An Advance Q Learning (AQL) Approach for Path Planning and Obstacle Avoidance of a Mobile Robot

Arpita Chakraborty, Bengal Institute of Technology, Kolkata, West Bengal, India

Jyoti Sekhar Banerjee, Bengal Institute of Technology, Kolkata, West Bengal, India

ABSTRACT

The goal of this paper is to improve the performance of the well known Q learning algorithm, the robust technique of Machine learning to facilitate path planning in an environment. Until this time the Q learning algorithms like Classical Q learning(CQL)algorithm and Improved Q learning (IQL) algorithm deal with an environment without obstacles, while in a real environment an agent has to face obstacles very frequently. Hence this paper considers an environment with number of obstacles and has coined a new parameter, called 'immediate penalty' due to collision with an obstacle. Further the proposed technique has replaced the scalar 'immediate reward' function by 'effective immediate reward' function which consists of two fuzzy parameters named as, 'immediate reward' and 'immediate penalty'. The fuzzification of these two important parameters not only improves the learning technique, it also strikes a balance between exploration and exploitation, the most challenging problem of Reinforcement Learning. The proposed algorithm stores the Q value for the best possible action at a state; as well it saves significant path planning time by suggesting the best action to adopt at each state to move to the next state. Eventually, the agent becomes more intelligent as it can smartly plan a collision free path avoiding obstacles from distance. The validation of the algorithm is studied through computer simulation in a maze like environment and also on KheperaII platform in real time. An analysis reveals that the Q Table, obtained by the proposed Advanced Q learning (AQL) algorithm, when used for path-planning application of mobile robots outperforms the classical and improved Q-learning.

Keywords: *Effective Immediate Reward, Immediate Penalty, Immediate Reward, Learning Time, Performance Index, Risk factor, Similar Q Value Problem*

INTRODUCTION

Reinforcement learning (RL) is one of the most rapidly developing machine learning methods in recent years (Jeni et al.,2007; Park et al.,2007). It includes the temporal difference algorithm proposed by Sutton and the Q-learning of Watkins (Watkins et al., 1992). Those algorithms have been extensively used in many applications such as industrial control, time sequence prediction, robot soccer competition, and many more. However, finding

DOI: 10.4018/ijimr.2013010105

the proper balance between exploration and exploitation in Q-learning is one of the major issues requiring further attention. Exploitation occurs if the action selection strategy is based purely on current values of the state-action pairs, i.e., when the selection is greedy. In the case of most of the optimization problems, this will lead to locally optimal policies, possibly differing from a globally optimal one. In contrast, exploration is the strategy based on the assumption that the agent selects a non optimal action in the current situation and obtains more knowledge about the problem. This knowledge allows it to neglect the locally optimal policies, and to reach the globally optimal one instead. On the other hand, excessive exploration will drastically decrease the performance of a learning algorithm, and in some cases might be even harmful with respect to the learning results themselves. Improved Q-learning (IQL) was proposed to overcome the space and time complexities of Classical Q-learning (CQL) and undoubtedly it has improved the performance. In spite of that, in case of IQL, the ability of learning the real environment amidst of static obstacles is not satisfactory (Gerke et al., 1997). Here the proposed learning algorithm, named Advanced Q learning (AQL) algorithm claims to be more efficient in learning endowing the agents with more intelligence during path planning (Xiao et al., 1997; Bien et al., 1992). RL has been simulated in different environments (Moll et al., 2004; Regele et al., 2006; Martin et al,2007) and in this paper the validation of the proposed algorithm is studied through computer simulation in a maze like environment and also on KheperaII platform in real time.

PRELIMINARIES OF Q LEARNING

Q learning is basically a model free Reinforcement Learning (Busoniu et al., 2010; Masoumzadeh et al., 2009), where a set of states S, a set of actions A, and a reward function R(S, A) are there. In each state $s \in S$, the agent (Hsu et al., 2008; Zhou et al., 2007) takes an action $a \in A$. Upon taking the action, the agent receives a reward R(s, a) and reaches to a new state $s^{/}$. Q learning (Cho et al., 2007; Pandey et al., 2010), which has been developed in several stages (Chen et al., 2009), are explained briefly in the following section.

Classical Q–Learning (CQL)

In classical Q-learning, every possible state of an agent and its possible actions in a given state are deterministically known. In other words, for a given agent A, let s_0, s_1, s_2... s_n, be n- possible states, and each state has m possible actions $a_0, a_1, a_2, ..., a_m$. At a particular state-action pair (s_i, a_j) the specific reward that the agent achieves is known as immediate reward $r(s_i, a_j)$ (shown in Figure 1). The agent selects its next state from its current state using a policy that attempts to maximize the cumulative reward that the agent could have in subsequent transition of states from its next state (Dean et al., 1993; Bellman, 1957; Watkins et al., 1992). For example, let the agent be in state s_i and is expecting to select the next best state. Then the Q-value at state S_i due to action of a_j is given in (1).

$$Q(s_i, a_j) = r(s_i, a_j) + \gamma \underset{a'}{Max} \quad Q(\delta(s_i, a_j), a') \tag{1}$$

where $\delta(s_i, a_j)$ denotes the next state due to selection of action a_j at state s_i. Let the next state selected be S_k. So, $Q(\delta(S_i, a_j), a') = Q(S_k, a')$. Consequently selection of a' that maximizing $Q(s_i, a_j)$ is an interesting problem. One main drawback for the above Q-learning is to know the Q value at a state s_k for all possible action a'. As a result, each time it accesses the memory to get Q value for all possible actions at a particular state to determine the most appropriate next state. So it consumes more time to select the next

On the Development of an Ants-Inspired Navigational Network for Autonomous Robots