Improving analogical extrapolation using case pair competence

Jean Lieber, Emmanuel Nauer, Henri Prade

Improving analogical extrapolation using case pair competence

Jean Lieber[1], Emmanuel Nauer[1], and Henri Prade[2]

[1] Université de Lorraine, CNRS, Inria, LORIA, F-54000 Nancy, France
[2] IRIT, Université de Toulouse, France

Abstract. An analogical proportion is a quaternary relation that is to be read "a is to b as c is to d", verifying some symmetry and permutation properties. As can be seen, it involves a pair of pairs. Such a relation is at the basis of an approach to case-based reasoning called analogical extrapolation, which consists in retrieving three cases forming an analogical proportion with the target problem in the problem space and then in finding a solution to this problem by solving an analogical equation in the solution space. This paper studies how the notion of competence of pairs of source cases can be estimated and used in order to improve extrapolation. A preprocessing of the case base associates to each case pair a competence given by two scores: the support and the confidence of the case pair, computed on the basis of other case pairs forming an analogical proportion with it. An evaluation in a Boolean setting shows that using case pair competences improves significantly the result of the analogical extrapolation process.

Keywords: analogical proportion, analogical inference, case-based reasoning, competence, extrapolation

1 Introduction

In a recent paper [15], the authors have advocated that reasoning about cases (or case-based reasoning, CBR [18, 17]) may not be only based on similarity-based reasoning, looking for the nearest solved cases, but may also use analogical proportions for extrapolation purposes. Extrapolation is based on analogical inference, that uses triples of cases for building the solution of a fourth (new) case through an adaptation mechanism [2].

Usually, several triples in the case base can be used for predicting the solution of the fourth case, and predictions may diverge. In fact, it has been established for Boolean features that such an inference makes no error (thus all the triples agree on the same prediction) if and only if the function that associates the solution to the description of a case is an affine Boolean function [5]. This is why, when the function is not assumed to be affine, a voting procedure is organized between the predicting triples.

Such a procedure is quite brute-force, and did not take really lesson from the case base. Indeed, it may happen that some triples in the base fail to predict

the correct answer of another case of the case base. In this paper, we propose to take into account this kind of information for restricting the number of triples used for making a prediction in a meaningful way.

The paper is organized as follows. The next section provides the necessary background on analogical proportions and the notations about CBR used throughout the paper. Section 3 discusses how to restrict the set of triples allowed to participate to a given prediction. Section 4 reports experimentations showing the gain in accuracy of the new inference procedure. Section 5 discusses related work, before concluding.

2 Preliminaries

This section presents first the formal framework of this study: the nominal representations and, more specifically, the representation by tuples of Boolean values. Then, it recalls some notions and gives some notations about analogical proportions and about case-based reasoning.

2.1 Nominal representations and Boolean setting

Feature-value representations are often used in CBR (see, e.g., [13]). A nominal representation is a feature-value representation where the range of each feature is finite (and, typically, small). More formally, let $\mathcal{U}_1$, $\mathcal{U}_2$, ..., $\mathcal{U}_p$ be p finite sets and $\mathcal{U} = \mathcal{U}_1 \times \mathcal{U}_2 \times \ldots \times \mathcal{U}_p$. A feature on $\mathcal{U}$ is one of the p projections $(x_1, x_2, \ldots, x_p) \in \mathcal{U} \mapsto x_i \in \mathcal{U}_i$ ($i \in \{1, 2, \ldots, p\}$).

A Boolean representation is a nominal representation where $\mathcal{U}_1 = \mathcal{U}_2 = \ldots = \mathcal{U}_p = \mathbb{B}$, where $\mathbb{B} = \{0, 1\}$ is the set of Boolean values: the value "false" is assimilated to the integer 0, and "true" is assimilated to 1. The Boolean operators $\neg$, $\wedge$ and $\vee$ are defined, for $\mathsf{a}, \mathsf{b} \in \mathbb{B}$, by $\neg \mathsf{a} = 1 - \mathsf{a}$, $\mathsf{a} \wedge \mathsf{b} = \begin{cases} 1 & \text{if } \mathsf{a} = \mathsf{b} = 1 \\ 0 & \text{otherwise} \end{cases}$ and $\mathsf{a} \vee \mathsf{b} = \neg(\neg \mathsf{a} \wedge \neg \mathsf{b})$. An element of $\mathbb{B}^p$ is denoted without commas and parentheses, e.g., 01101 stands for $(0, 1, 1, 0, 1)$.

2.2 Analogical proportions

Given a set $\mathcal{U}$, an analogical proportion on $\mathcal{U}$ is a quaternary relation on $\mathcal{U}$, denoted by $a\!:\!b\!::\!c\!:\!d$ for $(a, b, c, d) \in \mathcal{U}^4$, and satisfying the following postulates (for $a, b, c, d \in \mathcal{U}$):

(Reflexivity) $a\!:\!b\!::\!a\!:\!b$.
(Symmetry) If $a\!:\!b\!::\!c\!:\!d$ then $c\!:\!d\!::\!a\!:\!b$.
(Exchange of the means) If $a\!:\!b\!::\!c\!:\!d$ then $a\!:\!c\!::\!b\!:\!d$.

Given a finite set $\mathcal{U}_i$, the relation defined below is an analogical proportion:

$$a\!:\!b\!::\!c\!:\!d \stackrel{\text{def}}{=} (a = b \wedge c = d) \vee (a = c \wedge b = d)$$

2

Therefore, in the nominal representation, the 4-tuples (a, b, c, d) in analogical proportion have one of the three following forms: (s, s, s, s), (s, t, s, t) and (s, s, t, t) for $s, t \in \mathcal{U}_i$. In particular, if $\mathcal{U}_i = \mathbb{B}$, the set of $(\mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{d}) \in \mathbb{B}^4$ such that $\mathtt{a}{:}\mathtt{b}{::}\mathtt{c}{:}\mathtt{d}$ is $\{0000, 0011, 0101, 1111, 1100, 1010\}$.

Given a finite $\mathcal{U} = \mathcal{U}_1 \times \mathcal{U}_2 \times \ldots \times \mathcal{U}_p$ the following analogical proportion can be defined:

$$a{:}b{::}c{:}d \quad \overset{\text{def}}{=} \quad a_1{:}b_1{::}c_1{:}d_1 \quad \wedge \quad a_2{:}b_2{::}c_2{:}d_2 \quad \wedge \quad \ldots \quad \wedge \quad a_p{:}b_p{::}c_p{:}d_p$$

Given $a, b, c \in \mathcal{U}$, solving the *analogical equation* $a{:}b{::}c{:}y$ aims at finding the $y \in \mathcal{U}$ such that this relation holds. In a nominal representation, such an equation has 0 or 1 solution. More precisely:

- If $a = b$, the solution is $y = c$.
- If $a = c$, the solution is $y = b$.
- Otherwise, $a{:}b{::}c{:}y$ has no solution.

2.3 Notations and assumptions on CBR

Let $\mathcal{P}$ and $\mathcal{S}$ be two sets. A *problem* $\mathtt{x}$ is by definition an element of $\mathcal{P}$ and a *solution* $\mathtt{y}$, an element of $\mathcal{S}$. If $a \in \mathcal{P} \times \mathcal{S}$, then $\mathtt{x}^a$ and $\mathtt{y}^a$ denote its problem and solution parts: $a = (\mathtt{x}^a, \mathtt{y}^a)$. Let $\rightsquigarrow$ be a relation on $\mathcal{P} \times \mathcal{S}$. For $(\mathtt{x}, \mathtt{y}) \in \mathcal{P} \times \mathcal{S}$, $\mathtt{x} \rightsquigarrow \mathtt{y}$ is read "$\mathtt{x}$ has for solution $\mathtt{y}$" or "$\mathtt{y}$ solves $\mathtt{x}$". A *case* is a pair $(\mathtt{x}, \mathtt{y})$ such that $\mathtt{x} \rightsquigarrow \mathtt{y}$. The aim of a CBR system is to solve problems, i.e., it should approximate the relation $\rightsquigarrow$: given $\mathtt{x}^{\mathtt{tgt}} \in \mathcal{P}$ (the *target* problem), it aims at proposing $\mathtt{y}^{\mathtt{tgt}} \in \mathcal{S}$ such that it is plausible that $\mathtt{x}^{\mathtt{tgt}} \rightsquigarrow \mathtt{y}^{\mathtt{tgt}}$. For this purpose, a finite set of cases, called the case base and denoted by $\mathtt{CB}$, is used. An element of $\mathtt{CB}$ is called a *source case*. Besides the case base, other knowledge containers are often used [17], but they are not considered in this paper.

The classical way of defining a CBR process consists in selecting a set of k source cases related to $\mathtt{x}^{\mathtt{tgt}}$ (retrieve phase) and solve $\mathtt{x}^{\mathtt{tgt}}$ with the help of the retrieved cases (reuse phase). Other steps are considered in the classical 4 Rs model [1], but not in this paper. In [15], three approaches are presented for $k \in \{1, 2, 3\}$. The approach for $k = 3$, called analogical extrapolation, is recalled in the next section.

3 Improving extrapolation thanks to case pair competence

This section presents the proposed approach. First, it is shown how a notion of competence associated to case pairs can be used to improve extrapolation, an approach to CBR based on analogical proportions. Then, this notion of competence is formally defined. Finally, strategies for exploiting case pair competence are described.

3.1 Principles

The analogical proportion-based inference principle [20] can be stated as follows (using the notations on CBR introduced above; $a = (\mathtt{x}^a, \mathtt{y}^a)$, $b = (\mathtt{x}^b, \mathtt{y}^b)$, $c = (\mathtt{x}^c, \mathtt{y}^c)$ and $d = (\mathtt{x}^d, \mathtt{y}^d)$ are four cases):

$$\frac{\mathtt{x}^a\!:\!\mathtt{x}^b\!::\!\mathtt{x}^c\!:\!\mathtt{x}^d \text{ holds}}{\mathtt{y}^a\!:\!\mathtt{y}^b\!::\!\mathtt{y}^c\!:\!\mathtt{y}^d \text{ holds}}$$

In order to solve a new problem $\mathtt{x}^{\mathtt{tgt}}$, this leads to look for all triples of source cases (a, b, c) such that $\mathtt{x}^a\!:\!\mathtt{x}^b\!::\!\mathtt{x}^c\!:\!\mathtt{x}^{\mathtt{tgt}}$ holds and such that the equation $\mathtt{y}^a\!:\!\mathtt{y}^b\!::\!\mathtt{y}^c\!:\!y$ is solvable. Let $\mathcal{T}$ be the set of all these triples. Then the implementation of this inference pattern uses a vote among all triples of $\mathcal{T}$ and chooses the solution y found for the largest number of triples. This is the principle called analogical extrapolation (or, simply, extrapolation) in [15].

In the following, we assume for simplicity that all the features are nominal (e.g., Boolean). When there is only one feature for the solutions, the problem-solving task is a classification task (finding the class $\mathtt{y}^{\mathtt{tgt}} \in \mathcal{S}$ to be associated with $\mathtt{x}^{\mathtt{tgt}}$). When there are several features, one can handle them one by one only if they are logically independent, otherwise the vote should be organized between the whole vectors describing the different solutions. In the following, we assume independence, and we consider one of the components $\mathtt{y}_i$ of a solution $\mathtt{y}$ (thus, the index i is useless: $\mathtt{y}_i$ is denoted by $\mathtt{y}$).

Still, one may wonder if all triples of $\mathcal{T}$ involved in a vote for making a particular prediction have the same legitimacy. Indeed, one may take lesson from $\mathcal{T}$ by observing that if one wants to predict a solution for one problem taken from $\mathcal{T}$ from the rest of the examples, there may exist triples that make a wrong prediction, as suggested in [16]. The situation may be better analyzed in terms of pairs, as shown now.

Indeed look at Table 1. It exhibits three Boolean pairs such that $\mathtt{a}\!:\!\mathtt{b}\!::\!\mathtt{c}\!:\!\mathtt{d}$ and $\mathtt{a}'\!:\!\mathtt{b}'\!::\!\mathtt{c}\!:\!\mathtt{d}$ hold in all columns, except the last one (column 'S', as in solution). Note that the 'D' columns (first two columns, 'D' as in disagreement) show the possible patterns expressing that $\mathtt{a}$ and $\mathtt{b}$ differ in the same way as $\mathtt{c}$ and $\mathtt{d}$ and as $\mathtt{a}'$ and $\mathtt{b}'$.[1] The 'A' columns (as in agreement) show all the ways $\mathtt{a}$ and $\mathtt{b}$ agree, while $\mathtt{c}$ and $\mathtt{d}$ agree, and $\mathtt{a}'$ and $\mathtt{b}'$ also agree, maybe in different manners.[2] If we take out the value of $\mathtt{d}$ in the column 'S' and we try to predict it from the other values from this column, the equation $\mathtt{a}\!:\!\mathtt{b}\!::\!\mathtt{c}\!:\!y$ yields the good result (i.e., $0\!:\!1\!::\!0\!:\!y$ gives $y = 1$, i.e., the value of $\mathtt{d}$ in the table, column 'S'), while the equation $\mathtt{a}'\!:\!\mathtt{b}'\!::\!\mathtt{c}\!:\!y$ gives a wrong result (i.e., $0\!:\!0\!::\!0\!:\!y$ gives $y = 0$, whereas $\mathtt{d} = 1$ in the table, column 'S').

So, for each pair, like $(\mathtt{a}, \mathtt{b})$ or $(\mathtt{a}', \mathtt{b}')$ in the table, one may count the numbers of times where the pairs leads to a correct and to a wrong prediction for an

---

[1] $D(0/1)$ indicates the disagreement between $\mathtt{a}$ and $\mathtt{b}$ (respectively between $\mathtt{c}$ and $\mathtt{d}$ and between $\mathtt{a}'$ and $\mathtt{b}'$) when the former is equal to 0 and the latter is equal to 1. $D(1/0)$ is the reverse disagreement.

[2] $A(u, v, w)$ means that $\mathtt{a} = \mathtt{b} = u$, $\mathtt{c} = \mathtt{d} = v$ and $\mathtt{a}' = \mathtt{b}' = w$.

	$D(0/1)$	$D(1/0)$	$A(0,0,0)$	$A(0,0,1)$	$A(0,1,0)$	$A(0,1,1)$	$A(1,0,0)$	$A(1,0,1)$	$A(1,1,0)$	$A(1,1,1)$	S
a	0	1	0	0	0	0	1	1	1	1	0
b	1	0	0	0	0	0	1	1	1	1	1
c	0	1	0	0	1	1	0	0	1	1	0
d	1	0	0	0	1	1	0	0	1	1	1
a$'$	0	1	0	1	0	1	0	1	0	1	0
b$'$	1	0	0	1	0	1	0	1	0	1	0

Table 1. Double pairing of pairs $(\mathtt{a},\mathtt{b})$, $(\mathtt{c},\mathtt{d})$ and $(\mathtt{a}',\mathtt{b}')$: Analogy breaking on S.

example taken from the case base. This provides a basis for favoring triples containing pairs leading often to good predictions, in the voting procedure.

The above idea of looking at pairs of cases can be related to the reading of a pair of cases (a, b) as a potential rule expressing either that the change from $\mathtt{x}^a$ to $\mathtt{x}^b$ explains the change from $\mathtt{y}^a$ to $\mathtt{y}^b$, whatever the context (encoded by the features where the two examples agree), or that the change from $\mathtt{x}^a$ to $\mathtt{x}^b$ does not modify the solution (if $\mathtt{y}^a = \mathtt{y}^b$). This view of pairs as rules has already been proposed in CBR for finding adaptation rules [10, 6, 7] and later in an analogical proportion-based inference perspective in [4, 3].

So, roughly speaking, we are interested in a preprocessing process, in order to discover *analogy breakings* in $\mathcal{T}$. By an analogy breaking, we mean the existence of a quadruple of cases (a, b, c, d) such that (i) $\mathtt{x}^a : \mathtt{x}^b :: \mathtt{x}^c : \mathtt{x}^d$ holds, while (ii) $\mathtt{y}^a : \mathtt{y}^b :: \mathtt{y}^c : \mathtt{y}^d$ *does not* hold. If some analogy breaking(s) can be found in $\mathcal{T}$, this means that the partially unknown Boolean function associating to a problem a solution (or a class) cannot be affine [5]. In such a situation, analogical inference cannot be blindly applied with any triple, and we should take into account the analogy breaking(s), by introducing some further restrictions in the choice of the suitable triples.

More precisely, the idea is to make a preliminary preprocessing of the pairs $(a, b) \in \mathtt{CB}^2$, by associating with each of them a *competence*. The intuition behind this notion is that the more a case pair is competent for solving problems, the more it can play a role during the voting and selection process. To assess the competence of a pair $(a, b) \in \mathtt{CB}^2$, it has to be compared to other pairs $(c, d) \in \mathtt{CB}^2$ such that the triple (a, b, c) can be used to solve the problem $\mathtt{x}^d$ by extrapolation. When the outcome y of the extrapolation is equal to $\mathtt{y}^d$, then it increases the competence of the case pair (a, b). Otherwise, it lowers it. The definition of competence is detailed in the next section.

The case pair competence can be used at problem-solving time according to different strategies. Section 3.3 presents some of these strategies that are experimentally evaluated in a Boolean setting in Section 4.

3.2 Case pair competence: definition

Let (a, b) be a pair of source cases: $a = (\mathtt{x}^a, \mathtt{y}^a) \in \mathtt{CB}$ and $b = (\mathtt{x}^b, \mathtt{y}^b) \in \mathtt{CB}$. The *competence* of the pair (a, b) is defined by two scores: the support and the confidence of (a, b), defined below following the principle presented above.

First, let $\texttt{SolvableBy}(a, b)$ be the set of source case pairs $(c, d) \neq (a, b)$ such that the triple (a, b, c) can be used to solve $\mathbf{x}^d$ by extrapolation: $\mathbf{x}^a\!:\!\mathbf{x}^b\!::\!\mathbf{x}^c\!:\!\mathbf{x}^d$ and the equation $\mathbf{y}^a\!:\!\mathbf{y}^b\!::\!\mathbf{y}^c\!:\!y$ is solvable (and so, its solution y is unique in a nominal representation). Formally:

$$\texttt{SolvableBy}(a, b) = \left\{ (c, d) \in \texttt{CB}^2 \;\middle|\; \begin{array}{l} (c, d) \neq (a, b), \quad \mathbf{x}^a\!:\!\mathbf{x}^b\!::\!\mathbf{x}^c\!:\!\mathbf{x}^d \\ \text{and the equation } \mathbf{y}^a\!:\!\mathbf{y}^b\!::\!\mathbf{y}^c\!:\!y \text{ is solvable} \end{array} \right\}$$

In other words, $\texttt{SolvableBy}(a, b)$ is the set of source case pairs such that c can be adapted into a solution of $\mathbf{x}^d$ using (a, b) as an adaptation rule (without considering the trivial case when $(a, b) = (c, d)$). The support of (a, b), $\texttt{supp}(a, b)$, is simply the number of such pairs:

$$\texttt{supp}(a, b) = |\texttt{SolvableBy}(a, b)|$$

Among the $(c, d) \in \texttt{SolvableBy}(a, b)$ some leads to a correct solution $(y = \mathbf{y}^d)$ and some does not. The formers constitute the following set:

$$\texttt{CorrectlySolvableBy}(a, b) = \left\{ (c, d) \in \texttt{SolvableBy}(a, b) \mid \mathbf{y}^a\!:\!\mathbf{y}^b\!::\!\mathbf{y}^c\!:\!\mathbf{y}^d \right\}$$

For example, if $\texttt{supp}(a, b) = 6$ and $|\texttt{CorrectlySolvableBy}(a, b)| = 4$, it means that (a, b), considered as a rule, has been tested 6 times on the case base and has given 4 correct answers. Thus, the proportion of correct answers is $4/6 = 2/3$. This proportion is called the confidence of (a, b), denote by $\texttt{conf}(a, b)$. A special case has to be considered when $\texttt{supp}(a, b) = 0$. This means that the "adaptation rule" (a, b) cannot be tested on the case base. In such a situation, the value of the confidence is set to 0.5 (better than a confidence of, say, 3/7 for which the rule fails more often then it succeeds and worse then a confidence of 4/7 for which it succeeds more often then in fails). To summarize, the confidence of a pair (a, b) is:

$$\texttt{conf}(a, b) = \begin{cases} \dfrac{|\texttt{CorrectlySolvableBy}(a, b)|}{\texttt{supp}(a, b)} & \text{if } \texttt{supp}(a, b) \neq 0 \\ 0.5 & \text{otherwise} \end{cases}$$

3.3 Using case pair competence for selection and vote strategies

Given a target problem $\mathbf{x}^{\texttt{tgt}}$, extrapolation consists in retrieving triples $(a, b, c) \in \texttt{CB}^3$ such that $\mathbf{x}^a\!:\!\mathbf{x}^b\!::\!\mathbf{x}^c\!:\!\mathbf{x}^{\texttt{tgt}}$ and in adapting this triple by solving the equation $\mathbf{y}^a\!:\!\mathbf{y}^b\!::\!\mathbf{y}^c\!:\!y$ for each such triples (the triples (a, b, c) for which the equation has no solution are not considered). So, the result of extrapolation is the set $\mathcal{R}$ of $((a, b, c), y) \in \texttt{CB}^3 \times \mathcal{S}$, y being the result of the extrapolation of (a, b, c) in order to solve $\mathbf{x}^{\texttt{tgt}}$. Now, the question is how to consider all these solutions y to propose a sole solution $\mathbf{y}^{\texttt{tgt}}$ of $\mathbf{x}^{\texttt{tgt}}$. Four strategies for that purpose are detailed below.

The first one, called $\texttt{withoutComp}$, just makes a vote on all values of y, regardless of the competences. The proposed solution is thus

$$\mathbf{y}^{\texttt{tgt}} = \operatorname*{argmax}_{\widehat{y}} |\{((a, b, c), y) \in \mathcal{R} \mid y = \widehat{y}\}|$$

This is the strategy used in [15] and the baseline for the evaluation.

The second strategy, called `allConf`, considers all the $((a,b,c),y) \in \mathcal{R}$ and make a vote weighted by the confidence:

$$y^{\texttt{tgt}} = \underset{\widehat{y}}{\operatorname{argmax}} \sum_{((a,b,c),y) \in \mathcal{R}, y=\widehat{y}} \texttt{conf}(a,b)$$

The third strategy, called `topConf`, considers only the $((a,b,c),y) \in \mathcal{R}$ with the highest confidence, then makes a vote among them. Formally:

$$\text{with } \texttt{conf}_{\max} = \max\left\{\texttt{conf}(a,b) \mid ((a,b,c),y) \in \mathcal{R}\right\}$$
$$\text{and } \mathcal{R}^* = \left\{((a,b,c),y) \in \mathcal{R} \mid \texttt{conf}(a,b) = \texttt{conf}_{\max}\right\}$$
$$y^{\texttt{tgt}} = \underset{\widehat{y}}{\operatorname{argmax}} \left|\left\{((a,b,c),y) \in \mathcal{R}^* \mid y = \widehat{y}\right\}\right| \tag{1}$$

The fourth strategy, called `topConfSupp`, is similar to the previous one, except that it uses both the confidence and the support to make a preference. More precisely, it is based on the preference relation $\succcurlyeq$ on case pairs defined below (for $(a,b),(a',b') \in \texttt{CB}^2$:

$$(a,b) \succcurlyeq (a',b') \quad \text{if} \quad \left| \begin{array}{l} \texttt{conf}(a,b) > \texttt{conf}(a',b') \textbf{ or} \\ (\texttt{conf}(a,b) = \texttt{conf}(a',b') \textbf{ and } \texttt{supp}(a,b) \geq \texttt{supp}(a',b')) \end{array} \right.$$

In other words, confidence is the primary criterion, but in case of equality, the higher the support is, the more competent the case pair (a,b) is considered. For instance, if $\texttt{conf}(a,b) = \texttt{conf}(a',b') = 0.75$, $\texttt{supp}(a,b) = 8$ and $\texttt{supp}(a',b') = 4$, then (a,b) gives the good answer in 6 situations over 8, whereas (a',b') gives the good answer in 3 situations over 4. In this example, (a,b) is strictly preferred to (a',b') —$(a,b) \succ (a',b')$. Now, let $\mathcal{R}^*$ be the set of $((a,b,c),y) \in \mathcal{R}$ such that (a,b) is maximal for $\succcurlyeq$. Then, $y^{\texttt{tgt}}$ results from a vote, as described above in equation (1).

The interest of considering a triple (a,b,c) in the voting procedure at the end of the inference process is evaluated in terms of the competence of the pair (a,b). Since analogical proportions are stable under central permutation, one might think of considering the pair (a,c) as well. Preliminary investigations using different combinations (minimum, maximum, sum or product of the confidences of (a,b) and (a,c)) have not shown any clear improvement with respect to the simple use of the competence of (a,b); it is why we have restricted ourselves to this latter type of competence assessment. However, these preliminary investigations were only based on the `allConf` strategy, so it deserves to be reconsidered: this constitutes a future work.

4   Evaluation

The objective of the evaluation is to study the impact of the strategies for case pair selection and vote presented before on various types of Boolean functions.

4.1 Experiment setting

In the experiment, $\mathcal{P} = \mathbb{B}^8$ and $\mathcal{S} = \mathbb{B}$. $\rightsquigarrow$ is assumed to be functional: $\rightsquigarrow = \mathtt{f}$, meaning that $\mathtt{y}$ is a solution to $\mathtt{x}$ if $\mathtt{y} = \mathtt{f}(\mathtt{x})$.

The function $\mathtt{f}$ is randomly generated using the following generators that are based on two normal forms, with the purpose of having various types of functions:

DNF $\mathtt{f}$ is generated in a disjunctive normal form, i.e., $\mathtt{f}(\mathtt{x})$ is a disjunction of n_{disj} conjunctions of literals, for example $\mathtt{f}(\mathtt{x}) = (\mathtt{x}_1 \wedge \neg \mathtt{x}_7) \vee (\neg \mathtt{x}_3 \wedge \mathtt{x}_7 \wedge \mathtt{x}_8) \vee \mathtt{x}_4$. The value of n_{disj} is randomly chosen uniformly in $\{3, 4, 5\}$. Each conjunction is generated on the basis of two parameters, $p^+ > 0$ and $p^- > 0$, with $p^+ + p^- < 1$: each variable $\mathtt{x}_i$ occurs in the disjunct in a positive (resp. negative) literal with a probability p^+ (resp., p^-). In the experiment, the values $p^+ = p^- = 0.1$ were chosen.[3]

Pol $\mathtt{f}$ is generated in polynomial normal form: it is the same as **DNF**, except that the disjunctions ($\vee$) are replaced with exclusive or's ($\oplus$). As only positive literals occur in the polynomial normal form, the parameter $p^- = 0$.

The case base **CB** is generated randomly, with the values for its size: $|\mathtt{CB}| \in \{32, 64, 96, 128\}$, i.e. $|\mathtt{CB}|$ is between $\frac{1}{8}$ and $\frac{1}{2}$ of $|\mathcal{P}| = 2^8 = 256$. Each source case $(\mathtt{x}, \mathtt{y})$ is generated as follows: $\mathtt{x}$ is randomly chosen in $\mathcal{P}$ with a uniform distribution and $\mathtt{y} = \mathtt{f}(\mathtt{x})$.

Let `#tgt_pb` be the number of target problems posed to the system, `#ans` be the number of (correct or incorrect) answers (`#tgt_pb` − `#ans` is the number of target problems for which the system fails to propose a solution), and `#corr_ans` be the number of correct answers. For each selection and vote strategy, the following scores are computed:

The error rate `%err` is the average of $\left(1 - \dfrac{\texttt{\#corr\_ans}}{\texttt{\#ans}}\right) \times 100 \in [0, 100]$.

The answer rate `%ans` is the average of the ratios $\dfrac{\texttt{\#ans}}{\texttt{\#tgt\_pb}} \times 100 \in [0, 100]$.

If the system always gives an answer (correct or not) then `%ans` $= 100$.

The average is computed on 1 million problem solving for each function generator, requiring the generation of 1420 $\mathtt{f}$ for each of them. The average computing time of a CBR session (retrieval and adaptation for solving one problem) is about $2\,\mathrm{ms}$ on a current standard laptop.

---

[3] A generator **CNF**, generating formulas in CNF (conjunctive normal form: conjunction of disjunctions of literals) could also have been considered. However, this does not add anything new since it is dual with the **DNF** generator for two reasons. First, the drawn inferences are code-independent, meaning that replacing the attributes by their negations does not change the result of the inference, in particular, for $\mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{d} \in \mathbb{B}$, $\mathtt{a}\!:\!\mathtt{b}\!:\!:\!\mathtt{c}\!:\!\mathtt{d}$ iff $\neg\mathtt{a}\!:\!\neg\mathtt{b}\!:\!:\!\neg\mathtt{c}\!:\!\neg\mathtt{d}$. Second, if $\mathtt{f}$ is obtained from the **DNF** generator then $\neg\mathtt{f}$ can be put easily in a function g written in CNF using De Morgan laws, and the distribution of g obtained this way would be the same as the distribution from a **CNF** generator with the same parameters.
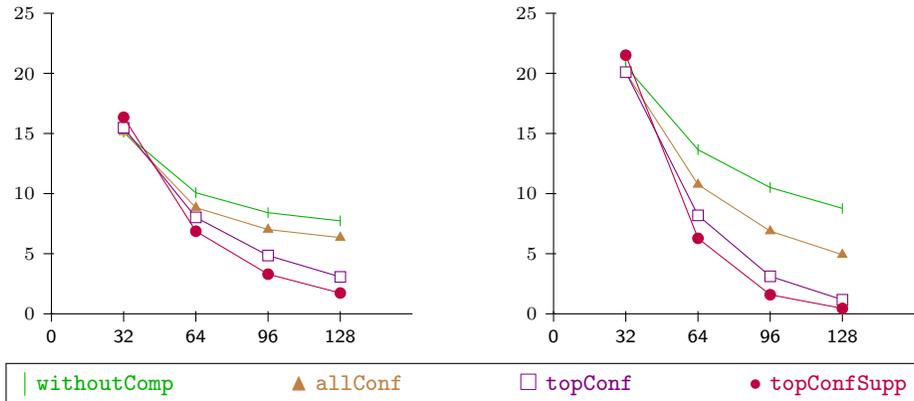
Fig. 1. Error rate function of |CB|, for each generator (DNF at the left, Pol at the right).

For the sake of reproducibility, the code for this experiment is available at https://tinyurl.com/analogyCBRTests, with the detailed results (generated functions and details of the evaluation).

4.2 Results

		|CB| = 32		|CB| = 64		|CB| = 96		|CB| = 128	
		%err	%ans	%err	%ans	%err	%ans	%err	%ans
DNF	withoutComp	**15.1**		10.1		8.4		7.7	
	allConf	15.1	97.5	8.8	100.0	7.0	100.0	6.3	100.0
	topConf	15.5		8.0		4.9		3.1	
	topConfSupp	16.4		**6.9**		**3.3**		**1.7**	
POL	withoutComp	20.6		13.7		10.5		8.8	
	allConf	**20.1**	95.8	10.8	100.0	6.9	100.0	4.9	100.0
	topConf	20.1		8.2		3.1		1.2	
	topConfSupp	21.5		**6.3**		**1.6**		**0.5**	

Table 2. %err and %ans for the different selection and vote strategies for the different generators.

Table 2 presents the error rate and the answer rate for the different case selection and vote strategies for the two different generators with an application on the different case base sizes. Error rate curves are given in Figure 1.

Given a function generator and a case base size, the answer rate is the same for the four strategies because all case pair selection strategies provide results

for a problem that could be solved, without using competences, by `withoutComp` (i.e. if a triple was found to solve a case $\mathtt{x^{tgt}}$ by the `withoutComp` strategy, this triple is considered by the three case pair selection strategies and either it will participate in solving $\mathtt{x^{tgt}}$ or it exists another "better" triple according to the selection procedure. The answer rate is high for all the methods: over 96% for $|\mathtt{CB}| = 32$ and 100% for $|\mathtt{CB}| \geq 64$.

Except for $|\mathtt{CB}| = 32$, which seems to be a too small training data set for computing competences, the error rate shows that the hypothesis of pair selection improves the precision. For both generators, all pair selection strategies give better results than the baseline (`withoutComp`). However, the improvement is rather different depending of the selection strategy: the more the selection of pairs is constrained, the more the error rate decreases. `allConf` decreases the error rate a little bit, `topConf` decreases the error rate a little bit more, and the best results are given by `topConfSupp`.

The benefit of all strategies is related to the case base size: the more the case base contains cases for competence acquisition, the better the results are. Comparing to the baseline, the benefit of the best selection strategy `topConfSupp` is noteworthy. Even if the error rate is already rather good with the baseline, and especially with a 100% answer rate, `topConfSupp` improves it, making it close to a 100% of correct answers. For `DNF`, according to the size of the case base (64, 96 and 128), the error rate `%err` decreases from 10.1 to 6.9 (decreasing of 32%), from 8.4 to 3.3 (decreasing of 61%) and from 7.7 to 1.7 (decreasing of 78%). For `Pol`, the results are even more impressive: according to the size of the case base (64, 96 and 128), the error rate `%err` decreases from 13.7 to 6.3 (decreasing of 54%), from 10.5 to 1.6 (decreasing of 85%), and from 8.8 to 0.5 (decreasing of 94%).

So, these first experimental results show that from a given case base size, the `topConfSupp` strategy overcomes all others and decreases drastically the error rate, while using less triples.

5   Discussion and Related Work

In this section, the approach presented in this paper is compared to related work in CBR according to two viewpoints: the notion of competence in CBR and the adaptation knowledge learning approaches.

Competence in CBR. In [15], three types of CBR processes are distinguished, in particular extrapolation, that retrieves and reuses cases by triples and approximation, that retrieves and reuses cases by singletons. It is argued here that previous researches on competence are related to approximation, whereas the work presented in this paper considers a notion of competence related to extrapolation.

The notion of competence in CBR is used in general for the purpose of case base maintenance, either for deleting the least competent cases [19] or adding the most competent ones [21]. In these previous studies, competence is assessed

to individual source cases, in relation to other cases from the case base. In particular, in the seminal paper [19], the competence of cases is assessed by putting source cases into categories (from pivotal cases who are the most competent ones to auxiliary cases), these categories being defined with the help of the binary relation of adaptability between a case and a problem. Thus, this notion of competence is linked with the approximation process (considering individually source cases).

By contrast, the current paper is concerned by competence related to the extrapolation process: cases are retrieved by triples. The competence of a triple $(a, b, c) \in \mathtt{CB}^3$ is reduced to the competence of a pair $(a, b) \in \mathtt{CB}^2$, which is related to the set of the other pairs $(c, d) \in \mathtt{CB}^2$. A common point of these two notions of competence is that the competence of an object (an object being a case for approximation and a case pair for extrapolation) is not an intrinsic property of the object, but is related to other objects (from $\mathtt{CB}$ or $\mathtt{CB}^2$).

A minor difference between previous studies on competence and the one defined in this paper is related to the use of competence: case base maintenance for the formers and problem-solving for the latter.

Relations with adaptation knowledge learning. The work presented in this paper has strong links with the issue of adaptation knowledge learning (AKL). The adaptation considered here is the one that follows the retrieval of a sole case (i.e., it is a single case adaptation). Such an adaptation has profit of the adaptation knowledge $\mathtt{AK}$, that can be informally defined by:

$$\mathtt{AK} = \text{"How does the solution changes when the problem changes."}$$

The approach generally applied for AKL is modelled in the seminal work of Kathleen Hanney and Mark T. Keane [10]. It uses the case base for learning adaptation knowledge according to the following principle. A set $\mathtt{TS}$ of source case pairs (a, b), with $a \neq b$, is built, either by considering all the distinct pairs from $\mathtt{CB}$ or by considering only the pairs (a, b) where a and b are judged as enough similar, according to some criterion. Then, $\mathtt{TS}$ is used as training set of a supervised learning process: for each pair (a, b) the input of an example is the pair $(\mathtt{x}^a, \mathtt{x}^b)$ and its output is the pair $(\mathtt{y}^a, \mathtt{y}^b)$. The supervised learning process provides a model of this knowledge $\mathtt{AK}$, used by the adaptation process.

Several work are based on this general scheme. In [12], $\mathtt{AK}$ consists in the representation of "adaptation cases". In [6], different techniques are used, in particular, decision tree induction and ensemble learning techniques. In [7] the frequent closed itemset extraction is used. The expert interpretation following this extraction produces adaptation rules to be added to $\mathtt{AK}$. In [9], similar techniques as in [7] are used (formal concept analysis and frequent closed itemset extraction are similar data-mining techniques), but, in this work, negative cases (i.e., pairs $(\mathtt{x}, \mathtt{y}) \in \mathcal{P} \times \mathcal{S}$ such that $\mathtt{y}$ is not a correct solution of $\mathtt{x}$) are used, which improves significantly the results of the learning process. In [11], an ensemble approach provides adaptation rules with categorical features.

The work presented in this paper could also be considered as an AKL approach. In fact, in this paper, the term of adaptation rule for considering a case

11

pair (a, b) has been used. Let us make this idea more accurate. Let $\sim$ be the relation defined, for (a, b) and (a', b'), two case pairs, by:

$$(a, b) \sim (a', b') \quad \text{if} \quad \mathtt{x}^a : \mathtt{x}^b :: \mathtt{x}^{a'} : \mathtt{x}^{b'} \text{ and } \mathtt{y}^a : \mathtt{y}^b :: \mathtt{y}^{a'} : \mathtt{y}^{b'}$$

For analogical proportions on nominal representations defined in Section 2.2, $\sim$ is an equivalence relation[4]. Thus, solving a problem $\mathtt{x}^{\mathtt{tgt}}$ by extrapolation from a triple $(a, b, c) \in \mathtt{CB}^3$ or from a triple $(a', b', c) \in \mathtt{CB}$ (with the same c) such that $(a, b) \sim (a', b')$ will give the same result: extrapolation is independent from the choice of a representative of the equivalent class of (a, b) for $\sim$. Such an equivalence class $C\ell$ can be used as an adaptation rule (where c is the retrieved case and $\mathtt{x}^{\mathtt{tgt}}$ is the problem to be solved):

> **with** (a, b) arbitrarily chosen in $C\ell$
>
> **if** $\mathtt{x}^a : \mathtt{x}^b :: \mathtt{x}^c : \mathtt{x}^{\mathtt{tgt}}$ and $\mathtt{y}^a : \mathtt{y}^b :: \mathtt{y}^c : y$ has a solution
>
> **then** this solution is a plausible solution to $\mathtt{x}^{\mathtt{tgt}}$

Thus, the set of equivalent classes of the restriction of $\sim$ to $\mathtt{CB}^2$ gives a set of candidate adaptation rules, but all these rules are not equivalently interesting: some gives more plausible results than the other ones. So, a criterion has to be defined for making a preference between these rules and, if it is decided to apply all of them, to do so by making a weighted vote (the more an adaptation rule is preferred, the higher its weight in the vote should be).

A simple way of doing this (used, e.g., in [7]) consists in using the cardinality of $C\ell$. This can be related to the notion of competence of a case pair: if $(a, b) \in C\ell$ then $|C\ell| = \mathtt{supp}(a, b) \times \mathtt{conf}(a, b)$. One limitation of this approach is that it counts only the examples (supporting the rule), not the counterexamples (penalizing the rule). By contrast, the approach presented in this paper takes into account counterexamples. For example, if $\mathtt{conf}(a, b) = 1/3$, then, for each example of the rule, there are two counterexamples, so, even if $\mathtt{supp}(a, b)$ is large, the rule associated to (a, b) is, at best, questionable.

Another difference with the work of [7] is that, in [7], when several case pairs have the same variations only on a subset of the features, they are still used to build an adaptation rule. For example, if (a, b) and (a', b') are two source case pairs such that for *most* attributes i, $\mathtt{x}_i^a : \mathtt{x}_i^b :: \mathtt{x}_i^{a'} : \mathtt{x}_i^{b'}$, the rules built on these common attributes are considered, neglecting the other attributes. In a formal framework in which analogies are rare (for example, when there are features with real number values), it could be justified to replace the exact analogical proportion with a gradual analogical proportion [8] in the approach described in this paper. Studying it constitutes a potential future work.

This discussion shows how some ideas related to AKL from the case base can be easily reformulated in the framework of analogical proportions: the links so established between these two fields is therefore potentially fruitful.

---

[4] Reflexivity and symmetry are direct consequences of the postulates with the same name. By contrast, there exist analogical proportions for which transitivity does not hold [14].

6 Conclusion

Classical case-based reasoning relies on the individual similarities of the problem at hand with each already solved problem that is known. We have shown that it may be also of interest to consider triples of cases (a, b, c) in order to equalize the change from a to b with the change from c to the problem to be solved with its tentative solution. This is the basis of analogical extrapolation based on analogical proportions. Still it has been observed that some triples may lead to wrong inferences.

In this paper, we have proposed to discriminate triples according to an evaluation of the "competence" of the pairs involved in the triples. Indeed an analogical proportion "a is to b as c is to d" can be viewed as establishing a parallel between two pairs. The differences between the components of a pair of problems are naturally related to the differences between solutions, but this relation may depend on the context expressed by the component values that do not change. We have shown that it was possible, at least to some extent, to evaluate the competence of pairs for selecting "good" triples and improving analogical inference results. This contributes to confirm the interest of analogical extrapolation for case-based reasoning.

Several future works follow these studies.

The first one has been mentioned at the end of Section 3. It consists, when choosing a triple (a, b, c), in considering not only the competence of the pair (a, b) but also the competence of the pair (a, c). Preliminary studies with the strategy `allConf` where carried out that does not give significant changes in the result. However, this may be different for the other strategies, and this remains to be studied.

Another future work will be to transfer contributions from the adaptation knowledge learning field to improve furthermore the performance of analogical extrapolation (cf. Section 5). In particular, a promising direction of work is the use of a base of negative cases, as it has been used in [9].

Finally, the competence of case pairs can be used in order to associate to a solution proposed by extrapolation an indication of its plausibility, according to the following idea: the higher are the competences of the case pairs used for giving a solution, the more plausible the proposed solution is. This can be used in order to combine analogical extrapolation with other approaches to CBR that also provides an indication of plausibility.

References

1. Aamodt, A., Plaza, E.: Case-based Reasoning: Foundational Issues, Methodological Variations, and System Approaches. AI Communications **7**(1) (1994) 39–59
2. Billingsley, R., Prade, H., Richard, G., Williams, M.A.: Towards analogy-based decision - A proposal. In H. Christiansen, H.J., ed.: Proc. 12th Conf. on Flexible Query Answering Systems (FQAS'17), London, Jun. 21-23. LNAI 10333, Springer (2017) 28–35

3. Bounhas, M., Prade, H., Richard, G.: Analogical classification: A rule-based view. In: Proc. of the International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems, Springer (2014) 485–495

4. Correa, W.F., Prade, H., Richard, G.: Trying to understand how analogical classifiers work. In: Proc. of the International Conference on Scalable Uncertainty Management, Springer (2012) 582–589

5. Couceiro, M., Hug, N., Prade, H., Richard, G.: Analogy-preserving functions: a way to extend Boolean samples. In: Proc. of the 26th Int. Joint Conf. on Artificial Intelligence (IJCAI'17), Morgan Kaufmann, Inc. (2017) 1575–1581

6. Craw, S., Wiratunga, N., Rowe, R.C.: Learning adaptation knowledge to improve case-based reasoning. Artificial Intelligence **170**(16-17) (2006) 1175–1192

7. d'Aquin, M., Badra, F., Lafrogne, S., Lieber, J., Napoli, A., Szathmary, L.: Case base mining for adaptation knowledge acquisition. In Veloso, M.M., ed.: Proc. of the 20th Int. Joint Conf. on Artificial Intelligence (IJCAI'07), Morgan Kaufmann, Inc. (2007) 750–755

8. Dubois, D., Prade, H., Richard, G.: Multiple-valued extensions of analogical proportions. Fuzzy Sets and Systems **292** (2016) 193–202

9. Gillard, T., Lieber, J., Nauer, E.: Improving Adaptation Knowledge Discovery by Exploiting Negative Cases: First Experiment in a Boolean Setting. In: Proc. of ICCBR 2018 - 26th International Conference on Case-Based Reasoning, Stockholm, Sweden (July 2018)

10. Hanney, K., Keane, M.T.: Learning adaptation rules from a case-base. In Smith, I., Faltings, B., eds.: Advances in Case-Based Reasoning – Proc. of the Third Eur. Workshop, EWCBR'96. LNAI 1168, Springer Verlag, Berlin (1996) 179–192

11. Jalali, V., Leake, D., Forouzandehmehr, N.: Learning and applying adaptation rules for categorical features: An ensemble approach. AI Communications **30**(3-4) (2017) 193–205

12. Jarmulak, J., Craw, S., Rowe, R.: Using case-base data to learn adaptation knowledge for design. In: Proc. of the 17th Int. Joint Conf. on Artificial Intelligence (IJCAI'01), Morgan Kaufmann, Inc. (2001) 1011–1016

13. Kolodner, J.: Case-Based Reasoning. Morgan Kaufmann, Inc. (1993)

14. Lepage, Y.: Proportional analogy in written language data. In Gala, N., Rapp, R., Bel-Enguix, G., eds.: Language, Production, Cognition and the Lexicon. Text, Speech and Language Technology 48. Springer International Publishing Switzerland (2014) 151–173

15. Lieber, J., Nauer, E., Prade, H., Richard, G.: Making the Best of Cases by Approximation, Interpolation and Extrapolation. In: ICCBR 2018 - 26th International Conference on Case-Based Reasoning, Stockholm, Sweden, Springer (July 2018)

16. Prade, H., Richard, G.: A discussion of analogical-proportion based inference. In Sánchez-Ruiz, A.A., Kofod-Petersen, A., eds.: Proc. ICCBR'17 Workshops (CAW, CBRDL, PO-CBR), Doctoral Consortium, and Competitions co-located with the 25th Int. Conf. on Case-Based Reasoning (ICCBR'17), Trondheim, June 26-28. Volume 2028 of CEUR Workshop Proceedings. (2017) 73–82

17. Richter, M.M., Weber, R.O.: Case-based reasoning, a textbook. Springer (2013)

18. Riesbeck, C.K., Schank, R.C.: Inside Case-Based Reasoning. Lawrence Erlbaum Associates, Inc., Hillsdale, New Jersey (1989) Available on line.

19. Smyth, B., Keane, M.T.: Remembering to forget. In: Proc. of the 14th Int. Joint Conf. on Artificial Intelligence (IJCAI'95), Montréal (1995)

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

21. Zhu, J., Yang, Q.: Remembering to add: competence-preserving case-addition policies for case base maintenance. In: Proc. of the 16th Int. Joint Conf. on Artificial Intelligence (IJCAI'99). (1999) 234–241