Probabilistic Argumentation Frameworks

Hengfei Li, Nir Oren, and Timothy J. Norman

Department of Computing Science, University of Aberdeen, Aberdeen, AB24 3UE, Scotland
{h.li,n.oren,t.j.norman}@abdn.ac.uk

Abstract. In this paper, we extend Dung's seminal argument framework to form a probabilistic argument framework by associating probabilities with arguments and defeats. We then compute the likelihood of some set of arguments appearing within an arbitrary argument framework induced from this probabilistic framework. We show that the complexity of computing this likelihood precisely is exponential in the number of arguments and defeats, and thus describe an approximate approach to computing these likelihoods based on Monte-Carlo simulation. Evaluating the latter approach against the exact approach shows significant computational savings. Our probabilistic argument framework is applicable to a number of real world problems; we show its utility by applying it to the problem of coalition formation.

1 Introduction

Likelihoods and probabilities form a cornerstone of reasoning in complex domains. When argumentation is used as a form of defeasible reasoning, uncertainty can affect the decisions reached during the reasoning process [27]. Uncertainty can also affect applications of argumentation technologies in other ways. For example, in the context of a dialogue, uncertainty regarding the knowledge of participants can affect both the dialogue outcome, and the utterances the participants choose to make. Furthermore, if uncertainty is viewed as a proxy for argument strength, questions immediately arise regarding argument interaction and the strength of conclusions given an argument system.

In this paper we examine the role of probabilities in an abstract argument framework. Within such a framework, an argumentation semantics defines a method by which a set of justified arguments can be deduced. As a reasoning approach, a semantics takes an argumentation framework as its knowledge base and produces a set of justified arguments as its output. The problem we address thus involves identifying the effects of probabilities on argument justification.

At the intuitive level, our approach is relatively simple. Starting with Dung's abstract argumentation framework[9] as its base[1], we assign probabilities to arguments and defeats. These probabilities represent the likelihood of existence of a specific argument or defeat, and thus capture the uncertainties inherent in the

[1] Though as discussed in Section 6, our techniques are applicable to nearly any other argumentation framework.

S. Modgil, N. Oren, and F. Toni (Eds.): TAFA 2011, LNAI 7132, pp. 1–16, 2012.

argument system. Within such a *probabilistic argument framework* (abbreviated PrAF), all possible arguments neither definitely exist, nor completely disappear. Instead, all elements of the framework have a different chance of existing. The semantics of such a framework then identify the likelihood of different sets of arguments being justified according to different types of extensions.

Now, since we are interested in the likelihood of a set of arguments being justified we are, in a sense, reversing the standard semantics of argumentation. Rather than identifying *which* arguments are in some sense compatible, we are instead identifying a set of arguments and asking what their likelihood of being compatible is (with respect to the other arguments, defeats and probabilities which make up the framework). Answering this type of question has a number of real world applications, including to the domains of trust and reputation [32] and coalition formation [28].

As we show, a naïve approach to computing the likelihood of some set of arguments being justified within a probabilistic argumentation framework based on the standard laws of probability has exponential computational complexity with respect to the number of arguments even in situations where the underlying semantics has linear complexity. Given that this is impractical for most real-life scenarios we propose, and evaluate, an approximation method based on the idea of Monte-Carlo simulation for calculating the likelihood of a set of arguments being justified.

The remainder of this paper is structured as follows. In the next section, we describe and formally define probabilistic argumentation frameworks, and explain the naïve method for performing computations over such PrAFs. Section 3 then details the Monte-Carlo simulation based approximation method. In Section 4, we empirically evaluate the performance of both of our techniques. An illustrative application for which PrAFs are particularly applicable is detailed in Section 5, following which Section 6 provides a more general discussion together with suggestions for future work. We then summarise our results and conclude the paper in Section 7.

2 Probabilistic Argumentation Frameworks

In this section, we extend Dung's argumentation framework to include uncertainty with respect to arguments and defeats. Essentially, we assign a probability to all elements of the argument framework, namely to every argument and element of the defeat relation. It should be noted that our approach can be easily extended to other frameworks such as the bipolar [7], evidential [20] and value based argumentation frameworks [5] as probabilities can be also assigned to the additional elements of these frameworks (e.g. to the members of the support relation in the case of bipolar frameworks). We begin this section by briefly describing Dung's system, following which we discuss our extensions and methods for reasoning about probabilistic frameworks.

Definition 1. ***(Dung Argumentation Framework)*** *A Dung argumentation framework DAF is a pair (Arg, Def) where Arg is a set of arguments, and $Def \subseteq Arg \times Arg$ is a defeats relation.*

A set of arguments S is conflict-free *if $\nexists a, b \in S$ such that $(a, b) \in Def$. An argument a is* acceptable *with respect to a set of arguments S iff $\forall b \in Arg$ such that $(b, a) \in Def$, $\exists c \in Arg$ such that $(c, b) \in Def$. A set of arguments S is* admissible *iff it is conflict free and all its arguments are acceptable with respect to S.*

From these definitions, different semantics have been defined [4]. These semantics identify sets of arguments which are, in some intuitive sense, compatible with each other. For example, the grounded semantics yield a single extension which is the least fixed point of the characteristic function $F_{AF}(S) = \{a | a \in Arg \text{ is acceptable w.r.t } S\}$. In the remainder of this paper, we will concentrate on the grounded semantics due to its computational tractability [11].

2.1 Formalising Probabilistic Argumentation Frameworks

A probabilistic argumentation framework extends Dung's argument framework by associating a likelihood with each argument and defeat in the original system. Intuitively, a PrAF represents an entire set of DAFs that exist *in potentia*. A specific DAF can then has a certain likelihood of being *induced* from the PrAF.

Definition 2. ***(Probabilistic Argumentation Framework)*** *A Probabilistic Argumentation framework PrAF is a tuple (A, P_A, D, P_D) where (A, D) is a DAF, $P_A : A \to (0:1]$ and $P_D : D \to (0:1]$.*

The functions P_A and P_D map individual arguments, and defeats to likelihood values. These represent the likelihood of existence of an argument within an arbitrary DAF induced from the PrAF. As discussed below, P_D is, implicitly, a conditional probability. It should be noted that the lower bound of these probabilities is not 0 (but approaches it in the limit). This requirement exists because any argument or defeat with a likelihood of 0 cannot ever appear within a DAF induced from the PrAF, and is thus redundant.

A PrAF represents the set of all DAFs that can potentially be created from it. We call this creation process the inducement of a DAF from the PrAF. All arguments and defeats with a likelihood of 1 will be found in the induced DAF, which can then contain additional arguments and defeats, as specified by the following definition.

Definition 3. ***(Inducing a DAF from a PrAF)*** *A Dung argument framework $AF = (Arg, Def)$ is said to be induced from a probabilistic argumentation framework $PrAF = (A, P_A, D, P_D)$ iff all of the following hold:*

- $Arg \subseteq A$
- $Def \subseteq D \cap (Arg \times Arg)$
- $\forall a \in A$ *such that* $P_A(a) = 1$, $a \in Arg$
- $\forall (f, t) \in D$ *such that* $P_D((f, t)) = 1$ *and* $P_A(f) = P_A(t) = 1$, $(f, t) \in Def$

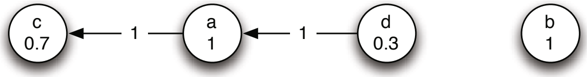

Fig. 1. A graphical depiction of a PrAF

We write $I(PrAF)$ to represent the set of all DAFs that can be induced from $PrAF$.

A DAF induced from a PrAF thus contains a subset of the arguments found in the PrAF, together with a subset of the defeats found in the PrAF, subject to these defeats containing only arguments found within the induced DAF. The process of inducing a DAF eliminates information regarding likelihoods found in the original PrAF.

Now, consider a situation where a number of entities are participating in a dialogue, and one of them (labelled α) would like to compute what conclusions might be drawn at the end of this interaction. Let us assume that α has arguments a and b in its knowledge base, and it believes that the other dialogue participants have arguments c and d in their knowledge base. This belief is however uncertain; c is believed to be known by the others with a likelihood of 0.7, and d with a likelihood of 0.3. Now let us assume that argument a defeats c and d defeats a. For simplicity, we assume that these defeat relations have no uncertainty associated with them (i.e. $P_D = 1$ for each of them). Formally, this can be represented by the PrAF following PrAF, illustrated in Figure 1.

$$(\{a, b, c, d\}, \{(a, 1), (b, 1), (c, 0.7), (d, 0.3)\}, \{(a, c), (d, a)\}, \{((a, c), 1), ((d, a), 1)\})$$

Given this PrAF, we can induce the following DAFs:

$$\begin{array}{ll} (\{a, b\}, \{\}), & (\{a, b, c\}, \{(a, c)\}), \\ (\{a, b, d\}, \{(d, a)\}), & (\{a, b, c, d\}, \{(a, c), (d, a)\}) \end{array}$$

Clearly, b appears in the grounded extension of all of these DAFs, while a appears in the grounded extension of 3 out of 4 induced DAFs. Now, α might want to identify the likelihood of a being justified (i.e. in the grounded extension) at the end of the dialogue, perhaps to decide whether to advance it or not (assuming that advancing an argument has some associated utility cost [21]).

2.2 Probabilistic Justification

Our goal is to compute the likelihood that some set of arguments exists and is justified according to some semantics within the DAFs induced from a PrAF. This likelihood can be obtained from the basic laws of probability, and we detail this procedure next. We make one critical simplifying assumption, namely that the likelihood of one argument (defeat) appearing in an induced DAF is independent of the likelihood of some other argument (defeat) appearing. With this assumption in hand, we begin by computing the likelihood of some DAF being induced from the PrAF.

As mentioned earlier, the P_D relation associates a *conditional* probability with each possible defeat. That is, for some arguments a, b

$$P_D(a,b) = P((a,b) \in Def | a,b \in Arg) \text{ for the induced DAF } (Arg, Def)$$

Informally, the probability of some DAF AF being induced from a PrAF can be computed via the joint probabilities of the arguments and defeat relations appearing in AF. In order to formalise this concept compactly, we must identify the set of defeats that *may* appear in an induced DAF. We label this set as *DefA*. Given a DAF with arguments $Args$, and a $PrAF$ containing defeats D

$$DefA = \{(a,b) | a,b \in Args \text{ and } (a,b) \in D\}$$

This allows us to compute the probability of some DAF AF being induced from a PrAF, written $P^I_{PrAF}(AF)$, by computing the joint probabilities of independent variables as follows:

$$P^I_{PrAF}(AF) = \prod_{a \in Arg} P_A(a) \prod_{a \in A \setminus Arg} (1 - P_A(a)) \prod_{d \in Def} P_D(d) \prod_{d \in DefA \setminus Def} (1 - P_D(d)) \quad (1)$$

Applying this to our earlier example, $P^I_{PrAF}((\{a,b\},\{\})) = 0.21$.

Proposition 1. *The sum of probabilities of all DAFs that can be induced from an arbitrary PrAF is 1. That is,* $\sum_{a \in I(PrAF)} P^I_{PrAF}(a) = 1$.

Now our goal is to identify the likelihood of some set of arguments being consistent with respect to some set of argumentation semantics. Such a semantics may return one or many extensions for a given argument framework, and we formalise our notion of consistency through the definition of a *semantic evaluation function*, $\xi^S(AF, X)$ which returns *true* if and only if the set of arguments X is deemed consistent using the semantics S when evaluated over the argument framework AF. Thus, for example $\xi^G(AF, X)$ could return true if the set of arguments X appears as a subset of the grounded extension of AF.

Then, following on from Proposition 1, given some $PrAF$, the likelihood of X being consistent according to the semantics S is defined as follows:

$$P_{PrAF}(X) = \sum_{AF \in I(PrAF)} P^I_{PrAF}(a) \text{ where } \xi^S(AF, X) = true \quad (2)$$

Referring again to our earlier example, $P_{PrAF}(\{a,b\}) = 0.7$.

While we can utilise Equations 1 and 2 to compute the exact likelihood of a set of arguments being justified with regards to some semantics, the size of the set of possible DAFs which can be induced from a PrAF grows exponentially with regards to the number of arguments and defeats within the PrAF, resulting in exponential time complexity (not including the computational costs associated with computing the results of ξ^S). This is clearly impractical for a large set of arguments, and in the next section, we examine an approximate method for determining these likelihoods.

3 Approximate Solutions in Probabilistic Argumentation Frameworks

In this section we describe a Monte-Carlo simulation based approach to computing $P_{PrAF}(X)$ for an arbitrary set of arguments X. At an abstract level, a Monte-Carlo simulation operates by repeatedly sampling a distribution many times in order to approximate it. More specifically, such a simulation has three basic steps. First, given a possible set of inputs, a subset of these inputs is selected according to some probability distribution. Second, some computation is performed using the selected inputs. Finally, the results of repeating the first two steps multiple times is aggregated. Monte-Carlo simulation has a long history, and has been applied to a variety of computationally difficult problems including inference in Bayesian Networks [19], reinforcement learning [31] and computer game playing [8].

In this context of probabilistic argumentation frameworks, this process involves randomly inducing DAFs from a PrAF, with the likelihood of an arbitrary DAF being induced being dependant on the underlying probability distribution of its individual members. We thus sample the space of possible DAFs in a way that approximates the DAFs true distribution in the probability space.

The only source of uncertainty in Equation 2 lies in P^I_{PrAF} which in turn depends only on the probabilities found in the underlying PrAF. Therefore, in order to approximate $P_{PrAF}(X)$ we need only sample the space of arguments and defeats found in the PrAF. Algorithm 1 describes this process more precisely.

The algorithm samples N DAFs from the set of inducible DAFs. A single DAF is generated by randomly selecting arguments and defeats according to their likelihood of appearance (Lines 4-7 and 10-14 respectively). This resultant DAF is then evaluated for the presence of X through the ξ^S function (Line 16), and if this function holds, the DAF is counted. $P_{PrAF}(X)$ is finally approximated as the ratio of the total number of DAFs in which $\xi^S(X)$ holds to the number DAFs sampled (Line 20).

The following proposition states that as our number of trials increases, the error in our approximation of $P_{PrAF}(X)$ shrinks.

Proposition 2. *If we denote the output of Algorithm 1 as $P'_{PrAF}(X)$, then as $N \to \infty$, $P_{PrAF}(X) - P'_{PrAF}(X) \to 0$. More specifically, there is some $N \in \mathbb{Z}^+$ and $\epsilon \in \mathbb{R}^+$ such that for all $M > N$, if M trials are run, $|P_{PrAF}(X) - P'_{PrAF}(X)| < \epsilon$.*

This proposition means that our algorithm has an anytime property: it may be terminated at any time, and earlier terminations will still provide an approximation to the true probability, albeit with a greater error than would be provided from a later termination.

While this proposition provides some guarantees regarding the accuracy of our results given enough trials, it does not answer one critical question: how many trials must be run to ensure (with some level of confidence) that our approximation has only a small level of error?

Algorithm 1. An algorithm to approximate $P_{PrAF}(X)$

Require: A Probabilistic Argumentation Framework $PrAF = (A, P_A, D, P_D)$
Require: A set of arguments $X \subseteq A$
Require: A number of trials $N \in \mathbb{N}$
Require: A semantic evaluation function, ξ^S

$Count = 0$
for $I = 0$ to N **do**
  $Arg = Def = \{\}$
  for all $a \in A$ **do**
    Generate a random number r such that $r \in [0, 1]$
    if $P_A(a) \geq r$ **then**
      $Arg = Arg \cup \{a\}$
    end if
  end for
  for all $(f, t) \in D$ such that $f, t \in Arg$ **do**
    Generate a random number r such that $r \in [0, 1]$
    if $P_D((f, t)) \geq r$ **then**
      $Def = Def \cup \{(f, t)\}$
    end if
  end for
  if $\xi^S((Arg, Def), X) = true$ **then**
    $Count = Count + 1$
  end if
end for
return $Count/N$

In order to answer this question, we note that the results of a Monte-Carlo simulation can be viewed as a normal distribution over possible values for $P_{PrAF}(X)$, and with $P'_{PrAF}(X)$ as its mean. Given this, we may make use of the notion of a *confidence interval* in order to answer our question. In statistics, a confidence level of l for a given a confidence interval CI and a mean p' can be read as stating that the true mean lies within $p' \pm CI$ with a likelihood of l. Such a confidence interval is dependant on the observed likelihood of an event and the number of trials used to make the observations. We can thus recast our problem to ask how many trials need to be run in order to ensure that the confidence interval around $P'_{PrAF}(X)$ (i.e. its error) is smaller than some value ϵ with some specific confidence level (e.g. 95%).

Probably the most common approach to computing such an interval is the normal approximation interval [18], which is defined as follows:

$$p' \pm z_{1-(\alpha/2)} \sqrt{\frac{p'(1-p')}{n}} \tag{3}$$

Here, p' is the observed mean, n is the number of trials, and $z_{1-(\alpha/2)}$ the $1-(\alpha/2)$ percentile of the normal distribution. In the experiments described in Section 4, we required a 95% confidence level, resulting in $z_{1-(\alpha/2)} = 1.96$. Then we get

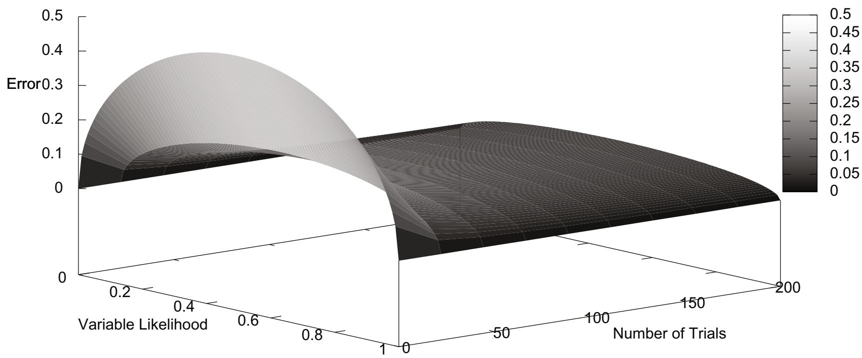

Fig. 2. The relationship between likelihood of a variable, the number of observations made and the error in the observed likelihood

the following equation to compute the number of trials required to achieve an error level below ϵ:

$$N > \frac{p'(1-p')}{\epsilon^2}(1.96)^2 \quad (4)$$

However, this approximation is problematic in our situation as p' is either 0 or 1 after a single trail, which will break down the calculation. To overcome this problem, we utilise the Agresti-Coull interval [1] instead. The general form of this interval is the same as that of Equation 3. However, the values of n and p' are computed differently:

$$n = N + z^2_{1-\alpha/2} \qquad p' = \frac{X + z^2_{1-\alpha/2}/2}{n}$$

Here, N is the number of trials and X is the number of "successes" observed. Intuitively, it adds two different trivial numbers to the number of trials and the number of "successes" respectively, which ensures p' will never be 0. Approximating $1 - \alpha/2$ with the value 2 yields the above new equation. This method guarantees the real "anytime" feature of our algorithm.

$$N > \frac{4p'(1-p')}{\epsilon^2} - 4 \quad (5)$$

Figure 2 provides a plot of this function. As seen here, initially, as the number of trials increase, the error falls off rapidly. However, this shrinking of the error quickly ceases, and additional trials serve to reduce the error by only a small amount. It should also be noted that the likelihoods of variables with extreme values (i.e. near 0 or 1) can be approximated far more quickly than variables with values near 0.5.

Given a desired error level ϵ and confidence level, Equation 5 provides us with a new stopping condition for Algorithm 1. The `for` loop of Line 2 can be substituted for a `while` loop which computes whether the expected error level falls below ϵ given the number of iterations that have been run so far. If this is the case, the loop can end, and the algorithm will terminate.

4 Evaluation

We have described, given some PrAF, two approaches to computing the likelihood of a chosen set of arguments being justified with respect to some semantics. While it is clear that the exact approach is exponential in complexity, it is useful to identify the approximate number of arguments in a PrAF at which point this becomes impractical. Similarly, in order to use it in real world settings, the approximate running time of the Monte-Carlo based approach must also be evaluated.

We implemented both of the approaches described in the paper using SWI-Prolog[2]. For simplicity, we associated likelihood values only with arguments within the PrAF; all defeats had a likelihood of 1. The goal of our first experiment was to identify the effects of differently sized PrAFs on the runtimes of the exact approach, and of the Monte-Carlo based approach with different error tolerances ($\epsilon = 0.01$ and $\epsilon = 0.005$). In order to do so, we evaluated the approaches on identical PrAFs with each PrAF containing between 1 and 16 arguments. Our semantic evaluation function $\xi^S(X)$ computed whether X formed a subset of the grounded extension. We ran our experiment 20 times for each unique number of arguments, and Figure 3 shows our results. As expected, the time taken by the exact approach increases exponentially; the Monte-Carlo based approaches overtake the exact approach at around 13 (when $\epsilon = 0.01$) and 15 (when $\epsilon = 0.005$) arguments. The introduction of uncertainty into the defeats relation would increase the number of DAFs that can be induced from the PrAF meaning that our results, in a sense, represent the best case for the exact approach.

In order to more closely examine the effect of ϵ and the size of the PrAF on the performance of our approximate algorithm, Figure 4 compares the average number of iterations, and runtime, required to achieve the desired level of accuracy against the number of arguments found in the PrAF. As expected, an increase in the size of the PrAF has only a linear effect on the runtime of our algorithm. This increase occurs due to an increase in the time required to computing the membership of grounded extension (as computing this has linear complexity) rather than additional iterations. In other words, the complexity level of our algorithm depends on the complexity of computing membership under some semantics. This result can clearly be seen from Figure 2; the number of iterations required to obtain a certain error level do not depend on the number of arguments and defeats in the PrAF, but only on the joint probabilities obtained from the PrAF. Figure 2 also predicts another result clearly seen in Figure 4, namely that as the permitted error shrinks, the standard deviation of the number of iterations that must be executed grows. This is because the number of iterations required to obtain an error ϵ when the joint probability in

[2] http://www.swi-prolog.org

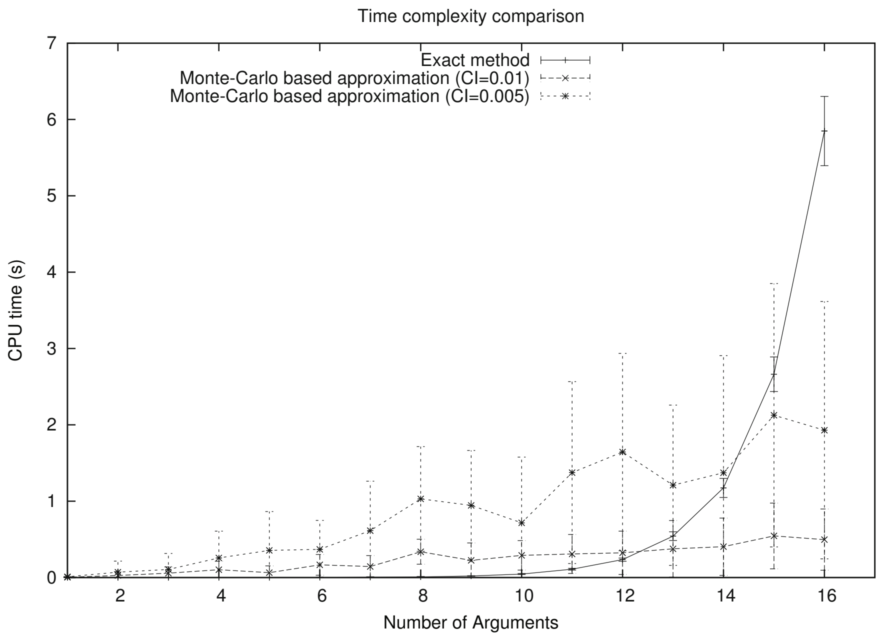

Fig. 3. Comparison of runtimes between the exact and Monte-Carlo based approaches. Error bars indicate 1 standard deviation.

question is close to 0 or 1 grows much more slowly than when the probability is close to 0.5.

Finally, it can also be seen that there exists some variability between the number of iterations required and the time to execute these iterations; this arises due to the underlying Prolog implementation, and the number of iterations is thus a better indicator of algorithm performance.

5 Applying PrAFs to Coalition Formation

In this section, we describe an application of our approach to a real world problem, namely coalition formation. According to [28], "Coalition formation is a fundamental form of interaction that allows the creation of coherent groupings of distinct, autonomous, agents in order to efficiently achieve their individual or collective goals". Coalition formation is applicable to both virtual domains such as e-commerce (where virtual organisations can form in order to satisfy a customer's requirements [24]), and physical domains where, for example, a search and rescue team must be composed of agents with specific capabilities in order to be able to undertake some mission [26].

Most approaches to coalition formation treat the problem as one of utility maximisation; agents will join a coalition if being in the coalition will yield a greater utility than not. Here, we show how to address the problem of coalition formation from a very different perspective. This different perspective allows us

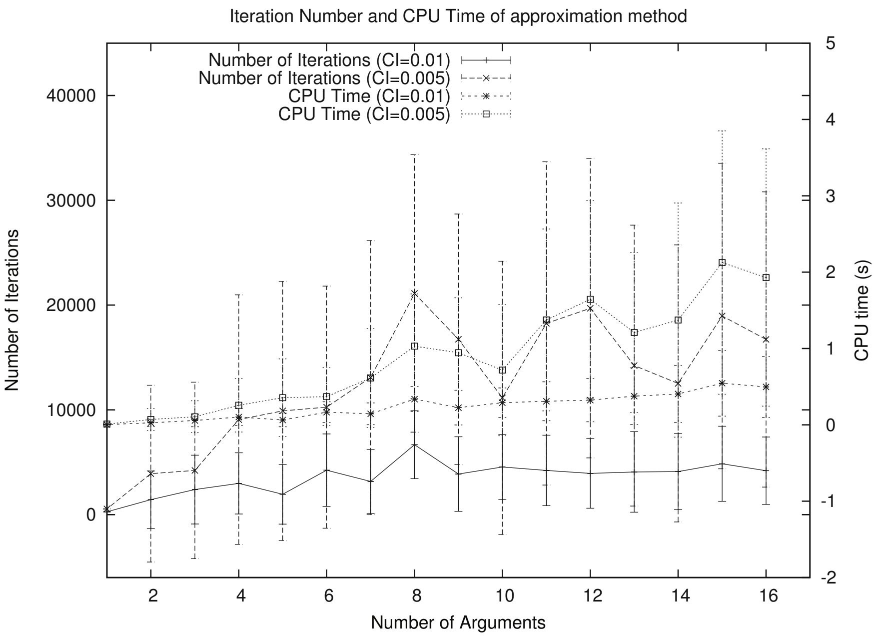

Fig. 4. Comparison of runtimes and number of iterations between the Monte-Carlo based approaches with different ϵ values. Error bars indicate 1 standard deviation.

to explore an aspect of the social dimensions involved in coalition formation; i.e. the notion of whether or not an individual's presence in a coalition may influence another's membership. More specifically, we model a system containing agents with different capabilities, each of which has a prior probability of joining the coalition, and a probability of preventing other potential coalition members from joining the coalition. We would then like to determine what the probability of a coalition forming which is capable of achieving some task.

Translating this problem into a PrAF is trivial. Each agent can be represented as an argument within the PrAF, an associated P_A equal to its prior probability of joining the coalition. Defeats then represent the likelihood of the presence of one member in the coalition preventing another member from joining. Computing the likelihood of a coalition containing specific members can then be computed by computing P_{PrAF}.

As an illustrative example[3], consider a small mercenary team consisting of a leader h, a pilot m, a mechanic b and an expert in persuasion f. Now assume that the presence of the pilot cannot be tolerated by the mechanic, and that f is generally disliked by other team members (to varying degrees); f's presence in a coalition will increase the risk that others will not join. Finally, assume that both f and h are often busy, and occasionally cannot join the team. This situation can be represented by the PrAF shown in Figure 5.

[3] This example is based on the characters from a 1980's television series.

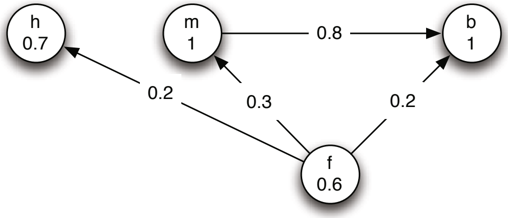

Fig. 5. A PrAF representing the coalition formation example

A system user could use the techniques presented in this paper to compute the likelihood of a specific team being formed, for example consisting of h, m and b (this would be 0.037632+0.056=0.093632. The first value is the likelihood of the full team forming, and the second, the probability of the team forming without f). Given this likelihood, the user might decide to change their goals, or add new agents to the system to increase the chances of success.

The discussion thus far has concentrated on determining whether a coalition can be formed containing some specific set of agents. However, in the context of coalition formation, the goal is often to form a coalition consisting of agents taking on some set of specific roles (e.g. a coalition requires two mechanics and a pilot). One approach to determining the likelihood of forming such a coalition involves identifying all possible ways in which such a coalition can form, and combining the probabilities of each individual coalition to obtain an aggregate probability. However, this approach does not scale well as the size of the system increases. We intend to investigate techniques for dealing with this issue in future work, and discuss it further in the next section.

6 Discussion and Future Work

The use of likelihood in different facets of argumentation for modelling strength or uncertainty of arguments has a long and rich history. Most commonly, such likelihood measures have served as a proxy for argument strength [27,14], or causal strength between arguments [23]. Argument framework incorporating uncertainty about arguments have utilised probabilities to compute the likelihood of some conclusion holding using a variety of different methods. For example, [17,16] consider probability in the construction of argument, deriving a probability of argument from the probabilities of its premises by several different methods. In the context of abstract argumentation frameworks, some approaches for modelling uncertainties such as assigning a numerical values [10] or preference ordering [2] to attacks have been developed. Another approach to strengths of argument involves counting the number of subsets which meet the requirements of some (multiple status) semantics [3], and in which the argument under question appears. The ratio of this number to the total number of extensions then serves to act as a measure of strength for the argument. Our approach is similar in spirit to this latter work as we compute the likelihood of some subset of arguments appearing. However, the introduction of probabilities, through the definition of a PrAF, makes the approach applicable to both single and multiple status semantics, with the distinct advantage of the former's tractability. Another similar work is by Janssen et al [15] in which they do not use probability but also a value between 0 and 1 to model the strength of arguments and attack relations. They borrow the idea from fuzzy logic and these values capture the concept of *degree*; they define their semantics based on fuzzy logic operations.

Within the machine learning and knowledge representation communities, Bayesian Networks [25] form a popular approach to modelling and reasoning about uncertainty. Such networks allow one to reason about the posterior probability of the value of variables within the network given observations about some other parts of the network. The idea of representing a PrAF as a Bayesian Network is intuitively appealing, but not trivial. The DAF induced from a PrAF captures the probability that an argument exists, while the PrAF based semantics computes the likelihood that an argument (or set of arguments) is present in an extension. It is not clear how both of these values can be encoded in, or computed from, a single Bayesian Network, and as future work, we intend to investigate whether, and how, a Bayesian Network representing a PrAF can be constructed.

In Section 5, we discussed one possible application for PrAFs, namely answering questions about the likelihood of a coalition with certain characteristics being formed. We discussed one shortcoming, namely the inability of the basic approach to deal with the notion of roles in coalition formation, and suggested one method for overcoming this shortcoming. Another more nuanced approach involves the use of resource bounded argumentation frameworks [30], which would allow us to place requirements on team composition via constraints, and thus also allow for more nuanced team formation. Another shortcoming involves our underlying Dung based model wherein only defeats between arguments are modelled. Constructing a PrAF on top of a bipolar framework (e.g. [7,20]) would allow us to cater for situations where one agent is more likely to enter into a coalition if some other agent will be present. Another way of achieving this would be to lift the independence assumption regarding the likelihood of argument and defeat relation likelihoods, and all of these form enticing possibilities for future work.

PrAFs and the techniques described in this paper can be applied to other argument frameworks and domains. For example, a value based argumentation framework (VAF) [5] provides a model of determining whether some set of arguments will be accepted by audiences containing agents with different preferences over the defeat relation. Constructing a PrAF on top of such a VAF can allow us to answer questions such as "what is the likelihood of all members in the audience accepting this argument". Clear applications of this include opponent modelling [21] and heuristics for argument [22,29,12]. Another interesting possibility lies in associating a probability distribution with the preferences of the audience within the VAF.

Apart from the coalition formation and argument strategy domains, the ideas associated with constructing and evaluating PrAFs can also play a role in other domains where the notion of the strength of an argument is relevant. For example, in the area of trust and reputation [32], PrAFs can be used to associate reputation information with individual agents. Distrust relationships (following [13]) or biases in trust relationships (following [6]) can then be constructed through the defeats relation , and, by using a bipolar framework, trust relationships can be created through support links. The resultant PrAF can then be used to compute the likelihood of some set of agents considering one another trustworthy.

7 Conclusions

In this paper we introduced *probabilistic argumentation frameworks*. These frameworks add the notion of likelihood to all elements of an abstract argument framework (in this paper, we concentrated on Dung argument frameworks, and thus associated likelihoods with arguments and defeats), and are used to determine the likelihood of some subset of arguments appearing within an extension. The exact method for determining this likelihood has exponential complexity, and is thus impractical for use with anything other than a small argumentation system. To overcome this limitation, we introduced a Monte-Carlo simulation based approach to approximate the likelihood. This latter technique scales up well, providing good results in a reasonable period of time, and has anytime properties, making it ideal for use in almost all situations.

PrAFs have applications to a myriad of domains. In this paper, we focused on one such domain, namely coalition formation, and described how PrAFs can be used to assist a system designer. While we have touched on the applications of PrAFs to other domains, and suggested a number of extensions to their basic representation, we intend to further explore their potential applicability to additional argumentation frameworks and application domains.

References

1. Agresti, A., Coull, B.A.: Approximate is better than “exact” for interval estimation of binomial proportions. The American Statistician 52(2), 119–126 (1998)
2. Amgoud, L., Cayrol, C.: Inferring from inconsistency in preference-based argumentation frameworks. Journal of Automated Reasoning 29, 125–169 (2002)
3. Baroni, P., Dunne, P.E., Giacomin, M.: On extension counting problems in argumentation frameworks. In: Proceeding of the 2010 Conference on Computational Models of Argument: Proceedings of COMMA 2010, pp. 63–74. IOS Press, Amsterdam (2010)
4. Baroni, P., Giacomin, M.: Semantics of abstract argument systems. In: Simari, G., Rahwan, I. (eds.) Argumentation in Artificial Intelligence, pp. 25–44. Springer, US (2009)
5. Bench-Capon, T.: Value based argumentation frameworks. In: Proceedings of the 9th International Workshop on Nonmonotonic Reasoning, Toulouse, France, pp. 444–453 (2002)

6. Burnett, C., Norman, T.J., Sycara, K.: Stereotypical trust and bias in dynamic multi-agent systems. ACM Transactions on Intelligent Systems and Technology (in press)
7. Cayrol, C., Lagasquie-Schiex, M.C.: Bipolar Abstract Argumentation Systems. In: Rahwan, I., Simari, G. (eds.) Argumentation in Artificial Intelligence, ch. 4, pp. 65–84. Springer, Heidelberg (2009), http://www.springerlink.com
8. Coulom, R.: Efficient Selectivity and Backup Operators in Monte-Carlo Tree Search. In: van den Herik, H.J., Ciancarini, P., Donkers, H.H.L.M(J.) (eds.) CG 2006. LNCS, vol. 4630, pp. 72–83. Springer, Heidelberg (2007)
9. Dung, P.M.: On the acceptability of arguments and its fundamental role in non-monotonic reasoning, logic programming and n-person games. Artificial Intelligence 77(2), 321–357 (1995)
10. Dunne, P.E., Hunter, A., McBurney, P., Parsons, S., Wooldridge, M.: Weighted argument systems: Basic definitions, algorithms, and complexity results. Artificial Intelligence 175(2), 457–486 (2011)
11. Dunne, P.E., Wooldridge, M.: Complexity of abstract argumentation. In: Simari, G., Rahwan, I. (eds.) Argumentation in Artificial Intelligence, pp. 85–104. Springer, US (2009)
12. Emele, C.D., Norman, T.J., Parsons, S.: Argumentation strategies for plan resourcing. In: Proceedings of the Tenth International Conference on Autonomous Agents and Multiagent Systems (2011)
13. Erriquez, E., van der Hoek, W., Wooldridge, M.: An abstract framework for reasoning about trust. In: Proceedings of AAMAS 2011 (to appear, 2011)
14. Gómez Lucero, M.J., Chesñevar, C.I., Simari, G.R.: Modelling Argument Accrual in Possibilistic Defeasible Logic Programming. In: Sossai, C., Chemello, G. (eds.) ECSQARU 2009. LNCS, vol. 5590, pp. 131–143. Springer, Heidelberg (2009)
15. Janssen, J., Cock, M.D., Vermeir, D.: Fuzzy argumentation frameworks. In: Proceedings of IMPU 2008 (12th International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems), pp. 513–520 (2008)
16. Kohlas, J., Haenni, R.: Assumption-based reasoning and probabilistic argumentation systems. Tech. Rep. 96–07, Institute of Informatics, University of Fribourg, Switzerland (1996)
17. Krause, P., Ambler, S., Elvang-Goransson, M., Fox, J.: A logic of argumentation for reasoning under uncertainty. Computational Intelligence 11(1), 113–131 (1995)
18. Lewicki, P., Hill, T.: Statistics: Methods and Applications. StatSoft Inc. (2005)
19. Mitchell, T.M.: Machine Learning. McGraw-Hill Higher Education (1997)
20. Oren, N., Norman, T.J.: Semantics for evidence-based argumentation. In: Computational Models of Argument: Proceedings of COMMA 2008, Toulouse, France, May 28-30, pp. 276–284 (2008)
21. Oren, N., Norman, T.J.: Arguing Using Opponent Models. In: McBurney, P., Rahwan, I., Parsons, S., Maudet, N. (eds.) ArgMAS 2009. LNCS, vol. 6057, pp. 160–174. Springer, Heidelberg (2010)
22. Oren, N., Norman, T.J., Preece, A.: Arguing with confidential information. In: Proceedings of the 18th European Conference on Artificial Intelligence, Riva del Garda, Italy, pp. 280–284 (August 2006)
23. Parsons, S.: Normative Argumentation and Qualitative Probability. In: Gabbay, D.M., Kruse, R., Nonnengart, A., Ohlbach, H.J. (eds.) FAPR 1997 and ECSQARU 1997. LNCS, vol. 1244, pp. 466–480. Springer, Heidelberg (1997)

24. Patel, J., Teacy, W.T.L., Jennings, N.R., Luck, M., Chalmers, S., Oren, N., Norman, T.J., Preece, A., Gray, P.M.D., Shercliff, G., Stockreisser, P.J., Shao, J., Gray, W.A., Fiddian, N.J., Thompson, S.: Agent-based virtual organisations for the grid. Multiagent and Grid Systems 1(4), 237–249 (2006)
25. Pearl, J.: Probabilistic reasoning in intelligent systems: networks of plausible inference. Morgan Kaufmann Publishers Inc., San Francisco (1988)
26. Pechoucek, M., Marík, V., Bárta, J.: A knowledge-based approach to coalition formation. IEEE Intelligent Systems 17, 17–25 (2002)
27. Pollock, J.L.: Cognitive Carpentry. Bradford/MIT Press (1995)
28. Rahwan, T.: Algorithms for Coalition Formation in Multi-Agent Systems. Ph.D. thesis, University of Southampton (2007)
29. Riveret, R., Prakken, H., Rotolo, A., Sartor, G.: Heuristics in argumentation: A game theory investigation. In: Computational Models of Argument: Proceedings of COMMA 2008, Toulouse, France, May 28-30, pp. 324–335 (2008)
30. Rotstein, N., Oren, N., Norman, T.J.: Resource bounded argumentation frameworks. Tech. rep., University of Aberdeen (2011)
31. Sutton, R.S., Barto, A.G.: Reinforcement Learning: An Introduction (Adaptive Computation and Machine Learning). The MIT Press (March 1998)
32. Teacy, W.T.L., Patel, J., Jennings, N.R., Luck, M.: Travos: Trust and reputation in the context of inaccurate information sources. Autonomous Agents and Multi-Agent Systems 12(2), 183–198 (2006)