The 33rd Annual Conference of the Japanese Society for Artificial Intelligence, 2019

Flexibility of Emulation Learning from Pioneers in Nonstationary Environments

Moto Shinriki*1 Hiroaki Wakabayashi*1 Yu Kono*1 Tatsuji Takahashi*1

*1School of Science and Engineering, Tokyo Denki University

In imitation learning, the agent observes specific action-state pair sequences of another agent (expert) and somehow reflect them into its own action. One of its implementations in reinforcement learning is the inverse reinforcement learning. We propose a new framework for social learning, emulation learning, which requires much less information from another agent (pioneer). In emulation learning, the agent is given only a certain level of achievement (accumulated rewards per episode). In this study, we implement emulation learning in the reinforcement learning setting by applying a model of satisficing action policy. We show that the emulation learning algorithm works well in a non-stationary reinforcement learning tasks, breaking the often observed trade-off like relationship between optimality and flexibility.

1. Introduction

Humans usually begin learning with some kind of prior knowledge. If the knowledge is about the structure of the environment, the past experience, or other's action history, the learning will be somehow model-based, by transfer, or supervised (or inverse), respectively. Imitation learning requires an *expert* who provides an exemplar behavior. However, the expert's action data may be very expensive or unavailable in general. On the other hand, there are cases where (only) the information of someone's achievement level is obtained. Our search is often accelerated by a peer's high achievement or record breaking, as in sports or in invention. As it is closely related to *end state emulation* in social learning[1], we call this form of learning *emulation learning*. In emulation, the outcome of an action sequence ("what") is socially learned, while in imitation the process or the procedure ("how") is observed and assimilated.

In this study, from two aspect, we test the performance of the three algorithms: the vanilla Q-learning, imitation learning with inverse reinforcement learning, and our emulation learning with satisficing reinforcement learning. The first aspect is the speed of the learning algorithms. The second aspect is the flexibility of the algorithms. The learning task is a nonstationary reinforcement learning task, and we evaluate how flexibly the algorithm can respond to the environmental changes.

2. The Reinforcement Learning Algorithms

Reinforcement Learning is a type of machine learning in which an agent learns an appropriate action sequence through interaction, trial and error, in the environment. Recently, as researches on game AI and autonomous robot have been actively conducted, reinforcement learning is gaining more attention as a method for autonomous learning in unknown environments.

Contact: Tatsuji Takahashi, School of Science and Engineering, Tokyo Denki University, Ishizaka, Hatoyama, Hikigun, Saitama, Japan 350-0394, Tel: 049-296-5416, tatsujit@mail.dendai.ac.jp

2.1 Q-Learning

There is a representative method of reinforcement learning called Q-learning. The action value $Q(s_t, a_t)$ of Q learning is updated based on the estimation policy. When the estimation policy is set as the greedy policy, the action value $Q(s_t, a_t)$ is updated as:

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha\Big(r_t + \gamma \max_a Q(s_{t+1}, a) - Q(s_t, a_t)\Big), \quad (1)$$

where α is the learning rate and γ is the discount rate.

2.2 Inverse Reinfocement Learning as Imitation

Inverse reinforcement learning (IRL) is a form of learning of which the goal is to infer a reward function $R(s)$ from the expert's behavior trajectory T. A reward function $R(s)$ is usually a function which returns a reward calculated by multiplying the parameter θ and the one-hot-vector of the state. The behavior trajectory is a pair of expert's states and actions, $(s_0, a_0, s_1, a_1, ...)$. In this study, we used the maximum entropy IRL (MaxEntIRL) as the implementation[3].

2.3 Emulation by Risk-sensitive Satisficing (RS)

Humans tend not to exhaustively search for optimazation. Rather, we satisfice. That is, we confront a task with certain reference level (aspiration) and finish searching when we find an satisfactory actoin better than the reference [6]. When the aspiration is given socially, satisficing means emulation. Satisicing policy will converge the actions to take, when the aspiration is satisfied, after the limited search. We implement emulation with the RS model, a cognitive satisficing value function with reflective risk attitudes as in the prospect theory in behavioral economics [2]. The RS value funtion is defined as follows:

$$RS(s_t, a_t) = \tau(s_t, a_t)\Big(Q(s_t, a_t) - \aleph(s_t)\Big) \quad (2)$$

$$a_t^{\mathrm{sel}} = \arg\max_a RS(s_t, a) \quad (3)$$

RS's valuation qualitatively changes according to the sign of the difference between the aspiration and the Q value. It considers the reliablity of the Q value with τ that approximates how many times the action has been chosen. τ is defined as follows, where γ_τ is the discount rate and α_τ is the learning rate.

$$\tau(s_t, a_t) = \tau_{\text{curr}}(s_t, a_t) + \tau_{\text{post}}(s_t, a_t) \quad (4)$$

$$\tau_{\text{curr}}(s_t, a_t) \leftarrow \tau_{\text{curr}}(s_t, a_t) + 1 \quad (5)$$

$$\tau_{\text{post}}(s_t, a_t) \leftarrow (1 - \alpha_\tau)\tau_{\text{post}}(s_t, a_t) + \alpha_\tau \gamma_\tau \tau(s_{t+1}, a^{\text{sel}}_{t+1}) \quad (6)$$

2.3.1 Global reference conversion (GRC)

While RS works as intended in the multi-armed bandit problems that may be considered as a single state reinforcement learning task, it is generally difficult to assign the optimal aspiration to each state, when the global aspiration (for the entire episode) is avaiable from a pioneer. The global reference conversion (GRC) allocates optimal reference values to each state by defining the global observed expectation E_G and the global satisficing reference value $\aleph_G$ defined below. It is this $\aleph_G$ that works as a social goal-setting trigger, such as someone's high performance or record breaking. E_G is defined using the temporary expectation (E_{tmp}) which is periodically reset.

$$E_G \leftarrow \frac{E_{\text{tmp}} + \gamma_G N_G E_G}{1 + \gamma_G N_G} \quad (7)$$

$$N_G \leftarrow 1 + \gamma_G N_G \quad (8)$$

$$\delta_G = \min(E_G - \aleph_G, 0) \quad (9)$$

$$\aleph(s_i) = \max_a Q(s_i, a) - \zeta(s_i)\delta_G \quad (10)$$

The parameter $\zeta(s_i)$ is introduced to adjust the scale of the global reference value and the Q value.

3. Task: UnsteadySwitchWorld

To test the speed and flexibility of the learning algorithms, we conducted an experiment, based a task called SwitchWorld introduced in the previous study by some of the authors [4]. In this experiment, we used a *UnsteadySwitchWorld* task in which the switches change their places periodically. The state space of the task is shown in Fig. 1. The red cells are where a switch is placed, and the green cell is where the agent is placed at the initial step of an episode. The agent can move to one of the upper, lower, left, or right adjacent cell in an action. When the agent passes through one of the switch cells, the agent is notified of it (an augmented state space [5]). When the agent has acted for 99 times, each episode ends. In order for the agent to gain a reward, the switches must be pressed in a correct order: switch 1, 2, and then 3. When the agent presses the last switch, a reward 1 is given, and the state of the switches resets. In this experiment, the agent ran 10000 episodes and calculated the average of 1000 simulations. The switch changes its place randomly every 1000 episodes, under the constraint that the new switch configuration is not the same at the previous one. The maximum reward in an episode is always six, because a lap (from switch 1 to 2 to 3) takes 16 moves.

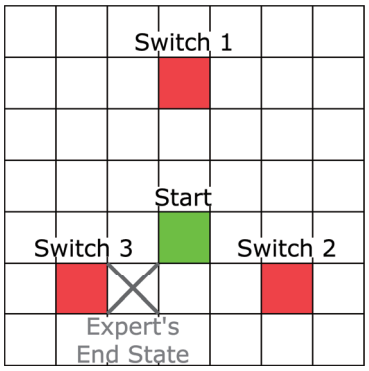

Fig. 1: Unsteady Switch World Task

4. Simulation and Result

QL and MaxEntIRL are operated under the ϵ-greedy policy, in which the agent selects an action at random at probability ϵ and selects the greedy action a with the highest action value $Q(s_t, a_t)$ at probability $1 - \epsilon$. ϵ starts from 1.0 and then decreases by 0.005 per episode, until it reaches 0.025 at episode 200. The aspiration level for *RS*+GRC, $\aleph(s_t)$, is assigned to each state by the Global Reference conversion (GRC). The global aspiration for entire episodes was $\aleph_G = 0.06$. γ_G was set to 0.9. The scaling parameter $\zeta(s_i) = 1.0$ for all s_i[7]. Learning rate α is 0.1 in all methods, and the discount rate γ is set to 0.5 for MaxEntIRL and 0.9 for all the other algorithms. The sample size of expert's trajectory for MaxEntIRL is 100, the learning rate β is 0.01 for reward function estimation and the epoch is 20. For generating the expert's trajectories, we used the Q values of the QL agent, which has already been learned. However, the starting position of the expert was uniformly randomly selected from the state space with an exception, setting the coordinates of the left upper cell $(0, 0)$, the $(2, 5)$ was avoided. The reason why cordinates $(2, 5)$ was avoided is because the experts trajectory starts from the coordinates that is set randomly and ends at $(2, 5)$. In addition, the parameter θ used in the estimated reward function is normalized while keeping the scale with the maximum value being 0.5.

Figure 2 shows the time development of the obtained reward for each episode. *RS*+GRC was overall capable to obtain the rewards, while MaxEntIRL failed to obtain the reward stably. QL can gradually adapt to the environmental change, slower than *RS*+GRC.

5. Discussion

From the results of this experiment, we see that *RS*+GRC can learn faster than QL and MaxEntIRL. MaxEntIRL could not cope well with unsteady environment. Considering the learning speed of *RS*+GRC, *RS*+GRC is conducting a search with an optimistic directionality based

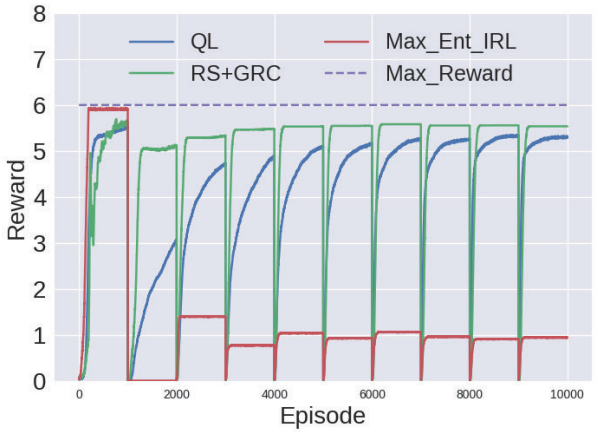

Fig. 2: Time development of reward per episode

on the satisficing policy and searching without much inefficient samplings, rather than ϵ-greedy applied to QL, which is a random search with no directionality.

We would discuss the difference in the behavior of *RS*+GRC and MaxEntIRL in the unsteady environment. The difference between the two is that *RS*+GRC is given only the reference value ($\aleph_G = 0.06$) and MaxEntIRL is given the expert's trajectory as the prior information. For that reason, *RS*+GRC judges the superiority or inferiority of the action sequence only with its result, compared to $\aleph_G$. MaxEntIRL compares the whole action sequence with the action sequence of the expert to judge the superiority or inferiority of the action series. Therefore, *RS*+GRC was able to cope with an unsteady environment because it can truncate the existing action sequence and search for a new action sequence if the result is lower than the reference level.

6. Conclusion

In this study, we showed that emulation implemented in RS can learn faster than QL, and has more flexible search capability than imitation implemented as MaxEntIRL. One of the future tasks is to test our emulation algorithm compared with imitation in continuous environments and to clarify the functional roles of imitation and emulation learning in a broader perspective such as general machine intelligence.

References

[1] Whiten, A. et al.: Emulation, imitation, over-imitation and the scope of culture for child and chimpanzee, *Phil. Trans. of the Royal Soc. B*, 364(1528), 2417–2428. 364 (2009)

[2] Takahashi, T., Kono, Y., Uragami, D., Cognitive Satisficing: Bounded Rationality in Reinforcement Learning, *Trans. Jap. Soc. AI*, 31, 6, AI30-M_1–11. (2016)

[3] Ziebart, B.D. et al.: Maximum Entropy Inverse Reinforcement Learning,*AAAI 2008*. (2008)

[4] Shinriki, M., Kono, Y., Takahashi, T., Emulation Learning from Pioneers, In: *Proc. of JNNS 2018*, PaperID-43, P1-30 ,(2018)

[5] Levy, K.Y., Shimkin, N.: Unified Inter and Intra Options Learning Using Policy Gradient Methods, In *EWRL*, 153164, (2011)

[6] Simon, H.A.: Rational choice and the structure of the environment, *Psychological Review*, Vol. 63, No. 2, pp. 129–138 (1956)

[7] Ushida, U., Kono, U., Takahashi, T., Proc. of JSAI 2017, 4C2-2in2. (2017)