

An Integrated Approach for the Identification of Compact, Interpretable and Accurate Fuzzy Rule-Based Classifiers from Data

Andri Riid

Laboratory of Proactive Technologies
Tallinn University of Technology
Ehitajate tee 5, Tallinn 19086, Estonia
Email: andri@dcc.ttu.ee

Ennu Rüstern

Department of Computer Control
Tallinn University of Technology
Ehitajate tee 5, Tallinn 19086, Estonia
Email: ennu.rustern@dcc.ttu.ee

Abstract—This paper presents three very simple and computationally undemanding symbiotic algorithms for the identification of compact fuzzy rule-based classifiers from data. The problem of interpretability is specifically addressed, resulting in a conclusion that due to the characteristics of classification tasks a major well-known interpretability condition - distinguishability - can be discarded. It is shown that despite the interpretability-accuracy tradeoff, accuracy of identified classifiers stands out to comparison. All obtained properties can be very useful in practical problems. The proposed method is validated on Iris, Wine and Wisconsin Breast Cancer data sets.

I. INTRODUCTION

A classifier is an algorithm that assigns a class label to an object, based on the object description. The object description comes in the form of a vector containing values of the features (attributes) that are considered to be relevant for the classification task. Typically, the classifier learns to predict class labels using a training algorithm and a training data set (alternatively, when a training data set is not available, a classifier can be designed from prior knowledge and expertise). Once trained, the classifier is ready for operation on unseen objects.

Importance of classification and the need for accurate, reliable and computationally efficient classifiers cannot be underestimated because many problems in very different fields can be represented as classification problems. The applications of classifiers run from the recognition of hand-written characters or faces to the problem of diagnosing a disease from registered symptoms.

In this paper we deal with fuzzy classifiers. According to [1], any classifier that uses fuzzy sets or fuzzy logic in the course of its training or operation can be considered a fuzzy classifier. However, a distinction can be made between the algorithms that assign partial (fuzzy) class membership to the objects (fuzzy clustering algorithms such as fuzzy c-means [2], Gustafson-Kessel [3] or Gath-Geva clustering [4]) and fuzzy rule-based systems that utilize fuzziness only in the reasoning mechanism. We focus on the latter because such classifiers can be interpreted linguistically that makes the reasoning that lies behind assigning an object into any given class much more

transparent, which can be very useful in giving hindsight into the decisions the classifier has made.

The paper is organized as follows. We first show that the concept of transparency in fuzzy rule-based classifiers is principally different from the one valid for fuzzy models/controllers and discuss its implications in section II. In sections III, IV and V, we present very simple and computationally low-cost algorithms for classification, subsequent rule base reduction and rule compression, respectively. These algorithms are tested on several benchmark classification problems in section VI to show that the proposed package of algorithms is indeed competitive.

II. FUZZY RULE-BASED CLASSIFIERS

In the field of classification, a classifier is expected to group the examples presented to it into a small number of distinct classes, that are labelled with discrete values $(1, 2, \dots, T)$ where T is the number of classes. Note that the actual numerical value assigned to a class is irrelevant, it just functions as a label.

Thus, a rule in a fuzzy rule-based classifier appears as

$$\begin{aligned} &\text{IF } x_1 \text{ is } A_{1r} \text{ AND } x_2 \text{ is } A_{2r} \text{ AND } \dots \\ &\dots \text{ AND } x_N \text{ is } A_{Nr} \text{ THEN } y \text{ belongs to class } c_r \quad (1) \\ &\text{OR } \dots, \end{aligned}$$

where A_{ir} denote the linguistic labels of the i -th input variable associated with the r -th rule ($i = 1, \dots, N, r = 1, \dots, R$) and c_r is a class assigned to the r -th rule ($c_r \in (1, \dots, T)$).

Note that in classification we typically deal with a high number of features/inputs and the number of data samples is rather scarce (thus distributed unevenly over the input (hyper)space).

A fuzzy system can be configured to produce neat discrete output values that are needed for classifiers. In [7], [8] this is accomplished by replacing defuzzification and aggregation stages in the inference algorithm with the winner-takes-it-all strategy so that output is the class related to the consequent of the rule that has the highest degree of activation (the only

inference parameter that remains relevant is the conjunction operator, which is minimum throughout the paper).

$$y = c_r, \arg \max_{1 \leq r \leq R} (\tau_r) \quad (2)$$

where τ_r is the activation degree of the r -th rule

$$\tau_r = \prod_{i=1}^N \mu_{ir}(x_i), \quad (3)$$

where normal and convex membership functions μ_{ir} represent A_{ir} in the numerical domain.

What it means is that proper **classification rules do not cooperate in producing the output of the fuzzy system, instead they compete with each other**. It also means that rule interpolation is never even involved in the inference algorithm thus the crucial interpretability requirement - transparency [5] - does not apply and there is no need to control the overlap of input MFs - fuzzy rule-based classifiers that utilize (2) are transparent by default.

This gives us extra freedom that becomes handy when the classifier has to recognize the class distributions that are overlapping in input dimensions (such as are classes 2 and 3 with class 1 respect to input x_2 in Fig. 1).

$$\begin{aligned} &\text{IF } x_1 \text{ is } A_{11} \text{ AND } x_2 \text{ is } A_{21} \\ &\text{THEN } y \text{ belongs to class 1} \\ &\text{IF } x_1 \text{ is } A_{12} \text{ AND } x_2 \text{ is } A_{22} \\ &\text{THEN } y \text{ belongs to class 2} \\ &\text{IF } x_1 \text{ is } A_{13} \text{ AND } x_2 \text{ is } A_{23} \\ &\text{THEN } y \text{ belongs to class 3} \end{aligned} \quad (4)$$

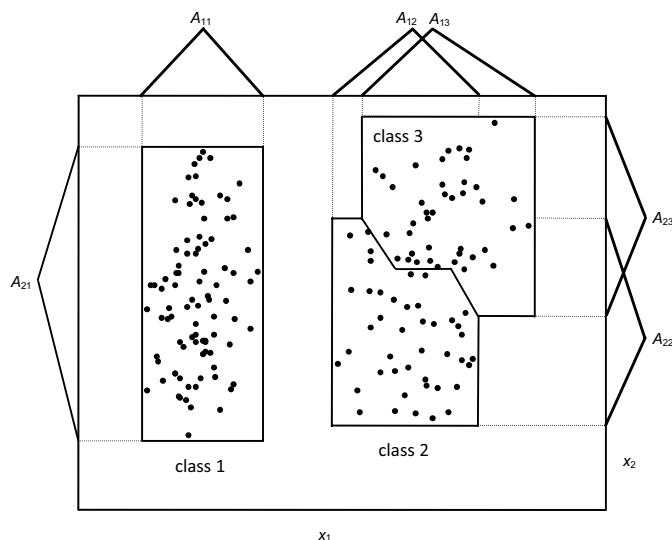


Fig. 1. Highly overlapping MFs do not compromise interpretability in classification problems.

III. BUILDING A RULE-BASED CLASSIFIER

The procedure for generating a fuzzy rule-based classifier is based on k-means clustering. K-means clustering [9] is a well-known algorithm that partitions (clusters) K observations into R (this figure is fixed beforehand) disjoint subsets S_j containing K_j observations so as to minimize the within-cluster sum of squares.

$$J = \sum_{j=1}^R \sum_{k \in S_j} \|\mathbf{x}_k - \mathbf{m}_j\|^2, \quad (5)$$

where $\mathbf{x}_k = (x_1(k), x_2(k), \dots, x_N(k))$ is a vector representing k -th observation and $\mathbf{m}_j = (m_{j1}, m_{j2}, \dots, m_{jN})$ is the geometric centroid of the data points in S_j . The algorithm itself consists of a simple re-estimation procedure as follows.

Initially, R cluster centroids are generated at random. In step 1, every observation is assigned to the cluster j whose centroid is closest to this observation

$$j = \arg \min_{j \in \{1, \dots, R\}} (\|\mathbf{x}_k - \mathbf{m}_j\|) \quad (6)$$

In step 2, a new centroid is computed for each cluster. For each j

$$\mathbf{m}_j = \sum_{k \in S_j} \mathbf{x}_k / K_j \quad (7)$$

These two steps are alternated until a stopping criterion is met, i.e., there is no further change in the assignments of observations. Ideally, the algorithm is able to find such centroids that objects within a cluster are as close to each other as possible and as far away as possible from objects in other clusters.

So, in order to build a fuzzy classifier, first the data representing T classes and consisting of K observations is partitioned into R ($R \geq T$) clusters using the k-means algorithm. All data is normalized into the unit interval.

Given a cluster mean \mathbf{m}_j and corresponding subset S_j , triangular MFs μ_{ir} given by parameters a_{ir}, b_{ir}, c_{ir} are created in all input dimensions. For each i

$$a_{ir} = \min_{k \in S_j} (x_i(k)), c_{ir} = \max_{k \in S_j} (x_i(k)), b_{ir} = m_{ji} \quad (8)$$

Note that the MFs are then slightly enlarged (Fig. 2) so as to give nonzero membership values to the samples located at the very edges of the cluster. Following this a rule is entered into the rulebase of the system.

$$\begin{aligned} &\text{IF } x_1 \text{ is } A_{1r} \text{ AND } x_2 \text{ is } A_{2r} \dots \\ &\text{AND } x_i \text{ is } A_{ir} \dots \text{ AND } x_N \text{ is } A_{Nr} \\ &\text{THEN } y \text{ belongs to class } c_r, \end{aligned} \quad (9)$$

where A_{ir} represent the MFs μ_{ir} and c_r is the class that is prevalent among the observations in cluster j . This is carried out until all clusters have been described with appropriate rules.

As an example we provide a classification experiment of the Iris data set that is a common benchmark problem in

TABLE I
IRIS DATA CLASSIFICATION

Exp. no.	variables				ϱ	R	n_ϵ	ϵ (%)
	1	2	3	4				
1	✓	✓			0.25	9	32	78.67
2	✓		✓		0.17	10	13	91.33
3	✓			✓	0.24	7	4	97.33
4		✓	✓		0.20	9	7	95.33
5		✓		✓	0.20	9	7	95.33
6			✓	✓	0.20	5	4	97.33
7	✓	✓	✓		0.30	9	6	96.00
8	✓	✓		✓	0.30	10	3	98.00
9	✓		✓	✓	0.22	9	3	98.00
10		✓	✓	✓	0.20	9	2	98.67
11	✓	✓	✓	✓	0.32	10	2	98.67

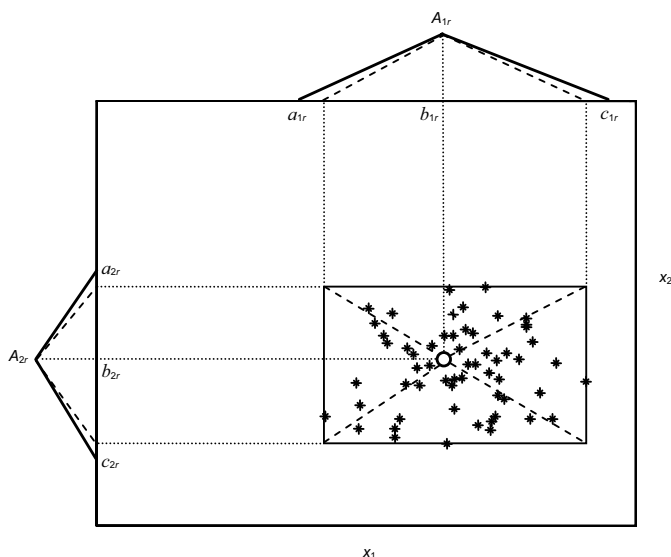


Fig. 2. Simultaneous generation of input MFs and a rule in product space based on given cluster ($N = 2$)

classification and pattern recognition studies. The data set contains 50 measurements of four features (sepal length, sepal width, petal length, petal width) from each of three species (setosa, versicolor, virginica). The first class is separate from others, while the second and third class overlap slightly.

As we do not know beforehand which selection of input variables is optimal, all combinations of two, three and four variables (that along with other experiment characteristic are given in Table I) have been employed.

The number of clusters is determined by a rule of thumb [10]

$$R = \sqrt{K/2} \quad (10)$$

Presently, (10) equals 8.67. Because of the random initialization of the k-means algorithm the results may vary. This is eliminated by supplying the initial centroids that have been computed by subtractive clustering algorithm [11] (finding the appropriate cluster radius (ϱ) by trial and error). This way the k-means converges to the same solutions so the experiments can be replicated at a later date. Concerning the number of rules, there are some notable exceptions, in some cases (Exps 2, 8 and 11) better results were obtained with 10 rules and in some cases (Exps 3 and 6) a smaller number of clusters gave better results. All in all, the results of this series are presented in Table I, where n_ϵ and ϵ denote the number of misclassified samples and overall accuracy, respectively.

Apart from the experiment no. 1 (a really unfortunate choice for input variables as evidenced in Fig. 3), and experiment no. 2 (to a lesser degree), every other classification run shows a pretty good accuracy.

IV. RULE BASE REDUCTION

It is not really good for interpretability of the rule-based classifier if the number of rules/clusters is several times higher

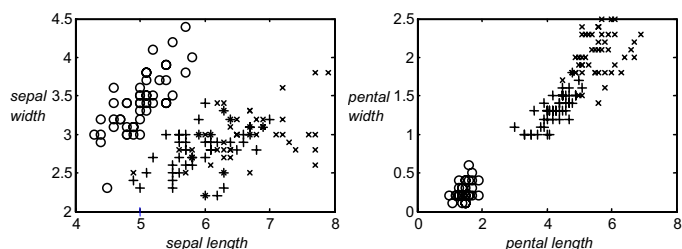


Fig. 3. Separation of classes in different input dimensions. Obviously, clustering the data in right leads to much better classification accuracy. o - iris setosa, + - iris versicolor, x - iris virginica

than the number of classes. One can, of course, specify a low enough R in the first place, which, however, more likely than not will result in accuracy loss. Sometimes it is more useful to be able to reduce the number of rules in the existing rule-based classifier so we can have two classifiers - one with better accuracy and poorly interpretable, another less accurate but with improved interpretability.

The procedure for reducing the number of rules is a very simple one. Two rules, r_a and r_b

$$\begin{aligned} \text{IF } x_1 \text{ is } A_{1a} \text{ AND } \dots \text{ AND } x_N \text{ is } A_{Na} \\ \text{THEN } y \text{ belongs to class } c_a, \\ \text{IF } x_1 \text{ is } A_{1b} \text{ AND } \dots \text{ AND } x_N \text{ is } A_{Nb} \\ \text{THEN } y \text{ belongs to class } c_b, \end{aligned} \quad (11)$$

can be merged into one if they are close enough to each other, i.e. the cluster centroids \mathbf{m}_a , \mathbf{m}_b are within a pre-determined threshold distance δ

$$\|\mathbf{m}_a - \mathbf{m}_b\| < \delta \quad (12)$$

and if $c_a = c_b$ (they assign the same class to the observations within the cluster).

If this is true a new rule r_c

$$\text{IF } x_1 \text{ is } A_{1c} \text{ AND } \dots \text{ AND } x_N \text{ is } A_{Nc} \\ \text{THEN } y \text{ belongs to class } c_c, \quad (13)$$

is generated, and the MFs associated with A_{ic} are created with parameters

$$\forall i, a_{ic} = \min(a_{ia}, a_{ib}), c_{ic} = \max(c_{ia}, c_{ib}), \\ b_{ic} = (b_{ia} + b_{ib})/2 \quad (14)$$

The resulting MF A_{ic} is a kind of union of A_{ia} and A_{ib} (Fig. 4). Obviously, original rules r_a and r_b and corresponding MFs must be deleted from the classifier after the merge. Note also, that the merged rule covers more space than two original rules combined (this can be also evidenced from Fig. 4), which means that it is more general (that does not necessarily mean that it is more accurate).

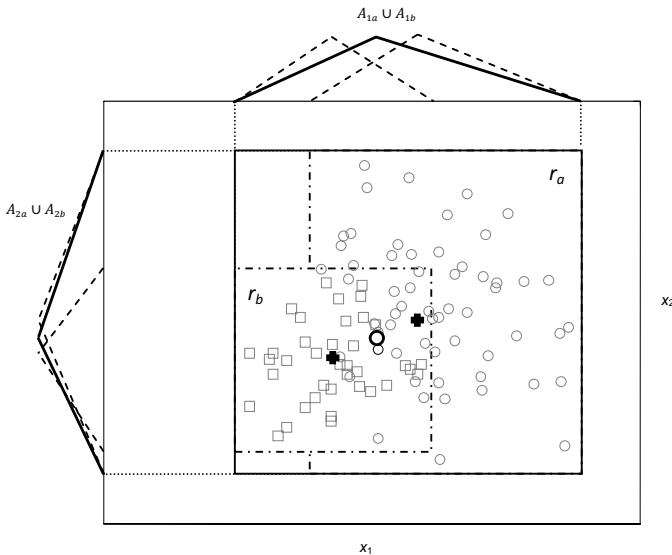


Fig. 4. Merge of two rules ($N = 2$). A new centroid is depicted by a circle.

We apply this reduction algorithm to the Iris classifiers generated in previous section (except for experiments 1 and 2). In the first series (columns 2-4 in Table II) the optimal number of rules is reduced (by choosing appropriate value of δ) that does not harm accuracy of the classifiers, In second series (columns 5-7 in Table II, minimal-rule ($R = T$) classifiers are produced. The blanks in the table mean that the optimal number of rules could not be found, i.e. the number of rules could not be reduced without "considerable" ($> 1\%$) performance loss.

TABLE II
IRIS CLASSIFIERS WITH RULE BASE REDUCTION

Exp. no.	δ	R	ϵ (%)	δ	R	ϵ (%)
3	-	-	-	>0.25	3	94.67
4	-	-	-	>0.18	3	92.67
5	0.20	4	96.00	>0.24	3	94.67
6	0.21	3	97.33	>0.20	3	97.33
7	0.23	5	96.00	>0.24	3	92.00
8	-	-	-	>0.25	3	93.33
9	0.20	7	98.00	>0.27	3	96.00
10	0.18	6	98.00	>0.26	3	95.33
11	0.21	8	98.67	>0.27	3	94.00

We can see, that generally the number of rules can be reduced substantially without any performance loss. Minimal-rule classifiers, on the other hand, are typically less accurate.

V. COMPRESSION OF RULES

Another and even more powerful way to improve interpretability is to remove less important variables from rules.

However, first the conditions for "importance" must be defined. For this purpose, let us define the intersection interval of two fuzzy sets μ_{ia} and μ_{ib} associated with rules r_a and r_b on i -th input variable

$$S_{ab}^{(i)} = \max(\text{supp}(\mu_{ia} \cap \mu_{ib})) - \min(\text{supp}(\mu_{ia} \cap \mu_{ib})) \quad (15)$$

where

$$\mu_{ia} \cap \mu_{ib} = \min(\mu_{ia}(x_i), \mu_{ib}(x_i)) \quad (16)$$

and

$$\text{supp}(\mu_{ia} \cap \mu_{ib}) = \{x_i \in X_i | \mu_{ia}(x_i) > 0, \mu_{ib}(x_i) > 0\}. \quad (17)$$

From this we can move to the concept of the overlap of two fuzzy rules. For two considered rules, r_a and r_b the overlap in input i (1st dimension)

$$O_{ab}^{(i)} = S_{ab}^{(i)} \quad (18)$$

The overlap of the same rules on inputs i and j (2nd dimension) is

$$O_{ab}^{(i,j)} = S_{ab}^{(i)} \cdot S_{ab}^{(j)} \quad (19)$$

Or in general M -th dimensional case, if $i \in (i_1, i_2, \dots, i_M)$.

$$O_{ab}^{(i_1, i_2, \dots, i_M)} = \prod_{i \in (i_1, i_2, \dots, i_M)} S_{ab}^{(i)} \quad (20)$$

Definition: A rule r can be compressed to contain only variables $[i_1, i_2, \dots, i_M]$ if

$$O_r^{(i_1, i_2, \dots, i_M)} = \sum_{\substack{p=1 \\ p \neq r}}^R O_{rp}^{(i_1, i_2, \dots, i_M)} = 0, \quad (21)$$

i.e. it does not intersect with any other rules with the given variable selection.

For example, consider a 4-rule classifier in 2-nd dimension with $i \in (x_a, x_b)$ (Fig. 5) We can see that $O_1^{(a,b)} = 0$ whereas $O_{23}^{(a,b)} > 0$ and $O_{34}^{(a,b)} > 0$. The latter means that all $O_2^{(a,b)}$, $O_3^{(a,b)}$, $O_4^{(a,b)}$ are larger than zero. Consequently, only rule r_1 is subject to compression and can be rewritten as

$$\begin{aligned} \text{IF } x_a \text{ is } A_{a1} \text{ AND } x_b \text{ is } A_{b1} \\ \text{THEN } y \text{ belongs to class } c_1, \end{aligned} \quad (22)$$

regardless of the count of input variables (N).

In practice, instead of zero, we usually use some overlap value θ , $O_r^{(i_1, i_2, \dots, i_M)}$ is compared to. Rule compression is an iterative procedure in which we look for compression into more general rules first. In each dimension $M = 1, \dots, N - 1$ all $N!/(M!(N - M)!)$ unique possible variable selections are verified and the one that gives the minimum value of $O_r^{(i_1, i_2, \dots, i_M)}$ for the given rule is chosen for execution. Rule overlaps in higher dimensions are numerically smaller than those in lower dimensions so using a constant θ favors

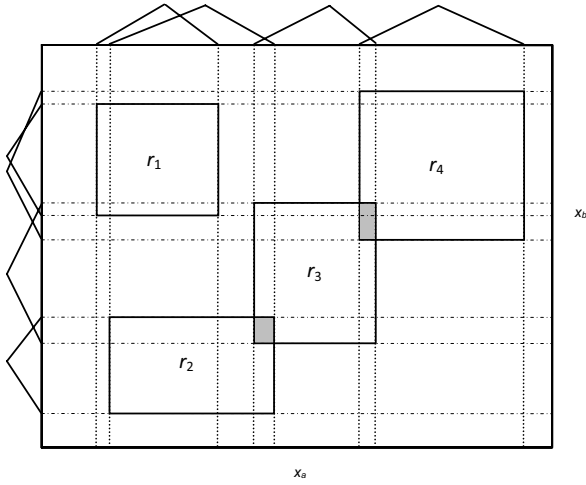


Fig. 5. Rule compression

rule compressions in higher dimensions (less variables are removed).

Let us note, however, that rule compression procedure is not without side-effects and introduces artifacts because compressed rules are much more general than non-compressed ones and cover more area in input space, sometimes conflicting with other rules. This can be observed in Fig. 6, where the following rule base has been obtained.

$$\begin{aligned}
 r_1 : & \text{IF } x_2 \text{ is } A_{21} \text{ THEN } y \text{ belongs to class 1,} \\
 r_2 : & \text{IF } x_1 \text{ is } A_{12} \text{ AND } x_2 \text{ is } A_{22} \\
 & \text{THEN } y \text{ belongs to class 2,} \\
 r_3 : & \text{IF } x_1 \text{ is } A_{13} \text{ THEN } y \text{ belongs to class 3,}
 \end{aligned} \tag{23}$$

Rule r_1 has been compressed in x_2 because assumed $\theta > O_{12}^{(2)}$ and $O_{13}^{(2)} = 0$. In result the rule expands so that it overlaps with r_2 even more than before. If rule r_3 is similarly compressed (in x_1 because $\delta > O_{32}^{(1)}$ and $O_{31}^{(1)} = 0$ - in this comparison original r_1 is used) its overlap with r_2 is slightly increased but more importantly r_3 now also overlaps with r_1 . As the latter overlap is in the area that does not contain any data, numerical performance of the classifier is not influenced, however, it may be when unseen data falls into the shared area.

Let us find out what the algorithm does to the classifiers from Table II. The columns 2-5 and 6-9 in Table III correspond to classifiers of columns 2-4 and 5-7 in Table II. η denotes the compression rate (number of compressed MFs in rules vs. the number of original MFs) and off (if available) contains the inputs that were removed from all rules of the classifier. We can see that in many cases rule compression actually improves accuracy of the classifier, whereas it is particularly interesting that all those minimum-rule classifiers that include inputs 3 and 4, converge to the same solution.

A known issue with the proposed methodology is that it performs poorly on unseen data. This stems from using triangular MFs that have compact supports. Even though rule compression positively influences this property, a more

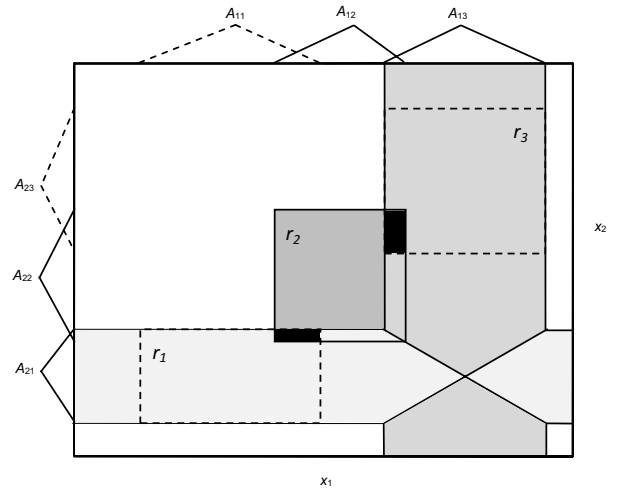


Fig. 6. Artifacts of rule compression. Compressed rules r_1 and r_3 have a "conflict of interests".

TABLE III
IRIS CLASSIFIERS AFTER RULE COMPRESSION

Exp. no.	θ	η (%)	off	ϵ (%)	θ	η (%)	off	ϵ (%)
3	0.30	28.6	-	97.33	0.20	50.0	1	96.00
4	-	-	-	-	0.20	50.0	2	95.33
5	0.30	37.5	-	96.00	0.20	50.0	2	96.00
6	0.01	16.7	-	97.33	0.20	50.0	4	95.33
7	0.05	40.0	-	96.00	0.07	44.4	1	93.33
8	0.05	23.3	-	98.00	0.10	44.4	1	94.67
9	0.02	28.6	-	98.00	0.03	44.4	1	97.33
10	0.036	27.8	-	98.00	0.03	44.4	2	97.33
11	0.02	37.5	2	98.67	0.03	58.3	1,2	97.33

efficient fix to the problem is to replace triangular MFs with double-sided Gaussian ones

$$\mu_{ir}(x_i) = \begin{cases} e^{-\frac{(x_i - b_{ir})^2}{2 \cdot (0.4247 \cdot (b_{ir} - a_{ir}))^2}}, & \text{if } x_i < b_{ir} \\ e^{-\frac{(x_i - b_{ir})^2}{2 \cdot (0.4247 \cdot (c_{ir} - b_{ir}))^2}}, & \text{if } x_i \geq b_{ir} \end{cases} \tag{24}$$

For the following experiments we use only half of data in the training process (every second item of the original data set goes to the checking data set). Table IV shows the obtained accuracies (the number of misclassified samples in training and testing data sets are given as parts of the sum and overall accuracy is given in parentheses) of the four-input classifiers at various stages of simplification before and after the conversion to Gaussian MFs. As we can see, the MF conversion is able to make the classifier a much better generalizer and does not affect classifier performance on training data.

Because of its simplicity, the Iris data set is very popular among classifier developers so there is plenty of material to compare our results to. Table V (where a blank means that the particular value is unknown) collects the accuracy measures available in literature along with our own results.

Note that different authors have used different techniques to determine testing errors, Wang, Wu and Nauck for example

TABLE IV
IRIS CLASSIFIER PERFORMANCE ON UNSEEN DATA

	before conversion	after conversion
initial classifier	1+33 (77.3%)	1+4 (96.7%)
after reduction (8 rules)	1+29 (80.0%)	1+4 (96.7%)
after reduction (3 rules)	4+14 (88.0%)	4+4 (94.7%)
after compression (8 rules)	1+17 (88.0%)	1+5 (96.0%)
after compression (3 rules)	4+2 (96.0%)	4+2 (96.0%)

TABLE V
COMPARISON OF RESULTS FOR IRIS DATA

	N	No. of MFs	R	ϵ_{tr}	ϵ_{test}
Wang et al. [12]	4	11	3	99.3	97.4
Wu et al. [13]	4	9	3	-	96.2
Shi et al. [14]	4	12	4	98.0	-
Ishibuchi et al. [15]	4	7	5	-	98.0
Tong et al. [16]	-	12	3	98.0	95.5
Xing et al. [17]	2	6	3	98.0	96.7
Russo [18]	4	18	5	100.0	-
Nauck et al. [19]	2	6	5	97.3	96.7
Abonyi et al. [20]	-	4	3	-	96.1
This paper	2	5	3	97.3	96.0

used the same approach as we here (original data set split into two equal sets of training and testing data), Xing, Tong, Abonyi used 5-fold and Ishibuchi 10-fold cross validation (both of which should generally yield lower testing errors). Moreover, note that unlike to our approach, usually a time-consuming optimization algorithm is involved in getting cited results, typically some kind of evolutionary algorithm. In this context, the performance of proposed algorithm can be counted as favorable.

VI. CLASSIFICATION OF OTHER DATA SETS

A. Wine data

The 178-element wine dataset collects data of 3 classes (59, 71, and 48 entries for each class) of wine from various places in Italy. There are 13 features corresponding to the values from chemical analysis, including Alcohol, Malic acid, Ash, Alcalinity of ash, Magnesium, Total phenols, Flavanoids, Nonflavanoid phenols, Proanthocyanins, Color intensity, Hue, OD280/OD315 of diluted wines and Proline.

Having learned from the experiments with Iris data we first try to obtain a classifier with the highest number of input variables possible. The result is a zero-error classifier with 8 rules. Via rule base reduction ($\delta = 0.5$) and rule compression ($\theta = 0.01$) we obtain a 6-rule zero-error classifier, of which, however, 5 input variables (1, 4, 5, 6 and 9) can be immediately removed as they are cancelled out by rule compression. Of these 6 rules, two classes (1 and 3) are determined by only a single rule with 4 MFs per each. For example, class 3 wines are characterized by flavanoids under 1.5, color intensity above 4, hue under 1 and OD280/OD315 under 2.5. The remaining four rules are for class 2 and have 4 or 5 premises. The total number of input premises is 28.

A minimal rule classifier can be obtained with 96.6% accuracy (6 misclassified samples), using only six inputs (1, 7, 10-13) and 15 MFs. This is as basic as it gets.

The listing in Table VI indicates that these results are in the same ballpark with those obtained by various researchers.

TABLE VI
COMPARISON OF RESULTS FOR WINE DATA

	N	No. of MFs	R	ϵ_{tr} (%)
Setnes et al. [21]	9	21	3	98.3
Wang et al. [12]	13	34	3	99.4
Ishibuchi et al. [15]	13	15	8	100.0
Roubos et al. [7]	5	10	3	99.4
Chang et al. [22]	6	13	5	98.9
this paper	13(8)	28	6	100.0

B. Wisconsin breast data

The Wisconsin Breast Cancer data set contains 699 patterns in two classes; 458 patterns belong to the benign class, the other 241 patterns are the malignant class. Each pattern is described by nine features. Since 16 of the data set have missing values, we (as many before us) use 683 patterns to evaluate the proposed classifier.

First we obtain a 12-rule classifier on half of the data achieving 100% accuracy on training data. On test data set 50 samples are misclassified which means that overall accuracy is 92.7%. After converting the MFs to Gaussian ones, accuracy rate increases to 96.9%. Of 21 misclassified patients, only one is false positive, which is a good news because being false diagnosed negative is far less dangerous for the patient.

Using different values of δ , 6-, 3- and 2-rule classifiers were obtained with accuracies 94.6%, 96.2% and 96.2%, respectively (which after the conversion transformed to 96.8%, 96.3%, and 96.3%, respectively).

However, rule compression does not work that well with this data set because with the 2-rule system no compression was possible without performance loss and with the 3-rule system only 23.3% (equivalent to 23 MFs) compression was established. With the 6-rule system we can choose either 94.7/96.6% (38 MFs) or 28.3%, 95.1/97.1% (43 MFs) classifiers. However, with this 97.1% accuracy the figures of false positives and false negatives are reversed (14 vs. 6).

Our most compact classifier is compared to several others found from literature in Table VII.

TABLE VII
COMPARISON OF RESULTS FOR BREAST CANCER DATA

	No. of MFs	R	ϵ_{tr}
Wang et al. [12]	18	2	96.3
Nauck et al. [19]	18	4	96.5
Ishibuchi et al. [15]	-	4	97.4
Abonyi et al. [20]	3	2	96.8
Chang et al. [22]	4	3	96.5
this paper	20	2	96.3

VII. FINAL THOUGHTS

When we set out with current project, our goal was to design a classification algorithm that would be fast and that would identify classifiers that could be validated by linguistic

analysis. Its accuracy was also a solid but still a secondary concern. Our results, however, indicate that the accuracy of the proposed three-step algorithm is competitive enough.

The idea for using the k-means in the classification in broader sense and as the basis for building fuzzy rule-based classifiers is, of course, not a new one. The implementations, however, vary in details. For example, the k-means based rule generation in [24] differs from ours in one aspect as the input MFs in their classifier are deemed to be symmetrical. The performance of the algorithm is demonstrated on only two simple data sets so the comparison material how this restriction affects the performance is insufficient.

In another example [23], the k-means is only used for prototyping the MFs and highly similar MFs are immediately merged. Subsequent rule generation is grid-based and the inference scheme employs a rounded off centre-of-gravity defuzzification. As this implementation is aimed at image segmentation we cannot say how it performs in direct comparison with our method but the number of rules tends to be high (around 35) even with few input variables and there seems to be a lot of computational overhead.

In fact, MF similarity based simplification is symptomatic to most rule base reduction approaches that can be found from literature. Typical in this sense is the method presented in [21] in which multivariable MFs are identified in the product space by a fuzzy clustering method. The univariate fuzzy sets $A_{i,r}$ are then obtained by projecting the clusters onto input axes and estimated by parametric functions. A similarity driven simplification is applied to merge similar fuzzy sets (special case: if all fuzzy sets for a feature are similar to the universal set then this feature can be removed from the classifier). Even not taking into account that similarity of MFs is not a real indicator of redundancy - MFs similar just in one dimension belong more likely than not to the rules that are far apart in product space - it is easy to see that doing the merge directly in input space (with a subsequent MF update) saves us from computational overhead.

Perhaps the biggest problem with the k-means is that it is known to converge to suboptimal solutions if the initial centroids (random initialization!) are unfortunate. During the years, many methods have been designed to deal with this problem, however, there is no definite solution yet, so finding or choosing one would easily be our next step to improve the results. On the other hand, current methodology is not strictly married to the k-means, so any algorithm that is able to yield compact and distinct clusters from data and is less sensitive to initial conditions could be considered as a viable candidate.

Rule compression is possibly the most important aspect in interpretability improvement and deserves a further treatment as well. Very often a priori feature selection procedure is carried out to shorten the rules, be it an application of Fisher interclass separability criterion in [7] or some other method. Rule compression without a priori feature selection in these days is usually carried out by evolutionary algorithms (examples of which can be found in several cited papers) - the path which we have avoided taking being drawn away by

long execution times of evolutionary algorithms.

REFERENCES

- [1] L. I. Kuncheva, *Fuzzy Classifier Design*, Springer-Verlag, Heidelberg, 2000.
- [2] J. C. Bezdek, *Pattern Recognition with Fuzzy Objective Function*, Plenum Press, New York, 1981.
- [3] D. E. Gustafson and W. C. Kessel, "Fuzzy clustering with a fuzzy covariance matrix," *Proc. IEEE Conf. Decision and Contr. (CDC '78)*, San Diego, Calif., pp. 761-766, 1979.
- [4] I. Gath and A. B. Geva, "Unsupervised optimal fuzzy clustering," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 7, pp. 773-781, 1989.
- [5] A. Riid and E. Rüstern, "Transparent fuzzy systems and modeling with transparency protection," *Proc. IFAC Symp. Artificial Intelligence in Real Time Control*, Budapest, pp. 229-234, 2000.
- [6] A. Riid and E. Rüstern, "Fuzzy logic in control: truck backer-upper problem revisited," *Proc. IEEE Int. Conf. Fuzzy Syst.*, Melbourne, Australia, vol. 1, 513-516, 2001.
- [7] H. Roubos, M. Setnes and J. Abonyi, "Learning Fuzzy Classification Rules from Data," *Inform. Sci.*, vol. 150, no. 1-2, pp. 77-93, 2003.
- [8] C. Mencar, C. Castiello and A. M. Fanelli, "Interpretability Assessment of Fuzzy Rule-Based Classifiers," *Proc. 8-th Int. Workshop Fuzzy Logic and Applications*, Palermo, Italy, pp. 155-162, 2009.
- [9] S. P. Lloyd, "Least squares quantization in PCM," *IEEE Trans. Information Theory*, vol. 28, no. 2, pp. 129-137, 1982.
- [10] K. V. Mardia, J. T. Kent and J. M. Bibby, *Multivariate Analysis*, Academic Press, 1980.
- [11] S. L. Chiu, "Fuzzy Model Identification Based on Cluster Estimation," *Journal of Intelligent and Fuzzy Systems*, vol.2, pp. 267-278, 1994.
- [12] J. S. Wang and G. C. S. Lee, "Self-adaptive neuro-fuzzy inference system for classification application," *IEEE Trans. Fuzzy Syst.*, vol. 10, pp. 790-802, 2002.
- [13] T. P. Wu and S. M. Chen, "A new method for constructing membership functions and fuzzy rules from training examples," *IEEE Trans. Syst. Man Cybern. Part B*, vol. 29, pp. 25-40, 1999.
- [14] Y. Shi, R. Eberhart and Y. Chen, "Implementation of evolutionary fuzzy system," *IEEE Trans. Fuzzy Syst.*, vol. 7, pp. 109-119, 1999.
- [15] H. Ishibuchi, T. Nakashima and T. Murata, "Three-objective genetic-based machine learning for linguistic rule extraction," *Information Science*, vol. 136, no. 1-4, pp. 109-133, 2001.
- [16] S. H. Tong, Y. Shen and Z. Y. Liu, "Approach to construct fuzzy classification system with clustering," *Control and decision*, vol. 16 pp. 737-740, 2001.
- [17] Z.-Y. Xing, Y. Zhang, L.-M. Jia and W.-L. Hu, "Construct Interpretable Fuzzy Classification System Based on Fuzzy Clustering Initialization," *Int. J. Inf. Technology*, vol. 11, no. 6, pp. 91-107, 2005.
- [18] M. Russo, "Genetic fuzzy learning," *IEEE Trans. Evol. Comp.*, vol. 4, pp. 259-273, 2000.
- [19] D. Nauck, U. Nauck and R. Kruse, "Generating Classification Rules with the Neuro-Fuzzy System NEFCLASS," *Proc. Biennial Conf. North American Fuzzy Information Processing Society (NAFIPS)*, Berkeley, pp. 466-470, 1996.
- [20] J. Abonyi, J. A. Roubos and F. Szeifert, "Data-driven generation of compact, accurate, and linguistically sound fuzzy classifiers based on a decision-tree initialization," *Int. J. Approx. Reason.*, vol. 23, pp. 1-21, 2003.
- [21] M. Setnes and H. Roubos, "GA-Fuzzy Modeling and Classification: Complexity and Performance," *IEEE Trans. Fuzzy Syst.*, vol. 8, no. 5, pp. 509-522, 2000.
- [22] X. Chang and J. H. Lilly, "Evolutionary Design of a Fuzzy Classifier From Data," *IEEE Trans. Syst., Man, Cybern.*, vol. 34, No. 4, pp. 1894-1906, 2004.
- [23] Z. Chi and H. Yan, "Image segmentation using fuzzy rules derived from K-means clusters," *J. Electronic Imaging*, vol 4, No. 2, pp. 199-206, 1995.
- [24] C.-C. Wong, C.-C. Chen and S.-L. Yeh, "K-Means-Based Fuzzy Classifier Design," *Proc. IEEE Int. Conf. Fuzzy Syst.*, San Antonio, TX, vol. 1., pp. 48-52, 2000.