

Learning Fuzzy Rules using Genetic Programming: Context-free grammar definition for high-dimensionality problems

Francisco José Berlanga, María José del Jesus
Department of Computer Science and
Artificial Intelligence
University of Jaén
Jaén E-20071, Spain
E-mail: berlanga,mjjesus@ujaen.es

Francisco Herrera
Department of Computer Science and
Artificial Intelligence
University of Granada
Granada E-18071, Spain
E-mail: herrera@decsai.ugr.es

Abstract—The inductive learning of a fuzzy rule-based classification system (FRBCS) with high interpretability is made difficult by the presence of a large number of features that increases the dimensionality of the problem being solved. The difficulty comes from the exponential growth of the fuzzy rule search space with the increase in the number of features considered.

In this work we tackle this problem, the FRBCS learning with high interpretability for high-dimensionality problems. We propose a genetic-programming-based method, where the evolved disjunctive normal form fuzzy rules compete in order to obtain an FRBCS with high interpretability (few rules and few antecedent conditions per rule) while maintaining a good performance.

I. INTRODUCTION

One of the most important areas for the application of the Fuzzy Set Theory are Fuzzy Rule-Based Systems (FRBSs). FRBSs have been successfully applied to various fields such as control, modelling and classification. While the main goal in the design of FRBSs has been the performance maximization, their interpretability has also been taken into account in some recent studies [3].

Regarding to the interpretability of linguistic FRBSs, the difficulty comes from the exponential growth of the fuzzy rule search space with the increase in the number of features considered. Usually human users do not want to check hundreds of fuzzy rules, so the number of fuzzy rules is closely related to the interpretability of FRBSs. On the other hand, the rule length is also closely related to the interpretability of FRBSs.

This problem can be tackled by different ways, a) compacting and reducing the rule set as a postprocessing approach ([13] [15]), and b) carrying out a feature selection process, that determines the most relevant variables before or during the FRBS inductive learning process. The feature selection approach reduces the fuzzy rule search space and

increases the efficiency of the learning. Several feature selection processes have been proposed involved in the learning of FRBCSs ([2][4][10][20][21]).

In this work, we tackle the FRBCS learning with high interpretability for high-dimensionality problems by means of a genetic-programming (GP) based approach.

The genetic programming [14] is an extension to the inspiration of GA, where the main problem of GA concerning the fixed problem definition is avoided by using variable-length trees instead of fixed-sized individuals.

The FRBCSs learning has already been tackled previously in the specialized literature by means of GP. Chien et al. learn fuzzy rules from discriminant functions in [7]. Sanchez et al. propose an FRBCS learning process in [22] and [9] by combining GP operators with simulated annealing and genetic algorithm respectively to establish the membership functions. Mendes et al. develop in [16] a co-evolutionary algorithm which includes a GP based algorithm for FRBCS learning and an evolutionary algorithm to evolve membership functions. Finally, other authors have also used the GP for the learning of FRBS for modelling (see [1] and [11], for instance).

In our proposal, the definition of a context-free grammar, together with the use of a competition mechanism between rules which allows for obtaining a reduced fuzzy rule set, with few antecedent conditions per rule and high-generalization capability. In this way the GP learning process obtains interpretable and compact FRBCSs with predefined membership functions, which can increase the interpretability of the system including expert knowledge about the problem.

In order to do that, the paper is organized as follows. Some previous considerations about the proposal FRBCS learning process using GP are shown in Section II. Components of our GP based learning process are explained in Section III. Section IV shows the results of the experimentation carried out. Finally, the conclusions obtained are presented in Section V.

II. LEARNING FUZZY RULE-BASED CLASSIFICATION SYSTEMS USING GENETIC PROGRAMMING

The first feature of our proposal is the coding approach used. It encodes one fuzzy rule per individual in the population. The rule set is provided by combining all the individuals in population.

The kind of rule used in this work is disjunctive normal form (DNF) rule, which has the following form:

If X_1 is \hat{A}_1 and ... and X_n is \hat{A}_n then Y is B , certainty degree C

where each input variable X_i takes as a value a set of linguistic terms $\hat{A}_i = \{A_{i1} \text{ or } \dots \text{ or } A_{iL_i}\}$, whose members are joined by a disjunctive operator, whilst the output variable (Y) has one of the class values.

This structure uses a more compact description that improves the interpretability. Moreover, the structure permits changes in the granularity by means the combination of linguistic terms using or operation and it is a natural support to allow the absence of some input variables in each rule (simply making \hat{A}_i be the whole set of linguistic terms).

The DNF rule also includes a certainty degree ($C \in [0,1]$), which represents the confidence of the classification in the class represented by the consequent to be true under the antecedent conditions of the rule.

The proposed FRBCS learning method encode one fuzzy DNF rule per individual in the population. However, we are not interested in obtaining only the best evolved rule, but we want to obtain the rule set which best covers all the search space. Therefore, it is necessary to maintain different groups of individuals in the population, exploring each different space part. Each group is referred to as species and the search space being explored by a species is referred to as a niche. The maintaining of the diversity of the population is important for the formation of niches. Moreover, the individuals must not be allowed to converge into a single niche and must be forced to explore different parts of the search space.

Several approaches have been designed to carry out this task (crowding, fitness sharing, ...) in GAs [18]. These approaches are based on two main principles:

- 1) The parents should be among the most similar individuals to the offspring.
- 2) The estimate of some similarity measure between individuals.

However, these ideas present problems when used with GP, since the parents and the offspring could be totally different due the variable-length nature of the individuals. Furthermore, it is much more complex to calculate how one individual is similar to another individual in GP. To solve these problems, it is necessary to use an approach which does not take into account the individual structure. In our method, the approach used is the Token Competition [24]. Token competition is applied in our method to evolve different multiple rules for prediction of each class in the data set as well as to preserve the diversity in the evolution.

The concept is as follows: In the natural environment, once an individual has found a good place to live, it will try to exploit this niche and prevent other newcomers from sharing the resources, unless the newcomer is stronger than it is. The other individuals are hence forced to explore and find their own niches. In this way, the diversity of the population is increased.

Based on this mechanism, it is assumed that each record in the training set can provide a resource called a token, for which all chromosomes in the population will compete to capture. If an individual (rule) can match the record, it sets a flag to indicate that the token is seized. Other weaker individuals then cannot get the token. The priority of receiving tokens is determined by the strength of the individuals. The individuals with a high fitness score can exploit the niche by seizing as many tokens as they can. The other ones entering the same niche will have their strength decreased because they cannot compete with the stronger ones. The fitness score of each individual is modified based on the tokens it can seize. The modified fitness is defined as:

$$\text{Modified_fitness} = \text{raw_fitness} \times \text{count} / \text{ideal} \quad (1)$$

where *raw_fitness* is the fitness score obtained from the evaluation function, *count* is the number of tokens that the individual actually seized and *ideal* is the total number of tokens that it can seize, which is equal to the number of records that the individual matches.

From another point of view, each individual contributes to the system by covering several records. If a training example has already been covered by one individual, then another individual covering the same example will make no contribution to the system. Thus the fitness of the latter individual should be discounted.

As a result of token competition, there exist individuals that cannot seize any token. These individuals are redundant as all of their examples are already covered by other stronger individual and, hence, they can be replaced by new individuals. The introduction of these new individuals can inject a larger degree of diversity into the population and provide extra changes for generating good ones. To create the new individuals, we can use seeds to generate better individuals. Possible seeds are those examples which tokens have not already been taken, i.e, examples which are not yet covered by any existing individuals, and thus introducing new ones covering them can improve the system. To create a new individual, a seed is randomly selected, and then an individual is generated to cover the seed.

Another important point of interest is the evaluation of the individuals, that is, the definition of a proper fitness function that allows good individuals to get high scores and hence have their tokens available. For that, our departure point are the following four well known measures:

- 1) True positives (tp): The number of examples that are covered by the individual and belong to the same class that is predicted by the individual.

2) False positives (fp): The number of examples that are covered by the individual but belong to a different class that is predicted by the individual.

3) True negatives (tn): The number of examples that not are covered by the individual and do not belong to the same class which is the predicted by the individual.

4) False negatives (fn): The number of examples that not are covered by the individual but belong to the same class that is predicted by the individual.

Note that true positives and true negatives correspond to the correct predictions made for the individual being evaluated, whereas false positives and false negatives correspond to wrong predictions made by the individual. From the previous definitions, two measures can be constructed:

1) *Confidence*: It measures the accuracy of an individual, that is, the confidence of the consequent to be true under the antecedent.

$$confidence = \frac{tp}{(tp + fp)} \times \frac{tn}{(fn + tn)} \quad (2)$$

2) *Support*: It measures the generalization capacity of an individual. An individual can have high accuracy but may be formed by chance and based on a few training examples. If support is below a user-defined minimum threshold (*min_support*), the confidence should not be considered to avoid the waste of effort to evolve those individuals with a high confidence but cannot be generalized.

$$support = \frac{tp}{(tp + fn)} \times \frac{tn}{(fp + tn)} \quad (3)$$

In our experiments, the definitions showed in (2) and (3) have been used. Finally, both measures are combined to form the fitness function in (4). In [24] a different way to compute support and confidence can be found.

$$raw_fitness = support, \quad \text{if } support < min_support$$

$$raw_fitness = support \times confidence, \quad \text{otherwise} \quad (4)$$

III. COMPONENTS OF THE GENETIC PROGRAMMING BASED FRBCS LEARNING PROCESS

Once the previous conditions have been completely clarified, we show an explanation of the principal components/steps of the proposed method.

- grammar definition
- data base definition
- genetic operators
- population evolution
- rule base simplification
- certainty degree calculation

1. The first step, consists on the definition of a grammar according to the problem to be solved. This grammar must specify the structure of a rule, which is in the form “if *antecedent* then *consequent*”. An example of the grammar for a classification problem with two features (X_1, X_2), three linguistic labels per feature (Low, Medium, High) and three classes (C_1, C_2, C_3), is given in Table I. Clearly, this example grammar, enables also the learning of DNF-type fuzzy rules in which each input variable takes as value a set of linguistic terms whose members are joined by a disjunctive operator. This structure uses a more compact description that improves the interpretability.

TABLE I
GRAMMAR EXAMPLE

start \rightarrow [If] antec [then] conseq. antec \rightarrow descriptor1 [and] descriptor2. descriptor1 \rightarrow [any]. descriptor1 \rightarrow [X_1 is] label. descriptor2 \rightarrow [any]. descriptor2 \rightarrow [X_2 is] label. label \rightarrow {member (?a, [L, M, H, L \vee M, L \vee H, M \vee H, L \vee M \vee H])}, [?a]. conseq \rightarrow [Class is] descriptorClass. descriptorClass \rightarrow {member (?a, [C_1, C_2, C_3])}, [?a].
--

2. Secondly, the data base (DB) is defined, fixing the parameters of the fuzzy sets associated with the labels present in the grammar. In our experiments, we have divided each feature definition interval in a uniform form, using triangular fuzzy sets.

3. Once the grammar and the DB are properly defined, an initial population of rules is randomly generated, according to the grammar production rules. In each iteration, individuals are selected to evolve offspring by one of the next three genetic operators:

1) *Crossover*: Produces one child from two parents. A part in the first parent is selected and replaced by another part in the second one. Both parts are randomly selected but under the constraint that the offspring produced must be valid according to the grammar.

2) *Mutation*: A part of the rule is selected and replaced by a randomly generated part. The new part is generated by the same derivation mechanism as in the population creation. Since the offspring have to be valid according to the grammar, a selected part can only mutate to another part with a compatible structure.

3) *Dropping Condition*: Due the probabilistic nature of GP, redundant constraints may be generated in the rule. Thus, it is necessary to generalize the rules, to represent the actual knowledge in a more concisely form. Dropping condition selects randomly one condition in the antecedent part and then turns it into “any”. The attribute in the condition is no longer considered in the rule, hence, the rule can be generalized.

This operator forces the feature selection in the rules, allowing us to get rules with a small number of variables per antecedent.

4. In each iteration, the number of new individual evolved equals the population size, thus the number of individuals in the population is doubled. At this moment, the token competition is carried out in order to maintain the diversity on the population. As a result of token competition, some individuals have their fitness modified to zero, hence they must be replaced by another which matches to uncovered records. If all the records are already covered, these individuals are eliminated from the population. Finally, the population size is set to half its current size.

Due the individuals can be eliminated from the population if their fitness is modified to zero by token competition and there are not uncovered records, the size of the final population can be smaller than the initial one. This shows that the proposed method is available to get reduced and compact fuzzy rule sets, which contains only the necessary rules to cover the whole training set. Our method is clearly orientated to get FRBCS with high interpretability without a significant performance loss.

5. Rule base simplification. Once the evolutionary process has finished, a post-processing step is carried out for eliminating redundant rules.

During the rule base learning process it may occur that the algorithm learn two rules, where one is included in the other. For example, consider the two rules showed in (5).

$$\begin{aligned} R1: & \text{ If } X_1 \text{ is Low then Class is } C_1 \\ R2: & \text{ If } X_1 \text{ is Low } \vee \text{ Medium then Class is } C_1 \end{aligned} \quad (5)$$

As we can see, the second rule includes the first one, hence, it does not make sense to keep both of them in the rule set. In this case, the logic solution deletes the first rule because the examples that it covers, are also covered by the second rule.

Both rules, R1 and R2, can exist in the population only if R1 had entered before R2 to the token competition process. If both would have done it in inverse order, R1 would have modified its fitness to zero (and, therefore, R1 would be eliminated from the population) because all its tokens would have already been seized by R2, which covers at least the same examples that R1, due R2 includes R1.

This process aims to increase the interpretability of the previously learned FRBCS, by deleting redundant rules.

6. Finally, our method ends with the certainty degree calculation for each individual (rule) in the population.

The certainty degree is obtained as the quotient S_j / S , where S_j is the sum of the matching degrees for the training examples belonging to class represented by the consequent which are covered by the antecedent of the rule, and S the sum of the matching degrees for all the training examples which are covered by the antecedent of the rule, independently of the class they belong.

IV. EXPERIMENTAL STUDY

In order to analyze the behaviour of the proposed method, an experimental study has been carried out using the Wisconsin Breast Cancer database [17].

This data base has been obtained from the University of Wisconsin Hospitals. The examples consist of the visual evaluation of the nuclear characteristics of the samples obtained by Fine Needles Aspirates (FNAs). Each samples is characterized by nine features in the range [1-10], where an 1 corresponds to a normal state while an 10 is associated to the most abnormal state. Finally, each sample has assigned one of the two next classes: benign or malignant (represented numerically as 2 and 4, respectively, in data sets). The measured variables are as follows:

- 1) Clump Thickness
- 2) Uniformity of Cell Size
- 3) Uniformity of Cell Shape
- 4) Marginal Adhesion
- 5) Single Epithelial Cell Size
- 6) Bare Nuclei
- 7) Bland Chromatin
- 8) Normal Nucleoli
- 9) Mitosis

The original data set is partitioned using 10-fold cross-validation procedure. The initial data set T , is randomly divided into 10 disjoint sets of equal size T_1, \dots, T_N . We maintain the original class distribution (before partitioning) within each set when carrying out the partitioning process. We then conduct 10 pairs of training and test sets.

Our method (from now on called FRBCS_GP) has been compared to other two fuzzy rules learning techniques and with a decision tree method:

1) *Wang & Mendel*: In [23], it is proposed a fuzzy control rules learning method, which Chi et al. extend for classification problems [5], [6].

This method generates a fuzzy rule for each example in the training set. It does not carry out any feature selection process. This method is used in our study in order to show the behaviour of a method without feature selection in a high dimensionality problem.

2) *Ravi et al.*: In [20] a process for deriving fuzzy rules for high-dimensionality classification problems is proposed. This approach extracts a more reduced set of features from the original ones by the Principal Component Analysis (PCA). After that, a fuzzy rule learning process is carried out following the method proposed in [12] which divides the pattern space in several fuzzy subspaces, learning a rule for each one. Finally, a modified threshold accepting algorithm [21] is used to build a compact rule subset with a high classification power, from rule set obtained in the previous stage.

3) *C4.5*: It is a classification algorithm proposed by Quinlan [19] as an extension of his previously proposed ID3 algorithm. It is based on information theory and it also includes a feature

selection method. This algorithm uses Divide-and-Conquer method and the criterion called information gain for constructing a decision tree, which can be later transformed into a crisp rule set.

We have used 5 linguistic labels per variable in all the experiments and all the data partitions have been normalised to the [0-1] interval. The specific parameters of Ravi et al. method are shown in Table II, while the FRBCS_GP parameter values are in Table III. Finally, it is important point out that our method learns DNF-type fuzzy rules.

TABLE II
RAVI ET AL. PARAMETERS

RAVI	
PCA	<i>Threshold</i> = 90% (reduces the original feature space from 9 to 5 features)
MTA	$U = 0.95\%$, $thresh = 0.035$, $thrtol = 10^{-8}$, $acc = 10^{-6}$, $old = 9999$, and $itrmax = 100$, $W_{NCP} = 10$ and $W_S = 1$

TABLE III
FRBCS_GP PARAMETERS

FRBCS_GP
$Iter = 100$, $pop_size = 20$, $P_{crossover} = 0.5$, $P_{mutation} = 0.4$, $P_{drop} = 0.1$, $min_support = 0.01$

The results are showed in Table IV. In this table, the first column indicates the name of the algorithm, the second one shows the average rule number (#R); the third one, the average antecedent variables per rule (#Var), the fourth, the average antecedent conditions number per rule (#Cond); and the last two columns stand for the correct percentage with training (%Tra) and test (%Test) examples respectively. In this table, the subscripts in our proposal, are related to the fuzzy reasoning method (FRM) used, so FRBCS_GP₁ correspond to the classical FRM (max-min) and the FRBCS_GP₂ with the normalised sum [8].

TABLE IV
WISCONSIN RESULTS

Method	#R	#Var	#Cond	%Tra	%Test
WM	296.5	9	9	100	66.335
RAVI	44.77	5	5	98.9263	86.2123
C4.5	25	4.46	5.08	99.69	94.43
FRBCS_GP ₁	7.77	1	2.22	92.88	93.92
FRBCS_GP ₂	7.77	1	2.22	94.133	93.90

Analyzing Table IV, we can point out the following considerations:

- The method that does not use feature selection (WM) learns a big number of rules showing overfitting on the training set.
- Our method learns the rule set with lower number of variables and labels per rule (average 1 and 2.2

respectively) than the remaining ones. It also learns rule bases with a few number of rules. Therefore the resulting FRBCS has a high interpretability level.

- Analysing the performance of our approach, we find a similar behaviour between training and test, without overfitting. In comparison with the other approaches, we find a good performance in test, better than Ravi's method and similar to C4.5. Regarding to the training performance, it is far from the other methods due to the fitness function does not use any measure based on the global classification performance.

In Table V we show an example consisting of a rule set, learned by our method for the Wisconsin Breast Cancer classification problem.

TABLE V
LEARNED RULE SET EXAMPLE

R1: If X ₂ is L1 then Class 2, certainty = 0.97
R2: If X ₂ is (L2 or L3 or L4 or L5) then Class 4, certainty = 0.81
R3: If X ₆ is (L1 or L3 or L4) then Class 2, certainty = 0.86
R4: If X ₆ is (L2 or L5) then Class 4, certainty = 0.84
R5: If X ₇ is L1 then Class 2, certainty = 0.96

V. CONCLUSIONS

In this work, we have proposed a genetic-programming-based method to obtain FRBCSs with a high interpretability. Since the GP individuals are represented by variable-length trees, they can naturally allow for the absence of any input feature, getting rules with fewer antecedent conditions. On the other hand, the use of a niche formation mechanism to increase the diversity into the population, makes the rules compete among themselves giving out a fewer number of rules with a high-generalization capability.

The effectiveness of the method is shown by an example, and the results are promising. Therefore, we consider this approach to be an interesting alternative for the learning of interpretable FRBCSs for high-dimensionality problems.

As future work we will analyse the behaviour of this proposal in different sintetic and real data sets, including huge DB in Data Mining problems. We also will incorporate a proper multiobjective approach in the genetic programming based leaning process.

REFERENCES

- [1] A. Bastian. "Identifying fuzzy models utilizing genetic programming". *Fuzzy Sets and Systems* 113(3) 1, pp. 333-350, 2000.
- [2] J. Casillas, O. Cordon, M.J. Del Jesús, and F. Herrera, "Genetic feature selection in a fuzzy rule-based classification system learning process for high-dimensional problems". *Information Sciences* 136 (1-4), pp. 135-157, 2001.
- [3] J. Casillas, O. Cordon, F.Herrera, L.Magdalena (Eds.). *Interpretability Issues in Fuzzy Modeling*. Springer-Verlag, 2003. Series Studies in Fuzziness and Soft Computing, Vol. 128.
- [4] D. Chakraborty, and N.R. Pal, "A neuro-fuzzy scheme for simultaneous

feature selection and fuzzy rule-based classification". *IEEE Transactions on Neural Networks* 15 (1), pp. 110-123, 2004.

- [5] Z. Chi, J. Wu, and H. Yan, "Handwritten numeral recognition using self-organizing maps and fuzzy rules". *Pattern Recognition* 28 (1), pp. 59-66, 1995.
- [6] Z. Chi, H. Yan, and T. Pham, *Fuzzy Algorithms with Applications to Image Processing and Pattern Recognition*. World Scientific, 1996.
- [7] B-C. Chien, J.Y. Lin, and T-P. Hong. "Learning discriminat functions with fuzzy attributes for classification using genetic programming". *Expert Systems with Applications* 23(1), pp. 31-37, 2002.
- [8] O. Cordón, M.J. del Jesús, and F. Herrera. "A Proposal on Reasoning Methods in Fuzzy Rule-Based Classification Systems". *International Journal of Approximate Reasoning* 20, pp. 21-45, 1999.
- [9] S. Garcia, F. González, and L. Sánchez. "Evolving fuzzy based classifiers with GA-P: A grammatical approach". *Genetic Programming: Second European Workshop, EuroGP'99*. Lecture Notes in Computer Science 1598, pp 203-210, 1999.
- [10] A. González, and R. Pérez, "Selection of relevant features in a fuzzy genetic learning algorithm". *IEEE Transactions on Systems, Man and Cybernetics – Part B* 31 (3), pp. 417-425, 2001.
- [11] F. Hoffmann, and O. Nelles. "Genetic programming for model selection of TSK-fuzzy systems". *Information Sciences* 136(1-4), pp. 7-28, 2001.
- [12] H. Ishibuchi, K. Nozaki, and H. Tanaka, "Distributed representation of fuzzy rules and its application to pattern classification". *Fuzzy Sets and Systems* 52, pp. 21-32, 1992.
- [13] H. Ishibuchi, K. Nozaki, N. Yamamoto, and H. Tanaka, "Selecting fuzzy if-then rules for classification problems using genetic algorithms". *IEEE Trans. Fuzzy Systems* 3 (3), pp. 260-270, 1995.
- [14] J.R. Koza, *Genetic programming – on the programming of computers by means of natural selection*. Cambridge MA, USA: The MIT Press. 1992.
- [15] A. Krone, P. Krause, and T. Slawinski. "A new rule reduction method for finding interpretable and small rule bases in high dimensional search spaces". *Proceedings of the Ninth IEEE International Conference on Fuzzy Systems* vol. 2, pp. 694-699, May 2000.
- [16] R.R.F. Mendes, F. de B. Voznika, A.A. Freitas, et al.. "Discovering Fuzzy Classification Rules with Genetic Programming and Co-evolution". *Principles of Data Mining and Knowledge Discovery: 5th European Conference (PKDD'01)*. Lecture Notes in Computer Science 2168, pp. 314, Springer-Verlag, 2001.
- [17] P.M. Murphy, and D.W. Aha. "UCI repository of machine learning Databases," Univ. California, Dept. Info. Comput. Sci., Irvine, CA, <http://www.ics.uci.edu/~mlearn/MLRepository.htm>, 1998.
- [18] E. Perez, F. Herrera, and C. Hernández, "Finding Multiple Solutions in Job Shop Scheduling by Niching Genetic Algorithm". *Journal of Intelligent Manufacturing* 14 (3-4), pp. 223-239, 2003.
- [19] J.R. Quinlan, *C4.5: Programs for Machine Learning*. Morgan Kaufmann, 1993.
- [20] V. Ravi, P.J. Reddy, and H.J. Zimmermann, "Pattern classification with principal component analysis and fuzzy rule bases". *European Journal of Operational Research* 126 (3), pp. 526-533, 2000.
- [21] V. Ravi, and H.J. Zimmermann, "Fuzzy rule based classification with FeatureSelector and modified threshold accepting". *European Journal of Operational Research* 123 (1), pp. 16-28, 2000.
- [22] L. Sánchez, I. Couso, and J.A. Corrales. "Combining GP operators with SA search to evolve fuzzy rule based classifiers" *Information Sciences* 136(1-4), pp. 175-191, 2001.
- [23] L.X. Wang, and J.M. Mendel, "Generating fuzzy rules by learning from examples". *IEEE Transactions on Systems, Man, and Cybernetics* 22 (6), pp. 1414-1427, 1992.
- [24] M.L. Wong, and K.S. Leung, *Data Mining using grammar based genetic programming and applications*. Kluwer Academics Publishers, 2000.