Piecewise-Linear Approximation of Nonlinear Models Based on Probabilistically/Possibilistically Interpreted Intervals' Numbers (INs)

S. E. Papadakis and Vassilis G. Kaburlasos

Department of Industrial Informatics

Technological Educational Institution of Kavala

65404 Kavala, Greece

Emails: {spap,vgkabs}@teikav.edu.gr

Abstract

Linear models are preferable due to simplicity. Nevertheless, non-linear models often emerge in practice. A popular approach for modeling nonlinearities is by piecewise-linear approximation. Inspired from Fuzzy Inference Systems (FISs) of TSK type as well as from Kohonen's Self-Organizing Map (KSOM) this work introduces a genetically optimized synergy based on Interval Numbers, or INs for short. The latter (INs) are interpreted here either probabilistically or possibilistically. The employment of mathematical lattice theory is instrumental. Advantages include accommodation of *granular* data, introduction of *tunable* nonlinearities, and induction of descriptive decision-making knowledge (rules) from the data. Both efficiency and effectiveness are demonstrated in three benchmark problems. The proposed computational method demonstrates invariably a better capacity for generalization; moreover, it learns orders of magnitude faster than alternative methods inducing clearly fewer rules.

Index Terms

. Fuzzy inference systems (FIS), Genetic optimization, Granular data, Intervals' number (IN), Lattice theory, Linear approximation, Rules, Self-organizing map (SOM), Similarity measure, Structure identification, TSK model

I. Introduction

The need to induce, efficiently, an effective model (real function) $y : \mathsf{R}^N \rightarrow \mathsf{R}^M$ arises frequently in practical applications. In particular, linear models $y(\mathbf{x}) = c_0 + c_1x_1 + c_2x_2 + ... + c_Nx_N$ are preferable due to simplicity. However, most often, the dependence of a system output y on the input variables $x_1, ..., x_N$ is nonlinear.

One way of modelling nonlinearities is by a piecewise-linear approximation. For instance, in the context of fuzzy sets and systems, the *TSK (Tagaki-Sugeno-Kang) fuzzy model*, described by Sugeno, Kang [47], Sugeno, Tanaka [48], Sugeno, Yasukawa [49], Takagi, Sugeno [52], combines linguistic (fuzzy) interpretations of its numeric inputs with a (locally, within a cluster) linear computation of an output in order to achieve a nonlinear input-to-output map. For the reader's convenience, the operation of a TSK model is summarized in the Appendix.

Critical for the computation of a TSK model is the computation of input data clusters. A popular clustering scheme is *Kohonen's self-organizing map (KSOM)* introduced by Kohonen [30] mainly for visualization of nonlinear relations of multidimensional data. Er, Li, Cai, Chen [10] have confirmed the capacity of KSOM for rapid data processing. Pascual-Marqui, Pascual-Montano, Kochi, Carazo [40] have reported a soft (fuzzy) KSOM synergy with conventional fuzzy c-means, where the code vectors are distributed on a regular low-dimensional grid. Moreover, Vuorimaa [59] has introduced a fuzzy extension of KSOM for function $f : \mathsf{R}^N \rightarrow \mathsf{R}$ approximation using triangular fuzzy membership functions, exclusively. Recently, Kaburlasos, Papadakis [21] have proposed *granular* (fuzzy) extensions of KSOM in classification applications.

This work introduces a synergy of TSK- with KSOM- inspired techniques towards an *efficient* as well as *effective* piecewise-linear approximation of nonlinear models as explained below. The proposed synergy builds on an established mathematical result, namely the "resolution identity theorem", presented by Zadeh [67], which specifies that a fuzzy set can (equivalently) be represented either by its membership function or by its α-cuts.

Note that even though a fuzzy set can be defined on any universe of discourse, in practice, the *real numbers* universe of discourse R is preferred as pointed out by Kaburlasos and Kehagias [20]. More specifically, "fuzzy numbers" are typically employed, for instance in Fuzzy Inference Systems (FISs). Recall that a *fuzzy number* is defined as a convex, normal fuzzy set, often with bounded support. A fuzzy number is defined on R with a *upper-semicontinuous* membership function as described in Kaburlasos [17], Vroman, Deschrijver, Kerre [58].

It turns out that a α-cut of a fuzzy number is an interval; hence, based on the aforementioned "resolution identity theorem", a fuzzy number can be represented by a set of intervals. In conclusion, Uehara, Fujise [54], Uehara, Hirota [55], Uehara, Koyama, Hirota [56] have proposed a novel FIS design in practical applications based on α-cuts (intervals) of fuzzy numbers — Advantages include faster (parallel) data processing "level-by-level", "orders-of-magnitude" smaller computer memory requirements, etc. Senturk, Erginel [43] have employed α-cuts for enhancing traditional control strategies. Furthermore, Cornelis, Deschrijver, Kerre [8], Nachtegael, Kerre [33] have considered α-cuts/intervals for fuzzy logic/morphology operations in theoretical studies involving ambiguity.

This work builds creatively on the "resolution identity theorem" by, first, considering the equivalent α-cuts (interval) representation for a fuzzy number and, second, by dropping the corresponding possibilistic interpretation. Hence, an *Intervals' Number (IN)* emerges as a mathematical object, which may admit either a *possibilistic* or a *probabilistic* interpretation as explained below. Advantages include an introduction of useful linear operations, tunable nonlinearities, a capacity to deal with granular data, etc. Instrumental for IN-based FIS analysis and design is (mathematical) *lattice theory (LT)* because the set of (closed) intervals on the real line is partially(lattice)-ordered.

For the reader's interest, the emergence of LT in information processing is outlined next.

Mathematical lattices have emerged in the first half of the nineteenth century as a spin off of work on formalizing propositional logic. During the next one hundred years LT was established, and compiled creatively by Garrett Birkhoff [3]. Currently, there is a number of research Communities that employ LT in various information processing domains including, first, *Logic and Reasoning* for automated decision-making (see in Xu, Ruan, Qin, Liu [65]), second, *Mathematical Morphology* for signal/image processing (see in Ritter, Wilson [42]), third, *Formal Concept Analysis* for knowledge-representation and information-retrieval (see in Ganter, Wille [12]), fourth, *Computational Intelligence* for clustering, classification, and regression applications (see in Kaburlasos [18]), etc.

There are two different approaches for employing LT in practice. The first approach, namely *order-based*, is based on semantics represented by the lattice(partial)-order as demonstrated also by Bloch [4], Ganter, Wille [12], Xu, Ruan, Qin, Liu [65]. The second approach, namely *algebra-based*, is based on the lattice(algebraic)-operations of meet ($\wedge$) and join ($\vee$) as demonstrated also by Graña, Villaverde, Maldonado, Hernandez [14], Ritter, Wilson [42], Soille [46], Valle, Sussner [57]. Various combinations of the aforementioned two approaches have also been reported, for instance in classification applications by da Silva, Sussner [9], Kaburlasos [18], Sussner, Esmi [50], [51]. In this work, we describe a novel combination of the aforementioned two approaches.

Previous work by Kaburlasos [17], [18], Kaburlasos, Kehagias [20], Kaburlasos, Papadakis [21], [23], has employed the term *Fuzzy Interval Number (FIN)* instead of the term *Intervals' Number (IN)*, because it stressed a fuzzy interpretation. Recently, Kaburlasos, Papadakis [22] have switched to the term IN, including also an improved mathematical notation. Likewise, the term "CALFIN", proposed previously for an algorithm which computes a "FIN" from a population of measurements, is eloquently replaced here by the term "CALCIN".

This paper presents significant enhancements over the preliminary work by Kaburlasos, Papadakis in [22] as follows. First, we introduce a novel similarity measure function (μ_θ). Second, we detail structure/parameter identification algorithms based on μ_θ rather than on metric d_θ, the latter was employed in [22]; here, we also compute the corresponding algorithm complexity. Third, we demonstrate an employment of a IN as either a probability- or a possibility- distribution. Fourth, we demonstrate three additional benchmark problems including improved experimental results; moreover, in all benchmark problems, we display the induced rules. Fifth, we discuss novel theoretical perspectives. Sixth, we cite a large number of additional references including comparative discussions.

This paper is organized as follows. Section II summarizes the mathematical background. Section III presents a novel structure identification. Section IV describes a novel parameter identification. Section V details, comparatively, experimental results. Section VI concludes by summarizing our contribution including also future work. The Appendix includes the proof of a proposition as well as two computational algorithms used in the experiments.

II. Mathematical Background

This section summarizes useful mathematical results and tools introduced by Kaburlasos [18], Kaburlasos, Kehagias [19], [20], Kaburlasos, Papadakis [21], [22], [23], Kaburlasos, Athanasiadis, Mitkas [24]. Mathematical lattice theory here is instrumental.

Recall from Birkhoff [3] that given a set P, a binary relation ($\leq$) on P is called *partial order* if and only if it satisfies the following conditions: $x \leq x$ (*reflexivity*), $x \leq y$ and $y \leq x \Rightarrow x = y$ (*antisymmetry*), and $x \leq y$ and $y \leq z \Rightarrow x \leq z$ (*transitivity*). A *partially ordered set*, or *poset* for short, is a pair $(P, \leq)$, where P is a set and $\leq$ is a partial order relation on P. A *(crisp) lattice* is a poset $(\mathsf{L}, \leq)$ any two of whose elements $x, y \in \mathsf{L}$ have both a *greatest lower bound*, or *meet* for short, and a *least upper bound*, or *join* for short, denoted by $x \wedge y$ and $x \vee y$, respectively. A lattice $(\mathsf{L}, \leq)$ is called *complete* when each of its subsets X has both a greatest lower bound and a least upper bound in L. For simplicity, we will use the same symbols O and I to denote the least and greatest element, respectively, in any complete lattice.

A. The Vector Lattice (Δ,≤) of Generalized Intervals

Consider the *complete latice* (R,$\leq$) of real numbers with *least* and *greatest* elements denoted, respectively, by $O = -\infty$ and $I = +\infty$. A *generalized interval* is defined in the following.

Definition 1: Generalized interval is an element of the product lattice (R,$\leq^{\partial}$)$\times$(R,$\leq$).

We remark that $\leq^{\partial}$ in Definition 1 denotes the *dual* (i.e. converse) of order relation $\leq$, i.e. $\leq^{\partial} \equiv \geq$. Product lattice ($\mathsf{R}$,$\leq^{\partial}$)$\times$($\mathsf{R}$,$\leq$) $\equiv$ ($\mathsf{R} \times \mathsf{R}$,$\geq \times \leq$) will be denoted, simply, by (Δ,$\leq$).

A generalized interval will be denoted by $[x, y]$, where $x, y \in \mathsf{R}$. The *meet* ($\wedge$) and *join* ($\vee$) in lattice (Δ,$\leq$) are given, respectively, by $[a, b] \wedge [c, d] = [a \vee c, b \wedge d]$ and $[a, b] \vee [c, d] = [a \wedge c, b \vee d]$, where $a \wedge c$ ($a \vee c$) denotes the *minimum* (*maximum*) of real numbers a and c.

The set of *positive* (*negative*) generalized intervals $[a, b]$, characterized by $a \leq b$ ($a > b$), is denoted by Δ_+ (Δ_-). Apparently, (Δ_+,$\leq$) is a poset, namely *poset of positive generalized intervals*. Furthermore, poset (Δ_+,$\leq$) is *isomorphic*[1] to the poset ($\tau(\mathsf{R})$,$\leq$) of intervals (sets) in R, i.e. ($\tau(\mathsf{R})$,$\leq$) $\cong$ (Δ_+,$\leq$). We augmented poset ($\tau(\mathsf{R})$,$\leq$) by a *least* (empty) interval, denoted by $O = [+\infty, -\infty]$ – Note that a *greatest* interval $I = [-\infty, +\infty]$ already exists in $\tau(\mathsf{R})$. Hence, the complete lattice ($\tau_O(\mathsf{R}) = \tau(\mathsf{R}) \cup \{O\}$,$\leq$)$\cong$ $(\Delta_+ \cup \{O\}, \leq)$ emerged. Due to the latter isomorphism, we will employ lattices $(\Delta_+ \cup \{O\}, \leq)$ and ($\tau_O(\mathsf{R})$,$\leq$), interchangeably.

A (strictly) decreasing *bijective*, i.e. "one-to-one", function $\theta_{\mathsf{R}} : \mathsf{R} \to \mathsf{R}$ implies isomorphism (R,$\leq$) $\cong$ (R,$\geq$); i.e. $x < y \Leftrightarrow \theta_{\mathsf{R}}(x) > \theta_{\mathsf{R}}(y)$, $x, y \in \mathsf{R}$. Furthermore, a strictly increasing function $v_{\mathsf{R}} : \mathsf{R} \to \mathsf{R}$ is a *positive valuation*[2] in lattice (R,$\leq$). We will refer to functions $\theta_{\mathsf{R}}(.)$ and $v_{\mathsf{R}}(.)$ as *dual isomorphism* and *positive valuation*, respectively. It follows that function $v_\Delta : \Delta \to \mathsf{R}$ given by $v_\Delta([a, b]) = v_{\mathsf{R}}(\theta_{\mathsf{R}}(a)) + v_{\mathsf{R}}(b)$ is a positive valuation in lattice (Δ,$\leq$). Furthermore, it follows a metric function $d_\Delta : \mathsf{R} \to \mathsf{R}^{\geq 0}$ given by $d_\Delta([a, b], [c, d]) = [v_{\mathsf{R}}(\theta_{\mathsf{R}}(a \wedge c)) - v_{\mathsf{R}}(\theta_{\mathsf{R}}(a \vee c))] + [v_{\mathsf{R}}(b \vee d) - v_{\mathsf{R}}(b \wedge d)]$. In particular, metric d_Δ is valid in lattice ($\tau_O(\mathsf{R})$,$\leq$).

[1] A map $\psi : (P, \leq) \to (Q, \leq)$ is called *(order) isomorphism* if and only if both "$x \leq y \Leftrightarrow \psi(x) \leq \psi(y)$" and "$\psi$ is onto Q". Two posets $(P, \leq)$ and $(Q, \leq)$ are called *isomorphic*, symbolically $(P, \leq) \cong (Q, \leq)$, if and only if there is an isomorphism between them.

[2] *Positive valuation* in a lattice $(\mathsf{L}, \leq)$ is a real function $v : \mathsf{L} \times \mathsf{L} \to \mathsf{R}$ that satisfies both $v(x) + v(y) = v(x \wedge y) + v(x \vee y)$ and $x < y \Rightarrow v(x) < v(y)$.

Functions $\theta_{\mathsf{R}}(.)$ and $v_{\mathsf{R}}(.)$ can be selected in various ways. For instance, choosing $\theta_{\mathsf{R}}(x) = -x$ and $v_{\mathsf{R}}(.)$ such that $v_{\mathsf{R}}(x) = -v_{\mathsf{R}}(-x)$ it follows positive valuation $v_{\Delta}([a,b]) = v_{\mathsf{R}}(b) - v_{\mathsf{R}}(a)$; hence, it follows metric $d_{\Delta}([a,b],[c,d]) = [v_{\mathsf{R}}(a \vee c) - v_{\mathsf{R}}(a \wedge c)] + [v_{\mathsf{R}}(b \vee d) - v_{\mathsf{R}}(b \wedge d)]$. In particular, for $v_{\mathsf{R}}(x) = x$ it follows metric $d_{\Delta}([a,b],[c,d]) = |a-c| + |b-d|$. In general, *parametric* functions $\theta_{\mathsf{R}}(.)$ and $v_{\mathsf{R}}(.)$ may introduce tunable nonlinearities.

The space Δ of generalized intervals is a *real linear space* with

- *addition* defined as $[a,b] + [c,d] = [a+c, b+d]$, and
- *multiplication* (by a scalar $k \in \mathsf{R}$) defined as $k[a,b] = [ka, kb]$.

A generalized interval in Δ is a *vector*. Moreover, the lattice-ordered vector space Δ is called *vector lattice*.

A subset C of a linear space is called *cone* if and only if for $x_1, x_2 \in C$ and real numbers $\lambda_1, \lambda_2 \geq 0$ it follows $(\lambda_1 x_1 + \lambda_2 x_2) \in C$. It turns out that the set Δ_+ is a cone. Likewise, the set Δ_- is a cone.

B. The Cone Lattice (F,$\leq$) of Intervals' Numbers (INs)

Generalized interval analysis in the previous subsection is extended to *intervals' numbers* (INs) in this subsection. A more general number type is defined in the first place, next.

Definition 2: Generalized interval number, or *GIN* for short, is a function $G : (0,1] \rightarrow \Delta$.

Let G denote the set of GINs. Since $(\mathsf{G}, \leq)$ is the Cartesian product of complete lattices $(\Delta, \leq)$ it follows that $(\mathsf{G}, \leq)$ is a complete lattice. Addition and multiplication can be extended from Δ to G as follows.

- *Addition* is defined as $G_s : G_s(\alpha) = (G_1 + G_2)(\alpha) = G_1(\alpha) + G_2(\alpha)$, $\alpha \in (0,1]$.
- *Multiplication* (by a scalar $k \in \mathsf{R}$) is defined as $G_p : G_p(\alpha) = kG_1(\alpha)$, $\alpha \in (0,1]$.

Our interest here focuses on the *sublattice*[3] of *intervals' numbers* defined next.

Definition 3: An *Intervals' Number*, or *IN* for short, is a GIN F such that both $F(\alpha) \in (\Delta_+ \cup \{O\})$ and $\alpha_1 \leq \alpha_2 \Rightarrow F(\alpha_1) \geq F(\alpha_2)$.

Let F denote the set of INs. Conventionally, a IN will be denoted by a capital letter in italics, e.g. $F \in \mathsf{F}$. Moreover, a N-tuple IN will be denoted by a capital letter in bold, e.g. $\mathbf{F} = (F_1, ..., F_N) \in \mathsf{F}^N$.

Definition 3 implies that a IN F equals the set union of (conventional) intervals, e.g. $F = \underset{\alpha \in (0,1]}{\cup} \{[a_\alpha, b_\alpha]\}$, where both interval-ends a_α and b_α are functions of $\alpha \in (0,1]$.

A IN is a mathematical object, which may be interpreted as a probability/possibility distribution, an interval, and/or a real number as explained in the following. IN $F = \underset{\alpha \in (0,1]}{\cup} \{[a,b]\}$ represents interval $[a,b]$ including real numbers for $a = b$. Moreover, IN $F = \underset{\alpha \in (0,1]}{\cup} \{F(\alpha)\}$ may represent a probability distribution such that interval $F(\alpha)$ includes $100(1-\alpha)\%$ of the distribution, whereas the remaining $100\alpha\%$ is split even both below and above

[3] A *sublattice* of a lattice $(\mathsf{L}, \leq)$ is another lattice $(\mathsf{S}, \leq)$ such that $\mathsf{S} \subseteq \mathsf{L}$.

interval $F(\alpha)$. In addition, due to the "resolution identity theorem", a IN $F = \underset{\alpha\in(0,1]}{\cup}\{F(\alpha)\}$ may also represent a fuzzy number, where $F(\alpha)$ is the corresponding α-cut. Hence, a IN $F : (0,1] \rightarrow \tau_O(\mathsf{R})$ may, equivalently, be represented by a membership function $m_F : \mathsf{R} \rightarrow (0,1]$ as explained next.

On one hand, $(\mathsf{F}, \leq)$ is a lattice with ordering $F_1 \leq F_2 \Leftrightarrow F_1(\alpha) \leq F_2(\alpha)$, $\alpha \in (0,1]$. On the other hand, using the conventional (membership) notation, it follows equivalence $F_1 \leq F_2 \Leftrightarrow m_{F_1}(x) \leq m_{F_2}(x)$, where $x \in \mathsf{R}$, and $m_F(.)$ denotes the membership function of fuzzy number F. In conclusion, there follows equivalence $m_{F_1}(x) \leq m_{F_2}(x) \Leftrightarrow F_1(\alpha) \leq F_2(\alpha)$, where $x \in \mathsf{R}$, $\alpha \in (0,1]$. In words, IN F_1 is smaller-than/equal-to IN F_2 if and only if either the membership function $m_{F_1}(x)$ is smaller-than/equal-to the membership function of $m_{F_2}(x)$ for all $x \in \mathsf{R}$, or (equivalently) interval $F_1(\alpha)$ is smaller-than/equal-to interval $F_2(\alpha)$ for all $\alpha \in (0,1]$.

The next proposition presents a metric in lattice $(\mathsf{F}, \leq)$ based on a positive valuation function $v_{\mathsf{R}} : \mathsf{R} \rightarrow \mathsf{R}^{\geq 0}$.

Proposition 2.1: Let F_1 and F_2 be INs in the lattice $(\mathsf{F}, \leq)$ of INs. Assuming that the following integral exists, a metric function $d_{\mathsf{F}} : \mathsf{F} \times \mathsf{F} \rightarrow \mathsf{R}^{\geq 0}$ is given by

$$d_{\mathsf{F}}(F_1, F_2) = \int_0^1 d_{\Delta}(F_1(\alpha), F_2(\alpha)) d\alpha \tag{1}$$

Moreover, a Minkowski metric $d_p : \mathsf{F}^N \times \mathsf{F}^N \rightarrow \mathsf{R}^{\geq 0}$ can be defined between two N-tuple INs $\mathbf{F}_1 = [F_{1,1}, ..., F_{1,N}]^T$ and $\mathbf{F}_2 = [F_{2,1}, ..., F_{2,N}]^T$ as

$$d_p(\mathbf{F}_1, \mathbf{F}_2) = [d_{\mathsf{F}}^p(F_{1,1}, F_{2,1}) + ... + d_{\mathsf{F}}^p(F_{1,N}, F_{2,N})]^{1/p} \tag{2}$$

Note that Minkowski metric $d_p(\mathbf{F}_1, \mathbf{F}_2)$ may involve a point $\mathbf{x} = [x_1, ..., x_N]^T \in \mathsf{R}^N$ such that an aforementioned point entry $x_i \in \mathsf{R}$ is represented by the *trivial* IN $x_i = \underset{\alpha\in(0,1]}{\cup}\{[x_i, x_i]\}$, $i = 1, ..., N$.

Space F is a *cone* for $F_1, F_2 \in \mathsf{F}$ and real numbers $\lambda_1, \lambda_2 \geq 0$ it follows $(\lambda_1 F_1 + \lambda_2 F_2) \in \mathsf{F}$.

C. A Hierarchy of Fuzzy Lattices

Consider the following definition.

Definition 4: Let $(\mathsf{L}, \leq)$ be a complete lattice with least and greatest elements O and I, respectively. An *inclusion measure* in $(\mathsf{L}, \leq)$ is a map $\sigma : \mathsf{L} \times \mathsf{L} \rightarrow [0,1]$, which satisfies the following conditions

IM0. $\sigma(x, O) = 0, \forall x \neq O$,

IM1. $\sigma(x, x) = 1, \forall x \in \mathsf{L}$,

IM2. $x \wedge y < x \Rightarrow \sigma(x, y) < 1$, and

IM3. $u \leq w \Rightarrow \sigma(x, u) \leq \sigma(x, w)$ (Consistency Property).

An inclusion measure in a lattice $(\mathsf{L}, \leq)$ fuzzifies the corresponding (crisp) lattice inclusion relation $(\leq)$. Therefore, notation $\sigma(x \leq y)$ may be used instead of $\sigma(x, y)$.

An alternative approach for crisp lattice fuzzification was proposed by Belohlavek [2] with emphasis on object-attribute fuzzy relations and fuzzy concept lattices without employment of positive valuation functions.

Lately, Hatzimichailidis, Kaburlasos [15] have proposed the following two inclusion measures in $(\tau_O(\mathsf{R}), \leq)$.

1) $\sigma_{\tau_O(\mathsf{R})}([a,b] \leq [c,d]; \vee) = \frac{v_\mathsf{R}(\theta_\mathsf{R}(c))+v_\mathsf{R}(d)}{v_\mathsf{R}(\theta_\mathsf{R}(a \wedge c))+v_\mathsf{R}(b \vee d)}$, and

2) $\sigma_{\tau_O(\mathsf{R})}([a,b] \leq [c,d]; \wedge) = \frac{v_\mathsf{R}(\theta_\mathsf{R}(a \vee c))+v_\mathsf{R}(b \wedge d)}{v_\mathsf{R}(\theta_\mathsf{R}(a))+v_\mathsf{R}(b)}$, if $a \vee c \leq b \wedge d$; otherwise, $\sigma_{\tau_O(\mathsf{R})}([a,b] \leq [c,d]; \wedge) = 0$,

where $\theta_\mathsf{R}(.)$ is a dual isomorphism and $v_\mathsf{R}(.)$ is a positive valuation function.

There follow two inclusion measures in the lattice $(\mathsf{F}, \leq)$ of INs, next.

1) $\sigma_\mathsf{F}(F_1 \leq F_2; \vee) = \int\limits_0^1 \sigma_{\tau_O(\mathsf{R})}(F_1(\alpha) \leq F_2(\alpha); \vee) d\alpha$, and

2) $\sigma_\mathsf{F}(F_1 \leq F_2; \wedge) = \int\limits_0^1 \sigma_{\tau_O(\mathsf{R})}(F_1(\alpha) \leq F_2(\alpha); \wedge) d\alpha$.

The latter inclusion measures will be employed below for calculating the similarity of two fuzzy sets, towards conditionally merging them, based on a novel *similarity measure* function defined next.

D. A Novel Similarity Measure

Various similarity measures have been presented by a number of authors including Adán, Adán [1], Torsello, Hidović-Rowe, Pelillo [53], Wu, Mendel [64], Zeng, Guo [68], Zhang, Zhang [69]. Setnes, Babu$\check{s}$ka, Kaymak, van Nauta Lemke [44] have reported similarity measure applications in fuzzy rule bases towards simplification. Pappis, Karacapilidis [39] have proposed a number of axioms for similarity measures regarding fuzzy sets, exclusively. In this work, we consider the more general definition by Kaburlasos, Moussiades, Vakali [25] presented next.

Definition 5: Similarity measure in a set U is a function $\mu : U \times U \to [0,1]$, which satisfies conditions: $(S1)$ $\mu(x,y) = 1 \Leftrightarrow x = y$, and $(S2)$ $\mu(x,y) = \mu(y,x)$.

Definition 5 retains the “common sense” essentials of similarity without “esoteric” redundancies.

We define a *similarity space* as a pair (U, μ) including a non-empty set U and a similarity measure function $\mu : U \times U \to [0,1]$. The following proposition introduces a novel similarity measure in a lattice based on an inclusion measure function.

Proposition 2.2: Let $(\mathsf{L}, \leq)$ be a lattice with an inclusion measure function $\sigma : \mathsf{L} \times \mathsf{L} \to \mathsf{R}$. Then, function $\mu_\wedge : \mathsf{L} \times \mathsf{L} \to [0,1]$ given by $\mu_\wedge(x,y) = \sigma(x \leq y) \wedge \sigma(y \leq x)$ is a similarity measure.

The proof of Proposition 2.2 is shown in the Appendix.

III. Novel structure identification

The term “structure identification” here originates from fuzzy system modeling as explained in the Appendix. The objective of structure identification is to partition the input data space into subspaces such that the output to an input $\mathbf{x} = [x_1, ..., x_N]^T \in \mathsf{R}^N$, within a subspace, is a linear combination of the N inputs $x_1, ..., x_N$. Some basic ideas are illustrated in the following.

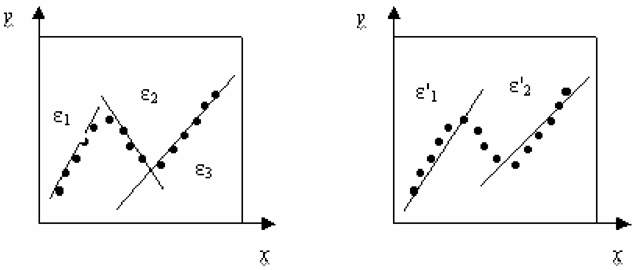


Fig. 1. Consequents (lines) of two different single-input-single-output piecewise-linear models. (a) This model partitions the input space effectively with a small error using three lines. (b) This model partitions the input space ineffectively with a large error using two lines.

The computation of a piecewise-linear model, characterized by both a small *mean square error (E)* and a minimum number of rules, is not trivial. For example, consider the data points shown together with the consequents (lines) of two different single-input-single-output piecewise-linear models in Fig. 1(a) and Fig. 1(b), respectively. On one hand, Fig. 1(a) demonstrates an effective partition (of the input space R) characterized by a small error. On the other hand, Fig. 1(b) demonstrates a non-effective partition characterized by a large error.

A structure identification method is proposed next for computing an effective partition based on (1) A novel SOM architecture inspired from KSOM, and (2) A novel structure identification algorithm, namely INSOM.

A. A Novel SOM Architecture

A cell $C_{i,j}$, where $i = 1, ..., I$ and $j = 1, ..., J$ in our proposed 2-dimensional SOM architecture grid stores both a N-dimensional IN $\mathbf{F}_{i,j} = [F_{i,j,1}, ..., F_{i,j,N}]^T$ and a $(N + 1)$-dimensional vector $\mathbf{c}_{i,j} = [c_{i,j,0}, c_{i,j,1}, ..., c_{i,j,N}]^T$. On one hand, IN $\mathbf{F}_{i,j} \in \mathsf{F}^N$ represents a population of data assigned to cell $C_{i,j}$ as detailed below. On the other hand, vector $\mathbf{c}_{i,j} \in \mathsf{R}^{N+1}$ stores the parameters of the following hyperplane.

$$p_{i,j}(\mathbf{x}) = c_{i,j,0} + c_{i,j,1}x_1 + c_{i,j,2}x_2, ..., c_{i,j,N}x_N \tag{3}$$

A cell is called *non-empty* if at least one datum is assigned to it. Vectors $\mathbf{c}_{i,j}$ are called *code vectors*. Structure identification is carried out by algorithm INSOM, next.

B. INSOM: A Novel Structure Identification Algorithm

Algorithm INSOM is applied in the aforementioned SOM architecture as detailed in the following.

After initialization (INSOM, lines 1-4) a loop of computations (INSOM, lines 6-23) repeats a user-defined number N_{epochs} of epochs. An epoch initially computes the neighborhood size $B_{p,q}(r)$ (INSOM, line 6) of the winner cell $C_{p,q}$ as follows.

$$B_{p,q}(r) = \frac{I+J}{4} \exp\left(-\frac{I+J}{N_{epochs}} r\right), \tag{4}$$

where $r \in \{1, ..., N_{epochs}\}$. Note that the following "decreasing" function $a(r)$, $r \in \{1, ...N_{epochs}\}$ (INSOM, line 6) equals the weight w_k in Eq. (13) of algorithm WRLS (in the Appendix).

$$a(r) = \exp\left(-\frac{r}{N_{epochs}}\right) = w_k \tag{5}$$

Of particular interest is function "FindTheWinner($\mathbf{x}_k, y_k$)" (INSOM, line 9). Recall that in order to compute an effective piecewise-linear model, the training data within a cluster have to be not only co-planar but also adjacent. Therefore, we proceed as follows.

First, we compute the distance $d_{i,j}(\mathbf{x}_k)$ of point $(\mathbf{x}_k, y_k) \in \mathsf{R}^N \times \mathsf{R}$ from hyperplane $p_{i,j}$ (Eq. (3)) as

$$d_{i,j}(\mathbf{x}_k) = \frac{|p_{i,j}(\mathbf{x}_k) - y_k|}{\sqrt{1 + \sum\limits_{t=1}^{N} c_{i,j}^2}}, \tag{6}$$

where $i = 1, ..., I$, $j = 1, ..., J$, $k = 1, ..., n$, and

Second, we compute the distance $D_{i,j}(\mathbf{F}_{i,j}, \mathbf{x}_k)$ of point $\mathbf{x}_k$ from the cluster of points within SOM cell $C_{i,j}$ using Eq. (2), where a cluster of points is represented by one IN per (data) dimension.

Then, a datum $(\mathbf{x}_k, y_k)$ is assigned to the (winner) cell $C_{p,q}$, which minimizes the following index $P_{i,j}$

$$P_{i,j} = b_d d_{i,j}(\mathbf{x}_k) + (1 - b_d) D_{i,j}(\mathbf{F}_{i,j}, \mathbf{x}_k), \tag{7}$$

where $i = 1, ..., I$, $j = 1, ..., J$, and $b_d \in [0, 1]$ is a user-defined *balancing factor for distance*.

We remark that a balanced consideration of both distances $d_{i,j}(\mathbf{x}_k)$ and $D_{i,j}(\mathbf{F}_{i,j}, \mathbf{x}_k)$, as in Eq. (7), has demonstrated an improved capacity for generalization in our computational experiments below.

Furthermore, we remark that an input datum $\mathbf{x}_k \in \mathsf{R}^N$ is assigned to the winner cell $C_{p,q}$ exclusively. However, input datum $\mathbf{x}_k$ is used to update the parameters of not only the winner cell's hyperplane but also of its neighbours' hyperplanes by algorithm WRLS (INSOM, line 16). In particular, algorithm WRLS (in the Appendix) is applied here locally, at each SOM grid cell, with $m = N$.

At the end of an epoch, if the total number of data assigned to a specific SOM cell is smaller than a user-defined threshold value $\ell_\theta = n/10$ then the aforementioned cell is "reset" (INSOM, line 21). That is, the cell's hyperplane parameter values are initialized to zero, moreover the corresponding INs are initialized to trivial values randomly; in addition, all data assigned to a "reset" cell are fed back to be assigned to different cells.

A N-dimensional IN is computed at each SOM cell by applying algorithm CALCIN once per data dimension for the data assigned to a cell (INSOM, line 22). Since neighboring cells in the SOM grid typically encode similar data, a post-processing simplification was carried out by assuming that two neighboring cells describe the same (input) subspace if their corresponding N-dimensional INs are "quite similar" to each other in a similarity measure $\mu_\wedge : \mathsf{F} \times \mathsf{F} \rightarrow [0, 1]$ sense as in Proposition 2.2 using either inclusion measure $\sigma_\mathsf{F}(.; \vee)$ or $\sigma_\mathsf{F}(.; \wedge)$. Only when

$\mu_\wedge(\mathbf{F}_1, \mathbf{F}_2)$ is above a user-defined threshold value μ_θ the aforementioned two cells are merged (INSOM, line 23). Note that preliminary work by Kaburlasos, Papadakis [22] has employed a user-defined threshold distance d_θ instead of μ_θ. An advantage of using μ_θ over d_θ is that the size of μ_θ is *normalized* (in the unit interval $[0, 1]$), whereas the size of d_θ is application-dependent.

C. INSOM Algorithm Complexity

We will consider only those routines of **Algorithm 1** (INSOM) with "substantial" complexity. The "outer" loop repeats N_{epochs} times, furthermore the "inner" loop repeats n times. Within the aforementioned "inner" loop, the complexity of routine "FindTheWinner($\mathbf{x}_k, y_k$)" equals $\mathsf{O}(I*J*m)$; moreover, routine "WRLS(i, j,w_k,$\mathbf{x}_k, y_k$)" repeats $I * J$ times, each time with complexity $\mathsf{O}(m * n)$. Therefore, the complexity of the "inner" loop equals $\mathsf{O}(I*J*m*n^2)$. After the "inner" loop, there follow, first, routine "ComputeINs()" with complexity $\mathsf{O}(I*J*log(n))$ and, second, routine "MergeSimilarCells(μ_θ)" with complexity $\mathsf{O}(I * J * m * (L + log(n)))$, where L is the total number of levels considered for a IN, e.g. IN $F = \overset{L}{\underset{i=1}{\cup}} \{[a_{\alpha_i}, b_{\alpha_i}]\}$, $\alpha_i \in (0, 1]$. Making the reasonable assumption $n^2 >> L$, it follows that the complexity of **Algorithm 1** (INSOM) equals $\mathsf{O}(I*J*m*n^2*N_{epochs})$. In conclusion, **Algorithm 1** (INSOM) computes an "initial" model, which is fine-tuned as detailed next.

Algorithm 1 INSOM: A Novel Structure Identification Algorithm

```
1:  $I \leftarrow$ Number of rows in a SOM grid/map
2:  $J \leftarrow$ Number of columns in a SOM grid/map
3:  $\mu_\theta \leftarrow 0.67$, $\ell_\theta \leftarrow n/10$ //user-defined parameters $\mu_\theta$ and $\ell_\theta$
4:  createANDinitializeMap($I, J$)
5:  for $r = 1$ to $N_{epochs}$ do //for each epoch do
6:     Calculate $B_{p,q}(r), a(r)$ //$B_{p,q}(r)$ is a neighborhood; $a(r)$ is a weight coefficient
7:     $w_k \leftarrow a(r)$
8:     for $k = 1$ to $n$ do //for each input datum $(\mathbf{x}_k, y_k)$ do
9:        FindTheWinner($\mathbf{x}_k, y_k$)
10:       $p \leftarrow$ the winner row
11:       $q \leftarrow$ the winner column
12:       Assign($\mathbf{x}_k, y_k$,$p$,$q$) //assign input datum $(\mathbf{x}_k, y_k)$ to the winner cell $C_{p,q}$
13:       for $i = 1$ to $I$ do //for each row in the SOM grid/map do
14:          for $j = 1$ to $J$ do //for each column in the SOM grid/map do
15:             if $C_{i,j} \in B_{p,q}(r)$ then //update hyperplane parameters for a cell in the neighborhood $B_{p,q}(r)$
16:                WRLS($i, j$,$w_k$,$\mathbf{x}_k, y_k$)
17:             end if
18:          end for//for $j$
19:       end for//for $i$
20:    end for//for $k$
21:    ResetCellsConditionally($\ell_\theta$)
22:    ComputeINs()
23:    MergeSimilarCells($\mu_\theta$)
24: end for//for $r$
```

IV. Novel Parameter Identification

The previous section has induced an "initial" piecewise-linear model from a series $(\mathbf{x}_k, y_k) \in \mathsf{R}^N \times \mathsf{R}$, $k = 1, 2, ..., n$ of training data. Recall that a IN, in the previous section, was interpreted *probabilistically* (*statistically*). Nevertheless, a IN is interpreted *possibilistically* in this section. More specifically, here we assume that a non-empty cell in our proposed SOM represents a fuzzy rule such that the N INs in $\mathbf{F}_{i,j}$, stored in cell $C_{i,j}$, define a fuzzy rule antecedent, whereas the $N+1$ hyperplane parameters, stored in $\mathbf{c}_{i,j}$, define the corresponding rule consequent, in a TSK-model-sense. The aforementioned model includes K rules in the form of Eq. (14) (in the Appendix) such that each rule is locally optimum. The objective in this section is to optimize our model, globally.

A. Hyperplane Parameter Optimization

Based on Eq. (14) the output of a piecewise-linear model in Eq. (15) (in the Appendix) equals

$$\widehat{y}(\mathbf{x}_k) = c_0 + \sum_{i=1}^{K} [(c_{i,0})(\sigma_i) + \sum_{j=1}^{N} (c_{i,j})(\sigma_i x_{k,j})] \tag{8}$$

where $\mathbf{x}_k = [x_{k,1}, ..., x_{k,N}]^T$, furthermore the σ_i's are functions of the (known) INs in the K rules. In conclusion, a globally optimum set of hyperplanes can be computed by algorithm WRLS (in the Appendix) with $m = K(N+1)$. Further improvement was sought by optimal parameter estimation of parametrically described INs as detailed next.

B. IN Parameter Optimization

Recall that a general IN has a non-parametric membership function. In the context of this work, based on the theory presented in section II-B, we replaced a IN $F_{i,j} \in \mathsf{F}$ by another IN $F'_{i,j} = a_{i,j}F_{i,j} + b_{i,j}$, where $a_{i,j} \in (0, 3]$ is a *scaling* parameter and $b_{i,j} \in [-1, 1]$ is a *translation* parameter, $i = 1, ..., K, j = 1, ..., N$. It follows that IN $F'_{i,j}$ is in the neighborhood (in a metric d_{F} sense) of IN $F_{i,j}$. The task now is to compute "optimal" INs $F'_{i,j}$, in a mean square error sense of Eq. (16) (in the Appendix), from INs $F_{i,j}$ by optimal parameter $a_{i,j}$, $b_{i,j}$ estimation.

Genetic algorithms are established optimization tools as explained by Chakraborty [6], Cordón, Gomide, Herrera, Hoffmann, Magdalena [7], Papadakis, Theocharis [37]. Hence, optimization was pursued here by genetic algorithms, where the phenotype of an "individual" consisted of specific values of the aforementioned parameters $a_{i,j}, b_{i,j}$. We remark that an "individual" encodes all rules' antecedents (IF part) of a piecewise-linear model.

There was a total number of $2 \times N \times K$ parameters binary-encoded in the chromosome of an individual. We included 25 individuals per generation. The genetic algorithm was enhanced by the *microgenetic hill-climbing* operator introduced by Kazarlis, Papadakis, Theocharis, Petridis [26] as well as by both operators *ASER* and *RWSCS* of Papadakis, Theocharis [36]. In addition, both *elitism* and *adaptive crossover/mutation* rates were implemented.

To avoid overtraining, the fitness value of an individual was computed as follows. The data set was divided in three subsets including (1) a training set (2) a validation set, and (3) a testing set. The consequent parameters of the fuzzy model were calculated from the training and validation sets. Let E_{trn} and E_{val} be the mean square errors on the training and validation sets, respectively. The fitness (Q) of an individual was calculated as follows.

$$Q = b_e E_{trn} + (1 - b_e) E_{val} \tag{9}$$

where $b_e \in [0, 1]$ is a user-defined *balancing factor for error*. The genetic algorithm was left to evolve until no improvement was observed in the fitness (Q) of the best "individual" for 50 generations in a row. Then, the testing (data) set was applied once. Finally, the mean square error E_{tst} on the testing set was recorded.

C. Comparative Algorithm Discussion

The TSK-model-based FIS scheme proposed in this work has followed, in general, the mainstream interpretability guidelines for FIS design presented by Guillaume [13]. For instance, care was taken to induce a small set of readable rules from the data, FIS optimization was pursued, etc.

Recall that FIS structure identification can be pursued by (1) *grid type* partition, (2) *guillotine cuts*, or (3) *scatter type* partition as explained by Cordón, Gomide, Herrera, Hoffmann, Magdalena [7], Papadakis, Theocharis [37]. On one hand, both grid type partition and guillotine cuts suffer from the *curse of dimensionality* problem, moreover they often result in redundant rules in large numbers. On the other hand, scatter type partition typically induces a small number of rules, furthermore it does not suffer from the *curse of dimensionality* problem for any number of inputs. Therefore, our INSOM architecture employed a *scatter type* partition of the input space.

The INSOM architecture was employed towards computing an "initial" piecewise-linear model. INs were computed per data dimension by algorithm CALCIN. Therefore, initially, INs were interpreted probabilistically (statistically). Next, the aforementioned "initial" model was employed as a TSK model. Therefore, ultimately, INs were interpreted possibilistically. In other words, a statistically computed "initial" model was ultimately interpreted linguistically. Finally, parameter optimization was pursued using standard FIS optimization techniques towards an improved practical performance.

In the interest of simplicity, this work considered only a single system output, i.e. $M = 1$, in a function $y : \mathsf{R}^N \rightarrow \mathsf{R}^M$ approximation problem. Multiple inputs, i.e. $M > 1$, can be accommodated by considering M single-output models. Furthermore, our intention here was neither an improvement of the TSK paradigm nor an improvement of the KSOM paradigm itself. Rather, inspired from both aforementioned paradigms, we sought a synergy towards an *efficient* as well as an *effective* piecewise-linear approximation of nonlinear models. Where, by *effective* we mean a good capacity for generalization, whereas by *efficient* we mean 1) fast computation, and 2) small computer memory storage requirements. Our proposed synergy has demonstrated both effectiveness and efficiency in the computational experiments presented in the following section.

V. Computational Experiments

The performance of our proposed piecewise-linear modeling method is demonstrated in this section in four experiments including (1) a single-input-single-output, non-linear, static system, (2) a two-input-single-output, non-linear, static system, (3) Fisher's Iris benchmark classification problem, and (4) a three-input-single-output, non-linear, dynamic system. In the interest of simplicity, positive valuation function $v_{\mathsf{R}}(x) = x$ was employed in every (data) dimension. Moreover, in all cases, both input and output data were normalized in the interval $[0, 1]$ by a straightforward linear transformation; at the end of all computations, the output data were restored in their original domain in order to enable meaningful comparisons. Only the first experiment has been published in a preliminary work by Kaburlasos, Papadakis [22]; whereas, the remaining three experiments are presented here for the first time.

A. *Single-Input-Single-Output, Non-linear, Static System*

Consider the system described by the following equation.

$$y = sin(10x) \tag{10}$$

where $x \in [0, 1]$. Forty input/output data pairs $(x_k, y_k) \in \mathsf{R} \times \mathsf{R}$, $k = 1, ..., 40$ were randomly (uniformly) generated. The scatter plot of the generated input/output data points is shown in Fig. 2(a). Following the practice of different authors, we employed the same data set for both training and testing. No validation set was employed here, i.e. $b_e = 1$ in Eq. (9).

A 4×4 SOM grid was used to compute a piecewise-linear model as described above. The structure identification algorithm was applied for $N_{epochs} = 100$ epochs resulting in five non-empty cells. Recall that a non-empty cell represents a rule. The IN/antecedent and the hyperplane/consequent (a line in this case) in each cell are shown, respectively, in Fig. 2(b) and Fig. 2(a). A visual inspection of Fig. 2 clearly shows that the proposed method partitions the input space well, in this simple example.

B. *Two-Input-Single-Output, Non-linear, Static System*

Consider the following system, also used by Kim, Park, Ji, Park [27], Papadakis, Theocharis [37], Sugeno, Yasukawa [49].

$$y = f(x_1, x_2) = (1 + x_1^{-2} + x_2^{-1.5})^2 \tag{11}$$

where $1 \leq x_1, x_2 \leq 5$. The surface in Fig. 3(a) plots the graph of Eq. (11). Fifty input vectors $\mathbf{x}_k = [x_{k,1}, x_{k,2}]$, $k = 1, 2, ..., 50$ were generated randomly in the square $[1, 5] \times [1, 5]$. For each vector $\mathbf{x}_k$ the corresponding output y_k was computed. Following the practice of different authors, we employed the same data set for both training and testing. No validation set was employed here, i.e. $b_e = 1$ in Eq. (9).

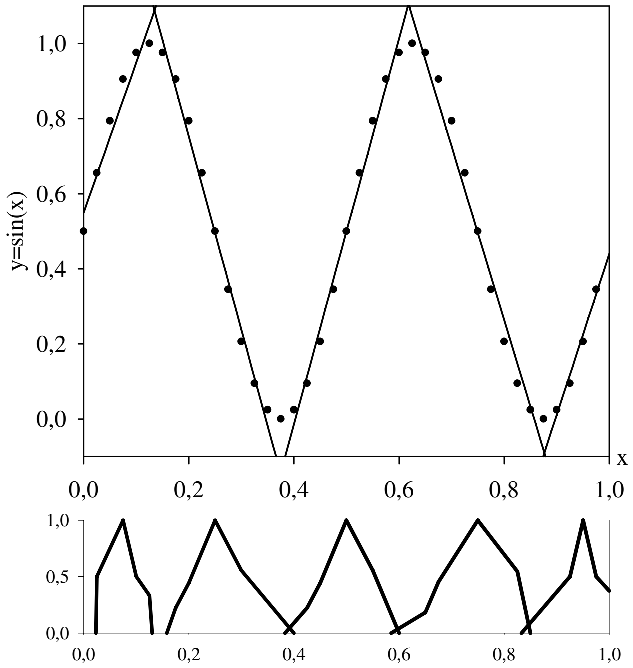


Fig. 2. (a) Scatter plot of model $y = sin(10x)$ including 40 randomly generated input/output data points. The five lines correspond, respectively, to the consequents of five piecewise-linear (fuzzy) model rules. (b) Five INs, each one corresponds to the antecedent of a rule. The corresponding rule consequent (line) is shown above a IN.

TABLE I

TWO-INPUT-SINGLE-OUTPUT SYSTEM: COMPARISON OF OUR PROPOSED METHOD WITH ALTERNATIVE METHODS

Method	Rules	Testing Error
Sugeno, Yasukawa [49]	6	0.0790
Kim, Park, Ji, Park [27]	3	0.0190
Papadakis, Theocharis [37]	4	0.0095
Our proposed method	4	0.0086

For structure identification a 4×4 grid of cells was employed for $N_{epochs} = 100$ epochs. The rule-base of the induced “initial” piecewise-linear model is shown in Fig. 4 including four rules. The four planes described analytically in the consequents of the (four) rules in Fig. 4 are shown in Fig. 3(b). Fig. 3(c) displays the surfaces in Fig. 3(a) and Fig. 3(b) superimposed.

The mean square error E_{tst} (on the testing data set) of our “initial” piecewise-linear model was $E_{tst} \approx 0.70$. After parameter identification, as described above, the testing error reduced down to $E_{tst} \approx 0.0086$. A comparison of our proposed method with alternative methods is summarized in Table I. We point out that all the methods in Table I, have used the same data.

The total computing time for our proposed method was around 22 sec on a Pentium IV 2.5 GHz computer. In stark contrast we remark that the genetic-based structure identification method presented by Papadakis, Theocharis [37] required around 50 min on the same data set.

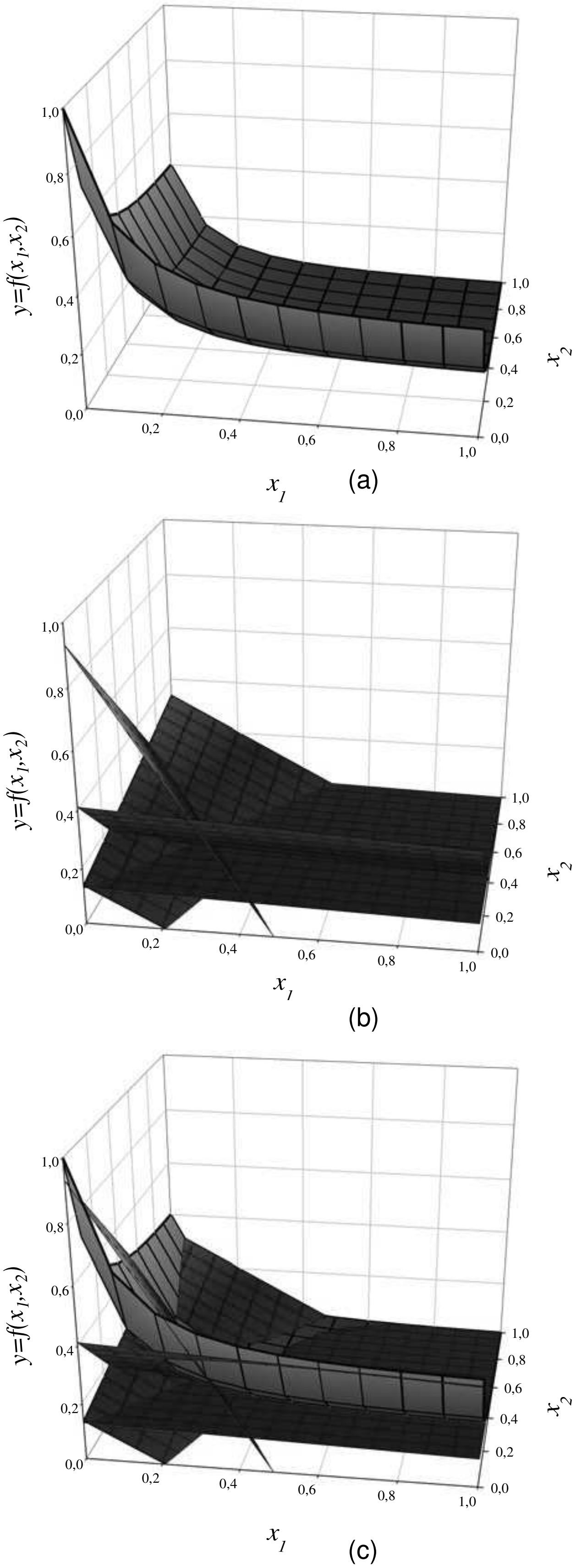


Fig. 3. (a) The input-output surface of the two-input-single-output, non-linear, static system example. (b) Four planes represent the consequents of four rules induced by our "initial" piecewise-linear model. (c) The surfaces in figures (a) and (b) are displayed superimposed.

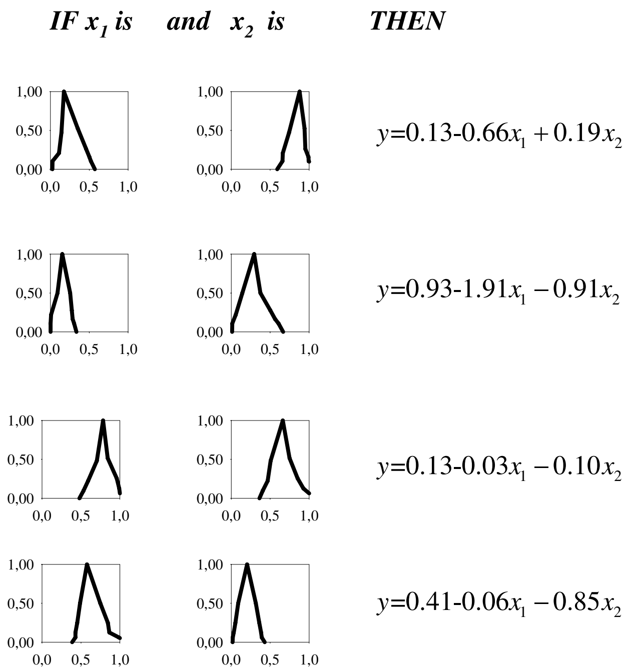


Fig. 4. The rule base of our "initial" piecewise-linear model in the two-input-single-output, non-linear, static system example.

C. Fisher's Iris Benchmark Classification Problem

The Fisher Iris benchmark data set was downloaded from the UCI machine learning repository [35]. There are fifty 4-dimensional vectors per class in three classes. Following a random data permutation we employed the first 75 data vectors "half" for training and "half" for validation, and the remaining 75 data vectors for testing.

This classification problem was dealt with by our proposed piecewise-linear modeling method. A 4×4 grid of cells was employed for $N_{epochs} = 100$ epochs resulting in, in one experiment, an "initial" model with the three rules shown in Fig. 5. Ten experiments were carried out using a different random data permutation per experiment. The sizes of the data sets used for training /validation /testing were as described above.

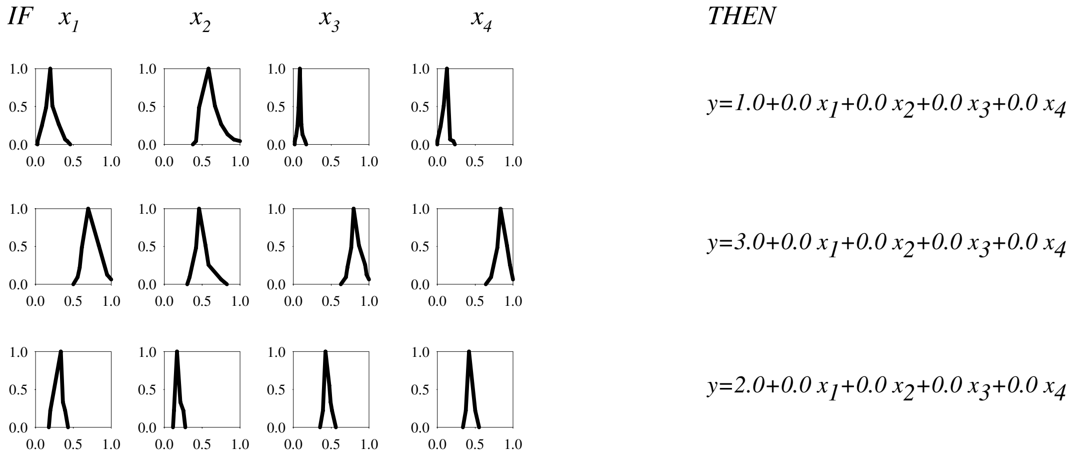
The experimental results in ten experiments are summarized in Table II. For comparison, the average training and testing classification accuracies in our ten experiments vs. the corresponding results by alternative classification methods are summarized in Table III. We point out that other authors do not, typically, specify the sizes of the training/testing data sets they use.

D. Three-Input-Single-Output, Non-linear, Dynamic System

Consider the dynamic system described by the following difference equation.

$$y_k = \frac{y_{k-1}y_{k-2}(y_{k-1}+2.5)}{1+y_{k-1}^2+y_{k-2}^2} + u_k \tag{12}$$

We remark that Eq. (12) was employed as a benchmark in a number of previous works by Farag, Quintana, Lambert-Torres [11], Narendra, Parthasarathy [34], Sugeno, Tanaka [48]. Note that the output (y_k) depends on both



(a) Rule antecedent.

(b) Rule consequent.

Fig. 5. The rule base of an "initial" piecewise-linear model in the Iris benchmark classification problem.

1) the previous output values y_{k-1} and y_{k-2}, and 2) the excitation signal u_k.

Using initial values $y_0 = y_1 = 0$ and random values (uniformly) for the excitation signal u_k in the interval $[-2, 2]$, we generated five hundred data vectors $[y_{k-1}, y_{k-2}, u_k, y_k]$, $k = 1, 2, ...500$. Half (250) of the aforementioned vectors were used for training and the remaining (250) vectors were used for validation. We considered a piecewise-linear model with three inputs $x_1 = y_{k-1}$, $x_2 = y_{k-2}$, $x_3 = u_k$ and one output $y = y_k$ according to the *series-parallel* approach described by Juang [16], Papadakis, Tzionas, Kaburlasos, Theocharis [38].

The application of our proposed structure identification method induced an "initial" piecewise-linear model with the three rules shown in Fig. 6. It is remarkable that the INs along the u_k axis are quite similar to one another, in particular they are "almost triangular" covering the whole universe of discourse – Note also that the coefficients of u_k in the consequent of every rule are (approximately) equal to one another (≈ 0.5). Therefore, we assumed that input $x_3 = u_k$ is redundant, and it can be omitted as discussed in the following section.

After parameter identification, our proposed modeling method achieved a training error $E_{trn} \approx 0.024$. The capacity for generalization of the final model was tested using five hundred data vectors $[y_{k-1}, y_{k-2}, u_k, y_k]$, $k = 1, 2, ...500$ generated using a sinusoid excitation signal $u_k = \sin(\frac{2k\pi}{25})$. The testing error in this case was $E_{tst} \approx 0.0159$. We remark that the smaller testing error 0.0159 (compared to 0.024 training error) is attributed to the sinusoid excitation signal used for testing. In particular, it appears that the outputs produced by our piecewise-linear model in response to a sinusoid excitation are more predictable than the outputs produced in response to a random excitation due to the input-output data law, which characterizes a signusoid. Furthermore, note that our experimental results did not change significantly when system input $x_3 = u_k$ was omitted, thus confirming our hypothesis regarding the redundancy of input $x_3 = u_k$ in this example.

Table IV shows the performance of our proposed method, comparatively with alternative methods from the literature. Note that all the methods in Table IV have used similar training /validation /testing data sets. In conclusion, our proposed method produced both a small testing error $E_{tst} \approx 0.015$ and a small number of (three) rules.

TABLE II

Experimental results for the Iris classification benchmark problem

# Exper	Classification Accuracy %		# Misclassified Data	
	Training	Testing	Training	Testing
1	97.33	96.00	2	3
2	100.00	97.33	0	2
3	97.33	100.00	2	0
4	97.33	98.66	2	1
5	97.33	98.66	2	1
6	98.66	98.66	1	1
7	100.00	97.33	0	2
8	96.00	100.00	4	0
9	100.00	96.00	0	3
10	96.00	100.00	3	0
Average:	98.00	98.26	1.60	1.30
Std. Dev:	1.57	1.55	1.35	1.16
Min:	96.00	96.00	0	0
Max:	100.00	100.00	4	3

TABLE III

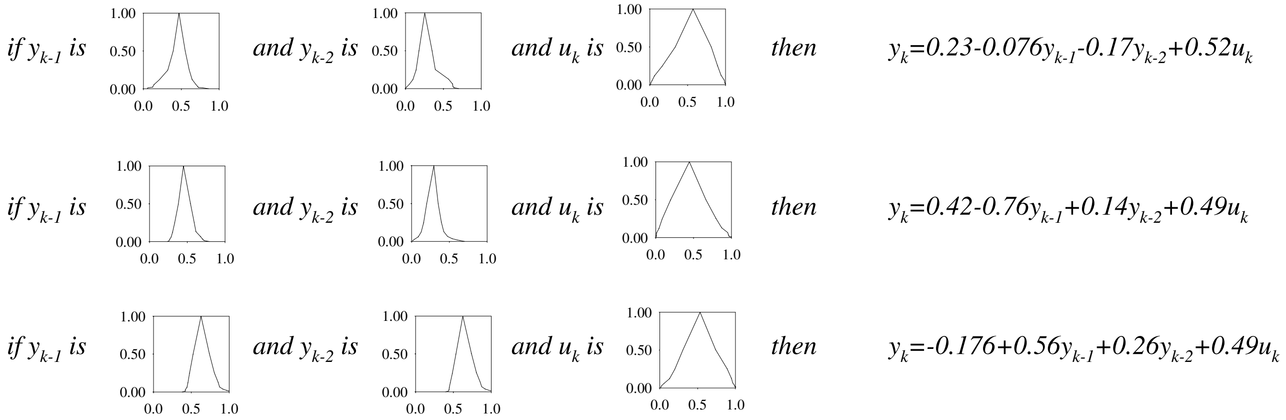
Iris Benchmark: Comparison of our proposed method with alternative methods

Method	Classification Accuracy %	
	Training	Testing
Lee, Chen, Chen, Jou [32]	96.70	97.12
Wang, Lee [60]	97.20	97.47
Lee, Chen, Jiang [31]	96.30	Not Available
Simpson [45]	97.30	Not Available
Our proposed method	98.00	98.26

TABLE IV

Three-Input-Single-Output System: Comparison of our proposed method with alternative methods

Method	Rules	Trn Error	Testing Error
Wang, De Baets, Kerre [62]	8	0.618	0.203
Sugeno, Tanaka [48]	12	0.507	0.244
Farag, Quintana, Lambert-Torres [11]	75	0.037	0.040
Papadakis, Theocharis [37]	3	0.018	0.025
Our proposed method	3	0.024	0.015



(a) Rule antecedent.

(b) Rule consequent.

Fig. 6. The rule base of our "initial" piecewise-linear model in the three-input-single-output, non-linear, dynamic system example.

VI. Discussion and Conclusion

This work has proposed a novel synergy of TSK type FISs with a SOM towards piecewise-linear approximation of nonlinear models based on INs. A IN was presented as a mathematical object interpreted as either a probability- or a possibility- distribution. Hence, we moved freely between different paradigms including a probabilistic (statistical) and a possibilistic (fuzzy) one towards an improved performance in practice. Note that here we employed a novel combination of an "order-based" employment of LT with a "lattice-based" employment of LT including the computation of a (novel) similarity measure function $\mu_\wedge(.,.)$, etc.

The employment of (fairly small) SOM grids here did not aim at data visualization. Rather, our objective here was practical efficiency as well as effectiveness in piecewise-linear modeling applications. The end result has justified our practices since they compared favorably with the results by alternative methods from the literature regarding the (small) number of rules, the (high) speed of learning, and the capacity for (comparatively accurate) generalization in three new experiments regarding benchmark problems.

Recall, on one hand, that based on experimental evidence, Kaburlasos, Papadakis [23] have considered the following *null hypothesis H0*: A system's input variable may be omitted when an underlying (sigmoid) positive valuation function remains practically constant over the aforementioned input variable's domain. On the other hand, this work has presented experimental evidence that an input variable may be omitted when, given a linear underlying positive valuation function, the induced INs (in different rules) are quite similar to one another, e.g. input variable $x_3 = u_k$ in example V-D may be omitted. Therefore, we propose here an "enhanced" *null hypothesis H0* as follows: An input variable may be omitted when the INs induced (in different rules) for the aforementioned input variable are "quite" near to one another, in a metric d_{F} sense. The latter hypothesis remains to be tested statistically in a future work since the problem of input variable selection is significant in practice.

Another promising potential of our techniques is their inherent capacity to deal, in principle, with the uncertainty as described by Wang [61] by treating, in particular, granular data such as intervals and (probability/possibility) distributions. Note that the need to accommodate the aforementioned data has been acknowledged in various applications by different authors including Boukezzoula, Foulloy, Galichet [5], Pedrycz, Bezdek, Hathaway, Rogers [41], etc. Furthermore, our techniques might be especially handy for (fuzzy) rule interpolation along the lines of work by Koczy and colleagues, e.g. Koczy, Hirota [28], [29], Wong, Tikk, Gedeon, Koczy [63], Yam, Koczy [66].

APPENDIX

This Appendix, in the first place, presents the proof of Proposition 2.2. Then, it summarizes the WRLS algorithm as well as the operation of a TSK model employed in this work.

Proposition 2.2 Let $(\mathsf{L}, \leq)$ be a lattice with an inclusion measure function $\sigma : \mathsf{L} \times \mathsf{L} \rightarrow \mathsf{R}$. Then, function $\mu_\wedge : \mathsf{L} \times \mathsf{L} \rightarrow [0, 1]$ given by $\mu_\wedge(x, y) = \sigma(x \leq y) \wedge \sigma(y \leq x)$ is a similarity measure.

Proof of Proposition 2.2. Function $\mu_\wedge(., .)$ satisfies both conditions ($S1$) and ($S2$) of Definition 5 as shown in the following – We will employ the following equivalence "$x \leq y \Leftrightarrow \sigma(x \leq y) = 1$" shown by Kaburlasos, Athanasiadis, Mitkas (see in [24], Proposition 3).

($S1$) In one direction, let $\mu_\wedge(x, y) = 1$.
Then, $\sigma(x \leq y) \wedge \sigma(y \leq x) = 1 \Rightarrow \sigma(x \leq y) = 1 = \sigma(y \leq x) \Rightarrow x \leq y$ and $y \leq x \Rightarrow x = y$.
In the other direction, let $x = y$.
Then, $\mu_\wedge(x, y) = \sigma(x \leq y) \wedge \sigma(y \leq x) = 1 \wedge 1 = 1$.
($S2$) $\mu_\wedge(x, y) = \sigma(x \leq y) \wedge \sigma(y \leq x) = \sigma(y \leq x) \wedge \sigma(x \leq y) = \mu_\wedge(y, x)$.

Therefore, function $\mu_\wedge : \mathsf{L} \times \mathsf{L} \rightarrow [0, 1]$ is a similarity measure.

A. WRLS Algorithm for Incremental Learning

Consider a series of data vectors $[x_{k,1}, ..., x_{k,m}, y_k]^T \in \mathsf{R}^m \times \mathsf{R}$, $k = 1, ..., n$. The WRLS (Weighted Recursive Least Squares) algorithm computes incrementally the parameters $\mathbf{c}_{k+1}$ of a hyperplane in R^{m+1}, *optimally fitted*, in a least square error sense, to the aforementioned data. The corresponding equations are displayed next.

$$\begin{aligned} &\mathbf{c}_{k+1} = \mathbf{c}_k + \left(y_{k+1} - \mathbf{x}_{k+1}^T \cdot \mathbf{c}_k\right) \mathbf{k}_k \\ &\mathbf{k}_k = \frac{\mathbf{S}_k \mathbf{x}_{k+1}}{\frac{1}{w_k} + \mathbf{x}_{k+1}^T \mathbf{S}_k \mathbf{x}_{k+1}} \\ &\mathbf{S}_{k+1} = \left(\mathbf{I} - \mathbf{k}_k \mathbf{x}_{k+1}^T\right) \mathbf{S}_k \\ &k = 1, 2, ..., n. \end{aligned} \tag{13}$$

The WRLS equations above are initialized at $k=0$ with $\mathbf{c}_0 = \mathbf{0}$ and $\mathbf{S}_0 = a\mathbf{I}$, where $\mathbf{I}$ is the identity matrix of dimension $m+1$ and $a \in \mathsf{R}$ is typically large, e.g. $a = 1000$. A "weight of significance" $w_k \in (0,1]$ may be attached to a data pair $(\mathbf{x}_k, y_k)$. In particular, for $w_k = 0$ the corresponding datum $(\mathbf{x}_k, y_k)$ is ignored. In this work we used $w_k = constant$, i.e. all data pairs $(\mathbf{x}_k, y_k), k = 1, ..., n$ are equally significant. Vector $\mathbf{c}_k = [c_{k,0}, c_{k,1}, ..., c_{k,m}]^T$ includes the *optimum* hyperplane parameters in a step. Moreover, vector $\mathbf{x}_{k+1} = [1, x_{k+1,1}, ..., x_{k+1,m}]^T$ is used to incrementally modify parameter vector $\mathbf{c}_{k+1}$. The value of parameter m depends on the application.

B. A TSK model

The TSK model version we employed in our computational experiments includes a set of K (integer number of) $IF - THEN$ rules. The antecedent (IF part) of a rule corresponds to a fuzzy subspace of the input space, whereas the corresponding consequent (THEN part) is a linear combination of N input values $x_j,\ j = 1, ..., N$. For instance, rule R_k is shown, next.

$$\begin{aligned} &R_k : IF\ x_1\ is\ A_{k,1}\ and\ x_2\ is\ A_{k,2}\ and \cdots and\ x_N\ is\ A_{k,N} \\ &THEN\ y_k = c_{k,0} + \sum_{j=1}^{N} c_{k,j} x_j = \mathbf{c}_k^T \cdot \mathbf{x} \end{aligned} \tag{14}$$

where $\mathbf{c}_k = [c_{k,0}, c_{k,1}, ..., c_{k,N}]^T \in \mathsf{R}^{N+1}$, $\mathbf{x} = [1, x_1, ..., x_N]^T \in \mathsf{R}^{N+1}$; furthermore, $A_{k,j}$ are fuzzy numbers, where $k = 1, ..., K$ and $j = 1, ..., N$.

A fuzzy rule is interpreted linguistically as: IF the system inputs $x_1, ..., x_N$ are within a fuzzy subspace, which is specified by the N-tuple $[A_{k,1}, ..., A_{k,N}]$, $THEN$ the system output y_k is on a hyperplane in R^{N+1} defined by the consequent part. When more than one rules are activated then the output is computed as follows.

Let $m_{A_{k,j}}(.)$ be the membership function of fuzzy number $A_{k,j}$. Assuming the "centroid" defuzzification technique, the output to an input vector $\mathbf{x} = [1, x_1, ..., x_N]^T$ is computed as follows.

$$\widehat{y}(\mathbf{x}) = \frac{\sum\limits_{k=1}^{K} g_k y_k}{\sum\limits_{k=1}^{K} g_k} = \sum_{k=1}^{K} \sigma_k f_k \tag{15}$$

where $g_k = \prod\limits_{j=1}^{N} m_{A_{k,j}}(x_j)$, moreover $\sigma_k = g_k / \sum\limits_{k=1}^{K} g_k$. We point out that output $\widehat{y}(\mathbf{x})$ may be used as an estimate of the system's "true" (sampled) output y in response to input vector $\mathbf{x}$.

Given a number n of input/output samples $(\mathbf{x}_i, y_i) \in \mathsf{R}^{N+1} \times \mathsf{R},\ i = 1, 2, ..., n$, the *predictive accuracy* of the model is measured by the following *mean square error (E)*.

$$E = \frac{1}{n} \cdot \sum_{i=1}^{n} [\widehat{y}(\mathbf{x}_i) - y_i]^2 \tag{16}$$

References

[1] A. Adán, M. Adán, A flexible similarity measure for 3d shapes recognition, IEEE Transactions on Pattern Analysis and Machine Intelligence 26 (11) (2004) 1507–1520.

[2] R. Belohlavek, Fuzzy Relational Systems: Foundations and Principles, Springer, Heidelberg, Germany, 2002.

[3] G. Birkhoff, Lattice Theory, Colloquium Publications 25, American Mathematical Society, Providence, RI, 1967.

[4] I. Bloch, O. Colliot, R.M. Cesar Jr., On the ternary spatial relation "between", IEEE Transactions on Systems, Man & Cybernetics - Part B 36 (2) (2006) 312–327.

[5] R. Boukezzoula, L. Foulloy, S. Galichet, Inverse controller design for fuzzy interval systems, IEEE Transactions on Fuzzy Systems 14 (1) (2006) 111–124.

[6] U.K. Chakraborty, Editorial: genetic and evolutionary computing, Information Sciences 178 (23) (2008) 4419–4420.

[7] O. Cordón, F. Gomide, F. Herrera, F. Hoffmann, L. Magdalena, Ten years of genetic fuzzy systems: current framework and new trends, Fuzzy Sets and Systems 141 (1) (2004) 5–31.

[8] C. Cornelis, G. Deschrijver, E.E. Kerre, Advances and challenges in interval-valued fuzzy logic, Fuzzy Sets and Systems 157 (5) (2006) 622–627.

[9] A.M. da Silva, P. Sussner, A brief review and comparison of feedforward morphological neural networks with applications to classification, In: V. Kurková, R. Neruda, J. Koutník (Eds.), Proc. Int. Conf. Artificial Neural Networks (ICANN) 2008, Part II, Lectures Notes in Computer Science (LNCS) 5164, Springer-Verlag, Heidelberg, Germany, 2008, pp. 783–792.

[10] M.J. Er, Z. Li, H. Cai, Q. Chen, Adaptive noise cancellation using enhanced dynamic fuzzy neural networks, IEEE Transactions on Fuzzy Systems 13 (3) (2005) 331–342.

[11] W.A. Farag, V.H. Quintana, G. Lambert-Torres, A genetic-based neuro-fuzzy approach for modeling and control of dynamical systems, IEEE Transactions on Neural Networks 9 (5) (1998) 756–767.

[12] B. Ganter, R. Wille, Formal Concept Analysis, Springer, Heidelberg, Germany, 1999.

[13] S. Guillaume, Designing fuzzy inference systems from data: an interpretability-oriented review, IEEE Transactions on Fuzzy Systems 9 (3) (2001) 426–443.

[14] M. Graña, I. Villaverde, J.O. Maldonado, C. Hernandez, Two lattice computing approaches for the unsupervised segmentation of hyperspectral images, Neurocomputing 72 (10-12) (2009) 2111–2120.

[15] A.G. Hatzimichailidis, V.G. Kaburlasos, A novel fuzzy implication stemming from a fuzzy lattice inclusion measure, In: V.G. Kaburlasos, U. Priss, M. Graña (Eds.) Proc. Lattice-Based Modeling (LBM 2008) Workshop, in conjunction with The Sixth International Conference on Concept Lattices and their Applications (CLA 2008), Palacký University, Olomouc, Czech Republic, 2008, pp. 59–66.

[16] C.F. Juang, A TSK-type recurrent fuzzy network for dynamic systems processing by neural network and genetic algorithms, IEEE Transactions on Fuzzy Systems 10 (2) (2002) 155–170.

[17] V.G. Kaburlasos, FINs: lattice theoretic tools for improving prediction of sugar production from populations of measurements, IEEE Transactions on Systems, Man & Cybernetics - Part B 34 (2) (2004) 1017–1030.

[18] V.G. Kaburlasos, Towards a Unified Modeling and Knowledge-Representation Based on Lattice Theory, ser. Studies in Computational Intelligence 27, Springer, Heidelberg, Germany, 2006.

[19] V.G. Kaburlasos, A. Kehagias, Novel fuzzy inference system (FIS) analysis and design based on lattice theory. part I: working principles, International Journal of General Systems 35 (1) (2006) 45–67.

[20] V.G. Kaburlasos, A. Kehagias, Novel fuzzy inference system (FIS) analysis and design based on lattice theory, IEEE Transactions on Fuzzy Systems 15 (2) (2007) 243–260.

[21] V.G. Kaburlasos, S.E. Papadakis, Granular self-organizing map (grSOM) for structure identification, Neural Networks 19 (5) (2006) 623–643.

[22] V.G. Kaburlasos, S.E. Papadakis, Piecewise-linear approximation of nonlinear models based on Interval Numbers (INs), In: V.G. Kaburlasos, U. Priss, M. Graña (Eds.) Proc. Lattice-Based Modeling (LBM 2008) Workshop, in conjunction with The Sixth International Conference on Concept Lattices and their Applications (CLA 2008), Palacký University, Olomouc, Czech Republic, 2008, pp. 13–22.

[23] V.G. Kaburlasos, S.E. Papadakis, A granular extension of the fuzzy-ARTMAP (FAM) neural classifier based on fuzzy lattice reasoning (FLR), Neurocomputing 72 (10-12) (2009) 2067–2078.

[24] V.G. Kaburlasos, I.N. Athanasiadis, P.A. Mitkas, Fuzzy lattice reasoning (FLR) classifier and its application for ambient ozone estimation, International Journal of Approximate Reasoning 45 (1) (2007) 152–188.

[25] V.G. Kaburlasos, L. Moussiades, A. Vakali, Fuzzy lattice reasoning (FLR) neural computation for weighted graph partitioning, Neurocomputing 72 (10-12) (2009) 2121–2133.

[26] S.A. Kazarlis, S.E. Papadakis, J.B. Theocharis, V. Petridis, Microgenetic algorithms as generalized hill-climbing operators for GA optimization, IEEE Transactions on Evolutionary Computation 5 (3) (2001) 204–217.

[27] E. Kim, M. Park, S. Ji, M. Park, A new approach to fuzzy modeling, IEEE Transactions on Fuzzy Systems 5 (3) (1997) 328–337.

[28] L.T. Koczy, K. Hirota, Approximate reasoning by linear rule interpolation and general approximation, International Journal of Approximate Reasoning 9 (3) (1993) 197–225.

[29] L.T. Koczy, K. Hirota, Size reduction by interpolation in fuzzy rule bases, IEEE Transactions on Systems, Man & Cybernetics - Part B 27 (1) (1997) 14–25.

[30] T. Kohonen, Self-Organizing Maps, ser. Information Sciences 30, Springer, Heidelberg, Germany, 1995.

[31] H.-M. Lee, K.-H. Chen, I-F. Jiang, A neural network classifier with disjunctive fuzzy information, Neural Networks 11 (6) (1998) 1113–1126.

[32] H.-M. Lee, C.-M. Chen, J.-M. Chen, Y.-L. Jou, An efficient fuzzy classifier with feauture selection based on fuzzy entropy, IEEE Transactions on Systems, Man & Cybernetics - Part B 31 (3) (2001) 426–432.

[33] M. Nachtegael, E.E. Kerre, Decomposing and constructing fuzzy morphological operations over α-cuts: continuous and discrete case, IEEE Transactions on Fuzzy Systems 8 (5) (2000) 615–626.

[34] K.S. Narendra, K. Parthasarathy, Identification and control of dynamical systems using neural networks, IEEE Transactions on Neural Networks 1 (1) (1990) 4–27.

[35] D.J. Newman, S. Hettich, C.L. Blake, C.J. Merz, UCI repository of machine learning databases, 1998 [available online at http://www.ics.uci.edu/˜mlearn/MLRepository.html].

[36] S.E. Papadakis, J.B. Theocharis, A novel genetic training algorithm with application to fuzzy neural networks. In: Proc. IASTED Intl. Conf. Syst. Modeling Identification and Control, Insburg, Austria, 1996, pp. 125–128.

[37] S.E. Papadakis, J.B. Theocharis, A GA-based fuzzy modeling approach for generating TSK models, Fuzzy Sets and Systems 131 (2) (2002) 121–152.

[38] S.E. Papadakis, P. Tzionas, V.G. Kaburlasos, J.B. Theocharis, A genetic based approach to the type I structure identification problem, Informatica 16 (3) (2005) 365–382.

[39] C.P. Pappis, N.I. Karacapilidis, Application of a similarity measure of fuzzy sets to fuzzy relational equations, Fuzzy Sets and Systems 75 (2) (1995) 135–142.

[40] R.D. Pascual-Marqui, A.D. Pascual-Montano, K. Kochi, J.M. Carazo, Smoothly distributed fuzzy c-means: a new self-organizing map, Pattern Recognition 34 (12) (2001) 2395–2402.

[41] W. Pedrycz, J.C. Bezdek, R.J. Hathaway, G.W. Rogers, Two nonparametric models for fusing heterogeneous fuzzy data, IEEE Transactions on Fuzzy Systems 6 (3) (1998) 411–425.

[42] G.X. Ritter, J.N. Wilson, Handbook of Computer Vision Algorithms in Image Algebra, 2nd ed., CRC Press, Boca Raton, FL, 2000.

[43] S. Senturk, N. Erginel, Development of fuzzy $\widetilde{\overline{X}} - \widetilde{R}$ and $\widetilde{\overline{X}} - \widetilde{S}$ control charts using α-cuts, Information Sciences 179 (10) (2009) 1542–1551.

[44] M. Setnes, R. Babu$\check{s}$ka, U. Kaymak, H.R. van Nauta Lemke, Similarity measures in fuzzy rule base simplification, IEEE Transactions on Systems, Man & Cybernetics - Part B 28 (3) (1998) 376–386.

[45] P.K. Simpson, Fuzzy min-max neural networks. I. classification, IEEE Transactions on Neural Networks 3 (5) (1992) 776–786.

[46] P. Soille, Morphological Image Analysis: Principles and Applications, 2nd ed., Springer-Verlag, New York, NY, 2004.

[47] M. Sugeno, G.T. Kang, Fuzzy modelling and control of multilayer incinerator, Fuzzy Sets and Systems 18 (3) (1986) 329–345.

[48] M. Sugeno, K. Tanaka, Successive identification of a fuzzy model and its applications to prediction of a complex system, Fuzzy Sets and Systems 42 (3) (1991) 315–334.

[49] M. Sugeno, T. Yasukawa, A fuzzy-logic-based approach to qualitative modeling, IEEE Transactions on Fuzzy Systems 1 (1) (1993) 7–31.

[50] P. Sussner, E.L. Esmi, Constructive morphological neural networks: some theoretical aspects and experimental results in classification, In: L. Franco, D. Elizondo, J. Jerez (Eds.) Constructive Neural Networks, Studies in Computational Intelligence, Springer, Heidelberg, 2009.

[51] P. Sussner, E.L. Esmi, An introduction to morphological perceptrons with competitive learning, Proc. 2009 International Joint Conference on Neural Networks (IJCNN 2009), Atlanta, Georgia, 2009.

[52] T. Takagi, M. Sugeno, Fuzzy identification of systems and its applications to modeling and control, IEEE Transactions on Systems, Man & Cybernetics 15 (1) (1985) 116–132.

[53] A. Torsello, D. Hidović-Rowe, M. Pelillo, Polynomial-time metrics for attributed trees, IEEE Transactions on Pattern Analysis and Machine Intelligence 27 (7) (2005) 1087–1099.

[54] K. Uehara, M. Fujise, Fuzzy inference based on families of α-level sets, IEEE Transactions on Fuzzy Systems 1 (2) (1993) 111–124.

[55] K. Uehara, K. Hirota, Parallel and multistage fuzzy inference based on families of α-level sets, Information Sciences 106 (1-2) (1998) 159–195.

[56] K. Uehara, T. Koyama, K. Hirota, Fuzzy inference with schemes for guaranteeing convexity and symmetricity in consequences based on α-cuts, Journal of Advanced Computational Intelligence and Intelligent Informatics 13 (2) (2009) 135–149.

[57] M.E. Valle, P. Sussner, A general framework for fuzzy morphological associative memories, Fuzzy Sets and Systems 159 (7) (2008) 747–768.

[58] A. Vroman, G. Deschrijver, E.E. Kerre, Solving systems of linear fuzzy equations by parametric functions, IEEE Transactions on Fuzzy Systems 15 (3) (2007) 370–384.

[59] P. Vuorimaa, Fuzzy self-organizing map, Fuzzy Sets and Systems, 66 (2) (1994) 223–231.

[60] J.-S. Wang, C.S.G. Lee, Self-adaptive neuro-fuzzy inference systems for classification applications, IEEE Transactions on Fuzzy Systems 10 (6) (2002) 790–802.

[61] P.P. Wang, Editorial: mathematics of uncertainty, Information Sciences 177 (23) (2007) 5141–5142.

[62] X. Wang, B. De Baets, E. Kerre, A comparative study of similarity measures, Fuzzy Sets and Systems 73 (2) (1995) 259–268.

[63] K.W. Wong, D. Tikk, T.D. Gedeon, L.T. Koczy, Fuzzy rule interpolation for multidimensional input spaces with applications: a case study, IEEE Transactions on Fuzzy Systems 13 (6) (2005) 809–819.

[64] D. Wu, J.M. Mendel, A comparative study of ranking methods, similarity measures and uncertainty measures for interval type-2 fuzzy sets, Information Sciences 179 (8) (2009) 1169–1192.

[65] Y. Xu, D. Ruan, K. Qin, J. Liu, Lattice-Valued Logic, ser. Studies in Fuzziness and Soft Computing 132, Springer, Heidelberg, Germany, 2003.

[66] Y. Yam, L.T. Koczy, Representing membership functions as points in high-dimensional spaces for fuzzy interpolation and extrapolation, IEEE Transactions on Fuzzy Systems 8 (6) (2000) 761–772.

[67] L.A. Zadeh, The concept of a linguistic variable and its application to approximate reasoning - I, Information Sciences 8 (3) (1975) 199–249.

[68] W. Zeng, P. Guo, Normalized distance, similarity measure, inclusion measure and entropy of interval-valued fuzzy sets and their relationship, Information Sciences 178 (5) (2008) 1334–1342.

[69] H.-Y. Zhang, W.-X. Zhang, Hybrid monotonic inclusion measure and its use in measuring similarity and distance between fuzzy sets, Fuzzy Sets and Systems 160 (1) (2009) 107–118.