Available online at www.sciencedirect.com

ScienceDirect

Computers & Operations Research 35 (2008) 1999–2017

computers & operations research

www.elsevier.com/locate/cor

Application of reinforcement learning to the game of Othello

Nees Jan van Eck*, Michiel van Wezel

Erasmus School of Economics, Erasmus University Rotterdam, P.O. Box 1738, 3000 DR Rotterdam, The Netherlands

Available online 5 December 2006

Abstract

Operations research and management science are often confronted with sequential decision making problems with large state spaces. Standard methods that are used for solving such complex problems are associated with some difficulties. As we discuss in this article, these methods are plagued by the so-called curse of dimensionality and the curse of modelling. In this article, we discuss reinforcement learning, a machine learning technique for solving sequential decision making problems with large state spaces. We describe how reinforcement learning can be combined with a function approximation method to avoid both the curse of dimensionality and the curse of modelling. To illustrate the usefulness of this approach, we apply it to a problem with a huge state space—learning to play the game of Othello. We describe experiments in which reinforcement learning agents learn to play the game of Othello without the use of any knowledge provided by human experts. It turns out that the reinforcement learning agents learn to play the game of Othello better than players that use basic strategies.

Keywords: Dynamic programming; Markov decision processes; Reinforcement learning; *Q*-learning; Multiagent learning; Neural networks; Game playing; Othello

1. Introduction

Many decision making problems that we face in real life are sequential in nature. In these problems, the payoff does not depend on an isolated decision but rather on a sequence of decisions. In order to maximize the total payoff, the decision maker may have to sacrifice immediate payoffs such that greater payoffs can be received later on. Finding a policy for making good sequential decisions is an interesting problem. Ideally, such a policy should indicate what the best decision is in each possible situation (or state) the decision maker may encounter.

A well-known class of sequential decision making problems are the Markov decision processes (MDPs), described in detail in Section 2. Their most important property is that the optimal decision in a given state is independent of earlier states the decision maker encountered. MDPs have found widespread application in operations research and management science. For a review see, e.g. [1].

For MDPs there exist a number of algorithms that are guaranteed to find optimal policies. These algorithms are collectively known as dynamic programming methods. A problem with dynamic programming methods is that they are unable to deal with problems in which the number of possible states is high (also called the curse of dimensionality). Another problem is that dynamic programming requires exact knowledge of the problem characteristics (also called the curse of modelling), as will be explained in Section 2.

* Corresponding author.
E-mail addresses: nvaneck@few.eur.nl (N.J. van Eck), mvanwezel@few.eur.nl (M. van Wezel).

0305-0548/$ - see front matter © 2006 Elsevier Ltd. All rights reserved.
doi:10.1016/j.cor.2006.10.004

A relatively new class of algorithms, known as reinforcement learning algorithms (see, e.g. [2–5]), may help to overcome some of the problems associated with dynamic programming methods. Multiple scientific fields have made contributions to reinforcement learning—machine learning, operations research, control theory, psychology, and neuroscience. Reinforcement learning has been applied in a number of areas, which has produced some successful practical applications. These applications range from robotics and control to industrial manufacturing and combinatorial search problems such as computer game playing (see, e.g. [3]). One of the most convincing applications is TD-Gammon, a system that learns to play the game of Backgammon by playing against itself and learning from the results [6–8]. TD-Gammon reaches a level of play that is almost as good as the best human players.

Recently, there has been some interest in the application of reinforcement learning algorithms to problems in the fields of operations research and management science. For example, an interesting article is [9], where reinforcement learning is applied to airline yield management and the aim is to find an optimal policy for the denial/acceptance of booking requests for seats in various fare classes. Another example is [10,11], where reinforcement learning is used to find an optimal control policy for a group of elevators. In the above examples, the authors report that reinforcement learning methods outperform frequently used standard algorithms. A marketing application is described in [12], where a target selection decision in direct marketing is seen as a sequential decision problem. Other examples from the management science literature are [13,14], which are more methodologically oriented.

The purpose of this article is to introduce the reader to reinforcement learning and to convince the reader of the usefulness of this method in helping to avoid both the curse of dimensionality and the curse of modelling. To achieve this, we will perform some experiments in which reinforcement learning is applied to a sequential decision making problem with a huge state space—the game of Othello (approximately 10^{28} states). In the experiments, reinforcement learning agents learn to play the game of Othello without the use of any knowledge provided by human experts.

Games like Othello are well-defined sequential decision making problems where performance can be measured easily. The problems they present are challenging and require complicated problem solving. Games have therefore proven to be a worthwhile domain for studying and developing various types of problem solving techniques. For an extensive overview of previously published work on techniques for improving game playing programs by learning from experience see, e.g. [15]. Here, we mention two articles in particular because they are in some way related to the research in this article. The first one is the work by Moriarty and Miikkulainen [16], who study the artificial evolution of game playing neural networks by using genetic algorithms. Some of the networks they use learn the game of Othello without any expert knowledge. The second one is the work by Chong et al. [17], who study neural networks that use an evolutionary algorithm to learn to play the game of Othello without preprogrammed human expertise.

The remainder of this article is organized as follows. In Section 2, we give an introduction to reinforcement learning and sequential decision making problems. We describe a frequently used reinforcement learning algorithm, *Q*-learning, in detail. In Section 3, we explain the game of Othello. In Section 4, we discuss the Othello playing agents that we use in our experiments. The experiments themselves are described in Section 5. Finally, in Section 6, we give a summary, some conclusions, and an outlook.

2. Reinforcement learning and sequential decision making problems

In this section, we give a brief introduction to reinforcement learning and sequential decision making problems. The reader is referred to [2–5] for a more extensive discussion of these topics.

We describe reinforcement learning from the intelligent agent perspective [18]. An intelligent agent is an autonomous entity (usually a computer program) that repeatedly senses inputs from its environment, processes these inputs, and takes actions in its environment. Many learning problems can conveniently be described using the agent perspective without altering the problem in an essential way.

In reinforcement learning, the agent/environment setting is as follows. At each moment, the environment is in a certain state. The agent observes this state, and depending solely on the state, the agent takes an action. The environment responds with a successor state and a reinforcement (also called a reward). Fig. 1 shows a schematic representation of this sense-act cycle.

The agent's task is to learn to take optimal actions, i.e., actions that maximize the sum of immediate rewards and (discounted) future rewards. This may involve sacrificing immediate rewards to obtain a greater cumulative reward in the long term or just to obtain more information about the environment.