



Making the Best of Cases by Approximation, Interpolation and Extrapolation

Jean Lieber^{1(✉)}, Emmanuel Nauer^{1(✉)}, Henri Prade^{2,3(✉)}, and
Gilles Richard^{2(✉)}

¹ Université de Lorraine, CNRS, Inria, LORIA, 54000 Nancy, France
Jean.Lieber@loria.fr

² IRIT, University of Toulouse, Toulouse, France

³ QCIS, University of Technology, Sydney, Australia

Abstract. Case-based reasoning usually exploits source cases (consisting of a source problem and its solution) *individually*, on the basis of the similarity between the target problem and a particular source problem. This corresponds to approximation. Then the solution of the source case has to be adapted to the target. We advocate in this paper that it is also worthwhile to consider source cases by two, or by three. Handling cases by two allows for a form of interpolation, when the target problem is between two similar source problems. When cases come by three, it offers a basis for extrapolation. Namely the solution of the target problem is obtained, when possible, as the fourth term of an analogical proportion linking the three source cases with the target, where the analogical proportion handles both similarity and dissimilarity between cases. Experiments show that interpolation and extrapolation techniques are of interest for reusing cases, either in an independent or in a combined way.

Keywords: Analogy · Analogical proportion · Approximation
Interpolation · Extrapolation

1 Introduction

Case-based reasoning (CBR) [18] aims at solving a new problem—the *target problem*—thanks to a set of cases (the *case base*), where a case is a pair consisting of a problem and a solution of this problem. A *source case* is a case from the case base, consisting of a *source problem* and one of its solutions. The classical approach to CBR consists in (i) selecting source cases *similar* to the target problem and (ii) adapting them to solve it. In such a view, the target and the source are compared in terms of similarity, which is a (two-valued or gradual) *binary* relation, while information about the way they *differ* is not really considered in general.

In this paper, in addition to the binary relation of similarity (which is the basis of approximation), two other relations are considered: the *betweenness* and the *analogical proportion*, which in a way or another leave room to dissimilarity.

Betweenness is a *ternary* relation stating that an object is “between” two other objects. It is used as an *interpolation* principle: if the target problem is between two source problems then it is plausible that a solution of it is between the solutions of these source problems. In a graded setting, this suggests that if the target problem is closer to one of two source problems, its solution should be closer as well to the solution to the corresponding source problem. In the Boolean setting, betweenness view may lead to several potential solutions, except if the solutions of the two source problems coincide.

Analogical proportion is a *quaternary* relation: four objects a , b , c and d are in analogical proportion if “ a is to b as c is to d ”. A logical modeling of it [14, 16] has pointed out that it expresses that “ a differs from b as c differs from d (and vice-versa)”, and that “what a and c have in common, b and d have it also”. Thus, analogical proportion is a matter of both dissimilarity and similarity. The fact that we are no longer considering similarity only, enables us to escape the strict vicinity of known cases, and to perform a form of adaptation for free. More precisely, in a CBR perspective, we use it as an extrapolation principle: if the target problem and three source problems are in analogical proportion, the solutions of these four problems are (likely to be) in analogical proportion as well. Such an analogical jump enables us to extrapolate the solution of the target problem from the solutions of three distinct source problems. An illustration of this is given in [2] where in three distinct situations the recommended actions are respectively to (i) serve tea without milk without sugar, (ii) serve tea with milk without sugar, (iii) serve tea without milk with sugar, while in a fourth situation that makes an analogical proportion with the three others, the action to do would be (iv) “serve tea with milk with sugar”.

The approach described in this paper combines the use of closeness, betweenness and analogical proportion for a knowledge-light approach to CBR. In fact, the only source of knowledge used in the inference lies in the case base: there is no domain knowledge nor adaptation knowledge and the similarity is based on some distance function.

Section 2 introduces the notions and notations used throughout the paper and the assumptions it is based on. Section 3 describes the approach for applying approximation, interpolation and extrapolation to CBR. Section 4 provides an evaluation in a Boolean setting. Section 5 presents a discussion and a comparison with related work, while Sect. 6 points out lines for future research.

2 Definitions, Notations and Assumptions

In this section, definitions are presented in a Boolean setting (objects are tuples of Boolean values).

2.1 Boolean Setting

Let $\mathbb{B} = \{0, 1\}$ be the set of Boolean values. The Boolean operators are denoted by the connector symbols of propositional logic: for $a, b \in \mathbb{B}$, $\neg a = 1 - a$,

$a \wedge b = \min(a, b)$, $a \vee b = \max(a, b)$, $a \oplus b = |b - a|$ (\oplus is the exclusive or) and $a \equiv b = \neg(a \oplus b)$.

Let p be a positive integer. In the examples, an element of \mathbb{B}^p is noted without parentheses and commas: $(0, 1, 0, 0, 1)$ is simply noted by 01001. The Hamming distance H on \mathbb{B}^p is defined by $H(a, b) = \sum_{i=1}^p |b_i - a_i|$. For example, with $p = 5$, $H(01001, 11011) = 2$.

2.2 CBR

CBR aims at solving the target problem with the help of the case base. It consists most of the time (1) in selecting relevant source cases (retrieval), (2) in reusing these source cases in order to solve the target problem (adaptation). The other classical steps of CBR are not considered in this paper: they concern the validation-repair of the newly formed case and its potential adding to the case base.

Let \mathcal{P} and \mathcal{S} be two sets called the universe of problems and the universe of solutions: a *problem* x (resp., a *solution* y) is by definition an element of \mathcal{P} (resp., of \mathcal{S}). Here, $\mathcal{P} = \mathbb{B}^m$, where $m \geq 1$ is a constant. Similarly, $\mathcal{S} = \mathbb{B}^n$, $n \geq 1$. A binary relation on $\mathcal{P} \times \mathcal{S}$ denoted by \rightsquigarrow and read “has for solution” is assumed to exist. Thus, “ y solves x ” is denoted by $x \rightsquigarrow y$. A *case* is a pair $(x, y) \in \mathcal{P} \times \mathcal{S}$ such that $x \rightsquigarrow y$. The *case base* \mathcal{CB} is a finite set of cases. A *source case* is an element of \mathcal{CB} . The *target problem* is the problem to be solved. Note that \rightsquigarrow is usually not completely known to the CBR system: such a system provides *plausible* solutions to problems, on the basis of what is known of \rightsquigarrow from the case base.

In some situations, it is assumed that \rightsquigarrow is functional. This assumption means that there exists a function $f : \mathcal{P} \rightarrow \mathcal{S}$ such that $x \rightsquigarrow y$ iff $y = f(x)$, for any $x \in \mathcal{P}$ and $y \in \mathcal{S}$.

2.3 Betweenness

Let \mathcal{U} be a set whose elements are represented by numerical features (including Boolean features). Let $a, b, c \in \mathcal{U}$; a is *between* b and c , denoted by $b-a-c$, if for every feature i , $((b_i \leq a_i \leq c_i) \text{ or } (c_i \leq a_i \leq b_i))$.¹ Let $\mathbf{Between}(b, c) = \{a \in \mathcal{U} \mid b-a-c\}$, for $b, c \in \mathcal{U}$. For example, in $\mathcal{U} = \mathbb{B}^5$, $\mathbf{Between}(01001, 11011) = \{01001, 11001, 01011, 11011\}$. For a logical view on betweenness, we refer to [22].

2.4 Analogical Proportion

Let \mathcal{U} be a set. An analogical proportion on \mathcal{U} is a quaternary relation between four elements a, b, c and d of \mathcal{U} , read “ a is to b as c is to d ” and denoted by $a:b::c:d$, having the following properties (see, e.g. [16]), for any $a, b, c, d \in \mathcal{U}$:

¹ When these values are Boolean, this can also be written $(b_i \wedge c_i \Rightarrow a_i) \wedge (a_i \Rightarrow b_i \vee c_i) = 1$.

reflexivity $a:b::a:b$,
symmetry if $a:b::c:d$ then $c:d::a:b$,
central permutation if $a:b::c:d$ then $a:c::b:d$.

In the Boolean setting, the analogical proportion considered is defined on \mathbb{B} by

$$a:b::c:d \text{ if } (a \wedge \neg b \equiv c \wedge \neg d) \wedge (\neg a \wedge b \equiv \neg c \wedge d) = 1$$

that can be read “ a differs from b as c differs from d ” and “ b differs from a as d differs from c ”. This can also be rewritten $b - a = d - c$, but mind that these differences belong to $\{-1, 0, 1\}$ ($-$ is not an operation that is closed in \mathbb{B}). Thus, the patterns $abcd$ such that $a:b::c:d$ are 0000, 1111, 0011, 1100, 0101 and 1010.

This analogical proportion can be extended on $\mathcal{U} = \mathbb{B}^p$ by

$$a:b::c:d \text{ if } a_i:b_i::c_i:d_i \text{ for each } i \in [1, p]$$

Given $a, b, c \in \mathcal{U}$, solving the *analogical equation* $a:b::c:y$ aims at finding the $y \in \mathcal{U}$ satisfying this relation. It may have no solution, e.g., when $a = 0$, $b = 1$ and $c = 1$. The equation $a:b::c:y$ in \mathbb{B} has a solution iff $(a \equiv b) \vee (a \equiv c) = 1$ and, when this is the case, the solution is unique: $y = c \equiv (a \equiv b)$.

3 Reusing Cases by Approximation, Interpolation and Extrapolation

This section describes the three mentioned approaches: approximation, interpolation and extrapolation. For an integer $k \geq 1$, case retrieval can be done by considering ordered sets of k source cases. This principle is detailed in Sect. 3.2 and applied in Sects. 3.3, 3.4 and 3.5 respectively for $k = 1, 2$, and 3. The combination of these three approaches is discussed in Sect. 3.6. Let us start with an example.

3.1 A Basic Example

In order to support the intuition, we consider the following example where a suitable dish type (described via 3 two-valued attributes, i.e., $\mathcal{S} = \mathbb{B}^3$) has to be suggested to an individual (described via 8 two-valued attributes, i.e., $\mathcal{P} = \mathbb{B}^8$). The 8 attributes representing an individual x have the following semantics: x_1 : x suffers from gout, x_2 : x has diabetes, x_3 : x is allergic to nuts, x_4 : x does not eat mammal meat (beef, pork, etc.), x_5 : x needs to have a regular calcium supplement, x_6 : x needs to have a regular iron supplement, x_7 : x likes vegetables, and x_8 : x does not like dairy products. A dish type y is represented via 3 attributes: y_1 : y is a dish with sauce, y_2 : y is based on starchy food (e.g., a pasta dish), y_3 : y is a dish with fish. $x \rightsquigarrow y$ can be read as: y is a suitable dish type for x . For a healthy individual 00010010 (with no specific requirement), all dishes are suitable. Therefore, \rightsquigarrow is not functional in this application, as several types of dishes might be suitable for the same individual. The 3 approaches of case reuse developed in this paper can be applied to this application as follows (where y^j is a suitable class of dish for x^j):

Approximation: If an individual \mathbf{x}^{tgt} is not far from \mathbf{x}^1 , it is plausible that a suitable dish for \mathbf{x}^{tgt} will not be far from a suitable dish \mathbf{y}^1 for \mathbf{x}^1 .

Interpolation: If an individual \mathbf{x}^{tgt} is between \mathbf{x}^1 and \mathbf{x}^2 ; it is plausible that a suitable dish for \mathbf{x}^{tgt} will be between a suitable dish \mathbf{y}^1 for \mathbf{x}^1 and a suitable dish \mathbf{y}^2 for \mathbf{x}^2 .

Extrapolation: If an individual \mathbf{x}^{tgt} is as similar to \mathbf{x}^3 as \mathbf{x}^2 is similar to \mathbf{x}^1 , it is plausible that a suitable dish for \mathbf{x}^{tgt} will be as similar to \mathbf{y}^3 as \mathbf{y}^2 is similar to \mathbf{y}^1 .

3.2 General Principle

Let $k \in \{1, 2, 3\}$: the approach presented here covers approximation ($k = 1$), interpolation ($k = 2$) and extrapolation ($k = 3$). Two $(k + 1)$ -ary relations are considered: $\text{Rel}_{\mathcal{P}}$ on \mathcal{P} and $\text{Rel}_{\mathcal{S}}$ on \mathcal{S} . Ideally, it would be assumed that these relations have the following properties, for $(\mathbf{x}^1, \dots, \mathbf{x}^k, \mathbf{x}^{k+1}) \in \mathcal{P}^{k+1}$ and $(\mathbf{y}^1, \dots, \mathbf{y}^k, \mathbf{y}^{k+1}) \in \mathcal{S}^{k+1}$, with the hypothesis that $\forall j \in [1; k + 1], \mathbf{x}^j \rightsquigarrow \mathbf{y}^j$:

$$\text{if } \text{Rel}_{\mathcal{P}}(\mathbf{x}^1, \dots, \mathbf{x}^k, \mathbf{x}^{k+1}) \text{ then } \text{Rel}_{\mathcal{S}}(\mathbf{y}^1, \dots, \mathbf{y}^k, \mathbf{y}^{k+1})$$

However, this assumption is usually too strong: since the relation \rightsquigarrow is only partially known to the CBR system, it seems odd to have such a certain relationship about \rightsquigarrow given by the pair $(\text{Rel}_{\mathcal{P}}, \text{Rel}_{\mathcal{S}})$. So, only the following relaxed form of this property is assumed:

$$\begin{aligned} &\text{if } \text{Rel}_{\mathcal{P}}(\mathbf{x}^1, \dots, \mathbf{x}^k, \mathbf{x}^{k+1}) \text{ and } \mathbf{x}^j \rightsquigarrow \mathbf{y}^j \text{ for } j \in [1; k + 1] \\ &\text{then it is plausible that } \text{Rel}_{\mathcal{S}}(\mathbf{y}^1, \dots, \mathbf{y}^k, \mathbf{y}^{k+1}) \end{aligned} \tag{1}$$

This property (1) can be used for CBR. Let \mathbf{x}^{tgt} be the target problem. A *candidate* is an ordered set of k cases $((\mathbf{x}^1, \mathbf{y}^1), \dots, (\mathbf{x}^k, \mathbf{y}^k)) \in \text{CB}^k$ such that $\text{Rel}_{\mathcal{P}}(\mathbf{x}^1, \dots, \mathbf{x}^k, \mathbf{x}^{\text{tgt}})$. Based on this notion, the following CBR steps can be specified:

retrieval: The set of candidates is computed.

adaptation: For a candidate $((\mathbf{x}^1, \mathbf{y}^1), \dots, (\mathbf{x}^k, \mathbf{y}^k))$, it is plausible that the solution \mathbf{y}^{tgt} of \mathbf{x}^{tgt} satisfies $\text{Rel}_{\mathcal{S}}(\mathbf{y}^1, \dots, \mathbf{y}^k, \mathbf{y}^{\text{tgt}})$.

Let **Candidates** be the set of candidates. When **Candidates** = \emptyset , the approach fails. Let **potentialSols** be the multiset of $\mathbf{y} \in \mathcal{S}$ such that $(\mathbf{x}, \mathbf{y}) \in \text{Candidates}$. When there are several *distinct* solutions in this multiset, there is a need for a function to integrate these solutions into one. Let **integrate** : **potentialSols** $\mapsto \mathbf{y} \in \mathcal{S}$ be such a function. The nature of this function can be different, depending on the solution space. When $\mathcal{S} = \mathbb{B}$ (or, more generally, a set of low cardinality) then **integrate** can be a simple vote. When $\mathcal{S} = \mathbb{B}^n$, the integration could be a component by component vote, for example:

$$\text{integrate}(\{\{001, 001, 010, 111, 111\}\}) = 011$$

(with an arbitrary choice in case of ties).

3.3 Reusing Cases by Singletons: Approximation

When $k = 1$, candidates are singletons and source cases are considered individually in relation to the target problem. Usually $\text{Rel}_{\mathcal{P}}$ and $\text{Rel}_{\mathcal{S}}$ binary relations are related to the notion of similarity and denoted $\mathbf{x}^1 \simeq \mathbf{x}^2$ and $\mathbf{y}^1 \simeq \mathbf{y}^2$. When a distance dist is available on the universe, \simeq is defined as $a \simeq b$ iff $\text{dist}(a, b) \leq \tau_{\text{dist}}$ where $\tau_{\text{dist}} > 0$ is a fixed threshold. Applying the general principle (1) leads to:

if $\mathbf{x}^1 \simeq \mathbf{x}^2$ and $\mathbf{x}^j \rightsquigarrow \mathbf{y}^j$ for $j \in \{1, 2\}$
 then it is plausible that $\mathbf{y}^1 \simeq \mathbf{y}^2$

i.e., similar problems have similar solutions, a principle often emphasized in CBR (see e.g., [9]).

```

function approximation(CB : a set of cases, xtgt : a problem)
begin
    Candidates ← {(x, y) ∈ CB | x ≃ xtgt}
    potentialSols ← multiset of the y such that (x, y) ∈ Candidates
    if potentialSols = ∅ then
        | return failure                                ▷ The approximation method fails.
    end
    ytgt ← integrate(potentialSols)
    return ytgt
end
    
```

Fig. 1. The approximation method.

Back to our initial example, let us consider 2 individuals with close profiles (in terms of Hamming distance), e.g., $\mathbf{x}^1 = 00010010$ and $\mathbf{x}^2 = 00010011$. In that example, $H(\mathbf{x}^1, \mathbf{x}^2) = 1$ (\mathbf{x}^2 does not like dairy product), and a dish type $\mathbf{y}^2 = 011$ without sauce, at distance 1 of $\mathbf{y}^1 = 111$ (a dish type suitable for \mathbf{x}^1) will be suitable for \mathbf{x}^2 .

Figure 1 summarizes the approximation method with an algorithm.

3.4 Reusing Cases by Pairs: Interpolation

When $k = 2$, candidates are pairs, relations $\text{Rel}_{\mathcal{P}}$ and $\text{Rel}_{\mathcal{S}}$ are betweenness on \mathcal{P} and on \mathcal{S} . The general principle (1) applied here leads to:

if $\mathbf{x}^1 - \mathbf{x}^3 - \mathbf{x}^2$ and $\mathbf{x}^j \rightsquigarrow \mathbf{y}^j$ for $j \in \{1, 2, 3\}$
 then it is plausible that $\mathbf{y}^1 - \mathbf{y}^3 - \mathbf{y}^2$

Applied to CBR, a retrieved pair $\{(\mathbf{x}^1, \mathbf{y}^1), (\mathbf{x}^2, \mathbf{y}^2)\} \in \text{Candidates}$ is such that $\mathbf{x}^1 - \mathbf{x}^{\text{tgt}} - \mathbf{x}^2$ and then adaptation takes advantage of the inferred information

$y^1 - y^{tgt} - y^2$. A way to get a unique solution is to have $y^1 = y^2$, which entails $y^{tgt} = y^1$ as a candidate solution for x^{tgt} . If **Candidates** = \emptyset , the equality $y^1 = y^2$ can be relaxed in $\text{dist}(y^1, y^2) \leq \tau_{\text{dist}}$ (where **dist** is here a distance function on \mathcal{S}). In that case, uniqueness of y^{tgt} is not guaranteed anymore. By contrast, if **Candidates** is considered to be too large, it can be restricted by allowing only the pairs $\{(x^1, y^1), (x^2, y^2)\}$ such that $\text{dist}(x^1, x^2) \leq \tau_{\text{between}}$, where $\tau_{\text{between}} > 0$ is a given threshold.

```

function interpolation(CB : a set of cases, xtgt : a problem)
begin
  Candidates  $\leftarrow \left\{ \{(x^1, y^1), (x^2, y^2)\} \mid \begin{array}{l} (x^1, y^1), (x^2, y^2) \in \text{CB} \\ \text{such that } \begin{cases} y^1 = y^2 \\ x^1 - x^{tgt} - x^2 \end{cases} \end{array} \right\}$ 
  potentialSols  $\leftarrow \left\{ \{(x^1, y^1), (x^2, y^2)\} \in \text{Candidates} \text{ and } y = y^1 = y^2 \right\}$ 
  if potentialSols =  $\emptyset$  then
    | return failure ▷ The interpolation method fails.
  end
  ytgt  $\leftarrow \text{integrate}(\text{potentialSols})$ 
  return ytgt
end

```

Fig. 2. The interpolation method (for the situation in which the solutions of the retrieved cases are equal).

Back to our initial example, let $x^{tgt} = 11010010$, $(x^1, y^1) = (01110010, 001)$, and $(x^2, y^2) = (10010010, 001)$ (i.e. $H(x^1, x^2) = 3$): x^{tgt} , x^1 and x^2 differ only in the fact that the chosen individuals have/do not have gout/diabetes/allergy to nuts. $y^1 = y^2 = 001$ is the solution “dish without sauce, not based on starchy food, with fish”. Since $x^1 - x^{tgt} - x^2$, a solution is $y^{tgt} \in \text{Between}(y^1, y^2) = \{y^1\}$, i.e., $y^{tgt} = y^1 = y^2$.

Figure 2 summarizes the interpolation method in an algorithm.

3.5 Reusing Cases by Triples: Extrapolation

When $k = 3$, candidates are triples. $\text{Rel}_{\mathcal{P}}$ and $\text{Rel}_{\mathcal{S}}$ are analogical proportions on \mathcal{P} and on \mathcal{S} . The property (1) applied here leads to:

$$\begin{aligned} &\text{if } x^1 : x^2 :: x^3 : x^4 \text{ and } x^j \rightsquigarrow y^j \text{ for } j \in \{1, 2, 3, 4\} \\ &\text{then it is plausible that } y^1 : y^2 :: y^3 : y^4 \end{aligned}$$

Applied to CBR, an element of **Candidates** is a triple $((x^1, y^1), (x^2, y^2), (x^3, y^3))$ of source cases such that (i) $x^1 : x^2 :: x^3 : x^{tgt}$ and (ii) the equation $y^1 : y^2 :: y^3 : y$ has a solution. Such a solution, unique in the Boolean setting, is considered as a plausible solution of x^{tgt} .

```

function extrapolation(CB : a set of cases,  $x^{tgt}$  : a problem)
begin
    Candidates  $\leftarrow \{((x^1, y^1), (x^2, y^2), (x^3, y^3)) \in CB^3 \mid x^1 : x^2 :: x^3 : x^{tgt}\}$ 
    potentialSols  $\leftarrow$ 
        | multiset of the  $y$  such that
        |  $((x^1, y^1), (x^2, y^2), (x^3, y^3)) \in$  Candidates
        | and  $y$  is a solution of  $y^1 : y^2 :: y^3 : y$ 
    if potentialSols =  $\emptyset$  then
        | return failure ▷ The extrapolation method fails.
    end
     $y^{tgt} \leftarrow$  integrate(potentialSols)
    return  $y^{tgt}$ 
end
    
```

(a) A naive algorithm in $O(|CB|^3)$.

```

function compute_key_table(CB : a set of cases)
begin
    key_table  $\leftarrow$  empty hash table
    for  $(x^1, y^1) \in CB$  do
        for  $(x^2, y^2) \in CB, (x^2, y^2) \neq (x^1, y^1)$  do
            |  $key \leftarrow x^2 - x^1$  ▷  $key \in \{-1, 0, 1\}^m$ 
            |  $value \leftarrow ((x^1, y^1), (x^2, y^2))$ 
            | Add the pair (key, value) to key_table
            | ▷ without erasing other pairs with the same key!
        end
    end
    ▷ key_table associates to a key a set of case pairs.
end
    
```

```

function extrapolation(CB : a set of cases,  $x^{tgt}$  : a problem)
begin
    Candidates  $\leftarrow \emptyset$ 
    for  $(x, y) \in CB, \text{with } x \neq x^{tgt}$  do
        |  $key \leftarrow x^{tgt} - x$ 
        | for  $((x^1, y^1), (x^2, y^2)) \in \text{access}(key, \text{key\_table}), \text{with } x^2 \neq x^{tgt}$  do
            | ▷ By construction of key_table,  $x^1 : x^2 :: x : x^{tgt}$  holds.
            | Candidates  $\leftarrow$  Candidates  $\cup \{((x^1, y^1), (x^2, y^2), (x, y))\}$ 
        end
    end
    potentialSols  $\leftarrow$ 
        | multiset of the  $y$  such that
        |  $((x^1, y^1), (x^2, y^2), (x^3, y^3)) \in$  Candidates
        | and  $y$  is a solution of  $y^1 : y^2 :: y^3 : y$ 
    if potentialSols =  $\emptyset$  then
        | return failure ▷ The extrapolation method fails.
    end
     $y^{tgt} \leftarrow$  integrate(potentialSols)
    return  $y^{tgt}$ 
end
    
```

(b) A more efficient algorithm (with an offline function *compute_key_table* in $O(|CB|^2)$ and an online function *extrapolation* in $O(|CB|^2)$).

Fig. 3. The extrapolation method (in 2 versions: a simple one (a) and a more efficient one (b)).

Back to our example where we are searching for a suitable dish type y^{tgt} for an individual x^{tgt} :

$x^1 = 00001011$	$y^1 = 010$
$x^2 = 00100010$	$y^2 = 110$
$x^3 = 10001011$	$y^3 = 001$
$x^{tgt} = 10100010$	$y^{tgt} = ???$

The relation $x^1:x^2::x^3:x^{tgt}$ holds, and the equation $y^1:y^2::y^3:y$ has a unique solution $y^{tgt} = 101$. Our principle tells us that $y^{tgt} = 101$ is a suitable option for x^{tgt} . If we consider the intended meaning of the parameters, x^{tgt} has gout, no diabetes, is allergic to nut, is not refractory to meat, has no need for calcium nor iron supplement, likes vegs and dairy. The dish type $y^{tgt} = 101$ describing fish with sauce and no starchy food is suitable for this type of individual.

Figure 3(a) summarizes the extrapolation method in an algorithm and Fig. 3(b) presents a more efficient method described further.

3.6 Combining These Approaches

Now, we have three methods, approximation, interpolation and extrapolation, to solve a problem. These methods are plausible and incomplete: for a target problem x^{tgt} , each of them may fail either by providing an incorrect solution or by not providing any solution at all. A complete discussion of the options for combining these methods is out of the scope of this paper and constitutes a future work. However, this section discusses some ideas about the design of a good combination method.

```

function combination(CB : a set of cases,  $x^{tgt}$  : a problem)
begin
    ( $m_1, m_2, m_3$ ) ← List of the three methods approximation,
                       interpolation and extrapolation,
                       ordered by decreasing preference
     $y^{tgt} \leftarrow m_1(CB, x^{tgt})$ 
    if  $y^{tgt} \neq failure$  then
        | return  $y^{tgt}$ 
    end
     $y^{tgt} \leftarrow m_2(CB, x^{tgt})$ 
    if  $y^{tgt} \neq failure$  then
        | return  $y^{tgt}$ 
    end
     $y^{tgt} \leftarrow m_3(CB, x^{tgt})$ 
    return  $y^{tgt}$ 
end

```

Fig. 4. The combination method based on a preference relation on the set of methods {approximation, interpolation, extrapolation}.

A simple way to combine these methods is to use a preference relation between them: if the preferred method provides a solution, this is the solution returned, else the second preferred method is tried, and so on (this simple combination method is summarized in Fig. 4). This makes sense since a method

may provide results unfrequently but with high plausibility and should be tried before a method providing frequent results with lower plausibility.

Now, given the three methods, how can such a preference relation be chosen? For this purpose, an analysis of the case base may help. In particular, for the extrapolation method, it has been shown that the functions \mathbf{f} such that $\mathbf{x}^1:\mathbf{x}^2::\mathbf{x}^3:\mathbf{x}^4$ entails $\mathbf{f}(\mathbf{x}^1):\mathbf{f}(\mathbf{x}^2)::\mathbf{f}(\mathbf{x}^3):\mathbf{f}(\mathbf{x}^4)$ for each $(\mathbf{x}^1, \mathbf{x}^2, \mathbf{x}^3, \mathbf{x}^4) \in \mathcal{P}^4$ are the affine functions [6].² Thus, if \rightsquigarrow is functional ($\rightsquigarrow = \mathbf{f}$) and \mathbf{f} is affine, then the extrapolation method never gives an incorrect solution. If \mathbf{f} is closed to an affine function, then this method gives good results (and should be highly ranked in the preference relation). A measure of closeness to affinity is the BLR test [3] that can be run on the case base.

Another approach to method combination consists in having a preference between methods depending on the target problem. For this purpose, an idea is to associate to the output of each method a score that is relative to the confidence of the method wrt its output: the preferred method is the one with the higher score. For example, the approximation method would have a high confidence if many source cases support the returned solution and few ones go against it. A similar principle can be applied for interpolation and extrapolation.

4 Evaluation

The objective of the evaluation is to study the behaviour of the three approaches, on various types of Boolean functions, in order to determinate which approach is the best and in which circumstances. First experimental results are presented and a combination method based on the preference relation is proposed according to these results.

4.1 Experiment Setting

In the experiment, $\mathcal{P} = \mathbb{B}^8$ and $\mathcal{S} = \mathbb{B}^3$, as in the running example. \rightsquigarrow is assumed to be functional: $\rightsquigarrow = \mathbf{f}$.

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

CNF \mathbf{f} is generated in a conjunctive normal form, i.e., $\mathbf{f}(\mathbf{x})$ is a conjunction of n_{conj} disjunctions of literals, for example $\mathbf{f}(\mathbf{x}) = (\mathbf{x}_1 \vee \neg \mathbf{x}_7) \wedge (\neg \mathbf{x}_3 \vee \mathbf{x}_7 \vee \mathbf{x}_8) \wedge \mathbf{x}_4$. The value of n_{conj} is randomly chosen uniformly in $\{3, 4, 5\}$. Each disjunction is generated on the basis of two parameters, $p^+ > 0$ and $p^- > 0$, with $p^+ + p^- < 1$: each variable \mathbf{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.

² An affine function $\mathbf{f} : \mathbb{B}^m \rightarrow \mathbb{B}$ has the form $\mathbf{f}(\mathbf{x}) = \mathbf{x}_{i_1} \oplus \dots \oplus \mathbf{x}_{i_q} \oplus c$ where $\{i_1, \dots, i_q\}$ is a subset of $[1, m]$ and $c \in \{0, 1\}$. An affine function $\mathbf{f} : \mathbb{B}^m \rightarrow \mathbb{B}^n$ is of the form $\mathbf{f}(\mathbf{x}) = (\mathbf{f}_1(\mathbf{x}), \dots, \mathbf{f}_n(\mathbf{x}))$ where $\mathbf{f}_j : \mathbb{B}^m \rightarrow \mathbb{B}$ is an affine function.

DNF \mathbf{f} is generated in a disjunctive normal form, i.e., it has the same form as for CNF except that the connectors \wedge and \vee are exchanged. The parameters n_{disj} , p^+ and p^- are set in the same way.

Pol is the same as DNF, except that the disjunctions (\vee) are replaced with exclusive or's (\oplus), thus giving a polynomial normal form. The only different parameter is $p^- = 0$ (only positive literals occur in the polynomial normal form).

The case base CB is generated randomly, with the values for its size: $|\text{CB}| \in \{16, 32, 64, 96, 128\}$, i.e. $|\text{CB}|$ is between $\frac{1}{16}$ and $\frac{1}{2}$ of $|\mathcal{P}| = 2^8 = 256$. Each source case (\mathbf{x}, \mathbf{y}) is generated as follows: \mathbf{x} is randomly chosen in \mathcal{P} with a uniform distribution and $\mathbf{y} = \mathbf{f}(\mathbf{x})$.

Each method may lead to several solutions. Let Y be the multiset of these solutions. The function `integrate` introduced at the end of Sect. 3.2 aims at associating to Y a unique element $\mathbf{y} \in \mathcal{S}$. Here, `integrate` consists in making a vote on each component: $y_i = \text{argmax}_{y_i \in Y_i} \text{multiplicity}(y_i, Y_i)$, where Y_i is the multiset of the y_i for $\mathbf{y} \in Y$.

Let `ntp` be the number of target problems posed to the system, `na` be the number of (correct or incorrect) answers (`ntp - na` is the number of target problems for which the system fails to propose a solution), and `sscapa` be the sum of the similarities between the correct answer (according to the generated function \mathbf{f}) and the predicted answer, where the similarity between two solutions \mathbf{y}^1 and \mathbf{y}^2 is computed by $1 - H(\mathbf{y}^1, \mathbf{y}^2)/n$ (with $n = 3$ since $\mathcal{S} = \mathbb{B}^n = \mathbb{B}^3$). For each method, the following scores are computed:

The precision `prec` is the average of the ratios $\frac{\text{sscapa}}{\text{na}}$.

The correct answer rate `car` is the average of the ratios $\frac{\text{sscapa}}{\text{ntp}}$.

The average is computed on 1 million problem solving for each function generator, requiring the generation of 1060 \mathbf{f} for each of them. The average computing time of a CBR session (retrieval and adaptation for solving one problem) is about 0.8 ms (for $k = 1$), 19 ms (for $k = 2$) and 2 ms (for $k = 3$) on an current standard laptop.

The parameters for each method has been chosen as follows, after preliminary tests based on precision:

approximation: $\tau_{\text{dist}} = 1$ on \mathcal{P} ;

interpolation: $\tau_{\text{dist}} = 0$ on \mathcal{S} (i.e., $\mathbf{y}^1 = \mathbf{y}^2$), $\tau_{\text{between}} = 2$;

extrapolation: All the triples in analogy with \mathbf{x}^{tgt} are considered.

combination: The chosen preference method is: interpolation preferred to approximation preferred to extrapolation for CNF and DNF, and extrapolation preferred to interpolation preferred to approximation for Pol.

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

A Note About Implementation. A naive algorithm for implementing each $k \in \{1, 2, 3\}$ method is in $O(|CB|^k)$ (cf. Figs. 1, 2 and 3(a)). However, this complexity can be reduced for interpolation to search **Candidates** by iterating only on pairs of cases $((x^1, y^1), (x^2, y^2)) \in CB^2$ such that $H(x^1, x^2) \leq \tau_{\text{dist}}$. For extrapolation, the complexity can be separated in an offline and on online part (cf. Fig. 3(b)). The offline part is in $O(|CB|^2)$ and generates a hashtable structure over keys representing the differences between x^1 and x^2 (considered as vectors in \mathbb{R}^m). For example, $\text{key}(01001, 11010) = (-1, 0, 0, -1, 0)$. The online part is in $O(|CB|^2)$ in the worst case (and frequently closer to $O(|CB|)$)³ and searches all the $(x, y) \in CB$ such that $\text{key}(x, x^{\text{tgt}}) = \text{key}(x^1, x^2)$, which is equivalent to $x^1 : x^2 :: x : x^{\text{tgt}}$.

4.2 Results

Figure 5 pictures the detailed results given in Table 1.

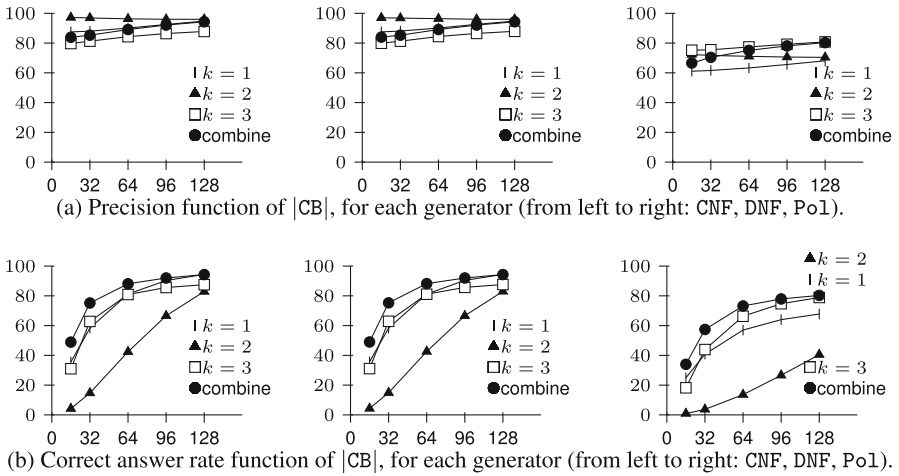


Fig. 5. Precision and correct answer rate function of the case base size for the three generators, for each method ($k = 1$, $k = 2$, $k = 3$ and “combine”, i.e. combination of the three other methods).

The result precisions shows that, for CNF and DNF, interpolation gives better results than approximation and extrapolation, and also that approximation gives better results than extrapolation. However, when examining the results wrt the correct answer rate, interpolation has a low performance, especially when the

³ The number of iterations in the for loop is $|CB| - 1$. In each iteration, the set $\text{access}(\text{key}, \text{key_table})$ contains at most $(|CB| - 1)$ elements, though in practice, this set contains in general a much smaller number of elements. So, the number of $O(1)$ operations of this online procedure in the worst case is not more than $(|CB| - 1) \times (|CB| - 1)$, hence a complexity in $O(|CB|^2)$.

Table 1. prec and car for $k \in \{1, 2, 3\}$ and combine (c.) for the different generators.

	$k =$	$ \text{CB} = 16$				$ \text{CB} = 32$				$ \text{CB} = 64$				$ \text{CB} = 96$				$ \text{CB} = 128$			
		1	2	3	c.	1	2	3	c.	1	2	3	c.	1	2	3	c.	1	2	3	c.
CNF	prec	.87	.97	.80	.84	.88	.97	.81	.85	.90	.96	.84	.89	.93	.96	.86	.92	.95	.96	.88	.94
	car	.36	.04	.31	.49	.58	.15	.63	.75	.81	.42	.81	.88	.90	.67	.86	.92	.94	.83	.88	.94
DNF	prec	.87	.97	.80	.84	.88	.97	.81	.85	.90	.96	.84	.89	.93	.96	.87	.92	.95	.96	.88	.94
	car	.36	.04	.31	.49	.58	.15	.63	.75	.81	.43	.81	.88	.90	.67	.86	.92	.94	.83	.88	.94
POL	prec	.61	.72	.75	.67	.62	.72	.75	.70	.63	.71	.77	.75	.66	.71	.79	.78	.68	.70	.81	.80
	car	.25	.01	.18	.34	.41	.04	.44	.57	.57	.14	.66	.73	.64	.27	.75	.78	.68	.40	.79	.80

case base is small ($|\text{CB}| \in \{16, 32\}$), due to the difficulty to find two candidates for the betweenness relations.

The results are different for `Pol`, for which, extrapolation provides better results. This result can be explained by the functions that have been generated which are close to affine functions, affine functions for which extrapolation always returns a correct answer [6]. For `DNF`, the second best approach wrt precision is approximation. One more time, the weakness of the two best methods (extrapolation and interpolation) is their low correct answer rates, and it must be noted that, in all situations, approximation provides a better `car`, especially when the case base is small.

Finally, combining the three approaches by preference ordering (interpolation, then approximation and then extrapolation for `DNF` and `CNF`, and extrapolation, then approximation and then interpolation for `Pol`) improves the results provided using only approximation, as expected.

The elaboration of a better combination method constitutes a future work. However, some elements relative to this issue are discussed here. A simple yet promising approach would be to estimate the average precision—and hence, the preference relation—on the basis of the case base, using a leave-one out approach: for each case $(\mathbf{x}, \mathbf{y}) \in \text{CB}$, the three methods are run with the target problem $\mathbf{x}^{\text{tgt}} = \mathbf{x}$ and the case base $\text{CB} \setminus \{(\mathbf{x}, \mathbf{y})\}$, and the results given by the three methods are compared with the known solution $\mathbf{y}^{\text{tgt}} = \mathbf{y}$, which enables us to compute the average precision for each method.

Another issue to be studied carefully is whether it is worth searching for a good combination method. Indeed, if all the methods are strongly correlated, their combination would not give much improvement. For this purpose, a preliminary test has been carried out that computes the covariance of the events “incorrect answer for method k ” and “incorrect answer for method ℓ ”, with $k, \ell \in \{1, 2, 3\}$ and $k \neq \ell$: $\text{cov}_{k\ell} = P_{k\ell} - P_k P_\ell$ where $P_{k\ell}$ estimates the probability that the methods k and ℓ provide an incorrect answer, given that they both provide an answer, P_k estimates the probability that the method k provides incorrect answers, given that it provides an answer, and P_ℓ is defined similarly. The results on the generated data are: $\text{cov}_{12} = \text{cov}_{21} \simeq 0.90$, $\text{cov}_{23} = \text{cov}_{32} \simeq 0.41$, and $\text{cov}_{13} = \text{cov}_{31} \simeq 0.27$. This can be interpreted as follows. The high correlation between approximation and interpolation is mainly due to the tight bounds

used in the experiments for interpolation. By contrast, extrapolation is *much less correlated* with the other methods. This suggests the complementarity of extrapolation, hence the worthiness of studying with more depth the combination issue.

5 Discussion and Related Work

An important issue raised in this paper is that the reuse of multiple cases ($k \geq 2$) does not necessarily rely on similarity (or not only): in the general situation, it relies on two $(k + 1)$ -ary relations on \mathcal{P} and \mathcal{S} , that may be non reducible to binary similarity relations. These relations structure the problem and solution spaces. Similar issues are addressed in other works of the CBR literature. For example, (simple or multiple) reuse may have profit of adaptation hierarchies on \mathcal{P} and \mathcal{S} [1, 5, 23]. Another example of relation used for multiple case retrieval and adaptation is the use of *diversity*: the retrieved cases should be diverse in order to better contribute to the solving of the target problem [13]. The main originality of this work is the use for CBR of betweenness and analogical proportion, two domain-independent relations that can be implemented in many formalisms (though it has been considered only in the Boolean setting here), e.g. on strings [12], and that enables to apply to CBR the principles of interpolation and extrapolation, in addition to the already frequently used principle of approximation.

The extrapolation approach based on analogical reasoning can be connected to the work of adaptation knowledge learning (AKL) in CBR, as explained hereafter. Most approaches to AKL consist in learning adaptation rules using as training set a set of source case pairs $((x^i, y^i), (x^j, y^j))$ [7, 8, 10]. A similar idea consists in considering one of such pairs as an *adaptation case* [11, 15]. With (x^1, y^1) a retrieved case, it can be adapted to solve x^{tgt} if the difference from x^1 to x^{tgt} equals the difference from x^i to x^j , which can be formalized using analogical proportion by $x^i:x^j::x^1:x^{\text{tgt}}$. Then, the adaptation consists in applying on y^1 the difference from y^i to y^j , which amounts to solving the equation $y^i:y^j::y^1:y$. Therefore, one contribution of this work is the formalization of *case-based adaptation* in terms of analogical proportions on the problem and solution spaces.

The idea of applying analogical inference based on analogical proportions in CBR has been first advocated and outlined recently [17], based on the fact that such an approach has given good quality results in classification [4] in machine learning on real datasets. Interestingly enough, analogical proportions can be always found in such datasets, and the results obtained can be favorably compared to those yielded by nearest neighbor methods. The ideas of interpolative and extrapolative reasoning can be found in the setting of conceptual spaces furnished with qualitative knowledge [20] with an illustration in [19]. Similar ideas have been applied to the completion of bases made of if-then rules in relation with the idea of analogical proportion [22] and to the interpolation between default rules [21]. However, none of these papers mentions a CBR perspective.

6 Conclusion

Classical CBR exploits each known case individually, on the basis of its similarity with the target problem. In this paper, we have proposed to extend this paradigm by taking advantage of betweenness and analogical proportion relations for linking the current situation to pairs and triples of known cases. By doing that, we are no longer just proposing the known solution of a case problem as an approximate solution to the current problem, but we are also taking advantage of interpolation and extrapolation ideas that are respectively embedded in betweenness and analogical proportion relations.

We have also provided a first experimental study that shows the precision and the correct answer rate of the approximation, interpolation, and extrapolation approaches on various types of functions. Each approach has its own merits. Approximation is simple and is often very efficient. Interpolation provides the most precise results, but may fail to give an answer. Extrapolation is superior to the two other methods when the underlying function is affine or close to be, i.e., exhibits a form of simplicity. Moreover, the results of extrapolation are not much correlated with the results of the two other methods. Experiments also show that combining the approaches may be beneficial. Clearly, one should investigate less straightforward ways for combining the approaches. More experiments would be also of interest for varying more the parameters and for dealing with non functional dependencies between problems and solutions. Experiments have been made on a variety of artificial datasets, showing that the present implementation of extrapolation is especially of interest for datasets close to obey to an affine Boolean function, studying if another form of extrapolation would do better, or if such datasets are often encountered in practice, is a matter of further work.

There are still two other perspectives worth of consideration. First, it would be of interest to see how incorporating domain/retrieval/adaptation knowledge in the process. Lastly, relations $\text{Rel}_{\mathcal{P}}$ and $\text{Rel}_{\mathcal{S}}$ have been defined here on \mathbb{B}^p , but allowing for gradual relations and for nominal and numerical features in the description of problems and solutions would be an important improvement. Such a latter development should be feasible since extensions of analogical proportions have been satisfactorily experienced in classification for handling nominal and numerical features [4], and the idea of betweenness seems as well to be susceptible of natural extensions to these situations.

References

1. Bergmann, R., Wilke, W.: Building and refining abstract planning cases by change of representation language. *J. Artif. Intell. Res.* **3**, 53–118 (1995)
2. Billingsley, R., Prade, H., Richard, G., Williams, M.-A.: Towards analogy-based decision - a proposal. In: Christiansen, H., Jaudoin, H., Chountas, P., Andreasen, T., Legind Larsen, H. (eds.) *FQAS 2017*. LNCS (LNAI), vol. 10333, pp. 28–35. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-59692-1_3
3. Blum, M., Luby, M., Rubinfeld, R.: Self-testing/correcting with applications to numerical problems. *J. Comput. Syst. Sci.* **47**(3), 549–595 (1993)

4. Bounhas, M., Prade, H., Richard, G.: Analogy-based classifiers for nominal or numerical data. *Int. J. Approx. Reasoning* **91**, 36–55 (2017)
5. Branting, L.K., Aha, D.W.: Stratified case-based reasoning: reusing hierarchical problem solving episodes. In: *Proceedings of the 14th International Joint Conference on Artificial Intelligence (IJCAI 1995)*, vol. 1, pp. 384–390, August 1995
6. Couceiro, M., Hug, N., Prade, H., Richard, G.: Analogy-preserving functions: a way to extend Boolean samples. In: *Proceedings of the 26th International Joint Conference on Artificial Intelligence (IJCAI 2017)*, pp. 1575–1581. Morgan Kaufmann, Inc. (2017)
7. Craw, S., Wiratunga, N., Rowe, R.C.: Learning adaptation knowledge to improve case-based reasoning. *Artif. Intell.* **170**(16–17), 1175–1192 (2006)
8. 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.) *Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI 2007)*, pp. 750–755. Morgan Kaufmann, Inc. (2007)
9. Dubois, D., Hüllermeier, E., Prade, H.: Flexible control of case-based prediction in the framework of possibility theory. In: Blanzieri, E., Portinale, L. (eds.) *EWCBR 2000*. LNCS, vol. 1898, pp. 61–73. Springer, Heidelberg (2000). https://doi.org/10.1007/3-540-44527-7_7
10. Hanney, K., Keane, M.T.: Learning adaptation rules from a case-base. In: Smith, I., Faltings, B. (eds.) *EWCBR 1996*. LNCS, vol. 1168, pp. 179–192. Springer, Heidelberg (1996). <https://doi.org/10.1007/BFb0020610>
11. Jarmulak, J., Craw, S., Rowe, R.: Using case-base data to learn adaptation knowledge for design. In: *Proceedings of the 17th International Joint Conference on Artificial Intelligence (IJCAI 2001)*, pp. 1011–1016. Morgan Kaufmann, Inc. (2001)
12. Lepage, Y., Denoual, É.: Purest ever example-based machine translation: detailed presentation and assessment. *Mach. Transl.* **19**, 251–282 (2005)
13. McSherry, D.: Diversity-conscious retrieval. In: Craw, S., Preece, A. (eds.) *ECCBR 2002*. LNCS (LNAI), vol. 2416, pp. 219–233. Springer, Heidelberg (2002). https://doi.org/10.1007/3-540-46119-1_17
14. Miclet, L., Prade, H.: Handling analogical proportions in classical logic and fuzzy logics settings. In: Sossai, C., Chemello, G. (eds.) *ECSQARU 2009*. LNCS (LNAI), vol. 5590, pp. 638–650. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-02906-6_55
15. Minor, M., Bergmann, R., Görg, S.: Case-based adaptation of workflows. *Inf. Syst.* **40**, 142–152 (2014)
16. Prade, H., Richard, G.: From analogical proportion to logical proportions. *Logica Universalis* **7**(4), 441–505 (2013)
17. Prade, H., Richard, G.: Analogical proportions and analogical reasoning - an introduction. In: Aha, D.W., Lieber, J. (eds.) *ICCBR 2017*. LNCS (LNAI), vol. 10339, pp. 16–32. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-61030-6_2
18. Richter, M.M., Weber, R.O.: *Case-Based Reasoning: A Textbook*. Springer, Heidelberg (2013). <https://doi.org/10.1007/978-3-642-40167-1>
19. Schockaert, S., Prade, H.: Interpolation and extrapolation in conceptual spaces: a case study in the music domain. In: Rudolph, S., Gutierrez, C. (eds.) *RR 2011*. LNCS, vol. 6902, pp. 217–231. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-23580-1_16
20. Schockaert, S., Prade, H.: Interpolative and extrapolative reasoning in propositional theories using qualitative knowledge about conceptual spaces. *Artif. Intell.* **202**, 86–131 (2013)

21. Schockaert, S., Prade, H.: Interpolative reasoning with default rules. In: Rossi, F. (ed.) Proceedings of the 23rd International Joint Conference on Artificial Intelligence (IJCAI 2013), Beijing, 3–9 August 2013, pp. 1090–1096 (2013)
22. Schockaert, S., Prade, H.: Completing symbolic rule bases using betweenness and analogical proportion. In: Prade, H., Richard, G. (eds.) Computational Approaches to Analogical Reasoning: Current Trends. SCI, vol. 548, pp. 195–215. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-642-54516-0_8
23. Smyth, B.: Case-based design. Ph.D. thesis, Trinity College, University of Dublin (1996)