EPiC Series in Computing

Volume 100, 2024, Pages 127–135

Proceedings of 25th Conference on Logic for Programming, Artificial Intelligence and Reasoning

A Tool for Reasoning about Trust and Belief

Aaron Hunter[1] and Alberto Iglesias[2]

[1] British Columbia Institute of Technology, Burnaby, Canada
aaron_hunter@bcit.ca
[2] British Columbia Institute of Technology, Burnaby, Canada
aiglesias4@my.bcit.ca

Abstract

We introduce a software tool for reasoning about belief change in situations where information is received from reports and observations. Our focus is on the interaction between trust and belief. The software is based on a formal model where the level of trust in a reporting agent increases when they provide accurate reports and it decreases when they provide innaccurate reports. If trust in an agent drops below a given threshold, then their reports no longer impact our beliefs at all. The notion of accuracy is determined by comparing reports to observations, as well as to reports from more trustworthy agents. The emphasis of this paper is not on the formalism; the emphasis is on the development of the prototype system for automatically calculating the result of iterated revision problems involving trust. We present an implemented system that allows users to flexibly specify and solve complex revision problems involving reports from partially trusted sources.

1 Introduction

Belief revision occurs when an agent receives new information that must be incorporated with some previous beliefs. The most influential approaches to belief revision make the assumption that new information is "better" than the original beliefs. Hence, an agent should believe the new information is true while keeping as much of the initial belief state as consistently possible. However, when the new information comes from an agent that may not be honest, then this is no longer sensible. In this paper, we present a prototype system that calculates the result of belief revision when new information includes observations and reports from other agents.

This work makes two main contributions to the existing literature on belief revision. First, we present a novel approach to integrating belief change and trust. When a series of reports is received, we not only describe the *belief change* that occurs, but we also describe the *trust change* that occurs. Second, while the theory of belief revision is well-studied, there continues to be relatively little work on practical tools. This paper provides a new revision tool that can address problems that were not solvable by existing implemented systems.

2 Preliminaries

The theory of belief revision is framed in the context of propositional logic. We assume an underlying propositional signature F, and we define propositional formulas in the usual manner

N. Bjørner, M. Heule and A. Voronkov (eds.), LPAR 2024 (EPiC Series in Computing, vol. 100), pp. 127–135

using connectives $\wedge, \vee, \rightarrow$, and $\neg$. A *belief state* K is a logically closed set of formulas. The most influential approach to belief revision is the AGM approach, where a belief state K and a formula ϕ are mapped to a new belief state $K * \phi$ [1]. AGM revision is defined with respect to a set of postulates, and the revision is calculated semantically by finding the most plausible states that are consistent with the new formula [9].

AGM revision can only be used for single-shot belief revision. If we want to perform iterated belief revision, then the dominant approach is the DP approach [5]. In DP revision, the initial beliefs are represented by an *epistemic state*. Although the exact composition of an epistemic state is flexible [12], one important feature is that it includes a total pre-order over states. Our formal approach is defined for DP revision, but the implemented tool is based on variations of the Dalal revision operator [4]. This is an operator where plausbility is defined in terms of the Hamming distance between states; it is the rare example of an AGM revision operator that can be iterated. When we discuss our theoretical framework, we give all definitions with respect to epistemic states in the DP sense. However, in our practical tool, epistemic states are specified by a set of formulas along with a distance-based operator that defines the total pre-order.

Most approaches to belief revision require that the new information *must* be believed; this is formalized by the so-called *success postulate* of AGM revision. As noted, this is not reasonable if information is reported by other agents. Two different approaches to this problem have been explored in the literature. There has been work on knowledge-based trust, where we only consider the 'part' of the information where the reporting agent has expertise[3, 13]. There has also been work on trust in terms of the reliability of information provided by different agents [2, 6, 8]. Throughout this paper, we assume an underlying revision operator which is then modified to capture some form of trust.

3 Trust Dynamics

3.1 Motivating Example

Suppose that Ali has two co-workers: Cindy and Juan. While they are both considered trustworthy, Ali tends to consider Cindy to be the more reliable source of information. So suppose that Cindy says it is raining, and then Juan says it is not raining. We would expect Ali to believe it is raining after this exchange, due to the stronger trust in Cindy.

Imagine that Cindy now reports it is hot outside. If we look at a thermometer and find it is actually cold, then this can change our perspective. Since Cindy has provided a report that conflicts with our direct observation, we may now trust her less. As a result, we might actually trust Juan more than Cindy following the full sequence of events. If this is indeed the case, then our beliefs should entail that it is cold outside (ignoring Cindy's report that conflicts with our observation) and that it is not raining (prefering Juan's report over Cindy's).

In practice, these kinds of relationships can be much more complicated. For example, if Cindy is "strongly" trusted, then it is unlikely that she would be discounted after just one false report; the strength of trust needs to be considered. Moreover, we also need to provide a mechanism for agents to "earn" more trust by providing accurate reports. The framework in this paper is intended to capture trust dynamics in a formal setting.

3.2 Framework

We sketch an approach to belief revision with trust change. We assume a set $\mathbf{A}$ of agents. A *report* is a pair (A, ψ) where ψ is a formula and $A \in \mathbf{A}$. We will write $\overline{R}$ for a finite sequence of reports. The following definition introduces an abstract structure that can filter out reports that are not trustworthy.

Definition 1. *A trust structure T is a function such that $T(\overline{R}, \phi) = \psi_1, \ldots, \psi_m$ where $\psi_m = \phi$ and $\psi_1, \ldots, \psi_{m-1}$ is a subsequence of the formulas appearing in $\overline{R}$.*

Intuitively, a trust structure filters out reports that are not trusted for some reason. The final formula ϕ represents an observation, so it must be included as the final revision. We define a useful class of trust structures.

Definition 2. *A trust structure T is* agent-based *if it satisfies the following property:*

- *If (A, ψ) in $\overline{R}$ and ψ is not in $T(\overline{R}, \phi)$, then for each report (A, τ) in $\overline{R}$, τ in $T(\overline{R}, \phi)$ only if there is an agent $B \neq A$ such that (B, τ) is in $\overline{R}$.*

An agent-based trust structure is one where, if we remove one report from an agent A, then we remove all reports from A. So the only way that a formula reported by A will be kept is if that formula happens to have also been reported by another agent. Another important class of structures is those that only include reports that are consistent with the final observation.

Definition 3. *A trust structure is* observation-consistent *if $T(\overline{R}, \phi) = \psi_1, \ldots, \psi_m$ implies that $\phi \not\models \neg\psi_i$ for any $i < m$*

We provide a concrete example.

Definition 4. *The trust structure T_{AO} is defined such that $T_{AO}(\overline{R}, \phi)$ is obtained by appending ϕ to the list of formulas ψ_i such that (A_i, ψ_i) is in $\overline{R}$ and there is no report (A_j, ϕ_j) in $\overline{R}$ such that $A_j = A_i$ and $\phi \models \neg\psi_j$.*

So T_{AO} is the trust structure defined by removing reports from agents that have provided information that is inconsistent with ϕ, and keeping everything else. The following is immediate.

Proposition 1. *T_{AO} is an agent-based, observation-consistent trust structure. Moreover, T_{AO} is maximal among such trust structures, with respect to the length of output sequence.*

Hence, if we would like an agent-focused model of trust where dishonest agents are ignored, then T_{AO} is the most inclusive structure. We give some more important examples.

Example Let α be the set of *honest* agents. Informally, agents in α are trusted and those outside α are not. Then we can define T_α such that $T_\alpha(\overline{R}, \phi)$ is obtained by appending ϕ to the sequence of formulas ψ_i such that (A_i, ψ_i) is in $\overline{R}$ and $A_i \in \alpha$. Note that this might not be observation consistent, unless we also enforce an additional constraint.

Example Suppose we have a total pre-order $<$ over agents where the highest ranked agents are the most trusted. Then we can define $T_<$ as follows. First, remove all reports that come from agents that have provided information that conflicts with the observation ϕ. Next, starting with the maximal elements of $<$, iteratively remove all reports provided by an agent that has supplied information that conflicts with a higher ranked agent.

We can unify the two examples above, by defining a larger class $T_{\alpha,<}$ of trust structures; we call these *order constrained* trust structures. We define $T_{\alpha,<}$ by first removing all reports from agents outside α and from agents that have provided reports conflicting with ϕ. Next, we iterate through $<$ and remove conflicting reports, as specified in the construction of $T_<$ above.

Proposition 2. *For any $\alpha \subseteq \mathbf{A}$ and any total pre-order $<$ over $\mathbf{A}$, $T_{\alpha,<}$ is an agent-based and observation-consistent trust structure.*

3.3 Changing Trust

The trust structures in the previous section provide a natural mechanism for representing static trust. They can also form the basis of a dynamic theory of trust, based on *epistemic trust states.*

Definition 5. *An* epistemic trust state *is a pair* $\langle \mathbb{E}, T_{\alpha,<} \rangle$ *where* $\mathbb{E}$ *is an epistemic state and* $T_{\alpha,<}$ *is an order constrained trust structure.*

An epistemic trust state gives a representation of the initial beliefs of an agent, along with the trust structure that they will use to decide which reports they can trust. However, it is not only the beliefs that change with new information; the trust stucture must also change.

Definition 6. *A report history (RH) operator is a function* $\circ$ *that maps an epistemic trust state* $\mathbb{E}$ *and a pair* $\langle \overline{R}, \phi \rangle$ *to a new epistemic trust state* $\mathbb{E} \circ \langle \overline{R}, \phi \rangle$.

An RH operator is both a *belief revision operator* and a *trust revision operator.* However, agents do not receive direct information about trust; it is updated by looking at the accuracy of reports. We can specify desirable properties for trust change the same way we specify properties for belief change. For example, one useful property is the following.

[D1] Suppose $\langle \mathbb{E}, T_{\alpha,<} \rangle \circ \langle (A, \neg\psi), \psi \rangle = \langle \mathbb{E}', T_{\beta,\prec} \rangle$. Then, $A < B$ implies $A \prec B$.

This property asserts that an agent does not become *more* trustworthy after providing a false report. We are not asserting this is true in general, we are asserting it is a property of interest for classifying *RH* operators. One notable class of *RH* operators is defined under a different name in [6], where the notion of *conflict* is used to remove agents from the honesty set if they provide *too many* reports that conflict with an observation.

The full set of trust change postulates will be presented in a companion paper, along with a semantic characterization of RH operators. In this paper, we focus on the RH operators used by our software tool. Towards that end, we provide a mechanism for defining and updating a trust structure defined by numeric parameters; this will be suitable for automating trust change.

Definition 7. *Let* $\min \in \mathbb{N}$ *and let* t *be a ranking function on agents. Then the* numeric trust structure $T_{\min,t}$ *is the trust structure* $T_{\alpha,\prec}$ *where*

1. $\alpha = \{A \mid t(A) < \min\}$
2. $A \prec B$ *if and only if* $t(A) < t(B)$

So, given a threshold and a ranking function, we can define a unique trust structure. We now define a change operation.

Definition 8. *Let* $T_{\min,t}$ *be a numeric trust structure and let* $P = \langle to^+, to^-, tr^+, tr^-, L \rangle$ *be a tuple of natural numbers (called the trust parameter). Define* $T_{\min,t} \cdot (\langle \overline{R}, \phi \rangle, P) = T_{\min,t'}$ *where* $t'(A)$ *is specified by applying the following procedure:*

1. *Initially, set* $t'(A) = t(A)$. *Update as follows by comparison with* ϕ:
   - *If there is a report* (A, ψ) *in* $\overline{R}$ *such that* $\phi \models \neg\psi$, *then* $t'(A) = t(A) - to^-$.
   - *If there is a report* (A, ψ) *in* $\overline{R}$ *such that* $\phi \models \psi$, *then* $t'(A) = t(A) + to^+$.
2. *Next, compare with reports. For all agents* B *with* $t(A) < t(B)$, *if* $t(B) - t(A) > L$, *then:*
   - *If there are reports* (A, ψ) *and* (B, τ) *in* $\overline{R}$ *with* $\tau \models \neg\psi$, *then* $t'(A) = t(A) - tr^-$.

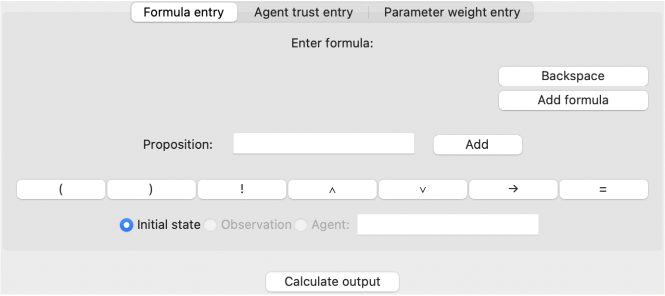


Figure 1: HBS Interface

- *If there are reports* (A, ψ) *and* (B, τ) *in* $\overline{R}$ *with* $\tau \models \psi$*, then* $t'(A) = t(A) + tr^+$*.*

The intuition here is the following. If A provides a report that disagrees with ϕ, then trust goes down by to^-. If A agrees with the observation ϕ, then trust in A goes up by to^+. The second two parameters have a similar usage. An agent that disagrees with a more trusted agent has their trust decreased by tr^-; an agent that agrees with a more trusted agent has their trust increased by tr^+. However, the trust differential must be greater than L for any change to happen based on reports alone. This constraint reflects the intuition that conflicting with a report may be less significant than conflicting with an observation.

Note that $T_{\min,t} \cdot (\langle \overline{R}, \phi \rangle, P)$ is a numeric trust structure; it is therefore agent-based and observation-consistent. This observation sets the stage for our definition of epistemic-trust change. In the following definition, we write $\mathbb{E} * T(\overline{R})$ as a short hand for the iterated revision of $\mathbb{E}$ by the list of formulas $T(\overline{R})$.

Definition 9. *Let* $*$ *be a DP revision operator, and let* $T = T_{\min,t}$ *be a numeric trust structure, and let* P *be a trust parameter. Define the RH operator* $\circ_{*,T,P}$ *such that* $\mathbb{E} \circ \langle \overline{R}, \phi \rangle = \langle \mathbb{E}', T' \rangle$ *where* $\mathbb{E}' = \mathbb{E} * T(\overline{R})$ *and* $T' = T \cdot (\langle \overline{R}, \phi \rangle P)$*.*

Hence, revision works as follows. We use T to determine which reports will be ignored; the new belief set is determined by iterated revision by this sequence of reports. Then, given a trust parameter P, we modify our trust in agents in accordance with Definition 8.

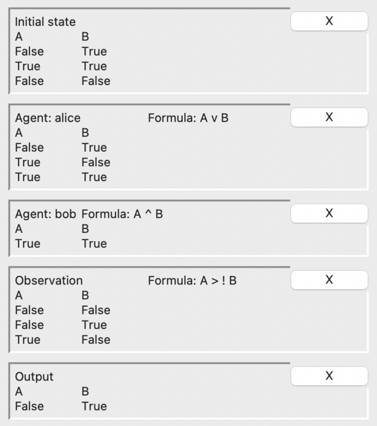
Proposition 3. *The RH operator in Definition 9 satisfies property D1.*

Thus, when agents provide information that is proven false, this does not lead to a trust increase. As noted previously, we leave a further examination of the formal properties of this operator for future work. However, we remark that it is easy to define P to make this operator consistent with the operators defined in [6], where agents are not trusted after some fixed number c of conflicts with observation. However, the approach here is significantly more flexible.

4 System Description

4.1 Interface

We now turn our attention to the main goal of this paper, which is to describe our software tool for implementing the combined epistemic-trust operation introduced above. The Honesty-based Belief revision System (HBS) is a tool written in Python to automatically solve belief revision



```
Initial state
A           B
False       True
True        True
False       False

Agent: alice          Formula: A v B
A           B
False       True
True        False
True        True

Agent: bob  Formula: A ^ B
A           B
True        True

Observation           Formula: A > ! B
A           B
False       False
False       True
True        False

Output
A           B
False       True
```

Figure 2: HBS Output

problems[1]. The system allows the user to specify an initial belief state, along with a sequence of reports and observations. The interface is shown in Figure 1.

When HBS is launched, it will initially be set to the *Formula Entry* tab. While the *Initial state* radio button is highlighted, this allows the user to enter a set of formulas that define the initial belief state. In interactive mode, formulas are entered with a simple graphic interface that prevents syntax errors. The formula being defined is displayed above the entry box, and it is entered as part of the initial belief state when the user clicks on *Add Formula.*

The initial belief state can be modified iteratively by adding more formulas, and a panel on the right will display the set of states believed possible. The radio button at the bottom can be toggled to add observations or reports. For observations, the formula is added to the right panel and labelled as an observation. For reports, an agent name must also be provided. All of the items listed in the right panel can be deleted at any time by clicking the X in the corner. As such, what the interface allows the user to do is to specify an expression of the form:

$$K * (A_1, \phi_1) * \psi_1 * \cdots * (A_n, \phi_n) * \psi_m.$$

The user can click *Calculate Output* to determine the new belief state after the given sequence of operations. Figure 2 shows the contents of the right panel after entering the following:

$$Cn[\neg(A \land \neg B)] * (\text{alice}, A \lor B) * (\text{bob}, A \land B) * (A \to \neg B)$$

The output at the bottom indicates the new belief state; we explain below how this is determined. We remark that the display can be modified to a simplified form; this will hide the lists of truth values for variables. This can be helpful for large examples with many variables.

4.2 Belief Revision Operators

We specify a default "idealized" revision operator. The idealized operator is the revision operator that would be used if we only had to incorporate observations. In HBS, the default revision operator is the Dalal operator based on the Hamming distance between states. However, this is not the only revision operator that HBS can capture.

[1] Software available at https://github.com/amhunter/HBS.

Under the edit menu, the user can change to a weighted Hamming distance operator. In this case, a weight needs to be associated with each variable and these weights are used in the distance calculation. It is easy to see that this approach to revision can capture many operators beyond the standard Dalal operator. For example, if we use powers of two for the weights, we can essentially specify a priority ordering over paramaters. Hence, the weighted Hamming distance can be used to capture any parametrized difference operator [11]. This is a natural class of revision operators suitable for iteration, with nice computational properties.

4.3 Trust

The trust held in different reporting agents is entered as a numeric value in the edit menu. In this menu, we can also specify trust parameters. There are six different parameters available in the software, corresponding to the formal parameters in Definition 8:

Trust Observation Decrease (to^-)	Trust Decrease (tr^-)	No Trust Threshold (min)
Trust Observation Increase (to^+)	Trust Increase (tr^-)	Difference Threshold (L)

The user can enter any natural number values for these parameters through the interface. The default values can also be changed, but this requires editing the `revision.py` file.

4.4 Calculations

Suppose that a series of reports followed by a single observation has been entered, and the user presses 'Calculate output.' Then the following calculations are performed:

1. First, the trust values are updated in accordance with Definition 8.
2. The new trust values are used to determine the subsequence of formulas for revision.
3. The revision is performed based on the selected revision operator.
4. The new belief state is displayed; it will be used for future revisions.

If the sequence of reports and observations involves several observations (rather than a single terminal observation), then step (1) and (2) involve multiple sweeps through the reports to remove those that are inconsistent with *any* observation.

4.5 Creating Test Cases

For examples involving many formulas, there is also a mechanism to load test cases from a text file in the following format:

```
(Av!A)^B/!A^!B
1,2
alice:AvB
bob:A^B
:A>!B
```

The first line specifies a set of formulas, separated by slashes. The second line gives weights to all variables, in the order that they appear in the input. These values are for the weighted Hamming distance; they should all be set to 1 if Dalal revision is preferred. The remaining lines specify reports in the form "agent:formula." If the agent part is left blank, then the line represents an observation. When a test case is loaded from a file in this format, then it automatically populates the right panel with all of the information.

5 Discussion

Trust has been explored in a variety of formal settings involving agents with limited beliefs exchanging information [2, 3, 6, 8, 10, 13]. The work in this paper differs in that we focus explicitly on the interaction of a belief revision operator with a dynamic notion of trust that changes as reports are received. This is basically what is required to model the kinds of reputation system that serve as the basis for trust in online communities. The work in this paper is also distinguished by the fact that we provide an implemented system for experimentation with trust and belief change.

There are many directions for future work. As already noted, a theoretical companion to the present paper is forthcoming; it will focus on the formal characterization of RH operators in terms of rationality postulates. We are also interested in extending the current framework, so that it can model the interaction between knowledge-based trust and honesty-based trust. Finally, there is also work to be done on the software side. The current version of HBS is a proof of concept that is only suitable for small toy problems, due to the computational complexity of belief revision. However, it is possible to signicantly improve the running time for revision solvers by using a competition-level ALLSAT solver [7]. In the next iteration of the software, we will use this approach to produce a tool that is useful for reasoning about much larger problems.

References

[1] Carlos E. Alchourrón, Peter Gärdenfors, and David Makinson. On the logic of theory change: Partial meet functions for contraction and revision. *Journal of Symbolic Logic*, 50(2):510–530, 1985.

[2] Yasser Ammar and Haythem O. Ismail. Trust is all you need: From belief revision to information revision. In *Proceedings of the 17th European Conference on Logics in Artificial Intelligence (JELIA)*, pages 50–65, 2021.

[3] Richard Booth and Aaron Hunter. Trust as a precursor to belief revision. *Journal of Artificial Intelligence Research*, 61:699–722, 2018.

[4] Mukesh Dalal. Investigations into a theory of knowledge base revision. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, pages 475–479, 1988.

[5] Adnan Darwiche and Judea Pearl. On the logic of iterated belief revision. *Artificial Intelligence*, 89(1-2):1–29, 1997.

[6] Aaron Hunter. Reports, observations, and belief change. In *Australasian Joint Conference on Artificial Intelligence (AJCAI)*, pages 54–65, 2023.

[7] Aaron Hunter and John Agapeyev. An efficient solver for parametrized difference revision. In *Proceedings of the Australasian Conference on Artificial Intelligence*, pages 143–152, 2019.

[8] David Jelenc, Luciano H. Tamargo, Sebastian Gottifredi, and Alejandro Javier García. Credibility dynamics: A belief-revision-based trust model with pairwise comparisons. *Artificial Intelligence*, 293:103450, 2021.

[9] Hirofumi Katsuno and Albert O. Mendelzon. Propositional knowledge base revision and minimal change. *Artificial Intelligence*, 52(2):263–294, 1992.

[10] Fenrong Liu and Emiliano Lorini. Reasoning about belief, evidence and trust in a multi-agent setting. In *PRIMA 2017: Principles and Practice of Multi-Agent Systems - 20th International Conference*, volume 10621, pages 71–89, 2017.

[11] Pavlos Peppas and Mary-Anne Williams. Parametrised difference revision. In *Proceedings of the International Conference on Principles of Knowledge Representation and Reasoning (KR)*, pages 277–286, 2018.

[12] Nicolas Schwind, Sebastien Konieczny, and Ramon Pino Perez. Darwiche and Pearl's epistemic states are not total preorders. In *Proceedings of the International Conference on Principles of Knowledge Representation and Reasoning (KR 2022)*, 2022.

[13] Joe Singleton and Richard Booth. Who's the expert? On multi-source belief change. In *Proceedings of the International Conference on Principles of Knowledge Representation and Reasoning (KR 2022)*, 2022.