Generating an Interpretable Family of Fuzzy Partitions From Data

Serge Guillaume and Brigitte Charnomordic

***Abstract*—In this paper, we propose a new method to construct fuzzy partitions from data. The procedure generates a hierarchy including best partitions of all sizes from n to two fuzzy sets. The maximum size n is determined according to the data distribution and corresponds to the finest resolution level. We use an ascending method for which a merging criterion is needed. This criterion is based on the definition of a special metric distance suitable for fuzzy partitioning, and the merging is done under semantic constraints. The distance we define does not handle the point coordinates, but directly their membership degrees to the fuzzy sets of the partition. This leads to the introduction of the notions of internal and external distances. The hierarchical fuzzy partitioning is carried independently over each dimension, and, to demonstrate the partition potential, they are used to build fuzzy inference system using a simple selection mechanism. Due to the merging technique, all the fuzzy sets in the various partitions are interpretable as linguistic labels. The tradeoff between accuracy and interpretability constitutes the most promising aspect in our approach. Well known data sets are investigated and the results are compared with those obtained by other authors using different techniques. The method is also applied to real world agricultural data, the results are analyzed and weighed against those achieved by other methods, such as fuzzy clustering or discriminant analysis.**

***Index Terms*—Distance, fuzzy partitioning, interpretability, learning, rule induction.**

I. Introduction

FUZZY inference systems have proven useful to represent a system behavior by means of IF–THEN fuzzy rules. Fuzzy rules can be based on expert knowledge available from human experts. This point of view, which seems natural, was historically the first one to be implemented, as in [1]. However, it soon appeared that for complex partially unknown systems the interactions are very difficult to grasp and expert rules are not sufficient to yield a satisfactory simulation of the system. For this reason, fuzzy rule induction from data has been given a lot of attention in the recent literature [2]. Such approaches are mostly inherited from numerical learning techniques, such as neural networks or evolutionist algorithms. They typically seek to optimize the numerical performance while interpretability of the induced rules is not their first concern. In many cases when induced rules meaning matters, this is a serious drawback. It is necessary to develop new fuzzy rule induction methods, so that the semantic integrity of the fuzzy inference system is guaranteed. A recent review of rule induction methods [3] has shown that the fuzzy partition for the system inputs is then of prime importance. The fuzzy sets are to be interpretable as linguistic labels to allow the cooperation between expert rules and induced rules. This may be contradictory with the numerical error minimization objective.

Focusing on the interpretability, this paper presents a new method for deriving fuzzy partitions from data. Although being generic, it has been designed for dealing with complex multidimensional systems, such as food processes.

The proposed approach is called hierarchical fuzzy partitioning (HFP) and is inspired from two different clustering methods. It has some similarities with hierarchical clustering which is widely used in Statistics, while it shares other points with clustering techniques adapted to the fuzzy formalism.

Hierarchical clustering makes clusters of multidimensional data pairs according to a given criterion. The starting point is a n-cluster partition, each cluster containing a single individual. The final partition obtained by recursive group aggregating is a one-cluster partition including all data pairs. At each stage the two "nearest" clusters are combined to form one bigger cluster. The commonly used Ward criterion combines the two clusters which least increase the within cluster variance.

Fuzzy clustering methods, such as fuzzy c-means [4], find a partition of the observations into a predetermined number of groups. The data points are divided into groups of points that are "close" to each other. Each data point belongs to a group or cluster with a given membership degree. Closeness between data points is defined by a metric distance, and each metric yields a different partitioning. The importance of the concept of distance and the sensitivity of the results with respect to the choice of different distances has often been underlined in clustering [4], [5], but not in fuzzy partitioning. Many metrics have been tried out, but none of them takes account of the partition structure. At best, it is related to cluster shape, as in [6]. Some authors also defined distances between fuzzy sets [7], [8] for approximate reasoning. Some of these distances fulfill the triangle inequality [9], [10], other ones are pseudo metrics only [11]. The distance introduced by [12], [13] is close to human appreciation.

In our approach, we wish to derive a fuzzy partition from data in each dimension. Instead of making data point clusters, we aggregate fuzzy sets. As in hierarchical clustering, we start from an initial n-item partition, and end with a one-item set partition. However, the items to be clustered are fuzzy sets instead of data points. At each stage the procedure aggregates two fuzzy sets to form a new wider range one. To aggregate we use a pairwise data

Manuscript received March 28, 2001; revised May 9, 2002, February 14, 2003, and October 8, 2003.

S. Guillaume is with Cemagref, 34196 Montpellier Cedex 5, France (e-mail: serge.guillaume@montpellier.cemagref.fr).

B. Charnomordic is with INRA, LASB, 34060 Montpellier, France (e-mail: bch@ensam.inra.fr).

Digital Object Identifier 10.1109/TFUZZ.2004.825979

point distance, which takes account of the particularities of the items to be merged: fuzzy sets within a given partition. In other words, this distance takes fuzzy partitioning into consideration, and is used as the basis of the aggregating criterion.

The aggregating procedure merges the fuzzy sets under semantic constraints, one of them being the fuzzy set distinguishability. Thus, the fuzzy partitions only contain fuzzy sets which can be read as linguistic labels. The method is carried out independently over each input dimension, and the partitions are used at a further stage to define the fuzzy rule premises in a fuzzy inference system.

These fuzzy inference systems can be useful for modeling a complex system and extracting elements of knowledge from data in a variety of cases. Indeed rule induction done this way is very different from rule induction based on fuzzy clustering. In Fuzzy Clustering, fuzzy rules are built from clusters of multidimensional data pairs. Each rule premise has its own fuzzy sets, which are obtained by projecting the corresponding cluster onto each dimension. For a given dimension, the fuzzy partition results from the union of the fuzzy sets for all rules. The fuzzy set distinguishability, which is essential for semantic integrity, is not guaranteed and is even unlikely to be met.

The paper is organized as follows. Section II outlines the overall HFP procedure, introducing the concepts of internal and external distance used in the merging criterion. Section III presents a distance metric suitable for fuzzy partitioning. Section IV is centered about the concept of partition validity.

The fuzzy partition families can serve as an input for other methods that intend to build fuzzy inference systems. Although the goal of this paper is not to propose a complete system generation method, the potential is illustrated by generating fuzzy inference systems of increasing complexity through a simple algorithm. Section V explains the fuzzy inference system generation and selection algorithm. Section VI gives the results obtained on several well-known data sets, available in the machine learning repository,[1] and compares them with those reported in [14]. Section VII presents a detailed case study of real world agricultural data, and comments the results with respect to other techniques.

Finally, Section VIII gives some conclusions.

II. Hierarchical Fuzzy Partitioning

The HFP method generates a collection of univariate fuzzy partitions from a multidimensional training dataset. The dataset, denoted E, is a collection of N multiple-input–single-output numerical data pairs (x_k, y_k), $k = 1, 2, \ldots, N$ where x_k is the p-dimensional input vector $x_k^1, x_k^2, \ldots, x_k^p$ and y_k is the one-dimensional output vector.

To make computation independent of measurement units, all data are scaled into the unit space.

A univariate fuzzy partition is composed of m fuzzy sets, the fth fuzzy set for the jth input variable being defined by its membership function $(x, \mu_j^f(x))$.

The procedure is carried independently over all dimensions. In each dimension it builds a family of fuzzy partitions as follows.

[1]http://www.ics.uci.edu/~mlearn/MLRepository.html

The initial fuzzy partition is determined by choosing M_j fuzzy sets according to the data sample distribution in the considered dimension, with $M_j \leq N$.

The family of fuzzy partitions is obtained using recursive fuzzy set merging so that at each step, the resulting partition is of size m, $2 \leq m \leq M_j - 1$, and best satisfies a merging criterion. Each merging modifies at most four fuzzy sets, the two being merged and their immediate neighbors when they exist. The final partition is composed of a single fuzzy set which covers the entire data range in the considered dimension.

The HFP procedure can be summarized as a sequence of $(M_j - 1, M_j - 2, \ldots, 2)$ iterations in each dimension, as shown in Algorithm 1. D_m is the criterion for merging two fuzzy sets. It will be given in Section II-C, (3).

Algorithm 1 Hierarchical fuzzy partitioning

```
base partition = FP^m = initial partition of size M_j
set m = M_j
while m > 2 do
  evaluate D_m; s = 1
  while s ≤ m − 1 do
    merge fuzzy sets s and s + 1, modify neighboring fuzzy sets
    evaluate D^s_{m−1}
    restore base partition
    s = s + 1
  end while
  select and store the partition FP^{m−1} for which:
    D_{m−1} = argmax(D^s_{m−1})
  base partition = FP^{m−1}; m = m − 1
end while
```

The next section explains how to choose M_j, and the initial fuzzy set location. The merging procedure is then presented. The last section gives the definition of the merging criterion, and introduces the important notions of internal and external distance.

A. Choice of the Initial Fuzzy Partition

The proposed approach is applicable regardless of the shape of the fuzzy sets. Both for computational time considerations and for clarity in demonstrating the method we chose all fuzzy sets of triangular shape, except at the domain edges, where they are semi trapezoidal.

The fuzzy sets are labeled $1, 2, \ldots, m$ and they overlap so that the fuzzy partition is standardized as follows:

$$\begin{cases} \forall x \sum\limits_{f=1,2,\ldots,m} \mu_j^f(x) = 1 \\ \forall f \; \exists x \; \mu_j^f(x) = 1 \end{cases}. \tag{1}$$

This choice is justified by the preoccupation of semantic integrity, which guarantees that the membership functions will represent a linguistic concept. It is discussed in great detail in [15]–[17].

Each triangle fuzzy set f is defined by its breakpoints left^f, c^f, right^f. A standardized fuzzy partition can be built by choosing fuzzy set breakpoints as shown in Fig. 1. Two contiguous fuzzy sets cross at the point of membership value $\mu = 0.5$, and three contiguous fuzzy sets f, g, h have such boundaries as

$$\begin{cases} \text{left}^g = c^f \\ \text{left}^h = c^g \\ \text{right}^f = c^g \\ \text{right}^g = c^h \end{cases}.$$

The first and last fuzzy sets in the partition are semi trapezoidal, with respective breakpoints $k^1_{inf}, k^1_{sup}, right^1$ and $left^m, k^m_{inf}, k^m_{sup}$ such as

$$\begin{cases} k^1_{\text{inf}} = \text{data lower bound} \\ k^1_{\text{sup}} = \text{left}^2 \text{ noted } c^1 \\ \text{right}^1 = c^2 \end{cases} \quad \begin{cases} left^m = c^{m-1} \\ k^m_{\text{inf}} = \text{right}^{m-1} \text{ noted } c^m \\ k^m_{\text{sup}} = \text{data upper bound}. \end{cases}$$

By construction, all points at most belong to two fuzzy sets. We also have $\forall x \max_{f=1,2,\ldots,m} \mu^f_j(x) \geq (1/2)$.

Each fuzzy set is assigned a weight equal to its cardinality, noted w^f for fuzzy set f

$$w^f = \sum_{x \in E} \mu^f_j(x). \tag{2}$$

In all rigor the initial partition could include as many fuzzy sets as N, the number of pairs in the training dataset. The kth triangular membership function would then be centered on x^j_k. In practice, the initial partition size can be reduced. The goal is to accelerate the procedure without a loss of performance. We therefore form M_j clusters of so-called unique c^j_m values. These unique values are determined by sorting the $x^j_k, k = 1, 2, \ldots, N$ values and setting an equality threshold tol. The cluster center $c^j_m, m = 1, \ldots, M_j$, is defined as the average of all values that fall within the cluster. Finally, each cluster center is used as a fuzzy set center. The initial fuzzy set weight, defined in (2), is equal to the number of observations in the cluster.

Sensitivity to the number of unique values and the choice of tol will be studied in Section VII. If data are numerical measurements, the meaning of tol can be related to their numerical resolution.

B. Fuzzy Set Merging

Recursive fuzzy set merging is a multistep procedure, that reduces the fuzzy partition size by one at each step. Merging is restricted to adjacent fuzzy sets, and seeks the best possible arrangement according to a given criterion. That criterion will be given in the next section and is to be computed for every possible fuzzy set combination.

Merging two fuzzy sets labeled 2 and 3 is illustrated in Fig. 2. The resulting fuzzy set is labeled $2'$ and defined as follows:

$$\begin{cases} \text{left}^{2'} = c^1 \\ c^{2'} = \frac{w^2 c^2 + w^3 c^3}{w^2 + w^3} \\ \text{right}^{2'} = c^4 \end{cases}.$$

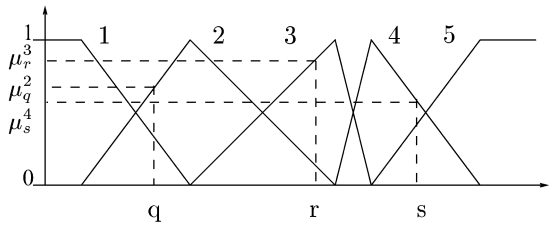


Fig. 1. Standardized fuzzy partition.

The neighboring fuzzy sets 1 and 4 are turned into $1'$ and $3'$. Their left and right breakpoints are modified so that the fuzzy partition is kept standardized. Fuzzy weights w^1, w^2, w^3 need to be updated after the merging, according to (2).

C. Merging Criterion

We seek a partition level index to be used in the merging process that summarizes the partition structure with regard to the data points. For that purpose, a special metric

$$d(x^j_q, x^j_r) = d(q, r)$$

is needed that allows to define pairwise dissimilarity coefficients, so called distances, while taking account of the fuzzy partition structure. Such a metric will be proposed in the next section. For now, let us outline only one key feature necessary to understand the approach.

Consider two data points with respective x^j_q, x^j_r coordinates in the jth dimension. Due to the fuzzification procedure, they can belong to several fuzzy sets with a non zero degree. To alleviate the notations we will denote $\mu^f_q = \mu^f_j(x^j_q)$.

Two nonexclusive cases are to be distinguished.

1) x^j_q and x^j_r partially belong to the same fuzzy set f, $\mu^f_q > 0$, $\mu^f_r > 0$.
2) x^j_q and x^j_r partially belong to two different sets f and g, $\mu^f_q > 0$, $\mu^g_r > 0$, $f \neq g$.

We introduce the terms of internal distance in the first case, external distance in the second one.

We impose a fundamental restriction to insure that the distance will reflect the partition structure and preserve the fuzzy set label semantic. Two points which mainly belong to the same fuzzy set will always be considered closer than others which mainly belong to distinct fuzzy sets.

The pairwise distance $d(q, r)$ will take into account q and r memberships to the various fuzzy sets by combining the respective parts of internal and external distances. A given size m partition can then be characterized by the sum of pairwise distances over all the data points

$$D_m = \frac{1}{N(N-1)} \sum_{q,r=1,2,\ldots,N,\ q \neq r} d(q, r). \tag{3}$$

During the merging process, the number of fuzzy sets is reduced by one at each stage. Obviously, some external distances become internal distances, inducing a change on the D_m index. On Fig. 2, this is the case for all $d(q, r)$, $x^j_q \in [c_1, c_3]$, $x^j_r \in [c_2, c_4]$, when considering memberships to fuzzy sets 2 and 3.

The best merge at a given stage can be considered as the one that minimizes the variation of D_m. The underlying idea is to

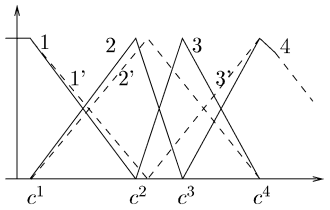


Fig. 2. Merging fuzzy sets 2 and 3 results in $2'$, $1 \Rightarrow 1'$, $4 \Rightarrow 3'$.

maintain as far as possible the homogeneity of the structure built at the previous stage.

Due to the fact that internal distances are smaller than external ones, the sum of distances decreases, except for some particular cases in the very first steps of the procedure.

The merging algorithm has a reduced complexity. Assuming a m size partition at a given step in a given dimension, the number of possible merges is equal to $m-1$. The D_m index is computed on the prototypes resulting from the preliminary stage, whose number can be reasonably bounded according to the chosen tolerance tol.

Let us now specify the distance metric in use.

III. Distance Metric Suitable for Fuzzy Partitioning

In the previous section, we introduced the notions of internal and external distances related to a fuzzy partition. We will now give a definition of both of them and see how they can be combined to deal with the multiple membership characteristic of fuzzy logic. Let us first recall some basic properties of a distance.

A. Distance Properties

A function d is a dissimilarity if

$$\forall\, q,r, \qquad \begin{cases} d(q,r) \geq 0 \\ d(q,q) = 0 \\ d(q,r) = d(r,q) \end{cases}. \tag{4}$$

A dissimilarity is semiproper if

$$d(q,r) = 0 \quad \Rightarrow \qquad \forall\, s \quad d(q,s) = d(r,s). \tag{5}$$

A dissimilarity is proper if

$$d(q,r) = 0 \quad \Rightarrow \quad q = r. \tag{6}$$

A semidistance is a dissimilarity which verifies the triangle inequality

$$\forall\, q,r,s \quad d(q,r) \leq d(q,s) + d(r,s). \tag{7}$$

A proper semidistance is called a distance.

We limit our study to convex standardized fuzzy sets and check the properties of the internal and external distances we define.

B. Internal Distance

The membership degree complement $(1-\mu_q^f)$ can be interpreted as the distance of x_q^j to the fuzzy set f. It measures the similarity of x_q^j to the fuzzy set prototypes that delimit the kernel. Recall that a prototype is such that $\mu(x) = 1$. Given two data points with (x_q^j, x_r^j) coordinates, we compute the internal distance by differencing the prototypes similarities, which comes to differencing the membership degrees

$$d_{\mathrm{int}}^f(q,r) = \left|\mu_q^f - \mu_r^f\right|.$$

Property (4) is trivial and (5) is easily checked. Counter examples for property (6) are also easy to find. Many distinct data pairs have an identical membership degree, yielding a null internal distance, as illustrated in Fig. 3.

A (q,r,s) triplet is relevant to one of the following three cases for which property (7) is to be checked.

1) Trivial case: identical membership for all three points: $\mu_q^f = \mu_r^f = \mu_s^f$.
2) Identical membership for two points: $\mu_q^f = \mu_r^f$ and $\mu_q^f \neq \mu_s^f$, with for instance $\mu_q^f > \mu_s^f$.

   The following inequalities are to be proven:

$$\begin{aligned} &d_{\mathrm{int}}^f(q,r) \leq d_{\mathrm{int}}^f(q,s) + d_{\mathrm{int}}^f(r,s), \ \text{i.e.,} \ 0 \leq 2(\mu_q^f - \mu_s^f) \\ &d_{\mathrm{int}}^f(q,s) \leq d_{\mathrm{int}}^f(q,r) + d_{\mathrm{int}}^f(r,s), \ \text{i.e.,} \ \mu_q^f - \mu_s^f \leq \mu_q^f - \mu_s^f \\ &d_{\mathrm{int}}^f(r,s) \leq d_{\mathrm{int}}^f(q,r) + d_{\mathrm{int}}^f(q,s), \ \text{i.e.,} \ \mu_q^f - \mu_s^f \leq \mu_q^f - \mu_s^f. \end{aligned}$$

3) All membership degrees are distinct, for instance $\mu_q^f < \mu_r^f < \mu_s^f$.

   The inequalities to be checked are written as

$$\begin{aligned} &\mu_r^f - \mu_q^f \leq \mu_s^f - \mu_q^f + \mu_s^f - \mu_r^f \ \text{ then } \ \mu_r^f \leq \mu_s^f \\ &\mu_s^f - \mu_q^f \leq \mu_r^f - \mu_q^f + \mu_s^f - \mu_r^f \ \text{ then } \ \mu_s^f - \mu_q^f \leq \mu_s^f - \mu_q^f \\ &\mu_s^f - \mu_r^f \leq \mu_r^f - \mu_q^f + \mu_s^f - \mu_q^f \ \text{ then } \ \mu_q^f \leq \mu_r^f. \end{aligned}$$

   In all cases, the proof is straightforward, therefore, the proposed internal distance function d_{int}^f is a semidistance.

C. Prototype Distance

We propose two different definitions of the distance $d_{\mathrm{prot}}(f,g)$ between the prototypes of fuzzy sets f and g.

1) A numerical prototype distance

$$d_{\mathrm{prot}}^{\mathrm{num}}(f,g) = \sqrt{(c_f - c_g)^2}$$

   where c_f, c_g are the respective fuzzy set kernel locations. This definition corresponds to the kernel Euclidean distance.

2) A more symbolic prototype distance

$$d_{\mathrm{prot}}^{\mathrm{sym}}(f,g) = \frac{g-f}{m-1} \tag{8}$$

   where m is the partition size, f and g are the indexes of the fuzzy sets sorted in ascending order relatively to the center coordinates.

Within the partition illustrated in Fig. 1, the symbolic choice for the prototype distance makes the fuzzy set 3 at the same distance from 2 and 4, while the numerical choice puts it closer to 4. The symbolic distance is more faithful to the symbolic representation.

Both definitions can easily be checked to fulfill conditions (6) and (7).

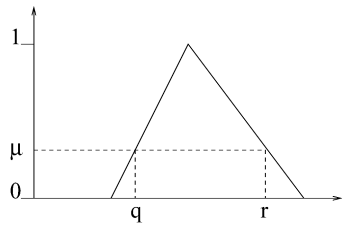


Fig. 3. Internal distance $d(q, r)$ equals zero.

D. External Distance

The external distance must take account of the point location within its reference fuzzy set, and of the relative fuzzy set location within the fuzzy partition, which implies combining the internal and the prototype distances.

We propose the following definition for the external distance between two points which belong to f and g:

$$d_{\text{ext}}^{f,g}(q,r) = \left|\mu_q^f - \mu_r^g\right| + d_{\text{prot}}(f,g) + D_c \tag{9}$$

where D_c is a constant correction factor, which ensures that the external distance is always superior to any internal distance. Note that the external distance reduces to the prototype distance plus the correction factor, when points q, r have internal identical membership degrees.

Fig. 1 can be used to illustrate external distances on the (q, r, s) triplet. When considering fuzzy sets 2–4, external distances can be written as

$$\begin{aligned} d_{\text{ext}}^{2,3}(q,r) &= \mu_r^3 - \mu_q^2 + d_{\text{prot}}(2,3) + D_c \\ d_{\text{ext}}^{3,4}(r,s) &= \mu_r^3 - \mu_s^4 + d_{\text{prot}}(3,4) + D_c \\ d_{\text{ext}}^{2,4}(q,s) &= \mu_q^2 - \mu_s^4 + d_{\text{prot}}(2,4) + D_c \end{aligned}$$

which proves the triangle inequality (7).

There are other external distances concerning the (q, r, s) triplet. They are dealt with in the same way, and we now examine the problem of distance combination.

E. Distance Combination and Continuity

To manage multiple memberships, the pairwise distance $d(q,r)$ is taken as a combination of the internal and external distances previously defined, depending on the number of fuzzy sets for which μ_q and μ_r are different from zero.

Let us denote $d_{f,g}(q,r)$ the partial (q, r) distance that represents respective memberships to f and g. It is an internal distance if $f = g$, an external distance otherwise.

$d(q,r)$ results from the combination of at most m^2 distances

$$d(q,r) = \frac{1}{\sum_{f=1}^{m} \mu_q^f} \sum_{f=1}^{m} \left[\mu_q^f \frac{1}{\sum_{g=1}^{m} \mu_r^g} \sum_{g=1}^{m} \left[\mu_r^g \; d_{f,g}(q,r)\right] \right]. \tag{10}$$

For a standardized fuzzy partition, as defined in (1), $d(q,r)$ is a combination of at most four distances and all denominators in the previous formula are equal to 1.

One point q is said to mainly belong to a fuzzy set f if $\mu_q^f \geq 0.5$. Consequently, the pairwise distance $d(q,r)$ will be mainly internal when both points q, r mainly belong to the same fuzzy set. These points must be closer than points whose distance is mainly external to enforce the fundamental constraint given in Section II-C. Due to our implementation, the maximum value of a mainly internal distance is 0.5. Therefore, we set $D_c = 0.5$ in (9).

The term $d(q,r)$ has been shown to be a combination of semidistances. Thus, it is a semidistance. Nevertheless, to alleviate the notations we will refer to d as a distance.

IV. Validity Criterion

What is a *good* partition? This question, widely studied, is still open. There is no universal answer. Within the supervised learning framework one can assess the performance of the corresponding fuzzy inference systems. Nevertheless, the performance depends on many factors: induction method, number of rules, variable selection, ... which makes it a challenge to decide on the quality of the partition itself. To assess the validity of a fuzzy partition we propose a new index based on the homogeneity of the fuzzy set densities.

The fuzzy set density, called d_f for fuzzy set f, is equal to the ratio of its weight, or fuzzy cardinality, w^f, defined in (2), to the fuzzy set area. The overlapping subareas are excluded, as shown in Fig. 4, for more robustness regarding the standardized fuzzy partition construction. The density homogeneity, σ^{FP}, is defined as the density standard deviation for all the fuzzy sets of the partition: $\sigma^{FP} = \sqrt{(1/m)\sum_{f=1}^{m}(d_f - \overline{d})^2}$, $\overline{d}$ being the mean of the fuzzy set densities. A good, steady partition is expected to be homogeneous, i.e., to have a small standard deviation.

σ^{FP} is not significant for the first steps of the merging process, all the fuzzy set are designed to be properly filled up. It is useful for the last steps. From the homogeneity point of view the best partition is the one for which σ^{FP} reaches a minimum. Checking the evolution of σ^{FP} versus the partition size is informative. Local minima and singular points can be found.

Illustration on the Iris Data*:* The iris data [18] are 150 items, representing four numerical measurements: Petal Length, Petal Width, Sepal Length, and Sepal Width, for three different species Setosa, Virginica, and Versicolor. We applied the HFP method to the four numerical features. The Petal Width histogram is plotted in Fig. 5. In the bottom part of this figure, the fuzzy set centers, for the last steps of the HFP procedure (partition size six to two), are reported together with σ^{FP} values. The Fig. 6 shows the corresponding results for Petal Length.

The results are in favor of a three fuzzy set partition. One can note that σ^{FP} reaches a lower minimum for Petal Width than for Petal Length, leading to believe that the Petal Length partition fuzzy set densities are more heterogeneous.

Complexity Analysis*:* To assess the computational load of the whole procedure we consider the generation of fuzzy partitions for a given mono dimensional variable (Algorithm 1 in Section II).

This algorithm complexity is measured according to the number of fuzzy sets in the initial partition, M. As explained in Section II-A, this number results of a clustering. The number of clusters depends both on the data distribution and the equality tolerance threshold tol. As the values are scaled into the unit space, tol expresses a relative variation: A value of 2% leads to

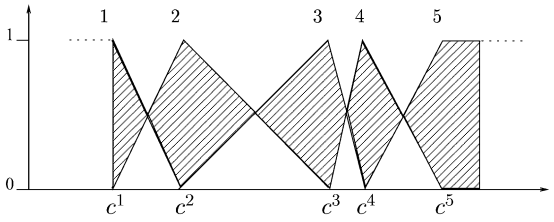
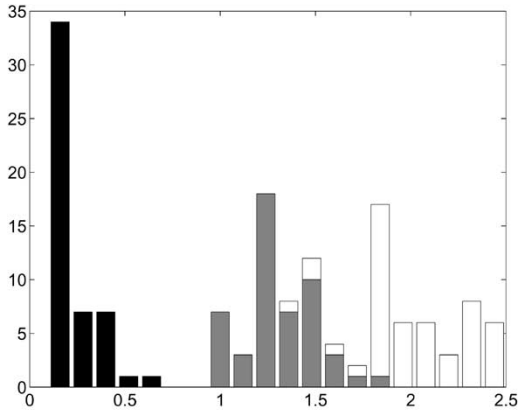
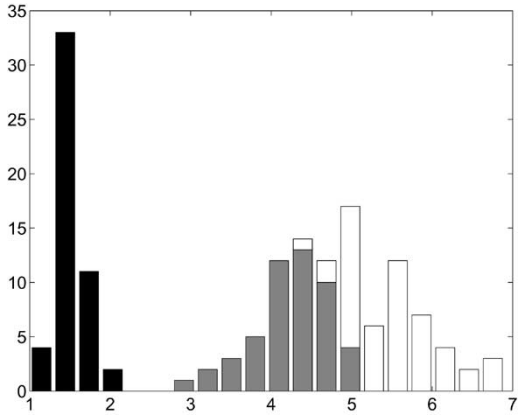


Fig. 4. Areas used for computing the fuzzy set densities.



σ^{FP}	Fuzzy set centers					
5.22	0.83	2.24				
4.87	0.10	1.40	2.24			
7.54	0.10	0.90	1.79	2.24		
12.75	0.10	0.32	1.36	1.79	2.24	
12.20	0.10	0.32	1.22	1.52	1.79	2.24

Fig. 5. Validity indicator evolution for iris petal width.



σ^{FP}	Fuzzy set centers					
13.34	3.18	6.67				
12.78	1.34	4.53	6.67			
19.41	1.34	3.17	5.16	6.67		
29.14	1.34	1.63	4.17	5.16	6.67	
29.41	1.34	1.63	4.17	4.81	5.57	6.68

Fig. 6. Validity indicator evolution for iris petal length.

a maximum of 50 clusters. The considered algorithm generates $M-2$ partitions, each of them results from $M-1$ attempts. A try is characterized by the the sum of $M \times M-1$ distances [see (3)]. The HFP algorithm global complexity is thus measured by $O(M^4)$.

V. Fuzzy Inference System Generation and Selection

Our objective in this part is to generate fuzzy inference systems (FIS), using a refinement procedure based on a known hierarchy of fuzzy set partitions of increasing size. This hierarchy can be the result of the HFP stage, it can also be obtained by other means, for instance a series of regular grids of different sizes. This will allow us to compare the results yielded by different hierarchies.

We start by considering the simplest system, which has only one rule including a single fuzzy set in each dimension $j = 1, \ldots, p$. The selection procedure builds new systems by refining the fuzzy partitions.

The refinement algorithm is detailed in Section V-B. It calls a FIS generation algorithm described in Section V-C.

First, we give the definition of some elements that will be used all along.

A. Definitions

Fuzzy Partition Notation: In each dimension, the HFP procedure yields a family of partitions of decreasing size. For a given dimension j, let us denote $FP_j^{n_j}$ the HFP generated fuzzy partition of size n_j, $n_j^{\max}$ being the maximum size of the partition (see Section II-A), $n_j^{\max} \leq M_j$ by construction. To improve interpretability, $n_j^{\max}$ is limited to a reasonable number (≈ 7) [19].

$FP_j^{n_j}$ is uniquely determined by its size n_j, the fuzzy set centers being the coordinates given by the hierarchy, $FP_j^{n_j} = \{MF_j^{k/n_j}, k = 1, \ldots, n_j\}$, where MF_j^{k/n_j} refers to the kth membership function of the fuzzy partition for the jth variable.

Performance Index*:* Two cases are to be considered.

1) Numerical output (regression case):
   The numerical performance index is chosen as the root mean square error over the training sample

$$\text{Perf} = \frac{1}{N}\sqrt{\sum_{i=1}^{N} ||\widehat{y}_i - y_i||^2}$$

   where $N$is the sample size, y_i the observed output for the ith example, and $\widehat{y}_i$ the inferred output for the ith example.
2) Nominal output (classification case):
   The performance index is equal to the number of misclassified items

$$\text{Perf} = \sum_{i=1}^{N} \delta_i \quad \begin{cases} \delta_i = 1, & \text{if } \widehat{y}_i \neq y_i \\ \delta_i = 0, & \text{otherwise} \end{cases}.$$

Blank Examples*:* A rule potentially covers the subset of the multidimensional input space corresponding to the combination of the fuzzy sets composing its premise. The rth rule will be activated by the ith example to a degree, called rule weight

$$rw^r(x_i) = \mu_{MF_1^{a/n_1}}\left(x_i^1\right) \wedge \ldots \wedge \mu_{MF_p^{q/n_p}}\left(x_i^p\right) \qquad (11)$$

where $\wedge$ is a T-norm operator for fuzzy set intersection.

The ith example will be considered as inactive or blank for a given rule r if $rw^r(i) \leq \mu_{\min}$, $\mu_{\min}$ being a fixed threshold value.

We call E^r the subset of nonblank examples for the rth rule. It is a subset of the learning sample E such as

$$E^r = \{(x_i, y_i) \in E \mid rw^r(x_i) > \mu_{\min}\}. \tag{12}$$

The examples in E^r are sorted by descending order of rw^r. They are said to fire the rule r.

In the same way, an example is said to be blank for the $(r_1, r_2, \ldots, r_R)$ rule base if $\sum_{r=1}^{R} rw^r(x_i) \leq \mu_{\min}$. Due to the cumulated sum, some examples which are blank for all rules may not be blank for the rule base. This could be avoided by using a max operator.

Note: The presence of blank examples should not always be considered as a drawback. It is a voluntary choice to make the rule base contain only general rules, and not too specific ones. This yields a smaller number of rules and a parsimonious fuzzy system.

B. Refinement Procedure

The iterative algorithm is presented below. It is not a greedy algorithm, unlike other techniques. It does not implement all possible combinations of the fuzzy sets, but only a few chosen ones.

Algorithm 2 Refinement procedure
1: iter $= 1$; $\forall j \ n_j = 1$
2: CALL *FIS Generation* (Algorithm 3)
3: **while** $iter \leq iter_{\max}$ **do**
4: Store system as base system
5: **for** $1 \leq j \leq p$ **do**
6: **if** $n_j = n_j^{\max}$ **then** next j (partition size limit reached for input j)
7: $n_j = n_j + 1$
8: CALL *FIS Generation* (Algorithm 3)
9: $\text{Perf}_j = \text{Perf}$
10: $n_j = n_j - 1$
11: Restore base system
12: **end for**
13: **if** $\forall j \ n_j = n_j^{\max}$ **then** exit (no more inputs to refine)
14: $s = \text{argmin} \ \{\text{Perf}_j, \ j = 1, \ldots, p, \quad n_j < n_j^{\max}\}$ (Select input to refine)
15: $n_s = n_s + 1$
16: CALL *FIS Generation* (Algorithm 3)
17: keep FIS_{iter}
18: iter $=$ iter $+ 1$
19: **end while**

The key idea is to introduce as many variables, described by a sufficient number of fuzzy sets, as necessary to get a good rule base. A good FIS represents a reasonable tradeoff between complexity, in relationship with the number of rules, and accuracy, measured by the performance index.

The refinement procedure is responsible for the selection of the variables or fuzzy sets to be introduced in the FIS. The initial FIS is the simplest one possible (Algorithm 2, lines 1 and 2). The search loop (lines 5–12) builds up temporary fuzzy inference systems. Each of them corresponds to adding to the initial FIS one fuzzy set in a given dimension. The selection of the dimension to retain is done in lines 14 and 15. Following this selection, a FIS to be kept is built up. It will serve as a base to reiterate the sequence (lines 3–19). Thus the result of the procedure is not a single FIS, but a series $\text{FIS}_1, \text{FIS}_2, \ldots$ of increased complexity.

When necessary, the procedure calls a FIS generation algorithm, referred to as Algorithm 3, which is now detailed.

C. FIS Generation

A fuzzy inference system is completely defined by its rule base and the inference method.

The rule generation is done by combining the fuzzy sets of the $FP_j^{n_j}$ partitions for $j = 1, \ldots, p$, as described by Algorithm 3. The algorithm then removes the less influential rules and evaluates the rule conclusions. The condition stated in line 5, where CV_t is a given threshold, ensures that the rule is significantly fired by the examples of the training set.

Algorithm 3 FIS generation
Require: $\{n_j | j = 1, \ldots, p\}$
1: get $FP_j^{n_j} \forall j = 1, \ldots, p$
2: Generate the $\prod_{j=1}^{p} n_j$ rule premises
3: **for all** Rule $r \in \text{FIS}$ **do**
4: $CV_r = \sum_{k=1}^{n} rw^r(x_k)$
5: **if** $CV_r < CV_t$ **then** remove rule r
6: **else** initialize rule conclusion
7: **end for**
8: Compute Perf

The rule conclusion initialization, line 6, depends on the system output type. In the following we consider the case of a nominal output (classification problem). The rule conclusion is then initialized as the most frequent output label in E_r, and the FIS output, $\hat{y}_i$, is an integer value obtained by rounding off the result of a simple weighted average defuzzification procedure (Sugeno type inference).

Other inference methods, including fuzzy rule conclusions and more sophisticated defuzzification procedures, can be implemented for a numerical output, without changing the FIS generation algorithm itself.

D. Final Choice

As we said above, the outcome of the procedure is not a single fuzzy inference system, but K FIS of increasing complexity. The selection of the best one takes into consideration the performance and the number of blank examples. We propose the following simple criterion.

$\text{FIS} = \text{argmin}(\text{Perf}(\text{FIS}_k), \ k = 1, \ldots, K)$ such as $B_k \leq \text{Card}(E)/10$, where B_k is the number of blank examples for the rule base in FIS_k.

Complexity Analysis: The refinement algorithm (Algorithm 2, given in Section V-B) complexity mainly depends on the number of input variables, p. The number of iterations can

be chosen according to p, for instance $iter_{\max} = k * p$ with $k = 3$, and each iteration calls the FIS generation algorithm (Algorithm 3) p times. To generate a fuzzy inference system, the weight of each rule is computed for each of the N data items. The number of rules corresponds, at each step, to all premise combinations. It reaches in the worst case k^p.

VI. Application to Benchmark Data

This section illustrates the potential of the Hierarchical Fuzzy Partitioning method by applying it to some well known benchmark data sets, the Wisconsin breast cancer data and the wine classification data, from the machine learning repository.[2]

These classification problems have been recently revisited by [14], who give an interesting summary that we will use as a basis for our analysis of the results.

A. Data Processing

The same protocol has been applied to each data set. First, sampling was done by extracting ten training samples–representative of the class distribution—from the whole set. The extraction consists of a random selection of 50% of the items of each class. The complement of each training set becomes the test set.

Then, the HFP partitions were induced from each training set and used by the selection algorithm, introduced in Section V, to generate fuzzy inference systems. We used the numerical distance, and a 0.01 tolerance threshold to build the initial partition. The regular hierarchy was built by splitting the whole data set range in equally spaced fuzzy sets, and, as with HFP, the selection algorithm was run with each training set.

Amongst the various configurations proposed by the selection algorithm, we kept the configuration which satisfied the constraints of accuracy and number of blank examples detailed in Section V-D. The configurations are characterized by their number of rules (#R) and their number of variables (#V). Their performances are assessed both on the training and the test sets, and given as a percentage. All results (configuration characteristics and performance) are given as an average on the ten samples.

B. Wisconsin Breast Cancer Diagnostic Data

The Wisconsin Breast Cancer Diagnostic data set contains 699 patterns distributed into two output classes, benign and malignant. Each pattern consists of nine input features: clump thickness, uniformity of cell size, uniformity of cell shape, marginal adhesion, single epithelial cell size, bare nuclei, bland chromatin, normal nucleoli, and mitoses. After removing examples containing missing values, 683 items remain available: 444 are in the benign class and the other 239 are in the malignant one.

Previous results found in the literature, from [14] and [20] are recalled in Table I.

TABLE I
Some Previous Results on the Wisconsin Breast Cancer Data

Models	Performance %
MSC [21]	94.9
NEFCLASS [22]	92.7
NNFS [23]	93.9
FEBFC [24]	94.7
SANFIS [14]	96.3
C4.5 [20]	94.7

As shown in Table II the proposed method leads to good performance results for the Wisconsin breast cancer data set. The 1.6% test error value appearing for HFP in Table II is particularly low. This is partly due to a relatively high number of blank examples in the test sample (20%). This comes from the fuzzy inference system selection procedure, which is focused on generalization, and dismisses too specific items in the data set. The average number of three input variables in the rule premise is smaller than in other results found in the literature, which makes the interpretation easier. The variables of most interest are variable 2 and 6 when using regular grids, while the most frequent combinations using HFP include variables 1 (clump thickness), 3 (uniformity of cell shape), and 6 (bare nuclei).

Fig. 7 shows one of the fuzzy partitions obtained using HFP for the clump thickness feature. It is clear that each fuzzy set can be assigned a readable linguistic label. The corresponding rule base system is defined by five input variables, given in Table III, together with their number of fuzzy sets (#MF) and tentative linguistic labels.

One of the rules is given as follows as an illustration of their intuitive interpretation:

If	Clump thickness	is	Large
And	Uniformity of cell size	is	High
And	Uniformity of cell shape	is	High
And	Bare nuclei	is	High
And	Normal nucleoli	is	High
Then	Class	is	Malignant

C. Wine Classification Data

The wine classification data set contains 178 wines that are grown in the same region of Italy but derived from three different cultivars. The numbers of instances in each class are: 59, 71, and 48. Each pattern consists of 13 continuous features resulting from chemical analysis: Alcohol, malic acid, ash, alkalinity of ash, magnesium, total phenols, flavonoids, nonflavonoid phenols, proanthocyanins, color intensity, hue, OD280/OD315 of diluted wines and proline. Some known results gained using this data, from [14], [25] and [20] are summarized in Table IV.

Table V shows the average results obtained by the proposed method. There is an important difference between the systems built using the HFP family partitions and the regular grid hierarchy. Contrary to what happens for the breast cancer data, the number of blank examples remains very small. Four variables are selected when the partitions are HFP type, the most frequent ones being variables 1, 6, 7, and 13. In the case of regular grids, only two variables (1 and 12) are selected.

One focusing on the numerical performance index may conclude that these results are not very good (10% represent nine examples). However, let us make some comparisons with the systems described in Table IV. The first two consist of 60 rules, instead of seven with ours. As fuzzy systems are

[2]http://www.ics.uci.edu/~mlearn/MLRepository.html

TABLE II
RESULTS ON THE WISCONSIN BREAST CANCER DATA

Type	#V	#R	Training Err %	Test Err %
Regular	2.9	8.6	4.1	5.2
Hfp	3.0	7.8	1.4	1.6

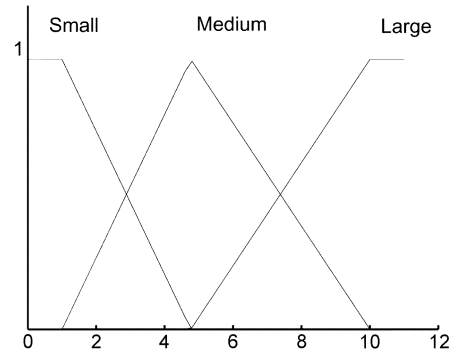


Fig. 7. Fuzzy partition for the clump thickness feature.

TABLE III
EXAMPLE OF SELECTED VARIABLES FOR THE BREAST CANCER DATA

Name	#MF	Labels
Clump thickness	3	Small Medium Large
Uniform. of cell size	3	Low Average High
Uniform. of cell shape	2	Low High
Bare nuclei	3	Low Average High
Normal nucleoli	3	Low Average High

TABLE IV
SOME PREVIOUS RESULTS ON THE WINE DATA

Models	#R	Error
GA-1 [26]	60	$0 \sim 3$
GA-2 [27]	60	$1 \sim 4$
GA + FCM [28]	3	3
SANFIS [14]	3	1
SLAVE [25]	5.2	3.24 %
C4.5 [20]		5.6 %

universal approximators [29], it is always possible to increase the performance by adding new rules. The drawback lies in the relevantness of these rules: how many examples do they concern?

The model built by [28], given on the third row of the table, has only three rules. However, due the clustering induction method, each fuzzy set is specific to a rule, making rule comparison impossible.

SANFIS provides three linguistic rules. Let us first notice the lack of test sets, the training being done on the whole data set. Moreover, all input features appear in each rule. This makes rule comparison and influential variable identification difficult. The lack of constraints in the partition optimization process leads to a loss of semantic. Fig. 8 displays the linguistic labels found by SANFIS for the 9th wine feature. We can see that the fuzzy set named *large* is not really distinguishable from the one labeled *medium*. This throws a shadow on the use of such fuzzy systems in interpretability concerned applications.

SLAVE has been applied on ten random samples (70% for the learning set and 30% for the test one). Two remarks have to be made. First, the way of designing variable partitioning is not

TABLE V
RESULTS ON THE WINE DATA

Type	#V	#R	Training Err %	Test Err %
Regular	1.7	3.1	24.2	26.6
Hfp	3.9	6.8	7.0	10.8

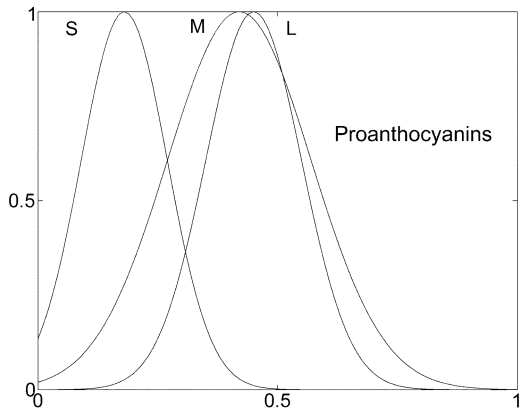


Fig. 8. SANFIS fuzzy partition for the 9th input wine feature.

detailed, but examples show that linguistic interpretability is not guaranteed. Second, several clauses of a given variable can be part of the same premise using a OR connector. This makes the rule number comparison with only AND connected premise rule impossible.

In [20], various ways of splitting numerical attributes are applied to decision tree induction using the famous C4.5 algorithm. The work includes a ten-fold cross validation by a nonpruned tree. As no depth level is given it is difficult to compare the tree structure with a number of rules.

The wine data benchmark shows that the proposed HFP method can achieve a good compromise between accuracy and interpretability, and that it is suitable for cases where knowledge induction is at least as important as numerical performance.

VII. CASE STUDY: A CORN CLASSIFICATION PROBLEM

We now apply a similar procedure to multidimensional agricultural data. The sample is made of 352 items, 80 corn, and 272 weeds. Spectrum data have been collected in order to discriminate corn crop from weeds. Eight spectrum wavelengths, corresponding to peaks, valleys, and other singular points have been preselected by the user. Consequently, the system to be modeled is made up of eight input variables, the output being a class label: one for corn, two for weed.

Our study focuses on the analysis of the results obtained by our HFP method. It includes a sensitivity analysis to initial parameters, and it also gives some complexity analysis elements. As no references are available in the literature, we also present some results that we obtained using other well known techniques, such as discriminant analysis or fuzzy clustering.

We worked either with the whole dataset, or by taking ten random samples from this dataset, as explained in the previous section.

A. HFP Partitions

As all data are numerical, we use the numerical distance in the HFP procedure.

TABLE VI
HFP UNIQUE VALUE SENSITIVITY TO THE TOLERANCE THRESHOLD

Tolerance	2 centers		3 centers		
Variable 1					
0.01	0.042	0.225	0.042	0.181	0.342
0.005	0.034	0.220	0.034	0.170	0.360
0.001	0.042	0.226	0.042	0.181	0.367
Variable 5					
0.01	0.171	0.320	0.103	0.210	0.320
0.005	0.095	0.260	0.095	0.226	0.317
0.001	0.095	0.260	0.095	0.227	0.320
Variable 7					
0.01	0.397	0.747	0.397	0.692	0.895
0.005	0.405	0.753	0.405	0.703	0.900
0.001	0.422	0.747	0.422	0.689	0.907

A study has been done to examine the sensitivity of the method to the tolerance threshold which determines the number of unique values used in the HFP procedure, as explained in Section II-A. Results on the whole dataset are summarized in Table VI, for a few variables (1, 5, 7).

The most important variations due to the tolerance threshold appear for Variable 5. If we observe the corresponding histogram, plotted in Fig. 9, we can see that it looks like a door function, with no clear structure. That could explain the variability in that case.

To save computational time, the HFP procedure has thus been applied using a tolerance threshold $\mathrm{tol} = 0.01$. Table VII gives the fuzzy set centers for the fuzzy partitions of size 2 and 3, and variables 1, 5, and 7. The first line is relative to the whole dataset, and the other two to the samples. The figures in the second line represent the average for the ten samples, and the standard deviation is given in brackets in the third line.

Fig. 10 displays the evolution of σ^{FP}. For the first and fifth variables, the minimum, obtained for a two fuzzy set partition, is well marked. This is not the case for the seventh variable. There are two close minima, corresponding to two and ten fuzzy sets. Their absolute values are much higher than for the other two variables. The indicator does not seem to be significant in this case.

Comparison of the fuzzy set centers with those of a regular grid, recalled in Table VIII shows that they are very different.

B. FIS Generation and Selection Using HFP

The FIS generation and selection are either done on the whole data set, or on each of the training samples. In the first case, the performance is measured on the whole data set, in the second one it is evaluated on the test sample. Table IX gives the test sample size (Test), the percentage of misclassified items (MIS), the number of variables in the FIS premises (#V), the number of rules (#R), and in the last column, the number of times that each variable has been selected.

From Table IX, we can see that the most often selected inputs are Variables 1, 5, and 7. This leads us to believe that these variables are of particular importance. The average number of 9.9 rules would drop to 6.6 if one configuration which includes 32 rules were replaced by a simpler one, with six rules only, which has a slightly lessened performance (three misclassified items instead of two).

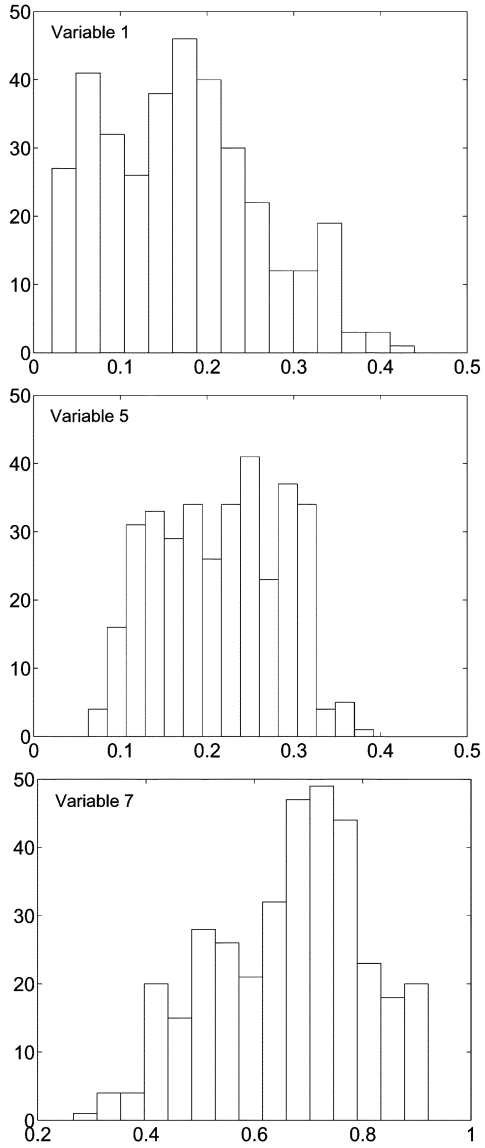


Fig. 9. Input variable histograms.

TABLE VII
MF CENTERS FOUND BY HFP

L. Set	2 centers		3 centers		
Variable 1					
Whole	0.042	0.225	0.042	0.181	0.342
2/3	0.054	0.236	0.039	0.173	0.346
	(0.030)	(0.024)	(0.005)	(0.011)	(0.048)
Variable 5					
Whole	0.171	0.320	0.103	0.210	0.320
2/3	0.108	0.264	0.102	0.222	0.318
	(0.027)	(0.018)	(0.016)	(0.007)	(0.005)
Variable 7					
Whole	0.397	0.747	0.397	0.692	0.895
2/3	0.512	0.818	0.384	0.682	0.872
	(0.092)	(0.064)	(0.062)	(0.021)	(0.036)

C. Comparison With Other Approaches

Using Regular Grid Hierarchies: The FIS generation and selection are now done using hierarchies based on regular grids, and the results are shown on Table X.

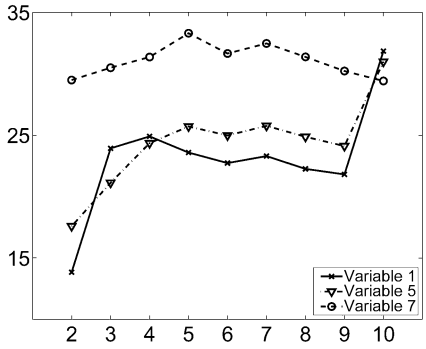


Fig. 10. σ^{FP} evolution for three input variables.

TABLE VIII
MF CENTERS WITHIN REGULAR GRIDS

Variable	2 centers		3 centers		
Variable 1	0.021	0.439	0.021	0.230	0.439
Variable 5	0.063	0.392	0.063	0.228	0.392
Variable 7	0.266	0 .921	0.266	0.594	0.921

TABLE IX
HFP BASED FIS CHARACTERISTICS

L. Set	Test	MIS	#V	#R	1	2	3	4	5	6	7	8
Whole	352	2.6	5	12	1		1	1	1		1	
2/3	115	2.3	3.1	9.9	6	3	2	3	8	1	6	2

TABLE X
REGULAR GRID BASED FIS CHARACTERISTICS

L. Set	Test	MIS	#V	#R	1	2	3	4	5	6	7	8
Whole	352	4.8	1	2	1							
2/3	115	3.0	1.7	3.2	9	2		3	2	1		

Most FIS based on regular grids, generated using the criterion given in Section V-D, include a single variable: This is the case for the whole data set, and 5 times out of ten for the random samples. Intermediate results (not given here) show that, in the first steps of the refinement procedure, when a variable is added into the rules, the performance is not improved. The performance gets better when the partition sizes are higher, but the number of blank examples rapidly goes over the limit of ten percent. This is due to the fact that the fuzzy set centers are not designed according to data distribution.

Compared to HFP, the regular grid performance is lower.

Subtractive Fuzzy Clustering: Subtractive clustering is a fuzzy clustering method introduced by [30]. It divides a multidimensional data set into an *a priori* unknown number of clusters. It estimates the cluster centers by setting a range of influence in each of the data dimensions, and choosing as cluster centers the points with the strongest attracting potential. We used the *matlab* implementation, which includes the generation of an order 1 Sugeno-type FIS with as many rules as clusters, the rule conclusions being optimized using a least squares method. The average misclassified number is equal to 1.6%. The 8 variables appear in the rule premises, the average number of rules being equal to 6.2.

A FIS generated with such a fuzzy clustering method is characterized by each rule using its own fuzzy sets, all different from one rule to the next.

The fuzzy sets not being shared by the rules makes any comparison between the rules impossible, and therefore prevents the identification of the influent variables.

Discriminant Analysis: Linear discriminant analysis is a well-known multivariate statistics technique, devised to distinguish between groups. It uses linear functions of the input variables, to define a new subspace, based on the maximization of the ratio of the between-group sum of squares to the within-group sum of squares. An observation can then be classified by computing its Euclidean distance from the group centroids, projected onto the new subspace. The observation is assigned to the closest group.

The projecting matrix and the group centroids have been calculated using each of the training sets, and the performance evaluated over the corresponding test set. The average misclassified number comes to 2.7%. The eight variables are included in the definition of the new subspace.

The results of discriminant analysis or subtractive clustering show that this spectrum data problem is not an easy one. Subtractive clustering obtains the best performance. However, the price to be paid is a greater number of parameters, a more sophisticated optimization procedure for the rule consequent parts, and a totally opaque model, disadvantage which also applies to discriminant analysis.

Generally speaking, we can consider that refinement based on fuzzy partition hierarchies leads to a good compromise between performance and interpretability of a fuzzy model. Moreover, when fuzzy set parameters are determined according to the data distribution, the results are better than with regular grids.

VIII. CONCLUSION

The hierarchical fuzzy partitioning presented in this paper aims to generate a family of fuzzy partitions from data. The originality is double. First the product is not one partition, but a hierarchy including partitions with various resolution levels. In each dimension, the initial partition is made up of fuzzy sets centered about the input values, if there are a few of them only. If the input values are too numerous, they are first clustered into so-called unique values.

Instead of a descending procedure, such as partition refinement [31]–[34], an ascending technique has been applied. It consists of merging two adjacent fuzzy sets at each step, the ones which best satisfy a merging criterion. The criterion preserves the previous step structure by considering a special sum of distances over the training data set. These distances are conceived to reflect the fuzzy partitioning under design. This concept is the second strong point of this paper. To enforce it, we introduced the notions of internal and external distances relative to fuzzy sets. The internal distance concerns the part of membership within a single fuzzy set, and the external distance the part of membership related to two distinct fuzzy sets.

The generated partitions can be used to build up the premises of a fuzzy inference system, or as an input for other rule induction techniques, such as fuzzy decision trees. We introduced in this paper a simple fuzzy inference system generation and selection procedure designed to alleviate the curse of dimensionality. To highlight its potential, we first applied the method to well-known benchmark data, for which reference results are available, then

to agricultural corn data. The comparison with other techniques, fuzzy clustering, or discriminant analysis, shows encouraging results.

The comparison includes the numerical performance, but is not restricted to it. Other important aspects, dealing with rule base interpretability, most influential variable identification or semantic integrity of the fuzzy partitions, are taken into account. The goal of the learning process is not only the numerical index improvement, but knowledge induction. In numerous cases, such as decision support system design, diagnosis applications, one may accept a controlled loss of performance to gain a better understanding.

The proposed approach does not try to compete with function approximation techniques, but is a promising way for managing the tradeoff between performance and interpretability in multidimensional complex problem modeling.

Further work should consider more sophisticated selection procedures, to take into account the model complexity: Number of rules, number of variables, and partition refinement degree.

ACKNOWLEDGMENT

The authors would like to thank B. Panneton, from Agriculture and Agri-Food Canada, for providing data and helpful information.

REFERENCES

[1] E. H. Mamdani and S. Assilian, "An experiment in linguistic synthesis with a fuzzy logic controller," *Int. J. Man-Mach. Stud.*, vol. 7, pp. 1–13, 1975.

[2] T. Takagi and M. Sugeno, "Fuzzy identification of systems and its applications to modeling and control," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-15, pp. 116–132, Jan. 1985.

[3] S. Guillaume, "Designing fuzzy inference systems from data: An interpretability-oriented review," *IEEE Trans. Fuzzy Syst.*, vol. 9, pp. 426–443, June 2001.

[4] J. C. Bezdek, *Pattern Recognition With Fuzzy Objective Functions Algorithms*. New York: Plenum Press, 1981.

[5] R. E. Hammah and J. H. Curran, "On distance measures for the fuzzy k-means algorithm for joint data," *Rock Mech. Rock Eng.*, vol. 32, no. 1, pp. 1–27, 1999.

[6] D. E. Gustafson and W. C. Kessel, "Fuzzy clustering with a fuzzy covariance matrix," in *Proc. IEEE Conf. Decision Control*, San Diego, CA, 1979, pp. 761–766.

[7] D. Dubois and H. Prade, *Fuzzy Sets and Systems: Theory and Applications*. New York: Academic, 1980.

[8] J. Fan and W. Xie, "Distance measure and induced fuzzy entropy," *Fuzzy Sets Syst.*, vol. 104, pp. 305–314, 1999.

[9] B. B. Chaudhuri and A. Rosenfeld, "On a metric distance between fuzzy sets," *Pattern Recogn. Lett.*, vol. 17, pp. 1157–1160, 1996.

[10] ——, "A modified hausdorff distance between fuzzy sets," *Inform. Sci.*, vol. 118, pp. 159–171, 1999.

[11] R. Lowen and W. Peeters, "Distance between fuzzy sets representing grey level images," *Fuzzy Sets Syst.*, vol. 99, pp. 135–149, 1998.

[12] L. Koczy and K. Hirota, "Ordering, distance and closeness of fuzzy sets," *Fuzzy Sets Syst.*, vol. 59, pp. 281–293, 1993.

[13] P. Subasic and K. Hirota, "Similarity rules and gradual rules for analogical interpolative reasoning with imprecise data," *Fuzzy Sets Syst.*, vol. 96, pp. 53–75, 1998.

[14] J.-S. Wang and C. S. G. Lee, "Self-adaptative neuro-fuzzy inference systems for classification applications," *IEEE Trans. Fuzzy Syst.*, vol. 10, pp. 790–802, Dec. 2002.

[15] J. Valente de Oliveira, "Semantic constraints for membership functions optimization," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-29, pp. 128–138, Jan. 1999.

[16] P.-Y. Glorennec, *Algorithmes d'apprentissage pour systèmes d'inférence floue*. Paris, France: Editions Hermès, 1999.

[17] J. Espinosa and J. Vandewalle, "Constructing fuzzy models with linguistic integrity from numerical data-afreli algorithm," *IEEE Trans. Fuzzy Syst.*, vol. 8, pp. 591–600, Oct. 2000.

[18] R. A. Fisher, "The use of multiple measurements in taxonomic problems," *Ann. Eugenics*, vol. 7, pp. 179–188, 1936.

[19] G. A. Miller, "The magic number seven, plus or minus two: some limits on our capacity for possessing information," *Psychol. Rev.*, vol. 63, pp. 81–97, 1956.

[20] T. Elomaa and J. Rousu, "General and efficient multisplitting of numerical attributes," *Mach. Learn.*, vol. 36, pp. 201–244, 1999.

[21] B. C. Lovel and A. P. Bradley, "The multiscale classifier," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 18, pp. 124–137, Feb. 1996.

[22] D. Nauck and R. Kruse, "A neuro-fuzzy method to learn fuzzy classification rules from data," *Fuzzy Sets Syst.*, vol. 89, no. 3, pp. 277–288, 1997.

[23] R. Setiono and H. Liu, "Neural-network feature selector," *IEEE Trans. Neural Networks*, vol. 88, pp. 654–662, June 1997.

[24] H. M. Lee, C. M. Chen, J. M. Chen, and Y. L. Jou, "An efficient fuzzy classifier with feature selection based on fuzzy entropy," *IEEE Trans. Syst., Man, Cybern.*, vol. 31, pp. 426–432, June 2001.

[25] L. Castillo, A. González, and R. Pérez, "Including a simplicity criterion in the selection of the best rule in a genetic fuzzy learning algorithm," *Fuzzy Sets Syst.*, vol. 120, pp. 308–321, 2001.

[26] A. L. Corcoran and S. Sen, "Using real-valued genetic algorithms to evolve rule sets for classification," in *Proc. 1st IEEE Conf. Evolutionary Computation*, Orlando, FL, June 1994, pp. 120–124.

[27] H. Ishibuchi, T. Nakashima, and T. Murata, "Performance evaluation of fuzzy classifier systems for multidimensional pattern classification problems," *IEEE Trans. Syst., Man, Cybern.*, vol. 29, pp. 601–618, Oct. 1999.

[28] M. Setnes and H. Roubos, "Ga-fuzzy modeling and classification: complexity and performance," *IEEE Trans.Fuzzy Syst.*, vol. 8, pp. 509–522, Oct. 2000.

[29] L.-X. Wang, "Fuzzy systems are universal approximators," in *Proc. 1st IEEE Conf. Fuzzy Systems*, San Diego, CA, 1992, pp. 1163–1169.

[30] S. L. Chiu, "Fuzzy model identification based on cluster estimation," *J. Intell. Fuzzy Syst.*, vol. 2, pp. 267–278, 1994.

[31] F. Klawonn and A. Keller, "Fuzzy clustering and fuzzy rules," presented at the *7th IFSA World Congr.*, Prague, Czech Republic, 1997.

[32] Y. Lin, G. A. Cunningham, and S. V. Coggeshall, "Using fuzzy partitions to create fuzzy systems from input-output data and set the initial weights in a fuzzy neural network," *IEEE Trans. Fuzzy Syst.*, vol. 5, pp. 614–621, Nov. 1997.

[33] P. Bortolet, "Modélisation et commande multivariable floues: Application à la commande d'un moteur thermique," Ph.D. dissertation, LAAS-CNRS, Institut National des Sciences Appliquées, Toulouse, France, Dec. 1998.

[34] I. Rojas, H. Pomares, J. Ortega, and A. Prieto, "Self-organized fuzzy system generation from training examples," *IEEE Trans. Fuzzy Syst.*, vol. 8, pp. 23–36, Feb. 2000.

Serge Guillaume received the Ph.D. degree from the University of Toulouse, Toulouse, France, in 2001.

He is an Engineer with the French Agricultural and Environmental Engineering Research Institute (Cemagref), Montpellier, France. He worked for several years in the field of image analysis and data processing applied to the food industry. From September 2002 to August 2003, he was hosted as a Visitor by Escuela Técnica Superior de Ingenieros de Telecomunicación, Madrid, Spain. He is involved in theoretical as well as applied developments, which are available in FisPro, an open source portable software for fuzzy inference system design and optimization.

Brigitte Charnomordic received the Ph.D. degree in physics from the University of Lyon, Lyon, France, in 1976.

She later became interested in computer science, and joined the Institut National de la Recherche Agronomique (INRA), Montpellier, France. She is currently a Research Engineer and works on projects regarding knowledge discovery and expert knowledge cooperation in the food industry and environmental areas. Her present interests include fuzzy logic, software engineering, and hybrid intelligent systems for process supervision, as well as open source software.