On Measuring Inconsistency in Relational Databases with Denial Constraints

Francesco Parisi[1] John Grant[2]

[1]Department of Informatics, Modeling, Electronics and System Engineering (DIMES),
University of Calabria, Italy,
fparisi@dimes.unical.it

[2]Department of Computer Science and UMIACS,
University of Maryland, College Park, USA,
grant@cs.umd.edu

24th European Conference on Artificial Intelligence
(Digital) ECAI 2020

August 29—September 8, 2020

Measuring the amount of inconsistency in databases

- Real-world databases are often inconsistent
- Extensive body of work on handling inconsistency in databases (e.g. consistent query answering, inconsistency management policies, data repairing/cleaning/exploration)
- Little work has been done on *measuring inconsistency* in databases (a problem extensively investigated in propositional logic)
- Measuring inconsistency in databases can help in, for instance,
  - assessing data quality (resp., dirtiness)
  - understanding the primary sources of conflicts
  - comparing the amount of inconsistency after updates (or in general between various chunks of information)
  - devising ways to deal with conflicting data, e.g., accepting an update (or merging different sources) only if the measure of inconsistency does not increase (to much) in the new state

Exploring database inconsistency measures

- We introduce the database counterpart $\mathcal{I}_x$ of several propositional inconsistency measures I_x, with $x \in \{B, M, \#, P, A, H, C, \eta\}$
- Every database inconsistency measure (IM) $\mathcal{I}_x$ quantifies the inconsistency by blaming database tuples only
- We introduce the database counterparts of several rationality postulates for IMs, and check for compliance

	Database Inconsistency Measures							
	$\mathcal{I}_B$	$\mathcal{I}_M$	$\mathcal{I}_\#$	$\mathcal{I}_P$	$\mathcal{I}_A$	$\mathcal{I}_H$	$\mathcal{I}_C$	$\mathcal{I}_\eta$
Free-Tuple Independence	✓	✓	✓	✓	✓	✓	✓*	✓
Penalty	✗	✓	✓	✓	✗	✗	✗	✗
Super-Additivity	✗	✓	✓	✓	✓*	✓	✓*	✗
MI-Separability	✗	✓	✓	✗	✗	✗	✗	✗
MI-Normalization	✓	✓	✗	✗	✗	✓	✓*	✗
Equal Conflict	✓	✓	✓	✓	✓	✓	✓*	✓

*: satisfied for database IM but not for proposition counterpart

Complexity of database inconsistency measures

- We investigate the data complexity of the problems of deciding whether a given value v is lower than (**LV**), greater than (**UV**), or equal to (**EV**) the inconsistency measured by a given IM $\mathcal{I}_x$,
- and the complexity of the problem of computing the actual value of an inconsistency measure (**IM** problem)

Measure(s)	$\mathbf{LV}_{\mathcal{I}}(D, v)$	$\mathbf{UV}_{\mathcal{I}}(D, v)$	$\mathbf{EV}_{\mathcal{I}}(D, v)$	$\mathbf{IM}_{\mathcal{I}}(D)$
$\mathcal{I}_B$, $\mathcal{I}_M$,$\mathcal{I}_\#$, $\mathcal{I}_P$	P	P	P	FP
$\mathcal{I}_A$	PP	PP	PP	$\#P$-c
$\mathcal{I}_H$, $\mathcal{I}_C$	$coNP$-c	NP-c	D^p-c	$FP^{NP[log\ n]}$-c
$\mathcal{I}_\eta$	$coNP$-c	NP-c	D^p	FP^{NP}

Outline

General concept of inconsistency measure

- **D** is the set of all databases over a fixed but arbitrary scheme $\mathcal{DS}$
- $\mathcal{C}$ is a fixed but arbitrary set of integrity constraints

Definition (Inconsistency Measure)

A function $\mathcal{I} : \mathbf{D} \to \mathbb{R}_{\infty}^{\geq 0}$ is an **inconsistency measure** if the following two conditions hold for all $D, D' \in \mathbf{D}$:

Consistency $\mathcal{I}(D) = 0$ iff D is consistent (w.r.t. $\mathcal{C}$)

Monotony If $D \subseteq D'$, then $\mathcal{I}(D) \leq \mathcal{I}(D')$

- Consistency and Monotony are called (rationality) postulates
- Postulates are desirable properties for IMs
- Consistency means that all and only consistent databases get measure 0
- Monotony means that the enlargement of a database cannot decrease its measure

Measures $\mathcal{I}_B$, $\mathcal{I}_M$, and $\mathcal{I}_\#$

Definition (Database Inconsistency Measures $\mathcal{I}_B$, $\mathcal{I}_M$, $\mathcal{I}_\#$)

For a database D, the IMs $\mathcal{I}_B$, $\mathcal{I}_M$, and $\mathcal{I}_\#$ are such that

- $\mathcal{I}_B(D) = 1$ if D is inconsistent, 0 otherwise
- $\mathcal{I}_M(D) = |\mathrm{MI}(D)|$
- $\mathcal{I}_\#(D) = \begin{cases} 0 & \text{if } D \text{ is consistent,} \\ \sum_{X \in \mathrm{MI}(D)} \frac{1}{|X|} & \text{otherwise.} \end{cases}$

- $\mathcal{I}_B$ is simply distinguishes between consistent and inconsistent databases (drastic measure)
- $\mathcal{I}_M$ counts the number of *minimal inconsistent subsets* (of D w.r.t. $\mathcal{C}$)
- $\mathcal{I}_\#$ also counts the number of minimal inconsistent subsets, but it gives larger sets a smaller weight

Measures $\mathcal{I}_P$, $\mathcal{I}_A$, and $\mathcal{I}_H$

Definition (Database Inconsistency Measures $\mathcal{I}_P$, $\mathcal{I}_A$, and $\mathcal{I}_H$)

For a database D, the IMs $\mathcal{I}_P$, $\mathcal{I}_A$, and $\mathcal{I}_H$ are such that

- $\mathcal{I}_P(D) = |\text{Problematic}(D)|$.
- $\mathcal{I}_A(D) = (|\text{MC}(D)| + |\text{Contradictory}(D)|) - 1$.
- $\mathcal{I}_H(D) = \min\{|X| \ s.t. \ X \subseteq D \text{ and } \forall M \in \text{MI}(D), \ X \cap M \neq \emptyset\}$.

- $\mathcal{I}_P$ counts the number of *problematic* tuples (i.e., tuples that are in one or more minimal inconsistencies)
- $\mathcal{I}_A$ uses the cardinality of the set of *maximal consistent subsets* (i.e, repairs); the number of contradictory tuples is added as they do not appear in any way in a maximal consistent set
- $\mathcal{I}_H$ counts the minimal number of tuples whose deletion makes the database consistent

A measure based on 3-valued logic (3VL): $\mathcal{I}_C$

- A *3VL-interpretation* is a function i that assigns to each tuple $R(\vec{t})$ in D one of the three truth values: T (*true*), F (*false*), or B (*both*)
- Semantics given by Priest's three-valued logic
- A 3VL interpretation is a 3VL model iff all the integrity constraints in $\mathcal{C}$ are satisfied and no tuple in the database D is assigned F (i.e., B is also allowed, in addition to T)
- For a 3VL interpretation i, $\text{Conflictbase}(i) = \{R(\vec{t}) \mid i(R(\vec{t})) = B\}$ is the set of tuples that have truth value B

Definition (Contension measure $\mathcal{I}_C$)

For a database D, $\mathcal{I}_C(D) = \min\{|\text{Conflictbase}(i)| \mid i \in \text{Models}(D)\}$.

- $\mathcal{I}_C$ counts the minimal number of tuples that if we could consider them both true and false would resolve all inconsistencies

A measure based on probabilistic satisfiability : $\mathcal{I}_\eta$

- We interpret a database D as a (probabilistic satisfiability) PSAT instance $\Gamma_{\mathcal{C},\eta}(D)$, where every tuple in D is assigned probability η, and every integrity constraint in $\mathcal{C}$ is assigned probability 1
- Let η be the maximum probability lower bound that one can consistently assign to all the tuples in the database (if $\eta = 1$ then D is consistent)
- $\mathcal{I}_\eta(D)$ is one minus the maximum probability lower bound η one can consistently assign to all tuples in D.

Definition (Probabilistic measure $\mathcal{I}_\eta$)

Given a database D and a set of integrity constraints $\mathcal{C}$, the inconsistency measure $\mathcal{I}_\eta$ is such that $\mathcal{I}_\eta(D) = 1 - \max\{\eta \in [0,1] \mid \Gamma_{\mathcal{C},\eta}(D) \text{ is satisfiable}\}$.

Outline

Postulates

Definition (Postulates for Database Inconsistency Measures)

Let D, D' be databases (over **D**, with constraints $\mathcal{C}$), $R(\vec{t})$ a tuple of D, and $\mathcal{I}$ an IM. The postulates for database IMs are as follows:

Free-Tuple Independence If $R(\vec{t}) \in \text{Free}(D)$, then $\mathcal{I}(D) = \mathcal{I}(D \setminus \{R(\vec{t})\})$.

Penalty If $R(\vec{t}) \in \text{Problematic}(D)$, then $\mathcal{I}(D) > \mathcal{I}(D \setminus \{R(\vec{t})\})$.

Super-Additivity If $D \cap D' = \varnothing$, then $\mathcal{I}(D \cup D') \geq \mathcal{I}(D) + \mathcal{I}(D')$.

MI-Separability If $\text{MI}(D \cup D') = \text{MI}(D) \cup \text{MI}(D')$ and $\text{MI}(D) \cap \text{MI}(D') = \varnothing$, then $\mathcal{I}(D \cup D') = \mathcal{I}(D) + \mathcal{I}(D')$.

MI-Normalization If $M \in \text{MI}(D)$, then $\mathcal{I}(M) = 1$.

Equal Conflict If $M, M' \in \text{MI}(D)$ and $|M| = |M'|$, then $\mathcal{I}(M) = \mathcal{I}(M')$.

- Independence means that free tuples (not involved in constraints violations) do not change the inconsistency measure
- Penalty states that deleting a problematic tuple (involved in some constraint violation) decreases the measure

Postulates

Definition (Postulates for Database Inconsistency Measures)

Let D, D' be databases (over **D**, with constraints $\mathcal{C}$), $R(\vec{t})$ a tuple of D, and $\mathcal{I}$ an IM. The postulates for database IMs are as follows:

Free-Tuple Independence If $R(\vec{t}) \in \text{Free}(D)$, then $\mathcal{I}(D) = \mathcal{I}(D \setminus \{R(\vec{t})\})$.

Penalty If $R(\vec{t}) \in \text{Problematic}(D)$, then $\mathcal{I}(D) > \mathcal{I}(D \setminus \{R(\vec{t})\})$.

Super-Additivity If $D \cap D' = \varnothing$, then $\mathcal{I}(D \cup D') \geq \mathcal{I}(D) + \mathcal{I}(D')$.

MI-Separability If $\text{MI}(D \cup D') = \text{MI}(D) \cup \text{MI}(D')$ and $\text{MI}(D) \cap \text{MI}(D') = \varnothing$, then $\mathcal{I}(D \cup D') = \mathcal{I}(D) + \mathcal{I}(D')$.

MI-Normalization If $M \in \text{MI}(D)$, then $\mathcal{I}(M) = 1$.

Equal Conflict If $M, M' \in \text{MI}(D)$ and $|M| = |M'|$, then $\mathcal{I}(M) = \mathcal{I}(M')$.

- Super-Additivity and MI-Separability deal with the union of two databases
- Super-Additivity: if the databases are disjoint, then the measure of the union is at least as great as the sum of the measures of the two databases

Postulates

Definition (Postulates for Database Inconsistency Measures)

Let D, D' be databases (over $\mathbf{D}$, with constraints $\mathcal{C}$), $R(\vec{t})$ a tuple of D, and $\mathcal{I}$ an IM. The postulates for database IMs are as follows:

Free-Tuple Independence If $R(\vec{t}) \in \text{Free}(D)$, then $\mathcal{I}(D) = \mathcal{I}(D \setminus \{R(\vec{t})\})$.

Penalty If $R(\vec{t}) \in \text{Problematic}(D)$, then $\mathcal{I}(D) > \mathcal{I}(D \setminus \{R(\vec{t})\})$.

Super-Additivity If $D \cap D' = \varnothing$, then $\mathcal{I}(D \cup D') \geq \mathcal{I}(D) + \mathcal{I}(D')$.

MI-Separability If $\text{MI}(D \cup D') = \text{MI}(D) \cup \text{MI}(D')$ and $\text{MI}(D) \cap \text{MI}(D') = \varnothing$, then $\mathcal{I}(D \cup D') = \mathcal{I}(D) + \mathcal{I}(D')$.

MI-Normalization If $M \in \text{MI}(D)$, then $\mathcal{I}(M) = 1$.

Equal Conflict If $M, M' \in \text{MI}(D)$ and $|M| = |M'|$, then $\mathcal{I}(M) = \mathcal{I}(M')$.

- Super-Additivity and MI-Separability deal with the union of two databases
- MI-Separability: if the minimal inconsistent sets of the two databases partition the minimal inconsistent sets of the union, then the measure of the union is the sum of the measures of the two databases

Postulates

Definition (Postulates for Database Inconsistency Measures)

Let D, D' be databases (over **D**, with constraints $\mathcal{C}$), $R(\vec{t})$ a tuple of D, and $\mathcal{I}$ an IM. The postulates for database IMs are as follows:

Free-Tuple Independence If $R(\vec{t}) \in \text{Free}(D)$, then $\mathcal{I}(D) = \mathcal{I}(D \setminus \{R(\vec{t})\})$.

Penalty If $R(\vec{t}) \in \text{Problematic}(D)$, then $\mathcal{I}(D) > \mathcal{I}(D \setminus \{R(\vec{t})\})$.

Super-Additivity If $D \cap D' = \varnothing$, then $\mathcal{I}(D \cup D') \geq \mathcal{I}(D) + \mathcal{I}(D')$.

MI-Separability If $\text{MI}(D \cup D') = \text{MI}(D) \cup \text{MI}(D')$ and $\text{MI}(D) \cap \text{MI}(D') = \varnothing$, then $\mathcal{I}(D \cup D') = \mathcal{I}(D) + \mathcal{I}(D')$.

MI-Normalization If $M \in \text{MI}(D)$, then $\mathcal{I}(M) = 1$.

Equal Conflict If $M, M' \in \text{MI}(D)$ and $|M| = |M'|$, then $\mathcal{I}(M) = \mathcal{I}(M')$.

- MI-Normalization and Equal Conflict deal with minimal inconsistent sets
- MI-Normalization: every minimal inconsistent set to has measure 1
- Equal Conflict requires minimal inconsistent sets of the same size to have the same measure

Satisfaction of postulates

	Database Inconsistency Measures							
	$\mathcal{I}_B$	$\mathcal{I}_M$	$\mathcal{I}_\#$	$\mathcal{I}_P$	$\mathcal{I}_A$	$\mathcal{I}_H$	$\mathcal{I}_C$	$\mathcal{I}_\eta$
Free-Tuple Independence	✓	✓	✓	✓	✓	✓	✓*	✓
Penalty	✗	✓	✓	✓	✗	✗	✗	✗
Super-Additivity	✗	✓	✓	✓	✓*	✓	✓*	✗
MI-Separability	✗	✓	✓	✗	✗	✗	✗	✗
MI-Normalization	✓	✓	✗	✗	✗	✓	✓*	✗
Equal Conflict	✓	✓	✓	✓	✓	✓	✓*	✓

*: satisfied for database IM but not for proposition counterpart

- Satisfaction for database IMs is not implied by satisfaction for propositional measures (tuples and constraints have different roles)
- For any D, $\mathcal{I}_C(D) = \mathcal{I}_H(D)$; no other pair of measures is identical
- Independence and Equal Conflict are satisfied by all the measures
- $\mathcal{I}_M$ satisfies all the postulates, and $\mathcal{I}_\#$ (the weighted version) satisfies all the postulates but MI-Normalization

Problems

Definition (Lower (**LV**), Upper (**UV**), and Exact Value (**EV**))

Let $\mathcal{I}$ be an IM. Given a database D over a fixed database scheme with a fixed set of constraints, and a positive value $v \in \mathbb{Q}^{>0}$,

- $\mathbf{LV}_{\mathcal{I}}(D, v)$ is the problem of deciding whether $\mathcal{I}(D) \geq v$.

Given D and a non-negative value $v' \in \mathbb{Q}^{\geq 0}$,

- $\mathbf{UV}_{\mathcal{I}}(D, v')$ is the problem of deciding whether $\mathcal{I}(D) \leq v'$, and
- $\mathbf{EV}_{\mathcal{I}}(D, v')$ is the problem of deciding whether $\mathcal{I}(D) = v'$.

Definition (Inconsistency Measurement (**IM**) problem)

Let $\mathcal{I}$ be an IM. Given a database D over a fixed database scheme with a fixed set of constraints, $\mathbf{IM}_{\mathcal{I}}(D)$ is the problem of computing the value of $\mathcal{I}(D)$.

Complexity results

Measure(s)	$\mathbf{LV}_{\mathcal{I}}(D, v)$	$\mathbf{UV}_{\mathcal{I}}(D, v)$	$\mathbf{EV}_{\mathcal{I}}(D, v)$	$\mathbf{IM}_{\mathcal{I}}(D)$
$\mathcal{I}_B$, $\mathcal{I}_M$,$\mathcal{I}_\#$, $\mathcal{I}_P$	P	P	P	FP
$\mathcal{I}_A$	PP	PP	PP	$\#P$-c
$\mathcal{I}_H$, $\mathcal{I}_C$	$coNP$-c	NP-c	D^p-c	$FP^{NP[log\ n]}$-c
$\mathcal{I}_\eta$	$coNP$-c	NP-c	D^p	FP^{NP}

- 4 measures (including $\mathcal{I}_M$ that satisfies all postulates) are polynomial, while they are hard in the propositional setting
- also the complexity of $\mathcal{I}_A$ decreases, compared with its propositional version
- the complexity of $\mathcal{I}_H$ and $\mathcal{I}_\eta$ remains the same, even under data complexity

Outline

Conclusions and future work

- We proposed a framework for measuring inconsistency in databases
- Our inconsistency measures and postulates are inspired by IMs for propositional logic but tailored to database
- We analyzed postulate satisfaction and complexity
- $\mathcal{I}_M$ satisfies all the postulates and can be computed in polynomial time
- $\mathcal{I}_M$, as well as $\mathcal{I}_\#$ and $\mathcal{I}_P$, can be evaluated by standard SQL
- FW1: extend our work to consider types of integrity constraints, (e.g. inclusion dependencies)
- FW2: identify tractable cases for the hard measures and devise efficient algorithms for evaluating IMs
- FW3: fine-grained IMs working at the attribute-level and dealing with incomplete information (e.g., databases with null values)

Thank you for your attention!

... questions?