nonlinear Architectures ................................. p. 155
3.2. Neural Networks ....................................................................................... p. 159
    3.2.1. Training of Neural Networks .............................................................. p. 163
    3.2.2. Multilayer and Deep Neural Networks .............................................. p. 164
3.3. Training of Cost Functions in Approximate DP ......................................... p. 166
3.4. Training of Policies in Approximate DP .................................................... p. 171
    3.4.1. Training of Truncated Rollout Policies - Multiprocessor Paralellization
           and Perpetual Rollout ........................................................................ p. 175
3.5. Notes and Sources .................................................................................. p. 176

4. Approximate Policy Iteration for Infinite Horizon Problems
  4.1. Stochastic Shortest Path and Discounted Problems .................................... p. 185
Preface

In this research monograph we will discuss the solution of large and challenging multistage decision problems using methods of reinforcement learning, also referred to by other names such as *approximate dynamic programming*, and *neuro-dynamic programming*. We will focus on a subset of methods which are based on the idea of policy improvement and policy iteration, i.e., starting from some policy and generating one or more improved policies. If only one improved policy is generated, this is called *rollout*. If multiple successively improved policies are generated, we obtain forms of approximate policy iteration, which we will also refer to as *perpetual rollout*. This is one of the most prominent type of reinforcement learning methods, and is central in the implementation of recent high profile successes, such as the AlphaGo and AlphaZero programs. They can be implemented using data generated by the system itself, a process known as *self-training*, and value and policy approximation architectures, including neural networks.

Fundamentally, our methods draw their validity from the algorithmic theory of dynamic programming, but they also rely on more modern approximation methods that originated in large part in learning ideas from artificial intelligence, such as the simulation-based training of compact approximation architectures, and the use of neural networks. Consequently we selectively summarize background or related material, some of which is covered in greater depth in the author’s RL book [Ber19a] (see also the slides and videolectures [Ber19d]).

On the other hand, we will aim to develop rollout and approximate policy iteration methods beyond the book [Ber19a]. In particular, we will present new research, relating to systems involving multiple agents, partitioned architectures, and distributed asynchronous computation. We will also apply our methods to the solution of challenging combinatorial/discrete optimization problems and partially observed Markov Decision Problems (POMDP).

Dimitri P. Bertsekas
ASU, February 2020