Conservative Adaptation in Metric Spaces

Julien Cojan, Jean Lieber

Conservative Adaptation in Metric Spaces

Julien Cojan and Jean Lieber

Orpailleur, LORIA, CNRS, INRIA, Nancy Universities,
BP 239, 54 506 Vandœuvre-lès-Nancy
{Julien.Cojan, Jean.Lieber}@loria.fr,

Abstract. Conservative adaptation consists in a minimal change on a source case to be consistent with the target case, given the domain knowledge. It has been formalised in a previous work thanks to the AGM theory of belief revision applied to propositional logic. However, this formalism is rarely used in case-based reasoning systems. In this paper, conservative adaptation is extended to a more general representation framework, that includes also attribute-value formalisms. In this framework, a case is a class of case instances, which are elements of a metric space. Conservative adaptation is formalised in this framework and is extended to α-conservative adaptation, that relaxes the conservativeness. These approaches to adaptation in a metric space transform adaptation problems to well-formulated optimization problems. A running example in the cooking domain is used to illustrate the notions that are introduced.

Keywords: adaptation, belief revision, conservative adaptation, case representation, metric spaces.

1 Introduction

Adaptation is an issue of CBR (case-based reasoning [1]) that still deserves a big amount of research. Conservative adaptation is an approach to adaptation that consists in a minimal change on a source case to be consistent with the target case, given the domain knowledge. It has been formalised in a previous work thanks to the AGM theory of belief revision applied to propositionnal logic (PL).

However, PL is rarely used in CBR systems. In this paper, conservative adaptation is extended to the general representation framework of "metric space formalisms", that includes PL and also attribute-value formalisms (which are widely used in CBR [2]).

Section 2 is a reminder about adaptation in CBR and introduces the running example in the cooking domain used throughout the paper. Section 3 presents the metric space formalisms. Section 4 formalises conservative adaptation in these formalisms. This approach to adaptation can be extended by relaxing the conservativeness: this is the α-conservative adaptation, presented and studied in section 5. Finally, section 6 concludes the paper and draws some future work.

2 Adaptation in Case-Based Reasoning

2.1 Principles of CBR and of Adaptation in CBR

Case-Based Reasoning (CBR) is a reasoning paradigm using cases, where a case encodes a particular piece of experience. The aim of a CBR system is to complete a *target case* `Target` for which some information is missing. To do so, a case base is assumed to be available. A case base is a finite set of cases, called the *source cases*. The application of CBR on a target case `Target` consists in two main steps:

- *Retrieval* of a source case `Source` from the case base, similar to `Target`.
- *Adaptation*, that consists in completing `Target` into `Target-completed` from `Source`.

`Target-completed` might still have to be completed. If so, it is used as a new target case for a new CBR session. Therefore several source cases may be involved in the final completion of `Target`.

Much work has been done on retrieval, but adaptation still needs investigation work. In most CBR implementations, adaptation is either basic or domain specific. The purpose of this paper is to present a general method for adaptation based on the principle of minimal change.

2.2 An Adaptation Example

Cooking provides many case-based reasoning examples, a recipe book is indeed a kind of case base. For simplicity, the focus is put on ingredients rather than on preparation, a problem consists in requirements on ingredients and portions, a solution is a recipe satisfying these requirements, i.e. an ingredient list and a text of instructions.

Léon wants to cook a fruit pie for six persons but he only has pears at disposal (and thus, no apple). He finds an apple pie recipe for four servings but no pear pie recipe. This can be formulated as a CBR adaptation problem:

`Target` = a requested recipe for a 6 portion pie with pears and no other fruit.
`Source` = a 4 portion apple pie recipe with 2 apples, 40 grams of sugar, and 120 grams of pastry as ingredients.

It is quite natural for Léon to think of the following adaptation which can be split into two steps: a substitution of apples by pears and an increase by half of the amount of each ingredient. These two adaptation steps involve different pieces of knowledge. The first one involves similarity between apples and pears. The second one is the following principle: the amount of ingredients is proportional to the number of portions.

In addition to this adaptation knowledge, some more knowledge is needed. The amount of apples and pears is expressed in number of fruits, however the relevant quantity here is their mass, thus the average mass per apple and pear is needed, say 120 grams for an apple and 100 grams for a pear. Moreover, to preserve the pie sweet, the amount of added sugar should be adjusted so as to compensate the different sweet amount contained in apples and pears —say 13 grams per pear and 14 grams per apple.

Knowing all this, from the source recipe Léon should infer he needs the following ingredients for his fruit pie:

- 3 or 4 pears as these values make the variation of fruit mass per person $\left|\frac{120\times 2}{4} - \frac{100\times x}{6}\right|$ minimal (for x: a natural integer).
- 50 grams of sugar (resp., 63 grams) if 4 pears (resp., 3 pears) were used , as it makes the variation of sweet mass per person $\left|\frac{40+2\times 14}{4} - \frac{x+4\times 13}{6}\right|$ (resp., $\left|\frac{40+2\times 14}{4} - \frac{x+3\times 13}{6}\right|$) minimal (for x: a real number).
- 180 grams of pastry as it makes the variation of pastry mass per person $\left|\frac{120}{4} - \frac{x}{6}\right|$ minimal (for x: a real number).

3 Metric Space formalism for case and domain knowledge representation

3.1 Background

Definition 1. *A* similarity measure *on a set* $\mathcal{U}$ *is a mapping* S *from* $\mathcal{U}\times\mathcal{U}$ *to* $[0,1]$ *such that:*

$$\textit{for all } x,y\in\mathcal{U} \qquad S(x,y)=1 \quad \textit{iff} \quad x=y$$

The notation S *is extended on* $y\in\mathcal{U}$ *and* $A,B\subseteq\mathcal{U}$*:*

$$S(A,y)=\sup_{x\in A} S(x,y) \qquad S(A,B)=\sup_{x\in A,y\in B} S(x,y) \tag{1}$$

with the following convention: $S(\emptyset,y)=S(A,\emptyset)=S(\emptyset,B)=0$.

A similarity measure S can be defined from a mapping $d:\mathcal{U}\times\mathcal{U}\to\mathbb{R}_+$ satisfying the separation postulate of metrics —for all $x,y\in\mathcal{U}$ $d(x,y)=0$ iff $x=y$— by the relation:[1]

$$\text{for all } x,y\in\mathcal{U} \qquad S(x,y)=e^{-d(x,y)} \tag{2}$$

[1] Any mapping $f:\mathbb{R}_+\to[0,1]$ continuous, strictly decreasing and such that $f(0)=1$ and $\lim_{x\to+\infty} f(x)=0$ can be used instead of $x\mapsto e^{-x}$. For instance, $f(x)=\frac{1}{1+x}$ is often chosen in CBR. This choice was made for simplifications (see further). And, as the values do not have any relevance but through comparisons by $\leq$, this choice has no other effect than simplicity.

3.2 Case representation

Cases are assumed to be represented by concepts of a *concept language* $\mathcal{L}_C$ where a concept C is interpreted by a subset $\texttt{Ext}(C)$ of a set $\mathcal{U}$ (the "universe of discourse"). $\mathcal{L}_C$ is supposed to be closed under negation, conjunction and the unary operators G^σ for $\sigma \in [0, 1]$:

$$\begin{aligned} \text{if } C, D \in \mathcal{L}_C \quad &\text{then} \quad \neg C,\ C \wedge D,\ G^\sigma(C) \in \mathcal{L}_C \\ C \vee D \quad &\text{is defined by} \quad \neg(\neg C \wedge \neg D) \end{aligned}$$

Moreover $\mathcal{L}_C$ is assumed to contain $\top$ and $\bot$. The semantics is given by the mapping $\texttt{Ext}$ from $\mathcal{L}_C$ to $2^{\mathcal{U}}$ (the subsets of $\mathcal{U}$) satisfying:

$$\begin{aligned} &\texttt{Ext}(\top) = \mathcal{U} && \texttt{Ext}(C \wedge D) = \texttt{Ext}(C) \cap \texttt{Ext}(D) \\ &\texttt{Ext}(\bot) = \emptyset && \texttt{Ext}(C \vee D) = \texttt{Ext}(C) \cup \texttt{Ext}(D) \\ &\texttt{Ext}(\neg C) = \mathcal{U} \setminus \texttt{Ext}(C) && \texttt{Ext}(G^\sigma(C)) = \{x \in \mathcal{U} \mid S(\texttt{Ext}(C), x) \geq \sigma\} \end{aligned}$$

Definition 2. *A* model *of $C \in \mathcal{L}_C$ is, by definition, an element of $\texttt{Ext}(C)$. The consequence $\models$ and equivalence $\equiv$ relations on $\mathcal{L}_C$ are defined by:*

$$\begin{aligned} C \models D \quad &\textit{if} \quad \texttt{Ext}(C) \subseteq \texttt{Ext}(D) \\ C \equiv D \quad &\textit{if} \quad \texttt{Ext}(C) = \texttt{Ext}(D) \end{aligned}$$

A concept $C \in \mathcal{L}_C$ is satisfiable *if $\texttt{Ext}(C) \neq \emptyset$, i.e. $C \not\models \bot$. For $A \in 2^{\mathcal{L}_C}$ and $C \in \mathcal{L}_C$, $A \models C$ means that if $x \in \mathcal{U}$ is a model of each $D \in A$, then it is a model of C. If $C, C_1, C_2 \in \mathcal{L}_C$, $C_1 \equiv_C C_2$ if $C \wedge C_1 \equiv C \wedge C_2$: $\equiv_C$ is the equivalence modulo C.*

In this paper, $\models$ (and thus, $\equiv$) are supposed to be computable: there is a program taking as inputs two concepts C and D and giving in finite time a boolean value that is equal to `True` iff $C \models D$.

The following notations are introduced for the sake of simplicity:

$$S(C, x) = S(\texttt{Ext}(C), x) \qquad S(C, D) = S(\texttt{Ext}(C), \texttt{Ext}(D)) \tag{3}$$

$$\mathcal{E} = \{\texttt{Ext}(C) \mid C \in \mathcal{L}_C\} \quad (\text{Thus, } \mathcal{E} \subseteq 2^{\mathcal{U}}) \tag{4}$$

3.3 Domain knowledge representation

Domain knowledge is about properties that can be inferred on cases. By contrast with adaptation knowledge that is about comparisons between cases, it is static, i.e. it applies to cases by their own. In the cooking example, the amount of fruit is inferred from the amount of apples and pears in the recipe. From the interpretation point of view, the domain knowledge comes to the restriction of the extension space, some interpretations are not *licit*. So, like cases, it can be represented by a concept $\texttt{DK}$ provided that the language $\mathcal{L}_C$ is expressive enough, which is assumed. Thus, $\texttt{DK} \in \mathcal{L}_C$.

3.4 Attribute-value Representation

Many CBR systems rely on attribute-values representation of cases. The formalism presented below is a general attribute-value representation formalism that specialises the (very) general framework presented above. In this formalism $\mathcal{U}$ is assumed to be a Cartesian product:

$$\mathcal{U} = V_1 \times V_2 \times \ldots \times V_n$$

where V_i are "simple values" spaces, i.e. either $\mathbb{R}$ (the real numbers), $\mathbb{R}_+$ (the positive or null real numbers), $\mathbb{Z}$ (the integers), $\mathbb{N}$ (the natural integers), $\mathbb{B} = \{\texttt{True}, \texttt{False}\}$, or another enumerated set given in extension ("enumerated type").

For $i \in \{1, \ldots, n\}$, the *attribute* a_i is the projection along the i^{th} coordinate:

$$a_i(x_1, x_2, \ldots, x_i, \ldots, x_n) = x_i \tag{5}$$

The language $\mathcal{L}_C$ is made of expressions with boolean values on the formal parameters $a_1, a_2, \ldots, a_n$: $C = P(a_1, a_2, \ldots, a_n)$. The extension of such a concept C is:

$$\begin{aligned}\texttt{Ext}(C) &= \{x \in \mathcal{U} \mid P(a_1(x), a_2(x), \ldots, a_n(x)) = \texttt{True}\} \\ &= \{(x_1, x_2, \ldots, x_n) \in \mathcal{U} \mid P(x_1, x_2, \ldots, x_n) = \texttt{True}\}\end{aligned}$$

$\mathcal{L}_C$ is still considered as closed for negation and conjunction.

3.5 Propositional Logic as a kind of Attribute-value Representation

The set of formulas on propositional variables $p_1, \ldots, p_n$ ($n \in \mathbb{N}$) can be put under the attribute-value representation with $\mathcal{U} = \mathbb{B}^n$. Indeed, to a propositional logic formula f on $p_1, \ldots, p_n$, can be associated the mapping $P_f : \mathbb{B}^n \to \mathbb{B}$ such that, for an interpretation I of the variables $p_1, \ldots, p_n$, I is a model of f iff $P_f(I(p_1), I(p_2), \ldots, I(p_n)) = \texttt{True}$. Reciprocally, to a mapping $P : \mathbb{B}^n \to \mathbb{B}$ it can be associated a formula f unique modulo logical equivalence such that $P = P_f$.

For example, to $f = a \wedge \neg(b \vee \neg c)$ is associated $P_f : (x, y, z) \in \mathbb{B}^3 \mapsto P(x, y, z) = \texttt{and}(x, \texttt{not}(\texttt{or}(y, \texttt{not}(z))))$.

For $I \in \mathcal{U}$, $i \in \{1, \ldots, n\}$ and f a propositional formula on $p_1, \ldots, p_n$, let $a_i(I) = I(p_i)$ and $\texttt{Ext}(f) = \{x \in \mathcal{U} \mid P_f(a_1(x), a_2(x), \ldots, a_n(x)) = \texttt{True}\}$. The following equivalence identifies the obtained semantics with the propositional logic semantics: I is a model of f iff $I \in \texttt{Ext}(f)$. This justifies the use of section 3.1 formalism in section 4 to generalise conservative adaptation defined on propositional logic in [3].

3.6 Formalisation of the Cooking Example Adaptation Problem

The section 2.2 example can be formalised as follows. The following attributes are introduced:

- $a_1 = \texttt{servings}$ for the number of servings the recipe is meant to, $V_1 = \mathbb{N} \setminus \{0\}$.
- $a_2 = \texttt{sweet}$ for the total amount of sweet (in equivalent saccharose grams), $V_2 = \mathbb{R}_+$.
- $a_3 = \texttt{sugar}$ for the amount of saccharose, in grams, $V_3 = \mathbb{R}_+$.
- $a_4 = \texttt{pastry-mass}$ for the amount of pastry, in grams, $V_4 = \mathbb{R}_+$.
- $a_5 = \texttt{fruit-mass}$ for the amount of fruits, in grams, $V_5 = \mathbb{R}_+$.
- $a_6 = \texttt{apples-nb}$ for the number of apples, $V_6 = \mathbb{N}$.
- $a_7 = \texttt{pears-nb}$ for the number of pears, $V_7 = \mathbb{N}$.

The space is then $\mathcal{U} = (\mathbb{N} \setminus \{0\}) \times \mathbb{R}_+ \times \mathbb{R}_+ \times \mathbb{R}_+ \times \mathbb{R}_+ \times \mathbb{N} \times \mathbb{N}$. The attributes `sugar`, `pastry-mass`, `apples-nb`, and `pears-nb` correspond to the possible ingredients that can be used in the recipes. The values corresponding to the attributes `sweet` and `fruit-mass` are deduced from the values of the "ingredient" attributes and from the domain knowledge `DK`: the amount of fruits is the sum of apple and pear masses, similarly, the sweet is equal to the sugar plus the sweet contained in apples and pears:

$$\begin{aligned}\texttt{DK} = &(\texttt{sweet} = \texttt{sugar} + 14 \times \texttt{apples-nb} + 13 \times \texttt{pears-nb}) \\ &\wedge (\texttt{fruit-mass} = 120 \times \texttt{apples-nb} + 100 \times \texttt{pears-nb})\end{aligned}$$

The source case, an apple pie for four servings, is represented by the concept `Source` stating the number of servings and the amount of each ingredient:

$$\begin{aligned}\texttt{Source} = &(\texttt{servings} = 4) \wedge (\texttt{pastry-mass} = 120) \wedge (\texttt{sugar} = 40) \\ &\wedge (\texttt{apples-nb} = 2) \wedge (\texttt{pears-nb} = 0)\end{aligned}$$

The target case, a pie for six servings, is represented by the concept `Target` stating the number of servings, the fact that no apple is available, and the fact that some fruit is required:

$$\texttt{Target} = (\texttt{servings} = 6) \wedge (\texttt{apples-nb} = 0) \wedge (\texttt{fruit-mass} > 0)$$

4 Conservative Adaptation in Metric Space Formalisms

4.1 Belief Revision

The belief revision theory aims at establishing how to incorporate new information into previous beliefs that can be inconsistent with this new information, i.e. to define an operator $\circ$ on beliefs such that if D is the new information to be added to prior beliefs C, then the resulting beliefs should be $C \circ D$. Requirements for a revision operator have been formalised in the AGM postulates [4]. In [5], Katsuno and Mendelzon give the following postulates which are equivalent to AGM postulates —they prove the equivalence in propositional logic but their demonstration is still valid in the formalism of section 3.1:

$$\text{Basic postulates}\begin{cases}(\text{R1}) & C\circ D\vDash D\\ (\text{R2}) & \text{if}\quad C\wedge D \text{ is satisfiable}\quad\text{then}\quad C\circ D\equiv C\wedge D\\ (\text{R3}) & \text{if}\quad D \text{ is satisfiable}\quad\text{then}\quad C\circ D \text{ too}\\ (\text{R4}) & \text{if}\quad C\equiv C' \text{ and } D\equiv D'\quad\text{then}\quad C\circ D\equiv C'\circ D'\end{cases}$$

$$\text{Minimality postulates}\begin{cases}(\text{R5}) & (C\circ D)\wedge F\vDash C\circ(D\wedge F)\\ (\text{R6}) & \text{if}\quad (C\circ D)\wedge F \text{ is satisfiable}\\ & \text{then}\quad C\circ(D\wedge F)\vDash (C\circ D)\wedge F\end{cases}$$

The postulate (R1) means that the new knowledge D must be kept, (R2) means that if C and D are compatible, then both should be kept. (R3) means that $C \circ D$ must be consistent whenever D is, (R4) states the irrelevance of syntax principle. (R5) and (R6) are less intuitive, according to [5], they express the minimality of change.

These postulates are not constructive and do not prove the existence nor the unicity of such a revision operator. However, provided a similarity measure S is given on $\mathcal{U}$, a candidate $\circ^S$ for being a revision operator is defined by $C \circ^S D$ where C and D are concepts and $\Sigma = S(C, D)$:

$$C \circ^S D = G^{\Sigma}(C) \wedge D \tag{6}$$

In terms of interpretations, this means that:

$$\texttt{Ext}(C \circ^S D) = \{x \in \texttt{Ext}(D) \mid S(C, x) \geq S(C, D)\} \tag{7}$$

The models of $C \circ^S D$ are the models of D which are the most similar to C.

Proposition 1. *(i)* $\circ^S$ *satisfies postulates* (R1), (R4), (R5), *and* (R6).
(ii) The postulate $C \wedge D \vDash C \circ^S D$, *weaker than* (R2), *is satisfied by* $\circ^S$.
(iii) $\circ^S$ *satisfies* (R2) *iff for all* $A \in \mathcal{E}$ *and* $x \in \mathcal{U}$:

$$S(A, x) = 1 \;\; \textit{implies} \;\; x \in A \tag{8}$$

(iv) $\circ^S$ *satisfies* (R3) *iff for all* $A, B \in \mathcal{E}$ *with* $B \neq \emptyset$:

$$\textit{if } S(A, B) = \Sigma \;\; \textit{then there is} \;\; x \in B \;\; \textit{such that } S(A, x) = \Sigma \tag{9}$$

The proof of this proposition is given in appendix B.

4.2 Conservative Adaptation

Conservative adaptation consists in completing `Target` by a minimal change on `Source`.

In [3], conservative adaptation is defined for CBR systems where each case is assumed to be decomposable in a fixed manner in a problem part and a solution part, both expressed in propositional logic. Below, conservative adaptation is formalised in the more general framework of this paper. Given a target case Target, a source case Source, and domain knowledge DK, conservative adaptation returns Target-completed such that:

$$(\mathtt{DK} \wedge \mathtt{Source}) \circ (\mathtt{DK} \wedge \mathtt{Target}) \ \equiv_{\mathtt{DK}} \ \mathtt{Target\text{-}completed} \tag{10}$$

Therefore, conservative adaptation depends on the chosen revision operator $\circ$. Consider Katsuno and Mendelzon postulates meaning from the conservative adaptation point of view:

- (R1) means that, modulo DK, Target-completed specialises Target, and thus, conservative adaptation realises a completion.
- (R2) means that if Source is not incompatible with Target modulo DK, then it completes Target correctly and Target-completed $\equiv_{\mathtt{DK}}$ Source $\wedge$ Target.
- (R3) is a success guarantee, if Source is consistent modulo DK, then conservative adaptation returns Target-completed which is consistent with DK too.[2]
- (R4) means that conservative adaptation satisfies the irrelevance of syntax principle.
- (R5) and (R6) mean that the adaptation from Source should be minimal, it consists in a minimal change on Source to be consistent with Target.

Proposition 1 states that postulates (R2) and (R3) are only satisfied if some conditions on d are satisfied. The non satisfaction of (R2) is not really a problem, interpretations with a similarity of 1 to the original belief can arguably be included in the extension of the revision. The non satisfaction of postulate (R3) is more problematic, no solution can be found, not because Source is too different to Target —(R3) can even be contradicted with $S(\mathtt{Source}, \mathtt{Target}) = 1$— but because the similarity condition is too restrictive, the inferior boundary in the definition of S on subsets (1) may not be reached. This concern leads to the study of α-conservative adaptation in section 5.

4.3 Conservative Adaptation in the Cooking Example

In the cooking example formalisation (section 3.6) the source and target cases and the domain knowledge have been formalised. However, conservative adaptation also depends on a revision operator which is chosen here to be of the (6) kind where the similarity measure S is defined from a mapping d as in (2). d is taken under the form:

$$d(x, y) = \sum_{i=1}^{7} w_i d_i(x, y)$$

[2] Note that the condition "Source is consistent with DK" should always be true: when adding a case Source to the case base, the consistency test DK $\wedge$ Source $\not\models \bot$ should be done. Indeed, since we adhere to the irrelevance of syntax principle, a source case that is inconsistent with domain knowledge is useless.

where $w_i > 0$ are weights and $d_i : \mathcal{U} \times \mathcal{U} \mapsto \mathbb{R}_+$ are defined as follows, for $x = (x_1, \ldots, x_7)$ and $y = (y_1, \ldots, y_7)$:

$$d_1(x, y) = |y_1 - x_1|, \qquad \text{for } i \in \{2, \ldots, 7\}, \;\; d_i(x, y) = \left| \frac{y_i}{y_1} - \frac{x_i}{x_1} \right|$$

The choice of d_2 to d_7 expresses proportionality knowledge: the quantity of each product is to be considered relatively to the number of servings —2 apples for 4 servings and 3 apples for 6 servings correspond to the same amount of apples per serving.

The conservative adaptation built upon S gives a concept `Target-completed` from the source case `Source` and a target case `Target` satisfying:

$$(\texttt{DK} \wedge \texttt{Source}) \circ^S (\texttt{DK} \wedge \texttt{Target}) \equiv_{\texttt{DK}} \texttt{Target-completed}$$

According to (7), its extension is equal to:

$$\begin{aligned}
&\texttt{Ext}(\texttt{Target-completed}) \\
&\quad = \{x \in \texttt{Ext}(\texttt{DK} \wedge \texttt{Target}) \mid S(\texttt{DK} \wedge \texttt{Source}, x) \text{ is maximal}\} \\
&\quad = \{x \in \texttt{Ext}(\texttt{DK} \wedge \texttt{Target}) \mid d(\texttt{DK} \wedge \texttt{Source}, x) \text{ is minimal}\}
\end{aligned}$$

Therefore, at this point, conservative adaptation is reduced to an optimisation problem. The way this specific optimisation problem is solved is presented in appendix A. However, the choice of w_i values could not be completely justified, in particular two sets of weights are proposed for which conservative adaptation results are respectively `Target-completed` and `Target-completed'`:

$$\begin{aligned}
\texttt{Target-completed} \equiv_{\texttt{DK}} & (\texttt{servings} = 6) \wedge (\texttt{pastry-mass} = 180) \wedge (\texttt{sugar} = 50) \\
& \wedge (\texttt{apples-nb} = 0) \wedge (\texttt{pears-nb} = 4) \\
\texttt{Target-completed'} \equiv_{\texttt{DK}} & (\texttt{servings} = 6) \wedge (\texttt{pastry-mass} = 180) \wedge (\texttt{sugar} = 63) \\
& \wedge (\texttt{apples-nb} = 0) \wedge (\texttt{pears-nb} = 3)
\end{aligned}$$

In the following, the values set corresponding to `Target-completed` is chosen. However, the distance difference with $\texttt{DK} \wedge \texttt{Source}$ is small:

$$d(\texttt{DK} \wedge \texttt{Source}, \texttt{Target-completed}) = 20 + \frac{1}{6}(10 + 40 + 10 \times 3 + 10 \times 4) = 40$$

$$d(\texttt{DK} \wedge \texttt{Source}, \texttt{Target-completed'}) = 20 + \frac{1}{6}(3 + 60 + 10 \times 3 + 10 \times 3) = 40.5$$

It may be interesting to include both in the result. Indeed, the adaptation process presented in section 2.2 is exactly $\texttt{Target-completed} \vee \texttt{Target-completed'}$. This can be done thanks to α-conservative adaptation.

5 α-Conservative Adaptation: a *less* conservative adaptation

Keeping only the models of `Target` closest to those of `Source` can be too restrictive, in particular when (R3) is not satisfied, the conservative adaptation result is not satisfiable. Some flexibility in what is meant by "closest to `Source`" is needed. For instance as the similarity difference between four and five pears is small, both possibilities could be proposed to Léon letting him choose whether he would rather have more or less fruits on his pie. To do so, a flexibility is introduced in the revision operator conservative adaptation stands on, a stretchable margin is added in the extension delimitation. This has also the merit to reduce the sensitivity of the adaptation on some parameters of the similarity measure (such as the weights w_i).

5.1 α-revision

Definition 3. *Given a similarity measure S, $\alpha \in [0,1]$, and $C, D \in \mathcal{L}_C$, the α-revision of C by D is $C \circ_\alpha^S D$ defined as follows where $\Sigma = S(C,D)$:*

$$C \circ_\alpha^S D = G^{\Sigma \times \alpha}(C) \wedge D$$

which entails that

$$\mathtt{Ext}(C \circ_\alpha^S D) = \{x \in \mathtt{Ext}(D) \,|\, S(C,x) \geq \Sigma \times \alpha\}$$

Proposition 2. $\circ_1^S = \circ^S$*, and for all* $1 \geq \alpha \geq \beta \geq 0$*:*

$$C \circ^S D \equiv C \circ_1^S D \vDash C \circ_\alpha^S D \vDash C \circ_\beta^S D \vDash C \circ_0^S D \equiv D$$

Moreover, for $\alpha < 1$, $\circ_\alpha^S$ satisfies postulates $(R1), (R3), (R4)$, and $(R5)$.

A proof of this proposition is given in appendix B.

However, if $\circ^S$ does not satisfy (R2), then for any $\alpha \in [0,1], \circ_\alpha^S$ neither does. The fact that, for $\alpha < 1$, $\circ_\alpha^S$ may not satisfy postulate (R6) is not surprising as the minimality criteria is loosened in α-revision.

5.2 α-conservative Adaptation

The α-conservative adaptation is defined from α-revision as conservative adaptation has been from revision. Given a target case `Target`, a source case `Source`, and domain knowledge `DK`, the α-conservative adaptation returns $\mathtt{Target\text{-}completed}_\alpha$ such that:

$$(\mathtt{DK} \wedge \mathtt{Source}) \circ_\alpha^S (\mathtt{DK} \wedge \mathtt{Target}) \equiv_{\mathtt{DK}} \mathtt{Target\text{-}completed}_\alpha \qquad (11)$$

From proposition 2, it comes that, for all $1 \geq \alpha \geq \beta \geq 0$:

$$\mathtt{Target\text{-}completed} \equiv \mathtt{Target\text{-}completed}_1 \vDash \mathtt{Target\text{-}completed}_\alpha$$
$$\vDash \mathtt{Target\text{-}completed}_\beta \vDash \mathtt{Target\text{-}completed}_0 \equiv \mathtt{Target}$$

5.3 α-conservative Adaptation in the Cooking Example

In example 4.3, given `DK` and `apples-nb` $= 0$, three parameters fully determine a model of `Target`: `pears-nb`, `sugar`, `pastry-mass`. In `Target-completed`, these parameters are fixed to precise values (`pastry-mass` $= 180$, `sugar` $= 50$, and `pears-nb` $= 4$). For $\alpha < 1$, $\texttt{Target-completed}_\alpha$ is less restrictive than `Target-completed`, and leaves some freedom in the parameter values. The representation of $\texttt{Target-completed}_\alpha$ needs 3D. Figure 1 represents cuts of its extension by the plane corresponding to the pair (`sugar`, `pastry-mass`), for `pears-nb` $= 4$ and `pears-nb` $= 3$. A point (x, y) of the graph `pears-nb` $= k$ is in the zone corresponding to α iff (`servings` $= 6$) $\wedge$ (`pastry-mass` $= y$) $\wedge$ (`sugar` $= x$) $\wedge$ (`apples-nb` $= 0$) $\wedge$ (`pears-nb` $= k$) is a model of $\texttt{Target-completed}_\alpha$.

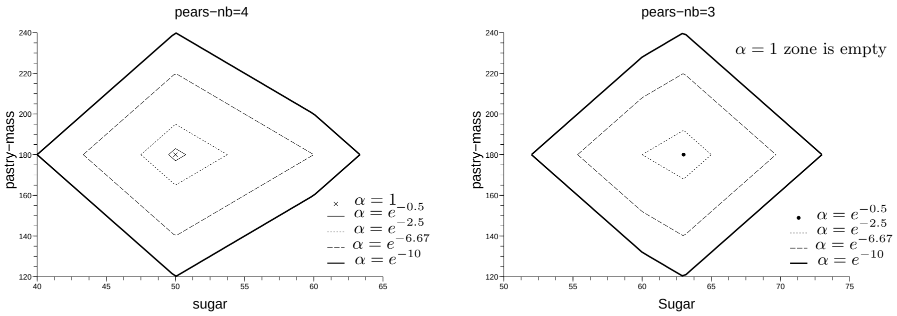

Fig. 1. Possible values for `sugar` and `pastry-mass` with `pears-nb` $= 4$ (left) and `pears-nb` $= 3$ (right). The graphs were made with Scilab [6].

For instance, with $\alpha = e^{-0.5}$, the possible values for `pears-nb`, `sugar`, and `pastry-mass` are: `pears-nb` $= 3$, `sugar` $= 63$, and `pastry-mass` $= 180$; or `pears-nb` $= 4$ and any values for `sugar` and `pastry-mass` in the corresponding zone of the left graph. In particular `Target-completed'` $\models$ $\texttt{Target-completed}_\alpha$.

6 Conclusion

The adaptation phase in CBR still lacks some formal definition. Conservative adaptation and its extensions can be considered as attempts of defining, at a semantic level, some approaches of adaptation based on revision operators. These latters may satisfy or not some of the AGM postulates, which has consequences on the properties of the adaptation function. A general question can be raised: *What are the adaptation approaches that can be covered by (more or less) conservative adaptation?* In [3], an answer is given in propositional logic. In the current paper, conservative adaptation is considered in the general framework of metric spaces.

Given a revision operator defined from a similarity measure S, conservative adaptation reduces the problem of adaptation to a problem of optimisation — determine the $x \in \mathcal{U}$ which maximise the function $y \mapsto S(\texttt{DK} \wedge \texttt{Source}, y)$ with the constraint $x \in \texttt{Ext}(\texttt{DK} \wedge \texttt{Target})$. The associated α-conservative adaptation is a relaxation of this optimisation problem —determine the $x \in \mathcal{U}$ such that $S(\texttt{DK} \wedge \texttt{Source}, x) \geq \alpha \times \sup_{y \in \texttt{ExtTarget}} S(\texttt{DK} \wedge \texttt{Source}, y)$— and is reduced to constraint programming problem. Powerful optimisation and constraint solvers as [7] could be used to solve large adaptation problems.

A prospect is to define fuzzy conservative adaptation that from a `Source` concept and `Target` concept would return a fuzzy concept `Target-completed` (an expression to be interpreted as a fuzzy subset `Ext(Target-completed)` of $\mathcal{U}$). The α-conservative adaptation is a first step towards it: from the parametered answer $\texttt{Target-completed}_\alpha$ can be built a fuzzy concept since a fuzzy set can be built from α-cuts [8]. However, in section 5, `Source` and `Target` are assumed to be classical concepts which prevents `Target-completed` to be further completed or retained as a new source case of the case base. The extension of fuzzy conservative adaptation to fuzzy concepts `Source` and `Target` is therefore a necessity for its coherence.

Another investigation direction is the construction of similarity measures so as to express adaptation rules, i.e. such that rule-based adaptation gives a result equivalent with conservative adaptation based on a similarity measure S. The obtained adaptation operators should be then compared to other formally defined adaptation approaches as, for example, the one presented in [9].

The implementation of a case-based reasoner based on conservative adaptation is a third objective. The previous concern is intended to make this reasoner as general as possible, applying the different adaptation rules that could be expressed under a similarity measure form. The claim is that such a reasoner could substitute many others as generalising them. This CBR reasoner should be applicable to a complex application, such as the one raised by the computer cooking contest (which explains, a posteriori, the choice of an example in the cooking domain).

A Fruit pie Adaptation Example Resolution

The minima of $x \mapsto d(\texttt{DK} \wedge \texttt{Source}, x)$ have to be found upon $\texttt{Ext}(\texttt{DK} \wedge \texttt{Target})$. However, some d_i are constant here, which simplifies the minima search, for all $x \in \texttt{Ext}(\texttt{DK} \wedge \texttt{Target})$, with the "$(x)$" dropped from the attributes:

$$d_1(\texttt{DK} \wedge \texttt{Source}, x) = |\texttt{servings} - 4| = 2$$
$$d_4(\texttt{DK} \wedge \texttt{Source}, x) = \left|\frac{\texttt{pastry-mass}}{6} - \frac{120}{4}\right| = 0$$
$$d_6(\texttt{DK} \wedge \texttt{Source}, x) = \left|\frac{\texttt{apples-nb}}{6} - \frac{2}{4}\right| = \frac{1}{2}$$

Indeed, $\texttt{servings} = 6$, $\texttt{apples-nb} = 0$, and $\texttt{pastry-mass}$ has no constraint and can be taken equal to 180. What remains to be minimised is:

$$\frac{1}{6}\left(w_2\left|\texttt{sugar} + 13 \times \texttt{pears-nb} - \frac{6}{4} \times 68\right| + w_3|\texttt{sugar} - 60|\right.$$
$$\left.+w_5|100 \times \texttt{pears-nb} - 360| + w_7 \times \texttt{pears-nb}\right)$$

which is a sum of affine per parts functions with two parameters. Minima can be searched one parameter at a time. First, let us focus on $\texttt{sugar}$, $\texttt{pears-nb}$ being taken as constant. The value of $\texttt{sugar}$ should be a minimum of the function $x \mapsto w_2|x - (102 - 13 \times \texttt{pears-nb})| + w_3|x - 60|$, i.e.:

- If $w_2 > w_3$ then $\texttt{sugar} = 102 - 13 \times \texttt{pears-nb}$ and the sweet mass per person is preserved.
- If $w_2 < w_3$ then $\texttt{sugar} = 60$ the sugar mass per person is preserved.
- If $w_2 = w_3$, any value between 60 and $102 - 13 \times \texttt{pears-nb}$ can be given to $\texttt{sugar}$.

It is assumed that the preservation of $\texttt{sweet}$ is to be preferred to the preservation of $\texttt{sugar}$ —sugar is used in cooking to adjust the sweet taste. Therefore $w_2 > w_3$. What remains to be minimised is then:

$$w_3\,|(42 - 13 \times \texttt{pears-nb})| + w_5\,|100 \times \texttt{pears-nb} - 360| + w_7 \times \texttt{pears-nb}$$

As previously, some relative importance relation between term considerations reduce the set of alternatives to explore. $\texttt{fruit-mass}$ preservation is more important than $\texttt{pears-nb}$'s, thus $100 \times w_5 > w_7$, 100 being the average pear mass: this coefficient is used in the inequality for normalisation. $x \mapsto w_5|100x - 360| + w_7x$ decreases for $x \leq \frac{360}{100} = 3.6$, and increases for $x \geq 3.6$. $x \mapsto w_3|42 - 13x|$ also decreases for $x < \frac{42}{13} \approx 3.23$ and then increases. As both decrease before 3 and increase after 4, the minima is then reached for $\texttt{pears-nb} = 3$ or 4:

- For $\texttt{pears-nb} = 4$, the term value is $w_3 \times 10 + w_5 \times 40 + w_7 \times 4$.
- For $\texttt{pears-nb} = 3$, the term value is $w_3 \times 3 + w_5 \times 60 + w_7 \times 3$.

Which one is minimal depends on the sign of $20 \times w_5 - 7 \times w_3 - w_7$. The previous considerations cannot help to determine it, consider the following two sets of w_i:

- $w_1 = 10$, $w_2 = 5$, $w_3 = 1$, $w_4 = 1$, $w_5 = 1$, $w_6 = w_7 = 10$, the constraints $w_2 > w_3$ and $100 \times w_5 > w_7$ are satisfied and $20 \times w_5 - 7 \times w_3 - w_7 > 0$. The minima of $x \mapsto d(\texttt{DK} \wedge \texttt{Source}, x)$ with $x \in \texttt{Ext(Target)}$ is then reduced to the single tuple $x = (6, 102, 50, 180, 400, 0, 4)$.
- $w_1 = 10$, $w_2 = 5$, $w_3 = 2$, $w_4 = 1$, $w_5 = 1$, $w_6 = w_7 = 10$, as before $w_2 > w_3$ and $100 \times w_5 > w_7$ but now $20 \times w_5 - 7 \times w_3 - w_7 < 0$. And $x \mapsto d(\texttt{DK} \wedge \texttt{Source}, x)$ with $x \in \texttt{Ext(Target)}$ minima is reduced to a single tuple too: $y = (6, 102, 63, 180, 300, 0, 3)$.

Unlike for the constraint $w_2 > w_3$ any choice of values for the w_i will not guarantee that `sugar` preservation will be given priority over `pears-nb` preservation as in the first case or the opposite as in the second case, it depends on the case attributes values. In this paper, the first set of weights is chosen and conservative adaptation will return the concept `Target-completed`:

$$\begin{aligned}\texttt{Target-completed} \equiv_{\texttt{DK}} & (\texttt{servings} = 6) \wedge (\texttt{pastry-mass} = 180) \wedge (\texttt{sugar} = 50) \\ & \wedge (\texttt{apples-nb} = 0) \wedge (\texttt{pears-nb} = 4)\end{aligned}$$

B Proofs

Proposition 1

(i) (R1) is satisfied by construction of $\circ^S$: $C \circ^S D = G^{\Sigma}(C) \wedge D \vDash D$.
(R4): If $C \equiv C'$ and $D \equiv D'$, then $G^{\Sigma}(C) \equiv G^{\Sigma}(C')$ so $C \circ^S D = G^{\Sigma}(C) \wedge D \equiv G^{\Sigma}(C') \wedge D' = C' \circ^S D'$.
For (R5) and (R6), two cases are to be considered:
First case: $(C \circ^S D) \wedge F \vDash \bot$, (R5) and (R6) are automatically satisfied.
Second case: $(C \circ^S D) \wedge F \nvDash \bot$, then $\texttt{Ext}((C \circ^S D) \wedge F) \neq \emptyset$. Let $x \in \texttt{Ext}((C \circ^S D) \wedge F)$. According to $\circ^S$ definition, since $x \in \texttt{Ext}(C \circ^S D)$:

$$\begin{aligned}S(C,x) = S(C,D) = \sup_{u \in \texttt{Ext}(D)} S(C,u) \geq \sup_{u \in \texttt{Ext}(D) \cap \texttt{Ext}(F)} S(C,u) \\ \geq S(C, D \wedge F)\end{aligned}$$

However, according to (R1), $\texttt{Ext}(C \circ^S D) \subseteq \texttt{Ext}(D)$, so $x \in \texttt{Ext}(D \wedge F)$ and $S(C, D \wedge F) = \sup_{u \in \texttt{Ext}(D \wedge F)} S(C,u) \geq S(C,x)$, therefore $S(C,D) = S(C, D \wedge F)$. And finally:

$$\begin{aligned}(C \circ^S D) \wedge F = G^{S(C,D)}(C) \wedge D \wedge F = G^{S(C,D \wedge F)}(C) \wedge D \wedge F \\ = C \circ^S (D \wedge F) \quad \text{thus, (R5) and (R6) are satisfied.}\end{aligned}$$

(ii) Satisfaction of $C \wedge D \vDash C \circ^S D$: the case $C \wedge D \vDash \bot$ is trivial. Consider now the case $C \wedge D \nvDash \bot$, let x be in $\texttt{Ext}(C \wedge D)$, $x \in \texttt{Ext}(C)$ thus $S(C,x) = 1$ and so $x \in \texttt{Ext}(G^1(C) \wedge D) = \texttt{Ext}(C \circ^S D)$. This shows that $\texttt{Ext}(C \wedge D) \subseteq \texttt{Ext}(C \circ^S D)$ and thus $C \wedge D \vDash C \circ^S D$.

(iii) **(8) implies (R2):** Assume $(S(A,x) \Rightarrow x \in A)$, then for $C \in \mathcal{L}_C$, $G^1(C) \equiv C$, indeed $\texttt{Ext}(G^1(C)) = \{x \in \mathcal{U} \mid S(\texttt{Ext}(C), x) = 1\} = C$. (R2) follows from this property: if $C \wedge D$ is satisfiable, then $\texttt{Ext}(C \wedge D) \neq \emptyset$ and $S(C,D) = 1$ ($\Sigma = 1$), thus

$$C \circ^S D = G^1(C) \wedge D \equiv C \wedge D$$

(R2) implies (8): Assume (R2) is satisfied, let A be in $\mathcal{E}$, x in $\mathcal{U}$, and C in $\mathcal{L}_C$ such that $\texttt{Ext}(C) = A$. Assume $S(A,x) = 1 > 0$, from the convention established in definition 1 it follows that $A \neq \emptyset$, so $A = \texttt{Ext}(C) = \texttt{Ext}(C) \cap \mathcal{U} = \texttt{Ext}(C) \cap \texttt{Ext}(\top) = \texttt{Ext}(C \wedge \top) \neq \emptyset$. (R2) implies that $C \circ^S \top \equiv C \wedge \top \equiv C$, thus $x \in \texttt{Ext}(C \circ^S \top) = \texttt{Ext}(C) = A$. and $x \in A$.

(iv) **(9) implies (R3):** Assume that (9) is satisfied, if D is satisfiable and $\Sigma = S(C, D)$, then (9) implies that there is an x in $\texttt{Ext}(D)$ such that $S(C, x) = \Sigma$. Thus $\texttt{Ext}(C \circ^S D) \neq \emptyset$ and $C \circ^S D$ is satisfiable.

(R3) implies (9): Assume that (R3) is satisfied, let A and B be in $\mathcal{E}$ with $B \neq \emptyset$, $\Sigma = S(A, B)$, and C and D in $\mathcal{L}_C$ such that $\texttt{Ext}(C) = A$ and $\texttt{Ext}(D) = B$. D is satisfiable so, according to (R3), $C \circ^S D$ is satisfiable too. However $\texttt{Ext}(C \circ^S D) = \{x \in B \mid S(A, x) = \Delta\}$, it follows that there is an x in B such that $S(A, x) = \Sigma$.

Proposition 2

– $\circ_1^S = \circ^S$, indeed, for C and D in $\mathcal{L}_C$ with $\Sigma = S(C, D)$:

$$C \circ_1^S D = G^{\Sigma \times 1}(C) \wedge D = G^{\Sigma} \wedge D = C \circ^S D$$

– Similarly, for C and D in $\mathcal{L}_C$ $C \circ_0^S D \equiv D$, indeed $\texttt{Ext}(G^0(C)) = \{x \in \mathcal{U} \mid S(C, x) \geq 0\} = \mathcal{U}$, thus $G^0(C) \equiv \top$. Let $\Sigma = S(C, D)$,

$$C \circ_0^S D = G^{\Sigma \times 0} \wedge D = G^0(C) \wedge D \equiv \top \wedge D \equiv D$$

– For α and β such that $1 \geq \alpha \geq \beta \geq 0$, and C, D in $\mathcal{L}_C$ with $\Sigma = S(C, D)$:

$$\begin{aligned}\texttt{Ext}(G^{\Sigma \times \alpha}(C)) &= \{x \in \mathcal{U} \mid S(C, x) \geq \Sigma \times \alpha\} \\ &\subseteq \{x \in \mathcal{U} \mid S(C, x) \geq \Sigma \times \beta\} = \texttt{Ext}(G^{\Sigma \times \beta}(C))\end{aligned}$$

Thus $G^{\Sigma \times \alpha}(C) \models G^{\Sigma \times \beta}(C)$ and

$$C \circ_\alpha^S D = G^{\Sigma \times \alpha}(C) \wedge D \models G^{\Sigma \times \beta}(C) \wedge D = C \circ_\beta^S D$$

References

1. Riesbeck, C.K., Schank, R.C.: Inside Case-Based Reasoning. Lawrence Erlbaum Associates, Inc., Hillsdale, New Jersey (1989)
2. Kolodner, J.: Case-Based Reasoning. Morgan Kaufmann, Inc. (1993)
3. Lieber, J.: Application of the Revision Theory to Adaptation in Case-Based Reasoning: The Conservative Adaptation. In: Proceedings of the 7th International Conference on Case-Based Reasoning (ICCBR). (2007)
4. Alchourrón, C.E., Gärdenfors, P., Makinson, D.: On the logic of theory change: partial meet functions for contraction and revision. Journal of Symbolic Logic **50** (1985) 510–530
5. Katsuno, H., Mendelzon, A.: Propositional knowledge base revision and minimal change. Artificial Intelligence **52**(3) (1991) 263–294
6. http://www.scilab.org/: Scilab software. last consult : March 2008
7. Puget, J.F.: A C++ implementation of CLP. In: Proceedings of the Second Singapore International Conference on Intelligent Systems, Singapore (1994)
8. Zadeh, L.A.: Fuzzy Sets. Information and Control **8** (1965) 338–353
9. Hüllermeier, E.: Credible case-based inference using similarity profiles. IEEE Transactions on Knowledge and Data Engineering **19**(6) (2007) 847–858