ECAI 2016
G.A. Kaminka et al. (Eds.)

doi:10.3233/978-1-61499-672-9-689

Analogical Classifiers: A Theoretical Perspective

Nicolas Hug[1] and **Henri Prade**[1,2] and **Gilles Richard**[1,3] and **Mathieu Serrurier**[1]

Abstract. In recent works, analogy-based classifiers have been proved quite successful. They exhibit good accuracy rates when compared with standard classification methods. Nevertheless, a theoretical study of their predictive power has not been done so far. One of the main barriers has been the lack of functional definition: analogical learners have only algorithmic definitions. The aim of our paper is to complement the empirical studies with a theoretical perspective. Using a simplified framework, we first provide a concise functional definition of the output of an analogical learner. Two versions of the definition are considered, a strict and a relaxed one. As far as we know, this is the first definition of this kind for analogical learner. Then, taking inspiration from results in k-NN studies, we examine some analytic properties such as convergence and VC-dimension, which are among the basic markers in terms of machine learning expressiveness. We then look at what could be expected in terms of theoretical accuracy from such a learner, in a Boolean setting. We examine learning curves for artificial domains, providing experimental results that illustrate our formulas, and empirically validate our functional definition of analogical classifiers.

1 Introduction

Analogical reasoning is widely recognized as a powerful ability of human intelligence. It can lead to conclusions for new situations by establishing links between apparently unrelated domains. One well known example is the Bohr's model of atom where electrons circle around the kernel, which is analogically linked to the model of planets running around the sun. It is not surprising that this kind of reasoning has generated a lot of attention from the artificial intelligence community. We can cite for instance [12, 13, 16, 30, 17] where the power of analogical reasoning is emphasized. The interested reader may find in [27] a survey of current trends. More recently, using analogy as a basis for the automatic solving of IQ tests [8, 29] or for machine learning tasks [15, 33] got more attention. In the case of classification, analogical classifiers are mainly based on a particular variant of analogy, namely analogical proportions and they have been proved successful [4, 28, 6], at least from an empirical viewpoint.

But analogy, as an essential ingredient of Artificial Intelligence, has also attracted theoretical investigations. In [10], a thorough investigation of analogical reasoning from a first order logic viewpoint has been done, leading to clearly specify safe conditions of usage of the *analogical jump*. More recently, in [14], an higher order logic framework has been developed, providing another logical theory for analogical reasoning in artificial intelligence and cognitive science. Instead of being described as an inference rule, the analogy-making process is described in terms of generalisation and anti-unification. On top of this work, a full implementation has been done leading to the so-called Heuristic-Driven Theory Projection (HDTP).

From another viewpoint, we have to mention the work of [18] which is an attempt to consider analogy-making as a particular case of machine learning where very few data are available. In the limit case, only one pair $(a, f(a))$ (a is the source) is available and one has to guess $f(b)$ for another element b, the target. This work describes a model which minimizes the computational cost of producing $(b, f(b))$ from $(a, f(a))$. This computational cost can be estimated via Kolmogorov complexity [7], a measure which is well-known to be hard to compute (but can be estimated via compression).

Finally, we can also recall the work of [2] where the particular case of analogical proportion is investigated in lattices and other algebraic structures, leading to elegant theoretical results and implementations.

Nevertheless, all these theoretical investigations are not directed to provide an analytical view of analogy-based learners. In that sense, they are not really helpful if we want to characterize the behaviour of an analogical classifier for instance. One of the reasons could be that, unlike the k-NN rule, the analogical learning rule is not easily amenable to a functional definition. In fact, each implemented algorithm provides a clean description of *how to compute* but we definitely miss a clean description of *what do we actually compute*. Since such a definition, even a simplified one, is paramount to investigate theoretical properties, we suggest here a concise functional definition and we prove that it fits with the main implementations of analogical classifiers.

Our paper is organized as follows. In Section 2, we recall the fundamentals about analogical proportions as a particular case of analogy. Then, in Section 3, we explain how such proportions underlie analogical classifiers and the principle of their implementations. Then we provide a unified functional view establishing the formal framework allowing to investigate their mathematical properties. In Section 4, we examine some general properties such as convergence and VC-dimension of analogical learners, considering only minimal constraints on the underlying domain. In Section 5, we investigate, from a probabilistic viewpoint, the expected accuracy of an analogical learner in the Boolean case. We empirically validate our formulas in Section 6 with a complete batch of experiments. We provide our final remarks in Section 7, linking the known results about analogical classifiers with their mathematical properties, noting some limitations of our study and suggesting directions for future research.

2 Analogical proportions

Given a set X, an analogical proportion[4] over X is a quaternary relation A over X satisfying 3 axioms [11, 20]:

1. $\forall a, b,\ A(a, b, a, b)$

[1] IRIT Univ. P. Sabatier, Toulouse, France, email: name.surname@irit.fr
[2] QCIS, University of Technology, Sydney, Australia
[3] BITE, London, UK

[4] For the remaining of this paper, the term *analogy* always means *analogical proportion*.

2. $\forall a, b, c, d,\ A(a, b, c, d) \implies A(c, d, a, b)$ (symmetry)
3. $\forall a, b, c, d,\ A(a, b, c, d) \implies A(a, c, b, d)$ (central permutation)

$A(a, b, c, d)$ is often denoted with infix notation $a : b :: c : d$ when there is no ambiguity over the relation A and its domain X. When dealing with natural language words, the third axiom may be debatable [3]. There are a lot of ways to define an analogical relation over a set X, depending on the available structure and operators.

When $X = \mathbb{R}$, some of the most well known examples are the arithmetic proportion $a : b :: c : d$ iff $a - b = c - d$, and the geometrical proportion $a : b :: c : d$ iff $\frac{a}{b} = \frac{c}{d}$ iff $a * d = b * c$.

When $X = \mathbb{B} = \{0, 1\}$, the previous definitions still work and have a logical translation as [23, 26]:

$$a : b :: c : d \text{ iff } (a \wedge d \equiv b \wedge c) \wedge (a \vee d \equiv b \vee c)$$

In fact, as soon as we have a proportion A over a set X, it is straightforward to build a proportion A^m over X^m with:

$$\forall a, b, c, d \in X^m, A^m(a, b, c, d) \iff \forall i \in [1, m], A(a_i, b_i, c_i, d_i).$$

Still, A^m will often be denoted A when there is no ambiguity. In [22, 31, 21], examples are given where X is equipped with some algebraic structure (words over an alphabet, lattices, sets, Boolean vectors, matrices, etc.).

2.1 Analogical equation

When an analogical proportion is defined on a set X, given 3 elements a, b, c of X and a variable x, a relation $a : b :: c : x$ turns into an equation that we may write $a : b :: c : x = 1$ where we have to find an element $x \in X$ such that the proportion holds. Depending of the set X, the proportion A and a, b, c, one may encounter one of the three situations: the equation is not solvable, has a unique solution, or has multiple solutions. When there is at most one solution, we say that A is *univocal*. For instance:

- When $X = 2^U$ (i.e. X is the powerset of a given universe) and $a : b :: c : x$ is defined as:

  $$(a \cup x = b \cup c) \text{ and } (a \cap x = b \cap c),$$

  the equation is solvable iff $b \cap c \subseteq a \subseteq b \cup c$ and the unique solution is then $x = ((b \cup c) \setminus a) \cup (b \cap c)$.
- With $X = \mathbb{R}^m$ and $a : b :: c : x$ iff $a - b = c - x$, the equation has always a solution $x = c - a + b$. When $m = 2$ and a, b, c, x are considered as points in $\mathbb{R}^2$, it simply means that starting from 3 points, we can always find a fourth one to build up a parallelogram as shown in Figure 1 [26]. In fact, from three non aligned points a, b, c, one can build two other parallelograms which correspond to the following equations: $b : a :: c : x'$ and $c : a :: b : x''$.
- With $X = \mathbb{B}^m$, the previous definition can still be used since $\mathbb{B}^m \subseteq \mathbb{R}^m$. But $\mathbb{B}^m$ is not closed for addition and subtraction, so the equation does not always have a solution in $\mathbb{B}^m$. For instance, the equation $(0, 0) : (0, 1) :: (1, 0) : x$ has a unique solution $(1, 1)$ whereas the equation $(0, 1) : (0, 0) :: (1, 0) : x$ has no solution in $\mathbb{B}^2$ (since in $\mathbb{B}$, $1 : 0 :: 0 : x$ has no solution), despite the fact that there exists a solution $x = (1, -1)$ in $\mathbb{R}^2$. Another way to put it is to say the 4th summit of the parallelogram, which always exists in $\mathbb{R}^m$, does not necessarily belong to $\mathbb{B}^m$.

In the following section, we investigate how the equation solving process can be used as the underlying principle to infer unknown information.

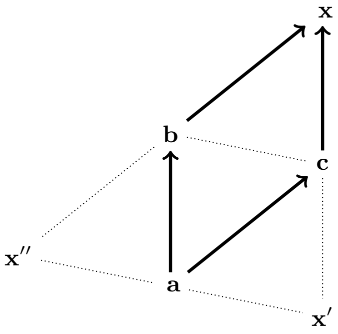


Figure 1. Three parallelograms issued from a, b, c.

2.2 Inference principle

It is recognised that analogical reasoning provides plausible conclusions only [10]. The analogical inference principle can be stated as [31] (where $\vec{a} = (a_1, a_2, \cdots a_n)$):

$$\frac{\forall j \in J \subset [1, n], a_j : b_j :: c_j : d_j}{\forall i \in [1, n] \setminus J, a_i : b_i :: c_i : d_i} \quad (analogical\ inference)$$

In words, this inference principle states that given four vectors $\vec{a}, \vec{b}, \vec{c}, \vec{d}$, if a proportion holds on a sufficient number of components (the J components), then it should also hold for the remaining ones. This principle leads to a prediction rule in the following context:

- 4 vectors $\vec{a}, \vec{b}, \vec{c}, \vec{d}$ are given where $\vec{d}$ is partially known: only the components of $\vec{d}$ with indexes in J are known.
- Using analogical inference, we can predict the missing components of $\vec{d}$ by solving (w.r.t. d_i) the set of equations (in the case they are solvable):

  $$\forall i \in [1, n] \setminus J, \quad a_i : b_i :: c_i : d_i.$$

In the case where the items are such that their last component is just a label, applying this principle to a new element $\vec{d}$ whose label is unknown leads to predict a candidate label for $\vec{d}$. This prediction technique has been successfully applied to classification problems in both Boolean [4, 6] and numerical settings [28].

In the next section, we describe the diverse ways this inference rule is implemented for classification purposes. This leads us to a unified functional definition of analogical classification.

3 Analogical classification

In the context of classification, items are represented as elements of a universe X having a (unique) label belonging to Y. For any $x \in X$, $\dot{x}$ denotes the ground truth label associated to x. The goal of a classifier is, given a sample set S (i.e. a set of elements $x \in X$ for which $\dot{x}$ is known), to correctly predict the label of other elements x that do not belong to the sample set. We call $\hat{x}$ the predicted label of x: this is the output of the classifier.

3.1 Conservative classifier

Let us first consider what we call a *conservative* classifier. Such a classifier is called conservative because it is not able to output a prediction for any x in X, but only for a subset of X. We need to define two crucial concepts: the *analogical extension* of a sample set S, and the *analogical root* of an element x.

Let A be an analogy relation over X and B an analogy relation over Y, the set of labels. The notion of analogical equation allows us to define the so-called *analogical extension* of S denoted as:

$$A_E^Y(S) = \{x \in X | \exists (a,b,c) \in S^3,\ a : b :: c : x \text{ and } \exists y \in Y, \dot{a} : \dot{b} :: \dot{c} : y\}.$$

An intuitive interpretation of $A_E^Y(S)$ is to see it as the set of all $x \in X$ that are solutions of the analogical equations which can be built over the sample set S, provided that the equation related to the associated labels is also solvable. We have the following properties:

1. $S \subseteq A_E^Y(S)$, since $x : x :: x : x$ always holds ;
2. $A_E^Y(\emptyset) = \emptyset$, $A_E^Y(X) = X$;
3. $S_1 \subseteq S_2 \implies A_E^Y(S_1) \subseteq A_E^Y(S_2)$.

The dual concept of the analogical extension is the so-called *analogical root* of a given element $x \in X$, denoted $R_S^Y(x)$:

$$R_S^Y(x) = \{(a,b,c) \in S^3 | a : b :: c : x \text{ and } \exists y \in Y, \dot{a} : \dot{b} :: \dot{c} : y\}$$

$R_S^Y(x)$ is the set of 3-tuples in S which are analogically linked to x and which provide a prediction for the label. It is clear that $R_S^Y(x)$ may contain more than one 3-tuple: for example in R^m, x may be the summit of more than one parallelogram.

For any element x of $A_E^Y(S)$, we define the *analogical label* of x as:

$$\overline{x} = \begin{cases} \dot{x} \text{ if } x \in S \\ \text{Mode}\{y | \dot{a} : \dot{b} :: \dot{c} : y\ \forall (a,b,c) \in R_S^Y(x)\} \text{ if } x \notin S \end{cases}$$

where $\text{Mode}(\Sigma)$ returns the most frequent element of the multiset Σ. In case of a tie, the returned element is chosen at random between the most frequent elements.

The analogical label will be used to estimate the label of every element. Obviously, in the first case, we do not want to change the label of the elements of S. For elements in $A_E^Y(S) \backslash S$ (i.e. the second case), the analogical label is the most frequent label out of all the labels inferred from the solution of the analogical equations that one can build from $R_S^Y(x)$. It is quite clear that, for these elements, we do not necessarily have $\overline{x} = \dot{x}$. To summarize, for a given element $x \in X$, we may potentially associate 3 labels:

- its true label $\dot{x}$;
- in the case where $x \in A_E^Y(S)$, its analogical label $\overline{x}$;
- its predicted label $\hat{x}$.

Conservative classifiers set the prediction of an element $x \in A_E^Y(S)$ as $\hat{x}$ as follows:

$$\text{if } x \in A_E^Y(S), \hat{x} = \overline{x} \text{ else } \hat{x} \text{ is undefined}$$

This kind of classifier cannot predict a label for an element which is not in $A_E^Y(S)$. In Algorithm 1, we provide the corresponding algorithm.

Let us note that $A_E^Y(S)$ is never explicitly computed. Instead, we look for every 3-tuple in S and check if they belong to $R_S^Y(x)$. Clearly, this is a supervised learning setting, where sample instances are stored for future use, without any generalization process. Conservative classifiers are Instance Based Learners as described in [1].

Such a conservative learner cannot generalize to any new input and is restricted to elements in $A_E^Y(S)$. This is not the case for instance-based learner like k-NN. This is why other options have been implemented to overcome this problem and to extend in some sense the generalization ability of analogical learner, as we will see in the next section.

Algorithm 1 *Conservative classifier*

```
Input: A sample set $S$ and an element $x \in X$ for which $\dot{x}$ is unknown.
Output: $\hat{x}$, an estimation of $\dot{x}$
Init: $C = \varnothing$        // multiset of candidate labels
for all $(a,b,c) \in S^3$ such that $a : b :: c : x$ do
  if $\exists y \in Y$ such that $\dot{a} : \dot{b} :: \dot{c} : y$ then
    // we are sure $(a,b,c) \in R_S^Y(x)$
    compute the solution $y$ of $\dot{a} : \dot{b} : \dot{c} : y$
    $C = C \cup y$
  end if
end for
$\hat{x} = \overline{x} = \text{Mode}(C)$ // undefined if $C = \varnothing$
```

3.2 Extended classifier

To relax the previous option, we need to be able to predict a label for elements outside $A_E^Y(S)$ i.e. elements which do not constitute a perfect analogy with elements in S. To this end, we can try to measure to what extent such elements are far from building a perfect analogy with those in S. The concept of *analogical dissimilarity*, first defined in [4], will be useful to quantify in some sense how far a relation $a : b :: c : d$ is from being a valid analogy. We keep the initial notation $AD(a,b,c,d)$ to denote the analogical dissimilarity between 4 elements. Some minimal properties have to be satisfied by such a dissimilarity $AD : X^4 \longrightarrow \mathbb{R}^+$ to fit with the intuition:

- $\forall a,b,c,d, \quad AD(a,b,c,d) = 0$ iff $a : b :: c : d$
- $\forall a,b,c,d, \quad AD(a,b,c,d) = AD(c,d,a,b) = AD(a,c,b,d)$
- $\forall a,b,c,d,e,f, AD(a,b,e,f) \leq AD(a,b,c,d) + AD(c,d,e,f)$

As the definition of an analogy strongly relies on the structure and operators available on X, we have the same situation for AD: there are a lot of possibilities. For instance:

- When $X = \mathbb{R}^m$ and $a : b :: c : d$ iff $a - b = c - d$, $AD(a,b,c,d) = ||(a-b) - (c-d)||_p$ is an analogical dissimilarity for any p, where $||.||_p$ denotes the standard p norm in $\mathbb{R}^m$.
- When $X = \mathbb{B}$ and $a : b :: c : d$ iff $(a \wedge b \equiv c \wedge d) \wedge (a \vee b) \equiv (c \vee d)$, one can define an analogical dissimilarity $AD(a,b,c,d)$ as the number of values that have to be switched to get a proper analogy. For instance, $AD(0,1,0,0) = 1$ and $AD(0,1,1,0) = 2$. The codomain of AD is just $\{0,1,2\}$. When extended to $X = \mathbb{B}^m$ with
$$AD(a,b,c,d) = \sum_{i=1}^{m} AD(a_i,b_i,c_i,d_i),$$
we get an analogical dissimilarity whose co-domain is $[0, 2m]$. In fact, this definition is just the restriction to $\mathbb{B}^m$ of the one coming from $\mathbb{R}^m$, when considering that $\mathbb{B}^m \subseteq \mathbb{R}^m$ and using the L_1 norm, i.e. $AD(a,b,c,d) = ||(a-b) - (c-d)||_1$.

As a measure of *how poorly an analogical proportion holds*, the analogical dissimilarity will help to define more flexible classifiers. The main underlying idea is to consider *approximate* analogies which are not valid stricto sensu, but not too far to be valid. In [4], after defining analogical dissimilarity, the authors build an extended classifier allowing classification of elements that do not belong to $A_E^Y(S)$. Algorithm 2 gives a description of their classifier.

This algorithm is similar to the conservative one but, instead of looking for pure analogies, we allow for some analogies not to be perfect when we need to. In their implementation [4], the authors

Algorithm 2 *Extended classifier*

```
Input: A sample set $S$, an element $x \in X$ for which $\dot{x}$ is unknown, a constant $k$.
Output: $\hat{x}$, an estimation of $\dot{x}$
Init: $C = \varnothing$        // multiset of candidate labels
for all $(a, b, c) \in S^3$ such that $\exists y \in Y$ with $\dot{a} : \dot{b} :: \dot{c} : y$ do
    compute $AD(a, b, c, x)$ and store it
end for
for all $k$ least values of $AD(a, b, c, x)$ do
    compute the solution $y$ of $\dot{a} : \dot{b} : \dot{c} : y$
    $C = C \cup y$
end for
$\hat{x} = \text{Mode}(C)$
```

actually look for all the 3-tuples that have the same analogical dissimilarity as the kth one: this allows them to fit with the previous conservative approach. For the sake of simplicity, we have chosen to ignore this small detail in our explanation.

In [4], the authors evaluated this classifier on a Boolean setting $\mathbb{B}^m$ over 8 benchmarks from the UCI repository. This approach led to remarkable results in terms of accuracy, when compared to off-the-shelf standard classifiers.

Nonetheless, this algorithm does not allow us to grasp its inherent working behaviour and it is difficult to extract theoretical properties. The aim of the next subsection is to give a functional translation of this algorithmic description.

3.3 Analogical classifier: a functional definition

As we have seen in the previous section, in the case of a Boolean setting, $AD(a, b, c, d) = ||(a - b) - (c - d)||_1$. A simple rewriting leads to:

$$AD(a, b, c, d) = ||d - (c - a + b)||_1 = ||d - d'||_1,$$

where $d' = c - a + b$. Actually, d' is nothing but the 4th vertex of the parallelogram $abcd'$ so this means that $AD(a, b, c, d)$ simply is the L_1 distance from d to this 4th vertex. Note that as $\mathbb{B}^m$ is not closed for addition, d' might not belong to $\mathbb{B}^m$ but to $\mathbb{R}^m$: this happens when one of the terms $AD(a_i, b_i, c_i, d_i)$ is equal to 2, as further discussed later.

As we have seen, for a given $x \in X$, algorithm 2 tries to minimise $AD(a, b, c, x)$ over all the 3-tuples $(a, b, c) \in S^3$. In the light of what has just been explained, we see that this is equivalent to finding the closest vertex $d' = c - a + b$ from x for any $(a, b, c) \in S^3$.

Denoting δ the L_1 distance, $AD(a, b, c, d) = \delta(a - b, c - d) = \delta(d, d')$, it is then natural to consider what we call the *nearest analogical neighbour* (or **nan**) of x from a sample S as the element of $A_E^Y(S)$ defined as:

$$\forall x \in X, \forall S \subseteq X, 1\text{-nan}(x, S) \overset{\text{def}}{=} \underset{d' \in A_E^Y(S)}{\arg\min}\ \delta(x, d')$$

When there is more than one nan, one can either proceed to a majority vote procedure among all their analogical labels, or randomly select one of these. This last option is the one we chose in our implementation.

Property 1 *We have the following equality:*

$$1\text{-}nan(x, S) = 1\text{-}nn(x, A_E^Y(S)).$$

The analogical classification rule simply is:

$$\hat{x} = \overline{1\text{-nan}(x, S)}.$$

In words, the predicted label of an element x is the analogical label of its nearest neighbour in $A_E^Y(S)$. In some sense, an analogical classifier behaves as a NN classifier but on an extended sample set.

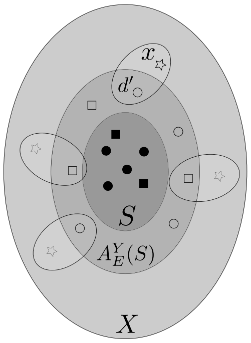
Obviously if x belongs to $A_E^Y(S)$ then x is its own nearest analogical neighbour: $1\text{-nan}(x, S) = x$ iff $x \in A_E^Y(S)$. Therefore, it is easy to see that this rule is a generalisation of the conservative approach. Instead of using only one nearest analogical neighbour, we can consider the set of the k nearest analogical neighbours, and implement a majority vote as it is done in [21].

The above definition leads to understand the process of analogical classification as follows:

1. First, extend the sample set S to its analogical extension $A_E^Y(S)$. $A_E^Y(S)$ can be viewed as an extended sample set that has **class noise**: the label associated with elements in $A_E^Y(S) \setminus S$ is their analogical label (as defined in 3.1), which may not be correct.
2. Then just apply a classical k-NN strategy over this extended sample set.

Figure 2 gives an illustration of the classification process: the label of $x \in X$ is unknown, and we set it to that of $d' \in A_E^Y(S)$ (a circle), which is its nearest analogical neighbour. To show that the analogical label of d' has itself been inferred, it is depicted as transparent instead of plain black. Let us note that topologically speaking, Figure 2 is not representative of a real case: even if we always have $S \subseteq A_E^Y(S) \subseteq X$, this does not mean that these sets are embedded into one another as shown in the drawing. Actually, elements of S (and thus of $A_E^Y(S)$) are usually scattered over the whole universe.

Figure 2. A graphical view of $A_E^Y(S)$ and the classification process.



As far as we know, this is the first time a functional definition of analogy-based classifiers is given. This definition clearly fits with the known algorithms but obviously, some implementation details cannot be exactly caught up by such a high level description. It is indeed possible to find a few edge cases where this functional definition may not output the same result as algorithm 2: this is the case for example when the nan of x is not unique. It is also the case when the closest vertex d' does not belong to B^m. However, as we will see in Section 6 these cases are not likely to occur and both approaches produce very similar results, thus empirically validating this functional definition.

Since we now have a clear functional definition of analogical classifiers, we are in position to examine some general properties such as convergence and VC-dimension of analogical learners. This is the purpose of the next section.

4 Some properties in the real case

Let us consider the case where $X = \mathbb{R}^m$, δ any distance issued from a norm, $AD(a, b, c, d) = \delta(a - b, c - d)$ and $x \in X$. In any case, just because $S \subseteq A_E(S)$, we have the following inequality:

$$\delta(x, \text{1-nan}(x, S)) \leq \delta(x, \text{1-nn}(x, S))$$

4.1 Study of convergence

Now, let us consider $x^{(i)}$ an i.i.d. sequence of random variables in $\mathbb{R}^m$, where $\mathbb{R}^m$ is equipped with a probability measure denoted P. As the set $S_n = \{x^{(i)},\ i \in [1, n]\}$ is random, then 1-nan(x, S_n) can also be considered as a random element of X. We then are in the exactly same context as the work of Cover & Hart ([9]), and we obtain the same result:

Property 2 *$plim_{n\to\infty}(1\text{-}nan(x, S_n)) = x$ almost surely,*

where $plim$ is the probability limit operator.

Proof: Exactly the same proof as in [9] could be applied. But it is simpler to remember that $\delta(x, \text{1-nan}(x, S_n)) \leq \delta(x, \text{1-nn}(x, S_n))$. Then, for a given x, the convergence in probability of $\delta(x, \text{1-nn}(x, S_n))$ to 0 implies the convergence in probability of $\delta(x, \text{1-nan}(x, S_n))$ to 0 which exactly means what needs to be proven. The subset of X where $plim_{n\to\infty}(\text{1-nan}(x, S_n)) \neq x$ is included into the subset of X where $plim_{n\to\infty}(\text{1-nn}(x, S)) \neq x$: Cover & Hart lemma tells us that this set has probability 0. Thus the final result. ■

Let us note the following points:

1. The lemma of Cover and Hart is more general than the one above. They have proven the result for any separable metric space, without any additional information. In fact, we cannot follow these lines here just because there is no known way to define an analogical dissimilarity on a metric space, without the help of other structure or operator (see [21] for a detailed discussion on this issue).
2. This result does not say anything regarding the prediction accuracy of 1-nan prediction rule as it is rather different than the 1-nn rule. Such consideration will be investigated in Section 5.
3. We have to be careful about the interpretation of this property in terms of machine learning. Indeed, a stronger property is proved in [9]: for an *integrable* function f over $\mathbb{R}^m$ w.r.t. the probability measure P, the expectation of $f(\text{1-nn}(x, S_n)) - f(x)$ converges to 0 when n goes to infinity. This means that asymptotically, the nearest neighbour of x has the same properties as x, and then the same label. Such a property has not yet been proven for 1-nan(x, S_n).
4. Finally, it is clear that when n goes to infinity, the behavior of an analogical classifier tends to that of a nearest neighbours classifier. Indeed, when S_n is very big, the nearest analogical neighbour of an element x simply is its nearest neighbour, in most cases. Moreover, when the nan and the nn are too close, paying the price of the noise related to the nan may not be worth it. This supports the common acknowledgement that analogical reasoning is mostly useful when very few data are available. In this later case extending a small sample set with its analogical extension may be particularly beneficial.

4.2 VC-dimension

The notion of VC-dimension was originally defined by Vapnik and Chervonenkis [32], and introduced into learnability theory by Blumer et al. [5]. Roughly speaking, the VC-dimension of a class of learners is a numerical measure of their discrimination power. It appears that this number is strongly linked to the confidence interval between the empirical risk (i.e. the error a learner makes on the sample set) and the true risk (the error a learner makes on the whole universe X). As such, the VC-dimension of a class of learners is an essential element of their theoretical study. We consider a universe X (usually a Cartesian product to represent the data) and a family $\mathcal{H} = \{h_i \subseteq X | i \in I\}$ of subsets of X. The elements of $\mathcal{H}$ will be referred as hypothesis or models. Given a subset A of X, we can consider the new family of subsets $tr(\mathcal{H}, A) = \{h_i \cap A \subseteq X | i \in I\}$: this family is called the *trace of $\mathcal{H}$ over A*. This is obviously a subset of the power set of A, 2^A i.e. $tr(\mathcal{H}, A) \subsetneq 2^A$. We say that $\mathcal{H}$ shatters A iff $tr(\mathcal{H}, A) = 2^A$. $VC\text{-}dim(\mathcal{H})$ is then the size of the largest finite subset which can be shattered by $\mathcal{H}$:

Definition 1 $VC\text{-}dim(\mathcal{H}) = \bigsqcup\{|A| \mid \mathcal{H} \text{ shatters } A\}$,

where $\bigsqcup$ is the least upper bound operator. In the case where $\forall n \in \mathbb{N}, \exists A \subset X, |A| = n$ such that $\mathcal{H}$ shatters A, we simply say that:

$$VC\text{-}dim(\mathcal{H}) = \infty.$$

As a binary classifier c over X defines a subset of X with $c^{-1}(1) = \{x \in X | c(x) = 1\}$, we can associate to a class $\mathcal{C}$ of classifiers a family of subsets $\{c^{-1}(1) | c \in \mathcal{C}\}$ and then the VC-dimension of a set of classifiers is as below:

Definition 2 $VC\text{-}dim(\mathcal{C}) = VC\text{-}dim(\{c^{-1}(1) | c \in \mathcal{C}\})$

For instance with $X = \mathbb{R}^n$ and with $\mathcal{C}$ the family of the k-NN classifiers: $\mathcal{C}_{\text{NN}} = \{k\text{-NN classifiers}, k \in \mathbb{N}^*\}$, then $VC\text{-}dim(\mathcal{C}_{\text{NN}}) = \infty$. Let us now consider the family of analogical binary classifiers $\mathcal{AC}_k$ whose classification rule is as below (where a majority vote is implemented):

$$\mathcal{AC}_k(x, S) = \overline{k\text{-}nan(x, S)}$$

In fact, an immediate result comes, derived from the core definition of an analogical proportion:

Property 3 $VC\text{-}dim(\mathcal{AC}_k) = \infty$

Proof: Given any x, the analogical proportion $x : x :: x : x$ always holds so that 1-nan$(x, S) = x$ then the label $\hat{x}$ allocated to x by AC_1 is just $\overline{x}$, which by definition equals $\dot{x}$. It means any set of items can be exactly labelled, thus the infinite $VC\text{-}dim$. ■

Regarding Property 3, the $\mathcal{AC}_k$ class behaves exactly as the k-NN class. Let us note that this is a very general result, which does not rely on any definition of distance. This is directly coming from a core property of analogical proportions.

5 Accuracy analysis in the Boolean case

In this section, we study the accuracy of an analogical classifier, and more particularly that of the 1-nan classifier (NaN). To do so, we restrict our view to a Boolean setting: elements to be classified belong to $X = \mathbb{B}^m$ and the label space is $Y = \mathbb{B}$.

As explained in Section 3.3, for a given $x \in X$ and a sample set $S \subset X$, we have:

$$\hat{x} = \overline{\text{1-nan}(x, S)} = \overline{\text{1-nn}(x, A_E^Y(S))}, \tag{1}$$

where $A_E^Y(S)$ is the analogical extension of S, that we will simply denote by A_E in what follows for notational brevity. We also denote $A_E^* \overset{\text{def}}{=} A_E \setminus S$ as the set of elements that belong A_E but not to S.

We now equip the set X with a probability distribution denoted P. The accuracy of the NaN_S classifier[5] over all the elements of X is defined as:

$$\text{Acc}(\text{NaN}_S, X) \overset{\text{def}}{=} P(\hat{x} = \dot{x} \mid x \in X)$$

By observing that for any x, its 1-nan either belongs to S or to A_E^*, the above equation can be split into two distinct parts as follows:

$$\begin{aligned}
&P\left(\hat{x} = \dot{x} \mid x \in X\right) = P\left(\overline{\text{1-nan}(x,S)} = \dot{x}\right)\\
&= P\left([\overline{\text{1-nan}(x,S)} = \dot{x}] \wedge [\text{1-nan}(x,S) \in S]\right) +\\
&\quad P\left([\overline{\text{1-nan}(x,S)} = \dot{x}] \wedge [\text{1-nan}(x,S) \in A_E^*]\right)\\
&= P\left([\overline{\text{1-nn}(x,A_E)} = \dot{x}] \wedge [\text{1-nn}(x,A_E) \in S]\right) +\\
&\quad P\left([\overline{\text{1-nn}(x,A_E)} = \dot{x}] \wedge [\text{1-nn}(x,A_E) \in A_E^*]\right)\\
&= P\left([\overline{\text{1-nn}(x,A_E)} = \dot{x}] \mid [\text{1-nn}(x,A_E) \in S]\right) \times\\
&\quad P\left([\text{1-nn}(x,A_E) \in S]\right) +\\
&\quad P\left([\overline{\text{1-nn}(x,A_E)} = \dot{x}] \mid [\text{1-nn}(x,A_E) \in A_E^*]\right) \times\\
&\quad P\left([\text{1-nn}(x,A_E) \in A_E^*]\right)
\end{aligned}$$

Let us denote $\alpha \overset{\text{def}}{=} P(\text{1-nn}(x,A_E) \in S)$[6]. The formula becomes:

$$\begin{aligned}
&\text{Acc}(\text{NaN}_S, X) =\\
&P\left([\overline{\text{1-nn}(x,A_E)} = \dot{x}] \mid [\text{1-nn}(x,A_E) \in S]\right) * \alpha +\\
&P\left([\overline{\text{1-nn}(x,A_E)} = \dot{x}] \mid [\text{1-nn}(x,A_E) \in A_E^*]\right) * (1-\alpha).
\end{aligned}$$

Let us focus on the first term (discarding the factor α):

$$P\left([\overline{\text{1-nn}(x,A_E)} = \dot{x}] \mid [\text{1-nn}(x,A_E) \in S]\right)$$

It is easy to see that the event $[\text{1-nn}(x,A_E) \in S]$ is equivalent to the event $[\text{1-nn}(x,A_E) = \text{1-nn}(x,S)]$. As a result, we can transform the first term to get a better grasp of its meaning:

$$\begin{aligned}
&P\left([\overline{\text{1-nn}(x,A_E)} = \dot{x}] \mid [\text{1-nn}(x,A_E) \in S]\right)\\
&= P\left([\overline{\text{1-nn}(x,A_E)} = \dot{x}] \mid [\text{1-nn}(x,A_E) = \text{1-nn}(x,S)]\right)\\
&= P\left([\overline{\text{1-nn}(x,S)} = \dot{x}] \mid [\text{1-nn}(x,A_E) \in S]\right).
\end{aligned}$$

In this form, the first term is just the accuracy of the NN_S algorithm over the elements that have their nearest analogical neighbour in S. As for the second term, the same process can be applied by observing that the event $[\text{1-nn}(x,A_E) \in A_E^*]$ is equivalent to the event $[\text{1-nn}(x,A_E) = \text{1-nn}(x,A_E^*)]$. This leads to

$$\begin{aligned}
&P\left([\overline{\text{1-nn}(x,A_E)} = \dot{x}] \mid [\text{1-nn}(x,A_E) \in A_E^*]\right)\\
&= P\left([\overline{\text{1-nn}(x,A_E^*)} = \dot{x}] \mid [\text{1-nn}(x,A_E) \in A_E^*]\right).
\end{aligned}$$

This second term is then the accuracy of the $\text{NN}_{A_E^*}$ algorithm over the elements that have their nearest analogical neighbour in A_E^*.

In the light of these interpretations, one can rewrite the accuracy formula in a concise form, using a few more definitions:

- $A \overset{\text{def}}{=} \{x \in X, \text{1-nan}(x,S) \in S\}$: the elements that have their nan in S.
- $B \overset{\text{def}}{=} \{x \in X, \text{1-nan}(x,S) \in A_E^*\}$: the elements that have their nan in A_E^*.

Naturally, $A \cup B = X$ and $A \cap B = \emptyset$. Also, $\alpha = P(x \in A)$ and $1 - \alpha = P(x \in B)$. Therefore, the accuracy of NaN_S over X can be understood as the weighted sum of the accuracy of NN over A and B, using a different sample set each time (respectively S and A_E^*):

$$\begin{aligned}
\text{Acc}(\text{NaN}_S, X) = \; &\text{Acc}(\text{NN}_S, A) \cdot \alpha +\\
&\text{Acc}(\text{NN}_{A_E^*}, B) \cdot (1-\alpha). \qquad (2)
\end{aligned}$$

The value $\text{Acc}(\text{NN}_S, A)$ is the accuracy of NN_S over all the elements in A. A theoretical study of this accuracy has been done in [19] when the size of A is known. Regarding $\text{Acc}(\text{NN}_{A_E^*}, B)$, this is the accuracy of 1-nn when the sample set is noisy, and has been studied in [24]. This last formula leads to the consistent facts:

1. The smaller A_E^* (i.e. analogical reasoning does not bring much more labels), the closer α is to 1, the closer A is to X and the more the accuracy of NaN_S tends towards the accuracy of NN_S over X.
2. In return, if A_E is much bigger than S, α is then small, B is close to X and the accuracy of NaN_S greatly depends on the quality of A_E, which can be measured by the value ω defined as:

$$\omega \triangleq P(\overline{x} = \dot{x} \mid x \in A_E^*).$$

Note that the value $1 - \omega$ corresponds to the class noise of A_E. As we will see in the next section, this situation where A_E is big with respect to S is actually extremely likely to occur.

6 Experiments and empirical validation

In order to get an empirical validation of our formulas, we have developed a set of experiments that we describe in the next subsection.

6.1 Validation protocol

Working with Boolean vectors, we have computed the accuracies of the NaN and NN algorithms over $X = \mathbb{B}^m$ for different values of m (namely 8 and 10). The ground truth label of elements of X is defined by different Boolean functions f in such a way that the $\forall x = (x_1, \cdots, x_m),\ \dot{x} = f(x)$. The different functions we have worked with are:

- $f(x) = x_m$: in that case, we can consider the $m-1$ first parameters are a kind of noise since they have no influence on the final label ;
- $f(x) = 1$ iff at least l components are equal to 1 (this kind of function is usually called l-of-m). We chose to set l to $\frac{m}{2}$;
- $f(x) = x_1 \oplus x_2$ (xor): we here have $m-2$ useless attributes ;
- $f(x) = 1$ iff $\sum x_i = 2$: all the attributes are relevant in that case ;
- $f(x) = 1$ iff $\sum x_i = m-1$: an extreme case of the previous one ;

[5] The S subscript is here to specify that the training set of the NaN algorithm is S. The same notation is used for the *nearest neighbour* algorithm: NN_Σ is the NN algorithm trained on the set Σ.

[6] Obviously, we also have $\alpha = P(\text{1-nan}(x,S) \in S)$.

- $f(x) = 1$ iff $x_1 \cdot x_m = 1$: only the first and the last elements are relevant.

Regarding the size of the training set, to be sure to fit with the size of the universe, we have investigated various sizes between 3 and 100. When dealing with a training set of size 100, the cubic complexity of the analogical classifier leads to explore a set of approximately 100^3 elements: as a consequence, we limit our investigation to a maximum of 100 elements in the training set in order to get realistic execution time.

All the accuracy (and other metrics) computations are averaged over a set of 100 experiments. The interested reader may find the Python source code that has generated all our plots and detailed results on Github[7]. For lack of space, we only provide a few examples which are representative of the global behavior. Please note that our implementation of the NaN algorithm is not that of algorithm 2, but is instead that of the functional definition of the analogical classifier developed in Section 3.3: we first construct the analogical extension set of S, and then proceed to a nearest neighbour strategy over this noisy extended training set. We have estimated probabilities by frequencies, thus implicitly assuming a uniform distribution on X.

In addition to these Boolean functions, we have also run the NaN algorithm over the Monk datasets over the UCI repository[8]. They are datasets of 432 binarized elements, among which exactly 169 of them have been used for training.

6.2 Experiments

Figure 3 shows the accuracies (left column) of the NaN and NN over six different Boolean settings with values of $|S|$ varying from 3 to 100. In the right column, we have plotted three different values that will help us analyse and validate the behaviour of the NaN algorithm:

- the theoretical accuracy as defined by equation (2) in Section 5. The probability $\alpha = P(x \in A)$ has been estimated by the frequency: $\frac{|A|}{|X|}$;
- the quality of $A_E^Y(S)$, measured by ω as defined in Section 5 which is estimated by the frequency: $\frac{|\{x \in A_E^* \mid \overline{x} = \dot{x}\}|}{|A_E^*|}$;
- finally, the quantity $\gamma = \frac{|A_E^Y(S)|}{|X|}$: the size of the analogical extension set with respect to that of the whole universe.

Table 1 shows the same metrics for the Monk datasets and also report the results of the Analogical Proportion Classifier (*APC*) from [21], which corresponds to algorithm 2 with $k = 100$.

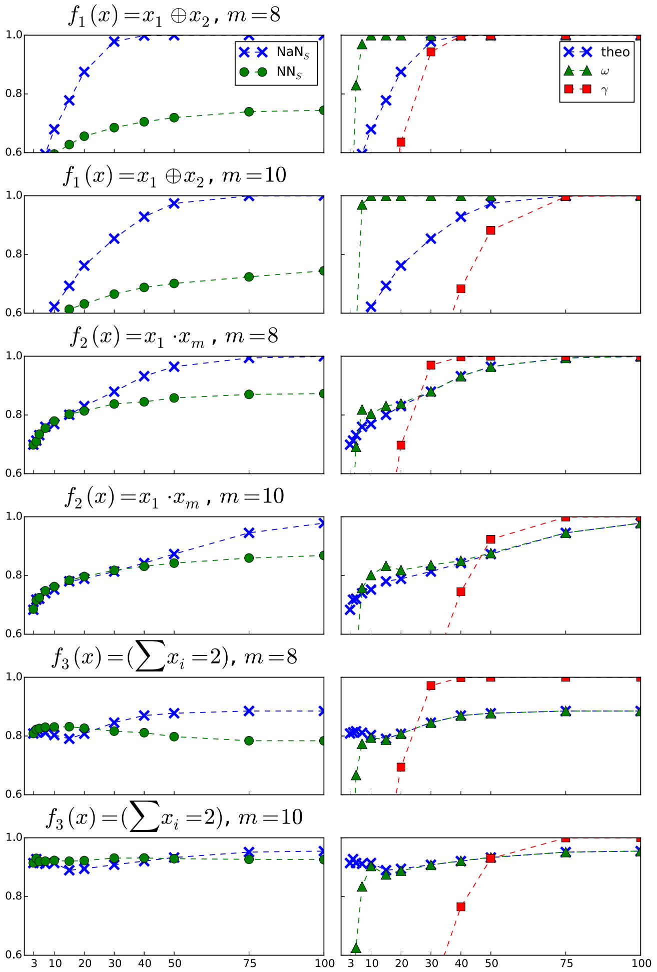
Table 1. Accuracies of the NaN, APC and NN algorithms over the Monk datasets

	NaN	APC	NN	ω	γ
Monk 1	.961	.98	.787	.961	1
Monk 2	.998	1	.738	.996	1
Monk 3	.963	.96	.829	.963	1

6.3 Comments and discussion

The experiments shown in figure 3 allow us to draw interesting conclusions about the behaviour of the NaN algorithm. We can observe one of the two cases:

Figure 3. Accuracies of the NaN and NN algorithms over different Boolean settings and training set sizes, with corresponding values of ω, γ, and theoretical accuracy. The x axis corresponds to the size of the training set.



- either the analogical labels are always correctly predicted[9] ($\omega = 1$, i.e. there is no class noise) as it is the case for f_1, $f(x) = x_m$ and (almost) for the Monk datasets ;
- or there is some class noise in $A_E^Y(S)$ ($\omega \neq 1$). In this case, we always observe that the NaN algorithm is outperformed by NN for small values of $|S|$, but eventually takes advantage once analogical prediction becomes more important than the nearest neighbour one, as we are going to see.

The theoretical accuracy seems to fit perfectly with the empirical accuracy of the NaN algorithm, thus validating our theoretical study that led to equation (2)[10].

An interesting observation is that the value of ω always converges to that of the theoretical accuracy (and therefore to the actual accuracy) of NaN. This can be easily explained by paying attention to the value of γ, the proportion of elements of X that belong to $A_E^Y(S)$. We see that in any setting, γ converges to 1 as $|S|$ grows. This means that when $|S|$ is big enough (but not necessarily *that* big with re-

[7] https://github.com/Niourf/nan_study

[8] https://archive.ics.uci.edu/ml/datasets/MONK's+Problems

[9] Note that for small values of $|S|$, it seems that $\omega \neq 1$. This is due to the fact that for such small values, it is sometimes impossible to construct $A_E^Y(S)$, thus leading to a value of $\omega = 0$ (which will be averaged afterwards over the 100 experiments).

[10] The maximal difference we observed between the theoretical accuracy and its actual value is of about 10^{-10}.

spect to X), the analogical extension of S covers the whole universe X[11]: every element x is then its own nearest analogical neighbour and $\hat{x} = \overline{x}$. It is therefore straightforward to see that in this case,

$$\begin{aligned} \omega = P(\overline{x} = \dot{x} \mid x \in A_E^*) &= P(\hat{x} = \dot{x} \mid x \in A_E^*) \\ &= \text{Acc}(\text{NaN}_S, A_E^*) \end{aligned}$$

When $\gamma = 1$, the only elements x we want to classify belong to A_E^* (otherwise they would be in S), so this last term exactly corresponds to the accuracy of the classifier. Another way to see it is to observe that the first term of equation (2) $\text{Acc}(\text{NN}_S, A) \cdot \alpha$ is null because $\alpha = 0$. Only the second term $\text{Acc}(\text{NN}_{A_E^*}, B) \cdot (1 - \alpha)$ is of importance, and its value corresponds to ω. This observation allows us to state that estimating the value of ω is paramount to have a precise idea of the accuracy of an analogical classifier. We will provide in the next subsection a method to accurately estimate this quantity ω with the only help of the training set S.

Regarding the Monk datasets (Table 1), we note that the functional NaN approach (almost) achieves the same results as the somewhat more complex algorithm described in Section 3.2, and that here again the analogical extension set covers the whole universe: this means that a conservative approach would have been sufficient! Actually, this raises the following question: why would we want to look for more than one analogical neighbour when every element of the universe is already in $A_E^Y(S)$, and therefore *analogically linked* to those in S? Our experiments tend to show that this becomes superfluous, provided that the training set is big enough.

6.4 Estimation of the prediction accuracy

We have seen in the previous subsection that the value ω is that of the actual accuracy of an analogical classifier when S is big enough. This leads to the following question: how can we get a precise estimation of this value ω ? Answering this would allow us to have a very precise idea of the accuracy we can expect from our classifier.

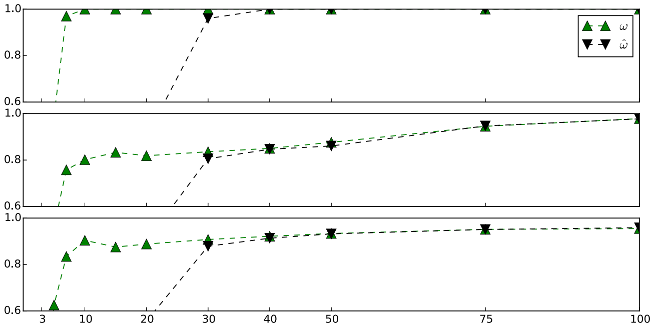
The method we propose for estimating ω only relies on the training set S and is very simple: it consists of applying the conservative algorithm to all the elements of S, and compute the fraction of these elements that have been correctly classified. A small yet important modification to the algorithm needs to be added: we only want to construct analogical proportions of the form $a : b :: c : x$ where a, b, c and x are all distinct elements. Indeed, the proportions $x : x :: x : x$ and $x' : x :: x' : x$ are always true, and the solution label related to these proportions would bias the final majority vote procedure in a significant way towards the real label $\dot{x}$.

We have applied this estimation protocol to all of the Boolean settings we have considered, and it has shown to be very accurate. Figure 4 illustrates a few of these settings (already considered in Figure 3). We can see that the estimation $\hat{\omega}$ converges to ω when S is big enough. For small values of S, this estimation is indeed imprecise as it is difficult to find a lot a 3-tuples such that an analogical proportion holds for every element.

7 Conclusion

In this paper, we have provided a functional definition of analogical learners. Starting from this definition, we are in a position to prove an analytic convergence result, similar to that of the nearest neighbour algorithm. Obviously, this is not enough to conclude regarding the predictive ability of analogy-based classifiers. We have also shown that their VC-dimension is infinite. It should not come as a surprise, as a very particular case of analogical rule (when the analogical proportion is trivial) is the k-NN rule.

Figure 4. Values of ω and its estimation $\hat{\omega}$ for f_1, f_2 and f_3 in $\mathbb{B}^{10}$.



In terms of accuracy in a Boolean setting, we have found a strong link between the accuracy of the NaN_S algorithm and that of the NN_S algorithm. At a first glance, we can consider the NaN algorithm as a NN strategy on an extended and noisy sample set: the analogical extension of S. In the end, we have seen that this extended sample set covers the entire universe provided that S is big enough, simplifying and bringing back the accuracy of the classifier to the value ω which corresponds to the quality of the analogical extension. We have also provided a method to accurately estimate the value of ω that only relies on elements of the S, thus allowing beforehand to have a precise idea of the accuracy of any analogical classifier in a Boolean setting. Some important points remain to be investigated, such as:

- What can we expect in terms of speed convergence from an analogical learner? In other words, what is the minimum size needed from a sample set to get a fixed accuracy threshold?
- If a clever learning strategy can (at least partially) overcome the problem of infinite VC-dimension, can we overcome the issue of the cubic complexity of analogical learners?
- Leaving the field of classification, can we provide a clear strategy for transfer learning with analogy? Indeed, the central goal of transfer learning is to identify and exploit analogies between source and target domains [25].

These points definitely constitute interesting challenges for future works. Nevertheless, we have to remember that analogical reasoning brings its whole power in the case where few data are available. If a lot of data are available, it is very likely that we have elements similar to the one at hand and, in that case, a k-NN style reasoning is natural. In the opposite case, when we only have a few relevant cases at hand, applying analogical proportion-based predictions appears to be a meaningful option.

[11] Obviously, the bigger the dimension m, the slower the convergence occurs.

REFERENCES

[1] C. C. Aggarwal, *Data Classification: Algorithms and Applications*, Data Mining and Knowledge Discovery Series, Chapman and Hall/CRC, 2014.

[2] N. Barbot and L. Miclet, 'La proportion analogique dans les groupes: applications aux permutations et aux matrices', Technical Report 1914, IRISA, (July 2009).

[3] M. Bayoudh, H. Prade, and G. Richard, 'Evaluation of analogical proportions through kolmogorov complexity', *Knowl.-Based Syst.*, **29**, 20–30, (2012).

[4] S. Bayoudh, L. Miclet, and A. Delhay, 'Learning by analogy: A classification rule for binary and nominal data', *Proc. Inter. Joint Conf. on Artificial Intelligence IJCAI07*, 678–683, (2007).

[5] A; Blumer, A. Ehrenfeucht, D; Haussler, and M; K. Warmuth, 'Learnability and the vapnik-chervonenkis dimension', *J. ACM*, **36**(4), 929–965, (1989).

[6] M. Bounhas, H. Prade, and G. Richard, 'Analogical classification: A new way to deal with examples', in *ECAI 2014 - 21st European Conference on Artificial Intelligence, 18-22 August 2014, Prague, Czech Republic*, volume 263 of *Frontiers in Artificial Intelligence and Applications*, pp. 135–140. IOS Press, (2014).

[7] A. Cornuéjols, 'Analogy as minimization of description length', in *Machine Learning and Statistics: The interface*, eds., G. Nakhaeizadeh and C. Taylor, pp. 321–336. Wiley and Sons, (1996).

[8] W. Correa, H. Prade, and G. Richard, 'When intelligence is just a matter of copying', in *Proc. 20th Eur. Conf. on Artificial Intelligence, Montpellier, Aug. 27-31*, pp. 276–281. IOS Press, (2012).

[9] T. M. Cover and P. E. Hart, 'Nearest neighbor pattern classification', in *IEEE Transactions on Information Theory*, volume 13, 21–27, IEEE, (1967).

[10] T. R. Davies and Russell S. J., 'A logical approach to reasoning by analogy', in *Proc. of the 10th International Joint Conference on Artificial Intelligence (IJCAI'87), Milan*, ed., J. P. McDermott, 264–270, Morgan Kaufmann, (1987).

[11] M. Dorolle, *Le Raisonnement par Analogie*, PUF, Paris, 1949.

[12] D. Gentner, 'Structure-mapping: A theoretical framework for analogy', *Cognitive Science*, **7**(2), 155–170, (1983).

[13] D. Gentner, K. J. Holyoak, and B. N. Kokinov, *The Analogical Mind: Perspectives from Cognitive Science*, Cognitive Science, and Philosophy, MIT Press, Cambridge, MA, 2001.

[14] H. Gust, K. Kühnberger, and U. Schmid, 'Metaphors and heuristic-driven theory projection (HDTP)', *Theoretical Computer Science*, **354**(1), 98 – 117, (2006).

[15] T. Hinrichs and K. D. Forbus, 'Transfer learning through analogy in games', *AAAI*, **32**(1), 70–83, (2011).

[16] D. Hofstadter and M. Mitchell, 'The Copycat project: A model of mental fluidity and analogy-making', in *Fluid Concepts and Creative Analogies: Computer Models of the Fundamental Mechanisms of Thought*, eds., D. Hofstadter and The Fluid Analogies Research Group, pp. 205–267, New York, NY, (1995). Basic Books, Inc.

[17] D. Hofstadter and E. Sander, *Surfaces and Essences: Analogy as the Fuel and Fire of Thinking*, Basic Books, 2013. French version: L'Analogie, Cœur de la Pensée - Odile Jacob, Paris.

[18] D. R. Hofstadter and The Fluid Analogy Research Group, *Fluid Concepts and Creative Analogies: Computer Models of the Fundamental Mechanisms of Thought*, Basic Books, Inc., New York, NY, USA, 1996.

[19] P. Langley and W. Iba, 'Average-case analysis of a nearest neighbor algorithm', in *Proceedings of the 13th Int. Joint Conf. on Artificial Intelligence. Chambéry, France, August 28 - September 3*, pp. 889–894. Morgan Kaufmann, (1993).

[20] Y. Lepage, 'De l'analogie rendant compte de la commutation en linguistique', *Habilit. à Diriger des Recher., Univ. J. Fourier, Grenoble*, (2003).

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

[22] L. Miclet and A. Delhay, 'Relation d'analogie et distance sur un alphabet défini par des traits', Technical Report 1632, IRISA, (July 2004).

[23] L. Miclet and 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*, pp. 638–650. Springer, LNCS 5590, (2009).

[24] S. Okamoto and N. Yugami, 'An average-case analysis of the k-nearest neighbar classifier for noisy domains', in *Proceedings of the Fifteenth Int. Joint Conf. on Artificial Intelligence, IJCAI 97, Nagoya, Japan, August 23-29*, pp. 238–245. Morgan Kaufmann, (1997).

[25] S. J. Pan and Q. Yang, 'A survey on transfer learning', *IEEE Trans. on Knowl. and Data Eng.*, **22**(10), 1345–1359, (2010).

[26] H. Prade and G. Richard, 'From analogical proportion to logical proportions', *Logica Universalis*, **7(4)**, 441–505, (2013).

[27] *Computational Approaches to Analogical Reasoning: Current Trends*, eds., H. Prade and G. Richard, volume 548 of *Studies in Computational Intelligence*, Springer, 2014.

[28] H. Prade, G. Richard, and B. Yao, 'Enforcing regularity by means of analogy-related proportions-a new approach to classification', *International Journal of Computer Information Systems and Industrial Management Applications*, **4**, 648–658, (2012).

[29] M. Ragni and S. Neubert, 'Solving Raven's IQ-tests: An AI and cognitive modeling approach', in *Proc. 20th Europ. Conf. on Artificial Intelligence (ECAI'12) Montpellier, Aug. 27-31*, eds., L. De Raedt, C. Bessière, D. Dubois, P. Doherty, P. Frasconi, F. Heintz, and P. J. F. Lucas, pp. 666–671, (2012).

[30] J. F. Sowa and A. K. Majumdar, 'Analogical reasoning', in *Proc. Inter. Conf. on Conceptual Structures*, pp. 16–36. Springer, LNAI 2746, (2003).

[31] N. Stroppa and F. Yvon, 'Analogical learning and formal proportions: Definitions and methodological issues', Technical Report D004, ENST-Paris, (2005).

[32] V. Vapnik, *Statistical Learning Theory*, Wiley, 1998.

[33] H. Wang and Q. Yang, 'Transfer learning by structural analogy', in *Proc. of the 25 AAAI Conference on Artificial Intelligence*, pp. 513–518. AAAI Press, (2011).