Algorithms for Reinforcement Learning

Algorithms for Reinforcement Learning

Csaba Szepesvári

ISBN: 978-3-031-00423-0 paperback
ISBN: 978-3-031-01551-9 ebook

DOI 10.1007/978-3-031-01551-9

A Publication in the Springer series
SYNTHESIS LECTURES ON ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING

Lecture #9
Series Editors: Ronald J. Brachman, *Yahoo! Research*
Thomas Dietterich, *Oregon State University*

Series ISSN
Synthesis Lectures on Artificial Intelligence and Machine Learning
Print 1939-4608 Electronic 1939-4616

Synthesis Lectures on Artificial Intelligence and Machine Learning

A Concise Introduction to Multiagent Systems and Distributed Artificial Intelligence
Nikos Vlassis
2007

Intelligent Autonomous Robotics: A Robot Soccer Case Study
Peter Stone
2007

Algorithms for Reinforcement Learning

Csaba Szepesvári
University of Alberta

SYNTHESIS LECTURES ON ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING #9

ABSTRACT

Reinforcement learning is a learning paradigm concerned with learning to control a system so as to maximize a numerical performance measure that expresses a long-term objective. What distinguishes reinforcement learning from supervised learning is that only partial feedback is given to the learner about the learner's predictions. Further, the predictions may have long term effects through influencing the future state of the controlled system. Thus, time plays a special role. The goal in reinforcement learning is to develop efficient learning algorithms, as well as to understand the algorithms' merits and limitations. Reinforcement learning is of great interest because of the large number of practical applications that it can be used to address, ranging from problems in artificial intelligence to operations research or control engineering. In this book, we focus on those algorithms of reinforcement learning that build on the powerful theory of dynamic programming. We give a fairly comprehensive catalog of learning problems, describe the core ideas, note a large number of state of the art algorithms, followed by the discussion of their theoretical properties and limitations.

KEYWORDS

reinforcement learning, Markov Decision Processes, temporal difference learning, stochastic approximation, two-timescale stochastic approximation, Monte-Carlo methods, simulation optimization, function approximation, stochastic gradient methods, least-squares methods, overfitting, bias-variance tradeoff, online learning, active learning, planning, simulation, PAC-learning, Q-learning, actor-critic methods, policy gradient, natural gradient

Contents

Preface

Reinforcement learning (RL) refers to both a learning problem and a subfield of machine learning. As a learning problem, it refers to learning to control a system so as to maximize some numerical value which represents a long-term objective. A typical setting where reinforcement learning operates is shown in Figure 1: A controller receives the controlled system's state and a reward associated with the last state transition. It then calculates an action which is sent back to the system. In response, the system makes a transition to a new state and the cycle is repeated. The problem is to learn a way of controlling the system so as to maximize the total reward. The learning problems differ in the details of how the data is collected and how performance is measured.

In this book, we assume that the system that we wish to control is stochastic. Further, we assume that the measurements available on the system's state are detailed enough so that the controller can avoid reasoning about how to collect information about the state. Problems with these characteristics are best described in the framework of Markovian Decision Processes (MDPs). The standard approach to 'solve' MDPs is to use dynamic programming, which transforms the problem of finding a good controller into the problem of finding a good value function. However, apart from the simplest cases when the MDP has very few states and actions, dynamic programming is infeasible. The RL algorithms that we discuss here can be thought of as a way of turning the infeasible dynamic programming methods into practical algorithms so that they can be applied to large-scale problems.

There are two key ideas that allow RL algorithms to achieve this goal. The first idea is to use samples to compactly represent the dynamics of the control problem. This is important for

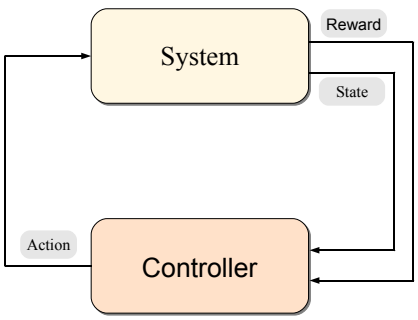

Figure 1: The basic reinforcement learning scenario

two reasons: First, it allows one to deal with learning scenarios when the dynamics is unknown. Second, even if the dynamics is available, exact reasoning that uses it might be intractable on its own. The second key idea behind RL algorithms is to use powerful function approximation methods to compactly represent value functions. The significance of this is that it allows dealing with large, high-dimensional state- and action-spaces. What is more, the two ideas fit nicely together: Samples may be focused on a small subset of the spaces they belong to, which clever function approximation techniques might exploit. It is the understanding of the interplay between dynamic programming, samples and function approximation that is at the heart of designing, analyzing and applying RL algorithms.

The purpose of this book is to allow the reader to have a chance to peek into this beautiful field. However, certainly we are not the first to set out to accomplish this goal. In 1996, Kaelbling et al. have written a nice, compact survey about the approaches and algorithms available at the time (Kaelbling et al., 1996). This was followed by the publication of the book by Bertsekas and Tsitsiklis (1996), which detailed the theoretical foundations. A few years later Sutton and Barto, the 'fathers' of RL, published their book, where they presented their ideas on RL in a very clear and accessible manner (Sutton and Barto, 1998). A more recent and comprehensive overview of the tools and techniques of dynamic programming/optimal control criteria, as well as various classes of controlled systems is given in the two-volume book by Bertsekas (2007a,b) which devotes one chapter to RL methods.[1] At times, when a field is rapidly developing, books can get out of date pretty quickly. In fact, to keep up with the growing body of new results, Bertsekas maintains an online version of his Chapter 6 of Volume II of his book, which, at the time of writing this survey counted as much as 160 pages (Bertsekas, 2010). Other recent books on the subject include the book of Gosavi (2003) who devotes 60 pages to reinforcement learning algorithms in Chapter 9, concentrating on average cost problems, or that of Cao (2007) who focuses on policy gradient methods. Powell (2007) presents the algorithms and ideas from an operations research perspective and emphasizes methods that are capable of handling large control spaces, Chang et al. (2008) focuses on adaptive sampling (i.e., simulation-based performance optimization), while the center of the recent book by Busoniu et al. (2010) is function approximation.

Thus, by no means do RL researchers lack a good body of literature. However, what seems to be missing is a self-contained and yet relatively short summary that can help newcomers to the field to develop a good sense of the state of the art, as well as existing researchers to broaden their overview of the field, an article, similar to that of Kaelbling et al. (1996), but with an updated contents. To fill this gap is the very purpose of this short book.

Having the goal of keeping the text short, we had to make a few, hopefully, not too troubling compromises. The first compromise we made was to present results only for the total expected discounted reward criterion. This choice is motivated by that this is the criterion that is both widely used and the easiest to deal with mathematically. The next compromise is that the background

[1] In this book, RL is called neuro-dynamic programming or approximate dynamic programming. The term neuro-dynamic programming stems from the fact that, in many cases, RL algorithms are used with artificial neural networks.

on MDPs and dynamic programming is kept ultra-compact (although an appendix is added that explains these basic results. Apart from these, the book aims to cover a bit of all aspects of RL, up to the level that the reader should be able to understand the whats and hows, as well as to implement the algorithms presented. Naturally, we still had to be selective in what we present. Here, the decision was to focus on the basic algorithms, ideas, as well as the available theory. Special attention was paid to describing the choices of the user, as well as the trade offs that come with these. We tried to be impartial as much as possible, but some personal bias, as usual, surely remained. The pseudocode of almost twenty algorithms was included, hoping that this will make it easier for the practically inclined reader to implement the algorithms described.

The target audience is advanced undergraduate and graduate students, as well as researchers and practitioners who want to get a good overview of the state of the art in RL quickly. Researchers who are already working on RL might also enjoy reading about parts of the RL literature that they are not so familiar with, thus broadening their perspective on RL. The reader is assumed to be familiar with the basics of linear algebra, calculus, and probability theory. In particular, we assume that the reader is familiar with the concepts of random variables, conditional expectations, and Markov chains. It is helpful, but not necessary, for the reader to be familiar with statistical learning theory, as the essential concepts will be explained as needed. In some parts of the book, knowledge of regression techniques of machine learning will be useful.

This book has three parts. In the first part, in Section 1, we provide the necessary background. It is here where the notation is introduced, followed by a short overview of the theory of Markov Decision Processes and the description of the basic dynamic programming algorithms. Readers familiar with MDPs and dynamic programming should skim through this part to familiarize themselves with the notation used. Readers, who are less familiar with MDPs, must spend enough time here before moving on because the rest of the book builds heavily on the results and ideas presented here.

The remaining two parts are devoted to the two basic RL problems (cf. Figure 2), one part devoted to each. In Section 2) the problem of learning to predict values associated with states is studied. We start by explaining the basic ideas for the so-called tabular case when the MDP is small enough so that one can store one value per state in an array allocated in a computer's main memory. The first algorithm explained is TD(λ), which can be viewed as the learning analogue to value iteration from dynamic programming. After this, we consider the more challenging situation when there are more states than what fits into a computer's memory. Clearly, in this case, one must compress the table representing the values. Abstractly, this can be done by relying on an appropriate function approximation method. First, we describe how TD(λ) can be used in this situation. This is followed by the description of some new gradient based methods (GTD2 and TDC), which can be viewed as improved versions of TD(λ) in that they avoid some of the convergence difficulties that TD(λ) faces. We then discuss least-squares methods (in particular, LSTD(λ) and λ-LSPE) and compare them to the incremental methods described earlier. Finally, we describe choices available for implementing function approximation and the trade offs that these choices come with.

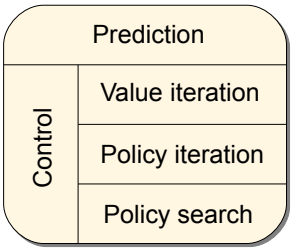

Figure 2: Types of reinforcement problems and approaches.

The second part (Section 3) is devoted to algorithms that are developed for control learning. First, we describe methods whose goal is optimizing online performance. In particular, we describe the "optimism in the face of uncertainty" principle and methods that explore their environment based on this principle. State of the art algorithms are given both for bandit problems and MDPs. The message here is that clever exploration methods make a large difference, but more work is needed to scale up the available methods to large problems. The rest of this section is devoted to methods that aim at developing methods that can be used in large-scale applications. As learning in large-scale MDPs is significantly more difficult than learning when the MDP is small, the goal of learning is relaxed to learning a good enough policy in the limit. First, direct methods are discussed which aim at estimating the optimal action-values directly. These can be viewed as the learning analogue of value iteration of dynamic programming. This is followed by the description of actor-critic methods, which can be thought of as the counterpart of the policy iteration algorithm of dynamic programming. Both methods based on direct policy improvement and policy gradient (i.e., which use parametric policy classes) are presented.

The book is concluded in Section 4, which lists some topics for further exploration.

Csaba Szepesvári
June 2010

Acknowledgments

I am truly indebted to my family for their love, support and patience. Thank you Mom, Beáta, Dávid, Réka, Eszter, Csongor! Special thanks to Réka who has helped me drawing Figure 1.1. A number of individuals have read various versions of the manuscript, full or in parts and helped me to reduce the number of mistakes by sending corrections. They include Dimitri Bertsekas, Gábor Balázs, Bernardo Avila Pires, Warren Powell, Rich Sutton, Nikos Vlassis, Hengshuai Yao and Shimon Whiteson. Thank You! Of course, all the remaining mistakes are mine. If I have left out someone from the above list, this was by no means intentional. If this is the case, please remind me in an e-mail (better yet, send me some comments or suggestions). Independently of whether they have contacted me before or not, readers are encouraged to e-mail me if they find errors, typos or they just think that some topic should have been included (or left out). I plan to periodically update the text and I will try to accommodate all the requests. Finally, I wish to thank Remi Munos and Rich Sutton, my closest collaborators over the last few years, from whom I have learned and continue to learn a lot. I also wish to thank all my students, the members of RLAI group and all researchers of RL who continue to strive to push the boundaries of what we can do with reinforcement learning. This book is made possible by you.

Csaba Szepesvári
June 2010