Applications of Product Boolean Algebras in Cluster Analysis

González, Carlos G.[a] Maia. E.C.[b]
[a]*Centro Universitário N. Sra. do Patrocínio, Itu–SP–Brazil* [b]*Uniso, Sorocaba–SP–Brazil*

Abstract. A structural analysis of product Boolean algebras is used as an intuitive ground for the formulation of algorithms in pattern recognition. In this analysis, we show some mathematical results to elucidate the properties of product Boolean algebras relevant to understand how its structure can be used for classification in the feature space. In this sense, we discuss two algorithms for clustering selection in feature spaces where the clusters are not clearly delimited.

1 Introduction

In [2] was proposed that the product Boolean algebras are the most natural way to deal with bit sequences, as long as we want to use a structured algebraic framework. In that paper, a study of the relationship between the bit sequences and the basic structural properties of product Boolean algebras is carried out. The aim of this paper is to continue that work and of the [3] in showing the manner how a deeper structural analysis of these algebras can be used to yield new ways to generate algorithms.

The organization of this paper is as follows. In Section 2 we discuss the distance concept in a general, abstract way. In Section 3 we present the definition of product Boolean algebra and define a distance function into this algebra: the Boolean valued distance. This distance concept is fundamental in the approach of this paper, and the sequel of this paper is based on it. In Section 4 we show how the Boolean space can be partitioned into clusters by using the Boolean valued distances. Finally, in Section 5 we present two algorithms that show how to apply the theoretical concepts.

2 Similarity and Distance Concepts

Let us assume as known the basic concepts of pattern recognition as feature vector, feature space, and so on (see [6], 3). Following [6], let X be a data set. The (hard) clustering of X is a partition of X into n sets $X_0, X_1, \ldots, X_{n-1}$ such that the vectors in X_i are "more similar" to each other than the feature vectors of the other clusters. But "more similar" is a vague and imprecise expression and it should be replaced for a more precise one. In order to formalize the concept of similarity, several measure concepts were proposed (see, for instance, [4], 271–272 and [1], 130–136). The most used is a concept of *dissimilarity measure*, or *distance*, formalized by using the framework of the topology of metric spaces.

DEFINITION 2.1. *Let E be a feature space. A map $d : E \times E \longrightarrow R$ (the set of real numbers) is a dissimilarity measure or distance if:*

a) $d(x,x) = 0,\ \forall x \in E;$
b) $d(x,y) = d(y,x),\ \forall x,y \in E;$
c) $d(x,y) = 0 \ \Rightarrow\ x = y,\ \forall x,y \in E;$
d) $d(x,z) \leq d(x,y) + d(y,z),\ \forall x,y,z \in E.$

The question we want to discuss is: can we replace R by another set appropriate to give measures? This requires an abstract analysis of the conditions of Definition 2.1 and in this analysis we prefer to talk about an indetermined set A, instead of R, and about a distinguished element d_0, instead of 0[1]. Condition a) states that each element of the space is identical to itself, i.e., that the distance function takes a d_0 value, where d_0 denotes the non-dissimilarity. Furthermore, it can be necessary to establish certain properties of this d_0 w.r.t. the operation $+$ in d). Condition b) is only the symmetry of the distance function. The c) is the classic identity of the indiscernibles, i.e., the indistinct elements of the space must be the same. At first look, it seems that this is not a very important condition, since we can set the indistinct elements in the same class, and then we work with these classes in some kind of quotient space. But this condition prevents the collapse of d, in the sense that $\forall a, b \in A\ d(a,b) = d_0$.

Whereas conditions a), b) and c) do not implicate structural properties on the set A, condition d) requires a relation $\leq$ and an operation $+$. We will assume once more an abstract viewpoint and we will discuss about the relation ρ and the operation $\star$, instead of $\leq$ and $+$. First, we analyze the question: what about the relation ρ? We want to use ρ for clustering. The minimum for this purpose seems to be that ρ can be used to fix limits and then we can take the elements inside these limits to select a cluster. On the one hand, if ρ has cyclic subgraphs[2], we cannot take these limits in a simple way (this means that ρ is a too connected relation). On the other hand, if there exists many isolated points in A, then ρ also seems to be not appropriate for clustering. If ρ is an order or an order alike relation, then ρ seems to induce a structure of A appropriate to take these limits. With an "order alike relation" we indicate a relation that with a little work can be transformed into an order, for instance, making the transitive closure or taking out some points. This order represents the intuitive meaning of "more similar" in a natural way. In brief, the use of a relation ρ to select clusters can be extremely difficult when ρ is not an order.

Let us assume that ρ is an (partial) order. The operation $\star$ represents some kind of cumulative condition on the set A, since condition d) imposes that $\star$ must satisfy $\forall a \in A\ a \leq a \star a$ (at least, for the image of d). Other properties of $\star$ can be necessary in order to represent the cumulative properties of the distances: for instance, $\forall a, b \in A\, a \star b = b \star a$, $\forall a \neq d_0\, \forall b\ b < a \star b$, and so on. If we pretend that ρ represents the degree of dissimilarity, it can be interesting that the d_0 will be the minimum of A with respect to ρ, and that $\forall a \in A\ a = a \star d_0$. In a general way, the condition d) implicates the lack of shortcuts by using $\star$. In other words, $d(x,y)$ is the shorter path between a and b, that is, the intuitive meaning of the triangular inequality. This premise can generate a further discussion, but it is outside the proposal of this paper.

Actually, the properties, relations and operations of the real numbers were used in clustering. For instance, a definition of the Euclidean distance is widely used in clustering, and requires not only the sum, but also the difference, the square and the square root. The purpose of this discussion is not how to eliminate the measures onto the reals, but in what cases an another set can be an appropriate candidate. In this sense, we propose a more general definition of a distance map:

[1][6], 358, also uses d_0, but $d_0 \in R$.
[2]Non trivial cyclic subgraphs, i.e. with more than one element.

DEFINITION 2.2. *Let E be a feature space, A a set, $\leq$ a partial order on A, $\star$ a binary operation on A and d_0 a distinguished element of A. Then, $d : E \times E \longrightarrow A$ is a distance map if:*

a) $d(x, x) = d_0, \ \forall x \in E$;
b) $d(x, y) = d(y, x), \ \forall x, y \in E$;
c) $d(x, y) = d_0 \ \Rightarrow \ x = y, \ \forall x, y \in E$;
d) $d(x, z) \leq d(x, y) \star d(y, z), \ \forall x, y, z \in E$.

The use of this abstract definition is limited by the choice of appropriate A, $\leq$, $\star$, and d_0.

3 Bit sequences and product Boolean algebras

In order to minimize the processing time, bit sequences have been used a long time ago for encoding information. Thus, instead of dealing with a single information each time, a parallel processing of information is carried out. In spite of the widely divulged use of bit sequences, the structure of the product Boolean algebras was rarely applied as their theoretical framework. Actually, the algebras that can be used in this sense are products of several (but finite number) two-valued, $\{0, 1\}$, algebras. The elements of these algebras are sequences of 0's and 1's, immediately related to bit sequences. In the sequel, we summarize the main concepts of the Boolean algebras needed in this paper[3]. First, we need to define a lattice.

DEFINITION 3.1. *A lattice $\langle L, \leq \rangle$ is a partially ordered set L with an order relation $\leq$, a join or supremum operation ($\vee$) and a meet or infimum operation ($\wedge$), such that for any $\{x, y\} \subseteq L$, has a supremum and an infimum.*

A lattice can be also denoted as $\langle L, \vee, \wedge \rangle$. Let us denote $\mathbb{0}$ the minimum of a lattice, and $\mathbb{1}$ the maximum of the lattice, if they exist. Now, we can define a Boolean algebra. For $x \in L$, a complement for x is a $y \in L$ such that $x \wedge y = \mathbb{0}$ and $x \vee y = \mathbb{1}$.

DEFINITION 3.2. *A Boolean algebra $\langle B, \wedge, \vee, \sim, \mathbb{0}, \mathbb{1} \rangle$ is a distributive lattice with $\mathbb{0}$ and $\mathbb{1}$, such that each $x \in B$ has a (unique) complement.*

For a family of Boolean algebras, we can yield a new Boolean algebra that is the product of this family (see [5], 40–42):

DEFINITION 3.3. *Let $\langle A_0, \ldots, A_{n-1} \rangle$ be a family of Boolean algebras. The product Boolean algebra is $\langle A, \wedge, \vee, \sim, \mathbb{0}, \mathbb{1} \rangle$, where $A = A_0 \times \ldots \times A_{n-1}$ and the operations defined point to point, i.e.,*

$$\langle a_0, \ldots, a_{n-1} \rangle \wedge \langle b_0, \ldots, b_{n-1} \rangle = \langle a_0 \wedge b_0, \ldots, a_{n-1} \wedge b_{n-1} \rangle$$
$$\langle a_0, \ldots, a_{n-1} \rangle \vee \langle b_0, \ldots, b_{n-1} \rangle = \langle a_0 \vee b_0, \ldots, a_{n-1} \vee b_{n-1} \rangle$$

and the order in the same way:

$\langle a_0, a_1, \ldots, a_{n-1} \rangle \leq \langle b_0, b_1, \ldots, b_{n-1} \rangle$ *iff* $a_0 \leq b_0$ *and* $a_1 \leq b_1$ *and* $\ldots$ *and* $a_{n-1} \leq b_{n-1}$.

[3]For a detailed treatment see [5].

The product Boolean algebras used in this paper are all (finite) products of $\{0, 1\}$ algebras, and therefore their elements are always sequences of 0's and 1's.

In the context of product Boolean algebras, we can choose as $d : B \times B \longrightarrow R$ the function that $d(x, y) = n$ where n is the number of bits in x that are different from thoose in y. We will show that this d satisfies the conditions of Definition 2.1. The only condition that requires an argument is d): let x, y and z be bit sequences and let $d(x, z) = \varepsilon$. We can suppose without loss of generality that $d(x, y) = \sigma$, with $\sigma < \varepsilon$. Let us consider the bits with the same value in x and y. We see that y and z must differ at least in $\varepsilon - \sigma$ bits. Otherwise, let y and z differ in κ bits, with $\kappa \leq \varepsilon - \sigma$. Then, since these bits have the same value in x and in y, then only κ of the considered bits of x and of z were different, being $d(x, z) = \kappa + \sigma < \varepsilon$, a contradiction. Thus the function d is a distance in the sense of Definition 2.1.

The approach that is based on the number of bits that are different in both sequences is useful, but it does not matter *which* are the different bits. In [3] we have introduced another notion of distance that takes into account this fact. This can be necessary, for instance, if we are looking for vectors that differ from the representative in a determined way, i.e., in certain bits and not in n whichever bits. For these cases, the values of the function d can be taken in the same Boolean algebra which represents the feature space and we define $d(x, y) = x$ XOR y, the point to point XOR. We need also the ρ and the $\star$ of Definition 2.2. The order of the Boolean algebra is the natural candidate for ρ. For the $\star$ we choose the join $\vee$. Thus, we can define:

DEFINITION 3.4. *Let B be a product Boolean algebra. $d : B \times B \longrightarrow B$ is a Boolean valued distance if:*

a) $d(x, x) = \mathbb{0},\ \forall x \in B$;
b) $d(x, y) = d(y, x),\ \forall x, y \in B$*;*
c) $d(x, y) = \mathbb{0} \ \Rightarrow\ x = y,\ \forall x, y \in B$*;*
d) $d(x, z) \leq d(x, y) \vee d(y, z),\ \forall x, y, z \in B.$

PROPOSITION 3.1. *The operation XOR is a distance in the sense of Definition 3.4.*

Proof: As above, a), b) and c) are obvious. For d), notice that it suffices the argument for one bit, since all the operations and the order were defined point to point, i.e., bit to bit. Let x_n, y_n and z_n the n-th bit of x, y e z, respectively. We suppose, without loss of generality, that x_n XOR $z_n = 1$, the n-th bit is different in x from z. We have only two cases: or $x_n \neq y_n$ or $y_n \neq z_n$. Then, x_n XOR $y_n \vee y_n$ XOR $z_n = 1$.

4 Distances and clustering

The most natural way to select clusters into the feature space is to group in the same cluster the closest vectors, i.e. closest with respect to the distance map d in the sense that we fix a threshold of dissimilarity ε such that for x and y in the same cluster we have $d(x, y) \leq \varepsilon$. Another way consists in fixing once more a distance threshold ε, but defining that x and y belong to the same cluster if there exists a path $a = a_0, a_1, \ldots, a_n = b$ such that $d(a_i, a_{i+1}) \leq \varepsilon$. With the first approach, the simplest way is the existence of a set of representatives. Such a set of representatives can be conceived as having a central vector a in each cluster, so that it suffices to take every x such that $d(a, x) \leq \varepsilon$ to select each cluster. In the best case, this set of representatives partitions the feature space. Such a set is named *a complete set of representatives*, and defined:

DEFINITION 4.1. *Let A be a set, E a feature space, $\leq$ an order relationship on A, $\star$ an operation on $\star : E \times E \longrightarrow A$ and let ε be a distance threshold. Then, $D \subseteq E$ is a complete set of representatives if:*
a) $d(x, y) > \varepsilon \star \varepsilon,\ \forall x, y \in D$;
b) $\forall x \in E\ \exists y \in D \quad d(x, y) \leq \varepsilon$.

The existence of a complete representative set yields a partition of the feature space in clusters that group the closest vectors.

PROPOSITION 4.1. *Let D be a set satisfying the conditions of Definition 4.1. Then D yields a partition of the feature space.*

Proof: Let us assume the notation of Definition 4.1. We define $\overline{x} = \{y \in E\ :\ d(x, y) \leq \varepsilon\}$, for each $x \in D$. We will show that $P = \{\overline{x}\ :\ x \in D\}$ is a partition of E. First, note that $\overline{x} \neq \emptyset$, since $x \in \overline{x}$. For $\bigcup P = E$, only $E \subseteq \bigcup P$ needs an argument, since the other inclusion is immediate from the definition of the classes $\overline{x}$. Let $x \in E$. Hence, $\exists y \in D$, by b) of Definition 4.1, such that $d(x, y) \leq \varepsilon$, and then $x \in \overline{y} \in P$. We will show that $\overline{x} \bigcap \overline{y} = \emptyset$, for $\overline{x} \neq \overline{y}$. To get a contradiction, suppose $\exists z\ z \in \overline{x}$ e $z \in \overline{y}$. Then we have $d(x, z) \leq \varepsilon$ e $d(y, z) \leq \varepsilon$. We sum these inequalities and have $d(x, z) \star d(y, z) \leq \varepsilon \star \varepsilon$. By using b) and d) of Definition 2.1, we have $\varepsilon \star \varepsilon < d(x, z) \leq d(x, z) + d(z, y) = d(x, z) + d(y, z) \leq \varepsilon \star \varepsilon$.

If the feature space is isomorphic to $R^n,\ n \in N^+$, then we will not find frequently such a set of representatives. For a more general definition, we consider an ε_x for each representative x:

DEFINITION 4.2. *Let $E, A, \varepsilon, \leq$ and $\star$ be as above. Furthermore, let D be the set of the couples $\langle x, \varepsilon_x \rangle$ with $x \in E$ and ε_x a distance threshold depending of x. Then D is a relative set of representatives if:*
a) $d(x, y) > \varepsilon_x \star \varepsilon_y,\ \forall \langle x, \varepsilon_x \rangle, \langle y, \varepsilon_y \rangle \in D$;
b) $\forall x \in E\ \exists \langle y, \varepsilon_y \rangle \in D \quad d(x, y) \leq \varepsilon_y$.

The first problem that we find by using these definitions is that frequently the set of representatives does not exist. Another problem is that by using R to give distances, the feature spaces are partitioned into hyperspheres, so that it can be hard to put any vector in some hypersphere, since the space is not the (finite) union of hyperspheres. Alternatively, we can choose to partition the feature space into hypercubes, instead of hyperspheres. But if we use hypercubes of different sizes, we can have a vector in the vertices which is closer of the representative of another cluster than the cluster that it belongs. This is not the case with a Boolean algebra, since the distances that satisfy the Definition 3.4 yield clusters that are Boolean hypercubes.

Actually, once determined a distance threshold in a Boolean algebra, it is very easy to define a partition of the feature space. We note that:

$$H_{x,\varepsilon} = \{y \in B\ :\ d(x, y) \leq \varepsilon\}$$

is a Boolean hypercube and also a Boolean algebra[4] with the restriction of the Boolean operations to $H_{x,\varepsilon}$. We denote with $\bigwedge_{H_x} = x_0 \wedge x_1 \wedge \ldots \wedge x_n$, with $x_i \in H_{x,\varepsilon}$ the minimum of

[4] It is possible that H_x would not be a sub-algebra of B since the minimum and the maximum H_x can be different in B and in $H_{x,\varepsilon}$, but the Boolean operations of $H_{x,\varepsilon}$ are a restriction of the operations in B.

this algebra, and $\bigvee_{H_{x,\varepsilon}} = x_0 \vee x_1 \vee \ldots \vee x_n$ the maximum. Anytime that ε results obvious from the context, we use H_x. We will now define the useful concept of mask. Let x_i be i-th bit of x. Then

$$(\equiv_{H_x})_i = 1 \quad \text{iff} \quad \forall a, b \in H_x \; a_i = b_i$$

In other words, $\equiv_{H_x}$ has the i-th bit equal to 1 if and only if this bit is equal for all the vectors in H_x. We note that, for $a \in H_x$, it holds $a \wedge \equiv_{H_x} = \bigwedge_{H_x}$. For $A \subseteq B$, we state $\sim A = \{\sim a \;:\; a \in A\}$. Then we have $a \vee \sim\equiv_{H_x} = \bigvee_{H_x}$, and for $a, b \in H_x$, we have $a \wedge \equiv_{H_x} = b \wedge \equiv_{H_x}$. Each $x \in B$, whereas is considered a mask, partitions B into equivalence classes. As above, each such equivalence class is a Boolean algebra with the restricted operations and the minimum and maximum defined from the mask and any $x \in B$. Furthermore, for any class it can be found a distance threshold $\varepsilon \in B$ such that, $d(a, b) \leq \varepsilon$ if a and b belong to the same class. In brief, any vector of the Boolean algebra can partition the algebra into classes of similar vectors, whereas we consider it as a mask.

We have analyzed above the distance concept based on the number of different bits in two sequences. We can define now more formally:

DEFINITION 4.3. *Let $q(x)$ be the number of bits equal to 1 in $x \in B$.*

And $Q(x, y)$ as:

DEFINITION 4.4.

$$Q : B \times B \longrightarrow N \qquad Q(x, y) = q\left(\equiv_{\{x,y\}}\right)$$

5 Algorithms

The concepts that we have discussed above yield a new viewpoint useful for the development of algorithms, since the purely pragmatic manners to work with bit sequences can be replaced for general methods in a powerful framework. Consider, for instance, the binary morphology clustering algorithms (see [6], 516). The first step, the discretization of the feature space, has no sense, since the Boolean spaces are always discrete. It can be necessary to consider the "resolution" or granularity, because of the impossibility of the computational treatment of all the hypercubes of the Boolean space. In these cases we can proceed by using blocks of bits instead of all the elements in the Boolean space. Actually, we have implicitly defining a homomorphism of the initial algebra in the one formed by blocks of bits.

To present an algorithm (see [3]), we first need a definition:

DEFINITION 5.1. *Let B be a product Boolean algebra. For $n \in N$, let F_n be the family of distances of degree n, defined by:*

$$F_n = \{\varepsilon \in B \;:\; q(\varepsilon) = n\}$$

We can now present the following algorithm:

ALGORITHM 5.1. *Cluster Detection by Valley Creation Algorithm, CDVCA.*

```
Initially no element is consider as processed.
Fix b,n ∈ N.
Repeat
  Choose a nonprocessed x.
  Consider in the sequel only the nonprocessed elements.
  If there is some class H_{x,ε} with more than n elements
  for ε ∈ F_b:
    a) Choose the H_{x,ε}'s with the greatest number of
       elements.
    b) For H_{x,ε}, set H'_{x,ε} = H_{x,ε} − ⋂_{δ∈F_b} H_{x,δ}.
    c) Choose one H'_{y,ε}, with y ∈ H'_{x,ε}, with the greatest
       number of elements, create a new cluster H_{y,ε} and
       consider x and the elements of H_{y,ε} as processed.
  Else
    consider x as processed
  EndIf
Until all elements have been processed.
```

This algorithm is specially useful when, in the one hand, the feature space has regions with difference of density, but there is not “valleys”, i.e., regions of the space without elements that make the cluster selection easier, but, in the other hand, we do no want too few clusters. This is the purpose when we take out the intersection of the classes selected in a first step (the $H_{y,\varepsilon}$'s): we want to create “artificial valleys” and then we will use these valleys to divide too big regions. In this sense, the CDVCA has a certain analogy with the k-closest neighbors algorithms (see [1], 88). Notice that the structure of the Boolean algebra allows that there exists a procedure that in other spaces (for instance R^n) would be trivial. The distances in F_b formalize the tentatives to define the classes of the element x in different directions of the Boolean space. If we conceive the feature space as encoding information by using bit sequences, then these tentatives are looking for sets with properties that are common to several objects. Furthermore, it can be interesting to set a weight for each element in the Boolean algebra, i.e. reformulate the CDVCA, so that in step b), we will sum the weight of elements, instead of the number of the elements in each class, and then we will select the classes with the greatest weight.

Another algorithm that results of the application of the Boolean ideas is:

ALGORITHM 5.2. *Minesweeper.*

```
Fix a k ∈ N and a distance threshold ε.
Divide the Boolean space in hypercubes (i.e. algebras with
the same number of elements).
For each hypercube, calculate the number of objects in it.
For each hypercube, calculate the sum of the elements of
itself and the other hypercubes in the ε-neighbor.
This result in a number n for each hypercube.
Consider the hypercubes with n < k as processed and the
others as nonprocessed.
```

```
Repeat
  Choose a hypercube c among the nonprocessed of the
  greatest n.
  If there exists nonprocessed elements in c then
    Choose a nonprocessed element x of c.
    Create the cluster A = {y ∈ B : d(x,y) ≤ ε },
    by using only nonprocessed y.
    Consider c and the elements in A as processed.
  Else
    Consider c as processed.
  EndIf.
Until all hypercubes have been processed.
```

To find the ε-neighbor hypercubes we can use, for instance, the minimum of each hypercube (we have seen that the hypercubes are Boolean algebras). Two hypercubes c and d are neighbors if the minimum x of c is different and contiguous to the minimum y of d:

$$x \leq y \quad \text{and} \quad \neg\exists z \; z \neq x, z \neq y \; x \leq z \leq y$$

or

$$y \leq x \quad \text{and} \quad \neg\exists z \; z \neq x, z \neq y \; y \leq z \leq x.$$

We can consider once more the weight for reformulate the algorithm.

6 Conclusions

In our opinion, the use of product Boolean Algebras to represent bit sequences seems to be a promissory field for research and development. The use of the Boolean valued distance that is exposed shows to be fruitful in creating new viewpoints and applications. Some well-known concepts can be moved into the framework of product Boolean algebras to yield new tools and techniques, and we have intended to show this in the algorithm section. The structure of the product Boolean algebras can seem complex in a first view, but then happens to be very motivating as a conceptual way of dealing with information encoded using bit sequences.

References

[1] Abdel Belaïd and Yolande Belaïd. *Reconnaissance des Formes*. Inter Editions, Paris, 1992.

[2] Carlos G. González. Aplicação de produtos de álgebras de Boole para codificar seqüências de bits. In Jair Minoro Abe, editor, *LAPTEC 2000. Anais do I Congresso de Lógica Aplicada à Tecnologia*, pages 679–689, São Paulo–SP, Brasil, setember 2000. Editora Plêiade.

[3] Carlos G. González and Edson Campos Maia. Aplicações de álgebras de Boole produto em reconhecimento de padrões. *Revista de Estudos Universitários*, 27(1):75–88, 2001.

[4] A. K. Jain, M. P. Murty, and P. J. Flynn. Data clustering: A review. *ACM Computing Surveys*, 31(3):264–322, 1999.

[5] Roman Sikorski. *Boolean Algebras*. Springer Verlag, Heildelberg, 1969.

[6] Sergios Theodoridis and Konstantinos Koutroumbas. *Pattern Recognition*. Academic Press, San Diego, 1998.