Learning Distributions of Image Features by Interactive Fuzzy Lattice Reasoning in Pattern Recognition Applications

Vassilis G. Kaburlasos, *Member, IEEE*, and George A. Papakostas

Human-Machines Interaction (HMI) Lab, Department of Computer and Informatics Engineering, Eastern Macedonia and Thrace Institute of Technology, 65404 Agios Loukas, Kavala, Greece

Abstract

This paper describes the recognition of image patterns based on novel representation learning techniques by considering higher-level (meta-)representations of numerical data in a mathematical lattice. In particular, the interest here focuses on lattices of (Type-1) Intervals' Numbers (INs), where an IN represents a distribution of image features including orthogonal moments. A neural classifier, namely fuzzy lattice reasoning (flr) fuzzy-ARTMAP (FAM), or flrFAM for short, is described for learning distributions of INs; hence, Type-2 INs emerge. Four benchmark image pattern recognition applications are demonstrated. The results obtained by the proposed techniques compare well with the results obtained by alternative methods from the literature. Furthermore, due to the isomorphism between the lattice of INs and the lattice of fuzzy numbers, the proposed techniques are straightforward applicable to Type-1 and/or Type-2 fuzzy systems. The far-reaching potential for deep learning in big data applications is also discussed.

Index Terms – Computer vision, fuzzy lattice reasoning, intervals' number, Jaccard similarity measure, type-2 fuzzy set

I. Introduction

Over the past decades, traditional computational intelligence has faced bottlenecks regarding algorithmic learning. In particular, one bottleneck has been the (slow) learning speed mainly due to gradient-based algorithms employed in the N-dimensional Euclidean space $\mathbb{R}^N$. Note that due to the conventional

Corresponding author Vassilis G. Kaburlasos contact e-mail: vgkabs@teikav.edu.gr

measurement procedure [12], traditional computational intelligence techniques are intimately linked to the notion of "feature space" such that an object is represented by a point (i.e., a vector of numbers) in $\mathbb{R}^N$. A vector data representation is popular mainly due to the abundance of analytical/computational tools available in $\mathbb{R}^N$. Nevertheless, a vector data representation itself is another bottleneck since it cannot represent sophisticated (data) semantics.

In response, on the one hand, novel techniques emerged to meet the (slow) learning speed including the extreme learning machines (ELMs) [2], [10]; the latter have reported good generalization performance even thousands of times faster than conventional feedforward neural networks. On the other hand, there is a sustained interest in learning in non-(geo)metric spaces involving data other than vectorial ones [9].

Non-vectorial data such as text, images, graphs, ontologies, hierarchies, schemata, etc, have proliferated with the proliferation of computers. Therefore, there is a need to deal with non-vectorial (or, equivalently, nonnumerical) data representations as well. We remark that in the context of "machine learning" it is accepted that the success of machine learning algorithms depends on the data representation [1]; moreover, representation learning might be the crux of the matter regarding *deep learning*, i.e., induction of more abstract – and ultimately more useful – data representations.

By departing from a vector space, one is confronted with the challenging task of defining (dis)similarity between non-vector data [9]. A popular approach for dealing with nonnumerical data is by "ad-hoc" transforming them to numerical ones. However, a problem with the aforementioned approach is that it introduces data distortions that might result in irreversible performance deterioration. Another approach for dealing with nonnumerical data is by developing domain-specific (mathematical) tools. Drawbacks of the latter approach include: first, different mathematical tools need to be "ad-hoc" devised in different (nonnumerical) data domains and, second, performance cannot, often, be tuned [3], [4]. Yet another approach has been proposed lately based on mathematical lattice theory as explained next.

The premise has been that popular types of data of interest in practical applications are lattice-ordered [12]. In conclusion, lattice computing, or LC for short, has been proposed as "an evolving collection of tools and mathematical modeling methodologies with the capacity to process lattice-ordered data *per se* including logic values, numbers, sets, symbols, graphs, etc" [6], [16], [29]. The existence of suitable real functions on lattice-ordered data allows for "fine-tuning" as demonstrated in this work. An advantage of LC is its capacity to rigorously compute with semantics represented by the lattice order relation. Specific examples of the LC approach are described in [13]. Recent trends in LC appear in [5], [14], [16], [27].

In the context of LC, of special interest here are Intervals' Numbers (INs) [13], [16], [21], [22], [26]. Recall that an IN is a mathematical object which can be interpreted either probabilistically or possibilistically [21]. Similarities as well as differences between Type-1 (respectively, Type-2) INs and Type-1 (respectively, Type-2) fuzzy sets have been reported [15]. In the remaining of this work "Type-1" will be denoted by "T1", for short; likewise, "Type-2" will be denoted by "T2". Our interest here is in digital image pattern recognition applications based on INs. We also discuss a potential enhancement of ELMs.

This paper builds on recently published work [16], [22], [24], [26] regarding human face recognition based on INs induced from orthogonal moments features. Differences with this work are summarized next. First, this work uses a (different) flrFAM classifier based on inclusion measure functions. Second, this work also employs 3-D T2 INs such that one 3-D T2 IN is represented by a $32 \times 32 \times 4$ matrix of real numbers. Third, this work employs comparatively additional features as well as additional classifiers in additional benchmark image pattern recognition problems. Furthermore, this work presents an improved mathematical notation as well as extensions of inclusion measure functions to the space of T2 INs. In addition, this work introduces an axiomatically extended Jaccard similarity measure.

The paper is organized as follows. Section II summarizes a hierarchy of mathematical lattices. Section III defines a similarity measure in a general lattice; moreover, it introduces Jaccard similarity measure extensions. Section IV outlines an flrFAM classifier. Section V describes the image pattern recognition problem as well as a technique for computing 3-D T2 INs. Section VI demonstrates, comparatively, computational experiments; it also includes a discussion of the results. Section VII concludes by summarizing our contribution as well as by describing potential future work.

II. A Hierarchy of Complete Lattices

This section introduces useful mathematical tools regarding INs [12], [13], [16], based on lattice theory.

Definition 2.1: Let $(\mathbb{P}, \sqsubseteq)$ be a mathematical lattice. A function $\sigma : \mathbb{P} \times \mathbb{P} \to [0, 1]$ is called *inclusion measure* iff the following two properties hold.

C1 $u \sqsubseteq w \Leftrightarrow \sigma(u, w) = 1$.

C2 $u \sqsubseteq w \Rightarrow \sigma(x, u) \leq \sigma(x, w)$.

We remark that an inclusion measure function $\sigma : \mathbb{P} \times \mathbb{P} \to [0, 1]$ can be interpreted as a fuzzy order relation on lattice $(\mathbb{P}, \sqsubseteq)$. Hence, the notations $\sigma(u, w)$ and $\sigma(u \sqsubseteq w)$ will be used interchangeably.

In the following we summarize a hierarchy of complete lattices in seven steps and define certain inclusion measure functions.

Step-1. We assume a totally-ordered, complete lattice $(\mathbb{L}, \leq)$ of real numbers, where $\mathbb{L} \subseteq \overline{\mathbb{R}} = \mathbb{R} \cup \{-\infty, +\infty\}$ with least and greatest elements denoted by o and i, respectively. The corresponding inf ($\wedge$) and sup ($\vee$) operators are the min and the max operators, respectively. In lattice $(\mathbb{L}, \leq)$ we consider both a strictly increasing function $v : \mathbb{L} \to [0, \infty)$, such that $v(o) = 0$ as well as $v(i) < +\infty$, and a strictly decreasing function $\theta : \mathbb{L} \to \mathbb{L}$, such that $\theta(o) = i$ as well as $\theta(i) = o$.

Step-2. We assume the partially-ordered, complete lattice $(\mathbb{I}_1, \subseteq)$ of (T1) intervals in lattice $(\mathbb{L}, \leq)$. The corresponding inf ($\cap$) and sup ($\dot{\cup}$) operations are given by $[a, b] \cap [c, d] = [a \vee c, b \wedge d]$ and $[a, b] \dot{\cup} [c, d] = [a \wedge c, b \vee d]$, respectively. We remark that if $a \vee c \not\leq b \wedge d$ then, by definition, $[a, b] \cap [c, d]$ equals the empty set ($\emptyset$). Two inclusion measure functions $\sigma_\cap : \mathbb{I}_1 \times \mathbb{I}_1 \to [0, 1]$ and $\sigma_{\dot{\cup}} : \mathbb{I}_1 \times \mathbb{I}_1 \to [0, 1]$ are given in lattice $(\mathbb{I}_1, \subseteq)$ based on a length function $V : \mathbb{I}_1 \to [0, \infty)$ as follows.

$$\sigma_\cap(x, y) = \begin{cases} 1, & \text{for } x = \emptyset. \\ \frac{V(x \cap y)}{V(x)}, & \text{for } x \supset \emptyset. \end{cases} \tag{1}$$

$$\sigma_{\dot{\cup}}(x, y) = \begin{cases} 1, & \text{for } x \dot{\cup} y = \emptyset. \\ \frac{V(y)}{V(x \dot{\cup} y)}, & \text{for } x \dot{\cup} y \supset \emptyset. \end{cases} \tag{2}$$

Recall that a length function $V : \mathbb{I}_1 \to [0, \infty)$ is defined as

$$V\left(x = [a_1, a_2]\right) = \begin{cases} 0, & x = \emptyset \\ v\left(\theta\left(a_1\right)\right) + v\left(a_2\right), & x \supset \emptyset \end{cases},$$

where functions $v(.)$ and $\theta(.)$ are as in Step-1.

Step-3. We assume the partially-ordered, complete lattice $(\mathbb{I}_2, \subseteq)$ of T2 intervals in lattice $(\mathbb{I}_1, \subseteq)$ – Recall that a T2 interval is defined as an interval of T1 intervals. The corresponding inf ($\cap$) and sup ($\dot{\cup}$) operations are given by $[[a_1, a_2], [b_1, b_2]] \cap [[c_1, c_2], [d_1, d_2]] = [[a_1 \wedge c_1, a_2 \vee c_2], [b_1 \vee d_1, b_2 \wedge d_2]]$, and $[[a_1, a_2], [b_1, b_2]] \dot{\cup} [[c_1, c_2], [d_1, d_2]] = [[a_1 \vee c_1, a_2 \wedge c_2], [b_1 \wedge d_1, b_2 \vee d_2]]$, respectively. We remark that if $[a_1 \wedge c_1, a_2 \vee c_2] \not\subseteq [b_1 \vee d_1, b_2 \wedge d_2]$ then, by definition, $[[a_1, a_2], [b_1, b_2]] \cap [[c_1, c_2], [d_1, d_2]]$ equals the empty set ($\emptyset$). Two inclusion measure functions $\sigma_\cap : \mathbb{I}_2 \times \mathbb{I}_2 \to [0, 1]$ and $\sigma_{\dot{\cup}} : \mathbb{I}_2 \times \mathbb{I}_2 \to [0, 1]$ are given in lattice $(\mathbb{I}_2, \subseteq)$ based on a length function $V : \mathbb{I}_2 \to [0, \infty)$ as follows.

$$\sigma_{\cap}\left([[a_1,a_2],[b_1,b_2]] \subseteq [[c_1,c_2],[d_1,d_2]]\right) = \begin{cases} 1, & b_1 > b_2. \\ 0, & b_1 \le b_2, b_1 \vee d_1 > b_2 \wedge d_2. \\ 0, & b_1 \le b_2, b_1 \vee d_1 \le b_2 \wedge d_2, \\ & \quad [a_1 \wedge c_1, a_2 \vee c_2] \not\subseteq [b_1 \vee d_1, b_2 \wedge d_2]. \\ \frac{V([[a_1,a_2],[b_1,b_2]] \cap [[c_1,c_2],[d_1,d_2]])}{V([[a_1,a_2],[b_1,b_2]])}, & \text{otherwise}. \end{cases} \tag{3}$$

$$\sigma_{\dot{\cup}}\left([[a_1,a_2],[b_1,b_2]] \subseteq [[c_1,c_2],[d_1,d_2]]\right) = \begin{cases} 1, & b_1 > b_2. \\ 0, & b_1 \le b_2, d_1 > d_2. \\ \frac{V([[c_1,c_2],[d_1,d_2]])}{V\left([[a_1,a_2],[b_1,b_2]] \dot{\cup} [[c_1,c_2],[d_1,d_2]]\right)}, & \text{otherwise}. \end{cases} \tag{4}$$

Recall that a length function $V : \mathbb{I}_2 \rightarrow [0, \infty)$ is defined as

$$V\left(x = [[a_1,a_2],[b_1,b_2]]\right) = \begin{cases} 0, & x = \emptyset. \\ v\left(a_1\right) + v\left(\theta\left(a_2\right)\right) + v\left(\theta\left(b_1\right)\right) + v\left(b_2\right), & x \supset \emptyset. \end{cases}$$

where functions $v(.)$ and $\theta(.)$ are as in Step-1.

Step-4. We assume the partially-ordered lattice $(\mathbb{F}_1, \preceq)$ of (T1) Intervals' Numbers, or (T1) INs for short. Recall that an IN is defined as a function $F : [0,1] \rightarrow \mathbb{I}_1$ that satisfies both $h_1 \le h_2 \Rightarrow F_{h_1} \supseteq F_{h_2}$ and $\forall X \subseteq [0,1] : \cap_{h \in X} F_h = F_{\bigvee X}$. In particular, an "interval (T1) IN F" is defined such that $F_h = [a,b], \forall h \in [0,1]$; in other words, the aforementioned interval (T1) IN $F \in \mathbb{F}_1$ represents the interval $[a,b] \in \mathbb{I}_1$. An IN is interpreted as an information granule [12]. An IN F can equivalently be represented either by a set of intervals $F_h, h \in [0,1]$ (an IN's *interval-representation*), or by a function $F(x) = \bigvee\limits_{h \in [0,1]} \{h : x \in F_h\}$ (an IN's *membership-function-representation*). For $F, G \in \mathbb{F}_1$ we have

$$F \preceq G \Leftrightarrow \left(\forall h \in [0,1] : F_h \subseteq G_h\right) \Leftrightarrow \left(\forall x \in \mathbb{L} : F(x) \le G(x)\right). \tag{5}$$

The height $hgt(F)$ of an IN F is defined as the supremum of its membership function values, i.e., $hgt(F) = \bigvee\limits_{x \in \mathbb{L}} F(x)$. The corresponding inf ($\curlywedge$) and sup ($\curlyvee$) operations in lattice $(\mathbb{F}_1, \preceq)$ are given by $(F \curlywedge G)_h = F_h \cap G_h$ and $(F \curlyvee G)_h = F_h \dot{\cup} G_h$, respectively, for $h \in [0,1]$. Next, we define two inclusion measure functions $\sigma_{\curlywedge} : \mathbb{F}_1 \times \mathbb{F}_1 \rightarrow [0,1]$ and $\sigma_{\curlyvee} : \mathbb{F}_1 \times \mathbb{F}_1 \rightarrow [0,1]$ based on the inclusion measure functions $\sigma_{\cap} : \mathbb{I}_1 \times \mathbb{I}_1 \rightarrow [0,1]$ and $\sigma_{\dot{\cup}} : \mathbb{I}_1 \times \mathbb{I}_1 \rightarrow [0,1]$, respectively.

$$\sigma_{\curlywedge}(E,F) = \int_{0}^{1} \sigma_{\cap}(E_h, F_h) dh. \tag{6}$$

$$\sigma_{\curlyvee}(E,F) = \int_{0}^{1} \sigma_{\dot{\cup}}(E_h, F_h) dh. \tag{7}$$

Specific advantages of an inclusion measure function in a Fuzzy Inference System (FIS) context have been reported [13].

Step-5. We assume the partially-ordered, complete lattice $(\mathbb{F}_2, \preceq)$ of T2 INs – Recall that a T2 IN is defined as an interval of T1 INs; that is, a T2 IN by definition equals $[U, W] \doteq \{X \in \mathbb{F}_1 \colon U \preceq X \preceq W\}$, where U is called lower IN, and W is called upper IN (of the T2 IN $[U, W]$). In the latter sense we say that X is encoded in $[U, W]$. The corresponding inf ($\curlywedge$) and sup ($\curlyvee$) operations in lattice $(\mathbb{F}_2, \preceq)$ are given by $(F \curlywedge G)_h = F_h \cap G_h$ and $(F \curlyvee G)_h = F_h \dot{\cup} G_h$, respectively. We can define two inclusion measure functions $\sigma_{\curlywedge} : \mathbb{F}_2 \times \mathbb{F}_2 \to [0, 1]$ and $\sigma_{\curlyvee} : \mathbb{F}_2 \times \mathbb{F}_2 \to [0, 1]$, based on the inclusion measure functions $\sigma_{\cap} : \mathbb{I}_2 \times \mathbb{I}_2 \to [0, 1]$ and $\sigma_{\dot{\cup}} : \mathbb{I}_2 \times \mathbb{I}_2 \to [0, 1]$, using equations (6) and (7), respectively. The computation of the join and meet operations in the lattice $(\mathbb{F}_2, \preceq)$ is demonstrated next.

Consider the two T2 INs $[f, F]$ and $[g, G]$ shown in Fig.1(a), where $f, F, g, G \in \mathbb{F}_1$ such that $f \preceq F$ and $g \preceq G$. The (join) T2 IN $[f, F] \curlyvee [g, G] = [f \curlywedge g, F \curlyvee G]$ is shown in Fig.1(b), where $(f \curlywedge g)_h = \emptyset, \forall h \in (h_1, 1]$. Fig.1(c) shows the (meet) T2 IN $[f, F] \curlywedge [g, G] = [f \curlyvee g, F \curlywedge G]$, where $(f \curlyvee g)_h = \emptyset, \forall h \in (h_3, 1]$, moreover $(F \curlywedge G)_h = \emptyset, \forall h \in (h_4, 1]$.

Step-6. The T1/T2 INs above have 2-dimensional (2-D) function representations, which can be extended to 3-dimensional (3-D) as follows. A 3-D T1 (respectively, T2) IN is defined as a function $F : [0, 1] \to \mathbb{F}$, where $\mathbb{F} = \mathbb{F}_1$ (respectively, $\mathbb{F} = \mathbb{F}_2$), which satisfies $z_1 \leq z_2 \Rightarrow F_{z_1} \succeq F_{z_2}$. In other words, a 3-D T1 (respectively, T2) IN F has 3-dimensional function representation F_z such that for constant $z = z_0$ the F_{z_0}, namely zSlice, is a 2-D T1 (respectively, T2) IN. A 3-D T2 IN example is plotted below. The symbol $\mathbb{F}_g$ denotes either the set of 3-D T1 INs or the set of 3-D T2 INs. It turns out that $(\mathbb{F}_g, \preceq)$ is a lattice whose order is $E \preceq F \Leftrightarrow E_z \preceq F_z, \forall z \in [0, 1]$. An inclusion measure function $\sigma_{\mathbb{F}_g} : \mathbb{F}_g \times \mathbb{F}_g \to [0, 1]$ is defined as

$$\sigma_{\mathbb{F}_g}(E,F) = \int_{0}^{1} \int_{0}^{1} \sigma_{\mathbb{I}}((E_z)_h, (F_z)_h) dh dz, \tag{8}$$

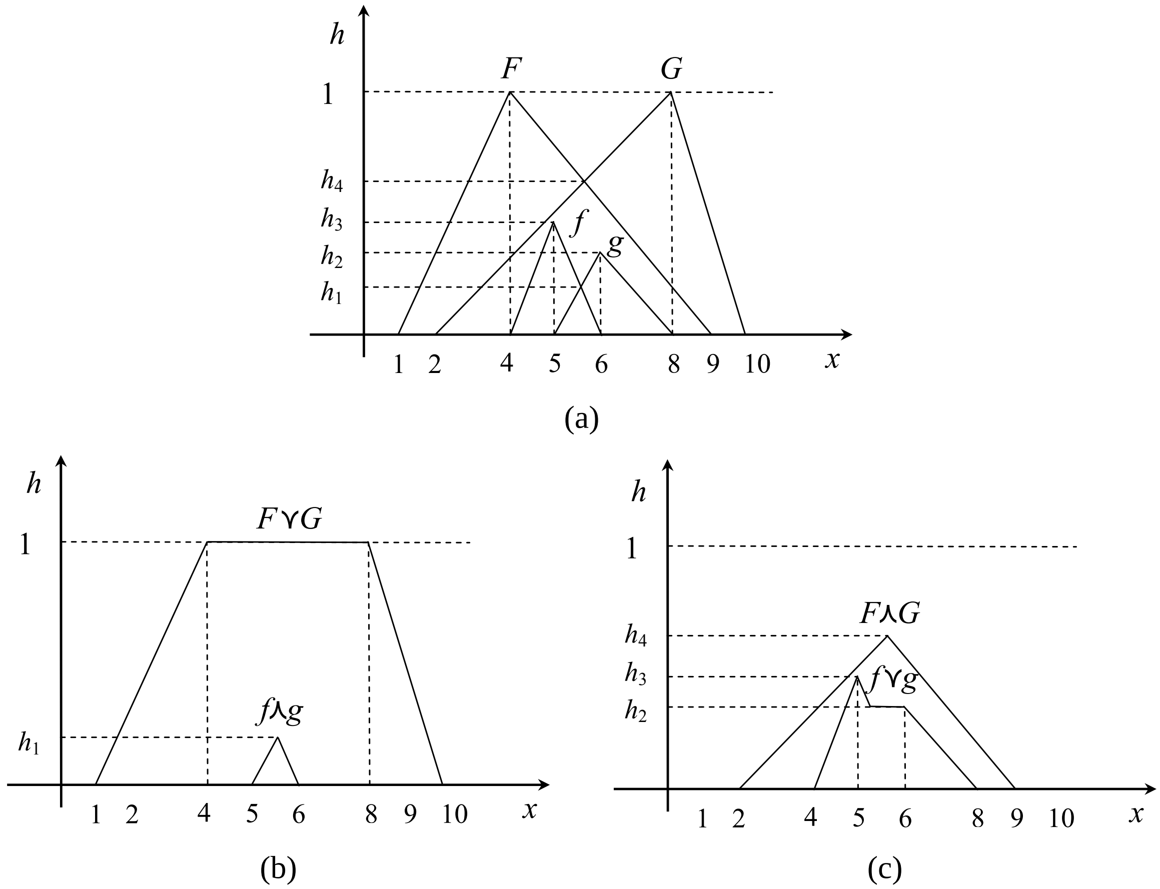


Fig. 1. (a) T2 INs $[f, F]$ and $[g, G]$, where $f, F, g, G \in \mathbb{F}_1$ such that $f \preceq F$ and $g \preceq G$. (b) The (join) T2 IN $[f, F] \curlyvee [g, G] = [f \curlywedge g, F \curlyvee G]$. (c) The (meet) T2 IN $[f, F] \curlywedge [g, G] = [f \curlyvee g, F \curlywedge G]$.

where $\sigma_{\mathbb{I}}(.,.)$ may be given by any one of the equations (1), (2), (3) and (4).

Step-7. We assume N-tuples of T1/T2 INs, where one N-tuple T1/T2 IN will be indicated by a boldface symbol, e.g. $\mathbf{X} = (X_1, \ldots, X_N)$. Given non-negative numbers $\lambda_1, \ldots, \lambda_N$ such that $\lambda_1 + \cdots + \lambda_N = 1$, an inclusion measure is defined in the complete lattice of N-tuple INs by the following convex combination

$$\sigma_c((X_1, \ldots, X_N), (Y_1, \ldots, Y_N)) = \sum_{i=1}^{N} \lambda_i \sigma_i(X_i, Y_i). \tag{9}$$

III. Similarity Measures on Lattices

Various definitions for (dis)similarity have been proposed in the literature in various data domains [3], [4], [25] without consensus. Motivated by a popular definition of similarity between fuzzy sets [25], we propose the following definition in a mathematical lattice.

Definition 3.1: Let $(\mathbb{P}, \sqsubseteq)$ be a mathematical lattice. A function $s : \mathbb{P} \times \mathbb{P} \rightarrow [0, 1]$ is called *similarity measure* iff the following three properties hold.

S1 $u = w \Leftrightarrow s(u,w) = 1$.

S2 $s(u,w) = s(w,u)$.

S3 $u \sqsubseteq v \sqsubseteq w \Rightarrow s(u,v) \geq s(u,w) \leq s(v,w)$.

Let $(\mathbb{P}, \sqsubseteq) = (\mathbb{P}_1, \sqsubseteq_1) \times \cdots \times (\mathbb{P}_N, \sqsubseteq_N) = (\mathbb{P}_1 \times \cdots \times \mathbb{P}_N, \sqsubseteq_1 \times \cdots \times \sqsubseteq_N)$ be the Cartesian product of N lattices; let function $s_i : \mathbb{P}_i \times \mathbb{P}_i \to [0,1]$ be a similarity measure on lattice $(\mathbb{P}_i, \sqsubseteq)$, $i \in \{1, \ldots, N\}$; let $\lambda_1, \ldots, \lambda_N$ be non-negative numbers such that $\lambda_1 + \cdots + \lambda_N = 1$. Then, as it will formally be proven elsewhere, the function $s : \mathbb{P} \times \mathbb{P} \to [0,1]$ given by the convex combination $s((U_1, \ldots, U_N), (W_1, \ldots, W_N)) = \lambda_1 s_1(U_1, W_1) + \cdots + \lambda_N s_N(U_N, W_N)$ is a similarity measure.

A. *Jaccard Similarity Measure Extensions*

Even though a number of similarity measures from the literature do not satisfy all the properties of Definition 3.1, the popular *Jaccard similarity measure* (or, equivalently, Jaccard coefficient) given by $\frac{|A \cap B|}{|A \cup B|}$ does satisfy them all. Next, we propose a parametric extension of the Jaccard similarity measure.

Let $(\mathbb{I}, \subseteq)$ be the complete lattice of either T1 intervals (i.e., $\mathbb{I} = \mathbb{I}_1$) or T2 intervals (i.e., $\mathbb{I} = \mathbb{I}_2$), and let $V : \mathbb{I} \to [0, \infty)$ be a length function on $\mathbb{I}$. Then, as it will formally be proven elsewhere, the function $J_{\mathbb{I}} : \mathbb{I} \times \mathbb{I} \to [0,1]$ given by $J_{\mathbb{I}}(A,B) = \frac{V(A \cap B)}{V(A \dot{\cup} B)}$, where $A \neq \emptyset$, is a similarity measure, namely extended Jaccard similarity measure. For non-overlapping intervals A and B, $J_{\mathbb{I}}(A,B)$ equals zero and vice versa. We extend $J_{\mathbb{I}}(.,.)$ to the complete lattice $(\mathbb{F}, \preceq)$ of T1/T2 INs.

Let $(\mathbb{F}, \preceq)$ be the complete lattice of either T1 INs (i.e., $\mathbb{F} = \mathbb{F}_1$) or T2 INs (i.e., $\mathbb{F} = \mathbb{F}_2$), and let $V : \mathbb{I} \to [0, \infty)$ be a length function on the corresponding lattice $(\mathbb{I}, \subseteq)$ of intervals. Then, as it will formally be proven elsewhere for INs with continuous membership functions, the function $J_{\mathbb{F}} : \mathbb{F} \times \mathbb{F} \to [0,1]$ given by $J_{\mathbb{F}}(A,B) = \int_0^1 J_{\mathbb{I}}(A_h, B_h) dh$ is a similarity measure.

Similarity measures can further be extended to 3-D T2 INs:

$$J_{\mathbb{F}_g}(E,F) = \int_0^1 \int_0^1 J_{\mathbb{I}}((E_z)_h, (F_z)_h) dh dz. \tag{10}$$

We remark that $J_{\mathbb{I}}((E_z)_h, (F_z)_h) = \frac{V((E_z)_h \cap (F_z)_h)}{V((E_z)_h)} \frac{V((E_z)_h)}{V((E_z)_h \dot{\cup} (F_z)_h)} = \sigma_{\cap}((E_z)_h, (F_z)_h) \sigma_{\dot{\cup}}((F_z)_h, (E_z)_h)$.

IV. An Interactive Fuzzy Lattice Reasoning (FLR) Neural Classifier

The flrFAM classifier is a single hidden layer neural architecture, inspired from the biologically motivated adaptive resonance theory [16] based on reasoning techniques [13]. This section proposes an enhancement of the flrFAM classifier in [16]. The latter was described by four algorithms: for clustering, for training (Structure Identification subphase), for training (Parameter Optimization subphase) and for testing. The difference between the algorithms employed here and the algorithms in [16] is that a neuron activation function $\alpha : \mathbb{F}_g^N \times \mathbb{F}_g^N \rightarrow [0, 1]$ here may be either a similarity measure or an inclusion measure function rather than the inclusion measure function $\sigma : \mathbb{I}_1^N \times \mathbb{I}_1^N \rightarrow [0, 1]$ in [16]. Hence, here we compute with distributions defined on a neighborhood rather than with the neighborhood alone.

Given $\mathbf{X} = (X_1, \ldots, X_N), \mathbf{W} = (W_1, \ldots, W_N) \in \mathbb{F}_g^N$, an activation function $\alpha : \mathbb{F}_g^N \times \mathbb{F}_g^N \rightarrow [0, 1]$ is computed by the convex combination $\alpha(\mathbf{X}, \mathbf{W}) = \lambda_1 \alpha_1(X_1, W_1) + \cdots + \lambda_N \alpha_N(X_N, W_N)$, where $\alpha_i : \mathbb{F}_g \times \mathbb{F}_g \rightarrow [0, 1]$, $i \in \{1, \ldots, N\}$, is an activation function in the lattice $(\mathbb{F}_g, \preceq)$. In particular, first, the activation function α_i can be an inclusion measure given by equation (8); therefore, in this case, the activation function $\sigma_{\mathbb{F}_g}(\mathbf{X}, \mathbf{W})$ filters h-level-wise an input datum $\mathbf{X} \in \mathbb{F}_g^N$ "bottom-up". Second, the activation function α_i can be the extended Jaccard similarity measure given by equation (10); hence, in this case, the activation function $J_{\mathbb{F}_g}(\mathbf{X}, \mathbf{W})$ simultaneously filters h-level-wise both an input datum $\mathbf{X} \in \mathbb{F}_g^N$ "bottom-up" and it filters a class code $\mathbf{W} \in \mathbb{F}_g^N$ "top-down" as indicated in the remark following equation (10).

The flrFAM algorithm here was inspired from Active Learning [19]. Nevertheless, active learning requires human intervention. We improved on active learning by assuming a "bottom-up"-"top-down" interplay between the training data and the class (learned) codes as it was explained above. In particular, a function $\sigma_{\mathbb{F}_g}(\mathbf{W_J}, \mathbf{X_i})$ always filters h-level-wise $\mathbf{W}_J \in \mathbb{F}_g^N$ "top-down". In conclusion, $\mathbf{W}_J \curlyvee \mathbf{X}_i$ may conditionally replace $\mathbf{W}_J$ depending on the (diagonal) size of $\mathbf{W}_J \curlyvee \mathbf{X}_i$ [12]. The capacity of the flrFAM classifier for generalization is demonstrated by the success rate S_{tst} on the testing dataset. An N-tuple of INs (granule) induced by the flrFAM classifier is interpreted as decision-making knowledge (i.e., a rule) induced from the data [15], [16], [21], [22].

V. The Image Pattern Recognition Problem and its Data Representation

This section demonstrates the capacity of our proposed techniques in image pattern recognition applications. The latter were selected due to the vast number of images generated globally, especially from mobile devices; hence, automated image learning as well as image pattern recognition is motivated, interesting as well as timely.

A. Data Preprocessing

We carried out the following three information processing tasks: #1. Image Acquisition, #2. Pattern Localization, and #3. Feature Extraction. Note that, typically, an image is represented in the literature as an N-dimensional point in the Euclidean space $\mathbb{R}^N$ by extracting features such as wavelet features, facial attributes, Gabor features, Zernike moments, etc [1], [16].

This paper retains a basic Feature Extraction employed elsewhere [16], [22], [24], [26]; that is, a population of numerical features is induced from an image to be learned/recognized. In particular, we induced orthogonal moments as well as other features due to their practical effectiveness [16], [22], [26]. Then, a distribution of features is "meta-represented" by an IN [24] induced by algorithm CALCIN [13]. A recent work [27] has demonstrated specific advantages for an IN meta-representation including a significant dimensionality reduction as well as a superior pattern recognition performance.

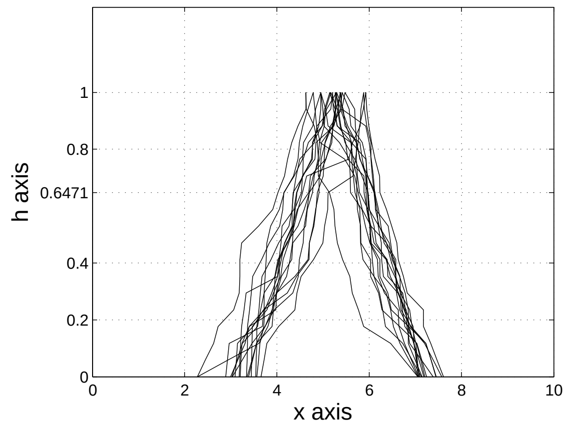
B. Image Pattern Representation

Recall that a population of features, which are induced from an image pattern, can be represented by a (T1) IN. This section investigates the representation of a class by a 3-D T1 IN (or a 3-D T2 IN) toward representing the distribution of T1 INs used for inducing it.

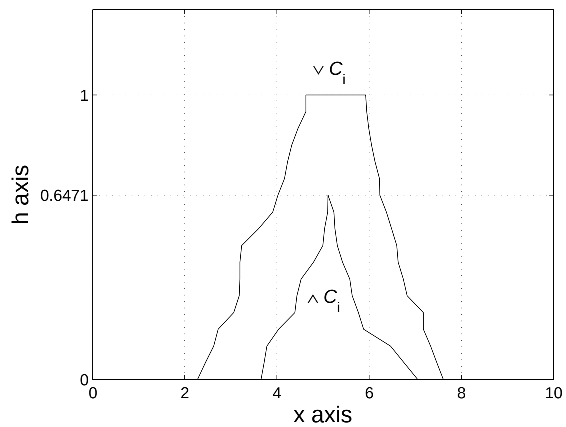
For example, consider the seventeen trivial T2 INs $[C_i, C_i], i \in \{1, \ldots, 17\}$ in Fig.2(a). Fig.2(b) displays the corresponding lattice join $\bigvee\limits_{i\in I}[C_i, C_i] = [\curlywedge\limits_{i\in I} C_i, \curlyvee\limits_{i\in I} C_i]$, $I = \{1, \ldots, 17\}$. Note that any inclusion measure function $\sigma : \mathbb{F}_2 \times \mathbb{F}_2 \to [0, 1]$ results in $\sigma([C_i, C_i], [\curlywedge\limits_{j\in I} C_j, \curlyvee\limits_{j\in I} C_j]) = 1$, $i \in I = \{1, \ldots, 17\}$ according to Definition 2.1. The lattice join $\bigvee\limits_{i\in I}[C_i, C_i] = [\curlywedge\limits_{j\in I} C_j, \curlyvee\limits_{j\in I} C_j]$ is a (2-D) T2 IN whose lower membership function $\curlywedge\limits_{j\in I} C_j$ has height $hgt(\curlywedge\limits_{j\in I} C_j) = 0.6471$ and whose upper membership function $\curlyvee\limits_{j\in I} C_j$ has height $hgt(\curlyvee\limits_{j\in I} C_j) = 1$, as shown in Fig.2(b).

A disadvantage of a 2-D T2 IN is that it does not retain any information regarding the distribution of INs used to induce it. We will try to turn the aforementioned disadvantage into an advantage by inducing an "h-secondary membership functions" as explained in Fig.3(a), where such functions will be induced at $h = 0.135$ and $h = 1$, respectively. Fig.3(b) displays two h-secondary membership functions along the line through $h = 0.135$, with supports $[2.65, 3.86]$ and $[6.3, 7.3]$, respectively; furthermore, Fig.3(c) displays one h-secondary membership function along the line through $h = 1$, with support $[4.6, 5.9]$.

Fig.4(a) displays the 3-D surface, which is induced from all the h-secondary membership functions, truncated by a plane through $z = 0.3$ parallel to the $x - h$ plane. By definition, a zSlice is the intersection



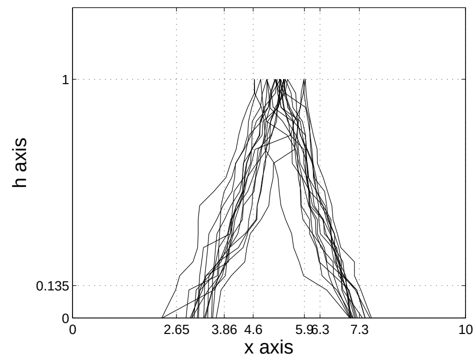
(a)



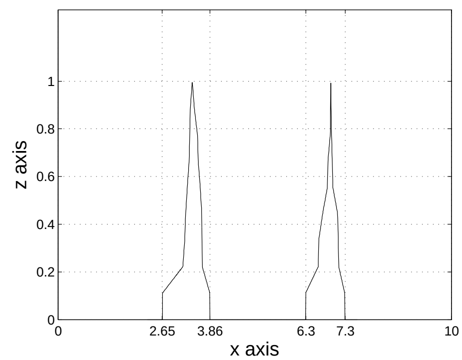
(b)

Fig. 2. (a) Seventeen trivial T2 INs $[C_i, C_i], i \in \{1, \ldots, 17\}$ are displayed in their membership-function-representation. (b) The lattice join $\bigvee_{i\in I}[C_i, C_i] = [\curlywedge_{i\in I} C_i, \curlyvee_{i\in I} C_i]$, where $I = \{1, \ldots, 17\}$.

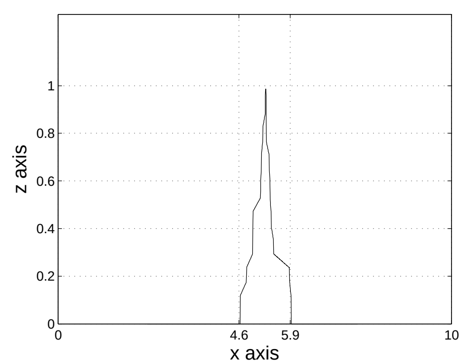
of the latter surface with a plane through $z \in [0, 1]$ parallel to the $x - h$ plane. By construction, a zSlice includes two functions, namely *primary membership functions*, defined by the ends of the supports of all the h-secondary membership functions on a zSlice. For example, the zSlice for $z = 0$ of the surface calculated from the INs in Fig.2(a) is the (2-D) T2 IN shown in Fig.2(b). Fig.4(b) displays the two primary membership functions on the zSlice shown in Fig.4(a) for $z = 0.3$. The (truncated) surface shown in Fig.4(a) is a 3-D T2 IN; whereas, the “T2 INs” shown in Fig.1 are 2-D T2 INs. Recall that the previously defined “(T1) INs” are alternatively called 2-D (T1) INs. Likewise, 3-D (T1) INs can be induced by computing h-secondary membership functions as detailed above.



(a)



(a))

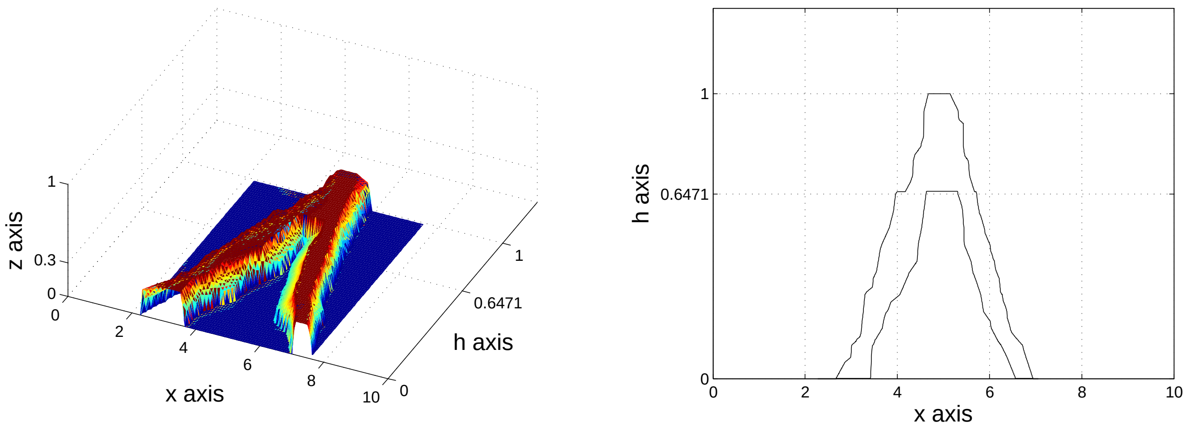


(b)

Fig. 3. (a) h-secondary membership functions will be induced at $h = 0.135$ and $h = 1$, respectively. (b) Two h-secondary membership functions at $h = 0.135$ with supports $[2.65, 3.86]$ and $[6.3, 7.3]$, respectively. (c) One h-secondary membership function at $h = 1$ with support $[4.6, 5.9]$.

VI. Experiments and Results

In this section we provide experimental evidence regarding the capacity of our proposed techniques in image pattern recognition applications. More specifically, we have dealt with image pattern recognition as a classification problem as explained below.



(a (b))

Fig. 4. (a) A 3-dimensional surface computed from the h-secondary membership functions of the INs in Fig.2(a), truncated (the surface) by a plane through $z = 0.3$ parallel to the $x - h$ plane. (b) The two primary membership functions on the zSlice of Fig.4(a) (for $z = 0.3$).

A. Benchmark Datasets and Feature Extraction

We have employed the following four benchmark datasets.

1) YALE dataset [30]: It regards face recognition. It contains 165 (8-bit) images (320×243 pixels each) of 15 individuals (i.e., classes). More specifically, there are 11 images per subject, one per different facial expression or configuration: center-light, w/glasses, happy, left-light, w/no glasses, normal, right-light, sad, sleepy, surprised, and wink. In order to remove irrelevant image content, the images were preprocessed by the Viola-Jones face detector followed by ellipse masking as described in [16]. The resulting localized faces were cropped to a fixed size of 32×32 pixels each.

2) TERRAVIC dataset [11]: It regards infrared face recognition. It contains $24,508$ images of 20 persons under different conditions such as front, left and right poses, indoor/outdoor environments with glasses and/or hat accessories; each image has an 8-bit, 320×240 pixels size. We used 70 images per person for the first 10 persons (i.e., classes) as described in [26].

3) JAFFE dataset [18]: It regards facial expression recognition. It contains 213 frontal images (256×256 pixels each) of 10 different persons corresponding to 7 common human facial expressions (i.e., classes), namely neutral (30), angry (30), disgusted (29), fear (32), happy (31), sad (31), and surprise (30) regarding Japanese female subjects, where a number within parentheses indicates the number of images available per facial expression. In order to remove irrelevant image content, the images were preprocessed as in the YALE dataset above. In conclusion, face images of 160×160 pixels each were produced.

4) TRIESCH I dataset [28]: It regards hand posture recognition. It contains 8-bit images (128×128

TABLE 1

CHARACTERISTICS OF THE IMAGE DATASETS USED IN 10-FOLD CROSS-VALIDATION EXPERIMENTS.

DATASET NAME	FEATURE TYPE (#FEATURES)	#INSTANCES	#TRAINING DATA PER FOLD	#TESTING DATA PER FOLD	#CLASSES
YALE	LBP (59)	165	149	16	15
TERRAVIC	ZMs (16)	700	630	70	10
JAFFE	dHMs (16)	213	192	21	7
TRIESCH I	HOG (9)	240	216	24	10

pixels each) of 10 hand postures (i.e., classes) regarding 24 persons in dark background.

It is understood that none of the above mentioned datasets is “big (data)”; nevertheless, any of the above datasets is big enough for the objectives here, where a large number of experiments were carried out toward comparing various classifiers.

We extracted six types of features per image. More specifically, we computed four different families of orthogonal moments including Zernike (ZMs), Gaussian-Hermite (GHMs), Tchebichef (TMs) and dual Hahn (dHMs) moments [23]; the order of each moment family was selected such that a 16-dimensional vector was produced. Another two types of features, popular in face recognition applications, were extracted, namely the Local Binary Pattern (LBP) and Histogram of Oriented Gradient (HOG) [24]. Regarding LBP, uniform patterns of $(R, N) = (1, 8)$ regions were computed. The vector length for LBP and HOG was 59 and 9, respectively.

Table 1 summarizes the characteristics of the image datasets used in our 10-fold cross-validation experiments. More specifically, the first column in Table 1 indicates the type of (image) feature that produced the best classification results for a dataset as well as the corresponding number of input features. For instance, for the YALE dataset, the best classification results were obtained for the 59 LBP input features, etc. The remaining columns in Table 1 display the number of instances (i.e., the total number of images used), the number of training data (per fold), the number of testing data (per fold), and the number of classes.

B. Experimental Setup

We employed ten traditional classifiers including three versions of the Minimum Distance Classifier (MDC) corresponding to the Chi Square (χ^2), Euclidean and Manhattan distances, respectively [26] – An MDC classifier here engaged “mean feature vectors” [24]. In addition, we employed a kNN (k=1), a Naïve-Bayes, an RBF ELM, a three-layer feedforward backpropagation Neural Network, and three types of Support Vector Machines (SVMs) including linear, polynomial (2^{nd} order) and RBF, respectively – The

Neural Network dimensions were $(no.features) \times (no.features) \times (no.classes)$. Both RBF SVM and RBF ELM [7] used a pair (C, γ) of tunable parameters computed optimally by the grid search method [8]. In conclusion, the pairs $(C, \gamma) = (2^5, 2^{-10})$ and $(C, \gamma) = (2^{12}, 2^4)$ were calculated and used for RBF SVM and RBF ELM, respectively, for all datasets. We also employed flrFAM classifiers as explained below. An flrFAM classifier processed INs induced from vectors of features, whereas an alternative classifier processed the corresponding vectors of features instead.

An flrFAM classifier represented a class by one N-tuple IN. In particular, we used two different class representations, namely per Feature (pF) and per Feature Vector (pFV), respectively [24] as follows. First, regarding pF, a T1 IN was induced from all values of a feature (i.e., dimension) in a class; second, regarding pFV, a T2 IN was induced from all T1 INs in a class, where one T1 IN was representing an image. Hence, the pF represented a class by one $(no.features)$-tuple of T1 INs, whereas the pFV represented a class by one T2 IN. Note that a T1 IN was either an interval in 2-D (thus resulting in a class representation by a hyperbox) of it could be a 3-D T1 IN; the former (hyperbox) IN representation was pursued by a “2-D T1 (interval) flrFAM” architecture, whereas the latter was pursued by a “3-D T1 flrFAM” architecture. Likewise, a T2 IN could be either a “2-D T2 (interval) IN” representation, pursued by a “2-D T2 (interval) flrFAM” architecture, or a “3-D T2 IN” one pursued by a “3-D T2 flrFAM” architecture. In all cases, we employed the IN interval-representation with $L = 32$ [13]. The activation function employed by an flrFAM classifier was based on either $\sigma_{\dot{\cup}}(.,.)$ or $J_{\mathbb{I}}(.,.)$.

For every classifier, on every dataset, we carried out a “10-fold cross-validation” computational experiment. More specifically, a dataset was partitioned in ten parts; nine-tenths of the dataset were used for training, whereas the remaining one-tenth was used for testing a classifier. In turn, all tenths of the dataset were used for testing. Care was taken so that all classes were represented fairly in both the training data and the testing data. The same training/testing data were used by all classifiers. As generalization rate we define the percentage $(\%)$ of the testing dataset classified correctly. For a “10-fold cross-validation” computational experiment we recorded both the minimum and the maximum generalization rates in 10 computational experiments as well as the corresponding average (ave) and standard deviation (std).

Regarding an flrFAM classifier we employed 10% of the training data for validation toward optimal parameter estimation [16]. More specifically, parameter optimization was pursued by a Genetic Algorithm (GA) such that the phenotype of an individual (flrFAM classifier) consisted of specific values for the two parameters $\lambda \in \mathbb{R}_0^+$ and $\mu \in \mathbb{R}$ of two functions, i.e. the (strictly increasing) sigmoid function

TABLE 2

YALE DATASET: PERCENTAGE (%) GENERALIZATION RATE STATISTICS IN 10 COMPUTATIONAL EXPERIMENTS BY SEVERAL CLASSIFIERS (LBP FEATURE)

CLASSIFIER NAME	[MIN, MAX]	AVE (STD)
01. MDC (Chisquare)	[40.00, 80.00]	56.67 (11.44)
02. MDC (Euclidean)	[26.67, 73.33]	42.67 (12.65)
03. MDC (Manhattan)	[33.33, 80.00]	53.33 (12.57)
04. kNN (k=1)	[20.00, 53.33]	38.67 (10.80)
05. Naïve-Bayes	[33.33, 60.00]	46.00 (7.98)
06. RBF ELM	[40.00, 73.33]	60.67 (9.66)
07. Neural Network (backprop)	[6.67, 20.00]	12.00 (4.22)
08. Linear SVM	[20.00, 53.33]	36.67 (10.06)
09. Polynomial SVM	[20.00, 33.33]	25.33 (5.26)
10. RBF SVM	[33.33, 66.67]	48.00 (10.80)
11. 2-D T1 flrFAM ($\sigma_{\dot\cup}$)	[20.00, 46.67]	35.33 (9.45)
12. 2-D T1 flrFAM ($J_{\mathbb{I}}$)	[6.67, 33.33]	22.00 (7.73)
13. 3-D T1 flrFAM ($\sigma_{\dot\cup}$)	[40.00, 73.33]	58.00 (11.78)
14. 3-D T1 flrFAM ($J_{\mathbb{I}}$)	[40.00, 73.33]	53.33 (12.96)

$v_s(x;\lambda,\mu) = 1/\left(1+e^{-\lambda(x-\mu)}\right)$ and the (strictly decreasing) function $\theta(x;\mu) = 2\mu - x$ as described in Step-1 of section II; an additional parameter was the baseline vigilance $\overline{\rho_a}$ [16]. Hence, a total number of three parameters per feature were binary-encoded in the chromosome of an individual. For both an inclusion measure and a Jaccard coefficient we used a convex combination with $\lambda_1 = \cdots = \lambda_N = \frac{1}{N}$.

To avoid a combinatorial explosion of the number of Tables with results presented in this paper, we selected the "best" feature type per benchmark dataset as follows. For each benchmark dataset, for each of the aforementioned ten traditional classifiers, we carried out a "10-fold cross-validation" computational experiment per feature type. Hence, for each benchmark dataset we recorded six Tables (i.e., one Table per feature type) including the generalization rate statistics of the ten traditional classifiers. Then, we selected the "best" feature type per benchmark dataset, that is the one that produced the highest overall classification results on the testing data. The "best" statistics of the aforementioned ten traditional classifiers, for the YALE dataset, are displayed in Table 2.

C. Computational Experiments and Results

We normalized the data by transforming them linearly to the unit interval $[0, 1]$. In the following, for lack of space, we display detailed generalization rate statistics regarding solely the YALE dataset.

1) Experiments with the YALE dataset: The LBP was the best feature selected as explained above. Table 2 displays the generalization rate statistics of the classifiers employed in this rather difficult classification problem. The 3-D T1 flrFAM classifiers on the average performed as good as or better than most classifiers, whereas the 2-D T1 (interval) flrFAM classifiers on the average performed around medium. Typically, an inclusion measure $\sigma_{\dot\cup}$ produced better results than a Jaccard similarity measure $J_{\mathbb{I}}$.

TABLE 3

AREA UNDER THE CURVE (AUC) STATISTICS "(AVERAGE, STANDARD DEVIATION)" IN 10 COMPUTATIONAL EXPERIMENTS BY SEVERAL CLASSIFIERS ON ALL DATASETS

	DATASET NAME			
CLASSIFER NAME	**YALE**	**TERRAVIC**	**JAFFE**	**TRIESCH I**
01. MDC (Chisquare)	0.73 (0.11)	0.99 (0.02)	0.68 (0.13)	0.95 (0.03)
02. MDC (Euclidean)	0.72 (0.11)	1.00 (0.01)	0.64 (0.12)	0.95 (0.03)
03. MDC (Manhattan)	0.73 (0.11)	1.00 (0.01)	0.65 (0.10)	0.94 (0.03)
04. kNN (k=1)	0.72 (0.10)	1.00 (0.00)	0.87 (0.07)	0.94 (0.03)
05. Naïve-Bayes	0.78 (0.12)	1.00 (0.00)	0.81 (0.10)	0.97 (0.03)
06. RBF ELM	0.90 (0.08)	0.83 (0.24)	0.80 (0.18)	0.92 (0.11)
07. Neural Network (backprop)	0.54 (0.09)	0.93 (0.05)	0.67 (0.09)	0.70 (0.09)
08. Linear SVM	0.86 (0.13)	1.00 (0.00)	0.84 (0.07)	0.88 (0.06)
09. Polynomial SVM	0.76 (0.16)	1.00 (0.00)	0.86 (0.04)	0.88 (0.07)
10. RBF SVM	0.87 (0.10)	0.95 (0.08)	0.76 (0.17)	0.92 (0.09)
11. 2-D T1 flrFAM ($\sigma_{\dot{\cup}}$)	0.72 (0.16)	1.00 (0.01)	0.71 (0.10)	0.95 (0.03)
12. 2-D T1 flrFAM ($J_{\mathbb{I}}$)	0.72 (0.17)	1.00 (0.01)	0.73 (0.13)	0.95 (0.03)
13. 3-D T1 flrFAM ($\sigma_{\dot{\cup}}$)	0.75 (0.15)	0.99 (0.04)	0.67 (0.12)	0.96 (0.03)
14. 3-D T1 flrFAM ($J_{\mathbb{I}}$)	0.75 (0.15)	0.99 (0.03)	0.65 (0.12)	0.96 (0.02)

2) Experiments with the TERRAVIC dataset: The ZMs was the best feature selected as explained above. Any T1 flrFAM (with $\sigma_{\dot{\cup}}$) classifier always gave the maximum generalization rate of 100%. Moreover, a 3-D T1 flrFAM classifier (with $J_{\mathbb{I}}$) performed clearly better than the corresponding 2-D T1 (interval) flrFAM classifier (with $J_{\mathbb{I}}$).

3) Experiments with the JAFFE dataset: The dHMs was the best feature selected as explained above. All flrFAM classifiers performed rather poorly. An inclusion measure $\sigma_{\dot{\cup}}$ typically produced better results than a Jaccard similarity measure $J_{\mathbb{I}}$.

4) Experiments with the TRIESCH I dataset: The HOG was the best feature selected as explained above. An flrFAM classifier on the average performed as good as or better than most classifiers. For the 2-D T1 (interval) flrFAM classifier, an inclusion measure $\sigma_{\dot{\cup}}$ on the average produced clearly larger generalization rates than a Jaccard similarity measure $J_{\mathbb{I}}$; it was vice versa for the 3-D T1 flrFAM classifier.

All computational experiments, for all benchmark datasets, with a T2 flrFAM classifier produced a generalization rate around $5\% - 15\%$ less than the generalization rate of its corresponding T1 flrFAM classifier. On the average, a 3-D T2 flrFAM classifier clearly outperformed its 2-D T2 (interval) counterpart.

In order to show the significance of the results for each classifier we carried out Receiver Operating Characteristics (ROC) curve analysis [16]. For lack of space, we only display the corresponding Area Under Curve (AUC) statistics (i.e., average and standard deviation) in Table 3 for all classifiers and all datasets in "10-fold cross-validation" computational experiments – Recall that the closer a classifier's AUC is to number 1 the better the classifier performs. Table 3 confirms comparatively the high confidence regarding the good performance of a 3-D T1 flrFAM classifier for all datasets but the JAFFE dataset.

D. Discussion of the Results

In this section we studied experimentally the performance of two different image representations, namely pF and pFV, in image pattern classification applications using a number of flrFAM classifiers applicable on INs induced from vectors of (image) features; alternatively, traditional classifiers were applied comparatively on the aforementioned vectors of features. In the context of this work, we did implement all the classifiers in order to compare them fairly on the same data for training/testing.

The pF image representation engaged one $(no.features)$-tuple of T1 INs per class, whereas the pFV image representation engaged one T2 IN per class. A pF (respectively, pFV) image representation was processed by a T1 (respectively, T2) flrFAM classifier. For any flrFAM classifier we tested all combinations of (2-D interval)/3-D INs with $\sigma_{\dot{\cup}}/J_{\mathbb{I}}$ functions. On the one hand, T1 flrFAM classifiers typically demonstrated a competitive image pattern recognition capacity compared to traditional classifiers. On the other hand, the T2 flrFAM classifiers on the average performed clearly ($5\% - 15\%$) less than their corresponding T1 counterparts thus confirming previous work [24]; that is, this work confirmed that the pF is a better image representation than the pFV representation. Our explanation is that the pFV representation mingles features from different data dimensions thus deteriorating their discriminative power.

Based on recorded experimental evidence, we confirmed that a 3-D T1 flrFAM classifier clearly outperformed its 2-D T1 (interval) counterpart. Our explanation is that a 3-D T1 IN represents more data statistics. More specifically, given that an IN represents a distribution, a 3-D T1 IN represents a distribution of (image features) distributions. Likewise, for the same reason, the (recorded) generalization rates for a 3-D T2 flrFAM classifier were clearly better than for its 2-D T2 (interval) counterpart.

An inclusion measure $\sigma_{\dot{\cup}}(.,.)$, in general, produced better generalization rates than a Jaccard similarity measure $J_{\mathbb{I}}(.,.)$. The latter was attributed to the fact that $J_{\mathbb{I}}(A,B)$ equals zero for nonoverlapping intervals A and B. In additional experiments we confirmed that the inclusion measure $\sigma_{\cap}(A,B)$ produced very similar results to the ones reported above for the $J_{\mathbb{I}}(A,B)$, for the same reason.

The average generalization rate of a 3-D T1 flrFAM classifier was not (statistically) significantly different from the corresponding average of the best of ten traditional classifiers in three benchmark image pattern recognition problems, namely YALE, TERRAVIC and TRIESCH I. Only for the JAFFE benchmark, the performance of any 3-D T1 flrFAM classifier clearly lagged behind the performance of three traditional classifiers, namely kNN (k=1), RBF ELM and (polynomial) SVM. We point out that our work in [16] has reported a competitive performance of an flrFAM classifier with the performance of a kNN (k=1) classifier due to two reasons: first, a flrFAM classifier in [16] employed 6-tuple INs induced from six different types of features (moments), concatenated and, second, a different FLR classifier in [16] induced more than one (N-tuple of INs) granule per class; whereas, here we used only one type of features as well as only one (N-tuple of INs) granule per class.

There is one more reason for characterizing “remarkable” the capacity of a 3-D T1 flrFAM classifier here for generalization. More specifically, recall that no flrFAM classifier was used for selecting the “best” feature type per benchmark dataset. Hence, the generalization rate of an flrFAM classifier here truly demonstrates its capacity for generalization.

Regarding the learning (time) complexity note that an flrFAM scheme involves both IN induction and parameter optimization. More specifically, first, IN induction typically requires negligible time. Second, parameter optimization typically requires substantial time due to the application of stochastic optimization techniques including a GA. However, note that previous work has demonstrated that GA application can be substantially accelerated by a parallel GA implementation [22].

Compared to the aforementioned traditional classifiers, an flrFAM scheme here was only faster than the (backprop) neural network. Nevertheless, recall that the performance of some traditional classifiers such as the RBF ELM/SVM depends critically on certain parameters (e.g., polynomial order, C, γ, etc) that can be tuned by grid search techniques [8]. We experimentally confirmed that a grid search applied to an flrFAM scheme here, results in a learning (time) complexity comparable to the learning (time) complexity of a traditional classifier. We also remark that an flrFAM testing (time) complexity is comparable to the testing (time) complexity of a traditional classifier.

VII. Conclusion

This work has introduced a mathematically sound extension of the flrFAM neural classifier to the (non-Euclidean) space of T1/T2 INs based on novel similarity/inclusion measure functions. A new type of learning was pursued, that is learning distributions of (image features) distributions. Comparative computational experiments regarding image pattern recognition have demonstrated the viability of the proposed techniques. The good generalizability of the proposed classification schemes was attributed to the synergy of INs, which can represent all order data statistics [14], with image feature descriptors.

The T1/T2 N-tuple of INs (granules) induced by an flrFAM classifier were interpreted as rules that may represent large numbers of data. In this sense, an flrFAM classifier could be valuable for data mining abstract representations in big data and deep learning applications [1]. Apart from interpreted probabilistically, an IN can also be interpreted possibilistically [21]. Hence, based on the isomorphism between T1/T2 INs and T1/T2 fuzzy sets, the mathematical tools proposed here are straightforward applicable to T2 fuzzy sets. The latter is significant due to the fact that within the fuzzy sets and systems community its "(linguistic) T2 component" currently clearly leads the way [20].

An advantageous ELM extension from space $(\mathbb{R}^N, \leq)$ to a cone $(\mathbb{F}^N, \preceq)$ of INs is feasible based on the fact that the algebraic operations of ELMs [2], [10] can be carried out in cones of T1/T2 INs [17]. Therefore, potential future work includes an extension of ELMs to a cone of INs with substantial expected benefits regarding the data processing speed. More specifically, even though an IN-based scheme (such as an flrFAM) processes its data (INs) slower than an ELM processes its own data (vectors), an IN-based scheme can improve the overall data processing speed by representing thousands/millions (or more) data by a single IN. In the aforementioned manner, an ELM extension to a cone of INs is promising toward further improving the data processing speed in certain big data applications.

Future work extensions of the flrFAM classifier will seek to optimize the number of (N-tuple of INs) granules induced per class as well as the induction of INs.

References

[1] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 8, pp. 1798–1828, 2013.

[2] E. Cambria, "Extreme learning machines," *IEEE Intelligent Systems*, vol. 28, no. 6, pp. 30–59, 2013.

[3] V. Cross, X. Yu, and X. Hu, "Unifying ontological similarity measures: A theoretical and empirical investigation," *Intl. J. of Approximate Reasoning*, vol. 54, no. 7, pp. 861–875, 2013.

[4] Y. Dou, L. Zhu, and H.S. Wang, "Solving the fuzzy shortest path problem using multi-criteria decision method based on vague similarity measure," *Applied Soft Computing*, vol. 12, no. 6, pp. 1621–1631, 2012.

[5] E. Esmi, P. Sussner, H. Bustince, and J. Fernández, "Theta-fuzzy associative memories (Theta-FAMs)," *IEEE Transactions on Fuzzy Systems*, vol. 23, no. 2, pp. 313–326, 2015.

[6] E. Esmi, P. Sussner, and S. Sandri, "Tunable equivalence fuzzy associative memories," *Fuzzy Sets and Systems*, doi: http://dx.doi.org/10.1016/j.fss.2015.04.004

[7] Extreme Learning Machines: http://www.extreme-learning-machines.org

[8] C.-W. Hsu and C.-J. Lin, "A comparison of methods for multiclass support vector machines," *IEEE Transactions on Neural Networks*, vol. 13, no. 2, pp. 415–425, 2002.

[9] D. Hofmann, A. Gisbrecht, and B. Hammer, "Efficient approximations of robust soft learning vector quantization for non-vectorial data," *Neurocomputing*, vol. 147, pp. 96–106, 2015.

[10] G.-B. Huang, L. Chen, and C.-K. Siew, "Universal approximation using incremental constructive feedforward networks with random hidden nodes," *IEEE Transactions on Neural Networks*, vol. 17, no. 4, pp. 879–892, 2006.

[11] IEEE OTCBVS WS Series Bench; Roland Miezianko, Terravic Research Infrared Database.

[12] V.G. Kaburlasos, *Towards a Unified Modeling and Knowledge-Representation Based on Lattice Theory*. Heidelberg, Germany: Springer, 2006.

[13] V.G. Kaburlasos and A. Kehagias, "Fuzzy inference system (FIS) extensions based on the lattice theory," *IEEE Transactions on Fuzzy Systems*, vol. 22, no. 3, pp. 531–546, 2014.

[14] V.G. Kaburlasos and T. Pachidis, "A lattice-computing ensemble for reasoning based on formal fusion of disparate data types, and an industrial dispensing application," *Information Fusion*, vol. 16, pp. 68-83, 2014.

[15] V.G. Kaburlasos, S.E. Papadakis, and A. Amanatiadis, "Binary image 2D shape learning and recognition based on lattice-computing (LC) techniques," *J. Math. Imag. & Vis.*, vol. 42, no. 2-3, pp. 118-133, 2012.

[16] V.G. Kaburlasos, S.E. Papadakis, and G.A. Papakostas, "Lattice computing extension of the FAM neural classifier for human facial expression recognition," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 24, no. 10, pp. 1526–1538, 2013.

[17] V.G. Kaburlasos, G.A. Papakostas, T. Pachidis, and A. Athinellis, "Intervals' numbers (INs) interpolation/extrapolation," in *Proc. IEEE Intl. Conf. on Fuzzy Systems (FUZZ-IEEE 2013)*, Hyderabad, India, July 7-10, 2013.

[18] M.J. Lyons, S. Akamatsu, M. Kamachi, and J. Gyoba, "Coding facial expressions with Gabor wavelets," in *Proc. 3rd Intl. Conf. on Automatic Face and Gesture Recognition (FG '98)*, Nara, Japan, April 14-16, 1998, pp. 200–205.

[19] J. Maiora, B. Ayerdi, and M. Graña, "Random forest active learning for AAA thrombus segmentation in computed tomography angiography images," *Neurocomputing*, vol. 126, pp. 71–77, 2014.

[20] J.M. Mendel, B. Bouchon-Meunier, H. Hagras, J. Kacprzyk, V. Kreinovich, and C.T. Lin, "Panel: Beyond Type-1: where is the non-standard fuzzy field heading and why?," in *2013 IEEE Intl. Conf. on Fuzzy Systems*, Hyderabad, India, July 7-10, 2013.

[21] S.E. Papadakis and V.G. Kaburlasos, "Piecewise-linear approximation of non-linear models based on probabilistically/possibilistically interpreted intervals' numbers (INs)," *Information Sciences*, vol. 180, no. 24, pp. 5060–5076, 2010.

[22] S.E. Papadakis, V.G. Kaburlasos, and G.A. Papakostas, "Two fuzzy lattice reasoning (FLR) classifiers and their application for human facial expression recognition," *Journal of Multiple-Valued Logic and Soft Computing*, vol. 22, no. 4-6, pp. 561–579, 2014.

[23] G.A. Papakostas, *Moments and Moment Invariants - Theory and Applications*. Xanthi, Greece: Science Gate Publishing, 2014, doi: http://dx.doi.org/10.15579/gcsr.vol1

[24] G.A. Papakostas and V.G. Kaburlasos, "Lattice Computing (LC) meta-representation for pattern classification," in *Proc. 2014 IEEE World Congress on Computational Intelligence (WCCI 2014), FUZZ-IEEE Program*, Beijing, China, 6-11 July 2014, pp. 39-44.

[25] G.A. Papakostas, A.G. Hatzimichailidis, and V.G. Kaburlasos, "Distance and similarity measures between intuitionistic fuzzy sets: a comparative analysis from a pattern recognition point of view," *Pattern Recognition Letters*, vol. 34, no. 14, pp. 1609–1622, 2013.

[26] G.A. Papakostas, V.G. Kaburlasos, and T. Pachidis, "Thermal infrared face recognition based on lattice computing (LC) techniques," in *Proc. IEEE Intl. Conf. on Fuzzy Systems (FUZZ-IEEE 2013)*, Hyderabad, India, July 7-10, 2013.

[27] G.A. Papakostas, A. Savio, M. Graña, and V.G. Kaburlasos, "A lattice computing approach to Alzheimer's disease computer assisted diagnosis based on MRI data," *Neurocomputing*, vol. 150, part A, pp. 37–42, 2015.

[28] J. Triesch and C. von der Malsburg, "Robust classification of hand postures against complex backgrounds," in *Proc. 2nd Intl. Conf. on Automatic Face and Gesture Recognition (FG '96)*, Killington, Vermont, USA, 1996, pp. 170–175.

[29] M.E. Valle and P. Sussner, "Quantale-based autoassociative memories with an application to the storage of color images," *Pattern Recognition Letters*, vol. 34, no. 14, pp. 1589–1601, 2013.

[30] Yale face database URL: http://vision.ucsd.edu/content/yale-face-database.