A Fuzzy Reinforcement Learning for a Ball Interception Problem

Tomoharu Nakashima, Masayo Udo, and Hisao Ishibuchi

Department of Industrial Engineering, Osaka Prefecture University
Gakuen-cho 1-1, Sakai, Osaka, 599-8531
{nakashi,udo,hisaoi}@ie.osakafu-u.ac.jp

Abstract. In this paper, we propose a reinforcement learning method called a fuzzy Q-learning where an agent determines its action based on the inference result by a fuzzy rule-based system. We apply the proposed method to a soccer agent that intercepts a passed ball by another agent. In the proposed method, the state space is represented by internal information the learning agent maintains such as the relative velocity and the relative position of the ball to the learning agent. We divide the state space into several fuzzy subspaces. A fuzzy if-then rule in the proposed method represents a fuzzy subspace in the state space. The consequent part of the fuzzy if-then rules is a motion vector that suggests the moving direction and velocity of the learning agent. A reward is given to the learning agent if the distance between the ball and the agent becomes smaller or if the agent catches up with the ball. It is expected that the learning agent finally obtains the efficient positioning skill.

1 Introduction

Fuzzy rule-based systems have been applied mainly to control problems [1]. Recently, fuzzy rule-based systems have also been applied to pattern classification problems. There are many approaches to the automatic generation of fuzzy if-then rules from numerical data for pattern classification problems.

Reinforcement learning [2] is becoming a more and more important research field for acquiring the optimal behavior of autonomous agents. One of the most well-known reinforcement learning methods is Q-learning [3]. In the original Q-learning, it is assumed that both a state space and an action space is discretely defined. The optimal discrete action is obtained for each discrete state in a learning environment through updating the mapping from a state-action pair to a real value called Q-value.

In order to deal with continuous state space and action space, various methods have been proposed such as tile coding, neural networks, linear basis functions and so on (see [2] for detail). Fuzzy theory has been also successfully applied in order to extend the Q-learning to fuzzy Q-learning. For example, Glorennec [4] proposed a fuzzy Q-learning algorithm for obtaining the optimal rule base for a fuzzy controller. Horiuchi et al. [5] proposed a fuzzy interpolation-based Q-learning where a fuzzy rule base is used to approximate the distribution of

D. Polani et al. (Eds.): RoboCup 2003, LNAI 3020, pp. 559–567, 2004.

Q-values over a continuous action space. In [5], action selection was performed by calculating Q-values for several discrete actions and then selecting one action through the roulette wheel selection scheme.

In this paper, we propose a fuzzy Q-learning that can deal with a continuous state space and a continuous action space. Q-values are calculated using fuzzy inference from the sensory information of the learning agent. We apply the proposed method to a soccer agent that tries to learn to intercept a passed ball. That is, it tries to catch up with a passed ball by another agent. In the proposed method, the state space is represented by internal information that the learning agent maintains such as the relative velocity and the relative position of the ball. We divide the state space into several fuzzy subspaces. We define each fuzzy subspace by specifying the fuzzy partition of each axis in the state space. A reward is given to the learning agent if the distance between the ball and the agent becomes smaller or if the agent catches up with the ball. Simulation results show that the learning agent can successfully intercept the ball over time.

2 Ball Interception Problem

The problem we solve in this paper is called a ball interception problem, where the task of the learning agent is to follow the passed ball by another agent. This problem is illustrated in Fig. 1. First, the passer approaches the ball to kick it. Then learning agent tries to catch up with the passed ball. Let (x_a, y_a) and (x_b, y_b) be the absolute position of the learning agent and the ball, respectively. We also denote the velocity of the learning agent and the ball as (v_{ax}, v_{ay}) and (v_{bx}, v_{by}), respectively. Suppose that the learning agent moves at the speed of (v_{ax}, v_{ay}) from the position (x_a, y_a) at the time step 0, and the learning agent can intercept the ball at the time step t. The position of the ball at the time step 0 is (x_b, y_b). Here we do assume that there is no noise nor friction in the movement of the objects. The positions of the learning agent and the ball at the time step t are $(x_a + v_{ax}t, y_a + v_{ax}t)$ and $(x_b + v_{bx}t, y_b + v_{bx}t)$, respectively. In order for the learning agent to successfully intercept the ball, it is necessary that the following two conditions hold:

$$x_a + v_{ax}t = x_b + v_{bx}t, \tag{1}$$

and

$$y_a + v_{ay}t = y_b + v_{by}t. \tag{2}$$

Thus, we have

$$t = \frac{x_a - x_b}{v_{bx} - v_{ax}} = \frac{y_a - y_b}{v_{by} - v_{ay}}. \tag{3}$$

The objective of the ball interception problem can be viewed as the minimization of the time to intercept the ball. There is a constraint that the learning agent can not move at more than some pre-specified maximal speed. Let us denote the maximal speed of the learning agent as $V_{\max}$. Then the ball interception problem can be rewritten as follows:

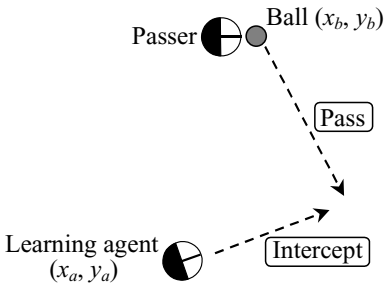


Fig. 1. Ball interception problem.

[Ball interception problem]
Minimize t,
subject to $V_{\max} \leq \sqrt{v_{ax}^2 + v_{ay}^2}$.

3 Q-Learning

In the conventional Q-learning, a Q-value is assigned to each state-action pair. The Q-value reflects the expected long-term reward by taking the corresponding action from the action set A in the corresponding state in the state space S. Let us denote $Q_t(s, a)$ as the Q-value for taking action a in state s at time step t. When a reward r is obtained immediately after taking action a, the Q-value $Q_t(s, a)$ is updated as follows:

$$Q_{t+1}(s, a) = (1 - \alpha) \cdot Q_t(s, a) + \alpha \cdot (r + \gamma \hat{V}_t), \tag{4}$$

where α is a learning rate, γ is a discount factor, and $\hat{V}$ is the maximum Q-value in the state s' after taking the action a in the state s at the time step t, which is defined as

$$\hat{V} = \max_{b \in A} Q_t(s', b), \tag{5}$$

where s' is the next state assuming that the agent took action a in the state s.

The action selection in the Q-learning is done considering a trade-off between exploration and exploitation of the state-action pairs. The roulette wheel selection scheme is often used for selecting an action where the following Boltzmann distribution is used with the selection probability $P(s, a)$ of the action a:

$$P(s, a) = \frac{\exp(Q(s, a)/T)}{\sum_{b \in A} \exp(Q(s, b)/T)}, \quad \text{for } a \in A. \tag{6}$$

The main procedure of the Q-learning is as follows. First, the learning agent observes the state of the environment. Next, action selection is performed according to the Q-values for the observed state. Then, Q-values corresponding to the selected action is updated using the reward from the environment. This procedure is iterated until some pre-specified stopping criterion is satisfied.

The Q-values are stored in a Q-table that is referred by the learning agent for retrieving Q-values for action selection. One difficulty in the Q-learning is so-called *curse of dimensionality*. That is, the number of Q-values to be stored in the Q-table is intractably large when the state space is large. Another problem is that the conventional Q-learning can not be applied when the state space and/or the action set is continuous. In order to overcome this problem, we propose fuzzy Q-learning that can deal with continuous state and continuous action.

4 Fuzzy Q-Learning

4.1 Fuzzy If-Then Rule

Let us assume that we would like an agent to learn the optimal behavior in a continuous state space with continuous actions. We also assume that the state space in the fuzzy Q-learning is described by n-dimensional real vector $\mathbf{x} = (x_1, \ldots, x_n)$ and there are m representative values W_k, $k = 1, \ldots, m$, for determining a single continuous action. In the fuzzy Q-learning, we use fuzzy if-then rules of the following type:

$$R_j : \text{If } x_1 \text{ is } A_{j1} \text{ and } \ldots \text{ and } x_n \text{ is } A_{jn} \text{ then } \mathbf{w}_j = (w_{j1}, \ldots, w_{jm}), \quad j = 1, \ldots, N, \tag{7}$$

where R_j is the label of the fuzzy if-then rule, A_{ji}, $i = 1, \ldots, n$, is a fuzzy set for a state variable, $\mathbf{w}_j$ is a consequent real vector of the fuzzy if-then rule R_j, and N is the number of fuzzy if-then rules. As fuzzy sets we can use any type of membership functions such as triangular, trapezoidal, and Gaussian type. Each element of the consequent vector $\mathbf{w}_j$ corresponds to the weight for a typical vector $\mathbf{a} = (a_1, \ldots, a_m)$ in the continuous action space.

4.2 Action Selection

From a state vector $\mathbf{x} = (x_1, \ldots, x_n)$, the overall weights of each typical point in the continuous output space is calculated through fuzzy inference as follows:

$$W_k = \frac{\sum_{j=1}^{N} w_{jk} \cdot \mu_j(\mathbf{x})}{\sum_{j=1}^{N} \mu_j(\mathbf{x})}, \qquad k = 1, \ldots, m, \tag{8}$$

where $\mu_j(\mathbf{x})$ is the compatibility of a state vector $\mathbf{x}$ with the fuzzy if-then rule R_j that is defined by a multiplication operator as follows:

$$\mu_j(\mathbf{x}) = \mu_{j1}(x_1) \cdot \mu_{j2}(x_2) \cdot \ldots \cdot \mu_{jn}(x_n), \tag{9}$$

where $\mu_{jk}(x_k)$ is the membership function of the fuzzy set A_{jk}, $k = 1, 2, \ldots, n$ (see (7)). While various schemes for action selection such as Boltzmann selection and ϵ-greedy selection can be used as in the conventional Q-learning, we use

a fuzzy inference scheme for selecting the action of the learning agent. That is, we do not use the explorative action selection method but the exploitative action selection method that is strictly determined by the fuzzy inference of the fuzzy rule base in (7). The similar approach was used in [6] where his proposed TPOT-RL method was used for optimizing the packet routing problem without any explorative action selection but only with greedy action selection.

The final output o is calculated as

$$o = \frac{\sum_{k=1}^{m} a_k \cdot W_k}{\sum_{k=1}^{m} W_k}. \tag{10}$$

4.3 Updating Fuzzy If-Then Rules

After the selected action was performed by the learning agent, the environment provides it with either a reward or a punishment according to the resultant state of the environment. Assume that the reward r is given to the learning agent after performing the selected action which is determined by (10). Weight values of each fuzzy if-then rule is updated by

$$w_{jk}^{\text{new}} = (1 - \alpha'_{jk}) \cdot w_{jk}^{\text{old}} + \alpha'_{jk} \cdot (r + \gamma \cdot W_{\max}), \tag{11}$$

where r is a reward, $W_{\max}$ is the maximum value among W_k, $k = 1, \ldots, m$ before the update, γ is a positive constant, and α' is an adaptive learning rate which is defined by

$$\alpha'_{jk} = \alpha \cdot \frac{\mu_j(\mathbf{x})}{\sum_{s=1}^{N} \mu_s(\mathbf{x})} \cdot \frac{W_k}{\sum_{t=1}^{m} W_t}, \tag{12}$$

where α is a positive constant.

5 Computer Simulations

In order to apply the fuzzy Q-learning to the ball interception problem, we use the fuzzy if-then rules of the following type:

$$\begin{aligned} R_j : \text{If } x_r \text{ is } A_{j1} \text{ and } y_r \text{ is } A_{j2} \text{ and } v_{rx} \text{ is } A_{j3} \text{ and } v_{ry} \text{ is } A_{j4} \\ \text{then } \mathbf{v} = (\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3, \mathbf{v}_4) \text{ with } \mathbf{w}_j = (w_{j1}, w_{j2}, w_{j3}, w_{j4}), \\ j = 1, \ldots, N, \end{aligned} \tag{13}$$

where (x_r, y_r) and (v_{rx}, v_{ry}) is the relative position and the relative velocity of the ball to the learning agent, $\mathbf{v}_j = (v_{jx}, v_{jy})$ is the velocity of the learning speed by the fuzzy if-then rule R_j, and $\mathbf{w}_j = (w_{j1}, w_{j2}, w_{j3}, w_{j4})$ is the vector of the recommendation degree for each velocity. Thus the following relations hold:

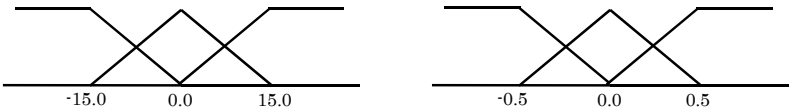
$$x_r = x_b - x_a, \tag{14}$$

$$y_r = y_b - y_a, \tag{15}$$

$$v_{rx} = v_{bx} - v_{ax}, \tag{16}$$

$$v_{ry} = v_{by} - v_{ay}. \tag{17}$$

As fuzzy sets for each state variable, we use triangular type of fuzzy partitions in Fig. 2. Since we partition each state variable into three fuzzy sets, the total number of combinations of the antecedent fuzzy sets is $N = 3^4 = 81$.



(a) Fuzzy partition for x_r and y_r. (b) Fuzzy partition for v_{rx} and v_{ry}.

Fig. 2. Fuzzy partitions.

Each of the consequent part $\mathbf{v} = (\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3, \mathbf{v}_4)$ represents the recommended velocity of the learning agent by R_j. In this paper, we use four combinations of v_{ax} and v_{ay} to calculate the motion vector of the learning agent. They are determined according to the maximum velocity of the learning agent in the constraint of the ball interception problem (see Section II). We use the following four recommended velocities (also see Fig. 3):

$$\mathbf{v}_1 = (v_x^{\max}, 0), \tag{18}$$

$$\mathbf{v}_2 = (0, v_y^{\max}), \tag{19}$$

$$\mathbf{v}_3 = (-v_x^{\max}, 0), \tag{20}$$

$$\mathbf{v}_4 = (0, -v_y^{\max}). \tag{21}$$

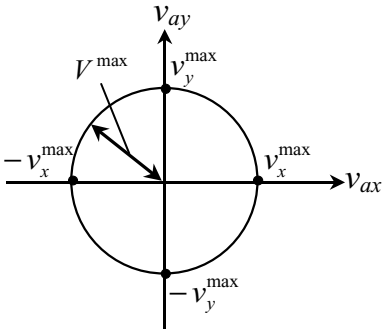


Fig. 3. Four typical velocities in the continuous action space.

After receiving the state vector $\mathbf{s} = (x_r, y_r, v_{rx}, v_{ry})$, the motion vector of the learning agent is determined using the weights for the recommended velocities as follows:

$$W_k = \frac{\sum_{j=1}^{81} w_{jk} \cdot \mu_j(\mathbf{s})}{\sum_{j=1}^{81} \mu_j^t}, \qquad k = 1, 2, 3, 4, \tag{22}$$

where $\mu_j(\mathbf{s})$ is the compatibility of the fuzzy if-then rule R_j to the state vector $\mathbf{s}$ and defined as the product operator as follows:

$$\mu_j(\mathbf{s}) = \mu_{j1}(x_r) \cdot \mu_{j2}(y_r) \cdot \mu_{j3}(v_{rx}) \cdot \mu_{j4}(v_{ry}), \tag{23}$$

where $\mu_{jk}(\cdot)$, $k = 1, 2, 3, 4$ is the membership of the antecedent fuzzy set A_{ji}.

The action is determined by the interpolation among the recommendation degrees of the four typical velocities. That is, the velocity of the learning agent (v_{ax}, v_{ay}) is determined by the following equation:

$$v_{ax} = \frac{W_1 \cdot v_x^{\max} - W_3 \cdot v_x^{\max}}{W_1 + W_2 + W_3 + W_4}, \tag{24}$$

$$v_{ay} = \frac{W_2 \cdot v_y^{\max} - W_4 \cdot v_y^{\max}}{W_1 + W_2 + W_3 + W_4}. \tag{25}$$

In this paper, we consider two types of reward to the learning agent. One is task reward r_t that is given when the learning agent can successfully intercept the passed ball. Since the task reward is sparsely given to the agent, we use an intermediate reward r_i that is given to the learning agent when the learning agent can reduce the distance between the passed ball and the learning agent. In our computer simulations in this paper, we specified those rewards as $r_t = 5$ and $r_i = 1$. Note that when the learning agent goes away from the passed ball, the negative value is given (i.e., $r_i = -1$) to the learning agent as the punishment.

The consequent weight vector $\mathbf{w}_j = (w_{j1}, w_{j2}, w_{j3}, w_{j4})$, $j = 1, \ldots, 81$, is updated by the following equation:

$$w_{jk}^{\text{new}} = (1 - \alpha'_{jk}) \cdot w_{jk}^{\text{old}} + \alpha' \cdot (r + \gamma \cdot W_k), \quad k = 1, 2, 3, 4, \tag{26}$$

where γ is the discount rate, and α'_{jk} is the learning rate, and r is the total reward to the learning agent. α'_{jk} and r are determined by the following equations:

$$\alpha'_{jk} = \alpha \cdot \frac{\mu_j(\mathbf{s})}{\sum_{l=1}^{81} \mu_l(\mathbf{s})} \cdot \frac{W_k}{\sum_{m=1}^{4} W_m}, \tag{27}$$

$$r = r_i + r_t. \tag{28}$$

We applied the proposed fuzzy Q-learning to RoboCup server 7.09 which is available from the URL *http://sserver.sourceforge.net/*. In our computer simulations, one trial ends when the learning agent can successfully intercept the passed ball or the maximum time step is reached. We specified the maximum time step as $t_{\max} = 300$ simulator cycles. Before the computer simulations, we set the initial values for the recommendation degree $\mathbf{w}_j = (w_{j1}, w_{j2}, w_{j3}, w_{j4})$ in the fuzzy if-then rule R_j randomly from the unit interval $[0, 1]$.

We performed the computer simulation for 300 trials. Every 25 trials we examined the performance of the fuzzy Q-learning by making the learning agent

intercept the passed ball with the fixed learned fuzzy if-then rules for 50 trials. We show simulation results of the fuzzy Q-learning in Fig. 4. Fig. 4 shows the success rates over the 50 runs of performance evaluation with the fuzzy if-then rules fixed. From Fig. 4, we can see that the number of the successful intercept increases as the number of trials increases. The learning curve in Fig. 4 did not monotonically increase to the number of trials. This is because the RoboCup server employs some degree of noise in the movement of objects such as the ball and agents and in the perception such as the information on the location and velocity of the objects. Also we observed the effect of the result of learning at the previous trials on the performance of succeeding learning. For example, when the learning agent successfully intercept the passed ball and received the positive reward, the next trial is likely to succeed as well. On the other hand, the learning agent is not likely to succeed when the negative reward was given to the agent in the previous trial.

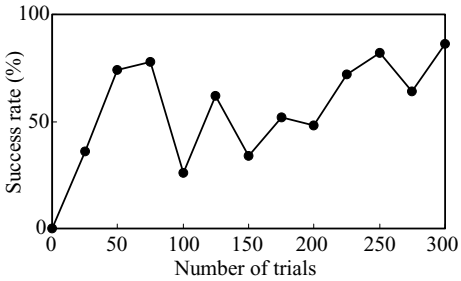


Fig. 4. Simulation results.

6 Conclusion

In this paper, we proposed a fuzzy Q-learning that can deal with the continuous state space and the continuous action space. In the fuzzy Q-learning, a fuzzy rule base is maintained that is used to calculate the recommendation degree by the fuzzy inference scheme. The action was determined by the exploitation-only scheme that includes no exploration procedure of continuous actions. The reward was used for updating the recommendation degrees of fuzzy if-then rules. In the computer simulation, we applied the fuzzy Q-learning to the ball intercept problem. Simulation results showed that the learning agent can gradually learn to successfully intercept the passed ball by another agent.

References

1. M. Sugeno, "An Introductory Survey of Fuzzy Control", *Information Science*, Vol. 30, No. 1/2 pp. 59-83, 1985.
2. R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, MIT Press, 1998.

3. C. J. C. H. Watkins and P. Dayan, "*Q*-Learning", *Machine Learning*, Vol. 8, pp. 279–292, 1992.
4. P. Y. Glorennec, "Fuzzy *Q*-Leaning and Dynamical Fuzzy *Q*-Learning", *Proc. of FUZZ-IEEE'94*, pp. 474–479, 1994.
5. T. Horiuchi, A. Fujino, O. Katai, and T. Sawaragi, "Fuzzy Interporation-Based *Q*-Learning with Continuous States and Actions", *Proc. of FUZZ-IEEE'96*, pp. 594–600, 1996.
6. P. Stone, *Layered Learning in Multiagent Systems – A winning Approach to Robotic Soccer*, MIT Press, 2000.