NORFREA: AN ALGORITHM FOR NON REDUNDANT FUZZY RULE EXTRACTION

R. Senhadji, S. Sánchez-Solano, A. Barriga, I. Baturone, F. J. Moreno-Velo

Instituto de Microelectrónica de Sevilla, Centro Nacional de Microelectrónica,
Avda. Reina Mercedes s/n, E-41012-Sevilla, Spain.

NORFREA: An algorithm for non redundant fuzzy rule extraction

R. Senhadji, S. Sánchez-Solano, A. Barriga, I. Baturone, F. J. Moreno-Velo
Instituto de Microelectrónica de Sevilla (CNM-CSIC)
Avda. Reina Mercedes s/n, Edif. CICA. 41012-Sevilla. Spain
raouf@imse.cnm.es

Abstract

This contribution presents a new algorithm (NORFREA) to select fuzzy rules from a *grid* partition of the input domain. Besides using an efficiency measure for the rules, this algorithm employs an heuristic technique to reduce the influence of the initial grid structure. Different benchmarks of classification problems are included to illustrate the advantages of this algorithm.

1. Introduction

The volume and variety of data that nowadays are included into data bases have increased enormously in the last few years due to their easy and low cost storage. It is very worthwhile analyzing these data to obtain valious information for supporting decision making. The application of techniques which automate the processes of analyzing and extracting information are unavoidable whenever the size of the data base is huge. A wide set of these techniques are comprised under the name of *Data Mining* within the area of *Knowledge Discovery in Databases* (KDD). In this area, a great interest is drawn by fuzzy logic-based reasoning techniques able to extract linguistic knowledge that could be easily understood by even non expert people.

A fuzzy inference system contains a rule base whose terms or labels are similar to those employed by natural language, and an inference mechanism to extract conclusions. The especification of a fuzzy system requires two tasks that can be carried out independently although they are related. One of them is the identification of the rule base (the coarse system structure) while the other is the especification of its label parameters like the membership functions parameters (the fine system structure).

Two different strategies are usually employed when defining the rule base of a fuzzy system from input/ output data: *cluster-oriented* and *structure-oriented fuzzy rule learning*. The first one organizes the training data into clusters and uses them to create the rules. Each fuzzy cluster is transformed into a rule by projecting it onto every dimension of the input variables [1][2]. As a result, each rule has its own fuzzy sets that do not appear in other rules. This is a disadvantage because the rule base so extracted is difficult to be expressed linguistically and, hence, to be understood.

Contrary to the cluster-oriented technique, which obtains the whole system description (rules and membership functions) simultaneously, the *structure-oriented fuzzy rule learning* (proposed by Wang & Mendel [3]) generates a partition of the input space prior to creating the rule base. Besides, the number of rules that will constitute the rule base is usually given from the beginning. Once these rules have been selected, an optimization process is performed to better adjust their membership functions. This learning is usually performed under constraints that preserve the linguistic meaning. The structure-oriented fuzzy rule learning can be understood as a particular case of the cluster-oriented technique in which the data are grouped into hiperrectangular clusters formed by the grid partition [4].

Several CAD tools reported in the literature are based on the structure-oriented fuzzy rule learning technique. Two examples are NEFCLASS [5] and NEFPROX [6] which have been developed to solve classification and function approximation classesproblems by Mamdani-type fuzzy systems. The algorithm that we describe in the following is also based in this technique.

2. The NORFREA algorithm

The NORFREA algorithm is aimed at reducing the main disadvantage of the structured-oriented technique: the influence of the initial partition. The idea is to not take for granted that the set of initially selected rules providing the best results for a given partition will also give the best results after learning. Our heuristic is to detect, prior to learning, those rules that would have not been significant if starting from a better initial partition.

For this purpose, the first step of the NORFREA algorithm is to find which clusters occupy more than a single grid cell. The objective is to rearrange the initial linguistic labels of the input variables so as to cover each cluster with only one cell. The rearrangement is performed by spreading as much as possible the support of the membership functions considered for the input variables. When this is done, the initial support of one label can be absorbed by another (we will name them "linked labels"), so that some rules can disappear (they are considered "redundant") and the cluster can be defined by less rules that are more efficient.

Since the number of rules that will constitute the rule base is given from the beginning, the ellimination of some of them means that new ones with new informa-
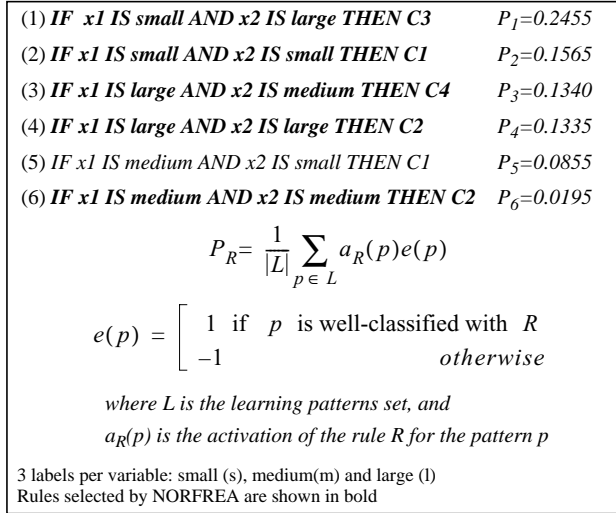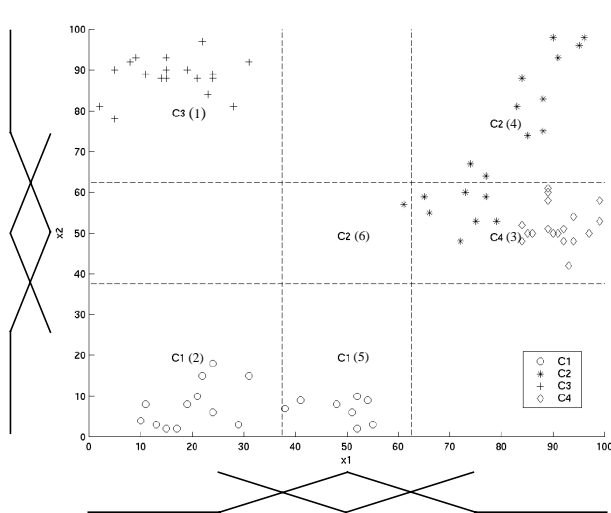
(a)   (b)

Figure 1: A simple example: a) artificial data set and initial partition, and b) rules extracted and its performance index.

The figure (b) panel contains:

(1) **IF x1 IS small AND x2 IS large THEN C3** $P_1=0.2455$
(2) **IF x1 IS small AND x2 IS small THEN C1** $P_2=0.1565$
(3) **IF x1 IS large AND x2 IS medium THEN C4** $P_3=0.1340$
(4) **IF x1 IS large AND x2 IS large THEN C2** $P_4=0.1335$
(5) *IF x1 IS medium AND x2 IS small THEN C1* $P_5=0.0855$
(6) **IF x1 IS medium AND x2 IS medium THEN C2** $P_6=0.0195$

$$P_R = \frac{1}{|L|} \sum_{p \in L} a_R(p)e(p)$$

$$e(p) = \begin{bmatrix} 1 \;\; \text{if} \;\; p \;\; \text{is well-classified with} \;\; R \\ -1 \hspace{3cm} otherwise \end{bmatrix}$$

where L is the learning patterns set, and
$a_R(p)$ is the activation of the rule R for the pattern p

3 labels per variable: small (s), medium(m) and large (l)
Rules selected by NORFREA are shown in bold

tion can be incorporated provided they do not break linked labels. In addition, if we maintain the number of labels per input, the spreading of some membership functions means that the other labels can be adjusted to best cover the rest of the input universes of discourse in the learning process.

This way, the NORFREA algorithm elliminates redundant rules sucessively as it incorporates new ones. The process finishes when either all the rules are non redundant or there are no rules containing new information.

2.1. A simple example

Although the algorithm can be applied to any n-dimmensional system, let us consider a simple two-dimmensional classification problem to illustrate more clearly the steps performed by NORFREA.

Figure 1-a shows a set of 80 data grouped into 4 different classes with 20 data each one (*C1*, *C2*, *C3*, and *C4*). Looking at the data cloud, it could be apparent that a fuzzy inference system with 5 rules and 3 labels per input (small, medium, and large) could provide a high classification rate. Considering the input partition of Figure 1a, Figure 1-b illustrates the 6 out of the 9 rules which have certain degree of efficiency (measured by the performance index, P_R).

If we use the tool NEFCLASS-J [6] to solve this classification problem with 5 rules, the rules 1-5 are selected. They provide a classification rate of 90% (8 errors, all of them corresponding to the class *C2*). In this example, the problem with NEFCLASS-J is that this result is not improved by the learning process, even if no constraint is considered. The reason is that the selected rules are unable to distinguish between class 4 and the bottom part of class *C2*, because both parts are described by rule 3.

If we analyze the initial *small* and *medium* labels of the variable *x1*, we can notice they could be the same, that is, they are linkable labels. If the membership function defining the small label is spread to cover also the support of the initially defined *medium* label, the *medium* and the *large* labels can be displaced to best cover the right part of the *x1* universe of discourse and, hence, define more efficient rules to distinguish between classes *C2* and *C4*. With this spreading, the rule 5 would become redundant since it provides the same information as the rule 2. Hence, it would be replaced by the next rule in the list (the rule 6).

This is what NORFREA does. It selects the rules 1-4 and 6. Since the rule 6 is less efficient than the rule 5, this initial rule base has a smaller classification rate (75%) than that selected by NEFCLASS-J (90%). However, if we apply the learning tool of NEFCLASS-J to adjust the membership functions, the classification rate increases until 100%. This total efficiency is achieved despite the learning was constrained to maintain the normalization of the membership functions, that is, the sum of the membership degrees of every input to all the labels is always the unity value. Although this constraint is very restrictive, it provides a very easy linguistic interpretation. Figure 3 illutrates how the membership functions are adjusted by this constrained learning.

2.2. Preliminar definitions

Let us consider a classifier system that: (a) has *N* inputs and 1 output, (b) covers each input variable, *i*, with L_i linguistic labels, and (c) classifies a pattern into one out of *M* classes (C_1, ..., C_M). The set of rules it can contain can be defined as follows:

$$R \subseteq \{\{\{l_{1,1} \ldots l_{1L_1}\} \times \ldots \times \{l_{N,1} \ldots l_{N,L_N}\}\} \rightarrow \{C_1 \ldots C_M\}\}$$
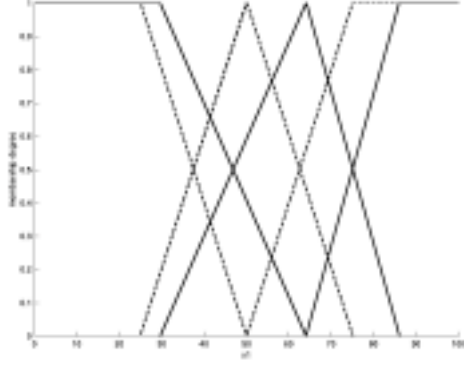
Figure 2: Membership functions of variable $x1$ before (dashed line) and after (solid line) the learning process.

Definition 1

Two linguistic labels of an input variable k, $l_{k,a}$ and $l_{k,b}$ with $a<b$, are said to be ***linkable*** when, if there are two rules that differ only in the label of the input k and being this label comprised between the beginning of the support of $l_{k,a}$ and the end of $l_{k,b}$, their consequents are the same. Mathematically, iff:

$$\not\exists\ r_p, r_q \in R \ | \ \text{if}\ r_p \equiv (l_{1,s_1}, ..., l_{k,i}, ..., l_{N,s_N}) \to c_p$$

and $r_q \equiv (l_{1,s_1}, ..., l_{k,j}, ..., l_{N,s_N}) \to c_q$

it happens that $c_p \neq c_q$ and $a \leq i < j \leq b$

Definition 2

Two rules r_p and r_q:

$r_p \equiv (l_{1,p_1}, ..., l_{k,p_k}, ..., l_{N,p_N}) \to c_p$

$r_q \equiv (l_{1,q_1}, ..., l_{k,q_k}, ..., l_{N,q_N}) \to c_q$

are said to be ***redundant*** iff:

$c_p = c_q$ and $(l_{k,p_i}$ and $l_{k,q_k})$ are linkable or equal $\forall k$

Definition 3

Two linkable labels $l_{k,a}$ and $l_{k,b}$ are said to be ***linked or to consitute a link*** when they are linked definitively after elliminating a redundant rule.

For the example described in Section 2.1, we will say that:
- The labels *small* and *medium* of the variable x1 are linkable;
- The rules $(small, small) \to C_1$ and $(medium, small) \to C_1$ are redundant rules;
- If we replace the rule $(medium, small) \to C_1$ by $(medium, medium) \to C_2$ the labels *small* and *medium* of the variable x1 are linked.

2.3. The algorithm definition

Figure 3-a shows the pseudo-code of NORFREA for solving a classification problem with k rules. Given an initial partition and its corresponding rule base, NOR-FREA divides the rule base into two sets: R_{select} and $R_{noselect}$. The most efficient k rules are included in R_{select} while all the other rules with a positive efficiency are included in $R_{noselect}$. The rules with a negative efficiency are elliminated because they never provide useful information.

The next step is to detect all the pairs of redundant rules within R_{select}, and to mark the least efficient rule of each pair as the candidate rule of the pair to be replaced. For every marked rule, NORFREA looks for a substitute in $R_{noselect}$. The process begins with the least efficient marked rules because it is better to not replace the most efficient redundant rules if it is not possible to replace all of them.

When looking for a rule to replace a redundant one, four cases can occur, as shown in Figure 3-b. In this figure, the redundant rule to be replaced is depicted with a shadow, the linked labels are surrounded by a solid ellipse, and the possible substituting rule is shown with a coming arrow. The four cases are the following:

- Case 1: The possible substituting rule breaks some of the created links. Hence, the algorithm looks for another substituting rule because no rule should break the link created by the ellimination of a more effective rule.

- Case 2: The rule does not break any link and it is non redundant. Hence, it is selected as substitute.

- Case 3: The rule does not break any link, it is redundant, and it is possible to create new links (shown in Figure 3-b with a dashed ellipse). In this case, the algorithm lookes for another substituting rule which is non redundant and creates the new tentative links (this is equivalent to select the rule but then trying to replace it by a non redundant rule).

- Case 4: The rule does not break any link, it is redundant but it is not possible to create new links. In this situation, the rule is selected as substitute despite being redundant because no link can be created to replace it.

The algorithm employs two variables related to the linkable labels: "links" and "tentative_links". This is done because the labels are not linked until the algorithm finds an adequate rule to replace the redundant rule (cases 2 and 4).

```
divide the rules into two groups:
    R_select ← {the most efficient k rules}
    R_noselect ← {the other rules with a positive performance index}
sort R_select by decreasing order of performance index
sort R_noselect by increasing order of performance index
detect redundant rules in R_select
links ← ∅
for each redundant rule r_s ∈ R_select and while R_noselect ≠ ∅ do
    while r_s does not substituted and R_noselect ≠ ∅ do
        tentative_links ← links ∪ {links created after elliminating the rule r_s}
        process the following rule r_r ∈ R_noselect
        if r_r do not break any tentative link then
            if r_r is a redundant rule then
                if it is possible to create the necessary links for elliminating the rule r_r then
                    (case 3)
                    tentative_links ← tentative_links ∪ {necessary links for elliminating the rule r_r}
                else
                    (case 4)
                    substitute r_s by r_r in R_select
                    links ← tentative_links
                end if
            else
                (case 2)
                substitute r_s by r_r in R_select
                links ← tentative_links
            end if
        end if
        (case 1)
        eliminate r_r from R_noselect
    end while
end
```

Case 1

Case 2

Case 3

Case 4
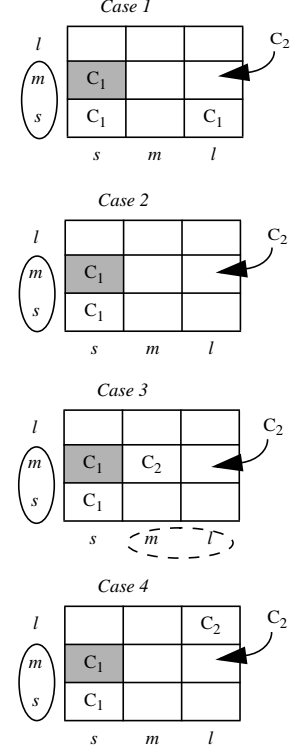
(a)   (b)

Figure 3: Algorithm for selecting fuzzy rules (NORFREA).

3. Results

The algorithm proposed in this contribution has been applied to two well-known clasification benchmarks: Fish [7] and Wisconsin Breast Cancer [8] datasets. The results obtained by using the rules selected by NEF-CLASS-J have been compared with those obtained by using the NORFREA rules. NEFCLASS-J has been also used to adjust the membership functions.

Fish

These data are seven measurements on six types of fishes caught in a lake in Finland. The last measurement has been excluded because it is a symbolic attribute and there are many samples in which this value is not defined. Our goal is to classify only 3 types of fish (Bream, Smelt and Perch), because the other ones have not an adequate number of samples. The incomplete samples have been removed, so that 104 samples remain.

At first, we tried to obtain a system with 3 rules, but only 2 classes were covered by these rules. At least 4 rules are required in order to cover all classes. NEF-CLASS-J selected the rules 1-4 shown in Figure 4. The classification rate was 75.00% and after the learning process without constraints the rate improved to 98.08%.

NORFREA detected that the rules 1 and 2 were redundant, and so it tried to substitute the rule 2. As the rule 5 correspond to the case 3 of the algorithm, the rule 6 was incorporated. Using the rules 1, 3, 4 and 6, the classification rate was 90.38%. So, the system performance was improved even before the optimization process. The reason is that some zones activate more than one rule and removing a rule does not necessarily lead to a lose in the system performance. Rules with a similar performance give better results when they are scattered than when they are closed. In this case, the improvement in the system performance is due to the NORFREA tendency to select scatted rules.

After the optimization process, our algorithm has improved the NEFCLASS-J results to a classification rate of 99.04%. Since no restriction has been applied in the optimization process the final results are not easy interpretable in linguictic terms. In this case, when the membership functions can be freely modified, the initial rule selection has a lower influence on the optimization results.

When the interpretation of results is required, it is necessary to impose some constraints to the learning process. In this case, and using 4 constraints at the same time: (a) keep relative order, (b) always overlap, (c) be symmetrical and (d) intersect at 0.5, the classification

rate obtained by NEFCLASS-J was 81.73% against 98.08% of NORFREA.

```
(1) if (s,s,s,s,s,m) then Perch        (P₁=0.1712)
(2) if (s,s,s,s,m,m) then Perch        (P₂=0.1471) Redundant with (1)
(3) if (m,m,m,m,l,m) then Bream        (P₃=0.1294)
(4) if (s,s,s,s,s,s) then Smelt        (P₄=0.1130)
(5) if (s,m,m,m,l,m) then Bream        (P₆=0.1070) Case 3, trying to replace (4)
(6) if (m,m,m,m,m,l) then Perch        (P₇=0.0856) Case 4, (2) is substituted
                                        ...


3 labels per variable: small (s), medium(m) and large (l)
Rules selected by NORFREA are shown in bold
```

Figure 4: Rules selected for the Fish dataset.

Wisconsin Breast Cancer (WBC)

These data are 9 different attributes of breast tumors grouped into 2 classes: 'Benign' and 'Malign'. Sixteen samples have been removed from the original dataset due to missing data, so that 683 samples remain. This example is included in the NEFCLASS distribution. The complete dataset is divided into two parts: one for traning and another for testing. In this example we have used the same partition.

We have needed 5 rules to cover both classes, because there are more samples for '*Benign*' than for '*Malign*', and, therefore, the rules with the '*Benign*' consequent have a higher performance index.

The Figure 5 shows the most efficient rules created by using 3 labels per variable. NEFCLASS-J selected the rules 1-5, getting a classfication rate of 65.20%. After the tuning process, this rate increased to 71.35% for the training data and 68.33% for the testing data. Considering all patterns, 69.84% of the samples was well-classified.

NORFREA detected three redundant rules (2-4) and substituted them for another three (8-10). After the change, the classification rate was the same, but the traning process improve this rate to 90.78% (92.98% for the training data and 88.56% for the testing data).

```
(1) if (s,s,s,s,s,s,s,s,s) then Benign    P₁=0.3910
(2) if (m,s,s,s,s,s,s,s,s) then Benign    P₂=0.1951 Redundant with (1)
(3) if (s,s,s,s,s,m,s,s,s) then Benign    P₃=0.0117 Redundant with (1)
(4) if (m,s,s,m,s,s,s,s,s) then Benign    P₄=0.0117 Redundant with (1)
(5) if (m,l,l,l,m,l,l,l,s) then Malign    P₅=0.0093
(6) if (m,s,m,s,s,s,s,s,s) then Benign    P₆=0.0074 Case 3, triying to replace (4)
(7) if (m,s,s,s,s,m,s,s) then Benign      P₇=0.0064 Case 3, triying to replace (4)
(8) if (m,l,l,l,m,l,m,m,s) then Malign    P₈=0.0058 Case 4, (4) is substituted
(9) if (l,l,l,l,l,l,l,l,s) then Malign    P₉=0.0056 Case 4, (3) is substituted
(10) if (l,l,l,m,m,s,l,l,s) then Malign   P₁₀=0.0056 Case 4, (2) is substituted
                                           ...


3 labels per variable: small (s), medium(m) and large (l)
Rules selected by NORFREA are shown in bold
```

Figura 5: Rules selected for the WBC dataset.

4. Conclusions

The algorithm presented in this paper, NORFREA, allows selecting fuzzy rules from a grid partition. Its novelty is that the selection criterion not only takes into account the performance index of each rule but also applies an heuristic which considers the subsequent learning process. This way, the algorithm selects an inital rule set that is better adjusted later. As a consequence, NORFREA increases not only the efficiency of the system but also its linguistic interpretation because the learning process provides good results even with restrictive constraints that ensure the linguistic meaning. The algorithm has been tested with different classification benchmarks and the results obtained have been better than those obtained with other algorithms based on the same structure-oriented fuzzy rule learning strategy.

References

[1] M. Sugeno, T. Yasukawa, *A Fuzzy-Logic-Based Aproach to Qualitative Modeling*, IEEE Trans. on Fuzzy Systems, Vol. 1, N° 1, February 1993, pp.7-31.

[2] F. Klawonn, R. Kruse, *Constructing a Fuzzy Controller from Data*, Fuzzy Sets and Systems,85, 1997, pp. 177-193.

[3] L. Wang, J.M. Mendel, *Generation Rules by Learning from Examples*. In International Symposium on Intelligent Control, pages 263-268, IEEE Press, 1991.

[4] D. Nauck, *Data Analysis with Neuro-Fuzzy Methods*. Habilitation thesis, Otto-von-Guericke University of Magdeburg, Faculty of Computer Science, Magdeburg, Germany, 2000.

[5] D. Nauck, R. Kruse, *NEFCLASS - A Neuro-Fuzzy Approach for the Classification of Data*. In K.M. George, J.H. Carrol, E. Deaton, D. Oppenheim and J. Hightower, eds.: Applied Computing 1995, Proc. 1995 ACM Symposium on Applied Computing, Nashville, Feb. 26-28, pp. 461-465. ACM Press, New York.

[6] D. Nauck, R. Kruse, *Function Approximation by NEFPROX*. In Proc. Second European Workshop on Fuzzy Decision Analysis and Neural Networks for Management, Planning, and Optimization (EFDAN'97), pp. 160-169, Dortmund.

[7] Obtained by Pekka Brofeldt and available from *http://www.amstat.org/publications/jse/datasets/fishcatch.dat*

[8] Obtained by Dr. William H. Wolbergthe and available from *ftp://ftp.ics.uci.edu/pub/machine-learning-databases/*