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

A Consideration on Immunity-based Reinforcement Learning in a Continuous State Space Environment

Shu Hosokawa, Kazushi Nakano

Dept. of Electronic Eng., The University of Electro-Communications
1-5-1 Chofu-ga-oka, Chofu, Tokyo 182-8585, Japan
(Tel : 81-42-443-5190; Fax : 81-42-443-5183)
(hosokawa@francis.ee.uec.ac.jp, nakano@ee.uec.ac.jp)

Abstract: Many reinforcement learning methods have been studied on the assumption that state is discretized and environment size is pre-determined. However, an operating environment may have a continuous state and its size may not be known in advance such as in robot navigation and control. When applying these methods to the environment described above, we may need a large amount of time for learning or fail to learn. In this study, we improve our previous immunity-based reinforcement learning method to work in continuous state space environment. Since our method selects an action based on the distance between the present state and the memorized action, environment information (e.g. environment size) is not required in advance. The validity of our method is demonstrated through simulations for a swing-up control of an inverted pendulum.

Keywords: Reinforcement learning, Continuous state space, Adaptive immune system

1 Introduction

The Immunity-based reinforcement learning method is built based on the adaptive immune system [1]. This learning method is superior to traditional methods[2][3] in learning speed regardless of the initial and reward values. But, since this approach has an assumption that it works well in a discrete state space environment, it is likely to fail to learn or decrease convergence speed in learning when applied to a continuous state space environment. Even if it learns successfully, it requires a lot of computer memory. For a continuous state space environment, there exist a method based on a combination of discrete learning methods[4], Actor-Critic[5], etc. However, these methods require in advance to set a probabilistic model and/or the number of divisions according to the environmental dimension.

In this study, we improve our previous immunity-based reinforcement learning method so as to extend it applicable to the continuous state space. For this, we reconsider the mechanism of the adaptive immune system, and re-model such kind of learning mechanism. The adaptive immune system can acquire immunity by ingesting pathogens in advance that are similar to other pathogens, such as in vaccinations. Previous learning methods have been used to select an action by using only the information that states memorizing past actions perfectly coincide with sensor observations. Focusing on this point, we take into account the fitness of memorized states and sensor observations, and make use of the fitness and the reward gained from the environment for action selection. The validity of the proposed method is demonstrated through simulations for the swing-up control of the inverted pendulum.

2 Adaptive Immunity-based Reinforcement Learning

This section explains how to eliminate pathogens that invade an human body, and introduce the immunity-based reinforcement learning algorithm to correspond with the above model.

2.1 Summary of adaptive immunity

Figure 1 shows a relationship between cells in adaptive immunity. The pathogen is called antigen. The antigen is captured and recognized by the antigen presenting cell. The antigen presenting cells include B cells, macrophage, etc. The information of the antigen

is presented to T cells. The T cells which the information is presented release the cytokines, and send the signal to the B cells for activation. The activated B cells then produce the antibody to neutralize the antigen. Therefore, the invaded antigen can be eliminated. The T cells playing the above role are called Helper T cells (Th cells). The relationship of B cells - antigens and B cells - Th cells is specific. Generally, T cells and B cells die after eliminating the antigen. But, some activated T cells and B cells have a long lifetime, circulate throughout in the body and survive as memory cells. As a result, the adaptive immunity becomes able to respond quickly and eliminate effectively the same type of antigen.

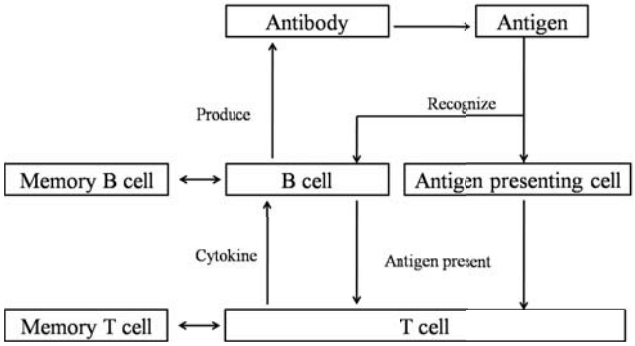

Figure 1: Adaptive immunity

2.2 Action selection algorithm

First of all, the set of all the states where the agent can exist is defined as S. The agent state is defined as $s_i(\in S)$. Th cells memorize state s_i, action a_k, and cytokine signal w_k. In addition, B cells that perform actions a_k is expressed as B_k. The state of B_k activated by antigens, is expressed as m_k. B_k became activated ($m_k = 1$) if the information matches the antigen, otherwise ($m_k = 0$). Action selection is performed based on the values of m_k and w_k by selecting B cells. B cells are selected to execute the action of antibody $A(s_i, k)$ that describes the current state s_i and actions are generated.

An algorithm for B cell selection using the Th database is presented as follows:

1. The agent exists in s_i, Th database releases a cytokine signal $w_k(s_i)$ according to the state. On the other side, B cells present the degree of stimulation m_k according to the current state.

2. After calculating $v_k = m_k \times w_k(s_i)$, a B cell is selected through the roulette selection using v_k of the selection probability for B_k.

3. Antibody $Ab(s_i, k)$ is produced by B_k. The antibody has the parameter called concentration which means the antibody's lifetime. When the antibody is produced, its concentration is set to 1 ($Ab(s_i, k) = 1$). If the same antibody has already been produced, or if the same B cell has already been selected in the past same state, a new antibody is not produced, and the existing antibody's concentration is reset to 1.

4. The concentrations of other antibodies produced in the past are updated with the following equation:

$$A_b \leftarrow \beta \times A_b \tag{1}$$

where $\beta(0 < \beta < 1)$ is the discount rate.

By performing the above process, the agent decides the B cell to be selected.

2.3 Update of Th database

When the agent receives a reward from its environment after it executed an action, the Th database is updated. This means that each $w_k(s_i)$ is updated as

$$w_k(s_i) \leftarrow w_k(s_i) + \alpha(r_k(s_i) - w_k(s_i)) \tag{2}$$

$$r_k(s_i) = \begin{cases} A_b(s_i, k) \times R & : A_b(s_i, k) \text{ produced} \\ 0 & : \text{otherwise} \end{cases} \tag{3}$$

where R is the reward which the agent receives from its environment, and $\alpha(0 < \alpha < 1)$ is the learning rate. This update formula is performed for all w. After updating, all antibodies are erased.

The agent becomes able to select an appropriate rule for its environment by repeating the learning with the above process of rule selection.

3 Immunity-based Reinforcement Learning in a Continuous State Space Environment

This section discusses how to improve the immunity-based reinforcement learning to work in a continuous state space environment. Traditional reinforcement

learning methods used cytokine w_k of Th cell with coincidence of the memorized states and sensor observations for computing $v(k)$ in action selection. However, there is almost no matching of the memorized states and sensor observations in a real continuous state space.

In the actual adaptive immune system, Th cells do not recognize the whole individual antigen. Th cells change with their activities based on the fitness of a part of the original antigen degraded by antigen-presenting cells[6]. Th cells are activated when antigens are presented with the degree of similarity with their own receptors, and release cytokine signals to B cells. Focusing on this mechanism, we make use of the distance between the present and the memorized states for the activity of Th cells. The cytokine signal and the activity of Th cells are used for action selection.

The following is presented as a modification of the immunity-based reinforcement learning algorithm. Th cell is generated as a cell which records the continuous state ($\boldsymbol{\xi} = [\xi_1, \xi_2 \cdots, \xi_n]$), action ($a$) and evaluation value which is explained below. Here, only a special Th cell (Th_0) which outputs the same cytokine signal to all states and actions is generated. First, set the initial value of the evaluation to this cell. Next, give the following formula to calculate the activity of the current state ($\boldsymbol{\xi}'$) and the memorized state $\boldsymbol{\xi}^j$:

$$L(Th_j, a_k) = \begin{cases} \gamma \sum_{p=1}^{n} |\xi'_p - \xi^j_p| & a_k \text{ memorized} \\ \infty & \text{otherwise} \end{cases} \quad (4)$$

Eq.(4) is the Manhattan distance, which is defined as the sum of the distances of all dimensions. γ is a gain parameter given as a positive value. The larger value of γ, the smaller number of cells. This corresponds exactly with fine discretization of a continuous state space. A cytokine is obtained which outputs Th cell by calculating the values of the activity and the evaluation.

$$w_k = \sum_{j=0}^{N} \frac{W_j}{\exp(L(Th_j, a_k))} \quad (5)$$

where N is the total number of Th cells and W_j is the evaluation value memorized in j-th Th (Th_j).

Our action selection algorithm with considering a continuous state space is given as follows:

1. The agent exists in $\boldsymbol{\xi}$, then B Cells present the degree of stimulation m_k according to current state.

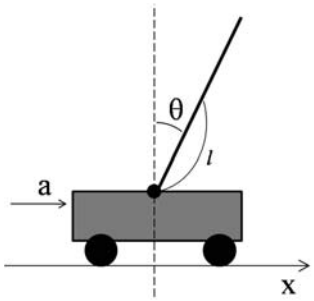

Figure 2: Swing-up of inverted pendulum

2. Th cell cytokine output(w_k) is calculated by using Eq (5).
3. A B cell is selected through the roulette selection with v_k as the selection probability for B_k after calculating $v_k = m_k \times w_k$.
4. Antibody $Ab(s_i, k)$ is produced by B_k. The antibody has the concentration which means the antibody's lifetime. When the antibody is produced, its concentration is set to 1 ($Ab(s_i, k) = 1$). If the same antibody has already been produced or the same B cell has already been selected in the past same state, a new antibody is not produced, and the existing antibody's concentration is reset to 1.
5. The concentrations of other antibodies produced in the past are updated with Eq.(1).

When the agent receives a reward from its environment, W_j is updated as follows:

1. Th cell is generated from the antibody's information. After setting the following evaluation, we erase the antibody information:

$$W_j = A_b(\boldsymbol{\xi}, k) \times R \quad (6)$$

2. All the evaluation values of Th cells are updated as

$$W_j \leftarrow W_j(1 - \alpha) \quad (7)$$

4 Simulation Results

This section shows simulation results of our proposed method applying to the swing up control of the inverted pendulum(Fig.2). The motion equation of the inverted pendulum is described by

$$(M+m)\ddot{x} + ml\cos\theta\ddot{\theta} + D_x\dot{x} + ml\sin\theta\dot{\theta} = a \quad (8)$$

$$-ml\cos\theta\ddot{x} + I\ddot{\theta} + D_\theta\dot{\theta} - mgl\sin\theta = 0 \quad (9)$$

Table 1: Initial state and target state

Parameter	Initial state	Target state
x	0	don't care
$\dot{x}$	0	0 ± 0.5
θ	π	0 ± 0.5
$\dot{\theta}$	0	0 ± 0.2

where, M is the mass of the truck, m is the weight of the pendulum, l is the length of the pendulum to the center of gravity, D_x is the friction on the truck, D_θ is the friction of the pendulum rotation, I is the moment on the rotation of the pendulum. In this simulation, we perform a learning of the swig-up control of the inverted pendulum through selections of $a \in A = [-10, 0, 10]$ (action list). The control task is to swing up the pendulum from its natural pendant position ($\theta = \pi$) and stabilize it in the inverted position ($\theta = 0$) under the assumption that all the physical parameters are unknown.

Table 1 shows the initial state and target state. The states $x, \dot{x}, \theta$ and $\dot{\theta}$ are inputted to the learning module, whose values are observed with the uniform random values (± 0.1) added. If we can reach the target, we give the reward of 10 and end one episode. The truck motion is limited on the range of $-10 \leq x \leq 10$. If going out of the range, the truck should be stopped ($\dot{x} = 0$). Also, if we could not reach the target even after 5000 steps, we restart to learn from the next episode without any reward. Figure 3 shows the results by using the proposed reinforcement learning method.

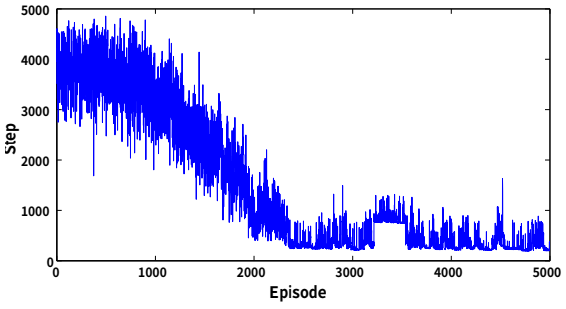

Figure 3: Result of swing-up control

Our method can acquire a swing-up control action even in an observation noise case. The discrete state - reinforcement learning method could not acquire any swing-up -.

5 Conclusions

We improved the immunity-based reinforcement learning method in order to extend it applicable to a continuous state space environment. Also the method was verified by simulations for a swing-up control of an inverted pendulum. As a result, our proposed method was able to perform a learning even in the continuous state space environment.

In our future works, we should handle negative rewards (penalty). However, our proposed method using a roulette selection for action selection, cannot handle negative evaluation values. We need to develop a new reward function or a new method for action selection where the negative reward can be used for stabilizing control.

References

[1] J. Ito, K. Nakano, K. Sakurama and S. Hosokawa: "Adaptive immunity based reinforcement learning", Artificial Life and Robotics, **13**, 1, pp.188–193 (2008).

[2] C. J. C. H. Watkins and P. Dayan: "Technical note: q-learning", Mach. Learn., **8**, 3-4, pp.279–292 (1992).

[3] J. J. Grefenstette: "Credit assignment in rule discovery systems based on genetic algorithms", Readings in Machine Learning (Eds. by J. W. Shavlik and T. G. Dietterich), Kaufmann, San Mateo, CA, pp.524–534 (1988).

[4] T. Matsui, N. Inuzuka and H. Seki: "Profit sharing with linear function approximation", The 16th Annual Conference of Japanese Society for Artificial Intelligence, pp.2D3–03 (2002) (in Japanese).

[5] H. Kimura and S. Kobayashi: "An analysis of actor-critic algorithms using eligibility traces : Reinforcement learning with imperfect value functions", Journal of Japanese Society for Artificial Intelligence, **15**, 2, pp.267–275 (2000) (in Japanese).

[6] S. Kakiuchi, K. Ikebuchi, K. Ota, Y. Kobayashi, R. Simosato and Z. Mizuguchi (Eds.): "Immunology handbook", Ohm sha (2006) (in Japanese).