Hidden-Mode Markov Decision Processes for Nonstationary Sequential Decision Making

Samuel P.M. Choi, Dit-Yan Yeung, and Nevin L. Zhang

Department of Computer Science,
Hong Kong University of Science and Technology
Clear Water Bay, Kowloon, Hong Kong
{pmchoi,dyyeung,lzhang}@cs.ust.hk

1 Introduction

Problem formulation is often an important first step for solving a problem effectively. In sequential decision problems, Markov decision process (MDP) (Bellman [2]; Puterman [22]) is a model formulation that has been commonly used, due to its generality, flexibility, and applicability to a wide range of problems. Despite these advantages, there are three necessary conditions that must be satisfied before the MDP model can be applied; that is,

1. The environment model is given in advance (a completely-known environment).
2. The environment states are completely observable (fully-observable states, implying a Markovian environment).
3. The environment parameters do not change over time (a stationary environment).

These prerequisites, however, limit the usefulness of MDPs. In the past, research efforts have been made towards relaxing the first two conditions, leading to different classes of problems as illustrated in Figure 1.

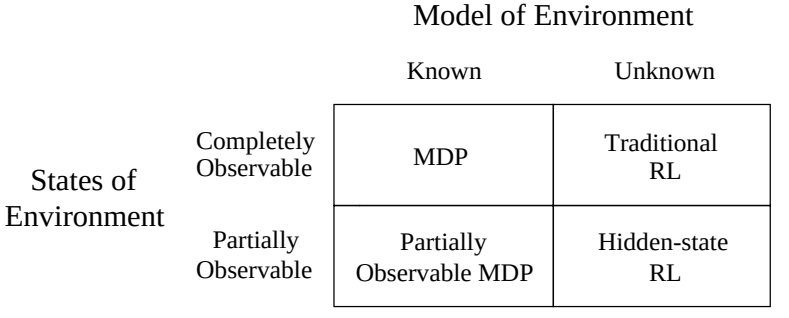

Fig. 1. Categorization into four related problems with different conditions. Note that the degree of difficulty increases from left to right and from upper to lower.

R. Sun and C.L. Giles (Eds.): Sequence Learning, LNAI 1828, pp. 264–287, 2000.
© Springer-Verlag Berlin Heidelberg 2000

This paper mainly addresses the first and third conditions, whereas the second condition is only briefly discussed. In particular, we are interested in a special type of nonstationary environments that repeat their dynamics in a certain manner. We propose a formal model for such environments. We also develop algorithms for learning the model parameters and for computing optimal policies.

Before we proceed, let us briefly review the four categories of problems shown in Figure 1 and define the terminology that will be used in this paper.

1.1 Four Problem Types

Markov Decision Process

MDP is the central framework for all the problems we discuss in this section. An MDP formulates the interaction between an agent and its environment. The environment consists of a state space, an action space, a probabilistic state transition function, and a probabilistic reward function. The goal of the agent is to find, according to its optimality criterion, a mapping from states to actions (i.e. policy) that maximizes the long-term accumulated rewards. This policy is called an *optimal policy*. In the past, several methods for solving Markov decision problems have been developed, such as value iteration and policy iteration (Bellman 1).

Reinforcement Learning

Reinforcement learning (RL) (Kaelbling *et al.* 12; Sutton and Barto 28) is originally concerned with learning to perform a sequential decision task based only on scalar feedbacks, without any knowledge about what the correct actions should be. Around a decade ago researchers realized that RL problems could naturally be formulated into incompletely known MDPs. This realization is important because it enables one to apply existing MDP algorithms to RL problems. This has led to research on *model-based* RL. The model-based RL approach first reconstructs the environment model by collecting experience from its interaction with the world, and then applies conventional MDP methods to find a solution. On the contrary, *model-free* RL learns an optimal policy directly from the experience. It is this second approach that accounts for the major difference between RL and MDP algorithms. Since less information is available, RL problems are in general more difficult than the MDP ones.

Partially Observable Markov Decision Process

The assumption of having fully-observable states is sometimes impractical in the real world. Inaccurate sensory devices, for example, could make this condition difficult to hold true. This concern leads to studies on extending MDP to partially-observable MDP (POMDP) (Monahan 20; Lovejoy 17; White III 29). A POMDP basically introduces two additional components to the original