Modelling the Persuadee in Asymmetric Argumentation Dialogues for Persuasion

Anthony Hunter
Department of Computer Science, University College London, London, UK
anthony.hunter@ucl.ac.uk

Abstract
Computational models of argument could play a valuable role in persuasion technologies for behaviour change (e.g., persuading a user to eat a more healthy diet, or to drink less, or to take more exercise, or to study more conscientiously, etc.). For this, the system (the persuader) could present arguments to convince the user (the persuadee). In this paper, we consider asymmetric dialogues where only the system presents arguments, and the system maintains a model of the user to determine the best choice of arguments to present (including counterarguments to key arguments believed to be held by the user). The focus of the paper is on the user model, including how we update it as the dialogue progresses, and how we use it to make optimal choices for dialogue moves.

1 Introduction
Persuasion is an activity that involves one party trying to induce another party to believe something or to do something. It is an important and multifaceted human facility. Persuasion technologies [Fogg, 1998] are being developed with an emphasis on building systems to help people make positive changes to their behaviour, particularly for healthcare and healthy life styles. Interestingly, argumentation is not central to the current manifestations of persuasion technologies [Hunter, 2014]. Rather there is an emphasis on either helping users to explore their issues (e.g. game playing) or helping users once they are persuaded to do something (e.g. diaries for recording calorie intake for weight management).

To address the lack of explicit argumentation in persuasion technologies, we propose a framework for argumentation dialogues. A system (the persuader running for example as an app) enters into a dialogue with a user (the persuadee using the app) to persuade them to believe an argument (for some action such as eating some fruit, or for not doing some action such as texting while driving, etc).

A key challenge for building a dialogical argumentation system is getting arguments from the user as we are unable (in the short term) to build a system to automatically understand natural language arguments from the user. Our solution to this problem is to have asymmetric dialogues where the kinds of move available to the system are different to those available to the user. In this paper, we allow the system to posit arguments but the user is unable to posit arguments.

Example 1. The system moves are starred: (1*) You believe that a cup cake is preferable to a banana? (2) Yes. (3*) It is late afternoon, and you think a cup cake will give you a sugar rush to help you work? (4) Yes. (5*) The sugar rush from a cup cake is brief, and therefore it won’t help you work. (6*) A banana gives a longer lasting energy supply, and so a banana is preferable to a cup cake.

Using asymmetric dialogues creates a challenge for the system to choose appropriate arguments to present in order to maximize the likelihood that the system is successful in persuading the user. To address this, the system uses a model of the user. In this paper, we investigate a probabilistic user model, including how the system updates the model at each step of the dialogue, how it uses the model to choose moves, and how it can query the user to improve the model.

2 Asymmetric dialogues
We base our paper on abstract argumentation [Dung, 1995]. We assume our dialogues concern an argument graph $G$ without self-attacks where $\text{Args}(G)$ is the set of arguments in $G$, and $\text{Attacks}(G)$ is the set of attack relations in $G$.

We focus on the following kinds of move in this paper: (1) Posit of an argument $A$ by the system, denoted $A!$; (2) Query by the system to the user about an argument, denoted $A?$; (3) Reply by the user of yes (denoted $Y$) or no (denoted $N$) to a query; And (4) termination of the dialogue by the system (denoted $\perp$).

A dialogue is a sequence of moves $D = \{m_1, \ldots, m_k\}$. Equivalently, we use $D$ as a function with an index position $i$ to return the move at that index (i.e. $D(i) = m_i$). We impose the following constraints on the dialogues in this paper.

- if $1 \leq i < k$, then $D(i) \neq \perp$.
- if $1 \leq i < k$, then $D(i) = A?$ iff $D(i + 1) \in \{Y, N\}$.

So a query in a dialogue is followed by the user answering, and a dialogue does not continue after the system has terminated. An example of a terminated dialogue is $[A!, B!, \perp]$ and examples of unterminated dialogues are $[A! , C!, D!, A!, C!, D!, A!]$ and $[A? , Y, B?, N]$.

Any unterminated dialogue may be extended by adding further moves,
and it may be terminated by adding \( \bot \). A dialogue may be infinite, but in this paper, we focus on finite dialogues.

For a dialogue \( D = [m_1, \ldots, m_k] \), let \( \text{Length}(D) = k \) and let \( \text{Set}(D) = \{m_1, \ldots, m_k\} \). For dialogues \( D' \) and \( D \), the subsequence relation, denoted \( D' \subseteq D \), holds iff for all \( i', j' \in \{1, \ldots, \text{Length}(D')\} \), if \( i' < j' \) then there are \( i, j \in \{1, \ldots, \text{Length}(D)\} \) such that \( i < j \) and \( D'(i') = D(i) \) and \( D'(j') = D(j) \). For example, \( [B!, D!] \subseteq [A!, B!, C!, D!, E!] \).

### 3 Probabilistic user models

We use epistemic probabilities [Thimm, 2012; Hunter, 2013; Hunter and Thimm, 2014b; Baroni et al., 2014].

**Definition 1.** A mass distribution \( P \) over \( \text{Args}(G) \) is such that \( \sum_{X \subseteq \text{Args}(G)} P(X) = 1 \). The probability of an argument \( A \) is \( P(A) = \sum_{X \subseteq \text{Args}(G)} \text{s.t. } A \in X P(X) \).

For a mass distribution \( P \), and \( A \in \text{Args}(G) \), \( P(A) \) is the belief that an agent has in \( A \) (i.e. the degree to which the agent believes the premises and the conclusion drawn from those premises). When \( P(A) > 0.5 \), then the agent believes the argument to some degree, whereas when \( P(A) < 0.5 \), then the agent disbelieves the argument to some degree.

We may wish to impose rationality (defined below) on our mass distributions [Hunter, 2013]. It forces the mass distribution to respect the structure of the graph, but it does not force an unattacked argument to be believed.

**Definition 2.** A mass distribution \( P \) is rational for \( G \) iff \( \forall (A, B) \in \text{Attacks}(G) \), if \( P(A) > 0.5 \), then \( P(B) \leq 0.5 \).

**Example 2.** Consider Figure 1. Mass distribution \( P_1(A) = 0.6 \), \( P_2(B) = 0.9 \), and \( P_3(C) = 0.9 \) is not rational, whereas \( P_2(A) = 0.6 \), \( P_2(B) = 0.3 \), and \( P_2(C) = 0.9 \) is rational, and \( P_2(B) = 0 \), \( P_3(B) = 1 \), and \( P_3(C) = 0.3 \) is rational.

In this paper, the system uses a mass distribution \( P \) as a model of the user, and so the system maintains \( P \) to represent the belief that the user has in each argument. The system can update the model at each stage of the dialogue depending on the move. We investigate how this can be done in the rest of the paper. Next is a non-exhaustive list of optional dynamic properties for when a mass \( P_{\text{t-1}} \) is updated to mass \( P_t \).

- (Credulous) If \( D(i) = A! \), then \( P_t(A) \geq 0.5 \).
- (Minimal) If \( D(i) \neq A! \), then \( P_t(A) = P_{t-1}(A) \).
- (Rational+) For \( G \), if \( P_{t-1} \) is rational, then \( P_t \) is rational.
- (Binary) If \( P_{t-1}(A) \neq P_t(A) \), then \( P_t(A) \in \{0, 1\} \).
- (Restricted) If \( P_{t-1}(A) \leq 0.5 \), and \( P_t(A) > 0.5 \), and \( (B, A) \in \text{Attacks}(G) \), then \( P_{t-1}(B) \leq 0.5 \).

These properties capture assumptions about the user, and we explain them as follows: (Credulous) A persuadee always believes an argument when it is posited by the persuader; (Minimal) A persuadee does not change belief in an argument that is unposited; (Rational) A rational distribution is only updated to a rational distribution; (Binary) Any update to belief in an argument is binary; (Restricted) Belief in an argument is only updated when the attackers of it are disbelieved.

The epistemic approach is useful for asymmetric dialogues where the user is not allowed to posit arguments or counterarguments. So the only way the user can treat arguments that s/he does not accept is by disbelieving them. In contrast, in symmetric dialogues, the user could be allowed to posit counterarguments to an argument that s/he does not accept. For example, suppose the user believes \( A \), and the system posits \( B \) where \( B \) attacks \( A \), then the user may disbelieve \( B \) and continue to believe \( A \), and this could be modelled by a rational mass distribution.

### 4 Persuasion goals

**A persuasion goal** is a Boolean combination of arguments. If \( A \in \text{Args}(G) \), then \( A \) is a positive literal, and \( \neg A \) is a negative literal. Let Formulaes(\( G \)) denote all the formulae that can be formed from the arguments in \( G \) using \( \land, \lor, \land \), and \( \neg \) as connectives in the usual way.

Informally, for an argument \( A \), the goal \( A \) means that the persuader aims to persuade the persuadee to accept \( A \), and the goal \( \neg A \) means that the persuader aims to persuade the persuadee to not accept \( A \). The goal \( A \land B \) means that the persuader aims to persuade the persuadee to accept \( A \) and to accept \( B \), and the goal \( A \lor B \) means that the persuader aims to persuade the persuadee to accept \( A \) or to accept \( B \).

In order to formalize the satisfaction of persuasion goals, we treat each subset of \( \text{Args}(G) \) as a model.

**Definition 3.** The satisfaction relation is defined as follows where \( X \subseteq \text{Args}(G) \), \( A \in \text{Args}(G) \), and \( \alpha, \beta \in \text{Formulaes}(G) \).

- \( X \models \alpha \text{ when } A \in X \)
- \( X \models \alpha \land \beta \text{ if } X \models \alpha \text{ and } X \models \beta \)
- \( X \models \alpha \lor \beta \text{ if } X \models \alpha \text{ or } X \models \beta \)
- \( X \models \neg \alpha \text{ if } X \not\models \alpha \)

Essentially \( \models \) is a classical satisfaction relation. So if \( \alpha \) is a classical tautology, then \( X \models \alpha \text{ for all } X \subseteq \text{Args}(G) \), and if \( \alpha \) is a classical contradiction, then \( X \not\models \alpha \text{ for all } X \subseteq \text{Args}(G) \). For \( \alpha \in \text{Formulaes}(G) \), let Models(\( \alpha \)) = \{ \( X \subseteq \text{Args}(G) \mid X \models \alpha \) \}.

For each graph \( G \), we assume an ordering over the arguments \( A_1, \ldots, A_n \), so that we can encode each model by a binary number: For a model \( X \), if the ith argument is in \( X \), then the ith digit is 1, otherwise it is 0. For example, for \( A, B, C \), the model \( \{A, C\} \) is represented by 101.

For a user model \( P \), the probability that a user believes a persuasion goal \( \phi \) is the sum of the probability of each model that satisfies the goal. This definition (below) is adapted from [Paris, 1994] where a probability distribution is defined over models of a propositional language.

**Definition 4.** The probability of the goal \( \phi \in \text{Formulaes}(G) \) is \( P(\phi) = \sum_{X \subseteq \text{Args}(G)} \text{s.t. } X \models \phi \, P(X) \).

Suppose \( \alpha \in \text{Formulaes}(G) \) and \( P \) is a mass distribution. If \( \alpha \) is a contradiction of classical logic, then \( P(\alpha) = 0 \), and if
\(\alpha\) is a tautology of classical logic, then \(P(\alpha) = 1\). Also, if \(\{\alpha\} \vdash \beta\), then \(P(\alpha) \leq P(\beta)\), and if \(\neg(\alpha \land \beta)\) is a classical tautology, then \(P(\alpha \lor \beta) = P(\alpha) + P(\beta)\).

We will use persuasion goals as outcomes for this. For this, we need the following subsidiary definitions. Persuasion goals \(\phi\) and \(\psi\) are disjoint iff \(\text{Models}(\phi) \cap \text{Models}(\psi) = \emptyset\). A set of persuasion goals \(\Phi\) is pairwise disjoint iff for each \(\phi, \psi \in \Phi\), \(\phi, \psi\) are disjoint. A set of persuasion goals \(\Phi\) is exhaustive iff \(\bigcup_{\phi \in \Phi} \text{Models}(\phi) = \text{Args}(G)\).

**Example 3.** Let \(\text{Args}(G) = \{A, B\}\). Consider the persuasion goals \(\Phi = \{\phi_1, \phi_2, \phi_3\}\) where \(\phi_1 = A \land B \land \neg C\), \(\phi_2 = ((A \land \neg B) \lor (\neg A \land B)) \lor \neg C\), and \(\phi_3 = \neg A \land \neg B\lor C\). Since \(\text{Models}(\phi_1) = \{110\}\), \(\text{Models}(\phi_2) = \{100, 010\}\), and \(\text{Models}(\phi_3) = \{111, 101, 011, 001, 000\}\). Hence, \(\Phi\) is pairwise disjoint and exhaustive.

**Proposition 1.** If \(\Phi = \{\phi_1, \ldots, \phi_n\}\) is a set of persuasion goals such that \(\Phi\) is exhaustive and pairwise disjoint, then \(P(\phi_1 \lor \ldots \lor \phi_n) = 1\).

For a persuasion goal \(\phi\), and a mass distribution \(P\) (i.e. the user model), \(P(\phi)\) is the probability that \(\phi\) is believed. If \(P(\phi)\) is low at the start of the dialogue, then the system aims to terminate the dialogue with \(P(\phi) > 0.5\) (i.e. according to the user model, the user believes \(\phi\)).

### 5 Redistributing mass

To update a user model during a dialogue, we need to redistribute mass. A mass redistribution function takes a probability distribution and formula \(\alpha\), and returns a revised probability distribution. There are many possibilities for this. In this paper, we focus on refinement which redistributes mass from models not satisfying \(\alpha\) to models satisfying \(\alpha\).

**Definition 5.** Let \(\alpha \in \text{Formulae}(G)\) be a literal, let \(P\) be a mass distribution, and let \(k \in [0, 1]\). A refinement function, denoted \(H_{\alpha}^k(P)\), returns the mass distribution \(P'\) as follows where \(X \in \text{Models}(G)\):

\[
P'(X) = \begin{cases} P(X) + (k \times P(h_\alpha(X))) & \text{if } X \models \alpha \\ (1 - k) \times P(X) & \text{if } X \not\models \alpha \end{cases}
\]

and where \(h_\alpha(X) = X \setminus \{A\}\) when \(\alpha\) is of the form \(\forall \beta\), and \(h_\alpha(X) = X \cup \{A\}\) when \(\alpha\) is of the form \(\neg A\).

See Table 1 for examples of redistribution. For redistribution, \(h_\alpha\) returns the model closest to \(X\) but with \(\alpha\) no longer satisfied. If \(k = 1\), then all the mass is transferred from the models not satisfying \(\alpha\) to models satisfying \(\alpha\). If \(k < 1\), then only a proportion is transferred. We use \(k < 1\) in the next section to give finer grained modelling of users.

For each \(\alpha\), we can partition \(\text{Models}(G)\) into the set of models that satisfy \(\alpha\), i.e. \(\text{Sat}(\alpha) = \{X \in \text{Models}(G) \mid X \models \alpha\}\), and the set of models that do not satisfy \(\alpha\), i.e. \(\text{Unsat}(\alpha) = \{X \in \text{Models}(G) \mid X \not\models \alpha\}\).

**Proposition 2.** For each \(\alpha \in \text{Formulae}(G)\), the function \(h_\alpha\) is a bijection from \(\text{Sat}(\alpha)\) to \(\text{Unsat}(\alpha)\).

**Proposition 3.** If \(\alpha \in \text{Formulae}(G)\), and \(k = 0\), and \(H_\alpha^0(P) = P'\), then \(P = P'\).

**Proposition 4.** If \(\alpha \in \text{Formulae}(G)\), \(H_\alpha^k(H_\alpha^k(P)) = H_\alpha^k(P)\).

**Proposition 5.** If \(A \in \text{Args}(G)\), \(P(A) = 1\) iff \(H_\alpha^k(P) = P\).

**Proposition 6.** Let \(\text{Args}(G) = \{A_1, \ldots, A_n\}\). If \(H_{A_1}^k(\ldots H_{A_n}^k(P)) = P'\), then there is a model \(X \in \text{Models}(G)\) such that \(X = \{A_1, \ldots, A_n\}\) and \(P(X) = 1\).

Note that the refinement update is not reversible. In other words, given an update \(H_\alpha^k\), there is no \(k'\) and \(\beta\) such that \(H_{\beta}^{k'}(H_{\beta}^{k'}(P)) = P\). We illustrate this in the next example.

**Example 4.** Consider \(P_3\) in the table. From \(P_3\), we can identify many distributions \(P\) such that \(H_{A_1}^k(P) = P_3\). We give two distributions \(P_1\) and \(P_2\), such that \(H_{A_1}^k(P_1) = P_3\) and \(H_{A_1}^k(P_2) = P_3\). So knowing \(\alpha\) and \(k\) is insufficient to identify \(\beta\) and \(k'\) such that \(H_{\beta}^{k'}(H_{\beta}^{k'}(P)) = P\).

The following property shows that the only argument with a changed assignment is the one being explicitly updated where \(\text{Atom}(A) = A\) and \(\text{Atom}(\neg A) = A\) for \(A \in \text{Args}(G)\).

**Proposition 7.** For literal \(\alpha \in \text{Formulae}(G)\), & \(k \in [0, 1]\), let \(H_\alpha^k(P) = P'\). If \(B \not\models \text{Atom}(\alpha)\), then \(P'(B) = P(B)\).

Refinement is associative for multiple updates as shown in the following result.

**Proposition 8.** If \(\alpha, \beta\) are literals, s.t. \(\alpha\) is not the complement of \(\beta\) (i.e. \(\alpha \not\equiv \neg \beta\)), and \(P\) is a mass distribution, and \(k, k' \in [0, 1]\), then \(H_{\alpha}^k(H_{\beta}^{k}(P)) = H_{\alpha}^{k}(H_{\beta}^{k'}(P))\).

Next, we introduce notation for a set of updates \(\Phi\).

**Definition 6.** Let \(\Phi \subseteq \text{Formulae}(G)\) be a set of literals, let \(P\) be a mass distribution, and let \(k \in [0, 1]\). A compound refinement function, denoted \(H_{\Phi}^k\), is defined as follows: (1) \(H_{\Phi}^k(P) = P\) when \(\Phi = \emptyset\); and (2) \(H_{\Phi}^k(P) = H_{\Phi}^k(H_{\Phi}^k(P))\) when \(\Phi = \Phi \setminus \{\alpha\}\) for some \(\alpha \in \Phi\).

Because of its useful properties, we use the refinement function for updating user models in the next section.

### 6 Updating user models

Given a mass distribution \(P\), representing a user’s beliefs at the current state of the dialogue, we want to update the model depending on the move made. For this, we introduce the notion of an update method which generates a mass distribution \(P_i\) from \(P_{i-1}\) based on the move \(D(i)\). Each method is defined as a rule with a condition (defined in terms of the current state of the dialogue, the current mass distribution, and the structure of the graph), and a consequent that specifies the redistribution.
Definition 7. For step \( i \) in the dialogue, the naive method generates \( P_i \) from \( P_{i-1} \) as follows.

If \( D(i) = A! \), then \( P_i = H_A^i(P_{i-1}) \).

Example 5. For Figure 1 with dialogue \([A!, B!; \perp] \) and the naive method. Let the initial mass be \( P_0(011) = 0.3, P_0(010) = 0.2, P_0(001) = 0.3, \) and \( P_0(000) = 0.2 \). After \( A! \), \( P_1(111) = 0.3 \), \( P_1(110) = 0.2 \), \( P_1(101) = 0.3 \), and \( P_1(100) = 0.2 \). After \( B! \), \( P_2(111) = 0.6 \), and \( P_2(110) = 0.4 \). Note, \( P_2 \) is not rational.

The naive method fails the rational+ property. The trusting method (defined next) satisfies the rational+ property by lowering belief in attackers and attackers of the posist.

Definition 8. For step \( i \) in the dialogue, the trusting method generates \( P_i \) from \( P_{i-1} \) as follows, where \( \Phi = \{ \neg C \mid (A, C) \in \text{Attacks}(G) \} \) or \( (C, A) \in \text{Attacks}(G) \).

If \( D(i) = A! \), then \( P_i = H_A^i(H_A^i(P_{i-1})) \).

Example 6. Consider Figure 1 with dialogue \([A!, !] \) and the trusting method. Let the initial mass be \( P_0(011) = 0.3, P_0(010) = 0.2, P_0(001) = 0.3, \) and \( P_0(000) = 0.2 \). After \( A! \), \( P_1(111) = 0.6, P_1(101) = 0.6, \) and \( P_1(100) = 0.4 \).

The strict method (defined next) only allows a posist to update the belief to 1 when there is no attacker of the posist that is believed.

Definition 9. For step \( i \) in the dialogue, the strict method generates \( P_i \) from \( P_{i-1} \) as follows, where \( \Phi = \{ \neg C \mid (A, C) \in \text{Attacks}(G) \} \).

If \( D(i) = A! \),
and for all \( (B, A) \in \text{Attacks}(G) \), \( P_{i-1}(B) \leq 0.5 \),
then \( P_i = H_{B}^i(H_A^i(P_{i-1})) \), else \( P_i = P_{i-1} \).

Example 7. Consider Figure 1 with dialogue \([A!, C!, A!, \perp] \) and the strict method. Let the initial mass be \( P_0(111) = 0.2, P_0(110) = 0.3, P_0(011) = 0.3, \) and \( P_0(010) = 0.2 \). After \( A! \), \( P_1(111) = 0.2, P_1(110) = 0.3, P_1(011) = 0.3, \) and \( P_1(010) = 0.2 \). After \( C! \), \( P_2(011) = 0.5, P_2(001) = 0.5 \). After \( A! \), \( P_3(011) = 1.0 \).

The ambivalent method is motivated by the idea that people do not entirely believe what they are told (unless corroborated), and as shown in [Rahwan et al., 2010] they do not entirely disbelieve an argument when defeated.

Definition 10. For step \( i \) in the dialogue, the ambivalent method generates \( P_i \) from \( P_{i-1} \) as follows, where \( \Phi = \{ \neg C \mid (C, A) \in \text{Attacks}(G) \} \).

If \( D(i) = A! \),
and for all \( (B, A) \in \text{Attacks}(G) \), \( P_{i-1}(B) \leq 0.5 \),
then \( P_i = H_B^{0.75}(H_A^{0.75}(P_{i-1})) \), else \( P_i = P_{i-1} \).

Example 8. Consider Figure 1 with dialogue \([C!, A!, \perp] \) with the ambivalent method. Let the initial mass be \( P_0(111) = 0.5, P_0(110) = 0.5, P_0(011) = 0.5, \) and \( P_0(010) = 0.5 \). After \( C! \), \( P_1(111) = 0.2, P_1(110) = 0.2, P_1(011) = 0.2, \) and \( P_1(010) = 0.2 \). After \( A! \), \( P_2(111) = 1/4, P_2(110) = 1/4, P_2(011) = 1/4, \) and \( P_2(010) = 1/4 \). After \( C! \), \( P_3(011) = 0.25, P_3(010) = 0.25 \).

Table 2: Dynamic properties for update methods

<table>
<thead>
<tr>
<th>Method</th>
<th>Naive</th>
<th>Trusting</th>
<th>Strict</th>
<th>Ambivalent</th>
</tr>
</thead>
<tbody>
<tr>
<td>Credulous</td>
<td>✓</td>
<td>✓</td>
<td>×</td>
<td>×</td>
</tr>
<tr>
<td>Minimal</td>
<td>✓</td>
<td>×</td>
<td>×</td>
<td>×</td>
</tr>
<tr>
<td>Rational+</td>
<td>×</td>
<td>✓</td>
<td>×</td>
<td>✓</td>
</tr>
<tr>
<td>Binary</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Restricted</td>
<td>×</td>
<td>×</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

Proposition 9. If \( P \) is a mass distribution, and \( P' \) is generated from \( P \) by the naive, trusting, strict or ambivalent method, then \( P' \) is a mass distribution.

Definition 11. Let \( \sigma \) be an update method (such as naive, trusting, strict, or ambivalent), let \( D = \{m_1, \ldots, m_n\} \) be a dialogue, and let \( \{P_0, P_1, \ldots, P_n\} \) be a sequence of user models such that for each \( i \in \{1, \ldots, n\}, P_i \) is obtained from \( P_{i-1} \) by the \( \sigma \) update method. We call \( P_0 \) the initial mass, and \( P_n \) the final mass obtained from \( D \) with respect to \( P_0 \) and \( \sigma \), which we denote by \( \sigma(P_0, D) = P_n \).

The update methods given in this section are meant to illustrate a range of methods. We give the satisfaction of dynamic properties in Table 2. Further options for update methods can use alternatives to Definition 5 or take meta-level criteria into account (e.g. preferences [Amgoud and Cayrol, 2002] and values [Oren et al., 2012]).

7 Maximizing expected utility

We start by briefly reviewing the notion of a lottery. A lottery \( L \) is a probability distribution over a set of possible outcomes. A lottery with possible outcomes \( \pi_1, \ldots, \pi_n \) that are pairwise disjoint and that is exhaustive (i.e. one of them is guaranteed to occur), with probabilities \( p_1, \ldots, p_n \), respectively, is written as \( \{\pi_1, \pi_1; \ldots; \pi_n, \pi_n\} \). For a utility function \( U \), the expected utility of a lottery \( L \) is \( \sum_{i=1}^{n} p_i \times U(\pi_i) \).

We adapt the notion of a lottery in a straightforward way for our purposes.

Definition 12. Let \( D \) be a dialogue, let \( S = \{\phi_1, \ldots, \phi_n\} \) be a set of disjunct and exhaustive outcomes, let \( P_0 \) be the initial mass, \( \sigma \) be an update method, and let \( \sigma(P_0, D) = P \), and let \( U \) be a utility function. The lottery for \( P, U, S \) is the following:

\[ \text{Lot}(P, U, S) = \{P(\phi_1) \times U(\phi_1); \ldots; P(\phi_n) \times U(\phi_n)\} \]

Then the expected utility for \( P, U, S \) is \( \text{EU}(P, U, S) = (P(\phi_1) \times U(\phi_1)) + \ldots + (P(\phi_n) \times U(\phi_n)) \).

A dialogue \( D \) is optimal with respect to an initial mass \( P_0 \), update method \( \sigma \), utility function \( U \), and \( \sigma(P_0, D) = P \), when \( \text{EU}(P, U, S) \) is maximized.

Example 9. Let \( \text{Args}(G) = \{A, B\} \) and \( \text{Attacks}(G) = \{(B, A)\} \). For the naive method \( \sigma(P_0, D) = P \) where \( P_0(11) = 0, P_0(10) = 0.6, P_0(01) = 0.2, P_0(00) = 0.2 \), \( U(A) = 1 \) and \( U(\neg A) = -1 \), \( D_1, D_2, D_3 \) and \( D_4 \) are optimal.

<table>
<thead>
<tr>
<th>( D )</th>
<th>( \text{EU}(P, U, S) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( D_1 )</td>
<td>(1 × 1) + (0 × −1) = 1</td>
</tr>
<tr>
<td>( D_2 )</td>
<td>(1 × 1) + (0 × −1) = 1</td>
</tr>
<tr>
<td>( D_3 )</td>
<td>(0.6 × 1) + (0.4 × −1) = 0.2</td>
</tr>
<tr>
<td>( D_4 )</td>
<td>(1 × 1) + (0 × −1) = 1</td>
</tr>
</tbody>
</table>
**Proposition 10.** Let $\sigma$ be the naive or trusting method. $A \in \text{Args}(G)$ and $P_0$ is the initial mass. If $(B, A) \in \text{Attacks}(G)$ and there is a $i$ such that $D(i) = B$, then there is a $D' \subseteq D$ such that $\text{EU}(P', U, S) = \text{EU}(P, U, S)$ where $\sigma(P_0, D) = P$ and $\sigma(P_0, D') = P'$.

Given the above result, the next example only considers dialogues without positing of attackers of the goal.

**Example 10.** Consider Figure 1 with the persuasion goal $A$ and the trusting method. If $P_0(111) = P_0(110) = P_0(011) = P_0(010) = 0.25$, then $D_1$, $D_2$ and $D_3$ are optional.

<table>
<thead>
<tr>
<th>$D$</th>
<th>$\text{EU}(P, U, S)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$D_1$</td>
<td>$(1 \times 1) + (0 \times -1) = 1$</td>
</tr>
<tr>
<td>$D_2$</td>
<td>$(1 \times 1) + (0 \times -1) = 1$</td>
</tr>
<tr>
<td>$D_3$</td>
<td>$(0.5 \times 1) + (0.5 \times -1) = 0$</td>
</tr>
</tbody>
</table>

The next result shows attackers cannot improve the utility of a dialogue with the strict/amibivalent methods.

**Proposition 11.** Let $\sigma$ be the strict or ambivalent method. Let $A \in \text{Args}(G)$ be the persuasion goal. If $(B, A) \in \text{Attacks}(G)$ and there is an $i$ such that $D(i) = B$, then there is a $D' \subseteq D$ such that $\text{EU}(P', U, S) \geq \text{EU}(P, U, S)$ where $\sigma(P_0, D) = P$ and $\sigma(P_0, D') = P'$.

We now turn to whether repeating a move can have a benefit under any of the assumed update methods.

**Definition 13.** For dialogues $D$ and $D'$, $D'$ is a maximal non-repeating subsequence of $D$ if the following conditions hold: (1) $D' \subseteq D$; (2) Set($D'$) = Set($D$); and (3) if $D'' \subseteq D'$, then Set($D''$) $\neq$ Set($D$).

**Example 11.** Consider $D = [A!, B!, A!, C!, D!]$. There are two maximal non-repeating subsequences $D_1 = [A!, B!, A!, C!, D!]$ and $D_2 = [B!, A!, C!, D!]$.

**Proposition 12.** Let $P_0$ be the initial mass, and $\sigma$ is the naive or trusting method. For all dialogues $D$, if $D'$ is a maximal non-repeating subsequence of $D$, and $\sigma(P_0, D) = P$ and $\sigma(P_0, D') = P'$, then $\text{EU}(P, U, S) = \text{EU}(P', U, S)$.

With naive and trusting methods, we can reduce a repeating sequence to a non-repeating sequence by deleting the repeated arguments. This is sufficient for the utility of the two sequences to be the same. However, with the strict and ambivalent methods, all we can ensure is that for every dialogue, there exists another dialogue that is non-repeating and with the same utility. To show this, we require the following.

**Definition 14.** An update tree $T$ for $A \in \text{Args}(G)$, and a mass distribution $P$, is the smallest tree such that

1. $T$ has a finite number of nodes and $A$ is the root
2. for each $A_i$ at an odd level in $T$, if $P(A_i) > 0.5$, and $(A_j, A_i) \in \text{Attacks}(G)$, then $A_j$ is a child of $A_i$ in $T$
3. for each $A_i$ at an even level in $T$, if $(A_j, A_i) \in \text{Attacks}(G)$, then $A_j$ is a child of $A_i$ in $T$
4. for each $A_i$ at an odd level in $T$, there is no $A_j$ at an odd level in $T$ such that $(A_j, A_i) \in \text{Attacks}(G)$,
   and where the root is at level 1, and for each node at level $i$, its children are at level $i + 1$.

In an update tree, all the odd level arguments are disbelieved, and all the even level arguments are believed. Furthermore, we can regard the odd level arguments as “defenders” of the root argument, and the even level arguments as direct or indirect “attackers” of the root argument. When every leaf is at an odd level, then we can use this tree to design a dialogue for a successful persuasion (as illustrated in Example 12), but as we illustrate in Example 13, not every graph and mass distribution has an update tree.

**Example 12.** Consider the following graph where $A$ is the persuasion goal.

In this graph and persuasion goal, with this mass distribution, there is no update tree.

**Proposition 13.** If $T$ is an update tree for $A \in \text{Arg}(G)$, and mass distribution $P$, the set $\Omega = \{B \in \text{Arg}(G) \mid B$ is a node in level $i$ and $i$ is odd$\}$ is conflictfree (i.e. there are no arguments $(B, C) \in \text{Attacks}(G)$ s.t. $(B, C) \subseteq \Omega$).

Given an update tree, the defenders in a branch can be posited in sequence starting from the leaf. Doing this leads to the following results.

**Proposition 14.** For persuasion goal $A \in \text{Arg}(G)$, with initial mass $P_0$, where $P_0(A) \leq 0.5$, there is an update tree for $A$ and $G$, where every leaf is at an odd level, if there is a dialogue $D = [B_1!, \ldots, B_z!, \perp]$ such that $P(A) > 0.5$, where $\sigma$ is the strict or ambivalent method and $\sigma(P_0, D) = P$.

**Proposition 15.** For any dialogue $D_1$, there is a dialogue $D_2$ such that $D_2$ is non-repeating, and $\text{EU}(P_1, U, S) = \text{EU}(P_2, U, S)$, where $P_0$ is the initial mass, and $\sigma$ is the strict or ambivalent method and $\sigma(P_0, D_1) = P_1$ and $\sigma(P_0, D_2) = P_2$ and $U$ is the utility function.

In general, shorter dialogues are preferable to longer dialogues if we want to decrease the risk of the user disengaging (because they become bored with the dialogue). We can take this into account in the expected utility assignment by normalizing it by the length of the dialogue (e.g. by dividing...
EU(P\(, U, S\)) by the number of posits in \(D\). Returning to Example 10, the revised expected utilities could be such that \(D_1\) is 0.5, \(D_2\) is 1, \(D_3\) is 0, and \(D_4\) is 0.5, and therefore, the optimal dialogue would be \(D_2\).

8 Accuracy of the user model

At each step \(i\) of the dialogue, the user model \(P_i\) is an estimate of the user’s actual mass distribution \(\hat{P}_i\). This raises the question of how to compare \(P_i\) and \(\hat{P}_i\). We use the total variation distance (defined below) because of simplicity. Alternatives include the Kullback-Leibler divergence or an \(f\)-divergence.

**Definition 15.** The total variation distance between mass distributions \(P_1\) and \(P_2\), denoted Distance\((P_1, P_2)\), is \(1/2 \times \sum_{A \in \text{Args}(G)} |P_1(A) - P_2(A)|\).

**Example 14.** Distance\((P_1, P_2)\) = 0.6. when \(P_1(11) = 0.4\), \(P_2(10) = 0.2\), \(P_1(01) = 0.2\), \(P_2(00) = 0.2\), \(P_2(11) = 0.0\), \(P_2(01) = 0.2\), and \(P_2(00) = 0.8\).

In order to decrease the distance between the user model and the user’s actual mass distribution, we can use queries. For instance, a dialogue can start with a sequence of queries before giving posits. So we have a sequence of queries/answers \([Q_1, R_1, \ldots, Q_n, R_n]\) where each \(Q_i\) is a query and each \(R_i\) is the response by the user and is either \(Y\) or \(N\). Note, we can use a finer grained scale for the user reply to \(A\)’s (e.g. from 0 to 10 where 0 denotes completely believes \(A\), 9 denotes strongly believes \(A\), etc.) instead of \(Y\) and \(N\). The model is updated as follows.

- If \(D(i) = A\) and \(D(i + 1) = Y\), then \(P_{i+1} = H_A(P_i)\)
- If \(D(i) = A\) and \(D(i + 1) = N\), then \(P_{i+1} = H_A(P_i)\)

**Example 15.** Consider the dialogue \([A?, Y, B?, N]\) where \(\text{Args}(G) = \{A, B, C\}\). Suppose the initial mass \(\rho_0\) is a uniform distribution. Then \(P_2(111) = P_2(110) = P_2(101) = P_2(100) = 1/4\), and \(P_2(01) = P_2(100) = 1/2\).

A user is categorial in dialogue \(D\) iff for all \(i \in \{1, \ldots, \text{Length}(D)\}\), if \(D(i) = A?\), then \(P_{i+1}(A) = P_{i+1}(A)\). The following results show that querying the user when s/he is categorical improves the user model.

**Proposition 16.** Assume that the user is categorical in dialogue \(D\). For each \(i\), if \(D(i) = A?\), then Distance\((P_i, \hat{P}_i) \geq\) Distance\((P_{i+1}, \hat{P}_{i+1})\)

**Proposition 17.** If \([Q_1, R_1, \ldots, Q_n, R_n]\) is the initial sequence of moves of dialogue \(D\) where each \(Q_i\) is a query and each \(R_i\) is a reply, and for every \(A \in \text{Args}(G)\), there is a query \(Q_i = A?\), then \(P_m = \hat{P}_m\).

If queries are at no cost, then we can query the user about all arguments in the graph. However, in practice, it may be inappropriate to query the user about every argument as asking the user more than a few queries raises the risk of the user disengaging. We will investigate this trade-off in future work.

9 Comparison with the literature

There are a number of proposals that formalize aspects of persuasion. Most are aimed at providing protocols for dialogues (e.g. Prakken, 2005; 2006; Fan and Toni, 2011; Caminada and Podlaszewski, 2012), but strategies for persuasion, in particular taking into account beliefs of the opponent are under-developed. See [Thimm, 2014] for a review of strategies in multi-agent argumentation.

There are a number of proposals for using probability theory in argumentation including the epistemic approach (e.g. [Thimm, 2012; Hunter, 2013]) and the constellations approach (e.g. [Li et al., 2011; Hunter, 2012]) but these do not consider dialogues.

Persuasion has been considered through uncertainty modelling of the audience [Oren et al., 2012], but this uncertainty is with respect to the structure of the graph rather than beliefs, and there is no consideration of dialogues or strategies. A probabilistic model of the opponent has been used in a dialogue strategy allowing the selection of moves for an agent based on what it believes the other agent believes [Rienstra et al., 2013]. But this assumes symmetric dialogues, and the uncertainty concerns what the opposing agent is aware of rather than what it believes. In another approach to a probabilistic opponent model, the history of previous dialogues is used to estimate the arguments that an agent might put forward [Hadjininikos et al., 2013]. But this assumes symmetric dialogues, with a method for updating the opponent model that is of exponential complexity, and there is no consideration of how utility theory could be employed.

Utility theory has been considered previously in argumentation (for example [Rahwan and Larson, 2008b; Riveret et al., 2008; Matt and Toni, 2008; Oren and Norman, 2009]) though none of these represent the uncertainty of moves made by each agent in argumentation. There is an approach that combines probability theory and utility theory to identify outcomes with maximum expected utility where outcomes are specified as particular arguments being included or excluded from extensions [Hunter and Thimm, 2014a], but it is based on the constellations approach to uncertainty in argumentation (i.e. uncertainty about what the structure of the graph is) as opposed to the epistemic approach considered in this paper, and there is no consideration of updates to the model. Strategies in argumentation have also been analyzed using game theory [Rahwan and Larson, 2008a; Fan and Toni, 2012], though these are more concerned with issues of manipulation, rather than persuasion.

10 Discussion

In this paper, we have considered a framework for asymmetric dialogues, with a general definition for probabilistic user models, and a general definition for updating user models in terms of mass redistributions. The type of dialogue is a persuasion dialogue. However, the persuader uses the user model to determine whether the persuasion goal is believed, and so it is an indirect judgment since the persuadee has not declared whether or not they are persuaded when the dialogue has terminated. Nonetheless, the user model may be a good representative of the persuadee as it can either be generated by querying the user, or by learning from previous interactions with the user or similar users. Some recent studies indicate the potential viability of an empirical approach [Cerutti et al., 2014; Rosenfeld and Kraus, 2015].
References


