Labeling Q-Learning for Maze Problems with Partially Observable States

Hae Yeon Lee*, Hiroyuki Kamaya**, Kenich Abe*

*Dept. Electrical and Communication Engineering, Graduate School of Engineering, Tohoku Univ.
Aoba05, 980-8579, Sendai, Japan
(Tel : +81-22-217-7075 ; Fax : +81-22-263-9290 ;
E-mail : yeon@abe.ecei.tohoku.ac.jp, abe@abe.ecei.tohoku.ac.jp)
**Dept. Electrical Engineering, Hachinohe National College of Technology
Tamonoki, 139-1192, Hachinohe, Japan
(Tel : +81-178-27-7283; Fax : +81-178-27-9379 ; Tel ; E-mail : kamaya-e@hachinohe-c.ac.jp)

Abstract

Recently, *Reinforcement Learning*(RL) methods have been used for learning problems in *Partially Observable Markov Decision Process*(POMDP) environments. Conventional RL-methods, however, have limited applicability to POMDP. To overcome the partial observability, several algorithms were proposed [5], [7].

The aim of this paper is to extend our previous algorithm for POMDP, called *Labeling Q-learning*(LQ-learning), which reinforces incomplete information of perception with labeling. Namely, in the LQ-learning, the agent percepts the current states by pair of observation and its label, and the agent can distinguish states, which look as same, more exactly. Labeling is carried out by a hash-like function, which we call *Labeling Function*(LF). Numerous labeling functions can be considered, but in this paper, we will introduce several labeling functions based on only 2 or 3 immediate past sequential observations.

We introduce the basic idea of LQ-learning briefly, apply it to maze problems, simple POMDP environments, and show its availability with empirical results, look better than conventional RL algorithms.

1. Introduction

Sequential decision problems in which an agent's observations provide it with the complete state of its environment can be formulated as *Markov Decision Processes*(MDP), for which a number of successful planning and RL methods have been developed. However, in many areas, e.g., mobile robotics, multi-agent or distributed control environments, etc., the agent's perception at best gives it partial information about the state of environment. Such agent-environment interactions suffer from hidden state or perceptual aliasing and can be formulated as POMDP. Therefore, finding efficient RL methods to solve interesting sub-classes of POMDP is of great practical interest to AI and engineering.

Recent researches on POMDP have concentrated on overcoming partial observability by using memory to estimate state and on developing planning and learning methods that work well with the agent's knowledge of state. In part, this emphasis on state estimation has come about because it has been widely observed and noted that the presence of partial obseravability renders popular and successful RL methods for MDP, such as *Q-learning* and *Sarsa*, useless on POMDP. Next, we introduce the basic algorithm of RL below.

Reinforcement Learning

In the widely used RL approaches, such as Q-learning by Watkins[16], TD(λ) by Sutton[14], and Sarsa(λ) by Dayan[3], the learning agent uses experience to learn estimation of optimal Q-value functions that map state-action pairs, (s,a) by receiving a scalar reinforcement signal, called reward, as a feedback performance from its environment, and changes its parameters so as to act optimally in the environment. In another words, agent learns the optimal policy that maximizes expected discounted rewards.

The environment is defined by a finite set of state S, and the agent has a finite set of actions A. In the RL process, the agent time-discretely observes its environment's state $s \in S$, and at each time step, determines it's action $a \in A$ based on the observation and receive a scalar reward r. Then, the object of learning is to construct an optimal action policy that maximizes the expected discounted reward $E[\Sigma_{t=0}^{\infty} \gamma^t r_t]$ where r_t is the reward at time step t and $0 \le \gamma \le 1$ is discount factor that makes immediate reward more available than reward more distant in time.

The action selection at each step is based on Q-values, $Q(s,a)$, about the relative goodness of actions. The Q-value, $Q(s,a)$, is the total discounted reward that the agent receives if it starts at a state s, performs an action a, and behaves optimally thereafter. In the RL approach, the Q-values are estimated based on the *Temporal Difference*(TD) method, see Sutton[13].

A standard form of the algorithms of estimating the Q-values with heuristic traces is expressed as follows;

$$Q(s,a) \leftarrow Q(s,a) + [r_t + \gamma Q(s_{t+1}, a_t) - Q(s_t, a_t)] e(s,a) \qquad (1)$$

where α is learning rate, and γ is discount factor, and $e(s,a)$ is eligibility traces(Loch and Singh, [8]).

$$e(s_t, a_t) \leftarrow \gamma \lambda e(s_t, a_t) + 1$$
$$e(s,a) \leftarrow \gamma \lambda e(s,a) \quad \text{for } s \neq s_t, \text{ or } a \neq a_t \qquad (2)$$

This eligibility traces emphasize more recent and frequent events more strongly. But we apply a modification form of the eligibility traces, called the *replacing traces*, see Singh and