Correlation Dependencies between Variables in Feature Selection on Boolean Symbolic Objects

Djamal Ziani
College of Business and Administration, Al Yamamah University, Saudi Arabia

Abstract: *Feature selection is an important process in data analysis and data mining. The increasing size, complexity, and multi-valued nature of data necessitate the use of Symbolic Data Analysis (SDA), which utilizes symbolic objects instead of classical tables, for data analysis. The symbolic objects are created by using abstraction or generalization techniques on individuals. They are a representation of concepts or clusters. To improve the description of these objects, and to eliminate incoherencies and over-generalization, using dependencies between variables is crucial in SDA. This study shows how correlation dependencies between variables can be processed on Boolean Symbolic Objects (BSOs) in feature selection. A new feature selection criterion that considers the dependencies between variables, and a method of dealing with computation complexity is also presented.*

Keywords: *Feature selection, dependencies; symbolic data analysis, discrimination criteria.*

Received September 13, 2016; accepted July 23, 2017

1. Introduction

In Symbolic Data Analysis (SDA), Boolean Symbolic Objects (BSOs) are used to represent groups or classes of individuals. The variables used in symbolic objects are multi-valued; they can handle single quantitative values, single categorical values, intervals, and sets of values. The description of a BSO is created via a generalization of the description of all individuals belonging to the class represented by that BSO. However, in general, the generalization introduces incoherence and overgeneralization, which weaken further analysis. Consider the following example of an animal class with two individuals described by three variables: "animal type," "number of wings," and "number of legs." The first individual is a bird, it is represented as (Bird, 2, 2); the second individual is a cat, it is represented as (Feline, 0, 4). The BSO representing the class of animals is described by [Animal type = {Bird, Feline}]$\wedge$[Number of wings = {0, 2}]$\wedge$[Number of legs={2, 4}]. This BSO implies that this class of animals can contain a feline with two wings and four legs, which is completely false. Thus, to be more precise and to avoid this kind of incoherence, we need to add a dependency between the variables "animal type" and "number of wings": if [animal type=feline] then [number of wings=NA] (where NA means not applicable).

This paper addresses feature selection on BSOs with dependencies between variables. Much research has been conducted on the problem of feature selection in classical data analysis, and many algorithms and feature selection criteria have been proposed. However, very little research, such as studies in [11, 12, 14, 16, 21, 22] been conducted in the area of SDA. The dependencies between variables problem has been studied extensively in classical data analysis. Hierarchical dependencies have been introduced [8, 10, 11, 13]; and the dependencies of semantic presence/absence (applicable or not) have also been treated [9, 15, 17]. The dependencies of the correlation semantic are used to indicate how the values taken by some variables influence the values taken by other variables; these dependencies have also been treated by researchers such as [1, 3, 19]. Dependencies in SDA were more recently introduced by [4, 7, 20, 23, 24].

2. Symbolic Objects and Dependencies

2.1. Boolean Symbolic Object

Y={y1,...,yn} is the set of variables. For example, Y={age, weight, illness, …}.

d=(*d*$_1$,...,*d*$_n$) is the description of the object, where $\mathrm{d_i}$ is the value taken by the variable $\mathrm{y_i}$. For example, d = ([20, 25], [80,90], {diabetes, cholesterol}).

$\Omega\{w_1, \ldots, w_p\}$ is the set of elementary observed objects.

$O=\{O_1,\ldots,O_n\}$ is the set of domains where each variable takes its values.

$\Omega' = O_1 \times \ldots \times O_n\}$ is the set of all possible elementary observed objects.

$R=\{R_1,\ldots,R_p\}$ is a set of relations, where R_i is the relation used by the variable y_i. For example, $R_1 =\subseteq$.

A symbolic object is defined as a triplet $s=(a,R,d)$; this explanatory expression defines a symbolic object called an assertion [21]. A BSO is an assertion represented by a symbolic expression:

$$a(w) = \wedge_{i=1,n}[y_i(w)R_i\, d_i] \quad (1)$$

Where $\varLambda$ is the standard logic operator "AND," and w is an elementary observed object.

For example, a(w) =[age(w) =[20, 25]] ∧ [weight(w) = [80, 90]] ∧ [illness(w) = {diabetes, cholesterol}].

An elementary event is represented by the symbolic expression e_i=[y_i=v_i], where $v_i \subset O_i$; it is defined by,

$e_i: \Omega \rightarrow \{true, false\}\ as\ e_i(w) = true \rightarrow y_i(w) \in v_i$.

For example, the elementary event $e_1 = [\text{hair} = \{\text{brown}, \text{black}\}]$ is such that e_1(Alain)= true, because hair(Alain) = brown ∈ {brown, black}.

We distinguish two types of extents: We distinguish two types of extents:

- The real extent of the symbolic object a is defined referring to Ω, and represents the set of elementary observed objects (individuals) that satisfy the expression $ext_\Omega(a) = \{w_l \in \Omega | a(w_l) = true\}$.
- The virtual extent of symbolic object a is defined referring to Ω', and represents the set of virtual elementary objects that satisfy the expression $ext_{\Omega'}(a) = \{w'_l \in \Omega' | a(w'_l) = true\}$; for example, $\forall y_i,\ y_i(w'_l) = v_i\ and\ v_i \in V_i\}$, where v_i is a value taken by variable y_i in object w'_l and v_i is a value taken by variable y_i in assertion.
- *Example* 1: Let Ω={Alain, John, Sam}, age(Alain) =20 and weight(Alain)=85,age(John)=25 and weight(John)=80, age (Sam)=26 and weight (Sam) =86,a(w)=[age(w)=[20, 25]] ∧ [weight(w)=[80, 85]],ext_Ω(a) ={Alain, John}.

Then, $ext_{\Omega'}$(a)={Alain, John and all virtual individuals with age between 20 and 25 and weight between 80 and 85}.

For each BSO, we associate a description space calculated by the function μ that is defined below:

We are given $a_l(w) = [y_1 = v_{l1}] \wedge ... \wedge [y_n = v_{ln}]$, where *w* is an individual.

O^n=O_1×..×O_n is the Cartesian product of all variables, *A* is the set of symbolic objects, a_l is a symbolic object.

We define the function μ as follows:μ

$$\mu\colon A \rightarrow O^n \quad \mu(a_l) = v_{l1} \times \ldots \times v_{ln} \quad (2)$$

- *Example* 2: Given two variables, age, defined in [18, 30], and weight, defined in [75, 85], let the BSO a be defined as a(w) = [age(w) =[20, 25]] ∧ [weight(w) = [80, 85]]; then, the description space of "a" is as shown in Figure 1.

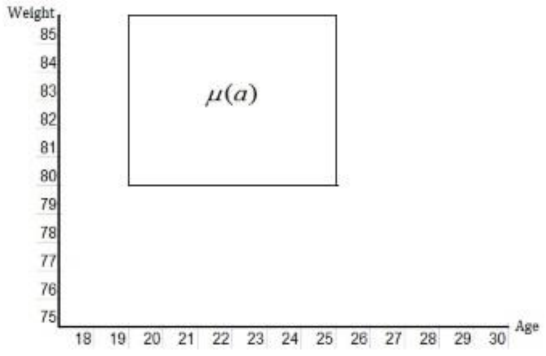


Figure 1. Description space of the object.

2.2. Variable Dependencies

There are two types of dependencies, correlation and hierarchical:

- Correlation Dependency (CD): This type of dependency is used to predict the values taken by some variables when the values taken by other variables are known. For example, [triangle= Equilateral] ⇒ [angle = 60].
- Hierarchical Dependency (HD): This type of dependency is used to predict the inefficacy of some variables, when the values taken by other variables are known.

In this paper, we focus only on CD. (HD will be treated in another paper.)

A dependency between variables is defined as follows:

$$R\colon \{O\} \rightarrow \{O\} \quad \wedge_{i=k,m}[y_i = v_i] \underset{,}{\Rightarrow} \wedge_{j=l,p}[y_j = v_j] \quad (3)$$

To facilitate the writing and processing of dependencies, we use only the AND operator (∧) in the premises and conclusions of the dependencies. This is not a simplification of the problem, but only conversion of the dependencies using the properties of logical operators. Thus, a dependency that uses the OR operator (∨) is replaced by a set of dependencies using only the AND operator (∧). When OR operators are used in the premise of the dependency, we use the following rules to change the dependencies: When the OR operators are used in the premise of the dependency, a dependency of the form $\vee_{i=k,m}[y_i = v_i] \underset{,}{\Rightarrow} Q$ will be replaced by a set of *m* dependencies:

$$\left\{[y_1 = v_1] \Rightarrow Q, \ldots, [y_m = v_m] \underset{,}{\Rightarrow} Q\right\} \quad (4)$$

- A dependency of the form $\vee_{i=k,m}[y_i = v_i] \underset{,}{\Rightarrow} Q$ is replaced by a set of m dependencies:

$$\left\{P \underset{,}{\Rightarrow} [y_1 = v_1], \ldots, P \underset{,}{\Rightarrow} [y_m = v_m]\right\} \quad (5)$$

The real extent of a dependency is the set of individuals that verifies the condition of this dependency. The dependency can be written in the form $R\colon P \Longrightarrow Q$, thus *R*(*w*)=*true iff P*(*w*)=*false or Q*(*w*)= *true*. Consequently, we calculate the real extent of a dependency *R*, as follows:

$$ext_{\Omega}(R) = \left(\Omega \setminus ext_{\Omega}(P)\right) \cup ext_{\Omega}(Q) \tag{6}$$

The virtual extent of a dependency is the set of virtual elementary objects that verifies the condition of this dependency. We calculate the virtual extent of dependency R as follows:

$$\mathrm{ext}_{\Omega'}(\mathrm{R}) = \left(\Omega' \setminus \mathrm{ext}_{\Omega'}(\mathrm{P})\right) \cup \mathrm{ext}_{\Omega'}(\mathrm{Q}) \tag{7}$$

We define the description space of a variable dependency as follows below. Given a CD $R: P \Longrightarrow Q$, where $\mathcal{D}$ is the set of dependencies, we define the function μ as follows:

$$\mu: \mathcal{D} \rightarrow \mathrm{O}^{n} \qquad \mu(\mathrm{R}) = \mu(\neg\mathrm{P}) \cup \mu(\mathrm{P} \wedge \mathrm{Q}), \tag{8}$$

Where $\mu(\neg\mathrm{P}) = \Omega' \setminus \mu(\mathrm{P})$.

- *Example* 3: Given a dependency (CD) R_1 = [x = [3, 5]] ⇒ [y = [2, 4]], and O = [1, 7] × [1, 7], the description space of R_1 is as shown in Figure 2.

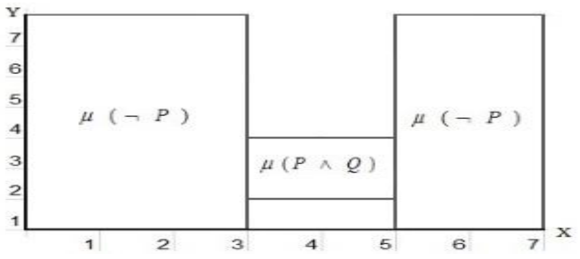


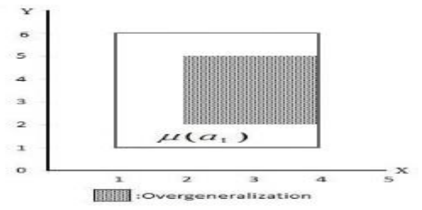
Figure 2. Description space of a CD.

2.3. Description Space of A Symbolic Object Verifying Dependencies

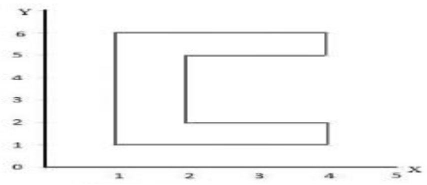
A BSO usually represents an individual cluster, with its description a generalization of the description of all individuals belonging to that cluster. The monothetic descriptive nature of BSOs often results in overgeneralization. This overgeneralization can be reduced and the quality of the description of the clusters consequently improved by inserting dependencies between variables in the dataset. The following example illustrates how dependencies between variables can help to solve the overgeneralization problem.

- *Example* 4: We are given two variables X (defined in [0, 5]) and Y (defined in [0, 6]) and a cluster C_1, described by the BSO a_1. Cluster C_1 represents all individuals where, when X is in [1, 2], Y is in [1, 5], and when X is in [2, 4], Y is in [1, 2] or in [5, 6]. Thus, $a_1 = [\,X\, = [0,\ 3]] \wedge [Y\ = [0,\ 5]\,]$.

Figure 3-a shows the description space of BSO a_1. It can be seen that a significant portion of the space is overgeneralized. We can solve the overgeneralization problem by adding a dependency (CD) $R_1 = [\,X\, = [2,\ 4]] \Rightarrow [Y\ = [1,\ 2],\ [5,6]\,]$, as shown in Figure 3-b.



a) Description space of a1.



b) Description space of a1 taking into account the dependency.

Figure 3. Object Description Space and Dependencies.

An individual w that belongs to the extent of a BSO "a", and verifies a dependency R:P $R: P \Rightarrow Q$, should verify this condition: $a(w)$=*true* and $R(w)$=*true*. Thus, the description space of a BSO "a" taking into consideration a dependency R is:$\mu(a\ \wedge R)$. The following formula will be used in order to calculate with less complex operations the description space of a BSO taking into consideration a dependency:

$$\mu(a\ \wedge R) = \mu\left(a\ \wedge\ \neg(a\ \wedge P \wedge \neg Q)\right) = \mu(a\,) \setminus \mu(a\ \wedge P \wedge \neg Q\,) \tag{9}$$

3. Processing Dependencies between Variables

3.1. Different Ways for Processing Dependencies

Various methods have been proposed to process the dependencies between variables. We will use the same notations for all measures.

Let v_{ik} and v_{jk} be the values taken by variable y_k in assertions a_i and a_j.

- *Weighting trend*: The idea in this trend is to give weighting for variables when they are involved in a dependency.
- Klin and Sassone in [13] proposed to label the implication by using the total weigh of the variables in the premise. This weighting is the basis of how the dependencies will be executed.
- Vignes in [20] uses a binary weighting in the calculation of the dissimilarity. Vignes calculates the dissimilarity between two BSOs as follows:

$$d(a_i, a_j) = \sum_{k=1}^{n} comp(v_{ik}, v_{jk}) \times \delta(v_{ik}, v_{jk}),$$
$$\text{where } comp(v_{ik}, v_{jk}) = \begin{cases} 1\ if\ v_{ik} \cap v_{jk} = \emptyset, \\ 0\ else \end{cases} \tag{10}$$
$$\text{and } \delta(v_{ik}, v_{jk}) = \begin{cases} 1\ if\ v_{ik} \neq NA\ and\ v_{jk} \neq NA \\ 0 \qquad else. \end{cases}.$$

- *Counting trend*: A number of researchers use codification and counting systems to calculate similarity and dissimilarity measures:
- Gower in [9] defines a dissimilarity between two objects a_i and a_j, described by n variables, as

follows:

$$d_{ij} = \sum_{l=1}^{n} d_{ijk} / \sum_{l=1}^{n} \delta_{ijk}, \tag{11}$$

Where $d_{ijk} = \begin{cases} 1 \ \ if\ v_{ik} = v_{jk}, \\ 0\ else. \end{cases}$ and

$$\delta_{ijk} = \begin{cases} 1\ if\ the\ variable\ k\ takes\ values\ in\ a_i\ and\ \ a_j \\ 0 \qquad\qquad else. \end{cases}$$

- *Dependency structure*: Ben-Bassat in [1] used segmentation tree to process hierarchical logical dependencies. Vignes in [20] used graphs of dependencies, a structure more complex than the tree used by Ben-Bassat, because in a graph a variable can be a premise in many dependencies. De Carvalho in [4] used graphs of dependencies to process hierarchical dependencies and correlation dependencies.
- *Decomposition trend*: In this trend, the dependencies are used to decompose the description space of the BSOs in a set of many sub-space parts, in which each one should describe a symbolic object that either satisfies all variable dependencies or does not [4, 5]. Given a BSO $a = \wedge_{i=1,n} e_i$, with e_i=[y_i=v_i], a set of dependencies R={R_1,...,R_p}, and a function π, which calculates the length of an object, the decomposition of dependencies leads to an exponential computation time value depending on the number of rules, as shown in the following:

Let have a BSO $a = \wedge_{i=1,n} e_i$, with e_i=[y_i=v_i]. Let have a set of dependencies: R={R_1,...,R_p}. And let use a function π, which calculate the length of an object.

$$\begin{aligned} d(a/R_1 \wedge \ldots \wedge R_t\ ) = & \prod_{i=1}^{n} \pi(e_i) \\ & - \sum_{j=1}^{t} \pi(a\ \wedge \neg R_j) \\ & + \sum_{j<k} \pi\left((a\ \wedge \neg R_j) \wedge \ldots \wedge \neg R_k\right) + \cdots \\ & +\ (-1)^{t+1} \pi\left((a\ \wedge \neg R_1) \wedge \neg R_2 \ldots \wedge \neg R_t\right). \end{aligned}$$

- Normal Symbolic Form (NSF): In this case, the idea is to represent the data in a manner that only coherent descriptions are represented, and the dependencies are not needed in the process. To accomplish this, the original tables, representing the symbolic objects, are decomposed into several tables, according to the number of different premise variables. The variables not associated with the dependencies remain in the original table and so the new tables contain variables on which the dependencies are applied [2].

The idea of NSF has been used by De Carvalho [6] to cluster symbolic objects constrained by a set of dependencies. To accomplish this, a graph of dependencies was used to generate NSFs, and then a dissimilarity measure was applied to the generated NSFs. The complexity involved in processing the dependencies using NSF is polynomial. However, decomposing the main symbolic tables into many tables completely changes the design of the symbolic object database, from the star schema to a normalized database.

3.2. Processing Dependencies Using Description Space

In this section, we present a new method that utilizes only the calculation of the description spaces to process the dependencies between variables. This method is inspired by research conducted on decomposition trend and in NSF; however, the symbolic objects are not decomposed. We propose three optimization methods to reduce the complexity: optimization of the description space calculation, optimization of the dissimilarity measure calculation, and optimization of the feature selection algorithm.

We calculate the description space of a BSO, a, given a set of dependencies $\mathcal{D} = \{R_1, \ldots, R_t\}$, by removing from the description space of BSO a, the part of the description space of BSO a that does not verify the dependencies. This is achieved by using the function Original Discriminant Power (ODP), which is defined as follows:

$$\mu\left(a\ \wedge\ (R_1 \wedge \ldots \wedge R_t)\right) = \mu(a\ ) \setminus \left(\mu(a\ \wedge \neg R_1\ ) \cup \ldots \mu(a\ \wedge \neg R_t)\right) \tag{13}$$

Where

$$\mu(a\ \wedge \neg R_k\ ) = \mu(a\ )\ \setminus \mu(a\ \wedge P_k \wedge \neg Q_k\ ) \tag{14}$$

Before discussing the complexity of Equations (13) and (11), let us look at the mathematical properties of function μ:

1. $\mu(A\ \wedge B\ ) = \mu(A)\ \cap \mu(B)$
2. $\mu(A \vee B\ ) = \mu(A)\ \cup \mu(B)$
3. $\mu(\neg A\ ) = \mu(A)^c = O \setminus \mu(A)$
4. The description space of an object not described by all variables is calculated by using as value, for each absent variable, the domain of the variable. For instance, if an object a is described by the first d variables, among a total of n variables, then the description space of a will be:

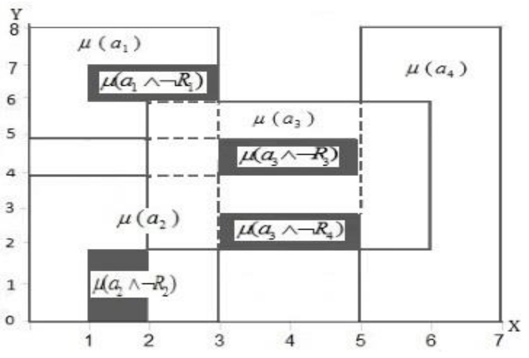
$$\mu\left(a = \bigwedge\nolimits_{i=1,d} [y_i = v_i]\ \right) = \ v_1 \times \ldots \times v_d \times o_{d+1} \times \ldots \times o_n \tag{15}$$

Here, v_i is a value taken by variable y_i in assertion a, and O_d is the domain of variable y_d.

5. Let have two BSOs: $\mathrm{a}_i = \wedge_{k=1,n}[y_i = v_{ik}]$ and $\mathrm{a}_j = \wedge_{k=1,n}[y_i = v_{jk}]$:

- If $\forall \mathrm{k} \in \{1,..,\mathrm{n}\}, \mathrm{v}_{\mathrm{jk}} \subseteq \mathrm{v}_{\mathrm{ik}}$ then $\mu(\mathrm{a}_\mathrm{i}\ \wedge \mathrm{a}_\mathrm{j}\ ) = \mu(\mathrm{a}_\mathrm{j})$.
- If $\forall \mathrm{k} \in \{1,..,\mathrm{n}\}, \mathrm{v}_{\mathrm{jk}} \subseteq \mathrm{v}_{\mathrm{ik}}$ then $\mu(\mathrm{a}_\mathrm{i}\ \vee \mathrm{a}_\mathrm{j}\ ) = \mu(\mathrm{a}_\mathrm{i})$.
- If $\exists \mathrm{k} \in \{1,..,\mathrm{n}\}, \mathrm{v}_{\mathrm{jk}} \cap \mathrm{v}_{\mathrm{ik}} = \emptyset$ then: $\mu(\mathrm{a}_\mathrm{i}\ \wedge \mathrm{a}_\mathrm{j}\ ) = \emptyset$.

Theoretically, calculating the description space of a BSO, given a set of dependencies is complex, as seen in Equation (13). It is based on the calculation of the union parts of description spaces. The part of a description space used in calculations, representing the space where the BSO does not verify the dependency, is also a complex operation, as seen in Equation (14). This is because it is using a subtraction operation between description spaces. However, in practice, a dependency is usually created to be used by one or only some objects of the dataset (see Figure 4).



Y
X
$\mu(a_1)$
$\mu(a_1 \wedge \neg R_1)$
$\mu(a_4)$
$\mu(a_3)$
$\mu(a_3 \wedge \neg R_3)$
$\mu(a_2)$
$\mu(a_3 \wedge \neg R_4)$
$\mu(a_2 \wedge \neg R_2)$
0
1
2
3
4
5
6
7
8
1
2
3
4
5
6
7

Figure 4. Dependencies in a whole object dataset.

This means that the calculation will be reduced to only the description space where the object is concerned. This check can be done using property 5.

- *Example* 5: This example shows how the description spaces, displayed in Figure 4, are calculated using Equation (13):

$$\mu\left(a_1 \wedge \left(\wedge_{i=1}^{4} R_i\right)\right) = \mu(a_1) \setminus \left(\cup_{i=1}^{4} \mu(a_1 \wedge \neg R_i)\right)$$
$$= \mu(a_1) \setminus (\mu(a_1 \wedge \neg R_1) \cup \emptyset \cup \emptyset \cup \emptyset)$$
$$= \mu(a_1) \setminus \mu(a_1 \wedge \neg R_1)$$
$$\mu\left(a_2 \wedge \left(\wedge_{i=1}^{4} R_i\right)\right) = \mu(a_2) \setminus \left(\cup_{i=1}^{4} \mu(a_2 \wedge \neg R_i)\right)$$
$$= \mu(a_2) \setminus (\emptyset \cup \mu(a_2 \wedge \neg R_2) \cup \emptyset \cup \emptyset)$$
$$= \mu(a_2) \setminus \mu(a_2 \wedge \neg R_2)$$
$$\mu\left(a_3 \wedge \left(\wedge_{i=1}^{4} R_i\right)\right) = \mu(a_3) \setminus \left(\cup_{i=1}^{4} \mu(a_3 \wedge \neg R_i)\right)$$
$$= \mu(a_3) \setminus (\emptyset \cup \emptyset \cup \mu(a_3 \wedge \neg R_3) \cup \mu(a_3 \wedge \neg R_4))$$
$$= \mu(a_3) \setminus \left(\mu(a_3 \wedge \neg R_3) \cup \mu(a_3 \wedge \neg R_4)\right)$$
$$\mu\left(a_4 \wedge \left(\wedge_{i=1}^{4} R_i\right)\right) = \mu(a_4) \setminus \left(\cup_{i=1}^{4} \mu(a_4 \wedge \neg R_i)\right)$$
$$= \mu(a_4) \setminus (\emptyset \cup \emptyset \cup \emptyset \cup \emptyset) = \mu(a_4)$$

4. Dissimilarity Measure and Dependencies Between Variables

4.1. Different Ways For Processing Dependencies

The feature selection criterion, which will be used in the algorithm, is based on a new dissimilarity measure between BSOs. To define this dissimilarity measure, we introduce a function π that calculates the potential of a description space:

$$\pi: O \to \mathcal{R}^{+} \qquad \pi(E_r) = \prod_{i=1}^{n} cap(v_{ri}), \tag{16}$$

Where $E_r = \prod_{i=1}^{n} v_{ri}$ and the *cap* function gives the length of a value.

The dissimilarity measure between two BSOs, a_i and a_i , is defined as follows:

$$d(a_i, a_j) = 1 - \frac{\pi\left(\mu(a_i \wedge a_j)\right)}{\pi\left(\mu(a_i \vee a_j)\right)} \tag{17}$$

The dissimilarity measure between two BSOs, a_i and a_j, taking into consideration a set of dependencies, $\{R_1, \ldots, R_t\}$, is defined as follows:

$$d(a_i, a_j) = 1 - \frac{\pi\left(\mu(a_i \wedge a_j \wedge_{p=1}^{t} R_p)\right)}{\pi\left(\mu\left((a_i \vee a_j) \wedge_{p=1}^{t} R_p\right)\right)} \tag{18}$$

This dissimilarity measure is reflexive and symmetric, and respects the triangle inequality. However, if Equation (12) is used the dissimilarity calculation is complex. This is why we optimize Equation (11) using the following property of function π:

$$\pi\left(\mu(a_i \vee a_j)\right) = \pi(\mu(a_i)) + \pi\left(\mu(a_j)\right) - \pi\left(\mu(a_i \wedge a_j)\right) \tag{19}$$

Thus, the dissimilarity measure between two BSOs taking into consideration a set of dependencies is practically calculated using:

$$d(a_i, a_j) = 1 - \frac{\pi\left(\mu(a_i \wedge a_j \wedge_{p=1}^{t} R_p)\right)}{\pi\left(\mu\left((a_i) \wedge_{p=1}^{t} R_p\right)\right) + \pi\left(\mu\left((a_j) \wedge_{p=1}^{t} R_p\right)\right) - \pi\left(\mu\left((a_i \wedge a_j) \wedge_{p=1}^{t} R_p\right)\right)} \tag{20}$$

Calculating the dissimilarity using Equation (13) is very important in order to reduce the complexity. Note that Equation (15) does not utilize $\pi\left(\mu\left((a_i \vee a_j) \wedge_{p=1}^{t} R_p\right)\right)$, which is a complex calculation, called the union part. Instead, the union part is replaced by the potential of object description space, and the following part. $\pi\left(\mu\left((a_i \vee a_j) \wedge_{p=1}^{t} R_p\right)\right)$, called the intersection part, which is the same as the numerator, is subtracted.

Moreover, to avoid calculating the same thing many times during the process of feature selection, we propose to save for each object a_i its potential of description space, calculated by $\pi\left(\mu(a_i \wedge a_j \wedge_{p=1}^{t} R_p)\right)$, and for each couple (a_i, a_j), we save the intersection part.

4.2. Dissimilarity Measure Between Two Elementary Events

The dissimilarity measure between two elementary events is calculated by quantifying the contribution of a variable in the discrimination between two objects. This contribution can be calculated by decomposing the description spaces of the two objects into many areas, depending on whether the variable contributes or not in the discrimination of each area. In the case where the variable contributes in the discrimination, the area is the space where the variable takes values that are not common to the two objects. This can be calculated using the following:

Given $a_i = \wedge_{l=1,n}[y_l = v_{il}]$ and $a_j = \wedge_{l=1,n}[y_l = v_{jl}]$, set $z_{ij}^{l} = O_l \setminus (v_{il} \cap v_{jl})$: This is the set of values that can be used by variable y_l to contribute to the dissimilarity between e_{il} and e_{jl}. Mathematically, it is the projection, on the axis of variable y_l, of the area where variable y_l contributes to the dissimilarity.

We define the description space of an object knowing the value taken by a variable as:

$$\mu(a_i/\, y_l = v) = v_1 \times \dots \times v_{l-1} \times v \times v_{l+1} \dots \times v_n \quad (21)$$

$$d(e_{il}, e_{jl}) = \frac{\pi\left(\mu(a_i \,/\, y_l = v_{il} \cap z^l_{ij}\,)\right) + \pi\left(\mu(a_j \,/\, y_l = v_{jl} \cap z^l_{ij}\,)\right)}{\pi(\mu(a_i)) + \pi\left(\mu(a_j)\right) - \pi\left(\mu(a_i \wedge a_j\,)\right)} \quad (22)$$

- *Property* 1

$$\pi\left(\mu(a_i \,/\, y_l = v\,)\right) = \pi\left(\mu(a_i\,)\right) \times \frac{cap(v)}{cap(v_{il})} \quad (23)$$

By taking into consideration the dependencies between variables, the dissimilarity between two elementary events will be calculated as follows:

$$d(e_{il}, e_{jl}) = \frac{\pi\left(\mu\left(a_i \wedge_{p=1}^{t} R_p /\, y_l = v_{il} \cap z^l_{ij}\right)\right) + \pi\left(\mu\left(a_j \wedge_{p=1}^{t} R_p /\, y_l = v_{jl} \cap z^l_{ij}\right)\right)}{\pi\left(\mu(a_i \wedge_{p=1}^{t} R_p)\right) + \pi\left(\mu(a_j \wedge_{p=1}^{t} R_p)\right) - \pi\left(\mu(a_i \wedge a_j \wedge_{p=1}^{t} R_p)\right)} \quad (24)$$

Knowing that,

$\pi\left(\mu(a_i \wedge_{p=1}^{t} R_p)\right) = \pi(\mu(a_i)) - \pi(\cup_{p=1}^{t} \mu(a_i \wedge \neg R_p\,)),$(14)

we will be calculated as follows:

$$d(e_{il}, e_{jl}) = \frac{\sum_{k=i,j} \pi\left(\mu\left(a_k /\, y_l = v_{kl} \cap z^l_{ij}\right)\right) - \pi\left(\cup_{p=1}^{t} \mu\left(\left(a_k /\, y_l = v_{kl} \cap z^l_{ij}\right) \wedge \neg R_p\,\right)\right)}{\sum_{k=i,j} \pi\left(\mu(a_k)\right) - \pi\left(\cup_{p=1}^{t} \mu(a_k \wedge \neg R_p)\right) - \pi\left(\mu\left(a_i \wedge a_j\,\right)\right) + \pi\left(\cup_{p=1}^{t} \mu\left(a_i \wedge a_j \wedge \neg R_p\right)\right)} \quad (25)$$

- *Example* 6: Let us consider the following dataset:

We have two variables: *X* defined in [0, 8] and *Y* defined in [0, 9]).

We have two BSOs: a_1=[X= [1, 6]] ∧ [Y= [3, 9]] and a_2=[X= [3, 8]] ∧ [Y= [1, 7]].

We also have three dependencies:

1. R_1 = [Y=[8, 9]] ⇒ [X={[1, 2],[3, 6]}].
2. R_2 = [Y=[2, 3]] ⇒ [X={[3, 5],[7, 8]}].
3. R_3 = [Y=[5, 6]] ⇒ [X={[1, 4],[6, 8]}].

Figure 5 represents the description space of all the dataset objects.

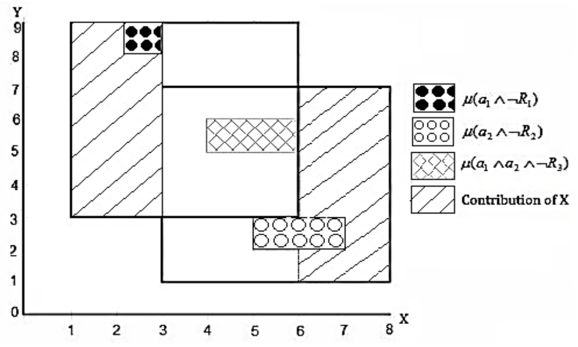


Figure 5. Description space of all objects in the dataset.

We wish to calculate *d*(e_{11},e_{21}), which represents the dissimilarity measure of the elementary events of objects a_1 and a_2, which use the first variable "*X.*"

We wish to calculate *d*(e_{11},e_{21}), which represents the dissimilarity measure of the elementary events of objects a_1 and a_2, which use the first variable "*X.*"

$z^1_{12} = [0,.8] \setminus [3,\ 6] = [0,3[\cup]6,8]$

$\pi\left(\mu(a_1 /\, X = [1,6] \cap z^1_{12})\right) = \pi([1,3] \times [3,9]) = 12$

$\pi\left(\mu(a_2 /\, X = [3,8] \cap z^1_{12})\right) = \pi([6,8] \times [1,7]) = 12$

$\neg R_1 = [X =]2,3[] \ \wedge \ [Y = [8,9]].$

$\neg R_2 = [X =]5,7[] \ \wedge \ [Y = [2,3]].$

$\neg R_3 = [X =]4,6[] \ \wedge \ [Y = [5,6]].$

$\pi\left(\cup_{p=1}^{3} \mu\left((a_1 /\, X = v_{11} \cap z^1_{12}) \ \wedge \neg R_p\right)\right) =$

$\pi(\,(]2,3[\times [8,9]) \cup \emptyset \cup \emptyset) = 1$

$\pi\left(\cup_{p=1}^{3} \mu(a_2 \wedge \neg R_p)\right) = \pi(\emptyset \cup ([6,7[\times [2,3]) \cup \emptyset) = 1$

(15)

$\pi\left(\mu\left(a_1 \wedge a_2 \wedge_{p=1}^{t} R_p\right)\right) = \pi\left(\mu(a_1 \wedge a_2\,)\right) - \pi\left(\mu\left(\cup_{p=1}^{3} a_1 \wedge a_2 \ \wedge \neg R_p\right)\right)$

$\pi\left(\mu\left(a_1 \wedge a_2 \wedge_{p=1}^{t} R_p\right)\right) = \pi([3,6] \times [3,7]) - \pi(]4,6[\times [5,6]) = 10$

$\pi\left(\mu(a_1)\right) = \pi([1,6] \times [3,9]) = 30$

$\pi\left(\mu(a_2)\right) = \pi([3,8] \times [1,7]) = 30$

$d(e_{11}, e_{21}) = \frac{12+12-1-1}{30+30-3-4-10} = 0.51.$

5. Feature Selection Algorithm

5.1. Selection Criteria

We previously developed an algorithm called *Minset-Plus* [21, 22]. This algorithm requires two criteria: the discriminant power, and the ODP.

Given a set of objects, *A*={a_1,…,a_m}, let *Y*={y_1, …, y_n} be a set of variables, *K* the set of object pairs *K*=*A*×*A*, *P*(*Y*) the set of all subsets of *Y*, and *P*(*K*) the set of all subsets of *K*.

- Discriminant Power (DP): DP is used as stopping criteria. The DP of a subset of variables Yd on the set K, noted by *DP*(*Yd*,*K*), calculates the maximum discrimination reached by the subset of variables. It uses the discrimination measure between elementary events:

$$DP: P(Y) \times P(K) \longrightarrow \mathcal{R}^+$$

$$DP(Yd, K) = \sum_{i=1}^{m-1} \sum_{j=i+1}^{m} \max_{y_l \in Yd} d(e_{il}, e_{jl}) \quad (26)$$

- Original Discriminant Power (ODP): ODP is the selecting criteria of the algorithm. The *ODP* of a variable y_1 referred to a set of variable *Yd*, quantifies how much variable y_1 contributes to discriminate the assertions pairs that are not discriminated by any variable of *Yd*.

$$ODP: Y \times P(Y) \times P(K) \longrightarrow \mathcal{R}^+$$

$$ODP(y_l, Yd, K) = \sum_{i=1}^{m-1} \sum_{j=i+1}^{m} \max_{(a_i,a_j) \in K} \left(d(e_{il}, e_{jl}) - \max_{y_p \in Yd}\left(d(e_{ip}, e_{jp})\right), 0\right) \quad (27)$$

5.2. Minset-Plus Algorithm

There is no change done in Minset-Plus algorithm; only the method used to calculate the selection criteria is changed.

The steps used in the algorithm are as follows:

1. Find the indispensible variables which permit us to discriminate assertion pairs not discriminated by others variables. This means we select the variables such that their ODP against all other variables is ≠0: ODP(y_i,Y-y_i, K)≠0. Set Y'=Y Set Yd=set of selected variables While DP(Yd,K) < DP(Y, K).
2. Select in each step the variable which has the highest ODP. The selected variable permits to discriminate the greatest number of assertion pairs, not already discriminated by the variables selected before.

Y'= Y' - Selected variables

$\mathrm{Yd} = \mathrm{Yd} \cup \{ y_l \,/\, y_l \text{ maximizes } \mathrm{ODP}(y_i, Y' - y_i, K) \;\; \forall y_i \in Y' \}$.

3. Eliminate in each step the variables which become redundant. This means the assertion pairs discriminated by these variables are discriminated by other selected variables.

$Yd = Yd - \{y_l \in Yd \text{ } where \text{ } ODP(y_l, Yd - y_l, K) = 0\}$

This algorithm is summarized as follows: Let Y and Y' represent two sets of variables, O and O$'$ represent the values taken by the variables of Y and Y', respectively, and K represent the set of assertion pairs. Initially, we have a knowledgebase (*Y,O,A*). The objective of the algorithm is to extract a knowledgebase (Y', O', A) such that $Y' \subseteq Y$ with$DP(Y', K) = DP(Y, K)$.

5.3. Algorithm Optimization

Because the symbolic objects are complex data, and that the processing of dependencies needs heavy operations, we reduce the complexity of the algorithm by saving intermediate results during calculation of the dissimilarity measure between two elementary events, and by avoiding repetitive calculations.

5.3.1. Complexity in The Calculation of Dissimilarity Measures Between Two Elementary Events

The dissimilarity measure between two elementary events is calculated using Equation (25):

$$d(e_{il}, e_{jl}) = \frac{\sum_{k=i,j} \pi\left(\mu\left(a_k/\, y_l = v_{kl} \cap z^l_{ij}\right)\right) - \pi\left(\bigcup_{p=1}^{t} \mu\left(\left(a_k/\, y_l = v_{kl} \cap z^l_{ij}\right) \wedge \neg R_p\right)\right)}{\sum_{k=i,j} \pi(\mu(a_k)) - \pi\left(\bigcup_{p=1}^{t} \mu(a_k \wedge \neg R_p)\right) - \pi\left(\mu(a_i \wedge a_j)\right) + \pi\left(\bigcup_{p=1}^{t} \mu(a_i \wedge a_j \wedge \neg R_p)\right)} \quad (28)$$

1. The denominator of this formula gives two important clues for optimizing the calculation:

- The denominator is the same for all elementary events of the same two objects. This means that if we have two objects, a_i and a_j, described by n variables, the $\frac{n(n-1)}{2}$ dissimilarity measures between two elementary events involved with the objects a_i and a_j use the same denominator.
- The denominator also uses calculation components that are used by other measures. If we have in our dataset m objects, the components $\pi(\mu(a_k))$ and $\pi\left(\bigcup_{p=1}^{t} \mu(a_k \wedge \neg R_p)\right)$ are used by the $\frac{m(m-1)}{2}$ dissimilarity measures between two elementary events involved with these m objects.

On the basis of these two observations, we save the calculation components in a matrix as follows:

Table 1. Calculation component matrix.

Object	Calculation Component
a_1	$\pi(\mu(a_1)) - \pi\left(\bigcup_{p=1}^{t} \mu(a_1 \wedge \neg R_p)\right)$
a_2	$\pi(\mu(a_2)) - \pi\left(\bigcup_{p=1}^{t} \mu(a_2 \wedge \neg R_p)\right)$
…	…
a_m	$\pi(\mu(a_m)) - \pi\left(\bigcup_{p=1}^{t} \mu(a_m \wedge \neg R_p)\right)$

2. In the numerator $d(e_{il}, e_{jl})$, and using property 1, the calculation of $\pi\left(\mu\left(a_k/\, y_l = v_{kl} \cap z^l_{ij}\right)\right)$ and $\pi\left(\bigcup_{p=1}^{t} \mu\left(\left(a_k/\, y_l = v_{kl} \cap z^l_{ij}\right) \wedge \neg R_p\right)\right)$ are carried out in a manner to use, respectively, $\pi(\mu(a_k))$ and $\pi\left(\bigcup_{p=1}^{t} \mu(a_i \wedge \neg R_p)\right)$, which have already been calculated in the denominator. Thus, the dissimilarity measure between two elementary events is calculated with Equation (29):

$$d(e_{il}, e_{jl}) = \frac{cap\left(v_{kl} \cap z^l_{ij}\right)}{cap(v_{kl})} \times \frac{\sum_{k=i,j} \pi(\mu(a_k)) - \sum_{k=i,j} \pi\left(\bigcup_{p=1}^{t} \mu(a_k \wedge \neg R_p)\right)}{D}, \quad (29)$$

Where

$$D = \sum_{k=i,j} \pi(\mu(a_k)) - \pi\left(\bigcup_{p=1}^{t} \mu(a_k \wedge \neg R_p)\right) - \pi\left(\mu(a_i \wedge a_j)\right) + \pi\left(\bigcup_{p=1}^{t} \mu(a_i \wedge a_j \wedge \neg R_p)\right) \quad (30)$$

The potential of a union of description space is a combinatory calculation:

$$\pi\left(\bigcup_{p=1}^{t} \mu(E_p)\right) = \sum_{p=1}^{t} \pi\left(\mu(E_p)\right).$$

$$+(-1)^1 \sum_{p=1}^{t-1} \pi\left(\mu(E_p) \cap \mu(E_{p+1})\right)$$

$$+\ldots+(-1)^{t-1} \pi\left(\mu(E_1) \cap \mu(E_2) \ldots \mu(E_t)\right).$$

This kind of calculation appears in Equation (29), in $\pi\left(\bigcup_{p=1}^{t} \mu(a_j \wedge \neg R_p)\right)$. The calculation is carried out by the union of the intersections of a description space with a negation of a dependency. To reduce the complexity, the calculation is carried out by level. In this way, we avoid the calculation of an intersection in level *p* if there is no intersection in level *p*-1. The following calculation matrix is used for this purpose.

Table 2. Calculation matrix.

	Level 1			**Level 2**			**Level 3**
	¬R_1	**¬R_2**	**¬R_3**	**¬R_1∧¬R_2**	**¬R_1∧¬R_3**	**¬R_2∧¬R_3**	**¬R_1∧¬R_2∧¬R_3**
E_1	0	5	7	X	X	2	X
E_2	5	6	0	3	2	X	X
E_3	0	0	5	X	X	X	X
E_4	8	4	0	4	2	X	X
E_5	5	0	0	3	X	X	X

X: means that the calculation is not done, and the result is equal to zero.

5.3.2. Using Dissimilarity Matrix

The discrimination matrix allows us to calculate $d_p(e_{li},e_{lk})$ only once, and in all the steps of the algorithm, the matrix is used to carry out all the necessary operations. This significantly optimizes the temporary complexity. In addition, at only $k \times n$, the matrix is not large; k=*card*(K) and n=*card*(Y), K is not a large number because we are dealing with classes of individuals. Example 8 illustrates how the algorithm uses the dissimilarity matrix.

- *Example* 7: Given Y ={y1, y2, y3, y4, y5} is the set of variables and A={a1, a2, a3, a4}, then K={(a1, a2), (a1, a3), (a1, a4), (a2, a3), (a2, a4), (a3, a4)}, see Table 3.

When we calculate *DP(Y, K)* for the stopping criteria of the algorithm, we fill the discrimination matrix (only one time). Thus, in the case corresponding to y_l and (a_i, a_j) we put the result of $d_p(e_{li},e_{lk})$. Then the *Max* Y_d is used to save: $\max_{y_p \in Yd}(d_p(e_{pi}, e_{pk}))$. This means, at the beginning the maximum of the result of $d_{pi}(e_{pi},e_{pk})$ is empty for the selected variables; and then we put the maximum for the indispensable variables. Here in this example, y_l is indispensable.

Table 3. Discrimination matrix.

	(a1, a2)	(a1, a3)	(a1, a4)	(a2, a3)	(a2, a4)	(a3, a4)
y1	0.7	0	0.3	0.1	0	0.1
y2	0	0.6	0.1	0.7	0	0.4
y3	0	0.6	0.5	0.3	0.6	0.3
y4	0	0.2	0.4	0.2	0.5	0.5
y5	0	0.3	0.4	0.3	0.6	0.3
Max Yd	0.7	0	0.3	0.1	0	0.1

Selection of a new variable in each step is accomplished by calculating *ODP*(y_l,*yd*,*K*) for each unselected variable. Using the discrimination matrix, selection of the new variable is achieved via the following operations:

- $d(e_{pi},e_{pk})$ is saved in the case corresponding to y1 and (a_i, a_j) of the discrimination matrix.
- $\max_{y_p \in Yd}\left(d_p(e_{pi}, e_{pk})\right)$ is saved in the *Max* Y_d row.

Thus, calculation of *ODP*(y_l,*yd*,*K*) is achieved via only one number subtraction operation and a comparison operation to find the maximum of two numbers.

- Finding the redundant variables in each step is achieved by checking each variable among the set of selected variables, to determine if redundancy will occur when the new selected variable is added. This means that we calculate the expression $ODP(y_l, (Y_d \cup y_s) - y_l, K) = 0$, where y_s is the new selected variable. The property defined in [22] enables calculation of the *ODP* based on $DP{:}ODP(y_i, Y_P, K) = DP(Y_P \cup y_i, K) - DP(Y_P, K)$.
  We check the redundancy as follows:

$$DP\big((Y_d \cup y_s) - (y_l - y_s), K\big) = DP(Y_d \cup y_s, K) \quad (32)$$

This test is not complex by using the discrimination matrix, since to calculate $DP(Y_d \cup y_s, K)$, we will calculate, for each couple (a_i, a_j), the maximum between the value saved in Max Yd and the value of the case$(a_i, a_j), y_s$: $\sum_{i=1,i\neq j}^{n} \sum_{j=1,j\neq i}^{n} Max\left(\text{Max}\, Y_d, d_p\left(e_{si}, e_{sj}\right)\right)$.

Therefore, the calculation of the Equation $DP((Y_d \cup y_s) - (y_l - y_s), K)$ will be done as follows:

$$\sum_{i\neq j}^{q} \sum_{j\neq i}^{q} Max\left(Max\left(\max_{y_{l \in Yd}} d_p(e_{li}, e_{sl}), d_p(e_{si}, e_{sj})\right) - Max\left(d_p(e_{li}, e_{sl}) - d_p(e_{si}, e_{sj})\right), 0\right). \quad (33)$$

When the algorithm selects the variably_s, it calculates $Max\left(\max_{y_{l \in Yd}} d_p(e_{li}, e_{sl}), d_p(e_{si}, e_{sj})\right)$ and its value is saved in the discrimination matrix row Max Yd. This means we will have to do, for each pair (a_i, a_j), only the subtraction of the value of the case $(a_i, a_j), y_l$ and the value of the case (a_i, a_j), y_s; and then we compare the value of the substation with the value found in *Max* Y_d corresponding to the same pair (a_i, a_j).

- Selecting the indispensible variables will be done by doing this test: y_l is indispensible if:

$$\exists\, (a_i, a_j) \in K \text{ where } d_p(e_{li}, e_{lj}) \neq 0 \quad \text{and} \quad \max_{y_p \in Y - y_l}\left(d_p(e_{pi}, e_{pk})\right) = 0 \quad (34)$$

This means that a variable is indispensable if we find a pair of objects discriminated by the variable and not discriminated in any way by any other variable. Using the discrimination matrix, we can find indispensable variables without any complex operations; only the values stored in the discrimination matrix are compared.

6. Application

We validated our algorithm using two categories of testing: quality testing and complexity testing.

6.1. Quality Testing

This validation was carried out on the Tristichacees, Aquatic Insects, and Phlebotomines datasets provided by [20]. In addition, we created two datasets, Phlebotomines Clustered and Tristichacees Clustered, from the result of clustering on the original datasets from Vignes (see Table 4). The clustered data sets have been created using the symbolic object generator program [22]. This program can use object similarity to cluster the objects and generate different types of symbolic objects: boolean or probabilistic objects.

We used the symbolic object generator program to generate individual test data. The program takes into account the domain variables and the dependencies to generate the individuals. Thus, all individuals satisfied the dependencies of their dataset.

In our study, the validation process with test data was carried out based on the calculation of the real object extents. If the intersection between object extents before feature selection is almost the same as that after feature selection, then it can be concluded that the selected feature maintained the same discrimination between objects. This assessment was carried out using the quality criteria Real Discrimination Power Variation (RDPV) defined in Equation (31), and which is based on the Real Discrimination Power (RDP) defined in Equation (36):

$$\left|\frac{RDP(Y_d,K)-RDP(Y,K)}{RDP(Y_d,K)}\right| \leq \beta, \qquad (35)$$

Where:

$$RDP(Y,K) = 1-\sum_{i=1,i\neq j}^{q}\sum_{j=1,j\neq i}^{q}\frac{card(ext\,(a_i)\cap ext(a_j))}{card(ext\,(a_i)\cup ext(a_j))}\ and$$
$$RDP(Y_d,K) = 1-\sum_{i=1,i\neq j}^{q}\sum_{j=1,j\neq i}^{q}\frac{card(ext(a_i')\cap ext(a_j'))}{card(ext(a_i')\cup ext(a_i'))}. \qquad (36)$$

a_i' and a_j' are objects describing a_i and a_j using only the selected variables Y_d.

Table 4. Dataset description.

Dataset	Feature Number	Object Number	Dependency Number	Individual Number	RDP before No Dep	RDP before Dep
Tristichacees	27	12	9	8933	100%	100%
Aquatic Insects	12	16	3	5974	99.7%	99.7%
Phlebotomines	53	73	5	13500	100%	100%
Phlebotomines Clustered	52	10	2	13500	75.8%	87.2
Tristichacees Clustered	26	4	4	8933	99.3%	99.9%

Table 5. Testing result.

Dataset	DP NO Dep	DP Dep	Selected features NO Dep	Selected features Dep	RDP No Dep	RDPV No Dep.	RDP Dep	RDPV Dep
Tristichacees	66	66	4	4	100%	0	100%	0
Aquatic Insects	119	115	8	8	99.6%	0.001	96.6%	0.001
Phlebotomines	2623	2623	16	16	99.9%	0.001	99.9%	0.001
Phlebotomines Cluster	25.66	27.91	7	8	71.5%	0.05	86.3%	0.01
Tristichacees Clustered	5.16	5.75	3	2	80.2%	0.19	94.5%	0.05

In Table 4, for the first three datasets, the calculated *RDP* without and with dependencies are approximately 100%. This means that the objects are totally and equally discriminated by the variables without and with the dependencies. In this case, we expected that the result of feature selection without and with dependencies for the datasets would be the same. We obtained the expected results, the same *DP* (first and second columns of Table 5) and the same number of selected features (third and fourth columns of Table 5). For the clustered datasets:

- From the values in columns two and three of Table 5, it is clear that the calculated DP with dependencies is greater that the calculated DP without dependencies. This means that the dependencies reduced the intersection between the clustered symbolic objects, by removing the overgeneralization area created during the process of generation of clustered symbolic objects.
- The RDP calculated for datasets with dependencies and using only the selected variables were much better than the RDP of the datasets without dependencies and using only the selected variables (see columns five and six of Table 5).
- The RDPVs of the feature selection with dependencies (between 0 and 0.05) are better than the RDPVs of the feature selection without dependency (between 0 and 0.18). This proves that feature selection with variable dependencies produce better results.

6.2. Complexity Testing

The complexity of our algorithm was first determined by comparing the execution time of the algorithm for selecting variables, with and without dependencies, on the datasets presented in Table 4. It can be seen that selecting variables with dependencies is more complex (see Table 6). On average, however, the time taken was twice as much as the time taken with dependencies, which implies that the complexity was not significant.

Table 6. Execution time on datasets.

DataSet	Execution Time No Dep (ms)	Execution Time with Dep (ms)
Tristichacees	89	165
Aquatic Insects	30	78
Phlebotomines	1290	1831
Phlebotomines Cluster	42	89
Tristichacees Clustered	28	31

The other category of complexity testing was applied to study the complexity when the number of dependencies and the number of objects in the datasets are varied.

- Execution Time vs. Number of Dependencies We used the Phlebotomines Clustered dataset and varied the number of dependencies from zero to 50. Figure 6 shows that the execution time was not exponential.
- Execution Time vs. Number of Objects For this test, we used the Phlebotomines data with five dependencies to create datasets with objects varying between five and 75. Figure 7 shows that the execution time for the varying number of symbolic objects was also not exponential.

The results of the two experiments clearly show that although feature selection with variable dependencies is complex, by using various techniques to avoid unnecessary calculations, we successfully reduced the complexity of the algorithm.

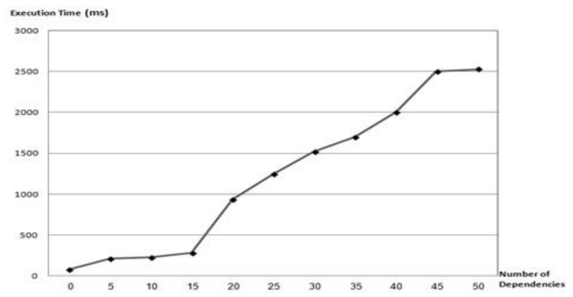


Figure 6. Execution time vs. number of dependencies.

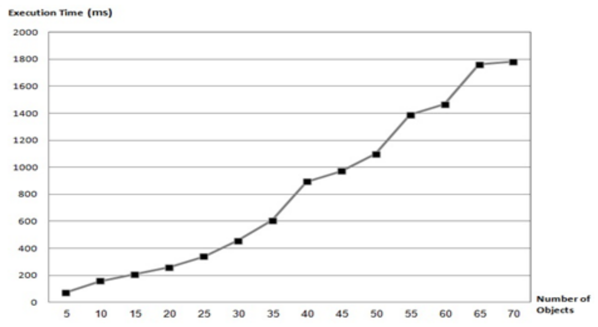


Figure 7. Execution time vs. number of objects7 conclusion.

This paper showed how the dependencies between variables can be used in feature selection on Boolean symbolic objects. The dependencies between variables were shown to be powerful structures that result in the symbolic objects representing individual clusters or concepts more logically and with more precision. Further, because the use of dependencies between variables leads to complex calculation in the criteria selection process, efforts were also made to reduce the complexity of the algorithm. This was achieved by using various mathematical properties to optimize the calculation of the discrimination between two elementary events that use dependencies. More specifically, a Dependency Calculation Matrix and a Discrimination Matrix were employed to avoid unnecessary calculations in the criteria selection process.

The results of experiments conducted on real and simulated data indicate that utilizing the dependencies between variables improves the quality of feature selection. They also prove that the *Minset-Plus* algorithm can deal with large datasets using variable dependencies.

Acknowledgements

The author would like to thank the College of Computer and Information Sciences and the Research Center at King Saud University for their sponsorship.

References

[1] Ben Bassat M. and Zaidenberg L., "Contextual Template Matching: A Distance Measure for Patterns with Hierarchically Dependent Features," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. Pami-6, no. 2, pp. 201-211, 1984.

[2] Csernel M. and De Carvalho F., "On Memory Requirement with Normal Symbolic Form," *in Proceedings of Exploratory Data Analysis in Empirical Research. Springer*, Berlin, pp. 22-30, 2002.

[3] Dale M., *Numerical Syntaxonomy*, Springer Netherlands, 1989.

[4] De Carvalho F., "Proximity Coefficients between Boolean Symbolic Objects," *in Proceedings of New Approaches in Classification and Data Analysis*, Berlim, 387-394, 1994.

[5] De Carvalho F., "Extension Based Proximity Coefficients Between Constrained Boolean Symbolic Objects," *in Proceedings of the 5th Conference of the International Federation of Classification Societies*, Kobe Berlin, pp. 370-378, 1998.

[6] De Carvalho F., Csernel M., and Lechevallier Y., "Clustering Constrained Symbolic Data," *Pattern Recognition Letters*, vol. 30, no. 11, pp. 1037-1045, 2009.

[7] Diday E., "An Introduction to Symbolic Data Analysis," *in Proceedings of the 4th International Conference of the Federation of Classification Societies*, Paris, pp. 53-55,1993.

[8] Gao Y., Koehn P., and Birch A., "Soft dependency Constraints for Reordering in Hierarchical Phrase-Based Translation," *in Proceedings of the Conference on Empirical Methods in Natural Language Processing*, Edinburgh, pp. 857-868, 2011.

[9] Gower J., "A General Coefficient of Similarity and Some of its Properties," *Biometrics*, vol. 27, no. 4, pp. 857-871, 1971.

[10] Gross S. and Huber C., "Hierarchical Dependency Models for Multivariate Survival Data With Censoring," *Lifetime Data Analysis*, vol. 6, no. 4, pp. 299-320, 2000.

[11] He C. and Jeng J., "Feature Selection of Weather Data with Interval Principal Component Analysis," *in Proceedings of international Conference on System Science and Engineering*, Puli, pp. 1-4, 2016.

[12] Kiranagi B., Guru D., and Gudivada V., "Unsupervised Feature Selection Scheme for Clustering of Symbolic Data Using The Multivalued Type Similarity Measure," *in Proceedings of the 2nd Workshop on Feature Selection for Data Mining*, Bethesda, pp. 67-74, 2006.

[13] Klin B. and Sassone V., "Structural Operational Semantics for Stochastic and Weighted Transition Systems," *Information and Computation*, vol. 227, pp. 58-83, 2013.

[14] Kosmelj K., Le-Rademacher J., and Billard L. "Symbolic Covariance Matrix for Interval-

Valued Variables and its Application to Principal Component Analysis: A Case Study," *Metodoloski Zvezki*, vol. 11, no. 1, pp. 1-20, 2014.

[15] Michalski R., Knowledge Acquisition Through Conceptual Clustering: A Theoretical Framework and An Algorithm for Partitioning Data into Conjunctive Concepts," *International Journal of Policy Analysis and Information Systems*, vol. 4, 219-244, 1980.

[16] Nagoya A., Ono Y., and Ichino M., "Detection of Chain Structures Embedded In Multidimensional Symbolic Data," *Pattern Recognition Letters*, vol. 30, no. 11, pp. 951-959, 2009.

[17] Pankhurst R., *Practical Taxonomic Computing*, Cambridge University Press, 1991.

[18] Sneath P., *Numerical Taxonomy*, Springer, 2005.

[19] Tlemsani R. and Benyettou A., "On Line Isolated Characters Recognition Using Dynamic Bayesian Networks," *The International Arab Journal of Information Technology*, vol. 8, no. 4, pp. 406-413, 2011.

[20] Vignes R., "Caractérisation Automatique de Groupes Biologiques," Doctorat Thesis, Paris VI University, 1991.

[21] Ziani D., "Feature Selection on Boolean Symbolic Objects," *International journal of Computer Science and Information Technology*, vol. 5, no. 6, pp. 1-20, 2013.

[22] Ziani D., "Feature Selection on Probabilistic Symbolic Objects," *Frontiers of Computer Science*, vol. 8, no. 6, pp. 933-947, 2014.

[23] Ziani D., "Sélection De Variables Sur Un Ensemble D'objets Symboliques: Traitement Des Dépendances Entre Variables," *University of Paris Dauphine, Dissertation for the Doctoral Degree* (in French), Paris, 1996.

[24] Ziani D., "Variable Hierarchical Dependencies in Feature Selection on Boolean Symbolic Objects," *in Proceedings of 6th International Conference of Soft Computing and Pattern Recognition*, Tunisia, pp. 11-16, 2014.

Djamal Ziani is an associate professor at Al Yamamah University, in Management Information Systems since 2019. He was Associate Professor at King Saud University in the Computer Sciences and Information Systems College from 2009 to 2018. Dr. Djamal is a researcher in ERP and in the data management group. He received a Master's degree in Computer Sciences from the University of Valenciennes, France in 1992, and Ph.D. in Computer Science from the University of Paris Dauphine, France in 1996. He has been a consultant and project manager in many companies in Canada, such as SAP, Bombardier Aerospace, and Montreal Stock Exchange, from 1998 to 2009.