Fuzzy-XCS: An Accuracy-Based Fuzzy Classifier System *

Jorge Casillas
Dept. Computer Science & Artif. Intell.
University of Granada
Granada E-18071, Spain
casillas@decsai.ugr.es

Brian Carse, Larry Bull
Fact. Comput., Eng. & Mathematical Sci.
University of the West of England
Bristol BS16 1QY, UK
Brian.Carse, Larry.Bull@uwe.ac.uk

Abstract

The issue of rule generalization has received a great deal of attention in the discrete-valued learning classifier system field. In particular, the accuracy based XCS is the subject of extensive ongoing research.

However, the same issue does not appear to have received a similar level of attention in the case of the fuzzy classifier system. This may be due to the difficulty in extending the discrete-valued system operation to the continuous case.

The intention of this contribution is to propose an approach to properly develop a fuzzy XCS system.

Keywords: learning classifier systems, XCS, evolutionary algorithm, Michigan-style learning fuzzy systems, fuzzy implications.

1 Introduction

The fuzzy classifier system is a machine learning system which employs linguistic rules and fuzzy sets in its representation and an evolutionary algorithm (EA) for rule discovery. It therefore combines an easily understood representation (as opposed to, for example, neural networks approaches) with a general purpose search method. In order to exploit the fuzzy representing to the full, the ability to learn generalization is of great importance.

Generalized rules allow more compact rule bases, scalability to higher dimensional spaces, faster inference, and better linguistic interpretability. The issue of rule generalization, and the interplay between general and specific rules in the same evolving population, has received a great deal attention in the discrete-valued classifier system research community (e.g. [8]). The same issue does not appear to have received a similar level of attention in the case of fuzzy classifier systems [1, 2, 3, 4, 5, 6, 7].

Traditional Michigan-style classifier systems have been "strength-based" in the sense that a classifier accrues strength during interaction with the environment (through rewards and/or penalties). This strength can then be used for two purposes: resolving conflicts between simultaneously matched classifiers during learning episodes; and as the basis of fitness for the EA. A completely different approach can be taken in which a classifier's fitness, from the point of view of the EA, is based on its "accuracy," i.e. how well a classifier predicts payoff whenever it fires. Such an accuracy-based approach offers a number of advantages such as avoiding over-general classifiers, obtaining optimally general classifiers, and learning of a complete "covering map." This accuracy-based classifier system, called XCS, was proposed in [8] and it is currently of major interest to the research community in this field.

This work aims at proposing a new approach to achieve accuracy-based Michigan-style fuzzy classifier systems. The proposal, Fuzzy-XCS, is based on XCS but properly adapted to fuzzy systems. The fuzzy inference and the reinforcement component are changed to consider a different, competitive interaction among rules that allow an accuracy-based operation.

The paper is organized as follows. Section 2 introduces some difficulties to develop an accuracy-based fuzzy classifier system. Section 3 describes a competitive inference mechanism. Section 4 introduces the proposed Fuzzy-XCS system. Section 5 shows some experimental results. Section 6 concludes.

* Part of this research has been developed during a research stay of J. Casillas at the Faculty of Computing, Engineering and Mathematical Sciences of the University of the West of England (Bristol, UK) supported by the *Secretaría General de Universidades e Investigación de la Junta de Andalucía* (Spain)

2 Difficulties in Accuracy-Based Fuzzy Classifier Systems

Most of the difficulty in accuracy-based fuzzy classifier system is the fuzzy inference process, therefore, it is briefly explained in the next subsection. Then, the problems when using strength-based Michigan fuzzy classifier systems, the advantages of considering accuracy-based fitness, and the difficulties in doing that are introduced.

2.1 Cooperation-Based Fuzzy Inference

This is the most commonly used fuzzy inference method. Linguistic (or Mamdani-type) fuzzy rule-based systems are formed by linguistic fuzzy rules with the following structure:

...
also
R^{r-1}: **IF** X_1 is A_1^{r-1} and ... and X_n is A_n^{r-1}
THEN Y_1 is B_1^{r-1} and ... and Y_m is B_m^{r-1} ,
also
R^r: **IF** X_1 is A_1^r and ... and X_n is A_n^r
THEN Y_1 is B_1^r and ... and Y_m is B_m^r ,
also
...

with X_i and Y_j being input and output linguistic variables respectively, and with A_i and B_j being linguistic labels with associated fuzzy sets defining their meaning. These linguistic labels use a global semantic defining the set of possible fuzzy sets used for each variable.

The inference process, which obtains an output as response to a specific input, consists basically of the following steps: for each fired (or matched) fuzzy rule, firstly the conjunction (***and***) operator is used to obtain the *matching degree* of the rule; secondly, the implication (***then***) operator is applied to scale the output fuzzy set to a degree according to the matching; finally, the last step is to apply the aggregation (***also***) operator to reduce the combined output of all the rules acting together to a single fuzzy set.

If we need a real-valued output, a last stage (defuzzification), converts the output fuzzy set to a number. Center of gravity, center of sums, or mean of maxima are some possibilities to do that.

The most widely inference scheme used in linguistic fuzzy systems is that proposed by Mamdani and Assilian in the first fuzzy controller of 1975. It involves the combination of operators (*min*,*min*,*max*), i.e., minimum for conjunction, minimum for implication, and maximum for aggregation. This inference is usually named Max-Min. Alternatively, sometimes it uses another t-norm (e.g. product) to play the role of conjunction and/or implication, or another t-conorm (e.g. bounded sum) to the also operator.

This is the most commonly used approach (we could say even the only one) followed in engineering problems such as fuzzy control and fuzzy modeling. It is because of, with these operators, the fuzzy rules "cooperate" to generate the output, in the sense that an Interpolative Reasoning is performed to define the output in the zones where several rules work to a medium matching degree. Indeed, the fact of using a t-conorm to aggregate the information involves that we are using a *union* and, therefore, the effect of each rule is added to the final consensual output.

2.2 Problems with "Strength-Based" Michigan Fuzzy Classifier Systems

As mentioned in the introduction, the strength-based fuzzy classifier system is characterized by using the same parameter (i.e. strength) to resolve conflicts between matched classifiers and to compute their fitness. A number of problems arise from this dual use of classifier strength. These include:

1. The cooperation/competition problem. High-strength, potentially cooperative classifiers go on to compete under the action of the EA.

2. Over-general rules with relatively high (but inconsistent) payoff can come to dominate the population.

3. In some environmental states, the maximum payoff achievable (by performing the best possible action for that state) may be relatively low. Although a classifier might be the best that can exist for that state, it can be eradicated from the population by other classifiers that achieve higher rewards in other states. This results in gaps in the system's "covering map."

2.3 Advantages of using "Accuracy-Based" Fitness

A completely different approach can be taken in which a classifier's fitness, from the point of view of the EA, is based on its "accuracy," i.e. how well a classifier predicts payoff whenever it fires. Such an accuracy-based approach offers a number of advantages. Firstly, it can distinguish between accurate and over-general classifiers: an over-general classifier will have relatively low accuracy since payoff will vary according to the input states covered by the classifier. Indeed, it has been shown (in the discrete valued case) that the accuracy-based approach can lead to evolution of optimally general classifiers. Additionally it can maintain both consistently correct and consistently incorrect classifiers which allows learning of a complete "covering map."

A potential drawback of the accuracy-based approach is that it is likely to require larger populations of classifiers.

2.4 Difficulties in Moving to "Accuracy-Based" Fuzzy Classifier Systems

Firstly, in a traditional fuzzy classifier system several rules fire in parallel (this is how the system achieves interpolation); credit assignment is much more difficult in the fuzzy case and it may well be that apportioning credit in proportion to a fuzzy classifier's activation level is not appropriate. A further difficulty is measuring the accuracy of a rule's predicted payoff since (particularly early in the search) a fuzzy rule will fire with many different other fuzzy rules at different time-steps, giving very different payoffs. Yet another difficulty is that the payoff a fuzzy rule receives depends on the input vector — an active fuzzy rule will receive different payoffs for different inputs. This further complicates payoff predictions used as the basis for accuracy-based fitness.

3 Competitive Fuzzy Inference

The problems explained in the previous section are mainly due to the fact that, usually, all the matched rules *cooperate* to define the final solution in a interpolative behavior as explained in Section 2.1. If it is the problem, why not to change our approach and look for competitive interaction? This section introduces such an approach.

3.1 Why Use Competitive Inference?

Michigan-style learning classifier systems in general, and XCS in particular, do not consider the interaction of the different rules as a cooperative action, instead they consider that each rule competes with the rest to be the best one for a particular input vector. This involves that the action is only due to a set of rules that were the winners among the rules composing the match set. Extending this approach to fuzzy modeling, it makes sense to deal with competitive fuzzy rules, in which case, we need to know how to work with such rules.

In off-line fuzzy system learning, it seems that cooperative inference has some clear advantages that make it be more suitable for this problem. In this case, we known the whole data set *a priori* and our objective is to build a landscape of the output data. To do that, it seems a good approach to fill the gaps between data with interpolation, and this is the reason why cooperative inference is successful.

However, in on-line fuzzy system learning, why is it necessary to interpolate the actions of the rules? The objective in this problem should not be to define a landscape of the output, but a landscape of the reward. This means taking the best action in each state. For this purpose, a competitive action makes more sense since our objective is to optimize the rules individually to have the best reward.

With competitive fuzzy inference we propose a decision making process where the matched rules compete among themselves, and only one is the winner. The output finally obtained is mainly due to the winner rule. That is, loser rules do not have an important influence on the output. Nevertheless, competition does not involve a loss of interaction. There is an interaction, but with a selfish objective. To perform competitive inference we only need to change the roles of the inference operators.

3.2 Fuzzy Operators for Competitive Inference

First of all, since each rule competes with the rest, we should use an intersection (t-norm) as *also* operator, instead of the union approach followed by the cooperative inference. We use the *minimum* in this paper. However, this change does not allow us to use the same kind of implication operator.

Analyzing the classical Boolean implication $p \Rightarrow q$, equivalent to $\neg p \vee q$, it is true when p is false. That is, when we have not information about the certainty of the fact (antecedent), we assume the "optimistic" view of thinking that the implication is true.

However, in cooperative inference, a t-norm (intersection) is used as implication. It is a "pessimistic" view that assumes the falseness of the implication when the fact is false. It works if we use a union (t-conorm) to aggregate the effect of the different rules. This behavior could be interpreted that the belief of the rules resides in the set of them, instead of in each of them.

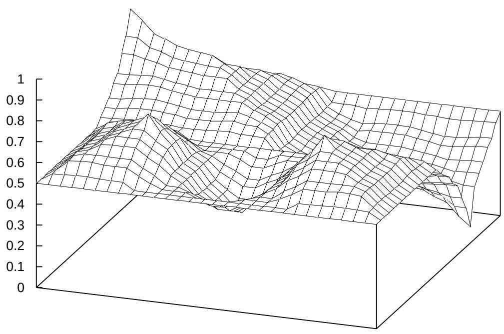
To develop a competitive inference where a intersection is considered as aggregation, the implication should follow the classical Boolean if-then view assuming that each rule has all the information about the relationship between fact and consequence. Thus, the belief develops separately for each rule. This family of implications are known as *logical* fuzzy implications, instead of the *engineering* fuzzy implications based on the t-norm. Table 1 includes some of the best known *logical* fuzzy implications.

To understand better the different behavior between cooperative and competitive inference, Figure 1 shows the output generated by two fuzzy systems that only

Table 1: Some *logical* fuzzy implications

Operator	$\mu_R(x,y)$
Kleene-Dienes	$max\{1-\mu_A(x), \mu_B(y)\}$
Zadeh	$max\{min\{\mu_A(x), \mu_B(y)\}, 1-\mu_A(x)\}$
Łukasiewicz	$min\{1, 1-\mu_A(x)+\mu_B(y)\}$
Dubois-Prade	$\begin{cases} 1-\mu_A(x) & \text{if } \mu_B(x)=0 \\ \mu_B(x) & \text{if } \mu_A(x)=1 \\ 1 & \text{otherwise} \end{cases}$
Gödel	$\begin{cases} 1 & \text{if } \mu_A(x) \leq \mu_B(x) \\ \mu_B(x) & \text{otherwise} \end{cases}$
Goguen	$min\{\mu_B(y)/max\{\mu_A(y), \mu_B(y)\}\}$

differ in the inference mechanism. In the cooperative inference, minimum, minimum, and maximum are used as conjunction, implication, and aggregation operators, respectively. In the competitive inference, minimum, Łukasiewicz, and minimum are used as conjunction, implication, and aggregation operators, respectively. The center of gravity is used as defuzzification process in both cases. Both systems consist of two input variables and one output variable, with 5 triangular fuzzy sets uniformly distributed in the universe of discourse [0,1] for each variable. Among the 25 fuzzy rules considered, only 7 have a consequent different from the medium linguistic term (with vertex at 0.5). As we can see, both approaches generate a smooth output but, while cooperative inference fills the gaps by interpolating, competitive inference isolates the effect of each rule.



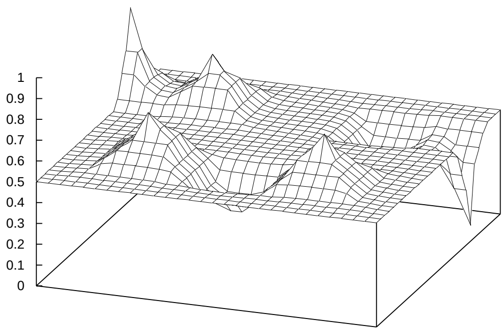


Figure 1: Output generated by *cooperative* (top) and *competitive* (bottom) inferences

On the other hand, Figure 2 illustrates some examples of competitive inference depending on the used implication operator. The *minimum* aggregation operator is used in all the cases. We can see how the action strength of the winner rule (the one with the highest matching degree) not only depends on its matching degree but also on the difference in the matching degree with respect to fuzzy rules with different actions (consequents), i.e., rival fuzzy rules. The more different these matching degrees, the higher the action strength of the selected rule. This involves that actually, the competition is performed among different linguistic actions. Therefore, we should say that competitive inference leads to competitive actions, but not necessarily competitive rules.

The competition among rules with the same action is only performed when using the minimum operator as aggregation, since in this case only the best rule of that action is considered in the inference. The use of other aggregation operators, such as the product, will lead the inference to a cooperation among rules with the same actions. According to XCS, this approach is not desired since it aims at obtaining the $X \times A \Rightarrow P$ map with the best classifier for each situation-action combination [8].

We can characterize slightly different interactions approaches among rules with different actions (i.e. rival rules) depending on the logical fuzzy implication operator used; these are shown in Table 2.

Table 2: Type of competition among rules with different actions depending on the used *implication* operator (a t-norm is used as *also* operator)

Implication	**Type of competition**
Kleene-Dienes	Rival rules reduce the importance of the winner rule in non-overlapped areas
Zadeh	Like Kleene-Dienes. No interaction among rules if the winner rule has a matching degree lower than 0.5
Łukasiewicz	Like Kleene-Dienes, but with higher preservation of overlapped areas than it
Dubois-Prade	Like Kleene-Dienes, but with the highest preservation of overlapped areas
Gödel	Output always in the overlapped area. No output with more than two different actions. If matchings greater than cross-point, output in the center of the overlapped area
Goguen	Like Gödel. Better with *mean of maximum* defuzzification

Finally, we should say that the consideration of competitive actions seems to fit properly with the XCS aim. Other fuzzy learning classifier systems, like Bonarini's work [1, 2], focus on the interaction among

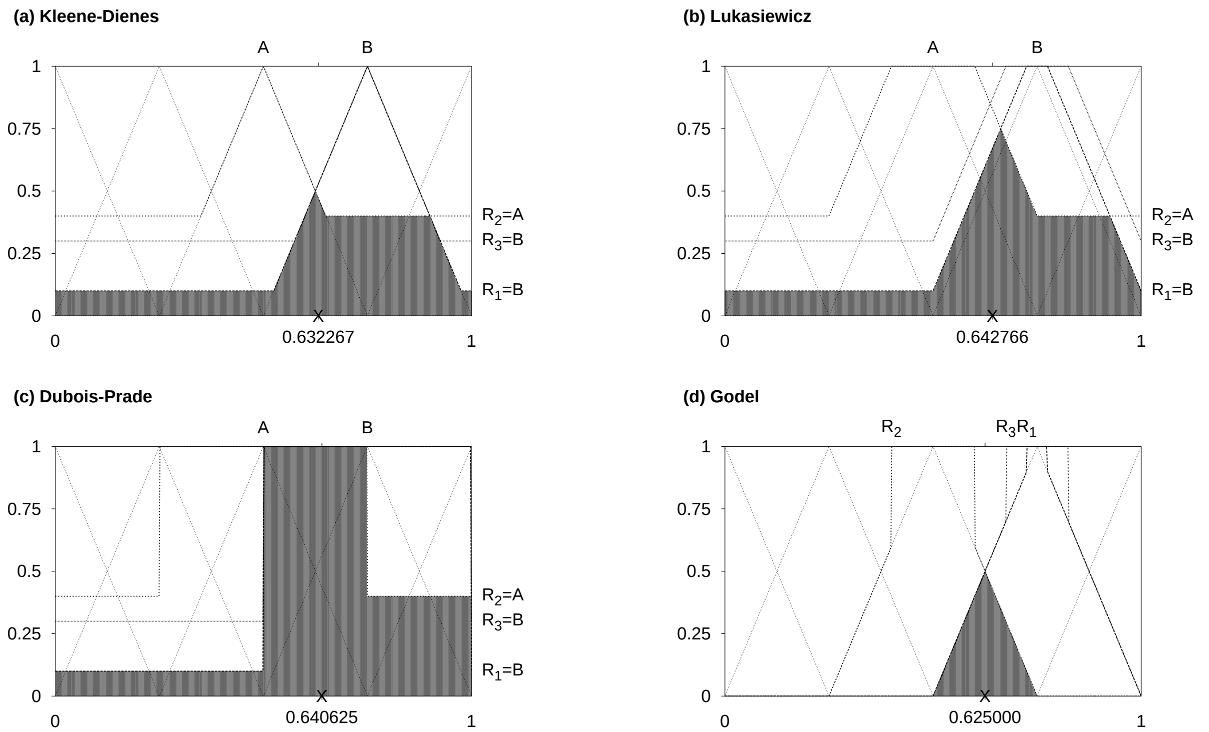


Figure 2: Results of four different fuzzy implications using minimum as *also* operator. Matching degrees: $\mu_{A_{R_1}} = 0.9$, $\mu_{A_{R_2}} = 0.6$, and $\mu_{A_{R_3}} = 0.7$

rules in the antecedent (state) instead of the consequent (action). He proposes a competition among rules with the same antecedent but a cooperation among rules with different ones. Besides, the implication and aggregation considered in his proposal leads implicitly to a cooperation between the consequents.

4 Fuzzy-XCS

The use of competitive fuzzy rules seems to solve the problems shown by the cooperative approach (Section 2.4). Let us now see how Fuzzy-XCS works.

4.1 Generalization Representation

First of all, a representation of fuzzy classifiers to allow proper generalization must be done. We propose the use of *disjunctive normal form* (DNF) fuzzy rules with the following structure:

$$\textbf{IF } X_1 \text{ is } \widetilde{A_1} \text{ and } \ldots \text{ and } X_n \text{ is } \widetilde{A_n} \textbf{ THEN } Y \text{ is } B$$

where each input variable X_i takes as a value a set of linguistic terms $\widetilde{A_i} = \{A_{i1} \vee \ldots \vee A_{il_i}\}$, whose members are joined by a disjunctive (t-conorm) operator, whilst the output variable remains a usual linguistic variable with a single label associated. We use the *bounded sum* ($min\{1, a+b\}$) as t-conorm.

This structure uses a more compact description that allows rules with different generalization degrees. Moreover, the structure naturally supports the absence of some input variables in each rule (simply making $\widetilde{A_i}$ be the whole set of linguistic terms).

In order to use this representation in Fuzzy-XCS, we propose a binary coding scheme for the antecedent of the fuzzy rule with size equal to the sum of the number of linguistic terms used in each input variable. The allele '1' means that the corresponding linguistic term is used in the corresponding variable. For the consequent of the rule, an integer coding scheme is used where each gene contains the index of the linguistic terms used for the corresponding output variable. For example, assuming we have three linguistic terms (S, M, and L) for each input/output variable, the fuzzy rule [IF X_1 is S and X_2 is {M or L} THEN Y_1 is M and Y_2 is L] is encoded as [100|011||23].

4.2 Performance Component

In XCS [8], the performance component consists of three stages: match set construction, prediction array computation, and action set selection. This process has the final objective of inferring an specific action from the set of classifiers that matches the current state. In Fuzzy-XCS the process is different, as described as follows:

• *Match set*: To avoid an excessive number of matched rules, only those rules with a matching degree greater or equal to a specific value (θ_M) are included in the match set.

• *Computation of candidate subsets*: This stage could be equivalent to the prediction array computation. It is necessary in discrete output systems like XCS to groups the matched classifiers according to the different actions, since it is not possible to select several actions. However, in real-valued output systems like Fuzzy-XCS, several "linguistic actions" (consequents) could be considered together. Thus, Fuzzy-XCS redefines the concept of prediction array computation. We can assume that what should not be accepted in our case is to have an action set with inconsistent rules, i.e. rules with the same antecedent and different consequent. Since DNF-type rules are considered, to have the same antecedent also involves rules where the antecedent of some of them are contained in others. Therefore, different groups of consistent fuzzy rule set (with the maximum number of rules in each group) are formed.

• *Action set selection*: The action set selection chooses the consistent classifier set with the highest mean prediction. In learning classifier systems the criterion used to select the action set alternately varies between exploration and exploitation approaches. According to this, our deterministic selection could be considered as a purely exploitation approach. Although we have experimented with several possibilities, the best results were obtained with this procedure.

4.3 Reinforcement Component

The p_j (prediction), ϵ_j (prediction error), and F_j (fitness) values are adjusted by the reinforcement learning standard techniques used in XCS (Q-learning, Widrow-Hoff, and MAM) for each fuzzy classifier C_j.

However, an important difference is considered in Fuzzy-XCS: the distribution among the classifiers must be made proportionally to the degree of contribution of each classifier to the obtained output. This is a crucial issue because the interaction among the classifiers (with competition actions in our case) is developed here.

The reinforcement performed in Fuzzy-XCS acts on the action set AS. The following subsections detail the reinforcement distribution process and the adjustment of the parameters.

4.3.1 Distribution of the Reinforcement

The reinforcement distribution among the classifiers of the action set AS is made by analyzing the contribution of each classifier to generate the aggregated output fuzzy set. Let B'_j be the scaled output fuzzy set generated by the fuzzy rule R_j:

$$B'_j = I(\mu_{A_{R_j}}(x), B_j), \tag{1}$$

where $\mu_{A_{R_j}}(x)$ is the matching degree of the rule R_j, I is the used logical fuzzy implication operator (see Table 1), and B_j the fuzzy set of the consequent of the rule R_j. Let $R_1 \in AS$ be the winner rule, i.e., $\mu_{A_{R_1}}(x) \geq \mu_{A_{R_j}}(x), \forall R_j \in AS - \{R_1\}$.

The process involves analyzing the area that the rival fuzzy rules (rules with lower matching degrees than R_1) "bite" into the area generated by the winner rule.

Thus, the weight of the winner rule is:

$$w_1 = \frac{\int \bigwedge_{j=1}^{|AS|} \mu_{B'_j}(y)\ dy}{\int \mu_{B'_1}(y)\ dy}, \tag{2}$$

with $|AS|$ being the action set size and $\wedge$ the t-norm used as aggregation operator (the minimum in our case), while the weights of the rival rules are computed as follows:

$$w_j = \frac{(1-w_1)\cdot\left(\int \mu_{B'_1}(y)\ dy - \int \mu_{B'_1}(y) \wedge \mu_{B'_j}(y)\ dy\right)}{\sum_{i=2}^{|AS|}\left(\int \mu_{B'_1}(y)\ dy - \int \mu_{B'_1}(y) \wedge \mu_{B'_i}(y)\ dy\right)} \tag{3}$$

This distribution is designed to take into account that competitive inference is being considered. To illustrate this behavior, we can see that, from the example shown in Figure 2(a), the competitive inference generates the weights $w_1 = 0.711$, $w_2 = 0.289$, and $w_3 = 0$, while a cooperative-based distribution proportional to the matching degrees generates the weights $w_1 = 0.409$, $w_2 = 0.318$, and $w_3 = 0.273$. Indeed, the selection pressure in the competitive inference is higher, thus allowing to discriminate between good and bad rules.

4.3.2 Adjustment of the Parameters

To adjust the parameters of each classifier, firstly the P (payoff) value is computed with the Q-learning technique as follows: $P = r + (\gamma \cdot \mu_{A_{R_1}}(x))$, with r being the external reward from the previous time-step, and $\gamma \in [0, 1]$ a constant decreasing factor.

Then, the following adjustment process is performed for each fuzzy classifier belonging to the action set:

1. Firstly, adjust the error values ϵ_j using the standard Widrow-Hoff delta rule with learning rate parameter β $(0 < \beta \leq 1)$ toward $|P - p_j|$ considering the weights w_j computed in eqs. (2) and (3) to distribute the adjustment, i.e.

$$\epsilon_j \leftarrow \epsilon_j + \beta \cdot w_j \cdot (|P - p_j| - \epsilon_j). \tag{4}$$

The MAM (moyenne adaptive modifiee) technique is used by adjusting ϵ_j to the average of the $|P - p_j|$ values, instead of the above equation, during the first $1/\beta$ times that the corresponding classifier is adjusted.

2. Then, adjust prediction values

$$p_j \leftarrow p_j + \beta \cdot w_j \cdot (P - p_j). \tag{5}$$

Again, weights are considered to distribute the reinforcement. MAM technique is also used here during the $1/\beta$ first adjustments.

3. Finally, recalculate the fitness values F_j from the updated values of ϵ_j, i.e.

$$F_j \leftarrow F_j + \beta \cdot (k'_j - F_j), \tag{6}$$

with

$$k'_j = \frac{k_j}{\sum_{R_i \in AS} k_i}, \quad k_j = \begin{cases} (\epsilon_j/\epsilon_0)^{-\nu} & \epsilon_j > \epsilon_0 \\ 1 & \text{otherwise.} \end{cases} \tag{7}$$

No weights w_j are considered to update the fitness since it depends on the prediction error instead of the received payoff. Again, MAM technique is used.

4.4 Discovery Component

The EA for Fuzzy-XCS acts only on the action set. It selects two classifiers with probabilities proportional to their fitness, applies crossover and mutation operators with probabilities χ and μ_{chom} (per chromosome), respectively, and inserts the offspring in the population. If the population contains the maximum number of fuzzy classifiers allowed, two individuals are deleted to make room. They are randomly selected proportionally to the prediction error weighted by the mean action set sizes where each fuzzy classifier was involved.

A simple two-point crossover operator that only acts on the antecedent part of the chromosomes (binary coding scheme) is considered. Prediction, prediction error, and fitness values of offspring are initialized to the mean values of the parents.

The mutation randomly selects an input/output variable of the rule. If an input variable is selected, one of the three following possibilities is applied: *expansion*, which flips to 1 a gene of the selected variable; *contraction*, which flips to 0 a gene of the selected variable; or *shift*, which flips to 0 a gene of the variable and flips to 1 the gene immediately before or after it. The selection of one of these mechanisms is made randomly among the available choices (e.g., contraction can not be applied if only a gene of the selected variable has the allele 1). If an output variable is selected, the mutation operator simply increases or decreases the integer value. Prediction, prediction error, and fitness values of mutated classifiers are not changed.

An EA subsumption is performed. Thus, if the offspring is logically contained by either of its parents and this parent is sufficiently experienced (it has been updated a threshold θ_{GA} number of times), the offspring is not added to the pool but the parent's numerosity is incremented.

When no fuzzy rules cover the state with the highest matching degree, a covering mechanism is used to include a fuzzy classifier with the input linguistic term set that best matches the state and a random action.

5 Experimental Results

Experiments have been performed to test the behavior of Fuzzy-XCS. We have developed a laboratory problem to play a similar role as the multiplexer problem for discrete-valued classifier systems. Thus, we have generated a example data set from a previously defined rule base with different degrees of generalization. Two input variables and one output variable are considered. A total of 576 examples uniformly distributed in the input space (24×24) were generated. Five linguistic terms are considered for each variable. Uniformly distributed triangular-shaped membership functions are used. The rule base considered to generate the data set is shown in Table 3. The reward depends inversely on the difference between the inferred and the desired output in a non-linear way. The objective is to obtain the set of rules that best approximate the data with the highest degree of generalization, i.e., a rule base as accurate and compact as possible.

Table 3: Rule base used to generate the data set

	X_1					X_2					Y				
	VS	S	M	L	VL	VS	S	M	L	VL	VS	S	M	L	VL
R_1	x	x				x	x				x				
R_2	x	x						x	x	x		x			
R_3			x	x	x	x							x		
R_4			x	x	x		x							x	
R_5			x	x	x			x	x	x					x

The used parameter values are the following: maximum number of classifiers = 200, $\beta = 0.2$, $\nu = 3$, $\epsilon_0 = 0.05$, $\alpha = 0.1$, $\theta_{Del} = 30$, $\theta_{GA} = 30$, $\chi = 0.8$, $\mu_{chrom} = 0.1$, $\theta_{timestep} = 20$, $\theta_M = 0.3$, *also* operator = min, and *then* operator = Łukasiewicz. γ parameter is not considered since it is a simple-step problem.

Figure 3 shows the average behavior of 10 runs of Fuzzy-XCS with competitive inference and reinforcement. The upper figure depicts the relative numerosity of the five optimum fuzzy classifiers. It shows the capability of the algorithm to find and keep the optimum solution. The bottom figure depicts the mean approximation error of the last 50 iterations. It shows the capability of the algorithm to provide the appropriate action (output) to the corresponding state (input). Note that the optimum solution is properly found.

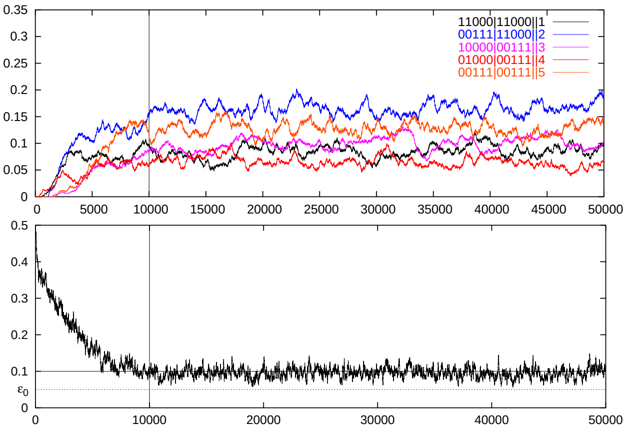


Figure 3: Results of Fuzzy-XCS with *competitive* inference and reinforcement approach

On the other hand, Figure 4 shows the behavior of an algorithm following a cooperative inference and reinforcement. To do that, the only changes made to the algorithm proposed in this paper are the use of *minimum* as implication and *maximum* as aggregation in the inference engine, and a distribution of the reward proportional to the matching degrees in the reinforcement component. This algorithm is tested with a data set generated with the same cooperative inference (Max-Min) to avoid biasing the experiment. Notice that the cooperative approach is unable to find the solution.

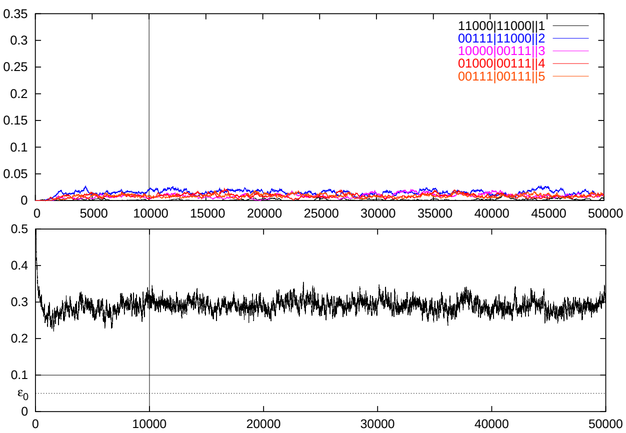


Figure 4: Results of Fuzzy-XCS with *cooperative* inference and reinforcement approach

6 Concluding Remarks

The paper has presented a proposal to properly develop an accuracy-based fuzzy classifier system. It is mainly based on a different inference approach that considers the interaction among fuzzy classifiers (rules) from a competitive point of view. The reinforcement component, based on XCS, is adapted to allow this behavior. The approach has the advantage of performing a higher selection pressure that results in a proper discrimination between good and bad fuzzy classifiers.

Promising results of the proposal have been obtained in a simple laboratory problem. Current and future work involves investigating the behavior of the proposal in multi-step and real-world problems.

References

[1] A. Bonarini. Evolutionary learning of fuzzy rules: competition and cooperation. In W. Pedrycz (Ed.), *Fuzzy modelling: paradigms and practice*, pages 265–284. Kluwer Academic, Norwell, MA, USA, 1996.

[2] A. Bonarini and V. Trianni. Learning fuzzy classifier systems for multi-agent coordination. *Information Sciences*, 136:215–239, 2001.

[3] H. Ishibuchi, T. Nakashima, and T. Murata. Performance evaluation of fuzzy classifier systems for multidimensional pattern classification problems. *IEEE Trans. Syst., Man, and Cybern.—Part B: Cybern.*, 29:601–608, 1999.

[4] A. Parodi and P. Bonelli. A new approach to fuzzy classifier systems. In *Proc. 5th Int. Conf. Genetic Algorithms*, pp. 223–230, San Mateo, CA, USA, 1993. Morgan Kaufmann Publishers.

[5] M. Valenzuela-Rendón. The fuzzy classifier system: a classifier system for continuously varying variables. In *Proc. 4th Int. Conf. Genetic Algorithms*, pp. 346–353, San Mateo, CA, USA, 1991. Morgan Kaufmann Publishers.

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

[7] J.R. Velasco. Genetic-based on-line learning for fuzzy process control. *Int. J. Intell. Syst.*, 13:891–903, 1998.

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