Oddness/evenness-based classifiers for Boolean or numerical data

Myriam Bounhas [a,b], Henri Prade [c,d,*], Gilles Richard [c]

[a] LARODEC Laboratory, ISG de Tunis, 41 rue de la Liberté, 2000 Le Bardo, Tunisia
[b] Emirates College of Technology, P.O. Box: 41009, Abu Dhabi, United Arab Emirates
[c] IRIT, UPS-CNRS, 118 route de Narbonne, 31062 Toulouse Cedex, France
[d] QCIS, University of Technology, Sydney, Australia

Keywords:
Logical proportion
Classification
Boolean data
Numerical data
k-Nearest neighbors method

A B S T R A C T

In this paper, we propose two viewpoints for estimating to what extent a new item, described in terms of binary-valued features, fits with a set of existing items. They are respectively based on an oddness index and an evenness index, which in spite of their names, are not exactly the opposite of each other. Both indicators, which refer to one feature, are built from heterogeneous logical proportions, and involve four items, the new item and three others. Logical proportions are Boolean functions that relate four variables through comparisons between pairs of them. Heterogeneous ones express that there is an intruder among four truth values, which is forbidden to appear in a specific position. Global oddness and evenness functions of an item with respect to a set are built from the corresponding indexes by taking all features into account, and then by considering all triples of items in the set. Moreover the oddness function naturally extends to numerical features and to subsets of items of different sizes (pairs, triples, etc.). Simple classification procedures can be based on these global functions: a new item is assigned to the class that minimizes oddness or maximizes evenness. Experiments on classical benchmarks with Boolean, or numerical data (for oddness) show that the results are competitive with other classification methods.

1. Introduction

It has been acknowledged for a long time that proportions play an important role in our perception and understanding of reality. Indeed proportions are a matter of comparisons expressed by differences or ratios that are equated to other differences or ratios. Two centuries ago, Gergonne [10,11] was the first to explicitly relate numerical (geometric) proportions to the ideas of interpolation and regression.

It is only in the last decade that *analogical* proportions, i.e., statements of the form A is to B as C is to D, where each capital letter refers to a situation described by a vector of feature values, have been formalized first in terms of subsets of properties that hold true in a given situation [12,26], and then in a logical manner [16]. Quite early, it was shown

* Corresponding author at: IRIT, UPS-CNRS, 118 route de Narbonne, 31062 Toulouse Cedex, France.
E-mail addresses: Myriam_Bounhas@yahoo.fr (M. Bounhas), prade@irit.fr (H. Prade), richard@irit.fr (G. Richard).

http://dx.doi.org/10.1016/j.ijar.2016.12.002

that a formal view of analogical proportions may be the basis of a new type of classifier that performs well on some difficult benchmarks [1,15]. This was confirmed by other implementations directly based on a logical view of analogical proportions [3].

Besides, it was shown that analogical proportions belong to a larger family of so-called logical proportions that relate a 4-tuple of Boolean variables [17], where the 8 code-independent logical proportions are of particular interest since their truth status remain unchanged if a property is encoded positively or negatively. These 8 logical proportions divide into 4 *homogeneous* proportions, which include the analogical proportion and 3 related proportions, and 4 *heterogeneous* proportions [20]. A heterogeneous proportion expresses the idea that there is an intruder among the 4 truth values, which is forbidden to appear in a specific position. Intuitively speaking, an item properly assigned to a class should not be (too much) an intruder in this class. It suggests that heterogeneous proportions may be also of interest as a basis for designing a new type of classifier. This is the topic of the paper.

It is a commonsense principle to consider that a class cannot be reasonably assigned to a new item if this item would appear to be at odds with respect to the known members of the class. On the contrary, the item should be even with respect to these class members for entering the class. The use of proportions leads to the idea of considering triples of elements in a class as a basis for estimating the evenness or the oddness of a new item with respect to the class. This departs from the usual view where the estimation of the (non) agreement of a new item with respect to a set of items amounts to compare the item, feature by feature, with a distribution of the feature values in the whole set.

An oddness index and an evenness index, which are not the exact opposite of each other, are proposed. In the evenness view, triples are the only subsets where when the new item conforms with the minority for a given Boolean feature, there is no longer any majority (with respect to this feature) in the triple augmented with the new item. Then one can estimate to what extent a new item fits with the majority of elements in any triple of members of a class on a set of features.

However, it is unclear how to extend the evenness-based approach to numerical data. A slightly different view, based on the direct estimation of oddness, which can still be related to heterogeneous logical proportions, can be also considered. This leads to an oddness measure that can be extended to numerical features in a straightforward manner, and that can be also generalized to subsets of any size and not only on triples. Thus in this paper, the evaluation of the evenness or of the oddness of an item with respect to a class relies on a local view, where the new item, should appear even/not appear at odds with a maximum number of (small) subsets of a considered class.

The paper is organized as follows. The next section provides the necessary background on Boolean logical proportions, introducing the two types of proportions: the homogeneous ones and the heterogeneous ones, by especially emphasizing a code independency property. Results are established that single out these proportions in terms of particular features that are meaningful when it comes to classification. Then, based on heterogeneous proportions, oddness and evenness indexes are introduced in Section 3, and their exact relationship established. The extension of oddness and evenness indexes to a numerical feature is discussed. Section 4 describes heterogeneous proportions-based classifiers. We show how the proposed oddness and evenness indexes can provide a basis for estimating the oddness or the evenness of an item with respect to a whole class. The related work Section 5 provides a brief overview of analogical proportions-based classifiers, which are based on a homogeneous logical proportion, but work quite differently from heterogeneous proportions-based classifiers. Section 6 is devoted to a set of experiments on standard benchmarks coming from the UCI repository. They are compared both with classical classifiers and with analogical proportions-based classifiers. Finally, we provide some hints for future works and concluding remarks in Section 7.

This paper gathers and substantially extends approaches and results partially reported in three conference papers [5–7].

2. Heterogeneous proportions vs. homogeneous proportions

Logical proportions are the basic ingredients of our approach. These proportions are Boolean formulas involving 4 variables. They have been deeply investigated in [19]. In the following section, we first recall how they are built, then we focus on the proportions that we will use in this paper.

As for notations, the paper uses the standard ones for Boolean connectives, namely $\vee, \wedge, \equiv, \to$ for disjunction, conjunction, equivalence and material implication respectively.

2.1. Background on logical proportions

A logical proportion states a relation between 4 items that is expressed in terms of comparisons between pairs of items, each item being represented as a set of Boolean features. Considering 2 Boolean variables a and b corresponding to the same feature attached to 2 items A and B, $a \wedge b$ and $\overline{a} \wedge \overline{b}$ indicate that A and B behave similarly w.r.t. the given feature (they are called "similarity" indicators), $a \wedge \overline{b}$ and $\overline{a} \wedge b$ the fact that A and B behave differently (they are called "dissimilarity" indicators). When we have 4 items A, B, C, D, for comparing their respective behavior in a pairwise manner, we are led to consider logical equivalences between similarity, or dissimilarity indicators, such as $a \wedge b \equiv c \wedge d$ for instance. This enables us to define a logical proportion [19]:

Definition 1. A logical proportion $T(a, b, c, d)$ is the conjunction of two equivalences between indicators for (a, b) on one side and indicators for (c, d) on the other side.

For instance, $((\overline{a} \wedge \overline{b}) \equiv (c \wedge \overline{d})) \wedge ((\overline{a} \wedge b) \equiv (\overline{c} \wedge d))$ is a logical proportion. It has been established that there are 120 syntactically and semantically distinct logical equivalences. There are two ways for distinguishing remarkable subsets among the 120 proportions: either by investigating their structure, or by investigating their semantics (i.e. their truth table). In this section, we shall see that both investigations lead to the same conclusion: there is a class of 8 proportions which stands out of the crowd. This class can be subdivided into 2 sub-groups of 4 proportions.

Indeed a property that appears to be paramount in many reasoning tasks is code independency: there should be no distinction when encoding information positively or negatively. In other words, encoding truth (resp. falsity) with 1 or with 0 (resp. with 0 and 1) is just a matter of convention, and should not impact the final result. When dealing with logical proportions, this property is called *code independency* and can be expressed as

$$T(a, b, c, d) \to T(\overline{a}, \overline{b}, \overline{c}, \overline{d})$$

From a structural viewpoint, remember that a proportion is built up with a pair of equivalences between indicators chosen among 16 equivalences. So, to ensure code independency, the only way to proceed is to first choose an equivalence then to pair it with its counterpart where every literal is negated: for instance $a \wedge b \equiv \overline{c} \wedge d$ should be paired with $\overline{a} \wedge \overline{b} \equiv c \wedge \overline{d}$ in order to get a code independent proportion. This simple reasoning shows that we have only $16/2 = 8$ code independent proportions whose logical expressions are given below.

A: $((a \wedge \overline{b}) \equiv (c \wedge \overline{d})) \wedge ((\overline{a} \wedge b) \equiv (\overline{c} \wedge d))$

R: $((a \wedge \overline{b}) \equiv (\overline{c} \wedge d)) \wedge ((\overline{a} \wedge b) \equiv (c \wedge \overline{d}))$

P: $((a \wedge b) \equiv (c \wedge d)) \wedge ((\overline{a} \wedge \overline{b}) \equiv (\overline{c} \wedge \overline{d}))$

I: $((a \wedge b) \equiv (\overline{c} \wedge \overline{d})) \wedge ((\overline{a} \wedge \overline{b}) \equiv (c \wedge d))$

$\mathbf{H_1}$: $((a \wedge \overline{b}) \equiv (c \wedge d)) \wedge ((\overline{a} \wedge b) \equiv (\overline{c} \wedge \overline{d}))$

$\mathbf{H_2}$: $((\overline{a} \wedge b) \equiv (c \wedge d)) \wedge ((a \wedge \overline{b}) \equiv (\overline{c} \wedge \overline{d}))$

$\mathbf{H_3}$: $((a \wedge b) \equiv (c \wedge \overline{d})) \wedge ((\overline{a} \wedge \overline{b}) \equiv (\overline{c} \wedge d))$

$\mathbf{H_4}$: $((a \wedge b) \equiv (\overline{c} \wedge d)) \wedge ((\overline{a} \wedge \overline{b}) \equiv (c \wedge \overline{d}))$

Only 4 among these proportions make use of similarity and dissimilarity indicators without mixing these types of indicators inside one equivalence: for this reason, these 4 proportions A, R, P, I are called *homogeneous proportions*. For instance, an informal reading of A would be: "a differs from b as c differs from d and vice versa." This expresses the meaning of an analogical proportion, i.e., a statement of the form "a is to b as c is to d". We can consider A as a Boolean counterpart to the idea of numerical proportion, either geometric, i.e., $\frac{a}{b} = \frac{c}{d}$, or arithmetic $a - b = c - d$. It may also be viewed as a qualitative form of comparison of differences, reminiscent to the concept of derivative where we study the ratio $\frac{f(a)-f(b)}{a-b}$ with $f(a) = c$ and $f(b) = d$, which is close to numerical proportions.

The idea of a proportion suggests that some stability properties hold w.r.t. permutations. Indeed, we can permute variables and check, for instance, if a given proportion still holds when permuting the 2 first variables. We denote p_{ij} the permutation of variable in position i with variable in position j. For instance, p_{14} permutes the variables in extreme positions 1 and 4, while p_{23} permutes variables in mean positions. And $p_{12}(a) = b$, $p_{12}(b) = a$, $p_{12}(c) = c$, $p_{12}(d) = d$.

Definition 2. A proportion T is stable w.r.t. permutation p_{ij} iff

$$T(a, b, c, d) \to T(p_{ij}(a), p_{ij}(b), p_{ij}(c), p_{ij}(d))$$

It can be checked that A is stable w.r.t. the extremes p_{14} or the means p_{23} permutations. P is stable for p_{12} and p_{34} permutations, while R is stable for p_{13} and p_{24}. Moreover A, R, P, I are symmetrical (i.e. $T(a, b, c, d) \to T(c, d, a, b)$). This is observable on their truth tables: See the top part of Table 1, where only the 6 patterns that make the logical proportions true appear). Besides, I is the only logical proportion that is stable w.r.t. any permutation of two of its variables. This noticeable result is proved in [18].

Moreover, R and P are closely related to A via permutations. Namely we have

$$A(a, b, c, d) \equiv P(c, b, a, d) \equiv R(b, a, c, d)$$

In fact, when d is fixed, exchanging the variables a, b, c amounts to move from one homogeneous proportion to another, or to remain stable ($A(a, b, c, d) = A(a, c, b, d)$; $P(c, b, a, d) = P(b, c, a, d)$; $R(b, a, c, d) = R(c, a, b, d)$), with I remaining an exception. Thus A, R, P collectively maintain a form of exchangeability property with respect to a, b, c, while I ensures it by itself. These exchangeability properties are of particular interest when applying homogeneous logical proportions to classification.

The 4 remaining code independent logical proportions H_1, H_2, H_3, H_4 are called *heterogeneous proportions*: it is clear from their logical expression that they mix similarity and dissimilarity indicators inside each equivalence. Their truth tables

Table 1
Homogeneous/heterogeneous proportions valid patterns.

A				R				P				I			
0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0
1	1	1	1	1	1	1	1	1	1	1	1	0	0	1	1
0	0	1	1	0	0	1	1	1	0	0	1	1	0	0	1
1	1	0	0	1	1	0	0	0	1	1	0	0	1	1	0
0	1	0	1	0	1	1	0	0	1	0	1	0	1	0	1
1	0	1	0	1	0	0	1	1	0	1	0	1	0	1	0

H_1				H_2				H_3				H_4			
1	1	1	0	1	1	1	0	1	1	1	0	1	1	0	1
0	0	0	1	0	0	0	1	0	0	0	1	0	0	1	0
1	1	0	1	1	1	0	1	1	0	1	1	1	0	1	1
0	0	1	0	0	0	1	0	0	1	0	0	0	1	0	0
1	0	1	1	0	1	1	1	0	1	1	1	0	1	1	1
0	1	0	0	1	0	0	0	1	0	0	0	1	0	0	0

are shown in the bottom part of Table 1, where only the 6 patterns that make them true appear. The index i in H_i refers to a position inside the formula $H_i(a,b,c,d)$. Namely, as can be checked, in each of these 6 patterns there is a minority value (i.e., the value having the smallest number of occurrences in the pattern, then this value is like an intruder among the other values), and i is the only position where the minority value never appears among the 6 4-tuples of values that make H_i true.

By examining the truth table of the heterogeneous proportions in Table 1, we get the following properties:

$$A(a,b,c,d) \wedge R(a,b,c,d) \wedge P(a,b,c,d) \wedge I(a,b,c,d) = \bot$$

together with

$$(A(a,b,c,d) \wedge R(a,b,c,d) \wedge P(a,b,c,d)) \equiv Eq(a,b,c,d)$$

where $Eq(a,b,c,d) = 1$ if $a = b = c = d$ and $Eq(a,b,c,d) = 0$ otherwise. Similarly, for heterogeneous proportions, we have

$$H_1(a,b,c,d) \wedge H_2(a,b,c,d) \wedge H_3(a,b,c,d) \wedge H_4(a,b,c,d) = \bot$$

which implies:

$$(H_1(a,b,c,d) \wedge H_2(a,b,c,d) \wedge H_3(a,b,c,d)) \to \neg H_4(a,b,c,d) \tag{1}$$

Obviously, we have similar properties by permuting the indexes of the H_i's. The meaning of the conjunction $H_1(a,b,c,d) \wedge H_2(a,b,c,d) \wedge H_3(a,b,c,d)$ will be discussed in the following section, devoted to heterogeneous proportions.

2.2. Specificity of heterogeneous proportions

In order to get a clear understanding of the heterogeneous proportions and to extract relevant properties, we now investigate their truth tables.

2.2.1. Heterogeneity and exchangeability

Still within Table 1, an obvious semantics appears: H_i holds when there are exactly 3 parameters with identical Boolean values (=1 for example) and the parameter in position i is one of these identical values.

Definition 3. Given 4 Boolean values a,b,c,d in this order such that 3 of them are identical and the remaining one is different, the position $i \in [1,4]$ of this remaining value is called *the intruder position* or the intruder for brevity.

Then, H_i holds iff there is an intruder among the 4 values a,b,c,d and the intruder position is not i. This suggests that H_i should be stable w.r.t. the permutations which do not affect position i. In fact a little bit more can be established:

Property 1. *Apart from I, H_i are the only logical proportions stable w.r.t any permutation which does not affect position i.*

The special case of I stable w.r.t. any permutation has been already proved in [19]. Table 1 allows to check that the H_i's are stable w.r.t. the permutations which do not affect position i. Showing that they are the only ones among the 120 logical properties stable w.r.t. these permutations requires a tedious checking procedure that cannot be summarized here.

Property 1 is quite satisfactory and confirms the informal semantics of H_i. This will be useful when using heterogeneous proportions to classification. More importantly, this gives a clear semantics to the conjunction $H_1(a,b,c,d) \wedge H_2(a,b,c,d) \wedge$

$H_3(a,b,c,d)$ above: $H_1(a,b,c,d) \wedge H_2(a,b,c,d) \wedge H_3(a,b,c,d)$ holds iff among the 4 values a, b, c, d, there is an intruder and this intruder is d. This will be the basis of our oddness measure. In the next subsection, we establish some results about the parity of the number of 1 or 0 in truth tables for heterogeneous proportions, which are contrasted with homogeneous proportions. This leads to a model of oddness of a given value, among a multiset of 4 values.

2.2.2. Parity of the number of 1 or 0 in tables

Since logical proportions are Boolean formulas involving 4 variables, their truth tables have 16 rows, where only 6 lead to 1 (see [19] for a complete investigation). One could ask if any truth table having 6 lines leading to 1 and 10 lines leading to 0 corresponds to a logical proportion. A simple numbering argument shows that this is not the case. On top of that, we can build classes of patterns which cannot be valid for any proportion:

Property 2. *There is no logical proportion that is true for the four elements of the set of valuations* $\{0111, 1011, 1101, 1110\}$. *The same holds for* $\{1000, 0100, 0010, 0001\}$.

Proof. An equivalence between indicators is of the form $l_1 \wedge l_2 \equiv l_3 \wedge l_4$. If this equivalence is valid for $\{0111, 1011\}$, it means that its truth value does not change when we switch the truth value of the 2 first literals from 0 to 1: there are only 2 indicators for a and b satisfying this requirement: $a \wedge b$ and $\overline{a} \wedge \overline{b}$. If this equivalence is still valid for $\{1101, 1110\}$, its truth value does not change when we switch the truth value of the 2 last literals from 0 to 1: there are only 2 indicators for c and d satisfying this requirement: $c \wedge d$ and $\overline{c} \wedge \overline{d}$. Then the equivalence $l_1 \wedge l_2 \equiv l_3 \wedge l_4$ is just $a \wedge b \equiv c \wedge d$, $a \wedge b \equiv \overline{c} \wedge \overline{d}$, $a \wedge b \equiv \overline{c} \wedge \overline{d}$ or $\overline{a} \wedge \overline{b} \equiv \overline{c} \wedge \overline{d}$. None of these equivalences is true for the four elements of the set of valuations $\{0111, 1011, 1101, 1110\}$. The same reasoning is still applicable for the other class. □

Applying a similar reasoning, we can build other set of valuations which cannot make simultaneously true a logical proportion.

Property 3. *A logical proportion cannot be made true by a set of 4 valuations including 3 valuations of one of the classes appearing in Property 2 and where the 4th valuation is just the negation componentwise of the remaining valuation of the class.*

For instance, there is no logical proportion true for $\{0111, 1011, 1101, 0001\}$ or for $\{0111, 0100, 1101, 1110\}$.

This remark helps establishing the following result:

Property 4. *Heterogeneous proportions are the only proportions whose all valid patterns have an odd number of 1.*

Proof. From the truth tables, we observe that only valid patterns for heterogeneous proportions have an odd number of 1. Let us now consider a proportion whose 6 valid patterns carry an odd number of 1. As there are exactly 8 patterns with an odd number of 1, and thanks to the previous property, this proportion includes necessarily 3 patterns from each of the previous classes. If the valid patterns in one class are obtained from the valid patterns from the other class just by negating all the variables, the proportion is code independent and then, it is a heterogeneous proportion. In the opposite case, it means that we have at least one pattern in the first class with no negated counterpart in the other class: for instance, $1110, 1101, 1011$ are valid but 0001 is not a valid pattern, leaving only $1000, 0100, 0010$ to complete the truth table of a logical proportion. Then Property 3 tells that there is no proportion valid for $1110, 1101, 1011, 1000$. □

A similar property holds for homogeneous proportions:

Property 5. *Homogeneous proportions are the only proportions whose all valid patterns have an even number of 1.*

We now show the specificity of heterogeneous (and homogeneous) proportions from a reasoning point of view.

2.2.3. Inference and univocal proportions

There is a way to infer unknown properties of a partially known object D starting from the knowledge we have about its other specified properties, and assuming that a logical proportion T holds componentwise with three other objects A, B, C, also represented in terms of the same n Boolean features. This can be done via an induction principle that can be stated as follows (where J is a subset of $[1, n]$, and x_i denotes the truth value of feature i for object $X \in \{A, B, C, D\}$):

$$\frac{\forall i \in [1,n] \setminus J,\ T(a_i, b_i, c_i, d_i)}{\forall i \in J,\ T(a_i, b_i, c_i, d_i)}$$

This can be seen as a continuity principle assuming that if it is known that a proportion holds for some attributes, this proportion should still hold for the other attributes. It generalizes the inference principle used with the analogical proportion [18,26] for prediction and classification purposes. From a strict logical viewpoint, this inference rule is unsound as there is no guarantee that the conclusion holds when the premisses hold. Nevertheless, specially when the ratio $\frac{|J|}{n}$ is close to 1, which means that proportions hold on a large number of attributes, it is natural to consider that such a proportion may also hold on the small number of remaining attributes.

This principle requires the unicity of the solution of equation $T(a,b,c,x)=1$ where x is unknown, when it exists. Namely, given 3 Boolean values a,b,c, we want to determine for what logical proportion T the equation $T(a,b,c,x)=1$ is solvable, and in such a case, if the solution is unique.

Definition 4. If, when the equation $T(a,b,c,x)=1$ is solvable, the solution is unique, then the proportion T is said to be 4-*univocal*. In a similar manner, one may define proportions that are 1, 2, or 3-univocal. T is *univocal* when it is i-univocal for every $i \in [1,4]$.

First of all, it is easy to see that there are always cases where the equation $T(a,b,c,x)=1$ has no solution, whatever the proportion T. Indeed, the triple a,b,c may take $2^3=8$ values, while any proportion T is true only for 6 distinct valuations, leaving at least 2 cases with no solution. For instance, when we deal with H_4, the equations $H_4(0,0,0,x)$ and $H_4(1,1,1,x)$ have no solution.

We have the following result:

Property 6. *The homogeneous and the heterogeneous proportions are the only proportions which are univocal.*

Proof. From the truth tables, we see that the 2 types of proportions satisfy the property. Now, a proportion which is not i-univocal is necessarily valid both for a pattern with an odd number of 1, and for a pattern with an even number of 1. Then, Properties 4 and 5 exclude homogeneous and heterogeneous proportions. □

At this step, we see that heterogeneous proportions allow to single out a particular position among an ordered list of 4 values. This position targets the value which is definitely not an intruder among the multiset of 4 items. For instance, when H_i is valid, the value in position i is not an intruder. We shall see in the following section that this property can be used to check the oddness of a given item w.r.t. a multiset of elements.

3. Evaluating the presence or the absence of an intruder in a multiset

Based on heterogeneous proportions, we define an oddness index and then an evenness index, beginning in each case with Boolean values, before investigating their extension to deal with graded truth values. We also lay bare the relation between evenness and oddness.

These oddness and evenness indexes pertain to a Boolean, or a numerical value (denoted x or d in the following) w.r.t. a multiset S of such values. For reasons explained in this section, it will appear desirable to keep this multiset small (i.e., $|S|=3$, or maybe 2 in the oddness case). Moreover, in this section the values x or in S may be thought of as the values of the same feature for different items. In Section 4, we shall build oddness and evenness measures from these indexes by cumulating them over features, and by considering collections of multisets S within the examples describing the same class $\mathcal{C}$ in a training set.

3.1. Oddness index

The idea of oddness introduced below directly relies on the evaluation of the extent to which a new item to be added to a subset reinforces its heterogeneity and so appears as an *intruder* in it.

3.1.1. An oddness measure for Boolean data

In the following, we first define an index to evaluate the *oddness* of a newcomer w.r.t. a multiset of Boolean values via the heterogeneous proportions, where the multiset is a triple. Then, we extend this index to multi-valued logic in the next subsection, before generalizing the extended oddness index to multisets of any size.

Let us remember the meaning of H_i: H_i holds iff there is an intruder among a,b,c,d and the parameter in position i is not this intruder. As shown in Table 2, each proportion H_i provides a piece of knowledge on the intruder and when combined with other pieces, we can pick out which one is the intruder among a,b,c and d. For example $H_1(a,b,c,d)=H_2(a,b,c,d)=H_3(a,b,c,d)=1$ means that there is an intruder which is out of the multiset $\{a,b,c\}$.

Then we define the oddness of d w.r.t. $\{a,b,c\}$ by the following formula:

$$Odd(\{a,b,c\},d) =_{def} H_1(a,b,c,d) \wedge H_2(a,b,c,d) \wedge H_3(a,b,c,d) \tag{2}$$

As an immediate consequence of equation (1), we have:

$$Odd(\{a,b,c\},d) \to \neg H_4(a,b,c,d)$$

Table 2
H_1, H_2, H_3 and Odd truth values.

a	b	c	d	$\mathbf{H_1}$	$\mathbf{H_2}$	$\mathbf{H_3}$	**Odd**
0	0	0	0	0	0	0	0
0	0	0	1	**1**	**1**	**1**	**1**
0	0	1	0	1	1	0	0
0	0	1	1	0	0	0	0
0	1	0	0	1	0	1	0
0	1	0	1	0	0	0	0
0	1	1	0	0	0	0	0
0	1	1	1	0	1	1	0
1	0	0	0	0	1	1	0
1	0	0	1	0	0	0	0
1	0	1	0	0	0	0	0
1	0	1	1	1	0	1	0
1	1	0	0	0	0	0	0
1	1	0	1	1	1	0	0
1	1	1	0	**1**	**1**	**1**	**1**
1	1	1	1	0	0	0	0

Due to the permutation properties of the H_i's, the right hand side of this definition is stable w.r.t. any permutation of a, b, c, then the multiset notation on the left hand side is justified. The truth table of Odd is given in Table 2.

It is clear that Odd holds only when the value of d is seen as *odd* among the other values: d is the intruder. Moreover Odd does not hold in the opposite situation where there is a majority among values in a, b, c, d and d belongs to this majority (e.g. $Odd(\{0, 1, 0\}, 0) = 0$), or there is no majority at all (e.g. $Odd(\{0, 1, 1\}, 0) = 0$).

A simple observation of Table 2 shows that the oddness index can be rewritten as

$$Odd(\{a, b, c\}, d) \equiv ((a \vee b \vee c) \not\equiv d) \wedge (a \wedge b \wedge c) \not\equiv d)) \tag{3}$$

Given a multiset a, b, c of 3 identical Boolean values, $Odd(\{a, b, c\}, d)$ can then act as a flag indicating if the 4th value d is different from the common value of a, b, c. Then the value d is at odds w.r.t. the other values.

3.1.2. Extension to numerical data

It is possible to extend the previous *oddness* measure in order to handle variables with graded values (i.e. variables whose values belong to $[0, 1]$, after a suitable normalization of numerical data). For now, this oddness is just 0 or 1 (i.e. the truth value of $Odd(\{a, b, c\}, d)$), but we would like to consider tuples such as $(0.1, 0.2, 0.1, 0.8)$ and still consider that the 4th value is somewhat odd w.r.t. the 3 other ones.

A direct translation of formula (2), taking min for $\wedge$, max for $\vee$, and $1 - |\cdot - \cdot|$ for $\equiv$ as in Łukasiewicz logic [23], leads to:

$$Odd(\{a, b, c\}, d) =_{def} \min(|\max(a, b, c) - d|, |\min(a, b, c) - d|) \tag{4}$$

First of all, it is an easy game to check that Odd remains code independent with graded values, i.e. changing values into their complement to 1.

Let us examine some examples to get a precise understanding of the formula for numerical data and to check if this *oddness* measure fits with the intuition.

- We see that $Odd(\{u, u, u\}, v) = |u - v|$. Indeed, if $u = v$ then obviously the 4th value is not an intruder. The larger $|u - v|$, the more v is at odds w.r.t the 3 values equal to u.
- We see also that $Odd(\{v, u, u\}, v) = 0$ which is consistent with the expected semantics of Odd.
- Generally, $Odd(\{u, v, w\}, \max(u, v, w)) = Odd(\{u, v, w\}, \min(u, v, w)) = 0$, and in any case, $Odd(\{u, v, w\}, u) \leq 0.5$.
- Let us now consider a numerical situation with 4 different numerical values, for instance: $(\{0, 0.1, 0.2\}, 0.9)$. We feel that $d = 0.9$ appears as an intruder in the multiset $(\{0, 0.1, 0.2\})$. This is consistent with the obtained truth value $Odd(\{0, 0.1, 0.2\}, 0.9) = 0.7$. Moreover, $Odd(\{0, 0.1, 0.1\}, 0.9) = 0.8$ and $Odd(\{0, 0.1, 0.3\}, 0.9) = 0.6$, which fits with the intuition.
- Conversely, the pattern $(\{0.7, 1, 1\}, 0.9)$ does not strongly suggest 0.9 as an intruder value. Indeed $Odd(\{0.7, 1, 1\}, 0.9) = 0.1$. $Odd(\{0.9, 1, 1\}, 0.7) = 0.2$ is a bit higher, as expected since we have moved towards more uniformity among a, b, c and slightly increased the differences between d and the elements of $\{a, b, c\}$. Moreover, note that $Odd(\{0.7, 1, 1\}, 0.9) = 0.1$, and $Odd(\{0, 0, 1\}, 0.9) = 0.1$, since the two cases illustrate two different ways of not being really an intruder. Indeed, although 0.9 is close to the majority value in $\{0.7, 1, 1\}$ in the first case, and far from the majority value in $\{0, 0, 1\}$ in the second case, closeness to majority value in $\{a, b, c\}$ is not at all what Odd estimates. Rather it is expected to find similar estimates in the two above cases, since they are respectively close to $Odd(\{1, 1, 1\}, 1) = 0$ and to $Odd(\{0, 0, 1\}, 1) = 0$ as shown in Table 2.
- Finally, $Odd(\{a, b, c\}, d)$ does not behave as $|d - average(\{a, b, c\})|$: the cases $\{a, b, c\} = \{0.5, 0.5, 0.5\}$, $d = 0.5$, and $\{a, b, c\} = \{0, 0.5, 1\}$, $d = 0.5$ cannot be distinguished by the second average-based expression, but, with our defini-

tion, $Odd(\{0.5, 0.5, 0.5\}, 0.5) = 0$, while $Odd(\{0, 0.5, 1\}, 0.5) = 0.5$. Thus $Odd(\{a, b, c\}, d)$ is a more accurate oddness measure than $|d - average(\{a, b, c\})|$ *when* the set $\{a, b, c\}$ contains heterogeneous values.

From the previous examples, we understand that the proposed definition fits with the initial intuition and provides high truth values when d appears to be at odds w.r.t. the multiset $\{a, b, c\}$ and low truth values in the opposite case where d is not very different from the other values. On top of that, the expression of Odd given here is no longer the conjunction of the *multiple-valued extensions* of H_1, H_2, H_3 as given in [20], which would lead to a less satisfactory measure of oddness. Indeed, we are here interested in the oddness of d w.r.t. a multiset $\{a, b, c\}$, and not in picking out an intruder in the multiset $\{a, b, c, d\}$ as in [20].

3.1.3. Oddness with respect to multisets of various size

Since we are interested in checking if d seems an intruder in a given multiset, we may consider multisets of any size when defining the oddness index. Indeed, the previous oddness index is not limited to multisets $\{a, b, c\}$ with 3 elements, and can be easily generalized to an index of oddness $Odd(S, x)$ of an item x w.r.t. a multiset S of values in $[0, 1]$ of any size, as follows:

$$Odd(S, x) =_{def} \min(|\max(S) - x|, |\min(S) - x|)$$

As can be seen, we only compare x to the upper and lower values in S, which may be really considered as a meaningful summary of S only if S is very small (when we have no additional information about the distribution of values in S), i.e. $|S| = 1, 2, 3$ or may be 4. Clearly, the computation of $Odd(S, x)$ reflects the smallest distance of x to an element in S only for $|S| = 1, 2$. Indeed, this is not the case as soon as $|S| \geq 3$ as can be seen on the following example: $Odd(\{a, b, c\}, x) = Odd(\{0, 0.5, 1\}, 0.5) = 0.5$, while $|b - x| = |0.5 - 0.5| = 0$.

As it may be the case for real datasets, we may have missing values. Obviously, when there is a missing value in the multiset S of size n, then a simple option is to consider the multiset S' of size $n - 1$ and to consider $Odd(S, x) = Odd(S', x)$ where S' has no longer any missing value.

3.2. Evenness index

Adopting a dual viewpoint, we may want to know if adding a new element to a given subset of items keeps it as homogeneous as it is, i.e., the newcomer does not appear as an intruder in this subset, and rather agrees with its majority. Homogeneity can be considered as a kind of *evenness* of the newcomer w.r.t. the existing items of the subset. In this subsection, we advocate a way of judging *evenness* on the basis of the majority, if any, inside the triples. Contrary to the oddness definition, where all $H_i, i = 1, 2, 3$ are required to define the oddness index, only H_4 is needed for defining an evenness index, thus denoted $Even_4$.

3.2.1. An evenness measure for Boolean data

Since the idea is to agree with a majority, we notice that the smallest multisets S of elements where majority makes sense are clearly triples. Let consider three Boolean values a, b, c in S. Then, in a Boolean world, there are two possibilities, either $a = b = c$, or two of the three are equal. In both cases, a strict majority takes place. Let m denote the majority value. Now consider the newcomer d, either $d = m$, and m remains the majority value in $\{a, b, c, d\}$, or $d \neq m$, and *there is no longer any majority in* $\{a, b, c, d\}$ (two values are equal to 1 and two values to 0). Only with the first case, d conforms to the majority.

Note that if we consider *larger* subsets S, even with only 4 elements rather than 3, it becomes possible that the newcomer increases an existing minority, without changing the majority. Indeed, the majority value that may be shared by 3 elements in the 4-elements multiset will then remain unchanged in the 5-elements multiset resulting from the arrival of a fifth element whatever its value. A similar phenomenon takes place if we start with larger subsets S having 5 elements or more. So we are losing a distinctive property of 3-elements subsets which have a different majority behavior depending if d conforms or not to the majority in the 3-elements subset. *This means that triples are the only subsets such that adding an item that conforms to the triple minority destroys the majority*. Thus, 3-elements subsets are able to clearly discriminate, among different d those that conform to the majority of the triple.

The idea of majority just described helps us to define a new *evenness* measure via the heterogeneous proportions. Let us recall the semantics of H_i: H_i holds iff there is an intruder among a, b, c, d and the parameter in position i is not this intruder. As a consequence, H_i implies that there is a majority of values among (a, b, c, d) and the value in position i conforms to the majority of values appearing among the 3 other positions (i.e. the multiset of values $\{a, b, c, d\}$ is more or less even). But the reverse implication does not hold since when the 4 parameters have identical value, $\forall i \in [1, 4], H_i(a, b, c, d) = 0$. Then, to have a concise Boolean definition for "there is a majority of values among the parameters a, b, c, d and the parameter in position i belongs to this majority of values", we need to consider the case where all the values are identical by using the following formula:

$$Even_i(a, b, c, d) =_{def} H_i(a, b, c, d) \vee Eq(a, b, c, d) \quad (5)$$

Table 3
H_4, Eq and $Even_4$ truth values.

				$\mathbf{H_4}$	**Eq**	$\mathbf{Even_4}$
0	0	0	0	0	1	1
0	0	0	1	0	0	0
0	0	1	0	1	0	1
0	0	1	1	0	0	0
0	1	0	0	1	0	1
0	1	0	1	0	0	0
0	1	1	0	0	0	0
0	1	1	1	1	0	1
1	0	0	0	1	0	1
1	0	0	1	0	0	0
1	0	1	0	0	0	0
1	0	1	1	1	0	1
1	1	0	0	0	0	0
1	1	0	1	1	0	1
1	1	1	0	0	0	0
1	1	1	1	0	1	1

where $Eq(a,b,c,d) =_{def} (a \equiv b) \wedge (b \equiv c) \wedge (c \equiv d)$. Thus, with $Even_i$ we take into account the special case where all the values are equal. The truth table of $Even_4$ is given in Table 3. It is clear that $Even_4$ holds only when the value of d belongs to a majority of the parameter's values. And $Even_4$ does not hold in an opposite situation where there is no majority of values as it is the case for $Even_4(0011)$ or $Even_4(0110)$.

The situations where $Even_4(a,b,c,d) = 1$ exactly cover the two cases already mentioned where d is identical to the majority value in the triple $\{a,b,c\}$ (is not the intruder), namely either $a = b = c$, or two of the three are equal to d. So the fact that d joins $\{a,b,c\}$, when $Even(a,b,c,d) = 1$, leaves the resulting subset as *even* as it was, hence the name, and in fact the majority is reinforced by the arrival of d. Note also that $Even_4(a,b,c,d)$ is left unchanged by any permutation of $\{a,b,c\}$. This means that the ordering inside triples does not matter. Besides, $Even_4(a,b,c,d) = Even_4(\overline{a},\overline{b},\overline{c},\overline{d})$ where $\overline{x} = 1$ if $x = 0$ and $\overline{x} = 0$ if $x = 1$, expressing that $Even_4(a,b,c,d)$ does not depend on the way the information is encoded. From now on, $Even_4$ will be denoted $Even$ as this is the only option we use with $i = 4$.

3.2.2. Relation between oddness and evenness in the Boolean case

The oddness and evenness Boolean functions have been built by truth tables inspection. However, these 2 functions exhibit noticeable links. Despite the fact that their name might suggest that oddness and evenness capture dual concepts, it is not the case that $Even(a,b,c,d) \equiv \neg Odd(\{a,b,c\},d)$. In fact, the relations between the 2 measures are as follows (where I denotes the inverse paralogy defined in section 2.1):

Property 7.

$$Even(a,b,c,d) \equiv \neg Odd(\{a,b,c\},d) \wedge \neg I(a,b,c,d)$$
$$Odd(\{a,b,c\},d) \equiv \neg Even(a,b,c,d) \wedge \neg I(a,b,c,d)$$
$$I(a,b,c,d) \equiv \neg Even(a,b,c,d) \wedge \neg Odd(\{a,b,c\},d)$$

This can be easily checked on the truth tables. This reflects the fact that Odd and $Even$ and I are mutually exclusive. Let us note that $Even(a,b,c,d) \to \neg Odd(\{a,b,c\},d)$, that $\neg Even(a,b,c,d) \to \neg H_4(a,b,c,d)$ and that $Odd(\{a,b,c\},d) \to \neg H_4(a,b,c,d)$ as well.

A more complete discussion of the relation between oddness and evenness can be found in [21].

3.2.3. Extension to numerical data

In order to deal with graded truth values, we need to extend the previous definition of Even given in formula (4) to the case where the truth values belong to $[0,1]$, as we have done for the oddness function.

A direct translation of formula (4), taking min for $\wedge$, max for $\vee$, and $1 - |\cdot - \cdot|$ for $\equiv$ as in Łukasiewics logic, leads to the following expression for $Even_4$:

$$max(min(1 - |min(a,b) - min(1-c,d)|, 1 - |min(1-a,1-b) - min(c,1-d)|),$$
$$1 - |max(a,b,c,d) - min(a,b,c,d)|)$$

Let us examine the behavior of this definition. In order to get a clear picture, we consider the 2 following curves:

- $f(x) = Even(0,x,x,x)$: we would expect f to get the constant value 1, since, whatever its value, the last element x, cannot be considered as an intruder in the multiset $\{0,x,x\}$.

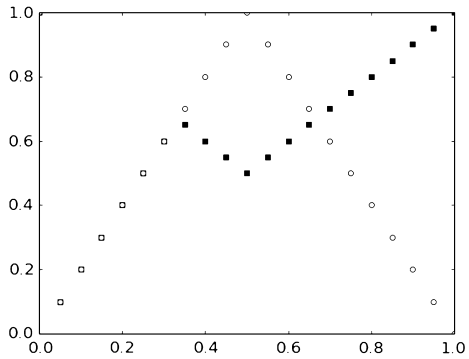

Fig. 1. f and g functions with standard definition of $Even$.

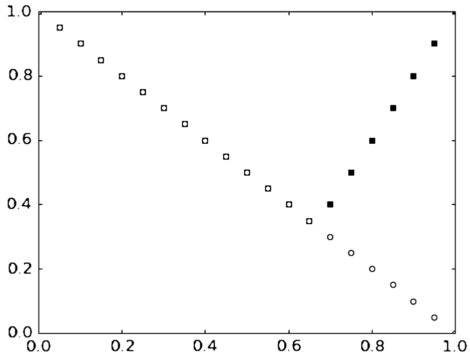

Fig. 2. f and g functions with the new definition of $Even$.

- $g(x) = Even(0, x, x, 0)$: we expect a function decreasing from 1 to 0 when x goes from 0 to 1. Indeed, the smaller x, the closer to 1 $Even(0, x, x, 0)$ should be, while the larger x the more 0 appears to be equal to the minority value in the multiset $\{0, x, x\}$.

The corresponding curves are in Fig. 1. Black square dots are for function f, empty circles for function g (mind that an empty circle may partially hide a black square when f and g coincide). As can be seen, f is not a constant function and g is not monotonically decreasing. This contrasts with $Odd(0, x, x, x) = 0$ and $Odd(0, x, x, 0) = 0$.

It appears that a direct translation of the Boolean definition (4) does not fit with the expected meaning of evenness in the case of graded truth values. But obviously, we could start from the property $Even \equiv \neg Odd \wedge \neg I$ to get another translation as:

$$Even(a, b, c, d) = min(1 - Odd(a, b, c, d), 1 - I(a, b, c, d))$$

This new definition leads via an easy computation to $Even(0, x, x, x) = 1 - x$ if $x \leq 0.5$ and $1 - min(x, 2 - 2x)$ when $x \geq 0.5$. With this new definition of $Even$, the curves corresponding to f and g are given in Fig. 2 (we use black square dots for function f and empty circles for function g again).

Observe that the behavior of $g(x) = Even(0, x, x, 0)$ is satisfactory since we get the decreasing function $1 - x$. However $f(x) = Even(0, x, x, x)$ may be far from 1 (in particular, $Even(0, \frac{2}{3}, \frac{2}{3}, \frac{2}{3}) = \frac{1}{3}$). Such a behavior is not satisfactory at all. It is still an open question to find a better definition for $Even$ in the graded case, which would coincide with the Boolean case when $a, b, c, d \in \{0, 1\}$. For this reason, we shall not experiment the evenness function with numerical data.

3.2.4. Dealing with missing values

Missing information is quite common in real life datasets and a way to extend the semantics of analogical proportion to deal with this issue has been deeply investigated in [20] for instance. In fact, such an approach can be also applied here, as explained now.

Still keeping a logical approach and considering that '?' denotes a missing value (i.e. an information is unknown), the idea is to extend the truth table of the $Even$ formula as follows: $Even(?, 0, 0, 0) = Even(0, ?, 0, 0) = Even(0, 0, ?, 0) = 1$, $Even(?, 1, 1, 1) = Even(1, ?, 1, 1) = Even(1, 1, ?, 1) = 1$, and $Even(x, y, z, t) = 0$ for any other pattern including at least a missing value '?'. It is clear that, with the 6 first patterns, whatever the candidate value of the missing feature, the 4th argument belongs to the majority and cannot be an intruder. In all the remaining cases, where we have no certainty regarding the status of d, we adopt a cautious behavior by considering that $Even$ does not hold.

4. Heterogeneous proportions-based classifiers

In the previous section, we have defined two new indexes that evaluate the *oddness* or *evenness* of an item with respect to a multiset S of a fixed size (especially S is a triple in case of evenness). In the context of classification, our aim is to maintain homogeneity or evenness inside each class $\mathcal{C}$ and to avoid oddness when classifying a new item. For this purpose, we first extend Odd and $Even$ indexes to deal with vectors instead of simple Boolean or numerical values, and then, build up a global oddness/evenness measure of an item x w.r.t. a class $\mathcal{C}$.

In the following, we propose a family of classifiers based on these global evenness/oddness measures and indexed by the size of the subsets (used in the comparison process for the oddness-based classifiers). But we first recall the Bayesian view of the conformity of an item w.r.t. a class, which contrasts with the views proposed in the rest of this section.

4.1. The Bayesian viewpoint

In a classical Bayesian view, we have $Prob(\mathcal{C}|\vec{x}) = \frac{1}{Z} \cdot Prob(\mathcal{C}) \cdot \prod_{i=1}^{n} Prob(x_i|\mathcal{C})$ assuming that the n features are independent. The evidence Z depends only on $\vec{x}$, $Prob(\mathcal{C})$ reflects some characteristics of $\mathcal{C}$ such as its size, and $\prod_{i=1}^{n} Prob(x_i|\mathcal{C})$ evaluates the conformity of $\vec{x} = (x_1, \cdots, x_n)$ with $\mathcal{C}$. Under some conditional independence assumptions, this probability can be rewritten as a weighted product of $Prob(x_i|\mathcal{C})$, i.e. the conditional probability to get value x_i for feature i in the class $\mathcal{C}$. Usually, $Prob(x_i|\mathcal{C})$ is estimated as the frequency of elements having value x_i for feature i in the whole class $\mathcal{C}$. Thus, the expression of $Prob(\mathcal{C}|\vec{x})$ involves the product of $Prob(\mathcal{C})$ with a kind of conjunctive combination expressed as the product of the proportion of elements of $\mathcal{C}$ identical to $\vec{x}$ for each feature i. A counterpart of this evaluation exists in the setting of possibility theory [2]. In the case of Boolean features, let $p(\mathcal{C}, i)$ be the proportion of the majority value for feature i in $\mathcal{C}$. Then, an elementary estimation of $Prob(\mathcal{C}|\vec{x})$ is:

$$\frac{1}{Z} \cdot Prob(\mathcal{C}) \cdot \prod_{i \in M} p(\mathcal{C}, i) \cdot \prod_{j \in \overline{M}} (1 - p(\mathcal{C}, j))$$

where $M \subseteq \{1, \cdots, n\}$ is the subset of features where $\vec{x}$ is conform to the majority in $\mathcal{C}$, and $\overline{M}$ is the complementary subset where $\vec{x}$ is not conform to the majority. The idea of conformity in this approach is thus related to the notion of majority w.r.t. the *whole* set $\mathcal{C}$ itself. In the following, we investigate the idea of judging conformity w.r.t. *a collection of smaller subsets* $\mathcal{S} \subset \mathcal{C}$, and then to cumulate the results of the comparison of $\vec{x}$ with the different subsets $\mathcal{S}$.

4.2. Classification indexes

When it comes to real life application, it is not enough to represent individuals with a single Boolean or real value. Generally, individuals are encoded by a set of features. Based on the previously defined oddness and evenness measures, we have to define new measures suitable for vectors.

4.2.1. Oddness and evenness measures for vectors

When dealing with vectors $\overrightarrow{x} \in [0, 1]^n$, Boolean vectors are also covered as a particular case. The Odd and $Even$ measures, defined respectively by (1) and (4), are used to estimate to what extent a value x can be considered as *odd* or *even* among a multiset S of values. Thanks to the two latter formulas, assuming the independence of features, it is natural to compute the *oddness* or *evenness* of a vector $\overrightarrow{x}$ as the *sum* of the *oddness* or *evenness* for each feature $x_i \in \overrightarrow{x}$, as follows:

$$Odd(\mathcal{S}, \overrightarrow{x}) =_{def} \Sigma_{i=1}^{n} Odd(S_i, x_i) \in [0, n]$$

$$Even(\mathcal{S}, \overrightarrow{x}) =_{def} \Sigma_{i=1}^{n} Even(S_i, x_i) \in [0, n]$$

where x_i is the i-th component of $\overrightarrow{x}$, S_i is the multiset gathering the i-th components of the vectors in $\mathcal{S}$. Note that in the case of $Even$, the set $\mathcal{S}$ has exactly 3 elements, but we keep the set notation for sake of notation uniformity.

If our aim is to measure oddness, high values of $Odd(\mathcal{S}, \overrightarrow{x})$ (close to n) means that, for *many* features, $\overrightarrow{x}$ appears as an intruder and may reduce the homogeneity when going from $\mathcal{S}$ to the multiset $\mathcal{S} \cup \{\overrightarrow{x}\}$. If $Odd(\mathcal{S}, \overrightarrow{x}) = 0$, no feature indicates that $\overrightarrow{x}$ behaves as an intruder and there is no obstacle for $\overrightarrow{x}$ to join the multiset S.[1]

On the opposite, if our aim is to compute evenness of the subset $\mathcal{S}$ when $\overrightarrow{x}$ is being added, the bigger $Even(\mathcal{S}, \overrightarrow{x})$, the larger the number of features for which $\overrightarrow{x}$ conforms to the majority in $\mathcal{S}$, the better $\overrightarrow{x}$ conforms to vectors in $\mathcal{S}$. If $Even(\mathcal{S}, \overrightarrow{x}) = n$, there does not exist a feature where $\overrightarrow{x}$ behaves as an intruder. Then, it is acceptable for $\overrightarrow{x}$ to join the set $\mathcal{S}$.

[1] It is clear that when dealing with classification task, $\mathcal{S}$ is just a set of examples, without any repetition, but obviously, its projections componentwise, the S_i's are multisets of Boolean or real values.

4.2.2. Global oddness and evenness measures

Given a set $\mathcal{C}$ of vectors belonging to the same class and a non-null integer m, we can compute $Odd(\mathcal{S}, \overrightarrow{x})$ for each distinct subset $\mathcal{S} \subseteq \mathcal{C}$ of cardinality m and $Even(\mathcal{S}, \overrightarrow{x})$ for each distinct subset $\mathcal{S} \subseteq \mathcal{C}$ of cardinality 3. The evenness or oddness measures of the vector $\overrightarrow{x}$ in the class $\mathcal{C}$ could simply be the sum of all these elementary values. For instance as follows:

$$\Sigma_{\mathcal{S}\subseteq\mathcal{C} s.t. |\mathcal{S}|=m} Odd(\mathcal{S}, \overrightarrow{x}) \text{ for Oddness}$$

and

$$\Sigma_{\mathcal{S}\subseteq\mathcal{C} s.t. |\mathcal{S}|=3} Even(\mathcal{S}, \overrightarrow{x}) \text{ for Evenness}$$

Obviously it is fair to take into account the relative size of the different classes $\mathcal{C}$ and, as a consequence, to introduce a normalization factor.

a. *Oddness measure*
Clearly, the number $\binom{|\mathcal{C}|}{m}$ of subsets $\mathcal{S} \subset \mathcal{C}$ of size m is an increasing function of $|\mathcal{C}|$. When $\mathcal{C}$ is large, this number is not far from $|\mathcal{C}|^m$. It is then relevant to consider the following definition:

$$ODD_m(\mathcal{C}, \overrightarrow{x}) =_{def} \frac{1}{|\mathcal{C}|^m} \Sigma_{\mathcal{S}\subseteq\mathcal{C} s.t. |\mathcal{S}|=m} Odd(\mathcal{S}, \overrightarrow{x})$$

When $m = 2$, we deal with pairs, when $m = 3$, we deal with triples, etc. In case of singletons ($m = 1$), $\mathcal{S} = \{\overrightarrow{y}\}$, and

$$ODD_1(\{\overrightarrow{y}\}, \overrightarrow{x}) = \Sigma_{i=1}^{n} Odd(y_i, x_i) = \Sigma_{i=1}^{n} |y_i - x_i|,$$

which is just the Hamming distance between $\overrightarrow{y}$ and $\overrightarrow{x}$. As a consequence, $ODD_1(\mathcal{C}, \overrightarrow{x})$ is the average distance between $\overrightarrow{x}$ and the elements in $\mathcal{C}$.

b. *Evenness measure*
A reasoning similar to the previous one leads to a definition of evenness as follows (since we use only subsets of cardinality 3):

$$EVEN(\mathcal{C}, \overrightarrow{x}) =_{def} \frac{1}{|\mathcal{C}|^3} \Sigma_{\mathcal{S}\subseteq\mathcal{C} s.t. |\mathcal{S}|=3} Even(\mathcal{S}, \overrightarrow{x}).$$

4.2.3. Optimization

It is clear that the calculation process of ODD_m may be time consuming for large values of m ($m \geq 3$). To reduce this complexity, we have chosen to take one element as a k nearest neighbor of the new item $\overrightarrow{x}$. Let us denote $\{\overrightarrow{y_j} | j \in [1, k]\}$ the k nearest neighbors of $\overrightarrow{x}$ in $\mathcal{C}$. So the oddness measure that will be used in practice is:

$$k\text{-}ODD_m(\mathcal{C}, \overrightarrow{x}) =_{def} \frac{1}{|\mathcal{C}|^{m-1}} \Sigma_{j=1}^{k} (\Sigma_{\mathcal{S}\subseteq\mathcal{C}\setminus\{\overrightarrow{y_j}\} s.t. |\mathcal{S}|=m-1} Odd(\mathcal{S} \cup \{\overrightarrow{y_j}\}, \overrightarrow{x}))$$

The oddness of an element is now the sum of k numbers, k being the number of nearest neighbors that we consider. Exactly the same approach applies for $EVEN$ measure leading to:

$$k\text{-}EVEN(\mathcal{C}, \overrightarrow{x}) =_{def} \frac{1}{|\mathcal{C}|^2} \Sigma_{j=1}^{k} (\Sigma_{\mathcal{S}\subseteq\mathcal{C}\setminus\{\overrightarrow{y_j}\} s.t. |\mathcal{S}|=2} Even(\mathcal{S} \cup \{\overrightarrow{y_j}\}, \overrightarrow{x}))$$

4.3. Algorithm

Let TS be a training set composed of instances $(\overrightarrow{z}, cl(\overrightarrow{z}))$, where $\overrightarrow{z} \in \mathbb{B}^n$ or $\mathbb{R}^n$, $cl(\overrightarrow{z})$ is the label of $\overrightarrow{z}$. Given a new instance $\overrightarrow{x} \notin TS$ without label, we have to allocate a label to $\overrightarrow{x}$ by looking for the class that better maintains its homogeneity when $\overrightarrow{x}$ is added to it. More formally, given the set $\mathcal{C}$ of instances in TS having the same label c, we estimate to what extent $\mathcal{C} \cup \{\overrightarrow{x}\}$ is odd or even. Based on the *oddness* and *evenness* measures defined before, the idea is then to assign to $\overrightarrow{x}$ the label corresponding to the class *minimizing* the oddness or *maximizing* the evenness when $\overrightarrow{x}$ is added.

Our implementations fit with the following simple procedure.

1. Choose a number k of nearest neighbors to be considered
2. For each class (or label) $\mathcal{C}$, compute $k\text{-}ODD_m(\mathcal{C}, \overrightarrow{x})/k\text{-}EVEN(\mathcal{C}, \overrightarrow{x})$.
3. Allocate to $\overrightarrow{x}$ the label $argmin_{\mathcal{C}}\ k\text{-}ODD_m(\mathcal{C}, \overrightarrow{x})/argmax_{\mathcal{C}}\ k\text{-}EVEN(\mathcal{C}, \overrightarrow{x})$

The previous procedure can be described with the pseudo-code of Algorithm 1. Algorithm 1 can deal with missing values thanks to the remark at the end of section 3.1.3 in case of oddness, and the extension presented in section 3.2.4 in case of evenness.

Algorithm 1 Oddness/Evenness-based algorithm.

Input: a training set TS of examples $(\overrightarrow{z}, cl(\overrightarrow{z}))$
a non-null integer m
an integer $k \geq 1$
a new item $\overrightarrow{x}$,
Partition TS into sets $\mathcal{C}$ of examples having the same label c.
for each $\mathcal{C}$ **do**
Compute $k\text{-}ODD_m(\mathcal{C}, \overrightarrow{x})/k\text{-}EVEN(\mathcal{C}, \overrightarrow{x})$
end for
$cl(\overrightarrow{x}) = argmin_{\mathcal{C}}\ k\text{-}ODD_m(\mathcal{C}, \overrightarrow{x})/argmax_{\mathcal{C}}\ k\text{-}EVEN(\mathcal{C}, \overrightarrow{x})$
return $cl(\overrightarrow{x})$

Our approach might appear somehow similar to k-nearest neighbors (k-nn) methods. However, the proposed method relies on the comparison of a newcomer with respect to subsets $\mathcal{S}$ involving $m = 2, 3$, or more elements, which are not singletons, while k-nn methods compare the newcomer with examples taken one by one. Obviously, this has a greater computational cost, since in the basic method, we have to consider all the subsets of size m in the training set.

Before reporting results of experiments with oddness or evenness-based classifiers, we briefly present analogical proportions-based classifiers, with which they will be compared, as well as with standard classifiers.

5. Related work. Analogical proportions-based classifiers

In the last decade, diverse classification approaches have been developed based on analogical proportion. We first review the Boolean case, before considering the numerical case.

5.1. Boolean and discrete cases

In [1], the authors use a measure of *analogical dissimilarity* between 4 objects. It estimates how far 4 objects are from building a perfect analogical proportion. Roughly speaking, the analogical dissimilarity *ad* between 4 Boolean values is the minimum number of bits that have to be switched to get a proper analogy. Thus we have:

$$\mathtt{ad}(1,0,1,0) = 0, \mathtt{ad}(1,0,1,1) = 1 \text{ and } \mathtt{ad}(1,0,0,1) = 2$$

It means, $a : b :: c : d$ holds if and only if $\mathtt{ad}(a,b,c,d) = 0$. Moreover $\mathtt{ad}$ differentiates two cases where analogy does not hold, namely the 8 cases with an odd number of 0 and an odd number of 1 among the 4 Boolean values, such as $\mathtt{ad}(0,0,0,1) = 1$ or $\mathtt{ad}(0,1,1,1) = 1$, and the two cases $\mathtt{ad}(0,1,1,0) = \mathtt{ad}(1,0,0,1) = 2$. When we deal with 4 Boolean vectors in $\mathbb{B}^n$, adding the $\mathtt{ad}$ evaluations componentwise generalizes the analogical dissimilarity to Boolean vectors, and leads to an integer belonging to the interval $[0, 2n]$. It is used in [1] in the implementation of a classification algorithm where the input parameters are a set TS of classified items, an integer k, and a new item $\overrightarrow{d}$ to be classified. It proceeds as follows:

Step 1: Compute the analogical dissimilarity $\mathtt{ad}$ between $\overrightarrow{d}$ and all the triples in TS^3 that produce a solution for the class of $\overrightarrow{d}$.

Step 2: Sort these n triples by the increasing value of $\mathtt{ad}$ w.r.t. with $\overrightarrow{d}$.

Step 3: Let p be the value of $\mathtt{ad}$ for the k-th triple, then find k' as being the greatest integer such that the k'-th triple has the value p.

Step 4: Solve the k' analogical equations on the label of the class. Take the winner of the k' votes and allocate this winner as the class of $\overrightarrow{d}$.

This approach provides remarkable results and, in several cases, outperforms the best known algorithms [15].

In the algorithm proposed in [17], there is no use of a dissimilarity measure but a straightforward implementation of the continuity principle, keeping flexibility by allowing to have some components where analogy does not hold. Triples of Boolean vectors $(\overrightarrow{a}, \overrightarrow{b}, \overrightarrow{c})$ are considered such that the class equation $cl(\overrightarrow{a}) : cl(\overrightarrow{b}) :: cl(\overrightarrow{c}) : x$ is solvable and such that the number of componentwise analogies $card(\{i \in \{1,n\} |\ a_i : b_i :: c_i : d_i \text{ holds}\})$ is maximal. Then the label solution of the corresponding class equation is allocated to $\overrightarrow{d}$, implementing a majority vote in case of multiple candidate triples.

Although the work of [26] does not deal with Boolean vectors strictly speaking, they are the first authors to suggest the proportional continuity principle as an underlying mechanism for building analogical learners. Their algebraic framework for defining analogical proportions allows to consider “viewing : reviewer :: searching : researcher” as a valid proportion, generalizing the work of [13]. Their approach provide satisfactory results in the field of natural language processing [25].

We note that all the previous works are focused on discrete data and none of them tackles the issue of dealing with numerical values.

Table 4
Description of datasets.

Datasets	Instances	Nominal att.	Binary att.	Numerical att.	Classes
Balance	625	4	20	–	3
Car	743	7	21	–	4
Spect	267	–	22	–	2
Voting	435	–	16	–	2
Monk1	432	6	15	–	2
Monk2	432	6	15	–	2
Monk3	432	6	15	–	2
Diabetes	768	–	–	8	2
W. B. Cancer	699	–	–	9	2
Heart	270	–	–	13	2
Iris	150	–	–	4	3
Wine	178	–	–	13	3
Satellite Image	346	–	–	36	6
Glass	214	–	–	9	7
Ecoli	336	–	–	7	8

5.2. Numerical case

As far as we know, there are only few works that apply logical proportion for classification of numerical data. The work presented in [22] is the first one to deal with such type of data. Starting from datasets coming from UCI repository [14], the data are normalized in order to get values in [0, 1] considered as truth degrees which allows the application of the graded semantics previously described in this paper. Given a new data $\vec{d}$ to be classified, the main idea is to consider *all* the triples $(\vec{a}, \vec{b}, \vec{c})$ such that the corresponding class equation is solvable. Actually, these triples are the only ones able to provide a prediction for the unknown label of $\vec{d}$. We compute for each of these triples the vector of truth values

$$(a_1 : b_1 : c_1 : d_1, \ldots, a_i : b_i :: c_i : d_i, \ldots, a_n : b_n :: c_n : d_n)$$

Then we order these vectors of truth values using the leximin[2] as a total order. The best triple, i.e. the one maximizing $(a_1 : b_1 : c_1 : d_1, \ldots, a_i : b_i :: c_i : d_i, \ldots, a_n : b_n :: c_n : d_n)$ is chosen to allocate a label to the new item $\vec{d}$. As highlighted in [22], the accuracy results of the corresponding classifier are quite good, and in some cases, outperform well-known algorithms.

6. Experimentations and discussion

In this section, we provide experimental results for the heterogeneous proportion-based classifiers presented above and we run our tests for different values for each parameter.

6.1. Datasets, protocols and other classifiers for comparison

The experimental study is based on several datasets taken from the U.C.I. machine learning repository [14]. A brief description of these data sets is given in Table 4. Since oddness-based classifier is able to deal with both Boolean and numerical features, Table 4 includes 7 datasets with Boolean attribute values (in the first part of this Table) and 8 datasets with only numerical features (in the second part). In terms of classes, we deal with a maximum number of 8 classes. In order to apply our Boolean and multiple-valued semantics framework, all discrete attributes are binarized and all numerical attributes are normalized. More precisely:

- For all categorical (non-binary) attributes where the range of attribute values is finite and strictly greater than 2, we apply the following procedure to convert them into Boolean attributes. Considering an attribute domain $v_1, \ldots, v_m$, we binarize it by means of m properties "having or not value v_i". For instance, a tri-valued attribute having candidate values v_1, v_2, v_3, can respectively be encoded as 100, 010, 001. It means, in that case, that, e.g., 110 does not represent a value and will never appear in the dataset.
- Regarding the numerical attributes, we just replace the value r with $\frac{r-r_{min}}{r_{max}-r_{min}}$, where r_{min} and r_{max} respectively represent the minimal and the maximal values for this attribute on this dataset. A real value is thus changed into a number that may be understood as a truth value.

[2] $(u_1, \ldots, u_i, \ldots, u_n) >_{leximin} (v_1, \ldots, v_i, \ldots, v_n)$, *once the components of each vector have been increasingly ordered*, iff $\exists j < n\ \forall i = 1, j\ u_i = v_i$ and $u_{j+1} > v_{j+1}$.

Table 5
Results of other classifiers on the benchmarks (Left part: classical ones, right part: analogical ones).

Datasets	C4.5	SVM		JRip	IBK ($k = 1$, $k = 10$)	Analogy1 [3]	Analogy2 [4] (Algo2:A, $k = 11$)	WAPC
		Poly-kernel	PUK-kernel					
Balance	78	90	89	76	83, 83	87	–	86
Car	95	91	87	91	92, 92	94	–	n/a
Voting	96	96	96	95	92, 93	78	–	n/a
Spect	81	81	83	81	75, 81	41	–	79
Monk1	99	75	100	98	100, 100	99	–	98
Monk2	95	67	67	73	44, 64	99	–	100
Monk3	100	100	100	100	100, 99	99	–	96
Diabetes	74	77	77	76	70, 71	–	73	–
Cancer	96	97	96	96	96, 97	–	97	–
Heart	77	84	81	81	75, 81	–	82	–
Iris	96	96	96	95	95, 96	–	97	–
Wine	94	98	99	93	95, 95	–	98	–
Sat. Image	94	94	95	93	95, 94	–	94	–
Glass	66	58	71	69	70, 64	–	72	–

In terms of protocol, we apply a standard 10 fold cross-validation technique to build the training and testing sets.
For nominal datasets:
– `Balance` and `Car` are multiple classes databases.
– `Voting`, `Spect`, `Monk1`, `Monk2`, `Monk3` data sets are binary class problems.
`Monk3` has noise added (in the training set only).
– `Voting` and `Spect` data sets contain only binary attributes. `Voting` dataset has missing attribute values.
For numerical datasets:
– `Iris`, `Wine`, `Sat.Image`, `Glass` and `Ecoli` data sets are multiple class problems.
– `Diabetes`, `Cancer` and `Heart` are binary class databases.

This experimental study is divided into two parts. In the first subsection, we evaluate the Oddness-based classifier in the case of *Boolean* and *numerical* data and we run our tests on 4 different sizes of subsets: subsets of one, two, tree or four items to compute the oddness measure, leading to algorithms Odd_1, Odd_2, Odd_3, Odd_4. These classifiers are also tested for diverse values of k.

In the second subsection, we test the efficiency of Evenness-based classifier to deal with *Boolean* datasets.

In order to evaluate the efficiency of Oddness/Evenness classifiers, we compare their accuracy to existing classification approaches. Table 5 includes classification results of some machine learning algorithms:

- **C4.5**: generating a pruned or unpruned C4.5 decision tree.
- **SVM**: a sequential minimal optimization algorithm for training a support vector classifier. We use two types of kernels: the Polynomial kernel and the Pearson VII function-based universal kernel denoted respectively Poly-Kernel and PUK-Kernel.
- **JRip**: propositional rule learner, Repeated Incremental Pruning to Produce Error Reduction (RIPPER), optimized version of IREP.
- **IBk**: a k-Nearest-neighbor classifier with normalized Euclidean distance with $k = 1$, $k = 10$.

Accuracy results for C4.5, SVMs, JRip and IBk given in Table 5 are obtained by using the free implementation of Weka software to the datasets described in Table 4.

The columns Analogy1, Analogy2 and WAPC in Table 5 refer to the results obtained respectively with analogy-based classifiers, reviewed in the previous Section 5, in the case of Boolean data [3], in the case of numerical data [4], and with the weighted analogical classifier (using analogical dissimilarity) presented in [15].

The comparative studies with existing classifiers of the oddness classifiers and the evenness classifiers, given in the following two subsections, are carried out through Signed-Ranks Test as proposed by Demsar [9]. They are parametric tests that check if the difference between the results of two classifiers over various datasets is significant enough [9].

6.2. Results for oddness-based classifiers

In Tables 6 and 7, we provide mean accuracies and standard deviations obtained with the three first implemented options using Odd_1, Odd_2 and Odd_3. For Odd_2 and Odd_3 alternatives, we also test different values of k (k being the number of nearest neighbors used). Let us note that when we have less than k elements in a given class, we do not check the version of our algorithm for the value k: this is the case for *Glass* and $k = 11$.

Table 8 shows classification results obtained with Odd_4. Since this is a time consuming option, we restrict our tests to datasets to small size.

Table 6
Classification accuracies given as mean and standard deviation with Odd_1 and Odd_2.

Datasets	Odd_1	Odd_2			
Value of k		1	3	5	11
Balance	$83{,}67 \pm 3{,}82$	$49{,}81 \pm 6{,}39$	$76{,}93 \pm 5{,}02$	$\mathbf{87{,}34 \pm 3{,}17}$	$86{,}29 \pm 3{,}45$
Car	$57{,}89 \pm 7{,}73$	$83{,}99 \pm 4{,}10$	$87{,}34 \pm 3{,}25$	$\mathbf{91{,}72 \pm 2{,}81}$	$91{,}04 \pm 2{,}91$
Spect	$44{,}02 \pm 6{,}63$	$78{,}68 \pm 6{,}96$	$83{,}72 \pm 6{,}17$	$83{,}11 \pm 6{,}06$	$83{,}19 \pm 5{,}27$
Voting	89.13 ± 5.34	90.26 ± 3.59	92.68 ± 3.11	93.75 ± 3.14	$\mathbf{94.74 \pm 2.97}$
Monk1	$75{,}01 \pm 6{,}53$	$99{,}12 \pm 1{,}49$	$99{,}77 \pm 0{,}51$	$\mathbf{99{,}86 \pm 0{,}33}$	$99{,}68 \pm 1{,}29$
Monk2	$50{,}74 \pm 9{,}11$	$34{,}52 \pm 6{,}70$	$36{,}57 \pm 6{,}21$	$43{,}37 \pm 4{,}70$	$\mathbf{58{,}98 \pm 4{,}13}$
Monk3	$97{,}23 \pm 1{,}78$	$99{,}96 \pm 0{,}13$	**100**	**100**	**100**
Diabetes	$74{,}85 \pm 4{,}39$	$69{,}95 \pm 4{,}16$	$73{,}81 \pm 4{,}42$	$74{,}28 \pm 4{,}70$	$75{,}73 \pm 3{,}77$
W. B. Cancer	$94{,}23 \pm 2{,}58$	$96{,}80 \pm 1{,}69$	$97{,}2 \pm 1{,}66$	$\mathbf{97{,}43 \pm 1{,}76}$	$97{,}31 \pm 1{,}71$
Heart	$\mathbf{83{,}18 \pm 7{,}74}$	$78{,}89 \pm 7{,}63$	$81{,}70 \pm 6{,}87$	$82{,}22 \pm 6{,}53$	$81{,}85 \pm 6{,}81$
Iris	$94{,}53 \pm 6{,}15$	$93{,}87 \pm 5{,}41$	$94{,}67 \pm 4{,}96$	$94{,}80 \pm 4{,}81$	$95{,}06 \pm 4{,}55$
Wine	$93{,}25 \pm 5{,}59$	$97{,}97 \pm 3{,}44$	$97{,}98 \pm 2{,}87$	$\mathbf{98{,}34 \pm 2{,}53}$	$97{,}52 \pm 3{,}21$
Sat. Image	$87{,}15 \pm 2{,}95$	$\mathbf{95{,}17 \pm 1{,}77}$	$95{,}06 \pm 2{,}16$	$95{,}04 \pm 1{,}97$	$94{,}33 \pm 2{,}12$
Glass	$35{,}93 \pm 9{,}51$	$75{,}15 \pm 8{,}00$	$\mathbf{76{,}53 \pm 7{,}71}$	$74{,}77 \pm 7{,}62$	–

Table 7
Classification accuracies given as mean and standard deviation with Odd_3.

Datasets	Odd_3			
Value of k	1	3	5	11
Balance	$52{,}05 \pm 6{,}98$	$74{,}99 \pm 3{,}59$	$87{,}16 \pm 2{,}71$	$86{,}63 \pm 2{,}90$
Car	$83{,}96 \pm 3{,}67$	$86{,}90 \pm 3{,}64$	$91{,}12 \pm 3{,}26$	$89{,}57 \pm 3{,}07$
Spect	$80{,}97 \pm 6{,}80$	$84{,}13 \pm 4{,}99$	$\mathbf{84{,}30 \pm 4{,}34}$	$\mathbf{84{,}30 \pm 4{,}35}$
Voting	88.68 ± 6.25	91.55 ± 3.72	92.18 ± 5.65	94.24 ± 5.02
Monk1	$99{,}63 \pm 0{,}71$	$97{,}15 \pm 3{,}46$	$98 \pm 2{,}46$	$91{,}99 \pm 6{,}40$
Monk2	34.28 ± 7.29	37.05 ± 6.44	41.91 ± 7.57	55.32 ± 7.21
Monk3	**100**	**100**	99.77 ± 0.68	99.32 ± 2.05
Diabetes	$70{,}31 \pm 4{,}06$	$74{,}03 \pm 3{,}75$	$74{,}55 \pm 4{,}25$	$\mathbf{76{,}41 \pm 4{,}32}$
W. B. Cancer	$96{,}10 \pm 2{,}35$	$97{,}03 \pm 1{,}94$	$97{,}08 \pm 1{,}87$	$97{,}03 \pm 1{,}85$
Heart	$77{,}63 \pm 7$	$81{,}26 \pm 5{,}63$	$82 \pm 6{,}75$	$82{,}44 \pm 6{,}28$
Iris	$94{,}93 \pm 5{,}22$	$94{,}79 \pm 4{,}81$	$94{,}66 \pm 4{,}76$	$\mathbf{95{,}73 \pm 5{,}07}$
Wine	$95{,}77 \pm 4{,}35$	$97{,}4 \pm 3{,}56$	$96{,}58 \pm 3{,}92$	$96{,}48 \pm 4{,}25$
Sat. Image	$94{,}12 \pm 1{,}90$	$94{,}18 \pm 2{,}08$	$94{,}18 \pm 2{,}30$	$93{,}51 \pm 2{,}36$
Glass	$70{,}12 \pm 6{,}06$	$74{,}26 \pm 6{,}42$	$72{,}44 \pm 7{,}23$	–

Table 8
Classification accuracies given as mean and standard deviation obtained with Odd_4.

Datasets	Odd_4			
Value of k	1	3	5	11
Spect	81.28 ± 5.2	$\mathbf{84.2 \pm 5.11}$	$\mathbf{84.2 \pm 3.89}$	$\mathbf{84.2 \pm 3.89}$
Monk1	99.55 ± 0.91	96.28 ± 4.31	90.07 ± 4.37	–
Heart	71.48 ± 7.95	78.89 ± 4.7	79.26 ± 5.79	80.37 ± 5.25
Iris	$95 \pm 5{,}68$	$95{,}67 \pm 5{,}01$	$95{,}50 \pm 4{,}98$	$\mathbf{96{,}34 \pm 4{,}84}$
Wine	$92{,}89 \pm 6{,}15$	$94{,}58 \pm 5{,}75$	$94{,}20 \pm 5{,}52$	$94{,}83 \pm 4{,}75$

6.2.1. Behavior of the different oddness classifiers and comparison with classical ones

In Tables 6, 7 and 8, we notice that:

- Odd_1 seems to be significantly less efficient than all other subset sizes for most data-sets. The worst accuracy for this option is noted for datasets: Car, Spect, Sat.Image, Wine and Glass having large number of attributes and/or classes. In fact, this option remains close to the basis of k-nn algorithm since both compute the distance to the training examples in an independent way without any further investigation on the relationship between these training data. Moreover, since this option computes the mean oddness measure to elements of classes, this makes it less informative than other options.
- For most datasets, best results are obtained with large values of k ($k = 5$ or 11) for the three alternatives using Odd_2, Odd_3 or Odd_4, except in the case of Monk1 where small values of k provide better accuracy for Odd_3 and Odd_4. Since subsets of pairs are generally less informative than subsets of triples or quadruples, it is better to consider, for this option, large values of k to take advantage of a larger variety of data. It remains to investigate what is the suitable k for a target dataset.
- If we compare Odd_2 to the other Odd_i's for $k = 5$, we note that this option provides the best accuracy for all datasets except for Spect., where it performs slightly worse than Odd_3 and Odd_4.

Table 9
Results for the Wilcoxon Matched-Pairs Signed-Ranks Test, the * means that the classifier in the row in statistically better than the classifier on the column.

		Odd_1	Odd_3 $(k=5)$	IB1	IB10
Odd_1	Without Monk2	–	–	**0.049**	**0.022**
	With Monk2	–	–	0.1	**0.013**
Odd_2 $(k=5)$	Without Monk2	**0.0076***	0,061	**0.026***	**0.0229***
	With Monk2	**0.023***	**0,034***	**0.034***	0.1158
Odd_3 $(k=5)$	Without Monk2	**0.0076***	–	0.091	0.1823
	With Monk2	**0.027***	–	0.136	0.463

- Odd_4 performs generally worse than Odd_2 and Odd_3 for all datasets expect for the Iris where it is slightly better. Especially for datasets Heart and Wine, the accuracy strongly decreases with Odd_4. This may reinforce the intuition that pairs and triples are appropriate to evaluate oddness.
- It is quite clear that the proposed classifier, especially Odd_2, is able to classify numerical as well as Boolean data sets almost in the same way. These results highlight that the proposed multi-valued oddness measure correctly extends the Boolean case.
- The comparison with Table 5 highlights the fact that oddness-based classifiers perform more or less in the same way as the best known algorithms. Especially, Odd_2 performs similarly well as other classifiers for data sets Monk1, Cancer, Sat. Image and Glass, has performances similar to SVM for datasets Car and Spect. For Monks3, Odd_2 behaves as most of ML classifiers.
- Odd_2 shows high efficiency to classify datasets Balance, Car, Sat.Image and Glass (which have multiple classes) which demonstrates its ability to deal with multiple class data sets.
- The oddness-based classifier seems to be also efficient when classifying data sets with a large number of instances and attributes as in the case of Car and Sat.Image for instance.

Results concerning optimized variants of Odd_2 classifier can be found in [8].

6.2.2. Comparison with analogical classifiers

We also note that:

- If we compare the best results obtained with Odd_2 in Table 6 with those obtained with the analogy-based classifier for numerical data [4], we can notice that the two classifiers perform similarly for most datasets, with maybe the exception of Iris. In that latter case, the analogy is significantly better, while for Diabetes the converse is observed.
- We notice that both oddness-based and analogy-based classifiers Analogy1 [3] in the case of Boolean datasets, exhibit good results for Balance, Car, Monk1 and Monk3, comparable to those obtained by classifiers like IBK or SVM. The results of oddness-based classifier Odd_2 are also comparable to those of the evenness-based classifier [6]; see also next subsection 6.3.
- Regarding Monk2, where analogical proportion-based classifiers perform very well, it is known that the underlying function ("having exactly two attributes with value 1") is more complicated than the functions underlying Monk1 and Monk3, and involves all the attributes (while in the two other functions only 3 attributes among 6 are involved in the discrete coding). We suspect that the existence of a large discontinuity in the classification of data (a nearest neighbor y of x will not generally be labeled with the same class $cl(\vec{x})$) may be too difficult to apprehend using oddness (or heterogeneous proportions Evenness in Table 5). Moreover, for this dataset, we expect that the classifier needs to consider more neighbors k to get better results. Thus, we also tested the approach using pairs with bigger values of k for Monk2 data set, we get an accuracy equal to 64.83 ± 2.06 for $k=17$.

6.2.3. Comparison with nearest neighbors classifiers

Lastly, on Table 5, we also observe that Odd_2 with $k=5$ significantly outperforms IBK on datasets Balance, Spect., Diabetes, Heart, Wine, Sat. Image and Glass and has similar results for Monk1, Monk3. This is confirmed by the Wilcoxon Matched-Pairs Signed-Ranks Test [9]. This test is a non-parametric alternative to the paired t-test enabling to compare two classifiers over multiple data sets. In our case, the null hypothesis states that the two compared algorithms performs in the same way. Table 9 summarizes the results of the computed p-values for each pair of compared classifiers. The null hypothesis has to be rejected when the p-value is *less* than the threshold 0.05. These values are highlighted in bold in Table 9. We add a * to each significant p-value (<0.05) if the classifier given in the row significantly outperforms the classifier given in the column. There is no * for any significant p-value if the classifier given in the column is rather statistically better than the classifier given in the row. From the computed p-values, we can draw the following conclusions:

- As expected, Odd_1 is statistically less efficient than IBK, Odd_2 and Odd_3.
- If we compare with Odd_3, the p-value confirms that Odd_2 is more efficient than Odd_3.

Table 10
Classification accuracies given as mean and standard deviation obtained with $Even$.

Datasets		$Even$			
Value of k		1	3	5	11
Balance	$l = n$	67.29 ± 5.2	71.48 ± 6.49	76.43 ± 4.85	78.23 ± 4.57
	$l = n - 1$	78.04 ± 4.91	83.28 ± 3.44	$\mathbf{87.08 \pm 3.22}$	86.48 ± 3.45
Car	$l = n$	92.6 ± 2.87	92.84 ± 2.82	93.05 ± 2.83	$\mathbf{93.27 \pm 2.7}$
	$l = n - 1$	89.63 ± 3.43	90.35 ± 3.0	91.58 ± 2.52	91.75 ± 2.42
Spect	$l = n$	81.53 ± 6.67	81.86 ± 7.93	$\mathbf{82.61 \pm 8.16} \pm$	82.32 ± 8.54
	$l = n - 1$	81.21 ± 6.52	81.14 ± 6.83	81.37 ± 6.82	81.74 ± 7.1
Voting	$l = n$	94.29 ± 3.67	94.99 ± 3.96	94.94 ± 3.96	$\mathbf{95.12 \pm 3.71}$
	$l = n - 1$	94.25 ± 3.94	94.95 ± 4.24	94.9 ± 4.24	94.99 ± 3.93
Monk1	$l = n$	**100**	**100**	**100**	**100**
	$l = n - 1$	**100**	99.95 ± 0.05	99.91 ± 0.64	99.95 ± 0.05
Monk2	$l = n$	38.31 ± 4.09	41.37 ± 4.66	45.54 ± 5.04	$\mathbf{50.68 \pm 4.3}$
	$l = n - 1$	30.87 ± 5.85	34.14 ± 4.46	37.46 ± 4.91	42.61 ± 5.23
Monk3	$l = n$	**100**	**100**	**100**	**100**
	$l = n - 1$	99.77 ± 0.71	99.22 ± 1.94	98.76 ± 2.42	98.49 ± 2.76

- Odd_2 is also significantly better than IB1. Our proposed algorithm statistically outperforms IB10 *only* if we remove Monk2 from the list of compared datasets for the Wilcoxon Ranks test (note that Odd_2 performs as IB10 for $k = 17$).

6.3. *Results for evenness-based classifiers*

In order to better control the meaning of $EVEN(\mathcal{C}, \vec{x})$, we may focus on triples for which $\vec{x}$ is an intruder for at most $n - l$ features, where $l = 0, 1, \cdots$. Instead of keeping all the triples, we can just choose a threshold $l \in [0, n]$, then consider $Even(\vec{a}, \vec{b}, \vec{c}, \vec{x})$ only for the triples $(\vec{a}, \vec{b}, \vec{c})$ in $\mathcal{C}^3$ such that

$$Even(\vec{a}, \vec{b}, \vec{c}, \vec{x}) \geq l,$$

i.e. we want $Even$ to hold over at least l features. We denote $EVEN^l(\mathcal{C}, \vec{x})$ this measure where we just reduce the number of candidate triples by filtering over l. As a consequence, the evenness-based algorithms have 2 parameters: k the number of considered nearest neighbors and l the minimum number of features where the $Even$ proportion should hold.

Table 10 provides accuracies results for the evenness-based classifier obtained with a 10-fold cross validation and for two values of k and l (k being the number of nearest neighbors of $\vec{x}$, l refers to the number of attributes j of d such that x_j belongs to a majority). The best results are in bold.

6.3.1. *Behavior of the different evenness classifiers and comparison with classical ones*

When we analyze results in Table 10, we can see that:

• In general, the best classification rates are obtained for $l = n$. This means that the classifier is likely to be more accurate when the classification is made on the basis of triples w.r.t. which $\vec{x}$ is not an intruder for *any* attributes. However, for some datasets such as Balance and Monk2, the classifier needs to consider more levels l when it is difficult to satisfy the constraint $Even(\vec{a}, \vec{b}, \vec{c}, \vec{x}) \geq l$ for $l = n$ or even $l = n - 1$. Thus, we also tested smaller levels of l and for "Balance" data set, we get an accuracy equal to 89.25 ± 2.4 for $l = n - 3$.

• The classifier shows good classification results for data sets "Balance", and "Car" (which have multiple classes). This shows that evenness-based classifiers are able to deal with multiple class data sets.

• If we compare results of the evenness-based classifier with machine learning algorithms in Table 5, we note that the proposed classifier is as good as the best known algorithms. Especially, the basic classifier, with large k works as well as any other classifiers for data sets "Balance" (for $l = n - 3$), "Spect.", "Monk1" and "Monk3" (for $l = n$). Moreover, evenness-based classifier outperforms IBK for *all* data sets except Monk2.

6.3.2. *Comparison with analogical classifiers*

• Both evenness-based classifiers and the analogy-based classifier [3], exhibit very good results for data sets "Balance", "Car", "Monk1" and "Monk3".

• However, as in the case of oddness-based classifiers, the evenness-based classifier is also less efficient when classifying "Monk2" data set. As said before for this dataset, we expect that the classifier needs to consider more neighbors k to achieve better accuracy.

• On the contrary, it is clear that evenness-based classifier outperforms the analogy-based classifier [3] for data sets "Spect" and "Voting" (see Table 5). From experiments, we notice that bad results for analogy-based classifier with "Spect" and "Voting" datasets seem to be due to the number of voters $(\vec{a}, \vec{b})$ which is equal to 0 for many examples to be classified.

Regarding the analogy-based classifiers, when considering a particular item $\vec{x}$, and a neighbor $\vec{c} \in \mathcal{B}_H(\vec{x}, r)$ (where $\mathcal{B}_H(\vec{x}, r)$ denotes the Hamming ball with center $\vec{x}$ and radius r) the number of voters $(\vec{a}, \vec{b})$ is only a small subset of the set of pairs differing on r attributes. Due to the fact that two constraints have to be satisfied in the analogical proportion-based approach: the pairs $(\vec{a}, \vec{b})$ and $(\vec{c}, \vec{x})$ differ on the same attribute(s) and the associated class equation should be solvable, if only one attribute in the pair $(\vec{a}, \vec{b})$ is not satisfied, this pair will be discarded.

In order to reduce the effect of the first constraint in the analogy-based classifier [3], we reimplemented the analogy-based classifier for *numerical* data described in [4] on the datasets "Spect" and "Voting" (this algorithm seeks for only triples which form with $\vec{x}$ an analogy on a *maximum* number of attributes, and not necessarily on *all* attributes as the algorithm used in [3]). We obtained an accuracy respectively equal to 73.38 ± 4.68 and 95.85 ± 3.09 (using the function: A and $k = 11$). This accuracy improvement shows that, for some datasets whose attributes are highly dissimilar (the case of "Spect" for example), it is faithful to relax the constraint "the pairs $(\vec{a}, \vec{b})$ and $(\vec{c}, \vec{d})$ differ on the same attribute(s)" by satisfying the analogical proportion only on a *maximum* (as it is the case in [4] and in evenness-based classifiers) instead of *all* attributes.

6.3.3. *Comparison between oddness- and evenness-based classifiers*

Although oddness and evenness indexes are not the exact opposite of each other, the results obtained by minimization in oddness-based classifiers and by maximization in evenness-based classifiers are quite close, as can be seen by comparing Tables 6 and 7 with Table 10, for Boolean features. However, the evenness index and measure have not been defined in the case of numerical features.

7. Conclusion

Many successful approaches have been proposed for classification purposes for a long time. Quite surprisingly, it has been recently shown that it was also possible to build another kind of classifiers based on analogical proportions. The fact that analogical proportions belong to a larger class of formulas, namely logical proportions, including heterogeneous proportions, led us to wonder if this latter kind of proportions might also be used with success to build classifiers. We have investigated this option in our paper.

We have first contrasted the two types of proportion in a formal way: homogeneous proportions, including analogical proportion, on the one hand, and heterogeneous proportions on the other hand. Then, starting from heterogeneous proportions, we have established a way to define an oddness measure and an evenness measure in order to estimate to what extent a new item does not conform, or conforms, to a candidate class. Then, testing on classical benchmarks coming from UCI repository, we have compared an oddness-based classifier and an evenness-based classifier with standards methods in classification, as well as with analogical proportion-based classifiers. Our experiments empirically highlight the good behavior of heterogeneous logical proportion-based classifiers.

As this paper is essentially devoted to classification, once we have explained where the oddness and the evenness evaluations come from, we have mainly focused on empirical results. We have discussed elementary properties of the building blocks of these measures, in order to make sure that they carry suitable semantics for a classification purpose. Nevertheless, we have seen that, regarding oddness measure, Odd_2 classifier (based on ODD_2 measure) stands out of the crowd. In fact, if we are back to the basic brick Odd_2 of ODD_2 measure, it is clear that, when $|S| = 2$, $Odd(S, x) = 0$ as soon as $x \in S$. This is not generally the case as soon as $|S| > 2$. This suggests that ODD_2, built up on the sum of atomic $Odd(S, x)$ with $|S| = 2$, may be a better marker of the oddness of a given element x inside a class $\mathcal{C}$ than any other measure ODD_i with $i > 2$. Still the global oddness and evenness measures have no obvious remarkable properties. This question may be addressed in future works.

Apart from a formal investigation of the properties of oddness and evenness measures, their merits would need to be studied in greater detail in order, for instance, to more precisely assessed the expected accuracy of oddness or evenness-based classifiers. Then, one might think to use them in conformal predictors [24,27,28], as first experimented in [6] with an evenness-based classifier. Indeed the oddness measure may be considered as a non-conformity measure, while the evenness measure would be a conformity measure.

References

[1] S. Bayoudh, L. Miclet, A. Delhay, Learning by analogy: a classification rule for binary and nominal data, in: Proc. Inter. Joint Conf. on Artificial Intelligence, IJCAI'07, 2007, pp. 678–683.

[2] M. Bounhas, Kh. Mellouli, H. Prade, M. Serrurier, Possibilistic classifiers for numerical data, Soft Comput. 17 (5) (2013) 733–751.

[3] M. Bounhas, H. Prade, G. Richard, Analogical classification: a new way to deal with examples, in: T. Schaub, G. Friedrich, B. O'Sullivan (Eds.), Proc. 21st Europ. Conf. on Artificial Intelligence, ECAI'14, Prague, Aug. 18–22, in: Frontiers in Artificial Intelligence and Applications, vol. 263, IOS Press, 2014, pp. 135–140.

[4] M. Bounhas, H. Prade, G. Richard, Analogical classification: handling numerical data, in: U. Straccia, A. Calì (Eds.), Proc. 8th Int. Conf. on Scalable Uncertainty Management, SUM'14, Oxford, Sept. 15–17, in: LNCS, vol. 8720, Springer, 2014, pp. 66–79.

[5] M. Bounhas, H. Prade, G. Richard, Evenness-based reasoning with logical proportions applied to classification, in: C. Beierle, A. Dekhtyar (Eds.), Proc. 9th Int. Conf. on Scalable Uncertainty Management, SUM'15, Quebec City, Sept. 16–18, in: LNCS, vol. 9310, Springer, 2015, pp. 139–154.

[6] M. Bounhas, H. Prade, G. Richard, A new view of conformity and its application to classification, in: L.C. van der Gaag (Ed.), Proc. 13th Eur. Conf. on Symb. and Quantit. Appr. to Reasoning with Uncertainty, ECSQARU'15, Compiègne, Jul. 15–17, in: LNAI, vol. 9161, Springer, 2015, pp. 221–232.

[7] M. Bounhas, H. Prade, G. Richard, Oddness-based classifiers for Boolean or numerical data, in: S. Hölldobler, M. Krötzsch, R. Peñaloza, S. Rudolph (Eds.), Advances in Artificial Intelligence, Proc. 38th Annual German Conference on AI, KI'15, Dresden, Sept. 21–25, in: LNCS, vol. 9324, Springer, 2015, pp. 32–44.

[8] M. Bounhas, H. Prade, G. Richard, Not being at odds with a class: a new way of exploiting neighbors for classification, in: G.A. Kaminka, M. Fox, P. Bouquet, E. Hüllermeier, V. Dignum, F. Dignum, F. van Harmelen (Eds.), Proc. 22nd Europ. Conf. on Artificial Intelligence, ECAI'16, the Hague, 29 Aug.–2 Sept., in: Frontiers in Artificial Intelligence and Applications, vol. 285, IOS Press, 2016, pp. 1662–1663.

[9] J. Demsar, Statistical comparisons of classifiers over multiple data sets, J. Mach. Learn. Res. 7 (2006) 1–30.

[10] J.D. Gergonne, Application de la méthode des moindres quarrés à l'interpolation des suites, Ann. Math. Pures Appl. 6 (1815) 242–252.

[11] J.D. Gergonne, Théorie de la règle de trois, Ann. Math. Pures Appl. 7 (1816) 117–122.

[12] Y. Lepage, Analogy and formal languages, in: Proc. FG/MOL 2001, 2001, pp. 373–378, see also http://www.slt.atr.co.jp/~lepage/pdf/dhdryl.pdf.gz.

[13] Y. Lepage, Analogy and formal languages, Electron. Notes Theor. Comput. Sci. 53 (2004) 180–191.

[14] J. Mertz, P.M. Murphy, UCI repository of machine learning databases, available at ftp://ftp.ics.uci.edu/pub/machine-learning-databases, 2000.

[15] L. Miclet, S. Bayoudh, A. Delhay, Analogical dissimilarity: definition, algorithms and two experiments in machine learning, J. Artif. Intell. Res. 32 (2008) 793–824.

[16] L. Miclet, H. Prade, Handling analogical proportions in classical logic and fuzzy logics settings, in: Proc. 10th Eur. Conf. on Symbolic and Quantitative Approaches to Reasoning with Uncertainty, ECSQARU'09, Verona, in: LNCS, vol. 5590, Springer, 2009, pp. 638–650.

[17] H. Prade, G. Richard, Reasoning with logical proportions, in: F.Z. Lin, U. Sattler, M. Truszczynski (Eds.), Proc. 12th Int. Conf. on Principles of Knowledge Representation and Reasoning, KR 2010, Toronto, May 9–13, 2010, AAAI Press, 2010, pp. 545–555.

[18] H. Prade, G. Richard, Homogeneous logical proportions: their uniqueness and their role in similarity-based prediction, in: G. Brewka, T. Eiter, S.A. McIlraith (Eds.), Proc. 13th Int. Conf. on Principles of Knowledge Representation and Reasoning, KR'12, Roma, June 10–14, AAAI Press, 2012, pp. 402–412.

[19] H. Prade, G. Richard, From analogical proportion to logical proportions, Log. Univers. 7 (4) (2013) 441–505.

[20] H. Prade, G. Richard, Homogenous and heterogeneous logical proportions, IfCoLog J. Log. Appl. 1 (1) (2014) 1–51.

[21] H. Prade, G. Richard, On different ways to be (dis)similar to elements in a set. Boolean analysis and graded extension, in: J.P. Carvalho, M.-J. Lesot, U. Kaymak, S.M. Vieira, B. Bouchon-Meunier, R.R. Yager (Eds.), Proc. 16th Int. Conf. on Information Processing and Management of Uncertainty in Knowledge-Based Systems, Part II, IPMU'16, Eindhoven, June 20–24, in: Commun. Comput. Inf. Sci., vol. 611, Springer, 2016, pp. 605–618.

[22] H. Prade, G. Richard, B. Yao, Enforcing regularity by means of analogy-related proportions – a new approach to classification, Int. J. Comput. Inf. Syst. Ind. Manag. Appl. 4 (2012) 648–658.

[23] N. Rescher, Many-Valued Logic, McGraw-Hill Inc., New York, 1969.

[24] C. Saunders, A. Gammerman, V. Vovk, Computationally efficient transductive machines, in: H. Arimura, S. Jain, A. Sharma (Eds.), Proc. of Algorithmic Learning Theory, 11th International Conference, ALT 2000, Sydney, Australia, December 11–13, in: LNCS, vol. 1968, Springer, 2000, pp. 325–333.

[25] N. Stroppa, F. Yvon, An analogical learner for morphological analysis, in: Online Proc. 9th Conf. Comput. Natural Language Learning, CoNLL-2005, 2005, pp. 120–127.

[26] N. Stroppa, F. Yvon, Analogical learning and formal proportions: definitions and methodological issues, Technical report, June 2005.

[27] V. Vovk, Cross-conformal predictors, Ann. Math. Artif. Intell. (2015) 9–28.

[28] V. Vovk, A. Gammerman, C. Saunders, Machine-learning applications of algorithmic randomness, in: Int. Conf on Machine Learning, 1999, pp. 444–453.