Journal of Machine Learning Research 4 (2003) 17-37 Submitted 1/02; Revised 8/02; Published 4/03

FINkNN: A Fuzzy Interval Number k-Nearest Neighbor Classifier for Prediction of Sugar Production from Populations of Samples

Vassilios Petridis PETRIDIS@ENG.AUTH.GR
Division of Electronics & Computer Engineering
Department of Electrical & Computer Engineering
Aristotle University of Thessaloniki
GR-54006 Thessaloniki, Greece

Vassilis G. Kaburlasos VGKABS@TEIKAV.EDU.GR
Division of Computing Systems
Department of Industrial Informatics
Technological Educational Institute of Kavala
GR-65404 Kavala, Greece

Editor: Haym Hirsh

Abstract

This work introduces *FINkNN*, a k-nearest-neighbor classifier operating over the metric lattice of conventional interval-supported convex fuzzy sets. We show that for problems involving populations of measurements, data can be represented by fuzzy interval numbers (*FIN*s) and we present an algorithm for constructing *FIN*s from such populations. We then present a lattice-theoretic metric distance between *FIN*s with arbitrary-shaped membership functions, which forms the basis for *FINkNN*'s similarity measurements. We apply *FINkNN* to the task of predicting annual sugar production based on populations of measurements supplied by Hellenic Sugar Industry. We show that *FINkNN* improves prediction accuracy on this task, and discuss the broader scope and potential utility of these techniques.

Keywords: k Nearest Neighbor (kNN), Fuzzy Interval Number (*FIN*), Metric Distance, Classification, Prediction, Sugar Industry.

1 Introduction

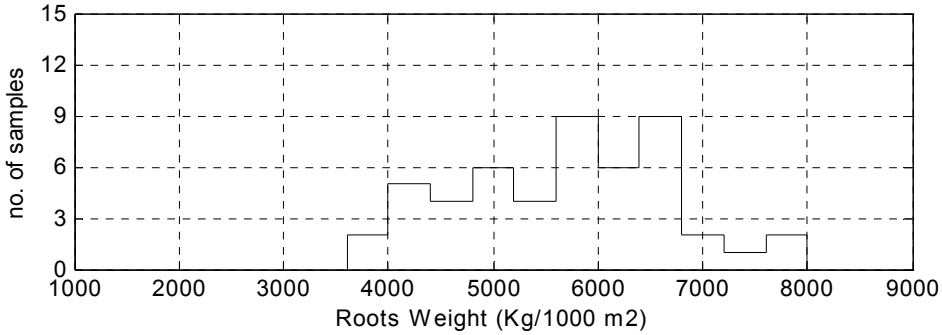
Learning and decision-making are often formulated as problems in *N*-dimensional Euclidean space $\mathbb{R}^N$, and numerous approaches have been proposed for such problems (Vapnik, 1988; Vapnik & Cortes, 1995; Schölkopf et al., 1999; Ben-Hur et al., 2001; Mangasarian & Musicant, 2001; Citterio et al., 1999; Ishibuchi & Nakashima, 2001; Kearns & Vazirani, 1994; Mitchell, 1997; Vidyasagar, 1997; Vapnik, 1999; Witten & Frank, 2000). Nevertheless, data representations other than flat, attribute-value representations arise in many applications (Goldfarb, 1992; Frasconi et al., 1998; Petridis & Kaburlasos, 2001; Paccanaro & Hinton, 2001; Muggleton, 1991; Hutchinson & Thornton, 1996; Cohen, 1998; Turcotte et al., 1998; Winston, 1975). This paper considers one such case, in which data take the form of populations of measurements, and in which learning takes place over the metric product lattice of conventional interval-supported convex fuzzy sets.

Our testbed for this research concerns the problem of predicting annual sugar production based on populations of measurements involving several production and meteorological variables supplied by the Hellenic Sugar Industry (HSI). For example, a population of 50 measurements, which correspond to the Roots Weight (RW) production variable from the HSI domain is shown in Figure 1. More specifically, Figure 1(a) shows 50 measurements on the real x-axis whereas

© 2003 Vassilios Petridis and Vassilis G. Kaburlasos

Figure 1(b) shows, in a histogram, the distribution of the 50 measurements in intervals of 400 Kg/1000 m^2. Previous work on predicting annual sugar production in Greece replaced a population of measurements by a single number, most typically the average of the population. Classification was performed using methods applicable to *N*-dimensional data vectors (Stoikos, 1995; Petridis et al., 1998; Kaburlasos et al., 2002).

(a)



(b)

Figure 1: A population of 50 measurements which corresponds to Roots Weight (RW) production variable from the HSI domain.
(a) The 50 RW measurements are shown along the x-axis.
(b) A histogram of the 50 RW measurements in steps of 400 Kg/1000 m^2.

In previous work (Kaburlasos & Petridis, 1997; Petridis & Kaburlasos, 1999) the authors proposed moving from learning over the Cartesian product $\mathsf{R}^N=\mathsf{R}\times...\times\mathsf{R}$ to the more general case of learning over a product lattice domain $\mathsf{L}=\mathsf{L}_1\times...\times\mathsf{L}_N$ (where R represents the special case of a totally ordered lattice), enabling the effective use of disparate types of data in learning. For example, previous applications have dealt with vectors of numbers, symbols, fuzzy sets, events in a probability space, waveforms, hyper-spheres, Boolean statements, and graphs (Kaburlasos & Petridis, 2000, 2002; Kaburlasos et al., 1999; Petridis & Kaburlasos, 1998, 1999, 2001). This work proposes to represent populations of measurements in the lattice of fuzzy interval numbers (*FIN*s). Based on results from lattice theory, a metric distance d_K is then introduced for *FIN*s with arbitrary-shaped membership functions. This forms the basis for the k-nearest-neighbor classifier *FINkNN* (Fuzzy Interval Number k-Nearest Neighbor), which operates on the metric product lattice F^N, where F denotes the set of conventional interval-supported convex fuzzy sets.

This work shows that lattice theory can provide a useful metric distance on the collection of conventional fuzzy sets defined over the real number universe of discourse. In other words, the learning domain in this work is the collection of conventional fuzzy sets (Dubois & Prade, 1980; Zimmerman, 1991). We remark that even though the introduction of fuzzy set theory (Zadeh,

1965) made an explicit connection to standard lattice theory (Birkhoff, 1967), to our knowledge no widely accepted lattice-inspired tools have been crafted in fuzzy set theory. This work explicitly employs results from lattice theory to introduce a useful metric distance d_K between fuzzy sets with arbitrary shaped membership functions.

Various distance measures have previously been proposed in the literature involving fuzzy sets. For instance, in Klir & Folger (1988) Hamming, Euclidean, and Minkowski distances are shown to measure the degree of fuzziness of a fuzzy set. The Hausdorf distance is used in Diamond & Kloeden (1994) to compute the distance between classes of fuzzy sets. Also, metric distances have been used in various problems of fuzzy regression analysis (Diamond, 1988; Yang & Ko, 1997; Tanaka & Lee, 1998). Nevertheless, all previous metric distances are restricted because they only apply to special cases, such as between fuzzy sets with triangular membership functions, between whole classes of fuzzy sets, etc. The metric distance function d_K introduced in this work can compute a unique distance for any pair of fuzzy sets with arbitrary-shaped membership functions. Furthermore the metric d_K is used here specifically to compute a distance between two populations of samples/measurements, and is shown to result in improved predictions of annual sugar production.

The layout of this work is as follows. Section 2 delineates an industrial problem of prediction based on populations of measurements. Section 3 presents the CALFIN algorithm for constructing a *FIN* from a population of measurements. Section 4 presents mathematical tools introduced by Kaburlasos (2002), including convenient geometric illustrations on the plane. Section 5 introduces *FINkNN*, a k-nearest-neighbor (*kNN*) algorithm for classification in metric product-lattice F^N of Fuzzy Interval Numbers (*FIN*s). *FINkNN* is employed in Section 6 on a real task, prediction of annual sugar production. Concluding remarks as well as future research are presented in section 7. Appendix A shows useful definitions in a metric space, furthermore Appendix B describes a connection between *FIN*s and probability density functions (*pdf*s).

2 An Industrial Yield Prediction Problem

The amount of sugar required for the needs of the Greek market is supplied, at large, by the production of Hellenic Sugar Industry (HSI). Sugar is produced in Greece from an annual (in farm practicing) plant, namely *Beta Vulgaris L* or simply *sugar-beet*. An early season accurate prediction of the annual production of sugar allows for both production planning and timely decision-making to fill efficiently the gap between supply and demand of sugar. An algorithmic prediction of annual sugar production can be effected based on populations of measurements involving both *production* and *meteorological* variables as explained below.

2.1 Data Acquisition

Sample measurements of ten *production variables* and eight *meteorological variables* were available in this work for eleven years from 1989 to 1999 from three agricultural districts in central and northern Greece, namely *Larisa*, *Platy*, and *Serres*. Tables 1 and 2 show, respectively, the production variables and the meteorological variables used in this work. Sugar production was calculated as the product *POL*RW*. The production variables were sampled every 20 days in a number of pre-specified *pilot fields* per agricultural district, whereas the meteorological variables were sampled daily in one local meteorological station per agricultural district. *Production* and *meteorological* variables are jointly called here *input variables*. The term *population of measurements* is used here to denote either 1) a number of production variable samples obtained during 20 days from each pilot field in an agricultural district, or 2) a collection of meteorological variable samples obtained daily during the aforementioned 20 days.

	Production Variable Name	Unit
1	Average Root Weight	*g*
2	*POL* - percentage of sugar in fresh root weight	-
3	α-amino-Nitrogen (α-*N*)	*meq*/100*g* root
4	Potassium (*K*)	*meq*/100*g* root
5	Sodium (*Na*)	*meq*/100*g* root
6	Leaf Area Index (*LAI*) - leaf area per field area ratio	-
7	*TOP*: plant top weight	*kg*/1000 m^2
8	Roots Weight (*RW*)	*kg*/1000 m^2
9	Nitrogen-test (*N-test*) - NO_3-*N* content in pedioles	$mg.kg^{-1}$
10	the Planting Date	-

Table 1: Production variables used for Prediction of Sugar Production.

	Meteorological Variable Name	Unit
1	Average (daily) Temperature	^oC
2	Maximum (daily) temperature	^oC
3	minimum (daily) Temperature	^oC
4	Relative Humidity	-
5	Wind Speed	*miles*/*hour*
6	Daily Precipitation	*mm*
7	Daily Evaporation	*mm*
8	Sunlight	*hours*/*day*

Table 2: Meteorological variables used for Prediction of Sugar Production.

2.2 Algorithmic Prediction of Sugar Production

Prediction of sugar production is made on the basis of the trend in current year compared to the corresponding trend in previous years. In previous work a population of measurements was typically replaced by a single number, the average value of the population. However, using the average value of a population of measurements in a prediction model can be misleading. For instance, two different daily precipitation patterns in a month may be characterized by identical average values, nevertheless their effect on the annual sugar production level might be drastically different. Previous annual sugar yield prediction models in Greece include neural networks (Stoikos, 1995), interpolation-, polynomial-, linear autoregression- and neural-predictors (Petridis et al., 1998), and intelligent clustering techniques (Kaburlasos et al., 2002). The best sugar prediction accuracy of 5% was reported in Kaburlasos et al. (2002).

2.3 Prediction by Classification

In order to capture to the fullest the diversity of a whole population of measurements this work proposes representing a population of measurements by a *FIN* (Fuzzy Interval Number) instead of representing it by a single number. Prediction is then made by classification.

In line with the common practice by the agriculturalists at the HSI, the goal in this work was to achieve prediction of sugar production by classification in one of the classes “good”, “medium” or “poor”. In particular, the goal here was to predict the sugar production level in

September based on data available by the end of July. The characterization of a sugar production level (in Kg/1000 m^2) as "good", "medium" or "poor" was not identical for different agricultural districts as shown in Table 3 due to the different sugar production capacities of the corresponding agricultural districts. For instance, "poor sugar production" for Larisa means 890 Kg/1000 m^2, whereas "poor sugar production" for Serres means 980 kg/1000 m^2. (Table 3 contains approximate values provided by an expert agriculturalist.)

Sugar Production Level	Agricultural District		
	Larisa	Platy	Serres
"good"	1040	1045	1165
"medium"	970	960	1065
"poor"	890	925	980

Table 3: Annual sugar production levels (in Kg/1000 m^2) for "good", "medium", and "poor" years, in three agricultural districts.

2.4 A Driving Idea for Prediction by Classification

Suppose that populations of measurements for various input variables are given for a year whose (unknown) sugar production level is to be predicted. The question is to predict the unknown sugar production level based on populations of measurements of other years whose sugar production level is known. The driving idea for prediction by classification in this work is the following. Compute a distance between populations of measurements, which correspond to a year, and populations of measurements, which correspond to the other years; then predict a sugar production level similar to the nearest year's (known) sugar production level.

There are two issues which need to be addressed for effecting the aforementioned prediction-by-classification. First, there is a representation issue. Second, there is an issue of defining a suitable distance. The first issue is addressed in section 3 where a population of measurements is represented by a *FIN* (*F*uzzy *I*nterval *N*umber); for instance, Figure 2 shows four *FIN*s, namely *MT89*, *MT91*, *MT95* and *MT98*, constructed from populations of 31 samples/measurements of the maximum daily temperatures (in centigrades) during the month of July in years 1989, 1991, 1995 and 1998 in the Larisa agricultural district. The second issue above is addressed in section 4 by a metric distance between fuzzy sets (*FIN*s) with arbitrary-shaped membership functions.

3 Algorithm CALFIN for Constructing a *FIN* from a Population of Measurements

Consider a population of n samples/measurements stored *incrementally* in vector $x= [x_1,\ldots,x_n]$, that is $x_1 \le x_2 \le \ldots \le x_n$. Algorithm CALFIN in Figure 3, in pseudo-code format, shows a recursive calculation of a *FIN* from vector x.

We remark that the median *median*(x) of a vector $x= [x_1, x_2,\ldots,x_n]$ of (real) numbers is a number such that half of the n entries $x_1, x_2,\ldots,x_n$ of vector x are smaller than *median*(x) and the other half ones are larger than *median*(x). For example, *median*([1, 3, 7])= 3, whereas the *median*([-1, 2, 6, 9]) might be any number in the interval [2, 6] for instance *median*([-1, 2, 6, 9])= (2+6)/2= 4.

The operation of algorithm CALFIN is explained in the following. Given a population of measurements stored incrementally in vector $x= [x_1, x_2,\ldots,x_n]$, algorithm CALFIN returns two vectors: 1) vector *pts*, and 2) vector *val*, the latter vectors represent a *FIN*. More specifically,

vector *pts* holds the *abscissae* whereas vector *val* holds the *ordinate* values of the corresponding *FIN*'s fuzzy membership function. Step-1 in Figure 3 computes vector *pts*; by construction, |*pts*| equals the smallest power of 2 which is larger than |*x*| (minus one). Step-3 computes vector *val*. By construction, a *FIN* attains its maximum value of 1 at one point.

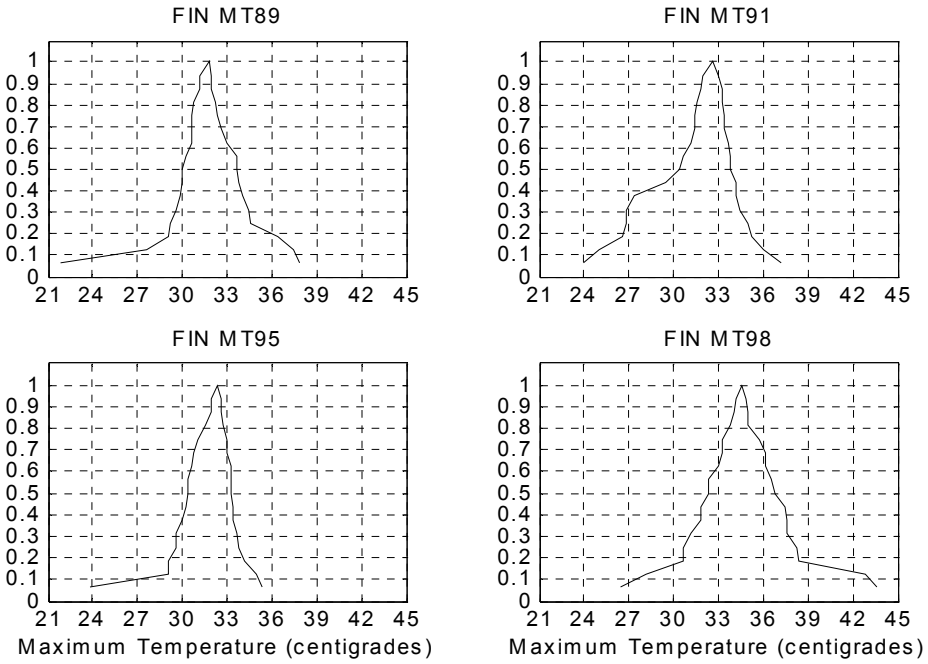


Figure 2: *FIN*s *MT89*, *MT91*, *MT95* and *MT98* constructed from maximum daily temperatures during July in the Larisa agricultural district, Greece.

```
Step-1: function abscissae(x)
        {  if (n ≠ 1)
              med ← median(x)
              x_left ← left half of vector x          % all numbers in x less-than med
              x_right ← right half of vector x        % all numbers in x larger-than med
              abscissae(x_left)
              abscissae(x_right)
           endif
           return med in vector pts
        }
Step-2: Sort vector pts incrementally.
Step-3: Let |pts| denote the cardinality of vector pts. Store in vector val, |pts|/2 numbers
        from 0 up to 1 in steps of 2/|pts| followed by another |pts|/2 numbers from 1
        down to 0 in steps of 2/|pts|.
```

Figure 3: Algorithm CALFIN above computes a Fuzzy Interval Number (*FIN*) from a population of measurements stored incrementally in vector *x*.

An application of algorithm CALFIN on the population of measurements shown in Figure 1(a) is illustrated in Figure 4. More specifically, a *FIN* is computed in Figure 4(b2) from a population of 50 samples/measurements of the Roots Weight (RW) input variable from 50 pilot fields in the last 20 days of July 1989 in the Larisa agricultural district. Identical figures Figure 4(a1) and Figure 4(a2) show the corresponding 63 *median* values computed in vector *pts* by algorithm CALFIN. Figure 4(b1) shows, in a histogram, the distribution of the 63 median values in intervals of 400 Kg/1000 m^2. Furthermore, Figure 4(b2) shows the ordinate values in vector *val* versus the abscissae values in vector *pts*.

A motivation for proposing algorithm CALFIN to represent a population of numeric data by a fuzzy set (*FIN*) is that algorithm CALFIN guarantees construction of convex fuzzy sets which comply with definition 4.2 in section 4, and thus proposition 4.4 can be used for computing a metric distance between two fuzzy sets with arbitrary-shaped membership functions. Any other algorithm that guarantees construction of convex fuzzy sets would also have this property. Finally, we point out that there is a one-one correspondence between *FIN*s constructed by algorithm CALFIN and probability density functions (*pdf*s). This connection is explained further in Appendix B.

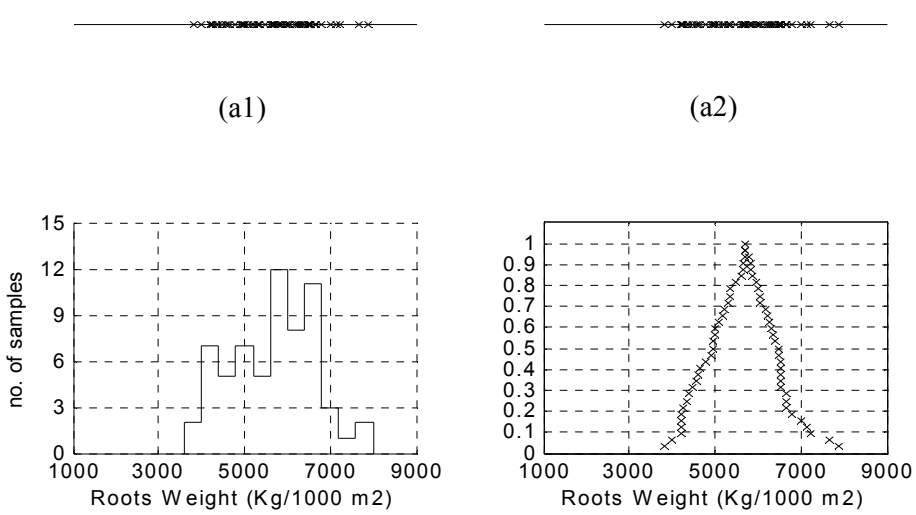


Figure 4: Calculation of a *FIN* from a population of samples/measurements.

(a1), (a2) 63 median values in vector *pts* computed by algorithm CALFIN from the 50 samples shown in Figure 1(a).

(b1) A histogram of the 63 median values in Figure 4(a1) in steps of 400 Kg/1000 m^2.

(b2) The 63 median values of vector *pts* in Figure 4(a2) have been mapped to the corresponding entries of vector *val* computed by algorithm CALFIN.

4 Metric Lattice F of Fuzzy Interval Numbers (*FIN*s)

A grounded example for computing a distance between *FIN*s is shown in the following. In particular, Figure 5 shows four *FIN*s, namely *RW89*, *RW91*, *RW95* and *RW98*, constructed by

algorithm CALFIN from populations of the Roots Weight (RW) input variable. We would like to quantify the proximity of two years based on the corresponding populations of measurements. Table 4 shows metric distances (d_K) computed between the abovementioned *FIN*s. The remaining of this section details the analytic computation of a metric distance d_K between arbitrary-shaped *FIN*s following the original work by Kaburlasos (2002).

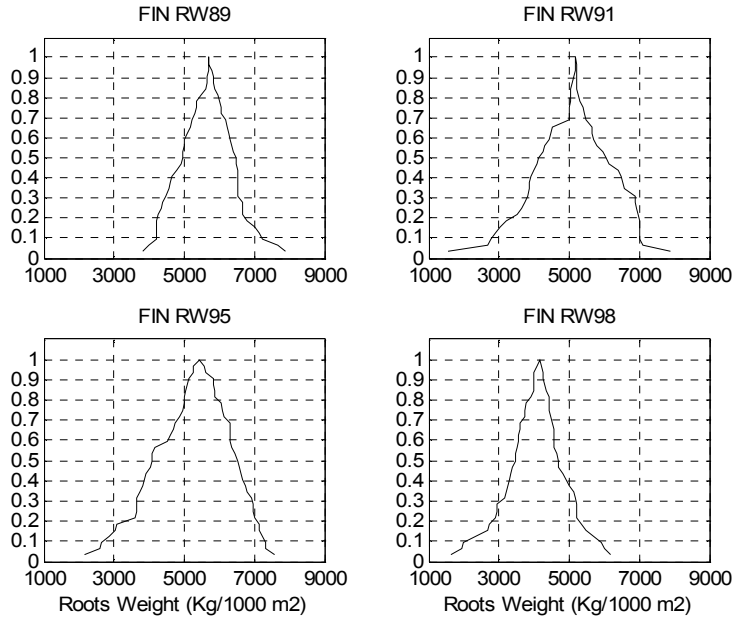


Figure 5: *FIN*s *RW89*, *RW91*, *RW95* and *RW98* were constructed from samples of Roots Weight (RW) production variable in 50 *pilot fields* during the last 20 days of July in the Larisa agricultural district, Greece.

FIN	*RW89*	*RW91*	*RW95*	*RW98*
RW89	0	541	349	1576
RW91	541	0	286	1056
RW95	349	286	0	1292
RW98	1576	1056	1292	0

Table 4: Distances d_K between *FIN*s *RW89*, *RW91*, *RW95* and *RW98* in (Figure 5)

The basic idea for introducing a metric distance between arbitrary-shaped *FIN*s is illustrated in Figure 6, where *FIN*s *RW89* and *RW91* are shown. Recall that a *FIN* is constructed such that any horizontal line ε_h, $h \in [0,1]$ intersects a *FIN* at exactly two points – without loss of generality only for h=1 there exists a single intersection point. A horizontal line ε_h at h=0.8 results in a "pulse" of height h=0.8 for a *FIN* as shown in Figure 6. More specifically, Figure 6 shows two pulses for the two *FIN*s *RW89* and *RW91*, respectively. The aforementioned pulses are called *generalized intervals of height h*=0.8. Apparently, if a metric distance could be defined between two generalized intervals of height h then a metric distance is implied between two *FIN*s simply by computing the corresponding definite integral from h=0 to h=1.

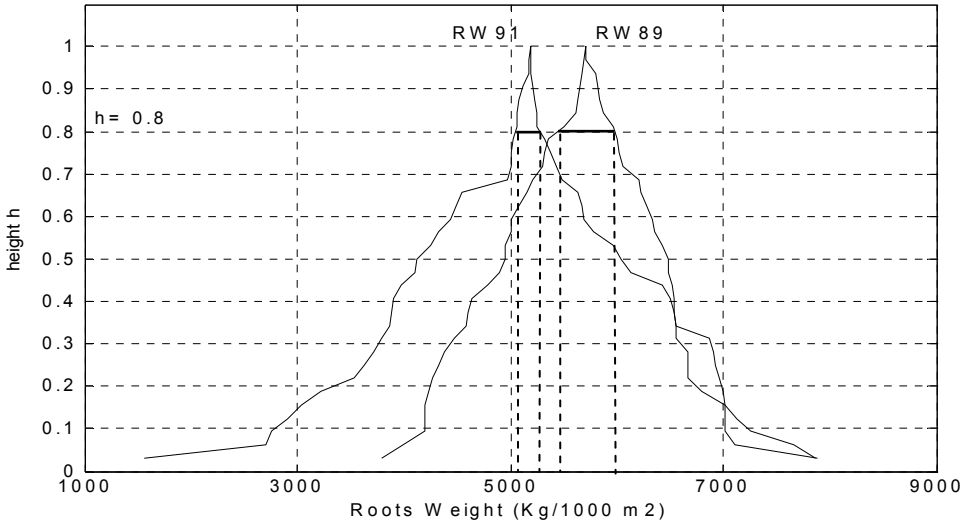


Figure 6: Generalized intervals of height h=0.8 which correspond to *FIN*s *RW89* and *RW91*.

4.1 Metric Lattices M^h of Generalized Intervals

Consider the notion *generalized interval* (*of height h*).

Definition 4.1 A *generalized interval of height h* is a real function given either by $\mu_{[x_1,x_2]^h_+}(x)=\begin{cases} h, & x_1 \le x \le x_2 \\ 0, & otherwise \end{cases}$, or by $\mu_{[x_1,x_2]^h_-}(x)=\begin{cases} -h, & x_1 \le x \le x_2 \\ 0, & otherwise \end{cases}$, where $h\in(0,1]$ is called *height* of the corresponding generalized interval.

A generalized interval may simply be denoted by $[x_1,x_2]^h_+$ (**positive** generalized interval) or by $[x_1,x_2]^h_-$ (**negative** generalized interval). The collection of generalized intervals of height h will be denoted by P^h. An ordering relation can be introduced in P^h as follows.

(R1) $[a,b]^h_+ \le_{\mathsf{P}^h} [c,d]^h_+ \Leftrightarrow c \le a \le b \le d$,

(R2) $[a,b]^h_- \le_{\mathsf{P}^h} [c,d]^h_- \Leftrightarrow [c,d]^h_+ \le_{\mathsf{P}^h} [a,b]^h_+$, and

(R3) $[a,b]^h_- \le_{\mathsf{P}^h} [c,d]^h_+ \Leftrightarrow [a,b]\cap[c,d]\neq\varnothing$, where $[a,b]$ and $[c,d]$ denote conventional intervals (sets) of numbers.

The ordering relation $\le_{\mathsf{P}^h}$ is a *partial ordering relation*, furthermore the set P^h is a lattice[1].

[1] Recall that a relation is called *partial ordering relation* if and only if it is 1) *reflexive* ($x\le x$), 2) *antisymmetric* ($x\le y$ and $y\le x$ imply $x=y$), and 3) *transitive* ($x\le y$ and $y\le z$ imply $x\le z$). A *lattice* L is a partially ordered set any two of whose elements have a unique greatest lower bound or *meet* denoted by $x\wedge_{\mathsf{L}} y$, and a unique least upper bound or *join* denoted by $x\vee_{\mathsf{L}} y$.

The set M^h with elements $[a,b]^h$ as described in the following is also a lattice: (1) if $a<b$ then $[a,b]^h \in \mathsf{M}^h$ corresponds to $[a,b]_+^h \in \mathsf{P}^h$, (2) if $a>b$ then $[a,b]^h \in \mathsf{M}^h$ corresponds to $[b,a]_-^h \in \mathsf{P}^h$, and (3) $[a,a]^h \in \mathsf{M}^h$ corresponds to both $[a,a]_+^h$ and $[a,a]_-^h$ in P^h. To avoid redundant terminology, an element of M^h is called *generalized interval* as well, and it is denoted by $[a,b]^h$. Figure 7 shows exhaustively all combinations for computing the lattice *join* $q_1 \vee_{\mathsf{M}^h} q_2$ and *meet* $q_1 \wedge_{\mathsf{M}^h} q_2$ for two different generalized intervals q_1, q_2 in M^h. No interpretation is proposed here for negative generalized intervals because it is not necessary. It will be detailed elsewhere how an interpretation of negative generalized intervals is application dependent.

Real function $v(.)$, defined as the area "under" a generalized interval, is a *positive valuation* function in lattice M^h therefore function $d(x,y)= v(x \vee_{\mathsf{M}^h} y)-v(x \wedge_{\mathsf{M}^h} y)$, $x,y \in \mathsf{M}^h$ defines a *metric* distance in M^h as explained in Appendix A. For example, the metric distance between the two generalized intervals $[5049, 5284]^{0.8}$ and $[5447, 5980]^{0.8}$ of height h=0.8 shown in Figure 6 equals $d([5049, 5284]^{0.8},[5447, 5980]^{0.8}])= v([5049, 5980])-v([5447, 5284])= 0.8(931)+0.8(163)= 875.2$.

Even though the set M^h of generalized intervals is a metric lattice for any $h>0$, the interest in this work is focused on metric lattices M^h with $h \in (0,1]$ because the latter lattices arise from a-cuts of convex fuzzy sets as explained below. The collection of all metric lattices M^h for h in (0,1] is denoted by M, that is $\mathsf{M}= \bigcup_{h\in(0,1]} \mathsf{M}^h$.

4.2 The Metric Lattice F of *FIN*s

A *Fuzzy Interval Number*, or *FIN* for short, is a conventional interval-supported convex fuzzy set. In order to facilitate mathematical analysis below, the following definition is proposed for a *FIN*.

Definition 4.2 A *Fuzzy Interval Number*, or *FIN*[2] for short, is a function F: $(0,1] \rightarrow \mathsf{M}$ such that $h_1 \leq h_2 \Rightarrow support(F(h_1)) \supseteq support(F(h_2))$, $0 < h_1 \leq h_2 \leq 1$.

We remark that the *support* of a generalized interval in M^h is a function which maps a generalized interval to its interval support (set), in particular $support([a,b]^h)=[a,b]$ if $a \leq b$, whereas $support([a,b]^h)=[b,a]$ if $a \geq b$. Figure 8 shows the supports $support(F(h_1))$ and $support(F(h_2))$ of two generalized intervals, respectively, $F(h_1)$ and $F(h_2)$ stemming from a *FIN* F.

The $support(F(a))$ of a generalized interval $F(a)$ equals, by definition, the a-cut Γ_a of the corresponding "fuzzy set F with membership function μ: $\mathsf{R} \rightarrow [0,1]$". Recall that an a-cut Γ_a has been defined in Zadeh (1965) as $\Gamma_a= \{x | \mu(x) \geq a\}$; that is Γ_a equals the set of real numbers x whose degree $\mu(x)$ of membership in F is greater-than or equal-to a. Apparently, an a-cut Γ_a for a *FIN* is an interval.

Let F denote the collection of *FIN*s. An ordering relation $\leq_{\mathsf{F}}$ is defined as follows.

Definition 4.3 Let $F_1, F_2 \in \mathsf{F}$, then $F_1 \leq_{\mathsf{F}} F_2$ if and only if $F_1(h) \leq_{\mathsf{M}^h} F_2(h)$, $h \in (0,1]$.

[2] We point out that the theoretical formulation presented in this work, regarding *FIN*s with negative membership functions, might be useful for interpreting significant improvements reported in Chang and Lee (1994) in fuzzy linear regression problems involving triangular fuzzy sets with negative spreads. Note that fuzzy sets with negative spreads are not regarded as fuzzy sets by some authors (Diamond and Körner, 1997).

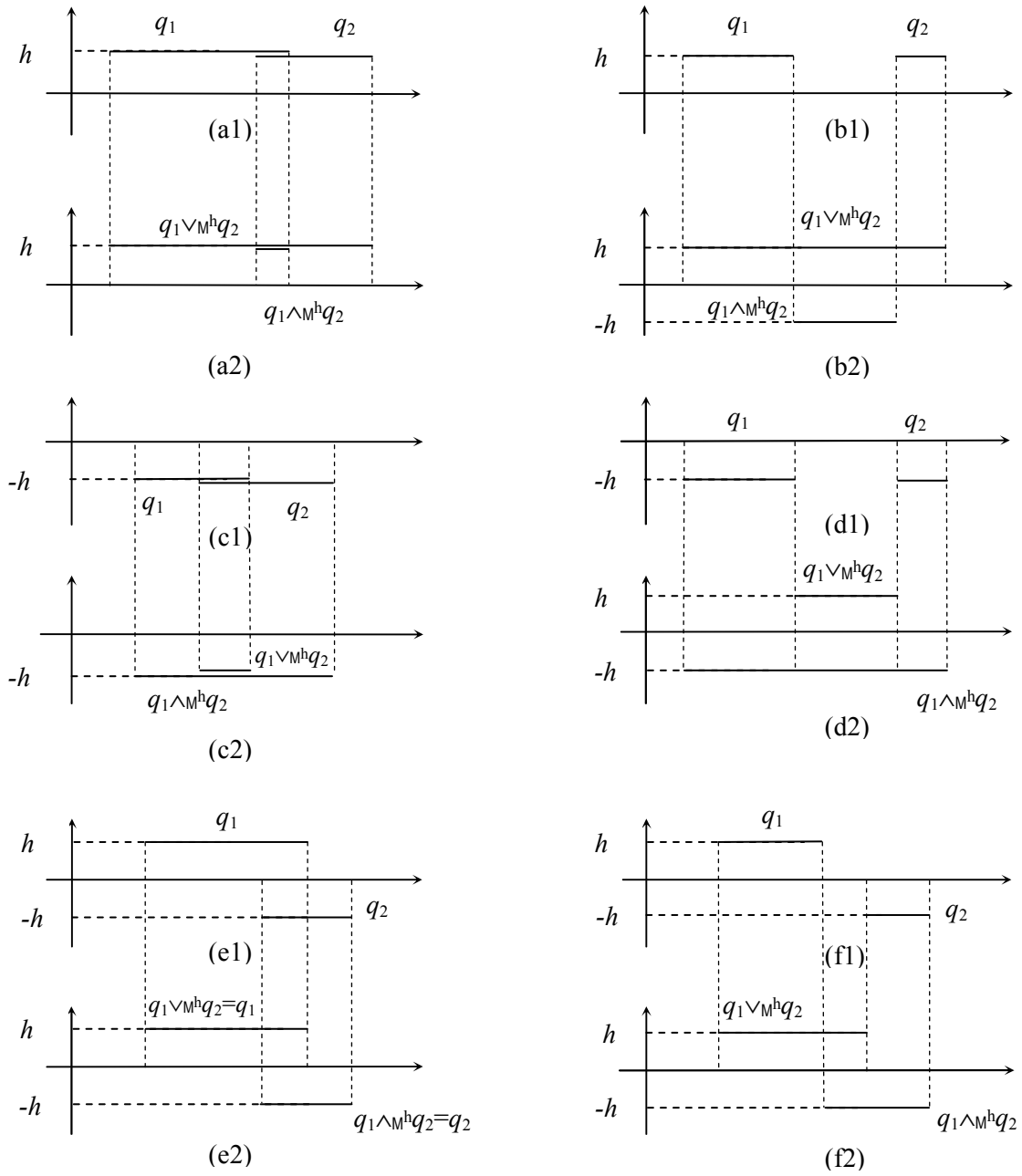


Figure 7: The *join* ($q_1 \vee_M^h q_2$) and *meet* ($q_1 \wedge_M^h q_2$) for generalized intervals $q_1, q_2 \in M^h$.

(a) "Intersecting" positive generalized intervals q_1 and q_2,
(b) "Non-intersecting" positive generalized intervals q_1 and q_2,
(c) "Intersecting" negative generalized intervals q_1 and q_2,
(d) "Non-intersecting" negative generalized intervals q_1 and q_2,
(e) "Intersecting" positive (q_1) and negative (q_2) generalized intervals, and
(f) "Non-intersecting" positive (q_1) and negative (q_2) generalized intervals.

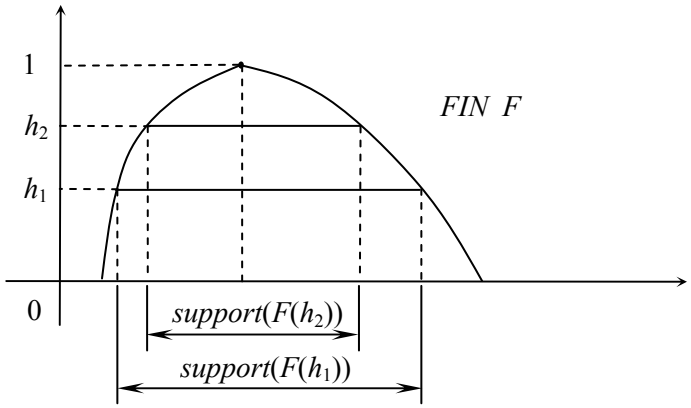


Figure 8: *FIN F*: $(0,1] \rightarrow \mathbb{M}$ maps a real number h in (0,1] to a generalized interval $F(h)$. The *domain* of function F is shown on the vertical axis, whereas the *range* of function F includes "rectangular shaped pulses" on the plane.

It has been shown that $\mathbb{F}$ is a lattice. More specifically, the lattice *join* $F_1 \vee_{\mathbb{F}} F_2$ and lattice *meet* $F_1 \wedge_{\mathbb{F}} F_2$ of two *incomparable FIN*s F_1 and F_2, i.e. neither $F_1 \leq_{\mathbb{F}} F_2$ nor $F_2 \leq_{\mathbb{F}} F_1$, are shown in Figure 9. The theoretical exposition of this section concludes in the following result.

Proposition 4.4 Let $F_1(h)$ and $F_2(h)$, $h \in (0,1]$ be *FIN*s in $\mathbb{F}$. A metric distance function d_K: $\mathbb{F} \times \mathbb{F} \rightarrow \mathbb{R}$ is given by $d_K(F_1,F_2) = \int_0^1 d(F_1(h), F_2(h))dh$, where $d(.,.)$ is the metric in lattice $\mathbb{M}^h$.

We remark that a similar metric distance between fuzzy sets has been presented and used previously by other authors (Diamond & Kloeden, 1994; Chatzis & Pitas, 1995) in a fuzzy set theoretic context. Nevertheless the calculation of $d_K(.,.)$ based on generalized intervals implies a significant capacity for "tuning" as it will be shown elsewhere. The following two examples demonstrate the computation of metric distance d_K.

Example 4.5

Figure 10 illustrates the computation of the metric distance d_K between *FIN*s *RW89* and *RW91* (Figure 10(a)), where generalized intervals *RW89*(h) and *RW91*(h) are also shown. *FIN*s *RW89* and *RW91* have been constructed from real samples of the Roots Weight (RW) production variable in the years 1989 and 1991, respectively.

For every value of the height $h \in (0,1]$ there corresponds a metric distance d(*RW89*(h),*RW91*(h)) as shown in Figure 10(b). Based on proposition 4.4 the area under the curve in Figure 10(b) equals the metric distance between *FIN*s *RW89* and *RW91*. It was calculated d_K(*RW89*,*RW91*)= 541.3.

A practical advantage of metric distance d_K is that it can capture sensibly the relative position of two *FIN*s as demonstrated in the following example.

Example 4.6

In Figure 11 distances $d_K(.,.)$ are computed between pairs of *FIN*s with triangular membership functions. In particular, in Figure 11(a) distances $d_K(F_1, H_1) \approx 5.6669$, $d_K(F_2, H_1) \approx 5$, and $d_K(F_3, H_1) \approx 4.3331$ have been computed. *FIN*s F_1, F_2, and F_3 have a common base and equal heights.

Figure 11(a) was meant to demonstrate the "common sense" results obtained analytically for metric d_K, where "the more a *FIN* F_i, i=1,2,3 leans towards *FIN* H_1" the smaller the corresponding distance d_K is. Similar results are shown in Figure 11(b), the latter has been produced from Figure 11(a) by shifting the top of *FIN* H_1 to the left. It has been computed analytically $d_K(F_1, H_2) \approx 5$, $d_K(F_2, H_2) \approx 4.3331$, and $d_K(F_3, H_2) \approx 3.6661$. Note that $d_K(F_i, H_2) \leq d_K(F_i, H_1)$, i=1,2,3 as expected by inspection because *FIN* H_2 leans more towards *FIN*s F_1, F_2, F_3 than *FIN* H_1 does. We also cite the following distances $d_K(F_1, F_2) \approx 0.6669$, $d_K(F_1, F_3) \approx 1.3339$, and $d_K(F_2, F_3) \approx 0.6669$.

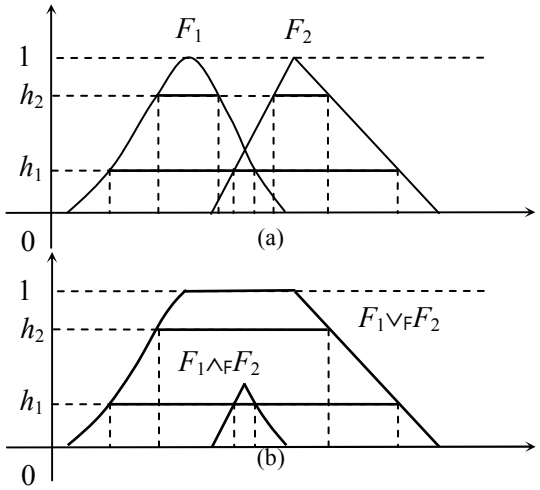


Figure 9: (a) Two *incomparable FIN*s F_1 and F_2, i.e. neither $F_1 \leq_F F_2$ nor $F_2 \leq_F F_1$.
(b) $F_1 \vee_F F_2$ is the lattice *join*, whereas $F_1 \wedge_F F_2$ is the lattice *meet* of *FIN*s F_1 and F_2.

5 *FINkNN*: A Nearest Neighbor Classifier

Let g be a *category* function g: $\mathsf{F} \rightarrow \mathsf{D}$ which maps a *FIN* in F to an element of a label set D. Classification in metric lattice (F, d_K) can be effected, first, by storing all the labeled training data pairs $(E_1, g(E_1)),\ldots,(E_n, g(E_n))$ and, second, by mapping a new *FIN* E to the category $g(E)$ which receives the majority vote among the k Nearest Neighbor (*kNN*) *FIN*s.

This work has considered N-dimensional vectors F of *FIN*s $F=(E_1,\ldots,E_N)$ where a vector component E_i, i=1,…,N corresponds to an *input* variable, i.e. a *production* variable or a *meteorological* variable. The *kNN* classifier described above has been applied, in principle, in product lattice F^N. In particular, since (F, d_K) is a metric lattice, it follows that $d^p(x,y)=\{d_K(E_1,H_1)^p+\ldots+d_K(E_N,H_N)^p\}^{1/p}$, p≥1, where $x=(E_1,\ldots,E_N), y=(H_1,\ldots,H_N)\in\mathsf{F}^N$, is a metric distance in product lattice F^N. In conclusion a *kNN* classifier, namely *FINkNN*, has been applied here in the metric lattice (F^N, d^1).

Classifier *FINkNN* has been cast in the framework of k Nearest Neighbor (*kNN*) classifiers, nevertheless *FINkNN* was applied in this work for k=1 for two reasons. First, there were only a few (11) pieces of data from 11 years partitioned in three categories and, second, k=1 gave better results than other values of k in this application. Classifier *FINkNN* is described below.

Classifier *FINkNN*

1. Store all labeled training data $(F_1, g(F_1)),\ldots,(F_n, g(F_n))$, where $F_i\in\mathsf{F}^N$, $g(F_i)\in\mathsf{D}$, i=1,…,n.
2. Classify a new datum $F\in\mathsf{F}^N$ to category $g(F_J)$, where $J=\arg\min_{i=1,\ldots,n}\{ d^1(F, F_i) \}$.

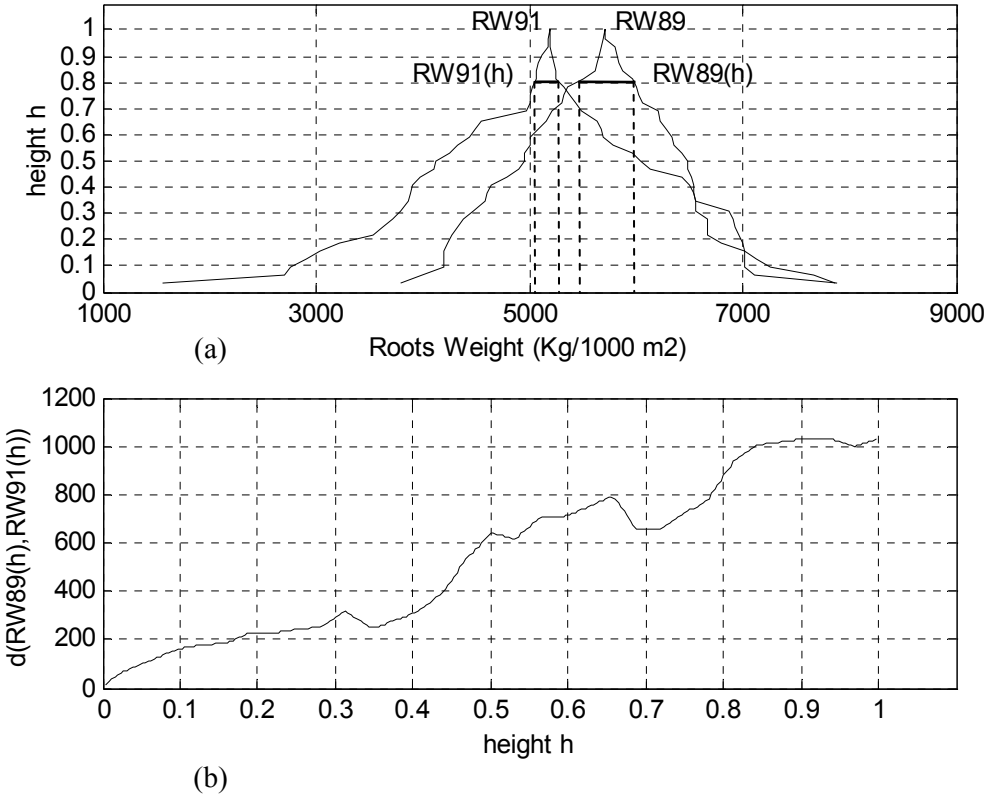


Figure 10: Computation of the metric distance $d_K(RW89,RW91)$ between *FINs RW89* and RW91.

(a) *FINs RW89* and *RW91*. Generalized intervals *RW89*(h) and *RW91*(h) are also shown.

(b) The metric distance $d(RW89(h),RW91(h))$ between generalized intervals *RW89*(h) and *RW91*(h) is shown as a function of the height $h \in (0,1]$. Metric $d_K(RW89,RW91)$= 541.3 equals the area under the curve $d(RW89(h),RW91(h))$.

Apparently, classifier *FINkNN* is "memory based" (Kasif et al., 1998) like other methods for learning including instance-based learning, case-based learning, k nearest neighbor (Aha et al., 1991; Kolodner, 1993; Dasarathy, 1991; Duda et al., 2001); the name "lazy learning" (Mitchell, 1997; Bontempi et al., 2002) has also been used in the literature for memory-based learning.

A critical difference between *FINkNN* and other memory-based learning algorithms is that the *FINkNN* can freely intermix "number attributes" and "*FIN* attributes" any place in the data, therefore "ambiguity", in a fuzzy set sense (Dubois & Prade, 1980; Ishibuchi & Nakashima, 2001; Klir & Folger, 1988; Zadeh, 1965; Zimmerman, 1991), can be dealt with.

6 Experiments and Results

In this section classifier *FINkNN* is applied on vectors of *FIN*s, the latter stem from populations of measurements of *production* and/or *meteorological* variables. The objective is prediction of annual sugar production by classification.

In the first place the significant differences in scale between different input variables, e.g. Maximum Temperature (Figure 2) versus Roots Weight (Figure 5), had to be smoothed out by a data preprocessing normalization procedure otherwise an input variable could be disregarded as

noise. Therefore a mapping to [0,1] was done by, first, translating linearly to 0 and, second, by scaling.

A "leave-one-out" series of eleven experiments was carried out such that one year among years 1989 to 1999 was left out, in turn, for testing whereas the remaining ten years were used for training.

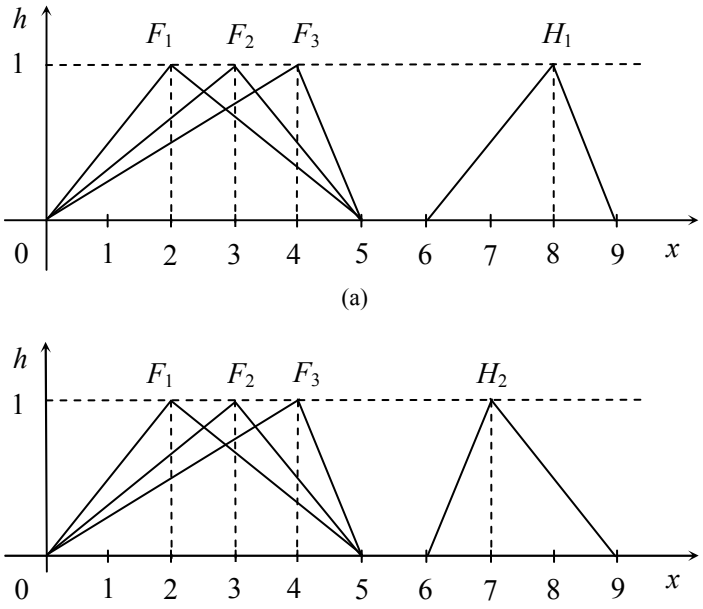


Figure 11: (a) It has been computed $d_K(F_1, H_1) \approx 5.6669$, $d_K(F_2, H_1) \approx 5$, and $d_K(F_3, H_1) \approx 4.3331$. That is "the more a *FIN* F_i, i=1,2,3 leans towards *FIN* H_1" the smaller is the corresponding distance d_K as expected intuitively by inspection.
(b) This figure has been produced from the above figure by shifting the top of *FIN* H_1 to the left. It has been computed $d_K(F_1, H_2) \approx 5$, $d_K(F_2, H_2) \approx 4.3331$, and $d_K(F_3, H_2) \approx 3.6661$.

6.1 Input Variable Selection

Prediction of sugar production was based on populations of selected input variables among 18 input variables $x_1,\ldots,x_{18}$. We remark that variable selection might itself be an important problem in both engineering system design (Hong & Harris, 2001) and in machine learning applications (Koller & Sahami, 1996; Boz, 2002). A subset of input variables have been selected based on an optimization of an *objective/fitness function* as described in this section.

Using data from ten training years a symmetric 10×10 matrix S_k of distances was calculated for each input variable x_k, k=1,…,18. Note that an entry in matrix S_k, say entry e_{ij}, i,j∈ {1,…,10}, quantifies a proximity between two years 'i' and 'j' based on the corresponding populations of input variable x_k. A sum matrix S was defined as $\mathsf{S}= \mathsf{S}_m+\ldots+\mathsf{S}_n$ for a subset {m,…,n} of input variables. A *training year* was associated with another one which corresponded to the shortest distance in a matrix S. A *contradiction* occurred if two *training years* (associated with the shortest distance) are in different categories among "good", "medium" or "poor". An *objective/fitness function* C(S) was defined as "the sum of contradictions". There follows the optimization problem: Find a subset of indices m,…,n∈ {1,…,18} such that C(S) is minimized. Apparently there exist a total number of 2^{18} subsets of indices to choose from.

The above optimization problem was dealt with using, first, a genetic algorithm (GA), second, a GA with local search and, third, human expertise, as described in the following. First, the GA implementation was a simple GA, that is no problem-specific-operators or other techniques were employed. The GA encoded the 18 input variables using 1 bit per variable resulting in a total genotype length of 18 bits. A population of 20 genotypes (solutions) was employed and it was left to evolve for 50 generations. Second, in addition to the GA above a simple local search steepest descent algorithm was employed by considering different combinations of input variables at Hamming distance one; note that the idea for local search around a GA solution has been inspired from the *microgenetic* algorithm for generalized hill-climbing optimization (Kazarlis et al., 2001). Third, a human expert selected the following input variables: variables Relative Humidity and Roots Weight were selected for *Larisa* agricultural district, variables Daily Precipitation, Sodium (Na) and Average Root Weight for *Platy*, and variables Daily Precipitation, Average Root Weight and Roots Weight were selected for the *Serres* agricultural district.

The optimization problem was solved eleven times leaving, in turn, each year from 1989 to 1999 out for testing whereas the remaining ten years were used for training. Two types of distances were considered between two populations of measurements: 1) the metric distance d_K, and 2) the "L1-distance" representing the distance between the average values of two populations.

6.2 Experiments and Comparative Results

The leave-one-out paradigm was used to evaluate comparatively *FINkNN*'s capacity for prediction-by-classification as it has been described above. After selecting a subset of input variables, prediction was effected by assigning the "left out" (*testing*) *year* to the category corresponding to the nearest training year. The experimental results are shown in Table 5.

The first line in Table 5 shows the average prediction accuracy over all *testing years* for Larisa, Platy and Serres, respectively, using algorithm *FINkNN* with expert selected input variables; line 2 shows the results using L1-distances *kNN* (with expert input variable selection). Line 3 shows the results using *FINkNN* (with GA local search input variable selection); line 4 in Table 5 shows the best results obtained using a L1-distances *kNN* (with a GA local search input variable selection). Line 5 reports the results obtained by *FINkNN* (with GA input variable selection); line 6 in Table 5 shows the results using L1-distances *kNN* (with GA input variable selection). The last three lines in Table 5 were meant to demonstrate that prediction-by-classification is *well posed* in the sense that a small prediction error is expected from the outset. In particular, selection "medium" each year resulted in error rates 5.22%, 3.44%, and 5.54% for the Larisa, Platy, and Serres factories, respectively (line 7). Line 8 shows the average errors when a year was assigned randomly (uniformly) among the three choices "good", "medium", "poor". Line 9 in Table 5 shows the *minimum prediction error* which would be obtained should each *testing year* be classified correct in its corresponding class "good", "medium" or "poor". The nearest to the latter *minimum prediction error* was clearly obtained by classifier *FINkNN* with an expert input variable selection.

Table 5 clearly shows that the best results were obtained for the combination of d_K distances (between *FIN*s) with expert-selected input variables. The L1-distance *kNN* results (lines 2, 4, and 6) use average values of populations of measurements, and were reported in previous work (Kaburlasos et al., 2002). In contrast, *FINkNN* is sensitive to the *skewness* of the distribution of measurements due to its use of *FIN*s and the d_K metric. In all but one of the nine possible comparisons in Table 5 (*FINkNN* versus L1-distance *kNN* for each region and for each selected set of input variables) results are improved using *FINkNN*. In general, it appears that an employment of *FIN*s tends to improve classification results. Finally, we also observe that the selection of input variables significantly affects the outcome of classification. Input variables selected by a human expert produced better results than input variables selected computationally through optimization of an objective/fitness function.

	Prediction Method	Larisa	Platy	Seres
1	*FINkNN* (with expert input variable selection)	1.11	2.26	2.74
2	L1-distances *kNN* (with expert input variable selection)	2.05	2.87	3.17
3	*FINkNN* (with GA local search input variable selection)	4.11	3.12	3.81
4	L1-distances *kNN* (with GA local search input variable selection)	3.89	4.61	4.58
5	*FINkNN* (with GA input variable selection)	4.85	3.39	3.69
6	L1-distances *kNN* (with GA input variable selection)	5.59	4.05	3.74
7	"medium" selection	5.22	3.44	5.54
8	Random prediction	8.56	4.27	6.62
9	minimum prediction error	1.11	1.44	1.46

Table 5: Average % prediction error rates using various methods for three factories of Hellenic Sugar Industry (HSI), Greece.

Computation time for algorithm "random prediction" in line 8 of Table 5 was negligible, i.e. the time required to generate a random number in a computer. However more time was required for the algorithms in lines 1 - 6 of Table 5 to select the input variables on a conventional PC using a Pentium (r) II processor. More specifically, algorithm "L1-distances *kNN*" (with GA input variable selection) required computer time of the order 5-10 minutes. In addition, algorithm "L1-distances *kNN*" (with GA local search input variable selection) required less than 5 minutes to select a set of input variables. In the last two cases the corresponding algorithm *FINkNN* required slightly more time due to the computation of distance d_K between *FIN*s. Finally, for either algorithm "*FINkNN*" or "L1-distances *kNN*" (with expert input variable selection) an expert needed around half an hour to select a set of input parameters. As long as input variables had been selected then computation time for all algorithms in lines 1 - 6 of Table 5 was less than 1 second to classify a year to a category "good", "medium" and "bad".

7 Conclusion and Future Research

A nearest neighbor classifier, *FINkNN*, was introduced that applies in the metric product-lattice F^N of fuzzy interval numbers (*FIN*s), which are conventional interval-supported convex fuzzy sets. *FINkNN* effectively predicted annual sugar production based on populations of measurements supplied by the Hellenic Sugar Industry. The algorithm CALFIN was presented for constructing *FIN*s from populations of measurements, and a novel metric distance was presented between fuzzy sets with arbitrary-shaped membership functions.

The improved prediction results presented in this work have been attributed to the capacity of *FIN*s to capture the state of the real world more accurately than single numbers because a *FIN* represents a whole population of samples/measurements. Future work includes an experimental comparison of *FINkNN* with alternative classification methods, e.g. decision trees, etc.

The metric d_K might potentially be useful in a number of applications. For instance, d_K could be used to compute a metric distance between populations of statistical samples. Furthermore, d_K could be useful in Fuzzy Inference System (FIS) design by calculating rigorously the proximity of

two fuzzy sets. Note also that a *FIN* can always be computed for any population size therefore a *FIN* could be useful as an instrument for data normalization and dimensionality reduction.

Acknowledgements

The data used in this work is a courtesy of Hellenic Sugar Industry S.A, Greece. Part of this research has been funded by a grant from the Greek Ministry of Development. The authors acknowledge the suggestions of Maria Konstantinidou for defining metric lattice M^h out of pseudo-metric lattice P^h. We also thank Haym Hirsh for his suggestions regarding the presentation of this work to a machine-learning audience.

Appendix A

This Appendix shows a metric distance in the lattice M^h of generalized intervals of height h. Consider the following definition.

Definition A.1 A *pseudo-metric distance* in a set S is a real function d: $\mathsf{S}\times\mathsf{S}\rightarrow\mathsf{R}$ such that the following four laws are satisfied for $x,y,z\in\mathsf{S}$:

(M1) $d(x,y) \geq 0$, (M3) $d(x,y) = d(y,x)$, and

(M2) $d(x,x) = 0$, (M4) $\mathrm{d}(x,y) \leq d(x,z) + \mathrm{d}(z,y)$ - Triangle Inequality

If, in addition to the above, the following law is satisfied

(M0) $d(x,y) = 0 \Rightarrow x=y$

then real function d is called a *metric distance* in S.

Given a set S equipped with a metric distance d, the pair (S, d) is called *metric space*. If $\mathsf{S}=\mathsf{L}$ is a lattice then metric space (L, d) is called, in particular, *metric lattice*.

A distance can be defined in a lattice L as follows (Birkhoff, 1967). Consider a *valuation* function in L, that is a real function v: $\mathsf{L}\rightarrow\mathsf{R}$ which satisfies $v(x)+v(y)= v(x\vee_{\mathsf{L}}y)+v(x\wedge_{\mathsf{L}}y)$, $x,y\in\mathsf{L}$. A valuation function is called *monotone* if and only if $x\leq_{\mathsf{L}}y$ implies $v(x)\leq v(y)$. If a lattice L is equipped with a *monotone valuation* then real function $d(x,y)=v(x\vee_{\mathsf{L}}y)-v(x\wedge_{\mathsf{L}}y)$, $x,y\in\mathsf{L}$ defines a *pseudo-metric* distance in L. If, furthermore, monotone valuation $v(.)$ satisfies "$x<_{\mathsf{L}}y$ implies $v(x)<v(y)$" then $v(.)$ is called *positive valuation* and function $d(x,y)=v(x\vee_{\mathsf{L}}y)-v(x\wedge_{\mathsf{L}}y)$ is a *metric* distance in L. A positive valuation function is defined in the set M^h of generalized intervals in the following.

Let v: $\mathsf{P}^h\rightarrow\mathsf{R}$ be a real function which maps a generalized interval to its area, that is function v maps a positive generalized interval $[a,b]^h_+$ to non-negative number $h(b-a)$ whereas function v maps a negative generalized interval $[a,b]^h_-$ to non-positive number $-h(b-a)$. It has been shown (Kaburlasos, 2002) that function v is a monotone valuation in P^h, nevertheless v is not a positive valuation. In order to define a *metric* in the set of generalized intervals, an equivalence relation ~ has been introduced in P^h such that $x\sim y \Leftrightarrow d(x,y)=0$, $x,y\in\mathsf{P}^h$. The *quotient* (set) of P^h with respect to equivalence relation ~ is lattice M^h, symbolically $\mathsf{M}^h= \mathsf{P}^h/\sim$. In conclusion M^h is a metric lattice with distance d given by $d(x,y)= v(x\vee_{\mathsf{M}^h}y)-v(x\wedge_{\mathsf{M}^h}y)$.

Appendix B

A one-one correspondence is shown in this Appendix between *FIN*s and *probability density functions* (*pdf*s). More specifically, based on the one-one correspondence between *pdf*s and *Probability Distribution Functions* (*PDF*s), a one-one correspondence is shown between *PDF*s and *FIN*s as follows. In the one direction, a *PDF* $G(x)$ was mapped to a *FIN* F with membership function $\mu_F(.)$ such that: if $G(x_0)= 0.5$ then $\mu_F(x)= 2G(x)$ for $x \leq x_0$, whereas $\mu_F(x)=2[1-G(x)]$ for $x \geq x_0$. In the other direction, a *FIN* F was mapped to a *PDF* $G(x)$ such that: if $\mu_F(x_0)= 1$ then $G(x)= \frac{1}{2}\mu_F(x)$ for $x \leq x_0$, whereas $G(x)= 1-\frac{1}{2}\mu_F(x)$ for $x \geq x_0$. Recall, from the remarks following algorithm CALFIN, that $\mu_F(x_0)= 1$ at exactly one point x_0.

A statistical interpretation of a *FIN* is presented in the following. Algorithm CALFIN implies that when a *FIN* F is constructed then approximately $100(1-h)$ % of the population of samples are included in interval $support(F(h))$. Hence, if a large number of samples is drawn independently from one probability distribution then interval $support(F(h))$ could be regarded as "an interval of confidence at level-h".

The previous analysis may also imply that *FIN*s could be considered as vehicles for accommodating synergistically tools from, on the one hand, probability-theory/statistics and, on the other hand, fuzzy set theory. For instance two *FIN*s F_1 and F_2 calculated from two *pdf*s $f_1(x)$ and $f_2(x)$, respectively, could be used for calculating a metric distance (d_K) between *pdf*s $f_1(x)$ and $f_2(x)$ as follows: $d_K(f_1(x), f_2(x))=d_K(F_1, F_2)$. Moreover two *FIN*s F_1 and F_2 calculated from two populations of measurements could be used for computing a (metric) distance between populations of measurements.

References

D.W. Aha, D.F. Kibler, and M.K. Albert. Instance-Based Learning Algorithms. *Machine Learning*, 6:37-66, 1991.

A. Ben-Hur, D. Horn, H.T. Siegelmann, and V. Vapnik. Support Vector Clustering. *Journal of Machine Learning Research*, 2(Dec):125-137, 2001.

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

G. Bontempi, M. Birattari, and H. Bersini. Lazy Learning: A Logical Method for Supervised Learning. In *New Learning Paradigms in Soft Computing*, L.C. Jain and J. Kacprzyk (editors), 84: 97-136, Physica-Verlag, Heidelberg, Germany, 2002.

O. Boz. Feature Subset Selection by Using Sorted Feature Relevance. In *Proceedings of the Intl. Conf. On Machine Learning and Applications* (*ICMLA'02*), Las Vegas, NV, USA, 2002.

P.-T. Chang, and E.S. Lee. Fuzzy Linear Regression with Spreads Unrestricted in Sign. *Computers Math. Applic.*, 28(4):61-70, 1994.

V. Chatzis, and I. Pitas. Mean and median of fuzzy numbers. In *Proc. IEEE Workshop Nonlinear Signal Image Processing*, pages 297-300, Neos Marmaras, Greece, 1995.

C. Citterio, A. Pelagotti, V. Piuri, and L. Rocca. Function Approximation – A Fast-Convergence Neural Approach Based on Spectral Analysis. *IEEE Transactions on Neural Networks*, 10(4):725-740, 1999.

W. Cohen. Hardness Results for Learning First-Order Representations and Programming by Demonstration. *Machine Learning*, 30:57-88, 1998.

B.V. Dasarathy, editor. *Nearest Neighbor* (*Nn Norms: Nn Pattern Classification Techniques*), IEEE Computer Society Press, 1991.

P. Diamond. Fuzzy Least Squares. *Inf. Sci*., 46:141-157, 1988.

P. Diamond, and P. Kloeden. *Metric Spaces of Fuzzy Sets*. World Scientific, Singapore, 1994.

P. Diamond, and R. Körner. Extended Fuzzy Linear Models and Least Squares Estimates. *Computers Math. Applic.*, 33(9):15-32, 1997.

D. Dubois, and H. Prade. *Fuzzy Sets and Systems - Theory and Applications*. Academic Press, Inc., San Diego, CA, 1980.

R.O. Duda, P.E. Hart., and D.G. Stork. *Pattern Classification*, 2nd edition. John Wiley & Sons, New York, N.Y., 2001.

P. Frasconi, M. Gori, and A. Sperduti. A General Framework for Adaptive Processing of Data Structures. *IEEE Transactions on Neural Networks*, 9(5):768-786, 1998.

L. Goldfarb. What is a Distance and why do we Need the Metric Model for Pattern Learning. *Pattern Recognition*, 25(4):431-438, 1992.

X. Hong, and C.J. Harris. Variable Selection Algorithm for the Construction of MIMO Operating Point Dependent Neurofuzzy Networks. *IEEE Trans. on Fuzzy Systems*, 9(1):88-101, 2001.

E.G. Hutchinson, and J.M. Thornton. PROMOTIF – A program to identify and analyze structural motifs in proteins. *Protein Science*, 5(2):212-220, 1996.

H. Ishibuchi, and T. Nakashima. Effect of Rule Weights in Fuzzy Rule-Based Classification Systems. *IEEE Transactions on Fuzzy Systems*, 9(4):506 -515, 2001.

V.G. Kaburlasos. Novel Fuzzy System Modeling for Automatic Control Applications. In *Proceedings 4th Intl. Conference on Technology & Automation*, pages 268-275, Thessaloniki, Greece, 2002.

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*, 4(2):31-43, 1997.

V.G. Kaburlasos, and V. Petridis. Fuzzy Lattice Neurocomputing (FLN) Models. *Neural Networks*, 13(10):1145-1170, 2000.

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), 84:55-96, Physica-Verlag, Heidelberg, Germany, 2002.

V.G. Kaburlasos, V. Petridis, P.N. Brett, and D.A. Baker. Estimation of the Stapes-Bone Thickness in Stapedotomy Surgical Procedure Using a Machine-Learning Technique. *IEEE Transactions on Information Technology in Biomedicine*, 3(4):268-277, 1999.

V.G. Kaburlasos, V. Spais, V. Petridis, L, Petrou, S. Kazarlis, N. Maslaris, and A. Kallinakis. Intelligent Clustering Techniques for Prediction of Sugar Production. *Mathematics and Computers in Simulation*, 60(3-5): 159-168, 2002.

S. Kasif, S. Salzberg, D.L. Waltz, J. Rachlin, and D.W. Aha. A Probabilistic Framework for Memory-Based Reasoning. *Artificial Intelligence*, 104(1-2):287-311, 1998.

S.A. Kazarlis, S.E. Papadakis, J.B. Theocharis, and V. Petridis, "Microgenetic algorithms as generalized hill-climbing operators for GA optimization", *IEEE Trans. on Evolutionary Computation*, 5(3): 204-217, 2001.

M.J. Kearns, and U.V Vazirani. *An Introduction to Computational Learning Theory*. The MIT Press, Cambridge, Massachusetts, 1994.

G.J. Klir, and T.A. Folger. *Fuzzy Sets, Uncertainty, and Information*. Prentice-Hall, Englewood Cliffs, New Jersey, 1988.

D. Koller, and M. Sahami. Toward Optimal Feature Selction. In *ICML-96*: *Proceedings of the 13th Intl. Conference on Machine Learning*, pages 284-292, San Francisco, CA, 1996.

J. Kolodner. *Case-Based Reasoning*. Morgan Kaufmann Publishers, San Mateo, CA, 1993.

O.L. Mangasarian and D.R. Musicant. Lagrangian Support Vector Machines. *Journal of Machine Learning Research*, 1:161-177, 2001.

T.M. Mitchell. *Machine Learning*. The McGraw-Hill Companies, Inc., New York, NY, 1997.

S. Muggleton. Inductive Logic Programming. *New Generation Computing*, 8(4):295-318, 1991.

A. Paccanaro, and G.E. Hinton. Learning Distributed Representations of Concepts Using Linear Relational Embedding. *IEEE Transactions on Knowledge and Data Engineering*, 13(2):232-244, 2001.

V. Petridis, and V.G. Kaburlasos. Fuzzy Lattice Neural Network (FLNN): A Hybrid Model for Learning. *IEEE Transactions on Neural Networks*, 9(5):877-890, 1998.

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*, 8(2):236, 2000).

V. Petridis, and V.G. Kaburlasos. Clustering and Classification in Structured Data Domains Using Fuzzy Lattice Neurocomputing (FLN). *IEEE Transactions on Knowledge and Data Engineering*, 13(2):245-260, 2001.

V. Petridis, A. Kehagias, L. Petrou, H. Panagiotou, and N. Maslaris. Predictive Modular Neural Network Methods for Prediction of Sugar Beet Crop Yield. In *Proceedings IFAC-CAEA'98 Conference on Control Applications and Ergonomics in Agriculture*, Athens, Greece, 1998.

B. Schölkopf, S. Mika, C.J.C. Burges, P. Knirsch, K.-R. Muller, G. Rätsch, and A.J. Smola. Input Space Versus Feature Space in Kernel-Based Methods. *IEEE Transactions on Neural Networks*, 10(5):1000-1017, 1999.

G. Stoikos. Sugar Beet Crop Yield Prediction Using Artificial Neural Networks (in Greek). In *Proceedings of the Modern Technologies Conference in Automatic Control*, pages 120-122, Athens, Greece, 1995.

H. Tanaka, and H. Lee. Interval Regression Analysis by Quadratic Programming Approach. *IEEE Trans. Fuzzy Systems*, 6(4):473-481, 1998.

M. Turcotte, S.H. Muggleton, and M.J.E. Sternberg. Learning rules which relate local structure to specific protein taxonomic classes. In *Proceedings of the 16th Machine Intelligent Workshop*, York, U.K., 1998.

V. Vapnik. The support vector method of function estimation. In *Nonlinear Modeling: Advanced Black-Box Techniques*, J. Suykens, and J. Vandewalle (editors), 55-86, Kluwer Academic Publishers, Boston, MA, 1988.

V. Vapnik. An Overview of Statistical Learning Theory. *IEEE Transactions on Neural Networks*, 10(5):988-999, 1999.

V. Vapnik, and C. Cortes. Support vector networks**.** *Machine Learning*, 20:1-25, 1995.

M. Vidyasagar. *A Theory of Learning and Generalization: With Applications to Neural Networks and Control Systems* (*Communications and Control Engineering*). Springer Verlag, New York, NY, 1997.

P. Winston, *Learning Structural Descriptions from Examples. The Psychology of Computer Vision*. P. Winston (ed.), 1975.

I.H. Witten, and E. Frank. *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations*. Morgan Kaufmann, 2000.

M.-S. Yang, and C.-H. Ko. On Cluster-Wise Fuzzy Regression Analysis. *IEEE Trans. Systems, Man, Cybernetics - Part B: Cybernetics*, 27(1):1-13, 1997.

L.A. Zadeh. Fuzzy Sets. *Information and Control*, 8:338-353, 1965.

H.-J. Zimmermann. *Fuzzy Set Theory - and Its Applications*. Kluwer Academic Publishers, Norwell, MA, 1991.