Using Shapley Inconsistency Values for Distributed Information Systems with Uncertainty

John Grant[1] and Anthony Hunter[2]

[1] Department of Computer Science and UMIACS, University of Maryland, College Park, MD 20742, USA
[2] Department of Computer Science, University College London, Gower Street, London WC1E 6BT, UK

Abstract. We study the problem of analyzing inconsistency in a distributed information system where the reliability of the sources is taken into account. We model uncertainty by assigning a probability to each source. This yields a definition of the expected inconsistency of the system. We also extend this with the use of Shapley values for determining the responsibility of each formula to inconsistency. Then we use the Shapley inconsistency values to assign an expected blame to each formula. From this we define the concept of weakness of a formula which represents the degree to which it should be deleted to resolve the inconsistency of the system.

1 Introduction

The general problem that we consider in this paper is the aggregation of information from multiple distributed sources (e.g. databases, information from the web, etc). As a user, we ask queries of the sources and as a result we get answers. We do not control the sources, and we cannot change them. Our primary concern is to evaluate the answers we get back from the sources by considering the inconsistency between them with respect to integrity constraints that we may have.

We let $(K_1, ..., K_n)$ denote a tuple of sources of information, where each K_i is a set of formulae. We do not necessarily know the contents of each K_i; however, we can query each source. We assume that we have a priori a set of integrity constraints based on the context. Suppose we ask a question Q, and we get the answer A_i from source i (i.e. $K_i \vdash A_i$). We assume each A_i is a nonempty set of facts (i.e. a set of atoms or propositional letters). Then, for any question Q, there is an answer tuple $(A_1, ..., A_n)$. We do not formalize the query process in this paper, and so our starting point is the set of integrity constraints and the answer tuple.

Example 1. Suppose we are searching the web on information about Paris. From source 1, we get the facts listed in A_1 below, and from source 2, we get the facts listed in A_2. So for that query, we have the answer tuple (A_1, A_2) where

$$A_1 = \{\texttt{population}(\texttt{7million}), \texttt{medianage}(\texttt{45})\}$$
$$A_2 = \{\texttt{population}(\texttt{4million}), \texttt{averagesalary}(\texttt{23KEuro})\}$$

We assume first-order predicate logic for our language for integrity constraints; however, we will rewrite the integrity constraints to suit what might appear in the answer tuples. Thus, $\forall \mathtt{x} \forall \mathtt{y}(\mathtt{population}(\mathtt{x}) \land \mathtt{population}(\mathtt{y}) \rightarrow \mathtt{x} = \mathtt{y})$ might be an integrity contraint for the first example. In this example we will use the instantiated version as $\neg\mathtt{population}(7\mathtt{million}) \lor \neg\mathtt{population}(4\mathtt{million})$. So we will assume that each integrity constraint is a disjunction of negated atoms and write it as A_0.

Given an answer tuple $(A_1, ..., A_n)$ and the corresponding set of integrity constraints A_0, we will be interested in the consistency of $\cup_{i=0}^{n} A_i$, that is, whether $\cup_{i=0}^{n} A_i \vdash \perp$ where $\vdash$ denotes the classical consequence relation and $\perp$ stands for falsity. Thus, Example 1 with the given integrity constraint is inconsistent.

Given an answer tuple, and a set of integrity constraints, we want to be able to resolve inconsistencies by removing facts from answers. Our goal is to find the formulae that are for some good reasons the best to eliminate in order to restore consistency. To support this process, we will use measures of inconsistency. We will review these in the next section, but essentially, they assess the number of conflicts, the inter-connectedness of conflicts, the proportion of the information that is in conflict, etc.

In order to help analyse the conflict, we will also take the reliability of the sources of information into account. Let P be a function that assigns a value in $[0, 1]$ to each source $i \in \{1, .., n\}$. We assume that $P(i)$ denotes the probability that a randomly selected formula in A_i is correct based on previous performance by the source where the previous performance is determined from the correctness of previous answers when checked by an oracle / expert / etc. We also assign $P(A_0) = 1$, that is, all integrity constraints are known to be correct.

We formalize a novel approach to analyzing inconsistency by using probabilistic information about sources of information in conjunction with measures of inconsistency.

2 Background to measuring inconsistency

We assume a propositional language $\mathcal{L}$ of formulae composed from a set of atoms $\mathcal{A}$ and the logical connectives $\land$, $\lor$, $\neg$. A knowledgebase K is a finite set of formulae. We let $\vdash$ denote the classical consequence relation, and write $K \vdash \perp$ to denote that K is inconsistent. $\mathcal{R}^{\geq 0}$ is the set of nonnegative real numbers.

For a knowledgebase K, $\mathsf{MI}(K)$ is the set of minimal inconsistent subsets of K. $\mathsf{Free}(K)$ is the set of formulae not involved in any inconsistency and $\mathsf{Problematic}(K)$ is $K \setminus \mathsf{Free}(K)$. For one of the inconsistency measures we will use we define a semantics that uses Priest's three valued logic (3VL) [7] with the classical two valued semantics augmented by a third truth value, B (for both), denoting inconsistency. The truth values for the connectives are defined in the following table.

α	T	T	T	B	B	B	F	F	F
β	T	B	F	T	B	F	T	B	F
$\alpha \lor \beta$	T	T	T	T	B	B	T	B	F
$\alpha \land \beta$	T	B	F	B	B	F	F	F	F
$\neg\alpha$	F	F	F	B	B	B	T	T	T

An interpretation i is a function that assigns to each atom in K one of the three truth values: $i : \mathsf{Atoms}(K) \to \{F, B, T\}$. For an interpretation i the atoms that are assigned the truth value B represent the inconsistency for which we obtain $\mathsf{Conflictbase}(i) = \{\alpha \mid i(\alpha) = B\}$. A model of K is an interpretation where no formula is assigned the truth value F: $\mathsf{Models}(K) = \{i \mid \text{ for all } \phi \in K, i(\phi) = T \text{ or } i(\phi) = B\}$ Then, as a measure of inconsistency for K we define $\mathsf{Contension}(K) = \mathsf{Min}\{|\mathsf{Conflictbase}(i)| \mid i \in \mathsf{Models}(K)\}$. So the contension gives the minimal number of atoms that must be assigned B in order to get a 3VL model of K.

Example 2. For $K = \{a, \neg a, a \vee b, \neg b\}$, there are two models of K, i_1 and i_2, where $i_1(a) = B$, $i_1(b) = B$, $i_2(a) = B$, and $i_2(b) = F$. Therefore, $\mathsf{Conflictbase}(i_1) = 2$ and $\mathsf{Conflictbase}(i_2) = 1$. Hence, $\mathsf{Contension}(K) = 1$.

An inconsistency measure assigns a nonnegative real value to every knowledgebase. We assume several requirements for inconsistency measures [4]. The conditions ensure that all and only consistent knowledgebases get measure 0, the measure is monotonic for subsets, the removal of a formula that does not participate in an inconsistency leaves the measure unchanged, and the addition of a logically weaker formula cannot lead to a larger inconsistency than the addition of a logically stronger formula.

Definition 1. *An inconsistency measure $I : \mathcal{K} \to \mathcal{R}^{\geq 0}$ is a function such that the following four conditions hold $\forall K, K' \in \mathcal{K}_{\mathcal{L}}$, $\forall \alpha, \beta \in \mathcal{L}$.*

- *Consistency: $I(K) = 0$ iff K is consistent.*
- *Monotony: $I(K \cup K') \geq I(K)$.*
- *Free Formula Independence: If α is a free formula of K, then $I(K) = I(K \setminus \{\alpha\})$.*
- *Dominance: If $\alpha \vdash \beta$ and $\alpha \nvdash \bot$, then $I(K \cup \{\alpha\}) \geq I(K \cup \{\beta\})$.*

There are many inconsistency measures in the literature but we will just focus on two in this paper (where K is a knowledgebase): $I_C(K) = |\mathsf{MI}(K)|$ is the inconsistency measure that counts the number of inconsistent subsets of K [3, 4]; and $I_B(K) = \mathsf{Contension}(K)$ is the inconsistency measure that counts the minimum number of atoms that need to be assigned B amongst the 3VL models of K [2, 5, 1].

We wish to compute the blame of each formula towards inconsistency. For this purpose we use a given inconsistency measure as the payoff function defining a game in coalitional form, and then use the Shapley value to compute the part of the inconsistency that can be imputed to each formula of the belief base [4]. Consider a game with players $1, \ldots, n$ whose utility function u assigns a nonnegative value to each coalition $C \subseteq \{1, \ldots, n\}$ such that if $C_1 \subseteq C_2$ then $u(C_1) \leq u(C_2)$. The Shapley value calculates each player's contribution to the utility of the coalitions the player joins in an optimal way. In our framework, following [4], we have a knowledgebase $K = \{\alpha_1, \ldots, \alpha_N\}$. The "utility" is the inconsistency measure; so for this purpose, the larger the inconsistency, the larger the utility of a set of formulae. The Shapley inconsistency value is defined as follows.

Definition 2. *The* **Shapley Inconsistency Value** *(SIV), denoted S_I, is the Shapley value of the coalitional game defined by the basic inconsistency measure I, where $|K| = n$,*

$|C| = c$, *and* $\alpha \in K$, *as follows.*

$$S_I^\alpha(K) = \sum_{C \subseteq K} \frac{(c-1)!(n-c)!}{n!}(I(C) - I(C \setminus \{\alpha\}))$$

Clearly, the only subsets of K that need to be considered are the ones where removing a formula changes the inconsistency measure, that is, the inconsistent sets. It will be convenient in the examples to first calculate the part of the formula that does not refer to the inconsistency measure for each inconsistent set C. We write $f(C) = \frac{(c-1)!(n-c)!}{n!}$. Hence, $S_I^\alpha(K) = \sum_{C \subseteq K} f(C)(I(C) - I(C \setminus \{\alpha\}))$.

Example 3. Let $K' = \{a, b, \neg a, \neg a \vee \neg b\}$. The subsets of K' for which removing a formula may change the inconsistency are: $C_1 = \{a, b, \neg a, \neg a \vee \neg b\}$ $C_2 = \{a, b, \neg a\}$ $C_3 = \{a, b, \neg a \vee \neg b\}$ $C_4 = \{a, \neg a, \neg a \vee \neg b\}$ $C_5 = \{a, \neg a\}$ Then $f(C_1) = \frac{3!}{4!} = \frac{1}{4}$, $f(C_2) = f(C_3) = f(C_4) = f(C_5) = \frac{2!}{4!} = \frac{1}{12}$. $I_C(C_1) = 2, I_C(C_i) = 1$ for $2 \leq i \leq 5$ and $I_B(C_i) = 1$ for $1 \leq i \leq 5$. We obtain $S_{I_C}^a(K) = \frac{2}{4} + \frac{4}{12} = \frac{5}{6}$; $S_{I_C}^{\neg a}(K) = \frac{1}{4} + \frac{1}{4} = \frac{1}{2}$; $S_{I_C}^b(K) = S_{I_C}^{\neg a \vee \neg b} = \frac{1}{4} + \frac{1}{12} = \frac{1}{3}$. $S_{I_B}^a(K) = \frac{1}{4} + \frac{4}{12} = \frac{7}{12}$; $S_{I_B}^{\neg a}(K) = \frac{3}{12} = \frac{1}{4}$; $S_{I_B}^b(K) = S_{I_B}^{\neg a \vee \neg b} = \frac{1}{12}$.

There are some interesting developments of Shapley values for inconsistency (see for example [6]), but there has been no consideration of the probabilistic uncertainty associated with an inconsistency measure.

3 Uncertainty of sources for answer tuples

There are many issues in managing distributed information. In this paper, we consider a specific problem of handling answer tuples as defined next. We assume that $\mathcal{A}$ denotes the set of atoms (propositional letters or ground predicates) in the language.

Definition 3. *Let* $\{1, \ldots, n\}$ *be the names for sources of information. An* **answer tuple***, denoted* $T = (A_1, \ldots, A_n)$*, is a tuple where for each* A_i, $1 \leq i \leq n$, $A_i \subseteq \mathcal{A}$.

An answer tuple, by itself, is never inconsistent. The inconsistency that occurs is the result of a set of integrity constraints that we assume is given a priori. We write A_0 for this set that contains formulae, each of which is a disjunction of negated atoms in $\mathcal{A}$. We say that T is inconsistent if $\bigcup_{i=0}^{n}(A_i) \vdash \bot$. Otherwise T is consistent. We will use the following subsidiary definitions: For the elements of T, $\mathsf{Elem}(T) = \bigcup_{i=1}^{n} A_i$; for the candidates of T, $\mathsf{Cand}(T) = \{S | S \subseteq \mathsf{Elem}(T)\}$.

To handle the issue of the reliability of the sources, we assume that a probability assignment is available for each source. Such an assignment may have been learnt from previous performance of sources, or obtained by some subjective judgment. We deal separately with the set of integrity constraints, A_0; basically we treat them as having probability 1.

Definition 4. *Let* $\{1, ..., n\}$ *be the names for sources of information. A* **probability assignment to sources***, denoted* P*, is a function* $P : \{1, ..., n\} \rightarrow [0, 1]$.

Given the probability assignment to sources, together with an answer tuple $T = (A_1, ..., A_n)$, we have further information to prefer some subsets of $\mathsf{Elem}(T)$ over others. To illustrate our concerns, we consider some scenarios next.

- At one extreme, suppose the probability is $P(i) = 0$ for each $i \in \{1, .., n\}$, then we need to consider only one candidate for the combination, which is $\emptyset$, since we believe that none of the formulae should be in the combination.
- At the other extreme, suppose the probability is $P(i) = 1$ for each $i \in \{1, .., n\}$, then we need to consider only one candidate for the combination, which is $\mathsf{Elem}(T)$, since we believe all should be in the combination.
- Between these two extremes, there may be multiple options for the combination. For example, suppose we have two sources, with the answers $A_1 = \{a\}$ and $A_2 = \{b\}$, while $A_0 = \{\neg a \vee \neg b\}$. Let $P(1) = 0.5$ and $P(2) = 0.5$. Then, there are four candidates for the combination to consider (i.e. $\mathsf{Cand}(T) = \{\{a, b\}, \{a\}, \{b\}, \{\}\}$), each with probability of 0.25, with the first candidate ($\{a, b\}$) being inconsistent.

The next step is to find the probability of each candidate. Consider that the sources may have different probability assignments and an atom may appear in several sets A_i. Suppose, for example, that the atom a appears in A_1 and A_2. Then, when we consider a candidate, such as $\{a, b\}$, we must consider all different cases where a was in A_1 but not in A_2, or a was in A_2 but not in A_1, or it was in both. As we need to take care of all of these cases, we start with a renaming where each atom is renamed using a superscript to indicate its source. So a in A_1 becomes a^1 and a in A_2 becomes a^2. We write r for this renaming and for answer tuple $T = (A_1, \ldots, A_n)$ we obtain $r(T) = (r(A_1), \ldots, r(A_n))$, where each $r(A_i)$ is obtained from A_i by adding the superscript i to each atom, that is, for $A_i = \{a, b, c\}$, $r(\{a, b, c\}) = \{a^i, b^i, c^i\}$. The inverse operator r^{-1} removes the subscripts. Thus if C' is a candidate of $r(T)$, then $r^{-1}(C')$ (which is a set, hence duplicates are removed) is a candidate of T.

We compute the probability of a candidate C' of $r(T)$ as follows. Let $y_i = |A_i| = |A'_i|$ and suppose that C' contains x_i elements from A'_i. We say that $(x_1, \ldots, x_n)$ is the generator of C' and write $Gen(C') = \{(x_1, \ldots, x_n)\}$. Using the renaming r, each candidate of $r(T)$ has a unique generator. Then computing the probability of a candidate C' of T' we get $P(C') = \prod_{i=1}^{n} P(i)^{x_i} \times (1 - P(i))^{y_i - x_i}$. Now suppose that C is a candidate of T. There may be several candidates C' of $r(T)$ such that $r^{-1}(C') = C$. Let $C_r = \{C' | r^{-1}(C') = C\}$. Then $P(C) = \sum_{C' \in C_r} P(C')$. We also write $Gen(C) = \{(x_1, \ldots, x_n) | (x_1, \ldots, x_n) \in Gen(C')$ and $C' \in C_r\}$. From this we obtain

$$P(C) = \sum_{(x_1,\ldots,x_n) \in \mathsf{Gen}(C)} P(1)^{x_1} \times (1 - P(1))^{(y_1 - x_1)} \times \ldots \times P(n)^{x_n} \times (1 - P(n))^{(y_n - x_n)}$$

Example 4. To illustrate the calculation of the probability distribution over candidates, we consider an example with $A_0 = \{\neg a \vee \neg c\}$ where $T = (A_1, A_2)$ with $A_1 = \{a, b\}$ and $A_2 = \{c\}$. Let the probability assignment for sources be $P(1) = 4/5$ and $P(2) = 2/3$. Here, $\mathsf{Cand}(T) = \wp(\{a, b, c\})$. In this example, for each candidate there is a unique generator because each atom appears in just one source's answer. For each candidate, we give the generator, and the probability for the candidate, in Table1.

Candidate	Generator	Probability of candidate
$\{a,b,c\}$	$(2,1)$	$4/5 \times 4/5 \times 2/3 = 32/75$
$\{a,b\}$	$(2,0)$	$4/5 \times 4/5 \times 1/3 = 16/75$
$\{a,c\}$	$(1,1)$	$4/5 \times 1/5 \times 2/3 = 8/75$
$\{a\}$	$(1,0)$	$4/5 \times 1/5 \times 1/3 = 4/75$
$\{b,c\}$	$(1,1)$	$1/5 \times 4/5 \times 2/3 = 8/75$
$\{b\}$	$(1,0)$	$1/5 \times 4/5 \times 1/3 = 4/75$
$\{c\}$	$(0,1)$	$1/5 \times 1/5 \times 2/3 = 2/75$
$\{\}$	$(0,0)$	$1/5 \times 1/5 \times 1/3 = 1/75$

Table 1. Calculations for Example 4

For a fact $\alpha \in \mathsf{Elem}(T)$, we have an a priori probability that it is true. This is the sum of the probability of each candidate that contains it. We denote this probability by the function $P : Atoms \rightarrow [0,1]$, where $P(\alpha) = \sum_{C \in \mathsf{Cand}(T) \text{ s.t. } \alpha \in C} P(C)$

Proposition 1. *For $T = (A_1, ..., A_n)$, $\sum_{C \in \mathsf{Cand}(T)} P(C) = 1$.*

The next proposition shows that if an atom that is an element of an answer tuple is removed, the probability of each candidate of the new answer tuple is the sum of the probabilities of the candidate and the candidate obtained by adding the atom.

Proposition 2. *Let $T = (A_1, \ldots, A_n)$, P be a probability assignment over sources, and $\alpha \in \mathsf{Elem}(T)$. Let $T' = (A_1 \setminus \{\alpha\}, \ldots, A_n \setminus \{\alpha\})$ (where if $A_i \setminus \{\alpha\} = \emptyset$, it is omitted from T') and write P' for the (same) probability assignment over sources for T'. Let $C' \in \mathsf{Cand}(T')$. Then $P'(C') = P(C') + P(C' \cup \{\alpha\})$.*

In the next section, we use the set of candidates to define a notion of expected inconsistency of a set of answers that is based on the probability distribution over the candidates.

4 Expected inconsistency of a set of formulae

We can measure the inconsistency of each candidate (using any inconsistency measure that is appropriate), and then aggregate the inconsistency measure for the combination as follows.

Definition 5. *Let I be an inconsistency measure, $T = (A_1, ..., A_n)$ an answer tuple, and P a probability distribution over the sources. The* **expected inconsistency** *of T with respect to I, denoted $E_{I,P}(T)$, is $E_{I,P}(T) = \sum_{C \in \mathsf{Cand}(T)} P(C) \times I(C)$.*

Example 5. To illustrate the definitions so far, consider the case where $A_0 = \{\neg a \vee \neg c, \neg b \vee \neg d\}$, $A_1 = \{a, b\}$ $A_2 = \{c, d\}$, $P(1) = 0.5$, and $P(2) = 0.5$. So $T = (A_1, A_2)$, and the set of candidates $\mathsf{Cand}(T)$ is the following

$$\begin{array}{cccccccc} \{a,b,c,d\} & \{a,c,d\} & \{b,c,d\} & \{c,d\} & \{a,b,c\} & \{a,c\} & \{b,c\} & \{c\} \\ \{a,b,d\} & \{a,d\} & \{b,d\} & \{d\} & \{a,b\} & \{a\} & \{b\} & \{\} \end{array}$$

Let $I = I_C$ or $I = I_B$. The numbers are the same for both measures. Hence, we obtain $I(\{a,b,c,d\}) = 2$, and for the remaining 6 inconsistent sets C', $I(C') = 1$. For each $C \in \mathsf{Cand}(T)$, $P(C) = 1/16$. Therefore, $E_{I,P}(T) = \frac{1}{16}(2 + (6 \times 1)) = \frac{8}{16} = \frac{1}{2}$.

So expected inconsistency takes into account the inconsistency measure as well as the probabilities of the sources and hence the candidates.

Proposition 3. *For $T = (A_1, ..., A_n)$, and an inconsistency measure I, if each source i is such that $P(i) = 1$, then $E_{I,P}(T) = I(\bigcup_{i=0}^{n} A_i)$.*

Proposition 4. *For $T = (A_1, ..., A_n)$, and an inconsistency measure I, if each source i is such that $P(i) = 0$, then $E_{I,P}(C) = 0$ for all $C \in \mathsf{Cand}(T)$.*

Proposition 5. *For an answer tuple $T = (A_1, ..., A_n)$, an inconsistency measure I, and a probability distribution P, $E_{I,P}(T) \leq I(\mathsf{Elem}(T))$.*

Proposition 6. *Let $T = (A_1, \ldots, A_n)$, I be an inconsistency measure, and P be a probability assignment and $T' = (A_1 \setminus \{\alpha\}, \ldots, A_n \setminus \{\alpha\})$ (if $A_i \setminus \{\alpha\} = \emptyset$, it is omitted from T'). Then $E_{I,P}(T) \geq E_{I,P}(T')$.*

Whilst the proposal for expected inconsistency is in terms of answer tuples, it is a trivial revision of the definition for expected inconsistency (i.e. Definition 5) to enable it to handle arbitrary knowledgebases of classical logic. Expected inconsistency is a simple extension of the approach of inconsistency measures. Intuitively, it involves discounting inconsistency that is unlikely to occur. So for instance, a small inconsistency that it very likely to occur can be worse than a large inconsistency that is unlikely to occur.

5 Expected blame of a formula

We use the Shapley Inconsistency Values of Definition 2 to ascribe the proportion of blame to each formula. Our definition of expected blame for an atom is the weighted sum of the blame for the atom in each candidate containing the atom.

Definition 6. *Let I be an inconsistency measure, $T = (A_1, ..., A_n)$ an answer tuple, and P a probability assignment. The* **expected blame** *of α in T with respect to I and P, denoted $B^{\alpha}_{I,P}(T)$, is $B^{\alpha}_{I,P}(T) = \sum_{C \in \mathsf{Candidates}(T)} P(C) \times S^{\alpha}_{I}(C)$*

Example 6. Consider $A_0 = \{\neg a \vee \neg b, \neg b \vee \neg c\}$, $A_1 = \{a\}$, $A_2 = \{b\}$, and $A_3 = \{c\}$, where $P(1) = 1$, $P(2) = 0.8$, and $P(3) = 0.5$. There are 4 candidates with non-zero probability: $C_1 = \{a,b,c\}$, $C_2 = \{a,b\}$, $C_3 = \{a,c\}$, and $C_4 = \{a\}$, where $P(C_1) = P(C_2) = 0.4$ and $P(C_3) = P(C_4) = 0.1$. We do the calculation separately for I_C and I_B.

- For $I = I_C$, The Shapley values are as follows: $S^a_I(C_1) = S^c_I(C_1) = S^a_I(C_2) = S^b_I(C_2) = 0.5$ and $S^b_I(C_1) = 1$. All other Shapley values are 0. Next we compute the expected blames: $B^a_{I,P}(T) = (0.4 \times 0.5) + (0.4 \times 0.5) = 0.4$, $B^b_{I,P}(T) = (0.4 \times 1) + (0.4 \times 0.5) = 0.6$, and $B^c_{I,P}(T) = (0.4 \times 0.5) = 0.2$.

- For $I = I_B$, The Shapley values are as follows: $S^a_I(C_1) = S^c_I(C_1) = \frac{1}{6}$ $S^a_I(C_2) = S^b_I(C_2) = 0.5$ and $S^b_I(C_1) = \frac{2}{3}$. All other Shapley values are 0. Next we compute the expected blames: $B^a_{I,P}(T) = (0.4 \times \frac{1}{6}) + (0.4 \times 0.5) = \frac{4}{15}$, $B^b_{I,P}(T) = (0.4 \times \frac{2}{3}) + (0.4 \times 0.5) = \frac{7}{15}$, and $B^c_{I,P}(T) = (0.4 \times \frac{1}{6}) = \frac{1}{15}$.

In both cases the blame for b is highest because it is involved in all the miminal inconsistent subsets, and the blame for a is higher than the blame for c because the probability of a is higher than c.

The probability assigned to a source directly affects the blame attributed to any atom given by that source. As formalized next, if α is given by a single source, the blame for α increases as the probability assigned to the source of α increases.

Proposition 7. *Let $T = (A_1, ..., A_n)$, and I be an inconsistency measure. Suppose that atom α appears in only one source as an answer, say A_1. Let P_1 and P_2 be probability assignments such that $P_1(1) \leq P_2(1)$ and $P_1(i) = P_2(i)$ for $i > 1$. Then $B^\alpha_{I,P_1}(T) \leq B^\alpha_{I,P_2}(T)$.*

In the following theorem, the first three properties are a restatement in this logical framework of the properties of the Shapley value: the distribution property states that the inconsistency values of the formulae sum to the total amount of expected inconsistency in the answer tuple; the symmetry property ensures that with equal probabilities only the amount of inconsistency brought by a formula matters for computing the expected blame; the minimality property expresses that a formula that is not embedded in any contradiction (i.e. does not belong to any minimal inconsistent subset) will not be blamed by the Shapley inconsistency values; and the dominance property states that logically stronger formulae bring (potentially) more conflicts.

Theorem 1. *Let I be basic inconsistency measure, and let P be a probability assignment to sources. Every expected blame value $B_{I,P}$ satisfies:*

- *Distribution:* $E_{I,P}(T) = \sum_{\alpha \in \mathsf{Elements}(T)} B^\alpha_{I,P}(T)$
- *Symmetry: If* $\alpha, \beta \in \mathsf{Elem}(T)$
  and for all $S \in \mathsf{Cand}(T)$ *such that* $\alpha \notin S$ *and* $\beta \notin S$,
  $P(S \cup \{\alpha\}) = P(S \cup \{\beta\})$ *and* $I(S \cup \{\alpha\}) = I(S \cup \{\beta\})$
  then $B^\alpha_{I,P}(T) = B^\beta_{I,P}(T)$.
- *Minimality: If α is a free formula of T, then* $B^\alpha_{I,P}(T) = 0$
- *Dominance: If* $\alpha \vdash \beta$ *and* $\alpha \nvdash \bot$, *then* $B^\alpha_{I,P}(T) \geq B^\beta_I(T)$

Expected blame is an extension of the approach of Shapley inconsistency values to the case of probabilities assigned to sources. Intuitively, it involves discounting blame that is unlikely to occur. So, for instance, blame for a small inconsistency that is very likely to occur can be greater than blame for a larger inconsistency unlikely to occur.

6 Weakness of a formula

Given an inconsistent answer tuple $(A_1, ..., A_n)$, we want to resolve some of the inconsistency by deleting an individual formula. We will use the blame of each formula, but using only blame is not enough. We need to use separately the probability of the formula to obtain a reasonable answer for determining the best formula to delete. There is an interplay between the inconsistency caused by a formula, and the uncertainty of the formula. To illustrate, consider the following example.

Example 7. Let $A_0 = \{\neg a \vee \neg b\}$, $A_1 = \{a\}$, and $A_2 = \{b\}$. In this case, for any I and P, $B^a_{I,P}(T) = B^b_{I,P}(T)$, but if $P(1) > P(2)$, then we would be more inclined to delete b as it has the same blame for the inconsistency, but the belief in it is lower.

Recall that in Section 3 we defined the probability of each fact. So now, we start with the Shapley value for each formula, and weight it by a function of the probability of the formula. We will consider the weighting function as a parameter that can be chosen by the user of the system. As an example of a weighting function, let $F(\alpha) = 1 - P(\alpha)$, because we prefer to delete formulae whose probability is small. It is certainly possible to consider other weighting functions. For example, let $F_1(\alpha) = k \times (1 - P(\alpha))$ for some number k. However, this merely expands or shrinks the difference between the weights but does not change the weight order. Another possibility is to use a step function, such as the following: $F_2(\alpha)$ = 4 if $P(\alpha) = 0$, $F_2(\alpha)$ = 3 if $0 < P(\alpha) \leq 0.5$, $F_2(\alpha)$ = 2 if $0.5 < P(\alpha) < 1$, and $F_2(\alpha)$ = 1 if $P(\alpha) = 1$. Such a function blurs the distinction between probabilities within a certain range. Hence, we will continue working with F as defined above.

Definition 7. *Let I be an inconsistency measure, $T = (A_1, ..., A_n)$ an answer tuple, $\alpha \in \mathsf{Elem}(T)$, P a probability function over sources, and F the weighting function. The* **weakness** *of α in T with respect to I and P, is $W^\alpha_{I,P}(T) = F(\alpha) \times B^\alpha_{I,P}(T)$*

Our goal is to use this definition of weakness, to reduce $B^\alpha_{I,P}(T)$ and $P(\alpha)$ to a single value for α which represents the degree to which we should delete it. The higher the degree of weakness (i.e. the greater the product of the weight and the blame for inconsistency for the formula), the greater the degree to which we should delete it.

Example 8. Let $A_0 = \{\neg a \vee \neg b\}$, $A_1 = \{a\}$, and $A_2 = \{b\}$. There is only one inconsistent candidate: $\{a, b\}$. For $I = I_C$ or $I = I_B$, the Shapley values are $S^a_I(T) = 0.5$ and $S^b_I(T) = 0.5$. We will use $F(\alpha) = 1 - P(\alpha)$ as the weighting function and consider the following scenarios for the probabilities for $P(1)$ and $P(2)$.

- $P(1) = 0.8$, $P(2) = 0.2$. Hence, $P(a) = 0.8$ and $P(b) = 0.2$. So, $W^a_I(T) = 0.1$ and $W^b_I(T) = 0.4$. Delete b.
- $P(1) = 0.6$ and $P(b) = 0.8$, Hence, $P(a) = 0.6$ and $P(b) = 0.8$. So, $W^a_I(T) = 0.2$ and $W^b_I(T) = 0.1$. Delete a.
- $P(1) = 0.5$ and $P(b) = 0.5$, Hence, $P(a) = 0.5$ and $P(b) = 0.5$. So, $W^a_I(T) = W^b_I(T) = 0.25$. As the probability of a and b is the same, there is no preference between deleting a or b.

Example 9. Continuing with Example 6 where we already computed the blames, we obtain the following weaknesses: When $I = I_C$, $W^a_{I,P}(T) = 0$, $W^b_{I,P}(T) = 0.12$, and $W^c_{I,P}(T) = 0.1$; And when $I = I_B$, $W^a_{I,P}(T) = 0$, $W^b_{I,P}(T) = \frac{7}{15}$, and $W^c_{I,P}(T) = \frac{1}{30}$. Note how close the weaknesses of b and c are for I_C. The reason is that while b

has higher blame, it also has higher probability and hence smaller weight. However, for I_B the blame is so much higher for b than for c that the higher probability does not compensate enough to make the weights close.

Proposition 8. *For $T = (A_1, ..., A_n)$, an inconsistency measure I, and a probability assignment over sources P, if $P(i) = 1$ for each source, then $W^{\alpha}_{I,P}(T) = 0$ for all $\alpha \in \mathsf{Elem}(T)$.*

The concept of weakness combines blame and the probability of the source that provides a fact. So if we try to resolve inconsistency by deleting some formulae, it is reasonable to start with the weakest one. Our examples illustrate the appropriateness of using this concept.

7 Summary and Future Work

We believe that this is the first paper that studies measuring inconsistency in the context where the uncertainty of the source of a formula is taken into account. We do not define a new inconsistency measure; our work applies to and combines with any given inconsistency measure. For such a measure we define the expected inconsistency of the answers based on the probabilities of the sources. We also define the expected blame of a formula and show that this definition has several useful properties. Finally, we combine blame with uncertainty to define the weakness of each formula, thereby providing a method to resolve inconsistencies by removing the weakest formulae.

In the future we plan to study additional properties of both expected blame and weakness. We will also consider the mechanism of inconsistency resolution in this framework, distinguishing between internal resolution (using weakness) and external resolution, where, in the latter case, we may request additional information from an outside source before deletion. Finally, we will consider how to measure the quality of the inconsistency resolution process.

References

1. Grant, J., Hunter, A.: Measuring inconsistency in knowledgebases. Journal of Intelligent Information Systems 27, 159–184 (2006)
2. Hunter, A.: Measuring inconsistency in knowledge via quasi-classical models. In: Proceedings of the National Conference on Artificial Intelligence (AAAI'02). pp. 68–73. MIT Press (2002)
3. Hunter, A., Konieczny, S.: Approaches to measuring inconsistent information. In: Inconsistency Tolerance, LNCS, vol. 3300, pp. 189–234. Springer (2004)
4. Hunter, A., Konieczny, S.: On the measure of conflicts: Shapley inconsistency values. Artificial Intelligence 174, 1007–1026 (2010)
5. Konieczny, S., Lang, J., Marquis, P.: Quantifying information and contradiction in propositional logic through epistemic actions. In: Proceedings of the 18th International Joint Conference on Artificial Intelligence (IJCAI'03). pp. 106–111 (2003)
6. Mu, K., Liu, W., Jin, Z.: Measuring the blame of each formula for inconsistent prioritized knowledge bases. Journal of Logic and Computation 22(3), 481–516 (2012)
7. Priest, G.: Logic of paradox. Journal of Philosophical Logic 8, 219–241 (1979)