In International Journal of Approximate Reasoning, **45** (**1**):152-188, 2007.

Available online at www.sciencedirect.com
ScienceDirect
International Journal of Approximate Reasoning
45 (2007) 152–188

INTERNATIONAL JOURNAL OF APPROXIMATE REASONING

www.elsevier.com/locate/ijar

Fuzzy lattice reasoning (FLR) classifier and its application for ambient ozone estimation ☆

Vassilis G. Kaburlasos [a,*], Ioannis N. Athanasiadis [b], Pericles A. Mitkas [c]

[a] *Department of Industrial Informatics, Technological Educational Institution of Kavala, GR 654 04 Kavala, Greece*

[b] *IDSIA – Istituto Dalle Molle di Studi sull'Intelligenza Artificiale, Galleria 2, CH-6928 Manno, Lugano, Switzerland*

[c] *Department of Electrical and Computer Engineering, Aristotle University of Thessaloniki, GR-54124 Thessaloniki, Greece*

Received 22 November 2004; received in revised form 11 July 2006; accepted 2 August 2006
Available online 28 August 2006

Abstract

The fuzzy lattice reasoning (FLR) classifier is presented for inducing descriptive, decision-making knowledge (rules) in a mathematical lattice data domain including space $\mathbb{R}^N$. Tunable generalization is possible based on non-linear (sigmoid) positive valuation functions; moreover, the FLR classifier can deal with missing data. Learning is carried out both incrementally and fast by computing disjunctions of join-lattice interval conjunctions, where a join-lattice interval conjunction corresponds to a hyperbox in $\mathbb{R}^N$. Our testbed in this work concerns the problem of estimating ambient ozone concentration from both meteorological and air-pollutant measurements. The results compare favorably with results obtained by C4.5 decision trees, fuzzy-ART as well as back-propagation neural

☆ This paper is an enhanced extension of a paper presented in the 1st International NAISO Symposium on Information Technologies in Environmental Engineering (ITEE 2003), Gdansk, Poland, 24–27 June 2003. V.G. Kaburlasos has been supported in part by the third European framework programme: Operational Programme in Education and Initial Vocational Training II, under project Archimedes I contract no. 04-3-001/1. Both I.N. Athanasiadis and P.A. Mitkas have been supported in part by a European Commission IST Programme under research contract no. IST-2000-31050.

* Corresponding author. Tel.: +30 2510 462 320; fax: +30 2510 462 348.
E-mail addresses: vgkabs@teikav.edu.gr (V.G. Kaburlasos), ioannis@idsia.ch (I.N. Athanasiadis), mitkas@eng.auth.gr (P.A. Mitkas).

0888-613X/$ - see front matter
doi:10.1016/j.ijar.2006.08.001

networks. Novelties and advantages of classifier FLR are detailed extensively and in comparison with related work from the literature.

Keywords: Fuzzy lattice reasoning (FLR); Classification; Machine learning; Missing values; Ambient ozone estimation

1. Introduction

Induction of rules from the training data towards classification has been popular due as well to the descriptive, decision-making knowledge represented by rules [94,95]. In order to be further useful a rule should also be able to generalize accurately.

Fuzzy inference systems (FISs) is a technology developed for *granular* rule induction and generalization based on fuzzy logic [41,44,75,78,87,88]. Note that since a data cluster can be interpreted as a (fuzzy) granule, data clustering [39,43,116] may be closely related to fuzzy rule induction. Neural implementations have provided conventional FISs a capacity for parallel implementation [44,73,86].

A novel analysis and design of FISs was proposed lately based on lattice theory [48,49]. In addition, previous work has introduced a series of fuzzy neural networks, namely *fuzzy lattice neural* (FLN) networks for clustering and classification in disparate data domains based on lattice theory. In particular, the σ-FLN, σ-FLNMAP, dσ-FLN, and FLNtf neural networks have demonstrated their effectiveness in disparate data classification applications involving numbers, (fuzzy) sets, symbols, linear operators, hyperspheres, Boolean propositions, events in a probability space, waveforms, and graphs [52,53,55,63,89,91]. This work presents the *fuzzy lattice reasoning* (FLR) classifier for inducing rules based on hyperboxes in the non-complete lattice R^N. Lattice theory equips the FLR classifier with sound tools whose effectiveness is demonstrated comparatively in several computational experiments below.

The computation of hyperboxes in the Euclidean space R^N has been a popular method for inducing rules towards classification [3,27,70,99,102,103,110,114]. A hyperbox may be assigned a class label thus corresponding to the following rule: If a point p is inside hyperbox h (let the latter be labeled by c) then p is in class c. For points outside all hyperboxes, as well as for points inside overlapping hyperboxes, various conventions have been proposed empirically. Advantages of hyperbox-based rule induction include fast computation as well as straightforward interpretation. Disadvantages often include (1) a shortage of tools for introducing tunable non-linearities, (2) restriction in the unit hypercube $[0,1]^N$, and (3) a shortage of tools for sound decision-making outside a hyperbox. The FLR classifier here retains the abovementioned advantages, moreover it mends the aforementioned disadvantages based on lattice theory as explained below.

Classifier FLR deals with rules all along. In particular, during training, the FLR classifier conditionally adapts a hyperbox-shaped core region of fuzzy sets according to a principle of minimal elongation of the hyperbox diagonal based on a series of training data pairs (a_i, c_i), $i = 1, \ldots, n$, where c_i is the class label of datum a_i; note that a training data pair (a_i, c_i) is interpreted here as rule 'if a_i then c_i', symbolically $a_i \rightarrow c_i$. During testing, the FLR classifies of a hitherto unknown rule antecedent a_0 based on the rules induced previously during training.

The FLR classifier stems from neural-fuzzy classifier σ-FLNMAP [52], the latter is enhanced here in two novel ways. First, the σ-FLNMAP is inherently restricted in complete lattices, whereas the FLR is applicable in both complete and non-complete lattices including the unit-hypercube $[0,1]^N$ and the Euclidean space R^N, respectively. Second, the σ-FLNMAP typically assumes solely a linear positive valuation function, whereas the FLR here assumes tunable non-linear- (sigmoid, etc.) as well as linear-positive valuation functions. Additional advantages of classifier FLR include a capacity for incremental learning, the potential for tunable generalization beyond a hyperbox, a capacity for granular computing, a capacity to deal with missing data, and applicability beyond R^N to a general lattice data domain. Another novelty of this work is that the FLR classifier is applied here in a real-world classification problem for ambient ozone estimation and comparative experimental results are reported extensively.

We point out that the term fuzzy lattice reasoning (FLR) has been introduced elsewhere [54] in a classification problem towards automation of a surgical drilling operation. Preliminary work including some of the experimental results presented here in the unit-hypercube, was reported in [4]. This work presents, in addition, (1) a theoretical substantiation of novel mathematical tools with emphasis in the non-complete lattice R^N, (2) an effective employment of tunable non-linearities, and (3) a large number of new experimental results.

The layout of this paper is as follows. Section 2 presents mathematical lattice notions and tools. Section 3 shows the practical relevance of mathematical tools. The FLR classifier is described in Section 4. Section 5 describes the physical problem of ambient ozone estimation. Section 6 demonstrates comparatively experimental results including useful discussions. Connections with related work from the literature are presented in Section 7. Section 8 summarizes the contribution of this work including directions for future research. Finally, the Appendix includes useful definitions followed by proofs of propositions.

2. Mathematical lattice notions and tools

This section summarizes useful mathematical lattice notions and results [8,17,47,48,51,90].

2.1. General lattices

Based on the notion *partially ordered set* (*poset*), whose definition is shown in the Appendix, a *lattice* is defined as a poset any two of whose elements have both a greatest lower bound (or *meet*), denoted by $x \wedge y$, and a least upper bound (or *join*), denoted by $x \vee y$. A lattice L is called *complete* when each of its subsets has both a least upper bound and a greatest lower bound in L. A non-void complete lattice has a *least* and a *greatest* element denoted, respectively, by O and I. Note that if x, y are elements of a lattice then x and y are either *comparable* (i.e. either $x \leqslant y$ or $y \leqslant x$) or they are *incomparable* (i.e. neither $x \leqslant y$ nor $y \leqslant x$, symbolically $x \| y$). A lattice without incomparable elements is called *totally-ordered* (lattice). An example of a totally-ordered lattice is the set R of real numbers.

Let $\mathsf{L} = \mathsf{L}_1 \times \cdots \times \mathsf{L}_N$ be the Cartesian product of N lattices $\mathsf{L}_1, \ldots, \mathsf{L}_N$, namely *constituent lattices*. A lattice inclusion relation can be defined in L as $(x_1, \ldots, x_N) \leqslant (y_1, \ldots, y_N)$ if and only if $x_1 \leqslant y_1, \ldots, x_N \leqslant y_N$. The *meet* in $\mathsf{L} = \mathsf{L}_1 \times \cdots \times \mathsf{L}_N$ is given by $(x_1, \ldots, x_N) \wedge$

$(y_1,\ldots,y_N)=(x_1\wedge y_1,\ldots,x_N\wedge y_N)$, moreover the *join* in L is given by $(x_1,\ldots,x_N)\vee(y_1,\ldots,y_N)=(x_1\vee y_1,\ldots,x_N\vee y_N)$. The *dual* of a lattice L is another lattice denoted by L^{∂}, which has, by definition, the same underlying set nevertheless its partial ordering relation ($\leqslant_{\partial}$) is the converse of L, i.e. $a\leqslant_{\partial} b$ in L^{∂} if and only if $b\leqslant a$ in L.

In this work a fuzzy set is denoted by (X,μ), where X is the universe of discourse and μ is a *fuzzy membership function* $\mu: X\to[0,1]$. The notion *fuzzy lattice*, next, was motivated for extending the crisp lattice ordering relation '$\leqslant$' to all pairs (x,y) in $\mathsf{L}\times\mathsf{L}$ including incomparable lattice elements.

Definition 1. A *fuzzy lattice* is a pair $\langle\mathsf{L},\mu\rangle$, where L is a crisp lattice and $(\mathsf{L}\times\mathsf{L},\mu)$ is a fuzzy set with membership function $\mu:\mathsf{L}\times\mathsf{L}\to[0,1]$ such that $\mu(x,y)=1$ if and only if $x\leqslant y$.

We remark that a fuzzy lattice is different from a *L-fuzzy set*; the latter is a mathematical generalization of a fuzzy set which maps a universe of discourse to a mathematical lattice [30,61,62,64,111]. Furthermore, we point out that a fuzzy lattice is different from a *type 2 fuzzy set*; the latter maps a universe of discourse to the collection of either conventional fuzzy sets or of intervals, for dealing with ambiguity in practical applications [58,71]. A requirement for a fuzzy lattice $\langle\mathsf{L},\mu\rangle$ is that L should be a crisp lattice. However, fuzzy relation μ also holds, to a fuzzy degree, between incomparable lattice elements. Note that the motivation for Definition 1 is similar to the motivation of other authors for introducing a 'fuzzy lattice' [13,79]. An instrument for fuzzifying a crisp lattice is defined next.

Definition 2. An *inclusion measure* σ in a complete lattice L is a real function $\sigma:\mathsf{L}\times\mathsf{L}\to[0,1]$ such that for $u,w,x,y\in\mathsf{L}$ the following conditions are satisfied:

(C0) $\sigma(x,O)=0,\ x\neq O$
(C1) $\sigma(x,x)=1,\ \forall x\in\mathsf{L}$
(C2) $u\leqslant w\Rightarrow\sigma(x,u)\leqslant\sigma(x,w)$ – The *Consistency Property*
(C3) $x\wedge y<x\Rightarrow\sigma(x,y)<1$.

For non-complete lattices condition C0 is dropped. Note that in every lattice L we have the equivalence $x\wedge y<x\iff y<x\vee y$ [8]; therefore condition C3 can be replaced by the following equivalent condition: (C3′) $y<x\vee y\Rightarrow\sigma(x,y)<1$.

Inclusion measure $\sigma(x,y)$ is similar to several other ones introduced in the literature between (fuzzy) sets [11,15,26,62,86,104,105,118]. Nevertheless, Definition 2 is more general since it applies to any kind of lattice, not only to a lattice of (fuzzy) sets. More specifically, $\sigma(x,y)$ can be interpreted as the fuzzy degree to which x is *less* than y; therefore notations $\sigma(x,y)$ and $\sigma(x\leqslant y)$ will be used interchangeably. Note that similar in spirit to an inclusion measure function $\sigma(x,y)$ is a zeta function $\zeta(x,y)$ in a poset [66]. The advantage of Definition 2 is that an inclusion measure σ in a crisp lattice L guarantees that $\langle\mathsf{L},\sigma\rangle$ is a fuzzy lattice as shown in the following proposition.

Proposition 3. *If* $\sigma:\mathsf{L}\times\mathsf{L}\to[0,1]$ *is an inclusion measure on lattice* L, *then* $\langle\mathsf{L},\sigma\rangle$ *is a fuzzy lattice.*

The proof of Proposition 3 is shown in the Appendix.

A valuation is defined as a real function $v : \mathsf{L} \to \mathsf{R}$ which satisfies $v(x \vee y) = v(x) + v(y) - v(x \wedge y)$, $x, y \in \mathsf{L}$. A valuation is called *positive* if and only if $x < y \Rightarrow v(x) < v(y)$ [8]. Based on a positive valuation, two different inclusion measure functions can be introduced as follows.

Proposition 4. *If* L *is a* (*complete*) *lattice and* $v : \mathsf{L} \to \mathsf{R}$ *is a positive valuation* (*with* $v(O) = 0$) *then* (1) $k(x,u) = \frac{v(u)}{v(x \vee u)}$, *and* (2) $s(x,u) = \frac{v(x \wedge u)}{v(x)}$ *are inclusion measures.*

The proof of Proposition 4 is shown in the Appendix.

We remark that both inclusion measures $k(x,u)$ and $s(x,u)$ have been introduced in [45]. Note that inclusion measure $s(x,u)$ as well as $k(x,u)$ can be interpreted as a degree of 'subsethood' of x in u [26]. Consider a subset (powerset) lattice, where the partial order is the subset relation. Then, inclusion measure $s(x,u)$ above, for a suitably selected positive valuation function v, can be used for defining a degree of inclusion of a subset x to a subset u. However, there is a serious drawback in the practical employment of $s(x,u)$. More specifically for $x \wedge u = \emptyset$ there follows $s(x,u) = 0$, hence a sensible decision is not possible. Nevertheless it is always $k(x,u) \neq 0$ for $x, u \neq \emptyset$, hence a sensible decision is always possible. The latter is an advantage for inclusion measure $k(x,u)$. Therefore, in the sequel, our interest mainly focuses on inclusion measure $k(x,u)$.

Another useful tool implied by a positive valuation function in a general lattice L is a *metric* $d : \mathsf{L} \times \mathsf{L} \to \mathsf{R}$ defined as $d(x,y) = v(x \vee y) - v(x \wedge y)$ – for the definition of a *metric* see in the Appendix.

Given (1) a product lattice $\mathsf{L} = \mathsf{L}_1 \times \cdots \times \mathsf{L}_N$, and (2) both a positive valuation $v_i : \mathsf{L}_i \to \mathsf{R}$, $i = 1, \ldots, N$ and an isomorphic function $\theta_i : \mathsf{L}_i^{\partial} \to \mathsf{L}_i$ in every constituent lattice L_i, $i = 1, \ldots, N$ – for the definition of an *isomorphism* see in the Appendix – then: (1) a positive valuation function $v : \mathsf{L} \to \mathsf{R}$ is given by $v(x_1, \ldots, x_N) = v_1(x_1) + \cdots + v_N(x_N)$, (2) an isomorphic function $\theta : \mathsf{L}^{\partial} \to \mathsf{L}$ is given by $\theta(x_1, \ldots, x_N) = (\theta_1(x_1), \ldots, \theta_N(x_N))$, and (3) countably infinite *Minkowski metrics* d_p in L are given by

$$d_p(x,y) = d_p((x_1, \ldots, x_N), (y_1, \ldots, y_N)) = [(d_1(x_1, y_1))^p + \cdots + (d_N(x_N, y_N))^p]^{1/p}$$

where $p = 1, 2, \ldots$ and $d_i(x_i, y_i) = v_i(x_i \vee y_i) - v_i(x_i \wedge y_i)$, $x_i, y_i \in L_i$, $i = 1, \ldots, N$. Note that, conventionally, a Minkowski metric d_p requires parameter p to be a natural (integer) number [42]. However, parameter p above can be a real number resulting in uncountably infinite Minkowski metrics d_p in L.

In the remaining of this work interest focuses on Cartesian products $\mathsf{L} = \mathsf{L}_1 \times \cdots \times \mathsf{L}_N$ of N 'totally-ordered' lattices L_i, $i = 1, \ldots, N$. Note that the aforementioned Cartesian product lattice L is not totally-ordered.

2.2. Extensions to lattices of intervals

Let L_i be a totally-ordered lattice. It is known that the set $\tau(\mathsf{L}_i) = \{[a,b] : a, b \in \mathsf{L}_i\}$ of *generalized intervals* is a mathematical lattice [46–48,50,92] with *lattice-meet* and *lattice-join* given, respectively, by $[a,b] \wedge [c,d] = [a \vee c, b \wedge d]$ and $[a,b] \vee [c,d] = [a \wedge c, b \vee d]$. Moreover, the corresponding lattice order relation $[a,b] \leqslant [c,d]$ in $\tau(\mathsf{L}_i)$ is equivalent to '$c \leqslant a$'.AND.'$b \leqslant d$'. The following proposition introduces a positive valuation in a lattice of generalized intervals.

Proposition 5. *Let* L_i *be a totally-ordered lattice, let* $v : \mathsf{L}_i \to \mathsf{R}$ *be a positive valuation, and let* $\theta : \mathsf{L}_i^{\partial} \to \mathsf{L}_i$ *be an isomorphic function in* L_i. *Then a positive valuation function* $v : \tau(\mathsf{L}_i) \to \mathsf{R}$ *is given by* $v([a, b]) = v(\theta(a)) + v(b)$.

The proof of Proposition 5 is shown in the Appendix.

A metric in a product lattice $\mathsf{L} = \mathsf{L}_1 \times \cdots \times \mathsf{L}_N$ can quantify the size of a lattice interval in $\tau(\mathsf{L})$ as follows.

Definition 6. Consider a product lattice $\mathsf{L} = \mathsf{L}_1 \times \cdots \times \mathsf{L}_N$. Let $v_i : \mathsf{L}_i \to \mathsf{R}$ be a positive valuation function in the constituent lattice L_i, $i = 1, \ldots, N$. Then the *diagonal* of an interval $[a, b] \in \tau(\mathsf{L})$, with $a \leqslant b$, is defined as a non-negative real function $\mathrm{diag}_p : \tau(\mathsf{L}) \to \mathsf{R}_0^+$ given by $\mathrm{diag}_p([a, b]) = d_p(a, b)$, $p = 1, 2 \ldots$

The following proposition establishes that $\mathrm{diag}_p([a, b])$ equals the largest distance for two points x and y in the interval $[a, b]$.

Proposition 7. *For* $p = 1, 2, \ldots$ *we have* $\mathrm{diag}_p([a, b]) = \max_{x,y \in [a,b]} d_p(x, y)$.

The proof of Proposition 7 is shown in the Appendix.

3. Practical relevance

Two specific lattices of practical interest will be considered next including, first, the complete lattice unit-hypercube I^N, where $\mathsf{I} = [0, 1]$ and, second, the non-complete lattice R^N – note that the treatment of space R^N is a novelty here. An interval in either I^N or R^N, corresponds to a *N-dimensional hyperbox*, or *hyperbox* for short. It is known that the set of hyperboxes in $\mathsf{I}^N(\mathsf{R}^N)$ is a complete (non-complete) lattice [90]. We point out that dealing with hyperboxes has been popular in machine learning as well as in neural computing [3,19,27,99,102,103,110,114] without, however, taking advantage of the lattice-ordering relation of hyperboxes. An important advantage of considering the latter relation is the capacity to introduce tunable non-linearities towards improving performance as shown below.

Any strictly increasing real function v_i, with $v_i(0) = 0$, in a constituent lattice L_i, $i = 1, \ldots, N$ is an eligible positive valuation. Moreover, any strictly decreasing function θ_i in a constituent lattice L_i, $i = 1, \ldots, N$ is an eligible isomorphic function. Below, we select functions v_i and θ_i such that equation $v([a, b]) = 1 + \mathrm{diag}_1([a, b])$ is satisfied in order to build on the basic equations used by the σ-FLN(MAP) model [52].

On the one hand, regarding the unit hypercube I^N, two convenient functions v_i and θ_i are, respectively, $v_i(x) = x$ and $\theta_i(x) = 1 - x$. It is known from [52] that the latter functions v_i and θ_i satisfy equality $v([a, b]) = 1 + \mathrm{diag}_1([a, b])$ in a constituent complete lattice $\mathsf{I} = [0, 1]$. On the other hand, regarding the Euclidean space R^N, this work introduces the following functions: $v_i(x) = \frac{1}{1+e^{-\lambda(x-x_0)}}$, and $\theta_i(x) = 2x_0 - x$. Fig. 1 shows plots of both aforementioned functions v_i and θ_i for selected parameter values. It can be easily shown that $v([x, x]) = v(\theta(x)) + v(x) = 1$. Hence,

$$v([a, b]) = v(\theta(a)) + v(b) = [1 - v(a)] + v(b) = 1 + [v(b) - v(a)] = 1 + \mathrm{diag}_1([a, b]).$$

Under the above assumptions, the *Consistency Property* C2 is demonstrated in Fig. 2, where

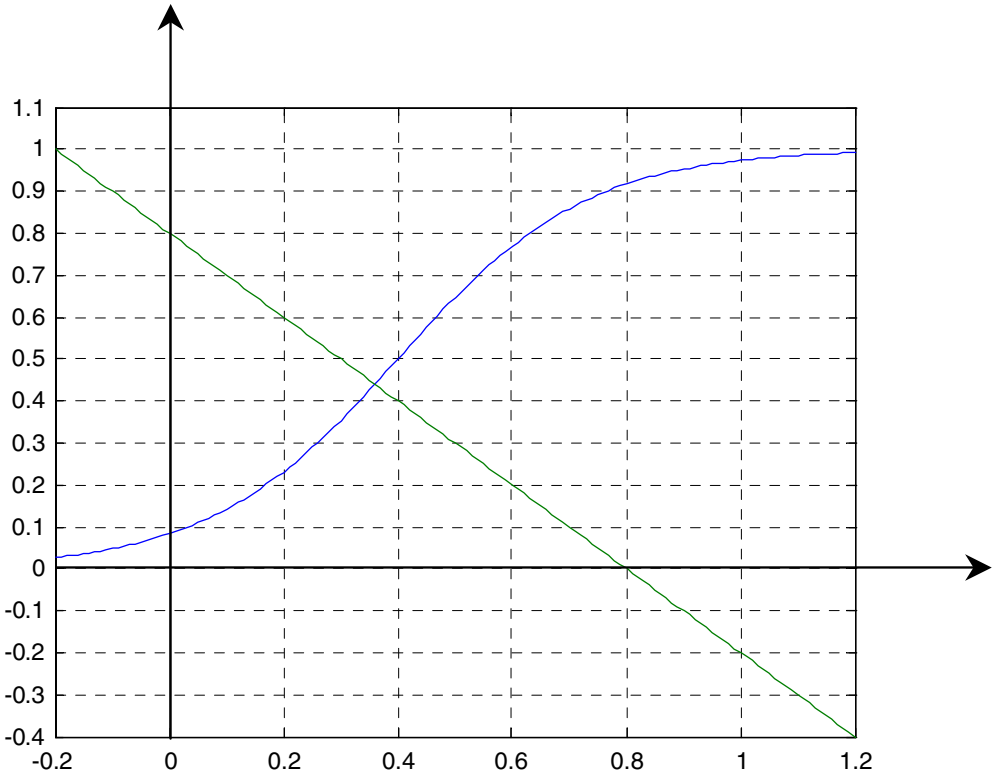

Fig. 1. A sigmoid positive valuation $v_i(x) = 1/(1 + \exp(-\lambda(x - x_0)))$ and a linear isomorphic function $\theta_i(x) = 2x_0 - x$, where $\lambda = 6$ and $x_0 = 0.4$, on the totally-ordered constituent lattice R of real numbers.

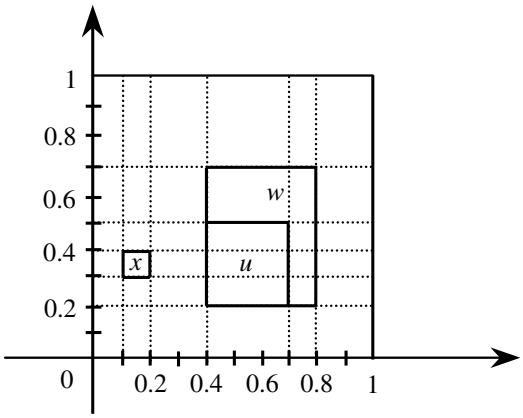

Fig. 2. Using the inclusion measure $k(x \leqslant u) = \frac{v(u)}{v(x \vee u)}$ for any choice of both a positive valuation $v(\cdot)$ and an isomorphic function $\theta(\cdot)$ in a constituent lattice $[0,1]$ it follows $k(x,u) \leqslant k(x,w)$ according to the *Consistency Property* $u \leqslant w \Rightarrow k(x \leqslant u) \leqslant k(x \leqslant w)$.

$$x = [0.1, 0.2] \times [0.3, 0.4], \quad u = [0.4, 0.7] \times [0.2, 0.5], \quad \text{and}$$
$$w = [0.4, 0.8] \times [0.2, 0.7].$$

An inclusion measure $k(\cdot,\cdot)$ of two boxes in Fig. 2 is computed as follows:

$$k(x \leqslant u) = \frac{v(u)}{v(x \vee u)} = \frac{v([0.4,0.7] \times [0.2,0.5])}{v([0.1,0.7] \times [0.2,0.5])} = \frac{v_1(\theta_1(0.4)) + v_1(0.7) + v_2(\theta_2(0.2)) + v_2(0.5)}{v_1(\theta_1(0.1)) + v_1(0.7) + v_2(\theta_2(0.2)) + v_2(0.5)}$$
$$k(x \leqslant w) = \frac{v(w)}{v(x \vee w)} = \frac{v([0.4,0.8] \times [0.2,0.7])}{v([0.1,0.8] \times [0.2,0.7])} = \frac{v_1(\theta_1(0.4)) + v_1(0.8) + v_2(\theta_2(0.2)) + v_2(0.7)}{v_1(\theta_1(0.1)) + v_1(0.8) + v_2(\theta_2(0.2)) + v_2(0.7)}$$

First, in the unit hypercube I^N, functions $v_i(x) = x$ and $\theta_i(x) = 1 - x$, $i = 1,2$ imply

$$k(x \leqslant u) = \frac{(1-0.4)+0.7+(1-0.2)+0.5}{(1-0.1)+0.7+(1-0.2)+0.5} \cong 0.89, \quad \text{and}$$

$$k(x \leqslant w) = \frac{(1-0.4)+0.8+(1-0.2)+0.7}{(1-0.1)+0.8+(1-0.2)+0.7} \cong 0.90$$

Second, in the space R^N, functions $v_i(x) = 1/(1+\exp(-\lambda(x-x_0)))$ and $\theta_i(x) = 2x_0 - x$ with parameter values $\lambda = 6$ and $x_0 = 0.4$, $i = 1,2$ imply

$$k(x \leqslant u) = \frac{0.5+0.8581+0.7685+0.6457}{0.8581+0.8581+0.7685+0.6457} \cong 0.8856, \quad \text{and}$$

$$k(x \leqslant w) = \frac{0.5+0.9168+0.7685+0.8581}{0.8581+0.9168+0.7685+0.8581} \cong 0.8947$$

That is, as guaranteed by the *Consistency Property* C2, the degree of inclusion of box x in box w is larger than the degree of inclusion of x in u because box u is contained in box w. The above example has demonstrated, in addition, four important points. First, the inclusion measure $k(x,u)$ has a capacity to generalize beyond the interval core of u or w. Second, the inclusion measure $k(x,u)$ can involve a non-trivial hyperbox x. Third, a larger value $k(x,w)$ is computed for a hyperbox (here for hyperbox w), which needs to be 'distorted' *the least* so as to include x; the latter corresponds to *Occam razor* semantics as explained below. Fourth, a positive valuation may introduce non-linearities in the data space.

The following example demonstrates advantages of the metric distance $d_1(\cdot,\cdot)$ compared with other distance functions from the literature.

Example

In [99] as well as in [114] a distance function $d_S(x,[a,b])$ between a point x and an interval $[a,b]$ is defined as

$$d_S(x,[a,b]) = \begin{cases} a-x, & x \leqslant a \\ 0, & a \leqslant x \leqslant b \\ x-b, & x \geqslant b \end{cases}$$

Specific distances between points and boxes are computed in the following with reference to Fig. 3. It turns out $d_S(x_1,u) = 0 = d_S(x_2,u)$, furthermore there follows the L1 distance $d_S(x_1,x_2) = 0.8$. Nevertheless, the aforementioned results are counter-intuitive because they violate the common sense *triangle-inequality* $d_S(x_1,x_2) \leqslant d_S(x_1,u) + d_S(u,x_2)$.

Here we decided to adhere to Fréchet's original definition for a *metric* [42] shown in the Appendix. Unfortunately not all *metrics* produce common sense results. For instance, a known metric (distance) between convex sets is the *Hausdorf* metric $d_{\mathrm{H}}(\cdot,\cdot)$ [18]. It is known that the distance $d_{\mathrm{H}}(\cdot,\cdot)$ between two intervals $[a,b]$ and $[c,d]$ on the real line equals $d_{\mathrm{H}}([a,b],[c,d]) = \max\{|a-c|,|b-d|\}$ [32,37]. In the following we compute the $L1$ distance between boxes in Fig. 3.

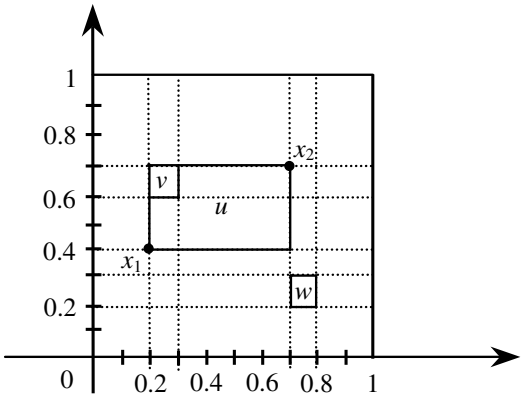

Fig. 3. Various distances have been computed involving boxes v, u, w and points x_1, x_2. First, for the distance $d_S(\cdot,\cdot)$ from [114] it follows $d_S(x_1,u)=0=d_S(x_2,u)$, moreover $d_S(x_1,x_2)=0.8$; the latter results are counter-intuitive because the *triangle inequality* $d_S(x_1,x_2)\leqslant d_S(x_1,u)+d_S(u,x_2)$ is violated. Second, for the Hausdorf metric distance $d_H(\cdot,\cdot)$ from [18] it follows $d_H(v,w)=0.9=d_H(u,w)$; by inspecting the above figure, the latter equality is counter-intuitive. Third, for the distance $d_1(\cdot,\cdot)$ presented in this work for any positive valuation $v(\cdot)$ and/or an isomorphic function $\theta(\cdot)$ in a constituent lattice it holds $d_1(v,w)>d_1(u,w)$; moreover, the triangle inequality $d_1(x_1,x_2)\leqslant d_1(x_1,u)+d_1(u,x_2)$ is preserved.

$$d_H(v,w)=d_H([0.2,0.3],[0.7,0.8])+d_H([0.6,0.7],[0.2,0.3])=0.5+0.4=0.9$$
$$d_H(u,w)=d_H([0.2,0.7],[0.7,0.8])+d_H([0.4,0.7],[0.2,0.3])=0.5+0.4=0.9$$

However, by inspecting Fig. 3, it is reasonable to expect $d(v,w)>d(u,w)$. In conclusion, even though the Hausdorf distance $d_H(\cdot,\cdot)$ is a *metric*, it produces counter-intuitive results. In the following we show that metric $d_1(\cdot,\cdot)$ produces intuitive results.

The metric distance $d_1(\cdot,\cdot)$ between two boxes in Fig. 3 is computed as follows:

$$\begin{aligned}d_1(v,u)&=d_1([0.2,0.3],[0.7,0.8])+d_1([0.6,0.7],[0.2,0.3])=v_1([0.2,0.3]\vee[0.7,0.8])\\&\quad-v_1([0.2,0.3]\wedge[0.7,0.8])+v_2([0.6,0.7]\vee[0.2,0.3])-v_2([0.6,0.7]\wedge[0.2,0.3])\\&=v_1([0.2,0.8])-v_1([0.7,0.3])+v_2([0.2,0.7])-v_2([0.6,0.3])=v_1(\theta_1(0.2))+v_1(0.8)\\&\quad-v_1(\theta_1(0.7))-v_1(0.3)+v_2(\theta_2(0.2))+v_2(0.7)-v_2(\theta_2(0.6))-v_2(0.3)\\d_1(u,w)&=d_1([0.2,0.7],[0.7,0.8])+d_1([0.4,0.7],[0.2,0.3])=v_1([0.2,0.7]\vee[0.7,0.8])\\&\quad-v_1([0.2,0.7]\wedge[0.7,0.8])+v_2([0.4,0.7]\vee[0.2,0.3])-v_2([0.4,0.7]\wedge[0.2,0.3])\\&=v_1([0.2,0.8])-v_1([0.7,0.7])+v_2([0.2,0.7])-v_2([0.4,0.3])=v_1(\theta_1(0.2))+v_1(0.8)\\&\quad-v_1(\theta_1(0.7))-v_1(0.7)+v_2(\theta_2(0.2))+v_2(0.7)-v_2(\theta_2(0.4))-v_2(0.3)\end{aligned}$$

First, in the unit hypercube I^N, functions $v_i(x)=x$ and $\theta_i(x)=1-x$, $i=1,2$ imply

$$\begin{aligned}d_1(v,w)&=v_1(0.8)+v_1(0.8)-v_1(0.3)-v_1(0.3)\\&\quad+v_2(0.8)+v_2(0.7)-v_2(0.4)-v_2(0.3)=1.8\\d_1(u,w)&=v_1(0.8)+v_1(0.8)-v_1(0.3)-v_1(0.7)\\&\quad+v_2(0.8)+v_2(0.7)-v_2(0.6)-v_2(0.3)=1.2\end{aligned}$$

Second, in the space R^N, functions $v_i(x)=1/(1+\exp(-\lambda(x-x_0)))$ and $\theta_i(x)=2x_0-x$ with parameter values $\lambda=6$ and $x_0=0.4$, $i=1,2$ imply

$$d_1(v,w) = v_1(0.6) + v_1(0.8) - v_1(0.1) - v_1(0.3) + v_2(0.6) + v_2(0.7) - v_2(0.2) - v_2(0.3) \cong 2.2299$$

$$d_1(u,w) = v_1(0.6) + v_1(0.8) - v_1(0.1) - v_1(0.7) + v_2(0.6) + v_2(0.7) - v_2(0.4) - v_2(0.3) \cong 1.4576$$

Therefore, $d_1(v,w) > d_1(u,w)$ as expected intuitively from Fig. 3 by inspection. In the following we show that the common sense triangle inequality $d_1(x_1,x_2) \leqslant d_1(x_1,u) + d_1(x_2,u)$ is satisfied

$$\begin{aligned} d_1(x_1,x_2) &= d_1([0.2,0.2],[0.7,0.7]) + d_1([0.4,0.4],[0.7,0.7]) \\ &= v_1([0.2,0.7]) - v_1([0.7,0.2]) + v_2([0.4,0.7]) - v_2([0.7,0.4]) \\ &= v_1(\theta_1(0.2)) + v_1(0.7) - v_1(\theta_1(0.7)) - v_1(0.2) + v_2(\theta_2(0.4)) + v_2(0.7) - v_2(\theta_2(0.7)) - v_2(0.4) \\ d_1(x_1,u) &= v_1(\theta_1(0.2)) + v_1(0.7) - v_1(\theta_1(0.2)) - v_1(0.2) + v_2(\theta_2(0.4)) + v_2(0.7) - v_2(\theta_2(0.4)) - v_2(0.4) \\ d_1(x_2,u) &= v_1(\theta_1(0.2)) + v_1(0.7) - v_1(\theta_1(0.7)) - v_1(0.7) + v_2(\theta_2(0.4)) + v_2(0.7) - v_2(\theta_2(0.7)) - v_2(0.7) \end{aligned}$$

First, in the unit hypercube I^N, functions $v_i(x) = x$ and $\theta_i(x) = 1 - x$, $i = 1,2$ imply

$$d_1(x_1,x_2) = 1.6, \quad d_1(x_1,u) = 0.8 \quad \text{and} \quad d_1(x_2,u) = 0.8$$

Second, in the space R^N, functions $v_i(x) = 1/(1 + \exp(-\lambda(x - x_0)))$ and $\theta_i(x) = 2x_0 - x$ with parameter values $\lambda = 6$ and $x_0 = 0.4$, $i = 1,2$ imply

$$d_1(x_1,x_2) = 1.9694, \quad d_1(x_1,u) = 0.9847 \quad \text{and} \quad d_1(x_2,u) = 0.9847$$

Therefore, the triangle inequality $d_1(x_1,x_2) \leqslant d_1(x_1, u) + d_1(x_2,u)$ is satisfied.

4. The fuzzy lattice reasoning (FLR) classifier

This section describes and analyzes the operation of the FLR classifier.

4.1. Rule induction (learning)

The fuzzy lattice reasoning (FLR) classifier induces rules from the training data by letting a rule's diagonal size increase up to a maximum threshold size D_{crit} (Fig. 4).

The FLR is a leader-follower classifier [24], which learns rapidly in a single pass through the training data. The order of input data presentation is significant. The FLR classifier may set out learning without *a priori* knowledge; however, *a priori* knowledge can be supplied to the FLR classifier in the form of an initial set of rules.

The total number of rules to be learned is not known *a priori* but, rather, it is determined on-line during learning. Further training of the FLR classifier, using additional training data, does not wash away previous learning. More specifically, retraining the FLR classifier with a new data set either enhances previously learned rules (step-5 in Fig. 4) or it creates new rules (step-2 in Fig. 4). There is only one (real number) parameter to tune, that is the maximum threshold size D_{crit}, which regulates the *granularity of learning*; the latter means the number of rules induced. It turns out that, in general, larger values of D_{crit} result in fewer (also, more generalized) rules, whereas smaller values of D_{crit} result in more (also, more specific) rules. Note that in step-4 (Fig. 4), condition

Step-0: A rule base RB= $\{A_1 \rightarrow C_1, \ldots, A_L \rightarrow C_L\}$ is given – Note that RB could initially be empty, i.e. L=0.
Step-1: Present the next input rule $a_i \rightarrow c_i$, i=1,…,n to the initially 'set' family of rules in RB.
Step-2: If no more rules in RB are 'set' then
  Store input rule $a_i \rightarrow c_i$ in RB;
  L←L+1;
  Goto step-1.
  Else, compute the fuzzy degree of inclusion $k(a_i \leq A_l)$, $l \in \{1,\ldots,L\}$ of antecedent a_i to the antecedents of all the 'set' rules in RB.
Step-3: Competition among the 'set' rules in RB.
  Winner is rule $A_J \rightarrow C_J$ such that $J = \arg \max_{l \in \{1,\ldots L\}} k(a_i \leq A_l)$.
Step-4: The *Assimilation Condition*: Both $diag_1(a_i \vee A_J)$ is less than a maximum user-defined threshold size D_{crit} and $c_i = C_J$.
Step-5: If the *Assimilation Condition* is satisfied then
  Replace the antecedent A_J of the winner rule $A_J \rightarrow C_J$ by the join-interval $a_i \vee A_J$.
  Else, 'reset' the winner rule $A_J \rightarrow C_J$.
  Goto step-2.

Fig. 4. Rule induction by the fuzzy lattice reasoning (FLR) classifier.

'$\mathrm{diag}_1(a_i \vee A_J) \leqslant D_{\mathrm{crit}}$' can equivalently be replaced by condition '$k(A_J \leqslant a_i) \geqslant \rho_{\mathrm{crit}}$', where $\rho_{\mathrm{crit}} = N/(N + D_{\mathrm{crit}}) \iff D_{\mathrm{crit}} = N(1 - \rho_{\mathrm{crit}})/\rho_{\mathrm{crit}}$ with the dimensionless *vigilance parameter* ρ_{crit} in the interval $[0.5, 1]$, i.e. $\rho_{\mathrm{crit}} \in [0.5, 1]$.

4.2. Generalization

Fig. 5 describes the FLR classifier's generalization based on a rule base RB induced previously.

We remark that during both rule induction and generalization the inclusion measure $k(\cdot,\cdot)$ is used to compute, in parallel, a fuzzy degree of activation for each rule. It was pointed out above that in the context of this work a positive valuation function $v_i : \mathsf{L}_i \rightarrow \mathsf{R}$ and an isomorphic function $\theta_i : \mathsf{L}_i^{\partial} \rightarrow \mathsf{L}_i$ are selected in a constituent lattice L_i, $i = 1, \ldots, n$ such that $v_i([a,b]) = 1 + \mathrm{diag}_1([a,b])$, where $[a,b] \in \tau(\mathsf{L}_i)$. It follows $k(a_0 \leqslant A_l) = \frac{v(A_l)}{v(a_0 \vee A_l)} = \frac{\mathrm{diag}_1(A_l)+N}{\mathrm{diag}_1(a_0 \vee A_l)+N}$. Based on the latter formula this work attaches *Occam razor* semantics to inclusion measure $k(\cdot,\cdot)$ as explained in the following. Let $A_1, \ldots, A_L$ be hyperboxes (rule antecedents) competing over hyperbox (rule antecedent) a_0, i.e. the largest $k(a_0 \leqslant A_l)$, $l = 1, \ldots, \mathsf{L}$ is sought. It follows that winner A_J among hyperboxes $A_1, \ldots, A_L$ will be the one whose diagonal size needs to be modified, comparatively, *the least* so as to include a_0. In this sense, the winner box A_J is the simplest hypothesis that fits the data, that is *Occam razor* semantics [77].

Step-0: Consider a rule base RB= $\{A_1 \rightarrow C_1, \ldots, A_L \rightarrow C_L\}$.
Step-1: Present a rule antecedent 'a_0' for classification based on the rule base RB.
Step-2: Calculate the fuzzy degree of inclusion $k(a_0 \leq A_l)$, $l \in \{1,\ldots,L\}$ of antecedent 'a_0' in the antecedents 'A_l', l=1,…,L of all the rules in RB.
Step-3: Competition among the rules in RB.
  Winner is rule $A_J \rightarrow C_J$, where $J = \arg \max_{l \in \{1,\ldots L\}} k(a_0 \leq A_l)$.
Step-4: The antecedent a_0 is classified to the class with label C_J.

Fig. 5. Generalization by the fuzzy lattice reasoning (FLR) classifier.

4.3. FLR classifier execution

This section illustrates the mechanics of classifier FLR. Consider three rules $A_1 \rightarrow c_0$, $A_2 \rightarrow c_1$, $A_3 \rightarrow c_1$, whose antecedents (boxes) A_1, A_2, and A_3 are shown in Fig. 6. Assume a sigmoid positive valuation function $v(x) = 1/(1 + \exp(-\lambda(x - x_0)))$ and a linear isomorphic function $\theta(x) = 2x_0 - x$ with parameter values $\lambda = 6$ and $x_0 = 0.4$ in both dimensions. The latter functions are shown in Fig. 1. In the following we demonstrate execution of classifier FLR for training assuming a used-defined threshold size $D_{\text{crit}} = 1.05$.

Let an input (rule) $a \rightarrow c_1$ appear as shown in Fig. 6(a). Recall that initially all rules are 'set', that is all rules can claim input $a \rightarrow c_1$. According to classifier FLR for training, all rules compete over input rule $a \rightarrow c_1$ (Fig. 6(b)) by calculating the following inclusion measures:

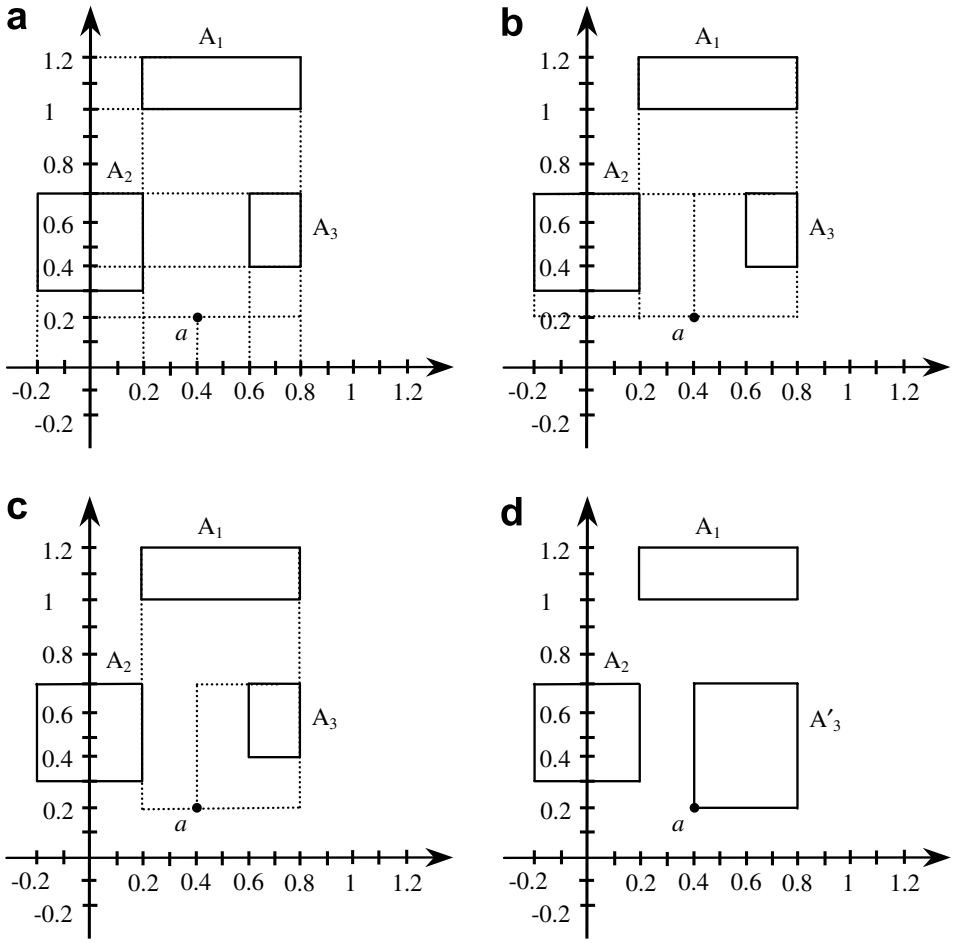

Fig. 6. Execution of classifier FLR for training. Functions $v(x) = 1/(1 + \exp(-\lambda(x - x_0)))$ and $\theta(x) = 2x_0 - x$ were used in both dimensions on the plane with parameter values $\lambda = 6$ and $x_0 = 0.4$. (a) Input rule $a \rightarrow c_1$, where 'a' is a point and 'c_1' is the corresponding class label, is presented to the initially 'set' rule base $\text{RB} = \{A_1 \rightarrow c_0, A_2 \rightarrow c_1, A_3 \rightarrow c_1\}$. (b) Competition among the 'set' rules in RB: Winner is rule $A_2 \rightarrow c_1$, since $k(a \leqslant A_1) = 0.7847$, $k(a \leqslant A_2) = 0.8738$, and $k(a \leqslant A_3) = 0.8236$. Nevertheless rule $A_2 \rightarrow c_1$ is 'reset' because it does not satisfy the *Assimilation Condition*; in particular $\text{diag}_1(a \vee A_2) = 1.10 > 1.05 = D_{\text{crit}}$. (c) Competition among the 'set' rules in RB: Winner is now rule $A_3 \rightarrow c_1$, which satisfies the *Assimilation Condition*; in particular, $\text{diag}_1(a \vee A_3) = 1.0434 < 1.05 = D_{\text{crit}}$. (d) The antecedent of the winner rule $A_3 \rightarrow c_1$ has been replaced by the join-interval $A'_3 = a \vee A_3$. The updated rule is $A'_3 \rightarrow c_1$.

$$k(a \leqslant A_1) = \frac{v(A_1)}{v(a \vee A_1)} = \frac{v([0.2,0.8] \times [1,1.2])}{v([0.2,0.8] \times [0.2,1.2])}$$
$$= \frac{v(\theta(0.2)) + v(0.8) + v(\theta(1)) + v(1.2)}{v(\theta(0.2)) + v(0.8) + v(\theta(0.2)) + v(1.2)}$$
$$= \frac{v(0.6) + v(0.8) + v(-0.2) + v(1.2)}{v(0.6) + v(0.8) + v(0.6) + v(1.2)} \cong 0.7847.$$
$$k(a \leqslant A_2) = \frac{v(A_2)}{v(a \vee A_2)} = \frac{v([-0.2,0.2] \times [0.3,0.7])}{v([-0.2,0.4] \times [0.2,0.7])} \cong 0.8738.$$
$$k(a \leqslant A_3) = \frac{v(A_3)}{v(a \vee A_3)} = \frac{v([0.6,0.8] \times [0.4,0.7])}{v([0.4,0.8] \times [0.2,0.7])} \cong 0.8236.$$

Hence, winner of the competition is rule $A_2 \to c_1$ because it includes input rule $a \to c_1$ more than any other rule. In order to test the Assimilation Condition we calculate the following diagonals:

$$\mathrm{diag}_1(a \vee A_1) = \mathrm{diag}_1([0.2,0.8] \times [0.2,1.2]) = d_1([(0.2,0.2),(0.8,1.2)]) = d_1((0.2,0.2),(0.8,1.2))$$
$$= d_1(0.2,0.8) + d_1(0.2,1.2) = v(0.8) - v(0.2) + v(1.2) - v(0.2) \cong 1.4456.$$
$$\mathrm{diag}_1(a \vee A_2) = \mathrm{diag}_1([-0.2,0.4] \times [0.2,0.7]) \cong 1.1000.$$
$$\mathrm{diag}_1(a \vee A_3) = \mathrm{diag}_1([0.4,0.8] \times [0.2,0.7]) \cong 1.0434.$$

It follows that winner rule $A_2 \to c_1$ is 'reset' because it does not satisfy the *Assimilation Condition since* $\mathrm{diag}_1(a \vee A_2) = 1.10 > 1.05 = D_{\mathrm{crit}}$. According to algorithm FLR (Fig. 4), competition resumes among the remaining rules as shown in Fig. 6(c). Winner is now rule $A_3 \to c_1$, which satisfies the Assimilation Condition since $\mathrm{diag}_1(a \vee A_3) = 1.0434 < 1.05 = D_{\mathrm{crit}}$. In conclusion, the antecedent A_3 of winner rule $A_3 \to c_1$ is replaced by the join-interval $a \vee A_3 = A_3'$ shown in Fig. 6(d).

Overlapping of rules (hyperboxes) is possible including 'contradictory overlapping' of hyperboxes with different labels. Our experience in a large number of computational experiments has shown that overlapping of contradictory rules (hyperboxes) is especially rare.

4.4. Fuzzy lattice reasoning (FLR) essentials

Four essentials of fuzzy lattice reasoning are summarized in this section.

First, according to the *Assimilation Condition*, rule induction may be effected by replacing a hyperbox A_J by a larger hyperbox $a_i \vee A_J$. It follows that there might be points within the larger hyperbox $a_i \vee A_J$ which (points) are assigned category label C_J 'inductively' without explicit evidence. The latter is called here *Type I Generalization*. Note that *Type I Generalization* may result in overlapping of hyperboxes. The latter can be avoided by conditionally augmenting the *Assimilation Condition* (Fig. 4, step-4) at the expense of longer computer processing times. Our extensive experimental experience using a number of data sets in various classification applications has shown that FLR's capacity for generalization, defined as the percent classification accuracy on a testing data set, behaves like a convex function in the threshold size D_{crit}; hence, a globally optimum threshold size can typically be sought by simple hill climbing. No theoretical analysis is currently available regarding the aforementioned 'convexity', nevertheless relevant plots have been reported [63]. Further theoretical study is a topic for future research.

Second, a rule $A_l \rightarrow C_l$, $l = 1, \ldots, L$ defines a fuzzy set $k(x \leqslant A_l)$ in the family of hyperboxes such that hyperbox A_l corresponds to the *core* of fuzzy set $k(x \leqslant A_l)$. Different positive valuation functions imply different membership functions for the fuzzy set $k(x \leqslant A_l)$. In all cases generalization becomes feasible beyond core (hyperbox) A_l. The latter generalization is called here *Type II Generalization*. Furthermore, note that x in $k(x \leqslant A_l)$ can be a hyperbox in order to compensate for data ambiguity.

Third, fuzzy lattice reasoning can deal with semantics in at least two different senses: (1) Occam razor semantics as explained above, and (2) non-numeric data, e.g. structured data (graphs), etc., can be accommodated in a constituent lattice [47,91].

Fourth, the FLR classifier can deal with a missing data value in a constituent lattice L_i by replacing a missing datum with a lattice interval $[a, b]$ such that $v_i([a, b]) = v_i(\theta_i(a)) + v_i(b) \cong 0$. The latter replacement is semantically interpreted as 'absence of information'. For instance, a missing feature in the constituent lattice $\mathsf{I} = [0, 1]$ in the N-dimensional unit hypercube I^N is replaced by $[1, 0]$. Note that dealing with missing data values is an important issue in practice [112].

4.5. Geometric interpretations on the plane and modes of reasoning

Assume that five rules, namely $a_1 \rightarrow 0$, $a_2 \rightarrow 0$, $a_3 \rightarrow 0$, $a_4 \rightarrow 1$, and $a_5 \rightarrow 1$, have been computed by the FLR classifier (Fig. 7). Let the first three rules map the antecedent boxes a_1, a_2 and a_3 to class label '0', whereas let the last two rules map the antecedent boxes a_4 and a_5 to class label '1'. In Fig. 7, no antecedent boxes overlap each other but it could be otherwise.

Each one of the rule antecedent boxes $a_1, \ldots, a_5$ can be regarded as a conjunctive logical expression. For example, the conjunctive logical expression which corresponds to antecedent box $a_3 = [0.1, 0.2] \times [0.4, 0.7]$ is true for a point (x_1, x_2) if and only if '$0.1 \leqslant x_1 \leqslant 0.2$'.AND.'$0.4 \leqslant x_2 \leqslant 0.7$'. It follows that class '0' is a disjunction of three conjunctions specified by the boxes a_1, a_2 and a_3. In other words, class '0' is true for a point (x_1, x_2) if and only if '$(x_1, x_2) \in a_1$'.OR.'$(x_1, x_2) \in a_2$'.OR.'$(x_1, x_2) \in a_3$'. Likewise, class '1' is true for a point (x_1, x_2) if and only if '$(x_1, x_2) \in a_4$'.OR.'$(x_1, x_2) \in a_5$'.

The FLR classifier supports at least two different modes of reasoning, namely *Generalized Modus Ponens* and *Reasoning by Analogy*. On the one hand, *Generalized Modus Ponens* is a common form of deductive reasoning whereby, in the context of this work, given both a rule $a_1 \rightarrow c_l$ and an antecedent x such that $x \leqslant a_1$ it follows c_l. Generalized modus ponens is directly supported by FLR. For instance given 'a_2' in Fig. 7(a) there follows class '0'. Moreover, in Fig. 7(b), given both $a_5 \rightarrow 1$ and $b \leqslant a_5$ there follows class '1'. On the other hand, *Reasoning by Analogy* is a mode of approximate reasoning suitable for dealing with incomplete knowledge. More specifically, given a set of rules $a_l \rightarrow c_l$, $l = 1, \ldots, L$ as well as an antecedent a_0, such that $a_0 \leqslant a_l$ for no $l \in \{1, \ldots, L\}$, the FLR classifier selects the rule which best fits the data (a_0) in an Occam razor sense as explained above. An example is illustrated in Fig. 7(c) where box b_1 partially overlaps boxes a_3 and a_4, each one of the latter boxes is mapped to a different class. The FLR classifier responds by calculating the fuzzy degrees of inclusion $k(b_1 \leqslant a_3)$ and $k(b_1 \leqslant a_4)$; finally, box b_1 is assigned to the class label attached to the winner of the competition between a_3 and a_4. A less obvious situation arises in Fig. 7(c) regarding box b_2. Note that a conventional subsethood criterion here fails because b_2 is outside all rule antecedents $a_1, \ldots, a_5$. Nevertheless, using

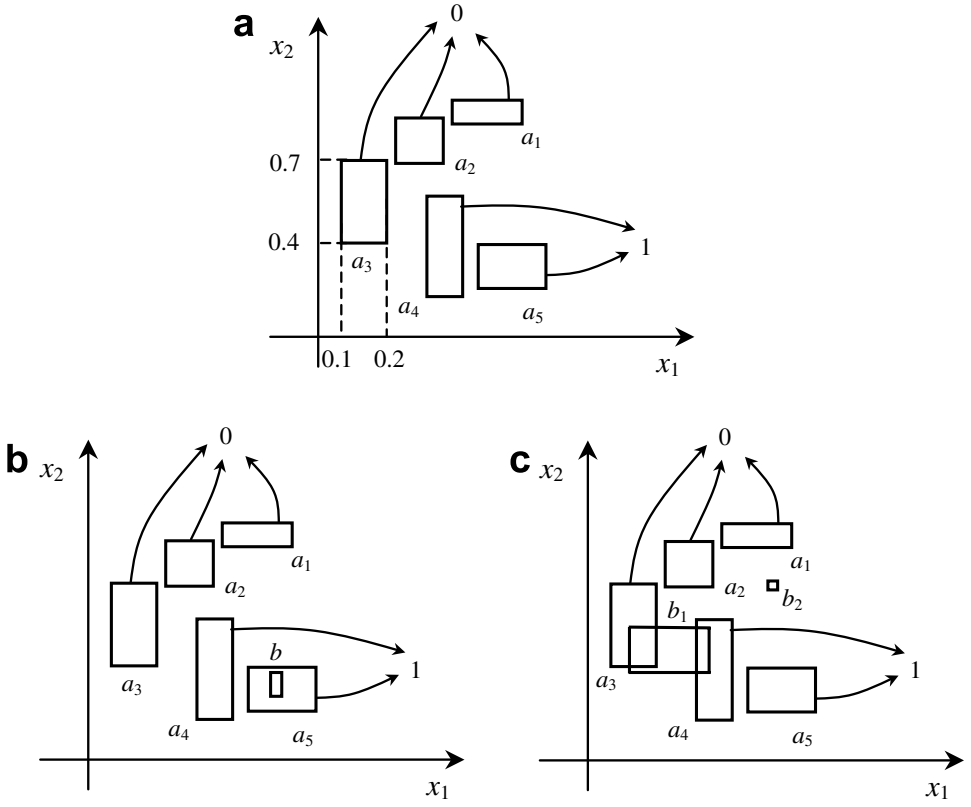

Fig. 7. (a) Class 0 can be regarded as the logical disjunction 'a_1.OR.a_2.OR.a_3' of three conjunctive logical expressions specified by the boxes a_1, a_2 and a_3, respectively. For instance, the conjunctive logical expression a_3 is true for a point (x_1, x_2) if and only if '$0.1 \leqslant x_1 \leqslant 0.2$'.AND.'$0.4 \leqslant x_2 \leqslant 0.7$'. Likewise, class 1 can be regarded as the logical disjunction 'a_4.OR.a_5' of two conjunctive logical expressions specified by the boxes a_4 and a_5, respectively. (b) Since box b is inside box a_5, class label 1 is assigned to box b by *Generalized Modus Ponens* based on the rule $a_5 \rightarrow 1$. (c) Inclusion measure function $k(\cdot,\cdot)$ is used by the FLR classifier to assign 'in principle' a class label to either an overlapping box b_1 or a non-overlapping box b_2.

inclusion measure k, box b_2 is assigned to the best fit, in an Occam razor sense as explained above, among hyperboxes $a_1, \ldots, a_5$.

4.6. Complexity of both training and testing

When a rule $a_i \rightarrow c_i$, $i = 1, \ldots, n$ for training is presented then the fuzzy inclusion measure $k(a_i \leqslant A_l)$, $l = 1, \ldots, L$ is calculated for all L rules in the rule base RB. The worst-case training scenario is to keep 'resetting' all L rules in RB for every input rule. Since both (1) the largest value for L is $L = n$, and (2) a single pass through the data suffices for learning, it follows that the *training complexity* of the FLR classifier is quadratic $\mathcal{O}(n^2)$ in the number n of rules/data for training. Likewise, it can be shown that the *testing complexity* of the FLR classifier is linear $\mathcal{O}(n)$.

5. The physical problem

The FLR classifier was applied in an environmental monitoring problem. This section presents relevant information regarding the physical problem. Experimental results as well as advantages of the FLR classifier are presented in the following section.

Growing public concern, regarding significant increases of air pollutants during the last decades, has spurred governments around the world to establish Air Quality Operational Centers (AQOC) for monitoring ambient air quality by collecting volumes of data from ground station networks. Human expertise is typically employed for real-time decision-making, whereas complex mathematical models are used 'off-line' for more accurate predictions at the expense of long computing times [69]. Computerized decision support could be useful for attaining accurate predictions on-line. In the aforementioned context a useful learning model would need to comply to the following specifications: (1) induction of descriptive, decision-making knowledge (rules) from the data, (2) a capacity for generalization using only a small number of rules, and (3) a capacity to deal effectively with missing data.

5.1. Environmental learning models

Several learning models have been phased in environmental monitoring software systems for predicting air pollution. For instance, neural networks were used for short-term ozone prediction [98,117]; case-based reasoning (CBR) as well as classification and regression trees (CART) were employed for predicting nitrogen dioxide [56]. A computational model for environmental monitoring networks was proposed in [101] using software agents. Along the guidelines of the latter, a multi-agent prototype, namely *distributed NEMO* or *DNEMO* for short, was developed for managing air pollution in Athens, Greece by integrating a number of machine learning and predictor modules [57]. A different multi-agent environmental monitoring system, namely O_3RTAA, was developed lately [5] for ambient air monitoring in Valencia, Spain. The FLR classifier has been accommodated experimentally in O_3RTAA.

5.2. Data acquisition

The data used in this work have been collected in one meteorological station in the vicinity of Valencia, Spain. Eight variables, including both meteorological and air-pollutant variables, have been sampled on a quarter-hourly basis during the year 2001. In all, $4 \times 24 \times 365 = 35{,}040$ data vectors have been available 6020 of which, that is around 17% of the total data vectors, had at least one missing value. The sampled variables as well as their corresponding units are shown in Table 1, where the ozone concentration level is

Table 1
Four air pollutant and three meteorological variables were used for estimating ozone concentration levels

	Data attribute name	Symbol	Data type	Units
1	Sulfur dioxide	SO_2	Real number	$\mu g/m^3$
2	Nitrogen oxide	NO	Real number	$\mu g/m^3$
3	Nitrogen dioxide	NO_2	Real number	$\mu g/m^3$
4	Nitrogen oxides	NO_x	Real number	$\mu g/m^3$
5	Wind velocity	VEL	Real number	m/s
6	Temperature	TEM	Real number	°C
7	Relative humidity	HR	Real number	%
8	Ozone concentration level	O_3	Class label	'low' (0–60 $\mu g/m^3$) 'med' (60–100 $\mu g/m^3$)

labeled either '*low*' or '*med*(*ium*)' for values in the ranges 0–60 μg/m^3 and 60–100 μg/m^3, respectively.

5.3. Specifics

An estimation of ambient ozone concentration from other variables is theoretically feasible based on 'first principles modeling'. For instance, ambient ozone concentration is known to be a function of both nitrogen oxides NO_x [14] and meteorological variables [40]. However, an estimation of ozone-level based on 'first principles modeling techniques' is not straightforward in practice due to the feedback of ozone in the corresponding chemical reaction. It has been demonstrated that estimation of environmental missing data is possible using regression techniques, e.g. linear extrapolation [35]. Nevertheless, conventional regressor models are restricted by *a priori* assumptions including the model structure.

An empirical approach is proposed here for estimating missing ozone measurements directly from the data by classification using the fuzzy lattice reasoning (FLR) classifier. Note that prediction by classification is an acknowledged practice [77]. In particular, we point out that several learning schemes have already been presented for prediction/estimation by classification based on fuzzy lattices. For instance, in [55] estimation of the stapes bone thickness has been effected by applying the dσ-FLN classifier in a lattice stemming from a metric space of linear operators. Furthermore, in [92] prediction of industrial sugar production has been demonstrated by applying the FINkNN classifier in a lattice of fuzzy numbers. In this work the FLR classifier is applied in lattices of hyperboxes towards estimating ozone concentration level.

6. Experimental results and discussion

We have employed the following four classifiers in our experiments:

1. back-propagation neural networks,
2. the fuzzy adaptive resonance theory (fuzzy-ART) classifier,
3. the C4.5 classifier, and
4. the fuzzy lattice reasoning (FLR) classifier.

In a preprocessing step the data attributes have been normalized in the interval $[0, 1]$ by straightforward translation followed by linear scaling using only the training data. Two series of experiments have been carried out: first, using data vectors without missing values and, second, using all data vectors including the ones with missing values. In both series of experiments the data collected from January 1, 2001 until mid June have been used for training, whereas the remaining data until year-end have been used for testing. The corresponding numbers of data vectors in classes 'low' and 'med' are shown in Table 2. Note that 197 and 368 data vectors have been removed from the training data set and the testing data set, respectively, because the aforementioned vectors included missing ozone attribute values.

The corresponding classification results using the four classifiers mentioned above are summarized in Tables 3 and 4, where a Confusion Matrix for each classifier is presented along with the best percentage of correct classifications on the testing data. Since the

Table 2
Numbers of data vectors available

	Data vectors in class		Total
	'low'	'med'	
Without missing values			
for training	6865	4761	11,626
for testing	12,256	5138	17,394
With missing values			
for training	9472	6074	15,546
for testing	13,483	5446	18,929

Table 3
Confusion matrices for the testing data for four classifiers after removing all vectors with missing data values. A total number of 17,394 data vectors were used for testing

Classifier name	Percentage of the testing data classified as:	
	'low'	'med'
Back-propagation neural network		
Percentage of the testing data in class 'low'	56.95	13.52
Percentage of the testing data in class 'med'	4.34	25.20
Overall success	82.15%	
Total number of induced rules	Not applicable	
Fuzzy-ART		
Percentage of the testing data in class 'low'	54.16	16.30
Percentage of the testing data in class 'med'	9.25	20.29
Overall success	74.45%	
Total number of induced rules	100	
C4.5 decision tree		
Percentage of the testing data in class 'low'	48.79	21.67
Percentage of the testing data in class 'med'	4.59	24.95
Overall success	73.74%	
Total number of induced rules	131	
FLR classifier (applied in $[0,1]^N$ for $N = 7$)		
Percentage of the testing data in class 'low'	64.64	5.82
Percentage of the testing data in class 'med'	10.95	18.59
Overall success	83.23%	
Total number of induced rules	3	
FLR classifier (applied in $\mathbb{R}^N$ for $N = 7$)		
Percentage of the testing data in class 'low'	65.02	5.55
Percentage of the testing data in class 'med'	9.23	20.20
Overall success	85.22%	
Total number of induced rules	3	

Table 4
Confusion Matrices for the testing data for two classifiers including vectors with missing data values. A total number of 18,929 data vectors were used for testing

Classifier name	Percentage of the testing data classified as:	
	'low'	'med'
C4.5 decision tree		
Percentage of the testing data in class 'low'	55.58	15.65
Percentage of the testing data in class 'med'	6.79	21.98
Overall success	77.56%	
Total number of induced rules	44	
FLR classifier (applied in $[0,1]^N$ for $N = 7$)		
Percentage of the testing data in class 'low'	68.56	2.67
Percentage of the testing data in class 'med'	12.74	16.04
Overall success	84.60%	
Total number of induced rules	19	
FLR classifier (applied in $\mathbb{R}^N$ for $N = 7$)		
Percentage of the testing data in class 'low'	68.61	3.17
Percentage of the testing data in class 'med'	11.30	16.92
Overall success	85.53%	
Total number of induced rules	19	

Table 5
Training and testing % classification accuracy using back-propagation neural networks (only data without missing values have been considered)

No. of hidden neurons	% classification accuracy	
	Training set	Testing set
Linear hidden layer		
3	75.24	78.37
5	75.68	80.32
7	76.33	74.07
9	76.79	73.01
11	**74.62**	**82.15**
13	76.80	72.95
15	77.48	74.34
17	76.43	72.45
19	77.51	72.56
Sigmoid hidden layer		
3	66.69	78.79
5	70.22	68.03
7	**68.68**	**82.06**
9	76.60	70.28
11	74.43	78.00
13	76.89	76.79
15	63.89	42.48
17	75.92	74.42
19	77.42	72.86

back-propagation as well as fuzzy-ART neural networks cannot deal directly with missing data values, classification results by both back-propagation and fuzzy-ART appear only in Table 3. Both Tables 3 and 4 also show the corresponding total number of induced rules. Details regarding the application of each classifier are presented next followed by a comparative classifier discussion.

6.1. Back-propagation neural networks

Many experiments have been carried out on various network architectures using different neuron activation functions as well as different numbers of hidden neurons. Table 5 summarizes the experimental results for selected parameter values. The transfer function in the hidden layer was either linear or sigmoid. The total number of hidden layer neurons varied from 3 to 19 in steps of 2. The transfer function for the output layer was always sigmoid, furthermore training was carried out using the resilient back-propagation algorithm with mean square error (MSE) target 0.01 and maximum number of training epochs 1500. A best classification accuracy of 82.15% has been attained on the testing data by a neural network with linear transfer functions including 11 hidden layer neurons. A comparable best classification accuracy of 82.06% has been attained by another neural network with sigmoid transfer functions including 7 hidden layer neurons.

6.2. The fuzzy adaptive resonance theory (fuzzy-ART) classifier

A standard fuzzy-ART classifier was employed as a typical representative of algorithms which handle hyperboxes. The fuzzy-ART was applied in the 7-dimensional unit hypercube for various values of its *vigilance parameter* (ρ). The corresponding classification accuracies are summarized in Table 6.

For parameter ρ values in the range 0.30–0.70 a constant number of 100 rules was induced. For larger values of ρ the number of rules increased exponentially. In the latter case the training- and testing-classification accuracies remained around 47% and 59%, respectively.

Table 6
Training and testing classification accuracy by the fuzzy-ART classifier. The corresponding numbers of induced rules are also shown (only data without missing values have been considered)

Vigilance parameter ρ	% classification accuracy		No. of induced rules
	Training set	Testing set	
0.30	41.48	40.57	100
0.35	**54.96**	**74.45**	**100**
0.40	48.56	64.54	100
0.45	44.01	46.29	100
0.50	41.85	40.23	100
0.55	41.13	41.93	100
0.60	52.52	73.12	100
0.65	47.39	48.52	100
0.70	44.40	41.07	100

6.3. The C4.5 classifier

The C4.5 classifier has been employed on a standard software platform (WEKA platform [115]), for generating decision trees (DT) in which the internal nodes specify inequalities for the values of environmental attributes, moreover the tree leaves specify an output class. Initially, the C4.5 classifier has been applied on the data without missing values, without pruning, resulting in a DT of height 697 with 1393 rules; the corresponding classification accuracy on the training set reached 94.8%, whereas on the testing set it was only 64.85% (Table 7). Obviously the latter DT overfits the training data, therefore two pruning methods were employed: (1) Confidence Factor Pruning (CFP), and (2) Reduced Error Pruning (REP). Results are shown in Table 7 for selected pruning parameter values. On the one hand, a best classification performance of 73.74% on the testing data has been recorded for REP and 100 folds. The corresponding induced DT height has been 66 including 131 leaves (rules). On the other hand, the application of CFP reduced the classification accuracy on the testing data to around 67%.

Table 7
Training and testing % classification accuracy using decision trees (C4.5 algorithm). Both the tree height and the number of rules are also shown

Pruning method	Parameter value	% classification accuracy		Tree height	No. of rules (tree leaves)
		Training set	Testing set		
Results on the data without missing values					
Unpruned	–	94.8	64.85	697	1393
Confidence factor pruning parameter: CF	0.10	91.33	67.31	288	575
	0.20	92.87	66.71	412	823
	0.30	93.92	67.40	528	1055
	0.40	94.10	67.39	551	1101
	0.50	94.31	67.19	585	1169
Reduced error pruning parameter: no. of folds	2	89.31	63.71	254	507
	10	89.01	71.85	233	465
	50	85.05	60.62	126	251
	100	**83.33**	**73.74**	**66**	**131**
	300	81.55	69.98	38	75
	500	77.73	72.48	16	31
Results on all the data including data with missing values					
Unpruned	–	92.18	58.67	1595	798
Confidence factor pruning parameter: CF	0.10	89.14	60.26	557	279
	0.20	89.98	59.19	735	368
	0.30	90.81	59.44	925	463
	0.40	91.37	59.30	1083	542
	0.50	91.59	59.32	1195	598
Reduced error pruning parameter: no. of folds	2	88.14	64.91	635	318
	10	88.28	59.19	575	288
	50	85.44	60.17	287	144
	100	84.01	61.36	167	84
	300	**82.48**	**77.56**	**87**	**44**
	500	81.33	70.19	63	32

In the following, the C4.5 classifier has been applied on all the data (including data with missing values), without pruning, resulting in a DT of height 1595 with 798 rules; the corresponding classification accuracy on the training data set has been 92.18%, whereas on the testing set has been only 58.67% (Table 7). Apparently the latter DT overfits the training data; hence the two pruning methods (CFP and REP) have been employed again resulting in best classification performances of 60.26% and 77.56%, respectively, on the testing data (Table 7). Note that CFP has significantly deteriorated the classification accuracy results on the testing data to around 59%.

In our experiments REP has produced, in general, better classification results than CFP on both testing data sets. The latter can be attributed to an enhanced capacity of REP for generalization implied by a smaller number of induced rules. The slightly deteriorated performance of the REP on the testing data without missing values compared to testing data with missing values, i.e. 73.74% versus 77.56%, can be attributed to the fact that in the former case fewer data vectors have been available for training.

6.4. The fuzzy lattice reasoning (FLR) classifier

The dimensionless *vigilance parameter* $\rho_{\text{crit}} = N/(N + D_{\text{crit}}) \iff D_{\text{crit}} = N(1 - \rho_{\text{crit}})/\rho_{\text{crit}}$ is more convenient than D_{crit} because ρ_{crit} varies in the interval $[0.5, 1]$ for any number of dimensions N. Therefore in the following experiments parameter ρ_{crit} has been employed instead.

First, experiments with the FLR classifier have been carried out in the 7-dimensional unit hypercube on normalized data using positive valuation function $v_i(x) = x$ and isomorphic function $\theta_i(x) = 1 - x$, $i = 1, \ldots, 7$ in a constituent lattice $[0, 1]$. No *a priori* rules have been used, i.e. initially the rule base RB has been empty. Learning took place incrementally in the order the data had been sampled. In all experiments, rules were induced in a single pass through the training data. The classification accuracies on both the training and the testing data sets as well as the corresponding number of induced rules are shown in Table 8 for selected values of ρ_{crit}. The fairly small number of induced rules in Table 8

Table 8
Training and testing classification accuracy by the FLR classifier applied on normalized data in the unit hypercube $[0,1]^N$ for $N = 7$. The corresponding numbers of induced rules are also shown

ρ_{crit}	% classification accuracy		No. of induced rules
	Training set	Testing set	
Results on the data without missing values			
0.50	59.16	70.46	2
0.60	**64.73**	**83.23**	**3**
0.70	73.68	74.85	20
0.80	67.43	72.59	139
Results on all the data including data with missing values			
0.50	60.99	71.22	5
0.60	60.99	71.22	8
0.70	**63.48**	**84.60**	**19**
0.80	69.00	66.54	43

Table 9
Three rules induced from the training data (without missing values) by the FLR classifier

Rule no.	SO_2	NO	NO_2	NO_x	VEL	TEM	HR	O_3 class
1	IF [3.0, 87.0]	& [2.0, 74.0]	& [4.0, 57.0]	& [6.0, 151.0]	& [0.1, 9.4]	& [4.0, 28.6]	& [8.0, 99.0]	THEN 'low'
2	IF [3.0, 47.0]	& [2.0, 24.0]	& [4.0, 36.0]	& [6.0, 54.0]	& [0.1, 11.1]	& [5.0, 35.0]	& [8.0, 99.0]	THEN 'med'
3	IF [3.0, 52.0]	& [2.0, 89.0]	& [4.0, 65.0]	& [6.0,176.0]	& [0.1, 7.5]	& [9.0, 35.0]	& [24.0, 99.0]	THEN 'low'

implies that FLR's capacity for 'Type I generalization' has resulted in a quite thorough coverage of the input data space. Again, the slightly better classification accuracy on the data set with missing values (compared to the data set without missing values), i.e. 84.60% versus 83.23%, can be attributed to the larger number of training data vectors in the former data set. Table 9 shows the three rules induced from the training data without missing values, moreover Table 10 displays explicitly rule no. 1 as a conjunctive 'if–then' statement.

Second, further experiments were carried out with the FLR classifier on non-normalized data using positive valuation function $v_i(x) = 1/(1 + \exp(-\lambda(x - x_m)))$ and isomorphic function $\theta_i(x) = 2x_m - x$ in each constituent lattice R, where $\lambda = \varsigma/(x_{\max} - x_{\min})$ with $\varsigma > 0$, $x_m = (x_{\min} + x_{\max})/2$, and $x_{\min}$ and $x_{\max}$ are the minimum and maximum attribute values in the corresponding constituent lattice of the training data. The classification accuracies on both the training and the testing data sets as well as the corresponding numbers of induced rules are shown in Table 11 for selected values of the parameters ς and ρ_{crit}. In this case the best testing classification accuracies for the data without- and with-missing values have been 85.22% and 85.53%, respectively. Therefore, it appears that an application of classifier FLR in R^N (on non-normalized data) rather than an application in I^N (on normalized data) tends to improve performance. The aforementioned improvement can be attributed, in part, to deterioration in performance due to truncation of some testing data when normalization is employed, whereas no truncation is necessary when classifier FLR is applied in R^N. Another reason for the aforementioned improvement is the employment of (sigmoid) non-linearities in R^N as detailed below.

A series of experiments has been carried out to confirm FLR's capacity for 'Type II generalization', i.e. generalization beyond a rule's core. The corresponding results are summarized in Table 12. More specifically, Table 12 shows that only a small percentage of the testing data lie outside any rule (hyperbox) core. The latter was expected in this application since the small number of induced rules here covers quite thoroughly the input data space as explained above. Table 12 shows that when data with missing attribute values have been left out then only 68 (out of a total of 469 testing data outside all rule hyperbox cores) have been classified correctly; whereas when data with missing attribute values have been used then as many as 497 (out of a total of 520 testing data outside all rule hyperbox cores) have been classified correctly. The aforementioned difference in performance was attributed to the different numbers of induced rules, i.e. 3 versus 19 rules have been induced using data without- and with-missing attribute values, respectively, as shown in Table 8.

Table 10
Rule no. 1 of Table 9 corresponds to the following "if–then" statement

IF	SO_2 is in the range [3.0, 87.0] $\mu g/m^3$ AND ...
	NO is in the range [2.0, 74.0] $\mu g/m^3$ AND ...
	NO_2 is in the range [4.0, 57.0] $\mu g/m^3$ AND ...
	NO_x is in the range [6.0, 151.0] $\mu g/m^3$ AND ...
	VEL is in the range [0.1, 9.4] m/s AND ...
	TEM is in the range [4.0, 28.6] °C AND ...
	HR is in the range [8.0, 99.0] %
THEN	O_3 is in the class 'low'

Table 11
Training and testing classification accuracy by the FLR classifier on non-normalized data for various values of parameters ς and ρ_{crit}. The corresponding numbers of induced rules are also shown

ς	ρ_{crit}	% classification accuracy		No. of induced rules
		Training set	Testing set	
Results on the data without missing values				
1	0.5	59.16	70.46	2
	0.6	59.16	70.46	2
	0.7	59.16	70.46	2
	0.8	**62.73**	**85.22**	**3**
5	0.5	59.16	70.46	2
	0.6	65.40	82.70	3
	0.7	70.48	79.64	19
	0.8	67.53	78.72	40
10	0.5	59.16	70.46	2
	0.6	64.27	83.43	3
	0.7	65.77	74.89	34
	0.8	69.56	82.87	115
15	0.5	59.16	70.46	2
	0.6	64.73	83.24	3
	0.7	68.85	78.88	23
	0.8	70.39	81.54	112
Results on the data with missing values				
1	0.5	60.99	73.37	2
	0.6	60.99	73.37	2
	0.7	60.99	73.37	3
	0.8	60.99	73.37	4
5	0.5	60.99	71.22	4
	0.6	60.99	71.22	6
	0.7	60.99	71.23	9
	0.8	**65.34**	**85.53**	**19**
10	0.5	60.99	71.22	6
	0.6	60.99	71.23	9
	0.7	60.99	71.22	14
	0.8	63.55	82.55	26
15	0.5	60.99	71.22	6
	0.6	60.99	71.23	10
	0.7	60.99	71.23	17
	0.8	64.00	82.59	31

The results in Table 12 imply that the improved performance of 'classifier FLR applicable in R^N' compared to 'classifier FLR applicable in the unit-hypercube' is mainly due to the (sigmoid) non-linearities introduced based on lattice theory. More specifically, a truncation of the testing data, does not appear to be able to deteriorate classification accuracy

Table 12
Demonstrating the capacity of classifier FLR for 'type II generalization'

	No. of data correctly classified	No. of data incorrectly classified	Total
Results on the testing data without missing values			
No. of data inside a rule core	14,410	2515	16,925
No. of data outside any rule core	68	401	469
Total	14,478	2916	17,394
Results on the testing data including data with missing values			
No. of data inside a rule core	15,516	2893	18,409
No. of data outside any rule core	497	23	520
Total	16,013	2916	18,929

significantly in this problem due to the small number of data outside any rule (hyperbox) core as shown in Table 12.

Additional computational experiments were carried out using different permutations of the training data in order to test FLR's sensitivity to different orders of data presentation. No significant deviations were observed from the aforementioned results. In particular, the testing data classification accuracy did not change more than 2 percentage points, whereas the number of induced rules did not change more than 2. Hence, the performance of classifier FLR has remained roughly the same in this classification problem. Nevertheless, it should be pointed out that a different order of presenting the data (than the original order the data were sampled in this real-world problem) is meaningless here.

6.5. Implementation details and comparative classifier discussion

The processing times for training/testing each classifier are shown in Table 13. All the experiments have been carried out on a workstation Pentium 4 processor at 1.5 GHz and 768 MB of RAM. More specifically, training and testing experiments with back-propagation networks have been performed using MATLAB version 5.3. Experiments with the fuzzy-ART were carried out using a standard MATLAB implementation downloaded from the Web. Experiments with C4.5 have been carried out using the J48 algorithm on the Waikato Environment for Knowledge Analysis (WEKA) [115]. The FLR classifier has already been included in WEKA platform version 3-4-2. Table 13 shows that the FLR classifier has been clearly faster than the C4.5 classifier furthermore the FLR classifier has been orders of magnitude faster than both fuzzy-ART and back-propagation neural networks in this application due, mainly, to both the simpler activation functions and

Table 13
Training/testing times for four classifiers

Classifier name	Training/testing time
1 Back-propagation neural networks	Between 3 min and 25 min
2 Fuzzy-ART classifier	Between 6 min and 9.5 min
3 C4.5 classifier	Around 6.5 s
4 FLR classifier	Around 1.5 s

the smaller number of rules induced by classifier FLR as well as to the low training complexity $\mathcal{O}(n^2)$ of classifier FLR as explained in Section 4.6.

It is worth pointing out that around 71% of the data vectors in a testing data set were in the class 'low' ozone concentration level; hence a random selection of category 'low' would result in a classification accuracy of around 71%. Therefore, an acceptable classifier is required to perform above 71% in this application. It turned out that back-propagation neural networks performed well in the experiments (82.51%) but without inducing descriptive knowledge (rules), moreover back-propagation neural networks have been slow and they could not deal with missing data values. The fuzzy-ART classifier, that is a typical hyperbox handling classifier, performed poorly (74.45%) in this application, moreover it induced a fairly large number (100) of rules; in addition, the fuzzy-ART classifier could not deal with missing data values. The C4.5 classifier could deal with missing data values, nevertheless it yielded a large number of rules (i.e. 131 and 44 rules, respectively, for data without- and with-missing values), and it performed only slightly better than random selection (i.e. 73.74% and 77.56%, respectively, for data without- and with-missing values). The FLR classifier has demonstrated a good capacity for generalization (i.e. 85.22% and 85.53%, respectively, for data without- and with-missing values), it induced considerably fewer rules than the C4.5 classifier (i.e. around 3 and 19 rules, respectively, for data without- and with-missing values), and it could deal with missing data values. In conclusion, the FLR classifier has outperformed the C4.5 classifier, the fuzzy-ART as well as the back-propagation neural networks in this real-world problem. The FLR classifier also demonstrated a capacity for generalization beyond rule (hyperbox) core. Another significant advantage of the FLR classifier includes faster training in a single pass through the training data. An additional, substantial advantage of the FLR classifier is its capacity to employ tunable (sigmoid) non-linearities for improving performance.

The experiments presented in this section were not meant for optimal parameter estimation. Rather, the experiments here were only meant for demonstrating comparatively the capacities of different classifiers for a range of parameter values. It turns out that classifier FLR in R^N performs better than any other classifier in this application. Note that had optimal parameter estimation been sought based only on the 'training data classification accuracies' displayed in Tables 5–8, and 11 then, clearly, classifier FLR prevails. In particular, classifier FLR applied on non-normalized data (Table 11) produced 79.64% and 85.53% testing classification accuracies on the data 'without missing values' and 'with missing values', respectively, for optimal parameter values $(\varsigma, \rho_{\text{crit}}) = (5, 0.7)$ and $(\varsigma, \rho_{\text{crit}}) = (5, 0.8)$ induced from maximum classification accuracies 70.48% and 65.34%, respectively, on the training data. The aforementioned percentages, i.e. 79.64% and 85.53%, are clearly higher than the corresponding percentages for classifiers back-propagation, fuzzy-ART, and C4.5 shown in Tables 5–7, respectively.

7. Connections with related work

The FLR is a rule-based classifier. A number of rule-based systems have been proposed in the literature in various contexts including decision trees, machine learning, fuzzy inference systems (FIS), (fuzzy) neural networks, etc. [2,41,44,73,74,96,100,109]. This section presents similarities and differences between the FLR classifier and other, rule-based systems. Additional, interesting connections are also shown.

The FLR classifier bears substantial similarities, as well as differences, with Decision Trees (DTs) for rule-induction [94–96] – note that the term *constituent lattice* used in the FLR classifier corresponds to the term *attribute* used in a DT. Both the FLR and a DT induce rules from the training data by partitioning the latter conditionally. However, a DT carries out a partition of the training data by a divide-and-conquer, batch-processing procedure based on an information-theoretic criterion; whereas, the FLR carries out a partition of the training data 'on-line' based on both a fuzzy inclusion measure and a maximum threshold criterion. Moreover, a rule induced by a DT typically consists of a variable number of tests carried out sequentially; whereas, a rule induced by FLR consists of a fixed number of tests, which can be carried out in parallel. Furthermore, it may appear that the lattice-ordering relation, assumed explicitly in a data domain by the FLR classifier, is too restrictive in comparison with a DT since the latter does not seem to impose any similar restrictions. Nevertheless a 'test', carried out by a DT in an attribute, typically employs a partial ordering relation, e.g. a relation '$\leqslant$' or a relation '$\subseteq$'. Hence, a partial ordering relation is assumed 'implicitly' in the data by a DT.

Compared to other classifiers note that the FLR classifier explicitly interprets a training datum (a_i, c_i) as a rule: $a_i \rightarrow c_i$. Hence, the FLR classifier deals with rules all along. It might be interesting to point out that an employment of rules has been proposed for unifying instance-based and rule-based induction in a machine-learning context [21]. In the aforementioned sense the FLR classifier may be interpreted as an instance-based classifier; in particular, the FLR classifier does not retain (lattice-ordered) instances but rather it generalizes them by the lattice-join operation. Likewise, the FLR classifier may be interpreted as a Case-Based Reasoning (CBR) classifier [67].

The FLR classifier, based on a lattice inclusion measure, operates similar to inclusion-based- or similarity-based-reasoning [16,23,107]. More specifically, it is known that 'in parallel to the mainstream approach to approximate reasoning based on CRI (Compositional Rule of Inference) an extensive body of literature employs *reasoning by analogy* such that similar inputs imply similar outputs' [16]. On the one hand, a similarity-measure or an inclusion-measure in the literature typically involves fuzzy sets. On the other hand, a lattice inclusion measure here can, more generally, compute a (fuzzy) degree of inclusion of a lattice element to another one. In particular, this work has computed a degree of inclusion of a hyperbox (rule antecedent) to another one. However, the rule consequents employed by the FLR classifier are (simply) class labels. Therefore, the full potential of a lattice inclusion measure remains to be fulfilled in a future work involving more sophisticated rule consequents, for instance in either approximate reasoning or case based building applications [16,107].

Of further interest is relevant work in machine-learning regarding rule induction including a combination of general-to-specific and specific-to-general learning [20]. Note that the emphasis of this work is in a novel classifier with such capacities as introduction of tunable non-linearities and fast processing of large data sets as demonstrated above. Such capacities do not usually characterize popular rule induction classifiers. Nevertheless, certain procedural practices of the latter classifiers can be considered creatively in future extensions of the FLR classifier.

Another related work regards induction of fuzzy graphs/rules from examples towards function approximation. For instance, the work in [7] presents an efficient algorithm for inducing locally independent fuzzy rules from examples. Note also that there is a substantial body of related work regarding (neuro-) fuzzy systems including (AN)FIS,

NEFCLASS, Mamdani-/Sugeno-type fuzzy rule-based systems, etc. [44,74,80–84,109]. It should be pointed out that, lately, a number of publications have proposed improvements in conventional FISs by introducing non-linear metrics based on lattice theory [48,49]. Moreover, the problem of structure identification in FIS design was dealt with effectively using a granular extension of Kohonen's self-organizing map based on lattice theory [50]. Furthermore, the problem of 'learning' in linguistically-interpreted FIS has been formulated as a function approximation problem based on lattice theory [47]. The FLR classifier here represents a simple (AN)FIS for inducing rules involving fuzzy sets with hyperbox-shaped cores. Comparative advantages of the FLR classifier include both fast data processing and the capacity to introduce tunable non-linearities as demonstrated above.

A rule induced by the FLR classifier has been interpreted here as a logical conjunction (logic AND) involving the constituent lattices. Since more than one rule may imply the same class there follows a logical disjunction (logic OR) of several conjunctions as it was shown in Section 4.5. Note that logical conjunctions and/or disjunctions have been studied in different learning contexts including the Probably Approximately Correct (PAC) learning framework [33,59,113]. Nevertheless, the latter publications deal solely with crisp Boolean variables, whereas a variable in a logical form here assumes values in the interval $[0,1]$. Hence, the vector of all fuzzified variables here corresponds to a point in the unit hypercube as in [68]. We point out that an employment of fuzzy AND/OR rules is not an innovation. For instance, fuzzy logic-based neural networks have been proposed in [73].

There are at least three substantial differences between the FLR classifier and other rule-based systems as explained in the following. First, a positive valuation can introduce tunable non-linearities in order to improve classification accuracy on the testing data as demonstrated experimentally above. Second, the FLR classifier retains a capacity for generalization beyond rule (hyperbox) core; more specifically, based on a positive valuation function it is possible to calculate a (fuzzy) degree of rule activation for a datum beyond a rule's (hyperbox') core. Third, the FLR classifier is applicable in a general lattice data domain including both complete and non-complete lattices such as the unit-hypercube $[0,1]^N$ and the Euclidean space R^N, respectively. All aforementioned differences are based on an explicit employment of lattice theory. Note also that a *linear* positive valuation function $v_i(x) = a_i x$, $i = 1,\ldots,N$ in a constituent lattice $\mathsf{L}_i = \mathsf{R}$, $i = 1,\ldots,n$ can be interpreted as a *weight* for the corresponding dimension. Even though the use of weights in a classifier is not an innovation [10,73], nevertheless the FLR classifier can employ a 'weight function' instead of employing a 'weight (constant)' in a data dimension.

Most of all, the FLR classifier bears similarities with classification algorithms, which handle hyperboxes in the N-dimensional space. Such algorithms include min–max neural networks [27,102,103] as well as adaptive resonance theory neural networks (ART) [3,34,70,85,110]. Moreover, in a machine-learning context, the class of axis-parallel rectangles has been shown to be efficiently *probably approximately correct* (PAC) learnable [9,60,72]. Further machine learning applications have also considered hyperboxes as instruments for learning [19,99,114]. Learning lattice intervals beyond space R^N is carried out implicitly in [113] where conjunctive normal forms (CNF) are computed in a Boolean lattice. Nevertheless the aforementioned learning schemes fail to employ lattice theory explicitly for their benefit. This work employs lattice theory for improving classification performance in practice. For the interested reader an employment of lattice theory in computational intelligence is delineated next.

Lattice theory has been employed by various authors in fuzzy logic applications [25,28]. Lattices have been employed for monitoring rule execution [6] as well as in information retrieval [12,93]. In machine-learning lattices are widely known since their employment in version spaces [77]. Lattice theory has been a popular tool in mathematical morphology, especially in digital image processing applications [22,31,36,76]. In addition, lattice theory has been employed for modeling the operation of neurons in a neural network [97,108]. Lattices have also been useful for knowledge representation in various contexts [1,29,106]. *Fuzzy lattices* have emerged from conventional (crisp) lattices by fuzzifying the corresponding ordering relation using an *inclusion measure* function $\sigma : \mathsf{L} \times \mathsf{L} \rightarrow [0,1]$ as shown above. An inclusion measure σ should not be confused with a *fuzzy measure*. The latter is a real function $g : \mathsf{F} \rightarrow [0,1]$ defined on a collection F of fuzzy sets; it follows that fuzzy measures subsume *probability measures* as a special case [64]. It turns out that an inclusion measure relates to the notion of *fuzzy subsethood indicator* [11,15,26,86,104,118]. Nevertheless an inclusion measure σ is more general since it applies to any lattice, not only to a lattice of (fuzzy) sets. For details regarding an employment of (fuzzy) lattice theory in computational intelligence the reader may refer to [47].

8. Conclusion and future work

A new classifier was presented in this work, namely fuzzy lattice reasoning (FLR) classifier, which can induce rules in a mathematical lattice data domain such that a rule antecedent corresponds to a lattice interval, moreover a rule consequent is a class label. The emphasis here has been in applications in lattice R^N, where a lattice interval corresponds to a N-dimensional hyperbox. Lattice theory enabled the introduction of useful non-linearities. Advantages of the FLR classifier, in comparison with other algorithms from the literature, have been presented extensively.

The FLR classifier was applied here in a environmental monitoring problem for ambient ozone estimation. The results have compared favorably with the results obtained by C4.5 decision trees, a conventional hyperbox handling classifier namely fuzzy-ART, as well as back-propagation neural networks. More specifically, the FLR classifier has demonstrated comparatively a capacity for fast learning, a good capacity for generalization on the testing data, it induced few rules in a single pass through the training data, it dealt effectively with missing data values, moreover it demonstrated a capacity for tunable, non-linear generalization beyond rule (hyperbox) core.

This work has demonstrated the potential of the rule-based classifier FLR. However, an optimization of the operation of classifier FLR remains a topic for future work. Note that, apart from the vigilance parameter, only the parameter values in a sigmoid positive valuation function have been optimized here. Future work will seek an optimization of the FLR classifier as described in the following.

First, even though the classification accuracy of FLR classifier has been fairly stable in this work for different orders of data presentation, there is no guarantee that it will remain stable in other classification problems. There is experimental evidence that an ensemble of voter classifiers can both stabilize and improve the classification accuracy of an individual voter classifier [63]. Therefore, the development of a 'voting FLR' classifier will be pursued in the future. Second, rule (hyperbox) overlapping can be contained using genetic algorithms 'likewise' as in conventional fuzzy system design where it was treated as a constraint in a multiple constraint satisfaction problem [84]. Third, for a large number

of rules, an employment of standard database indexing techniques [38,65] could reduce time complexity in practice. Additional types of optimization may also be pursued. Finally, an interesting future work could interpret the lattice inclusion measure presented in this work as an equality relation towards the development of approximate reasoning optimization techniques.

Acknowledgements

The authors thankfully acknowledge both Eibe Frank and Mark Hall for including the FLR classifier in the WEKA platform. The dataset used in the experiments has been a courtesy of the Fundación Centro de Estudios Ambietales del Mediterráneo (CEAM) and IDI-Eikon, Valencia, Spain. This work has been partially supported by the third European framework programme: Operational Programme in Education and Initial Vocational Training II, under project Archimedes I contract no. 04-3-001/1 as well as by a European Commission IST Programme under research contract no. IST-2000-31050.

Appendix

Useful definitions are listed in this Appendix followed by the proofs of propositions.

Definition A1. A *partially ordered set* (*poset*) is a set P in which a binary relation $x \leqslant y$ is defined, which satisfies the following conditions for all $x, y, z \in P$:

P1. $x \leqslant x$ *Reflexive*

P2. $x \leqslant y$ and $y \leqslant x \Rightarrow x = y$ *Antisymmetry*

P3. $x \leqslant y$ and $y \leqslant z \Rightarrow x \leqslant z$ *Transitivity*

Definition A2. A *metric* in a non-empty set S is real function $d : S \times S \to \mathrm{R}$ satisfying

M0. $d(x, y) \geqslant 0$,

M1. $d(x, y) = 0 \Longleftrightarrow x = y$,

M2. $d(x, y) = d(y, x)$, and

M3. $d(x, z) \leqslant d(x, y) + d(y, z)$ – The *Triangle Inequality*.

for $x, y, z \in S$.

Definition A3. Let P and Q be partially ordered sets. A mapping $\psi : P \to Q$ is called

(i) *Order-preserving* (or, alternatively, *monotone*), if $x < y$ in P implies $\psi(x) < \psi(y)$ in Q.
(ii) *Order-isomorphism* (or, simply, *isomorphism*), if both '$x < y$ in $P \iff \psi(x) < \psi(y)$ in Q' and 'ψ is onto Q'.

When there is an isomorphism from P to Q, then P and Q are called *isomorphic*, symbolically $P \cong Q$.

Proposition 3. *If* $\sigma : \mathsf{L} \times \mathsf{L} \to [0,1]$ *is an inclusion measure on lattice* L, *then* $\langle \mathsf{L}, \sigma \rangle$ *is a fuzzy lattice.*

Proof. Given an inclusion measure $\sigma : \mathsf{L} \times \mathsf{L} \to [0,1]$ we need to prove the equivalence $x \leqslant y \iff \sigma(x,y) = 1$.

(1) In the one direction let $x \leqslant y$.
From conditions C1 and C2 it follows $x \leqslant y \Rightarrow \sigma(x,x) \leqslant \sigma(x,y) \Rightarrow 1 \leqslant \sigma(x,y) \Rightarrow \sigma(x,y) = 1$.
(2) In the other direction let $\sigma(x,y) = 1$.
Consider the following three mutually exclusive and exhaustive cases (i) $x \leqslant y$, (ii) $y < x$, and (iii) $x \| y$.
In the following we reject cases (ii) and (iii).
(ii) Let $y < x$. Then, based on condition C3, it follows $y < x \Rightarrow x \wedge y < x \Rightarrow \sigma(x,y) <$ 1-contradiction.
(iii) Let $x \| y$. The latter implies "$x \wedge y < x < x \vee y$".AND. "$x \wedge y < y < x \vee y$". Based on conditions C3 and C3′ it follows

$$x \| y \Rightarrow \left\{ \begin{array}{l} x \wedge y < x < x \vee y \Rightarrow x \wedge y < x \\ x \wedge y < y < x \vee y \Rightarrow y < x \vee y \end{array} \right\} \Rightarrow \sigma(x,y) < \text{1-contradiction.}$$

Therefore we have to accept case (i) $x \leqslant y$. □

Proposition 4. *If* L *is a* (*complete*) *lattice and* $v : \mathsf{L} \to \mathsf{R}$ *is a positive valuation* (*with* $v(O) = 0$) *then* (1) $k(x,u) = \frac{v(u)}{v(x \vee u)}$, *and* (2) $s(x,u) = \frac{v(x \wedge u)}{v(x)}$ *are inclusion measures.*

Proof

(C0) Only for complete lattices: $k(x,O) = \frac{v(O)}{v(x \vee O)} = 0,\ x \neq O \quad s(x,O) = \frac{v(x \wedge O)}{v(x)} = 0,\ x \neq O.$
(C1) $k(x,x) = \frac{v(x)}{v(x \vee x)} = 1. \quad s(x,x) = \frac{v(x \wedge x)}{v(x)} = 1.$
(C2) The truth of $u \leqslant w \Rightarrow k(x,u) \leqslant k(x,w)$ is known from [51].
The truth of $u \leqslant w \Rightarrow s(x,u) \leqslant s(x,w)$ is known from [91].

(C3) $$x \wedge y < x \Rightarrow \left\{ \begin{array}{l} y < x \Rightarrow y < x = x \vee y \\ y \| x \Rightarrow x \wedge y < y < x \vee y \end{array} \right\} \Rightarrow k(x,y) = \frac{v(y)}{v(x \vee y)} < 1.$$

$$x \wedge y < x \Rightarrow \left\{ \begin{array}{l} y < x \Rightarrow x \wedge y = y < x \\ y \| x \Rightarrow x \wedge y < x < x \vee y \end{array} \right\} \Rightarrow s(x,y) = \frac{v(x \wedge y)}{v(x)} < 1. \quad \square$$

Proposition 5. *Let* L_i *be a totally-ordered lattice, let* $v : \mathsf{L}_i \to \mathsf{R}$ *be a positive valuation, and let* $\theta : \mathsf{L}_i^{\partial} \to \mathsf{L}_i$ *be an isomorphic function in* L_i. *Then a positive valuation function* $v : \tau(\mathsf{L}_i) \to \mathsf{R}$ *is given by* $v([a,b]) = v(\theta(a)) + v(b)$.

Proof. First, we have to show that $v([a,b]) + v([c,d]) = v([a,b] \wedge [c,d]) + v([a,b] \vee [c,d])$.
Second, we have to show that $[a,b] < [c,d] \Rightarrow v([a,b]) < v([c,d])$.

First, $v([a,b]) + v([c,d])$

$$= v(\theta(a)) + v(b) + v(\theta(c)) + v(d) = [v(\theta(a)) + v(\theta(c))] + [v(b) + v(d)]$$
$$= [v(\theta(a \wedge c)) + v(\theta(a \vee c))] + [v(b \wedge d) + v(b \vee d)]$$
$$= [v(\theta(a \wedge c)) + v(b \vee d)] + [v(\theta(a \vee c)) + v(b \wedge d)]$$
$$= v([a \wedge c, b \vee d]) + v([a \vee c, b \wedge d])$$
$$= v([a,b] \wedge [c,d]) + v([a,b] \vee [c,d]).$$

Second, $[a,b] < [c,d]$

$$\Rightarrow \left\{ \begin{array}{llll} c < a & \text{and} & b \leqslant d \\ c \leqslant a & \text{and} & b < d \end{array} \right\} \Rightarrow \left\{ \begin{array}{llll} \theta(a) < \theta(c) & \text{and} & b \leqslant d \\ \theta(a) \leqslant \theta(c) & \text{and} & b < d \end{array} \right\}$$
$$\Rightarrow \left\{ \begin{array}{llll} v(\theta(a)) < v(\theta(c)) & \text{and} & v(b) \leqslant v(d) \\ v(\theta(a)) \leqslant v(\theta(c)) & \text{and} & v(b) < v(d) \end{array} \right\} \Rightarrow v(\theta(a)) + v(b) < v(\theta(c)) + v(d)$$
$$\Rightarrow v([a,b]) < v([c,d]). \quad \square$$

Proposition 7. *For* $p = 1, 2, \ldots$ *we have* $\mathrm{diag}_p([a,b]) = \max_{x,y \in [a,b]} d_p(x,y)$.

Proof. Consider an interval $[a,b]$ in a product lattice $\mathsf{L} = \mathsf{L}_1 \times \cdots \times \mathsf{L}_N$. If $a = (a_1, \ldots, a_N)$ and $b = (b_1, \ldots, b_N)$ it follows $a_i \leqslant b_i$, $i = 1, \ldots, N$. Let $x = (x_1, \ldots, x_N)$ and $y = (y_1, \ldots, y_N)$ be two points in the interval $[a,b]$. Note that $x \in [a,b]$ is equivalent to $a_i \leqslant x_i \leqslant b_i$, $i = 1, \ldots, N$. In the following we show that $d_i(x_i, y_i) \leqslant d_i(a_i, b_i)$.

$$\left. \begin{array}{l} x_i \leqslant b_i \Rightarrow x_i \vee y_i \leqslant b_i \vee y_i = b_i \\ a_i \leqslant x_i \Rightarrow a_i \wedge y_i = a_i \leqslant x_i \wedge y_i \end{array} \right\}$$
$$\Rightarrow a_i \leqslant x_i \wedge y_i \leqslant x_i \vee y_i \leqslant b_i \Rightarrow v_i(a_i) \leqslant v_i(x_i \wedge y_i) \leqslant v_i(x_i \vee y_i) \leqslant v_i(b_i)$$
$$\Rightarrow d_i(x_i, y_i) = v_i(x_i \vee y_i) - v_i(x_i \wedge y_i) \leqslant v_i(b_i) - v_i(a_i) = d_i(a_i, b_i).$$

It follows $d_p(x,y) \leqslant d_p(a,b)$, hence $\mathrm{diag}_p([a,b]) = \max_{x,y \in [a,b]} d_p(x,y)$. $\square$

References

[1] N. Ajmal, K.V. Thomas, Fuzzy lattices, Information Sciences 79 (3–4) (1994) 271–291.

[2] P.P. Angelov, Evolving Rule-Based Models, Studies in Fuzziness and Soft Computing, vol. 92, Physica-Verlag, Heidelberg, Germany, 2002.

[3] G.C. Anagnostopoulos, M. Georgiopoulos, Category regions as new geometrical concepts in Fuzzy-ART and Fuzzy-ARTMAP, Neural Networks 15 (10) (2002) 1205–1221.

[4] I.N. Athanasiadis, V.G. Kaburlasos, P.A. Mitkas, V. Petridis, Applying machine learning techniques on air quality data for real-time decision support, in: Proceedings of the 1st ICSC-NAISO Symposium on Information Technologies in Environmental Engineering (ITEE'2003), ICSC-NAISO Academic Press, Gdansk, Poland, June 2003, p. 51.

[5] I.N. Athanasiadis, P.A. Mitkas, An agent-based intelligent environmental monitoring system, Management of Environmental Quality 15 (3) (2004) 238–249.

[6] E. Baralis, S. Ceri, S. Paraboschi, Compile-time and runtime analysis of active behaviors, IEEE Transactions on Knowledge Data Engineering 10 (3) (1998) 353–370.

[7] M.R. Berthold, K.-P. Huber, Constructing fuzzy graphs from examples, International Computer Science Institute, Berkeley, California, Technical Report TR-97-053, 1997.

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

[9] A. Blumer, A. Ehrenfeucht, D. Haussler, M.K. Warmuth, Learnability and the Vapnik–Chervonenkis dimension, Journal of the ACM 36 (4) (1989) 929–965.
[10] L. Breiman, Bagging predictors, Machine Learning 24 (2) (1996) 123–140.
[11] P. Burillo, N. Frago, R. Fuentes, Inclusion grade and fuzzy implication operators, Fuzzy Sets and Systems 114 (3) (2000) 417–429.
[12] C. Carpineto, G. Romano, A lattice conceptual clustering system and its application to browsing retrieval, Machine Learning 24 (2) (1996) 95–122.
[13] K. Chakrabarty, On fuzzy lattice, in: W. Ziarko, Y. Yao (Eds.), Lecture Notes in Artificial Intelligence, vol. 2005, Springer-Verlag, Berlin, 2001, pp. 238–242.
[14] L.J. Clappa, M.E. Jenkin, Analysis of the relationship between ambient levels of O_3, NO_2 and NO as a function of NO_X in the UK, Atmospheric Environment 35 (36) (2001) 6391–6405.
[15] C. Cornelis, Van der Donck, E. Kerre, Sinha–Dougherty approach to the fuzzification of set inclusion revisited, Fuzzy Sets and Systems 134 (2) (2003) 283–295.
[16] C. Cornelis, M. De Cock, E. Kerre, Efficient approximate reasoning with positive and negative information, in: M.Gh. Negoita et al. (Eds.), Lecture Notes in Artificial Intelligence, vol. 3214, Springer-Verlag, Berlin, 2004, pp. 779–785.
[17] B.A. Davey, H.A. Priestley, Introduction to Lattices and Order, Cambridge University Press, Cambridge, UK, 1990.
[18] P. Diamond, P. Kloeden, Metric Spaces of Fuzzy Sets, World Scientific, Singapore, 1994.
[19] T.G. Dietterich, R.H. Lathrop, T. Lozano-Perez, Solving the multiple-instance problem with axis-parallel rectangles, Artificial Intelligence 89 (1–2) (1997) 31–71.
[20] P. Domingos, Two-way induction, in: Proceedings of the 7th International Conference on Tools with Artificial Intelligence, Herndon, VA, 1995, pp. 182–189.
[21] P. Domingos, Unifying instance-based and rule-based induction, Machine Learning 24 (2) (1996) 141–168.
[22] E.R. Dougherty, D. Sinha, Computational gray-scale mathematical morphology on lattices (a comparator-based image algebra) part II: image operators, Real-Time Imaging 1 (1995) 283–295.
[23] D. Dubois, H. Prade, L. Ughetto, A new perspective on reasoning with fuzzy rules, International Journal of Intelligent Systems 18 (5) (2003) 541–563.
[24] R.O. Duda, P.E. Hart, D.G. Stork, Pattern Classification, second ed., John Wiley & Sons, New York, NY, 2001.
[25] E.A. Edmonds, Lattice fuzzy logics, International Journal of Man–Machine Studies 13 (1980) 455–465.
[26] J. Fan, W. Xie, J. Pei, Subsethood measure: new definitions, Fuzzy Sets and Systems 106 (2) (1999) 201–209.
[27] B. Gabrys, A. Bargiela, General fuzzy min–max neural network for clustering and classification, IEEE Transactions on Neural Networks 11 (3) (2000) 769–783.
[28] B.R. Gaines, Fuzzy and probability uncertainty logics, Information and Control 38 (1978) 154–169.
[29] B. Ganter, R. Wille, Formal Concept Analysis: Mathematical Foundations, Springer-Verlag, Heidelberg, Germany, 1999.
[30] J.A. Goguen, *L*-fuzzy sets, Journal of Mathematical Analysis and Applications 18 (1967) 145–174.
[31] J. Goutsias, H.J.A.M. Heijmans, Nonlinear multiresolution signal decomposition schemes. I. Morphological pyramids, IEEE Transactions on Image Processing 9 (11) (2000) 1862–1876.
[32] P. Grzegorzewski, Distances between intuitionistic fuzzy sets and/or interval-valued fuzzy sets based on the Hausdorff metric, Fuzzy Sets and Systems 148 (2) (2004) 319–328.
[33] D. Haussler, Learning conjunctive concepts in structural domains, Machine Learning 4 (1) (1989) 7–40.
[34] M.J. Healy, T.P. Caudell, Acquiring rule sets as a product of learning in a logical neural architecture, IEEE Transactions on Neural Networks 8 (3) (1997) 461–474.
[35] S. Hedges, Ozone monitoring, mapping, and public outreach: delivering real-time ozone information to your community, Technical Report EPA/625/R-99/007, United States Environmental Protection Agency, 1999.
[36] H.J.A.M. Heijmans, J. Goutsias, Nonlinear multiresolution signal decomposition schemes. II. Morphological wavelets, IEEE Transactions on Image Processing 9 (11) (2000) 1897–1913.
[37] S. Heilpern, Representation and application of fuzzy numbers, Fuzzy Sets and Systems 91 (2) (1997) 259–268.
[38] J.M. Hellerstein, E. Koutsoupias, C.H. Papadimitriou, On the analysis of indexing schemes, in: Proceedings ACM Symposium on Principles of Database Systems, 1997, pp. 249–256.
[39] K. Hirota, W. Pedrycz, Fuzzy computing for data mining, Proceedings of the IEEE 87 (9) (1999) 1575–1600.

[40] L.-S. Huang, R.L. Smith, Meteorologically-depended trends in urban ozone, Environmetrics 10 (1) (1999) 103–118.
[41] H. Ishibuchi, T. Nakashima, T. Murata, Performance evaluation of fuzzy classifier systems for multidimensional pattern classification problems, IEEE Transactions on Systems, Man and Cybernetics – Part B 29 (5) (1999) 601–618.
[42] K. Itô (Ed.), Encyclopedic Dictionary of Mathematics, second ed., The Mathematical Society of Japan, Cambridge, MA, 1987 (English translation by MIT Press).
[43] A.K. Jain, M.N. Murty, P.J. Flynn, Data clustering: a review, ACM Computing Surveys 31 (3) (1999) 264–323.
[44] J.-S.R. Jang, C.-T. Sun, Neuro-fuzzy modeling and control, Proceedings of the IEEE 83 (3) (1995) 378–406.
[45] V.G. Kaburlasos, Adaptive resonance theory with supervised learning and large database applications, Ph.D. dissertation, Dept. Electrical Engineering, University of Nevada, Reno, 1992.
[46] V.G. Kaburlasos, FINs: Lattice theoretic tools for improving prediction of sugar production from populations of measurements, IEEE Transactions on Systems, Man and Cybernetics – Part B 34 (2) (2004) 1017–1030.
[47] V.G. Kaburlasos, Towards a unified modeling and knowledge-representation based on lattice theory – computational intelligence and soft computing applications, Studies in Computational Intelligence, vol. 27, Springer, Heidelberg, Germany, 2006.
[48] 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.
[49] V.G. Kaburlasos, A. Kehagias, Novel fuzzy inference system (FIS) analysis and design based on lattice theory, IEEE Transactions on Fuzzy Systems, in press.
[50] V.G. Kaburlasos, S.E. Papadakis, Granular self-organizing map (grSOM) for structure identification, Neural Networks 19 (5) (2006) 623–643.
[51] V.G. Kaburlasos, V. Petridis, Fuzzy lattice neurocomputing (FLN): a novel connectionist scheme for versatile learning and decision making by clustering, International Journal of Computers and Their Applications 4 (3) (1997) 31–43.
[52] V.G. Kaburlasos, V. Petridis, Fuzzy lattice neurocomputing (FLN) models, Neural Networks 13 (10) (2000) 1145–1170.
[53] V.G. Kaburlasos, V. Petridis, Learning and decision-making in the framework of fuzzy lattices, in: L.C. Jain, J. Kacprzyk (Eds.), New Learning Paradigms in Soft Computing, Studies in Fuzziness and Soft Computing, vol. 84, Physica-Verlag, Heidelberg, Germany, 2002, pp. 55–96.
[54] V.G. Kaburlasos, V. Petridis, B. Allotta, P. Dario, Automatic detection of bone breakthrough in orthopedics by fuzzy lattice reasoning (FLR): the case of drilling in the osteosynthesis of long bones, in: Proceedings of the Mechatronical Computer Systems for Perception and Action, Pisa, Italy, 1997, pp. 33–40.
[55] V.G. Kaburlasos, V. Petridis, P. Brett, D. Baker, Estimation of the stapes-bone thickness in stapedotomy surgical procedure using a machine-learning technique, IEEE Transactions on Information Technology in Biomedicine 3 (4) (1999) 268–277.
[56] E. Kalapanidas, N. Avouris, Applying machine learning techniques in air quality prediction, in: Proceedings of the ACAI, Chania, Greece, 1999, pp. 58–64.
[57] E. Kalapanidas, N. Avouris, Air quality management using a multi-agent system, International Journal of Computer Aided Civil and Infrastructure Engineering 17 (2) (2002) 119–130.
[58] N.N. Karnik, J.M. Mendel, Qilian Liang, Type-2 fuzzy logic systems, Transactions on Fuzzy Systems 7 (6) (1999) 643–658.
[59] M. Kearns, M. Li, L. Valiant, Learning Boolean formulas, Journal of the Association of Computing Machinery 41 (6) (1994) 1298–1328.
[60] M.J. Kearns, U.V. Vazirani, An Introduction to Computational Learning Theory, MIT Press, Cambridge, MA, 1994.
[61] A. Kehagias, An example of *L*-fuzzy join space, Rendiconti del Circolo Matematico di Palermo 51 (2) (2002) 503–526.
[62] A. Kehagias, M. Konstantinidou, *L*-fuzzy valued inclusion measure, *L*-fuzzy similarity and *L*-fuzzy distance, Fuzzy Sets and Systems 136 (3) (2003) 313–332.
[63] A. Kehagias, V. Petridis, V.G. Kaburlasos, P. Fragkou, A comparison of word- and sense-based text categorization using several classification algorithms, Journal of Intelligent Information Systems 21 (3) (2003) 227–247.

[64] G.J. Klir, T.A. Folger, Fuzzy Sets, Uncertainty, and Information, Prentice-Hall International Editions, London, UK, 1988.
[65] D.E. Knuth, Sorting and Searching, second ed., The Art of Computer Programming, vol. 3, Addison-Wesley, Reading, MA, 1998.
[66] K.H. Knuth, Lattice duality: the origin of probability and entropy, Neurocomputing 67 (2005) 245–274.
[67] J. Kolodner, Case-Based Reasoning, Morgan Kaufman Publishers, San Mateo, CA, 1993.
[68] B. Kosko, Fuzzy Engineering, Prentice-Hall, Upper Saddle River, NJ, 1997.
[69] S. Larrsen, L.O. Hagen, Air pollution monitoring in Europe. Problems and trends, Topic Report 26/1996, European Environment Agency, Copenhagen, Denmark, 1996.
[70] P. Lavoie, J.-F. Crespo, Y. Savaria, Generalization, discrimination, and multiple categorization using adaptive resonance theory, IEEE Transactions on Neural Networks 10 (4) (1999) 757–767.
[71] H.M. Liang, J.M. Mendel, Interval type-2 fuzzy logic systems: theory and design, Transactions on Fuzzy Systems 8 (5) (2000) 535–550.
[72] P.M. Long, L. Tan, PAC learning axis-aligned rectangles with respect to product distributions from multiple-instance examples, Machine Learning 30 (1) (1998) 7–21.
[73] Z.-Q. Liu, F. Yan, Fuzzy neural network in case-based diagnostic system, IEEE Transactions on Fuzzy Systems 5 (2) (1997) 209–222.
[74] E.H. Mamdani, S. Assilian, An experiment in linguistic synthesis with a fuzzy logic controller, International Journal of Man–Machine Studies 7 (1975) 1–13.
[75] C.J. Mantas, J.M. Puche, J.M. Mantas, Extraction of similarity based fuzzy rules from artificial neural networks, International Journal of Approximate Reasoning. Available online 3 May 2006.
[76] P. Maragos, Lattice image processing: a unification of morphological and fuzzy algebraic systems, Journal of Mathematical Imaging and Vision 22 (2–3) (2005) 333–353.
[77] T.M. Mitchell, Machine Learning, McGraw-Hill Series in Computer Science, The McGraw-Hill Companies, Inc., New York, NY, 1997.
[78] F. Mugica, A. Nebot, Reasoning under uncertainty with FIR methodology, International Journal of Approximate Reasoning 41 (3) (2006) 287–313.
[79] S. Nanda, Fuzzy lattice, Bulletin Calcutta Mathematical Society 81 (1989) 201–202.
[80] D. Nauck, Special issue on hybrid methods for adaptive systems, Fuzzy Sets and Systems 147 (1) (2004) 1–2.
[81] D. Nauck, R. Kruse, A neuro-fuzzy method to learn fuzzy classification rules from data, Fuzzy Sets and Systems 89 (3) (1997) 277–288.
[82] D. Nauck, R. Kruse, Obtaining interpretable fuzzy classification rules from medical data, Artificial Intelligence in Medicine 16 (2) (1999) 149–169.
[83] D. Nauck, R. Kruse, Neuro-fuzzy systems for function approximation, Fuzzy Sets and Systems 101 (2) (1999) 261–271.
[84] 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.
[85] E. Parrado-Hernández, E. Gómez-Sánchez, Y.A. Dimitriadis, Study of distributed learning as a solution to category proliferation in Fuzzy ARTMAP based neural systems, Neural Networks 16 (7) (2003) 1039–1057.
[86] S. Paul, S. Kumar, Subsethood-product fuzzy neural inference system (SuPFuNIS), IEEE Transactions on Neural Networks 13 (3) (2002) 578–599.
[87] W. Pedrycz, Granular networks and granular computing, in: L.C. Jain, J. Kacprzyk (Eds.), New Learning Paradigms in Soft Computing, Studies in Fuzziness and Soft Computing, vol. 84, Physica-Verlag, Heidelberg, Germany, 2002, pp. 30–54.
[88] W. Pedrycz, A. Bargiela, Granular clustering: a granular signature of data, IEEE Transactions on Systems, Man and Cybernetics – Part B 32 (2) (2002) 212–224.
[89] V. Petridis, V.G. Kaburlasos, Fuzzy lattice neural network (FLNN): a hybrid model for learning, IEEE Transactions on Neural Networks 9 (5) (1998) 877–890.
[90] V. Petridis, V.G. Kaburlasos, Learning in the framework of fuzzy lattices, IEEE Transactions on Fuzzy Systems 7 (4) (1999) 422–440, Errata in IEEE Transactions on Fuzzy Systems 8 (2) (2000) 236.
[91] V. Petridis, V.G. Kaburlasos, Clustering and classification in structured data domains using fuzzy lattice neurocomputing (FLN), IEEE Transactions on Knowledge and Data Engineering 13 (2) (2001) 245–260.
[92] V. Petridis, V.G. Kaburlasos, FINkNN: a fuzzy interval number k-nearest neighbor classifier for prediction of sugar production from populations of samples, Journal of Machine Learning Research 4 (Apr.) (2003) 17–37.

[93] U. Priss, Lattice-based information retrieval, Knowledge Organization 27 (3) (2000) 132–142.
[94] J.R. Quinlan, Induction of decision trees, Machine Learning 1 (1) (1986) 81–106.
[95] J.R. Quinlan, Decision trees and decision-making, IEEE Transactions on Systems, Man and Cybernetics 20 (2) (1990) 339–346.
[96] J.R. Quinlan, C4.5: Programs for Machine Learning, Morgan Kaufman Publishers, San Mateo, CA, 1993.
[97] G.X. Ritter, G. Urcid, Lattice algebra approach to single-neuron computation, IEEE Transactions on Neural Networks 14 (2) (2003) 282–295.
[98] J.C. Ruiz-Suarez, O.A. Mayora-Ibara, J. Torres-Jimenez, L.G. Ruiz-Suarez, Short-term ozone forecasting by artificial neural networks, Advances in Engineering Software 23 (3) (1995) 143–149.
[99] S. Salzberg, A nearest hyperrectangle learning method, Machine Learning 6 (3) (1991) 251–276.
[100] R. Setiono, W.K. Leow, J.M. Zurada, Extraction of rules from artificial neural networks and nonlinear regression, IEEE Transactions on Neural Networks 13 (3) (2002) 564–577.
[101] Y. Shoham, An overview of agent-oriented programming, in: J.M. Bradshaw (Ed.), Software Agents, MIT Press, Cambridge, MA, 1997, pp. 271–290.
[102] P.K. Simpson, Fuzzy min–max neural networks – Part 1: Classification, IEEE Transactions on Neural Networks 3 (5) (1992) 776–786.
[103] P.K. Simpson, Fuzzy min–max neural networks – Part 2: Clustering, IEEE Transactions on Fuzzy Systems 1 (1) (1993) 32–45.
[104] D. Sinha, E.R. Dougherty, Fuzzification of set inclusion: theory and applications, Fuzzy Sets and Systems 55 (1) (1993) 15–42.
[105] D. Sinha, E.R. Dougherty, A general axiomatic theory of intrinsically fuzzy mathematical morphologies, IEEE Transactions on Fuzzy Systems 3 (4) (1995) 389–403.
[106] J.F. Sowa, Knowledge Representation: Logical, Philosophical, and Computational Foundations, Brooks Cole Publishing Co., Pacific Grove, CA, 2000.
[107] Z. Sun, G. Finnie, K. Weber, Case base building with similarity relations, Information Sciences 165 (1–2) (2004) 21–43.
[108] P. Sussner, Associative morphological memories based on variations of the kernel and dual kernel methods, Neural Networks 16 (5–6) (2003) 625–632.
[109] T. Tagaki, M. Sugeno, Fuzzy identification of systems and its applications to modeling and control, IEEE Transactions on Systems, Man, and Cybernetics 15 (1985) 116–132.
[110] A.H. Tan, Cascade ARTMAP: integrating neural computation and symbolic knowledge processing, IEEE Transactions on Neural Networks 8 (2) (1997) 237–250.
[111] A. Tepavcevic, G. Trajkovski, *L*-fuzzy lattices: an introduction, Fuzzy Sets and Systems 123 (2) (2001) 209–216.
[112] H. Timm, C. Döring, R. Kruse, Different approaches to fuzzy clustering of incomplete datasets, International Journal of Approximate Reasoning 35 (3) (2004) 239–249.
[113] L.G. Valiant, A theory of the learnable, Communications of the ACM 27 (11) (1984) 1134–1142.
[114] D. Wettschereck, T.G. Dietterich, An experimental comparison of the nearest-neighbor and nearest-hyperrectangle algorithms, Machine Learning 19 (1) (1995) 5–27.
[115] I.H. Witten, E. Frank, Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations, Morgan Kaufman Publishers, San Mateo, CA, 1999, The WEKA platform is available at: <http://www.cs.waikato.ac.nz/ml/weak/>.
[116] Y. Xie, V.V. Raghavan, P. Dhatric, X. Zhao, A new fuzzy clustering algorithm for optimally finding granular prototypes, International Journal of Approximate Reasoning 40 (1–2) (2005) 109–124.
[117] J. Yi, V.R. Prybutok, A neural network model for the prediction of daily maximum ozone concentration in an industrialized urban area, Environmental Pollution 92 (3) (1996) 349–357.
[118] V.R. Young, Fuzzy subsethood, Fuzzy Sets and Systems 77 (3) (1996) 371–384.