Context-Aware Distance Semantics for Inconsistent Database Systems

Anna Zamansky[1], Ofer Arieli[2], and Kostas Stefanidis[3]

[1] Department of Information Systems, University of Haifa, Israel
annazam@is.haifa.ac.il
[2] School of Computer Science, The Academic College of Tel-Aviv, Israel
oarieli@mta.ac.il
[3] Institute of Computer Science, Foundation for Research and Technology
Hellas (FORTH), Greece
kstef@ics.forth.gr

Abstract. Many approaches for consistency restoration in database systems have to deal with the problem of an exponential blowup in the number of possible repairs. For this reason, recent approaches advocate more flexible and fine grained policies based on the reasoner's preference. In this paper we take a further step towards more personalized inconsistency management by incorporating ideas from context-aware systems. The outcome is a general distance-based approach to inconsistency maintenance in database systems, controlled by context-aware considerations.

1 Introduction

Inconsistency handling in constrained databases is a primary issue in the context of consistent query answering, data integration, and data exchange. The standard approaches to this issue are usually based on the principle of minimal change, aspiring to achieve consistency via a minimal amount of data modifications (see, e.g., [2, 7, 10]). A key question in this respect is how to *choose* among the different possibilities of restoring the consistency of a database (i.e., 'repairing' it).

Earlier approaches to inconsistency management were based on the assumption that there should be some fixed, pre-determined way of repairing a database. Recently, there has been a paradigm shift towards user-controlled inconsistency management policies. Works taking this approach provide a possibility for the user to express some *preference* over all possible database repairs, preferring certain repairs to others (see [18] for a survey and further references). While such approaches provide the user with flexibility and control over inconsistency management, in reality they entail a considerable technical burden on the user's shoulders of calibrating, updating and maintaining preferences or policies. Moreover, in many cases these preferences may be *dynamic*, changing quickly on the go (e.g., depending on the user's geographical location). In the era of ubiquitous computing, users want *easy* – and sometimes even *fully automatic* – inconsistency management solutions with little cognitive load, while still expecting them to be *personalized* to their particular needs. This leads to the idea of introducing *context-awareness* into inconsistency management.

Context-awareness is defined as the use of contexts to provide task-relevant information and services to a user (see [1]). We believe that inconsistency management has natural relations to the concept of context. To capture this idea, we incorporate notions and techniques that have been studied by the context-aware computing community to consistency management for database systems, by combining the following two ingredients:

- *Distance-based semantics* for restoring the consistency of inconsistent databases according to the principle of minimal change, and
- *Context-awareness considerations* for incorporating user preferences.

Example 1. Let us consider the following simple database instance:

empNum	**name**	**address**	**salary**
1	John	Tower Street 3, London, UK	70K$
1	John	Herminengasse 8, Wien, AT	80K$
2	Mary	42 Street 15, New York, US	90K$

Two functional dependencies that may be violated here are empNum → address and empNum → salary. Thus, a database with the above relation and integrity constraints is not consistent. Minimal change considerations (which will be expressed in what follows by distance functions) imply that it is enough to delete either the first or the second tuple for restoring consistency. Now, the decision which tuple to delete may be *context-dependent*. For instance, for tax assessments tuples with higher salaries may be preferred, while tuples with lower salaries may have higher priority when loans or grants are considered. The choice between the first two tuples may also be determined by more dynamic considerations, such as geographic locations, etc.

2 Inconsistent Databases and Distance Semantics

For simplicity of presentation, in this paper we remain on the propositional level and reduce first-order databases to our framework by grounding them. In the sequel, $\mathcal{L}$ denotes a propositional language with a *finite* set of atomic formulas $\mathsf{Atoms}(\mathcal{L})$. An *$\mathcal{L}$-interpretation* I is an assignment of a truth value in $\{T, F\}$ to every element in $\mathsf{Atoms}(\mathcal{L})$. Interpretations are extended to complex formulas in $\mathcal{L}$ in the usual way, using the truth tables of the connectives in $\mathcal{L}$. The set of two-valued interpretations for $\mathcal{L}$ is denoted by $\Lambda_{\mathcal{L}}$. An interpretation I is a *model* of an $\mathcal{L}$-formula ψ if $I(\psi) = T$, denoted by $I \models \psi$, and it is a model of a set Γ of $\mathcal{L}$-formulas, denoted by $I \models \Gamma$, if it is a model of every $\mathcal{L}$-formula in Γ. The set of models of Γ is denoted by $mod(\Gamma)$. We say that Γ is *satisfiable* if $mod(\Gamma)$ is not empty.

Definition 1. A *database* $\mathcal{DB}$ in $\mathcal{L}$ is a pair $\langle \mathcal{D}, \mathcal{IC} \rangle$, where $\mathcal{D}$ (the *database instance*) is a finite subset of $\mathsf{Atoms}(\mathcal{L})$, and $\mathcal{IC}$ (the *integrity constraints*) is a finite and consistent set of $\mathcal{L}$-formulas.

The meaning of $\mathcal{D}$ is determined by the conjunction of its facts, augmented with Reiter's *closed world assumption*, stating that each atomic formula that does not appear in $\mathcal{D}$ is false: $\mathsf{CWA}(\mathcal{D}) = \{\neg p \mid p \notin \mathcal{D}\}$. Henceforth, a database $\mathcal{DB} = \langle \mathcal{D}, \mathcal{IC} \rangle$ will be associated with the theory $\Gamma_{\mathcal{DB}} = \mathcal{IC} \cup \mathcal{D} \cup \mathsf{CWA}(\mathcal{D})$.

Definition 2. A database $\mathcal{DB}$ is *consistent* iff $\Gamma_{\mathcal{DB}}$ is satisfiable.

When a database is not consistent at least one integrity constraint is violated, and so it is usually required to look for "repairs" of the database, that is, changes of the database instance so that its consistency will be restored. There are numerous approaches for doing so (see, e.g., [2, 7, 10] for some surveys on this subject). Here we follow the distance-based approach described in [3, 5], which we find suitable for our purposes since it provides a modular and flexible framework for a variety of methods of repair and consistent query answering. In the context of database systems this approach aims at addressing the problem that when $\mathcal{DB}$ is inconsistent $mod(\Gamma_{\mathcal{DB}})$ is empty, so reasoning with $\mathcal{DB}$ is trivialized. This may be handled by replacing $mod(\Gamma_{\mathcal{DB}})$ with the set $\Delta(\mathcal{DB})$ of interpretations that, intuitively, are 'as close as possible' to (satisfying) $\mathcal{DB}$, while still satisfying the integrity constraints. When $\mathcal{DB}$ is consistent, $\Delta(\mathcal{DB})$ and $mod(\Gamma_{\mathcal{DB}})$ coincide (see Proposition 3 below), which assures that distance-based semantics is a conservative generalization of standard semantics for consistent databases.

In what follows, we recall the relevant definitions for formalizing the intuition above (see also [3, 5]).

Definition 3. A *pseudo-distance* on a set U is a total function $d : U \times U \to \mathbb{R}^+$, which is symmetric (for all $\nu, \mu \in U$, $d(\nu, \mu) = d(\mu, \nu)$) and preserves identity (for all $\nu, \mu \in U$, $d(\nu, \mu) = 0$ if and only if $\nu = \mu$). A pseudo-distance d is called a *distance* (*metric*) on U, if it satisfies the triangular inequality: for all $\nu, \mu, \sigma \in U$, $d(\nu, \sigma) \leq d(\nu, \mu) + d(\mu, \sigma)$.

Definition 4. A *(numeric) aggregation function* is a function f, whose domain consists of multisets of real numbers and whose range is the real numbers, satisfying the following properties:
1. f is non-decreasing when a multiset element is replaced by a larger element,
2. $f(\{x_1, \ldots, x_n\}) = 0$ if and only if $x_1 = x_2 = \ldots x_n = 0$, and
3. $f(\{x\}) = x$ for every $x \in \mathbb{R}$.
An aggregation function f is *hereditary*, if $f(\{x_1, \ldots, x_n\}) < f(\{y_1, \ldots, y_n\})$ entails that $f(\{x_1, \ldots, x_n, z_1, \ldots, z_m\}) < f(\{y_1, \ldots, y_n, z_1, \ldots, z_m\})$.

In what follows we shall aggregate distance values. Since distances are non-negative numbers, aggregation functions in this case include the summation and the maximum functions, the former is also hereditary.

Example 2. One may define the following distances on $\Lambda_{\mathcal{L}}$:

$$d_U(I, I') = \begin{cases} 1 & \text{if } I \neq I', \\ 0 & \text{otherwise.} \end{cases} \qquad d_H(I, I') = \mid \{p \in \mathsf{Atoms}(\mathcal{L}) \mid I(p) \neq I'(p)\} \mid .$$

d_U is sometimes called the uniform distance and d_H is known as the Hamming distance. More sophisticated distances are considered, e.g., in [5] and [12].

Definition 5. A *distance setting* (for a language $\mathcal{L}$) is a pair $\mathsf{DS} = \langle d, f \rangle$, where d is a pseudo-distance on $\Lambda_{\mathcal{L}}$ and f is an aggregation function.

The next definition is a common way of using distance functions for maintaining inconsistent data (see, e.g, [15, 16]).

Definition 6. For a finite set $\Gamma = \{\psi_1, \ldots, \psi_n\}$ of formulas in $\mathcal{L}$, an interpretation $I \in \Lambda_{\mathcal{L}}$, and a distance setting $\mathsf{DS} = \langle d, f \rangle$ for $\mathcal{L}$, we denote: $d_{\mathsf{DS}}(I, \psi_i) = \min\{d(I, I') \mid I' \models \psi_i\}$ and $\delta_{\mathsf{DS}}(I, \Gamma) = f(\{d_{\mathsf{DS}}(I, \psi_1), \ldots, d_{\mathsf{DS}}(I, \psi_n)\})$.

Proposition 1. [3, 16] *For every interpretation $I \in \Lambda_{\mathcal{L}}$ and a distance setting $\mathsf{DS} = \langle d, f \rangle$, it holds that $I \models \psi$ iff $d_{\mathsf{DS}}(I, \psi) = 0$ and $I \models \Gamma$ iff $\delta_{\mathsf{DS}}(I, \Gamma) = 0$.*

Definition 7. Given a database $\mathcal{DB} = \langle \mathcal{D}, \mathcal{IC} \rangle$ in $\mathcal{L}$ and a distance setting $\mathsf{DS} = \langle d, f \rangle$ for $\mathcal{L}$, the set of *the most plausible interpretations of* $\mathcal{DB}$ (with respect to DS) is defined as follows:

$$\begin{aligned}\Delta_{\mathsf{DS}}(\mathcal{DB}) = \big\{ I \in \mathsf{mod}(\mathcal{IC}) \mid I' \in \mathsf{mod}(\mathcal{IC}) \Longrightarrow \\ \delta_{\mathsf{DS}}(I, \mathcal{D} \cup \mathsf{CWA}(\mathcal{D})) \leq \delta_{\mathsf{DS}}(I', \mathcal{D} \cup \mathsf{CWA}(\mathcal{D})) \big\}.\end{aligned}$$

Note 1. Since $\mathcal{IC}$ is satisfiable, for every database $\mathcal{DB} = \langle \mathcal{D}, \mathcal{IC} \rangle$ and a distance setting DS for its language, it holds that $\Delta_{\mathsf{DS}}(\mathcal{DB}) \neq \emptyset$.

Definition 8. Let $\mathcal{DB} = \langle \mathcal{D}, \mathcal{IC} \rangle$ be a database and $\mathsf{DS} = \langle d, f \rangle$ a distance setting. We say that $\mathcal{R}$ is a DS-*repair* of $\mathcal{DB}$, if there is an $I \in \Delta_{\mathsf{DS}}(\mathcal{DB})$ such that $\mathcal{R} = \{p \in \mathsf{Atoms}(\mathcal{L}) \mid I(p) = T\}$. We shall sometimes denote this repair by $\mathcal{R}(I)$ and say that it is *induced by* I (or that I is the *characteristic model* of $\mathcal{R}$). The set of all the DS-repairs is denoted by $\mathsf{Repairs}_{\mathsf{DS}}(\mathcal{DB}) = \{\mathcal{R}(I) \mid I \in \Delta_{\mathsf{DS}}(\mathcal{DB})\}$.

An alternative characterization of the DS-repairs of $\mathcal{DB}$ is given next:

Proposition 2. *Let $\mathcal{DB} = \langle \mathcal{D}, \mathcal{IC} \rangle$ be a database and $\mathsf{DS} = \langle d, f \rangle$ a distance setting. Let I_S be the characteristic function of $S \subseteq \mathsf{Atoms}(\mathcal{L})$ (that is, $I_S(p) = T$ if $p \in S$ and $I_S(p) = F$ otherwise). The* DS-inconsistency value *of S is:*

$$\mathsf{Inc}_{\mathsf{DS}}(S) = \begin{cases} \delta_{\mathsf{DS}}(I_S, \mathcal{D} \cup \mathsf{CWA}(\mathcal{D})) & \text{if } I_S \in mod(\mathcal{IC}), \\ \infty & \text{otherwise.} \end{cases}$$

Then $\mathcal{R} \subseteq \mathsf{Atoms}(\mathcal{L})$ is a DS-repair of $\mathcal{DB}$ iff its DS-inconsistency value is minimal among the DS-inconsistency values of the subsets of $\mathsf{Atoms}(\mathcal{L})$.

Proof. Let $\mathcal{R} \subseteq \mathsf{Atoms}(\mathcal{L})$ such that $\mathsf{Inc}_{\mathsf{DS}}(\mathcal{R}) \leq \mathsf{Inc}_{\mathsf{DS}}(S)$ for every $S \subseteq \mathsf{Atoms}(\mathcal{L})$. Since $\mathcal{IC}$ is satisfiable, $\mathsf{Inc}_{\mathsf{DS}}(\mathcal{R}) < \infty$, and so $I_{\mathcal{R}} \in mod(\mathcal{IC})$. Let now $\mathcal{R}'$ be a DS-repair of $\mathcal{DB}$. Then there is an element $I' \in \Delta_{\mathsf{DS}}(\mathcal{DB})$ such that $\mathcal{R}' = \{p \in \mathsf{Atoms}(\mathcal{L}) \mid I'(p) = T\}$. But $\delta_{\mathsf{DS}}(I_{\mathcal{R}}, \mathcal{D} \cup \mathsf{CWA}(\mathcal{D})) \leq \delta_{\mathsf{DS}}(I', \mathcal{D} \cup \mathsf{CWA}(\mathcal{D}))$, and so $I_{\mathcal{R}} \in \Delta_{\mathsf{DS}}(\mathcal{DB})$ as well, which implies that $\mathcal{R}$ is a DS-repair of $\mathcal{DB}$.

For the converse, let $\mathcal{R}$ be a DS-repair of $\mathcal{DB}$ and let $S \subseteq \mathsf{Atoms}(\mathcal{L})$. We have to show that $\mathsf{Inc}_{\mathsf{DS}}(\mathcal{R}) \leq \mathsf{Inc}_{\mathsf{DS}}(S)$. Indeed, if $I_S \notin mod(\mathcal{IC})$ then $\mathsf{Inc}_{\mathsf{DS}}(S) = \infty$ and so the claim is obtained. Otherwise, both $I_{\mathcal{R}}$ and I_S are models of $\mathcal{IC}$, and since $\mathcal{R}$ is a DS-repair of $\mathcal{DB}$, $I_{\mathcal{R}} \in \Delta_{\mathsf{DS}}(\mathcal{DB})$. It follows that $\delta_{\mathsf{DS}}(I_{\mathcal{R}}, \mathcal{D} \cup \mathsf{CWA}(\mathcal{D})) \leq \delta_{\mathsf{DS}}(I_S, \mathcal{D} \cup \mathsf{CWA}(\mathcal{D}))$ and so $\mathsf{Inc}_{\mathsf{DS}}(\mathcal{R}) \leq \mathsf{Inc}_{\mathsf{DS}}(S)$. □

By Proposition 1 and Definition 8, we also have the following result:

Proposition 3. *Let $\mathcal{DB} = \langle \mathcal{D}, \mathcal{IC} \rangle$ be a database and* DS *a distance setting. The following conditions are equivalent: (1) $\mathcal{DB}$ is consistent, (2) $\Delta_{\mathsf{DS}}(\mathcal{DB}) = mod(\Gamma_{\mathcal{DB}})$, (3) $\mathsf{Repairs}_{\mathsf{DS}}(\mathcal{DB}) = \{\mathcal{D}\}$, (4) The* DS*-inconsistency value of every* DS*-repair of $\mathcal{DB}$ is zero.*

Example 3. Let us return to the database in Example 1. The projection of the database table on the attributes id and salary is: $\{\langle 1, \mathsf{70K\$}\rangle, \langle 1, \mathsf{80K\$})\rangle, \langle 2, \mathsf{90K\$}\rangle\}$. After grounding the database and representing the tuple $\langle \mathsf{empNum}, \mathsf{salary}\rangle$ by a propositional variable $\mathsf{T}^{\mathsf{empNum}}_{\mathsf{salary}}$, we have:

$$\mathcal{D} \cup \mathsf{CWA}(\mathcal{D}) = \left\{\mathsf{T}^1_{70K\$}, \mathsf{T}^1_{80K\$}, \neg\mathsf{T}^1_{90K\$}, \neg\mathsf{T}^2_{70K\$}, \neg\mathsf{T}^2_{80K\$}, \mathsf{T}^2_{90K\$}\right\},$$

and the functional dependency $\mathsf{empNum} \rightarrow \mathsf{salary}$ is formulated as follows:

$$\mathcal{IC} = \left\{\mathsf{T}^x_y \rightarrow \neg\mathsf{T}^x_z \mid y \neq z,\ y, z \in \{70K\$, 80K\$, 90K\$\},\ x \in \{1, 2\}\right\}.$$

Using the distance d_H from Example 2 and $f = \Sigma$, we compute:

I	$d_H(I, \mathsf{T}^1_{70})$	$d_H(I, \mathsf{T}^1_{80})$	$d_H(I, \neg\mathsf{T}^1_{90})$	$d_H(I, \neg\mathsf{T}^2_{70})$	$d_H(I, \neg\mathsf{T}^2_{80})$	$d_H(I, \mathsf{T}^2_{90})$	$\delta_{d_H,\Sigma}(I, \Gamma_{\mathcal{DB}})$
$\emptyset$	1	1	0	0	0	1	3
$\{\mathsf{T}^1_{70}\}$	0	1	0	0	0	1	2
$\{\mathsf{T}^1_{80}\}$	1	0	0	0	0	1	2
...	...	...	...	...	...	...	...
$\{\mathsf{T}^1_{70}, \mathsf{T}^2_{90}\}$	0	1	0	0	0	0	**1**
$\{\mathsf{T}^1_{80}, \mathsf{T}^2_{90}\}$	1	0	0	0	0	0	**1**
...	...	...	...	...	...	...	...

It follows that $\Delta_{\langle d_H, \Sigma\rangle}(\mathcal{DB}) = \{I_1, I_2\}$ and $\mathsf{Repairs}_{\mathsf{DS}}(\mathcal{DB}) = \{\mathcal{R}(I_1), \mathcal{R}(I_2)\}$, where $\mathcal{R}(I_1) = \{\mathsf{T}^1_{70}, \mathsf{T}^2_{90}\}$ and $\mathcal{R}(I_2) = \{\mathsf{T}^1_{80}, \mathsf{T}^2_{90}\}\}$. Thus, only T^2_{90} holds in all the repairs of $\mathcal{DB}$, that is, only the salary of employee 2 is certain.

3 Context-Aware Inconsistency Management

3.1 Context Modeling

As defined in [1], *"Context is any information that can be used to characterize the situation of an entity. An entity is a person, place or object that is considered relevant to the interaction between a user and an application, including the user and application"*. This notion has been found useful in several domains, such as machine learning and knowledge acquisition (see, e.g., [8, 9]). We shall consider as a context any data that can be used to characterize database-related situations, involving database entities, user contexts and preferences, etc. [11]. There is a wide variety of methods for modeling contexts. Here we follow the data-centric approach introduced in [20], and refer to contexts using a finite set of special purpose variables (which may not be part of the database).

Definition 9. A *context environment* (or just a *context*) C is a finite tuple of variables $\langle c_1, \ldots, c_n\rangle$, where each variable c_i ($1 \leq i \leq n$) has a corresponding range $\mathsf{Range}(c_i)$ of possible values. A *context state* for C (a C-state, for short) is an assignment S such that $S(c_i) \in \mathsf{Range}(c_i)$. The set of context states is denoted by $\mathsf{States}(C)$.

Intuitively, a context environment C represents the parameters that may be taken into consideration for the database inconsistency maintenance.

We are now ready to incorporate context-awareness into distance considerations. We do so by making the 'most plausible' interpretations in $\mathcal{DB}$, the elements in $\Delta_{\mathsf{DS}}(\mathcal{DB})$, *sensitive to context*, in the sense that more 'relevant' formulas have higher impact on the distance computations than less 'relevant' formulas. Thus, while we still strive to minimize change, the latter will be measured in a more subtle, context-aware way.

Definition 10. A *relevance ranking* for a set Γ of formulas and a context environment C, is a total function $R : \Gamma \times \mathsf{States}(C) \to (0, 1]$.

Given a set Γ and a context environment C, a relevance ranking function for Γ and C assigns to every formula $\psi \in \Gamma$ and every state S of C a (positive) *relevance factor* $R(\psi, S)$ indicating the relevance of ψ according to S. Intuitively, higher values of these factors correspond to higher relevance of their formulas, which makes changes to these formulas in computing database repairs less desirable.[4]

Definition 11. A *context setting* for a set of formulas Γ is a triple $\mathsf{CS}(\Gamma) = \langle C, S, R\rangle$, where C is a context environment, $S \in \mathsf{States}(C)$ is a C-state, and R is a relevance ranking function for Γ and C. In what follows we shall sometimes denote by $\mathsf{CS}(\mathcal{L})$ a context setting $\mathsf{CS}(\Gamma)$ in which Γ is the set of all the well-formed formulas of $\mathcal{L}$.

Consistency restoration for databases can now be defined as before (see Definitions 6 and 7), except that the underlying distance setting $\mathsf{DS} = \langle d, f\rangle$ should now be context-sensitive in the sense that d_{DS} preserves the order induced by ranking in the following way:

Definition 12. Let $\mathsf{CS}(\mathcal{L}) = \langle C, S, R\rangle$ be a context setting for a language $\mathcal{L}$. A distance setting $\mathsf{DS} = \langle d, f\rangle$ is called CS-*sensitive*, if for every two atomic formulas p_1 and p_2 such that $R(p_1, S) > R(p_2, S)$, it holds that $d_{\mathsf{DS}}(I_2, p_1) > d_{\mathsf{DS}}(I_1, p_2)$ for every $I_1 \in mod(p_1) \setminus mod(p_2)$ and $I_2 \in mod(p_2) \setminus mod(p_1)$.

Clearly, Proposition 3 holds also for context-sensitive distance settings.

Next, we demonstrate the effect of incorporating context sensitive distance settings on inconsistency management.

Proposition 4. *Let* $\mathcal{DB} = \langle \mathcal{D} \sqcup \{p_1, p_2\}, \mathcal{IC}\rangle$ *be a database*[5], $\mathsf{CS} = \langle C, S, R\rangle$ *a context setting and* $\mathsf{DS} = \langle d, f\rangle$ *a* CS*-sensitive distance setting in which* f *is hereditary. If* $R(p_1, S) > R(p_2, S)$, *for every* $\mathcal{D}' \subseteq \mathsf{Atoms}(\mathcal{L}) \setminus \{p_1, p_2\}$ *such that* $\mathcal{D}' \sqcup \{p_1\} \models \mathcal{IC}$, *the* DS*-inconsistency value of* $\mathcal{D}_1 = \mathcal{D}' \sqcup \{p_1\}$ *is smaller than the* DS*-inconsistency value of* $\mathcal{D}_2 = \mathcal{D}' \sqcup \{p_2\}$.

[4] Relevance factors may be thought of as a context-dependent interpretation of weights in prioritized theories (see, for example, [4]).

[5] We denote by $\mathcal{D} \sqcup \{p_1, p_2\}$ the disjoint union of $\mathcal{D}$ and $\{p_1, p_2\}$.

Proof. Let $\mathcal{D}' \subseteq \mathsf{Atoms}(\mathcal{L}) \setminus \{p_1, p_2\}$ and $\mathcal{D}_1 = \mathcal{D}' \cup \{p_1\}$. Since $\mathcal{D}_1 \models \mathcal{IC}$, we have that $\mathsf{Inc}_{\mathsf{DS}}(\mathcal{D}_1) < \infty$. Thus, $\mathsf{Inc}_{\mathsf{DS}}(\mathcal{D}_1) < \mathsf{Inc}_{\mathsf{DS}}(\mathcal{D}_2)$ whenever $\mathcal{D}_2 \not\models \mathcal{IC}$. Suppose then that $\mathcal{D}_2 \models \mathcal{IC}$ as well. In this case, in the notations of Proposition 2, we have that $I_{\mathcal{D}_1}$ and $I_{\mathcal{D}_2}$ differ only in the assignments for p_1 and p_2 (I.e., $I_{\mathcal{D}_1}$ satisfies p_1 and falsifies p_2 while $I_{\mathcal{D}_2}$ satisfies p_2 and falsifies p_1. Elsewhere, both interpretations are equal to $I_{\mathcal{D}'}$). Now, since DS is CS-sensitive, by the facts that (i) $R(p_1, S) > R(p_2, S)$, (ii) $I_{\mathcal{D}_1} \in mod(p_1) \setminus mod(p_2)$ and (iii) $I_{\mathcal{D}_2} \in mod(p_2) \setminus mod(p_1)$, we have that $d_{\mathsf{DS}}(I_{\mathcal{D}_1}, p_2) < d_{\mathsf{DS}}(I_{\mathcal{D}_2}, p_1)$. Let $\mathcal{D} \cup \mathsf{CWA}(\mathcal{D} \sqcup \{p_1, p_2\}) = \{\psi_1, \ldots, \psi_n\}$. By the assumption that f is hereditary,

$$
\begin{aligned}
&\mathsf{Inc}_{\mathsf{DS}}(\mathcal{D}_1) = \delta_{\mathsf{DS}}(I_{\mathcal{D}_1}, \mathcal{DB}) = \\
&\quad f(\{d_{\mathsf{DS}}(I_{\mathcal{D}_1}, \psi_1), \ldots, d_{\mathsf{DS}}(I_{\mathcal{D}_1}, \psi_n), d_{\mathsf{DS}}(I_{\mathcal{D}_1}, p_1), d_{\mathsf{DS}}(I_{\mathcal{D}_1}, p_2)\}) = \\
&\quad f(\{d_{\mathsf{DS}}(I_{\mathcal{D}_1}, \psi_1), \ldots, d_{\mathsf{DS}}(I_{\mathcal{D}_1}, \psi_n), 0, d_{\mathsf{DS}}(I_{\mathcal{D}_1}, p_2)\}) = \\
&\quad f(\{d_{\mathsf{DS}}(I_{\mathcal{D}_2}, \psi_1), \ldots, d_{\mathsf{DS}}(I_{\mathcal{D}_2}, \psi_n), 0, d_{\mathsf{DS}}(I_{\mathcal{D}_1}, p_2)\}) < \\
&\quad f(\{d_{\mathsf{DS}}(I_{\mathcal{D}_2}, \psi_1), \ldots, d_{\mathsf{DS}}(I_{\mathcal{D}_2}, \psi_n), 0, d_{\mathsf{DS}}(I_{\mathcal{D}_2}, p_1)\}) = \\
&\quad f(\{d_{\mathsf{DS}}(I_{\mathcal{D}_2}, \psi_1), \ldots, d_{\mathsf{DS}}(I_{\mathcal{D}_2}, \psi_n), d_{\mathsf{DS}}(I_{\mathcal{D}_2}, p_2), d_{\mathsf{DS}}(I_{\mathcal{D}_2}, p_1)\}) = \\
&\quad \delta_{\mathsf{DS}}(I_{\mathcal{D}_2}, \mathcal{DB}) = \mathsf{Inc}_{\mathsf{DS}}(\mathcal{D}_2). \qquad \square
\end{aligned}
$$

It follows that when context-sensitive distances are incorporated, "more relevant" formulas are preferred in the repairs. This is shown next.

Corollary 1. *Let $\mathcal{DB} = \langle \mathcal{D} \sqcup \{p_1, p_2\}, \mathcal{IC} \rangle$ be a database, $\mathsf{CS} = \langle C, S, R \rangle$ a context setting and $\mathsf{DS} = \langle d, f \rangle$ a CS-sensitive distance setting in which f is hereditary. If $\mathcal{DB}_1 = \langle \mathcal{D} \sqcup \{p_1\}, \mathcal{IC} \rangle$ is a consistent database, $R(p_1, S) > R(p_2, S)$, and $\mathcal{IC} \cup \{p_1, p_2\}$ is (classically) inconsistent, then no DS-repair of $\mathcal{DB}$ contains p_2.*

Corollary 1 can be generalized as follows:

Corollary 2. *Let $\mathcal{DB} = \langle \mathcal{D}, \mathcal{IC} \rangle$ be a database, $\mathsf{CS} = \langle C, S, R \rangle$ a context setting and $\mathsf{DS} = \langle d, f \rangle$ a CS-sensitive distance setting in which f is hereditary. Suppose that $\mathcal{D} = \mathcal{D}' \sqcup \mathcal{D}''$ can be partitioned to two disjoint nonempty subsets $\mathcal{D}'$ and $\mathcal{D}''$ such that (1): $\mathcal{DB}' = \langle \mathcal{D}', \mathcal{IC} \rangle$ is a consistent database, (2): $\forall p'' \in \mathcal{D}''$ $\exists p' \in \mathcal{D}'$ such that $\mathcal{IC} \cup \{p', p''\}$ is not consistent, and (3): $\forall p' \in \mathcal{D}'$ and $\forall p'' \in \mathcal{D}''$ it holds that $R(p', S) > R(p'', S)$. Then for every DS-repair $\mathcal{R}$ of $\mathcal{DB}$, $\mathcal{R} \cap \mathcal{D}'' = \emptyset$.*

3.2 A Simple Construction of Context-Sensitive Distance Settings

Below we provide a concrete method for defining context-sensitive distance settings and exemplify some of the properties of the settings that are obtained.

Definition 13. Let $\mathsf{CS}(\mathcal{L}) = \langle C, S, R \rangle$ be a context setting for $\mathcal{L}$ and let g be an aggregation function. The (pseudo) distance d_g^{CS} on $\Lambda_{\mathcal{L}}$ is defined as follows:

$$d_g^{\mathsf{CS}}(I, I') = g(\{R(p, S) \cdot |I(p) - I'(p)| \;\mid\; p \in \mathsf{Atoms}(\mathcal{L})\}).$$

It is easy to verify that for any CS and g, the function d_g^{CS} is a pseudo-distance on $\Lambda_{\mathcal{L}}$. In particular, for any context setting $\mathsf{CS}(\mathcal{L}) = \langle C, S, R \rangle$ where R is uniformly 1, d_{Σ}^{CS} coincides with the Hamming distance d_H in Example 2. The next proposition provides a general way of constructing context-sensitive distance settings, based on the functions in Definition 13.

Proposition 5. *Let* $\mathsf{CS} = \langle C, S, R\rangle$ *be a context setting and let* $\mathsf{DS} = \langle d_g^{\mathsf{CS}}, f\rangle$ *be a distance setting, where* g *is a hereditary. Then* DS *is* CS*-sensitive.*

Proof. Let p_1 and p_2 be atomic formulas such that $R(p_1, S) > R(p_2, S)$, and let $I_1 \in \mathit{mod}(p_1) \setminus \mathit{mod}(p_2)$ and $I_2 \in \mathit{mod}(p_2) \setminus \mathit{mod}(p_1)$. Below, we denote $g(\overline{0}, x) = g(\{0, \ldots, 0, x, 0, \ldots, 0\})$. By Definition 6,

$$\begin{aligned} d_{\mathsf{DS}}(I_1, p_2) &= \min\{d_g^{\mathsf{CS}}(I_1, J) \mid J \models p_2\} \\ &= \min\{g(\{R(p, S) \cdot |I_1(p) - J(p)| \mid p \in \mathsf{Atoms}(\mathcal{L})\}) \mid J \models p_2\}. \end{aligned}$$

Since g is hereditary, the minimum above must be obtained for a model J of p_2 that coincides with I_1 on every atom $p \neq p_2$. It follows, then, that $d_{\mathsf{DS}}(I_1, p_2) = g(\overline{0}, R(p_2, S))$. By similar considerations, $d_{\mathsf{DS}}(I_2, p_1) = g(\overline{0}, R(p_1, S))$. Now, since $R(p_1, S) > R(p_2, S)$ and since g is hereditary, $d_{\mathsf{DS}}(I_2, p_1) > d_{\mathsf{DS}}(I_1, p_2)$. □

The next proposition demonstrates how CS-sensitive distance settings of the form defined above give precedence to "more relevant" facts.

Proposition 6. *Let* $\mathsf{CS} = \langle C, S, R\rangle$ *be a context setting and let* $\mathsf{DS} = \langle d_g^{\mathsf{CS}}, f\rangle$ *be a distance setting, where* g *and* f *are hereditary aggregation functions. Let* $\mathcal{DB} = \langle \mathcal{D} \sqcup \{p_1, p_2\}, \mathcal{IC}\rangle$ *be a database such that:*

1. $R(p_1, S) > R(p_2, S)$ *(i.e.,* p_1 *is more relevant than* p_2*), and*
2. $\mathcal{IC} \cup \{p_1, p_2\}$ *is not consistent but* $\mathcal{DB}_1 = \langle \mathcal{D} \sqcup \{p_1\}, \mathcal{IC}\rangle$ *is consistent*[6]*.*

Then $\Delta_{\mathsf{DS}}(\mathcal{DB}) = \{I_1\}$*, where* I_1 *is the (unique) model of* $\mathcal{DB}_1$*.*

Proof. Again, we denote: $g(\overline{0}, x) = g(\{0, \ldots, 0, x, 0, \ldots, 0\})$. Then, for all p, I,

$$d_{\mathsf{DS}}(I, p) = \begin{cases} 0 & \text{if } I \models p, \\ g(\overline{0}, R(p, S)) & \text{otherwise.} \end{cases}$$

Suppose that $\mathcal{D} \cup \mathsf{CWA}(\mathcal{D} \sqcup \{p_1, p_2\}) = \{\psi_1, \ldots, \psi_n\}$. Let $I \in \Delta_{\mathsf{DS}}(\mathcal{DB})$ and $I_1 \in \mathit{mod}(\Gamma_{\mathcal{DB}_1})$ (Such a model exists, since $\mathcal{DB}_1$ is consistent). By Corollary 1, since DS is CS-sensitive (Proposition 5), $I \not\models p_2$, and so $d_{\mathsf{DS}}(I, p_2) = g(\overline{0}, R(p_2, S))$. Now,

$$\begin{aligned} \delta_{\mathsf{DS}}(I, \mathcal{DB}) = {} & f(\{d_{\mathsf{DS}}(I, \psi_1), \ldots, d_{\mathsf{DS}}(I, \psi_n), d_{\mathsf{DS}}(I, p_1), d_{\mathsf{DS}}(I, p_2)\}) = \\ & f(\{d_{\mathsf{DS}}(I, \psi_1), \ldots, d_{\mathsf{DS}}(I, \psi_n), d_{\mathsf{DS}}(I, p_1), g(\overline{0}, R(p_2, S))\}). \end{aligned}$$

and so, since f is non-decreasing,

$$\begin{aligned} \delta_{\mathsf{DS}}(I, \mathcal{DB}) \geq {} & f(\{0, \ldots, 0, 0, g(\overline{0}, R(p_2, S))) = \\ & f(\{d_{\mathsf{DS}}(I_1, \psi_1), \ldots, d_{\mathsf{DS}}(I_1, \psi_n), d_{\mathsf{DS}}(I_1, p_1), d_{\mathsf{DS}}(I_1, p_2)\}) = \\ & \delta_{\mathsf{DS}}(I_1, \mathcal{DB}). \end{aligned}$$

Thus, $I_1 \in \Delta_{\mathsf{DS}}(\mathcal{DB})$. On the other hand, if there is some $q \in \{\psi_1, \ldots, \psi_n, p_1\}$ for which $d_{\mathsf{DS}}(I, q) \neq 0$, then since f is hereditary the above inequality becomes strict, which contradicts the assumption that $I \in \Delta_{\mathsf{DS}}(\mathcal{DB})$. It follows that for every $q \in \{\psi_1, \ldots, \psi_n, p_1\}$ $d_{\mathsf{DS}}(I, q) = d_{\mathsf{DS}}(I_1, q) = 0$, i.e., $I \models q$. One concludes, then, that I is a model of $\mathcal{DB}_1$, that is, $I = I_1$. □

Proposition 6 may be extended in various ways. Below is one such extension.

[6] $\mathcal{DB}_2 = \langle \mathcal{D} \sqcup \{p_2\}, \mathcal{IC}\rangle$ may be consistent as well, but this is not a prerequisite.

Proposition 7. *Let $\mathcal{DB} = \langle \mathcal{D}, \mathcal{IC} \rangle$ $\mathsf{CS} = \langle C, S, R \rangle$ and $\mathsf{DS} = \langle d_g^{\mathsf{CS}}, f \rangle$, where g and f are hereditary aggregation functions. Suppose that $\mathcal{D}$ can be partitioned to two nonempty subsets $\mathcal{D}'$ and $\mathcal{D}''$, such that*

1. $\mathcal{DB}' = \langle \mathcal{D}', \mathcal{IC} \rangle$ is a consistent database,
2. $\forall p'' \in \mathcal{D}''$ $\exists p' \in \mathcal{D}'$ s.t. $\mathcal{IC} \cup \{p', p''\}$ is not consistent, and
3. $\forall p' \in D'$ and $\forall p'' \in \mathcal{D}''$, $R(p', S) > R(p'', S)$.

Then $\Delta_{\mathsf{DS}}(\mathcal{DB}) = \{I'\}$, where I' is the (unique) model of $\mathcal{DB}'$.

Proof. The proof is similar to that of Proposition 6. We omit the details. □

Example 4. Consider again the database in Example 1. By Example 3, the distance setting $\mathsf{DS} = \langle d_H, \Sigma \rangle$ leads to the following two equally good repairs:

Repair 1 :

eNum	name	address	salary
1	John	..., UK	70K$
2	Mary	..., US	90K$

Repair 2 :

eNum	name	address	salary
1	John	..., AT	80K$
2	Mary	..., US	90K$

Sensitivity to context may differentiate between these repairs, preferring one to another. Let us again denote by $\mathsf{T}^1_{\mathsf{UK}}$, $\mathsf{T}^1_{\mathsf{AT}}$ and $\mathsf{T}^2_{\mathsf{US}}$ the tuple according to which John lives in the UK and is payed 70K$, John lives in Austria and is payed 80K$, and the tuple with the information about Mary.

Now, consider the context setting $\mathsf{CS}(\mathcal{L}) = \langle C, S, R \rangle$ and the distance setting $\mathsf{DS} = \langle d_{\Sigma}^{\mathsf{CS}}, \Sigma \rangle$, where $C = \{\mathsf{country}\}$, $\mathsf{Range}(\mathsf{country}) = \{\mathsf{US}, \mathsf{UK}, \mathsf{AT}\}$, $S(\mathsf{country}) = \mathsf{UK}$, and the relevance ranking is given by the following functions:

$$R(\mathsf{T}^{\mathsf{i}}_{\mathsf{c}}, S) = \begin{cases} 1, & \text{if } c = S(\mathsf{country}), \\ 0.5, & \text{otherwise.} \end{cases} \qquad R(\neg\mathsf{T}^{\mathsf{i}}_{\mathsf{c}}, S) = \begin{cases} 0.5, & \text{if } c = S(\mathsf{country}), \\ 1, & \text{otherwise.} \end{cases}$$

Computation of Δ_{DS} is given below (where we abbreviate $d(\psi, S)$ for $d_{\Sigma}^{\mathsf{CS}}(\psi, S)$).

I	$d(I, \mathsf{T}^1_{\mathsf{UK}}, S)$	$d(I, \mathsf{T}^1_{\mathsf{AT}}, S)$	$d(I, \neg\mathsf{T}^1_{\mathsf{US}}, S)$	$d(I, \neg\mathsf{T}^2_{\mathsf{UK}}, S)$	$d(I, \neg\mathsf{T}^2_{\mathsf{AT}}, S)$	$d(I, \mathsf{T}^2_{\mathsf{US}}, S)$	$\delta_{\mathsf{DS}}(I, \Gamma, S)$
$\emptyset$	1	0.5	0	0	0	0.5	2
$\{\mathsf{T}^1_{\mathsf{UK}}\}$	0	0.5	0	0	0	0.5	1
$\{\mathsf{T}^1_{\mathsf{AT}}\}$	1	0	0	0	0	0.5	1.5
$\{\mathsf{T}^1_{\mathsf{US}}\}$	1	0.5	1	0	0	0.5	3
...	...	...	...	...	...	...	...
$\{\mathsf{T}^1_{\mathsf{UK}}, \mathsf{T}^2_{\mathsf{US}}\}$	0	0.5	0	0	0	0	**0.5**
$\{\mathsf{T}^1_{\mathsf{AT}}, \mathsf{T}^2_{\mathsf{US}}\}$	1	0	0	0	0	0	1
...	...	...	...	...	...	...	...

According to CS, the single element in $\Delta_{\mathsf{DS}}(\mathcal{DB})$ satisfies $\{\mathsf{T}^1_{\mathsf{UK}}, \mathsf{T}^2_{\mathsf{US}}\}$, and so Repair 1 is preferred. Dually, in a state S' where $S'(\mathsf{country}) = \mathsf{AT}$, Repair 2 is preferred. Thus, context-aware considerations lead us to choose different repairs according to the relevance ranking, as indeed guaranteed by Propositions 6 and 7.

4 Conclusion

As observed in [13], contexts have largely been ignored by the AI community. In the database community context awareness has only recently been addressed in relation to user preference in querying (consistent) databases [17, 19]. To the best of our knowledge, the approach presented here is the first one to combine inconsistency management with context-aware considerations. Combined with the extensive work available on personalization and automatically determining user's context and preferences (see, e.g., [6, 14]), it may open the door to new inconsistency management solutions and novel database technologies. Implementation and evaluation of the methods in this paper is currently a work in progress.

References

1. G. D. Abowd, A. K. Dey, P. J. Brown, N. Davies, M. Smith, and P. Steggles. Towards a better understanding of context and context-awareness. In *Handheld and Ubiquitous Computing*, pages 304–307. Springer, 1999.
2. M. Arenas, L. Bertossi, and J. Chomicki. Answer sets for consistent query answering in inconsistent databases. *TPLP*, 3(4–5):393–424, 2003.
3. O. Arieli. Distance-based paraconsistent logics. *International Journal of Approximate Reasoning*, 48(3):766–783, 2008.
4. O. Arieli. Reasoning with prioritized information by iterative aggregation of distance functions. *Journal of Applied Logic*, 6(4):589–605, 2008.
5. O. Arieli, M. Denecker, and M. Bruynooghe. Distance semantics for database repair. *Annals of Mathematics and Artificial Intelligence*, 50(3–4):389–415, 2007.
6. M. Baldauf, S. Dustdar, and F. Rosenberg. A survey on context-aware systems. *International Journal of Ad Hoc and Ubiquitous Computing*, 2(4):263–277, 2007.
7. L. Bertossi. Consistent query answering in databases. *SIGMOD Record*, 35(2):68–76, 2006.
8. C. Bolchini, C. Curino, G. Orsi, E. Quintarelli, R. Rossato, F. A. Schreiber, and L. Tanca. And what can context do for data? *Comm. ACM*, 52(11):136–140, 2009.
9. P. Brézillon. Context in artificial intelligence: I. A survey of the literature. *Computers and Artificial Intelligence*, 18(4), 1999.
10. J. Chomicki. Consistent query answering: Five easy pieces. In *Proc. ICDT'07*, LNCS 4353, pages 1–17. Springer, 2007.
11. A. Dey. Understanding and using context. *Personal and Ubiquitous Computing*, 5(1):4–7, 2001.
12. T. Eiter and H. Mannila. Distance measure for point sets and their computation. *Acta Informatica*, 34:109–133, 1997.
13. H. R. Ekbia and A. G. Maguitman. Context and relevance: A pragmatic approach. In *Modeling and Using Context*, pages 156–169. Springer, 2001.
14. K. Henricksen and J. Indulska. Developing context-aware pervasive computing applications: Models and approach. *Pervasive and Mobile Comput.*, 2:37–64, 2006.
15. S. Konieczny, J. Lang, and P. Marquis. DA2 merging operators. *Artificial Intelligence*, 157(1–2):49–79, 2004.
16. S. Konieczny and R. Pino Pérez. Merging information under constraints: A logical framework. *Logic and Computation*, 12(5):773–808, 2002.
17. E. Pitoura, K. Stefanidis, and P. Vassiliadis. Contextual database preferences. *IEEE Data Engeneering Bulletin*, 34(2):19–26, 2011.
18. S. Staworko, J. Chomicki, and J. Marcinkowski. Prioritized repairing and consistent query answering in relational databases. *Ann. Math. Artif. Intel.*, 64:209–246, 2012.
19. K. Stefanidis and E. Pitoura. Fast contextual preference scoring of database tuples. In *Proceedings of EDBT'2008*, pages 344–355. ACM, 2008.
20. K. Stefanidis, E. Pitoura, and P. Vassiliadis. Managing contextual preferences. *Information Systems*, 36(8):1158–1180, 2011.