plain
definition
remark

Universität Hamburg
Department Informatik
Knowledge Technology, WTM

Interactive Reinforcement Learning: Learning with Advice

Seminar Paper Outline

Bio-Inspired Artificial Intelligence

Peter Wüppen, Alvin Rindra Fazrie
Matr.Nr. 605308, 6641834
5wueppen@informatik.uni-hamburg.de
4fazrie@informatik.uni-hamburg.de

11.11.2015

Abstract

Interaction between Artificial Intelligence and Human has been growing to be a vital part in daily life in the past decade. Reinforcement Learning has become one of the fundamental topic for scientists in the field of robotic and machine learning. In this paper, we....

Contents

1 Introduction

Current technology recently have been built with Artificial Intelligence that humans could interact with. Interaction between humans and AI is getting popular in part of daily life e.g., drone, autonomous car, and even video games since AI player could make the games becoming more impressive and challenging. In this regards, AI player should implement a learning approach which could make them to learn acting unpredicted scenarios through another AI and humans.

Interactive Reinforcement learning has proven to be great potential in adapting and learning multiple tasks especially. The idea behind RL is to respond according to a set of desired actions and to continuously observe the current and potential environment. With a reinforcement learning approach, the agent explores the environment around it with an aim to maximize the rewards that it could gain by interacting with the environment. The core of the reinforcement learning concept is based on Markov decision process(MDP).

In this paper, an overview of basic elements of reinforcement learning will be provided in the beginning. In the following chapter, there will be different interactive approaches, and applying the approaches system introduced in cruz paper will be provided in chapter.

2 Reinforcement Learning Basics

In these decades, several fields have already implemented Reinforcement learning method and it is proven to show great performance in machine learning paradigms. In this section we will provide a brief understanding of reinforcement learning, the basic setup, value function and how to find the optimal policies.

Reinforcement learning is a learning mechanism where an agent learns to observe the environment continuously and respond with a set of allowed actions within the environment. The agent once performs the desired action will receive a feedback from the environment. Each of the actions performed by the agent are selected based on the decision according to an internal decision making policy where the policy is simply a probability of an action a being taken for a given s.

The feedback given to the agent will be in the form of a scalar reward from the RL framework, which will be decided by a reward function. The reward could diverse based on what kind of agent behavior is expected and it will be utilized as a factor to update the action selection policy to gain an optimal policy. In other terms the agents ultimate goal is to achieve an optimal behavior by learning the optimal action for each state.

2.1 Basic Setup

Reinforcement Learning is arranged in certain steps continuously, the basic step reinforcement learning model can be seen as listed below:

1. Observe state, s_t

2. Decide on an action, a_t
3. Perform action
4. Get the reward
5. Observe new state, $st + 1$
6. Update policy based on the given reward
7. Repeat

The model has an aim to find a control policy which will maximize the observed rewards for the agent.

In RL an agent will face a certain state each time ($s_t \in S$) and in the current state the agent needs to select a possible action and then perform it ($a_t \in A(s_t)$). Once the action has been done by the agent, a positive or a negative value as a reward will be provided by the environment (r_t) and the following state (s_{t+1}) to the agent in the next step [4].

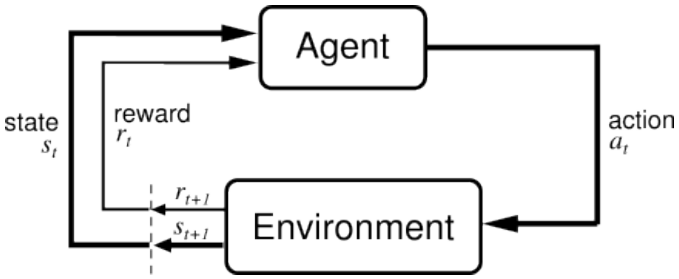

Figure 1: Reinforcement learning diagram showing an agent's interaction with its environment [4].

Policy maps a state to an action through the process. ($\pi_t(s_t, a_t)$); moreover, learning process is changing the policy in order to gain experience from the environment and the main goal of agent is maximizing the accumulated reward.[4]. The basic model of standard reinforcement learning is depicted in figure 1.

Markov Decision Process is the basis of core components for Reinforcement Learning, and several core components which define the Reinforcement Learning could be formalized as follows [1]:

- State space, denoted by $s \in S$, is a discrete set of environment states.
- Action space, denoted by $a \in A$, is a discrete set of actions from the environment's agent.
- Reward function, denoted by $r : S \times A \to \mathbb{R}$, is a function which turns each transition for a given state into a scalar value.
- State Transition function, denoted by $\delta : S \times A \to S$, is a function which gives the potential state s' when the action a is conducted.

- Policy, denoted by π, is a function which specifies the agents behavior. It maps the action to be taken for each given state. i.e., $\pi_t : S \rightarrow A$, where S is the environment state set and A is the agent action.

- State value function, denoted by $V^\pi : S \rightarrow \mathbb{R}$, is a function that will be used to obtain the highest reward as the agent will always try to learn. It specifies the value for each state and maps the state to the reward an agent can expect to accumulate.

2.2 Q-Value functions

In Reinforcement Learning, the agent needs to achieve its main goal which is to take an action in each step that could maximize the total reward. The total reward estimation which an agent could obtain by performing an action in the current state (action-value pairs) could be calculated with the equation below [4].

$$Q^\pi(s,a) = E_\pi\{\sum_{k=0}^{\infty} \gamma^k r_t + k + 1 | s_t = s, a_t = a\} \tag{1}$$

From the equation, it can be seen that $Q(s,a)$ is the state-action pairs value and r_t is the action reward $a = a_t$ under the policy π in the state $s = s_t$. Moreover, γ is discount rate of future actions influence ($0 \leq \gamma < 1$).

From that equation, the optimized estimation of state-action pairs value can be formularized in equation below which is called Bellman Equation [4]:

$$Q^*(s,a) = \sum_{s'} P(s'|s,a)[r(s,a,s') + \gamma \max_{a'} Q(s',a')] \tag{2}$$

In this equation, $Q(s,a)$ is the optimal function to do state-action pairs value estimation where p is the probability to achieve the subsequent state $s' = s_t + 1$ which is performed bey a as an action in the current state s and a' representing the possible actions set in the future state s' [4]. state action pair value.

2.3 Finding optimal policies

The easiest way to learn the optimal policy is to learn the optimal value function. There are several Reinforcement Learning algorithms which could maximize the finding optimal policies. In this section, Q Learning and Sarsa algorithm are provided.

Q-Learning is an Off-Policy algorithm for Temporal Difference learning. The algorithm gathers with probability from 1 to a close approximation of the action-value function for an arbitrary target policy. Q-Learning learns the optimal policy even when actions are performed according to a more exploratory or even random policy. The Q learning algorithm can be seen as follows [4] :

Algorithm 1 Q learning

1. Initialize $Q(s, a)$ arbitrary $(\forall s, a)$.
2. For $t = 0, 1, 2, \ldots$

   Initialize s

   Choose the action a_t for the current state s_t. (E.g., an ϵ greedy policy.)

   Take action a_t, observe $R(s_t)$, s_{t+1}

   $Q(s_t, a_t) \leftarrow (1 - \alpha_t)Q(s_t, a_t) + \alpha_t[R(s) + \gamma \max_a Q(s_{t+1}, a)]$

   $s \leftarrow s_{t+1}$;
3. until s is terminal

The Sarsa algorithm is an On-Policy algorithm for TD-Learning. Sarsa is almost identical to Q-learning and the major difference between Sarsa and Q-Learning, is that the maximum reward for the next state is not necessarily utilized for updating the Q-values. But the actiona and reward, is selected using the same policy which determined the original action. The name Sarsa actually derives from the updates which are done using the quintuple Q(s, a, r, s', a'). Where: s, a are the original state and action, r is the reward observed in the following state and s', a' are the new state-action pair. The Sarsa algorithm algorithm can be seen as follows [4]:

Algorithm 2 SARSA

1. Initialize $Q(s, a)$ arbitrary $(\forall s, a)$.
2. Choose the action a_t for the current state s_t. (E.g., an ϵ greedy policy.)
3. For $t = 0, 1, 2, \ldots$

   Initialize s

   Take action a_t, observe $R(s_t)$, s_{t+1}

   Choose the action a_{t+1} for the current state s_{t+1}. (E.g., an ϵ greedy policy.)

   $Q(s_t, a_t) \leftarrow (1 - \alpha_k)Q(s_t, a_t) + \alpha_k[R(s) + \gamma Q(s_{t+1}, a_{t+1})]$

   $s \leftarrow s_{t+1}; a \leftarrow a_{t+1}$;
4. until s is terminal

3 RL with Interactive Feedback

3.1 Approaches for Human Advising

[6] [7] [3]

3.2 Approaches for Agent Advising

[5]

4 Implementation

[2]

4.1 Description of the implemented algorithm

4.2 Parameter Optimizations for different scenarios

4.3 Discussion and Results

5 Conclusion

References

[1] T Baier-Lowenstein and Jianwei Zhang. Learning to grasp everyday objects using reinforcement-learning with automatic value cut-off. In *Intelligent Robots and Systems*, page 15511556. IEEE, 2007.

[2] Francisco Cruz, Johannes Twiefel, Sven Magg, Cornelius Weber, and Stefan Wermter. Interactive reinforcement learning through speech guidance in a domestic scenario. In *Neural Networks (IJCNN), 2015 International Joint Conference on*, pages 1–8. IEEE, 2015.

[3] W Bradley Knox and Peter Stone. Reinforcement learning from human reward: Discounting in episodic tasks. In *RO-MAN, 2012 IEEE*, pages 878–885. IEEE, 2012.

[4] Richard S. Sutton and Andrew G. Barto. *Introduction to Reinforcement Learning*. MIT Press, Cambridge, MA, USA, 1st edition, 1998.

[5] Matthew E. Taylor, Nicholas Carboni, Anestis Fachantidis, Ioannis Vlahavas, and Lisa Torrey. Reinforcement learning agents providing advice in complex video games. *Connect. Sci*, 26(1):45–63, January 2014.

[6] Andrea L. Thomaz and Cynthia Breazeal. Reinforcement learning with human teachers: Evidence of feedback and guidance with implications for learning performance. In *Proceedings of the 21st National Conference on Artificial Intelligence - Volume 1*, AAAI'06, pages 1000–1005. AAAI Press, 2006.

[7] Andrea Lockerd Thomaz, Guy Hoffman, and Cynthia Breazeal. Real-time interactive reinforcement learning for robots. In *AAAI 2005 workshop on human comprehensible machine learning*, 2005.