The Prototyping and Focused Discriminating Strategy for Pattern Recognition and one Instantiation: the MELIDIS System

Nicolas Ragot, Eric Anquetil

The Prototyping and Focused Discriminating Strategy for Pattern Recognition and one Instantiation: the MÉLIDIS System

Nicolas Ragot and Éric Anquetil

Abstract

This paper presents the Prototyping and Focused Discriminating (PFD) strategy for pattern recognition. This strategy takes benefits from the duality between model generation and discrimination. Both collaborate through a focusing mechanism that detects the conflicts between the class models and drive the discrimination. Classifiers based on this collaboration benefit from a set of useful properties. The Mélidis system illustrates this strategy and extends its possibilities, using a fuzzy framework. As shown by experiments, the resulting system provides an interesting compromise between accuracy and compactness. Experiments also demonstrate the interest of the new strategy and of its focusing mechanism.

Index Terms

Pattern recognition, Prototype modeling, Discriminative modeling, Collaboration between classifiers, Fuzzy clustering, Fuzzy decision trees, Fuzzy inference systems.

N. Ragot is with the Laboratoire d'Informatique (LI), Université de Tours, 64 Av. Jean Portalis, 37200 Tours, France (e-mail: nicolas.ragot@univ-touts.fr)

E. Anquetil is with the Institut de Recherche en Informatique et Systèmes Aléatoires (IRISA), INSA de Rennes, Av. du Gal. Leclerc, 35042 Rennes cedex, France (e-mail: Eric.Anquetil@irisa.fr)

The Prototyping and Focused Discriminating Strategy for Pattern Recognition and one Instantiation: the MÉLIDIS System

I. Introduction

FOR complex pattern recognition problems, multiple classifiers systems (MCS) have shown their superiority over single classifiers. For the design of such MCS, there are at least four critical points to consider: which classifier to use; which feature space to choose for each classifier; which data to use for the learning of each classifier (should they be specialized and how); and finally, which architecture should be used to combine the classifiers? These choices will determine the properties of the system. Nevertheless, they are rarely considered all together. Particularly, the choice of the classifiers suffer from a lack of justification. Indeed, for most of MCS, the recognition rates seems to be the unique objective occluding other general properties that could be required for real life applications such as the ability of the recognizer: to detect its weaknesses, i.e. outliers or unrecognizable data and possible ambiguities in the decision; to deal with complex problems with possibly a high number of classes; to deal with multimodal classes; to be modular and flexible to simplify its optimization and its maintenance; etc. In other words, MCS tends to be reduced to a combination of classifiers instead of being considered as a collaboration between classifiers. In this paper, we try to adopt this alternative point of view and we detail a collaboration strategy that provide accuracy and such kind of general properties.

Our first concern is about the choice of the classifiers and other choices will be deduced. Of course, it seems difficult to know precisely which classifier to use at a given place in a MCS. Nevertheless we can think about using systems that could collaborate to provide complementary properties. This way, there are at least two kinds of approaches for pattern recognition that could collaborate. The first one consists in the explicit description of each class through the use of patterns, prototypes or models. These class models are used afterward in a competition process to recognize unknown shapes. This is usually called *prototype modeling*, *generative modeling* or *model-based approach* [1]–[4]. The second kind of approaches consists in modeling explicit boundaries between the different classes by a discrimination process. This is called *discriminative modeling* [5], [6]. Both kind of approaches have been used independently in MCS to deal with complex problems, on the basis of mixtures of experts [7]–[10]. Authors have also compared both of them to illustrate their complementary strengths [11]. The link between generative and discriminative learning have also been studied and it was shown that learning discriminatively a generative model could improve the accuracy of the recognizer [12], [13]. Specific classifiers can also implicitly combine both approaches such as Radial Basis Function network (RBF) presented in [14]. Nevertheless, the explicit collaboration between model-based and discriminant modeling has only been considered recently, in the framework of MCS [15]–[20]. All these systems are based on a two-stage hierarchical architecture in which local classifiers (that can be seen as parallel or mixture of experts) improve the accuracy of a global classifier. We can mention two different strategies. In [16]–[18], the global classifier is a discriminant system (mainly MLP or RBF) whereas the local classifiers are k-NNs. This choice allows to limit the complexity of the entire system: the global classifier, simpler, recognize most of the samples. It is only when a decision is ambiguous that local classifiers, more complex, are used. The main drawback is that the global classifier must deal itself with the entire problem, without reducing first its intrinsic complexity. Moreover, these approaches do not exploit all the advantages of model-based systems. In contrast, in [15], [19], [20], the global classifier is a model-based approach and the local classifiers are discriminant-based approaches. The reason is that model-based classifiers are explicit, flexible (prototypes can be studied separately, modified, removed or added) and they can handle problems with numerous classes since the models can be trained on each class separately. Consequently, this kind

of approaches is much more interesting than the previous one. Moreover, they can be refined to offer other useful properties by choosing an appropriate collaboration that take into account the two other critical points: feature space and data selection. This is the aim of the *Prototyping and Focused Discriminating* (PFD) strategy that makes collaborate prototype class models and discriminant models through a focusing mechanism. This one detects conflicts between the class models and drive the discrimination. The Mélidis recognition system instantiate this strategy to illustrate its interestingness and extends its possibilities using a fuzzy framework.

This paper has the following organization. First, section II explains the PFD strategy. Next, the Mélidis system is detailed. Section III describes the learning mechanism. The corresponding modeling formalized by fuzzy inference systems is presented in section IV. Section V explains the decision process by describing how the different kinds of knowledge are aggregated and fused for the classification task. Finally, the section VI gives experimental results that demonstrate the interest of the strategy, both on classical benchmarks, and on the more specific problems of on-line handwritten digit recognition.

II. Prototyping and Focused Discriminating (PFD) strategy

The PFD strategy exploits the four critical points cited in the introduction to provide general useful properties. It is based on the hybrid architecture illustrated in Fig. 1 which is generic and flexible enough to deal with most of the supervised classification problems, in the same way as MLP, RBF, SVM and MCS in general. In comparison with other two-stage MCS as those detailed in [15], [19], [20], its specificity

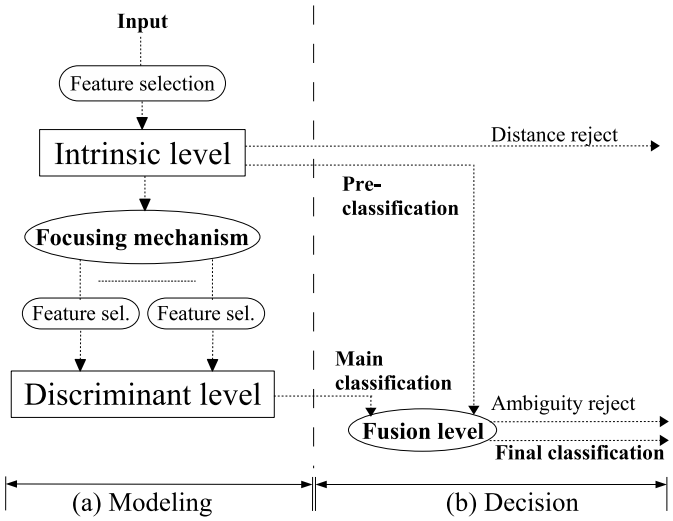

Fig. 1. The hybrid architecture of the PFD strategy.

comes mainly from two points. Firstly, the nature of the knowledge used to model each level and the way they collaborate through the focusing mechanism are original, as explained below. Secondly, the modeling part of the system (cf. Fig. 1 (a) and section II-A) is independent from the decision mechanism (cf. Fig. 1 (b) and section II-B), which makes the architecture more flexible.

A. Modeling part of the strategy

The *intrinsic level* describes the classes by their intrinsic or typical properties i.e. by a set of prototypes. These prototypes must be defined independently from one another, so that they can overlap if necessary (cf. Fig. 2). Therefore, they correspond to the most typical and representative samples of a given class. This kind of modeling has the same advantages as general model-based approaches. It is flexible: each model can be learnt and optimize separately; prototypes can be added or removed. Thus it allows to address the complexity of the problem (division according to the classes) and of the algorithms used

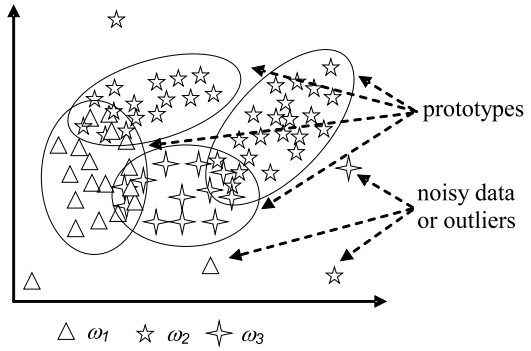

Fig. 2. Example of intrinsic modeling for 3 classes (ω_1, ω_1, ω_1) by prototypes (represented by ellipsis) in a two dimensional feature space.

(parallelism allowed). Moreover, prototypes allow to deal with multimodal classes (e.g. class ω_2 in Fig. 2) which is not true for all model-based approaches. One should notice here that these prototypes should be extracted from a specific feature space that could be chosen either to make intrinsic properties more robust (by minimizing intra-class distance for example) or to make the corresponding *pre-classification* (cf. section II-B) more accurate.

As most of other two-stage systems, the second level is composed of local discriminant classifiers that solve the conflicts resulting from the previous level. Since the discrimination process has a highly contextual behavior, the corresponding knowledge must be extracted in precise contexts to be more accurate. In classical MCS [21]–[23], discriminant classifiers operate the discrimination of one class against all the others or discriminate pairs of classes. In fact, the entities to discriminate are not necessarily the classes themselves. It is rather subclasses or more generally subsets of samples that we call here *discrimination contexts*. The *focusing mechanism* determines these discrimination contexts by detecting the overlaps between classes, using a distance criterion from the prototypes. One discrimination context is extracted for each class and each one contains only samples that have intrinsic properties similar to those of this class (cf. Fig. 3). This process is much stable and robust than the one used in other two-stage systems which are based on the result of a first classifier and then depends on arbitrary decision boundaries. Of course, it might be possible to refine the process and to extract one discrimination context for each prototype but this solution is not considered here. Finally for each discrimination context, the feature space used should be automatically adapted to make the discrimination more accurate. This can be done by selecting only interesting features from the original feature space.

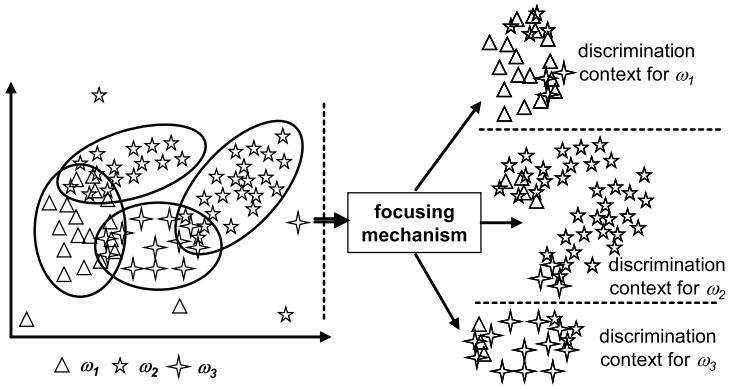

Fig. 3. Example of the focusing mechanism based on the intrinsic modeling: one discrimination context is extracted for each class.

B. Decision mechanism of the system

For the recognition of unknown samples, the decision function of the PFD strategy fully exploits the previous modeling. As mentioned above, this decision mechanism does not intervene in the modeling process, especially for the elaboration of the discriminant level (cf. Fig. 1 (b)). Therefore, the same modeling can be used with several kinds of decision functions which is useful for the optimization of the system. The decision mechanism is composed of three subfunctions. The first one is provided by the intrinsic level. This *pre-classification* is based on the computation of scores giving the adequacy of an unknown sample to each class. Similarly, the discriminant level produces a second decision, named the *main classification*. This one is not only the result of the discriminant modeling but rather the result of the exploitation of the intrinsic **and** the discriminant knowledge **through** the focusing mechanism as shown in the section V. Finally, the third subfunction is represented by the *fusion level*. It tries to take advantage of the dual modeling by combining both pre-classification and main classification. It yields to the *final classification*.

The decision scheme can also be further refined by adding two rejection processes [24]. The first one is based on the intrinsic modeling (distance to prototypes) and reject outliers that the system cannot recognize. The second one is based on the discriminant modeling and ambiguities of the final classification.

III. The Mélidis system: learning and knowledge extraction

To illustrate the PFD strategy, we designed the Mélidis system that uses fuzzy logic as a backbone. This was chosen for three reasons. Firstly, the fuzzy algorithms used here particularly highlights the PFD strategy. Secondly, this provides an homogeneous formalism for the entire system. Finally, it illustrates how properties inherited from the PFD strategy could be extended. Indeed, fuzzy logic allow to deal with imprecisions and variability of the inputs. It also provides a compact and legible modeling. Consequently, the Mélidis system provides an interesting compromise between the following properties: accuracy, compactness, ability to deal with noisy data, flexibility and legibility. Of course, other choices could enhance or add other properties.

The learning is based on a database B_{app} composed of n samples $e_j,\ j = 1, \ldots, n$. Each sample is described by a vector of N numerical features. In the following, e_j will designate both a sample and the vector of its features. Thus, $e_j^m,\ m = 1, \ldots, N$ represents the m^{th} feature of e_j. The learning is supervised and the samples of B_{app} are labeled by their class $\omega_i,\ i = 1, \ldots, K$. This label is noted λ_j. Consequently, the learning is based on the couples (e_j, λ_j). Fig. 4 illustrates the entire process, driven by the focusing mechanism.

A. Intrinsic level: description of the classes by fuzzy prototypes

To model each class $\omega_i,\ i = 1, \ldots, K$ according to its intrinsic characteristics, the corresponding models $MI(\omega_i)$ are extracted from specific data set B_{app}^i. Each one is created by using all the samples from B_{app} that belong to ω_i (cf. Fig. 4, step 1): $B_{app}^i = \{e_j \in B_{app} \,|\, \lambda_j = \omega_i\}$. Then, an appropriate algorithm must be run on each B_{app}^i (cf. Fig. 4, step 2). Learning Vector Quantization [25], Expectation-Maximization [26] or data condensation [27] are all classical techniques that can find relevant prototypes. Nevertheless, these algorithms are generally sensible to noisy data such as outliers. Moreover, there ability to extract independent prototypes that could overlap by sharing data (fuzzy clusters) is limited. These two points are important limitations to represent intrinsic properties of the classes. It is why we choose instead to use the Possibilistic C-Means (PCM) [28]–[30] fuzzy clustering algorithm. Contrary to other partitioning algorithms, the PCM describes the clusters by independent fuzzy prototypes (N-dimensional[1] fuzzy sets) defined by their center and their shape (i.e. membership function). Moreover, this algorithm was designed to be less sensitive to outliers. Thus, it corresponds exactly to the notion of *typicality* needed here. At each step of the iterative process, the centers are updated using the following

[1]A feature selection algorithm could be used before as proposed in section II. This is not described here.

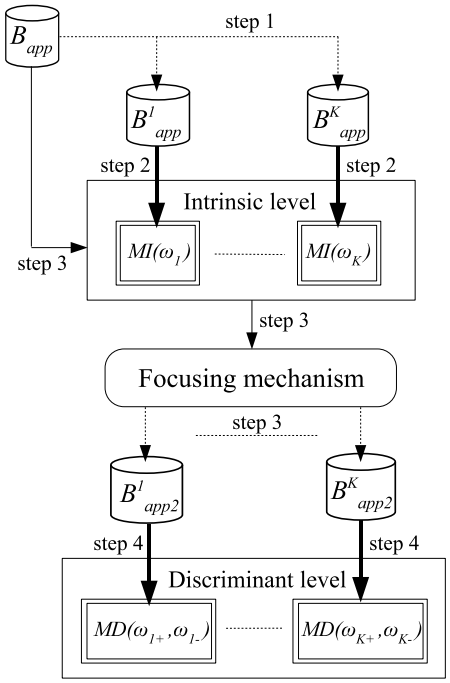

Fig. 4. The modeling process of the Mélidis system.

equation: $P_c = \sum_{j=1}^{n}(\mu_{cj})^m e_j / \sum_{j=1}^{n}(\mu_{cj})^m$, where $P = \{P_1, \ldots, P_C\}$ are the centers of the C clusters to find, $E = \{e_j | j = 1, \ldots, n\}$ is the data set, $m > 1$ is the fuzzifier (here we use $m = 2$) and μ_{cj} is the membership degree of the sample e_j to the cluster c. This degree is computed and updated using: $\mu_{cj} = 1/(1 + (e_j - P_c)^T Cov_c^{-1}(e_j - P_c))$, where Cov_c is the fuzzy covariance matrix of the cluster c (see [28] for more details).

The PCM algorithm is run separately on each class ω_i, by setting $E = B^i_{app}$ and $C = L_i$, where L_i is determined by using the Xie and Beni validity measure [31] for several values of C (generally no more than 4). Even if there exists other methods to determine the number of clusters, such as split and merge techniques [29], one advantage of the PCM is that it is robust enough to find a correct representation of the data for a given value of C. Then, a second run can be done to limit the impact of outliers: the first run is used as initialization and the samples e_j in B^i_{app} for which μ_{cj} is low for all c are eliminated. Finally, the result for a class ω_i, is a set of N-dimensional fuzzy prototypes (or fuzzy sets) $F^l_{\omega_i}$, $l = 1, \ldots, L_i$, defined by their membership function corresponding to the μ_{cj} of the PCM with $c = l$: $\mu_{F^l_{\omega_i}}(e_j) = 1/(1 + (e_j - P^l_{\omega_i})^T (Cov^l_{\omega_i})^{-1}(e_j - P^l_{\omega_i}))$, and where $P^l_{\omega_i}$ are the centers of the clusters/subclasses found and $Cov^l_{\omega_i}$ are the final covariance matrices. The intrinsic model $MI(\omega_i)$ for a class ω_i is the union of these prototypes. The membership degree of a sample e_j to $MI(\omega_i)$ is computed using a t-conorm $\perp$:

$$\mu_{MI(\omega_i)}(e_j) = \perp_{l=1,\ldots,L_i} \mu_{F^l_{\omega_i}}(e_j)\,. \tag{1}$$

B. Focusing mechanism

For a given class ω_i, its discrimination context corresponds to a learning database B^i_{app2} composed of the samples that potentially belong to ω_i. It is obtained by operating a fuzzy filtering, based on the intrinsic model $MI(\omega_i)$ and an α-cut α_i (cf. Fig. 4, step 3): $B^i_{app2} = \{e_j \in B_{app} \,|\, \mu_{MI(\omega_i)}(e_j) \geq \alpha_i\}$. These α parameters can be hard to determine. A solution is to use a relative threshold based on the higher

membership degree of the sample to the K intrinsic models:

$$B^i_{app2} = \{e_j \in B_{app} \,|\, \mu_{MI(\omega_i)}(e_j) \geq \frac{\max_{k=1,\ldots,K}(\mu_{MI(\omega_k)}(e_j))}{\alpha}\}. \quad (2)$$

Thus, a sample will be put only in the discrimination contexts corresponding to the intrinsic models for which its membership degree is significant, relatively to the higher one. In practice, it works well with a value for α between 2 and 3. For the discrimination task, the samples that truly belong to ω_i are considered as positive samples (examples) and relabeled ω_{i+}, whereas the others are considered as negative samples (counter-examples) and relabeled ω_{i-}.

C. Discriminant level: discrimination by specific fuzzy decision trees

For each database B^i_{app2}, a discriminant model $MD(\omega_{i+}, \omega_{i-})$ (cf. Fig. 4, step 4) is built to separate examples from counter-examples. In the Mélidis system, Fuzzy Decision Trees (FDT) [32]–[37] are used. Even if other algorithms (possibly more accurate) could be used, this choice highlights the principle of the focused discrimination and allow an homogeneous representation with the first level (cf. section IV). Indeed, FDT rely on a progressive discrimination which is performed in discrimination contexts represented by subset of samples and adapted feature space, at each node of the tree. This corresponds exactly to the PFD strategy. Moreover, the fuzzy formalism allows more shaded representations and so robust decisions. It also makes the interpretation of the tree's structure easier.

The FDT used here [37] were designed to improve their discrimination power. They have the structure illustrated in Fig. 5. The nodes N_{Id} are identified by a label Id. For the root, $Id = 1$. For the following

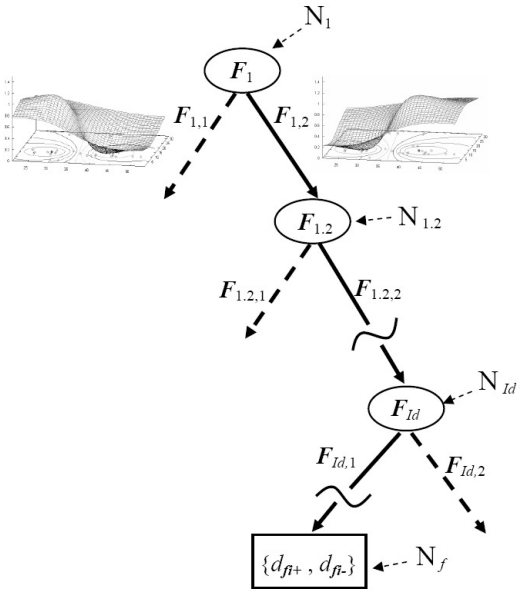

Fig. 5. The structure of the fuzzy decision trees used at the discriminant level.

child nodes, Id is the label of the parent node concatenated by '.' and the index of the child node (considering its parent, in a left-right order). At each node N_{Id}, a binary partitioning is operated on a local data set B_{Id} ($B_1 = B^i_{app2}$). This avoid having too many early splittings that would be irrelevant and unstable. The partitioning is performed using the Fuzzy C-Means (FCM) [38] with $C = 2$ which describes the separation properties of the data set by defining two relative clusters. Indeed, each sample

e_j has a degree μ_{cj} of overlapping (or "sharing") between the 2 clusters:

$$\mu_{cj} = \frac{1}{\sum_{l=1}^{2} \left(\frac{d^2(e_j, P_c)}{d^2(e_j, P_l)} \right)^{\frac{1}{m-1}}} \tag{3}$$

with P_c the centers of the clusters. To make the corresponding boundary meaningful for the discrimination, it is extracted in a local feature subspace F_{Id} determined by a Genetic Algorithm (GA) based on the star entropy criterion [39], [40] from the entire N-dimensional feature space. The result of the partitioning is two fuzzy sets $Fs_{Id.1}$ and $Fs_{Id.2}$ defined by their membership function corresponding to equation 3. Next, the local data set B_{Id} is divided into two new data sets, $B_{Id.1}$ and $B_{Id.2}$, using an α-cut: $B_{Id.t} = \{e_j \in B_{Id} \,|\, \mu_{Fs_{Id.t}}(e_j) \geq \beta\}, \quad t = \{1, 2\}$. Since the fuzzy sets $Fs_{Id.1}$ and $Fs_{Id.2}$ are defined relatively to each other, β is given here a value of less than 0.5 (0.35 is a good value in practice). Thus, the samples near the discriminant boundary are duplicated in each new context. Two new nodes are next added, based on these new discrimination contexts. The process is repeated until a stopping criteria is met. Here, we have chosen to stop when the representativity of a class exceeds 99% and when there are not enough samples in the discrimination context to operate another partitioning. 10 is a good value in general. When a leaf N_f is built, the fuzzy conditional probabilities d_{fi+} and d_{fi-} to obtain the classes ω_{i+} and ω_{i-} are estimated in the following way. Each sample e_j of B^i_{app2} has a membership degree $\mu_{N_{Id}}(e_j)$ to each node N_{Id}. At the root, $\mu_{N_1}(e_j) = 1$ and the membership degree to a child node $N_{Id.t}$ is defined by: $\mu_{N_{Id.t}}(e_j) = \top\left(\mu_{N_{Id}}(e_j), \mu_{Fs_{Id.t}}(e_j)\right)$, where $\top$ is a t-norm representing the conjunction. Finally, the membership degree to a leaf N_f, which expresses how much a sample satisfies all the conditions along the path from the root node N_1 to the leaf N_f, is defined by:

$$\mu_{N_f}(e_j) = \top\left(\mu_{N_1}(e_j), \top_{Id=1}^{Id=f} \mu_{Fs_{Id}}(e_j)\right), \tag{4}$$

where Fs_{Id} (with $Id = 1, \ldots, f$) represents all the fuzzy sets found along the path. The conditional probability d_{fi+} (d_{fi-}) is then computed by: $d_{fi+} = P^*(\omega_{i+}|f) = \frac{\sum_{e_j \in B^i_{app2},\, \lambda_j = \omega_{i+}} \mu_{N_f}(e_j)}{\sum_{e_j \in B^i_{app2}} \mu_{N_f}(e_j)}$.

IV. Formalization by fuzzy inference systems

We describe in this section how Fuzzy Inference Systems (FIS) express the relationships between the previous modeling and the classes. The choices made here are based on previous works [41] that have proven their effectiveness.

A. Formalization of the intrinsic level

The models $MI(\omega_i)$, that were extracted for each class at the intrinsic level, are used to design a unique FIS. There is one rule R_{ω_i} for each model:

$\boldsymbol{R_{\omega_i}}$: **If** a sample x is similar to $MI(\omega_i)$
Then its similarity with ω_1 is a_{i1} and ...
and its similarity with ω_K is a_{iK}.

The premise corresponds to the membership degree $\mu_{MI(\omega_i)}(x)$ of an unknown sample x to the intrinsic model of the class ω_i. It is computed using (1) with the t-conorm *max* which expresses the disjunction of the prototypes that compose the model:

$$\mu_{MI(\omega_i)}(x) = max_{l=1,\ldots,L_i}\, \mu_{F^l_{\omega_i}}(x)\,. \tag{5}$$

In the consequent part of the rule, the a_{ik} represents the weight of the model $MI(\omega_i)$ in the description of the class ω_k. They are obtained directly by using a basic optimizing technique: the pseudo-inverse algorithm [5]. Finally, the rules are aggregated using a *sum-product* inference coupled with a "defuzzification". This way, the intrinsic level produces a pre-classification vector $s^1(x) = \{s^1_1(x), \ldots, s^1_i(x), \ldots, s^1_K(x)\}$ where $s^1_i(x)$ represents the adequacy of a sample x to the class ω_i, according to the intrinsic modeling:

$$s^1_i(x) = \frac{\sum_{k=1}^{K} \mu_{MI(\omega_k)}(x) \cdot a_{ki}}{\sum_{j=1}^{K} \mu_{MI(\omega_j)}(x)}\,. \tag{6}$$

B. Formalization of the discriminant level

Each discriminant model (i.e. tree) $MD(\omega_{i+}, \omega_{i-})$ is formalized by a FIS, noted FIS_i, where each rule R_f represents the path from the root to the leaf N_f:

$\boldsymbol{R_f}$: **If** x satisfies the conditions leading to N_f
Then its similarity with ω_{i+} is b_{fi+} and
its similarity with ω_{i-} is b_{fi-}.

The premise represents the adequacy $\mu_{N_f}(x)$ of a sample x to the leaf N_f and it is determined using (4). The conjunction operator used is the t-norm *product* that takes into account the accumulation of imprecisions resulting from numerous partitionings. Doing so, short branches, more robust, are favored. The consequent part of the rule represents the weight of the leaf N_f in the description of each class ω_{i+} and ω_{i-}. These weights can be chosen to be the information contained in the leaf ($b_{fi+} = d_{fi+}$ and $b_{fi-} = d_{fi-}$). But here, as in the FIS of the intrinsic level, they are determined by the pseudo-inverse algorithm. The same inference operator is also used to provide homogeneous scores $s_i^2(x)$ representing the adequacy of a sample x to a class ω_i according to the discriminant modeling:

$$s_i^2(x) = \frac{\sum_{f=1}^{F} \mu_f(x) \cdot b_{fi+}}{\sum_{g=1}^{F} \mu_g(x)}, \tag{7}$$

where F is the number of leaves in the FDT that is considered.

V. Decision process

During the recognition process, the collaboration between the two kinds of knowledge through the focusing mechanism is exploited by fusing the two classification vectors provided by the two levels. When an unknown sample x must be recognized, the *pre-classification* vector $s^1(x)$ is first computed by inferring the FIS of the intrinsic level by (6). Then, the discriminant models corresponding to x are selected using the focusing mechanism (equation (2)): the FIS_i is selected if

$$\mu_{MI(\omega_i)}(x) \geq \frac{\max_{k=1,\ldots,K}(\mu_{MI(\omega_k)}(x))}{\alpha} \ . \tag{8}$$

The *main classification* vector $s^2(x) = \{s_1^2(x), \ldots, s_i^2(x), \ldots, s_K^2(x)\}$ is obtained by computing $s_i^2(x)$ using (7) if FIS_i is selected, or otherwise by setting $s_i^2(x) = 0$. The final classification vector $s(x) = \{s_1(x), \ldots, s_i(x), \ldots, s_K(x)\}$ is computed by an appropriate combination of the two intermediary classification vectors $s^1(x)$ and $s^2(x)$. The choice of the fusion operator is based on the properties of the information to combine. $s^1(x)$ is more stable but also less precise than $s^2(x)$. Therefore, it is difficult to know which one to favor. Moreover, $s^1(x)$ and $s^2(x)$ are both complementary: if one of the two scores is near 0 for a class, this one must be discarded. Finally, the fusion operator must provide a precise and graduated decision. Among the numerous classifier combination methods [42]–[45], the product operator is attractive since it is simple and has an adequate behavior considering the desired properties [44], especially thanks to its conjunctive aspect. Moreover, the focusing mechanism considerably limits the possible instabilities of this fusion operator [43]. The final classification is consequently obtained by:

$$s_i(x) = s_i^{1'}(x) \times s_i^{2'}(x) \ , \tag{9}$$

where $s_i^{1'}(x)$ and $s_i^{2'}(x)$ are the normalized version of the classification vectors: $s_i^1(x)$ and $s_i^2(x)$ divided by $\sum_{i=1}^{K} s_i^1(x)$ and $\sum_{i=1}^{K} s_i^2(x)$ respectively.

The rejection possibilities of the PFD strategy explained in section II-B were not used here but they can be implemented on the basis of the works described in [24].

VI. Experimental results

Two kinds of experiments were carried out to evaluate the Mélidis system and the PFD strategy. The first one is about the general performances of Mélidis and especially about its ability to deal with several recognition problems using the standard parameters given previously and its compactness. The comparison is made with classical algorithms whose performances were reported in the literature. The comparison with other MCS is only performed here with SVM [46] that operate for multiclass problems as a *one against all* multiple system. For other MCS, most of them are tested on specific benchmarks and the libraries are not easily accessible. Considering classical architectures of MCS such as Mixtures of Experts [7]–[10], their performances depend significantly on the type of classifiers used as experts (MLP, SVM, GMM, etc.) and on the way they are trained. Moreover, to have a significant comparison and to show the improvements of Mélidis and the PFD strategy over such systems, the main and/or final classification should also be based on such mixture principle which was not studied here.

Finally, the second kind of experiments illustrates the interest of the PFD strategy (i.e. the complementarity between intrinsic and discriminant knowledge through the focusing mechanism) by evaluating the improvements provided by each part of the system.

A. Benchmarks used

The benchmarks chosen here differ in the number of classes, the number of samples used for the learning and their intrinsic complexities. They are on the one hand, classical benchmarks and on the other hand, on-line handwritten character recognition benchmark.

Classical benchmarks:

The classifier has been evaluated using three databases of the UCI Repository[2]: the Breiman's waveforms (Wav.), the StatLog satellite images (Sat.) and the Pima indians diabetes (Diab.). The waveforms is a problem with 3 classes. The samples are described by 21 numerical features. 600 samples were used for learning and another $3\,000$ for tests. The satellite images, which is a problem with 6 classes, has 36 features. There is $4\,435$ samples for the learning and another $2\,000$ for tests. Finally, the diabetes dataset contains 768 instances from 2 classes in a 8 dimensional feature space.

On-Line handwritten character recognition benchmark:

Preliminary experiments for on-line digit recognition were carried out to evaluate the system on more complex and real-world problems. We report in this paper the results for on-line digit recognition using the IRONOFF [47] dataset (Iron.). The $4\,000$ initial samples were divided into 50% for the learning and the other 50% for the test. There are around 400 writers in the database and they are different in the learning set and in the test set (writer independent evaluation). There is no pre-processing of the digits which are directly described by a set of 44 high level features [41]. For this benchmark, the 20 most discriminative features on the training set were pre-selected by a third part algorithm (GA+classical RBF) to allow classifiers to work with less features if it is better.

B. Performances on classical benchmarks

The comparison is based on the results reported in the appendix of [48] (at the url given in the paper). Among the 33 algorithms evaluated in the paper, we report only those that are the best for at least one of the benchmarks. We also added the MDA (Mixture Discriminant Analysis) which is a Gaussian Mixture classifier and the Nearest Neighbor (NN) algorithms for their similarity with the concept of the PFD strategy or part of it. The Mélidis system has been evaluated using the same test protocols as those used in [48] to provide results as comparable as possible. More particularly, for the diabetes database, the data has been cleaned and a 10-fold cross validation used, as indicated in [48]. Table I reports the recognition rates and rank obtained by the different classifiers. Mélidis performs very well for all these problems: its recognition rates are very similar to the best ones or better. Moreover, it has the best mean recognition

[2] www.ics.uci.edu/∼mlearn/MLRepository.html

TABLE I
RECOGNITION RATES AND RANKS OF SEVERAL CLASSIFIERS FOR CLASSICAL BENCHMARKS.

	Wav. reco ; rank	Sat. reco ; rank	Diab. reco ; rank	Mean reco. ; rank
Mélidis	83.4% ; 5/34	89.6% ; 2/34	78.1% ; 1/34	83.4% ; 2.7
LVQ	83.0% ; 7/34	90.2% ; 1/34	75.7% ; 23/34	83.0% ; 10.3
RBF	84.9% ; 1/34	87.9% ; 3/34	77.0% ; 12/34	83.3% ; 5.3
LDA	82.2% ; 11.5/34	84.0% ; 17/34	77.9% ; 2.5/34	81.4% ; 10.3
FTL	82.1% ; 13.5/34	84.3% ; 24/34	77.9% ; 2.5/34	81.4% ; 13.3
MDA	83.7% ; 3/34	85.6% ; 10.5/34	76.9% ; 14/34	82.1% ; 9.2
NN	60.4% ; 33/34	78.3% ; 32/34	70.5% ; 33/34	69.7% ; 32.7

rate and the best mean rank over the 33 other algorithms in the same experimental conditions. To provide an estimation of the stability of the learning algorithm, a complementary experiment has been carried out: a 10 fold Cross-Validation (CV) has been performed, using only the learning sets for the Wav. and Sat. datasets (for the Diab. database, the whole set was yet used for a 10 CV in table I). The results obtained are for the waveforms: 83.3% of recognition rates with a standard error of 3.48; for the satellite images: 88.6% of recognition rates with a standard error of 1.42; and for the diabetes: 78.1% for recognition and 4.6 of standard error. These results confirm the stability of the learning process. So, these experiments are a first element showing the accuracy and the generic aspect of the classifier and of the PFD strategy. These points are strengthened by the results obtained for on-line handwritten digits recognition problems.

C. Performances on handwritten digits recognition

The Mélidis system has been compared on the IRONOFF data set with three other classifiers: a MLP (Multi-Layer Perceptron), a radial basis function network (RBF) and a support vector machine (SVM) which is known to be one of the most powerful classifiers at the moment. The MLP has one hidden layer, for which different numbers of neurons have been tested. Weights are learnt in a classical fashion by the back-propagation algorithm. The neurons of the RBF [14] are determined by a clustering algorithm and the weights by the pseudoinverse algorithm. Finally, the SVM comes from the SVMTorch II software proposed by [46]. It was trained in the multi-class mode (i.e. as a *one against all* MCS) using gaussian kernels, and different sets of parameters were tested to find the most accurate ones.[3] Considering the feature space, the MLP and the RBF give the best results with the 20 features pre-selected. The SVM uses the entire feature space and Mélidis uses the 20 features at the intrinsic level and the entire feature space at the discriminant level.

The results are summarized in table II where recognition rates (reco) and the number of parameters (param) are reported. For the Mélidis system, the number of parameters is evaluated on the basis of the number of fuzzy sets used at each level and their dimensionality. For example, at the intrinsic level, if the fuzzy sets $F^l_{\omega_i}$ are defined in N dimensions by their position $P^l_{\omega_i}$ and their covariance matrix $Cov^l_{\omega_i}$, they need $(N \times N) + N$ parameters. In the same way, the number of parameters for other classifiers are determined on the basis of the number of weights, neurons, radial basis functions, support vectors used, and on the basis of the number of parameters needed to define them. These results show that the Mélidis

TABLE II
COMPARISON OF THE PERFORMANCES OF THE MÉLIDIS SYSTEM AND SEVERAL CLASSIFIERS FOR ON-LINE DIGITS RECOGNITION (IRONOFF DATABASE).

	MLP	MLP	MLP	RBF	RBF	RBF	SVM	**Mélidis**
param	560	5 510	8 260	8 600	12 900	17 200	137 565	12 416
reco	91.1%	94.9%	93.9%	93.7%	94.6%	94.4%	95.5%	95.8%

[3]These parameters were estimated classically by splitting the training set into two sets: one for training and the other for validating.

system outperforms the MLP and the RBF classifiers. The SVM is the only one that obtains comparable performances. Nevertheless, it needs 10 times more parameters.

D. Complementarity between intrinsic and discriminant knowledge

Table III reports the recognition rates of the Mélidis system at each step of the decision process for the previous benchmarks. On these problems, we can see at each classification level that the recognition rates

TABLE III
RECOGNITION RATES OF THE DIFFERENT LEVELS OF MÉLIDIS ON DIFFERENT BENCHMARKS.

	Wav.	Sat.	Diab.	Iron.
Pre-classification	81.9%	84.3%	73.9%	94.5%
Main classification	82.4%	89.3%	77.6%	94.3%
Final classification	83.4%	89.8%	78.1%	95.8%

are improved. This means that both intrinsic and discriminant levels are complementary through this kind of architecture. The only exception concerns the IRONOFF database for which the main classification is not better than the pre-classification. But even in this case, the final classification takes advantage of the two modeling levels since the fusion process increases the recognition rates. This complementarity is also proved by considering the examples that produce errors: the number of errors that are common to both levels is less than the total number of errors made by each level separately. For example, on the waveforms, the intrinsic level makes 543 errors, the discriminant level makes 528 errors and only 304 are common to both levels. Similarly, for the satellite images, the intrinsic level makes 314 errors and only 132 are common with the 213 errors made by the discriminant level. This observation is similar for all the datasets. This means that the intrinsic and the discriminant levels do not make the same errors. This is why the fusion process is able to provide a higher final recognition rate.

To illustrate more precisely the contribution of each level to the final decision, Fig. 6 gives the relative error reduction of each classification level. The "pre-classification" error rates is the reference on which the error reduction of the other levels is computed (i.e. the reference). The "error reduction of the main classification" indicates the reduction obtained at the discriminant level. The "error reduction of the final classification" indicates the reduction obtained by the fusion of pre-classification and main classification in comparison with the pre-classification. The interest of the several levels and their collaboration appears

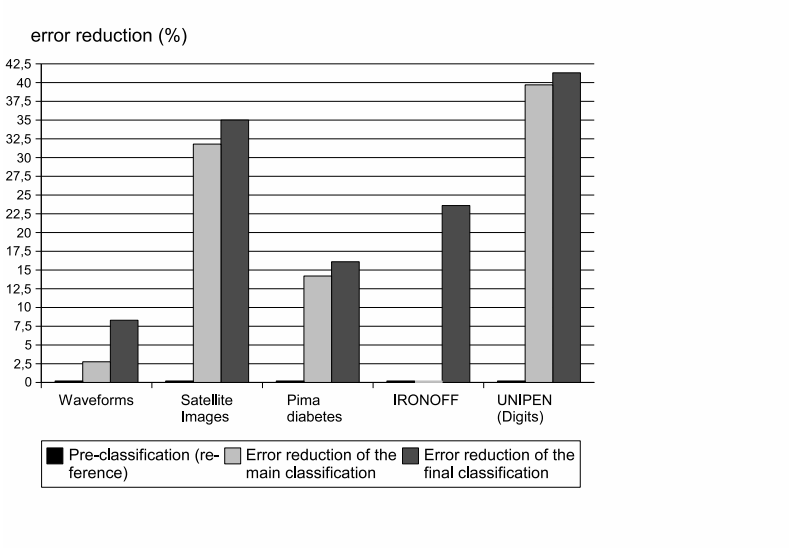

Fig. 6. Relative error reduction at each level of the Mélidis system.

more clearly on this chart.

E. Interest of the focusing mechanism

Another kind of experiments was also carried out to evaluate the interest of the focusing mechanism. The discriminant level of the Mélidis system has been compared with a forest of FDT: each FDT is trained to operate the discrimination of one class against all the others. Thus, the two approaches operate in a same way. Moreover, the learning algorithm and the configuration used for Mélidis and the forest are exactly the same. The classification vector for the forest of FDT is obtained in the exact same way as the main classification of Mélidis. Doing so, the only difference between the two approaches is the result of the focusing mechanism used in Mélidis and its filtering process based on the intrinsic knowledge.

The classification rates obtained with the forest of FDT are given in table IV and compared with results of the main classification of the Mélidis system for most of the previous benchmarks. The benefits of the

TABLE IV
EVALUATION OF THE INTEREST OF THE FOCUSING MECHANISM BY COMPARING RECOGNITION RATES OF A FOREST OF FDT AND THE MAIN CLASSIFICATION IN THE MÉLIDIS SYSTEM.

	Wav.	Sat.	Iron.
Forest of FDT	81.3%	83.3%	94.0%
Main classification of Mélidis	82.4%	89.3%	94.3%

focusing mechanism appears clearly since it provides an error reduction for all of the four experiments. This reduction goes up to 36% for the satellite images.

VII. CONCLUSION

In this paper, we have presented a strategy for pattern recognition based on a prototyping and focused discriminating (PFD) strategy. The main contribution lies in the particular collaboration of intrinsic and discriminant knowledge thanks to a focusing mechanism. Another specificity of the strategy is that the modeling is independent from the decision mechanism which makes optimization easier. Consequently, classifiers based on this strategy could benefits from several properties: accuracy, dealing with multimodal classes, rejection possibilities, flexibility for easier optimization. The Mélidis recognition system was designed to illustrate the PFD strategy in a fuzzy framework. Moreover this gives to the resulting classifier new properties: the ability to deal with noisy and variable data, compactness and legibility since the classifier is homogeneously formalized by a set of rules. Tests were carried out on classical benchmarks and on on-line handwritten digits recognition. In all cases, the results are the best or close to the best approaches using standard parameters. Moreover, the compactness of the modeling is greater than the one of an SVM.

Considering the Mélidis system, we would like to show how the optimization of the parameters (number of fuzzy prototypes per class, pruning of the FDT, choice of fuzzy operators, inference mechanism and fusion process, etc.) can be done thanks to the flexibility and the legibility of the modeling. Another work will also show how the to add rejection management to the Mélidis system thanks to the dual modeling of the PFD strategy. Considering the PFD strategy, other instantiation using other algorithms than the one used in the Mélidis system should be studied to show how general properties of the strategy could be enhanced and to show how new ones could be added.

ACKNOWLEDGMENT

The authors would like to thank Prof. Guy Lorette for its contribution to this work.

REFERENCES

[1] E. Fix and J. L. Hodges, "Discriminatory analysis–nonparametric discrimination: Consistency properties," USAF School of Aviation Medicine, Randolph Field, Texas, Project Number 21-49-004 4, 1951.

[2] R. Bajcsy and S. Kovačič, "Multiresolution elastic matching," *Comput. Vision, Graphics, Image Process.*, vol. 46, pp. 1–21, 1989.

[3] G. E. Hinton, C. K. I. Williams, and M. D. Revow, "Adaptative elastic models for hand-printed character recognition," *Advances in Neural Information Processing Systems*, pp. 512–519, 1992.

[4] H. Schwenk and M. Milgram, "Transformation invariant autoassociation with application to handwritten character recognition," in *Advances in Neural Information Processing Systems*, G. Tesauro, D. Touretzky, and T. Leen, Eds., vol. 7. The MIT Press, 1995, pp. 992–998.

[5] C. M. Bishop, *Neural Networks for Pattern Recognition*. Oxford University Press, 1995.

[6] C. J. C. Burges, "A tutorial on support vector machines for pattern recognition," *Data Mining and Knowledge Discovery*, vol. 2, no. 2, pp. 121–167, 1998.

[7] R. A. Jacobs, M. I.Jordan, S. J. Nowlan, and G. E. Hinton, "Adaptative mixtures of local experts," *Neural Computation*, vol. 3, no. 1, pp. 79–87, 1991.

[8] S. B. Ronan Collobert and Y. Bengio, "A parallel mixture of svms for very large scale problems," *Neural Computation*, vol. 14, no. 2, pp. 1105–1114, 2002.

[9] M. K. Titsias and A. Likas, "Mixture of experts classication using a hierarchical mixture model," *Neural Computation*, vol. 14, pp. 2221–2244, 2002.

[10] S. Akaho and H. J. Kappen, "Nonmonotonic generalization bias of gaussian mixture models," *Neural Computation*, no. 12, pp. 1411–1427, 2000.

[11] I. Ulusoy and C. M. Bishop, "Generative versus discriminative methods for object recognition," in *Proc. of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition(CVPR 2005)*, vol. 2, 2005, pp. 258– 265.

[12] G. Bouchard and B. Triggs, "The tradeoff between generative and discriminative classifiers," in *Proceedings in Computational Statistics, 16th Symposium of IASC*, J. Antoch, Ed., vol. 16. Prague: Physica-Verlag, 2004. [Online]. Available: http://lear.inrialpes.fr/pubs/2004/BT04/Bouchard-compstat04.pdf

[13] C. M. B. Julia A. Lasserre and T. P. Minka, "Principled hybrids of generative and discriminative models," in *EEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 1, 2006, pp. 87– 94.

[14] E. Anquetil, B. Coüasnon, and F. Dambreville, "A symbol classifier able to reject wrong shapes for document recognition systems," in *Graphics Recognition, Recent Advances*, ser. Lecture Notes in Computer Science, vol. 1941. Springer-Verlag, 2000, pp. 209–218.

[15] C. S. K.T. Abou-Moustafa and M. Cheriet, "A generative-discriminative hybrid for sequential data classification," in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '04)*, vol. 5, 2004, pp. 805–808.

[16] K. R. Ianakiev and V. Govindaraju, "Improvement of recognition accuracy using 2-stage classification," in *Proc. of the Seventh International Workshop on Frontiers in Handwriting Recognition*, L. Schomaker and L. Vuurpijl, Eds., 2000, pp. 153–165.

[17] N. Giusti, F. Masulli, and A. Sperduti, "Theorical and experimental analysis of a two-stage system for classification," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 7, pp. 893–904, 2002.

[18] E. Alpaydin, C. Kaynak, and F. Alimoğlu, "Cascading multiple classifiers and representations for optical and pen-based handwritten digit recognition," in *Proc. of the 7th International Workshop on Frontiers in Handwriting Recognition*, 2000, pp. 453–462.

[19] L. G. Vuurpijl and L. R. Schomaker, "Two-stage character classification: A combined approach of clustering and support vector classifiers," in *Proc. of the Seventh International Workshop on Frontiers in Handwriting Recognition*, L. Schomaker and L. Vuurpijl, Eds., 2000, pp. 423–432.

[20] L. Prevost, A. Moises, C. Michel-Sendis, L. Oudot, and M. Milgram, "Combining model-based and discriminative classifiers: application to handwritten character recognition," in *International Conference on Document Analysis and Recognition (ICDAR'03)*, vol. 1, 2003, pp. 31–35.

[21] C. Hsu and C. Lin, "A comparison of methods for multi-class support vector machines," 2001.

[22] E. Mayoraz and E. Alpaydin, "Support vector machines for multi-class classification," in *Proceedings of the International Workshop on Artifical Neural Networks (IWANN99)*, 1999, pp. 833–842.

[23] D. M. J. Tax and R. P. W. Duin, "Using two-class classifiers for multiclass classification," in *Proc. of the 16th International Conference on Pattern Recognition*, vol. 2, 2002, pp. 124–127.

[24] H. Mouchre and E. Anquetil, "A unified strategy to deal with different natures of reject," in *Proc. of the International Conference on Pattern Recognition (ICPR'06)*, 2006, pp. 792–795.

[25] T. Kohonen, "The self-organizing map," *Proc. of IEEE*, vol. 78, no. 9, pp. 1464–1480, 1990.

[26] A. Dempster, N. Laird, and D. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *Journal of the Royal Statistical Society, Series B*, vol. 39, no. 1, pp. 1–38, 1977.

[27] P. Mitra, C. Murthy, and S. K. Pal, "Density-based multiscale data condensation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 6, pp. 734–747, 2002.

[28] R. Krishnapuram and J. M. Keller, "A possibilistic approach to clustering," *IEEE Transactions on Fuzzy Systems*, vol. 1, no. 2, pp. 98–110, 1993.

[29] R. Krishnapuram, "Generation of membership functions via possibilistic clustering," in *IEEE World congress on computational intelligence*, 1994, pp. 902–908.

[30] R. Krishnapuram and J. M. Keller, "The possibilistic *c*-means algorithm: Insights and recommendations," *IEEE Transactions on Fuzzy Systems*, vol. 4, no. 3, pp. 385–393, 1996.

[31] X. L. Xie and G. Beni, "A validity measure for fuzzy clustering," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, no. 8, pp. 841–847, 1991.

[32] C. Z. Janikow, "Fuzzy decision trees: Issues and methods," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 28, pp. 1–14, 1998.

[33] C. Marsala and B. Bouchon-Meunier, "Choice of a method for the construction of fuzzy decision trees," in *Proc. of the Int. Conf. on Fuzzy Systems , FUZZ-IEEE'03*, 2003, pp. 584–589.

[34] C. Olaru and L. Wehenkel, "A complete fuzzy decision tree technique," *Fuzzy Sets and Systems*, no. 138, pp. 221–254, 2003.

[35] J. Y. jen Hsu and I.-J. Chiang, "Fuzzy classification trees," in *Ninth International Symposium on Artificial Intelligence in Joint Cooperation with the Sixth International Conference on Industrial Fuzzy Control and Intelligent Systems*, 1996, pp. 431–8.

[36] Y. Yuan and M. J. Shaw, "Induction of fuzzy decision trees," *Fuzzy sets and systems*, no. 69, pp. 125–139, 1995.

[37] N. Ragot and E. Anquetil, "A new hybrid learning method for fuzzy decision trees," in *Proc. of FUZZ-IEEE 2001*, vol. 3, 2001, pp. 1380–1383.

[38] J. C. Bezdek, *Pattern recognition with fuzzy objective function algorithms*. Plenum Press, 1981.

[39] H. Tanaka, T. Okuda, and K. Asai, "Fuzzy information and decision in statistical model," in *Advances in Fuzzy Set Theory and Applications*, 1979, pp. 303–320.

[40] C. Marsala and B. Bouchon-Meunier, "Measures of discrimination for the construction of fuzzy decision trees," in *Proc. of Fuzzy Information Processing (FIP'03)*, 2003, pp. 709–714.

[41] E. Anquetil and G. Lorette, "Automatic generation of hierarchical fuzzy classification systems based on explicit fuzzy rules deduced from possibilistic clustering: Application to on-line handwritten character recognition," in *Information Processing and Management of Uncertainty in Knowledge-Based Systems (IPMU'96)*, 1996, pp. 259–264.

[42] D. Bahler and L. Navarro, "Methods for combining heterogeneous sets of classifiers," in *Proc. of the 7th National Conference on Artificial Intelligence (AAAI 2000), Workshop on New Research Problems for Machine Learning*, 2000, http://citeseer.ist.psu.edu/470241.html.

[43] D. M. J. Tax, M. van Breukelen, R. P. W. Duin, and J. Kittler, "Combining multiple classifiers by averaging or by multiplying?" *Pattern Recognition*, vol. 33, pp. 1475–1485, 2000.

[44] I. Bloch, "Information combination operators for data fusion: A comparative review with classification," *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, vol. 26, no. 1, pp. 52–67, 1996.

[45] J. Kittler, M. Hatef, R. P. Duin, and J. Matas, "On combining classifiers," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 3, pp. 226–239, 1998.

[46] R. Collobert and S. Bengio, "SVMTorch: Support vector machines for large-scale regression problems," *Journal of Machine Learning Research*, vol. 1, pp. 143–160, 2001.

[47] C. Viard-Gaudin, P. M. Lallican, S. Knerr, and P. Binter, "The IRESTE on/off (IRONOFF) dual handwriting database," in *Proceedings of the Fifth International Conference on Document Analysis and Recognition (ICDAR'99)*, 1999, pp. 455–458.

[48] T.-S. Lim, W.-Y. Loh, and Y.-S. Shih, "A comparison of prediction accuracy, complexity, and training time of thirty-three old and new classification algorithms," *Machine Learning*, vol. 40, no. 3, pp. 203–228, 2000. [Online]. Available: http://www.stat.wisc.edu/~loh/quest.html