Resolving the Conflict Between Competitive and Cooperative Behavior in Michigan-Type Fuzzy Classifier Systems

Peter Haslinger and Ulrich Bodenhofer
Software Competence Center Hagenberg
A-4232 Hagenberg, Austria
{peter.haslinger|ulrich.bodenhofer}@scch.at

Abstract

A vital problem for fuzzy classifier systems of the Michigan type is the conflict of competition and cooperation of rules. Whereas the classical approach of a classifier system circumvents this complicacy by the total lack of collaboration of classifiers, the fuzzification approach has to deal with it. This paper proposes a solution to this dilemma by introducing a special encoding of the classifiers and by performing a selection algorithm on a sub-group of firing rules.

Keywords: fuzzy classifier systems, fuzzy systems, genetic algorithms, learning classifier systems.

1 Introduction

Classifier systems have potential as learning paradigms, but due to the limitation of the classifier syntax they are seldom applied to continuously varying variables. Fuzzy systems on the other hand are a commonly used tool to deal with such continuously represented data. On account of the similar structure of the if-then rules it seems obvious that a combination of these two systems might prove to be an interesting approach.

The first one to introduce such a fuzzy classifier system (FCS) of the Michigan type was M. Valenzuela-Rendón [8, 9], which is more or less a straightforward fuzzification of a Holland Classifier System [5]. Instead of a discrete-valued classifier system, linguistic rules and fuzzy sets are introduced to achieve an easily interpretable and understandable machine learning system. It has a production system similar to common classifier systems with the difference of fuzzification of the input variables and a defuzzification applied to outgoing messages. Both steps work with the concept of minimal messages. Of special interest is the coding of the rules, which are represented as binary strings that encode the membership function of the fuzzy sets involved.

Another approach in this area—a continuation of the FCS—is presented in [7], where the whole knowledge base is modified. A drawback of this system is the loss of interpretability of the resulting rules and rule base due to the rearranging of the membership function.

A. Bonarini presented in [1, 2, 3] the ELF method (evolutionary learning of fuzzy rules) and applied it to a robot learning environment. Here a first attempt to solve the cooperation/competition problem was introduced. The rule base is split up into sub-populations which cooperate to generate the output. Within a sub-population the rules compete to propose their output. A central concept of ELF is to keep the rule base as small as possible. So for the case that no or too few rules for a given state exist, the so-called cover detector operation is accomplished, which generates new rules.

A recent approach [4] deals with the issue of rule generalization within the field of fuzzy classifier systems. A fuzzification of the XCS [10] (a classifier system operating on classifiers with fitness based on accuracy) is accomplished.

2 An Improved Fuzzy Classifier System

Fuzzy rules interact—the rules “collaborate” to generate a desired output. On the other hand, classifier systems need to evaluate the contribution of a *single member* or a *small fraction* of classifiers to the performance of the system. Classifier systems and fuzzy systems, therefore, contradict in this point. There is a big difference in the selection scheme of a classifier system and the fuzzy classifier system. Classifier Systems mostly select only one single rule and send its action to the environment, they receive a payoff which is directly applied to the selected classifier. The FCS performs a sum-product inference on all firing classifiers, sending the generated action to the environment and receiving a payoff that must be distributed to all selected classifiers. However, by selecting all classifiers—good and bad ones—the outgoing messages to the environment become distorted and the system is unable to determine which rules are good and which are not.

We will introduce a modified version of the FCS to overcome some drawbacks. This new system can be seen as a fusion of Valenzuela-Rendón's FCS [8] and Bonarini's ELF Algorithm [2]. The structure of the fuzzy classifier system (similarly to Michigan classifier system) is combined with the idea of competition and cooperation and the cover-detector operation.

As a first important difference, the modified variant uses fuzzy partitions consisting of trapezoidal membership functions instead of bell-shaped ones as in [8, 7]. This adaptation grants already two minor improvements: a)We expect an easier interpretability of the resulting rule base, b) fewer classifiers are activated during one learning step. The latter is due to the given maximum of two active membership functions for each variable. Whereas in the bell-shaped case all membership functions have a truth value above zero and are active all the time.

2.1 The Production System

In the suggested classifier system the structure of the classifiers differs strongly from Valenzuela-Rendón's fuzzy classifier system: Not all fuzzy sets are represented in the binary condition and action string. Instead, only one fuzzy set per variable is addressed, represented by an integer value. The position in the condition string or integer list, respectively, corresponds to the number of the input or internal variable. Within the action, the number of the internal or output variable is represented by the action tag. Consider two input variables x_0 and x_1 and one output variable x_2 which all have a partition with three component sets ("low"$= 0$,"medium"$= 1$, and "high"$= 2$). An example of such a classifier would look as follows:

$$0:1,1:2/2:0$$

The corresponding decoded rule would be the following:

> "***if*** x_0 *is medium* ***and*** x_1 *is high* ***then*** x_2 *should be low*"

To be compatible with the coding of the rules, the incoming messages are structured like the action part of the classifiers:

They consist of a tag which names the variable the message corresponds to and an integer value which represents the fuzzy set. For example, the message $2:1 \rightarrow 0.5$ means "*the value of variable 2 is medium with a degree of 0.5*". Note that this kind of message is already a minimal message, so the evaluation of the matching scheme and payoff distribution is simplified. See Figure 1 for an example of the creation of fuzzy messages.

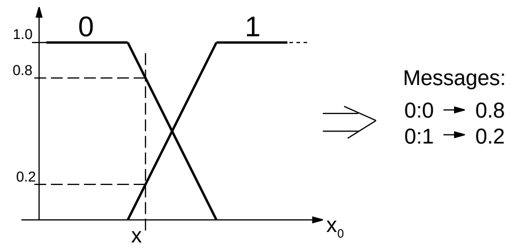

Figure 1: The creation of fuzzy messages in the improved FCS

The matching of the conditions is done in the following way: The tag of the message is compared with the tags of the conditions. Then the value of the condition with the same tag is checked with the value from the message. The condition matches the message if the message value and the condition value correspond to the same fuzzy set. The activity level $a \in [0,1]$ of the condition is then set to the same degree as that of the message. This is done for all messages and all classifiers.

After the matching procedure the activity level of the classifier is set to the minimum of all matching degrees of the conditions (i.e. conjunction by means of the minimum t-norm). This activity level a of the classifier is then used as the degree of satisfaction for the output message. Figure 2 gives an example of the matching procedure.

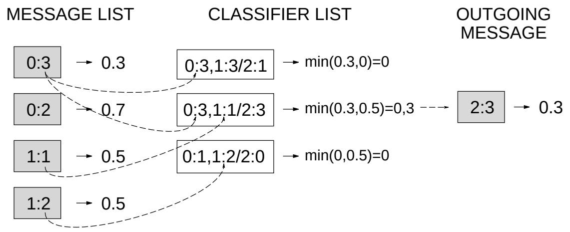

Figure 2: The matching procedure of the modified fuzzy classifier system

For the case that no matching classifier is found, the cover-detector operation is performed to create a new classifier with a condition that matches the messages and a randomly chosen action. This new classifier is added to the classifier list, either as additional one or as replacement for a weak classifier, similarly to ELF [1, 2, 3].

Now let us turn to the most important modification—the selection scheme:

In Valenzuela-Rendón's FCS, all matching classifiers are allowed to fire and to send their action messages to the output unit. The modified version uses another selection scheme. If there are activated classifiers which have the same condition and the same action tag (but a different action value), a selection scheme is performed (e.g. roulette wheel) according to the classifiers strength. The winning classifier is then allowed to post its message to the message list. With this kind of selection, we have found a compromise between competition and cooperation of fuzzy rules with some similarities to the ELF method (see [1, 2, 3]). Fuzzy classifiers with the same condition and action tag compete to post a message, and all others "work together" to create the output of the classifier system by a sum-prod inference.

2.2 Credit Assignment

The payoff distribution is solved in a very simple way. All classifiers that did post a message receive a payoff according to their activity level, that means the classifier R_i (i index of rules in rule base) receives a payoff $P_{i,t}$ at time step t:

$$P_{i,t} = P_t \cdot a_{i,t},$$

where P_t is the payoff from the environment at time step t, and $a_{i,t}$ is the activity level of the classifier R_i at time step t.

An active classifier has to pay a bid to be allowed to post a message. For simplicity the bid is set to the activity level of the classifier. With this bid and the above payoff we can calculate the new strength $u_{i,t+1}$ of the classifier R_i:

$$u_{i,t+1} = u_{i,t} - a_{i,t} + P_t \cdot a_{i,t}$$

Where $u_{i,t}$ is the current strength of the classifier. This strength adjustment works properly, if the payoff $P_t \in [0, p_{max}]$ with maximal payoff $p_{max} > 1$ to allow an increase of the classifiers strength.

2.3 Rule Discovery

The rule discovery system is similar to common classifier systems. Classifiers with a higher strength are selected more often than weaker ones. The reproduction of the selected classifiers is done by crossover and mutation. The crossover algorithm works on the condition part only and is applied in case that there is more than one condition per classifier.

3 Experimental Results

The modified FCS is employed in two different application tasks. At first its behaviour as function approximator is investigated, and then a more sophisticated problem — the well known inverted pendulum — is used as test environment for the FCS.

3.1 Function Approximation

To achieve comparable results, the same test functions as proposed in [8] are applied to the modified FCS. It should learn to approximate the following functions (straight line and parabola):

$$x_1 = f_1(x_0) = x_0 \quad (1)$$

$$x_1 = f_2(x_0) = 4(x_0 - 0.5)^2 \quad (2)$$

The payoff P_t is determined as in [8] by $P_t = P_0 \cdot (1 - |f_j(x_0) - y|)$ with a constant P_0 and $j \in \{1,2\}$ for (1) and (2) and y is the output value of the FCS. During these tests $P_0 = 1.6$ was chosen. After learning a given number of iterations, an evaluation cycle is performed. To eliminate the random effect of the roulette-wheel selection (which is mainly necessary for exploration during the learning phase) two evaluation scenarios have been considered. The first

Table 1: Classifier's strengths after learning f_1

classifier	strength	classifier	strength
0:0/1:0	100.00	0:1/1:0	13.56
0:1/1:1	100.00	0:1/1:2	13.48
0:2/1:2	100.00	0:2/1:0	1.06
0:2/1:1	15.73	0:0/1:2	1.05
0:0/1:1	13.79		

one allows just the the strongest rules to post their messages. This leads to a linear interpolation of the function with nodes (sampling points) at the positions where the fuzzy sets have their maximum membership degree. The other chosen evaluation allows all rules to post their messages, but with a weighted activity level according to their strength. In this case the activity level is adjusted by the following formula: $a_{i,out} = (\frac{u_i}{u_{average}})^2 \cdot a_{i,t}$ with $u_{average} = \sum_{i=0}^{n} u_i$ and n is the number of classifiers in the rule base. We will call this evaluation "strength adjusted". To allow a comparison with the results presented in [8], we will calculate the absolute error by $\frac{1}{x_f - x_0} \int_{x_0}^{x_f} |f_{out} - y| dx$, with $[x_0, x_f]$ interval of x, f_{out} output of classifier system, and the actual value y.

For the function f_1 of (1), a partition of 3 fuzzy sets is used and all possible rules are stored in the classifier list. After 2000 iterations an evaluation was performed. Using only the strongest classifiers a perfect interpolation is achieved with an absolute error of 0.0%. The "strength adjusted" evaluation had an error of 0.62%, compared to Valenzuela-Rendón's FCS which had an absolute error of 1.72% after 64000 cycles. Table 1 shows the strength of the classifiers after the learning phase. The important rules have all reached the maximum strength of 100, whereas the others have a significant lower strength.

The function f_2 of (2) was approximated by the modified FCS with a partitioning of 5 fuzzy sets. Starting with 18 randomly chosen classifiers, a genetic operation (mutation with probability $p_m = 0.08$) is applied every 4000 iterations. In total 14000 iterations are performed. The resulting absolute errors are 4.18% for the evaluation with the strongest classifiers and 4.61% for the "adjusted strength" method. Valenzuela-Rendón's FCS (in [8]) showed a slightly better result with an absolute error of 3.76% after 54000 cycles. Using more than 5 fuzzy sets would decrease the absolute error below Valenzuela-Rendón's FCS, as we would have a more accurate interpolation due to the increased number of interpolation nodes. For example tests with 7 sets led to an absolute error of 1.85%. Figure 3 shows the interpolation result with 5 fuzzy sets and Table 2 shows the resulting classifiers.

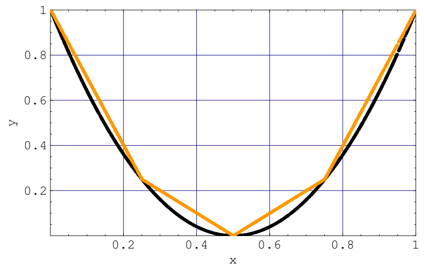

Figure 3: Interpolation result of the evaluation of f_2 (with 5 fuzzy sets)

Table 2: Classifier's strengths after learning f_2

classifier	strength	classifier	strength
0:0/1:4	152.23	0:2/1:1	59.64
0:0/1:3	43.14	0:2/1:3	1.89
0:0/1:2	16.30	0:2/1:4	1.06
0:1/1:1	201.06	0:3/1:1	131.81
0:1/1:0	121.57	0:3/1:0	71.37
0:1/1:2	50.97	0:3/1:2	43.71
0:1/1:3	7.22	0:4/1:4	172.70
0:1/1:4	1.43	0:4/1:2	16.73
0:2/1:0	370.86	0:4/1:1	6.16

3.2 Inverted Pendulum

The FCS is also applied to an online learning task. The inverted pendulum from [6] is chosen, where the fuzzy partition and a perfect rule set are known. The modified FCS should now try to balance the pendulum of length $l = 1$ and mass $m = 1$ by determining a moment M. A friction coefficient of $k = 0.01$ and a random noise moment of variance $\sigma = 1$ is applied to the pendulum. The chosen time step $\Delta t = 0.01$. The payoff of the environment for each time step is calculated by $P_t = 1 + c \cdot \mathrm{sgn}(\varphi_i) \cdot (\varphi_{i-1} - \varphi_i)$, with the constant $c = 1000$. The fuzzy classifier system receives positive payoff from the environment if the messages from the FCS cause a moment that moves the pole nearer to the vertical position. The classifier consists of two input variables (angle ϕ and angle velocity $\dot{\phi}$) and one output variable (moment M).

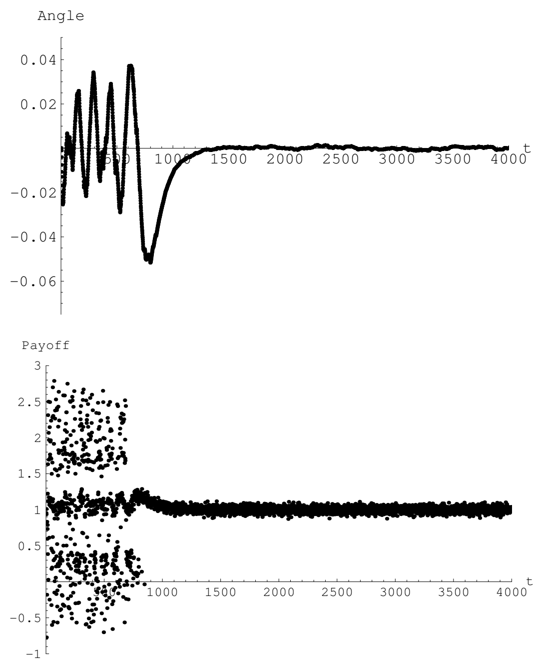

Figure 4: Results of the FCS eliminating 4 bad rules

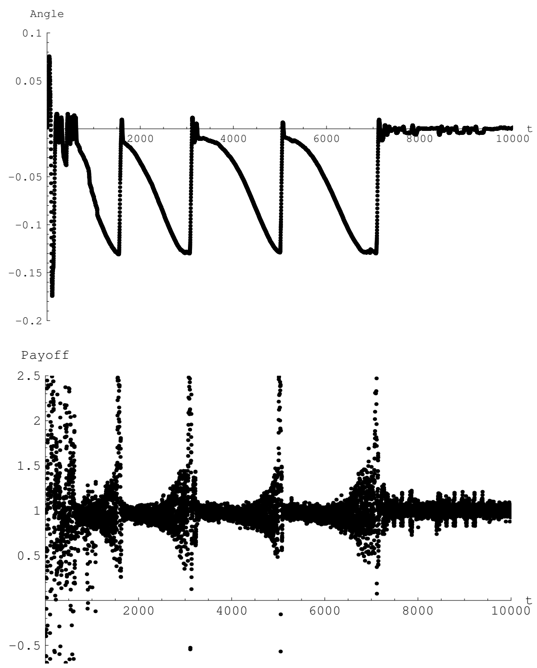

Figure 5: Results of the FCS with a starting random rule set of size 50

Table 3: Classifier's strengths of the "perfect set" and 4 bad rules

classifier	strength	classifier	strength
0:3,1:3/2:3	195.63	0:4,1:4/2:2	7.01
0:3,1:4/2:2	31.00	0:2,1:2/2:4	6.53
0:2,1:3/2:4	20.65	0:3,1:1/2:5	5.78
0:3,1:2/2:4	19.84	0:3,1:3/2:1	2.30
0:4,1:3/2:2	16.20	0:4,1:4/2:4	2.15
0:3,1:0/2:6	10.01	0:3,1:3/2:5	1.02
0:4,1:0/2:4	10.01	0:3,1:2/2:0	1.00

Two tests are performed. At first a complete perfect fitting rule set (as in [6]) with four interfering bad rules is taken as starting population for the system. For the second test a random start population in the classifier list is used.

In the first case (working with a perfect rule set and four bad rules) 4000 iterations without any genetic algorithm are performed to see if the system is capable to eliminate the negative rules. The results are presented in Figure 4. After around 1000 iterations/time steps and a lot of oscillations a stable state is achieved.

The resulting rule set is shown in Table 3 (only rules at least activated once are presented) and the bad rules are the last four with the lowest strength.

In the last test fifty randomly created classifiers are used to learn and control the inverted pendulum. 10000 iterations are performed, crossover (with probability $p_C = 0.08$) and mutation ($p_M = 0.02$) was applied every 3000 time steps. Results are shown in Figure 5. A cyclic behavior can be viewed. There seems to be a negatively influencing rule moving the pendulum into the negative direction, but after a high negative angle is achieved the rules 0:1,1:1/2:6 and 0:2,1:2/2:5 are activated which move the pole back to the center position. This happens 4 times until the "bad" rule's strength is reduced enough and a stable state is achieved. The resulting rules after the 10000 iterations can be viewed in Table 4.

It has to be mentioned that if a random starting population contains mainly false rules—as other tests have shown— the FCS was not always able to find a proper rule base.

4 Conclusion

This modified approach of the FCS deals with the competition/cooperation dilemma by performing a selection scheme to activated classifiers which have the same condition and the same tag on the action side, whereas the remaining classifiers cooperate via the inference system to generate the output.

Table 4: Classifier's strengths of the random rule base after 10000 iterations

classifier	strength	classifier	strength
0:2,1:5/2:4	178.80	0:3,1:3/2:5	11.80
0:3,1:3/2:3	157.09	0:4,1:2/2:5	10.79
0:2,1:4/2:4	117.12	0:4,1:0/2:4	10.01
0:3,1:5/2:3	53.01	0:1,1:1/2:6	8.27
0:3,1:5/2:2	45.55	0:2,1:1/2:6	7.97
0:4,1:4/2:2	36.98	0:4,1:4/2:0	7.90
0:3,1:5/2:1	32.76	0:1,1:2/2:2	7.40
0:4,1:3/2:2	23.61	0:2,1:2/2:4	5.70
0:2,1:4/2:2	21.69	0:2,1:1/2:4	4.73
0:1,1:4/2:5	19.64	0:4,1:5/2:3	3.75
0:4,1:1/2:3	16.54	0:2,1:2/2:1	1.62
0:1,1:3/2:5	14.58	0:2,1:0/2:5	1.00
0:5,1:5/2:0	14.40	0:2,1:0/2:6	1.00
0:1,1:4/2:3	13.16	0:4,1:4/2:6	1.00
0:1,1:3/2:6	12.69	0:3,1:0/2:1	1.00
0:2,1:2/2:5	12.28	0:3,1:0/2:2	1.00
0:1,1:2/2:6	11.82	0:2,1:0/2:1	1.00

As shown in Section 3, the modified FCS is capable to perform a refinement to a given rule base. Hence, it can be seen as an optimization method for partly incorrect rule bases. The starting population has a crucial influence to the performing of the FCS. If a fitting payoff scheme is found and at least some basic information of the environment is known, the FCS seems to be capable to improve the set of classifiers.

For future approaches it should be considered to completely remove the activity level from the payoff distribution scheme. A given rule which is the perfect description of a given environmental state, but fires with a low activity level will receive only a small reward and its strength will keep low. The genetic algorithm will most probably remove such classifiers from the rule base, and essential information might be lost. One possible solution for this problem is presented in [4]. It has to be investigated in further detail how this approach can be introduced into the concept of the modified FCS.

Acknowledgments

This work has been supported by the Austrian Government, the State of Upper Austria, and the Johannes Kepler University Linz in the framework of the K*plus* Competence Center Program.

References

[1] A. Bonarini. ELF: Learning incomplete fuzzy rule sets for an autonomous robot. In *Proc. 1st European Congress on Fuzzy and Intelligent Technologies*, volume 1, pages 69–75, Aachen, 1993.

[2] A. Bonarini. Evolutionary learning of fuzzy rules: Competition and cooperation. In W. Pedrycz, editor, *Fuzzy Modeling: Paradigms and Practice*, pages 265–283. Kluwer Academic Publishers, Dordrecht, 1996.

[3] A. Bonarini. Anytime learning and adaptation of hierarchical fuzzy logic behaviors. *Adaptive Behavior*, 5:281–315, 1997.

[4] J. Casillas, B. Carse, and L. Bull. Fuzzy XCS: An accuracy-based fuzzy classifier system. In *Proc. 12th Spanish Conference on Fuzzy Logic and Technologies*, pages 369–376, Jaen, 2004.

[5] J. H. Holland. *Adaptation in Natural and Artificial Systems*. The MIT Press, Cambridge, MA, first MIT Press edition, 1992. First edition: University of Michigan Press, 1975.

[6] R. Kruse, J. Gebhardt, and F. Klawonn. *Foundations of Fuzzy Systems*. John Wiley & Sons, New York, 1994.

[7] A. Parodi and P. Bonelli. A new approach to fuzzy classifier systems. In S. Forrest, editor, *Proc. 5th Int. Conf. on Genetic Algorithms*, pages 223–230, Los Altos, CA, 1993. Morgan Kaufmann.

[8] M. Valenzuela-Rendón. The fuzzy classifier system: A classifier system for continuously varying variables. In R. K. Belew and L. B. Booker, editors, *Proc. 4th Int. Conf. on Genetic Algorithms*, pages 346–353, San Mateo, CA, 1991. Morgan Kaufmann.

[9] M. Valenzuela-Rendón. Reinforcement learning in the fuzzy classifier system. *Expert Systems with Applications*, 14:237–247, 1998.

[10] S. M. Wilson. Classifier fitness based on accuracy. *Evolutionary Computation*, 3(2):149–175, 1995.