Building A Fuzzy Inference System By An Extended Rule Based Q-Learning

Min-Soeng Kim, Sun-Gi Hong and Ju-Jang Lee

* Dept. of Electrical Engineering and Computer Science, KAIST
373-1 Kusung-Dong Yusong-Ku
Taejon 305-701, Korea
Tel : +82-42-869-8032, +82-42-869-3432
{mibella,sghong}@odyssey.kaist.ac.kr, jjlee@ee.kaist.ac.kr

Building a Fuzzy Inference System (FIS) generally requires experts' knowledge. However, experts' knowledge is not always available. When there is few experts' knowledge, it becomes hard to build a FIS using one of supervised learning methods.

Meanwhile, Q-learning is a kind of reinforcement learning where an agent can acquire knowledge from its experiences even without the model of the environment and experts' knowledge. The Q-learning, however, has weakness that the original algorithm cannot deal with the continuous states and continuous actions.

In this paper, we proposed a FIS that can do Q-learning. The proposed FIS structure is made up of several extended rules. Based on these extended rules, Q-learning algorithm for the proposed structure is developed. It is shown that this combination results in a FIS that can learn through its experience without experts' knowledge. Also the proposed structure can resolve the continuous state/action problem in Q-learning by virtue of a FIS. The effectiveness of the proposed structure is shown through simulation on the cart-pole system.

1 Introduction

A FIS consists of several fuzzy if-then rules. An extraction of these fuzzy rules requires a priori experts' knowledge. However, it is not always easy to obtain experts' knowledge. Generally, determining consequent parts of fuzzy rules are considered to be more difficult than determining antecedent parts. There have been various approaches to obtain fuzzy rules in a supervised way.[1][2]

Most of those methods needs a large set of experimental data, which becomes expensive as necessary data increase or the complexity of a problem increases.

Meanwhile, Q-learning[3] is a kind of reinforcement learning where an agent can acquire knowledge from its experiences even without the model of the environment and experts' knowledge. Q-value represents a quality value for a certain action in a certain state. An agent can learn to select its proper action in each state based by using and updating these Q-values. However, most research done in the field of Q-learning has focused on discrete domains, although the environment with which the agent must interact is usually continuous. There has been much research to make Q-learning deal with the continuous state/action spaces.[4-6] In these methods, however, the action selection method is based on a step search in action space. That is, after the Q-value for all actions in the possible action range is calculated, the action of which the Q-value is the maximum is selected and this action is performed. The continuous property of action is important from the practical view because the control input cannot change rapidly by an actuator.

Meanwhile, a FIS usually has a fuzzifier that translates real-valued input into fuzzy values and a defuzzifier that translates fuzzy output values into real-valued output. Thus, continuous states or continuous actions can be described by a finite set of fuzzy variables. Thus, the FIS seems to be an effective model to overcome the continuous state/action problem in Q-learning.

In this paper, we proposed a FIS that can do Q-learning. The proposed FIS structure is made up of several extended rules. Based on these extended rules, Q-learning algorithm for the proposed structure is developed. It is shown that this combination results in a FIS that can learn through its experience without experts' knowledge. Also the proposed structure can resolve the continuous state/action problem in Q-learning by virtue of a FIS.

The remainder of this paper is organized as follows. In Section 2, after the basic fuzzy rule is briefly reviewed, the concept of an extended rule is developed. The representation scheme of Q-value based on an extended rule is discussed. The proposed structure and the learning algorithm for this structure are shown in Section 3. In Section 4, the simulation results are presented to show the effectiveness of the proposed methods. Section 5 presents conclusion about the characteristics of the proposed methods.
2 An Extended Fuzzy Rule

2.1 A basic fuzzy rule

In fuzzy rule, premises and conclusions are expressed by means of linguistic terms. A FIS rule base is made of N different fuzzy rules of the following general form:

\[ R_i : \text{if } s'_1 \text{ is } A'_1 \text{ and } s'_2 \text{ is } A'_2 \cdots \text{ and } s'_n \text{ is } A'_n \text{ then } y' \text{ is } B'_i \]  

where \( s \in S^k \) is an input state variable. \( R_i \) represents i-th rule of a rule base. \( A' \) and \( B' \) are fuzzy memberships characterized by linguistic labels (e.g., small, large and big etc) and by a function \( s \rightarrow \mu(s) \in [0,1] \) and \( s \rightarrow \mu(s) \in [0,1] \) respectively.

2.2 An extended fuzzy rule

To incorporate Q-learning in FIS, an extended rule is proposed in this paper. The extended rule is different from the basic fuzzy rule and has several characteristics as follows:

- There are several candidate consequent parts (i.e., candidate actions), for one antecedent part of each rule.
- Each candidate action is a fuzzy membership function and has its own Q value.
- Rule action is selected among candidate actions based on each candidate’s Q-value.
- Rule action becomes an actual consequent part of the rule.
- Final action is obtained by defuzzification.

This extended rule can be described as follows:

\[ R_i : \text{if } s'_1 \text{ is } L'_1 \text{ and } s'_2 \text{ is } L'_2 \cdots \text{ and } s'_n \text{ is } L'_n \text{ then } a' \text{ is } \pi(u'_1, Q'_i) \]  

where \( s \in S^k \) is an input state variable and \( L'_n \) is a linguistic label related to each input state. \( a' \) is a rule action which corresponds to the actual consequent part of i-th fuzzy rule, \( R_i \). \( \pi(.,.) \) is a policy that determines the rule action from a candidate action set \( A_i \). And \( A_i \) is a collection of action \( u'_j \), which is a candidate action for rule action. Every individual \( u'_j \) has its own Q value, \( Q'_i \). Superscript \( i \) and subscript \( i \) represent the rule number and \( i=1 \cdots n \).

In each extended fuzzy rule, one candidate action from \( A_i \) is selected based on its Q-value and the policy. To apply Q-learning, Q-value for each candidate action must be updated. A Rule’s candidate action is currently selected action, among candidate consequent parts, and become an actual consequent part. These selected rule actions generate final action by defuzzification. The final action, however, is generally different from each rule’s rule action. Because Q value is the functional value of the state and the action, different value of action cannot be used in obtaining Q-value for the current state and the final action. That is, we must have Q-value for the final action in each rule to obtain Q(current state, final action) by defuzzification. Thus some kind of technique to obtain the Q-value corresponding to final action in each rule have to be devised, the interpolation technique is used to obtain a Q value for the final action in each rule in this paper.

2.3 Q-value representation in an extended rule

Now let us consider how to represent an appropriate Q-value in an extended rule. The objective is to obtain a Q-value of which action parameter is equal \( a_f \) in each rule. A \( Q(s,a) \) value for a certain action \( a \) which is not included in the discrete action set \( A \) can be calculated using the interpolation technique. Once the final action \( a_f \) is determined from the fuzzy rule base, the Q value for the final action in i-th rule \( R_i \) can be calculated as follows.

\[ Q(R_i,a_f) = \frac{1}{\sum_{j=1}^{p} K(u'_j,a_f)} K(u'_j,a_f)Q'_i \]  

where \( K(.,.) \) is a kernel function which determines the degree of how much each \( (u'_j, Q'_i) \) pair in i-th rule will contribute to calculating \( Q(R_i,a_f) \) and has the following form:

\[ K(u'_j,a_f) = \exp\left(-\frac{(a_f - u'_j)^2}{\sigma^2}\right) \]  

where \( \sigma \) should be determined according to the distribution of candidate actions.

3 Self-Organizing Fuzzy Inference System

3.1 The proposed structure

In this section, the self-organizing fuzzy inference system that can perform Q-learning is proposed. The term, self-organizing, means that the consequent part of the fuzzy rule is automatically selected from the discrete action set based on the policy and on the Q value for each candidate action. The whole system is shown in Fig.1. This structure performs the fuzzy inference based on the fuzzy rule base that consists of several extended rules in (2). Inputs for the network are the states of the environment or plant. The network generate a final action and also approximate \( Q(s,a) \)-function. The network consists of 5 layers in total. Layer 1 to layer 3 achieves fuzzification in the input state, which resolves the continuous state representation problem in Q-learning. Layer 4 and layer 5
generate continuous action by fuzzy inference and generate the associated Q value for that inferred action by fuzzy inference mixed with the interpolation technique.

Layer 1
The function of the first layer is to receive the value of the state and transmit this value to the next layer. Each node in this layer corresponds to one input state variable \( s \in S \).

Layer 2
This layer consists of several linguistic labels for each input variable. Linguistic label \( \mu_{m}^{i}() \) in (2) has its own membership function denoted by \( \mu_{m}^{i}() \). Triangular and trapezoidal membership functions are used here. For these membership functions, \( \mu_{m}^{i}(s) \) is defined by four parameters: \( L, R, S_L, S_R \), where \( L \) and \( R \) correspond to the left center, right center, left spread and right spread respectively. For triangular membership function \( LC = RC \), that is, \( L = R \). The output value from layer 2 indicates a membership degree of the current state and can be calculated using the following equation.

\[
\mu_{m}^{i}(s) = \begin{cases} 
1 - \frac{|s - C_R|}{S_R} & s \in [C_R, C_R + S_R] \\
1 - \frac{|s - C_L|}{S_L} & s \in [C_L - S_L, C_L] \\
1 & s \in [C_L, C_R] \\
0 & \text{otherwise}
\end{cases}
\]  

(5)

Layer 3
One link from a node in layer 1 to a node in layer 3 corresponds to one antecedent part of FIS. That is, it represents the If part of each rule \( R_i \) in (2). The value of each node in layer 3 simply represents how much current state \( s \) belongs to rule \( R_i \) and can be obtained by fuzzy and operation. This value is the firing strength of rule \( i \) for state \( s \) and denoted as \( \alpha_{R_i} \). Then \( \alpha_{R_i} \) can be calculated as the following.

\[
\alpha_{R_i}(s) = \prod_{m=1}^{M} \mu_{m}^{i}(s)
\]  

(6)

where the fuzzy and operator is implemented by product operation.

Layer 4
Layer 4 corresponds to the then part of each fuzzy rule. Based on Q values and policy, one candidate action from the discrete action set is selected as a rule action \( a^i \) as in (7) and those selected rule actions from each rule are combined to generate the final action (i.e., defuzzification process)

\[
a^i = \pi(a^i_j, Q^i_j)
\]  

(7)

where \( j \) is the number of candidate actions and \( \pi(\cdot, \cdot) \) represents the policy used to select the action. In this paper, the \( \epsilon \)-greedy policy[7] is used. \( \epsilon \)-greedy policy selects an action of which Q value is the maximum with the probability of \((1-\epsilon)\). Besides the selection of the rule action, the Q value for the final action is calculated by (3) after the final action is calculated in layer 5.

Layer 5
In layer 5, the final action is obtained by using a weighted sum of each rule action with their firing strength as follows.

\[
a_f = \sum_{i=1}^{n} \frac{1}{\sum_{l=1}^{n} \alpha_{R_l}} \alpha_{R_i} a^i
\]  

(8)

Once the value of \( Q(R_i, a_f) \) is calculated for each rule using (3), the \( Q(s, a_f) \) value can be obtained by the following equation.

\[
Q(s, a_f) = \sum_{i=1}^{n} \frac{1}{\sum_{l=1}^{n} \alpha_{R_l}} \alpha_{R_i} Q(R_i, a_f)
\]  

(9)

Although the rule action \( a^i \) is determined by the \( \epsilon \)-greedy policy, the final action obtained by combining (defuzzifying) each rule action cannot be said a \( \epsilon \)-greedy action because the final action is generally not equal to the action that maximize the Q-value for the current state. Thus the policy conducted in the proposed FIS can be viewed as SARSA(\( \lambda \))[8].

3.2 Learning algorithm
Update is performed through the change in the Q value of each candidate action for each rule. To speed up the learning, eligibility is adopted and calculated as
\[ \frac{\partial Q(s, a)}{\partial Q_j} \text{. The eligibility is as the following equation.} \]

\[ e_{ij} = \begin{cases} 
\frac{\alpha_k}{\sum_{i=1}^{m} \alpha_k K(u_i, a_f)} & \text{if } R_i \text{ and } u^j_i \\
\lambda e_{ij} & \text{otherwise} 
\end{cases} \]

where \( \lambda \) is an eligibility rate which is used to weight each rule and rule action pair according to their proximity to the occurring time step from the current state.

The learning procedure is shown below:
1. Initialization of Q values of all candidate actions to zero in each rule
2. Perceive the current state \( s_{curr} \) and generate final action \( a_{f, curr} \) by (8).
3. Calculate \( Q(s_{curr}, a_{f, curr}) \) by (3) and calculate eligibility traces by (10).
4. Perform action \( a_{f, curr} \), receive reward \( r_{curr} \) and transition to the next state \( s_{next} \).
5. Generate final action for the next state, \( a_{f, next} \) and calculate \( Q(s_{next}, a_{f, next}) \) by (3) and calculate temporal error \( \delta \).
6. For each candidate action of every rule, update their Q values as follows:
   \[ Q_j = Q_j + \alpha \delta e_{ij} \] (12)
7. If trial is not ended, then let \( a_{f, curr} = a_{f, next} \) and go to step 2.

4 Simulation Study

4.1 Cart-pole balancing problem

The cart-pole balancing problem is concerned with how to balance an upright pole. The pole has only one degree of freedom and the primary control task is to keep the pole vertically balanced while keeping the cart within the rail track boundaries. Four state variables are used in describing this system and one variable represents the force applied to the cart. These are:

\( \theta \) : the angle of the pole from upright position (radian);
\( \dot{\theta} \) : the angular velocity of the pole (radian/seconds);
\( x \) : the position of the cart’s center (meters);
\( \dot{x} \) : the velocity of the cart (meters/seconds);
\( f \) : force applied to the cart (Newton)

The dynamics model used for the simulation is shown below:

\[ g \sin \theta + \cos \theta \left( -f - m_p \dot{x}^2 \sin \theta + \mu_c \text{sgn}(\dot{x}) \right) - \frac{\mu_c \dot{\theta}}{m_p} \]

\[ \dot{x} = f + m_p \left( \dot{\theta}^2 \sin \theta - \dot{\theta} \cos \theta \right) - \mu_c \text{sgn}(\dot{x}) \]

Each parameter value can be found in [9]. Simulation is done by the Euler method using a time step of 0.02s. Constraints on the variables are \(-12^\circ \leq \theta \leq 12^\circ\), \(-2.4m \leq x \leq 2.4m\) and \(-10N \leq f \leq 10N\).

It is assumed that the dynamics of the system is unknown to the FIS. The controller (FIS) can be informed of only the values of the state variables and the reward signal at each time step. The only available feedback that the proposed FIS can be acquired is failure signal when \( |p| > 12^\circ \) or \( |x| > 12.4m \). The reward signal given to the controller is as follows:

\[ r = \begin{cases} 
-1 & |p| > 12^\circ \text{ or } |x| > 12.4m \\
0 & \text{otherwise} 
\end{cases} \] (13)

Parameter values used for Q-learning are as follows: \( \gamma = 0.9, \alpha = 0.3 \) and \( \lambda = 0.7 \).

4.2 Simulation results

Figure 2 show learning curves when five candidate actions are used with zero initial condition. The curve consists of ten consecutive runs where a run consists of several trials. Each trial ends in two cases. First, if the pole has fallen (i.e., \( |p| > 12^\circ \)) or the cart position exceeds the limits of the rail track (i.e., \( |x| > 12.4m \)). Second, if the time step exceeds 10000. 10000 time step corresponds to 200
seconds in simulation time. When 5 discrete actions are used as candidate actions in consequent part of the extended fuzzy rule, the proposed FIS can learn to balance the pole after about 30-40 failures. In other words, the proposed FIS can solve the given task by using its experience without any experts' knowledge or any supervisor. Figure 3 to Fig. 5 show resulting trajectories of $\theta$, $x$ and $f$ for a learned FIS. After learning is occurred, most preferable consequent candidate action has the highest Q-value than any other candidate actions in the corresponding rule. Thus we can know what value of consequent part is good for a certain state or a certain antecedent part. In other words, we can obtain a local solution, which can be used to understand the characteristics of the system or the environment. Moreover, because this local solution is in the form of a fuzzy linguistic label, the information is physically explicit and easily understandable.

4.3 Exploration vs exploitation in Q-learning

Figure 6 to Fig. 8 show the resulting trajectories of $\theta$, $x$ and $f$ with 5 candidate actions with the same conditions. Although the same number of candidate actions and the same parameters are used, the obtained performance is somewhat different. These results can be explained by famous exploration vs exploitation problem in reinforcement learning. Since the agent cannot experience every state during finite times, it faces the problem: whether to explore unknown state/action to achieve better performance or to exploit known state/action based on already obtained experience. As shown in Fig. 6, the results seem worse than those in Fig. 3. But both of the resulting FIS is enough to fulfill the condition. It is a very difficult problem in reinforcement learning whether the agent performs exploration to obtain a better solution that might exist or performs exploitation. The exploration can result in better performance, but the more time is necessary. Tradeoff between the exploration and the exploitation scheme is important factor in reinforcement learning because the agent cannot live forever and cannot visit every possible state/action pair.

5 Conclusions

In this paper, the self-organizing fuzzy inference system that can perform Q-learning was proposed. By the extended rule and the interpolation technique, the proposed FIS behaves as a fuzzy inference system while approximating the Q(s,a) function. By incorporating the fuzzy inference system with Q-learning based on an extended fuzzy rule, without the model and knowledge about the environment, the proposed FIS can organize its fuzzy rule base automatically. As in this way, we can build a FIS(or an agent) that can learn to solve the given problem by itself through the interaction with the given problem, that is, through its experience like an animal. Also, by fuzzification, the continuous input state problem in Q-learning is easily resolved. By defuzzification, the continuous action problem is also resolved.
References


