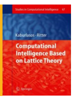
In **Computational Intelligence Based on Lattice Theory**, (Kaburlasos, V. G. and Ritter, Gerhard X., Eds.), Springer-Verlag, 2007, pp.175-193.



9

The Fuzzy Lattice Reasoning (FLR) Classifier for Mining Environmental Data

Ioannis N. Athanasiadis

Istituto Dalle Molle di Studi sull'Intelligenza Artificiale,
Galleria 2, CH-6928 Manno, Lugano, Switzerland
ioannis@athanasiadis.info

Summary. This chapter introduces a rule-based perspective on the framework of fuzzy lattices, and the Fuzzy Lattice Reasoning (FLR) classifier. The notion of fuzzy lattice rules is introduced, and a training algorithm for inducing a fuzzy lattice rule engine from data is specified. The role of positive valuation functions for specifying fuzzy lattices is underlined and non-linear (sigmoid) positive valuation functions are proposed, that is an additional novelty of the chapter. The capacities for learning of the FLR classifier using both linear and sigmoid functions are demonstrated in a real-world application domain, that of air quality assessment. To tackle common problems related to ambient air quality, a machine learning approach is demonstrated in two applications. The first one is for the prediction of the daily vegetation index, using a dataset from Athens, Greece. The second concerns with the estimation of quartely ozone concentration levels, using a dataset from Valencia, Spain.

9.1 Introduction

The framework of fuzzy lattices has been utilized lately in machine learning applications, mainly by utilizing artificial neural network architectures, i.e. as in [9, 10, 11, 12, 14, 15]. In this chapter, the fuzzy lattice framework is approached from a rule-based, reasoning perspective. Two issues related to fuzzy lattice reasoning are discussed here. The first one is the foundation of the fuzzy lattice rule engines, as a remedy for classification problems. The notion of fuzzy lattice rule is introduced, which employes fuzzy lattice elements as the rule antecedents, while the fuzzy inclusion measure serves as a truth function for deriving to rule consequences (conclusions). On top of the fuzzy lattice rule, a fuzzy lattice rule engine is specified, with which classification tasks can be performed. A fuzzy lattice rule engine induction algorithm is presented in this chapter as well. The second issue deals with the positive valuation functions for defining fuzzy lattices from partially ordered sets. Previous works have employed only linear positive valuation functions for defining fuzzy lattices. In this work, non-linear positive valuation functions are investigated and a sigmoid one is introduced. The sigmoid positive valuation functions are immidiately applicable to the framework of fuzzy lattices, and its variety of applications. Non-linear positive valuation functions can be considered as an extention to existing artificial neural network architectures based on fuzzy lattices, as the Fuzzy Lattice Neural Networks (FLNN) [14], and the Fuzzy Lattice Neurocomputing models (FLN) [11]. In this chapter, the employment of non-linear positive valuation functions is demonstrated in the context of the Fuzzy Lattice Reasoning classifier presented. Specifically, the Fuzzy Lattice Reasoning classifier is used for addressing classification tasks related to ambient air quality, is comparison with other rule-based classifiers. In conclusion, the FLR classifier turned out with credible models both for the prediction of the daily vegetation index in the metropolitan area of Athens, Greece, and the estimation of quartely ozone concentration levels in the region of Valencia, Spain. In both cases the use of sigmoid positive valuation functions improved the performance of the classifier, while it didn't increase the complexity of model. Actually, in one of the cases it was reduced significantly.

The rest of the chapter is organized as follows. Section 9.2 summarizes briefly the required mathematics, and is provided for the reader of the stand alone chapter. Readers who are are comfortable with the terms and notations of the book and can proceed to section 9.3, where the Fuzzy Lattice Reasoning classifier is presented. The introduction of a sigmoid positive valuation function is detailed in section 9.4. Section 9.5 demonstrates the two test-cases and presents comparative results with other classification methods. The main findings of this chapter are discussed in the last section 9.6.

9.2 Mathematical Background

A **lattice** L is a partially ordered set (poset), so that any two of its elements $a, b \in \mathsf{L}$ have a greatest lower bound (or meet) denoted by $a \wedge b := \texttt{inf}\{a, b\}$ and a least upper bound (or join) denoted by $a \vee b := \texttt{sup}\{a, b\}$. A lattice L is called complete when each of its subsets has a least upper bound and a greatest lower bound in L. A non-void complete lattice has a least element and a greatest element denoted by O and I, respectively.

The Cartesian product $\mathsf{L} = \mathsf{L}_1 \times \ldots \times \mathsf{L}_N$ of N constituent lattices $\mathsf{L}_1 \ldots \mathsf{L}_N$ (**product lattice**) is a lattice [7]. In a product lattice $\mathsf{L} = \mathsf{L}_1 \times \ldots \times \mathsf{L}_N$ inclusion can be defined as:

$$(x_1, \ldots, x_N) \leq (y_1, \ldots, y_N) \iff (x_1 \leq y_1) \& \ldots \& (x_N \leq y_N) \tag{9.1}$$

The meet in a product lattice $\mathsf{L} = \mathsf{L}_1 \times \mathsf{L}_N$ is given by $(x_1, \ldots, x_N) \wedge (y_1, \ldots, y_N) = (x_1 \wedge y_1, \ldots, x_N \wedge y_N)$, whereas the join is given by $(x_1, \ldots, x_N) \vee (y_1, \ldots, y_N) = (x_1 \vee y_1, \ldots, x_N \vee y_N)$ [7, 8].

A product lattice could combine diverse constituent lattices thus implying the potential to deal either separately and/or jointly, in any combination, with disparate types of data such as vectors of real numbers, propositions, (fuzzy) sets, events in a probability space, symbols, graphs, etc.

A **fuzzy lattice** is a pair $\langle \mathsf{L}, \mu \rangle$, where L is a 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 $\mu(a,b) = 1 \iff a \leq b$ [11].

The set of all fuzzy lattices $\langle \mathsf{L}, \mu \rangle$ is called framework of fuzzy lattices and has been used for decision-making in various applications [10, 11]. This paper approaches the fuzzy lattice framework from a rule-based perspective as presented in Section 9.3 below. Some more usefull instruments of the fuzzy lattice framework are the followings.

A **valuation function** $v : \mathsf{L} \to \mathsf{R}$ is defined on a lattice L as any real function that satisfies: $v(a) + v(b) = v(a \wedge b) + v(a \vee b), \forall a, b \in \mathsf{L}$. A valuation function is called **positive** if and only if $a < b \iff v(a) < v(b)$ [7]. Linear positive valuations function have been used in previous works [10] for defining an inclusion measure (σ) in a complete lattice L.

In general, an inclusion measure on a complete lattice L is defined as a real mapping function $\sigma : \mathsf{L} \times \mathsf{L} \to [0,1]$, such that for each $a, b, x \in \mathsf{L}$ the following conditions are satisfied:

$$\sigma(a, O) = 0, \forall a \neq O \tag{9.2}$$

$$\sigma(a, a) = 1 \tag{9.3}$$

$$a < b \Rightarrow \sigma(x, a) < \sigma(x, b) \tag{9.4}$$

$$a \wedge b < a \Rightarrow \sigma(a, b) < 1 \tag{9.5}$$

Given a lattice L and an inclusion measure $\sigma : \mathsf{L} \times \mathsf{L} \to [0,1]$ it turns out that $\langle \mathsf{L}, \sigma \rangle$ is a fuzzy lattice, with σ the membership function.

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

A positive valuation function $v : \mathsf{L} \to \mathsf{R}$ in a lattice L with $v(O) = 0$ is a sufficient condition for two inclusion measures [10]:

$$k(a, b) = \frac{v(b)}{v(a \vee b)} \tag{9.6}$$

$$s(a, b) = \frac{v(a \wedge b)}{v(a)} \tag{9.7}$$

Ultimately, given a lattice L, for which a positive valuation function $v : \mathsf{L} \to \mathsf{R}$ can be defined with $v(O) = 0$, then both $\langle \mathsf{L}, k \rangle$ and $\langle \mathsf{L}, s \rangle$ are fuzzy lattices. It becomes apparent that the only requirement for specifing a fuzzy lattice on a lattice L is the selection of an appropriate positive valuation function $v(\cdot)$. Based on this remark, any kind of partially ordered data, as numbers, sets, graphs, etc that can define a lattice, to which if a positive valuation function is ascribed, then it becomes available in the framework of fuzzy lattices.

The framework of fuzzy lattices has been extended to lattices of closed intervals [11], which are of particular interest for the introduction of fuzzy lattice rules in Section 9.3. In a complete lattice L, a closed interval of lattice elements is defined as

$$[a, b] = \{x | a \leq x \leq y\} \tag{9.8}$$

A singleton interval is defined as $[a, a] = a, \forall a \in \mathsf{L}$. The set $\tau(\mathsf{L})$ of all closed interval of lattice elements in L (including singletons) is also a complete lattice with upper bound $[O, I]$ and lower bound $[I, O]$. In $\tau(\mathsf{L})$ an ordering relation is defined as:

$$[a, b] \leq [c, d] \equiv \{c \leq a \quad \& \quad b \leq d\} \tag{9.9}$$

Join of two $\tau(\mathsf{L})$ elements $[a, b]$ and $[c, d]$ is defined as:

$$[a, b] \vee [c, d] = [a \wedge c, b \vee d] \quad (join) \tag{9.10}$$

And meet of two $\tau(\mathsf{L})$ elements $[a, b]$ and $[c, d]$ is defined as:

$$[a, b] \wedge [c, d] = \begin{cases} [a \vee c, b \wedge d], \text{ if } \quad a \vee c \leq b \wedge d \\ 0, \text{ otherwise} \end{cases} \qquad \text{(meet)} \tag{9.11}$$

The definition of a valuation function v_τ for the lattice of closed intervals $\tau(\mathsf{L})$ has been discussed in [11, 15]. Based on the positive valuation function $v : \mathsf{L} \to \mathsf{R}$ of lattice L and an isomorphic function $\theta : \mathsf{L}^\partial \to \mathsf{L}$, a valuation function in $\tau(\mathsf{L})$ is defined as:

$$v_\tau([a, b]) = v(\theta(a)) + v(b) \tag{9.12}$$

Both inclusion measures defined in equations (9.6), (9.7) using v_τ can be applied on $\tau(\mathsf{L})$:

$$k_\tau([a, b], [c, d]) = \frac{v_\tau([c, d])}{v_\tau([a, b] \vee [c, d])} \tag{9.13}$$

$$s_\tau([a, b], [c, d]) = \frac{v_\tau([a, b] \wedge [c, d])}{v_\tau([a, b])} \tag{9.14}$$

As a result it turns out that $\langle \tau(\mathsf{L}), k_\tau \rangle$, $\langle \tau(\mathsf{L}), s_\tau \rangle$ are fuzzy lattices. Note that the inclusion measure k_τ is more usable compared to s_τ, due to the conditional definition of the meet in $\tau(\mathsf{L})$, which appears in the nominator of Eq. (9.14). In the contrary, k_τ is unconditionally defined as:

$$\begin{aligned} k_\tau([a, b], [c, d]) &= \frac{v_\tau([c, d])}{v_\tau([a, b] \vee [c, d])} \\ &= \frac{v_\tau([c, d])}{v_\tau([a \wedge c, b \vee d])} \\ &= \frac{v(\theta(c)) + v(d)}{v(\theta(a \wedge c)) + v(b \vee d)} \end{aligned} \tag{9.15}$$

9.3 Fuzzy Lattice Reasoning (FLR) Classifier

Many data structures of practical interest are lattice ordered. The objective here is to present a classifier for inducing a rule-based inference engine from data, based on the instruments of the fuzzy lattice framework presented in the previous section.

9.3.1 Fuzzy lattice rule engine

A fuzzy lattice rule engine is based on fuzzy lattice rules. A fuzzy lattice rule employes a fuzzy lattice element as the rule antecedent, while the fuzzy inclusion measure serves as a truth function for deriving to rule consequences (conclusions).

A **fuzzy lattice rule** is a pair $\langle a, c\rangle$ where a is an element in a fuzzy lattice $\langle \mathsf{L}, \mu\rangle$ and $c \in \mathsf{C}$ is a categorical label. Note that this definition applies to the whole framework of fuzzy lattices, including product lattices and lattices of closed intervals. A fuzzy lattice rule can be considered as the mapping $a \rightarrow c$ of a fuzzy lattice $\langle \mathsf{L}, \mu\rangle$ element a to a categorical label c, where a is the rule antecedent and c is the consequence of the rule.

Let a and b be two lattice L elements, c a categorical label in C and function k, as defined in Eq. (9.6) be a fuzzy membership function in L. We define the **degree of truth** of the fuzzy lattice rule $a \rightarrow c$ against the perception b to be defined by the fuzzy membership function of the fuzzy lattice $\langle \mathsf{L}, \mu\rangle$, as:

$$\mu(b, a) = k(b, a) = \frac{v(a)}{v(b \vee a)} \tag{9.16}$$

Similarly holds for $\tau(\mathsf{L})$ using k_τ as defined in equation (9.13).

A **fuzzy lattice rule engine** $\mathcal{E}_{\langle \mathsf{L},\mu\rangle,\mathsf{C}}$ can be considered as a set of N fuzzy lattice rules that are commonly activated:

$$\mathcal{E}_{\langle \mathsf{L},\mu\rangle,\mathsf{C}} \equiv \{a_i \rightarrow c_i\}, a_i \in \langle \mathsf{L}, \mu\rangle, c_i \in \mathsf{C}, i = 1 \ldots N \tag{9.17}$$

Reasoning with a fuzzy lattice rule engine implies the calculation of the degree of truth for each one of engines rules. For example consider the following engine that consists of three rules:
$\mathcal{E}_{\langle \mathsf{L},\mu\rangle,\mathsf{C}} = \{a_1 \rightarrow c_1, a_2 \rightarrow c_2, a_3 \rightarrow c_3\}$, where a_1, a_2, a_3, are elements of a fuzzy lattice $\langle \mathsf{L}, \mu\rangle$ and c_1, c_2, c_3 a set of predifined labels. Against an input element a_0, the engine will result with the following table of degree of truth for each consequence: $c_1 = \sigma(a_0, a_1)$, $c_2 = \sigma(a_0, a_2)$, and $c_3 = \sigma(a_0, a_3)$. The fuzzy lattice reasoning engine will respond with the class $c = arg\underset{i}{\mathtt{max}}(\sigma(a_0, a_i))$. In another mode of generalization, $\mathcal{E}$ may respond with the label that is additively included the most. In this way, a fuzzy lattice reasoning engine can be used for generalization.

9.3.2 Fuzzy lattice rule induction (training)

The task of inducing a fuzzy lattice rule engine can be described as follows: Let a training set of M partially ordered objects $\{u_1, u_2, \ldots, u_M\} \in \mathcal{U}$, each one of which is associated with a class label $c \in \mathsf{C}$, where $\mathsf{C} = \{c_1, c_2, \ldots, c_K\}$ is a set of K predefined labels (classes). The objective is to induce a set of fuzzy lattice rules that implement a function $h : \mathsf{U} \to \mathsf{C}$, associating any object $u \in \mathsf{U}$ with a classification label $c \in \mathsf{C}$.

In general, the universe U of the training objects can include any type of complex data structures, as vectors of real numbers, graphs or sets. Obviously, U is a complete lattice. Given a positive valuation function $v : \mathsf{U} \to \mathsf{R}$, an inclusion measure $\sigma : \mathsf{U} \times \mathsf{U} \to [0, 1]$ can be defined in U, as denoted above in Eq. (9.6), (9.7), (9.13), (9.14), which implies a fuzzy membership function $\mu : \mathsf{U} \times \mathsf{U} \to [0, 1]$. In this respect, it turns out that $\langle \mathsf{U}, \mu \rangle$ is a fuzzy lattice. The classifier to be built is equivalent to a map $h' : \langle \mathsf{U}, \mu \rangle \to \mathsf{C}$, which is a set of fuzzy lattice rules, i.e. a fuzzy lattice rule engine: $h \equiv h' \equiv \mathcal{E}_{\langle \mathsf{U}, \mu \rangle, \mathsf{C}}$.

Each object u of the training set is an element of U and each training pair $\langle u, c \rangle$ can be expressed as a fuzzy lattice rule $u \to c$, where u is an element of the fuzzy lattice $\langle \mathsf{U}, \mu \rangle$ and c the corresponding class. This means that the instances of a training set could be treated as fuzzy lattice rules. For example consider the simple case where the universe of the training instances is a closed interval of real numbers $[O, I]$. Then any training pair $\langle x, c \rangle$ where $x \in [O, I]$ and $c \in \mathsf{C}$ can be expressed as a fuzzy lattice rule consisted from a lattice interval singleton mapped to class c, as: $\langle x, c \rangle \equiv \langle [x, x], c \rangle$. Likewise for alternative universes of discourse.

A **naive fuzzy lattice reasoning classifier** that can be induced directly from a set of M training pairs $(u_1, c_1), \ldots, (u_M, c_M)$, $u_i \in \mathsf{U}$, and $c_i \in \mathsf{C}$, is the one that memorizes all training instances as fuzzy lattice rules. Given a positive valuation function v, each training element u_i is an element of the fuzzy lattice $\langle \mathsf{U}, \sigma \rangle$, where σ is an inclusion measure defined in Eq. (9.6), (9.7), (9.13). In this way, the most simple fuzzy lattice rule engine will consist at most out of M (trivial) rules and will be: $\mathcal{E} = \{u_1 \to c_1, \ldots, u_i \to c_i, \ldots, u_M \to c_M\}$, where $u \in \langle \mathsf{U}, \sigma \rangle$ and $c \in \mathsf{C}$.

A **training process** for inducing a fuzzy lattice rule engine is based on joining lattice rules pointing to the same class for formulating lattice rules of higher size, and potentially higher ability for generalization. The training procedure for inducing the classifier h, through single pass iteration over all training instances is presented below. Note that a simplified version of the FLR algorithm was presented previously implemented as neural network σ-FLN architecture [10, 11].

FLR training algorithm

Step-0: Let a fuzzy lattice rule engine $\mathcal{E}_{\langle \mathsf{L}, \sigma \rangle, \mathsf{C}} = \{a_1 \to c_1, \ldots, a_R \to c_R\}$ of size R.

Note that $\mathcal{E}_{\langle \mathsf{L},\sigma\rangle,\mathsf{C}}$ could be initially empty, i.e. $R = 0$, and a user-defined threshold size D_{crit}.

Step-1: Present the next training pair $\langle u, c\rangle$, in the form of a fuzzy lattice rule $u \rightarrow c$ to the initially set rules in $\mathcal{E}_{\langle \mathsf{L},\sigma\rangle,\mathsf{C}}$.

Step-2: If no more rules in $\mathcal{E}$ are set then append input rule $u \rightarrow c$ in $\mathcal{E}$ and go to Step-1.
Else, compute the fuzzy degree of inclusion $\sigma(u \leq a_r)$, $\forall l = 1 \ldots R$ of the antecedent u to the antecedents of all the set rules in $\mathcal{E}$.

Step-3: Competition among the set rules in $\mathcal{E}$. Winner is the rule $a_J \rightarrow c_J$, where

$$J = arg \max_{r\in\{1,\ldots R\}} \sigma(u \leq a_r) \tag{9.18}$$

Step-4: If both $c = c_J$ and $diag(u \vee a_J) < D_{crit}$ (assimilation condition), then replace the antecedent a_J of the winner rule $a_J \rightarrow c_J$ by the join-lattice $u \vee a_J$, i.e. with the rule: $u \vee a_J \rightarrow c_J$. Go to Step-1.
Else, reset the winner rule $a_J \rightarrow c_J$, and go to Step-2.

Previous works has employed for the algorithm tuning, instead of D_{crit}, the dimensionless vigilance parameter:

$$\rho_{crit} = \frac{N}{N + D_{crit}} \Leftrightarrow D_{crit} = \frac{N(1 - \rho_{crit})}{\rho_{crit}} \tag{9.19}$$

Note that ρ_{crit} varies in the interval $[0.5, 1]$ for any number of dimensions N as shown in [11]. In the following experiments ρ_{crit} has been employed, as its range is not related to the dimension of the lattice.

9.3.3 Decision making with fuzzy lattice rules (testing)

The decision making process (**testing phase**) of an (induced) fuzzy rule engine $\mathcal{E}_{\langle \mathsf{L},\sigma\rangle,\mathsf{C}}$ of size R, involves the competition of its rules over a perception $x \in U$, of unknown label. The element x is presented to each rule of the engine: $a_r \rightarrow c_r$, and the inclusion measure $\sigma(x \leq a_r) \equiv \sigma(x, a_r)$ is calculated. Finally, x is assigned to the category c_J, where

$$J = arg \max_{r\in\{1,\ldots,R\}} \sigma(x \leq a_r) \tag{9.20}$$

A second mode of reasoning for the an (induced) fuzzy rule engine $\mathcal{E}_{\langle \mathsf{L},\sigma\rangle,\mathsf{C}}$ of size R may involve a contributing competition, where all rules pointing to the same class c_i will add-up their inclusion measures and the perception $x \in \mathsf{U}$ will be assigned to the class that additively includes it the most:

$$J = arg\max_{c_i\in\mathsf{C}} \sum_{r=\{1,\ldots,R/c_r=c_i\}} \sigma(x \leq a_r) \tag{9.21}$$

In principal, in any universe of partially ordered data, that can be formalized as lattices, product lattices or lattices of intervals a fuzzy lattice reasoning classifier can be induced. A similar lattice algorithm, namely Find-S algorithm, has been presented in a machine learning context [13], but without an employment of positive valuation functions. In the following section the capacity of the algorithm is further broaden, by introducing non-linear positive valuation functions.

9.4 Non-linear Positive Valuation Functions

The whole procedure for inducing an FLR classifier is related to the selection of an appropriate valuation function in U, that implies the formation of the fuzzy lattice $\langle \mathsf{U}, \sigma \rangle$, where σ is an inclusion measure as those defined in Eq. (9.6), (9.7), (9.13), (9.14). This remark holds also for other decision making schemes built upon the framework of fuzzy lattices that map data to lattices. Typically, prior works within the Framework of Fuzzy Lattices [2, 10, 12, 14, 15] have focused in forming fuzzy lattices from numerical datasets by employing linear valuation functions. In cases that data reside in the N-dimensional unit hypercube $\mathsf{I}^N = [0,1] \times [0,1] \times \ldots \times [0,1]$, the positive valuation function selected for each consistuent lattice is the simple function $v_i(x) = x$. In other cases, where data reside in R^N the training dataset can be formulated as $\mathsf{T} = [O_1, I_1] \times [O_2, I_2] \times \ldots \times [O_N, I_N]$, and the positive valuation function for each constituent lattice is given by the following equation that which linearly scales T to the N-dimensional unit hypercube I^N:

$$v_i(x) = \frac{(x - O)}{(I - O)} \tag{9.22}$$

In this paper, both linear and non-linear positive valuation functions are considered for inducing an FLR classifier from a numerical dataset. The sigmoid function is an example non-linear increasing function with range $[0,1]$ that could be used as a positive valuation function for mapping an interval of real numbers to a fuzzy lattice. In the particular case of lattice I a non-linear positive valuation function is defined by:

$$v_\lambda(x) = \frac{1}{1 + e^{-\lambda(x - 1/2)}}, \lambda > 0 \tag{9.23}$$

In generic the case of data residing within the interval $[O, I]$, a positive valuation function can be defined by the sigmoid function:

$$v_\varsigma(x) = \frac{1}{1 + e^{-\lambda(x - x_{med})}}, \tag{9.24}$$

where

$$x_{med} = \frac{I + O}{2}, \quad \lambda = \frac{\varsigma}{I - O}, \quad \varsigma > 0$$

The single parameter ς can be used for tuning the slope of $v_\varsigma(x)$. Fig. 9.1 plots function $v_\varsigma(x)$ for various values of ς, in contrast with a linear valuation function $v(x) = x$.

The capacity of non-linear positive valuation functions to improve performance has been demonstrated lately in classification and regression applications [1, 6]. In the following section, the capacity for learning of the FLR classifier is evaluated in two air quality data sets, by employing both linear and sigmoid positive valuation functions.

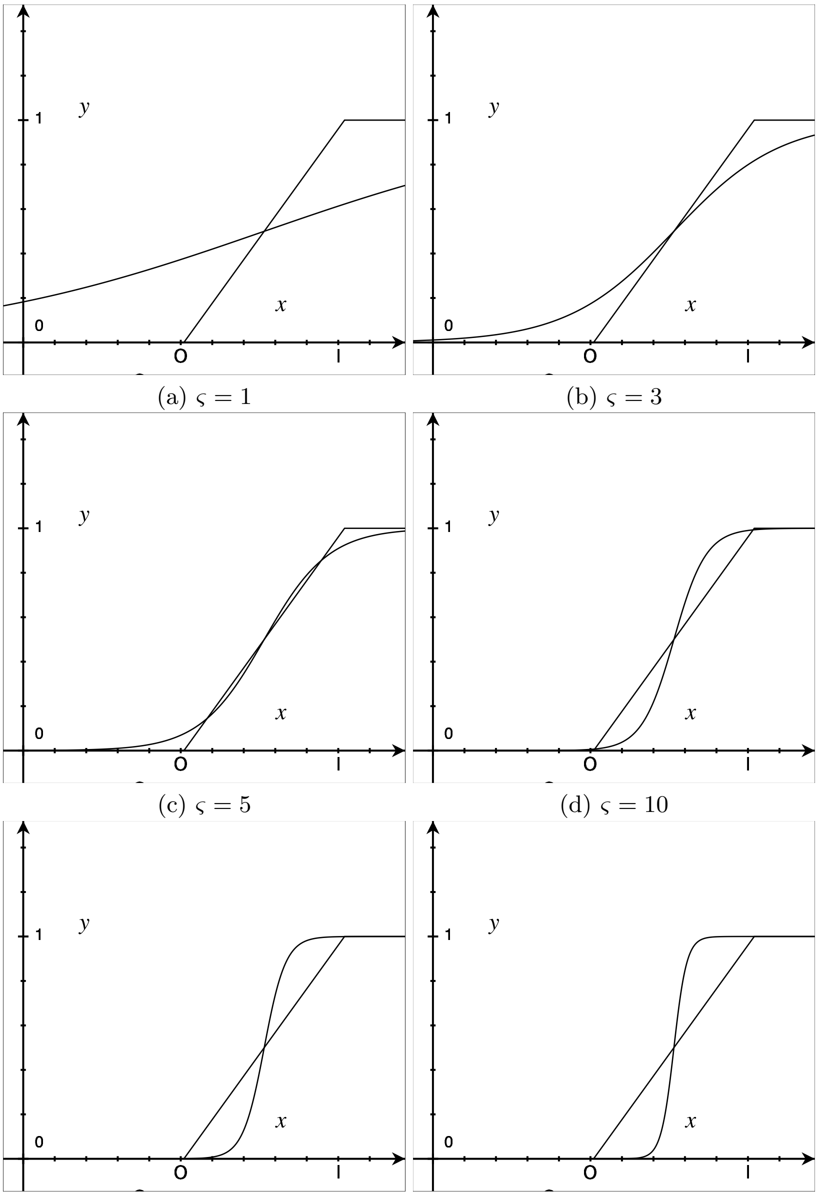
9.5 Application on Environmental Datasets

In this work, the problem of operational decision support related to air pollution is tackled by utilizing a machine learning approach. Specifically, the FLR Classifier is demonstrated in comparison with other state-of the-art algorithms in two application cases related to urban air quality assessment. The first one concerns with the prediction of the daily vegetation index in the metropolitan area of Athens, Greece, while the second one is for the estimation of ambient ozone concentration levels, in a rural area in Valencia, Spain.

9.5.1 Air quality assessment

Ambient air quality assessment and management is characterized by complexity and uncertainty mainly due to the difficulties of atmospheric chemistry and physics and the stochastic processes involved in air pollutant generation. These boundaries raise the major obstacles in building simple models for credible prediction. In most cases, decision making relies on human expertise, as analytical models are too complex and slow for operational decision support. Legislation in Europe, the US, and elsewhere, define environmental quality indicators, which could be communicated to the public on-time (or even in advance) for informing population about air quality, especially in urban areas.

In both application cases, focus is given on ambient ozone, which is a secondary pollutant formed as a result of catalytic reactions between pollutants emitted from industrial sources and automobiles. In the presence of sunlight (ultra-violet radiation) and under suitable meteorological conditions, the precursors react photo-chemically to *produce* ozone. Due to the chemical reaction dynamics, the analytical models for describing ozone formation in ambient air are very complex. As a consequence, simple, yet credible prediction models are required for achieving both the requirements of accurate air quality assessment and capabilities for fast decision making (in contrast with the analytical complex models). These properties can be realized by learning from data, using knowledge discovery techniques as discussed in previous works [3, 4, 5], and presented below.



(a) $\varsigma = 1$

(b) $\varsigma = 3$

(c) $\varsigma = 5$

(d) $\varsigma = 10$

(e) $\varsigma = 15$

(f) $\varsigma = 25$

Fig. 9.1. Sigmoid positive valuation functions $v_\varsigma(x)$ illustrated in the interval $[O, I]$ in contrast with the linear positive valuation function $v_i(x)$ for various values of the normalized parameter ς.

9.5.2 Daily vegetation index prediction in Athens, Greece

The first demonstration case concentrates in the metropolitan area of Athens, Greece, that suffers from air-pollution problems, mainly due to the traffic and industrial emission, but also because of the urban landscaping. A measure of the impact of air pollution to human quality of life is the daily vegetation index introduced by the European Commission with the Directive 92/72/EEC. It is an early warning indicator of the overall air quality and specifies a threshold on the ozone's mean concentration (O_3) over a 24 hours period. The same directive sets the daily vegetation threshold at the limit of 65 $\mu g/m^3$.

The prediction of the daily vegetation index threshold (exceeded or not) is the actual goal of the first demonstration case, where FLR has been employed. Specifically, a dataset was available that contained daily observations from the Maroussi station of the Ministry of Environment lined up with meteorological data from the Athens National Observatory. The selection of the monitoring station was based on the frequency of high exceedances of the selected index. The dataset covers a 3.5 year period (January 1999 - June 2001) and has been split in two parts: one for training (that corresponds to the period January 1999 - December 2001) and one for testing (January - June 2002). The purpose of this selection was the ability to make comparisons with the statistical methods used for the same test case as previously reported [3]. Tables 9.1, 9.2 present the dataset attributes and statistics.

Table 9.1. Athens dataset attributes.

Attribute	Symbol	Datatype	Units
1 Carbon monoxide	CO	real number	mg/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 Sulfur dioxide	$SO2$	real number	$\mu g/m^3$
6 Ozone	$O3$	real number	$\mu g/m^3$
7 Air temperature	Ta	real number	deg C
8 Soil temperature	Ts	real number	deg C
8 Relative humidity	RH	real number	
9 Wind speed	WS	real number	m/s
10 Wind direction	WD	real number	rad
11 Mean temperature	$meantemp$	real number	deg C
12 Max temperature	$maxtemp$	real number	deg C
13 Min temperature	$mintemp$	real number	deg C
14 Solar radiation	ws	real number	Wm^{-2}
16 Vegetation index	O_3 alert	class label	Yes/No

Table 9.2. Athens dataset statistics.

	Records in class	
	Yes	No
Training set	571	525
Testing set	62	120

9.5.3 Comparative results for the Athens dataset

A set of cross-evaluation experiments were conducted for the Athens test-case. FLR with both linear and sigmoid valuation functions has been applied for predicting the daily vegetation index, for a range of values for the parameters ρ and ς. For comparison purposes, results are presented in Table 9.3 along with previous results obtained for the same test case with statistical methods (LRA, ARIMA and PCA) [16] and other ten classification algorithms and neural networks (ADTree, C4.5, Conjunctive Rule, Decision Table, IBk, KStar, NaiveBayes, Nnge, OneR, Voted Perceptron) [3]. Note that WEKA platform [17] implementations of the algorithms have been used. The statistical methods resulted low overall accuracy rates (less than 60%), with high false positive rates, i.e. there are several alarms missed, but the confidence on those identified is very high. On the contrary, classification techniques manage higher classification accuracies, with the cost of a lower credibility on their decisions.

The application of FLR with the linear valuation function resulted up to an overall classification accuracy of 80.11% with a fuzzy lattice rule engine of 212 rules. The accuracy compared to the rest classification algorithms is relatively good (ADTree and NaiveBayes performed better). Also, the false positive rates of FLR with linear positive valuation function is the best achieved among classification algorithms. In this terms FLR performance is competitive.

Next, the FLR with a sigmoid positive valuation function was employed, resulting a Fuzzy Lattice Rule Engine of 123 rules. In this case, the overall accuracy improved to 84.53%, while the false positive rate decreased significantly to 5.88%. Overall, the introduction of a sigmoid positive valuation function achieved to decrease the number of extracted rules almost to half, result an overall accuracy similar to that of ADTree, while arriving the optimal false positive rate. Based on these remarks, the FLR with a sigmoid positive valuation function can be considered as the most credible model for predicting the daily vegetation index in this particular application.

9.5.4 Ozone level estimation in Valencia, Spain

The second test-case concerns with the peri-urban area of Valencia, Spain where ambient air quality is diminished due to industrial activities. The goal here is to identify the level of ozone concentration, a critical photochemical pollutant, which is commonly used as an indicator of the overall ambient

Table 9.3. Results for the Athens dataset.

Model	Accuracy (%)	False positive rate (%)
LCA	53.37	0.37
PCA	21.47	0.55
ARIMA	31.58	2.94
FLR (sigmoid)	84.53	5.88
FLR (linear)	80.11	8.4
Nnge	59.68	8.4
Conjunctive Rule	64.52	10.92
OneR	70.97	11.76
Decision Table	69.35	12.61
IBk	59.68	15.13
Voted Perceptron	80.65	15.97
ADTree	85.48	17.65
NaiveBayes	80.65	17.65
C4.5 (J48)	75.81	19.33
KStar	61.67	43.7

air quality. In this case, the objective is to estimate the ozone concentration levels from the concurrent observations of other pollutants and meteorological attributes, a task related with both quality assurance and control activities and operational decision making.

In this case, data were available from a single metrological stations, that monitors eight parameters, including both meteorological attributes and air-pollutant concentrations, as shown in Table 9.4. Data are sampled on a quarter-hourly basis during the year 2001. In total there are available 35,040 data vectors, out of which 565 records have the ozone label missing, and thus where excluded in the analysis below. Values were missing in other attributes, and in total there are 6,020 records (that is around 17% of the total) with at least one missing value. In the following experiments, both the original dataset with missing values and a preprocessed one that excluded all records with missing values. 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 available in classes `low` and `med(ium)`, respectively, are shown in Table 9.5.

9.5.5 Comparative results for the Valencia dataset

For estimating the ozone concentration level three classifiers were employed: (a) The C4.5 classifier, (b) The FLR classifier, with a linear positive valuation function, and (c) The FLR classifier, with a sigmoid positive valuation function. Two series of experiments have been carried out: first, using the dataset without missing values and, second, the original set including the ones with missing values.

Table 9.4. Valencia dataset attributes.

Attribute	Symbol	Datatype	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	deg C
7 Relative humidity	HR	real number	%
8 Ozone level	O_3	class label	`low`, `med`

Note: Class `low` corresponds to concentration levels in range $0 - 60$ $\mu g/m^3$, and class `med(ium)` to $60 - 100$ $\mu g/m^3$.

Table 9.5. Valencia dataset statistics for (a) the dataset without missing values, and (b) the dataset with missing values (original).

(a)

	Records in class	
	low	medium
Training set	6,865	4,761
Testing set	12,256	5,138

(b)

	Records in class	
	low	medium
Training set	9,472	6,074
Testing set	13,483	5,446

First, the C4.5 classifier has been employed on a standard software platform (WEKA platform [17]), for generating decision trees, 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 decision tree with 1393 leaves (rules). The corresponding classification accuracy on the training set reached 94.8%, whereas on the testing set it was only 64.85%. Similar results have been obtained for the dataset with no missing values. Obviously, C4.5 over-fits the training data, therefore two pruning methods have been employed: (1) Confidence Factor Pruning (CFP), and (2) Reduced Error Pruning (REP). Results are shown in Tables 9.6 and 9.9 for selected pruning parameter values. The highest accuracy achieved on the testing split was 73.74% and 77.56% respectively for each dataset.

The FLR classifier has been implemented on the same software platform (WEKA) using both linear and sigmoid valuation functions. Initially, the FLR Classifier has been employed using a linear valuation function. In this case, the valuation function used was $v_i(x) = \frac{(x-O)}{(I-O)}$, where [O, I] are the minimum and maximum values of the training data in each dimension. Results are presented in Tables 9.7 and 9.10 for selected values of the vigilance parameter ρ. The FLR Classifier achieved a classification accuracy of 83.23% with only three rules for the dataset without missing values and 84.60% with 19 rules for the dataset with missing values. Note that the FLR classifier outperforms C4.5.

Then, experiments have been conducted for the FLR Classifier using the sigmoid function of Eq. (9.24) on both datasets. In this case the FLR Classifier has been tuned using two parameters: The vigilance parameter ρ and the slope parameter ς of the sigmoid valuation function. Results obtained by FLR with sigmoid valuation function are presented in Tables 9.8 and 9.11. For the dataset without missing values the FLR with sigmoid positive valuation function achieved a classification accuracy of 85.22% with three rules. Note that using the sigmoid positive valuation function the best performance has improved by 2% without increasing the number of induced rules. For the dataset with missing values, the best accuracy improved by nearly 1%, again without increasing the number of rules, as shown in Tables 9.8 and 9.11.

9.6 Discussion

This chapter introduced the Fuzzy Lattice Reasoning (FLR) classifier, by considering Fuzzy Lattices as the foundation for specifying rules, both for Fuzzy Lattices and Fuzzy Lattices of intervals. Modes of generalization in a Fuzzy Lattice Rule Engine have been identified and a training procedure was detailed. Also, here non-linear positive valuation functions are introduced as an instrument for further improving the capacity for decision-making within the framework of Fuzzy Lattices. The FLR classifier was demonstrated for assessing ambient air quality on two test-cases. Results obtained with FLR Classifier for the case of the prediction of the daily vegetation index in Athens have compared favorably with the results obtained by other state-of-the-art classifiers and statistical approaches used in previous works. The FLR Classifier achieved the best performance in terms of false positive rates (5.88%), while keeping the overall accuracy at very high levels (84%). The introduction of the non-linear positive valuation function in this case resulted to an improvement of performance by 5%, while reducing the number of the model complexity (induced rules) to half. In the case of the estimation of ozone level concentrations in Valencia, the FLR Classifier resulted positively, with respect to the performance achieved with C4.5 decision trees. The FLR classifier with linear positive valuation function, compared to C4.5, improved classification accuracy by 9.5% for the dataset without missing values and by 7% for the dataset with missing values. Furthermore, the employment of a sigmoid positive valuation function by the FLR classifier achieved further improvement without increasing the complexity (number of induced rules) of the model. Finally, the approach presented here for tackling with the complexity and the uncertainties of the air quality assessment by using machine-learning techniques and in particular the FLR classifier rendered with trustworthy and credible results with a great potential for the application domain.

Table 9.6. Results with C4.5 for the Valencia dataset without missing values.

Parameter	Classification accuracy (%)		No. of Rules
value	Training set	Test set	(Tree leaves)
Unpruned			
-	94.80	64.85	1393
Confidence factor pruning (parameter: CF)			
0.1	91.33	67.31	575
0.2	92.87	66.71	823
0.3	93.92	67.40	1055
0.4	94.10	67.39	1101
0.5	94.31	67.19	1169
Reduced error pruning (parameter: no. of Folds)			
2	89.31	63.71	507
10	89.01	71.85	465
50	85.05	60.62	251
100	83.33	73.74	131
300	81.55	69.98	75
500	77.73	72.48	31

Table 9.7. Results with FLR with linear valuation function for the Valencia dataset without missing values.

Parameter	Classification accuracy (%)		No. of Rules
ρ	Training set	Test set	(Tree leaves)
0.5	59.16	70.46	2
0.6	64.73	83.23	3
0.7	73.68	74.85	20
0.8	67.43	72.59	139

Table 9.8. Results with FLR with sigmoid valuation function for the Valencia dataset without missing values.

Parameter		Classification accuracy (%)		No. of Rules
ς	ρ	Training set	Test set	(Tree leaves)
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

Table 9.9. Results with C4.5 for the Valencia dataset with missing values.

Parameter	Classification accuracy (%)		No. of Rules
value	Training set	Test set	(Tree leaves)
Unpruned			
-	94.80	64.85	1393
Confidence factor pruning (parameter: CF)			
0.1	89.14	60.26	279
0.2	89.98	59.19	368
0.3	90.81	59.44	463
0.4	91.37	59.30	542
0.5	91.59	59.32	598
Reduced error pruning (parameter: no. of Folds)			
2	88.14	64.91	318
10	88.28	59.19	288
50	85.44	60.17	144
100	84.01	61.36	84
300	82.48	77.56	44
500	81.33	70.19	32

Table 9.10. Results with FLR with linear valuation function for the Valencia dataset with missing values.

Parameter	Classification accuracy (%)		No. of Rules
ρ	Training set	Test set	(Tree leaves)
0.5	60.99	71.22	5
0.6	60.99	71.22	8
0.7	63.48	84.60	19
0.8	69.00	66.54	43

Table 9.11. Results with FLR with sigmoid valuation function for the Valencia dataset with missing values.

Parameter		Classification accuracy (%)		No. of Rules
ς	ρ	Training set	Test set	(Tree leaves)
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

References

1. Athanasiadis IN, Kaburlasos VG (2006) Air quality assessment using Fuzzy Lattice Reasoning (FLR). In: Proc World Congress Computational Intelligence (WCCI) FUZZ-IEEE Program pp 231–236
2. Athanasiadis IN, Kaburlasos VG, Mitkas PA, Petridis V (2003) Applying machine learning techniques on air quality data for real-time decision support. In: First Intl Symposium Information Technologies in Environmental Engineering (ITEE-2003) ICSC-NAISO Academic Press p 51
3. Athanasiadis IN, Karatzas KD, Mitkas P (2006) Classification techniques for air quality forecasting. In: Fifth ECAI Workshop on Binding Environmental Sciences and Artificial Intelligence, 17th European Conf on Artificial Intelligence
4. Athanasiadis IN, Mitkas PA (2004) Supporting the decision-making process in environmental monitoring systems with knowledge discovery techniques. Vol III Knowledge-based Services for the Public Sector Symposium: KDnet pp 1–12
5. Athanasiadis IN, Mitkas PA (accepted) Knowledge discovery for operational decision support in air quality management. J of Environmental Informatics
6. Cripps A, Nguyen N (2007) Fuzzy lattice reasoning (FLR) classification using similarity measures. This volume, chapter 13
7. Birkhoff G (1967) Lattice Theory. Colloquium Publications, Providence, RI
8. Davey B, Priestley H (1990) Introduction to Lattices and Order. Cambridge University Press, Cambridge, UK
9. Georgiopoulos M, Fernlund H, Bebis G, Heileman G (1996) Order of search in fuzzy ART and fuzzy ARTMAP: effect of the choice parameter. Neural Networks 9(9):1541–1559
10. Kaburlasos V (2006) Towards a Unified Modeling and Knowledge-Representation Based on Lattice Theory: Computational Intelligence and Soft Computing Applications, ser Studies in Computational Intelligence 27. Springer, Heidelberg, Germany
11. Kaburlasos VG, Petridis V (2000) Fuzzy lattice neurocomputing (FLN) models. Neural Networks 13(10):1145–1170
12. Kaburlasos VG, Petridis V (2002) Learning and decision-making in the framework of fuzzy lattices. In: Jain LC, Kacprzyk J (eds) New Learning Paradigms in Soft Computing, ser Studies in Fuzziness and Soft Computing 84:55–96. Physica-Verlag, Heidelberg, Germany
13. Mitchell T (1997) Machine Learning. McGraw-Hill, New York, NY
14. Petridis V, Kaburlasos VG (1998) Fuzzy lattice neural network (FLNN): a hybrid model for learning. IEEE Trans Neural Networks 9(5):877–890
15. Petridis V, Kaburlasos V (1999) Learning in the framework of fuzzy lattices. IEEE Trans Fuzzy Systems 7(4):422–440
16. Slini T, Karatzas K, Moussiopoulos N (2005) Ozone forecasting supported by data mining statistical methods. In: R Sokhi, J Brexhler (eds) 5th International Conference on Urban Air Quality Measurement, Modelling and Management
17. Witten IH, Frank E (1999) Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations. Morgan Kaufmann, USA