Strategic Sequences of Arguments for Persuasion Using Decision Trees

Emmanuel Hadoux and **Anthony Hunter**
Department of Computer Science, University College London
London, UK
{e.hadoux, anthony.hunter}@ucl.ac.uk

Abstract

Persuasion is an activity that involves one party (the persuader) trying to induce another party (the persuadee) to believe or do something. For this, it can be advantageous for the persuader to have a model of the persuadee. Recently, some proposals in the field of computational models of argument have been made for probabilistic models of what the persuadee knows about, or believes. However, these developments have not systematically harnessed established notions in decision theory for maximizing the outcome of a dialogue. To address this, we present a general framework for representing persuasion dialogues as a decision tree, and for using decision rules for selecting moves. Furthermore, we provide some empirical results showing how some well-known decision rules perform, and make observations about their general behaviour in the context of dialogues where there is uncertainty about the accuracy of the user model.

Introduction

Computational models of argument can potentially be used for systems to persuade users to change their behaviour (*e.g.*, to eat less, to exercise more, to vote) (Hunter 2014a).

In this work, we make no assumption on the knowledge or on the behaviour of the persuadee. We propose a general framework representing persuasion problems as decision problems to solve them exactly using state-of-the-art methods in decision theory. Composed of several independent units, the framework is general enough to be able to take into account various types of behaviour for both the persuader and the persuadee. The determination of a dialogue as being succesful in the persuasion can be done using, for instance, different Dung's semantics (1995) or other types of functions (*e.g.*, degree of belief in a persuasion goal).

Persuasion Dialogues

A *persuader* (the proponent) has a dialogue with a *persuadee* (the opponent) to make her believe (or disbelieve) some combination of arguments (*e.g.*, to do more exercise or to eat healthier food). For the sake of simplicity, in this paper, we deal with two agents and a singleton goal. However, our work can be extended to more agents and any number of goals as long as only one persuader is involved. Building upon Dung's abstract argumentation (1995), a dialogue concerns an argument graph G without self-attacks where $\mathsf{Args}(G)$ is the set of arguments in G, and $\mathsf{Attacks}(G)$ is the set of attack relations in G.

More formally, a *persuasion dialogue* is a sequence of moves $D = [m_1, \dots, m_h]$. In this work, a move consists in positing an argument $a \in \mathsf{Args}(G)$. The attacks to and from this argument in relation to the arguments already posited come from the original graph. Equivalently, we use D as a function with an index position i to return the move at i (*i.e.*, $D(i) = m_i$). The parameter h is the *horizon* of the debate, *i.e.*, the maximum number of moves that can be played. It is justified by the need to keep the persuadee engaged. A shorter debate (*i.e.*, a smaller value for h) gives more chance to keep the persuadee in the debate until the end. However, it also lowers the number of ways to make a valid point in the debate (see Section experiments for a discussion).

Probabilistic User Models

Each odd (resp. even) move in the dialogue is a persuader (resp. persuadee) move. However, the persuadee moves are played with respect to the arguments she believes in, in reaction to the persuader positing an argument. Therefore, an efficient strategy needs to take into account the possible subsets of arguments the persuadee believes in. Indeed, an agent is unlikely to posit arguments she does not have faith in. To that end, the persuader keeps and updates a *belief model* of the persuadee and uses it in her decision process. We use the epistemic approach to probabilistic argumentation (Thimm 2012; Hunter 2013; Hunter and Thimm 2014; Baroni, Giacomin, and Vicig 2014), defining a model as a *mass distribution* over all possible subsets of believed arguments.

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

For a mass distribution P and $A \in \mathsf{Args}(G)$, $P(A)$ 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 $P(A) > 0.5$ the agent believes the argument to some degree, whereas when $P(A) \leq 0.5$ the agent disbelieves the argument to some degree. Each

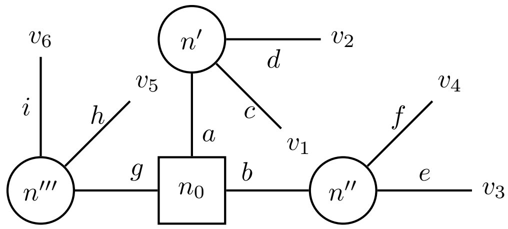


Figure 1: Example of a decision tree

element of the mass distribution is a possible world $X \subseteq \mathsf{Args}(G)$, where X (resp. $\mathsf{Args}(G) \setminus X$) is the subset of arguments believed (resp. disbelieved) by the persuadee.

The persuader uses a mass distribution P as a belief model of the persuadee and updates it at each stage of the dialogue depending on the move made. For this, we consider the notion of an update method $\sigma(P_{i-1}, D(i)) = P_i$ generating a mass distribution P_i from P_{i-1} based on the move $D(i)$.

Decision Trees for Dialogues

In order to reach the persuasion goal, the persuader has to posit the right sequence of arguments with respect to this model. We use the very well known notion of *decision trees* and adapt them to persuasion, in order to compute a *policy*, an action to perform (*i.e.*, an argument to posit) in each possible state of the dialogue (see next section for more details).

A decision tree represents all the possible combinations of decisions and outcomes of a sequential decision-making problem. In a two-agents problem, a path from the root to any leaf crosses alternatively nodes associated to the proponent (called *decision nodes* in this work) and nodes associated to the opponent (called *chance nodes* or *nature nodes*). In the case of a dialogue represented as a decision tree, a path is one possible permutation of the argument set, *i.e.*, one possible complete dialogue between the two agents. If horizon h is smaller than the number of arguments, every execution (and thus path) is at most of length h. In this case, it is a permutation of a subset of the argument set. An edge between any two nodes n and n' in the tree is the decision that has to be taken by the corresponding agent in order to transition from node n to node n'.

Figure 1 shows an example of a decision tree for $h = 2$. Square nodes are decision nodes and circle nodes represent chance nodes. The central square node is the root of the tree while $v_1, \ldots, v_6$ are the leaves.

Note that no assumption is made on the behaviour of the persuadee, *i.e.*, we do not consider that the opponent is adversarial, or compliant, or plays strategically, etc. However, we assume that the behaviour of the opponent can be represented by a probability distribution in each node, called the *decision model* of the persuadee. This probability distribution gives the probability of each of the possible moves that can be taken in the node. The decision model is unknown to the persuader. However, this stochastic assumption is not a restriction. Indeed, as each node of the decision tree is unique and represents a unique part of history, any global behaviour of the persuadee can be represented as a set of the probability distributions, one for each chance node.

Algorithm 1: Decision tree building

Function `Build` (*arguments, h, goal*)
    $tree$ = root node
3   **foreach** *permutation* p *of arguments of size* h **do**
        $current$ = root node
        $type$ = chance node
        $model$ = uniform distribution
        foreach *argument* a *in* p *except last one* **do**
            n = add a node of type $type$ linked to $currentNode$ by edge tagged a
            $currentNode = n$
            $type$ = other type
            $model = \texttt{UpdateBelief}(model, a)$
12      $val = \texttt{Evaluate}(model, p, goal)$
        add leaf of value val to $currentNode$ linked by edge tagged $last(p)$
    return $tree$

Function `UpdateBelief` (*model, a*)
16 apply `Ambivalent` (model, a)
    return *model*

Function `Evaluate` (*model, p, goal*)
    return $\frac{1}{2}$ *belief of goal in model* $+ \frac{1}{2}\mathbb{1}_{goal \in p}$

Valuation of an execution

The values associated with the leaves are the valuation given to one particular dialogue. For instance, v_1 is the outcome of the dialogue $D = [a, c]$, representing how desirable is this execution with respect to (1) the persuasion goal of the persuader and (2) the final belief in this goal by the persuadee. Therefore, they are computed only for the persuader, from her point of view. For the sake of simplicity, in this paper, the value v_i of dialogue i is the average of (2) the value of the final belief in the goal and (1): a value of 1 if the goal has been posited in dialogue i or 0 otherwise.

This function can be replaced to take into account different goals and interactions between them (*e.g.*, synergies) or combinations more complex than the average (*e.g.*, ordered weighted average (Yager 1988)). Note that Dung's dialectical semantics (1995) can also be used. For instance, the value of the goal can be 1 if it is in the grounded extension, 0 otherwise.

Algorithm 1 presents the whole building process of the decision tree started with $\texttt{Build}(\mathsf{Args}(G), horizon, goal)$. Line 12 (resp. 16) is the function to replace in order to use a different evaluation (resp. update) function. Note that the notion of awareness to some arguments can be captured by a different way to build the permutations (see Line 3). Indeed, instead of the permutations of size h, any subset of each permutation, representing the arguments the agent is aware of, can be added as a valid execution. This will lead to paths of different lengths, handled by decision trees without any modification.

Example 1 *We use a refinement function for redistributing mass from possible worlds not satisfying α to possible worlds satisfying α. We define the satisfaction operator such as $X \subseteq \mathsf{Args}(G)$ satisfies an argument A (resp. its negation) (denoted $X \models A$) if A is (resp. is not) in X. Let α be an argument or its negation, P a mass distribution, and* $k \in [0, 1]$. *The refinement function (Hunter 2015), denoted $H^k_\alpha(P)$, returns the mass distribution P' as follows:*

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

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

In Example 1, h_α returns the possible world closest to X but with α no longer satisfied. If $k = 1$, then all the mass is transferred from the worlds not satisfying α to worlds satisfying α. If $k < 1$, then only a proportion is transferred. This gives flexibility to model updates in different types of persuadee. For instance, if we want to model a persuadee who does not fully believe arguments played by the persuader, we can use $k < 1$ to update the model so that the argument is not fully believed in the model.

Note that, even though the definition of the refinement function is given for one subset X, it needs to be applied on all possible subsets to update the whole belief model. For $|\mathsf{Args}(G)|$ arguments, $2^{|\mathsf{Args}(G)|}$ subsets exist. This may lead to a computationaly intractable problem. To address this issue, a method exploiting the structure of the argument graph G has been recently developed (Hadoux and Hunter 2016). It uses conditional probabilistic independance and graphs of 50 arguments have been handled satisfactorily.

Example 2 *(Example 1 cont'd) Let us use the ambivalent update method (Hunter 2015), that raises the belief in a posit, and lowers the belief in its attackees. At step i in the dialogue, the **ambivalent method** generates P_i from P_{i-1} as follows, where $\Phi = \{\neg C \mid (A, C) \in \mathsf{Attacks(G)}\}$.*

$$\textit{If } D(i) = A!, \textit{ then } P_i = H^{0.75}_\Phi(H^{0.75}_A(P_{i-1})).$$

where $H^k_{\{\alpha_1,\ldots,\alpha_n\}}(P) = H^k_{\alpha_1}(\ldots H^k_{\alpha_n}(P))$.

The ambivalent method can be replaced by any update method (see (Hunter 2015) for more methods). The aim of this flexibility is to model different kinds of persuadees with different kinds of behaviours, some of which are not rational.

Decision Rules and Optimization

Once the decision tree is built, we need to select, in each decision node, an action to perform (*i.e.*, an argument to posit in each state of the debate) from the point of view of the persuader. This association of a node with the action to perform in this node is called a *policy*. The aim is to compute an *optimal policy*, the best action to perform in each decision node. For this, we use a decision rule, composed by two parts: one aggregating the values of all children of a decision node and the other for aggregating all the children of a chance node.

For the sake of simplicity, we will only consider the rules maximizing the value in each decision node. The difference will be made on the part of the rule dealing with the children of each chance node. This work presents (but is not limited to) commonly used rules in decision theory.

The second part of a decision rule corresponds to an assumption of the decision model of the persuadee such that it can be translated to a probability distribution. We call this distribution the *profile* of the persuadee, with respect to a given decision rule. Note that the profile is a consequence of the choice of a decision rule and changing it amounts to a change of decision rule. The performance of the rule is directly correlated with the difference between the profile and the actual decision model. We call this difference the *error*.

We list in the following four decision rules. Each rule is presented with the desired behaviour induced in the persuader, its formula and the profile associated.

Optimistic selection (MaxiMax) The MaxiMax rule is applied if the persuader wants to adopt an optimistic behaviour, *i.e.*, to consider that the persuadee wants to maximize the outcome as well. Recall that the outcome is given from the point of view of the persuader only. Let n be a chance node and $(n_1, \ldots, n_i, \ldots, n_k)$ the children (decision nodes and/or leaves) of node n. The MaxiMax rule defines the value $V(n)$ is node n as follows:

$$V(n) = \gamma \times \max_{i \in \{1,\ldots,k\}} V(n_i)$$

Note $\gamma \in [0, 1]$ is a *discount factor* making an outcome less desirable than the same one at least one step closer from the current step. It enables the agent to choose the shortest sequence amongst several with identical outcomes. We only define the rules in the chance nodes as for this work we always maximize in the others. The associated profile of the persuadee is a probability distribution concentrated on the decision yielding the highest outcome.

Example 3 *If a chance node n has three children n_1, n_2 and n_3 with values $v_1 = 1, v_2 = 2, v_3 = 3$. Therefore, the profile is a distribution* $(0, 0, 1)$.

Pessimistic selection (MaxiMin) (Wald 1950) The strict opposite of the optimistic behaviour is the pessimistic one. This time, the persuader anticipates the opponent will always try to minimize the outcome. It can be related to a two-players zero-sum game where a negative outcome for one player is positive for the other. A pessimistic behaviour is suitable when performing better than a minimum threshold is critical. The MaxiMin rule is defined as follows:

$$V(n) = \gamma \times \min_{i \in \{1,\ldots,k\}} V(n_i)$$

The profile is a distribution concentrated on the decision yielding the lowest reward.

Example 4 *(Example 3 cont'd) The profile associated to children n_1, n_2 and n_3 is* $(1, 0, 0)$.

Hurwicz α-criterion (Hurwicz 1952) Hurwicz α-criterion is a generalization of both MaxiMax and MaxiMin. It defines a parameter, α, acting as an optimism factor. This rule allows us to define a balance between optimism and pessimism to have a more nuanced solution that is less sensitive to the error. This criterion is defined as follows:

$$V(n) = \gamma \times (\alpha \times \max_{i\in\{1,\dots,k\}} V(n_i) + (1-\alpha) \times \min_{i\in\{1,\dots,k\}} V(n_i))$$

Note that an $\alpha = 1$ (resp. 0) is equivalent to the MaxiMax (resp. MaxiMin). The profile is a distribution with a probability of α for the best decision (from the persuader point of view), $(1-\alpha)$ for the worst and 0 for all the others.

Example 5 *(Example 3 cont'd) The profile for an* $\alpha = 0.8$ *is* $(0.2, 0, 0.8)$.

Laplace insufficient reason criterion The assumption made by this rule is that having no information amounts to having an equal chance for all the outcomes. This rule is a maximum expected utility with a uniform probability distribution and is defined as follows:

$$V(n) = \gamma \times \sum_{i\in\{1,\dots,k\}} \left(\frac{1}{k} \times V(n_i)\right)$$

The profile is a uniform distribution over all the decisions that can be made in the chance node considered.

Example 6 *(Example 3 cont'd) The profile is* $(\frac{1}{3}, \frac{1}{3}, \frac{1}{3})$.

Choosing a decision rule

The choice of a decision rule is driven by the desired behaviour for the persuader, the behaviour assumed for the persuadee but also by the amount of risk we are willing to take.

For instance, in an informal discussion where we do not fear to lose the debate, the MaxiMax rule can be used to increase the impact of our point if we believe the persuadee to be credulous. However, in a more serious situation, for instance a police negotiator trying to make a hostage-taker surrender, being optimistic can be dramatic if the worse situation is not addressed effectively.

Optimizing the tree

The optimization of a decision tree amounts to the computation of the optimal decision to take in each decision node, *i.e.*, the optimal policy. It is done by applying the decision rule by backward induction, starting from the leaves up to the root. For instance, in the case of MaxiMin, each leaf returns its value, each chance node returns the value of its lowest yielding child and each decision node the value of its highest yielding child. Moreover, each decision node keeps track of which child yields the highest outcome.

Example 7 *Let us define the outcomes of the decision tree depicted in Figure 1 as given in Table 1a. Table 1b shows the resulting optimal decision for each decision rule. As we can see, the optimal decision is different. For instance the MaxiMax decision rule tells us to play argument b. However, while this can lead to the best outcome (3) we also have a chance to obtain the worst one (-10).*

Example 7 shows the need to choose carefully the decision rule to apply.

v_1	v_2	v_3	v_4	v_5	v_6
0.8	0.5	-10	3	0.6	1

(a) Outcome values

method	n'	n''	n'''	action in n_0
MaxiMax	0.8	**3**	1	b
MaxiMin	0.5	-10	**0.6**	g
Hurwicz $\alpha = 0.2$	0.56	0.4	**0.92**	g
Hurwicz $\alpha = 0.5$	0.65	-3.5	**0.8**	g
Hurwicz $\alpha = 0.8$	**0.74**	-7.4	0.68	a
Laplace	0.65	-3.5	**0.8**	g

(b) Optimal action for each decision rule

Table 1: Example of optimal decision for each rule

Experiments

In this section, we present the experiments we conducted in order to study the behaviour of each decision rule with respect to the behaviour of the persuadee and the error in the representation of this behaviour by the persuader.

Graph generation and tree optimization

First, we randomly generate 100 graphs with 8 arguments without cycles. For each graph, 5 different persuasion goals are randomly selected giving a final set of 500 persuasion problems. Note that the low number of arguments is due to the update method. It can be overcome by using structure optimizations such as (Hadoux and Hunter 2016).

We studied the evolution of the performance of each rule when the actual decision model of the persuadee differs from the profile with a divergence interval d. This divergence is computed using the *Jensen-Shannon divergence* (JSD) (Lin 1991), a symmetrical bounded (between 0 and 1) extension of the *Kullback-Leibler divergence*, defined as follows:

$$\text{JSD}(P \parallel Q) = \frac{1}{2}\text{D}(P \parallel M) + \frac{1}{2}\text{D}(Q \parallel M)$$

where P is the profile, Q the actual distribution, $M = \frac{1}{2}(P+Q)$ and D is the Kullback-Leibler divergence. The divergence can be seen as a measure of the error where $d = 0$ means the assumed profile and the actual decision model of the persuadee are identical.

We also compare using a naive random decision rule, picking a uniform random decision in each node, for each player, as a baseline. For each of the 500 graphs, the optimal policy is computed for each decision rule.

Experimental run

One run of experiments for one of the 500 graphs and a lower bound lb for the divergence d is computed as presented in Algorithm 2. Starting from the root, with a profile associated to the chosen decision rule, $\texttt{Run}(profile, lb)$ is called. In the root (if it is not a leaf), the optimal decision is applied (Line 6), leading to a chance node or a leaf. If it is a leaf, the run terminates, returning the value in this leaf (Line 8).

Algorithm 2: Experimental run algorithm

```
   Function Run (profile, divergence)
      return DecNode (root, profile, divergence)

   Function DecNode (node, profile, divergence)
4     if node is a leaf then return node.value
      else
6        return ChnNode (node.optimalChild, profile,
           divergence)

   Function ChnNode (node, profile, divergence)
8     if node is a leaf then return node.value
      else
10       p = random distribution s.t. JSD(p, profile) ∈
           [divergence, divergence + 0.1]
11       n = draw a child of node using p
12       return DecNode (n, profile, divergence)
```

Otherwise, a distribution of probabilities over the children of this node is created at the given divergence from the profile (Line 10). The next node is drawn following this discrete distribution (Lines 11 and 12). If it is a leaf, the run terminates and returns the value in this leaf (Line 4). Otherwise (*i.e.*, a decision node), the process is reiterated (Line 6).

Figures 2a, 2b, 2c and 2d present the results averaged on 1000 runs performed on each of the 500 graphs for horizons of 2, 4, 6, and 8. Each x on the x-axis is the lower bound of an interval of 0.1 of divergence between the profile of the persuadee and the created distribution (Line 13 of Algorithm 2). For instance, $x = 2$ means the divergence is between 0.2 and 0.3 in each chance node. Distributions at a divergence superior to 0.7 could not always be created. The y-axis presents the average outcome of the 1000 runs on each of the 500 graphs.

Interpretation of the results

Interestingly, we can see that some intuitive observations can be found in each figure.

- The naive random decision rule is not affected by the error on the persuadee.
- The MaxiMax rule outperforms the other rules when the error is low.

Intuitively, MaxiMax is the best rule to apply when the error is zero. Indeed, in this case, the persuadee is perfectly represented and she is maximizing as well. Therefore, no policy can yield a better outcome. A big divergence does not mean the persuadee will play the minimum yielding action but rather will have a probability close to zero on playing the maximum yielding one.

- In the same way, we can see that Hurwicz with $\alpha = 0.8$, *i.e.* closer to the MaxiMax than to the MaxiMin, follows the same trend as MaxiMax.

However, while performing possibly slightly worse than MaxiMax when the error is zero, the results are less impacted by the increase in the error. Therefore, this rule may be more suitable than MaxiMax when the model of the persuadee is less sure to be accurate.

- At the other end of the spectrum, MaxiMin and Hurwicz with $\alpha = 0.2$ perform better when the error is larger.

Indeed, MaxiMin and Hurwicz with $\alpha = 0.2$ aim for the highest minimal value. Therefore, with a large error, the persuadee has a smaller probability to play the minimum value and will therefore yield a higher outcome than expected. By including a part of the highest outcome, Hurwicz rule manages to circumvent the inability of MaxiMin to differentiate two decisions with an identical minimum and a different maximum, thus yielding a highest average outcome.

During the optimization phase, Laplace and Hurwicz with $\alpha = 0.5$ rules maximize an average of the values of several children nodes. Laplace rule uniformly averages on all children while Hurwicz uniformly averages on the minimum and the maximum only. Therefore, the results are relatively insensitive to the error. They are between the maximum and the minimum for each interval of divergence. These two rules perform better than the naive random because the averages are made only in chance nodes (where naive random draws uniformly for each type of node). They maximize in decision nodes. Note that the curves are shorter because no distribution could be computed at a divergence greater than 0.4 of those profiles. Indeed, the more uniform the profile is, the lower the maximum divergence to any distribution is.

Increasing the horizon of the dialogue has a double effect. Allowing more plays for the persuader gives her a higher chance to posit the goal argument as well as defenders. However, it also enables the persuadee to posit more arguments to try to defeat the goal argument. Therefore, the effects of the horizon are hard to determine *a priori* and it needs to be tuned for a given persuasion problem.

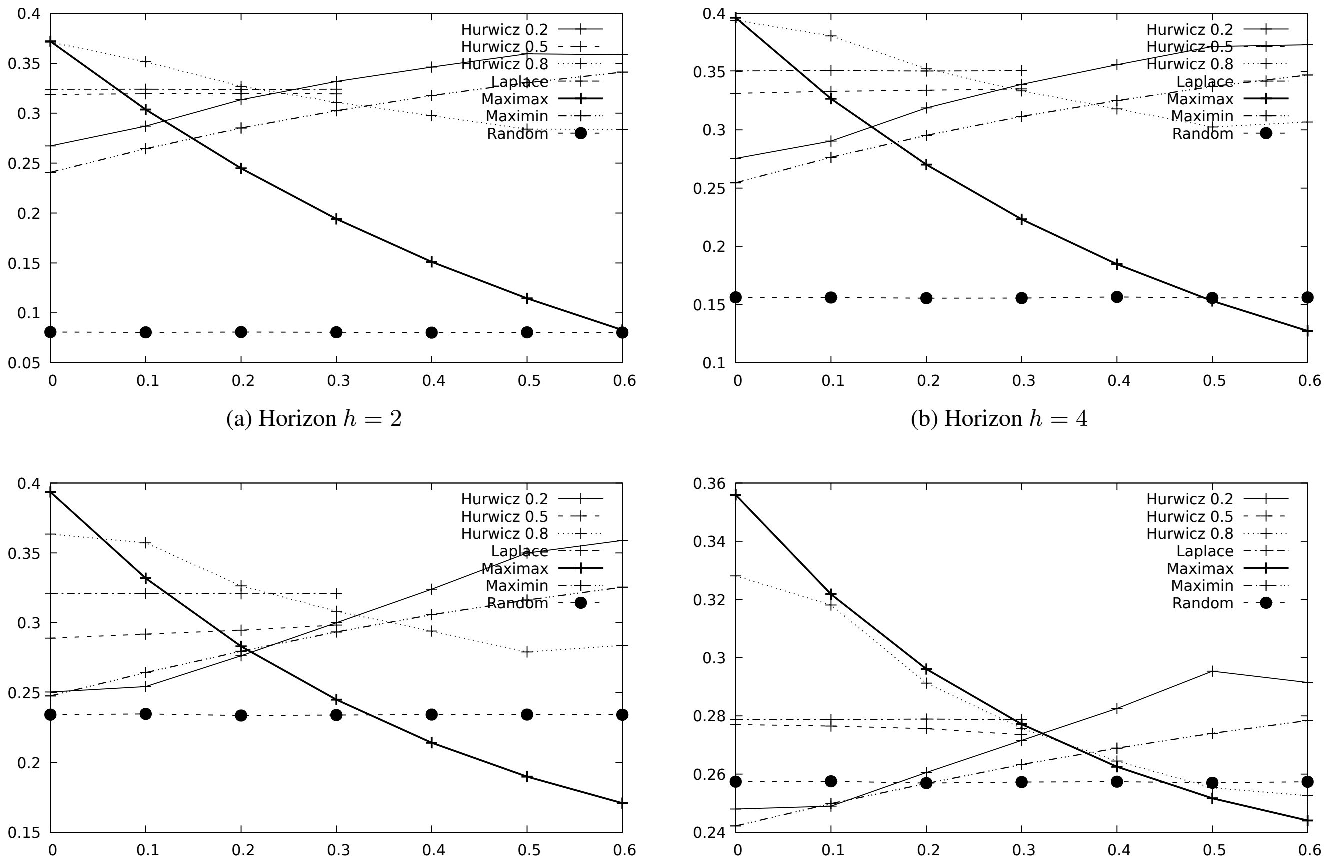
Besides the numerical value of the outcome, the horizon has a role in the amount of risk taken when choosing a particular decision rule.

- With a longer horizon, playing MaxiMax will perform at least as good as the others with a bigger error.

Indeed, as we can see, with a longer horizon, the error needs to be bigger as well for the MaxiMax curves to fall below all the others (except the naive random).

Finally, these observations mean the outcome is dependent of the combination of three factors: the decision rule, the error and the horizon. Empirical assumptions can be derived from these results:

- With optimistic rules (MaxiMax and Hurwicz with an α close to 1), the less accurate the profile is, the bigger the horizon needs to be.
- Optimistic rules perform better with a low error.
- If the error cannot fall under a known threshold, pessimistic rules may perform better (in these experiments, if the error is at least of 0.3).
- If no information is available on the possible error, an averaging rule (Laplace or Hurwicz with $\alpha = 0.5$) will yield a minimum guaranteed outcome. This lower bound can be formally derived from the problem definition.



(a) Horizon $h = 2$ (b) Horizon $h = 4$

(c) Horizon $h = 6$ (d) Horizon $h = 8$

Figure 2: Results for different horizons

Note that MaxiMax is the only rule that does not take into account the minimum value. Therefore, a minimal outcome cannot be guaranteed with this rule. If the error is big, the MaxiMax rule can perform worse than the naive random.

Conclusions

In this paper, we have presented a systematic procedure to transform a persuasion problem into a decision problem and how to apply state-of-the-art methods to solve it. We also theoretically and empirically reviewed and compared the best known decision rules in this context. We derived some intuitions to efficiently choose the decision rule. In order to exploit this framework, one needs to: (1) determine a valuation function suitable to the problem, (2) choose an update function depending on the behaviour of the persuadee and (3) select a decision rule based on the desired behaviour and the amount of risk to consider for the persuader.

Most proposals for dialogical argumentation focus on protocols (*e.g.*, (Prakken 2005), (Prakken 2006), (Fan and Toni 2011), (Caminada and Podlaszewski 2012)) with strategies being under-developed. See (Thimm 2014) for a review of strategies in multi-agent argumentation. Strategies in argumentation have been analyzed using game theory (*e.g.*, (Rahwan and Larson 2008), (Fan and Toni 2012)), but these are more concerned with issues of manipulation, rather than persuasion. There are also some proposals for using probability theory to, for instance, select a move based on what an agent believes the other is aware of (Rienstra, Thimm, and Oren 2013), or, to approximately predict the argument an opponent might put forward based on an history (Hadjinikolis et al. 2013). Neither consider belief in the arguments. Other works represent the problem as a probabilistic finite state machine with a restricted protocol (Hunter 2014b), and generalize it to POMDPs when there is uncertainty on the internal state of the opponent (Hadoux et al. 2015).

More decision rules can be studied in the context of persuasion problems in order to define other behaviours. In future work, we will examine regret methods, such as *MinMax Regret* (Savage 1961). This class of rules efficiently captures the notion of safety with respect to the error and yields policies within a given bound to the optimal one.

Acknowledgments

This research was partly funded by EPSRC grant EP/N008294/1 for the Framework for Computational Persuasion project.

References

Baroni, P.; Giacomin, M.; and Vicig, P. 2014. On rationality conditions for epistemic probabilities in abstract argumentation. In *Proceedings of the 5th International Conference on Computational Models of Argument (COMMA)*, 121–132.

Caminada, M., and Podlaszewski, M. 2012. Grounded semantics as persuasion dialogue. In *Proceedings of the 4th International Conference on Computational Models of Argument (COMMA)*, 478–485.

Dung, P. M. 1995. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming, and n-person games. *Artificial Intelligence* 77:321–357.

Fan, X., and Toni, F. 2011. Assumption-based argumentation dialogues. In *Proceedings of the 22nd International Joint Conference on Artificial Intelligence (IJCAI)*, 198–203.

Fan, X., and Toni, F. 2012. Mechanism design for argumentation-based persuasion. In *Proceedings of the 4th International Conference on Computational Models of Argument (COMMA)*, 322–333.

Hadjinikolis, C.; Siantos, Y.; Modgil, S.; Black, E.; and McBurney, P. 2013. Opponent modelling in persuasion dialogues. In *Proceedings of the 23rd International Joint Conference on Artificial Intelligence (IJCAI)*, 164–170.

Hadoux, E., and Hunter, A. 2016. Computationally viable handling of beliefs in arguments for persuasion. In *Proceedings of the 28th International Conference on Tools with Artificial Intelligence (ICTAI)*, In press.

Hadoux, E.; Beynier, A.; Maudet, N.; Weng, P.; and Hunter, A. 2015. Optimization of probabilistic argumentation with Markov decision models. In *Proceedings of the 24th International Joint Conference on Artificial Intelligence (IJCAI)*, 2004–2010.

Hunter, A., and Thimm, M. 2014. Probabilistic argumentation with incomplete information. In *Proceedings of the 21st European Conference on Artificial Intelligence (ECAI)*, 1033–1034.

Hunter, A. 2013. A probabilistic approach to modelling uncertain logical arguments. *International Journal of Approximate Reasoning* 54(1):47–81.

Hunter, A. 2014a. Opportunities for argument-centric persuasion in behaviour change. In *Proceeding of the 14th European Conference on Logics in Artificial Intelligence (JELIA)*, volume 8761 of *LNCS*, 48–61. Springer.

Hunter, A. 2014b. Probabilistic strategies in dialogical argumentation. In *Proceedings of the 8th International Conference on Scalable Uncertainty Management (SUM)*, volume 8720 of *LNCS*, 190–202. Springer.

Hunter, A. 2015. Modelling the persuadee in asymmetric argumentation dialogues for persuasion. In *Proceedings of the 24th International Joint Conference on Artificial Intelligence (IJCAI)*, 3055–3061.

Hurwicz, L. 1952. A criterion for decision making under uncertainty. Technical Report 355, Cowles Commission.

Lin, J. 1991. Divergence measures based on the shannon entropy. *IEEE Transactions on Information theory* 37(1):145–151.

Prakken, H. 2005. Coherence and flexibility in dialogue games for argumentation. *Journal of Logic and Computation* 15(6):1009–1040.

Prakken, H. 2006. Formal sytems for persuasion dialogue. *Knowledge Engineering Review* 21(2):163–188.

Rahwan, I., and Larson, K. 2008. Pareto optimality in abstract argumentation. In *Proceedings of the 23rd AAAI Conference on Artificial Intelligence*, 150–155.

Rienstra, T.; Thimm, M.; and Oren, N. 2013. Opponent models with uncertainty for strategic argumentation. In *Proceedings of the 23rd International Joint Conference on Artificial Intelligence (IJCAI)*, 332–338.

Savage, L. J. 1961. *The Foundations of Statistics Reconsidered*. University of California Press.

Thimm, M. 2012. A probabilistic semantics for abstract argumentation. In *Proceedings of the 20th European Conference on Artificial Intelligence (ECAI)*, volume 242 of *Frontiers in Artificial Intelligence and Applications*, 750–755. IOS Press.

Thimm, M. 2014. Strategic argumentation in multi-agent systems. *Kunstliche Intelligenz* 28:159–168.

Wald, A. 1950. Basic ideas of a general theory of statistical decision rules. In *Proceedings of the International congress of Mathematicians*, volume 1, 308–325.

Yager, R. R. 1988. On ordered weighted averaging aggregation operators in multicriteria decision-making. *IEEE Transactions on systems, Man, and Cybernetics* 18(1):183–190.