Hindawi Publishing Corporation
Advances in Fuzzy Systems
Volume 2012, Article ID 206121, 8 pages
doi:10.1155/2012/206121

Research Article

Fuzzy Lattice Reasoning for Pattern Classification Using a New Positive Valuation Function

Yazdan Jamshidi Khezeli and Hossein Nezamabadi-pour

Department of Electrical Engineering, Shahid Bahonar University of Kerman, P.O. Box 76169-133, Kerman, Iran

Correspondence should be addressed to Hossein Nezamabadi-pour, nezam_h@yahoo.com

Received 20 April 2012; Accepted 23 July 2012

Academic Editor: F. Herrera

This paper describes an enhancement of fuzzy lattice reasoning (FLR) classifier for pattern classification based on a positive valuation function. Fuzzy lattice reasoning (FLR) was described lately as a lattice data domain extension of fuzzy ARTMAP neural classifier based on a lattice inclusion measure function. In this work, we improve the performance of FLR classifier by defining a new nonlinear positive valuation function. As a consequence, the modified algorithm achieves better classification results. The effectiveness of the modified FLR is demonstrated by examples on several well-known pattern recognition benchmarks.

1. Introduction

Much attention has been paid lately to applications of lattice theory [1] in different fields including neural networks [2]. Artificial neural networks whose computation is based on lattice algebra have become known as morphological neural networks [3, 4]. Lattices are popular in mathematical morphology including image processing applications [5, 6]. Moreover, algebraic lattices have been used for modeling associative memories [7]. In [8], the problem of capacity storage limitation in associative memories [9, 10] has been eliminated by proposing one-way and bidirectional lattice associative memories. Furthermore, lattices are used implicitly in some neural networks such as fuzzy ART and min-max [11, 12] as explained in [2, 13]. A practical advantage of lattice theory is the ability to model both uncertain information and disparate types of lattice-ordered data [14]. The term of a fuzzy lattice was proposed by Nanda in 1989 on the basis of the concept of a fuzzy partial-order relation [15]. Several authors have employed the notion "fuzzy lattice" in mathematics emphasizing algebraic properties of lattice ideals [16, 17]. Furthermore, the notion of fuzzy concept lattice has been studied in [18–20]. Sussner and Esmi [21] have introduced the morphological perceptron with a fusion of fuzzy lattices for competitive learning. Fuzzy lattices have also been used in clustering and classification algorithms. More specifically, independently from the development of morphological neural networks, Petridis and Kaburlasos [13] have found inspiration in lattice theory and versions of the ART model and have devised another successful approach to lattice-based computational intelligence. Hence, they proposed a fundamentally new and inherently hierarchical approach in neuron computing named fuzzy lattice neurocomputing (FLN) [14]. Moreover, fuzzy lattice reasoning (FLR) classifier was announced for inducing descriptive, decision-making knowledge (rules) in a mathematical data domain including space R^N, and it has been successfully applied to a variety of problems such as ambient ozone estimation [22] as well as air quality assessment [23]. Decision making in FLR is based on an inclusion measure function; moreover, the definition of an inclusion measure is based on a positive valuation function.

The original FLR model employs a linear positive valuation function to define an inclusion measure. Liu et al. [24] proposed a nonlinear valuation function (arctan) for computing the inclusion measure function and successfully applied it to several data set benchmarks.

In this work, we apply FLR algorithm to solve pattern classification problems without feature extraction and improve its performance based on a new nonlinear positive valuation function. As a consequence, the modified algorithm achieves better classification results. The effectiveness of the modified FLR is demonstrated by examples on several well-known benchmarks.

The layout of this paper is as follows. In Section 2, the mathematical background of fuzzy lattices is reviewed. Section 3 explains modified fuzzy lattice reasoning classifier model. Section 4 provides experimental results that demonstrate the performance of modified FLR. Finally, Section 5 summarizes the results of this work.

2. Mathematical Background

A lattice $(L, \leq)$ is a partially ordered set (or, simply, poset) such that any two of its elements $a, b \in L$ have a greatest lower bound $a \wedge b = \inf\{a, b\}$ and a least upper bound $a \vee b = \sup\{a, b\}$. The lattice operations $\wedge$ and $\vee$ are also called meet and join, respectively. A lattice $(L, \leq)$ is called complete when each of its subsets has a least upper bound and a greatest lower bound in L [1]. A nonvoid complete lattice has a least element and a greatest element denoted by O and I, respectively. The inverse $\geq$ of an order relation $\leq$ is itself an order relation. The order $\geq$ is called the *dual* order of $\leq$ symbolically $\leq^{\partial}$. A lattice $(L, \leq)$ can be Cartesian product of N constituent lattices $L_1, \ldots, L_N$, that is, $L = L_1 \times \cdots \times L_N$. The lattice operations meet and join of product lattice are defined as follows:

$$\begin{aligned} a \wedge b &= (a_1, \ldots, a_N) \wedge (b_1, \ldots, b_N) = (a_1 \wedge b_1, \ldots, a_N \wedge b_N), \\ a \vee b &= (a_1, \ldots, a_N) \vee (b_1, \ldots, b_N) = (a_1 \vee b_1, \ldots, a_N \vee b_N). \end{aligned} \tag{1}$$

A valuation on a crisp lattice L is a real-valued function $v : L \rightarrow R$ which satisfies $v(a) + v(b) = v(a \vee b) - v(a \wedge b)$, $a, b \in L$. A valuation is called monotone if and only if $a \leq b$ in L implies $v(a) \leq v(b)$ and positive if and only if $a < b$ implies $v(a) < v(b)$. We remark that the goal of positive valuation function v is to deal with lattice elements. Choosing a suitable valuation function is problem dependent. Decision making by the FLR is based on an inclusion measure function; therefore,a proper positive valuation function might improve performance.

Definition 1. An inclusion measure σ with least element O and great element I in a complete lattice L is a mapping $\sigma : L \times L \rightarrow [0, 1]$ such that it satisfies the following conditions [2]:

(i) $\sigma(a, O) = 0, \;\; a \neq 0$,

(ii) $\sigma(a, a) = 1$, for all $a \in L$,

(iii) $a \leq b \Rightarrow \sigma(c, a) \leq \sigma(c, b)$, for all $a, b, c \in L$ (consistency property).

It reveals that an inclusion measure indicates the degree to which one fuzzy set is contained in another one.

Theorem 2. *A positive valuation function $v : L \rightarrow R$ in a lattice $(L, \leq)$ with $v(O) = 0$ is a sufficient condition for two inclusion measures [2]:*

$$\begin{aligned} k(x, u) &= \frac{v(u)}{v(x \vee u)}, \quad (x \vee u \neq O), \\ s(x, u) &= \frac{v(x \wedge u)}{v(x)}, \quad (x \neq O). \end{aligned} \tag{2}$$

In our experiments, the data have been normalized in lattice $L = [0, 1]^N$, that is, the unit N-dimensional hypercube, where N is the dimension of the input data. Furthermore, we propose the following nonlinear positive valuation function:

$$\forall x \in \left([0, 1]^N, \leq\right), \quad v(x) = \sum_{i=1}^{N} x_i \ln(x_i + \gamma), \quad \gamma \in [1, +\infty), \tag{3}$$

where γ is called location parameter. Without loss of generality, let $a = (a_1, \ldots, a_N) \geq b = (b_1, \ldots, b_N)$, that is, $a_i \geq b_i$, $i = 1, \ldots, N$:

$$\begin{aligned} v(a \vee b) + v(a \wedge b) & \\ &= \sum_{i=1}^{N} v_i(a_i \vee b_i) + \sum_{i=1}^{N} v_i(a_i \wedge b_i) \\ &= \sum_{i=1}^{N} (a_i \vee b_i) \ln(a_i \vee b_i + \gamma) \\ &\quad + \sum_{i=1}^{N} (a_i \wedge b_i) \ln(a_i \wedge b_i + \gamma) \\ &= \sum_{i=1}^{N} a_i \ln(a_i + \gamma) + \sum_{i=1}^{N} b_i \ln(b_i + \gamma) \\ &= \sum_{i=1}^{N} v_i(a_i) + \sum_{i=1}^{N} v_i(b_i) = v(a) + v(b). \end{aligned} \tag{4}$$

Furthermore, the proposed valuation function is a strictly increasing function, thus for any $a, b \in ([0, 1]^N, \leq), a < b \Rightarrow v(a) < v(b)$. Finally, the aforementioned function (3) maps the least element of lattice $([0, 1]^N, \leq)$ to zero. On the other hand, $v(O) = \sum_{i=1}^{N} 0 \times \ln(0 + \gamma) = 0$.

The aforementioned valuation function operates in a more flexible manner compared with other valuations proposed in the literature. First, the performance of FLR can be optimized by selecting different values of the location parameter. Second, if the first variable x is assumed to be constant, then it will be converted to $v(x) = \ln(x + \gamma)$, in the space R, and in some special applications, it might be a proper valuation function. One may say that it does not satisfy the condition $v(O) = 0$, in this case, it can be defined as follows [14]:

$$v'(x) = v(x) - v(O), \quad \forall x \in L. \tag{5}$$

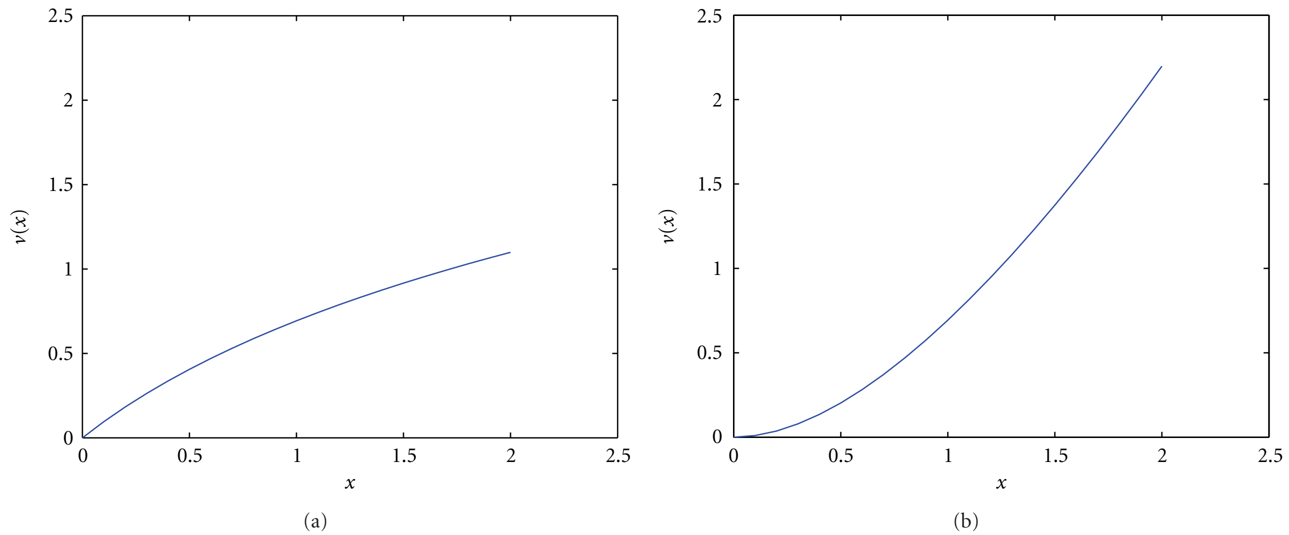


FIGURE 1: The positive valuation functions: (a) $\ln(x+1)$ and (b) $x\ln(x+1)$.

Figure 1(a) plots the positive valuation function $v(x) = \ln(x+\gamma)$, whereas Figure 1(b) plots $v(x) = x\ln(x+\gamma)$ for $\gamma = 1$.

A lattice $(L, \leq)$ is totally ordered if and only if for any $a, b \in L$, either $a \geq b$ or $a < b$. The lattice $([0,1]^N, \leq)$ under inequality relation is not a totally ordered lattice.

A fuzzy lattice is a pair (L, μ) where $(L, \leq)$ is a crisp lattice and $(L \times L, \mu)$ is a fuzzy set with membership function $\mu : L \times L \rightarrow [0,1]$ such that $\mu(a,b) = 1 \Leftrightarrow a \leq b$. Note that given a lattice L, for which a positive valuation function $v : L \rightarrow R$ can be defined with $v(O) = 0$, then both (L, k) and (L, s) are fuzzy lattices [2].

Consider the set R of real numbers. It turns out that $(\overline{R} = R \cup \{-\infty, +\infty\}, \leq)$ under the inequality relation $\leq$ between $a, b \in R$ is a complete lattice with the least element $-\infty$ and the greatest element $+\infty$ [25].

For lattice $(L, \leq)$, we define the set of (closed) intervals as $\tau(L) = \{[a,b] \mid a, b \in L \text{ and } a \leq b\}$. We remark that $(\tau(L), \leq)$ is a lattice with the ordering relation, lattice join and meet defined as follows [25]:

$$\begin{aligned} [a,b] &\leq [c,d] \equiv c \leq a, \quad b \leq d, \\ [a,b] \vee [c,d] &= [a \wedge c, b \vee d], \\ [a,b] \wedge [c,d] &= [a \vee c, b \wedge d]. \end{aligned} \tag{6}$$

Including a least (the empty) interval, denoted by $[I, O]$, to $(\tau(L), \leq)$ leads to a complete lattice $(\tau_O(L), \leq) = (\tau(L) \cup \{[I,O]\}, \leq)$. Note that in case of $L = \overline{R}$, the lattice $(\tau_O(L), \leq)$ is equal to conventional intervals (sets) in $\overline{R}$. Our particular interest here is in the complete lattice $(\tau_O([0,1]), \leq)$ with greatest element $[0,1]$ and least element $[1,0]$.

An isomorphic function φ from poset P to poset Q is a map if both "$x \leq y \, in \, P \Leftrightarrow \varphi(x) \leq \varphi(y) \, in \, Q$" *and* "$\varphi$ *is onto* Q." Based on the positive valuation function v of lattice $(L, \leq)$ and an isomorphic function $\theta : (L, \leq^{\partial}) \rightarrow (L, \leq)$, a valuation function v_{τ_O} in $(\tau_O(\overline{R}), \leq)$ is defined as

$$v_{\tau_O}([a,b]) = v(\theta(a)) + v(b). \tag{7}$$

As a consequence, the degree of inclusion of an interval in another one in lattice $(\tau_O(L), \leq)$ is computed as follows [25]:

$$\begin{aligned} k_{\tau_O}([a,b],[c,d]) &= \frac{v_{\tau_O}([c,d])}{v_{\tau_O}([a,b] \vee [c,d])}, \\ s_{\tau_O}([a,b],[c,d]) &= \frac{v_{\tau_O}([a,b] \wedge [c,d])}{v_{\tau_O}([a,b])}. \end{aligned} \tag{8}$$

For two N-dimensional hypercubes $A = [a_1, b_1] \times \cdots \times [a_N, b_N]$ and $B = [c_1, d_1] \times \cdots \times [c_N, d_N]$, the following inclusion measure between two intervals A and B is defined:

$$\begin{aligned} k_{\tau_O}(A,B) &= \frac{v_{\tau_O}(B)}{v_{\tau_O}(A \vee B)} \\ &= \frac{\sum_{i=1}^{N} [v\theta(c_i) + v(d_i)]}{\sum_{i=1}^{N} [v\theta(a_i \wedge c_i) + v(b_i \vee d_i)]}. \end{aligned} \tag{9}$$

3. FLR Model

This section presents a classifier for extracting rules from the input data based on fuzzy lattices. One of FLR important properties is the ability of dealing with disparate type of data, including real vectors, fuzzy sets, symbols, graphs, images, waves, and even any combination of the aforementioned data and this shows the ability of FLR in combining different types of data. Furthermore, FLR can handle both complete and noncomplete lattices, and it can cope with both points and intervals. Moreover, stable learning is carried out both incrementally and fast in a single pass through the training data. In some applications, we might face with "missing" or "do not care" data. In this case, FLR can manage "missing" and "do not care" data by replacing them with least element O and great element I, respectively. For example, if the constituent lattice is $([0,1], \leq)$, then we can replace "missing" and "do not care" data by intervals of $O = [1,0]$ and $I = [0.1]$, respectively [13, 14, 22].

It should be mentioned that an input datum to the FLR classifier (model) is represented as (a_i, C_K) where C_K is the class label of the datum a_i, and it can be interpreted as a rule "*if* a_i *then* C_k." We remark that a single real number $a \in R$ corresponds to the trivial interval $[a,a]$. Learning and generalization in FLR is based on the computation of hyperboxes in space R^N, that is, a rule induced by FLR corresponds to an N-dimensional hypercube.

Suppose a knowledge base $\text{KB} = \{(a_1, C_1), \ldots, (a_c, C_c)\}$ is given. KB can be empty at first. Decision making in FLR is based on an inclusion measure. During learning phase, when an input datum (a_0, C_0) is presented to the network, the degree of inclusion between input and each stored rules in KB will be calculated as $k(a_0, a_1), \ldots, k(a_0, a_c)$, respectively. The fuzzy lattice reasoning classifier will choose the rule with $\arg\max_{i \in \{1,\ldots,c\}} \{k(a_0, a_i)\}$ as the winner. If the winner rule a_J and input datum a_0 have the same class label and the size of $(a_0 \vee a_J)$, denoted by Z, is less than a user defined threshold, then the winner rule will be updated. Note that the size of an interval $[a,b]$ is computed as $Z[a\ b] = v(b) - v(a)$. Otherwise, this process is repeated; if no more rules are left, then the input datum (a_0, C_0) will be a new member of KB. Algorithm for training is described in Algorithm 1.

Note that ρ is the threshold size which is used to specify the maximum size of a hyperbox to be learned. The decision boundaries which can be formed by FLR endowed with logarithmic valuation function are illustrated in the following example.

Example 3. The Simpson benchmark is a two-dimensional data set consisting of 24 points which is used for testing the performance of a clustering algorithm [26]. This is a perceptual grouping problem in vision, which deals with the detection of the right partition of an image into subsets [27]. We have divided the data into three classes.

Figure 2 shows the decision surfaces of the FLR endowed with proposed logarithmic valuation function without any misclassified data. Here, due to lack of space, we have classified the data four times for location parameter (γ) equal to $\gamma = 1$ and with four different vigilance parameter ρ values. Note that the size of a hyperbox is tuned by the vigilance parameter ρ; more specifically, larger values of ρ result in more hyperboxes.

As it was said in the previous section, one of the FLR properties is the ability of knowledge representation. Indeed, FLR is capable of extracting implicit features beyond the data and represents them as rules. Each rule is represented as *if* $(a_1\ AND \cdots AND\ a_M)$ *then* C_i where $a_j, j = 1, \ldots, M$ are attributes, each one corresponding to an interval, and $C_i, i = 1, \ldots, c$ are class labels. Table 1 shows three induced rules corresponding to Figure 2(d).

4. Experimental Results and Discussions

4.1. Benchmark Dataset Description. In this section, we evaluate the classification performance of the optimized FLR in a series of experiments on six well-known benchmarks.

4.1.1. Object Recognition. We evaluate the classification performance of the FLR model using images of the Columbia Image database [28]. Columbia object image library (COIL-100) is a database of color images of 100 objects. We selected the 10 objects from the dataset shown in Figure 3. The objects were placed on a motorized turntable against a black background. The turntable was rotated through 360° to vary object pose with respect to a fixed color camera. Images of the objects were taken at pose intervals of 5°. This corresponds to 72 poses per object. The images were size normalized. There are 720 128×128-dimensional instances divided into 10 separate classes; 72 for each class; only six randomly selected instances per each object were sufficient for whole training set, and the remaining patterns were used for testing set. The aim is the correct classification of the testing data to their corresponding classes.

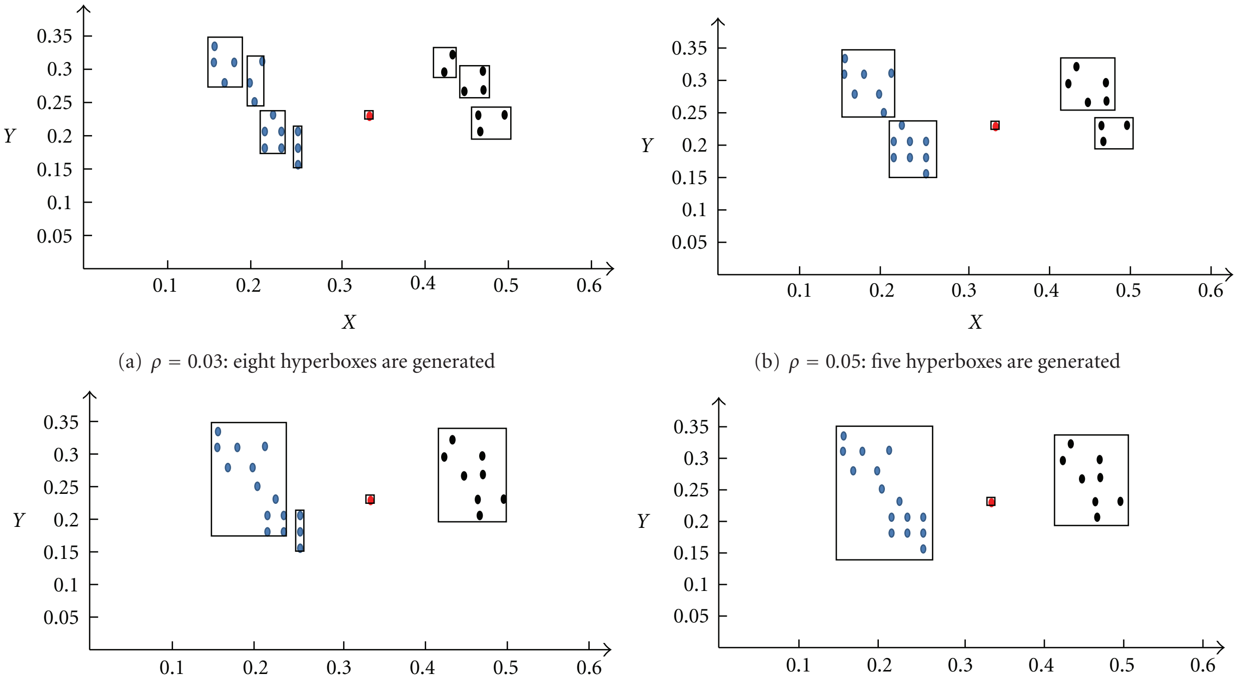
4.1.2. Image Segmentation. The image segmentation data set was donated by the Vision Group, University of Massachusetts, and is included in the Machine Learning Repository of the University of the California, Irvine [29]. The Image Segmentation data set consisted of data relating numerous analyses of the colors in subdivided images to the type of surface in the image. Each image was divided up into small subsections, each of which comprised one point of data. Each data point was composed of 18 different attributes, including one that determined what the image was of: brick face, foliage, grass, sky, window, concrete, and dirt. This data set consists of 210 samples for training and 2100 samples for testing. The goal is to distinguish between seven different classes.

4.1.3. Pen-Based Recognition of Handwritten Digits. The Pen-based recognition of handwritten digits dataset was taken from the UCI repository of machine learning databases [29]. It was created by collecting 250 digit samples from 44 writers. The data have 16 continuous attributes distributed in 10 separated classes. A training set is given explicitly. For a faster simulation, we have resized the training set by selecting randomly six instances per each class. Distribution of digits between 0 and 9 in the dataset is shown in Figure 4.

4.1.4. Letter Recognition. The letter recognition benchmark was employed from the UCI repository of machine learning databases [29]. The data set consists of 20,000 unique letter images generated by randomly distorting pixel images of the 26 uppercase letters from 20 different commercial fonts. The parent fonts represented a full range of character types including script, italic, serif, and gothic. The features of each of the 20,000 characters were summarized in terms of 16 primitive numerical attributes. A training set is not given explicitly. We have divided all the data into a training set

```
S0. The first input $(a_0, C_0)$ is memorized. At an instant, there are $c$ Known
Classes $C_1, \ldots, C_c$ memorized in the memory, initially $c = 0$.
S1. Present the next input $(a_i, C_k)$, $i = 1, \ldots, m$ to the initially "set" family of rules.
S2. If no rules are "set" then
Store input $(a_i, C_K)$,
$c = c + 1$,
        Go to S1.
    Else
Compute $k(a_0, a_i)$, $i = 1, \ldots, c$ of the "set" rules.
S3. Competition among the "set" rules:
        Winner is rule $(a_J, C_J)$ such that $J = \arg\max \{k(a_0, a_i)\}$, $i = 1, \ldots, c$ .
S4. The Assimilation Condition:
        Both $Z(a_i \vee a_J) \leq \rho$ and $C_i = C_J$.
S5. If the Assimilation Condition is satisfied then
        Replace $a_J$ by $a_0 \vee a_J$.
    Else
        "reset" the winner $(a_J, C_J)$, Go to S2.
```

Algorithm 1: FLR training algorithm.



(a) $\rho = 0.03$: eight hyperboxes are generated

(b) $\rho = 0.05$: five hyperboxes are generated

(c) $\rho = 0.08$: four hyperboxes are generated

(d) $\rho = 0.09$: three hyperboxes are generated

Figure 2: Decision boundaries generated by the modified FLR with four different threshold parameters.

Table 1: Three induced rules generated by the modified FLR.

Rule no.		Attributes				Class label
		a1		a2		
1	IF	[0.18, 0.27]	AND	[0.16, 0.3]	THEN	1
2	IF	[0.34, 0.34]	AND	[0.22, 0.22]	THEN	2
3	IF	[0.43, 0.49]	AND	[0.2, 0.28]	THEN	3

Table 2: Characteristics of 6 used data sets.

Data set	No. of data set elements	No. of training data	No. of testing data	No. of input attributes	No. of classes
Columbia Image	720	60	620	16384	10
Image Segmentation	2310	210	2100	19	7
Pen-based recognition	3558	60	3498	16	10
Letter recognition	20000	2000	18000	16	26
Semion hand recognition	1593	162	1431	256	10
Optical recognition of handwritten digits	2182	385	1797	64	10

Table 3: Recognition results along with relative ranking by different methods over 6 benchmarks.

Data set\Algorithm	SOM	Fuzzy-ART	GRNN	FLR$_x$	FLR$_a$	FLR$_l$
Columbia Image	43.87 (4)	40.81 (5)	90.16 (3)	94.03 (2)	**94.36** (**1**)	**94.36** (**1**)
Image Segmentation	79.86 (5)	88.71 (4)	92.43 (3)	**93.00** (**1**)	92.71 (2)	**93.00** (**1**)
Pen-based recognition	79.24 (6)	65.70 (5)	81.85 (2)	81.48 (3)	80.85 (4)	**81.90** (**1**)
Letter recognition	68.65 (5)	65.22 (6)	82.33 (4)	84.26 (2)	83.42 (3)	**84.48** (**1**)
Semion hand recognition	71.56 (4)	68.27 (5)	78.34 (3)	78.97 (2)	78.97 (2)	**79.18** (**1**)
Optical recognition of handwritten digits	87.87 (5)	88.76 (4)	**94.49** (**1**)	93.27 (3)	93.60 (2)	93.27 (3)

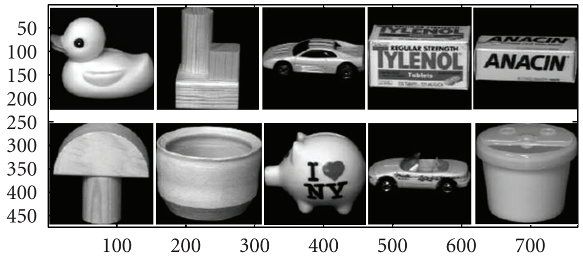


Figure 3: Ten objects used to train the networks.

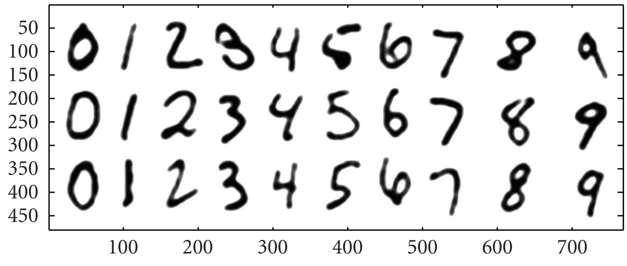


Figure 4: Sample digits from handwritten digits datasets.

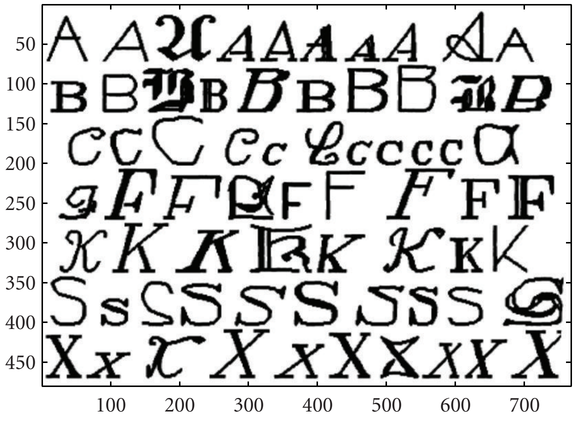


Figure 5: Examples of the character images.

consisting of 10 percent of patterns which have been selected randomly and a testing set consisting of the remaining patterns. Examples of the character images are presented in Figure 5.

4.1.5. Semion Hand Recognition. The semion hand recognition benchmark was taken from the UCI repository of machine learning databases [29]. This dataset consists of 1593 handwritten digits from around 80 persons was scanned and stretched in a rectangular box 16 × 16 in a gray scale of 256 values. Then each pixel of each image was scaled into a boolean value using a fixed threshold. Each person wrote on a paper all the digits from 0 to 9, twice. The commitment was to write the digit the first time in the normal way and the second time in a fast way. We have used 10 percent of data for whole training set.

4.1.6. Optical Recognition of Handwritten Digits. The optical recognition of handwritten digits benchmark was employed from the UCI repository of machine learning databases [29]. In this data set, 32 × 32 bitmaps of handwritten digits from a total of 43 people are divided into nonoverlapping blocks of 4 × 4, and the numbers of on pixels are counted in each block. This generates an input matrix of 8 × 8 where each element is an integer in the range of 0 to 16. This reduces dimensionality and gives invariance to small distortions. Training and testing sets are given explicitly including 3823 and 1797 64-dimensional samples. For a faster simulation, the way we have used the data set was to employ 10 percent of the training set for actual training.

Table 4: Average classification accuracy on the entire data sets.

Algorithm	SOM	Fuzzy ART	GRNN	FLR_x	FLR_a	FLR_l
Average	71.84 (5)	69.58 (6)	86.60 (4)	87.50 (2)	87.32 (3)	**87.70** (**1**)

Table 5: Sum of ranking of Table 3, on each column.

Algorithm	SOM	Fuzzy-ART	GRNN	FLR_x	FLR_a	FLR_l
Sum of ranks	29 (5)	29 (5)	16 (4)	13 (2)	14 (3)	**8** (**1**)

Table 2 shows briefly the characteristic of the selected benchmark data sets.

4.2. Experiments and Results. In order to provide a meaningful comparison, all the algorithms have been implemented in the same environment using the C++ object-oriented programming language, the same partitioning of data sets for training and testing, the same order of input patterns, and a full range of parameters, and we have employed the isomorphic function $\theta(x) = 1 - x$. Furthermore, all the N-dimensional data have been normalized into space $[0,1]^N$ by the function $x_{\text{norm}} = (x - x_{\min})/(x_{\max} - x_{\min})$, where $x_{\min}$ and $x_{\max}$ stand for the least and the greatest attribute values, respectively, in a data dimension. In this work, the FLR algorithm endowed with linear valuation function x, nonlinear valuation arctan, and nonlinear logarithmic valuation function is denoted, respectively, by FLR_x, FLR_a, and FLR_l. To compare the learning capability, Table 3 shows the comparison of the experimental results of FLR_l with the ones produced by the FLR_x and FLR_a, the SOM [30], fuzzy ART [11], and GRNN [31].

In all our experiments in order to achieve the best performance, we have considered GRNN for different values of variance parameter between 0 and 0.5 in steps of 0.001. For fuzzy ART, we have set the choice parameter to 0.01, and the values of vigilance and learning parameters have been adopted between 0 and 1 in steps of 0.01. Computational experiments for the SOM algorithm have been carried out using $M \times M$ $(M = 1, \ldots, 10)$ grids of units and number of 100 epochs. Since the results produced by SOM depend on the initialization of the weights, we have chosen the weights that yielded best results on the testing set for 10 random initializations. The only parameter for FLR algorithm that should be tuned is the threshold size parameter ρ. In our simulations, the size of ρ was set from 0.01 up to N, where N is the dimension of the input data, in steps of 0.01 except for object recognition data set, we have set ρ in steps of 50 due to high-dimensional input data. Just for FLR_l, the location parameter γ should be tuned too. We have set γ between 1 and 50 in steps of 1.

Table 3 cites the classification accuracy and ranking of different methods for each benchmark. In other words, each table cell, which belongs to a specific learning algorithm and the data set, contains the percentages of correct classification of that model over the corresponding data set. The number in brackets in each table cell shows the ranking of each method after running on a specific data set. The best results have been shown in bold face.

As can be seen in Table 3, FLR_l has obtained acceptable results in comparison with other methods, and in five cases, it gets the first rank. In Table 4, the average classification accuracy on all data sets for each of the different learning algorithms along with the new relative obtained ranks has been shown. In other words, first the average of each column of the previous table has been calculated, then the corresponding ranking is shown within brackets. As it can be seen from among all the methods, FLR_l has achieved the best ranking among all other methods.

In Table 5, this comparison has been made according to the sum of the ranks available in Table 3 per each column. Although this quantity is of lower precision degree for reporting results in some cases, it is common in nonparametric statistics. As can be seen in this table, FLR_l, FLR_x, and FLR_a get first, second, and third rankings, respectively. It should be pointed out that although there is no universal learning algorithm that can get the best results on the all benchmarks, the results obtained by FLR_l confirm that our proposed model is an efficient classifier compared with established classifiers from the literature.

5. Conclusion

In this work, we introduced an improvement of fuzzy lattice reasoning (FLR) classifier using a new nonlinear positive valuation function. We have investigated the performance of new FLR model in several well-known classification problems. Experimental results demonstrated that our proposed methods outperformed established classification models in terms of classification accuracy on the testing data.

References

[1] G. Birkhoff, *Lattice Theory*, American Mathematical Society, Providence, RI, USA, 3rd edition, 1967.

[2] V. G. Kaburlasos, “Towards a unified modeling and knowledge-representation based on lattice theory: computational intelligence and soft computing applications,” in *Studies in Computational Intelligence*, Springer, New York, NY, USA, 2006.

[3] G. X. Ritter and G. Urcid, “Lattice algebra approach to single-neuron computation,” *IEEE Transactions on Neural Networks*, vol. 14, no. 2, pp. 282–295, 2003.

[4] M. E. Valle and P. Sussner, “Storage and recall capabilities of fuzzy morphological associative memories with adjunction-based learning,” *Neural Networks*, vol. 24, no. 1, pp. 75–90, 2011.

[5] G. X. Ritter and J. N. Wilson, *Handbook of Computer Vision Algorithms in Image Algebra*, CRC Press, Boca Raton, Fla, USA, 1996.

[6] M. Graña, A. M. Savio, M. García-Sebastián, and E. Fernandez, "A lattice computing approach for on-line fMRI analysis," *Image and Vision Computing*, vol. 28, no. 7, pp. 1155–1161, 2010.

[7] G. X. Ritter and G. Urcid, "Learning in lattice neural networks that employ dendritic computing," *Studies in Computational Intelligence*, vol. 67, pp. 25–44, 2007.

[8] T. Kohonen, "Correlation matrix memory," *IEEE Transactions on Computers*, vol. C-21, no. 4, pp. 353–359, 1972.

[9] J. J. Hopfield, "Neurons with graded response have collective computational properties like those of two-state neurons," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 81, no. 10 I, pp. 3088–3092, 1984.

[10] B. Kosko, "Bidirectional associative memories," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 18, no. 1, pp. 49–60, 1988.

[11] G. A. Carpenter, S. Grossberg, and D. B. Rosen, "Fuzzy ART: fast stable learning and categorization of analog patterns by an adaptive resonance system," *Neural Networks*, vol. 4, no. 6, pp. 759–771, 1991.

[12] P. K. Simpson, "Fuzzy min-max neural networks–I: classification," *IEEE Transactions on Neural Networks*, vol. 3, no. 5, pp. 776–786, 1992.

[13] V. Petridis and V. G. Kaburlasos, "Fuzzy lattice neural network (FLNN): a hybrid model for learning," *IEEE Transactions on Neural Networks*, vol. 9, no. 5, pp. 877–890, 1998.

[14] V. G. Kaburlasos and 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*, vol. 4, no. 3, pp. 31–43, 1997.

[15] S. Nanda, "Fuzzy lattices," *Bulletin Calcutta Mathematical Society*, vol. 81, pp. 1–2, 1989.

[16] Y. Bo and W. Wangming, "Fuzzy ideals on a distributive lattice," *Fuzzy Sets and Systems*, vol. 35, no. 2, pp. 231–240, 1990.

[17] N. Ajmal and K. V. Thomas, "Fuzzy lattices," *Information Sciences*, vol. 79, no. 3-4, pp. 271–291, 1994.

[18] G. Georgescu and A. Popescu, "Non-dual fuzzy connections," *Archive for Mathematical Logic*, vol. 43, no. 8, pp. 1009–1039, 2004.

[19] R. Bělohlávek, "Fuzzy Galois connections," *Mathematical Logic Quarterly*, vol. 45, no. 4, pp. 497–504, 1999.

[20] S. Q. Fan, W. X. Zhang, and W. Xu, "Fuzzy inference based on fuzzy concept lattice," *Fuzzy Sets and Systems*, vol. 157, no. 24, pp. 3177–3187, 2006.

[21] P. Sussner and E. L. Esmi, "An introduction to morphological perceptrons with competitive learning," in *Proceedings of the International Joint Conference on Neural Networks (IJCNN'09)*, pp. 3024–3031, Atlanta, Ga, USA, June 2009.

[22] V. G. Kaburlasos, I. N. Athanasiadis, and P. A. Mitkas, "Fuzzy lattice reasoning (FLR) classifier and its application for ambient ozone estimation," *International Journal of Approximate Reasoning*, vol. 45, no. 1, pp. 152–188, 2007.

[23] I. N. Athanasiadis and V. G. Kaburlasos, "Air quality assessment using fuzzy lattice reasoning (FLR)," in *Proceedings of the IEEE International Conference on Fuzzy Systems*, pp. 29–34, Vancouver, Canada, July 2006.

[24] H. Liu, S. Xiong, and Z. Fang, "FL-GrCCA: a granular computing classification algorithm based on fuzzy lattices," *Computers and Mathematics with Applications*, vol. 61, no. 1, pp. 138–147, 2011.

[25] V. G. Kaburlasos and T. Pachidis, "A Lattice-computing ensemble for reasoning based on formal fusion of disparate data types, and an industrial dispensing application," *Information Fusion*. In press.

[26] P. K. Simpson, "Fuzzy min-max neural networks. Part 2. Clustering," *IEEE Transactions on Fuzzy Systems*, vol. 1, no. 1, pp. 32–45, 1993.

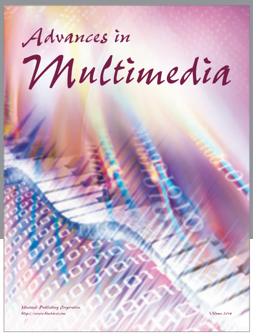
[27] K. C. Tan and H. J. Tang, "New dynamical optimal learning for linear multilayer FNN," *IEEE Transactions on Neural Networks*, vol. 15, no. 6, pp. 1562–1570, 2004.

[28] A. Nene, S. K. Nayar, and H. Murase, "Columbia object image library (COIL100)," Tech. Rep. CUCS-005-96, Department of Computer Science, Columbia University, 1996.

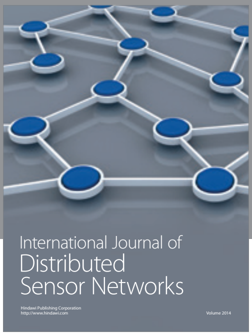
[29] C. J. Merz and P. M. Murphy, *UCI Repository of Machine Learning Databases*, Department of Information and Computer Science, University of California, Irvine, Irvine, Calif, USA, 1996, http://archive.ics.uci.edu/ml/datasets.html.

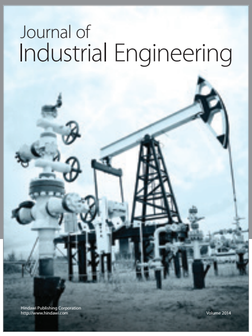
[30] T. Kohonen and P. Somervuo, "How to make large self-organizing maps for nonvectorial data," *Neural Networks*, vol. 15, no. 8-9, pp. 945–952, 2002.

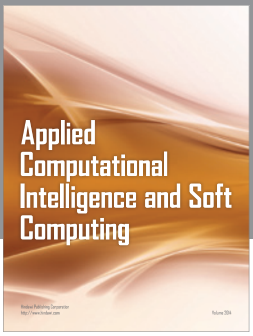
[31] D. F. Specht, "A general regression neural network," *IEEE Transactions on Neural Networks*, vol. 2, no. 6, pp. 568–576, 1991.

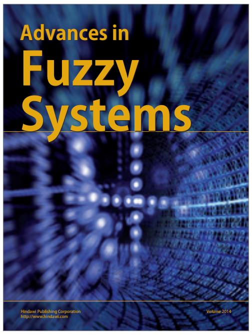

Advances in
Multimedia

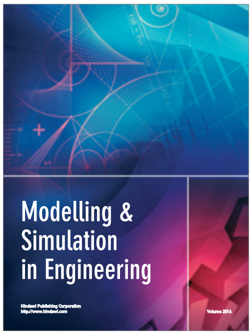
The Scientific
World Journal

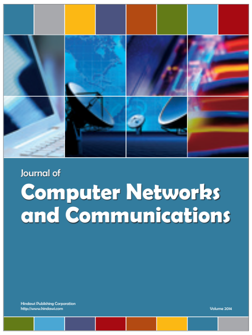

International Journal of
Distributed
Sensor Networks

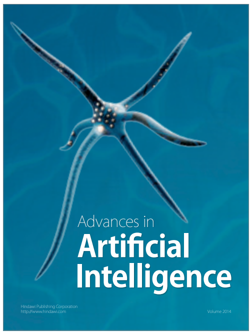

Journal of
Industrial Engineering

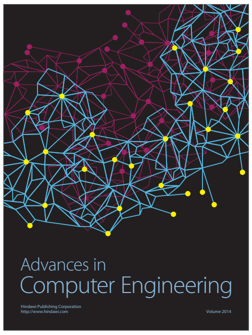

Applied
Computational
Intelligence and Soft
Computing


Advances in
Fuzzy
Systems

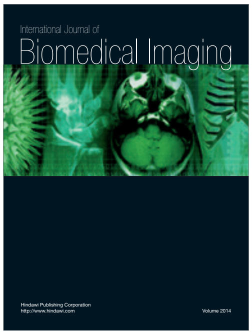

Modelling &
Simulation
in Engineering

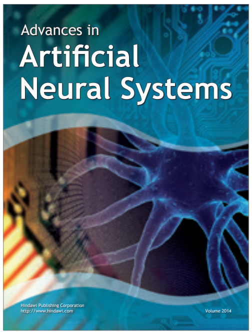

Journal of
Computer Networks
and Communications

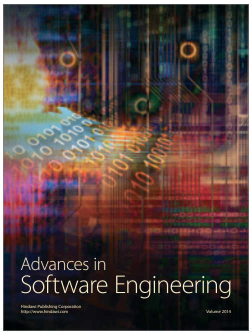

Advances in
Artificial
Intelligence


Advances in
Computer Engineering

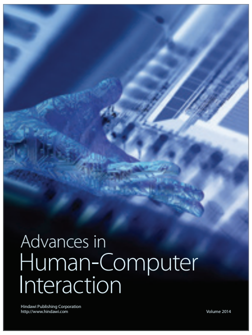
International Journal of
Computer Games
Technology

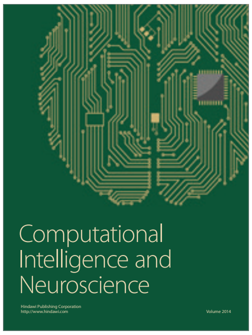

International Journal of
Biomedical Imaging


Advances in
Artificial
Neural Systems


Advances in
Software Engineering

Journal of
Robotics


Advances in
Human-Computer
Interaction


Computational
Intelligence and
Neuroscience

International Journal of
Reconfigurable
Computing

Journal of
Electrical and Computer
Engineering