

Induction of formal concepts by lattice computing techniques for tunable classification

Vassilis G. Kaburlasos and Lefteris Moussiades

*Dept. of Computer and Informatics Engineering, TEI of Eastern Macedonia & Thrace,
Kavala, 65404 Greece*

Abstract

This work proposes an enhancement of Formal Concept Analysis (FCA) by Lattice Computing (LC) techniques. More specifically, a novel Galois connection is introduced toward defining tunable metric distances as well as tunable inclusion measure functions between formal concepts induced from *hybrid* (i.e., nominal and numerical) data. An induction of formal concepts is pursued here by a novel extension of the Karnaugh map, or K-map for short, technique from digital electronics. In conclusion, granular classification can be pursued. The capacity of a classifier based on formal concepts is demonstrated here with promising results. The formal concepts are interpreted as descriptive decision-making knowledge (rules) induced from the training data.

Keywords: Classification, Formal Concept Analysis, Lattice Computing

1. Introduction

With the proliferation of computer-supported devices there is a sustained interest in human/machine-to-machine interaction applications involving numerical and/or nominal data. In the aforementioned context, Machine Learning (ML) and Computational Intelligence (CI) techniques are promising. In particular, Formal Concept Analysis (FCA) supported by Lattice Computing (LC) techniques might have a significant potential as demonstrated in this work.

Formal Concept Analysis, or FCA for short, is a branch of applied (mathematical) lattice theory [4] which is used in data-mining and knowledge-representation applications. Typically, FCA induces lattice-ordered structures, namely *formal concepts*, restricted in finite power-sets. Nevertheless, traditional FCA lacks a capacity for (tunable) generalization. This work equips an extended FCA with LC tools for tunable generalization, also in spaces of (un)countably infinite cardinality involving hybrid (i.e., nominal and numerical) data; linguistic data processing is another possibility as explained below.

The LC paradigm has been defined as “an evolving collection of tools and mathematical modelling methodologies with the capacity to process lattice-ordered data *per se* including logic values, numbers, sets, symbols, graphs, etc” [13], [22]. The LC paradigm suggests an alternative to conventional “number crunching” in computational intelligence applications by considering semantics, represented by a lattice-order relation [7]. Specific examples of the LC paradigm are described in [9], [20]. In a broad sense, the LC paradigm subsumes any lattice-theory-based information processing approach. In the aforementioned sense, a number of cross-fertilizations

have been proposed in the LC paradigm by different authors [1], [17]. Trends in LC appear in [5], [8].

This work pursues an enhancement of the traditional FCA using LC techniques. In particular, this work is an extension of recent work regarding (optimized) granular kNN classifiers [6], [21] based on *Type-1 Intervals' Numbers*, or (*Type-1*) *INs* for short. Recall that an IN has been described as a mathematical object that may represent either a fuzzy interval or a distribution of numeric data samples [14]. The space F_1 of INs has been studied. In particular, it has been shown that F_1 is a metric lattice with cardinality \aleph_1 , where “ \aleph_1 ” is the cardinality of the set \mathbb{R} of real numbers; the space F_1 is a cone in a linear space; non-linear transformations in F_1 have been shown [14]. This work considers INs jointly with symbols in data mining and pattern recognition applications. In conclusion, tunable, hybrid formal concepts are induced from both numerical and nominal data as explained below.

The novelties of this work are summarized next. First, a measure (set) function is introduced in a power-set of (un)countable cardinality toward employing LC instruments in traditional FCA. Second, an extension of the K-map technique from digital electronics is introduced toward an optimal calculation of formal concepts. Third, hybrid formal concepts are considered including both nominal and numerical data (IN) representations; in particular, a (Type-1) IN here represents a population of numerical data.

The layout of this work is as follows. Section 2 summarizes the mathematical background. Section 3 presents FCA extensions based on LC techniques. Section 4 describes a K-map extension. Section 5 presents computational experiments. Section 6 concludes by summarizing our contribution and discussing future work. The Appendix includes useful mathematical definitions.

2. Mathematical Background

This section presents general notions from lattice theory followed by the presentation of two lattices, namely a power-set and the set of INs.

2.1 General

A binary relation Ξ on a set P is a *partial order* iff it satisfies the following three conditions:

(P1) $x \Xi x$ — *Reflexivity*.

(P2) $x \Xi y$ and $y \Xi x \Rightarrow x = y$ — *Antisymmetry*.

(P3) $x \Xi y$ and $y \Xi z \Rightarrow x \Xi z$ — *Transitivity*.

In this case (P, Ξ) is called a *partially ordered set*, or *poset* for short.

A *lattice* is a poset (L, Ξ) with the additional property that any two elements $x, y \in L$ have both a greatest lower bound, namely *infimum* (or, equivalently, *meet*), denoted by $x \sqcap y$ and a least upper bound, namely *supremum* (or, equivalently, *join*), denoted by $x \sqcup y$. If in a lattice (L, Ξ) every (x, y) pair satisfies either $x \Xi y$ or $x \supseteq y$ then we say that the lattice (L, Ξ) is *totally-ordered*. A lattice (L, Ξ) is called *complete* iff each of its subsets X has both an infimum and a supremum in L (hence, taking $X = L$, we see that a complete lattice has both a *least* element and a *greatest* element denoted by O and I , respectively).

A *valuation* on a lattice (L, Ξ) is a real function $v: L \rightarrow \mathbb{R}$ which satisfies $v(x) + v(y) = v(x \sqcap y) + v(x \sqcup y)$. A valuation is called *positive* iff $x \sqsubset y \Rightarrow v(x) < v(y)$.

Given a positive valuation on a lattice (L, Ξ) , a *metric distance* function $d: L \times L \rightarrow \mathbb{R}_0^+$ is given by $d(x, y) = v(x \sqcup y) - v(x \sqcap y)$ — The definition for a metric is given in the Appendix. Furthermore, a positive valuation results in an *inclusion measure* function $\sigma: L \times L \rightarrow [0, 1]$ given by $\sigma_{\sqcup}(x, y) = \frac{v(y)}{v(x \sqcup y)}$ — The definition for an

inclusion measure is given in the Appendix.

Our interest here focuses on a complete lattice (L, Ξ) .

Of special interest is a *dual isomorphic* function $\theta: L \rightarrow L$ which, by definition, satisfies both “ $x \sqsubset y \Leftrightarrow \theta(x) \supseteq \theta(y)$ ” and “ θ is onto L ”. Note that function $\theta(\cdot)$ may be interpreted as a negation operator in lattice (L, Ξ) . Function $\theta(\cdot)$ may extend, as explained below, the metric given by $d(x, y) = v(x \sqcup y) - v(x \sqcap y)$ to the set I_1 of a lattice (L, Ξ) (*Type-1*) intervals defined by

$$I_1 = \{[a_1, a_2] \mid a_1, a_2 \in L \text{ such that } a_1 \Xi a_2\} \cup \{\emptyset\}$$

The empty set (\emptyset) is also considered to be an interval $[a_1, a_2]$ with any $a_1, a_2 \in L$ such that $a_1 \not\Xi a_2$. In particular, the empty interval in a complete lattice (L, Ξ) is typically represented by $[I, O]$. In turn, (I_1, \subseteq) is a complete lattice, where \subseteq is the conventional set-inclusion relation with order $[a, b] \subseteq [c, d] \Leftrightarrow (c \Xi a \text{ and } b \Xi d)$ [24] and *least* (resp. *greatest*) element denoted by $[I, O]$ (resp. $[O, I] = L$).

It is known that given both a positive valuation $v: L \rightarrow \mathbb{R}$ and a dual isomorphic function $\theta: L \rightarrow L$ on a lattice (L, Ξ) , the function $v_{\Delta}: L \times L \rightarrow \mathbb{R}$ given by $v_{\Delta}([a, b]) = v(\theta(a)) + v(b)$ is a positive valuation on lattice $(L \times L, \Xi \times \Xi)$ [10]. Hence, the function $v_{\Delta}(\cdot)$ can be used to define a metric distance on the lattice (I_1, \subseteq) of (*Type-1*) intervals. Furthermore, function $v_{\Delta}(\cdot)$ can be used to define an inclusion measure in (I_1, \subseteq) . This work focuses on metrics. Future work will consider inclusion measures.

2.2 A power-set

Based on the definitions of a σ -algebra as well as of a *measure* given in the Appendix, a *measure space* is defined as a triplet $(\Omega, \Sigma_{\Omega}, m_{\Sigma_{\Omega}})$, where Ω is a set, Σ_{Ω} is a σ -algebra over Ω , and $m_{\Sigma_{\Omega}}$ is a measure over Σ_{Ω} . Note that a *probability space* is a measure space such that $m_{\Sigma_{\Omega}}(\Omega) = 1$.

Given a measure space $(\Omega, \Sigma_{\Omega}, m_{\Sigma_{\Omega}})$ it follows that $(\Sigma_{\Omega}, \subseteq)$ is a complete lattice ordered by set-inclusion (\subseteq). The corresponding lattice meet and join operators are the conventional set intersection (\cap) and union (\cup) operators, respectively. In particular, note that if $A, B \in \Sigma_{\Omega}$ then (1) $(A \cup B) \in \Sigma_{\Omega}$ and (2) $A^c, B^c \in \Sigma_{\Omega} \Rightarrow (A^c \cup B^c)^c = (A \cap B) \in \Sigma_{\Omega}$. Furthermore, note that the measure $m_{\Sigma_{\Omega}}$ is a positive valuation function on lattice $(\Sigma_{\Omega}, \subseteq)$. A dual isomorphic function $\theta: \Sigma_{\Omega} \rightarrow \Sigma_{\Omega}$ is given by the set complement $\theta(A) = A^c = (\Omega - A) \in \Sigma_{\Omega}$. Our interest in the context of this work focuses on a measure space $(\Omega, \Sigma_{\Omega}, m_{\Sigma_{\Omega}})$ such that $\Omega = \{1, \dots, K\}$ is a finite set, $\Sigma_{\Omega} = 2^{\Omega}$, and $m_{\Sigma_{\Omega}}$ remains to be defined in a specific application.

2.3 Intervals' Numbers (INs)

We present a hierarchy of lattices stemming from the totally-ordered set \mathbb{R} of real numbers.

Real numbers

Consider the set $\overline{\mathbb{R}} = \mathbb{R} \cup \{-\infty, \infty\}$, namely set of *extended real numbers*. It turns out that $(\overline{\mathbb{R}}, \leq)$ is a complete- as well as a totally-ordered lattice, where \leq is the usual order relation of real numbers. The

corresponding infimum and supremum operators are denoted by \wedge and \vee , respectively. There are both a least element denoted by “ $-\infty$ ” and a greatest element denoted by “ ∞ ” in lattice $(\overline{\mathbb{R}}, \leq)$.

Intervals

Given $a_1, a_2 \in \overline{\mathbb{R}}$ such that $a_1 \leq a_2$, an *interval* $A = [a_1, a_2]$ is defined as

$$[a_1, a_2] = \{x: x \in \overline{\mathbb{R}} \text{ and } a_1 \leq x \leq a_2\}$$

The empty set (\emptyset) is also considered an interval, i.e. the *empty interval*; we denote the collection of intervals of $\overline{\mathbb{R}}$ (including the empty interval) by $I_1(\overline{\mathbb{R}})$, or I_1 for short, namely set of (Type-1) intervals.

The pair (I_1, \subseteq) is a complete lattice with respect to the \subseteq (set theoretic inclusion) order. The least element (\emptyset) of I_1 is typically represented as $[\infty, -\infty]$, whereas the greatest element of I_1 is represented as $[-\infty, \infty] = \overline{\mathbb{R}}$. Given nonempty intervals $A = [a_1, a_2]$ and $B = [b_1, b_2]$, their infimum and supremum in I_1 are given, respectively, by

$$A \cap B = [a_1 \vee b_1, a_2 \wedge b_2] \quad (1)$$

$$A \cup B = [a_1 \wedge b_1, a_2 \vee b_2] \quad (2)$$

Fuzzy Intervals

A *fuzzy subset* F of $\overline{\mathbb{R}}$ is identical to its *membership function* $F: \overline{\mathbb{R}} \rightarrow [0, 1]$. A partial order can be defined for fuzzy subsets as follows

$$F \leq G \Leftrightarrow (\forall x: F(x) \leq G(x)).$$

It is easy to check that the infimum and supremum of two fuzzy sets F, G is a fuzzy set denoted by $F \wedge G$ and $F \vee G$, respectively, and defined for every $x \in \overline{\mathbb{R}}$ as

$$(F \wedge G)(x) = F(x) \wedge G(x) \quad (3)$$

$$(F \vee G)(x) = F(x) \vee G(x) \quad (4)$$

Given a fuzzy subset F , the h-cut of F is the set

$$F_h = \{x: F(x) \geq h\}.$$

Recall that we use the term “h-cut” instead of the (equivalent) term “ α -cut” used in the literature for fuzzy sets due to the two different interpretations for an *Intervals’ Number* (IN) [9].

It is known that for a fuzzy membership function F it is

$$(\forall h \in [0, 1]: F_h = G_h) \Leftrightarrow (\forall x: F(x) = G(x)) \quad (5)$$

Consider the following definition for a *fuzzy interval* or, equivalently, *fuzzy number*.

Definition 1: A *fuzzy interval* is a fuzzy subset F whose every h-cut is a closed interval: $(\forall h \in [0, 1]: F_h \in I_1)$. We denote the set of all fuzzy intervals by F_1 .

It is known that the structure (F_1, \leq) , equipped with the fuzzy sets order \leq , is a complete lattice. The infimum

operation is \wedge as defined in (3). The supremum operation ($\dot{\vee}$) is defined as follows

$$F \dot{\vee} G = \inf\{H: H \in F_1, F \leq H, G \leq H\} \quad (6)$$

That is, $F \dot{\vee} G$ is the “smallest” fuzzy interval which is greater than both F and G .

Intervals’ Numbers

Consider the following definition.

Definition 2: A *Type-1 Intervals’ Number*, or (*Type-1*) *IN* for short, is a function $F: [0, 1] \rightarrow I_1$, which satisfies

$$h_1 \geq h_2 \Rightarrow F_{h_1} \subseteq F_{h_2}, \text{ and}$$

$$\forall X \subseteq [0, 1]: \bigcap_{h \in X} F_h = F_{\bigvee X}.$$

We denote the class of all (Type-1) INs by F_1 . The following theorem shows that we can associate every IN to a fuzzy interval.

Theorem 1: Given IN $E \in F_1$, define a fuzzy set \tilde{E} as

$$\forall x: \tilde{E}(x) = \sup\{h: x \in E_h\}.$$

The h-cuts of \tilde{E} are denoted by \tilde{E}_h and, by definition, satisfy $\forall h \in [0, 1]: \tilde{E}_h = \{x: \tilde{E}(x) \geq h\}$. Then, for all $h \in [0, 1]$ we have $\tilde{E}_h = E_h$. Hence, \tilde{E} is a *fuzzy interval*.

In other words, the h-cuts of the fuzzy set \tilde{E} are the intervals of the original IN E . Hence, Theorem 1 indicates two equivalent representations for an IN, namely the *interval-representation* and the *membership-function-representation*. An advantage of the interval-representation is that it enables useful algebraic operations, whereas an advantage of the membership-function-representation is that it enables convenient (e.g., fuzzy logic, etc) interpretations. INs are equipped with a partial order \preceq as follows.

Definition 3: For every pair $F, G \in F_1$ we define the relationship \preceq as follows

$$F \preceq G \Leftrightarrow (\forall h \in [0, 1]: F_h \subseteq G_h).$$

Lattices (F_1, \leq) and (F_1, \preceq) are (*order*) *isomorphic* as a consequence of the following theorem — The definition for (order) isomorphic posets is given in the Appendix.

Theorem 2: For all $F, G \in F_1$ we have

$$F \preceq G \Leftrightarrow \forall h \in [0, 1]: F_h \subseteq G_h \Leftrightarrow \forall x \in \overline{\mathbb{R}}: F(x) \leq G(x)$$

The relation \preceq is a lattice order and the lattice (F_1, \preceq) of INs is complete. If we denote the infimum operation by \wedge and the supremum operation by $\dot{\vee}$, then $\forall h \in [0, 1]$ we have

$$(F \wedge G)_h = F_h \cap G_h \text{ and } (F \dot{\vee} G)_h = F_h \cup G_h \quad (7)$$

Metric Distances Between INs

The following two functions will be useful.

A1. A strictly increasing function $\nu: \overline{\mathbb{R}} \rightarrow \overline{\mathbb{R}}$.

A2. A strictly decreasing function $\theta: \overline{\mathbb{R}} \rightarrow \overline{\mathbb{R}}$.

Based on two functions ν and θ that satisfy **A1-A2**, a metric function $d_{I_1}: I_1 \times I_1 \rightarrow \mathbb{R}_0^+$ is defined as

$$d_{I_1}([a,b],[c,d]) = \nu(\theta(a \wedge c)) - \nu(\theta(a \vee c)) + \nu(b \vee d) - \nu(b \wedge d) \quad (8)$$

A metric function $d_{F_1}: F_1 \times F_1 \rightarrow \mathbb{R}_0^+$ is defined as

$$d_{F_1}(F, G) = \int_0^1 d_{I_1}(F_h, G_h) dh \quad (9)$$

Both functions $\nu(\cdot)$ and $\theta(\cdot)$ can be parametric, therefore all the above metrics are tunable.

Assume N metric spaces (S_i, d_i) , $i \in \{1, \dots, N\}$, where S_i is a space and function $d_i: S_i \times S_i \rightarrow \mathbb{R}_0^+$ is a metric.

Then, the k -Minkowski metric $d^k: S \times S \rightarrow \mathbb{R}_0^+$, where $S = S_1 \times \dots \times S_N$, is defined as

$$d^k(\mathbf{a}, \mathbf{b}) = \left[\sum_{i=1}^N (d_i(a_i, b_i))^k \right]^{\frac{1}{k}} \quad (10)$$

where $\mathbf{a} = (a_1, \dots, a_N)$, $\mathbf{b} = (b_1, \dots, b_N)$, $k \in \{1, 2, \dots, \infty\}$.

Apparently, k -Minkowski metrics can be used between N -tuples of INs.

3. FCA Extensions Based on LC Techniques

Formal Concept Analysis (FCA) deals with a formal context defined next [3].

Definition 4: A formal context is a triplet (G, M, I) , where G and M are two finite sets called *objects* and *attributes*, respectively; moreover, $I \subseteq G \times M$ is a binary relation, namely *incidence relation*.

Consider two derivation operators as follows.

Definition 5: Given a formal context (G, M, I) as well as the subsets $A \subseteq G$ and $B \subseteq M$, two derivation operators, namely *Galois connectors*, are defined as

$$\begin{aligned} A' &:= \{m \in M \mid (g, m) \in I, \forall g \in A\} \\ B' &:= \{g \in G \mid (g, m) \in I, \forall m \in B\} \end{aligned}$$

The aforementioned derivation operators satisfy the following conditions for $A_1, A_2 \subseteq G$ and $B_1, B_2 \subseteq M$:

1. $A_1 \subseteq A_2 \Rightarrow A_2' \subseteq A_1'$; dually, $B_1 \subseteq B_2 \Rightarrow B_2' \subseteq B_1'$.
2. $A_1 \subseteq A_1''$, $A_1' = A_1'''$; dually, $B_1 \subseteq B_1''$, $B_1' = B_1'''$.
3. $A_1 \subseteq B_1' \Leftrightarrow B_1 \subseteq A_1' \Leftrightarrow A_1 \times B_1 \subseteq I$.

A formal concept is defined next.

Definition 6: Let (G, M, I) be a formal context. A *formal concept* C is a pair $C = (A, B)$, where $A \subseteq G$ and $B \subseteq M$ such that $A = B'$ and $A' = B$.

We remark that A is called *extent* and B is called *intent* of the formal concept (A, B) .

Next, we define formally a Galois connection [4].

Definition 7: Let $\varphi: P \rightarrow Q$ and $\psi: Q \rightarrow P$ be maps between two posets (P, \sqsubseteq) and (Q, \sqsubseteq) . Such pair of maps is called a *Galois connection* between the posets if

1. $p_1 \sqsubseteq p_2 \Rightarrow \varphi(p_1) \sqsupseteq \varphi(p_2)$.
2. $q_1 \sqsubseteq q_2 \Rightarrow \psi(q_1) \sqsupseteq \psi(q_2)$.
3. $p \sqsubseteq \psi(\varphi(p))$ and $q \sqsubseteq \varphi(\psi(q))$.

The following Theorem [4] will be instrumental below.

Theorem 3: A pair (φ, ψ) of maps is a Galois connection iff $p \sqsubseteq \psi(q) \Leftrightarrow \varphi(p) \sqsupseteq q$.

FCA focuses on a Galois connection $\varphi: 2^M \rightarrow 2^N$ and $\psi: 2^N \rightarrow 2^M$ between power-sets lattices $(2^M, \subseteq)$ and $(2^N, \subseteq)$ of two (finite) sets M and N , respectively.

If (A_1, B_1) and (A_2, B_2) are concepts of a context, we define an order $(A_1, B_1) \sqsubseteq (A_2, B_2)$ iff $A_1 \subseteq A_2$ (which is equivalent to $B_2 \subseteq B_1$). The set of all concepts of a formal context (G, M, I) ordered in this way is denoted by $\underline{\mathcal{B}}(G, M, I)$ and it is called the *concept lattice* of the context (G, M, I) .

Theorem 4 (The Basic Theorem of Concept Lattices): The concept lattice $\underline{\mathcal{B}}(G, M, I)$ is a complete lattice in which the meet and join are given, respectively, by

$$\bigwedge_{t \in T} (A_t, B_t) = \left(\bigcap_{t \in T} A_t, \left(\bigcup_{t \in T} B_t \right)'' \right)$$

and

$$\bigvee_{t \in T} (A_t, B_t) = \left(\left(\bigcup_{t \in T} A_t \right)', \bigcap_{t \in T} B_t \right)$$

Various techniques have been proposed for computing formal concepts in FCA [3], [16]. This work proposes a novel computation of a novel formal concept as explained below.

3.1 A Novel Formal Concept Definition

This work proposes defining a formal concept as an interval $[a, b]$ in a complete lattice (L, \sqsubseteq) data domain. Recall from subsection 2.1 that the interval $[a, b]$ is an element of the complete lattice (I_1, \subseteq) of intervals.

We assume both a dual isomorphic function $\theta: L \rightarrow L$ and a positive valuation function $v: L \rightarrow \mathbb{R}$ toward defining a metric as well as an inclusion measure function in lattice (L, Ξ) as explained in subsection 2.1. An additional constraint, namely $\theta(\theta(x))=x$, results in the following Theorem.

Theorem 5: Let function $\theta: L \rightarrow L$ on a lattice (L, Ξ) satisfy both $x \sqsubseteq y \Leftrightarrow \theta(x) \sqsupseteq \theta(y)$ and $\theta(\theta(x))=x$. Then, the pair (θ, θ) is a Galois connection — For simplicity we say that “ θ is a Galois connection”.

Proof: $p = \theta(\theta(p)) \sqsubseteq \theta(q) \Leftrightarrow \theta(p) \sqsupseteq \theta(\theta(q)) = q$.

Due to Theorem 3, θ is a Galois connection. ■

We remark that a Galois connection is a critical instrument for representing a formal concept both in the context of FCA as well as in the context of this work. In particular, recall that a formal concept in FCA is represented by (A, B) such that $B = A' = \varphi(A)$ and $A = B' = \psi(B)$, where (φ, ψ) is a Galois connection; whereas, a formal concept (i.e., a lattice interval $[a, b]$) in this work is represented by $(\theta(a), b)$, where (θ, θ) is a Galois connection.

In either case, a formal concept is interpreted in this work as an *information granule*. Recall that the term (information) granule has been introduced in fuzzy set theory to denote a clump of values drawn together by indistinguishability, similarity, proximity or functionality. Computing with granules, namely *Granular Computing*, is important because granules may accommodate ambiguity and/or uncertainty [19].

Even though both aforementioned formal concepts are induced from data there is a fundamental difference between them. More specifically, a formal concept in FCA “overfits the data” in the sense that it exclusively accommodates the data it was induced from; whereas, a (lattice interval) formal concept might include more data. The latter is called *Type I generalization* [12]. In addition, only a (lattice interval) formal concept provides the instruments, namely a metric as well as an inclusion measure function, to relate a datum outside all formal concepts to its nearest /most-similar one; the latter is called *Type II generalization* [12].

3.2 Hybrid Formal Concepts

An FCA “object” (see in Definition 4) in standard engineering terms here is also called *instance* or, equivalently, *datum*; likewise, an FCA “attribute” (see in Definition 4) here is also called *dimension*. Each data dimension is a complete lattice.

We are interested in datasets including instances with a fixed number of dimensions per instance (per dataset). Certain dimensions include nominal data, whereas the remaining dimensions in an instance include numerical data. In this sense we are interested in *hybrid* datasets (including both nominal and numerical data). In

conclusion, likewise hybrid formal concepts will be induced from the hybrid data as explained below.

A nominal data dimension X_i , $i \in \{1, \dots, L_a\}$ obtains a number K_i of values in a set $V_i = \{i_1, \dots, i_{K_i}\}$, whereas a numerical data dimension Y_i , $i \in \{1, \dots, R\}$ is treated as a real variable, i.e. $Y_i \in \mathbb{R}$. In all, there are L_a nominal data dimensions, namely $X_i \in \{1, \dots, L_a + R_a\}$ where $i \in \{1, \dots, L_a\}$, and R_a numerical data dimensions, namely $Y_i \in \{1, \dots, L_a + R_a\}$ where $i \in \{1, \dots, R_a\}$, such that $X_1 < \dots < X_{L_a}$ and $Y_1 < \dots < Y_{R_a}$. A datum (instance) is called *trivial* iff each of its $L_a + R_a$ dimensions is either a single nominal value or a single number. Likewise, a formal concept might be trivial. Furthermore, we say that the nominal data of an instance (datum) D are *included* in a formal concept FC iff (in every nominal data dimension) the nominal datum of D is included in the nominal data of FC . The next section describes how a hybrid formal concept can be induced, first, by inducing sets of nominal data in every nominal data dimension and, second, by inducing one IN in every numerical data dimension by algorithm CALCIN [21] presented next.

Algorithm CALCIN

Consider a finite set X of (numeric) data samples, i.e., $X = \{x_1, \dots, x_n\}$. Two entries x_i, x_j of the set X are called *successive* iff there is no other entry x_k , $k \in \{1, \dots, n\}$ such that $x_i \wedge x_j < x_k < x_i \vee x_j$, where \wedge and \vee are the min and max operators, respectively. A strictly increasing, cumulative real function $c: \mathbb{R} \rightarrow \mathbb{R}_0^+$ is computed from the set X by, first, defining

$$c(x_i) = \frac{1}{n} |\{x_j : x_j \leq x_i\}|, \quad (11)$$

where $i, j \in \{1, \dots, n\}$ and $|S|$ denotes the cardinality of the set S . Finally, function $c: \mathbb{R} \rightarrow \mathbb{R}_0^+$ is defined by straight-line connecting two points $(x_i, c(x_i))$ and $(x_j, c(x_j))$, where x_i, x_j are successive entries of the set X . Note that there is a unique real number $x_{0.5}$ such that $c(x_{0.5})=0.5$. An IN is calculated from function $c(\cdot)$ such that for values less-than or equal-to $x_{0.5}$ the corresponding IN envelope function equals $2c(x)$, whereas for values larger than $x_{0.5}$ the corresponding IN envelope function equals $2(1-c(x))$.

In its interval-representation, an IN envelope is represented by L equally spaced intervals from $h=0$ to $h=1$ included. Therefore, any population of data samples in the set X is represented by L intervals.

4. A K-map Extension

The Karnaugh map, or K-map for short, is a technique popular in digital electronics for simplifying a Boolean algebra expression [15]. More specifically, the contents of a Boolean Truth Table are transferred onto a two-dimensional grid whose cells are ordered in a Gray code. In conclusion, a K-map simplifies a Boolean

logic expression so that the latter can be implemented using a minimum number of physical logic gates [15]. In particular, a *sum-of-products* can be implemented using AND gates feeding into an OR gate.

In the conventional K-map, a variable shifts between 1 and 0. This work describes an extension of K-map such that a variable may shift among more than two values as illustrated next.

Consider three nominal data dimensions, i.e. $L_a=3$, such that $V_1= \{0,1,2\}$, $V_2= \{0,1,2,3\}$ and $V_3= \{0,1\}$. Furthermore, consider the four instances (data) shown in the lines of Table 1.

Table 1: A line of this Table displays nominal data

Nominal data dimension		
1	2	3
0	3	0
2	3	0
1	3	0
1	3	1

In terms of the K-map, each line in Table 1 is interpreted as a product term. The sum of all the four terms in Table 1 equals “030 + 230 + 130 + 131 = (0+2+1)30 + 13(0+1)= *30 + 13*”, where the term “131” was repeated twice to facilitate simplification. Hence, the four data in Table 1 can be represented by the disjunction of two strings of characters, namely “*30” and “13*”, respectively. Note that an asterisk (*) in a nominal data dimension denotes all the nominal data values in the corresponding dimension. In the aforementioned manner, we have extended the conventional K-map from binary (Boolean) values to N-ary (nominal) values. In conclusion, a simplified sum-of-products expression is produced.

Fig.1 presents a simple algorithm, namely “K-map*”, for inducing formal concepts from nominal data exclusively as it was illustrated above in this section. Note that algorithm “K-map*” induces labelled formal concepts per class. Fig.2 presents another simple algorithm, namely “K-mapExt”, for inducing hybrid formal concepts including both nominal data (induced by algorithm “K-map*”) and numerical data such that one IN is induced per numerical data dimension per (hybrid) formal concept. Optimized extensions of the basic K-map will be pursued in a future work.

Testing by algorithm “K-mapExt” is straightforward by computing the k-Minkowski metric distance of a testing datum T_S from all the labelled formal concepts. In conclusion, T_S is assigned the class label of its nearest formal concept. That is, the “K-mapExt” classifier carries out kNN classification for k=1.

```

for c=1 to c=Nclasses do
  for i=1 to i=La do
    if all the Ki discrete values of the nominal
      attribute Xi appear in dimension i then replace
        all the Ki discrete values by “*”.
    end for
  end for

```

Fig.1. Algorithm “K-map*” for inducing formal concepts from nominal data attributes alone.

```

Engage the “K-map*” algorithm (of Fig. 1) to
induce formal concepts from nominal data attributes
exclusively.
for each formal concept FC do
  for i=1 to i=Ra do
    Use algorithm CALCIN to induce an IN from
      all the numerical data in dimension i whose
        corresponding nominal data are included in FC.
    end for
  end for

```

Fig.2. Algorithm “K-mapExt” for inducing hybrid formal concepts including both nominal and numerical data.

5. Computational Experiments

In this section we comparatively present preliminary computational experiments and results regarding an application of the K-mapExt classifier on a benchmark dataset from the UCI Machine Learning Repository [2]. In particular, we employ the Statlog (Australian-Credit-Approval) benchmark dataset, which includes 690 instances (with 14 attributes including 6 numerical and 8 nominal attributes per instance) partitioned in 2 classes. There are missing values; more specifically, 37 instances (5% of the data) have one or more missing values, which were replaced by the mode and the mean of the corresponding attribute for categorical and numerical attributes, respectively. Since a training/testing dataset is not given explicitly, in our computational experiments we engaged a randomly selected 75% of the data instances for training, whereas the remaining 25% data instances were used for testing. A nominal dimension X takes values in the set $\Omega = \{0, \dots, K-1\}$; we assumed a measure $m_{\Sigma_\Omega} : \Sigma_\Omega \rightarrow \mathbb{R}_0^+$

such that $m_{\Sigma_\Omega}(\{0\}) = \dots = m_{\Sigma_\Omega}(\{K-1\}) = \frac{1}{K}$. Hence,

$$d(\{i\}, \{k\}) = \frac{2}{K}, \text{ for } i \neq k, \text{ and } d(\{i\}, \Omega) = 1 - \frac{1}{K}. \text{ Each}$$

numerical dimension was normalized by a linear transformation such that a dimension’s min and max values were mapped to the numbers 0 and 1, respectively. In conclusion, one IN was induced from

the normalized training data per numerical dimension by algorithm CALCIN. We used $L=32$ levels for representing an IN. For every numerical data dimension we employed the same functions $v(x) = x$ and $\theta(x) = 1-x$ for $x \in [0,1]$.

The best “K-mapExt” classification accuracy of 74.67% for the Statlog (Australian-Credit-Approval) benchmark dataset is shown in Table 2 as well as the accuracy of the ZeroR classifier from the literature [2]. Note that the difference in the accuracies of the two classifiers is not statistically significant. The corresponding number of induced formal concepts was 143 and 122 for class-1 and class-2, respectively. Fig.3 and Fig.4 display a hybrid formal concept regarding class-1 and class-2, respectively. Note that the nominal data in Fig.1/2(a) as well as the INs in Fig.1/2 (b)-(g) can be interpreted as descriptive decision-making knowledge (rules) induced from the training data.

6. Discussion and Conclusion

This work has proposed an enhancement of Formal Concept Analysis (FCA) by Lattice Computing (LC) techniques toward classification. A novel formal concept definition, that is an interval in a lattice-ordered data domain, has been proposed. It was explained how the proposed formal concept is also based on a Galois connection. Moreover, the proposed formal concept can rigorously include both nominal and numerical data.

A simple kNN algorithm, namely K-mapExt, was introduced as an extension of the K-map technique from digital electronics. The capacity of the K-mapExt algorithm was demonstrated in a benchmark classification problem including both nominal and numerical data. Apart from its tunability as well as its capacity for disparate data unification, an advantage of the proposed classifier is its capacity to deal with non-trivial data toward accommodating vagueness.

Future work may proceed along several directions. In particular, intervals of (Type-1) INs, namely Type-2 INs, might be considered toward computing with words [18], [23]; alternative (tunable) measure functions may be considered per nominal data dimension as well as alternative (tunable) functions $v(x)$ and $\theta(x)$ may be considered per numerical data dimension toward optimizing classification performance. Using an inclusion measure σ instead of a metric distance d is another promising direction toward a semantic optimization based on the computation of “maximal” formal concepts [11] as it will be detailed elsewhere.

Table 2: Testing data accuracy of two different classifiers regarding the Statlog (Australian-Credit-Approval) benchmark dataset

Classifier name	Accuracy (%)
ZeroR	75.93
K-mapExt	74.67

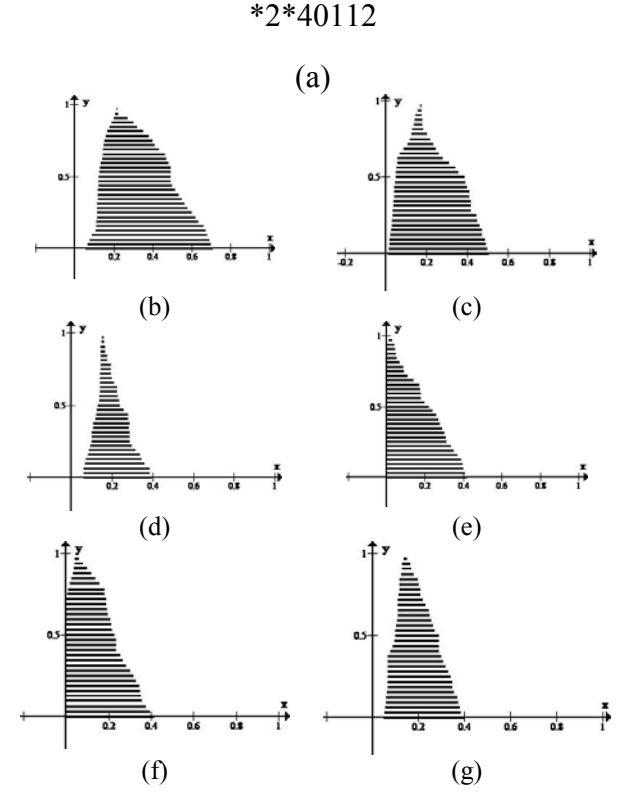


Fig.3. A hybrid formal concept induced from class-1 training data regarding the “Statlog (Australian-Credit-Approval)” benchmark dataset. (a) The nominal data term “*2*40112” computed by Algorithm K-map* (Fig. 1). The remaining figures display INs, in their interval-representation, induced from normalized numerical data in dimensions (b) 2, (c) 3, (d) 7, (e) 10, (f) 13 and (g) 14.

Appendix

This Appendix includes five mathematical definitions.

A **metric** in a space S is a function $d: S \times S \rightarrow \mathbb{R}_0^+$ that satisfies:

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

(M2) $d(x,y) = d(y,x)$ — *Symmetry*.

(M3) $d(x,z) \leq d(x,y) + d(y,z)$ — *Triangle Inequality*.

An **inclusion measure** in a lattice (L, \sqsubseteq) is a function $\sigma: L \times L \rightarrow [0,1]$ that satisfies:

(C1) $u \sqsubseteq w \Leftrightarrow \sigma(u,w) = 1$.

(C2) $u \sqsubseteq w \Leftrightarrow \sigma(x,u) \leq \sigma(x,w)$.

A σ -algebra Σ_Ω over a set Ω is a collection of subsets of Ω that satisfies:

($\Sigma 1$) $\emptyset \in \Sigma_\Omega$.

($\Sigma 2$) $A \in \Sigma_\Omega \Rightarrow A^c = (\Omega - A) \in \Sigma_\Omega$.

($\Sigma 3$) given $A_i \in \Sigma_\Omega$, where $i \in D$ and D is a countable indexing set, it follows $(\bigcup_{i \in D} A_i) \in \Sigma_\Omega$.

A **measure** is a set function $m_S: S \rightarrow \mathbb{R}_0^+$ that assigns a size to every “measureable set” element in S such that:

(S1) $m_S(\emptyset) = 0$, and

(S2) for any countable (including finite) indexing set D and any collection of pairwise disjoint sets $A_i \in S$ indexed by $i \in D$ it holds $m_S(\bigcup_{i \in D} A_i) = \sum_{i \in D} m_S(A_i)$.

A function $\psi: (P, \Xi) \rightarrow (Q, \Xi)$ between posets (P, Ξ) and (Q, Ξ) is called (**order**) **isomorphic** iff both “ $x \Xi y \Leftrightarrow \psi(x) \Xi \psi(y)$ ” and “ ψ is onto Q ”. Posets (P, Ξ) and (Q, Ξ) are called **isomorphic**, symbolically $(P, \Xi) \cong (Q, \Xi)$.

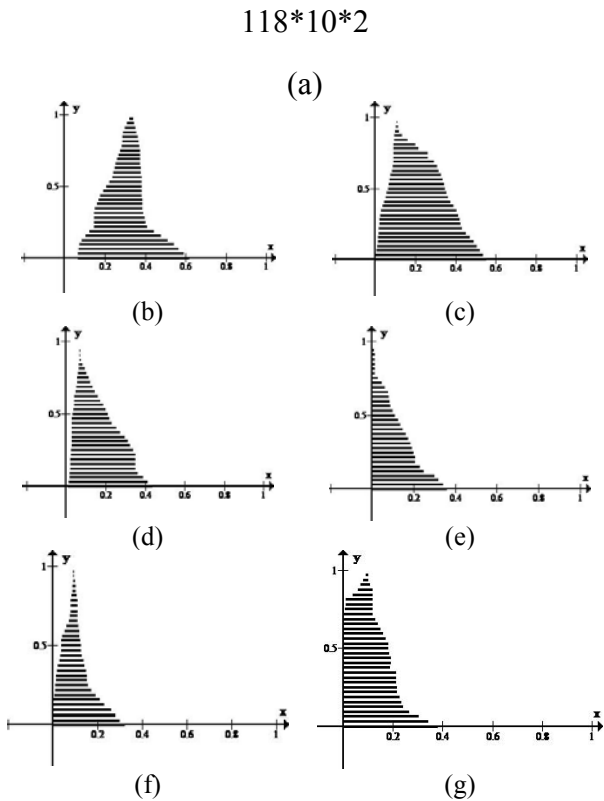


Fig.4. A hybrid formal concept induced from class-2 training data regarding the “Statlog (Australian-Credit-Approval)” benchmark dataset. (a) The nominal data term “118*10*2” computed by Algorithm K-map* (Fig. 1). The remaining figures display INs, in their interval-representation, induced from normalized numerical data in dimensions (b) 2, (c) 3, (d) 7, (e) 10, (f) 13 and (g) 14.

References

1. Atif, J., Bloch, I., Distel, F., Hudelot, C., “Mathematical morphology operators over concept lattices”, in *Intl. Conf. on Formal Concept Analysis (ICFCA’13)* (Lecture Notes in Computer Science Series 7880), P. Cellier, F. Distel, B. Ganter, Eds. Springer-Verlag, Berlin, Germany, 2013, pp. 28-43.
2. Bache, K., Lichman, M., UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science, 2013.
3. Caro-Contreras, D.E., Mendez-Vazquez, A., “Computing the concept lattice using dendritcal neural networks”, in *Proc. of the Tenth Intl. Conf. on Concept Lattices and Their Applications (CLA 2013)*, M. Ojeda-Aciego and J. Outrata, Eds. CEUR WS 1062, pp. 141-152.
4. Ganter, B., Wille, R., *Formal Concept Analysis*. Springer, Heidelberg, Germany, 1999.
5. Graña, M., “Lattice computing in hybrid intelligent systems”, in *Proc. 12th Intl. Conf. Hybrid Intel. Systems (HIS)*, Dec. 2012, pp. 1-5.
6. Jamshidi, Y., “gsaINknn: a GSA optimized, lattice computing kNN classifier”, *Engineering Applications of Artificial Intelligence*, under revision
7. Kaburlasos, V.G., *Towards a Unified Modeling and Knowledge-Representation Based on Lattice Theory*. Springer ser. Studies in Computational Intelligence 27, Heidelberg, Germany, 2006.
8. Kaburlasos V.G., Ed., Special issue on: Information engineering applications based on lattices, *Information Sciences* 181(10), 2011, pp. 1771-1773.
9. Kaburlasos, V.G., Kehagias, A., “Fuzzy inference system (FIS) extensions based on lattice theory”, *IEEE Trans. Fuzzy Systems*, Digital Object Identifier: 10.1109/TFUZZ.2013.2263807
10. Kaburlasos, V.G., Pachidis, T., “A Lattice-Computing ensemble for reasoning based on formal fusion of disparate data types, and an industrial dispensing application”, *Information Fusion* 16, 2014, pp. 68-83.
11. Kaburlasos, V.G., Petridis, V., “Fuzzy lattice neurocomputing (FLN) models”, *Neural Networks*, 13(10), 2000, pp. 1145-1170.
12. Kaburlasos, V.G., Athanasiadis, I.N., Mitkas, P.A., “Fuzzy lattice reasoning (FLR) classifier and its application for ambient ozone estimation”, *International Journal of Approximate Reasoning* 45(1), 2007, pp. 152-188.
13. Kaburlasos, V.G., Papadakis, S.E., Papakostas, G.A., “Lattice computing extension of the FAM neural classifier for human facial expression recognition”, *IEEE Trans. Neural Networks and Learning Systems* 24(10), 2013, pp. 1526-1538.
14. Kaburlasos, V.G., Papakostas, G.A., Pachidis, T., Athinellis, A., “Intervals’ numbers (INs) interpolation/extrapolation”, in *Proc. IEEE Intl. Conf. on Fuzzy Systems (FUZZ-IEEE)*, July 2013.
15. Karnaugh, M. “The map method for synthesis of combinational logic circuits”, *Trans. American Institute of Electrical Engineers - I* 72(9) 1953, pp. 593-599.
16. Krajca, P., Outrata, J., Vychodil, V., “Advances in algorithms based on CbO”, in *Proc. of the Seventh Intl. Conf. on Concept Lattices and Their Applications (CLA 2010)*, M. Kryszkiewicz, S. Obiedkov, Eds. CEUR WS 672, pp. 325-337.

17. Maragos, P., *Advances in Imaging and Electron Physics* 177. Chapter Two: Representations for Morphological Image Operators and Analogies with Linear Operators, pp. 45-187, Elsevier Inc., 2013.
18. Mendel, J.M., John, R.I., Liu, F. "Interval type-2 fuzzy logic systems made simple", *IEEE Trans. Fuzzy Systems* 14(6), 2006, pp. 808-821.
19. Pedrycz, W., Skowron, A., Kreinovich, V. (Eds.), *Handbook of Granular Computing*. John Wiley & Sons, 2008.
20. Sussner, P., Nachtegaal, M., Mélangé, T., Deschrijver, G., Esmi, E., Kerre, E., "Interval-valued and intuitionistic fuzzy mathematical morphologies as special cases of L-fuzzy mathematical morphology", *Journal of Math Imaging and Vision* 43(1), 2012, pp. 50-71.
21. Tsoukalas, V.T., Kaburlasos, V.G., Skourlas, C., "A granular, parametric KNN classifier", in *Proc. 17th Panhellenic Conf. on Informatics (PCI)*, Sept. 2013, pp. 319-326.
22. Valle, M.E., Sussner, P., "Quantale-based autoassociative memories with an application to the storage of color images", *Pattern Recognition Letters* 34(14) 2013, pp. 1589-1601.
23. Wang, D., Wang, Y., "Disease-related gene identification by an improved type-2 fuzzy method on microarrays", *Journal of Engineering Science and Technology Review* 6(2) 2013, pp. 78-84.
24. Zapata, F., Kreinovich, V., Joslyn, C., Hogan, E., "Orders on intervals over partially ordered sets: extending Allen's algebra and interval graph results", *Soft Computing* 17 2013, pp. 1379-1391.