In **First Int'l Symposium on Information Technologies in Environmental Engineering (ITEE-2003)**, Gdansk, Poland, ICSC-NAISO Academic Press, June 2003, pp.51.

Applying Machine Learning Techniques on Air Quality Data for Real-Time Decision Support

Ioannis N. Athanasiadis[1,3], **Vassilis G. Kaburlasos**[2], **Pericles A. Mitkas**[1,3] **and Vassilios Petridis**[3]

[1] Informatics and Telematics Institute
Centre for Research and Technology – Hellas
GR570 01 Thermi, Greece

[2] Department of Industrial Informatics
Technological Educational Institute of Kavala
GR654 04 Kavala, Greece

[3] Department of Electrical and Computer Engineering
Aristotle University of Thessaloniki
GR541 24 Thessaloniki, Greece

Abstract

Fairly rapid environmental changes call for continuous surveillance and decision making, areas where IT technologies can be valuable. In the aforementioned context this work describes the application of a novel classifier, namely *σ-FLNMAP*, for estimating the ozone concentration level in the atmosphere. In a series of experiments on meteorological and air pollutants data, the *σ–FLNMAP* classifier compares favorably with both back-propagation neural networks and the C4.5 algorithm; moreover *σ–FLNMAP* induces only a few rules from the data. The *σ–FLNMAP* classifier can be implemented as either a neural network or a decision tree. We also discuss the far reaching potential of *σ–FLNMAP* in IT applications due to its applicability on partially (lattice) ordered data.

1 Introduction

Air quality is typically assessed based on either expert meteorologist knowledge or on sophisticated "first principles" mathematical models. Air Quality Operational Centers have been established worldwide in areas with (potential) air pollution problems. These centers monitor critical atmospheric variables and they publish regularly their analysis results [KA99]. Currently, real-time decisions are made by human experts, whereas mathematical models are used for off-line study and understanding of the atmospheric phenomena involved.

The goal of this work is real time assessment of air quality. Specific problems in real-time air quality assessment include: sensor malfunction, instrument polarization, noise, etc. Moreover, rapid environmental changes have rendered previous assessment methods obsolete [FA96]. At the same time, state regulations worldwide have defined stricter pollution levels. There is a need for new techniques for reliable real-time assessment of air quality based on sampled data. In this context Information Technology (IT) techniques including Machine Learning, Data Mining, Multi-Agent Systems, etc. are promising to assist human experts.

2 Decision support systems for assessing air quality in real–time

The Centre for Research and Technology - Hellas is currently collaborating, in the context of the Agent Academy project, with Investigación y Desarrollo Informático Eikon (IDI–EIKON), Valencia, Spain on the development of an agent-based decision support system for real-time assessment of air quality [AA00].

Email Addresses:
I.N. Athanasiadis: `ionathan@ee.auth.gr`
V.G. Kaburlasos: `vgkabs@teikav.edu.gr`
P.A. Mitkas: `mitkas@eng.auth.gr`
V. Petridis: `petridis@eng.auth.gr`

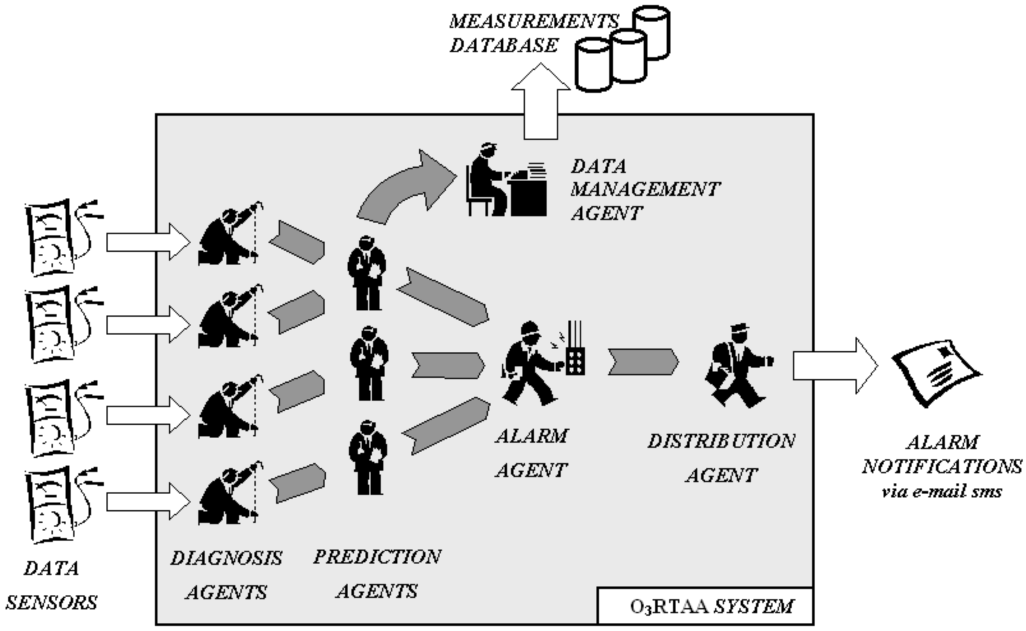

Figure 1: The O3RTAA decision support system.

The decision support system being developed, namely O_3RTAA, will be installed in the Fundación Centro de Estudios Ambietales del Mediterráneo (CEAM), Valencia, Spain to assist human experts in assessing air quality. The system's architecture has been described in [MS02] and [KS02].

The O_3RTAA system is currently developed as a multi-agent decision support system. Several software agents co–operate in a distributed agent society, in order to monitor both meteorological and air–quality attributes in an effort to to evaluate air quality and, ultimately, to trigger alarms. The aforementioned multi-agent system architecture uses the following types of agents:

1. *Diagnosis agents.* These are agents running on sensors and their goal is to monitor various air quality attributes including NO, NO_2, NO_x, O_3, etc.

2. *Prediction agents*, which are in charge of both pulling the data from monitoring agents and of verifying that the latter agents operate properly. In case of a sensor breakdown, prediction agents are in charge of predicting the missing values.

3. *Alarm agents*, which evaluate the inputs and decide whether an alarm should be triggered or not.

4. *Distribution agents*, which are in charge of triggering alarms selectively.

The O_3RTAA system architecture is shown in Fig.1

The function of the prediction agent is outlined next. A "sensor breakdown" results in a missing air quality parameter value. The predictor agent is expected to supply an approximation of the missing value. Likewise, a missing air quality attribute value may trigger an alarm. In the latter case, a prediction agent is in charge of producing an approximation of the missing value based on the remaining attribute values.

This work demonstrates real time prediction of ozone concentration by classification, based on meteorological and air quality data. More specifically, an *σ–FLNMAP* classifier has been employed here and is compared with both back-propagation neural networks and decision trees. In the following section the classifier *σ–FLNMAP* is described in a lattice theoretic context.

3 The *σ–FLNMAP* Classifier

The *σ–FLNMAP* classifier has been introduced in [PK99, KP00, KP02]. It is applicable in a fuzzy lattice data domain including the N-dimensional Euclidean space $\Re^N$. An important advantage of a "lattice data domain" is that disparate types of data can be dealt with in principle, the latter might be advantageous in designing IT decision support systems. The framework of fuzzy lattices is outlined next, followed by a presentation of the *σ–FLNMAP* algorithm for classification.

3.1 The Framework of Fuzzy Lattices

A lattice L is a partially ordered set any two of whose elements have 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 a least upper bound and a greatest lower bound in L. A non-void complete lattice has a least element (O) and a greatest element (I).

It is important to point out that a lattice L could be the Cartesian product $L=L_1\times\ldots\times L_N$ of N lattices $L_1,\ldots,L_N$, namely constituent lattices. A product lattice L involving disparate constituent lattices implies the potential of dealing 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 useful function in a lattice L is a *valuation* function v: L → R defined by $v(x)+v(y) = v(x\wedge y)+v(x\vee y)$, $x,y\in L$. A valuation, is called *positive* if and only if $x<y \Rightarrow v(x)<v(y)$. A positive valuation function implies an *inclusion measure* function (σ). The latter is defined in the following.

Definition 1
An *inclusion measure* σ on a complete lattice L is a map σ: $L\times L\to[0,1]$ such that for $u,w,x\in L$ the following three axioms are satisfied:

(A1) $\sigma(x,O) = 0$, $x\neq O$.
(A2) $\sigma(x,x) = 1$, $\forall x\in L$.
(A3) $u \leq w \Rightarrow \sigma(x,u) \leq \sigma(x,w)$ – Consistency Property

Note that $\sigma(x,u)$ denotes the degree of inclusion of lattice element x in lattice element u, therefore notations $\sigma(x,u)$ and $\sigma(x\leq u)$ will be used interchangeably.

Given a positive valuation function v in a lattice L an inclusion measure can be defined by the ratio $\sigma(x\leq u) = v(u)/v(x\vee u)$ [KP00]. It has been shown in [KP00] how the aforementioned tools and notions can be extended in the complete lattice τ(L) of intervals of lattice elements.

It is known from [KP00] that an inclusion measure in a lattice L implies a *fuzzy lattice*, which is defined in the following.

Definition 2
A *fuzzy lattice* is a pair <L,μ>, where L is a crisp lattice and (L×L,μ) is a fuzzy set with membership function μ: $L\times L\to[0,1]$ such that $\mu(x,y) = 1$ if and only if $x\leq y$.

The set of all fuzzy lattices <L,σ> has been dubbed *framework of fuzzy lattices* [PK99], [KP00]. It turns out that <L,σ> is a fuzzy lattice.

This work deals with complete product lattice "unit hypercube" U= $[0,1]\times\ldots\times[0,1]$, where a constituent complete lattice is the closed interval [0,1] of real numbers. A positive valuation function in [0,1] is given by $v(x)=x$. An interval in the N-dimensional unit hypercube U corresponds to an N-dimensional hyperbox. Learning and decision making can be effected in U by hyperboxes as described in the following.

3.2 The *σ–FLNMAP* algorithm for classification

The *σ–FLNMAP* classifier is a synergy of two *σ–FLN* schemes for clustering. Critical, in the operation of scheme the *σ–FLN* is the notion *size* of an interval $x=[a,b]\in\tau(L)$.

Definition 3
Let v be a positive valuation function in a lattice L. The *size* of an interval $x=[a,b]\in\tau(L)$ is a function Z: L→R, given by $Z([a,b])= v(b)-v(a)$.

All constituent lattices L_i, i=1,…,N involved in this work are *complete lattices* [KP00]. It follows, after normalization, that a positive valuation function $v_i(.)$ takes values in the interval [0,1]. In conclusion, the critical size Z_{crit} in algorithm *σ–FLN* below equals $Z_{crit}=\frac{N(1-\rho)}{\rho}$, where $\rho\in[0.5,1]$ is called *vigilance parameter* [KP00].

Let the training data set consist of n pairs (x_i, $y_i = g(x_i)$), i = 1,...,n, where x_i is a lattice interval and g: $\tau(L)\to D$ is a *category* function, where D is a finite set of category labels. "Learning" according to *σ–FLN* is described in the following:

0. A set $H=\{H_1,\ldots,H_L\}$ of hyperboxes is given – Note that set H could be empty.

1. An input lattice interval $x= [a,b]$ is presented to the *σ–FLN.*

2. The degree of inclusion of $x= [a,b]$ is calculated for each huperbox $H_1,\ldots,H_L$.

3. Hyperboxes $H_1,\ldots,H_L$ compete over input $x = [a,b]$. Winner is the hyperbox H_J which includes x the most.

4. Winner hyperbox H_J is augmented tentatively to hyperbox $H_J\vee_{\tau(L)}x$ so as to include input x.

Table 1: Dataset attributes and various statistics

attribute	data type	Training Data Set mean	Training Data Set std.dev.	Testing Data Set mean	Testing Data Set std.dev.
SO_2 (*Sulfur dioxide*)	real	5.08	4.66	5.87	5.84
NO (*Nitrogen oxide*)	real	5.09	4.54	6.49	7.55
NO_2 (*Nitrogen dioxide*)	real	9.55	7.74	6.98	5.80
NO_x (*Nitrogen oxides*)	real	17.09	12.61	15.83	13.97
VEL (*Wind velocity*)	real	2.19	1.56	1.91	1.25
TEM (*Temperature*)	real	18.34	6.77	23.41	7.60
HR (*Relative humidity*)	real	60.23	22.93	82.59	17.54
O_3 (*ozone level*)	'low'	*6,265 records*		*12,255 records*	
O_3 (*ozone level*)	'med'	*4,761 records*		*5,138 records*	

5. If $Size(H_J \vee_{\tau(L)} x)$ is smaller than a user-defined size Z_{crit} then hyperbox H_J is replaced by $H_J \vee_{\tau(L)} x$. Otherwise, reset occurs and the next winner is selected among the remaining (non-reset) hyperboxes.

6. If all the hyperboxes have been reset then input $x = [a,b]$ is learned as a new hyperbox.

Two *σ–FLN* schemes for clustering, namely *σ–FLN*$_a$ and *σ–FLN*$_b$, are employed synergistically to produce a *σ–FLNMAP* classifier. In particular, the *σ–FLN*$_a$ clusters the input data, *σ–FLN*$_b$ clusters the corresponding "category" data, whereas an intermediate MAP field is used to associate clusters in *σ–FLN*$_a$ with clusters in *σ–FLN*$_b$ as detailed in [KP00]. In conclusion, the MAP field assigns a category label (in *σ–FLN*$_b$) to a data cluster (in *σ–FLN*$_a$).

The *σ–FLNMAP* classifier learns in "one-pass" through the training data. The vigilance parameter $\rho \in [0,1]$ specifies the *σ–FLNMAP*'s sensitivity. As ρ increases, the calculated hyperboxes' size decreases. At the end of "learning", there exist M hyperboxes partitioned in K classes, where K is the cardinality of the finite set *D* of category labels. As soon as "learning" is complete, the set of calculated hyperboxes can be implemented as either a decision tree or as a neural network [PK99, KP00].

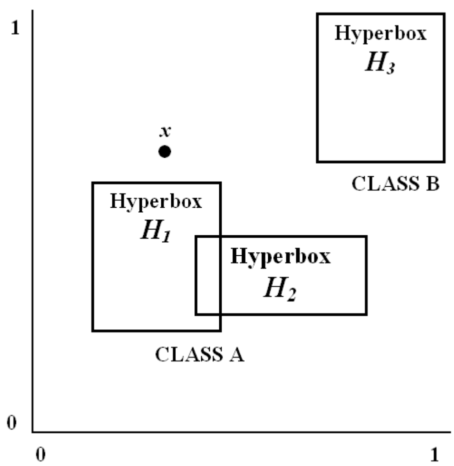

Figure 2: Learning in the FLN framework

Figure 2 shows three hyperboxes H_1, H_2 and H_3 representing two different classes A and B. Let a lattice interval *x* lie outside both classes A and B. Based on the calculated inclusion measure values $\sigma(x \le H_1)$, $\sigma(x \le H_2)$ and $\sigma(x \le H_3)$, input *x* is assigned to the class whose label is attached to the hyperbox in which *x* is included most.

A Fuzzy Lattice can be represented using its least element (*O*) and its greatest element (*I*). A product lattice is defined using the least element and greatest element for each one of the constituent lattices. For example, in Figure 2, Hyperbox H_3 can be defined as: $H_3 = L_1 \times L_2 = [0.65,1] \times [0.7,1]$, where L_1 and L_2 are the constituent lattices in the two-dimensions.

In the following section experimental results on real world data are presented and compared.

4 Experiments on Environmental Data

4.1 Data Preprocessing

The *σ–FLNMAP* classifier was applied on a dataset of meteorological and air-pollutants measurements for estimating ozone concentration levels. The dataset, labeled C2ONDA01 and supplied by CEAM, contained data from a meteorological station in the district of Valencia, Spain. More specifically, several meteorological attributes and air-pollutants values were recorded on a quarter-hourly basis during the year 2001.

There are approximately 35,000 records, with seven attributes per record plus a class attribute. After removing records with missing values, the data set was split into two subsets: one subset for training and another subset for testing containing around 40% of the data and around 60% of the data, respectively. Attributes as well as various statistics for both the training data set and the testing data set are shown in Table 1.

A prediction model, be it a Neural Network, an Expert System, or a Decision Tree, is to be embedded in the *Prediction Agent* in order for the latter agent to produce accurate estimates of the ozone concentration level in the atmosphere from other pollutants and meteorological measurements, even in case the ozone sensor is out of order. Ozone level is characterized as either '*low*' or '*medium*', respectively, for values in the ranges 0 - 60μg/m^3 and 60 - 100 μg/m^3.

Table 2: Confusion Matrices for three classifiers

***σ-FLNMAP* classifier**

Records classified as:	'low'	'med'
No. records in class 'low':	11,243	1,012
No. records in class 'med':	1,904	3,234
Correctly classified records:	83.14%	

C4.5 algorithm for classification

Records classified as:	'low'	'med'
No. records in class 'low':	8,487	3,769
No. records in class 'med':	798	4,340
Correctly classified records:	73.69%	

Backpropagation Neural Networks

Records classified as:	'low'	'med'
No. records in class 'low':	9,905	2,351
No. records in class 'med':	752	4,384
Correctly classified records:	82.05%	

4.2 Experimental Results

Real time estimation of ozone concentration level from other environmental and meteorological attributes, was attained using three different classifiers:

1. The C4.5 classification algorithm;

2. Back-propagation neural networks; and

3. The *σ–FLNMAP* classifier.

The above classifiers have been selected among others because they can be easily embedded into artificial agents [MS02]. The prediction accuracy achieved using the three algorithms is presented comparatively in Table 2. In this table, the Confusion Matrix for each one of the algorithms is presented along with the percentage of the correctly classified records.

The C4.5 algorithm for classification produced a decision tree whose nodes specify inequalities for the values of environmental attributes and the tree leaves specify an output class. After several training sessions for various values of the C4.5 algorithm parameters, the "best" pruned decision tree included 66 leaves and and had a size of 131. The classification performance on the testing data set was 73.65%. The corresponding confusion matrix for the testing data set, is shown in Table 2.

Back-propagation neural networks were employed next. Various network architectures were tested characterized by different activation functions and different numbers of hidden neurons. The best performance was obtained by a neural network with 11 hidden neurons, linear transfer functions for the hidden layer neurons, sigmoid transfer functions for the output layer neurons, and the resilient backpropagation algorithm. The corresponding classification success was 82.05% on testing data.

Learning with the *σ–FLNMAP* classifier was rather simple and fast, compared to back propagation training, as the *σ–FLNMAP* learns in one pass through the training data. Furthemore, there is only one parameter to be tuned (the vigilance parameter ρ). Vigilance ρ was given several values. The best results were obtained for ρ = 0.59. The corresponding confusion matrix is shown in Table 2. The classification performance on the testing data set was 83.14%.

The rules extracted with *σ–FLNMAP* are represented as hyperboxes. There were only three rules induced

Table 3: Rules induced empirically from the training data by the *σ-FLNMAP* classifier

Rule#	SO_2	*NO*	NO_2	NO_x	*VEL*	*TEM*	*HR*	O_3 *Class*
0	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'
1	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'
2	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'

empirically from the training data set and they are shown in Table 3. The extracted rules can be also expressed as "if-then" statements. *Rule#0* as an "if-then statement" is shown in Figure 3.

```
IF SO2 is in range [3.0,87.0]
AND NO is in range [2.0,74.0]
AND …
AND HR is in range [8.0,99.0]
THEN O3 IS CLASSIFIED IN Class 'Low'
```

Figure 3: *Rule#0* represented as an "if then" statement.

4.3 Discussion

Even though the classification results by *σ–FLNMAP* are marginally better than those by Backpropagation, and they clearly outperform the classification results by C4.5, additional advantages of the *σ–FLNMAP* classifier include faster training in one pass through the data. Also, very few rules/hypercubes (only 3) were induced from the training data. Note that rules represented as hyperboxes, can be easily comprehended by human experts, whereas conventional backpropagation neural networks are "black boxes" whose answers are not easily deciphered.

5 Conclusion - Future Work

In this work the problem of air quality assessment was addressed in real-time as a classification problem with satisfactory results. The *σ–FLNMAP* classifier resulted in comparatively good results. Furthermore, rules extracted by *σ–FLNMAP* in the form of hypercubes are easily understood by humans and can be embedded on software agents in order to support and generalize air quality assessment.

Future work will focus in two directions. First, the decision support system O_3RTAA for assessing air quality in real time will be further developed. Second, the applicability of classifier *σ–FLNMAP* on other types of agents beyond the O_3RTAA system will be studied. Note that the applicability of *σ–FLNMAP* on partially (lattice) ordered data is expected to be useful in various IT applications.

Acknowledgements

The authors would like to express their appreciation a) to the IDI-EIKON team for their efforts within the Agent Academy project towards the design and development of the O_3RTAA system, and b). to CEAM for supplying the C2ONDA01 dataset.

This work has been partially supported by the European Commission IST Programme, under the contract IST-2000-31050.

References

[AA00] The Agent Academy Consortium. *Agent Academy: A Data Mining Framework for Training Intelligent Agents*, 2000. http://www.AgentAcademy.iti.gr

[FA96] S. Fine and J. Ambrosiano. The Environmental Decision Support System: Overview and air quality application. Symposium on Environmental Applications, Jan. 28-Feb 2, Atlanta, GA, American Meteorological Society, pp.152-157, 1996.

[KP00] V. G. Kaburlasos and V. Petridis. Fuzzy lattice neurocomputing (FLN) models. *Neural Networks*, 13(10):1145-1170, 2000.

[KP02] V. G. Kaburlasos and V. Petridis. Learning and decision-making in the framework of fuzzy lattices. In *New Learning Paradigms in Soft Computing*, L.C. Jain and J. Kacprzyk (editors), vol. 84, pp. 55-96, 2002, Heidelberg, Germany: Physica-Verlag.

[KA99] E. Kalapanides and N. Avouris. Applying Machine Learning Techniques in Air Quality Prediction. In Proc. *ACAI 99*, pp. 58-64, Chania, July 1999.

[KS02] D. D. Kechagias, A. L. Symeonidis, P. A. Mitkas and M. Alborg. Towards improving multi-agent simulation in safety management and hazards control environments. In *AIS'2002: AI, Simulation and Planning in High Autonomous Systems,* F. Barros, N. Giambiasi (editors), pp.335-340, Lisbon, Portugal, SCS press.

[MS02] P. Mitkas, A. Symeonidis, D. Kechagias, I. Athanasiadis, G. Laleci, G. Kurt, Y. Kabak, A. Acar, A. Dogac. An Agent Framework for Dynamic Agent Retraining: Agent Academy. In *eBusiness and eWork 2002 (e2002) 12th annual conference and exhibition*, 16-18 October 2002, Prague, Czech Republic.

[PK99] V. Petridis and V. G. Kaburlasos. Learning in the framework of fuzzy lattices. *IEEE Transactions on Fuzzy Systems*, 7(4):422-440, 1999. Errata in *IEEE Transactions on Fuzzy Systems*, vol. 8, no. 2, p. 236, 2000.