Attributes for Understanding Groups of Binary Data

Arthur Chambon, Frédéric Lardeux, Frédéric Saubion, Tristan Boureau

Attributes for Understanding Groups of Binary Data

Arthur Chambon, Frédéric Lardeux, Frédéric Saubion
Université d'Angers
LERIA
Angers, France
{firstname.lastname}@univ-angers.fr

Tristan Boureau
Université d'Angers
UMR1345 IRHS
Angers, France
tristan.boureau@univ-angers.fr

***Abstract*—The analysis of groups of binary data can be achieved by logical based approaches. These approaches identify subsets of relevant Boolean attributes to characterize observations and may help the user to better understand their properties. In this work, our purpose is to highlight that different techniques may be used to compute subsets of attributes. We compare three different methods and propose a new algorithm for computing Boolean patterns. Experiments are performed on real biological data sets.**

***Index Terms*—Logical analysis of data; multiple characterizations; diagnostic test**

I. Introduction

Let us consider two groups (sets) of observations P and N (respectively positive and negative observations) defined over a set $\mathcal{A}$ of Boolean attributes. Our purpose is to compute a subset of $\mathcal{A}$ that may be used to explain/justify a priori the memberships of observations to their respective groups. Logical analysis of data (LAD) [11] combines concepts from partially defined Boolean functions and optimization in order to characterize such sets of data. Many applications have been pointed out, e.g., for diagnosis purposes when a physician wants to identify common symptoms that are shared by a group of patients suffering from similar diseases. Contrary to classification approaches issued from machine learning techniques (e.g., clustering algorithms) the purpose here is to provide an explicit justification of the data instead of an algorithm that assigns groups to incoming data. Note that we assume that the two groups are built by experts, or using expert knowledge (this is thus definitely not a classification nor a clustering problem).

As an example, let us consider a set of 8 Boolean attributes (labeled from a to h) and 7 observations, dispatched into two groups P and N.

Observ.	Groups	Attributes							
		a	b	c	d	e	f	g	h
1	P	0	1	0	1	0	1	1	0
2		1	1	0	1	1	0	0	1
3		0	1	1	0	1	0	0	1
4	N	1	0	1	0	1	0	1	1
5		0	0	0	1	1	1	0	0
6		1	1	0	1	0	1	0	1
7		0	0	1	0	1	0	1	0

In LAD methodology, a key concept consists in identifying patterns of similar values in groups. For instance, $a = 0$ and $b = 1$ is a pattern that is shared by observations 1 and 3 in P and such that no observation in N is covered by this pattern. Therefore this pattern could be interpreted as a partial explanation of the observations of group P. Among the sets of patterns, one has to decide which compromise has to be achieved between their size and the covering that they provide. Concerning the size of the patterns, some properties have been exhibited in order to focus on the most relevant one. In particular, prime patterns are patterns whose number of attributes cannot be reduced unless they are not patterns anymore. Prime patterns correspond to the simplicity requirement (in terms of attributes) while strong patterns correspond to an evidential preference where a larger cover is preferred (we refer the reader to [10] for a survey on LAD).

Alternatively, attributes f and g can also be used to generate a Boolean formula $\phi \equiv (f \wedge g) \vee (\neg f \wedge \neg g)$, which is true for observations in P (interpreted as Boolean assignments on attributes) and false for observations in N. Note that, the attribute b is not sufficient to explain group P since observation 6 in N has also this attribute set to 1. ϕ is presented here in disjunctive normal form. Note that such formula could be convenient for users, either by minimizing the number of attributes (for instance, to simplify their practical implementation in diagnosis routines) or by minimizing the size of the formula (for instance, to improve their readability). This approach focuses on minimal characterizations in terms of number of attributes and can be extended to consider several groups simultaneously [9].

Motivations: In this paper, our main goal is to study different techniques to select relevant attributes to help the user to better understand groups of binary data and eventually to identify relevant properties of data. Hence, we consider the two previously described approaches: patterns and minimal sets of attributes. We also aim to compare these methods with a feature selection technique (note that feature selection methods are commonly used to improve classification algorithms by identifying relevant attributes, but may also be useful for data visualization).

Contributions: We propose (1) A new algorithm to compute all prime patterns, (2) to compare the attributes that

are selected by the two above-mentioned logical characterization techniques, (3) to compare these approaches with a simple feature selection and (4) to study if the groups defined by experts could be explained by classification using the selected attributes. At last, we perform experiments on different real benchmarks issued from biology.

II. Logical Analysis of Data

Logical Analysis of Data (LAD) ([6], [3], [10], [14]) considers two groups of observations represented by Boolean vectors. The purpose is to find a justification/explanation of these groups. LAD is mainly based on the concept of partially defined Boolean functions [20]. A justification can be a formula that is satisfied by observations of one group (called the positive group) while being falsified by the observations of the other group (called the negative group). Such a formula is then built on a subset of attributes that discriminate one group against the other one. In this context, LAD focuses on the notion of pattern that corresponds to a subset of attributes whose values are similar for several observations in the positive group, which may help the user to identify common characteristics of these observations. From a practical point of view, LAD has been applied to many domains: biology and medicine ([18], [2], [1]), engineering [4], transportation [12]. The Characterization Problem [9] is an extension of the LAD methodology that considers simultaneously several groups of observations. The Characterization Problem consists in minimizing the number of attributes that are necessary to discriminate mutually several groups of observations.

From these previous works, we propose a unified presentation of these possible characterizations of groups of binary data.

A. Basic Principles: Boolean Functions

A Boolean function f of n variables, $n \in \mathbb{N}$, is a mapping $f : \mathbb{B}^n \mapsto \mathbb{B}$, where $\mathbb{B}$ is the set $\{0,1\}$. A vector $x \in \mathbb{B}^n$ is a *true vector* (resp. *false vector*) of the Boolean function f if $f(x) = 1$ (resp. $f(x) = 0$). $T(f)$ (resp. $F(f)$) is the set of *true vectors* (resp. *false vectors*) of a Boolean function f. A partially defined Boolean function (pdBf) on $\mathbb{B}^n$ is a pair (P,N) such that $P,N \subseteq \mathbb{B}^n$ and $P \cap N = \emptyset$. P is thus the set of positive vectors, and N the set of negative vectors of the pdBf (P, N). The notion of partially defined Boolean function is generalized by the following notion of term proposed in ([14], [5]).

A term is a Boolean function t_{σ^+,σ^-} whose true set $T(t_{\sigma^+,\sigma^-})$ is of the form: $T(t_{\sigma^+,\sigma^-}) = \{x \in \mathbb{B}^n | x_i = 1 \ \forall i \in \sigma^+, \ x_j = 0 \ \forall j \in \sigma^-\}$ for some set $\sigma^+, \sigma^- \subseteq \{1, 2, ..., n\}$, $\sigma^+ \cap \sigma^- = \emptyset$. A term t_{σ^+,σ^-} can be represented by a Boolean formula of the form: $t_{\sigma^+,\sigma^-}(x) = (\bigwedge_{i\in\sigma^+} x_i) \wedge (\bigwedge_{j\in\sigma^-} \neg x_j)$.

B. Formulation of the Problem

We define now a Binary Data Characterization Problem (BDCP).

Definition 1. *An instance of the Binary Data Characterization Problem is a tuple $(\Omega, \mathcal{A}, D, G)$ defined by a set of observations $\Omega \subseteq \mathbb{B}^{|\mathcal{A}|}$ of Boolean vectors built on a set $\mathcal{A}$ of Boolean attributes. The observations are recorded in a Boolean matrix $D_{|\Omega|\times|\mathcal{A}|}$. A function $G : \Omega \to \{P, N\}$ assigns a group $G(o)$ to the observation $o \in \Omega$.*

The matrix D is defined as:

- *the value $D[o, a]$ represents the presence/absence of the attribute a in the observation o.*
- *a line $D[o, .]$ represents the Boolean vector of presence/absence of the different attributes in the observation o.*
- *a column $D[., a]$ represents the Boolean vector of presence/absence of the attribute a in all the observations.*

Given a subset $A \subset \mathcal{A}$, D^A is a matrix reduced to the attributes of A.

As already mentioned, two possible notions of solution are considered:

- computing minimal sets of attributes that discriminate the groups,
- computing patterns that are shared by observations of the positive group.

C. Minimal Sets of Attributes

Given a BDCP instance $(\Omega, \mathcal{A}, D, G)$ the purpose here is to find a subset of attributes $S \subseteq \mathcal{A}$ such that two observations from two different groups are always different on at least one attribute in S.

Definition 2. *Given an instance $(\Omega, \mathcal{A}, D, G)$, a subset of attributes $S \subseteq \mathcal{A}$ is a solution iff $\forall (o, o') \in \Omega^2, G(o) \neq G(o') \rightarrow D^S[o, .] \neq D^S[o', .]$. In this case, the matrix D^S is called a solution matrix.*

An instance may have several solutions of different sizes. It is therefore important to define an ordering on solutions in order to compare and classify them. In particular, for a given solution S, adding an attribute generates a new solution $S' \supset S$. In this case we say that S' is dominated by S. We can also compute solutions of minimal size with regards to the attributes they involve.

A solution $S \subseteq \mathcal{A}$ is non-dominated iff $\forall s \in S, \exists (o, o') \in \Omega^2$ s.t. $G(o) \neq G(o')$ and $D^{S\setminus\{s\}}[o, .] = D^{S\setminus\{s\}}[o', .]$. A solution $S \subseteq \mathcal{A}$ is minimal iff $\nexists S' \subseteq \mathcal{A}$ with $|S'| < |S|$ s.t. S' is a solution.

According to previous works [9], given an instance $(\Omega, \mathcal{A}, D, G)$, the BDCP can be formulated as the following 0/1 linear program :

$$
\begin{aligned}
& min : \sum_{i=1}^{|\mathcal{A}|} y_i \\
& s.t. : \\
& C \ . \ Y^t \geqslant \mathbb{1}^t \\
& Y \in \{0,1\}^{|\mathcal{A}|}, Y = [y_1, ..., y_{|\mathcal{A}|}]
\end{aligned}
$$

where Y is a Boolean vector that encodes the presence/absence of the set of attributes in the solution. C is a

matrix that defines the constraints that must be satisfied in order to ensure that Y is a solution. Let us denote Θ the set of all pairs $(o, o') \in \Omega^2$ such that $G(o) \neq G(o')$. For each pair of observations (o, o') that does not belong to the same group, one must insure that the value of at least one attribute differs from o to o'. This will be insured by the inequality constraint involving the $\mathbb{1}$ vector (here a vector of dimension $|\Theta|$ that contains only values equal to 1). The minimization objective function insures that we aim to find a minimal solution.

More formally, C is a Boolean matrix of size $|\Theta| \times |\mathcal{A}|$ defined as:

- Each line is numbered by a couple of observations $(o, o') \in \Omega^2$ such as $G(o) \neq G(o')$ $((o, o') \in \Theta)$.
- Each column represents an attribute.
- $C[(o, o'), a] = 1$ if $D[o, a] \neq D[o', a]$, $C[(o, o'), a] = 0$ otherwise.
- We denote $C[(o, o'), .]$ the Boolean vector representing the differences between observations o and o' on each attribute. This Boolean vector is called constraint since one attribute a such $C[(o, o'), a] = 1$ must be selected in order to insure that no identical observations can be found in different groups.

Two algorithms have been proposed to compute solutions [8]:

- NDS (Non Dominated Solutions) that computes the set of all non-dominated solutions.
  This algorithm find all $Y \in \{0, 1\}^{|\mathcal{A}|}$ of the linear program above such as $C \,.\, Y^t \geqslant \mathbb{1}^t$
- MWNG (Merging with negative variables) that computes all minimal non dominated solutions.
  This algorithm finds all the solutions of the linear program above.

Note that the computation of all minimal non dominated solutions is related to the Min Set Cover problem and the Hitting Set problem [17].

D. Patterns

Let us consider $P = \{o \in \Omega | G(o) = P\}$ the group of positive observations and $N = \{o \in \Omega | G(o) = N\}$ the group of negative ones. A pattern aims to identify a set of attributes that have identical values for several observations in P. Of course this pattern must not appear in any observation of N.

Definition 3. *A pattern of a pdBf (P,N) is a term t_{σ^+,σ^-} such that $|P \cap T(t_{\sigma^+,\sigma^-})| > 0$ and $|N \cap T(t_{\sigma^+,\sigma^-})| = 0$.*

Given a term t, $Var(t_{\sigma^+,\sigma^-})$ is the set of attributes (also called variables) defining the term ($Var(t_{\sigma^+,\sigma^-}) = \{x_i | i \in \sigma^+ \cup \sigma^-\}$) and $Lit(t_{\sigma^+,\sigma^-}) = \{x_i \cup \bar{x}_j | i \in \sigma^+,\ j \in \sigma^-\}$ the set of literals (i.e. a logic variable or its complement) in t_{σ^+,σ^-}. Given a pattern p, the set $Cov(p) = P \cap T(p)$ is said to be covered by the pattern p.

Example 1. *Let us recall the introductory example.*

Observ.	*Group*	*Attributes*							
		a	*b*	*c*	*d*	*e*	*f*	*g*	*h*
1	*P*	*0*	*1*	*0*	*1*	*0*	*1*	*1*	*0*
2		*1*	*1*	*0*	*1*	*1*	*0*	*0*	*1*
3		*0*	*1*	*1*	*0*	*1*	*0*	*0*	*1*
4	*N*	*1*	*0*	*1*	*0*	*1*	*0*	*1*	*1*
5		*0*	*0*	*0*	*1*	*1*	*1*	*0*	*0*
6		*1*	*1*	*0*	*1*	*0*	*1*	*0*	*1*
7		*0*	*0*	*1*	*0*	*1*	*0*	*1*	*0*

$p_1 = \neg a \wedge b$ and $p_2 = \neg f \wedge \neg g$ are two patterns covering respectively observations 1 and 3 (p_1) and 2 and 3 (p_2).

Let us consider now $p_3 = f \wedge g$. p_2 and p_3 are two patterns using identical attributes: $Var(p_2) = Var(p_3)$ but $Lit(p_2) \neq Lit(p_3)$. $p_2 \cup p_3$ cover the positive group (since $Cov(p_2) \cup Cov(p_3) = P$) with only two attributes.

We consider different types of patterns:

Definition 4. *A pattern p is called* prime *if and only if the removal of any literal from $Lit(p)$ results in a term which is not a pattern.*

Obviously, a pattern is prime if and only if the removal of any variable from $Var(p)$ results in a term which is not a pattern.

In Example 1, $p_2 = \neg f \wedge \neg g$ is a prime pattern. $p_4 = \neg a \wedge b \wedge \neg c$ is not prime because pattern $p_1 = \neg a \wedge b$ is prime.

Definition 5. *A pattern p_1 is called* strong *if there does not exist a pattern p_2 such that $Cov(p_1) \subset Cov(p_2)$.*

In Example 1, $p_2 = \neg f \wedge \neg g$ is a strong pattern. $p_4 = \neg a \wedge b \wedge \neg c$ is not strong because $p_1 = \neg a \wedge b$ is a pattern and $Cov(p_4) = \{1\} \subset Cov(p_1) = \{1, 3\}$.

Definition 6. *A pattern p_1 is called* strong prime *if and only if*

1) *p_1 is a strong pattern and,*
2) *if there exists a pattern p_2 such as $Cov(p_2) = Cov(p_1)$ then p_1 is prime*

In [15], it has been proved that a pattern is strong prime if and only if it is both strong and prime. Therefore, the set of all strong prime patterns is the intersection of the set of all prime patterns and the set of all strong patterns.

In the experiments, we are interested in covering the whole set of observations P by a subset of prime and strong prime patterns. Such a complete cover of group P is considered as a solution for the pattern approach. This cover is thus considered as the justification of group P from the pattern point of view.

III. Computation of Prime Patterns and Group Covers

We propose a new algorithm that uses the computation of all non dominated solutions of the BDCP problem in order to compute prime patterns.

In LAD, the aim is to find a pattern that covers a maximum number of observations of P, such as no observation

of N contains this pattern. From BDCP point of view, the notion of solution is rather different. Given a solution S of a BDCP instance $(\Omega, \mathcal{A}, D, G)$ defined as above, the attributes of S do not generally correspond to a pattern for the observations in P, unless all observations are identical on S. In this case a solution of the BDCP obviously coincides with a prime pattern in terms of attributes.

In particular, if $|P| = 1$, the set of all solutions of the BDCP coincides in terms of attributes with the set of all prime patterns that cover the only observation in P, because in both cases no attribute can be removed.

Given a non-dominated solution S of the BDCP (computed by previously mentioned algorithms for instance), it is easy to transform an observation o of the group P into prime pattern p. Each attribute a of S appears positively (resp. negatively) in p if $D[o, a] = 1$ (resp. $D[o, a] = 0$). This transformation will be insured by the Transformation_Pattern procedure in Algorithm 1.

For each observation, we can generate all prime patterns that cover this observation. If we generate all prime patterns for all observations, we generate prime patterns p, and determine $Cov(p)$ for each one.

Algorithm 1 returns the set Pat of all prime patterns, and the set Cov of coverage of all patterns $p \in Pat$. Cov is a set of elements V_p, $\forall p \in Pat$. Each element V_p is a set of all observations covered by p.
Note that it is not necessary to compute the set Cov to generate the set Pat. Hence, each step that involves the set Cov can be removed.

Algorithm 1. Prime Patterns Computation (PP).

Data: D: matrix of data, with two groups {P,N}.
Result: Pat: set of all prime patterns
Result: Cov: set of covers of each prime pattern.
$Pat = \emptyset$
$Cov = \emptyset$
forall $o \in P$ **do**
  Generate the constraint matrix C_o as if o was the only one observation in P
  Sol={set of all non dominated solutions for C_o}
  forall $s \in Sol$ **do**
    p=Transformation_Pattern(s,o)
    if $p \notin Pat$ **then**
      $Pat = Pat \cup \{p\}$
      //Create a new element V_p of Cov which will be a set of observations covered by p.
      $V_p = \{o\}$
      $Cov = Cov \cup \{V_p\}$
    end
    else
      //V_p is already in Cov; update
      $V_p = V_p \cup \{o\}$
    end
  end
end
return Pat and Cov;

Note that Algorithm NDS [8] can also generate solutions of smaller size than a given bound B. Given a bound B, we can only generate prime patterns with a size inferior to B.

Now, using the set Cov we can run Algorithm 2 to compute only strong prime patterns. From the set of all covers, we can compute the subset of strong patterns among prime patterns.

Algorithm 2. Strong Prime Patterns Computation (SPP).

Data: Cov: set of coverage of each prime pattern.
Pat: set of all prime patterns
Result: SPP: set of all strong prime patterns.
$SPP = \emptyset$
forall $p \in Pat$ **do**
  if $\nexists p' \in Pat$ *s.t.* $Cov(p) \subset Cov(p')$ **then**
    $SPP = SPP \cup \{p\}$
  end
end
return SPP;

IV. Find a covering for a group with the minimal number of patterns

When computing all non-dominated solutions (see [8]), many solutions are generated. Hence, it is difficult to identify the most suitable solutions with regards to user-defined criteria. Another criterion could be to minimize the size of the Boolean formula of the solution for each group in disjunctive normal form (DNF) for sake of simplicity (as mentioned in Introduction). In DNF, conjunctions of literals are connected by the logical connector $\vee$ (disjunction), where each conjunction corresponds to a pattern. The size of the Boolean DNF formula is thus the number of patterns.

Example 2. *Let us consider again Example 1*

Observ.	*Groups*	*Attributes*							
		a	*b*	*c*	*d*	*e*	*f*	*g*	*h*
1	*P*	*0*	*1*	*0*	*1*	*0*	*1*	*1*	*0*
2		*1*	*1*	*0*	*1*	*1*	*0*	*0*	*1*
3		*0*	*1*	*1*	*0*	*1*	*0*	*0*	*1*
4	*N*	*1*	*0*	*1*	*0*	*1*	*0*	*1*	*1*
5		*0*	*0*	*0*	*1*	*1*	*1*	*0*	*0*
6		*1*	*1*	*0*	*1*	*0*	*1*	*0*	*1*
7		*0*	*0*	*1*	*0*	*1*	*0*	*1*	*0*

$S_1 = \{f, g\}$ is a solution of the corresponding BDCP. The Boolean formula corresponding to solution S_1 for the group P is (in DNF): $(f \wedge g) \vee (\neg f \wedge \neg g)$. The size of this Boolean formula is 2, because it contains two Boolean conjonctions (i.e. two patterns): $(f \wedge g)$ and $(\neg f \wedge \neg g)$.

When searching for a cover by means of patterns, we may again turn to a Min-Set Cover problem since we

search the smaller set of patterns that cover all observations of the group P. Using the set Cov generated by Algorithm 1, we can build a constraint matrix M for this problem, where each line i represents observations of the studied group P and each column j represents a prime pattern. $M_{[i,j]} = 1$ if the pattern j covers the observation i, 0 otherwise.

Now, finding a covering for the positive group with the minimal number of patterns leads to solving the min-set cover problem represented by the following linear program:

$$
\begin{aligned}
&min : \sum_{i=1}^{|\mathcal{A}|} y_i \\
&s.t. : \\
&M \,.\, Y^t \geqslant \mathbb{1}^t \\
&Y \in \{0,1\}^{|\mathcal{A}|}, Y = [y_1, ..., y_{|\mathcal{A}|}]
\end{aligned}
$$

where Y is a Boolean vector that encodes the presence/absence of each pattern in the solution. So, we can use any classic covering algorithm as the MWNG algorithm mentioned above to solve the problem and find the minimal number of patterns that fully cover the positive group.

V. Correlation Based Feature Selection (CFS)

In this section, we describe an attributes selection technique that could be relevant in our context of binary data. Among the feature selection methods, the filtering methods consist in classifying the attributes according to an appropriate selection criterion. This criterion generally depends on the relevance (i.e. correlation) of the attribute on a given cluster (i.e., group). Given a cluster, CFS aims at computing the subset of attributes that are relevant to justify this group from a classification point of view.

The **CFS** method [13] will be based on a measure μ evaluating a set of attributes $A \subseteq \mathcal{A}$ with regards to a group of observations G, taking into account the correlation between these attributes:

$$\mu(A, G) = \frac{m \times \bar{\rho}_{G,A}}{\sqrt{m + m \times (m-1) \times \bar{\rho}_{A,A}}}$$

where $m = |A|$. Value $\bar{\rho}_{G,A}$ is the average of the correlations between the attributes chosen and a cluster/group G:

$$\bar{\rho}_{G,A} = \frac{1}{m} \sum_{i=1}^{m} \rho_{G,A_i}$$

and $\bar{\rho}_{A,A}$ is the average of the cross-correlations between the selected attributes:

$$\bar{\rho}_{A,A} = \frac{2}{m \times (m-1)} \sum_{i=1}^{m-1} \sum_{j=i+1}^{m} \rho_{A_i,A_j}$$

where A_i is the attribute i in A. Our purpose consists thus in determining the subset A with the highest value $\mu(A, P)$ (where P is the set of positive observations). In experiments, we use a R library [1]. Once the attributes have been selected by CFS, we will check their relevance by running a classification method and observe if these attributes allow us to rebuild the initial groups. Moreover, we will be interested in comparing the attributes selected by CFS and the two previously described approaches.

VI. Experimental Study

In this section, our purpose is to compare the different sets of attributes computed by the different methods that have been presented. Remember that these attributes aim at characterizing the groups of data.Therefore, our experimental study can be sketched by Figure 1

In the first part of this section, given a set of instances, we compare :

- the number of attributes obtained by the attributes minimization approach and by the patterns minimization approach
- the number of required patterns for covering the positive group using the attributes computed by these two approaches.

In the second part of this section, we attribute sets obtained by minimization with those obtained by the variable selection method `CFS`. Finally, we check if the attributes obtained by the variable selection method allows us to highlight pertinent attributes for the characterization problem (i.e., minimization).

A. Data Instances

In order to evaluate and compare the previously described approaches, we consider different sets of observations.

- Instances ra100_phv, ra100_phy, rch8, ralsto, ra_phv, ra_phy, ra_rep1 and ra_rep2 correspond to biological identification problems. Each observation is a pathogenic bacterial strain and attributes represent genes (e.g., resistance genes or specific effectors). These bacteria are responsible of serious plant diseases and their identification is thus important.

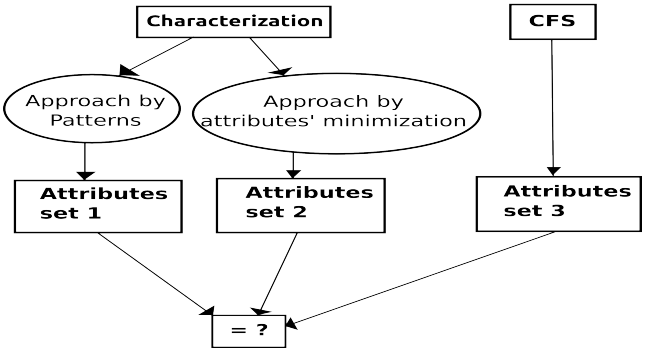

Fig. 1. Overview of experimental process

[1] https://www.rdocumentation.org/packages/FSelector/versions/0.21/topics/cfs

The main challenge for biologists is to characterize groups of bacteria using a limited number of genes to design simple and cheap diagnosis tests [7]. The original files are available[2]. In the original data sets, several groups were considered. Here, we have considered the first group of bacteria as the positive group and the union of the other groups as the negative group. Note that similar results have been obtained when considering other groups as positive groups.

- Instance vote_r is available on the Tunedit repository[3] for machine learning tests.

The characteristics of the instances (number of observations, size of the positive group and number of attributes) are described in Table I.

TABLE I
CHARACTERISTICS OF THE INSTANCES

Instances	Observations	Positive group size	# Attributes
ra100_phv	101	21	50
ra100_phy	105	31	51
rch8	132	5	37
ralsto	73	27	23
ra_phv	108	22	70
ra_phy	112	31	73
ra_rep1	112	38	155
ra_rep2	112	37	73
vote_r	435	168	16

B. Experiments

Let us recall the main characteristics of the three methods that we consider in these experiments. Our purpose is to compare these methods in terms of selected attributes.

- *MinS*: computes minimal sets of attributes to identify the group P (see Section II-C).
- *Pattern*: computes a cover of P using patterns (see Section II and IV).
- *CFS*: computes a set of relevant attributes for P with regards to correlation considerations (see Section V).

1) Number of Attributes: In Table II, we compare for (*MinS* and *Pattern*) the total number of attributes that are used. Concerning *MinS*, these attributes correspond to the attributes in a minimal solution. Concerning *Pattern*, we consider the minimal number of attributes that must be used to fully cover the set P using patterns. Note that, in both cases, several solutions may exist for the same instance.

And also note that the number of attributes is rather similar for both methods when considering small instances. Nevertheless, we observe a difference on ra_rep1 and ra_rep2 instances. Let us note that for ra_rep2 this difference represents only 11% of the total number of attributes of the instance, and only 6.5% for ra_rep1.

[2]http://www.info.univ-angers.fr/~gh/Idas/Ccd/ce_f.php
[3]http://tunedit.org/repo/UCI/vote.arff

TABLE II
NUMBER OF ATTRIBUTES USED BY *MinS* AND *Pattern*

Instances	# Attributes *MinS*	# Attributes *Pattern*
ra100_phv	2	2
ra100_phy	3	4
rch8	3	3
ralsto	5	5
ra_phv	2	2
ra_phy	3	4
ra_rep1	12	22
ra_rep2	11	19
vote_r	10	14

2) Number of Patterns for Covering P: In Table III, we focus on the patterns. The value # Patterns *MinS* corresponds to the minimal number of patterns that are required to cover the positive group when using only the minimal set of attributes computed by *MinS* (in this case we recompute the patterns for the set of attributes selected by *MinS*, see relationships in Section IV). The value # Patterns is the minimal number of patterns that are necessary to cover the positive group but here the cover is based on all possible patterns.

TABLE III
NUMBER OF PATTERNS FOR *MinS* AND *Pattern*

Instances	# Patterns *MinS*	# Patterns *Pattern*
ra100_phv	1	1
ra100_phy	5	2
rch8	1	1
ralsto	7	3
ra_phv	1	1
ra_phy	5	2
ra_rep1	27	7
ra_rep2	25	6
vote_r	71	10

On Table III, we observe that the attributes computed by *MinS* are not suitable for finding a good set of covering patterns (based on these attributes the number of patterns for building a cover of the positive set increases). It means that the explanation provided by *MinS* differs from the pattern approach in terms of patterns (i.e., the attributes are different).

3) Comparing MinS with CFS: In Table IV, we present the following observations:

- The number of minimal solutions computed by *MinS*,
- The number of attributes of these minimal solutions,
- The number of attributes of the best solution computed by *CFS*,
- The maximum number of common attributes between the subset computed by *CFS* and minimal solutions of *MinS*. Since many solutions can maximize the number of common attributes, we also indicate in parentheses how many solutions satisfy this criterion.

Remember that the CFS method computes a subset of relevant attributes. In order to evaluate the relevance of this subset of attributes, we propose to check that they

may be useful for data clustering (i.e., grouping together similar data). Therefore we use a clustering algorithm and check that the resulting clusters correspond to the initial positive and negative groups. We use here a k-means algorithm ([19], [16]) since it is a simple, efficient and well-known clustering technique (of course the number of clusters is set to 2).

The accuracy of the clustering is evaluated according to the similarity between predicted cluster c and the real original group r. Given n observations we consider two n dimensional vectors C and R such as c_i is the predict group of observation i and r_i corresponds to its real original group. We define the accuracy as:

$$Acc(C,R) = |2 \times (\frac{1}{n}\sum_{i=1}^{n}(r_i - c_i)^2) - 0.5|$$

Note that $Acc(C,R) \in [0,1]$. Values close to 1 correspond thus to a high accuracy.

For each instance the accuracy of the clustering based on the attributes selected by *CFS* is presented between parentheses. Since k-means is a statistical method we repeat the clustering process and evaluate the mean value of accuracy over 20 independent runs. The last column corresponds to the maximal number of common attributes between the subset computed by CFS and minimal solutions computed by *MinS*. Since several solutions may maximize the number of common attributes we also indicate in parentheses how many solutions satisfy this criterion.

TABLE IV
MinS vs. *CFS*

Instances	# *MinS* sol	# att. in min sol.	# att. *CFS* (accur.)	# com. att. (# sol)
ra100_phv	1	2	4 (0.465)	0 (1)
ra100_phy	1	3	1 (0.562)	1 (1)
rch8	1	3	6 (0.439)	0 (1)
ralsto	5	5	3 (0.288)	2 (1)
ra_phv	1	2	3 (0.519)	0 (1)
ra_phy	1	3	1 (0.589)	1 (1)
ra_rep1	134	12	6 (0.393)	3 (1)
ra_rep2	106	11	11(0.536)	3 (26)
vote_r	1	10	4 (0.706)	3 (1)

Using an attribute selection technique such as CFS in order to explain our data by clustering techniques (i.e., distance based methods) appears not really relevant here. Note that attributes selected by *CFS* are different from attributes computed for *MinS*. Moreover, *MinS* reduces the number of attributes used for characterization. Note that we have also performed clustering using the attributes computed in *MinS* solutions leading also to poor clustering accuracy (but *MinS* is definitely not a feature selection method since it searches for combinations of attributes to characterize data).

Better results are obtained for the vote_r instance, where 3 of the 4 attributes selected by *CFS* are also involved in the unique minimal solution for *MinS*. Moreover, the accuracy is higher (0.7). Nevertheless *CFS* selects only 4 attributes, while 10 are necessary to characterize the instance in the *MinS* approach. Therefore this feature selection technique could not really be used to improve the characterization problem's results.

4) Evaluation of the Attributes for `MinS` *and* `Pattern`*:* In Table V and Table VI, we evaluate the relevance of attributes using scores. We want to assess the ability of an attribute to discriminate observations according to the initial groups. Given an attribute a, the score denoted $score(a)$ is computed by counting the number of observations where the value of this attribute differs in a same group.

$$score(a) = |2 \times (\frac{1}{n}\sum_{j=1}^{n}(g_j - a_j)^2) - 0.5|$$

where:

- a_j is the value of attribute a for observation j,
- $g_j \in \{0,1\}$ is the group assigned to observation j.

Note that an attribute that fully discriminates the groups (i.e., whose values will be always identical or always different for all observations of the two groups) has a score value of 1. Given a set of attributes (for instance representing a *MinS* solution or a *Pattern* solution), its score is the average of the scores of its attributes.

In Table V, we focus on *MinS*:

- The first column is the best possible score obtained by a solution among the set of minimal solutions.
- In the second column, in order to evaluate this best score, we compute the ratio of it with regards to the maximal score that can be obtained when selecting the same number of attributes but using the best attributes with regards to the score function. A ratio of 1 means that the minimal solution is built using only attributes with highest scores.
- The third column indicates the best possible score among attributes that do not appear in any solution (i.e., which are not selected by *MinS*).
- The fourth column indicates the best possible score among the whole set of attributes.

TABLE V
Attributes Scores for *MinS*

Instances	score sol max	ratio	max att no sol	max att
ra100_phv	0.624	0.663	0.96	0.96
ra100_phy	0.549	0.653	0.829	0.905
rch8	0.323	0.35	0.924	0.924
ralsto	0.43	0.602	0.836	0.89
ra_phv	0.648	0.686	0.963	0.963
ra_phy	0.577	0.634	0.929	0.929
ra_rep1	0.217	0.593	0.321	0.518
ra_rep2	0.239	0.636	0.357	0.5
vote_r	0.439	0.7	0.683	0.894

Except on the vote_r instance, we note that the value of the ratio is low, which shows that the scores of the *MinS*

solutions are rather low. These solutions are therefore composed of attributes with low scores. Moreover, the best score of the attributes not appearing in any solution is close to the best score. It means that attributes with a high score do not participate to solutions. This suggests that *MinS* solutions are not built with attributes that are the most correlated to the groups. And again, this scoring system does not allow preprocessing of data and offers information that is different from *MinS*.

In Table VI, we study solutions using *Pattern*. Since the number of attributes is variable, we indicate in the first column the number of attributes in the solution with the best scores between parentheses.

TABLE VI
ATTRIBUTES SCORES FOR *Pattern*

Instances	score sol (#att)	ratio	max att no sol	max att
ra100_phv	0.624(2)	0.663	0.96	0.96
ra100_phy	0.467(4)	0.58	0.829	0.905
rch8	0.515(4)	0.557	0.924	0.924
ralsto	0.333(6)	0.514	0.836	0.89
ra_phv	0.648(2)	0.686	0.963	0.963
ra_phy	0.527(4)	0.581	0.893	0.929
ra_rep1	0.211(26)	0.616	0.339	0.518
ra_rep2	0.158(19)	0.447	0.5	0.5
vote_r	0.468(15)	0.930	0.531	0.894

Similarly to *MinS*, we observe that the scores of the variables are not correlated to the requirements to build a cover by means of patterns.

Experimental Highlights:

- As expected, *MinS* computes the smallest subsets of attributes, which is relevant when attributes are costly to generate (e.g., biological complex routines);
- The proposed *Pattern* algorithm computes the smallest sets of patterns for covering the positive set P, which is useful to observe common characteristics shared by observations; attributes involved in these patterns differ from those selected by *MinS*;
- *MinS* and *Pattern* constitute indeed complementary methods for practitioners who need to better understand and analyze groups of binary data by focusing on different characterizations of the observations;
- Feature selection processes like *CFS* do not provide relevant information with regards to logical characterization or patterns computation.

VII. CONCLUSION

In this paper, we focus on attributes in the general context of logical characterization and analysis of binary data. We have defined new algorithms to generate complete sets of patterns. Our experiments show that patterns computation and computation of minimal solutions for the characterization problem constitute interesting and complementary alternatives to classic statistical based methods for features selection.

REFERENCES

[1] G. Alexe, S. Alexe, D. E Axelrod, T. O Bonates, I. I Lozina, M. Reiss, and P. L Hammer. Breast cancer prognosis by combinatorial analysis of gene expression data. *Breast Cancer Research*, 8(4):1–20, 2006.

[2] G. Alexe, S. Alexe, DE Axelrod, P. L Hammer, and D Weissmann. Logical analysis of diffuse large b-cell lymphomas. *Artificial Intelligence in Medicine*, 34(3):235 – 267, 2005.

[3] G. Alexe, S. Alexe, T. O. Bonates, and A. Kogan. Logical analysis of data - the vision of peter l. hammer. *Ann. Math. Artif. Intell.*, 49(1-4):265–312, 2007.

[4] A. Bennane and S. Yacout. Lad-cbm; new data processing tool for diagnosis and prognosis in condition-based maintenance. *Journal of Intelligent Manufacturing*, 23(2):265–275, 2012.

[5] E. Boros, Y. Crama, P. L. Hammer, T. Ibaraki, A. Kogan, and K. Makino. Logical analysis of data: classification with justification. *Annals OR*, 188(1):33–61, 2011.

[6] E. Boros, P. L. Hammer, T. Ibaraki, and A. Kogan. Logical analysis of numerical data. *Math. Program.*, 79:163–190, 1997.

[7] T. Boureau, M. Kerkoud, F. Chhel, G. Hunault, A. Darrasse, C. Brin, K. Durand, A. Hajri, S. Poussier, C. Manceau, F. Lardeux, F. Saubion, and M.-A. Jacques. A multiplex-pcr assay for identification of the quarantine plant pathogen xanthomonas axonopodis pv. phaseoli. *Journal of Microbiological Methods*, 92(1):42 – 50, 2013.

[8] A. Chambon, T. Boureau, F. Lardeux, F. Saubion, and M. Le Saux. Characterization of multiple groups of data. In *27th IEEE International Conference on Tools with Artificial Intelligence, ICTAI*, pages 1021–1028, 2015.

[9] F. Chhel, F. Lardeux, A. Goëffon, and F. Saubion. Minimum multiple characterization of biological data using partially defined boolean formulas. In *Proceedings of the ACM Symposium on Applied Computing, SAC*, pages 1399–1405, 2012.

[10] I. Chikalov, V. Lozin, I. Lozina, M. Moshkov, H. Nguyen, A. Skowron, and B. Zielosko. Logical analysis of data: Theory, methodology and applications. In *Three Approaches to Data Analysis*, volume 41 of *Intelligent Systems Reference Library*, pages 147–192. Springer Berlin Heidelberg, 2013.

[11] Y. Crama, P. L. Hammer, and T. Ibaraki. Cause-effect relationships and partially defined boolean functions. *Annals of Operations Research*, 16(1):299–325, 1988.

[12] C. Dupuis, M. Gamache, and J.-F. Page. Logical analysis of data for estimating passenger show rates at air canada. *Journal of Air Transport Management*, 18(1):78–81, 2012.

[13] M. A Hall and L. A Smith. Feature subset selection: a correlation based filter approach. 1997.

[14] P. L. Hammer and T. O. Bonates. Logical analysis of data - an overview: From combinatorial optimization to medical applications. *Annals OR*, 148(1):203–225, 2006.

[15] P. L Hammer, A. Kogan, B. Simeone, and S. Szedmák. Pareto-optimal patterns in logical analysis of data. *Discrete Applied Mathematics*, 144(1):79–102, 2004.

[16] J. A Hartigan and M. A Wong. Algorithm as 136: A k-means clustering algorithm. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 28(1):100–108, 1979.

[17] Richard M Karp. Reducibility among combinatorial problems. In *Complexity of computer computations*, pages 85–103. Springer, 1972.

[18] V. Kholodovych, J. R. Smith, D. Knight, S. Abramson, J. Kohn, and W. J. Welsh. Accurate predictions of cellular response using qspr: a feasibility test of rational design of polymeric biomaterials. *Polymer*, 45(22):7367 – 7379, 2004.

[19] J. MacQueen et al. Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, pages 281–297. Oakland, CA, USA., 1967.

[20] K. Makino, K. Hatanaka, and T. Ibaraki. Horn extensions of a partially defined boolean function. *SIAM J. Comput.*, 28(6):2168–2186, 1999.