Reinforcement–driven adaptation of control relations

Hans - Arno Jacobsen
Berkeley Initiative in Soft Computing
University of California
Berkeley, CA 94720-1776 USA
jacobsen@icsi.berkeley.edu

Joachim Weisbrod
Institut für Programmstrukturen und Datenorganisation
Universität Karlsruhe
Deutschland
weisbrod@ipd.info.uni-karlsruhe.de

Abstract

The conceptual framework of a hybrid control system architecture is briefly motivated. It employs neural and fuzzy techniques on a side–by–side basis using each one for the task it is best suited for. In this paper, our main interest is with the adaptation of the fuzzy control knowledge. The adaptation algorithm is based on reinforcement signals and directly optimizes the global fuzzy relation representing the complete knowledge base. The new approach is experimentally evaluated.

I Introduction

Neural networks are well suited for learning and adaptation tasks. In general, however, a neural network constitutes a black box. This means it is not possible to understand how a neural controller works. Furthermore, it is very hard to incorporate human a priori knowledge into a neural network. This is mainly due to the fact that the connectionist paradigm gains most of its power from a distributed knowledge representation.

Fuzzy knowledge based systems, on the other hand, exhibit complementary characteristics. The incorporation and interpretation of knowledge is straight forward, whereas learning and adaptation constitute major problems.

If in a supervised learning setting a *teacher* is used to adapt to a process it is only possible to come up to the teacher's performance. In order to manage complex problems, however, the availability of such a teacher cannot be guaranteed. The general goal is to control a process that has not been prior controlled. This problem has been addressed in the past by reinforcement learning paradigmes, e.g. [1, 13]. In reinforcement learning a *critic* evaluates each controller action or each sequence of controller actions by returning feedback to the controller on how well it meets the performance criteria. The adaptation process adapts the critic as well. The basic idea of our work is to implement the critic by a neural network

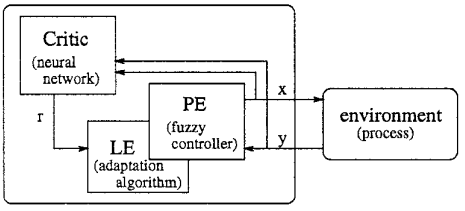

Figure 1: Hybrid control system architecture. (PE – performance element; LE – learning element).

because the adaptation of the critic is more complex than the adaptation of the controller, whereas an understanding of the critic's behavior is far less important. However, we do not discuss the realization of the critic here. Figure 1 depicts the conceptual architecture of the envisioned hybrid control system.

Since controller performance is far more important than interpretability we propose a two step process. (1) *adaptation*: a given fuzzy controller is tuned without regard to interpretability; (2) *interpretation*: the adapted fuzzy knowledge is analyzed in order to extract a set of fuzzy rules that, at least 'qualitatively', represents the given knowledge base. This paper addresses the first step, i.e. the adaptation of fuzzy control knowledge.

The key idea of our scheme is to adapt the knowledge representing fuzzy relation locally according to reinforcements generated by the critic.

Several approaches exist which combine neural and fuzzy techniques to so called *hybrid neural-fuzzy* systems, e.g. [2, 11, 7]. Most of these attempts combine both techniques successfully but relax fundamental concepts of either technique in order to better 'merge' both approaches. Our approach uses both techniques in complementary fashion, side-by-side, using each one for the task it is better suited for.

All of the above cited approaches adapt membership function parameters. Few approaches exist which adapt the control knowledge forming fuzzy relation.

0-7803-3225-3-6/96 $5.00 © 1996 IEEE

Early work by Mamdani and Procyk [12] on adaptive fuzzy systems updates fuzzy relations based on performance evaluations generated by incorporating a process model. Their approach updates the entire relation at each iteration which can be rather time consuming. We are looking at local modifications of the fuzzy relation based on a reinforcement signal generated by a separate adaptive component, the critic.

A supervised learning algorithm for off-line adaptation of fuzzy controllers is presented by Moore and Harris [10] who study supervised fuzzy relation adaptation, i.e. learning is based on a pre-determined set of state-action pairs. Our approach is independent of the availability of such a training set and operates on-line.

The paper is organized as follows. Section II reviews the essential theoretic concepts underlying fuzzy control. Section III presents the reinforcement-driven fuzzy relation adaptation algorithm. Experiments evaluating the adaptation schemes are presented in section IV. The paper concludes by pointing out future directions to pursue.

II Formal approach to fuzzy control

A fuzzy controller is a rule based system. It consists of a set of fuzzy rules which are applied to the actual controller input to infer the controller output. In the following, without loss of generality, we will consider only a *one input/one output* system. All results are easily extended to systems with many input and many output variables. We will therefore, consider here just two variables, the input variable x and the output variable y, with their respective universes of discourse $\mathcal{U}_x$ and $\mathcal{U}_y$. Additionally, we denote the generic elements of $\mathcal{U}_x$ and $\mathcal{U}_y$ by u and v, respectively. Furthermore, let $\tilde{A}_i$ and $\tilde{B}_i$ represent fuzzy sets on the universes of discourse $\mathcal{U}_x$ and $\mathcal{U}_y$, respectively ($i \in \{1, \ldots, n\}$). We will denote the fuzzy rule "IF x is $\tilde{A}_i$ THEN y is $\tilde{B}_i$" by $[\tilde{A}_i \Rightarrow \tilde{B}_i]$.

The general framework for handling a fuzzy rule base $[\tilde{A}_i \Rightarrow \tilde{B}_i]$ is to transform each rule into a fuzzy relation $\tilde{R}_i = \mathit{transform}(\tilde{A}_i, \tilde{B}_i)$ on $\mathcal{U}_x \times \mathcal{U}_y$, to aggregate these implication relations to $\tilde{R} = \mathit{aggregate}(\tilde{R}_i)$, and to apply the resulting so called meta rule $\tilde{R}$ by using max–min composition. That is, given the actual input $\tilde{A}'$ on $\mathcal{U}_x$, the result $\tilde{B}'$ on $\mathcal{U}_y$ of applying the fuzzy rule base $[\tilde{A}_i \Rightarrow \tilde{B}_i]$ is determined by computing

$$\tilde{B}' = \tilde{A}' \circ \tilde{R}, \tag{1}$$

$$\mu_{B'}(v) = \max_{u \in \mathcal{U}_x} \min \{\mu_{A'}(u), \mu_R(u, v)\}. \tag{2}$$

This general framework has been derived by generalizing mechanisms for crisp sets. Obviously, with the choice of *transform*() and *aggregate*() there are some important design decisions left [8]. In the past, there have been lots of efforts to support these design decisions, most of them by conducting and evaluating experiments, others by theoretical considerations [14, 15]. In our work we use two complementary inference mechanisms, the well known possibilistic approach [3, 4, 5] and a new method called σ–reasoning that has been developed recently [16]. This new approach builds the theoretical justification for the well known MAMDANI approach to fuzzy control [9], where

$$\tilde{R}_i = \mathit{transform}(\tilde{A}_i, \tilde{B}_i) = \tilde{A}_i \cap \tilde{B}_i, \text{ and} \tag{3}$$

$$\tilde{R} = \mathit{aggregate}(\tilde{R}_i) = \bigcup \tilde{R}_i = \bigcup_i (\tilde{A}_i \cap \tilde{B}_i). \tag{4}$$

Using eq. (2), we get

$$\begin{aligned}
&\mu_{B'}(v) \\
&= \max_{u \in \mathcal{U}_x} \min \{\mu_{A'}(u), \mu_R(u, v)\} \\
&= \max_{u \in \mathcal{U}_x} \min \{\mu_{A'}(u), \max_i \min \{\mu_{A_i}(u), \mu_{B_i}(v)\}\} \\
&= \max_{u \in \mathcal{U}_x} \max_i \min \{\mu_{A'}(u), \min \{\mu_{A_i}(u), \mu_{B_i}(v)\}\} \\
&= \max_i \max_{u \in \mathcal{U}_x} \min \{\min \{\mu_{A'}(u), \mu_{A_i}(u)\}, \mu_{B_i}(v)\} \\
&= \max_i \{\min \{\max_{u \in \mathcal{U}_x} \min \{\mu_{A'}(u), \mu_{A_i}(u)\}, \mu_{B_i}(v)\}\},
\end{aligned}$$

with this last expression being exactly MAMDANI's way of implementing fuzzy inference. That is, MAMDANI's approach is just a special case of the general fuzzy inference by means of fuzzy relations (eq. (1)).

Nevertheless, during the adaptation step there is no need to consider the way the meta rule is constructed. We may just take the fuzzy relation $\tilde{R}$ for granted and adapt it according to the critic's reinforcements. This is due to the fact, that *any* meta rule is processed using max–min composition according to eq. (1).

But in the second step, i.e. the knowledge interpretation phase, this information will become essential because we will have to look for the inverse operations of *transform* and *aggregate* in order to reduce the resulting fuzzy relation $\tilde{R}'$ into a set of simple fuzzy rules.

III Adaptation of fuzzy relations

The adaptation process can be divided into two stages which are repeated in a cyclic manner: an *action selection* stage and a *knowledge update* stage. The former selects a control action to be transmitted to the process, the latter adapts the fuzzy relation underlying the knowledge representation. Figure 2 depicts the adaptation process in a more detailed manner making reference to figure 1.

```
while   (learning has not converged)
        select an initial state x^t at random
        while   (not out of control)
                y^t ← Performance_Element (x^t)
                apply y^t to process
                observe new state x^{t+1}
                r^t ← Critic^H (x^t, x^{t+1}, y^t)
                Learning_Element (r^t)
        end
end
```

Figure 2: Loop executed by the control system. The different functions correspond to the components shown in the previous figure.

Action selection stage After inferring the output fuzzy set $\mu_{B'}(v)$ from the meta rule $\tilde{R}$ (cf. section II) a crisp control action has to be derived and emitted to the process. This step is known as *defuzzification*. Several operators have been defined for this purpose (see [8] for an overview). For example the *maximum defuzzification* operation selects the control action with the maximum membership value among all maxima from the output fuzzy set.

Any deterministic defuzzification scheme, however, does not allow the control agent to *experiment* with the available control actions since the same action is selected over and over again for the same input situation. Hence, the agent cannot experience better or worse situations as a result of applying different actions in the same state and can therefore not adapt its knowledge. If the represented knowledge is sufficient for meeting the performance goals and there is no need for improvements a deterministic defuzzification is all that is needed. But if the available control knowledge is incomplete, inconsistent or even partly wrong the agent needs mechanisms to acquire, fine-tune, or reconfirm it. This is accomplished by the *trial and error* strategy undertaken in reinforcement learning.

The objective is that the agent learns to reliably judge the expected outcome of taking a specific action in a given state. This information will then be reflected in the output fuzzy set associated with the current input situation.

To stimulate this kind of explorative behavior we introduce a *randomized defuzzification* scheme. It is similar to the maximum defuzzification operation discussed above. Control actions are chosen randomly from the set of possible control actions according to their degrees of membership in the output fuzzy set. Control actions with higher degrees of membership have a greater chance of being selected as output as ones with lower degree of membership.

Knowledge update stage Given the crisp input x and the crisp output y we know exactly *how* and *why* the selected control action was chosen from the set of possible control actions. Observing the effect of the output on the process it is now possible to reinforce the selection of the same control action or to suppress its selection in future situations. This is achieved by directly modifying the underlying knowledge relation. Clearly, the objective is to reinforce good actions and to suppress bad actions. Several different *reinforcement schemes* for updating the relation have been considered:

Point-wise update:

$$R(x,y) = \min\{1, \max\{0, R(x,y)\} + \alpha\,\kappa\}\},$$

with $0 \leq \alpha \leq 1$ a learning rate and κ the reinforcement signal ($\kappa > 0$ for rewards and $\kappa < 0$ for punishments). The min, max operations serve to enforce the boundary conditions. From now on we denote $\mu_R(u,v)$ by $\tilde{R}(u,v)$. The above update operates on a single point in the relation, it is therefore very precise. The fuzzy relation, however, is a topological representation of the control knowledge. It has localized meaning. Updating a whole region centered around the point specified by the state–action pair has therefore a generalizing effect on the learning process. Since an entire neighborhood profits from a single update the number of learning cycles decreases for most learning situations.

Neighborhood incorporating update:
$\forall u_i \in \mathcal{U}_x$ and $\forall v_j \in \mathcal{U}_y$

$$R^t(u_i,v_j) = \min\{1, \max\{0, R^t(u_i,v_j) + \alpha\,\kappa\,R^t(u_i,v_j)\,e^{(-(d^{x,y}_{u_i,v_j})^2)/\sigma_t^2}\}\}$$

with α and κ as above, σ_t an adaptive variance[1] and d a distance measure. The adaptation is here additionally a function of time. With increasing time (number of iterations) the updated neighborhood decreases, finally converging to the center point.

Fuzzy set oriented update:
$\forall u_i \in \mathcal{U}_x$ and $\forall v_j \in \mathcal{U}_y$

$$R(u_i,v_j) = \max\{\gamma\,R(u_i,v_j), \min\{\mu_{I_{w_2}}(u_i), \mu_{O_{w_1}}(v_j)\}\},$$

with $0 < \gamma \leq 1$ a discount factor and w_1, w_2 parameters specifying the fuzzy set I on the input domain and the fuzzy set O on the output domain centered around the crisp state-action pair (x,y). The discount factor

[1]The adaptive variance: $\sigma^t = \sigma_{initial}\left(\frac{\sigma_{final}}{\sigma_{initial}}\right)^{t/t_{max}}$.

γ was introduced to discount the relation in situations where the process response patterns change. A similar operator was introduced in [10].

IV Experiments

In this section the adaptation scheme is applied to function approximation tasks. In these tasks a controller is adapted such that it approximates a given function. The static character of these tasks allows us to study the fuzzy relation adaptation process isolated from the adaptive critic since the internal reinforcement signal can be easily generated by comparing the real value of the function to be approximated with the through fuzzy inference computed value. If this difference is sufficiently small the controller performance is judged as *good* and otherwise as *bad*. We observe the adaptation process by means of the mean square error between the real function and the, through the fuzzy controller approximated function. The below presented, error curves show the development of the error over the number of iterations, averaged over the number of repetitions. One iteration involves an entire learning cycle (i.e. one execution of the inner loop of the control algorithm, cf. figure 2). It should be noted that the error measure depends on the chosen discretization and the range of function values. The error should therefore be interpreted in a relative manner.

IV.1 Adaptation with and without a priori knowledge

Figure 3 shows error curves for the approximation of $f(x) = x, x \in [0, 10]$. The meta relation was initialized with a set of rules describing f in straight forward manner (uniform partitioning of domain, 4 rules). The rules were randomly distorted. The positive effect of initializing the meta relation with prior knowledge (i.e. known rules) can be seen by comparing the error with the error in figure 5. Both figures make reference to approximating the same function.

Figures 4 compares the behavior of the above developed updates. It follows that the fuzzy set oriented update performs best with regard to speed of learning and smoothness of adaptation. The neighborhood incorporating update lays between the fuzzy set oriented update and the point-wise update. These results are conform with other experiments completed in [6].

Figure 5 shows the influence of the parameter γ. Due to the static character of the task only slight effects are observable. The best performance is achieved with γ close to 1. The second graph in figure 3 shows the influence of the parameter r. As the width of the fuzzy set, defining the update operator increases, the

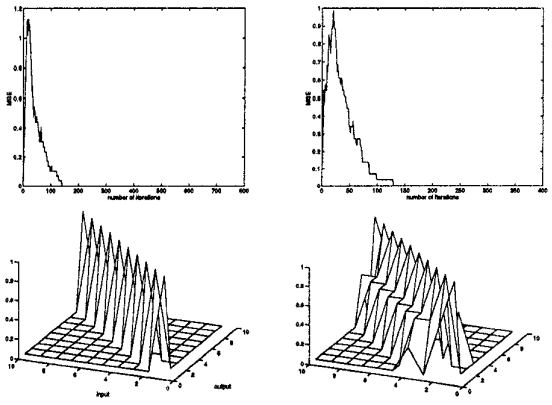

Figure 3: Error for point-wise update ($\alpha = 1$ and fuzzy set oriented update ($r = 2, \gamma = 1$). Learning with a priori knowledge. Second row: Adapted fuzzy relations for the approximation of $f_1(x) = x, x \in [0, 10]$.

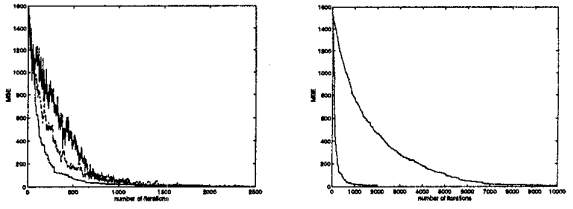

Figure 4: Comparison of the error measures for the approximation of a complex partly linear function. Fuzzy set oriented update (both figures, solid line, $\gamma = 1$, $r = 5$), two dimensional update fixed variance (left figure, dashed line, $\sigma = 0.4$), two dimensional update time-decaying variance (left figure, solid line, $\sigma_i = 1.5, \sigma_f = 0.2$) and point-wise update (right figure, solid upper line, $\alpha = 0.7$).

speed and accuracy of learning augments. Intuitively this can be interpreted as: if one updates more in a single step, less overall updates have to be effected.

IV.2 Adaptation in changing environment

To investigate the *on-line adaptive* capabilities of the proposed algorithm we defined a learning task that simulates processes which exhibit sudden changes in their response patters. This is achieved by altering the function underlying the approximation task in the above experiments. Figures 6 and 7 show the results for different parameters. (cf. [10] for similar experiment in a supervised learning setting.)

V Conclusions

We discuss a reinforcement-driven fuzzy relation adaptation algorithm which is part of a complexer hybrid control system architecture. The adaptation

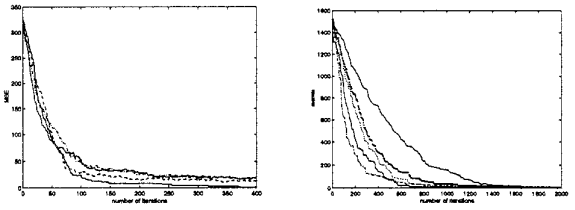

Figure 5: First figure: Influence of the discount factor γ ($\gamma = 1$ (solid line), $\gamma = 0.8$ (dash dot line), $\gamma = 0.5$ (dashed line), $\gamma = 0.3$ (solid line)). Function f from above. Second figure: Study of parameter r (half-width of fuzzy set) ($r = 5, 1.25, 1.0, 0.75, 0.5$ from left to right in the figure; $\gamma = 1$). Complex partly linear function. Learning without a priori knowledge.

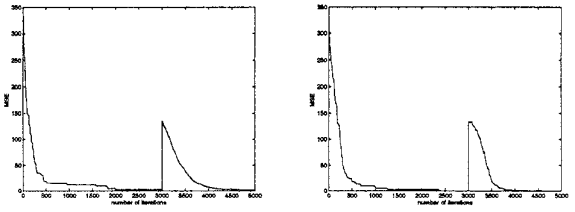

Figure 6: Approximation of $g_1(x) = x$ changing after $T = 3000$ to $g_2(x) = x^2$ for $x \in [0, 1]$. (Fuzzy set oriented updates, $r = 2, r = 4$, respectively; $\gamma = 0.93$ in both runs).

algorithm modifies the meta control relation according to reinforcements generated by a separate adaptive critic component. The adaptation algorithm was experimentally evaluated and proved excellent performance. We demonstrated how the incorporation of a priori knowledge improved its performance. Experiments were set up to evaluate the effects changing environments have on the behavior of the adaptation. The overall conclusion is that fuzzy relations can be adapted locally provided reliable reinforcements on the controller performance are available.

The fuzzy relation adaptation scheme has so far been evaluated in isolation from the discussed hybrid control system architecture. In a further step the adaptation algorithm will be combined with a neural adaptive critic. The incorporation of a critic will then permit us to test the control system architecture on real control problems and oppose it to alternate approaches. Apart from these extensions are we currently working on ways to extract fuzzy rules from the adapted meta control relation.

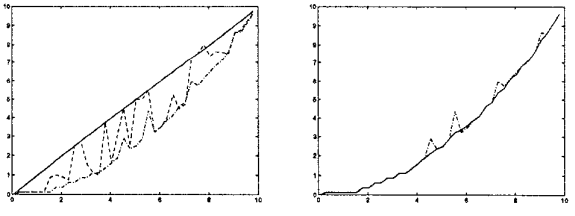

Figure 7: Approximation behavior during the change in response characteristic in process occurred. Left figure : solid line at $T = 3000$; dashed line after $T = 3500$; dashed dotted line after $T = 4000$. Right figure : dash dotted line after $T = 4500$; solid line after $T = 5000$.

Acknowledgments

The authors would like to thank Ben Gomes for proof reading the final manuscript. The first author would like to express his sincere thanks to Professor Zadeh for his help and constructive criticism.

References

[1] A. Barto, R. Sutton, and C. Anderson. Neurolike adaptive elements that can solve difficult learning control problems. *IEEE Trans. Systems, Man & Cybernetics*, 1983.

[2] H. R. Berenji and P. Khedkar. Learning and tuning fuzzy logic controllers through reinforcements. *IEEE Trans. Neural Networks*, 3:724–740, September 1992.

[3] D. Dubois and H. Prade. *Possibility theory: an approach to computerized processing of uncertainty.* Plenum Press, New York, 1988.

[4] D. Dubois and H. Prade. Fuzzy sets in approximate reasoning, part 1: Inference with possibility distributions. *Fuzzy Sets and Systems*, 40:143–201, 1991.

[5] D. Dubois and H. Prade. Fuzzy sets in automated reasoning: problems and methods. In F. Esteva and P. Garćia, editors, *Tecnologías y Lógica Fuzzy (FUZZY'94)*, Blanes, Spain, 1994.

[6] H.-A. Jacobsen. Adaptive fuzzy systems. Master's thesis, Universität Karlsruhe (TH), August 1995.

[7] R. Jang. ANFIS: Adaptive-network-based fuzzy inference systems. *IEEE Trans. Systems, Man & Cybernetics*, 1991. submitted.

[8] C. C. Lee. Fuzzy logic in control systems: Fuzzy logic controller (part I&II). *IEEE Transactions on Systems, Man & Cybernetics*, 20(2):404–435, 1990.

[9] E. H. Mamdani and S. Assilian. An experiment in linguistic synthesis with a fuzzy logic controller. *International Journal of Man–Machine Studies*, 7, 1975.

[10] C. G. Moore and C. J. Harris. *Advances in Intelligent Control*, chapter Indirect Adaptive Fuzzy Control. Burgess Science Press, 1994.

[11] D. Nauck and R. Kruse. A neural fuzzy controller learning by fuzzy error propagation. In *NAFIPS92*, pages 388–397, Puerto Vallarta, December 1992.

[12] T. J. Procyk and E. H. Mamdani. A linguistic self-organising process controller. *Automatica*, 15:15–30, 1979.

[13] R. Sutton. Learning to predict by the method of temporal differences. *Machine Learning*, 1988.

[14] I. B. Turksen and Y. Tian. Combination of rules or their consequences in fuzzy expert systems. *Fuzzy Sets and Systems*, 58:3–40, 1993.

[15] J. Weisbrod. On fuzzy implication relations. *Fuzzy Sets and Systems*, 67:211–219, 1994.

[16] J. Weisbrod. Fuzzy control revisited — why is it working? In P. P. Wang, editor, *Advances in Fuzzy Theory and Technology, Vol. III*, pages 219–244. Bookwrights, Durham (NC), 1995.