The Fifteenth International Symposium on Artificial Life and Robotics 2010 (AROB 15th '10),
B-Con Plaza, Beppu,Oita, Japan, February 4-6, 2010

A Reinforcement Learning with Switching Controllers for Continuous Action Space

Masato Nagayoshi[a], Hajime Murao[b], and Hisashi Tamaki[c]

[a] Niigata College of Nursing, 240, Shinnan, Joetsu 943-0147, Japan
nagayosi@niigata-cn.ac.jp

[b] Faculty of Cross-Cultural Studies, Kobe Univ. 1-2-1, Tsurukabuto, Nada-ku, Kobe 657-8501, Japan
murao@i.cla.kobe-u.ac.jp

[c] Graduate School of Engineering, Kobe University, Rokko-dai, Nada-ku, Kobe 657-8501, Japan
tamaki@al.cs.kobe-u.ac.jp

Abstract

Reinforcement Learning (RL) attracts much attention as a technique of realizing computational intelligence such as adaptive and autonomous decentralized systems. In general, however, it is not easy to put RL into practical use. This difficulty includes a problem of designing a suitable action space of an agent, i.e., satisfying two requirements in trade-off: (i) to keep the characteristics (or structure) of an original search space as much as possible in order to seek strategies that lie close to the optimal, and (ii) to reduce the search space as much as possible in order to expedite the learning process.

In order to design a suitable action space adaptively, in this paper, we propose a RL model with switching controllers based on Q-learning and Actor-Critic to mimic a process of an infant's motor development in which gross motor skills develop before fine motor skills. Then, a method for switching controllers is constructed by introducing and referring to the "entropy". Further, through computational experiments by using a path planning problem with continuous action space, the validity and the potential of the proposed method have been confirmed.

1 Introduction

In recent years, artificial systems have become extremely complicated and enlarged. The conventional way, in which systems are controlled in a top-down manner mainly by humans, is facing up to the difficulties of not only optimality but also adaptability and flexibility. As one of the solutions to this issue, the development of an autonomously adaptive system has been ongoing. Engineers and researchers are paying more attention to Reinforcement Learning (RL)[1] as a key technique of realizing autonomous systems. In general, however, it is not easy to put RL into practical use. Such issues as satisfying the requirement of learning speed, resolving the perceptual aliasing problem, and designing reasonable state and action spaces of an agent, etc. must be resolved. Our approach mainly deals with the problem of designing the action space. By designing a suitable action space adaptively, it can be expected that the other two problems will be resolved simultaneously. Here, the problem of designing the action space involves the following two requirements: (i) to keep the characteristics (or structure) of an original search space as much as possible in order to seek strategies that lie close to the optimal, and (ii) to reduce the search space as much as possible in order to expedite the learning process. These requirements are, in general, in conflict.

In order to design a suitable action space adaptively, in this paper, we propose a RL model with switching controllers based on Q-learning and Actor-Critic to mimic a process of an infant's motor development in which gross motor skills develop before fine motor skills. Here, the controller based on Q-learning acquires gross motor skills, and the other controller based on Actor-Critic acquires fine motor skills. Then, a method for switching controllers, i.e., adjusting the search space adaptively, is constructed by introducing and referring to the "entropy" which is defined on action selection probability distributions in a state. Some models which combine Q-learning and Actor-Critic have been proposed so far [3, 4]. However, none of the existing models aim to mimic the process of the infant's motor development, nor do they function to switch from Q-learning to Actor-Critic depending on the state, nor is the action selected by Q-learning performed directly.

Through some computational experiments by using a path planning problem, the proposed method is compared with an Actor-Critic method and three Q-learning methods that divide the action space evenly into 4, 8, and 16 spaces.

2 Typical Reinforcement Learning Methods

2.1 Q-learning

Q-learning works by calculating the Quality of a state-action combination, namely Q-value, that gives the expected utility of performing a given action in a given state. By performing an action $a \in \boldsymbol{\mathcal{A}}_{\mathrm{Q}}$, where $\boldsymbol{\mathcal{A}}_{\mathrm{Q}} \subset \boldsymbol{\mathcal{A}}$ is the set of available actions in Q-learning and $\boldsymbol{\mathcal{A}}$ is the action space of the agent, the agent can move from state to state. Each state provides the agent a reward r.

The Q-value is updated according to the following formula, when the agent is provided the reward:

$$Q(s(t\text{-}1)\ a(t\text{-}1)) \leftarrow Q(s(t\text{-}1)\ a(t\text{-}1)) + \ _{\mathrm{Q}}\ r(t\text{-}1) + \gamma \max_{b \in \boldsymbol{\mathcal{A}}_{\mathrm{Q}}} Q(s(t)\ b) \quad Q(s(t\text{-}1)\ a(t\text{-}1)) \quad (1)$$

where $Q(s(t\text{-}1)\ a(t\text{-}1))$ is the Q-value for the state and the action at the time step t-1, $_{\mathrm{Q}} \in [0\ 1]$ is the learning rate of Q-learning, $\gamma \in [0\ 1]$ is the discount factor.

The agent selects an action according to the stochastic policy, $(a|s)$, which based on the Q-value. $(a|s)$ specifies probabilities for taking each action a in each state s. Boltzmann selection, which is one of the typical action-selection methods, is used in this research. Therefore, the policy $(a|s)$ is calculated as follows:

$$(a|s) = \frac{\exp(Q(s\ a)\quad)}{\sum_{b \in \boldsymbol{\mathcal{A}}} \exp(Q(s\ b)\quad)} \quad (2)$$

where is a positive parameter labeled the temperature.

2.2 Actor-Critic

Actor-Critic methods have a separate memory structure to explicitly represent the policy independent of the value function. The policy structure is called "Actor", which selects actions, and the estimated value function is called "Critic", which criticizes the actions made by the Actor. The Critic is a state-value function. After each action selection, the Critic evaluates the new state to determine whether things have gone better or worse than expected. That evaluation is TD-error:

$$(t\text{-}1) = r(t\text{-}1) + \gamma V(s(t)) \quad V(s(t\text{-}1)) \quad (3)$$

where $V(s)$ is the state Value. This TD-error can be used to evaluate the action just selected. If $(t\text{-}1)$ is positive, it suggests that the tendency to select $a(t\text{-}1)$ should be strengthened for the future, whereas if $(t\text{-}1)$ is negative, it suggests the tendency should be weakened.

Then, $V(s(t\text{-}1))$ is updated according to Eq. (4) in the Critic, based on this $(t\text{-}1)$. In parallel, it is updated for the stochastic policy, $(a|s)$, in the Actor.

$$V(s(t\text{-}1)) \leftarrow V(s(t\text{-}1)) + \ _{\mathrm{C}}\ (t\text{-}1) \quad (4)$$

where $_{\mathrm{C}} \in [0\ 1]$ is the learning rate of the Critic.

It is typical for the normal distribution to be used, shown in Eq. (5), as the stochastic policy in the Actor, when Actor-Critic is applied to a continuous action space[2]. In this case, both the mean (s) and the standard error of the mean $\sigma(s)$ about the normal distribution are calculated using TD-error $(t\text{-}1)$ in the Actor, as Eq. (6),(7).

$$(a|s) = \frac{1}{\sigma(s)\sqrt{2}} \exp(\frac{(a \quad (s))^2}{2\sigma(s)^2}) \quad (5)$$

$$(s(t\text{-}1)) \leftarrow \ (s(t\text{-}1)) + \quad (t\text{-}1)(a(t\text{-}1) \quad (s(t\text{-}1))) \quad (6)$$

$$\sigma(s(t\text{-}1)) \leftarrow \sigma(s(t\text{-}1)) + \ _{\sigma}\ (t\text{-}1)((a(t\text{-}1) \quad (s(t\text{-}1)))^2 \quad \sigma(s(t\text{-}1))^2) \quad (7)$$

where $\in [0\ 1]$ $_{\sigma} \in [0\ 1]$ are the learning rate of the mean and the standard error of the mean respectively. Here, if Eq. (7) is used directly, the standard error could be 0 or a negative value. So, it is necessary for the setting of the standard error to be creative to specify the range, etc.

3 A Switching Reinforcement Learning from Q-learning to Actor-Critic

3.1 Outline of a Computational Model

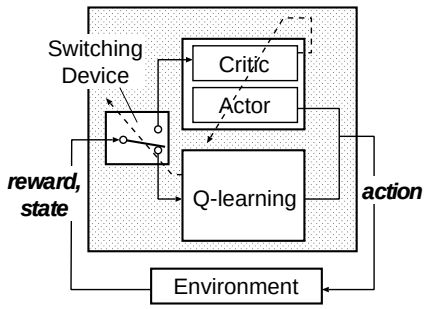

Figure 1: Proposed switching reinforcement learning model.

In this section, we propose a RL model with switching controllers based on Q-learning and Actor-Critic to mimic a process of an infant's motor development.

The proposed model is constructed by two learning controllers and a switching device, as shown in Fig. 1.

Then, the following procedure is conducted to mimic a process of an infant's motor development in which gross motor skills develop before fine motor skills.

1. The controller based on Q-learning (hereafter called "QL controller") estimates the Q-values for each typical action pre-designed by humans. The actions selected by the QL controller correspond to gross motor skills.
2. After "sufficient learning" has been achieved in a state s, the learning controller for the state s is switched from Q-learning to Actor-Critic, as will be described in detail below.
3. The controller based on Actor-Critic (hereafter called "AC controller") adjusts actions continuously. The actions selected by the AC controller correspond to fine motor skills.

It can be expected that the proposed model 1) demonstrates a good performance with regard to the ultimately obtained control rule, because it can adjust actions continuously unlike models using only a QL controller. 2) has a better performance in the early stages of learning than the model using only an AC controller do by switching from the QL controller, where the possible actions are limited beforehand. 3) reduces a designer's load and responsibilities in designing the action space of the QL controller, because the AC controller adjusts actions after switching the controller.

3.2 A Method to Switch Controllers

A variety of methods to switch from the QL controller to the AC controller can be considered. In this paper, we propose a switching method referring to the "entropy", which is defined on action selection probability distributions in a state, and the number of learning opportunities in the state.

The entropy of action selection probability distributions using Boltzmann selection in a state, $H(s)$, is defined by

$$H(s) = \quad (1 \quad \log |\boldsymbol{\mathcal{A}}_{\mathrm{Q}}|) \sum_{a \in \boldsymbol{\mathcal{A}}_{\mathrm{Q}}} \quad (a|s) \log \quad (a|s) \qquad (8)$$

where $|\boldsymbol{\mathcal{A}}_{\mathrm{Q}}|$ is the number of available actions of the QL controller.

The switching method treats this entropy $H(s)$ as an index of sufficiency for the number of learning opportunities in the state.

The controller is switched to Actor-Critic, if the following formula is satisfied:

$$H(s) < \theta_{\mathrm{H}} \qquad (9)$$

In prior studies, Ito at el.[5] have referred to the entropy as the residual entropy, and used the average of the residual entropies when switching from the coarse-graining state space to the fine-graining one. In our early studies[6, 7], we have used the entropy as an index of a correctness of state aggregation when adjusting the aggregation size of s.

In parallel, the controller is also switched to Actor-Critic, if the following formula regarding the number of learning opportunities in s, $L(s)$, is satisfied:

$$L(s) > \theta_{\mathrm{L}} \qquad (10)$$

where θ_{L} is set at a sufficiently big number. This is used because the entropy can not be small after the controller learned a sufficient number of times, if the state space is designed too coarse-graining[6, 7].

When switching controllers, the following procedure is conducted:
i) the state value of the Critic,$V(s)$, is initialized by

$$V(s) = \max_{a \in \boldsymbol{\mathcal{A}}_{\mathrm{Q}}} Q(s \ a) \qquad (11)$$

ii) the normal probability distribution used by the Actor is calculated by

$$(s) = \arg \max_{a \in \boldsymbol{\mathcal{A}}_{\mathrm{Q}}} Q(s \ a) \qquad (12)$$

$$\sigma(s) = |\boldsymbol{\mathcal{A}}| \ \ (6 \cdot |\boldsymbol{\mathcal{A}}_{\mathrm{Q}}|) \qquad (13)$$

where $|\boldsymbol{\mathcal{A}}|$ is a size of the action space of the AC controller. Here, Eq. (13) is presupposed to be designed such that the action space of the QL controller is divided evenly.

4 Computational Example

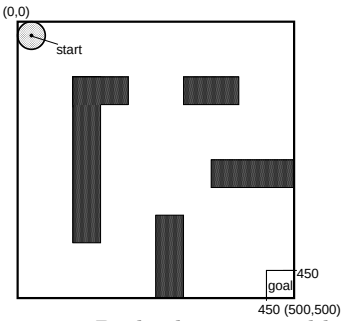

Figure 2: Path planning problem.

The proposed method is applied to a so-called "path planning problem" where an agent is navigated from a start point to a goal area in a continuous space as shown in Fig. 2. Here, the agent has a circular shape (diameter = 50[mm]), and the continuous space

Table 1: Parameters for Experiments.

Method	Parameter	Value
A, Q4, Q8, and Q16	Q	0 1
A and AC	C σ	0 1
all	γ	0 9
A, Q4, Q8, and Q16		0 1
A	θ_H	0 3
A	θ_L	10000

is 500[mm] 500[mm] bounded by the external wall with internal walls as shown in black. The agent can observe the center position of the agent: $(x_A\ y_A)$ as the input, and move 25[mm] in a direction, i.e., decide the direction: θ_A as the output.

The positive reinforcement signal $r_t = 10$ (reward) is given to the agent only when the center of the agent arrives at the goal area and the reinforcement signal $r_t = 0$ at any other steps. The period from when the agent is located at the start point to when the agent is given a reward, labeled as 1 episode, is repeated.

After dividing the state space evenly into 20 20 spaces, the proposed method (hereafter called "method A") is compared with an Actor-Critic method (hereafter called "method AC") and three Q-learning methods that divide the action space evenly into 4, 8, and 16 spaces (hereafter called "method Q4", "method Q8", and "method Q16" respectively).

All initial values of the state and standard errors are set at 0.0 and 1.0 respectively, and all initial means are set to randomize within a range of [0 1 1 0] for the method AC. Then, all initial Q-values are set at 5.0 as the optimistic initial values[1] for Q-learning methods. Here, the initial values and the maximum limit of $\sigma(x)$ are set so that $\pm 3\sigma$ becomes the size of the action space: 2 .

Computer experiments have been done with parameters as shown in Table 1. Here, θ_H was set referring to about 0.312: the maximal value of the entropy when the highest selection probability for one action is 0.9, θ_L was set in consideration of the enough big number.

The number of average steps required to accomplish the task was observed during learning over 20 simulations with various methods as described in Fig. 3.

Learning speed and obtained control rule: It can be confirmed from Fig. 3 that, 1) method A and Q4 have good performances with regard to the learning speed, 2) method A has good performance as well as method AC with regard to the obtained control rule, 3) any proper control rule by method Q4, Q8, and Q16 couldn't be obtained.

Therefore, we have confirmed that method A demonstrates better performance than any other method on the path planning problem with the continuous action space.

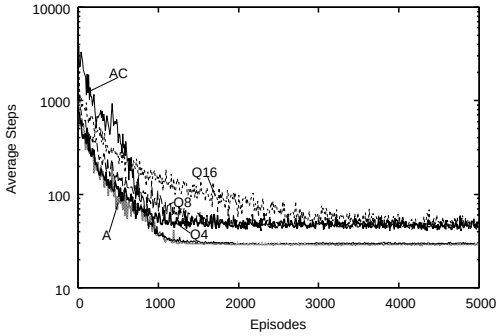

Figure 3: Required steps.

5 Conclusion

In order to design the suitable action space adaptively, we propose in this paper the RL model with switching controllers based on Q-learning and Actor-Critic, and the method for switching controllers referring to the "entropy". Then, through some computational experiments by using the path planning problem with continuous action space, the validity and the potential of the proposed method have been confirmed.

Our future projects include: 1) to apply more complicated problems, 2) to investigate multi-step models for mimicking the process of the infant's motor development, 3) to mimic the process of an infant's perceptual development, etc.

Acknowledgment

This work was supported in part by a Grant-in-Aid for Young Scientists (B), (No. 21700258), from MEXT, Japan.

References

[1] R.S. Sutton and A.G. Barto: Reinforcement Learning, A Bradford Book, MIT Press (1998).

[2] H. Kimura and S. Kobayashi: An Analysis of Actor-Critic Algorithms using Eligibility Traces: Reinforcement Learning with Imperfect Value Functions, *JSAI Jour.*, **15**(2), 267-275 (2000)(in Japanese).

[3] J. Morimoto and K. Doya: Acquisition of stand-up behavior by a real robot using hierarchical reinforcement learning, *Robotics and Autonomous System*, **36**, 37-51 (2001).

[4] K. Shibata, T. Nishino, and Y. Okabe: Active Perception Learning System Based on Actor-Q Architecture, *T. IEICE Japan*, **J84-D-II**(9), 2121-2130 (2001) (in Japanese).

[5] A. Ito and M. Kanabuchi: Speeding up Multi-Agent Reinforcement Learning by Coarse-Graining of Perception - Hunter Game as an Example -, *T. IEICE Japan*, **J84-D-I**(3), 285-293 (2001) (in Japanese).

[6] M. Nagayoshi, H. Murao, and H. Tamaki: A State Space Filter for Reinforcement Learning, *Proc. AROB 11th'06*, 615-618(GS1-3) (2006).

[7] M. Nagayoshi, H. Murao, and H. Tamaki: A State Space Filter for Reinforcement Learning in POMDPs - Application to a Continuous State Space -, *Proc. of the SICE-ICSE International Joint Conference 2006*, 6037-6042(SE18-4) (2006).