



Belief Merging-based Case Combination

Julien Cojan, Jean Lieber

► **To cite this version:**

Julien Cojan, Jean Lieber. Belief Merging-based Case Combination. 8th International Conference on Case-Based Reasoning - ICCBR 2009, Jul 2009, Seattle, United States. pp.105-119, 10.1007/978-3-642-02998-1_9. inria-00421724

HAL Id: inria-00421724

<https://hal.inria.fr/inria-00421724>

Submitted on 2 Oct 2009

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Belief Merging-based Case Combination

Julien Cojan and Jean Lieber

Orpailleur, LORIA, CNRS, INRIA, Nancy University,
BP 239, 54506 Vandœuvre-lès-Nancy
{Julien.Cojan, Jean.Lieber}@loria.fr

Abstract. Integrity constraint belief merging aims at producing from several knowledge bases, that may be mutually inconsistent, a synthetic knowledge base satisfying a given integrity constraint. It is applied here to case combination for case-based reasoning. This approach is shown to extend Eyke Hüllermeier's credible case-based inference and to be reducible under some assumptions to linear programming.

1 Introduction

Case-based reasoning (CBR [1]) aims at solving problems thanks to a set of previously solved problems accompanied with their solutions (the source cases). At least, two general approaches of CBR exist. The first one consists in the adaptation of a sole retrieved case. The second one consists in the combination of $k \geq 1$ retrieved cases. Actually, case combination is a generalization of case adaptation: this latter is a case combination with $k = 1$.

In [2], an approach to adaptation based on a belief revision operator, the so-called conservative adaptation, is presented. A belief revision operator $\dot{+}$ associates to two knowledge bases A and B a knowledge base $A \dot{+} B$ that entails B and keeps as much as possible from A . Conservative adaptation of a source case $Srce$ by a target case Tgt consists in a revision $ctxt(Srce) \dot{+} ctxt(Tgt)$ where $ctxt(Srce)$ and $ctxt(Tgt)$ are respectively $Srce$ and Tgt interpreted according to the domain knowledge.

Now, belief revision is generalized by integrity constraint (IC) belief merging [3] that integrates several knowledge bases altogether with constraints.

The purpose here is to substitute belief revision by IC belief merging in conservative adaptation to define a case combination approach.

After the introduction of a running example and a preliminary section, the IC belief merging theory is presented in section 4 and its application to case combination in section 5. Section 6 shows that credible case-based inference [4] is a kind of case combination. The computation of belief merging in numerical spaces is studied in section 7, before a related work review and the conclusion.

2 Introduction of the running example

Assume you have an egg allergic guest and you have the experience of some dishes that can be made without eggs. Your guest loves chocolate and chocolate mousse would be

a perfect desert, even if you don't have an egg-free recipe at disposal. Several recipes –the source cases Srce_i – can be combined to solve the target case Tgt :

Tgt : egg-free chocolate mousse recipe
 Srce_1 : chocolate mousse recipe
 Srce_2 : egg-free Chantilly recipe
 Srce_3 : egg-free chocolate cream recipe

The expected solution would be to follow the main lines from Srce_1 but to substitute the egg-snow by egg-free Chantilly from Srce_2 and the chocolate mix –that contains egg yolks– by the chocolate cream from Srce_3 .

3 Preliminaries

3.1 Set theory notations

A boolean interpretation \mathcal{I} on a set of variables $\mathcal{V} = \{a_1, a_2, \dots, a_n\}$ is a mapping from \mathcal{V} to the set of boolean values $\mathbb{B} = \{\text{true}, \text{false}\}$. It can be assimilated to $(x_1, \dots, x_n) \in \mathbb{B}^n$ where $\mathcal{I}(a_i) = x_i$ for $i = 1$ to n . Thus, the models $[f]$ of a formula f –the set of the interpretations satisfying f – is assimilated to a subset of $\mathcal{U} = \mathbb{B}^n$. Following the *principle of irrelevance of syntax*, the formulas are assimilated in this paper to their models and thus to subsets of \mathcal{U} . In particular conjunction \wedge , entailment \vDash , and logical equivalence \equiv represent intersection \cap , inclusion \subseteq , and set equality $=$ on subsets of \mathcal{U} . More precisely, $[f \wedge g] = [f] \cap [g]$, $f \vDash g$ iff $[f] \subseteq [g]$, and $f \equiv g$ iff $[f] = [g]$.

Propositional logic can then be generalized considering any set \mathcal{U} , in particular attribute-values formalisms correspond to sets of the kind $\mathcal{U} = D_1 \times \dots \times D_n$ where D_1, \dots, D_n are more elementary sets like integers \mathbb{Z} , positive real numbers \mathbb{R}^+ , etc.

In order to ease the reading, a variable x_i that can take any value from D_i is just written ' $_$ ', eg. if $n = 3$ and $a_1 \in D_1$, $a_3 \in D_3$, $\{(a_1, _, a_3,)\} = \{(a_1, x_2, a_3) \mid x_2 \in D_2\}$. So, $\mathcal{U} = \{(_, _, _)\}$.

3.2 Metric spaces

Definition 1. A pseudo-distance¹ on a set \mathcal{U} is a function $d : \mathcal{U} \times \mathcal{U} \rightarrow [0; +\infty]$ satisfying the separation axiom:

$$\text{for any } x, y \in \mathcal{U}, d(x, y) = 0 \text{ iff } x = y$$

A distance on \mathcal{U} is a pseudo-distance on \mathcal{U} taking only finite values (for any $x, y \in \mathcal{U}$, $d(x, y) < +\infty$) that satisfies the symmetry axiom:

$$\text{for any } x, y \in \mathcal{U}, d(x, y) = d(y, x)$$

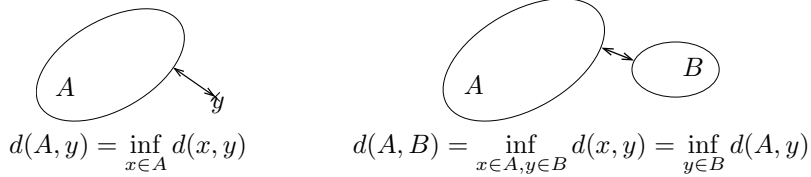
¹ We slightly abuse words here as this may not be in agreement with common definition of pseudo-distance.

and the triangular inequality:

$$\text{for any } x, y, z \in \mathcal{U}, d(x, z) \leq d(x, y) + d(y, z)$$

A (pseudo-)metric space is a pair (\mathcal{U}, d) where d is a (pseudo-)distance on the set \mathcal{U} .

$2^{\mathcal{U}}$ denotes the set of subsets of \mathcal{U} . A pseudo-distance d on \mathcal{U} is extended on subsets of \mathcal{U} as follows:



$$d(A, y) = \inf_{x \in A} d(x, y) \qquad d(A, B) = \inf_{x \in A, y \in B} d(x, y) = \inf_{y \in B} d(A, y)$$

where $A, B \in 2^{\mathcal{U}}$, and $x, y \in \mathcal{U}$ (note that $d : 2^{\mathcal{U}} \times 2^{\mathcal{U}} \rightarrow [0; +\infty]$ is not necessary a pseudo-distance).

$A \in 2^{\mathcal{U}}$ is bounded if there exists $K \in \mathbb{R}^+$ such that for each $x, y \in A$, $d(x, y) \leq K$. Given a pseudo-distance on \mathcal{U} , $x, y \in \mathcal{U}$ and $r \in [0; +\infty]$, the *right closed ball of center x and radius r* is the set $\mathcal{B}_r^{\mathcal{U}}(x) = \{y \in \mathcal{U} \mid d(x, y) \leq r\}$ and the *left closed ball of center y and radius r* is the set $\mathcal{B}_r^{\mathcal{U}}(y) = \{x \in \mathcal{U} \mid d(x, y) \leq r\}$. The distinction between these two definitions is relevant when d is not symmetrical, otherwise they are equal.

Definition 2. A (pseudo-)discrete metric space (\mathcal{U}, d) is a (pseudo-)metric space such that, for any $x \in \mathcal{U}$ and $r \in [0; +\infty[$, $\mathcal{B}_r^{\mathcal{U}}(x)$ and $\mathcal{B}_r^{\mathcal{U}}(x)$ are finite.

This definition is stronger than the usual definition of (pseudo-)discrete metric space which states that for any $x \in \mathcal{U}$, there is an $r > 0$ such that $\mathcal{B}_r^{\mathcal{U}}(x) = \mathcal{B}_r^{\mathcal{U}}(x) = \{x\}$.

If (\mathcal{U}, d) is discrete then for A, B two subsets of \mathcal{U} , if A is bounded, then A is finite and $d(A, B) = d(A, y)$ for some $y \in B$. Moreover $\{z \in \mathcal{U} \mid d(A, z) = 0\} = A$ (i.e. A is closed).

To avoid continuity issues, like no minimum for a distance, only discrete spaces will be considered in the following. In particular \mathbb{R} will be approximated by decimals of a fixed maximum length.

3.3 CBR: Definitions and Hypotheses

Cases and domain knowledge. Case-based reasoning (CBR) aims at solving problems of a given application domain with the help of previous solving episodes, or *cases*. In this paper, these notions are formalized as follows. Let \mathcal{U}_{pb} and \mathcal{U}_{sol} be two sets: $x \in \mathcal{U}_{\text{pb}}$ is a *problem instance*, $X \in \mathcal{U}_{\text{sol}}$ is a *solution instance*. A *problem pb* is a class of problem instances: $\text{pb} \in 2^{\mathcal{U}_{\text{pb}}}$. A *solution sol* is a class of solution instances: $\text{sol} \in 2^{\mathcal{U}_{\text{sol}}}$.

Let $\mathcal{U} = \mathcal{U}_{\text{pb}} \times \mathcal{U}_{\text{sol}}$. A *case* is a subset of \mathcal{U} . A *singleton case* is a case with only one element: $\text{Case} = \{(x, X)\}$. Given x and Case , a problem instance and a case, $\Phi_x(\text{Case})$ denotes the projection of x on solution instances:

$$\Phi_x(\text{Case}) = \{X \in \mathcal{U}_{\text{sol}} \mid (x, X) \in \text{Case}\}$$

In particular, if $\text{Case} = \{(x, X)\}$ is a singleton case then $\Phi_x(\text{Case}) = \{X\}$.

The relationship stating that a solution sol solves a problem pb is formalized by a binary relation \rightsquigarrow on $\mathcal{U} = \mathcal{U}_{\text{pb}} \times \mathcal{U}_{\text{sol}}$: pb is solved by sol if, for every $x \in \text{pb}$ exists $X \in \text{sol}$ such that $x \rightsquigarrow X$.

\rightsquigarrow is not assumed to be completely known. By contrast, it is assumed that a finite set of cases, the *case base* CB and some domain knowledge are available, and that any case of the case base CB –called a *source case* and denoted by Srce – has the following property: for each $x \in \mathcal{U}_{\text{pb}}$, if $\Phi_x(\text{Srce}) \neq \emptyset$ then there exists $X \in \Phi_x(\text{Srce})$ such that $x \rightsquigarrow X$ (see figure 1).

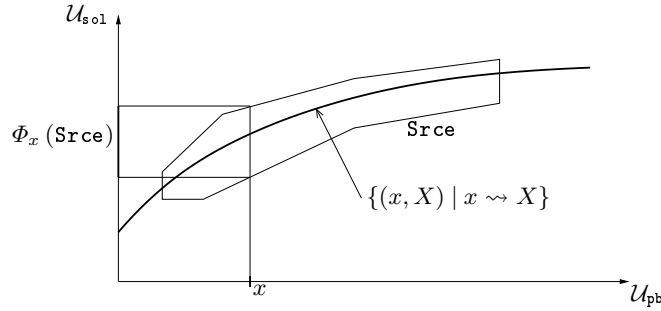


Fig. 1. A source case Srce .

The target case, denoted by Tgt , is the case for which the solution part has to be made more precise by the current CBR session. In general, before the CBR inference, nothing is known about this solution: $\text{Tgt} = \text{tgt} \times \mathcal{U}_{\text{sol}}$ with $\text{tgt} \in 2^{\mathcal{U}_{\text{pb}}}$, the *target problem* (Fig. 2). A *singleton target problem* is a target problem with only one element: $\text{tgt} = \{x_0\}$.

The *domain knowledge* states that some pairs (x, X) are not licit: $x \not\rightsquigarrow X$. Thus, it corresponds to a necessary condition for $x \rightsquigarrow X$. It is formalized by a subset DK of $\mathcal{U} = \mathcal{U}_{\text{pb}} \times \mathcal{U}_{\text{sol}}$ and satisfies the implication: $x \rightsquigarrow X$ implies $(x, X) \in \text{DK}$ (or, by contraposition, $(x, X) \notin \text{DK}$ implies $x \not\rightsquigarrow X$). For Case , a given case, its elements (x, X) that are not consistent with DK have not to be considered (they are known to be illicit). Thus, Case is to be considered in conjunction with DK , i.e. in its context $\text{ctxt}(\text{Case}) = \text{DK} \cap \text{Case}$. If no domain knowledge is available, then $\text{DK} = \mathcal{U}$: every pair $(x, X) \in \mathcal{U}$ is *a priori* licit.

Case-based inference. Given a target case Tgt , a case base CB and the domain knowledge $\text{DK} \subseteq \mathcal{U}$, the case-based inference aims at proposing a case SolvedTgt that makes Tgt more precise (Fig. 2):

$$\text{DK} \cap \text{SolvedTgt} \subseteq \text{DK} \cap \text{Tgt} \quad (1)$$

Two main approaches for this inference are described in the CBR literature. The first one is based on adaptation and the second one on combination.

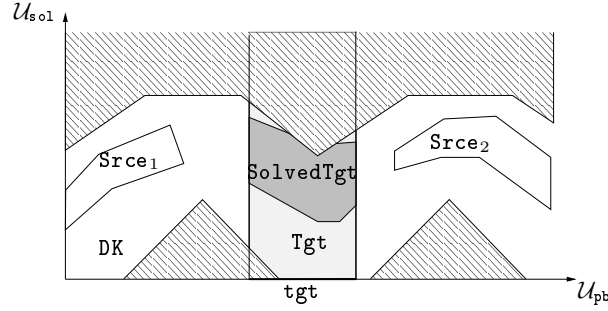


Fig. 2. Result of a CBR session, Tgt has been specialized into SolvedTgt.

3.4 Formalization of the example

The cooking problem specification consists in the three following conditions: whether the dish is frothy, whether it contains eggs, and whether it contains chocolate. $\mathcal{U}_{pb} = \mathcal{U}_1 \times \mathcal{U}_2 \times \mathcal{U}_3$.

Only the ingredient amounts and the volume of froth will be considered for the solution. All the values are taken for a single serving –which explains the real values for the eggs number. $\mathcal{U}_{so1} = \mathcal{U}_4 \times \mathcal{U}_5 \times \dots \times \mathcal{U}_9$. And finally $\mathcal{U} = \mathcal{U}_{pb} \times \mathcal{U}_{so1}$. The cases values are given in table 1.

	attribute	Tgt	Srce ₁	Srce ₂	Srce ₃
$\mathcal{U}_1 = \mathbb{B}$	is frothy	true	true	true	false
$\mathcal{U}_2 = \mathbb{B}$	has eggs	false	true	false	false
$\mathcal{U}_3 = \mathbb{B}$	has chocolate	true	true	false	true
$\mathcal{U}_4 = \mathbb{R}$	froth volume (ml)	–	200	225	0
$\mathcal{U}_5 = \mathbb{R}$	number of eggs	–	0.67	0	0
$\mathcal{U}_6 = \mathbb{R}$	chocolate mass (g)	–	35	0	25
$\mathcal{U}_7 = \mathbb{R}$	cream mass (g)	–	10	0	125
$\mathcal{U}_8 = \mathbb{R}$	sugar mass (g)	–	65	50	–
$\mathcal{U}_9 = \mathbb{R}$	soya volume (ml)	–	0	170	0

Table 1. Formalization of the egg free chocolate mousse example.

The domain knowledge is given by DK:

$$DK = R_{vFroth} \cap R_{hasFroth} \cap R_{hasEggs} \cap R_{hasChocolate}$$

where R_{vFroth} states that the froth obtained from an egg is at most 200 ml and 220 ml from 170 ml of soya milk ($220/170 \simeq 1.32$):

$$R_{vFroth} = \{(_, _, _, vFroth, eggs, _, _, _, soya) \mid vFroth \leq 200 \times eggs + 1.32 \times soya\}$$

R_{hasFroth} , R_{hasEggs} , and $R_{\text{hasChocolate}}$ force `hasFroth` (resp. `hasEggs` and `hasChocolate`) to be true only when there is froth (resp. eggs and chocolate) in the recipe:

$$\begin{aligned}
R_{\text{hasFroth}} &= \{(\text{hasFroth}, _, _, \text{vFroth}, _, _, _, _) \mid \\
&\qquad\qquad\qquad \text{hasFroth} = \text{true iff vFroth} \neq 0\} \\
R_{\text{hasEggs}} &= \{(_, \text{hasEggs}, _, _, \text{eggs}, _, _, _) \mid \text{hasEggs} = \text{false iff eggs} = 0\} \\
R_{\text{hasChocolate}} &= \{(_, _, \text{hasChocolate}, _, _, \text{chocolate}, _, _) \mid \\
&\qquad\qquad\qquad \text{hasChocolate} = \text{false iff chocolate} = 0\}
\end{aligned}$$

Therefore, the fifth component of $\text{DK} \cap \text{Tgt}$ is $(\text{DK} \cap \text{Tgt})_5 = 0$.

The following distance d is defined on \mathcal{U} , for $x, y \in \mathcal{U}$, by:

$$\begin{aligned}
d(x, y) &= d_{\text{pb}}(x, y) + d_{\text{so1}}(x, y) \\
d_{\text{pb}}(x, y) &= \sum_{i=1}^3 w_i \times \begin{cases} 0 & \text{if } x_i = y_i \\ 1 & \text{otherwise} \end{cases} \\
d_{\text{so1}}(x, y) &= \sum_{i=4}^9 w_i |y_i - x_i|
\end{aligned}$$

The choice of the weights w_i reflects the relative importance of the different dimensions. In particular here, the eggs and soya milk are used to generate froth, thus froth's dimension should get a higher importance than egg's and soya's.

4 Integrity Constraint Belief Merging

The following scenario illustrates the notion of integrity constraint belief merging. Let us consider an agent that has some knowledge about the world that he/she considers to be inviolable: this is his/her integrity constraints (IC). Now, he/she receives from several sources some knowledge about the world. Taking the conjunction of all these sources does not necessarily lead to a knowledge base consistent with the IC of the agent.

Various operators may be used to merge these sources of knowledge in a result consistent with the integrity constraints. Such an IC merging operator should satisfy some postulates [3], as it is explained in section 4.1. A straightforward generalization of this work to a more general formalism is presented in section 4.2. Then, an example of IC merging operator is presented (section 4.3).

4.1 IC Merging in Propositional Logic

The definition below is a reformulation from [3], with substitution of propositional formulas on \mathcal{V} by subsets of \mathbb{B}^n :

Definition 3. Let $\mathcal{U} = \mathbb{B}^n$. An IC merging operator on \mathcal{U} is a mapping $\Delta : (\text{IC}, M) \mapsto \Delta_{\text{IC}}(M)$, where $\text{IC} \in 2^{\mathcal{U}}$ is the integrity constraint and M –the set of beliefs to be merged– is a finite multi-set of non empty subsets of \mathcal{U} , satisfying the following postulates:

- (Δ -1) $\Delta_{IC}(M) \subseteq IC$
- (Δ -2) If $IC \neq \emptyset$ then $\Delta_{IC}(M) \neq \emptyset$
- (Δ -3) If $\bigcap M \cap IC \neq \emptyset$ then $\Delta_{IC}(M) = \bigcap M \cap IC$
- (Δ -4) If $A_1 \subseteq IC$ and $A_2 \subseteq IC$ then
 $\Delta_{IC}(\{A_1, A_2\}) \cap A_1 \neq \emptyset$ iff $\Delta_{IC}(\{A_1, A_2\}) \cap A_2 \neq \emptyset$
- (Δ -5) $\Delta_{IC}(M_1) \cap \Delta_{IC}(M_2) \subseteq \Delta_{IC}(M_1 \cup M_2)$
- (Δ -6) If $\Delta_{IC}(M_1) \cap \Delta_{IC}(M_2) \neq \emptyset$ then $\Delta_{IC}(M_1 \cup M_2) \subseteq \Delta_{IC}(M_1) \cap \Delta_{IC}(M_2)$
- (Δ -7) $\Delta_{IC_1}(M) \cap IC_2 \subseteq \Delta_{IC_1 \cap IC_2}(M)$
- (Δ -8) If $\Delta_{IC_1}(M) \cap IC_2 \neq \emptyset$ then $\Delta_{IC_1 \cap IC_2}(M) \subseteq \Delta_{IC_1}(M)$

where $IC, IC_1, IC_2, A_1, A_2 \in 2^{\mathcal{U}}$, $A_1 \neq \emptyset$, $A_2 \neq \emptyset$ and M, M_1 , and M_2 are three multisets of non empty subsets of \mathcal{U} . $M_1 \cup M_2$ denotes the multi-set union of M_1 and M_2 .

Note that there is another postulate in [3] that expresses the principle of irrelevance of syntax. By working on interpretations the independence to the syntax is already implied, thus this postulate is reformulated in the following tautology: if $M_1 = M_2$ and $IC_1 = IC_2$ then $\Delta_{IC_1}(M_1) = \Delta_{IC_2}(M_2)$.

IC merging and belief revision. Belief revision is usually presented as the change of an agent belief A after some facts B are known by him. Some beliefs in A may have to be left as being in contradiction with B but others may not have interference and should be kept. The resulting belief $A \dot{+} B$ should entail B and keep “as much as possible” of A . The notion of IC merging can be considered as a generalization of the notion of revision in the sense that if $\dot{+}$ is defined by

$$A \dot{+} B = \Delta_B(\{A\}) \quad (2)$$

with Δ an IC merging operator, then $\dot{+}$ satisfies the postulates of revision (i.e., the postulates of the “AGM theory” [5], that have been applied to propositional logic in [6]).

4.2 Generalization

The definition of an IC merging operator on a given set \mathcal{U} is the same as definition 3, except that \mathcal{U} is any set, not necessarily \mathbb{B}^n .

Definition 4. A pre-IC merging operator on a set \mathcal{U} is a mapping $\Delta : (IC, M) \mapsto \Delta_{IC}(M)$, where $IC \in 2^{\mathcal{U}}$ and M is a finite multi-set of subsets of \mathcal{U} , satisfying the postulates (Δ -1), (Δ -3) to (Δ -8), and (Δ -2') a weakened version of (Δ -2):

$$(\Delta$$
-2') If $IC \neq \emptyset$ and every set in M is bounded, then $\Delta_{IC}(M) \neq \emptyset$

Definition 5. An IC merging operator is a pre-IC merging operator that satisfies postulate (Δ -2).

4.3 Example of pre-IC Merging Operator

The operator presented in this section is inspired from one of the DA² operators [3].

Let (\mathcal{U}, d) be a pseudo-metric space. Let d^Σ be the function associating to the pair (M, y) –where M is a finite multiset of subsets of \mathcal{U} , and $y \in \mathcal{U}$ – the real number

$$d^\Sigma(M, y) = \sum_{A \in M} d(A, y)$$

For $IC \in 2^\mathcal{U}$, let $d^\Sigma(M, IC) = \inf_{y \in IC} d^\Sigma(M, y)$. Let $\Delta^{d, \Sigma}$ be the operator defined as follows:

$$\Delta_{IC}^{d, \Sigma}(M) = \{y \in IC \mid d^\Sigma(M, y) = d^\Sigma(M, IC)\}$$

Proposition 1. *If (\mathcal{U}, d) is discrete, $\Delta^{d, \Sigma}$ satisfies the postulates $(\Delta-1)$, $(\Delta-2')$, $(\Delta-3)$, $(\Delta-5)$, $(\Delta-6)$, $(\Delta-7)$, and $(\Delta-8)$.*

If d is symmetrical (i.e., it satisfies the symmetry axiom) then $\Delta^{d, \Sigma}$ satisfies $(\Delta-4)$. It is then a pre-IC merging operator.

A proof for this proposition is given in appendix.

Note, that $(\Delta-2)$ is not satisfied in general. Consider $\mathcal{U} = \{\log(n) \mid n \in \mathbb{N} \setminus \{0\}\}$ with $d(x, y) = |y - x|$, $IC = \{\log(2p) \mid p \in \mathbb{N} \setminus \{0\}\}$, $M = \{A\}$ with $A = \{\log(2p + 1) \mid p \in \mathbb{N}\}$. Then $\Delta_{IC}^{d, \Sigma}(M) = \emptyset$.

5 Case Combination based on a pre-IC Merging Operator

5.1 Conservative Adaptation

Conservative adaptation [2] is an approach to adaptation based on belief revision. Its principle is to reuse “as much as possible” of Srce while being consistent with Tgt . Both Srce and Tgt must be considered according to domain knowledge DK . As for belief revision, the meaning of “as much as possible” is variable. The idea is to define conservative adaptation parameterized by a belief revision operator $\dot{+}$:

$$\text{SolvedTgt} = (\text{DK} \cap \text{Srce}) \dot{+} (\text{DK} \cap \text{Tgt})$$

This inference is called $\dot{+}$ -conservative adaptation.

5.2 Δ -combination of cases

Definition. Let $\text{SCS} = \{\text{Srce}_1, \dots, \text{Srce}_k\}$ be a subset of the case base CB . Let Δ be a pre-IC merging operator on $\mathcal{U} = \mathcal{U}_{\text{pb}} \times \mathcal{U}_{\text{so1}}$. Δ -combination of cases is a generalization of $\dot{+}$ -conservative adaptation: $\dot{+}$ is generalized in Δ and the sole selected case is generalized in the set of source cases SCS . Thus, $\text{SolvedTgt} = \text{CC}_{\Delta}^{\text{DK}}(\text{SCS}, \text{Tgt})$ with

$$\text{CC}_{\Delta}^{\text{DK}}(\{\text{Srce}_1, \dots, \text{Srce}_k\}, \text{Tgt}) = \Delta_{\text{DK} \cap \text{Tgt}}(\{\text{DK} \cap \text{Srce}_1, \dots, \text{DK} \cap \text{Srce}_k\}) \quad (3)$$

I.e., the contribution of the source cases are merged in a result that specializes the target case.

If $k = \text{card}(\text{SCS}) = 1$, then $\text{CC}_{\Delta}^{\text{DK}}(\text{SCS}, \text{Tgt})$ is a $\dot{+}$ -conservative adaptation, with $\dot{+}$ defined by (2).

Properties. In this section, the consequences of the postulates of (pre-)IC merging operators are discussed from a Δ -combination of cases viewpoint.

(Δ -1) entails that $DK \cap \text{SolvedTgt} \subseteq DK \cap \text{Tgt}$ which is the property (1) required for the case-based inference (cf. section 3.3).

(Δ -2') entails that if the target case is consistent with the domain knowledge $DK \cap \text{Tgt} \neq \emptyset$ and each source case is bounded, –e.g., singleton cases– then $DK \cap \text{SolvedTgt}$ is satisfiable. (Δ -2) is stronger as it does not require the source cases to be bounded.

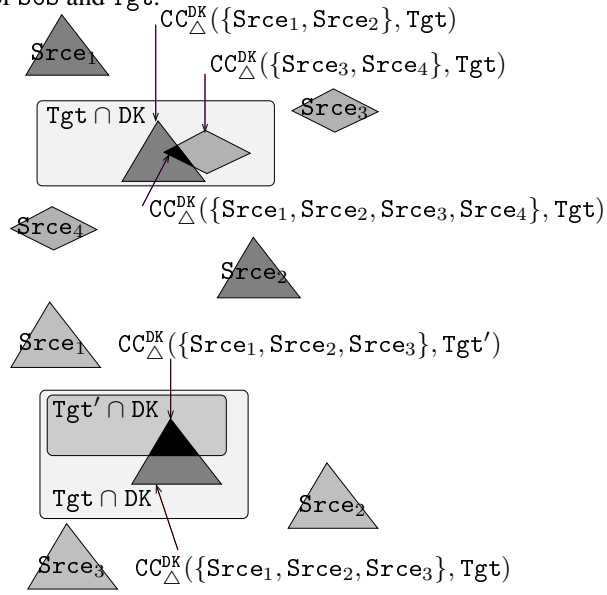
(Δ -3) entails that if the cases of SCS are consistent altogether with DK , then SolvedTgt is the conjunction of DK , Tgt , and every $\text{Srce} \in \text{SCS}$.

(Δ -4) enforces equity between the source cases: if two source cases are consistent with the target context, then either both of them are taken into account in the combination or none of them.

(Δ -5) to (Δ -8) characterize the maximal preservation of the source cases according to the local decomposition of SCS and Tgt .

(Δ -5) and (Δ -6) state that if the combination of two subsets of SCS provide consistent solutions, then the combination of the whole is the conjunction of both solutions.

(Δ -7) and (Δ -8) state that if Tgt is specialized into Tgt' that is consistent with the case combination for Tgt , then the case combination for Tgt' is obtained by conjunction with Tgt' .



5.3 Application to the example

Consider the merging operator defined in section 4.3 using the distance d defined in section 3.4. The values of dimensions \mathcal{U}_i for $i = 1, 2, 3, 5$ are fixed in $DK \cap \text{Tgt}$, so the space to be explored for the minima of d^Σ corresponds to $i = 4, 6, 7, 8, 9$.

From the structure of d , it can be shown that the minimum can be searched independently for dimensions \mathcal{U}_i with $i = 6, 7, 8$ as there is no constraint relating their values. Then the minima does not depend on the weight w_i , it is reached for:

$$x_6 = 25 \quad x_7 = 10 \quad 50 \leq x_8 \leq 65$$

$\mathcal{U}_4, \mathcal{U}_5$, and \mathcal{U}_9 are related through R_{vFroth} , the search must then follow this restriction: $x_4 \leq 200 \times x_5 + 1.32 \times x_9$. As $x_5 = 0$ it becomes: $x_4 \leq 1.32 \times x_9$.

As seen in section 3.4, the volume of froth should be given priority over the ways to making it. From the structure of d^Σ used in $\Delta^{d,\Sigma}$, it can be shown that the condition $w_4 > 3 \times (w_9 + w_5)$ is enough to ensure that priority (3 for the number of cases and w_9, w_5 because these are the dimensions in competition). Under this assumption, the minima is obtained for:

$$x_4 = 200 \quad x_9 = 165$$

This result matches with what was expected in section 2: the froth volume and chocolate mass are close to those of Srce_1 . The use of soya milk to generate froth is taken from Srce_2 with an adaptation according to the froth volume. Srce_3 had little influence, however its selection as source case offsets the absence of chocolate in Srce_2 .

6 Application to CCBI

6.1 Credible Case-Based Inference

Assumptions. Credible case-based inference (CCBI [4]) is an approach to CBR for which the problem-solution relation is assumed to be a partial function: if $x \rightsquigarrow X$ and $x \rightsquigarrow X'$ then $X = X'$. Moreover, each source case is a singleton $\text{Srce}_i = \{(x_i, X_i)\}$ and the target case is specified by a singleton target problem: $\text{Tgt} = \{x_0\} \times \mathcal{U}_{\text{so1}}$.

CCBI is based on the idea that the CBR principle ‘‘Similar problems have similar solutions’’ can be modeled thanks to d_{pb} , a symmetrical pseudo-distance on \mathcal{U}_{pb} , d_{so1} , a symmetrical pseudo-distance on \mathcal{U}_{so1} , and h , a *similarity profile*, i.e., a function $h : [0; +\infty] \rightarrow [0; +\infty]$ such that for *most* $x, y \in \mathcal{U}_{\text{pb}}$, if $x \rightsquigarrow X$ and $y \rightsquigarrow Y$ then

$$d_{\text{so1}}(X, Y) \leq h(d_{\text{pb}}(x, y)) \quad (4)$$

Thus, the similarity between solutions is constrained by the similarity between problems.² A way to learn h from the case base is also described in [4] and it is proven, under technical assumptions, that the probability of having the constraint (4) violated converges to 0 as the size of the case base grows.

Definition of CCBI. Given a set SCS of source cases $\text{Srce}_i = \{(x_i, X_i)\}$ and a target problem x_0 , if $x = x_i$ and $y = x_0$ satisfies (4) for any i , then the solution X_0 of x_0 satisfies $d_{\text{so1}}(X_i, X_0) \leq h(d_{\text{pb}}(x_i, x_0))$. In other words (Fig. 3):

$$X_0 \in \mathcal{C}_{\text{CCBI}} = \bigcap_i \mathcal{B}_{h(d_{\text{pb}}(x_i, x_0))}^{\mathcal{U}_{\text{so1}}}(X_i) \quad (5)$$

Therefore, $\text{SolvedTgt} = \{x_0\} \times \mathcal{C}_{\text{CCBI}}$ solves Tgt .

This inference is only credible and not certain since (4) is only satisfied for most $x, y \in \mathcal{U}_{\text{pb}}$.

² In fact, CCBI is introduced in [4] thanks to similarity measures \mathcal{S}_{pb} and \mathcal{S}_{so1} on \mathcal{U}_{pb} and \mathcal{U}_{so1} , but the definition presented in the current paper is equivalent. Indeed, a similarity measure \mathcal{S} on \mathcal{U} verifying $\mathcal{S}(x, y) = 1$ iff $x = y$ can be defined thanks to a pseudo-distance d on \mathcal{U} by $\mathcal{S}(x, y) = \frac{1}{1+d(x,y)}$ and vice-versa.

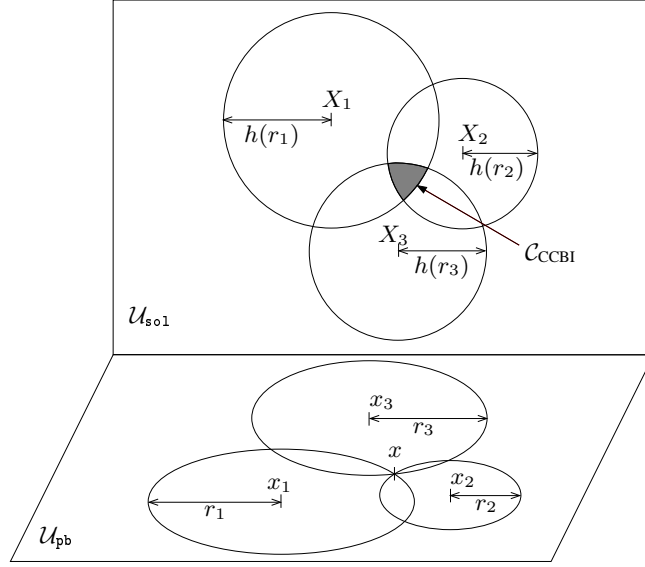


Fig. 3. Credible Case-Based Inference.

6.2 $\Delta^{d,\Sigma}$ -combination of cases extends CCBI

Proposition 2. *CCBI assumption about the case base is made (cf. section 6.1).*

Let d be the pseudo-distance on $\mathcal{U} = \mathcal{U}_{pb} \times \mathcal{U}_{so1}$ defined for $(x, X), (y, Y) \in \mathcal{U}$ by

$$d((x, X), (y, Y)) = \max\{h(d_{pb}(x, y)), d_{so1}(X, Y)\}$$

Let \mathcal{C}_{CCBI} be the result of CCBI, and $\mathcal{C}_{\Delta^{d,\Sigma}}$ be the result of the $\Delta^{d,\Sigma}$ -combination of cases without domain knowledge ($DK = \mathcal{U}$):

$$\mathcal{C}_{CCBI} = \bigcap_{i=1}^n \mathcal{B}_{h(d_{pb}(x_i, x_0))}^{\mathcal{U}_{so1}}(X_i)$$

$$\mathcal{C}_{\Delta^{d,\Sigma}} = \Phi_{x_0} \left(\mathcal{C}_{\Delta^{d,\Sigma}}^{\mathcal{U}}(SCS, Tgt) \right)$$

If CCBI provides a consistent result $-\mathcal{C}_{CCBI} \neq \emptyset-$ then it coincides with the $\Delta^{d,\Sigma}$ -case combination:

$$\mathcal{C}_{CCBI} = \mathcal{C}_{\Delta^{d,\Sigma}}$$

A proof for this proposition is given in appendix.

7 Computing IC merging in numerical spaces

The computation of the merging $\Delta_{IC}^{d,\Sigma}(M)$ is considered in this section with the assumptions:

- $\mathcal{U} = D_1 \times \dots \times D_n$ with D_i an interval of \mathbb{Z} or of \mathbb{R} .
- For $x, y \in \mathcal{U}$ $d(x, y) = \sum_{i=1}^n w_i |y_i - x_i|$.
- IC can be defined by a finite set of linear inequalities.
- $M = \{A_1, \dots, A_p\}$ and every A_j can also be defined by a finite set of linear inequalities.

The computation of $\Delta_{IC}^{d, \Sigma}(M)$ is equivalent to the minimization of the function $y \mapsto d^\Sigma(M, y)$ under the constraint $y \in IC$.

Proposition 3. *The computation of $\Delta_{IC}^{d, \Sigma}(M)$ is reducible to a linear programming problem [7].*

Sketch of proof. Minimizing $d^\Sigma(M, y)$ is reducible to minimizing $\sum_{j=1}^p \sum_{i=1}^n w_i |y_i - x_i^j|$ under the constraints $x^j \in A_j$. It is itself reducible to minimizing $\sum_{j=1}^p \sum_{i=1}^n w_i z_i^j$ with the additional constraints: $z_i^j \geq y_i - x_i^j$ and $z_i^j \geq x_i^j - y_i$, which is a linear programming problem. \square

This property shows that if every D_i is an interval of \mathbb{R} , i.e. it is reducible to a real linear programming problem, then the computation cost is polynomial in $n \times p$. If any D_i is a subset of \mathbb{Z} it is a mixed integer linear programming (which is an NP-hard problem).

In particular, the running example of this paper has been computed this way (cf. section 5.3): the boolean dimensions are replaced by the integer interval $[0, 1]_{\mathbb{Z}} = \{0, 1\}$.

8 Conclusion, Related work, and Future work

The main contribution of this paper is to define an approach to case combination based on an IC belief merging operator. It can be applied to case adaptation as a particular case combination. It is shown to extend credible case-based inference. This approach is, *a priori*, applicable to any formalism on which a pre-IC merging operator can be defined, eg. a pseudo-metric space. Provided that cases are represented by numerical attributes and that the constraints and the distance are linear, belief merging can be reduced to linear programming. The complexity is then polynomial if there are only real value attributes (linear programming) and NP-hard otherwise (mixed integer linear programming). In propositional logic, IC merging and thus Δ -combination is NP-hard, see [3].

Ongoing works are the implementation of a case combination based on an IC merging operator as defined in section 7, with the purpose of experimentation. The possibility to reduce other merging operators to linear programming should be investigated too.

As future work a systematic comparison with other case combination approaches should be performed. The convergence of these approaches should be investigated to

determine in particular which ones can be covered by an IC merging operator. The following criteria –inspired by the case combination approaches review given in [8]– can guide the comparison: how is structured the participation of each source case, whether the source cases are reused simultaneously or iteratively, and how the consistency is maintained.

Different ways of structuring the combination exist. Static structures as in Decentralized CBR (DzCBR) [9] where a set of contexts (or viewpoints) is set for the system. Every context generates a local solution according to its local domain knowledge and adaptation knowledge. Structure contained in cases as in DÉJÀ VU where the problem solving episodes are decomposed into a hierarchy of cases from the most abstract one that gives the main frame of the solution to the most concrete ones that solve sub-problems. Coverage of the target case by a set of source cases as in IDIOM [10] and COMPOSER [11] –a source case represent a partial solution with constraints for its inclusion in a global solution. In the approach presented in this paper all the source cases are equally considered, no explicit structure appears.

While COMPOSER, DzCBR, and the approach presented in this paper reuse the cases simultaneously, DÉJÀ VU and IDIOM do it iteratively. In DÉJÀ VU the resolution of a new query starts from the reuse of an abstract source case and is iterated on the resulting subproblems. In IDIOM a solution is built by iteratively incorporating source cases. A further investigation could be to investigate the possibility to express this approach to an iteration of conservative adaptation³ and to relate it to a combination based on an IC merging operator.

Finally the approaches can be distinguished by the way the consistency is maintained. In DzCBR bridge rules between the contexts enforce the coherence of the local solutions altogether to form a global solution. IDIOM and COMPOSER use a conflict resolution algorithm. In our approach the inconsistencies between cases are managed by an IC merging operator. This motivates the investigation of relationships between conflict resolution and IC merging.

Appendix

Proof of property 1

(Δ -1) The satisfaction of (Δ -1) is straightforward from the definition of $\Delta_{IC}^{d,\Sigma}(M)$.

(Δ -3) If $\bigcap M \cap IC \neq \emptyset$, let y be an element of $\bigcap M \cap IC$, $y \in A$ for any $A \in M$, thus $d(A, y) = 0$ and $d^\Sigma(M, y) = 0$. So $d^\Sigma(M, IC) = 0$ and $y \in \Delta_{IC}^{d,\Sigma}(M)$.

On the other way round, if $y \in \Delta_{IC}^{d,\Sigma}(M)$, then $d^\Sigma(M, y) = 0$ and $d(A, y) = 0$ for any $A \in M$ ($d(A, y) \geq 0$ for any A). The discreteness assumption of (\mathcal{U}, d) implies then that $y \in A$. Indeed, consider the set $\{x \in A \mid d(x, y) \leq 1\} \subseteq \mathcal{B}_1^{\mathcal{U}}(y)$, it is finite and since $d(A, y) < 1$ it is not empty, thus $d(A, y)$ is reached for an $x \in A$. $d(x, y) = 0$ which implies by the separation assumption that $y \in A$. Finally, $y \in \bigcap M$ and by definition of $\Delta_{IC}^{d,\Sigma}(M)$ $y \in IC$ which entails the result.

³ I.e.. first computing $\text{SolvedTgt}_1 = \text{ctxt}(\text{Srce}_1) \dot{+} \text{ctxt}(\text{Tgt})$ and then iteratively $\text{SolvedTgt}_{i+1} = \text{ctxt}(\text{Srce}_i) \dot{+} \text{ctxt}(\text{SolvedTgt}_i)$.

(Δ -5) and (Δ -6) Their satisfaction is obvious when $\Delta_{IC}^{d,\Sigma}(M_1) \cap \Delta_{IC}^{d,\Sigma}(M_2) = \emptyset$.
 In the contrary, let y be an element of $\Delta_{IC}^{d,\Sigma}(M_1) \cap \Delta_{IC}^{d,\Sigma}(M_2)$,

$$\begin{aligned} d^\Sigma(M_1, IC) + d^\Sigma(M_2, IC) &= d^\Sigma(M_1, y) + d^\Sigma(M_2, y) \\ &\geq \inf_{z \in IC} (d^\Sigma(M_1, z) + d^\Sigma(M_2, z)) \geq d^\Sigma(M_1 \cup M_2, IC) \end{aligned}$$

However

$$\begin{aligned} d^\Sigma(M_1, IC) + d^\Sigma(M_2, IC) &= \inf_{z \in IC} d^\Sigma(M_1, z) + \inf_{z \in IC} d^\Sigma(M_2, z) \\ &\leq \inf_{z \in IC} (d^\Sigma(M_1, z) + d^\Sigma(M_2, z)) \leq d^\Sigma(M_1 \cup M_2, IC) \end{aligned}$$

Thus all the inequalities can be replace by equalities, in particular the lower bound of $d^\Sigma(M_1, z) + d^\Sigma(M_2, z)$ for $z \in IC$ is $d^\Sigma(M_1, y) + d^\Sigma(M_2, y) = d^\Sigma(M_1 \cup M_2, IC)$, and as for $i = 1, 2$ $d^\Sigma(M_i, z) \geq d^\Sigma(M_i, y)$ this lower bound is reached when $d^\Sigma(M_i, z) = d^\Sigma(M_i, y) = d^\Sigma(M_i, IC)$, ie. $z \in \Delta_{IC}^{d,\Sigma}(M_1 \cup M_2)$ iff $z \in \Delta_{IC}^{d,\Sigma}(M_1) \cap \Delta_{IC}^{d,\Sigma}(M_2)$.

(Δ -7) and (Δ -8) Similarly, if $\Delta_{IC_1}^{d,\Sigma}(M) \cap IC_2 = \emptyset$, the result is obvious.

Otherwise, let y be an element of $\Delta_{IC_1}^{d,\Sigma}(M) \cap IC_2$.

$$\begin{aligned} d^\Sigma(M, IC_1) &= \inf_{z \in IC_1} d^\Sigma(M, z) \leq \inf_{z \in IC_1 \cap IC_2} d^\Sigma(M, z) = d^\Sigma(M, IC_1 \cap IC_2) \\ d^\Sigma(M, y) &= d^\Sigma(M, IC_1) \leq d^\Sigma(M, IC_1 \cap IC_2) \leq d^\Sigma(M, y) \end{aligned}$$

Thus $d^\Sigma(M, IC_1) = d^\Sigma(M, IC_1 \cap IC_2)$ and $\Delta_{IC_1}^{d,\Sigma}(M) \cap IC_2 = \Delta_{IC_1 \cap IC_2}^{d,\Sigma}(M)$.

(Δ -2') Assume $IC \neq \emptyset$ and every set in M is bounded.

If $M = \emptyset$ then $\bigcap M = \mathcal{U}$ and according to postulate (Δ -3), $\Delta_{IC}^{d,\Sigma}(M) = IC \neq \emptyset$.
 If $M \neq \emptyset$, then there exists an $A \in M$, $A \neq \emptyset$ and is bounded so finite. Let $r = d^\Sigma(M, IC) + 1$ and consider $S = \{y \in IC \mid d^\Sigma(M, y) \leq r\}$. S is finite, indeed $S \subseteq \{y \in \mathcal{U} \mid d^\Sigma(A, y) \leq r\} = \bigcup_{a \in A} \mathcal{B}_r^{\mathcal{U}}(a)$ which is a finite disjunction of finite sets. Moreover $S \neq \emptyset$ as $IC \neq \emptyset$ and the lower bound of $x \mapsto d^\Sigma(x, M)$ on IC and S are equal. S being finite this lower bound is reached and $\Delta_{IC}^{d,\Sigma}(M) \neq \emptyset$.

(Δ -4) Assume d is symmetrical and consider three sets IC , $A_1 \subseteq IC$ and $A_2 \subseteq IC$.
 If $\Delta_{IC}^{d,\Sigma}(\{A_1, A_2\}) \cap A_1 \neq \emptyset$, let y_1 be an element of this set.

$$d^\Sigma(\{A_1, A_2\}, IC) = d^\Sigma(\{A_1, A_2\}, y_1) = d(A_1, y_1) + d(A_2, y_1)$$

as $y_1 \in A_1$, $d(A_1, y_1) = 0$ thus $d^\Sigma(\{A_1, A_2\}, IC) = d(A_2, y_1)$. From the discretion assumption, as $A_2 \neq \emptyset$ there is $y_2 \in A_2$ such that $d(y_1, y_2) = d(y_2, y_1) = d(A_2, y_1)$.

$$d^\Sigma(\{A_1, A_2\}, IC) = d(y_1, y_2) \geq d(A_1, y_2) = d^\Sigma(\{A_1, A_2\}, y_2)$$

As $y_2 \in IC$ $d^\Sigma(\{A_1, A_2\}, y_2) \geq d^\Sigma(\{A_1, A_2\}, IC)$ thus, there is equality and $y_2 \in \Delta_{IC}^{d,\Sigma}(\{A_1, A_2\})$ which entails that $\Delta_{IC}^{d,\Sigma}(\{A_1, A_2\}) \cap A_2 \neq \emptyset$.

The other implication is symmetrical. \square

Proof of property 2

For $X \in \mathcal{C}_{\text{CCBI}}$, $d_{\text{so1}}(X_i, X) \leq h(d_{\text{pb}}(x_i, x_0))$ for any $1 \leq i \leq n$, so

$$d((x_i, X_i), (x_0, X)) = h(d_{\text{pb}}(x_i, x_0)) \leq \min_{Y \in \mathcal{U}_{\text{so1}}} d((x_i, X_i), (x_0, Y))$$

and

$$\begin{aligned} d^{\Sigma}(\text{CB}, (x_0, X)) &= \sum_{i=1}^n d((x_i, X_i), (x_0, X)) \leq \sum_{i=1}^n \left(\min_{Y \in \mathcal{U}_{\text{so1}}} d((x_i, X_i), (x_0, Y)) \right) \\ &\leq \min_{Y \in \mathcal{U}_{\text{so1}}} \left(\sum_{i=1}^n d((x_i, X_i), (x_0, Y)) \right) = \min_{(x_0, Y) \in \mathcal{U}} d^{\Sigma}(\text{CB}, (x_0, Y)) \end{aligned}$$

Thus, $(x_0, X) \in \mathcal{CC}_{\Delta^d, \Sigma}^{\mathcal{U}}(\text{CB}, \text{Tgt})$ and $X \in \mathcal{C}_{\Delta^d, \Sigma}$, which shows that $\mathcal{C}_{\text{CCBI}} \subseteq \mathcal{C}_{\Delta^d, \Sigma}$.

Moreover, if $\mathcal{C}_{\text{CCBI}} \neq \emptyset$, (ie. there is such an X), then $\min_{Y \in \mathcal{U}_{\text{so1}}} d^{\Sigma}(\text{CB}, (x_0, Y)) \leq d^{\Sigma}(\text{CB}, (x_0, X))$ and the previous inequalities are equalities. Thus, if $Y \in \mathcal{U}_{\text{so1}}$ minimizes $d^{\Sigma}(\text{CB}, (x_0, Y))$, then $\sum_{i=1}^n d((x_i, X_i), (x_0, Y)) = \sum_{i=1}^n d((x_i, X_i), (x_0, X))$. As for any $1 \leq i \leq n$ $d((x_i, X_i), (x_0, Y)) \geq d((x_i, X_i), (x_0, X))$, this means that

$$d((x_i, X_i), (x_0, Y)) = d((x_i, X_i), (x_0, X)) = h(d_{\text{pb}}(x_i, x_0))$$

ie. $Y \in \mathcal{C}_{\text{CCBI}}$, which shows that $\mathcal{C}_{\Delta^d, \Sigma} \subseteq \mathcal{C}_{\text{CCBI}}$. □

References

1. Riesbeck, C.K., Schank, R.C.: Inside Case-Based Reasoning. Lawrence Erlbaum Associates, Inc., Hillsdale, New Jersey (1989)
2. Cojan, J., Lieber, J.: Conservative adaptation in metric spaces. In: Proc. of ECCBR-08. (2008) 135–149
3. Konieczny, S., Lang, J., Marquis, P.: DA² merging operators. Artificial Intelligence **157**(1-2) (2004) 49–79
4. Hüllermeier, E.: Credible Case-Based Inference Using Similarity Profiles. IEEE Transaction on Knowledge and Data Engineering **19**(6) (2007) 847–858
5. Alchourrón, C.E., Gärdenfors, P., Makinson, D.: On the logic of theory change: partial meet functions for contraction and revision. J. of Symbolic Logic **50** (1985) 510–530
6. Katsuno, H., Mendelzon, A.: Propositional knowledge base revision and minimal change. Artificial Intelligence **52**(3) (1991) 263–294
7. Karmarkar, N.: A new polynomial-time algorithm for linear programming. Combinatorica **4**(4) (1984) 373–396
8. Gebhardt, F., Voß, A., Gräther, W., Schmidt-Belz, B.: Reasoning with complex cases. Kluwer, Boston (1997)
9. d’Aquin, M., Lieber, J., Napoli, A.: Decentralized case-based reasoning for the semantic web. In: International Semantic Web Conference. (2005) 142–155
10. Smith, I., Lottaz, C., Faltings, B.: Spatial composition using cases: IDIOM. In: ICCBR-95 Proceedings. 88–97
11. Purvis, L., Pu, P.: An Approach to Case Combination. In Voß, A., ed.: Proc. of the ECAI’96 Workshop: Adaptation in Case-Based Reasoning. (1996) 43–46