Evolutionary Reduction of Fuzzy Rule-Based Models

Witold Pedrycz, Kuwen Li and Marek Reformat

Abstract In the design of fuzzy rule-based models we strive to develop models that are both accurate and interpretable (transparent). The approach proposed here is aimed at the enhancement of transparency of the fuzzy model already constructed with the accuracy criterion in mind by proposing two modifications to the rules. First, we introduce a mechanism of reduction of the input space by eliminating some less essential input variables. This results in rules with the reduced subspaces of input variables making the rules more transparent. The second approach is concerned with an isolation of input variables: fuzzy sets defined in the n-dimensional input space and forming the condition part of the rules are subject to a decomposition process in which some variables are isolated and interpreted separately. The reduced dimensionality of the input subspaces in the first approach and the number of isolated input variables in the second one are the essential parameters controlling impact of enhanced transparency on the accuracy of the obtained fuzzy model. The two problems identified above are of combinatorial character and the optimization tasks emerging there are handled with the use of Genetic Algorithms (GAs). A series of numeric experiments is reported where we demonstrate the effectiveness of the two approaches and quantify the relationships between the criterion of accuracy and interpretability.

W. Pedrycz (✉) · K. Li · M. Reformat
Department of Electrical and Computer Engineering, University of Alberta,
Edmonton, AB T6R 2V4, Canada
e-mail: wpedrycz@ualberta.ca

K. Li
e-mail: kuwen@ualberta.ca

M. Reformat
e-mail: reformat@ualberta.ca

W. Pedrycz
Faculty of Engineering, Department of Electrical and Computer Engineering,
King Abdulaziz University, Jeddah 21589, Saudi Arabia

W. Pedrycz
Systems Research Institute, Polish Academy of Sciences, Warsaw, Poland

D.E. Tamir et al. (eds.), *Fifty Years of Fuzzy Logic and its Applications*,
Studies in Fuzziness and Soft Computing 326,
DOI 10.1007/978-3-319-19683-1_23

Keywords Rule-based models · Interpretability · Reduction · Isolation of variables · Fuzzy clustering · Curse of dimensionality

1 Introductory Notes

In the construction of fuzzy rule based systems, we have been witnessing a wealth of design strategies and detailed algorithms involving the technology of Evolutionary Computing and neurocomputing. Just recent developments reported in this realm can be found in a series of studies [1–3, 5, 6, 8]. Predominantly, the development of fuzzy models is guided by the criterion of accuracy. Another fundamental criterion being at the heart of fuzzy modeling is interpretability (transparency) of resulting fuzzy models. This criterion is central to fuzzy models however its multifaceted nature requires a thorough formulation and a detailed quantification of essential aspects of interpretability. Subsequently, it calls for engaging advanced optimization techniques supporting the realization of the ensuing design.

The concept of interpretability of fuzzy rule-based models has been around for several decades and attracted a significant deal of attention. The transparency of fuzzy models is one of the outstanding and important features of fuzzy models. In contrast to the criterion of accuracy, whose quantification is relatively straightforward and easy to come up with performance indexes, transparency of fuzzy rules is more difficult to describe. What makes the fuzzy rule-based easier to interpret and comprehend is still an open issue. It is quite subjective to assess and in one way or another invokes a factor of subjective judgment given that a human user is ultimately involved in the evaluation process. What also becomes apparent, is a multifaceted nature of the problem and a multitude of various approaches supported by various optimization technologies including evolutionary optimization. When it comes to the main factors worth considering when discussing a concept of interpretability, we can enumerate a list of factors that may be involved in the reduction process:

- number of rules forming a rule base of the model,
- number of sub-conditions (input variables) forming a condition part of a given rule,
- number or rules and the number of input variables,
- complexity of local regression models forming the conclusion part of the rules (in case of Takagi-Sugeno model)
- interpretability of a family of fuzzy sets formed in the input space for individual variables.

As a result, given this diversity of possible ways of reduction of rules, it is difficult to quantify the effect of reduction. For instance, it is not always clear if it would be better to have a larger number of simple rules (whose condition parts are linear functions) or a smaller number of rules of a more complex conclusion parts (say, those of polynomial form).

The reader may refer to a large body of studies devoted to the issue of interpretability, cf. [3, 10]. Quite often, given the combinatorial nature of the reduction problems, Genetic Algorithms are used in the optimization process, see [9]. There are also more specialized algorithmic vehicles to support the reduction process such as e.g., singular value decomposition [14] however one has to be cognizant as to the interpretability of the reduced rules.

As to the overall strategy of rule reduction and interpretability enhancement, there are two general design strategies: either the reduction is completed once the model has been constructed or the reduction mechanism of rule-based modeling is incorporated into the design process from its very beginning.

The objective of the study is to pursue a fundamental issue of building more transparent and user-centric fuzzy rule-based models by starting from an already designed fuzzy model. The intent is to make the rules more readable in two different ways: (a) by reducing the input space (the number of antecedents) of the individual rules, and (b) by isolating input variables completed for the input variables treated *en block* in the condition parts of the rules.

In both these fundamental scenarios we can establish and quantify a tradeoff between a gradual reduction of accuracy (which is inevitable when realizing any of the reduction mechanisms of the rules and enhancing its intensity) and the increased interpretability of the rules. The presentation of the material is structured in the following way. We briefly revisit the essentials of Takagi-Sugeno rule- based systems, stressing the proposed design approach in which we use fuzzy clustering (Sect. 2). The reduction of input space and a formation of input subspaces for each rule is introduced in Sect. 3; in the same section we also present an optimization process realized with the use of genetic algorithm. The second approach to the enhancement of the interpretability of the rule – an isolation of input variables is discussed in Sect. 5. Detailed experimental studies are reported in Sects. 3 and 6.

In the study, we experiment with a number of publicly available datasets coming from eight datasets from UCI Machine learning repository and DELVE repository; their main characteristics are listed in Table 1.

Table 1 Main characteristics of datasets used in the experiments (number of data and dimensionality –number of input variables)

Dataset	Number of input variables	Number of data	Origin of the data
Abalone	8	4177	UCI Machine learning repository (http://archive.ics.uci.edu/ml/)
Auto MPG	7	392	UCI Machine learning repository
Boston Housing	13	506	UCI Machine learning repository
Computer Activity	21	8,192	DELVE repository (http://www.dcc.fc.up.pt/~ltorgo/Regression/DataSets.html)
Concrete strength	8	1030	UCI Machine learning repository
Forest fires	12	517	UCI Machine learning repository
Red wine quality	11	1599	UCI Machine learning repository
White wine quality	11	4898	UCI Machine learning repository

2 Fuzzy Rule-Based Models: An Overview and Main Design Issues

Our point of departure of the reduction processes of the rules is a "standard" Takagi-Sugeno fuzzy model comprising c rules coming in the following form

$$-\ if\ \boldsymbol{x}\ is\ A_i\ then\ y = f_i(\boldsymbol{x}, \boldsymbol{a}_i) \tag{1}$$

for $i = 1, 2, .., c$ where $\boldsymbol{x} \in \boldsymbol{R}^n$. A_i is a fuzzy set defined in the n-dimensional space while f_i is the corresponding local model (linear or nonlinear) endowed with its parameters $\boldsymbol{a}_i$ forming the conclusion part of the i^{th} rule. The design of such models is well reported in the literature and as usual consists of the two main steps, namely (a) a construction of condition parts (through clustering of input data done in the input space) and (b) estimating parameters of the linear models (which leads to the problem of linear regression). The number of rules (c) is determined by monitoring the behavior of the model on the training and testing data. The rules in the form (1) come with several essential properties. The condition part is a fuzzy set expressed in $\boldsymbol{R}^n$ making the rules concise, which helps avoid a curse of dimensionality we are commonly faced with in rule based systems with a higher number of input variables. In the case of treating all input variables at the same time, the number of rules becomes small (c) and the rule base itself is compact. Unfortunately, the interpretability could be negatively impacted as no individual variables in the condition part are treated and visualized separately.

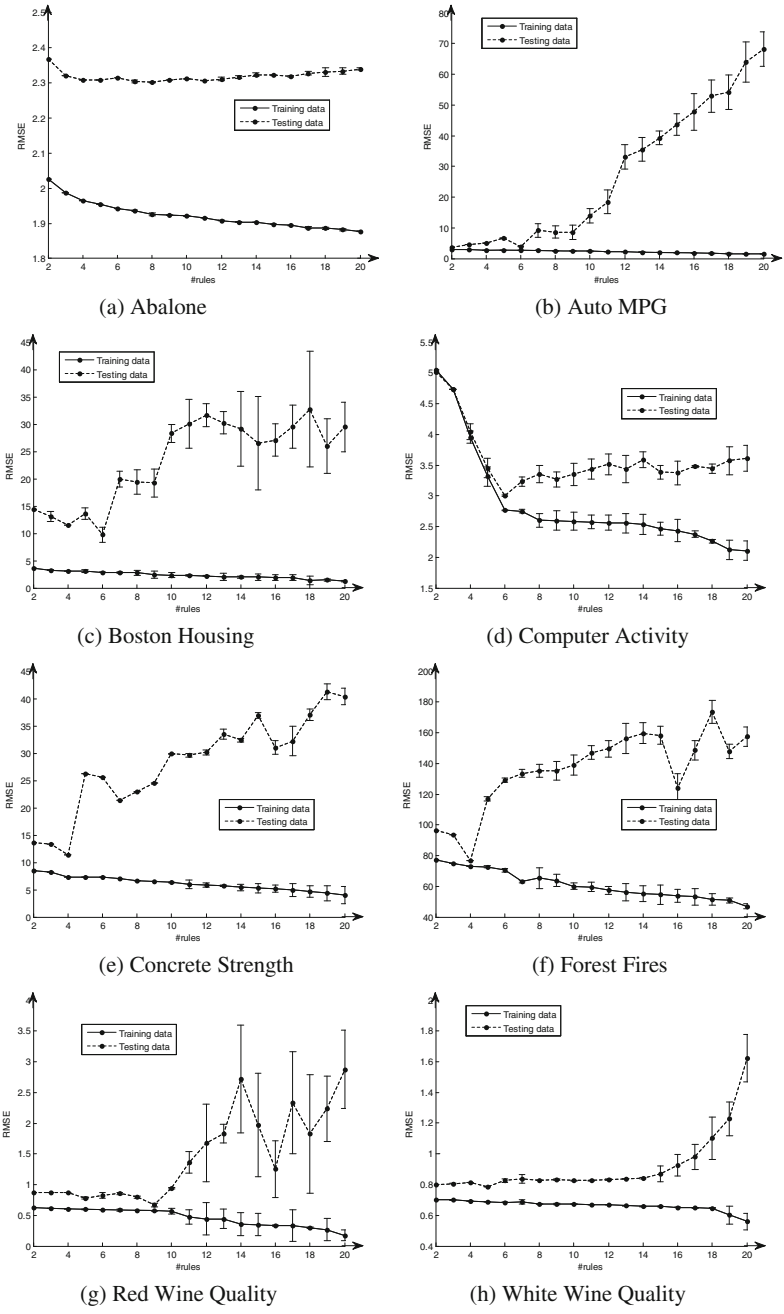
When it comes to the quantification of the accuracy of the fuzzy model (1) its performance is commonly expressed by the RMSE index computed for the training set

$$\sqrt{\frac{1}{N}\sum_{k=1}^{N}(\mathrm{FM}(\mathbf{x}_k) - target_k)^2} \tag{2}$$

where N denotes the number of data in the training set. In the same way, quantified is the performance of the constructed model on the testing data (consisting of M data points)

$$\sqrt{\frac{1}{\mathrm{M}}\sum_{\mathrm{k}=1}^{\mathrm{M}}(\mathrm{FM}(\mathbf{x}_\mathrm{k}) - \mathrm{target}_\mathrm{k})^2} \tag{3}$$

Proceeding with the data summarized in Table 1, the performance of the corresponding models visualized versus the number of rules is illustrated in a series of figures shown below, Fig. 1. The results are reported both for the training and testing data. In the design, we use a standard version of the Fuzzy C-Means (FCM) [4] with the fuzzification coefficient (m) set to 2.0. The algorithm was run for 10 iterations (more specifically, we completed 10 runs with different splits of data into training/testing data)



(a) Abalone (b) Auto MPG

(c) Boston Housing (d) Computer Activity

(e) Concrete Strength (f) Forest Fires

(g) Red Wine Quality (h) White Wine Quality

Fig. 1 Performance of fuzzy models versus the number of rules reported for the training and testing data for datasets (Table 1)

For the training sets, there is a general tendency of having lower values of the RMSE with the increase of the number of rules. The performance reported on the testing sets points at the memorization effect where the some models tend to lose their generalization capabilities. By eyeballing these plots, we choose a suitable number of rules (clusters) where sound approximation abilities come hand in hand with the generalization of the models. The values selected in this way are collected in Table 2.

Table 2 Number of rules of fuzzy models constructed for the corresponding data

Data name	Selected number of rules
Abalone	7
Auto MPG	6
Boston Housing	6
Computer Activity	6
Concrete strength	4
Forest fires	4
Red wine quality	9
White wine quality	5

3 Reduction of Input Subspaces in Rule-Based Models

The essence of the enhancement of interpretability of the rules is accomplished by reducing the number of input variables standing in the condition parts of the rules. The reduced rules are concisely described in the form

$$- \textit{if } \boldsymbol{x} \textit{ is } [A_i]_{Xi} \textit{ then } y = f_i(\boldsymbol{x}, \boldsymbol{a}_i) \tag{4}$$

where the symbol $[]_{Xi}$ stresses the fact that the fuzzy set A_i is now effectively confined to the reduced input space $\boldsymbol{X}_i \subset \boldsymbol{R}^n$ where some original input variables have been removed. In other words, dim $(\boldsymbol{X}_i) = n_i < n$.

The computing of the activation level of A_i positioned in this new reduced space is realized as follows

$$[\mathrm{A_i}]_{\mathbf{X_i}}(\mathbf{x}) = 1 / \sum_{\mathrm{j}=1}^{\mathrm{c}} \left(\frac{\cdot \left\| \underline{\mathbf{x}} - \underline{\mathbf{v}_\mathrm{i}} \right\|_{\aleph_\mathrm{i}}}{\cdot \left\| \underline{\mathbf{x}} - \underline{\mathbf{v}_\mathrm{j}} \right\|_{\aleph_\mathrm{i}}} \right)^{\frac{2}{m-1}}, \tag{5}$$

for $i = 1,2,..,\ c;\ m > 1$ where the computations of the distance are realized in the reduced input space $\boldsymbol{X}_i$.

The reduction of the rules can be quantified in terms of a reduction factor ν, which relates with the number $n*c$ (expressing an overall number of variables across all the rules) in the following way

$$p = n(n*c). \tag{6}$$

where p represents a reduced number of input variables used in c rules.

The reduction of the input variables existing in the new, more interpretable rules is done via engaging the optimization capabilities of Genetic Algorithms (GAs). More specifically, we optimize a matrix of allocation of input variables $W = [w_{ij}]$ with c rows and n input variables. The p largest entries of W are selected giving rise to a binary 0–1 matrix. The p entries with the largest values are set to 1 while the remaining ones are suppressed to zero. Each row of the matrix formed in this way identifies the variables to be used in the corresponding reduced rule. If all entries of the i^{th} row of W are equal to zero, this entails that the corresponding rule does not exist in the reduced set of rules.

The process of identifying which input variables should be kept in the antecedents of rules is translated into an optimization problem. Its solution is obtained through evolutionary optimization, namely Genetic Algorithm (GA) [7]. GA uses elements of natural selection to determine the best solution to a problem by minimizing a certain fitness function capturing the essence of the optimization problem. The best solution is obtained via selecting and modifying a population of potential solutions (chromosomes). A main flow of GA computing is outlined in Table 3.

Table 3 A flow of optimization realized by genetic algorithm

Parameters of genetic algorithm:
iter – number of generations
Z – size of population
p_c – crossover rate
p_m – mutation rate
Algorithm
1 *Initialization:*
2 Random generation of chromosomes c_k using uniform distribution in [0,1]), for $k = 1, 2, \ldots, Z$
3 *Iterate (*
4 Calculate the fitness value for each chromosome c_k
5 Select the candidates for the next generation based on their fitness values
6 Retain the overall best chromosome so far, say c_{best}
7 Apply the crossover and mutation operations based on the crossover (p_c) and mutation (p_m) rates
8 *Until* the number of iterations does not exceed the predetermined limit, iter*

The two aspects of GA that require special attention and directly impact the quality of results of the optimization process are: 1) construction of a suitable fitness function, and 2) mapping a solution to the problem onto a structure of the chromosome.

A fitness function is used to evaluate possible solutions. Fitness values associated with solutions (chromosomes) represent their ability to solve a given problem. The values are used during selection of chromosomes for further processing (Table 3, line 5). The fitness function has to reflect the objective of optimization process and ensure that the obtained fitness values are adequate for mechanisms of selection, i.e., the fitness values should be such that a selection process leads to a diversified set of potential solutions [11]. In the case of our problem of selecting input attributes that should be kept in the rules, the fitness function evaluates quality of fuzzy rules that model a given dataset. The form of the fitness function used in the optimization is given by (2).

As noted earlier, each chromosome represents a solution to the considered problem. It consists of simple elements, called genes, that can assume values 1 or 0 (Binary Coded GA), or any real numbers from a specified range (Real Coded GA). A chromosome with such a simple structure, called genotype, has to represent a solution to the problem, i.e., it should be mapped into a form called phenotype that is adequate for a given problem domain.

For the optimization problem considered in the paper, we use a Real Coded GA (RCGA). The size of chromosome, i.e., a number of genes *gsize*, is equal to the product of a number of rules c and a number of input attributes n of a given dataset: $gsize = cn$. Therefore, each chromosome contains information which input attributes are present in each rule. A single gene is a real number between 0 and 1. The translation of such a chromosome (genotype) into a solution to our problem (phenotype) occurs in the following way. The user has to determine a number p of input attributes or isolated attributes (Sect. 5) that should be kept in all c rules. This means that among all genes of a chromosome only p of them should be used to construct c rules. The selection of these genes is done via sorting all genes based on their values (between 0 and 1) and identifying the first p of them. The process is presented in Fig. 2.

Once an initial population of chromosomes c_k has been generated, RCGA starts an iteration process. A single iteration consists of calculating fitness values (Table 3, line 4), selecting best chromosomes (line 6), and performing crossover and mutation operations on selected chromosomes (line 7). All this constitutes a single generation. After executing a number of generations, GA provides an optimal solution in a form of a chromosome that gives the highest value of the fitness function.

The purpose of crossover is to exchange information between chromosomes. In its simplest form it means exchange of genes between a pair of randomly selected, with the probability p_c given by the user, chromosomes. For our RCGA, a linear crossover is applied [13]. The linear crossover generates three offspring (new chromosomes): $O_k = (o_1^k, \ldots, o_i^k, \ldots o_n^k), k = 1, 2, 3,$ where o_i^k represents the ith gene of the kth chromosome. The genes of offspring are built from the genes of two parents. Let us assume that the parents are: $C_k = (c_1^k, \ldots, c_i^k, \ldots c_n^k),\ \mathrm{k} = 1, 2$. The values of genes of the offspring are calculated in the following way:

$$o_i^1 = \frac{1}{2}c_i^1 + \frac{1}{2}c_i^2,\ o_i^2 = \frac{3}{2}c_i^1 - \frac{1}{2}c_i^2 \text{ and } o_i^3 = -\frac{1}{2}c_i^1 + \frac{3}{2}c_i^2 \tag{7}$$

Each of the offspring is evaluated, i.e., a fitness value is determined for each of them. The two most promising offspring are selected to substitute their two parents in the population.

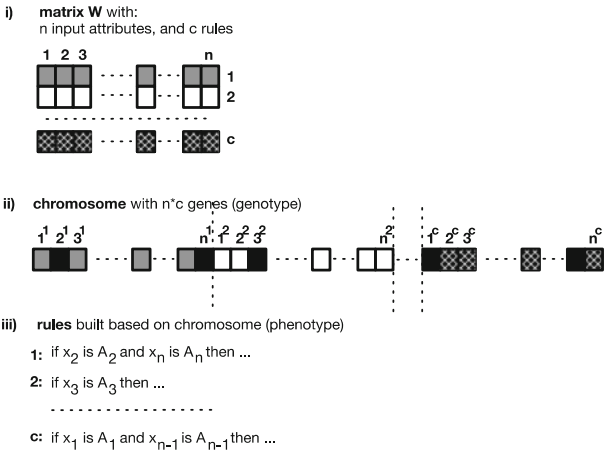


Fig. 2 GA chromosome: (i) original matrix of rules, (ii) a single chromosome built from the whole matrix, (iii) rules built based on this chromosome (there are p black boxes representing attributes selected for c rules)

The mutation operator modifies genes of a single chromosome in order to introduce diversity to the population. The frequency of modifications is controlled by the mutation probability p_m provided by the user. Mutation ensures that a search process covers the whole space of possible solutions. In the case of RCGA, the

Muhlenbein's mutation is adopted [12]. For a given parent chromosome $C_k = (c_1, \ldots, c_i, \ldots c_n)$, the gene values of a new – mutated – chromosome are calculated as $c_i^{'} = c_i \pm rang_i \bullet g$, where $rang_i$ defines the mutation range, and it is normally set to $0.1*(b_i - a_i)$, where b_i and a_i are the maximum and the minimum of c_i, respectively. The sign + or − is chosen with a probability of 0.5, while $g = \sum_{k=0}^{15} a_k 2^{-k}, a_k \in \{0, 1\}$ is randomly generated with $p(a_i = 1) = \frac{1}{16}, i = 0.15$.

The operations on a single chromosome are implemented in such a way that at least one input attribute is kept in each rule.

For reduction of input spaces, if p is a number of input attributes to be kept, the top p genes will be used to calculate the model output. The rest *gsize-p* attributes will be omitted in the calculation.

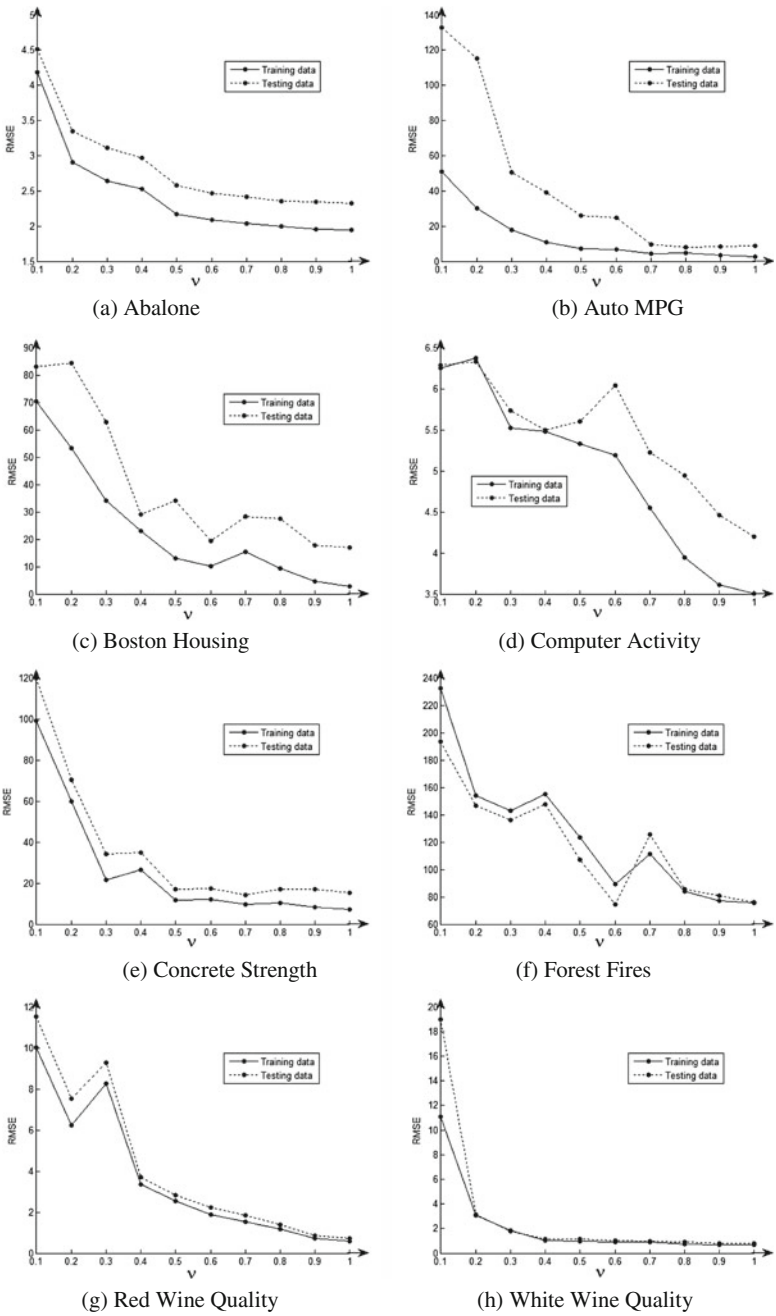
For isolation of attributes (Sect. 5), if p is a number of isolated attributes in each rule, the top p attributes in each rule (row of the matrix W) will be marked as isolated. The rest $n - p$ attributes will be treated as the remaining group of attributes. Thus, the total number of isolated attributes is pc.

4 Experimental Studies

In the following experiments we present how the reduction of the rules proceeds and how the reduced, more interpretable rules perform. We use different values of ν and report the corresponding values of the RMSE for the training and the testing data. The GA used a population of 100 individuals and was run for 100 generations. The crossover rate was set to 0.8 while the mutation rate was equal to 0.1. The choice of these numeric values was a result of some preliminary experimentation. The results are quantified by reporting the RMSE values obtained for different values of the reduction index; refer to Fig. 3.

It becomes apparent (and intuitively anticipated) that lower values of ν result in higher RMSE values. The detailed behavior varies across data with regard to how far the rules can be reduced and how the differences shape up for the training and testing data. For example, the reduction could be made quite substantial not compromising the performance of the model as this becomes present in case of abalone, auto, concrete, and white wine. In some case, we witness a phenomenon of increased generalization abilities of the model (lower differences of the RMSE for the training and testing data for lower values of ν). Figure 4 illustrates the performance of the GA for some selected data; most of the improvement is visible at the beginning of the optimization (first 20–30 generations).

The detailed results of reduction of the number of variables in the rules are contained in Fig. 5. The shaded regions identify the input variables being retained in the corresponding rules. This offers a better view as to which input variables can be dropped and points at a sequence of the variables, which have been eliminated.



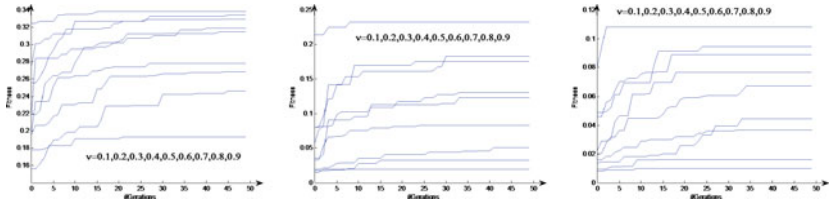
(a) Abalone (b) Auto MPG

(c) Boston Housing (d) Computer Activity

(e) Concrete Strength (f) Forest Fires

(g) Red Wine Quality (h) White Wine Quality

Fig. 3 RMSE values of the reduced rule-based models versus reduction level ν



(a) Abalone (b) Auto MPG (c) Concrete Strength

Fig. 4 Values of fitness function reported in successive GA generations for ν ranging from 0.1 to 0.9 with step 0.1 show direction of ν

5 Isolation of Input Variables

To enhance the transparency of the rules, we express the fuzzy set A_i as a Cartesian product of a single *isolated* fuzzy set defined in $\boldsymbol{R}$ and a *relational* remainder expressed in $\boldsymbol{R}^{n-1}$. In other words, we form the expression describing the condition part as follows

$$A_i^{\wedge}(x_j) \times A_i^{\wedge} \sim (x^{\sim}) \tag{8}$$

where $A_i^{\wedge}$ is a fuzzy sets defined in $\boldsymbol{R}$ and $A_i^{\sim}$ is expressed in $\boldsymbol{R}^{n-1}$. Then the rules of the form read as follows

$$-\textit{if } x_j \textit{ is } A_i^{\wedge}(x_j) \textit{ and } x^{\sim} A_i^{\sim}(x^{\sim}) \textit{ then } y \textit{ is } f_i(x, a_i) \tag{9}$$

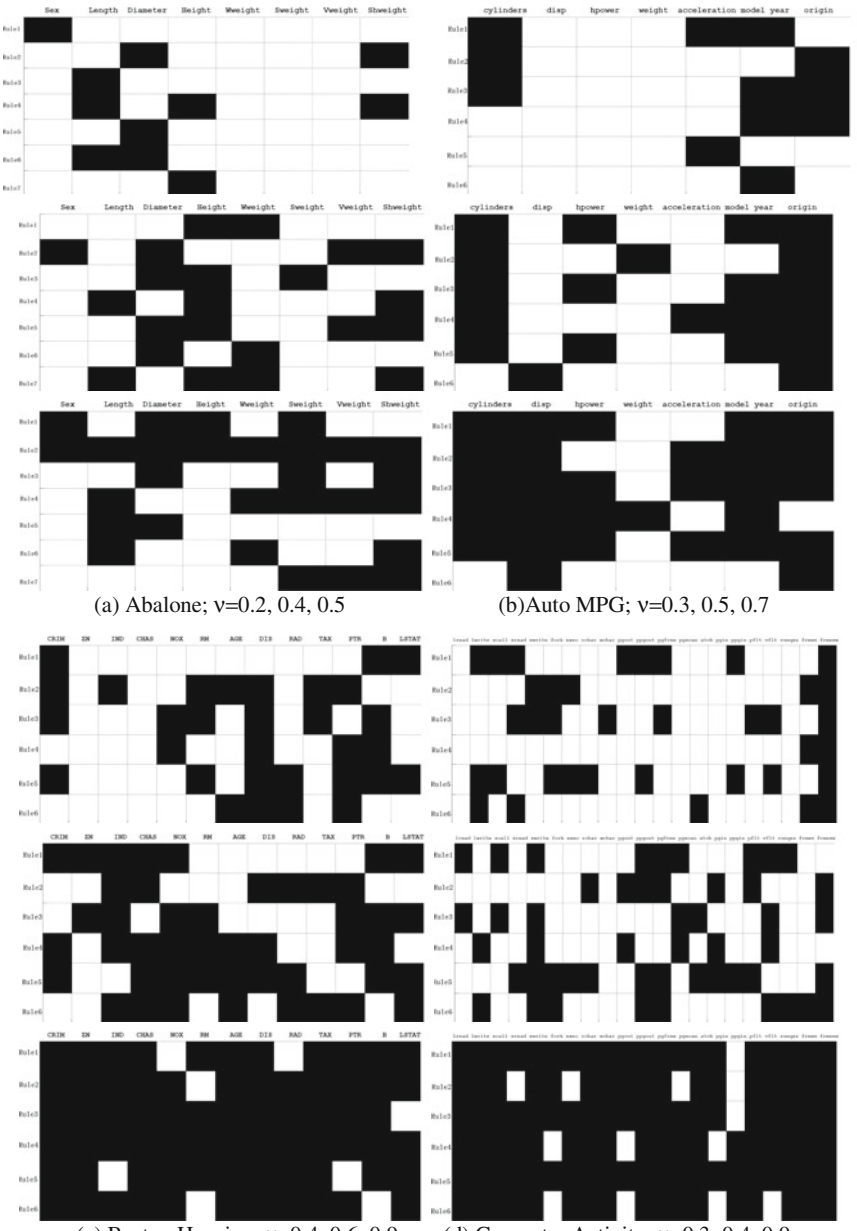
Here a certain input variable (*j*th one) has been selected to be isolated. The term isolation pertains to the fact that a certain variable has been chosen and subsequently a fuzzy set isolated from the fuzzy set A_i is treated separately and thus becomes more visible and interpretable.

Obviously, the above Cartesian product is not identical to the original A_i that is the following holds

$$A_i \neq (A_i^{\wedge} \times A_i^{\sim}) \tag{10}$$

In terms of membership functions this means that the following relationship holds

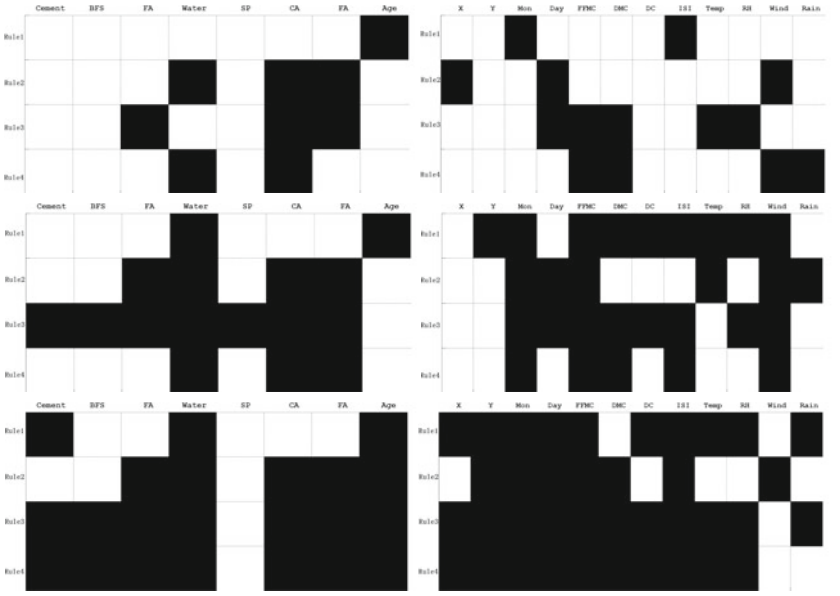
$$A_i(\boldsymbol{x}) \neq \min(Ai^{\wedge}(xj), Ai^{\sim}(x^{\sim})) \tag{11}$$



(a) Abalone; ν=0.2, 0.4, 0.5 (b)Auto MPG; ν=0.3, 0.5, 0.7

(c) Boston Housing; ν=0.4, 0.6, 0.9 (d) Computer Activity; ν=0.3, 0.4, 0.9

Fig. 5 Visualization of reduced rules: shaded regions identify the input variables being retained



(e) Concrete Strength ν=0.3, 0.5, 0.7 (f) Forest Fires ν=0.3, 0.6, 0.8

(g) Red Wine Quality; ν=0.2, 0.4, 0.8 (h) White Wine Quality; ν=0.2, 0.3, 0.4

Fig. 5 (continued)

Note that the corresponding membership functions of $A_i^{\wedge}(x_j)$ and $A_i^{\sim}(\boldsymbol{x}^{\sim})$ are computed as follows

$$A_i^{\wedge}(x_j) = \frac{1}{\sum_{l=1}^{c} \left(\frac{x_j - v_{ij}}{x_j - v_{il}} \right)^{2/(m-1)}} \tag{12}$$

$$A_i^{\sim}(x^{\sim}) = \frac{1}{\sum_{l=1}^{c} \left(\frac{||x^{\sim} - v_i^{\sim}||}{||x^{\sim} - v_l^{\sim}||} \right)^{2/(m-1)}} \tag{13}$$

The consequence is that if $A_i^{\wedge}(x_j) \times A_i^{\sim}(\boldsymbol{x}^{\sim})$ is used as the condition part of the ith rule, the output of the model is going to be different than the original rule. It is likely that the accuracy of the model could be reduced as a result of the increased interpretability of the rules because of the isolation of the input variables. In the above formulation, one is interested in choosing an individual variable (jth one) for which the results provided by the rule-based model are as close as possible to those formed by the original fuzzy model. The selection of the input variable is quite straightforward through a direct enumeration.

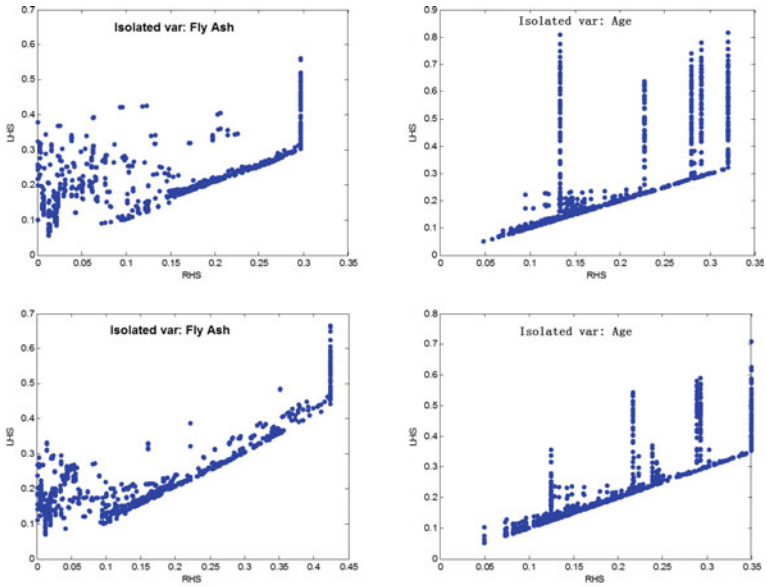
The plots showing the LHS and the RHS relationship is shown for concrete strength with one or five isolated input variables.

From the plots above, several observations of a general character can be drawn. First, the values of the LHS are higher than the corresponding ones for the RHS. This is reflective of the fact that the separation of the variable(s) leads to the higher activation levels of the rules with eventual reduction of the specificity of the results of reasoning. It is also apparent that with the increase of the variables being isolated – compare Fig. 6a, b, the differences between the values produced by the LHS and RHS of the expression (10) are more profound. Again, this is not surprising as by isolating more variables we depart from the RHS more vigorously.

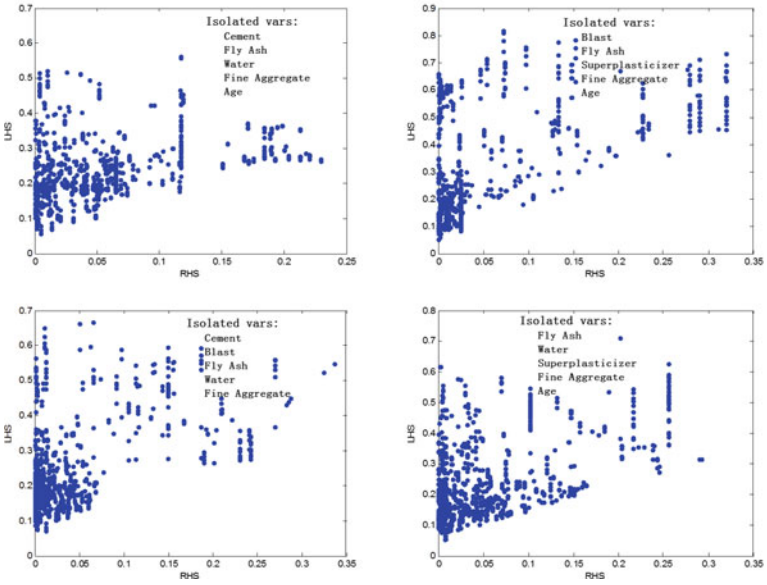
In a general setting, one can realize an isolation of L input variables, which as a result leads to the rules in the form

$$\begin{aligned} &- \text{if } \mathrm{x_{j1}} \text{ is } \mathrm{A}_i^{\wedge}\, \mathrm{x_{j1}} \textit{ and } \mathrm{x_{j2}} \textit{ is } \mathrm{A}_i^{\wedge}(\mathrm{x_{j2}}) \\ &\textit{and} \cdots \textit{and } \mathrm{x_{jL}} \text{ is } \mathrm{A}_i^{\wedge}(\mathrm{x_{jL}}) \textit{ and } x^{\sim}\ \mathrm{A}_i^{\sim}(\mathbf{x}^{\sim}) \\ &\qquad\qquad\qquad\qquad\qquad\qquad \text{then y is } \mathrm{f_i}(\mathbf{x}, \mathbf{a_i}) \end{aligned} \tag{14}$$

note that in this case $\boldsymbol{x}^{\sim}$ is defined in $\boldsymbol{R}^{n-L}$.

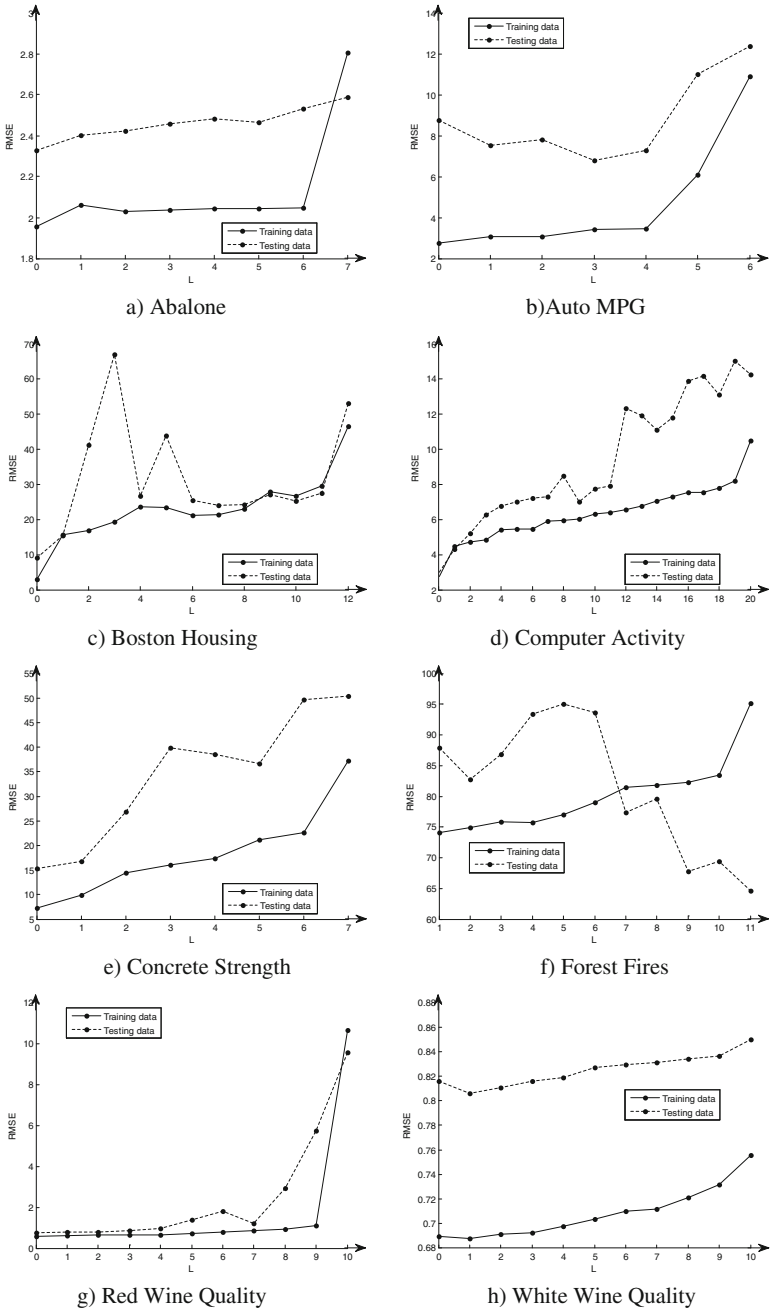


(a) single input variable isolated



(b) 5 isolated input variables

Fig. 6 Values of the original activation of the rules value versus the one with isolated input variables - concrete strength data set, 4 rules in the rulebase



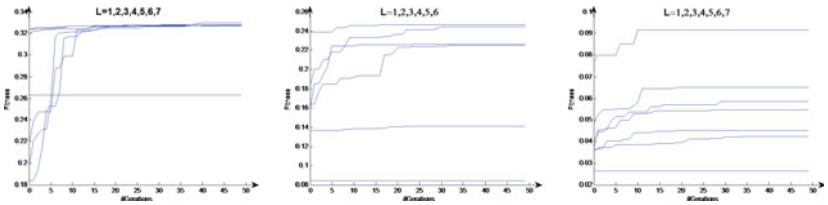
a) Abalone b)Auto MPG

c) Boston Housing d) Computer Activity

e) Concrete Strength f) Forest Fires

g) Red Wine Quality h) White Wine Quality

Fig. 7 Performance of the fuzzy model versus the number of isolated input variables



(a) Abalone (b) Auto MPG (c) Concrete Strength

Fig. 8 Fitness function reported in successive GA generations

Here an optimal choice of L variables gives rise to a combinatorial optimization problem. This could be solved by GA optimization. Considering that the value of L is specified in advance, GA forms an optimal isolation matrix $\mathbf{I}$ consisting of c rows (number of rules) and n rows (number of input variables) where in each row there are L 1 s indicating the variables which are isolated in the rule. For instance, for $L = 3$ the matrix with the entries

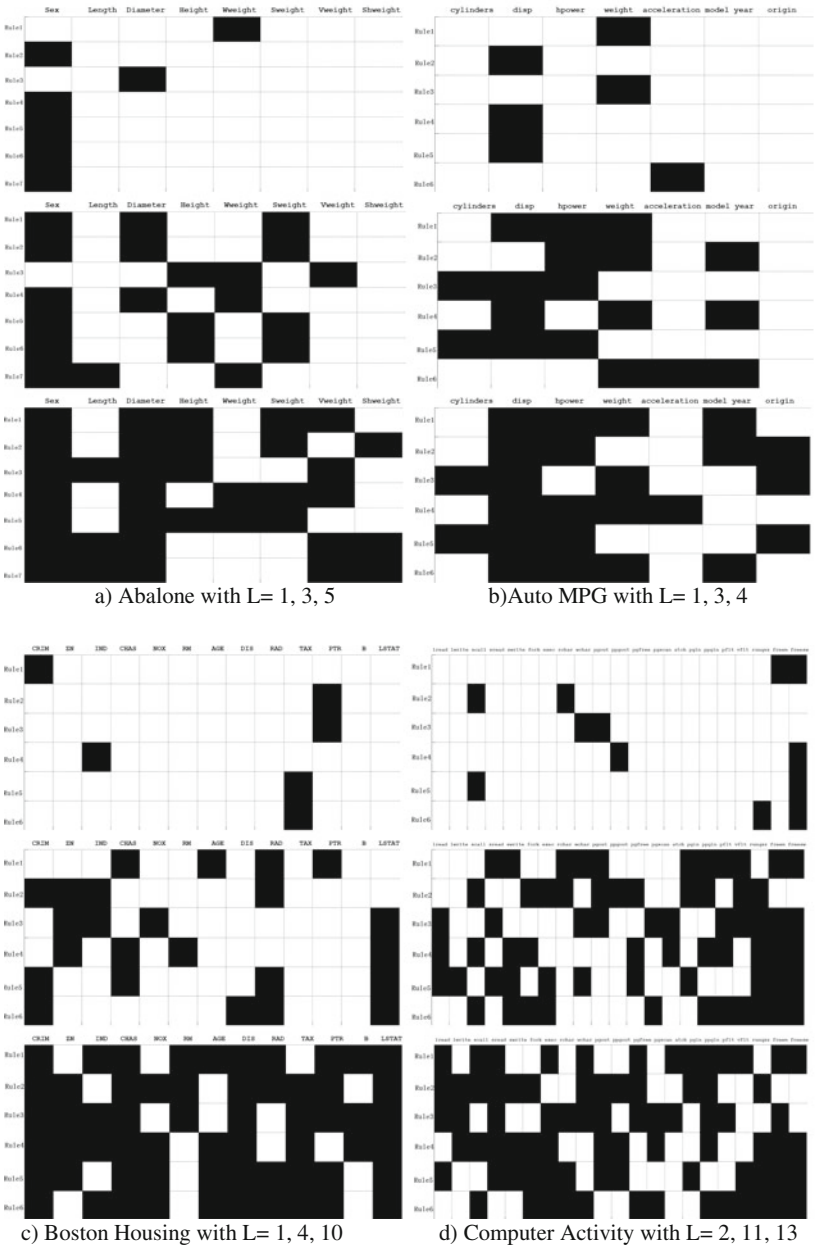
$$\mathbf{I} = \begin{bmatrix} 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 \\ \cdots & & & & & \end{bmatrix} \tag{15}$$

states that in the first rule isolated are variables 1, 4, and 5; in the second rule we isolate variables 2, 3, and 4, and so on.

6 Experiments

The GA was carried out with 100 populations and maximum 50 generations. Preliminary experiments with GA indicated lack of improvement before 50th generation, so the maximum generation is set to 50. The population is not large, so relative large values of moderate crossover and mutation rates are used. The crossover rate is set as 0.8 and mutation rate is 0.1.

We present the results in a similar way as before by focusing on the presentation of the rules with isolated variables and showing how the families of isolated variables impact the performance of the model (Figs. 7, 8 and 9).



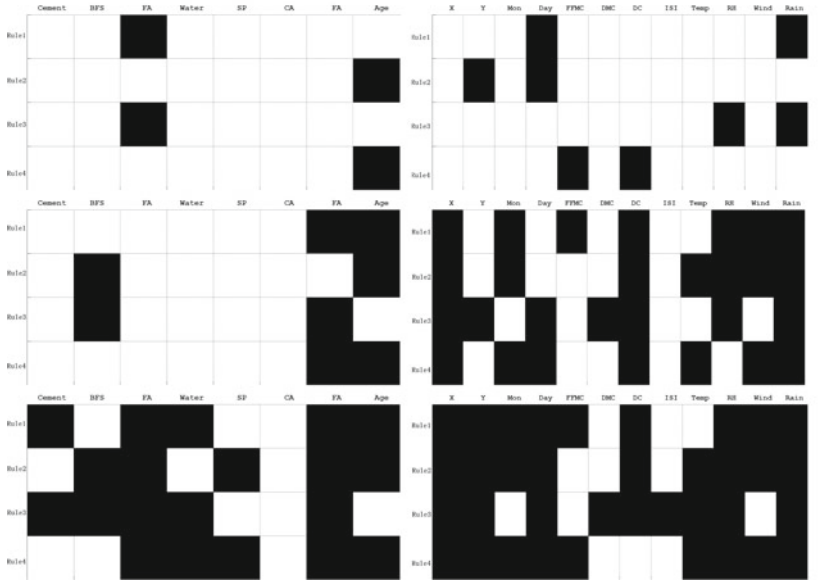
a) Abalone with L= 1, 3, 5

b)Auto MPG with L= 1, 3, 4

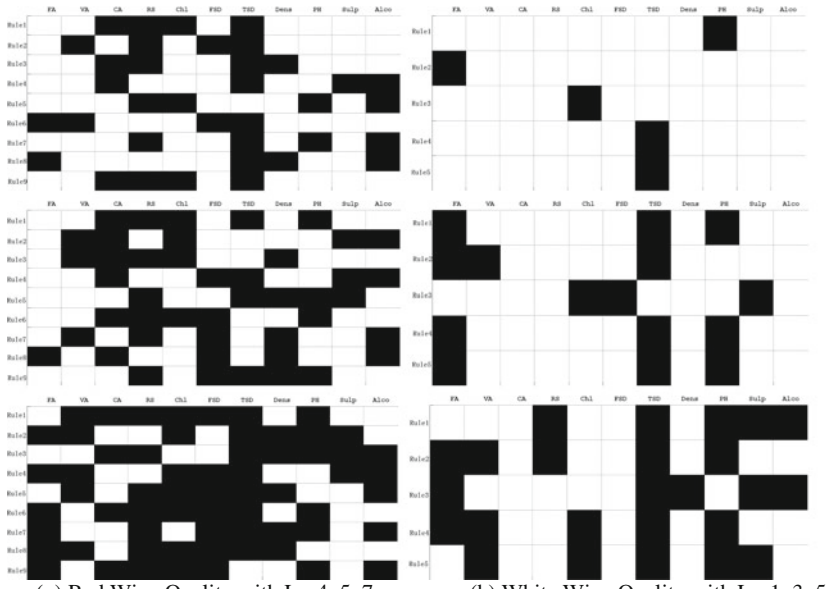
c) Boston Housing with L= 1, 4, 10

d) Computer Activity with L= 2, 11, 13

Fig. 9 Isolated variables (shaded) obtained for selected values of L



e) Concrete Strength with L= 1, 2, 5 f) Forest Fires with L= 2, 7, 9



(g) Red Wine Quality with L= 4, 5, 7 (h) White Wine Quality with L= 1, 3, 5

Fig. 9 (continued)

7 Conclusions

The two approaches enhancing the interpretability of rule-based models are directly applied to the already constructed Takagi-Sugeno fuzzy models realized with the use of fuzzy clustering. Both the formation of the input subspace of conditions as well as the isolation of the input variables are the methods refining multivariable fuzzy sets produced through fuzzy clustering. The proposed approaches are quantifiable in terms of the level of the interpretability abilities offered by them (expressed either in terms of the number of variables eliminated or the variables isolated). This aspect is helpful in determining how much the interpretability could be enhanced without any significant sacrifice of accuracy of the model. Furthermore in this way one could reveal input variables (or their combinations) that are essential in rule-based modeling.

The approach offers a certain new view at the enhancement of fuzzy rule-based models. There could be several avenues worth pursuing in the future including (a) development of a hybrid arrangement of the formation of subspaces of conditions of the rules associated with some further isolation of variables from such subspaces, (b) use of other techniques of Evolutionary Optimization in the entire process, (c) construction of interpretability measures quantifying various facets of the interpretation mechanisms.

References

1. Alcala, R., Gacto, M.J., Herrera, F.: A fast and scalable multiobjective genetic fuzzy system for linguistic fuzzy modeling in high-dimensional regression problems. IEEE Trans. Fuzzy Syst. **19**(4), 666–681 (2011)
2. Alonso, J.M., Magdalena, L., Guillaume, S.: Linguistic knowledge base simplification regarding accuracy and interpretability. Mathware Soft Comput. **13**(3), 203–216 (2006)
3. Ayouni, S., Yahia, S.B., Laurent, A.: Extracting compact and information lossless sets of fuzzy association rules. Fuzzy Sets Syst. **183**(1, 16), 1–25 (2011)
4. Bezdek, J.C.: Pattern recognition with fuzzy objective function algorithms. Plenum Press, New York (1981)
5. Bodenhofer, U., Bauer, P.: A formal model of interpretability of linguistic variables. In: Casillas, J., Cordón, O., Herrera, F., Magdalena, L. (eds.) Interpretability Issues in Fuzzy Modeling, pp. 524–545. Springer, Berlin (2003)
6. Chen, M.Y., Linkens, D.A.: Rule-base self-generation and simplification for data-driven fuzzy models. Fuzzy Sets Syst. **142**(2), 243–265 (2004)
7. Goldberg, D.: Genetic Algorithms in Search, Optimization and Machine Learning. Addison-Wesley (1989)
8. Gacto, M.J., Alcalá, R., Herrera, F.: Integration of an index to preserve the semantic interpretability in the multiobjective evolutionary rule selection and tuning of linguistic fuzzy systems. IEEE Trans. Fuzzy Syst. **18**(3), 515–531 (2010)
9. Ishibuchi, H., Yamamoto, T.: Fuzzy rule selection by multi-objective genetic local search algorithms and rule evaluation measures in data mining. Fuzzy Sets Syst. **141**(1), 59–88 (2004)

10. Krone, A., Krause, H., Slawinski, T.: A new rule reduction method for finding interpretable and small rule bases in high dimensional search spaces. In: Proceedings of 9th IEEE International Conference on Fuzzy System, San Antonio, TX, pp. 693–699 (2000)
11. Mitchell, M.: Introduction to Genetic Algorithms. MIT Press (1998)
12. Muhlenbein, H., Schlierkamp-Voosen, D.: Predictive models for the Breeder genetic algorithm I. Continuous Parameter Optim. Evol. Comput. **1**, 25–49 (1993)
13. Wright, A.: Genetic algorithms for real parameter optimization. In: Rawlin, G.J.E (ed.) Foundations of Genetic Algorithms 1, pp. 205–218. Morgan Kaufmann, San Mateo (1991)
14. Yam, Y., Baranyi, P., Yang, C.T.: Reduction of fuzzy rule base via singular value decomposition. IEEE Trans. Fuzzy Syst. **7**, 120–132 (1999)

Authors Biography

Witold Pedrycz is a Professor and Canada Research Chair (CRC) in Computational Intelligence in the Department of Electrical and Computer Engineering, University of Alberta, Edmonton, Canada. He is also with the Systems Research Institute of the Polish Academy of Sciences, Warsaw, Poland. He also holds an appointment of special professorship in the School of Computer Science, University of Nottingham, UK. In 2009 Dr. Pedrycz was elected a foreign member of the Polish Academy of Sciences. In 2012 he was elected a Fellow of the Royal Society of Canada. Witold Pedrycz has been a member of numerous program committees of IEEE conferences in the area of fuzzy sets and neurocomputing. In 2007 he received a prestigious Norbert Wiener award from the IEEE Systems, Man, and Cybernetics Council. He is a recipient of the IEEE Canada Computer Engineering Medal 2008. In 2009 he has received a Cajastur Prize for Soft Computing from the European Centre for Soft Computing for "pioneering and multifaceted contributions to Granular Computing". In 2013 has was awarded a Killam Prize. In the same year he received a Fuzzy Pioneer Award 2013 from the IEEE Computational Intelligence Society. His main research directions involve Computational Intelligence, fuzzy modeling and Granular Computing, knowledge discovery and data mining, fuzzy control, pattern recognition, knowledge-based neural networks, relational computing, and Software Engineering. He has published numerous papers in this area. He is also an author of 15 research monographs covering various aspects of Computational Intelligence, data mining, and Software Engineering. Dr. Pedrycz is intensively involved in editorial activities. He is an Editor-in-Chief of Information Sciences and Editor-in-Chief of WIREs Data Mining and Knowledge Discovery (Wiley). He currently serves as an Associate Editor of IEEE Transactions on Fuzzy Systems and is a member of a number of editorial boards of other international journals.

Kuwen Li received the B.S. and M.S. degrees in computer science from Harbin Engineering University, P. R. China and M. S. degree in computer engineering in 2005 and Ph. D. degree in computer engineering in 2014 from the University of Alberta, Canada. His research interests include Computational Intelligence, fuzzy modeling, knowledge based neural networks, and software engineering.

Marek Reformat received his M.Sc. degree (with honors) from Technical University of Poznan, Poland, and his Ph.D. from University of Manitoba, Canada. Presently, he is a professor with the Department of Electrical and Computer Engineering, University of Alberta. The goal of his research activities is to develop methods and techniques for intelligent data modeling and analysis leading to translation of data into knowledge, as well as to design systems that possess abilities to imitate different aspects of human behavior. He recognizes the concepts of Computational Intelligence – with fuzzy computing and possibility theory in particular – are key elements necessary for capturing relationships between pieces of data and knowledge, and for mimicking human ways of reasoning about opinions and facts. Dr. Reformat applies elements of fuzzy sets to social networks, Linked Open Data, and Semantic Web in order to handle inherently imprecise information, and provide users with unique facts retrieved from the data.

Dr. Reformat is a past president of the North American Fuzzy Information Processing Society, and a vice president of the International Fuzzy Systems Association. He has been a member of program committees of multiple international conferences related to Computational Intelligence and Software Engineering.