Stream-based Inconsistency Measurement

Matthias Thimm

Institute for Web Science and Technologies,
Universität Koblenz-Landau, Germany

Abstract

Inconsistency measures have been proposed to assess the severity of inconsistencies in knowledge bases of classical logic in a quantitative way. In general, computing the value of inconsistency is a computationally hard task as it is based on the satisfiability problem which is itself NP-complete. In this work, we address the problem of measuring inconsistency in knowledge bases that are accessed in a *stream* of propositional formulæ. That is, the formulæ of a knowledge base cannot be accessed directly but only once through processing of the stream. This work is a first step towards practicable inconsistency measurement for applications such as Linked Open Data, where huge amounts of information is distributed across the web and a direct assessment of the quality or inconsistency of this information is infeasible due to its size. Here we discuss the problem of stream-based inconsistency measurement on classical logic, in order to make use of existing measures for classical logic. However, it turns out that inconsistency measures defined on the notion of minimal inconsistent subsets are usually not apt to be used in the streaming scenario. In order to address this issue, we adapt measures defined on paraconsistent logics and also present a novel inconsistency measure based on the notion of a *hitting set*. We conduct an extensive empirical analysis on the behavior of these different inconsistency measures in the streaming scenario, in terms of runtime, accuracy, and scalability. We conclude that for two of these measures, the stream-based variant of the new inconsistency measure and the stream-based variant of the *contension* inconsistency measure, large-scale inconsistency measurement in streaming scenarios is feasible.

Keywords: inconsistency measures, approximation algorithm

1. Introduction

Inconsistency measurement is a subfield of Knowledge Representation and Reasoning (KR) that is concerned with the quantitative assessment of the severity of inconsistencies in knowledge bases. Consider the following two knowledge bases $\mathcal{K}_1$ and $\mathcal{K}_2$ formalized in propositional logic:

$$\mathcal{K}_1 = \{a, b \vee c, \neg a \wedge \neg b, d\} \qquad\qquad \mathcal{K}_2 = \{a, \neg a, b, \neg b\}$$

Both knowledge bases are classically inconsistent as for $\mathcal{K}_1$ we have $\{a, \neg a \wedge \neg b\} \models \perp$ and for $\mathcal{K}_2$ we have, e. g., $\{a, \neg a\} \models \perp$. These inconsistencies render the knowledge bases useless for reasoning if one wants to use classical reasoning techniques. In order to make the knowledge bases useful again, one can either use non-monotonic/paraconsistent reasoning techniques (Makinson, 2005; Priest, 1979) or one revises the knowledge bases appropriately to make them consistent (Hansson, 2001). Looking at the knowledge bases $\mathcal{K}_1$ and $\mathcal{K}_2$ one can observe that the *severity* of their inconsistency is different. In $\mathcal{K}_1$, only two out of four formulæ (a and $\neg a \wedge \neg b$) are *participating* in making $\mathcal{K}_1$ inconsistent while for $\mathcal{K}_2$ all formulæ contribute to its inconsistency. Furthermore, for $\mathcal{K}_1$ only two propositions (a and b) are conflicting and using e. g. paraconsistent reasoning one could still infer meaningful statements about c and d. For $\mathcal{K}_2$ no such statement can be made. This leads to the assessment that $\mathcal{K}_2$ should be regarded *more* inconsistent than $\mathcal{K}_1$. Inconsistency measures can be used to quantitatively assess the inconsistency of knowledge bases and to provide a guide for how to repair them. Moreover, they can be used as an analytical tool to assess the quality of knowledge representation. For example, one simple inconsistency measure, see e. g. (Grant and Hunter, 2011), is to take the number of *minimal inconsistent subsets* (MIs) as an indicator for the inconsistency: the more MIs a knowledge base contains, the more inconsistent it is. For $\mathcal{K}_1$ we have then 1 as its inconsistency value and for $\mathcal{K}_2$ we have 2. A lot of different approaches of inconsistency measures and postulates for inconsistency measures have been proposed, mostly for classical propositional logic (Knight, 2001; Hunter, 2002; Hunter and Konieczny, 2004, 2008, 2010; Ma et al., 2009; Mu et al., 2011b,a; Xiao and Ma, 2012; Grant and Hunter, 2011, 2013; Besnard, 2014; McAreavey et al., 2014; Jabbour et al., 2014b), but also for classical first-order logic (Grant and Hunter, 2006, 2008), description logics (Ma et al., 2007; Deng et al., 2007; Qi and Hunter, 2007; Zhou et al., 2009), default logics (Doder et al., 2010), and probabilistic and other weighted logics (Daniel, 2009; Muiño, 2011; Ma et al., 2012; Thimm, 2013, 2014a; Potyka, 2014; Mu et al., 2014).

Inconsistencies arise easily when many experts share their knowledge in order to construct a joint knowledge base, particularly for large knowledge bases as they appear in, e. g., Semantic Web applications (Sacramento et al., 2012). So far, the field of inconsistency measurement is focused on the problem on what a *reasonable* inconsistency measure is and what properties it should satisfy. In this paper, we consider the *computational* problems of inconsistency measurement, particularly with respect scenarios where the knowledge base can only be processed in a step-by-step fashion, i. e., in *streams*. More precisely, we consider a scenario where, instead of a knowledge base $\mathcal{K}$ we are faced with a stream $\mathcal{S}$ that for any point in time $i \in \mathbb{N}$ gives us a propositional formula $\phi = \mathcal{S}(i)$. The measures we are interested in update for every time step i the currently computed inconsistency value and therefore approximate the actual inconsistency value of $\bigcup_{j=1}^{i}\{\mathcal{S}(j)\}$ with the limiting case $i \to \infty$.

To address the issue of stream-based inconsistency measurement, we present a novel inconsistency measure $\mathcal{I}_{hs}$ that is inspired by the η-inconsistency measure of Knight (2002) and is particularly apt to be applied to the streaming scenario. This measure bases on the notion of a *hitting set* which (in our context) is a minimal set of classical interpretations such that every formula of a knowledge base is satisfied by at least one element of the set. We then formalize the problem of stream-based inconsistency measurement, describe desirable properties of stream-based inconsistency measures by relating the problem to the classical setting of inconsistency measurement, and propose specific instantiations for stream-based inconsistency measures. We investigate the properties and the behavior of our new measures both analytically and empirically. For the latter, we conduct an extensive empirical evaluation on artificial data. Our findings show that the stream-based variant of our novel measure, as well as a measure based on paraconsistent logics are suitable in terms of runtime, accuracy, and scalability for the stream-based scenario. In summary, the contributions of this paper are as follows:

1. We present a novel inconsistency measure $\mathcal{I}_{hs}$ based on hitting sets and show how this measure relates to other measures (Section 3).
2. We formalize a theory of inconsistency measurement in streams and relate it to the classical setting of inconsistency measurement (Section 4).
3. We provide a window-based approach for applying classical inconsistency measures to the streaming case and develop specific approaches for some concrete classical measures (Section 5).
4. We conduct an extensive empirical study on the behavior of those inconsistency measures in terms of runtime, accuracy, and scalability. In particular,

we show that the stream variants of $\mathcal{I}_{hs}$ and of the *contension* measure $\mathcal{I}_c$ are effective and accurate for measuring inconsistency in the streaming scenario (Section 6).

Additionally, we give necessary preliminaries for propositional logic in Section 2, provide some review of related work in Section 7 and conclude the paper in Section 8. Proofs of technical results can be found in the appendix. This paper extends and revises the previously published paper (Thimm, 2014b) by correcting and extending technical results, providing proofs, and adding further discussion.

2. Preliminaries

Let At be a propositional signature, i. e., a (finite) set of propositions (also called atoms), and let $\mathcal{L}(\mathsf{At})$ the corresponding propositional language constructed using the usual connectives $\wedge$ (*and*), $\vee$ (*or*), and $\neg$ (*negation*).

Definition 1. A knowledge base $\mathcal{K}$ is a finite set of formulæ $\mathcal{K} \subseteq \mathcal{L}(\mathsf{At})$. Let $\mathbb{K}(\mathsf{At})$ be the set of all knowledge bases.

We write $\mathbb{K}$ instead of $\mathbb{K}(\mathsf{At})$ when there is no ambiguity regarding the signature. If X is a formula or a set of formulæ we write $\mathsf{At}(X)$ to denote the set of propositions appearing in X. Semantics to a propositional language $\mathcal{L}(\mathsf{At})$ is given by *interpretations* and an *interpretation* ω on At is a function $\omega : \mathsf{At} \rightarrow \{\mathsf{true}, \mathsf{false}\}$. Let $\mathsf{Int}(\mathsf{At})$ denote the set of all interpretations for At. An interpretation ω *satisfies* (or is a *model* of) an atom $a \in \mathsf{At}$, denoted by $\omega \models a$, if and only if $\omega(a) = \mathsf{true}$. For $\omega \in \mathsf{Int}(\mathsf{At})$ and $\phi, \phi' \in \mathcal{L}(\mathsf{At})$ we define

- $\omega \models \neg\phi$ if and only if $\omega \not\models \phi$
- $\omega \models \phi \wedge \phi'$ if and only if $\omega \models \phi$ and $\omega \models \phi'$
- $\omega \models \phi \vee \phi'$ if and only if $\omega \models \phi$ or $\omega \models \phi'$

As an abbreviation we sometimes identify an interpretation ω with its *complete conjunction*, i. e., if $a_1, \ldots, a_n \in \mathsf{At}$ are those propositions that are assigned true by ω and $a_{n+1}, \ldots, a_m \in \mathsf{At}$ are those propositions that are assigned false by ω we identify ω by $a_1 \ldots a_n \overline{a_{n+1}} \ldots \overline{a_m}$ (or any permutation of this). For example, the interpretation ω_1 on $\{a, b, c\}$ with $\omega(a) = \omega(c) = \mathsf{true}$ and $\omega(b) = \mathsf{false}$ is abbreviated by $a\bar{b}c$.

For $\Phi \subseteq \mathcal{L}(\mathsf{At})$ we also define $\omega \models \Phi$ if and only if $\omega \models \phi$ for every $\phi \in \Phi$. Define furthermore the set of models $\mathsf{Mod}(X) = \{\omega \in \mathsf{Int}(\mathsf{At}) \mid \omega \models X\}$ for every formula or set of formulæ X. Two formulæ or sets of formulæ X and Y are *equivalent*, denoted by $X \equiv Y$, if and only if $\mathsf{Mod}(X) = \mathsf{Mod}(Y)$. Furthermore, two knowledge bases $\mathcal{K}$, $\mathcal{K}'$ are *semi-extensionally equivalent* ($\mathcal{K} \equiv^{\sigma} \mathcal{K}'$) if there is a bijection $\sigma : \mathcal{K} \to \mathcal{K}'$ such that for all $\alpha \in \mathcal{K}$ we have $\alpha \equiv \sigma(\alpha)$ (Thimm, 2013). If $\mathsf{Mod}(X) = \emptyset$ we also write $X \models \perp$ and say that X is *inconsistent*. Note that checking $X \not\models \perp$ is an NP-complete problem as it is equivalent to the satisfiability problem SAT (Cook, 1971).

Let $\mathbb{R}_0^+$ be the set of non-negative real numbers. Inconsistency measures are functions $\mathcal{I} : \mathbb{K} \to \mathbb{R}_0^+ \cup \{\infty\}$ that aim at assessing the severity of the inconsistency in a knowledge base $\mathcal{K}$, cf. (Grant and Hunter, 2011). The basic idea is that the larger the inconsistency in $\mathcal{K}$ the larger the value $\mathcal{I}(\mathcal{K})$. However, inconsistency is a concept that is not easily quantified and there have been a couple of proposals for inconsistency measures so far, in particular for classical propositional logic, see e. g. (Knight, 2001; Hunter, 2002; Hunter and Konieczny, 2004, 2008, 2010; Ma et al., 2009; Mu et al., 2011b,a; Xiao and Ma, 2012; Grant and Hunter, 2011, 2013; Besnard, 2014; McAreavey et al., 2014; Jabbour et al., 2014b). Below we recall some popular measures but we first introduce some necessary notations. Let $\mathcal{K} \in \mathbb{K}$ be some knowledge base.

Definition 2. A set $M \subseteq \mathcal{K}$ is called *minimal inconsistent subset* (MI) of $\mathcal{K}$ if $M \models \perp$ and there is no $M' \subset M$ with $M' \models \perp$. Let $\mathsf{MI}(\mathcal{K})$ be the set of all MIs of $\mathcal{K}$.

Definition 3. A formula $\alpha \in \mathcal{K}$ is called *free formula* of $\mathcal{K}$ if there is no $M \in \mathsf{MI}(\mathcal{K})$ with $\alpha \in M$. Let $\mathsf{Free}(\mathcal{K})$ denote the set of all free formulæ of $\mathcal{K}$.

We adopt the following definition of a (basic) inconsistency measure from (Grant and Hunter, 2011).

Definition 4. A *basic inconsistency measure* is a function $\mathcal{I} : \mathbb{K} \to \mathbb{R}_0^+ \cup \{\infty\}$ that satisfies the following three conditions:

1. $\mathcal{I}(\mathcal{K}) = 0$ if and only if $\mathcal{K}$ is consistent,
2. if $\mathcal{K} \subseteq \mathcal{K}'$ then $\mathcal{I}(\mathcal{K}) \leq \mathcal{I}(\mathcal{K}')$, and
3. for all $\alpha \in \mathsf{Free}(\mathcal{K})$ we have $\mathcal{I}(\mathcal{K}) = \mathcal{I}(\mathcal{K} \setminus \{\alpha\})$.

The first property (also called *consistency*) of a basic inconsistency measure ensures that all consistent knowledge bases receive a minimal inconsistency value and every inconsistent knowledge base receives a positive inconsistency value. The second property (also called *monotony*) states that the value of inconsistency

α	β	$\upsilon(\alpha \wedge \beta)$	$\upsilon(\alpha \vee \beta)$
T	T	T	T
T	B	B	T
T	F	F	T
B	T	B	T
B	B	B	B
B	F	F	B
F	T	F	T
F	B	F	B
F	F	F	F

α	$\upsilon(\neg\alpha)$
T	F
B	B
F	T

Table 1: Truth tables for propositional three-valued logic (Priest, 1979).

cannot decrease when adding new information. The third property (also called *free formula independence*) states that removing harmless formulæ from a knowledge base—i. e., formulæ that do not contribute to the inconsistency—does not change the value of inconsistency. If $\mathcal{I}$ is a basic inconsistency measure and $\mathcal{K} \in \mathbb{K}$ is a knowledge base we say that $\mathcal{I}(\mathcal{K})$ is the *inconsistency value* of $\mathcal{K}$ wrt. $\mathcal{I}$. In the following we will drop the "basic" and refer to measures satisfying the above three conditions simply as inconsistency measures. For the remainder of this paper we consider the following selection of inconsistency measures: the MI measure $\mathcal{I}_{\mathsf{MI}}$, the MI^c measure $\mathcal{I}_{\mathsf{MI}^c}$, the contension measure $\mathcal{I}_c$, and the η-measure $\mathcal{I}_\eta$, which will be defined below, cf. (Grant and Hunter, 2011; Knight, 2002).

In order to define the contension measure $\mathcal{I}_c$ we need to consider three-valued interpretations for propositional logic (Priest, 1979). A three-valued interpretation υ on At is a function $\upsilon : \mathsf{At} \to \{T, F, B\}$ where the values T and F correspond to the classical truth values true and false, respectively. The additional truth value B stands for *both* and is meant to represent a conflicting truth value for a proposition. The function υ is extended to arbitrary formulæ as shown in Table 1. Then, an interpretation υ satisfies a formula α, denoted by $\upsilon \models^3 \alpha$ if either $\upsilon(\alpha) = T$ or $\upsilon(\alpha) = B$.

For defining the η-inconsistency measure (Knight, 2002) we need to consider probability functions P of the form $P : \mathsf{Int}(\mathsf{At}) \to [0, 1]$ with $\sum_{\omega \in \mathsf{Int}(\mathsf{At})} P(\omega) = 1$. Let $\mathcal{P}(\mathsf{At})$ be the set of all those probability functions and for a given probability function $P \in \mathcal{P}(\mathsf{At})$ define the probability of an arbitrary formula α via $P(\alpha) = \sum_{\omega \models \alpha} P(\omega)$.

Definition 5. Let $\mathcal{I}_{\mathsf{MI}}$, $\mathcal{I}_{\mathsf{MI}^c}$, $\mathcal{I}_c$, and $\mathcal{I}_\eta$ be defined via

$$\begin{aligned}
\mathcal{I}_{\mathsf{MI}}(\mathcal{K}) &= |\mathsf{MI}(\mathcal{K})| \\
\mathcal{I}_{\mathsf{MI}^c}(\mathcal{K}) &= \sum_{M \in \mathsf{MI}(\mathcal{K})} \frac{1}{|M|} \\
\mathcal{I}_c(\mathcal{K}) &= \min\{|\upsilon^{-1}(B)| \mid \upsilon \models^3 \mathcal{K}\} \\
\mathcal{I}_\eta(\mathcal{K}) &= 1 - \max\{\xi \mid \exists P \in \mathcal{P}(\mathsf{At}) : \forall \alpha \in \mathcal{K} : P(\alpha) \geq \xi\}
\end{aligned}$$

The measure $\mathcal{I}_{\mathsf{MI}}$ takes the number of minimal inconsistent subsets of a knowledge base as an indicator for the amount of inconsistency: the more minimal inconsistent subsets the more severe the inconsistency. The measure $\mathcal{I}_{\mathsf{MI}^c}$ refines this idea by also taking the size of the minimal inconsistent subsets into account. Here the idea is that larger minimal inconsistent subsets are less severe than smaller minimal inconsistent subsets (the less formulæ are needed to produce an inconsistency the more "obvious" the inconsistency). The measure $\mathcal{I}_c$ considers the set of three-valued models of a knowledge base (which is always non-empty) and uses the minimal number of propositions with conflicting truth value as an indicator for inconsistency. Finally, the measure $\mathcal{I}_\eta$ (which always assigns an inconsistency value between 0 and 1) looks for the maximal probability one can assign to every formula of a knowledge base[1]. All these measures are basic inconsistency measures as defined in Definition 4.

Example 1. For the knowledge bases $\mathcal{K}_1 = \{a, b \vee c, \neg a \wedge \neg b, d\}$ and $\mathcal{K}_2 = \{a, \neg a, b, \neg b\}$ from the introduction we obtain the following inconsistency values.

The knowledge base $\mathcal{K}_1$ contains one minimal inconsistent subset $\{a, \neg a \wedge \neg b\}$, i. e. $\mathsf{MI}(\mathcal{K}_1) = \{\{a, \neg a \wedge \neg b\}\}$, while $\mathcal{K}_2$ contains two minimal inconsistent subsets $\{a, \neg a\}$ and $\{b, \neg b\}$, i. e. $\mathsf{MI}(\mathcal{K}_2) = \{\{a, \neg a\}, \{b, \neg b\}\}$. This results in $\mathcal{I}_{\mathsf{MI}}(\mathcal{K}_1) = 1$ and $\mathcal{I}_{\mathsf{MI}}(\mathcal{K}_2) = 2$.

As the size of the only minimal inconsistent subset of $\mathcal{K}_1$ is 2 we have $\mathcal{I}_{\mathsf{MI}^c}(\mathcal{K}_1) = 1/2$. For $\mathcal{K}_2$ we have two minimal inconsistent subsets each of size 2, resulting in $\mathcal{I}_{\mathsf{MI}^c}(\mathcal{K}_2) = 1/2 + 1/2 = 1$.

For the propositional signature $\mathsf{At}_1 = \{a, b, c, d\}$ consider the three-valued interpretation υ_1 defined via

$$\upsilon_1(a) = B \qquad \upsilon_1(b) = F \qquad \upsilon_1(c) = T \qquad \upsilon_1(d) = T$$

[1]Note that we modified the definition of $\mathcal{I}_\eta$ slightly compared to the original definition in order to fit our framework.

and observe $\upsilon_1 \models^3 \mathcal{K}_1$. Note that $|\upsilon^{-1}(B)| = 1$ and that there cannot be another υ which assigns to fewer propositions the value B. So we have $\mathcal{I}_c(\mathcal{K}_1) = 1$. For the propositional signature $\mathsf{At}_2 = \{a, b\}$ consider the three-valued interpretation υ_2 defined via

$$\upsilon_2(a) = B \qquad\qquad \upsilon_2(b) = B$$

and observe $\upsilon_2 \models^3 \mathcal{K}_2$. Note that $|\upsilon^{-1}(B)| = 2$ and that there cannot be another υ which assigns to fewer propositions the value B. So we have $\mathcal{I}_c(\mathcal{K}_2) = 2$.

For At_1 consider the probability function $P_1 : \mathsf{Int}(\mathsf{At}_1) \to [0, 1]$ defined via $P_1(abcd) = 1/2$, $P_1(\overline{a}\overline{b}cd) = 1/2$ and $P_1(\omega) = 0$ for all remaining $\omega \in \mathsf{Int}(\mathsf{At}_1)$. Then we have

$$\begin{aligned}
P_1(a) &= P_1(abcd) = 0.5\\
P_1(b \vee c) &= P_1(abcd) + P_1(\overline{a}\overline{b}cd) = 0.5 + 0.5 = 1\\
P_1(\neg a \wedge \neg b) &= P_1(\overline{a}\overline{b}cd) = 0.5\\
P_1(d) &= P_1(abcd) + P_1(\overline{a}\overline{b}cd) = 0.5 + 0.5 = 1
\end{aligned}$$

and therefore for all $\alpha \in \mathcal{K}_1$ it is $P_1(\alpha) \geq 1/2$. Observe that there cannot be another P which assigns larger probability to all formulas, so we have $\mathcal{I}_\eta(\mathcal{K}_1) = 1 - 1/2 = 1/2$. For At_2 consider the probability function $P_2 : \mathsf{Int}(\mathsf{At}_2) \to [0, 1]$ defined via $P_2(ab) = 1/2$, $P_2(\overline{a}\overline{b}) = 1/2$ and $P_2(\omega) = 0$ for all remaining $\omega \in \mathsf{Int}(\mathsf{At}_2)$. Then we have $P_2(a) = P_2(b)$ and also $\mathcal{I}_\eta(\mathcal{K}_2) = 1 - 1/2 = 1/2$.

In summary, these are the inconsistency values obtained for the discussed inconsistent measures:

$$\begin{array}{llll}
\mathcal{I}_{\mathsf{MI}}(\mathcal{K}_1) = 1 & \mathcal{I}_{\mathsf{MI}^c}(\mathcal{K}_1) = 1/2 & \mathcal{I}_c(\mathcal{K}_1) = 1 & \mathcal{I}_\eta(\mathcal{K}_1) = 1/2\\
\mathcal{I}_{\mathsf{MI}}(\mathcal{K}_2) = 2 & \mathcal{I}_{\mathsf{MI}^c}(\mathcal{K}_2) = 1 & \mathcal{I}_c(\mathcal{K}_2) = 2 & \mathcal{I}_\eta(\mathcal{K}_2) = 1/2
\end{array}$$

Example 2. In the previous example, all considered inconsistency measures agreed that $\mathcal{K}_1$ is not more inconsistent than $\mathcal{K}_2$. While e. g. $\mathcal{I}_\eta$ is indifferent about $\mathcal{K}_1$ and $\mathcal{K}_2$ the measure $\mathcal{I}_{\mathsf{MI}^c}$ evaluates $\mathcal{K}_1$ to be less inconsistent than $\mathcal{K}_2$. It can also be the case that inconsistency measures behave completely incomparable. Consider the knowledge bases $\mathcal{K}_3$ and $\mathcal{K}_4$ defined via

$$\begin{aligned}
\mathcal{K}_3 &= \{a, b, c, d, \neg(a \vee b \vee c \vee d), e, f, g, h, \neg(e \vee f \vee g \vee h)\}\\
\mathcal{K}_4 &= \{a, \neg a\}
\end{aligned}$$

Observe that $\mathsf{MI}(\mathcal{K}_3) = \{m_1, m_2\}$ with $m_1 = \{a, b, c, d, \neg(a \vee b \vee c \vee d)\}$ and $m_2 = \{e, f, g, h, \neg(e \vee f \vee g \vee h)\}$, and $\mathsf{MI}(\mathcal{K}_4) = \{m_3\}$ with $m_3 = \{a, \neg a\}$. Then we have

$$\mathcal{I}_{\mathsf{MI}}(\mathcal{K}_3) = |\mathsf{MI}(\mathcal{K}_3)| = 2$$
$$\mathcal{I}_{\mathsf{MI}}(\mathcal{K}_4) = |\mathsf{MI}(\mathcal{K}_4)| = 1$$

but

$$\mathcal{I}_{\mathsf{MI}^c}(\mathcal{K}_3) = \frac{1}{|m_1|} + \frac{1}{|m_2|} = \frac{2}{5}$$
$$\mathcal{I}_{\mathsf{MI}^c}(\mathcal{K}_4) = \frac{1}{|m_3|} = \frac{1}{2}$$

So $\mathcal{I}_{\mathsf{MI}}$ and $\mathcal{I}_{\mathsf{MI}^c}$ completely disagree on the order of $\mathcal{K}_3$ and $\mathcal{K}_4$.

For a more detailed introduction to inconsistency measures see e. g. (Grant and Hunter, 2006) and for some recent developments see e. g. (Besnard, 2014; Jabbour et al., 2014a; Mu et al., 2014; McAreavey et al., 2014; Jabbour et al., 2014b).

3. An Inconsistency Measure based on Hitting Sets

The basic idea of our novel inconsistency measure $\mathcal{I}_{hs}$ is inspired by the measure $\mathcal{I}_\eta$ which seeks a probability function that maximizes the probability of all formulæ of a knowledge base. Basically, the measure $\mathcal{I}_\eta$ looks for a minimal number of models of parts of the knowledge base and maximizes their probability in order to maximize the probability of the formulæ. By just considering this basic idea we arrive at the notion of a *hitting set* for inconsistent knowledge bases.

Definition 6. A subset $H \subseteq \mathsf{Int}(\mathsf{At})$ is called a *hitting set* of $\mathcal{K}$ if for every $\alpha \in \mathcal{K}$ there is $\omega \in H$ with $\omega \models \alpha$.

Some observations on hitting sets are as follows.

Proposition 1. *Let $\mathcal{K}$ be a knowledge base. The following two statements are equivalent:*

1. *there is no $\phi \in \mathcal{K}$ with $\phi \models \perp$*
2. *there exists a hitting set H of $\mathcal{K}$*

Proposition 2. *Let $\mathcal{K}$ be a knowledge base.*

1. *If H is a hitting set of $\mathcal{K}$ then every H' with $H \subseteq H'$ is a hitting set of $\mathcal{K}$.*
2. *$H = \emptyset$ is a hitting set of $\mathcal{K}$ if and only if $\mathcal{K} = \emptyset$.*
3. *$\mathcal{K}$ is consistent if and only if there is a hitting set H of $\mathcal{K}$ with $|H| = 1$.*
4. *If H is a hitting set of $\mathcal{K}$ then H is a hitting set of every $\mathcal{K}'$ with $\mathcal{K}' \subseteq \mathcal{K}$.*

We then define the measure $\mathcal{I}_{hs}$ as the minimal cardinality of a hitting set of the knowledge base minus one.

Definition 7. The function $\mathcal{I}_{hs} : \mathbb{K} \to \mathbb{R}_0^+ \cup \{\infty\}$ is defined via

$$\mathcal{I}_{hs}(\mathcal{K}) = \min\{|H| \mid H \text{ is a hitting set of } \mathcal{K}\} - 1$$

with $\min \emptyset = \infty$ for every $\mathcal{K} \in \mathbb{K} \setminus \{\emptyset\}$ and $\mathcal{I}_{hs}(\emptyset) = 0$.

Note that, following Proposition 1, we have $\mathcal{I}_{hs}(\mathcal{K}) = \infty$ if and only if $\mathcal{K}$ contains a contradictory formula (e. g. $a \wedge \neg a$). Observe also that we need a case differentiation for $\mathcal{K} = \emptyset$ as $\emptyset$ has also the hitting set $\emptyset$, see Proposition 2.

Example 3. We continue Example 1 and consider $\mathcal{K}_1 = \{a, b \vee c, \neg a \wedge \neg b, d\}$ and $\mathcal{K}_2 = \{a, \neg a, b, \neg b\}$. Let $H_1 \subseteq \mathsf{Int}(\mathsf{At})$ be defined via $H_1 = \{abcd, \bar{a}\bar{b}cd\}$. Observe that for both $\mathcal{K}_1$ and $\mathcal{K}_2$ we have that H_1 is a hitting set, i. e., every formula of the knowledge base is satisfied by at least one interpretation of H_1. Furthermore, H_1 is also a minimal hitting set (with respect to set cardinality) as, e. g., for $\mathcal{K}_2$ the two formulas a and $\neg a$ require at least two different interpretations to be satisfied. Therefore we have $\mathcal{I}_{hs}(\mathcal{K}_1) = \mathcal{I}_{hs}(\mathcal{K}_2) = 1$.

Example 4. Consider the knowledge base $\mathcal{K}_5$ defined via

$$\mathcal{K}_5 = \{a \vee d, a \wedge b \wedge c, b, \neg b \vee \neg a, a \wedge b \wedge \neg c, a \wedge \neg b \wedge c\}$$

Then $H_2 = \{abcd, a\bar{b}cd, ab\bar{c}d\} \subseteq \mathsf{Int}(\mathsf{At})$ is a minimal hitting set for $\mathcal{K}_5$ and therefore $\mathcal{I}_{hs}(\mathcal{K}_5) = 2$.

As the following result shows, $\mathcal{I}_{hs}$ is indeed a suitable inconsistency measure.

Proposition 3. *The function $\mathcal{I}_{hs}$ is a (basic) inconsistency measure.*

The result below shows that $\mathcal{I}_{hs}$ also behaves well with some more properties mentioned in the literature (Hunter and Konieczny, 2010; Thimm, 2013).

Proposition 4. *The measure $\mathcal{I}_{hs}$ satisfies the following properties:*

- *If $\alpha \in \mathcal{K}$ is such that $\mathsf{At}(\alpha) \cap \mathsf{At}(\mathcal{K} \setminus \{\alpha\}) = \emptyset$ then $\mathcal{I}_{hs}(\mathcal{K}) = \mathcal{I}_{hs}(\mathcal{K} \setminus \{\alpha\})$ (*safe formula independence*).*
- *If $\mathcal{K} \equiv^{\sigma} \mathcal{K}'$ then $\mathcal{I}_{hs}(\mathcal{K}) = \mathcal{I}_{hs}(\mathcal{K}')$ (*irrelevance of syntax*).*
- *If $\alpha \models \beta$ and $\alpha \not\models \perp$ then $\mathcal{I}_{hs}(\mathcal{K} \cup \{\alpha\}) \geq \mathcal{I}_{hs}(\mathcal{K} \cup \{\beta\})$ (*dominance*).*

However, $\mathcal{I}_{hs}$ is incompatible with some other properties such as *super-additivity* and *MinInc-separability* (Hunter and Konieczny, 2010).

Example 5. A measure $\mathcal{I}$ satisfies *super-additivity* if for $\mathcal{K} \cap \mathcal{K}' = \emptyset$ we have $\mathcal{I}(\mathcal{K} \cup \mathcal{K}') \geq \mathcal{I}(\mathcal{K}) + \mathcal{I}(\mathcal{K}')$. A measure $\mathcal{I}$ satisfies *MinInc-separability* if $\mathsf{MI}(\mathcal{K} \cup \mathcal{K}') = \mathsf{MI}(\mathcal{K}) \cup \mathsf{MI}(\mathcal{K}')$ and $\mathsf{MI}(\mathcal{K}) \cap \mathsf{MI}(\mathcal{K}') = \emptyset$ implies $\mathcal{I}(\mathcal{K} \cup \mathcal{K}') = \mathcal{I}(\mathcal{K}) + \mathcal{I}(\mathcal{K}')$. Consider

$$\mathcal{K}_4 = \{a, \neg a\}$$
$$\mathcal{K}_6 = \{b, \neg b\}$$

Then we have $\mathcal{I}_{hs}(\mathcal{K}_4) = \mathcal{I}_{hs}(\mathcal{K}_6) = 1$. However, we also have $\mathcal{I}_{hs}(\mathcal{K}_4 \cup \mathcal{K}_6) = 1$ as $\{ab, \overline{a}\overline{b}\}$ is a hitting set of $\mathcal{K}_4 \cup \mathcal{K}_6$. It follows that $\mathcal{I}_{hs}$ violates *super-additivity*. Furthermore, observe that $\mathsf{MI}(\mathcal{K}_4) = \{\mathcal{K}_4\}$ and $\mathsf{MI}(\mathcal{K}_6) = \{\mathcal{K}_6\}$ and therefore $\mathsf{MI}(\mathcal{K}_4 \cup \mathcal{K}_6) = \mathsf{MI}(\mathcal{K}_4) \cup \mathsf{MI}(\mathcal{K}_6)$ and $\mathsf{MI}(\mathcal{K}_4) \cap \mathsf{MI}(\mathcal{K}_6) = \emptyset$. It follows that $\mathcal{I}_{hs}$ also violates *MinInc-separability*.

The measure $\mathcal{I}_{hs}$ can also be nicely characterized by a consistent *partitioning* of a knowledge base.

Definition 8. A set $\Phi = \{\Phi_1, \ldots, \Phi_n\}$ with $\Phi_1 \cup \ldots \cup \Phi_n = \mathcal{K}$ and $\Phi_i \cap \Phi_j = \emptyset$ for $i, j = 1, \ldots, n$, $i \neq j$, is called a *partitioning* of $\mathcal{K}$. A partitioning $\Phi = \{\Phi_1, \ldots, \Phi_n\}$ is consistent if $\Phi_i \not\models \perp$ for $i = 1, \ldots, n$.

Proposition 5. *For every knowledge base $\mathcal{K}$*

$$\mathcal{I}_{hs}(\mathcal{K}) = \min\{|\Phi| \mid \Phi \textit{ is a consistent partitioning of } \mathcal{K}\} - 1$$

with $\min \emptyset = \infty$ for every $\mathcal{K} \in \mathbb{K} \setminus \{\emptyset\}$ and $\mathcal{I}_{hs}(\emptyset) = 0$.

As $\mathcal{I}_{hs}$ is inspired by $\mathcal{I}_{\eta}$ we go on by comparing these two measures.

Proposition 6. *Let $\mathcal{K}$ be a knowledge base. If $\infty > \mathcal{I}_{hs}(\mathcal{K}) > 0$ then*

$$\mathcal{I}_\eta(\mathcal{K}) \leq 1 - \frac{1}{\mathcal{I}_{hs}(\mathcal{K}) + 1}$$

Note that for $\mathcal{I}_{hs}(\mathcal{K}) = 0$ we always have $\mathcal{I}_\eta(\mathcal{K}) = 0$ as well, as both are basic inconsistency measures. Furthermore, $\mathcal{I}_{hs}(\mathcal{K}) = \infty$ is equivalent to the existence of a $\phi \in \mathcal{K}$ with $\phi \models \perp$, cf. Proposition 1, which is equivalent to $\mathcal{I}_\eta(\mathcal{K}) = 1$ (Knight, 2002). Although Proposition 6 describes a loose relationship between $\mathcal{I}_\eta$ and $\mathcal{I}_{hs}$ both measures are in general different as we will see below.

We say that an inconsistency measure $\mathcal{I}_1$ *is subsumed by* an inconsistency measure $\mathcal{I}_2$, denoted by $\mathcal{I}_1 \sqsubseteq \mathcal{I}_2$, if the order on knowledge bases imposed by $\mathcal{I}_1$ is a subset of the order imposed by $\mathcal{I}_2$. More formally, $\mathcal{I}_1 \sqsubseteq \mathcal{I}_2$ if and only if $\mathcal{I}_1(\mathcal{K}) < \mathcal{I}_1(\mathcal{K}')$ implies $\mathcal{I}_2(\mathcal{K}) < \mathcal{I}_2(\mathcal{K}')$ for all $\mathcal{K}, \mathcal{K}' \in \mathbb{K}$. Two inconsistency measures $\mathcal{I}_1$ and $\mathcal{I}_2$ are *equivalent*, denoted by $\mathcal{I}_1 \simeq \mathcal{I}_2$, if and only if $\mathcal{I}_1 \sqsubseteq \mathcal{I}_2$ and $\mathcal{I}_2 \sqsubseteq \mathcal{I}_1$.

It turns out that $\mathcal{I}_{hs}$ is neither equivalent nor is subsumed by any of the previously discussed inconsistency measures[2].

Proposition 7. *There is no subsumption relation between $\mathcal{I}_{hs}$ and any $\mathcal{I} \in \{\mathcal{I}_{\mathsf{MI}}, \mathcal{I}_{\mathsf{MI}^c}, \mathcal{I}_c, \mathcal{I}_\eta\}$.*

Corollary 1. *$\mathcal{I}_{hs} \not\simeq \mathcal{I}_{\mathsf{MI}}$, $\mathcal{I}_{hs} \not\simeq \mathcal{I}_{\mathsf{MI}^c}$, $\mathcal{I}_{hs} \not\simeq \mathcal{I}_c$, and $\mathcal{I}_{hs} \not\simeq \mathcal{I}_\eta$.*

Example 6. Consider the knowledge bases $\mathcal{K}_7$ and $\mathcal{K}_8$ given as

$$\mathcal{K}_7 = \{a \wedge b \wedge c, \neg a \wedge \neg b \wedge \neg c\}$$
$$\mathcal{K}_8 = \{a \wedge b, \neg a \wedge b, a \wedge \neg b\}$$

Then we have e. g. $\mathcal{I}_{hs}(\mathcal{K}_7) = 2 < 3 = \mathcal{I}_{hs}(\mathcal{K}_8)$ but $\mathcal{I}_c(\mathcal{K}_7) = 3 > 2 = \mathcal{I}_c(\mathcal{K}_8)$.

4. Inconsistency Measurement in Streams

In the following, we introduce and formalize the problem of inconsistency measurement in streams of propositional formulaæ. The goal of this formalization is to obtain stream-based inconsistency measures that approximate given inconsistency measures when the latter would have been applied to the knowledge base as a whole. We first formalize this setting and, afterwards, provide concrete approaches for some inconsistency measures.

We use a very simple formalization of a stream that is sufficient for our needs.

[2]Note that this result corrects Corollary 1 from (Thimm, 2014b) where $\mathcal{I}_{hs} \sqsubseteq \mathcal{I}_\eta$ was claimed.

Definition 9. A *propositional stream* $\mathcal{S}$ is a function $\mathcal{S} : \mathbb{N} \to \mathcal{L}(\mathsf{At})$. Let $\mathbb{S}$ be the set of all propositional streams.

A propositional stream models a sequence of propositional formulæ. On a wider scope, a propositional stream can also be interpreted as a very general abstraction of the output of a linked open data crawler (such as LDSpider (Isele et al., 2010)) that crawls knowledge formalized as RDF (*Resource Description Framework*) from the web, possibly enriched with OWL semantics to have a well-defined notion of consistency. For notational convenience, we write a propositional stream $\mathcal{S}$ with $\mathcal{S}(0) = \phi_0$, $\mathcal{S}(1) = \phi_1$, $\mathcal{S}(2) = \phi_2$,... also as a tuple $\mathcal{S} = \langle \phi_0, \phi_1, \phi_2, \ldots \rangle$

Using the abstraction of a propositional stream, we can also model large knowledge bases by propositional streams that indefinitely repeat the formulæ of the knowledge base. For that, we assume for a knowledge base $\mathcal{K} = \{\phi_1, \ldots, \phi_n\}$ the existence of a *canonical enumeration* $\mathcal{K}^c = \langle \phi_1, \ldots, \phi_n \rangle$ of the elements of $\mathcal{K}$. This enumeration can be arbitrary and has no specific meaning other than to enumerate the elements in an unambiguous way.

Definition 10. Let $\mathcal{K}$ be a knowledge base and $\mathcal{K}^c = \langle \phi_1, \ldots, \phi_n \rangle$ its canonical enumeration. The *$\mathcal{K}$-stream* $\mathcal{S}_\mathcal{K}$ is defined as $\mathcal{S}_\mathcal{K}(i) = \phi_{(i \,\mathsf{mod}\, n)+1}$ for all $i \in \mathbb{N}$.

Using $\mathcal{K}$-streams we can formalize the desired behavior of stream-based inconsistency measures as follows. Given a $\mathcal{K}$-stream $\mathcal{S}_\mathcal{K}$ and an inconsistency measure $\mathcal{I}$ we aim at defining a measure $\mathcal{J}_\mathcal{I}$ that processes the elements of $\mathcal{S}_\mathcal{K}$ one by one and approximates (or converges to) $\mathcal{I}(\mathcal{K})$.

Definition 11. A *stream-based inconsistency measure* $\mathcal{J}$ is a function $\mathcal{J} : \mathbb{S} \times \mathbb{N} \to \mathbb{R}_0^+ \cup \{\infty\}$.

Definition 12. Let $\mathcal{I}$ be an inconsistency measure and $\mathcal{J}$ a stream-based inconsistency measure. Then $\mathcal{J}$ *approximates* (or *is an approximation of*) $\mathcal{I}$ if for all $\mathcal{K} \in \mathbb{K}$ we have $\lim_{i\to\infty} \mathcal{J}(\mathcal{S}_\mathcal{K}, i) = \mathcal{I}(\mathcal{K})$.

A stream-based inconsistency measure $\mathcal{J}$ is supposed to maintain some state information (which is hidden in the formal definition) that is updated when processing the i-th element of a propositional stream. For $i \in \mathbb{N}$ we say that $\mathcal{J}(\mathcal{S}, i)$ is the *inconsistency value* of $\mathcal{S}$ wrt. $\mathcal{J}$ at time point i. We also require that $\mathcal{J}$ is not able to process formulas from the future, i. e., the value $\mathcal{J}(\mathcal{S}, i)$ is independent of every value $\mathcal{S}(j)$ for $j > i$. More formally:

Definition 13. A stream-based inconsistency measure $\mathcal{J}$ is *future-ignorant* if and only if for all $\mathcal{S}, \mathcal{S}' \in \mathbb{S}$, if $\mathcal{S}(i) = \mathcal{S}'(i)$ for all $i = 0, \ldots, n$ then $\mathcal{J}(\mathcal{S}, n) = \mathcal{J}(\mathcal{S}', n)$.

In the following, we only consider future-ignorant stream-based inconsistency measures.

5. Stream-based Inconsistency Measures

In this section we develop concrete approaches for adopting classical inconsistency measures, including the $\mathcal{I}_{hs}$ measure developed above, to the streaming scenario. First, we present an approach based on considering a window on the stream at any time point $i \in \mathbb{N}$. Second, we provide approximation algorithms for both $\mathcal{I}_{hs}$ and $\mathcal{I}_c$ that use concepts of the programming paradigms of *simulated annealing* and *evolutionary algorithms*.

5.1. A Window-based Approach for Stream-based Inconsistency Measures

The simplest form of implementing a stream-based variant of any algorithm or function is to use a window-based approach, i. e., to consider at any time point a specific excerpt from the stream and apply the original algorithm or function on this excerpt, cf. (Beck et al., 2015). This approach gives us for each time point $i \in \mathbb{N}$ the inconsistency value of the considered excerpt. In order to not dismiss the inconsistency value determined at time point i in time point $i+1$, we aggregate the newly determined inconsistency value at time point $i+1$ with the one from the previous step using an aggregation function.

Definition 14. An *aggregation function* g is a function $g : (\mathbb{R}_0^+ \cup \{\infty\}) \times (\mathbb{R}_0^+ \cup \{\infty\}) \to \mathbb{R}_0^+ \cup \{\infty\}$ with

1. $g(x, y) \in [\min\{x, y\}, \max\{x, y\}]$ for all $x, y \in \mathbb{R}_0^+$,
2. $g(x, \infty) \geq x$ for all $x \in \mathbb{R}_0^+$,
3. $g(\infty, y) \geq y$ for all $y \in \mathbb{R}_0^+$, and
4. $g(\infty, \infty) = \infty$.

Possible aggregation functions are, e. g., the maximum function $\max$ or a smoothing function $g_\alpha(x, y) = \alpha x + (1 - \alpha)y$ for some $\alpha \in [0, 1]$ (for every $x, y \in \mathbb{R}_0^+ \cup \{\infty\}$).[3]

[3] With $\max(x, \infty) = \max(\infty, x) = \max(\infty, \infty) = \infty$, $g_\alpha(x, \infty) = g_\alpha(\infty, x) = g_\alpha(\infty, \infty) = \infty$ for $x \in \mathbb{R}_0^+$.

For any propositional stream $\mathcal{S}$ let $\mathcal{S}^{i,j}$ (for $i \leq j$) be the knowledge base obtained by taking the formulæ from $\mathcal{S}$ between positions i and j, i. e., $\mathcal{S}^{i,j} = \{\mathcal{S}(i), \ldots, \mathcal{S}(j)\}$.

Definition 15. Let $\mathcal{I}$ be an inconsistency measure, $w \in \mathbb{N} \cup \{\infty\}$, and g an aggregation function. We define the *window-based inconsistency measure* $\mathcal{J}_{\mathcal{I}}^{w,g} : \mathbb{S} \times \mathbb{N} \to \mathbb{R}_0^+ \cup \{\infty\}$ via

$$\mathcal{J}_{\mathcal{I}}^{w,g}(\mathcal{S}, i) = \begin{cases} \mathcal{I}(\{\mathcal{S}(0)\}) & \text{if } i = 0 \\ g(\mathcal{I}(\mathcal{S}^{\max\{0,i-w+1\},i}), \mathcal{J}_{\mathcal{I}}^{w,g}(\mathcal{S}, i-1)) & \text{otherwise} \end{cases}$$

for every $\mathcal{S}$ and $i \in \mathbb{N}$.[4]

The intuition behind the window-based inconsistency measure $\mathcal{J}_{\mathcal{I}}^{w,g}$ is as follows. At a specific time point i the current inconsistency value $\mathcal{J}_{\mathcal{I}}^{w,g}(\mathcal{S}, i)$ is determined by, first, determining the inconsistency value of the knowledge base obtained from joining the previously encountered $w \in \mathbb{N}$ formulas, and then aggregating this value with the previously determined value $\mathcal{J}_{\mathcal{I}}^{w,g}(\mathcal{S}, i-1)$. If $w = \infty$, the value $\mathcal{J}_{\mathcal{I}}^{w,g}(\mathcal{S}, i)$ is determined by aggregating the inconsistency value of the union of all but the last encountered formula with the inconsistency value of the union of all encountered formulas. Observe that $\mathcal{J}_{\mathcal{I}}^{w,g}$ is indeed a future-ignorant stream-based inconsistency measure.

Example 7. Consider the propositional stream $\mathcal{S}_1$ given via

$$\mathcal{S}_1 = \langle a \wedge b, \neg a, \neg b, a \vee b, \neg b \wedge \neg a, \ldots \rangle$$

All further elements of $\mathcal{S}_1$ are unimportant for this example. Consider further the inconsistency measure $\mathcal{I}_{\mathsf{MI}}$, the aggregation function $g_{0.7}$ (smoothing function for $\alpha = 0.7$), and the window size 3. We consider the first four timepoints in the evaluation of $\mathcal{J}_{\mathcal{I}_{\mathsf{MI}}}^{3,g_{0.7}}$.

- At timepoint $i = 0$ we obtain

$$\mathcal{J}_{\mathcal{I}_{\mathsf{MI}}}^{3,g_{0.7}}(\mathcal{S}_1, 0) = \mathcal{I}_{\mathsf{MI}}(\{\mathcal{S}_1(0)\}) = \mathcal{I}_{\mathsf{MI}}(\{a \wedge b\}) = 0$$

[4] For $w = \infty$ we define $\max\{0, -\infty\} = 0$

- For $i = 1$ note that $\mathcal{S}^{\max\{0,i-w+1\},i} = \mathcal{S}^{\max\{0,-1\},1} = \mathcal{S}^{0,1} = \{a \wedge b, \neg a\}$ and we have

$$\begin{aligned}\mathcal{J}^{3,g_{0.7}}_{\mathcal{I}_{\mathsf{MI}}}(\mathcal{S}_1, 1) &= g_{0.7}(\mathcal{I}_{\mathsf{MI}}(\mathcal{S}_1^{\max\{0,1-w\},1}), \mathcal{J}^{3,g_{0.7}}_{\mathcal{I}_{\mathsf{MI}}}(\mathcal{S}_1, 0)) \\ &= g_{0.7}(\mathcal{I}_{\mathsf{MI}}(\{a \wedge b, \neg a\}), 0) \\ &= 0.7 \cdot 1 + (1 - 0.7) \cdot 0 = 0.7\end{aligned}$$

- For $i = 2$ we have

$$\begin{aligned}\mathcal{J}^{3,g_{0.7}}_{\mathcal{I}_{\mathsf{MI}}}(\mathcal{S}_1, 2) &= g_{0.7}(\mathcal{I}_{\mathsf{MI}}(\mathcal{S}_1^{\max\{0,2-w\},2}), \mathcal{J}^{3,g_{0.7}}_{\mathcal{I}_{\mathsf{MI}}}(\mathcal{S}_1, 1)) \\ &= g_{0.7}(\mathcal{I}_{\mathsf{MI}}(\{a \wedge b, \neg a, \neg b\}), 0.7) \\ &= 0.7 \cdot 2 + (1 - 0.7) \cdot 0.7 = 1.61\end{aligned}$$

- For $i = 3$ we have

$$\begin{aligned}\mathcal{J}^{3,g_{0.7}}_{\mathcal{I}_{\mathsf{MI}}}(\mathcal{S}_1, 3) &= g_{0.7}(\mathcal{I}_{\mathsf{MI}}(\mathcal{S}_1^{\max\{0,3-w+1\},3}), \mathcal{J}^{3,g_{0.7}}_{\mathcal{I}_{\mathsf{MI}}}(\mathcal{S}_1, 2)) \\ &= g_{0.7}(\mathcal{I}_{\mathsf{MI}}(\{\neg a, \neg b, a \vee b\}), 1.61) \\ &= 0.7 \cdot 1 + (1 - 0.7) \cdot 1.61 = 1.183\end{aligned}$$

Some observations on the properties of $\mathcal{J}^{w,g}_{\mathcal{I}}$ are as follows.

Proposition 8. *Let $\mathcal{I}$ be an inconsistency measure, $w \in \mathbb{N} \cup \{\infty\}$, and g an aggregation function.*

1. *If w is finite then $\mathcal{J}^{w,g}_{\mathcal{I}}$ is* not *an approximation of $\mathcal{I}$.*
2. *If $w = \infty$ and $g(x, y) \geq (x + y)/2$ then $\mathcal{J}^{w,g}_{\mathcal{I}}$ is an approximation of $\mathcal{I}$.*
3. *$\mathcal{J}^{w,g}_{\mathcal{I}}(\mathcal{S}_{\mathcal{K}}, i) \leq \mathcal{I}(\mathcal{K})$ for every $\mathcal{K} \in \mathbb{K}$ and $i \in \mathbb{N}$.*

As can be seen from Example 7 and item 1. of Proposition 8, the main issue with the window-based approach to measuring inconsistency in streams is that only local information (wrt. the current window) can be used to determine the inconsistency value. If, for example, there is a minimal inconsistent subset not covered by any window—as $\{\mathcal{S}_1(0), \mathcal{S}_1(4)\} = \{a \wedge b, \neg b \wedge \neg a\}$ in Example 7—the inconsistency value obtained by the window-based approach will always be an underestimation of the actual inconsistency value, cf. item 2 of Proposition 8.

5.2. Stream-based Approximation Algorithms for $\mathcal{I}_{hs}$ and $\mathcal{I}_c$

The approximation algorithms for $\mathcal{I}_{hs}$ and $\mathcal{I}_c$ that are presented in this subsection are using concepts of the programming paradigms of *simulated annealing* and *evolutionary algorithms*, which are both approaches to solve non-convex optimization problems (Lawrence, 1987). Let $f : X \to \mathbb{R}$ be some function that has to be maximized. The basic idea of evolutionary algorithms is to maintain a population of domain elements $X_p \subseteq X$. In each iteration step a subset of X_p with maximal values wrt. f is selected and the rest discarded. From the selected set, new domain elements are generated by *crossover* (combing two or more of the selected domain elements) and *mutation* (random alteration of a selected domain element). This process is repeated until some convergence criterium is satisfied. In the best case the population converges to the global maximum of f. Simulated annealing works roughly as follows. In the beginning, a single domain element $x \in X$ is selected at random. In each iteration a random choice is made whether to locally improve x (select an $x' \in X$ in the vicinity that has larger value wrt. f than x) or to "jump" to a different part in X. The probability of jumping decreases with the number of iterations (this feature is also called *cooling*) and the algorithm stops in some local maximum, which is, in the best case, also the global maximum.

The basic idea for the stream-based approximation of $\mathcal{I}_{hs}$ is as follows. At any processing step we maintain a candidate set $C \in 2^{\mathsf{Int(At)}}$ (initialized with the empty set) that approximates a hitting set of the underlying knowledge base. At the beginning of a processing step we make a random choice (with decreasing probability the more formulæ we already encountered) whether to remove some element of C. This action ensures that C does not contain superfluous elements and mirrors the cooling step in simulated annealing. Afterwards we check whether there is still an interpretation in C that satisfies the currently encountered formula. If this is not the case we add some random model of the formula to C (as in the mutation step of evolutionary algorithms). Finally, we update the previously computed inconsistency value with $|C| - 1$, taking also some aggregation function g (as for the window-based approach) into account. In order to increase the probability of successfully finding a minimal hitting set we do not maintain a single candidate set C but a (multi-)set $Cand = \{C_1, \ldots, C_m\}$ (as in evolutionary algorithms) for some previously specified parameter $m \in \mathbb{N}$ and use the minimum size of these candidate hitting sets.

We call a function $f : \mathbb{N} \to [0, 1]$ *null-bound* if there is $k > 0$ such that f is strictly decreasing on $\{0, \ldots, k\}$ and $f(i) = 0$ for all $i > k$.

Algorithm 1 $\texttt{hs-stream}^{m,g,f}(\mathcal{S}, i)$

```
currentValue = 0
Cand = {∅, ..., ∅}
N = 0
for all j = 0, ..., i do
    currentValue = update_hs^{m,g,f}(S(j))
return currentValue
```

Algorithm 2 $\texttt{update}_{hs}^{m,g,f}(form)$

```
N = N + 1
if form ≡ ⊥ then
    currentValue = ∞
if currentValue = ∞ then
    return currentValue
newValue = ∞
for all C ∈ Cand do
    rand ∈ [0, 1]
    if rand < f(N) then
        Remove some random ω from C
    if ¬∃ω ∈ C : ω |= form then
        Add random ω ∈ Mod(form) to C
    newValue = min(newValue, (|C| − 1))
currentValue = g(newValue, currentValue)
return currentValue
```

Definition 16. Let $m \in \mathbb{N}$, g an aggregation function, and $f : \mathbb{N} \to [0, 1]$ null-bound. We define the approximation algorithm $\mathcal{J}_{hs}^{m,g,f}$ via[5]

$$\mathcal{J}_{hs}^{m,g,f}(\mathcal{S}, i) = \texttt{hs-stream}^{m,g,f}(\mathcal{S}, i)$$

for every $\mathcal{S}$ and $i \in \mathbb{N}$. The algorithm $\texttt{hs-stream}^{m,g,f}(\mathcal{S}, i)$ is given in Algorithm 1 and its subroutine $\texttt{update}_{hs}^{m,g,f}$ is depicted in Algorithm 2.

[5]Note that $\mathcal{J}_{hs}^{m,g,f}$ is not strictly a stream-based inconsistency measure (as a mathematical function) according to Definition 11 as it is a randomized algorithm.

At the first call of the algorithm $\texttt{hs-stream}^{m,g,f}$ the value of $currentValue$ (which contains the currently estimated inconsistency value) is initialized to 0, the (multi-)set $Cand \subseteq 2^{\mathsf{Int(At)}}$ (which contains a population of candidate hitting sets) is initialized with m empty sets, and N is initialized with 0. The function f can be any null-bound function and ensures that every candidate C reaches some stable result eventually. The parameter m increases the probability that at least one of the candidate hitting sets attains the global optimum of a minimal hitting set. The algorithm $\texttt{hs-stream}^{m,g,f}$ then repeatedly calls the subroutine $\texttt{update}_{hs}^{m,g,f}$ that updates the inconsistency value with each formula in the stream, up to the i-th formula.

In order to address the special case of contradictory formulas, lines 2–5 of $\texttt{update}_{hs}^{m,g,f}$ ensure that $currentValue$ and thus the inconsistency value is set to ∞.

Example 8. We continue Example 7 and consider again the propositional stream $\mathcal{S}_1$ given via

$$\mathcal{S}_1 = \langle a \wedge b, \neg a, \neg b, a \vee b, \neg b \wedge \neg a, \ldots \rangle$$

We consider $m = 1$ (only one candidate hitting set is maintained) and use the maximum function $\max$ as our aggregation function ($g = \max$). Let $f = f_0$ be defined as $f_0(n) = 1/(n+1)$ for $n \in \{0, \ldots, 10\}$ and $f_0(n) = 0$ for $n > 10$. We consider the first four timepoints in the evaluation of $\mathcal{J}_{hs}^{1,\max,f_1}$.

- For $i = 0$, we first initialize $currentValue = 0$, $Cand = \{\emptyset\}$, and $N = 0$ in Algorithm 1, and then set $N = 1$ in line 1 of Algorithm 2. Lines 2–5 in Algorithm 2 are skipped as we do not have any contradictory formulas in $\mathcal{S}_1$. In line 6 we set $newValue = \infty$. In line 7 we select $C_1 = \emptyset$. Suppose in line 8 we determine $rand = 0.7$. As $f_0(1) = 1/2$ we do not execute line 10. As there is no $\omega \in C_1$ that satisfies $form = \mathcal{S}_1(0) = a \wedge b$ we add some model, e. g., ab, to C_1 in line 12. We set $newValue = 0$ in line 13 and $currentValue = \max(0, 0) = 0$ in line 14.

- For $i = 1$, suppose in line 8 of Algorithm 2 we determine $rand = 0.4$. As $f_0(2) = 1/3$ we do not execute line 10. As there is no $\omega \in C_1 = \{ab\}$ that satisfies $form = \mathcal{S}_1(1) = \neg a$ we add some model, e. g., $\overline{a}\overline{b}$, to C_1 in line 12 (now we have $C_1 = \{ab, \overline{a}\overline{b}\}$). We set $newValue = 1$ in line 13 and $currentValue = \max(1, 0) = 1$ in line 14.

- For $i = 2$, suppose in line 8 of Algorithm 2 we determine $rand = 0.5$. As $f_0(3) = 1/4$ we do not execute line 10. Note that there is $\omega \in C_1$ with $\omega \models form = \mathcal{S}_1(2) = \neg b$ ($\omega = \overline{a}\overline{b}$). Therefore we skip line 12 and set $newValue = 1$ in line 13 and $currentValue = \max(1, 1) = 1$ in line 14.

- For $i = 3$, suppose in line 8 of Algorithm 2 we determine $rand = 0.1$. As $f_0(3) = 1/5 > rand$ we execute line 10 and remove ab from C_1. As there is no $\omega \in C_1$ that satisfies $form = \mathcal{S}_1(3) = a \vee b$ we add some model, e. g., $a\overline{b}$, to C_1 in line 12. We set $newValue = 1$ in line 13 and $currentValue = \max(1, 1) = 1$ in line 14.

As $\mathcal{J}_{hs}^{m,g,f}$ is a random process we cannot show that $\mathcal{J}_{hs}^{m,g,f}$ is an approximation of $\mathcal{I}_{hs}$ in the general case. However, we can give the following result.

Proposition 9. *For every $p \in [0, 1)$, g some aggregation function with $g(x, y) \geq (x + y)/2$, $f : \mathbb{N} \to [0, 1]$ a null-bound function, and $\mathcal{K} \in \mathbb{K}$ there is $m \in \mathbb{N}$ such that with probability greater or equal p it is the case that* $\lim_{i\to\infty} \mathcal{J}_{hs}^{m,g,f}(\mathcal{S}_{\mathcal{K}}, i) = \mathcal{I}_{hs}(\mathcal{K})$.

This result states that $\mathcal{J}_{hs}^{m,g,f}$ indeed approximates $\mathcal{I}_{hs}$ if we choose the number of populations large enough. In the next section we will provide some empirical evidence that even for small values of m results are satisfactory. As for the runtime, note that in lines 2 and 11 of Algorithm 2 an FNP-complete problem is solved (determining some model of a propositional formula). However, under the reasonable assumption that formulas are usually quite small compared to the size of the whole knowledge base the impact of this step is negligible.

Both Definition 16 and Algorithms 1 and 2 can be modified slightly in order to approximate $\mathcal{I}_c$ instead of $\mathcal{I}_{hs}$, yielding a new measure $\mathcal{J}_c^{m,g,f}$.

Definition 17. Let $m \in \mathbb{N}$, g an aggregation function, and $f : \mathbb{N} \to [0, 1]$ some null-bound function. We define the approximation algorithm $\mathcal{J}_c^{m,g,f}$ via[6]

$$\mathcal{J}_c^{m,g,f}(\mathcal{S}, i) = \texttt{c-stream}^{m,g,f}(\mathcal{S}, i)$$

for every $\mathcal{S}$ and $i \in \mathbb{N}$. The algorithm $\texttt{c-stream}^{m,g,f}(\mathcal{S}, i)$ is given in Algorithm 3 and its subroutine $\texttt{update}_c^{m,g,f}$ is depicted in Algorithm 4.

[6]Note that $\mathcal{J}_c^{m,g,f}$ is not strictly a stream-based inconsistency measure (as a mathematical function) according to Definition 11 as it is a randomized algorithm.

Algorithm 3 $\texttt{c-stream}^{m,g,f}(\mathcal{S}, i)$

$currentValue = 0$
$Cand = \{\upsilon_1, \ldots, \upsilon_m\}$
$N = 0$
for all $j = 0, \ldots, i$ **do**
    $currentValue = \texttt{update}_c^{m,g,f}(\mathcal{S}(j))$
return $currentValue$

In $\texttt{c-stream}^{m,g,f}(\mathcal{S}, i)$ and $\texttt{update}_c^{m,g,f}$, the set of candidates $Cand$ contains three-valued interpretations instead of sets of classical interpretations (initialized with randomly chosen interpretations $\upsilon_1, \ldots, \upsilon_m$ with $\upsilon_i^{-1}(B) = \emptyset$ for $i = 1, \ldots, m$). In line 6 of $\texttt{update}_c^{m,g,f}$, we flip some arbitrary proposition from B to T or F. Similarly, in lines 8–13 of $\texttt{update}_c^{m,g,f}$ we flip some propositions to B in order to satisfy the new formula. Finally, the inconsistency value is determined by taking the number of B-valued propositions (the minimum of all candidates in $Cand$).

With respect to the accuracy of $\mathcal{J}_c^{m,g,f}$, we can make a similar statement as for $\mathcal{J}_{hs}^{m,g,f}$.

Proposition 10. *For every $p \in [0, 1)$, g some aggregation function with $g(x, y) \geq (x + y)/2$, $f : \mathbb{N} \to [0, 1]$ a null-bound function, and $\mathcal{K} \in \mathbb{K}$ there is $m \in \mathbb{N}$ such that with probability greater or equal p it is the case that $\lim_{i\to\infty} \mathcal{J}_c^{m,g,f}(\mathcal{S}_\mathcal{K}, i) = \mathcal{I}_c(\mathcal{K})$.*

In order to evaluate the accuracy and performance of these stream-based inconsistency measures in more detail, we perform some empirical experiments in the following section.

6. Empirical Evaluation

In this section we describe our empirical experiments on runtime, accuracy, and scalability of the discussed stream-based inconsistency measures. Our Java implementations[7] have been added to the *Tweety Libraries for Knowledge Representation* (Thimm, 2014c).

[7] $\mathcal{I}_{\mathsf{MI}}, \mathcal{I}_{\mathsf{MI}^c}, \mathcal{I}_\eta, \mathcal{J}_\mathcal{I}^{w,g}$: `http://mthimm.de/r?r=tweety-inc-commons`
$\mathcal{I}_c, \mathcal{I}_{hs}$: `http://mthimm.de/r?r=tweety-inc-pl`
$\mathcal{J}_{hs}^{m,g,f}$: `http://mthimm.de/r?r=tweety-stream-hs`
$\mathcal{J}_c^{m,g,f}$: `http://mthimm.de/r?r=tweety-stream-c`
Evaluation framework: `http://mthimm.de/r?r=tweety-stream-eval`

Algorithm 4 $\texttt{update}_c^{m,g,f}(form)$

$N = N + 1$
$newValue = \infty$
for all $\upsilon \in Cand$ **do**
    rand $\in [0, 1]$
    if $rand < f(N)$ and $\upsilon^{-1}(B) \neq \emptyset$ **then**
        Set random proposition in υ from B to T or F
    if $\upsilon \not\models^3 form$ **then**
        Select random $\omega \in \mathsf{Mod}(form)$
        for all $p \in \mathsf{At}$ **do**
            if $\omega \models p$ and $\upsilon(p) = F$ **then**
                $\upsilon(p) = B$
            if $\omega \not\models p$ and $\upsilon(p) = T$ **then**
                $\upsilon(p) = B$
    $newValue = min(newValue, |\upsilon^{-1}(B)|)$
$currentValue = g(newValue, currentValue)$
return $currentValue$

6.1. Evaluated Approaches

For our evaluation, we considered the inconsistency measures $\mathcal{I}_{\mathsf{MI}}, \mathcal{I}_{\mathsf{MI}^c}, \mathcal{I}_{\eta}, \mathcal{I}_c$, and $\mathcal{I}_{hs}$. We used the SAT solver *lingeling*[8] for the sub-problems of determining consistency and to compute a model of a formula.

For enumerating the set of MIs of a knowledge base (as required by $\mathcal{I}_{\mathsf{MI}}$ and $\mathcal{I}_{\mathsf{MI}^c}$) we used MARCO[9], a tool for computing all minimal unsatisfiable sets of clauses from a knowledge base given in conjunctive normal form (CNF). In order to apply MARCO to our general non-CNF knowledge bases, we used the following approach. First, a general knowledge base $\mathcal{K}$ is converted to CNF, i. e., each formula of $\mathcal{K}$ is converted to a set of clauses. In doing so, we retain a mapping from each original formula to its set of clauses (note that clauses may appear multiple times in the resulting knowledge base, if they originate from different formulas). On the knowledge base in CNF we apply MARCO, which returns the set of all minimal sets of unsatisfiable clauses. Using the mapping to the original formulas, from each minimal set of unsatisfiable clauses a set of formulas is derived.

[8] `http://fmv.jku.at/lingeling/`
[9] `http://sun.iwu.edu/~mliffito/marco/`

By construction, the resulting set of formulas is inconsistent, but not necessarily minimally inconsistent. Therefore, after all these sets have been computed, a final minimality check is performed and all non-minimal sets are filtered out. This approach is similar to the one employed by MIMUS (McAreavey et al., 2014), a tool which also determines MIs from a general knowledge base and is based on CAMUS[10]. We decided to use MARCO with the above preprocessing step instead of MIMUS directly, as initial experiments suggested that the former one is usually faster if the knowledge base contains at least one minimal inconsistent subset (which is the standard case in our evaluation). This observation is consistent with the observations made by Liffiton and Malik (2013), where CAMUS is criticized to be slower than MARCO for determining many minimal unsatisfiable sets. While CAMUS is a multi-purpose tool that also computes *minimal correction sets*, MARCO is optimized for computing minimal unsatisfiable sets of clauses fast.

The measure $\mathcal{I}_\eta$ was implemented using the linear optimization solver *lp_solve*[11].

The measures $\mathcal{I}_{\mathsf{MI}}$, $\mathcal{I}_{\mathsf{MI}^c}$, and $\mathcal{I}_\eta$ were used to define three different versions of the window-based measure $\mathcal{J}_{\mathcal{I}}^{w,g}$ (with $w = 500, 1000, 2000$ and $g = \max$). For the measures $\mathcal{I}_c$ and $\mathcal{I}_{hs}$ we tested each three versions of their streaming variants $\mathcal{J}_c^{m,g_{0.75},f_1}$ and $\mathcal{J}_{hs}^{m,g_{0.75},f_1}$ (with $m = 10, 100, 500$) with $f_1 : \mathbb{N} \to [0, 1]$ defined via $f_1(i) = 1/(i+1)$ for all $i \in \mathbb{N}$ with $i \leq 2^{32}$ and $f_1(i) = 0$ otherwise. Furthermore, $g_{0.75}$ is the smoothing function for $\alpha = 0.75$ as defined in the previous section.

6.2. Experiment Setup

For measuring the runtime of the different approaches we generated 100 random knowledge bases in CNF with each 5000 formulæ and 30 propositions.[12] A knowledge base was generated by randomly determining the propositions appearing in a clause (uniformly distributed and up to a maximum of 4) and randomly negating some of these propositions (uniformly for each proposition). For each generated knowledge base $\mathcal{K}$ we considered its $\mathcal{K}$-stream and processing of the stream was aborted after 40000 iterations. We fed the $\mathcal{K}$-stream to each of the evaluated stream-based inconsistency measures and measured the average runtime per iteration and the total runtime. For each iteration, we set a time-out of 2 minutes and aborted processing of the stream completely if a time-out occurred.

[10] `http://sun.iwu.edu/~mliffito/camus/`

[11] `http://lpsolve.sourceforge.net`

[12] All sampling algorithms can be found at `http://mthimm.de/r?r=tweety-sampler`

Measure	RT (iteration)	RT (total)	Measure	RT (iteration)	RT (total)
$\mathcal{J}^{500,\max}_{\mathcal{I}_{\mathsf{MI}}}$	198ms	133m	$\mathcal{J}^{10,g_{0.75},f_1}_{c}$	0.16ms	6.406s
$\mathcal{J}^{1000,\max}_{\mathcal{I}_{\mathsf{MI}}}$	359ms	240m	$\mathcal{J}^{100,g_{0.75},f_1}_{c}$	1.1ms	43.632s
$\mathcal{J}^{2000,\max}_{\mathcal{I}_{\mathsf{MI}}}$	14703ms	9812m	$\mathcal{J}^{500,g_{0.75},f_1}_{c}$	5.21ms	208.422s
$\mathcal{J}^{500,\max}_{\mathcal{I}_{\mathsf{MI}^c}}$	198ms	134m	$\mathcal{J}^{10,g_{0.75},f_1}_{hs}$	0.07ms	2.788s
$\mathcal{J}^{1000,\max}_{\mathcal{I}_{\mathsf{MI}^c}}$	361ms	241m	$\mathcal{J}^{100,g_{0.75},f_1}_{hs}$	0.24ms	9.679s
$\mathcal{J}^{2000,\max}_{\mathcal{I}_{\mathsf{MI}^c}}$	14812ms	9874m	$\mathcal{J}^{500,g_{0.75},f_1}_{hs}$	1.02ms	40.614s

Table 2: Runtimes for the evaluated measures; each value is averaged over 100 random knowledge bases of 5000 formulæ; the total runtime is after 40000 iterations

In order to measure accuracy, for each of the considered approaches we generated another 100 random knowledge bases (not necessarily in CNF) with specifically set inconsistency values, used otherwise the same settings as above, and measured the returned inconsistency values.

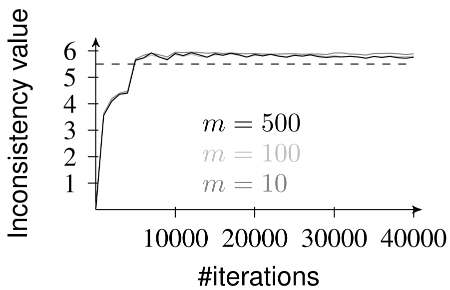
To evaluate the scalability of our stream-based approach of $\mathcal{I}_{hs}$ we conducted a third experiment[13] where we fixed the number of propositions (60) and the specifically set inconsistency value (200) and varied the size of the knowledge bases from 5000 to 50000 (with steps of 5000 formulæ). We measured the total runtime up to the point when the inconsistency value was within a tolerance of ± 1 of the expected inconsistency value.

The experiments were conducted on a server with two Intel Xeon X5550 QuadCore (2.67 GHz) processors with 8 GB RAM running SUSE Linux 2.6.

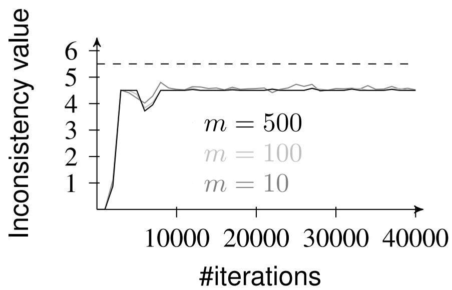
6.3. Results

Our first observation concerns the inconsistency measure $\mathcal{I}_\eta$ which proved to be not suitable to work on large knowledge bases. Computing the value $\mathcal{I}_\eta(\mathcal{K})$ for some knowledge base $\mathcal{K}$ includes solving a linear optimization problem over a number of variables which is (in the worst-case) exponential in the number of propositions of the signature. In our setting with $|\mathsf{At}| = 30$ the generated optimization problem contained therefore $2^{30} = 1073741824$ variables. Hence, even

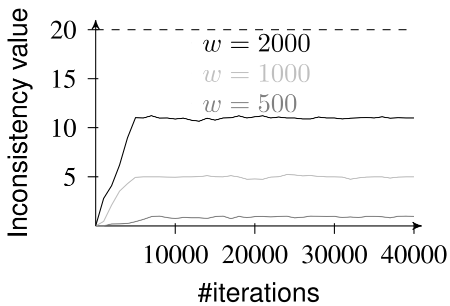
[13] We did the same experiment with our stream-based approach of $\mathcal{I}_c$ but do not report the results due to the similarity to $\mathcal{I}_{hs}$.



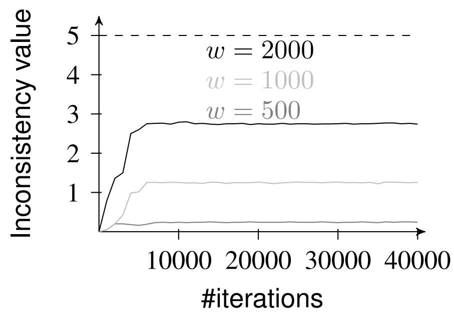
(a) Accuracy $\mathcal{J}_{hs}^{m,g_{0.75},f_1}$



(b) Accuracy $\mathcal{J}_{c}^{m,g_{0.75},f_1}$



(c) Accuracy $\mathcal{J}_{\mathcal{I}_{\mathsf{MI}}}^{w,\max}$



(d) Accuracy $\mathcal{J}_{\mathcal{I}_{\mathsf{MI}^c}}^{w,\max}$

Figure 1: Accuracy performance for the evaluated measures (dashed line is actual inconsistency value); each value is averaged over 100 random knowledge bases of 5000 formulæ (30 propositions) with varying inconsistency values

the optimization problem itself could not be constructed within the timeout of 2 minutes for every step. In the following, we will therefore not report on further results for $\mathcal{I}_\eta$.

As for the runtime of the window-based approaches of $\mathcal{I}_{\mathsf{MI}}$ and $\mathcal{I}_{\mathsf{MI}^c}$ and our stream-based approaches for $\mathcal{I}_c$ and $\mathcal{I}_{hs}$ see Table 2. There one can see that $\mathcal{J}_{\mathcal{I}_{\mathsf{MI}}}^{w,g}$ and $\mathcal{J}_{\mathcal{I}_{\mathsf{MI}^c}}^{w,g}$ on the one hand, and $\mathcal{J}_{c}^{m,g,f}$ and $\mathcal{J}_{hs}^{m,g,f}$ on the other hand, have comparable runtimes, respectively. The former two have almost identical runtimes, which is obvious as the determination of the MIs is the main problem in both their computations. Clearly, $\mathcal{J}_{c}^{m,g,f}$ and $\mathcal{J}_{hs}^{m,g,f}$ are significantly faster per iteration (and in total) than $\mathcal{J}_{\mathcal{I}_{\mathsf{MI}}}^{w,g}$ and $\mathcal{J}_{\mathcal{I}_{\mathsf{MI}^c}}^{w,g}$, only very few milliseconds for the latter and several hundreds and thousands of milliseconds for the former (for all variants of m and w). The impact of increasing m for $\mathcal{J}_{c}^{m,g,f}$ and $\mathcal{J}_{hs}^{m,g,f}$ is expectedly linear while the impact of increasing the window size w for $\mathcal{J}_{\mathcal{I}_{\mathsf{MI}}}^{w,g}$ and $\mathcal{J}_{\mathcal{I}_{\mathsf{MI}^c}}^{w,g}$ is exponential (this

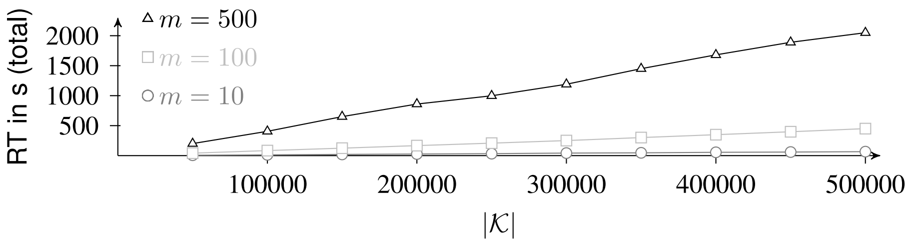


Figure 2: Evaluation of the scalability of $\mathcal{J}_{hs}^{m,g_{0.75},f_1}$; each value is averaged over 10 random knowledge bases of the given size

is also clear as both solve an FNP-hard problem).

As for the accuracy of the different approaches see Figure 1. There one can see that both $\mathcal{J}_{hs}^{m,g,f}$ and $\mathcal{J}_{c}^{m,g,f}$ (Figures 1a and 1b) converge quite quickly (almost right after the knowledge base has been processed once) into a $[-1, 1]$ interval of the actual inconsistency value, where $\mathcal{J}_{hs}^{m,g,f}$ is even closer to it. The window-based approaches (Figures 1c and 1d) have a comparable bad performance (this is clear as those approaches cannot *see* all MIs at any iteration due to the limited window size). Surprisingly, the impact of larger values of m for $\mathcal{J}_{hs}^{m,g,f}$ and $\mathcal{J}_{c}^{m,g,f}$ is rather small in terms of accuracy which suggests that the random process of our algorithm is quite robust. Even for $m = 10$ the results are quite satisfactory.

As for the scalability of $\mathcal{J}_{hs}^{m,g_{0.75},f_1}$ see Figure 2. There one can observe a linear increase in the runtime of all variants wrt. the size of the knowledge base. Furthermore, the difference between the variants is also linear in the parameter m (which is also clear as each population is an independent random process). It is noteworthy, that the average runtime for $\mathcal{J}_{hs}^{10,g_{0.75},f_1}$ is about 66.1 seconds for knowledge bases with 50000 formulæ. As the significance of the parameter m for the accuracy is also only marginal, the measure $\mathcal{J}_{hs}^{10,g_{0.75},f_1}$ is clearly an effective and accurate stream-based inconsistency measure.

7. Related Work

This work is the first to address inconsistency measurement in streaming scenarios. The closest family of related works are approaches for efficient inconsistency measure computation for the classical setting (where a knowledge base is given as a whole).

In (Ma et al., 2009, 2010; Xiao et al., 2010) Ma and colleagues present an anytime algorithm that approximates an inconsistency measure based on a 4-valued paraconsistent logic (similar to the contension inconsistency measure). The algorithm provides lower and upper bounds for this measure and can be stopped at any point in time with some guaranteed quality. The main difference between our framework and the algorithm of (Ma et al., 2009, 2010) is that the latter needs to process the whole knowledge base in each atomic step and is therefore not directly applicable for the streaming scenario. The empirical evaluation in (Ma et al., 2009, 2010) also suggests that our streaming variant of $\mathcal{I}_{hs}$ is much more performant as Ma et al. report an average runtime of their algorithm of about 240 seconds on a knowledge base with 120 formulæ and 20 propositions (no evaluation on larger knowledge bases is given) while our measure has a runtime of only a few seconds for knowledge bases with 5000 formulæ with comparable accuracy[14].

In (McAreavey et al., 2014) an approach is developed for computing measures based on minimal inconsistent subsets (such as $\mathcal{I}_{\mathsf{MI}}$ and $\mathcal{I}_{\mathsf{MI}^c}$) more efficiently. Core to the computation of these measures is the determination of $\mathsf{MI}(\mathcal{K})$ for an arbitrary knowledge base $\mathcal{K}$. While the computational challenges of determining the set of minimal inconsistent subsets for knowledge bases in CNF has been studied for some time in the SAT community—see e. g. (Liffiton and Sakallah, 2005; Büning and Kullmann, 2009; Liffiton and Malik, 2013; Previti and Marques-Silva, 2013)—additional issues arise when considering knowledge bases that are not in CNF. These issues are addressed by McAreavey et al. (2014) where an approach for efficiently computing minimal inconsistent sets for arbitrary knowledge is presented. The approach has been implemented in the tool MIMUS which is based on the tool CAMUS[15] for computing minimal inconsistent subsets of knowledge bases in CNF. In (McAreavey et al., 2014) this tool has been empirically evaluated, also in the context of measuring inconsistency with measures based on minimal inconsistent subsets. Compared to (McAreavey et al., 2014) we consider arbitrary inconsistency measures and not just those based on minimal inconsistent subsets. Still, the work of McAreavey et al. is relevant for applying those inconsistency measures to our streaming scenario. We slightly adapted the general approach of McAreavey et al. (2014) and used MARCO[16], instead of the predecessor CAMUS, for our empirical evaluation (see Section 6).

[14] Although hardware specifications for these experiments are different this huge difference is quite relevant.

[15] http://sun.iwu.edu/~mliffito/camus/

[16] http://sun.iwu.edu/~mliffito/marco/

8. Summary and Conclusion

In this paper we introduced and discussed the problem of stream-based inconsistency measurement. For that, we developed a novel inconsistency measure $\mathcal{I}_{hs}$ that is based on the notion of a *hitting set* and analyzed its properties. We presented a general framework for applying classical inconsistency measures to the streaming scenario and developed specific approximation algorithms for $\mathcal{I}_{hs}$ and the contension measure $\mathcal{I}_c$. Our empirical evaluation showed that the latter two approaches outperform the baseline window-based approaches to measure inconsistency and streams and provide general evidence of the feasibility of stream-based inconsistency measurement.

All discussed inconsistency measures (classical and stream-based ones), as well as the evaluation framework have been implemented in Java and added to the open source project *Tweety*[17] (Thimm, 2014c). Current work is about the application of our work on linked open data sets (Isele et al., 2010) enriched with OWL semantics.

Acknowledgements. I thank the anonymous reviewers for their valuable comments to improve a previous version of this paper.

References

Beck, H., Dao-Tran, M., Eiter, T., Fink, M., 2015. LARS: A Logic-based Framework for Analyzing Reasoning over Streams. In: Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence (AAAI'15).

Besnard, P., 2014. Revisiting Postulates for Inconsistency Measures. In: Fermé, E., Leite, J. a. (Eds.), Logics in Artificial Intelligence. Vol. 8761 of Lecture Notes in Computer Science. Springer International Publishing, pp. 383–396.

Büning, H. K., Kullmann, O., 2009. Minimal Unsatisfiability and Autarkies. In: Handbook of Satisfiability. IOS Press, pp. 339–401.

Cook, S. A., 1971. The Complexity of Theorem-Proving Procedures. In: Proceedings of the 3rd Annual ACM Symposium on Theory of Computing. pp. 151–158.

[17] http://tweetyproject.org

Daniel, L., December 2009. Paraconsistent Probabilistic Reasoning. Ph.D. thesis, L'École Nationale Supérieure des Mines de Paris.

Deng, X., Haarslev, V., Shiri, N., 2007. Measuring Inconsistencies in Ontologies. In: Proceedings of the 4th European conference on The Semantic Web (ESWC '07). Springer International Publishing, Berlin, Heidelberg, pp. 326–340.

Doder, D., Raskovic, M., Markovic, Z., Ognjanovic, Z., 2010. Measures of Inconsistency and Defaults. International Journal of Approximate Reasoning 51, 832–845.

Grant, J., Hunter, A., 2006. Measuring Inconsistency in Knowledgebases. Journal of Intelligent Information Systems 27, 159–184.

Grant, J., Hunter, A., May 2008. Analysing Inconsistent First-Order Knowledgebases. Artificial Intelligence 172 (8-9), 1064–1093.

Grant, J., Hunter, A., 2011. Measuring Consistency Gain and Information Loss in Stepwise Inconsistency Resolution. In: Liu, W. (Ed.), Proceedings of the 11th European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty (ECSQARU'11). No. 6717 in Lecture Notes in Artificial Intelligence. Springer International Publishing, pp. 362–373.

Grant, J., Hunter, A., 2013. Distance-based Measures of Inconsistency. In: Proceedings of the 12th Europen Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty (ECSQARU'13). Vol. 7958 of Lecture Notes in Computer Science. Springer International Publishing, pp. 230–241.

Hansson, S. O., 2001. A Textbook of Belief Dynamics. Kluwer Academic Publishers.

Hunter, A., 2002. Measuring Inconsistency in Knowledge via Quasi-Classical Models. In: Proceedings of the 18th American National Conference on Artificial Intelligence (AAAI'02). MIT Press, pp. 68–73.

Hunter, A., Konieczny, S., 2004. Approaches to Measuring Inconsistent Information. In: Inconsistency Tolerance. Vol. 3300 of Lecture Notes in Computer Science. Springer International Publishing, pp. 189–234.

Hunter, A., Konieczny, S., September 2008. Measuring Inconsistency through Minimal Inconsistent Sets. In: Brewka, G., Lang, J. (Eds.), Proceedings of the Eleventh International Conference on Principles of Knowledge Representation and Reasoning (KR'2008). AAAI Press, Menlo Park, California, Sydney, Australia, pp. 358–366.

Hunter, A., Konieczny, S., July 2010. On the Measure of Conflicts: Shapley Inconsistency Values. Artificial Intelligence 174 (14), 1007–1026.

Isele, R., Umbrich, J., Bizer, C., Harth, A., 2010. LDSpider: An Open-Source Crawling Framework for the Web of Linked Data. In: Proceedings of 9th International Semantic Web Conference (ISWC'10).

Jabbour, S., Ma, Y., Raddaoui, B., 2014a. Inconsistency Measurement Thanks to MUS Decomposition. In: Proceedings of the 13th International Conference on Autonomous Agents and Multiagent Systems (AAMAS'14). pp. 877–884.

Jabbour, S., Ma, Y., Raddaoui, B., Sais, L., 2014b. Prime Implicates Based Inconsistency Characterization. In: Proceedings of the 21st European Conference on Artificial Intelligence (ECAI'14). pp. 1037–1038.

Knight, K. M., 2001. Measuring inconsistency. Journal of Philosophical Logic 31, 77–98.

Knight, K. M., 2002. A Theory of Inconsistency. Ph.D. thesis, University Of Manchester.

Lawrence, D., 1987. Genetic Algorithms and Simulated Annealing. Pitman Publishing.

Liffiton, M. H., Malik, A., 2013. Enumerating Infeasibility: Finding Multiple MUSes Quickly. In: Proceedings of the 10th International Conference on Integration of Artificial Intelligence and Operations Research techniques in Constraint Programming (CPAIOR'13). pp. 160–175.

Liffiton, M. H., Sakallah, K. A., 2005. On Finding All Minimally Unsatisfiable Subformulas. In: Proceedings of the 8th International Conference on Theory and Applications of Satisfiability Testing (SAT'05). Springer International Publishing, pp. 173–186.

Ma, J., Liu, W., Miller, P., 2012. A Characteristic Function Approach to Inconsistency Measures for Knowledge Bases. In: Proceedings of the 6th International Conference on Scaleable Uncertainty Management (SUM'12). pp. 473–485.

Ma, Y., Qi, G., Hitzler, P., Lin, Z., 2007. Measuring Inconsistency for Description Logics Based on Paraconsistent Semantics. In: Proceedings of the 9th European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty (ECSQARU'07). Springer International Publishing, Berlin, Heidelberg, pp. 30–41.

Ma, Y., Qi, G., Xiao, G., Hitzler, P., Lin, Z., 2009. An Anytime Algorithm for Computing Inconsistency Measurement. In: Knowledge Science, Engineering and Management. No. 5914 in Lecture Notes in Computer Science. Springer International Publishing, pp. 29–40.

Ma, Y., Qi, G., Xiao, G., Hitzler, P., Lin, Z., 2010. Computational Complexity and Anytime Algorithm for Inconsistency Measurement. International Journal of Software and Informatics 4 (1), 3–21.

Makinson, D., 2005. Bridges from Classical to Nonmonotonic Logic. College Publications.

McAreavey, K., Liu, W., Miller, P., 2014. Computational Approaches to Finding and Measuring Inconsistency in Arbitrary Knowledge Bases. International Journal of Approximate Reasoning 55, 1659–1693.

Mu, K., Liu, W., Jin, Z., 2011a. A general framework for measuring inconsistency through minimal inconsistent sets. Knowledge and Information Systems 27, 85–114.

Mu, K., Liu, W., Jin, Z., Bell, D. A., 2011b. A Syntax-based Approach to Measuring the Degree of Inconsistency for Belief Bases. International Journal of Approximate Reasoning 52 (7), 978–999.

Mu, K., Wang, K., Wen, L., 2014. Approaches to Measuring Inconsistency for Stratified Knowledge Bases. International Journal of Approximate Reasoning 55, 529–556.

Muiño, D. P., September 2011. Measuring and Repairing Inconsistency in Probabilistic Knowledge Bases. International Journal of Approximate Reasoning 52 (6), 828–840.

Potyka, N., 2014. Linear Programs for Measuring Inconsistency in Probabilistic Logics. In: Proceedings of the 14th International Conference on Principles of Knowledge Representation and Reasoning (KR'14). pp. 568–577.

Previti, A., Marques-Silva, J., 2013. Partial MUS Enumeration. In: Proceedings of the 27th Conference on Artificial Intelligence (AAAI'13). pp. 818–825.

Priest, G., 1979. Logic of Paradox. Journal of Philosophical Logic 8, 219–241.

Qi, G., Hunter, A., 2007. Measuring Incoherence in Description Logic-based Ontologies. In: Proceedings of the 6th International Semantic Web Conference and 2nd Asian Semantic Web Conference (ISWC'07/ASWC'07). Springer International Publishing, Berlin, Heidelberg, pp. 381–394.

Sacramento, E. R., Casanova, M. A., Breitman, K. K., Furtado, A. L., de Macĕdo, J. A. F., Vidal, V. M. P., 2012. Dealing with Inconsistencies in Linked Data Mashups. In: Proceedings of the 16th International Database Engineering and Applications Sysmposium (IDEAS'12). pp. 175–180.

Thimm, M., April 2013. Inconsistency Measures for Probabilistic Logics. Artificial Intelligence 197, 1–24.

Thimm, M., August 2014a. Coherence and Compatibility of Markov Logic Networks. In: Proceedings of the 21st European Conference on Artificial Intelligence (ECAI'14). pp. 891–896.

Thimm, M., September 2014b. Towards Large-scale Inconsistency Measurement. In: Proceedings of the 37th German Conference on Artificial Intelligence (KI'14). pp. 195–206.

Thimm, M., 2014c. Tweety - A Comprehensive Collection of Java Libraries for Logical Aspects of Artificial Intelligence and Knowledge Representation. In: Proceedings of the 14th Int. Conference on Principles of Knowledge Representation and Reasoning (KR'14). pp. 528–537.

Xiao, G., Lin, Z., Ma, Y., Qi, G., 2010. Computing Inconsistency Measurements under Multi-Valued Semantics by Partial Max-SAT Solvers. In: Proceedings of the Twelfth International Conference on the Principles of Knowledge Representation and Reasoning (KR'10). pp. 340–349.

Xiao, G., Ma, Y., 2012. Inconsistency Measurement based on Variables in Minimal Unsatisfiable Subsets. In: Proceedings of the 20th European Conference on Artificial Intelligence (ECAI'12). pp. 864–869.

Zhou, L., Huang, H., Qi, G., Ma, Y., Huang, Z., Qu, Y., 2009. Measuring Inconsistency in DL-Lite Ontologies. In: Proceedings of the 2009 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT'09). IEEE Computer Society, Washington, DC, USA, pp. 349–356.

Appendix A. Proofs of Technical Results

Proposition 1. *Let $\mathcal{K}$ be a knowledge base. The following two statements are equivalent:*

1. *there is no $\phi \in \mathcal{K}$ with $\phi \models \perp$*
2. *there exists a hitting set H of $\mathcal{K}$*

Proof. Let $\mathcal{K} = \{\phi_1, \ldots, \phi_n\}$. First, assume that there is no $\phi_i \in \mathcal{K}$ with $\phi_i \models \perp$ for $i = 1, \ldots, n$. It follows $\mathsf{Mod}(\phi_i) \neq \emptyset$ for every $i = 1, \ldots, n$. Let $\omega_i \in \mathsf{Mod}(\phi_i)$, then by definition $\{\omega_1, \ldots, \omega_n\}$ is a hitting set of $\mathcal{K}$. Let now $H = \{\omega_1, \ldots, \omega_m\}$ be a hitting set of $\mathcal{K}$. Then for every $\phi \in \mathcal{K}$ there is $\omega \in H$ with $\omega \models \phi$. Therefore there can be no $\phi \in \mathcal{K}$ with $\mathsf{Mod}(\phi) \neq \emptyset$. □

Proposition 2. *Let $\mathcal{K}$ be a knowledge base.*

1. *If H is a hitting set of $\mathcal{K}$ then every H' with $H \subseteq H'$ is a hitting set of $\mathcal{K}$.*
2. *$H = \emptyset$ is a hitting set of $\mathcal{K}$ if and only if $\mathcal{K} = \emptyset$.*
3. *$\mathcal{K}$ is consistent if and only if there is a hitting set H of $\mathcal{K}$ with $|H| = 1$.*
4. *If H is a hitting set of $\mathcal{K}$ then H is a hitting set of every $\mathcal{K}'$ with $\mathcal{K}' \subseteq \mathcal{K}$.*

Proof. Let $\mathcal{K} = \{\phi_1, \ldots, \phi_n\}$.

1. Let H be a hitting set of $\mathcal{K}$ and let H' be such that $H \subseteq H'$. Then for every $\phi \in \mathcal{K}$ we have $\omega \in H \subseteq H'$ such that $\omega \models \phi$. Therefore H' is a hitting set of $\mathcal{K}$.
2. Let $\mathcal{K} = \emptyset$. Then $H = \emptyset$ is a trivial hitting set of $\mathcal{K}$ by definition of universal quantification. Note also that for any $\mathcal{K}$ with $\mathcal{K} \neq \emptyset$ the set $H = \emptyset$ cannot be a hitting set.
3. Let $\mathcal{K} \neq \emptyset$ be consistent. Then there is $\omega \in \mathsf{Int}(\mathsf{At})$ with $\omega \models \mathcal{K}$, i. e., $\omega \models \phi$ for every $\phi \in \mathcal{K}$. Therefore, $\{\omega\}$ is a hitting set of $\mathcal{K}$ with $|\{\omega\}| = 1$. Let H be any hitting set of $\mathcal{K}$ with $|H| = 1$, i. e., $H = \{\omega\}$. Then $\omega \models \phi$ for all $\phi \in \mathcal{K}$ and, hence, $\phi \models \mathcal{K}$. Therefore, $\mathcal{K}$ is consistent. For the case $\mathcal{K} = \emptyset$ note that every subset of $\mathsf{Int}(\mathsf{At})$ is a hitting set of $\mathcal{K}$.
4. Let H be a hitting set of $\mathcal{K}$ and let $\mathcal{K}' \subseteq \mathcal{K}$. Then for every $\phi \in \mathcal{K}'$ there is $\omega \in H$ with $\omega \models \phi$ as $\mathcal{K}' \subseteq \mathcal{K}$. Hence, H is a hitting set of $\mathcal{K}$. □

Proposition 3. *The function $\mathcal{I}_{hs}$ is a (basic) inconsistency measure.*

Proof. We have to show that properties 1.), 2.), and 3.) of Definition 4 are satisfied.

1. This follows directly from items 2.) and 3.) of Proposition 2.
2. This follows directly from item 4.) of Proposition 2.
3. Let $\alpha \in \mathsf{Free}(\mathcal{K})$ and define $\mathcal{K}' = \mathcal{K} \setminus \{\alpha\}$. Let H be a hitting set of $\mathcal{K}'$ with $|H|$ being minimal and let $\omega \in H$. Furthermore, let $\mathcal{K}'' \subseteq \mathcal{K}'$ be the set of all formulæ β such that $\omega \models \beta$. It follows that $\mathcal{K}''$ is consistent. As α is a free formula it follows that $\mathcal{K}'' \cup \{\alpha\}$ is also consistent (otherwise there would be a minimal inconsistent subset of $\mathcal{K}''$ containing α). Let ω' be a model of $\mathcal{K}'' \cup \{\alpha\}$. Then $H' = (H \setminus \{\omega\}) \cup \{\omega'\}$ is a hitting set of $\mathcal{K}$ and due to 2.) also of minimal cardinality. Hence, we have $\mathcal{I}_{hs}(\mathcal{K}') = \mathcal{I}_{hs}(\mathcal{K})$. □

Proposition 4. *The measure $\mathcal{I}_{hs}$ satisfies the following properties:*

- *If $\alpha \in \mathcal{K}$ is such that $\mathsf{At}(\alpha) \cap \mathsf{At}(\mathcal{K} \setminus \{\alpha\}) = \emptyset$ then $\mathcal{I}_{hs}(\mathcal{K}) = \mathcal{I}_{hs}(\mathcal{K} \setminus \{\alpha\})$ (*safe formula independence*).*
- *If $\mathcal{K} \equiv^{\sigma} \mathcal{K}'$ then $\mathcal{I}_{hs}(\mathcal{K}) = \mathcal{I}_{hs}(\mathcal{K}')$ (*irrelevance of syntax*).*
- *If $\alpha \models \beta$ and $\alpha \not\models \perp$ then $\mathcal{I}_{hs}(\mathcal{K} \cup \{\alpha\}) \geq \mathcal{I}_{hs}(\mathcal{K} \cup \{\beta\})$ (*dominance*).*

Proof.

- This is satisfied as safe formula independence follows from free formula independence, cf. (Hunter and Konieczny, 2010; Thimm, 2013).
- Let H be a hitting set of $\mathcal{K}$ with minimal cardinality. So, for every $\alpha \in \mathcal{K}$ we have $\omega \in H$ with $\omega \models \alpha$. Due to $\alpha \equiv \sigma(\alpha)$ we also have $\omega \models \sigma(\alpha)$ and, thus for very $\beta \in \mathcal{K}'$ we have $\omega \in H$ with $\omega \models \beta$. So H is also a hitting set of $\mathcal{K}'$. Minimality follows from the fact that σ is a bijection.
- Let H be a minimal hitting set of $\mathcal{K}_1 = \mathcal{K} \cup \{\alpha\}$ with minimal cardinality and let $\omega \in H$ be such that $\omega \models \alpha$. Then we also have that $\omega \models \beta$ and H is also a hitting set of $\mathcal{K}_2 = \mathcal{K} \cup \{\beta\}$. Hence, $\mathcal{I}_{hs}(\mathcal{K}_1) \geq \mathcal{I}_{hs}(\mathcal{K}_2)$. □

Proposition 5. *For every knowledge base $\mathcal{K}$*

$$\mathcal{I}_{hs}(\mathcal{K}) = \min\{|\Phi| \mid \Phi \textit{ is a consistent partitioning of } \mathcal{K}\} - 1$$

with $\min \emptyset = \infty$ *for every* $\mathcal{K} \in \mathbb{K} \setminus \{\emptyset\}$ *and* $\mathcal{I}_{hs}(\emptyset) = 0$.

Proof. For $\mathcal{K} = \emptyset$ and the case that $\mathcal{K}$ contains ϕ with $\phi \models \perp$, the statement is trivially satisfied so assume $\mathcal{K} \neq \emptyset$ and that $\mathcal{K}$ does not contain an inconsistent formula. Let $\Phi = \{\Phi_1, \ldots, \Phi_n\}$ be a consistent partitioning with $|\Phi|$ being minimal and let $\omega_i \in \mathsf{Int}(\mathsf{At})$ be such that $\omega_i \models \Phi_i$ (for $i = 1, \ldots, n$). Observe that $\omega_i \neq \omega_j$ for all $i \neq j$, otherwise $\Phi_i \cup \Phi_j$ would have a model $\omega_i = \omega_j$ and $\Phi' = \Phi \setminus \{\Phi_i, \Phi_j\} \cup \{\Phi_i \cup \Phi_j\}$ would be a consistent partitioning with $|\Phi'| < |\Phi|$. Then $H = \{\omega_1, \ldots, \omega_n\}$ is a hitting set of $\mathcal{K}$ and we have $\mathcal{I}_{hs}(\mathcal{K}) \leq |\Phi| - 1$. Let now $H = \{\omega_1, \ldots, \omega_n\}$ be a hitting set of $\mathcal{K}$ with $|H|$ being minimal. Let $\Phi = \{\Phi_1, \ldots, \Phi_n\}$ be a set such that $\phi \in \Phi_i$ implies $\omega_i \models \phi$ for every $\phi \in \mathcal{K}$ (note that there may be multiple partitionings satisfying this property but they all have the same cardinality). Note that Φ is a partitioning of $\mathcal{K}$ and that Φ_i is consistent for every $i = 1, \ldots, n$. It follows $\mathcal{I}_{hs}(\mathcal{K}) \geq |\Phi| - 1$ and therefore the claim. □

Proposition 6. *Let $\mathcal{K}$ be a knowledge base. If* $\infty > \mathcal{I}_{hs}(\mathcal{K}) > 0$ *then*

$$\mathcal{I}_{\eta}(\mathcal{K}) \leq 1 - \frac{1}{\mathcal{I}_{hs}(\mathcal{K}) + 1}$$

Proof. Let H be a minimal hitting set of $\mathcal{K}$ with minimal cardinality, i. e., we have $\mathcal{I}_{hs}(\mathcal{K}) = |H| - 1$. Define a probability function $P : \mathsf{Int}(\mathsf{At}) \to [0, 1]$ via $P(\omega) = 1/|H|$ for every $\omega \in H$ and $P(\omega') = 0$ for every $\omega' \in \mathsf{Int}(\mathsf{At}) \setminus H$ (note that P is indeed a probability function). As H is a hitting set of $\mathcal{K}$ we have that $P(\phi) \geq 1/|H|$ for every $\phi \in \mathcal{K}$ as at least one model of ϕ gets probability $1/|H|$ in P. So we have $\mathcal{I}_{\eta}(\mathcal{K}) \leq 1 - 1/|H| = 1 - 1/(\mathcal{I}_{hs}(\mathcal{K}) + 1)$. □

Proposition 7. *There is no subsumption relation between $\mathcal{I}_{hs}$ and any $\mathcal{I} \in \{\mathcal{I}_{\mathsf{MI}}, \mathcal{I}_{\mathsf{MI}^c}, \mathcal{I}_c, \mathcal{I}_{\eta}\}$.*

Proof.

1. $\mathcal{I}_{hs} \not\sqsubseteq \mathcal{I}_{\mathsf{MI}}$: Consider the knowledge bases $\mathcal{K}_1$ and $\mathcal{K}_2$ given as

$$\mathcal{K}_8 = \{a \wedge b, \neg a \wedge b, a \wedge \neg b\}$$
$$\mathcal{K}_9 = \{a, b, c, \neg a \wedge \neg b \wedge \neg c\}$$

   Then we have $\mathcal{I}_{hs}(\mathcal{K}_9) = 1 < 2 = \mathcal{I}_{hs}(\mathcal{K}_8)$ but $\mathcal{I}_{\mathsf{MI}}(\mathcal{K}_9) = 3 = \mathcal{I}_{\mathsf{MI}}(\mathcal{K}_8)$.

2. $\mathcal{I}_{\mathsf{MI}} \not\sqsubseteq \mathcal{I}_{hs}$: Consider the knowledge bases $\mathcal{K}_4$ and $\mathcal{K}_{10}$ given as

$$\mathcal{K}_4 = \{a, \neg a\}$$
$$\mathcal{K}_{10} = \{a, \neg a \wedge \neg b, b\}$$

Then we have $\mathcal{I}_{\mathsf{MI}}(\mathcal{K}_4) = 1 < 2 = \mathcal{I}_{\mathsf{MI}}(\mathcal{K}_{10})$ but $\mathcal{I}_{hs}(\mathcal{K}_4) = 1 = \mathcal{I}_{hs}(\mathcal{K}_{10})$.
3. $\mathcal{I}_{hs} \not\sqsubseteq \mathcal{I}_{\mathsf{MI}^c}$: For the knowledge bases from item 1.) we also have $\mathcal{I}_{\mathsf{MI}}(\mathcal{K}_8) = 1.5 = \mathcal{I}_{\mathsf{MI}}(\mathcal{K}_9)$.
4. $\mathcal{I}_{\mathsf{MI}^c} \not\sqsubseteq \mathcal{I}_{hs}$: For the knowledge bases from item 2.) we also have $\mathcal{I}_{\mathsf{MI}^c}(\mathcal{K}_4) = 1/2 < 1 = \mathcal{I}_{\mathsf{MI}^c}(\mathcal{K}_{10})$.
5. $\mathcal{I}_{hs} \not\sqsubseteq \mathcal{I}_c$: see Example 6.
6. $\mathcal{I}_c \not\sqsubseteq \mathcal{I}_{hs}$: see Example 6.
7. $\mathcal{I}_{hs} \not\sqsubseteq \mathcal{I}_\eta$: Consider the knowledge bases $\mathcal{K}_4$ and $\mathcal{K}_{11}$ given as

$$\begin{aligned}
\mathcal{K}_4 = \{&a, \neg a\}\\
\mathcal{K}_{11} = \{&(a \wedge b \wedge c) \vee (a \wedge b \wedge \neg c) \vee (a \wedge \neg b \wedge c),\\
&(a \wedge b \wedge c) \vee (a \wedge b \wedge \neg c) \vee (a \wedge \neg b \wedge \neg c),\\
&(a \wedge b \wedge c) \vee (a \wedge b \wedge \neg c) \vee (\neg a \wedge b \wedge c),\\
&(a \wedge b \wedge c) \vee (a \wedge \neg b \wedge c) \vee (a \wedge \neg b \wedge \neg c),\\
&(a \wedge b \wedge c) \vee (a \wedge \neg b \wedge c) \vee (\neg a \wedge b \wedge c),\\
&(a \wedge b \wedge c) \vee (a \wedge \neg b \wedge \neg c) \vee (\neg a \wedge b \wedge c),\\
&(a \wedge b \wedge \neg c) \vee (a \wedge \neg b \wedge c) \vee (a \wedge \neg b \wedge \neg c),\\
&(a \wedge b \wedge \neg c) \vee (a \wedge \neg b \wedge c) \vee (\neg a \wedge b \wedge c),\\
&(a \wedge b \wedge \neg c) \vee (a \wedge \neg b \wedge \neg c) \vee (\neg a \wedge b \wedge c),\\
&(a \wedge \neg b \wedge c) \vee (a \wedge \neg b \wedge \neg c) \vee (\neg a \wedge b \wedge c)\}
\end{aligned}$$

Then we have $\mathcal{I}_{hs}(\mathcal{K}_4) = 1 < 2 = \mathcal{I}_{hs}(\mathcal{K}_{11})$ but $\mathcal{I}_\eta(\mathcal{K}_4) = 1/2 > 2/5 = \mathcal{I}_\eta(\mathcal{K}_{11})$. Let us discuss $\mathcal{K}_{11}$ a bit more. Consider the five interpretations $\omega_1, \ldots, \omega_5$ of the propositional signature $\mathsf{At} = \{a, b, c\}$ defined via

$$\omega_1 = abc$$
$$\omega_2 = ab\overline{c}$$
$$\omega_3 = a\overline{b}c$$
$$\omega_4 = a\overline{b}\overline{c}$$
$$\omega_5 = \overline{a}bc$$

Then $\mathcal{K}_{11}$ comprises of formulas ϕ that are satisfied by exactly three out of these five interpretations (and for each 3-element subset of $\{\omega_1, \ldots, \omega_5\}$ there is exactly one corresponding formula). It follows that a probability function P assigning probability $1/5$ to each of these five interpretations (and zero to the remaining interpretations) yields $P(\phi) = 3/5$ for each $\phi \in \mathcal{K}_2$ (and this is maximal), thus yielding $\mathcal{I}_\eta(\mathcal{K}_{11}) = 1 - 3/5 = 2/5$. Further, any 3-element subset of $\{\omega_1, \ldots, \omega_5\}$ is also a hitting set of $\mathcal{K}_{11}$: as every $\phi \in \mathcal{K}_2$ is satisfied by exactly three interpretations, one can remove any two of them and still maintain the hitting set property. So $H = \{\omega_1, \omega_2, \omega_3\}$ is a hitting set and one can easily see that there is no smaller one, yielding $\mathcal{I}_{hs}(\mathcal{K}_{11}) = |H| - 1 = 2$.

8. $\mathcal{I}_\eta \not\sqsubseteq \mathcal{I}_{hs}$: Consider the knowledge bases $\mathcal{K}_4$ and $\mathcal{K}_{12}$ given as

$$\mathcal{K}_4 = \{a, \neg a\}$$
$$\mathcal{K}_{12} = \{a, b, \neg a \vee \neg b\}$$

Then we have $\mathcal{I}_\eta(\mathcal{K}_{12}) = 1/3 < 1/2 = \mathcal{I}_\eta(\mathcal{K}_4)$ but $\mathcal{I}_{hs}(\mathcal{K}_{12}) = 1 = \mathcal{I}_{hs}(\mathcal{K}_4)$. □

Corollary 1. *$\mathcal{I}_{hs} \not\sim \mathcal{I}_{\mathsf{MI}}$, $\mathcal{I}_{hs} \not\sim \mathcal{I}_{\mathsf{MI}^c}$, $\mathcal{I}_{hs} \not\sim \mathcal{I}_c$, and $\mathcal{I}_{hs} \not\sim \mathcal{I}_\eta$.*

Proof. This follows directly from Proposition 7 and the definition of equivalence. □

Proposition 8. *Let $\mathcal{I}$ be an inconsistency measure, $w \in \mathbb{N} \cup \{\infty\}$, and g an aggregation function.*

1. *If w is finite then $\mathcal{J}_\mathcal{I}^{w,g}$ is* not *an approximation of $\mathcal{I}$.*
2. *$\mathcal{J}_\mathcal{I}^{w,g}(\mathcal{S}_\mathcal{K}, i) \leq \mathcal{I}(\mathcal{K})$ for every $\mathcal{K} \in \mathbb{K}$ and $i \in \mathbb{N}$.*
3. *If $w = \infty$ and $g(x, y) \geq (x + y)/2$ then $\mathcal{J}_\mathcal{I}^{w,g}$ is an approximation of $\mathcal{I}$.*

Proof.

1. Assume $\mathcal{K}$ is a minimal inconsistent set with $|\mathcal{K}| > w$. Then $\mathcal{I}(\mathcal{S}^{\max\{0,i-w\},i}) = 0$ for all $i > 0$ (as every subset of $\mathcal{K}$ is consistent) and $\mathcal{J}_\mathcal{I}^{w,g}(\mathcal{S}, i) = 0$ for all $i > 0$ as well. As $\mathcal{I}$ is an inconsistency measure $\mathcal{I}(\mathcal{K}) > 0$ and, hence, $\mathcal{J}_\mathcal{I}^{w,g}$ does not approximate $\mathcal{I}$.
2. This follows from the fact that $\mathcal{I}$ is a basic inconsistency measure and therefore satisfies $\mathcal{I}(\mathcal{K}) \leq \mathcal{I}(\mathcal{K}')$ for $\mathcal{K} \subseteq \mathcal{K}'$.

3. If $w = \infty$ there is $i_0 \in \mathbb{N}$ such that $\mathcal{I}(\mathcal{S}^{\max\{0,i-w\},i}) = \mathcal{I}(\mathcal{K})$ for all $i > i_0$. Due to item 2 above (all previous values estimated the inconsistency value from below) and as $g(x, y) > (x + y)/2$ (in each step the new value is the average of the previous value and the actual inconsistency value) the value $\mathcal{I}(\mathcal{K})$ will be approximated by $\mathcal{J}_{\mathcal{I}}^{w,g}$ eventually. □

Proposition 9. *For every $p \in [0, 1)$, g some aggregation function with $g(x, y) \geq (x + y)/2$, $f : \mathbb{N} \to [0, 1]$ a null-bound function, and $\mathcal{K} \in \mathbb{K}$ there is $m \in \mathbb{N}$ such that with probability greater or equal p it is the case that $\lim_{i\to\infty} \mathcal{J}_{hs}^{m,g,f}(\mathcal{S}_{\mathcal{K}}, i) = \mathcal{I}_{hs}(\mathcal{K})$.*

Proof. Let $p \in [0, 1)$, g some aggregation function with $g(x, y) \geq (x + y)/2$, $f : \mathbb{N} \to [0, 1]$ a null-bound function, and $\mathcal{K} \in \mathbb{K}$. Let $H = \{\omega_1, \ldots, \omega_h\} \subseteq \mathsf{Int}(\mathsf{At})$ be a hitting set of $\mathcal{K}$ such that $\mathcal{I}_{hs}(\mathcal{K}) = |H| - 1$. Consider the evolution of a single candidate set during the iterated execution of $\texttt{update}_{hs}^{m,g,f}(form)$. If $\mathcal{K}$ contains a contradictory formula then lines 2–5 ensure that the return value of Algorithm 2 is always ∞ and thus the claim holds trivially. We now assume that $\mathcal{K}$ contains no contradictory formula.

Let $C_0 = \emptyset$ be the initial candidate set and let C_i for $i \in \mathbb{N}$ denote the candidate set after iteration i. In the first iteration, C_0 does not contain any interpretations yet, so lines 9 and 10 of Algorithm 2 are vacuous. As the condition in line 11 evaluates to true, we add some interpretation to C_0. As H is a hitting set of $\mathcal{K}$ there is $\omega \in H$ with $\omega \in \mathsf{Mod}(form)$. The probability of choosing ω in line 12, so the probability of C_0 evolving to $C_1 = \{\omega\}$, is $p_0 = 1/|\mathsf{Mod}(form)| > 0$. In the second iteration, the probability q_1 that line 10 is *not* executed is greater than zero (as f is bounded by 1 and the condition involves a strictly less comparison). Assume that $\omega \not\models form$ (otherwise simply continue with the next formula in the next iteration). Then, again, there is $\omega' \in H$ with $\omega' \models form$ and the probability of choosing ω' in line 12 is $1/|\mathsf{Mod}(form)| > 0$. Therefore, the probability of C_0 evolving through C_1 to $C_2 = \{\omega, \omega'\}$ is $p_1 = p_0 q_1/|\mathsf{Mod}(form)| > 0$. It follows that the probability of C_0 evolving to $C_h = H$ in its h-th iteration is strictly greater than zero. Note that beginning in the $h + 1$-th iteration the condition in line 11 is not satisfied anymore and that $C_i = C_{i+1}$ for every $i > h$ with positive probability as well as there is a positive probability that line 10 will not be executed (with increasing probability over the iterations as f is a null-bound function). So for every candidate set $C \in Cand$ there is a positive probability $\hat{p}$ that C evolves to H and does not change anymore thereafter.

In general, observe that every candidate set $C \in Cand$ evolves into a hitting set of $\mathcal{K}$ as the probability of executing line 10 becomes zero eventually and lines

11 and 12 ensure that every formula in $\mathcal{K}$ has a model in C. Furthermore, each evolution of a candidate set $C \in Cand$ is an independent random process. So for $|Cand| = m$ the probability that at least one element of $Cand$ evolves to H in the above described manner is $1-(1-\hat{p})^m$ (here $(1-\hat{p})^m$ is the probability that none of the candidate sets evolve to H). Observe that $1-(1-\hat{p})^m$, due to $\hat{p} > 0$, is monotonously increasing in m with $\lim_{m\to\infty} 1-(1-\hat{p})^m = 1$. Therefore we can choose $\hat{m}$ such that $1-(1-\hat{p})^{\hat{m}} \geq p \in [0,1)$ and with probability at least p the set of candidate sets $Cand$ with $|Cand| = \hat{m}$ contains at least one candidate set that evolves to H. Let $C_0, C_1, \ldots, C_h, C_{h+1}, \ldots$ be the evolution of this candidate set with $C_0 \subseteq C_1 \ldots \subseteq C_h = C_{h+1} = \ldots$ with $C_h = H$. It follows that the variable $newValue$ has never a value larger than $|H|-1$ whenever line 14 is executed. As $currentValue$ is initialized with 0 it also follows that $currentValue$ has never a larger value than $|H|-1$ in line 15, as g is an aggregation function.

Consider now an iteration step i where all candidate sets in $Cand$ have stabilized and do not change thereafter. Then $newValue$ always has the value $|H|-1$ in line 14. If at iteration step i $currentValue$ has already the value $|H|-1$ then the value of $currentValue$ is not changed (as g is an aggregation function). Then we have that the return value of Algorithm 2 in line 15 is always $|H|-1 = \mathcal{I}_{hs}(\mathcal{K})$ and therefore $\lim_{i\to\infty} \mathcal{J}_{hs}^{m,g,f}(\mathcal{S}_{\mathcal{K}}, i) = \mathcal{I}_{hs}(\mathcal{K})$. If at iteration step i $currentValue$ has a value $\alpha_0 < |H|-1$ then observe that $currentValue$ is updated to some value $\alpha_1 \geq (\alpha_0 + (|H|-1))/2$ with $\alpha_1 \leq |H|-1$ (as g is an aggregation function). In subsequent iterations this value is updated while satisfying $\alpha_j \geq (\alpha_{j+1} + (|H|-1))/2$ with $\alpha_j \leq |H|-1$ which converges to $|H|-1$ and thus proves the claim. □

Proposition 10. *For every $p \in [0,1)$, g some aggregation function with $g(x,y) \geq (x+y)/2$, $f : \mathbb{N} \to [0,1]$ a null-bound function, and $\mathcal{K} \in \mathbb{K}$ there is $m \in \mathbb{N}$ such that with probability greater or equal p it is the case that $\lim_{i\to\infty} \mathcal{J}_c^{m,g,f}(\mathcal{S}_{\mathcal{K}}, i) = \mathcal{I}_c(\mathcal{K})$.*

Proof. Let $p \in [0,1)$, g some aggregation function with $g(x,y) \geq (x+y)/2$, $f : \mathbb{N} \to [0,1]$ a null-bound function, and $\mathcal{K} \in \mathbb{K}$. Let $\hat{\upsilon} : \mathsf{At} \to \{T, F, B\}$ be a three-valued interpretation of the atoms appearing in $\mathcal{K}$ such that $\hat{\upsilon} \models^3 \mathcal{K}$ and $\mathcal{I}_c(\mathcal{K}) = |\hat{\upsilon}^{-1}(B)| = t$. Consider the evolution of a single three-valued interpretation $\upsilon \in Cand$ during the iterated execution of $\texttt{update}_c^{m,g,f}(form)$.

Let υ_0 be the initial interpretation with $\upsilon^{-1}(B) = \emptyset$ set and let υ_i for $i \in \mathbb{N}$ denote the interpretation after iteration i. As f is null-bound there is an iteration $k > 0$ from which on line 6 is not executed anymore. Furthermore, observe that lines 7–13 ensure that υ is changed in such a way that it satisfies the formula $form$. Note that once we are in an iteration $k' \geq k$ and υ satisfies all formulas in $\mathcal{K}$ lines 7–13 will also not be executed anymore. Consequently, the evolution of υ always converges at some iteration $l > 0$ and υ_l satisfies all formulas in $\mathcal{K}$. Similarly to the analysis in the proof of Proposition 9 the probability of υ evolving to $\hat{\upsilon}$, i. e., $\upsilon_l = \hat{\upsilon}$, is strictly greater than zero (albeit potentially quite small). In particular, the probability $\hat{p}$ of the evolution $\upsilon_0, \upsilon_1, \ldots, \upsilon_t, \ldots$ with $|\upsilon_0^{-1}(B)| = 0$,$|\upsilon_1^{-1}(B)| = 1$,..., $|\upsilon_t^{-1}(B)| = t$, and $\upsilon_{j+1} = \upsilon_j = \hat{\upsilon}$ for all $j \geq t$ is strictly greater zero. So for $|Cand| = m$ the probability that at least one element of $Cand$ evolves to $\hat{\upsilon}$ in the above described manner is $1-(1-\hat{p})^m$ (here $(1-\hat{p})^m$ is the probability that none of the interpretations evolve to $\hat{\upsilon}$). Therefore we can choose $\hat{m}$ such that $1 - (1 - \hat{p})^{\hat{m}} \geq p \in [0, 1)$ and with probability at least p the set $Cand$ with $|Cand| = \hat{m}$ contains at least one interpretation that evolves to $\hat{\upsilon}$. It follows that the variable $newValue$ has never a value larger than t whenever line 15 is executed. As $currentValue$ is initialized with 0 it also follows that $currentValue$ has never a larger value than t in line 16, as g is an aggregation function.

Consider now an iteration step i where all interpretations in $Cand$ have stabilized and do not change thereafter. Then $newValue$ always has the value t in line 15. If at iteration step i $currentValue$ has already the value t then the value of $currentValue$ is not changed (as g is an aggregation function). Then we have that the return value of Algorithm 4 in line 16 is always $t = \mathcal{I}_c(\mathcal{K})$ and therefore $\lim_{i\to\infty} \mathcal{J}_c^{m,g,f}(\mathcal{S}_{\mathcal{K}}, i) = \mathcal{I}_c(\mathcal{K})$. If at iteration step i $currentValue$ has a value $\alpha_0 < t$ then observe that $currentValue$ is updated to some value $\alpha_1 \geq (\alpha_0 + t)/2$ with $\alpha_1 \leq t$ (as g is an aggregation function). In subsequent iterations this value is updated while satisfying $\alpha_j \geq (\alpha_{j+1} + t))/2$ with $\alpha_j \leq t$ which converges to t and thus proves the claim. $\square$