Relaxing Independence Assumptions in Probabilistic Argumentation

Hengfei Li, Nir Oren, and Timothy J. Norman

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

Abstract. Probabilistic argumentation frameworks (PrAFs) are a novel extension to standard argumentation systems, enabling one to reason about the likelihood of a set of arguments appearing within an extension. However, PrAFs assume that the likelihood of arguments appearing is independent of the presence of other arguments. In this paper, we lift this restriction through the introduction of probabilistic evidential argumentation frameworks (PrEAFs). Our extension captures probabilistic dependencies through the use of a support relation, as used in bipolar argumentation frameworks. After describing PrEAFs and their properties, we present algorithms for computing PrEAF semantics.

1 Introduction

Both argumentation and probability theory are important to automated reasoning. Argumentation is a promising approach to defeasible reasoning. Probability theory has a long history of modelling uncertainties in classical deductive logic. While the need to combine both of these approaches has long been recognised, work in this area is relatively sparse, and aimed at meeting very different goals. For example, in [23] probabilities serve as a measure of argument strength. In Li et al. [9], probabilities are assigned to arguments and defeat relations in a Dung argumentation framework [3]. A probability-based semantics is then defined to give the likelihood of a set of arguments being justified. This approach models uncertainties in abstract argumentation frameworks using probabilities and is able to probabilistically quantify the justifiability of arguments according to the underlying argumentation semantics. In this paper, we relax a critical, unrealistic assumption made by [9], namely that arguments are probabilistically independent. Since arguments often build on one another, such an assumption places too many restrictions to enable the modelling of real world scenarios.

In this paper, we develop probabilistic evidential argumentation frameworks, built on top of the bipolar argumentation framework described in [13]. The latter framework models inter-argument dependencies via a support relation. By assigning probabilities to supports, we are able to capture conditional likelihoods between arguments.

We start by describing probabilistic argument frameworks, then in Section 3 we introduce probabilistic evidential argument frameworks and their semantics.
We describe how these frameworks are instantiated in Section 4, before placing this research in context and presenting our conclusions.

2 Probabilistic Argumentation Framework

In this section, we briefly introduce probabilistic argumentation frameworks and explain the probabilistic independence assumption. We start with the definition of Dung’s abstract argumentation framework [3] as necessary background.

Definition 1. (Dung Argumentation Framework) A Dung argumentation framework (DAF) is a pair \( \langle A, D \rangle \) where \( A \) is a set of arguments, and \( D \subseteq A \times A \) is a defeats relation.

A set of arguments \( S \) is conflict-free iff there does not exist \( a, b \in S \) such that \( (a, b) \in D \). An argument \( a \) is acceptable w.r.t. a set of arguments \( S \) iff for all \( b \in A \) such that \( (b, a) \in D \), there exists \( c \in A \) such that \( (c, b) \in D \). A set of arguments \( S \) is admissible iff it is conflict free and all its arguments are acceptable w.r.t. \( S \).

From these definitions, different semantics have been defined by Dung and others [1]. The purpose of these semantics is to identify sets of arguments which are, in some intuitive sense, compatible with each other. For example, the grounded semantics yields a single extension which is the least fixed point of the characteristic function \( F_{AF}(S) = \{ a | a \in A \text{ is acceptable w.r.t } S \} \). We will utilise the function \( \xi^S : 2^A \times AF \rightarrow [0,1] \) to determine whether some set of arguments within an argument framework has some desirable property. For example, the function \( \xi^{Gnd}(A, D) \) could return 1 if \( A \) is in the grounded extension of \( D \), and 0 otherwise.

A probabilistic argumentation framework [9] associates uncertainties with a DAF by assigning probabilities to its arguments and defeats. These represent the likelihood that the arguments and defeats appear in the framework.

Definition 2. (Probabilistic Argumentation Framework) A probabilistic argumentation framework (PrAF) is a tuple \( \langle A, P_A, D, P_D \rangle \) where \( \langle A, D \rangle \) is a DAF, \( P_A : A \rightarrow (0 : 1] \) and \( P_D : D \rightarrow (0 : 1] \) map arguments and defeats to probabilities.

A PrAF implicitly specifies a probability distribution over the set of DAFs that can be generated by selecting arguments and defeats from it. This process is called inducing a DAF.

Definition 3. (Inducible DAF) Given a PrAF \( \langle A, P_A, D, P_D \rangle \), a DAF \( I = \langle A^I, D^I \rangle \) can be induced iff all of the following hold.

- \( A^I \subseteq A \); and
- \( D^I \subseteq D \cap (A^I \times A^I) \); and
- \( \forall a \in A \text{ such that } P_A(a) = 1, a \in A^I \); and
- \( \forall (s, t) \in D \text{ where } P_A(s) = P_A(t) = 1 \text{ and } P_D((s, t)) = 1 \) it is the case that \( (s, t) \in D^I \).
We call such a DAF an inducible DAF (of the PrAF). We denote the set of all inducible DAFs of a PrAF as $I(\text{PrAF})$.

Note that $P_D$ is therefore conditional — the induced DAF must include its source and target arguments in order for the defeat to be present.

The semantics of a PrAF capture the likelihood that a set of arguments $X$ has the properties described by the function $\xi$.

**Definition 4. (Probability-based Semantics for PrAFs)**

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

$P^I_{\text{PrAF}}(AF)$ denotes the probability that $AF$ is induced from the PrAF. This latter probability is computed from the joint probability of all arguments and defeats appearing or not appearing within the induced framework. Assuming probabilistic independence, this is computed as follows (where $A^I, D^I$ respectively represent the arguments and defeats in the induced AF, and $A, D$ the arguments and defeats of the PrAF).

$$P^I_{\text{PrAF}}(AF) = \prod_{a \in A^I} P_A(a) \prod_{a \in A \setminus A^I} (1 - P_A(a)) \prod_{d \in D^I} P_D(d) \prod_{d \in D \setminus D^I} (1 - P_D(d)) \quad (1)$$

where $DA = \{(a, b) | a, b \in A^I \text{ and } (a, b) \in D\}$ represents the defeats that may appear in $AF$.

[9] argue that without further information about the contents of the arguments, the independence assumption is justifiable. This view is also taken up by [7], who identifies situations where this assumption is acceptable. In structured argumentation frameworks, however, arguments are constructed from sub-arguments, and the absence of one such sub-argument could prevent the full argument from appearing. Therefore, this independence assumption is too restrictive in many situations. For instance, consider the “John is married” example of [2]: argument $a_1$ claims “John is married” backed by the premise “John wears something that looks like a wedding ring”, while this argument is in turn a sub-argument supporting argument $a_2$ for “John has a wife”. Argument $a_3$ attacks $a_1$ by stating “John is a bachelor because he often goes out until late with his friends”. Clearly, argument $a_2$ is dependent on argument $a_1$. For one thing, in terms of the dependence of knowledge, one cannot know $a_2$ without the awareness of $a_1$. For another, from the aspect of evaluation of arguments, the justification status of $a_2$ is affected by that of $a_1$, i.e., $a_2$ can be justified only when $a_1$ is justified. However, PrAFs fail to correctly model such situations. By assuming independence, unrealistic argument frameworks such as frameworks with arguments $a_2$ but without $a_1$ can be induced in PrAFs.
3 Probabilistic Evidential Argumentation Framework

We utilise evidential argumentation frameworks [13] to lift the PrAF independence assumption. EAFs model both defeat and support between arguments, allowing us to capture conditional dependencies between individual arguments. Within an evidential argumentation framework (EAF), the acceptability of an argument depends on the acceptability of its supporting arguments. EAFs thus abstractly capture the sub-argument relation found in works such as [17, 18].

Definition 5. (Evidential Argumentation Framework) An evidential argumentation system is a tuple $\langle A, R_d, R_s \rangle$ where $A$ is a set of arguments, $R_d$ and $R_s$ are defeat and support relations of the form $(2^A \setminus \emptyset) \times A$, and there are no $x \in 2^A$, $y \in A$ such that $xR_dy$ and $xR_sy$. Furthermore, we assume the existence of an argument $\eta \in A$.

Argument $\eta$ represents incontrovertible premises, and is used as the basis for support of other arguments.

Definition 6. (Evidential Support) An argument $a \in A$ has evidential support from a set $S$ iff either

- $SR_s a$ where $S = \{\eta\}$; or
- $\exists T \subset S$ such that $TR_s a$ and $\forall x \in T$, $x$ has evidential support from $S \setminus \{x\}$.

$S$ is a minimum support for $a$ if $a$ has evidential support from $S$ and there is no $T \subset S$ such that $s$ is supported by $T$. We say that $a$ is evidentially supported by $S$.

Definition 7. (Evidence Supported Defeat) A set $S$ carries out an evidence supported defeat on an argument $a$ iff $XR_d a$ where $X \subseteq S$, and all elements $x \in X$ are supported by $S$.

An evidence supported defeat by a set $S$ on $a$ is minimal iff $S$ carries out an evidence supported defeat on $a$, and there is no $T \subset S$ such that $T$ carries out an evidence supported defeat on $a$.

Definition 8. (Auxiliary Notions for EAFs) Argument $a$ is acceptable w.r.t. a set $S$ iff $S$ supports $a$, and for any minimal evidence-supported defeat by $X \subseteq 2^A$ against $a$, $\exists T \subseteq S$ s.t. $TR_d x$, where $x \in X$ so that $X \setminus \{x\}$ is no longer an evidence-supported defeat on $a$.

$S \subseteq A$ is conflict free iff $\forall y \in S$, $\exists X \subseteq S$ s.t. $XR_d y$.

$S \subseteq A$ is self supporting iff $\forall x \in S$, $S$ supports $x$.

If an argument is not evidentially supported, then it is not acceptable, and will not take part in any evidence-supported defeat. Such an argument cannot, therefore, affect the justification status of any other argument in an EAF.

By assigning a probability to the support relation, as was done in PrAFs for the defeat relation, we can obtain what is effectively a conditional probability between arguments and their supporting arguments.
Definition 9. (Probabilistic Evidential Argumentation Framework) A probabilistic evidential argumentation framework (PrEAF) is a tuple $\langle A, R_d, R_s, P_s \rangle$ where $\langle A, R_d, R_s \rangle$ is an EAF and $P_s : R_s \rightarrow (0, 1]$ represents the probability of a support. We assume $|A|$ is finite.

If $a$ is evidentially supported by $S$, then for any set $S'$ s.t. $a \in S'$, it is not the case that $S'R_d a'$ where $a' \in S$.

Our PrEAF definition prohibits support loops, and we assume that PrEAFs contain a finite set of arguments. These assumptions are necessary for the correctness of our algorithms.

As for PrAFs, we can define inducible EAFs.

Definition 10. (Inducible EAFs of a PrEAF) An EAF $I = \langle A_I, R_I^d, R_I^s \rangle$ can be induced from a PrEAF $P = \langle A, R_d, R_s, P_s \rangle$ iff all of the following hold.

- $A_I \subseteq A$ and $\eta \in A_I$
- $R_I^s \subseteq R_s$
- $R_I^d = R_d \cap (A_I \times A_I)$
- $\forall S \subseteq A, S' \subseteq A' \text{ s.t. } P_s(SR_s a) = 1, SR_s a \in R_I^s$
- $\forall a \in A \setminus \{\eta\} \text{ s.t. } \exists SR_s a \in R_I^s, a \in A_I$

$I$ is referred to as an inducible EAF of $P$. The set of all inducible EAFs of a PrEAF $P$ is denoted $I(P)$.

PrAFs model uncertain DAFs while PrEAFs model uncertain EAFs. We can therefore define PrEAF semantics in an analogous manner, identifying the likelihood that a set of arguments is justified under some EAF semantics function $\xi^S$, defined as for PrAFs.

Definition 11. (Probability-based Semantics for PrEAFs)

$$P_{PrEAF}(X) = \sum_{EAF \in I(PrEAF)} P_{PrEAF}(EAF)\xi^S(X, EAF)$$

An EAF $E$ for which $P_{PrEAF}(E) = 0$ is an invalid inducible EAF, and we need only consider valid inducible EAFs when computing $P_{PrEAF}(X)$.

Given a PrEAF $\langle A, R_s, R_d, P_s \rangle$, a na"ive way to identify valid inducible EAFs is to check every element in $2^A$. If such a combination does not include any
unsupported argument, then this set of arguments, together with the associated supports and defeats, forms a valid inducible EAF. Given a valid inducible EAF, its probability of being induced is the joint probability that those arguments in the EAF appear and those not in the EAF do not appear. The former depends on the probabilities that its supporting arguments are present, which, in turn, depends on the probabilities that those supporting the supporting arguments appear, etc. For every inducible EAF, therefore, we need to propagate the probabilities from the argument η. For different inducible EAFs, the propagation goes through the same paths but ends at different positions. We can, therefore, construct a recursive algorithm for identifying inducible EAFs, and to compute their likelihood.

4 Finding Inducible EAFs

We begin by describing the process used to generate all valid EAFs. Following this, we describe how the probability of an EAF being induced can be computed.

4.1 Inducible EAF Generation

Recall that every argument (except η) must be evidentially supported in a valid inducible EAF, and that η is self supported, providing support to all arguments with a path to it from η via the support relation. This means we can visualise all inducible EAFs of a PrEAF as a tree. Starting from the inducible EAF that contains only η as the root, at each level a valid EAF is generated from its parent by adding arguments that can be supported by the parent’s arguments. Figure 2 illustrates this for the PrEAF of Figure 1.

To formalise the expansion procedure, we define the relationship between a parent EAF and its children, these being the EAFs obtained by expanding the parent through the introduction of additional arguments.
Definition 12. (Child EAFs) Let PrEAF = \(\langle A, R_d, R_s, P_s \rangle\) be a PrEAF and \(F = \langle A^F, R^F_d, R^F_s \rangle\) be an inducible EAF of PrEAF. \(F\) is the parent of an inducible EAF \(C = \langle A^C, R^C_d, R^C_s \rangle\) iff

1. \(A^F \subset A^C\);
2. whenever \(F \neq \langle \{\eta\}, \emptyset, \emptyset \rangle\), then \(\forall a \in A^C \setminus A^F, \exists B \subseteq A^F\) such that \((B, a) \in R^C_s\) and \(B \not\subseteq A^F\). Here, \(A^F\) are the arguments present in the parent of \(F\).
3. \(R^C_d = R^F_d \cup (R_d \cap (A^C \times A^C))\)
4. \(R^C_s = R^F_s \cup (R_s \cap (A^C \times A^C))\)

The second requirement in the definition above means that for every argument in the child EAF but not in the parent EAF, there must exist at least one support linking this argument and a set of supporting arguments in \(A^F\). This set of supporting arguments must contain ‘new’ arguments; i.e., arguments in \(F\) but not in the parent of \(F\). Doing so ensures that no EAFs can have identical children. Effectively, this definition allows us to create EAFs by creating EAFs in a breadth-first manner over the PrEAF argument graph. Starting with \(\langle \{\eta\}, \emptyset, \emptyset \rangle\), we recursively generate all its child EAFs, their children, and so on, until no further children can be generated.

4.2 Computing the Probabilities of EAFs being Induced

We now turn our attention to computing the likelihoods associated with each of these EAFs. Given a PrEAF, the probability of obtaining an inducible EAF is the joint probability that those arguments in the EAF appear and those not in the EAF do not appear. We first write this probability as a product of these two terms. For a given set of arguments \(x\), \(P(x)\) denotes the probability associated with this set of arguments appearing in some specific EAF induced from the PrAF. Similarly, \(P(\bar{x})\) is the probability that arguments \(x\) do not appear within this EAF. The probability that a specific EAF, \(I = \langle A^I, R^I_d, R^I_s \rangle\), is induced is

\[
P_{PrEAF}^{I}(\langle A^I, R^I_d, R^I_s \rangle) = P(A^I) \times P((A \setminus A^I)|A^I)\] (2)

For example, considering the EAF of Figure 1, the likelihood of inducing an EAF containing arguments \(\eta, a\) and \(d\) is the probability that \(\eta, a\) and \(d\) appear in the EAF, times and the probability of arguments \(b\) and \(c\) not being present given that \(\eta, a\) and \(d\) do appear, i.e.

\[
P(\{\eta, a, d\}) \times P(\{b, c\}| \{\eta, a, d\})
\]

We consider the computing each part of the joint distribution separately. First, consider \(P((A \setminus A^I)|A^I)\), which is the probability that all arguments except those in \(A^I\) are not present within the induced EAF given that the arguments in \(A^I\) are present. Now let us consider two situations.

1. One of these undesirable arguments supports an argument in \(A^I\). Here, we can identify two further cases.
(a) An undesirable argument forms part of a unique chain of supports to some argument in $A^I$. In this case, requiring the undesirable argument to be present means that there is some argument in $A^I$ that is not supported, resulting in the probability of inducing $A^I$ being 0. We can therefore disregard this case.

(b) While the undesirable arguments form part of a chain of supports to an argument in $A^I$, there is some other chain of support containing only arguments in $A^I$. In such a situation, some part of $A^I$ will potentially support the undesirable argument (as $\eta$ must always form part of $A^I$ for the induced EAF to have a non-zero likelihood). This situation therefore reduces to the next case.

2. Some arguments in $A^I$ supports an undesirable argument.

The set of support links we must consider for an undesirable argument $a$, is defined by

$$\{r_s | r_s \in R_s \setminus R^I_s \text{ s.t. } r_s = (S,a) \text{ and } S \subseteq A^I\}$$

$P((A \setminus A^I)|A^I)$ is then equivalent to the probability that each of these support links is not induced for every undesirable argument. In other words,

$$P((A \setminus A^I)|A^I) = \prod_{\{r_s | r_s \in R_s \setminus R^I_s \text{ and } \text{Src}(r_s) \subseteq A^I\}} (1 - P_s(r_s)) \quad (3)$$

Here $\text{Src}(r_s)$ denotes the set of source arguments of $r_s$. From hereon, we will often refer to the first or second element within the attack and support relations. Therefore, given an element $r = (S,t)$ of this relation, we write $\text{Src}(r)$ to refer to $S$, and $\text{Tgt}(r)$ to refer to $t$.

To compute $P(A^I)$, we utilize recursion within our expansion algorithm, i.e., we compute $P(A^I)$ based on the $P(A^F)$ where $F = (A^F, R_s^F, R_d^F)$, the parent EAF.

$$P(A^I) = \begin{cases} 1 & \text{if } I = \langle \{\eta\}, \emptyset, \emptyset \rangle \\ P(A^F) \times P((A^I \setminus A^F)|A^F) & \text{otherwise} \end{cases} \quad (4)$$

Here, $P((A^I \setminus A^F)|A^F)$ is the conditional probability of the arguments present in the induced EAF, but not in the parent, given the arguments already present in the parent EAF.

The arguments in $A^I \setminus A^F$ are evidentially supported by the arguments in $A^F$. $P(A^F)$ thus represents the marginal probabilities, with the prior obtained from the PrEAF’s $P_s$ value for every argument in $A^I \setminus A^F$. Now consider those support relations for arguments in $A^I \setminus A^F$. We define the support relations by which such an argument is obtained.

**Definition 13.** (Sups) The $\text{Sups}$ function identifies the support relations introduced between a parent EAF and a child EAF $I$ for a specific argument $a$, given the arguments found in the parent’s parent EAF $A^F$ and the arguments found in the parent EAF $A^F$.

$$\text{Sups}(a, I, A^F, A^F') = \{r_s | r_s \in R_s, \text{Src}(r_s) \subseteq A^F, \text{Src}(r_s) \not\subseteq A^F' \text{ and } \text{Tgt}(r_s) = a\} \quad (5)$$
where \( a \in A^I \backslash A^F, A^F' \) is the set of arguments within the parent of \( F \) and if \( F = (\{\eta\}, \emptyset, \emptyset) \), then \( A^F = \emptyset \).

If \( |\text{Sups}(a, I, A^F, A^F')| = 1 \), then there is only a single support from the parent to this argument, and so \( P(a|A^F) = P_s(r_s) \), meaning that:

\[
P(A^I \backslash A^F|A^F) = \prod_{\{a|a \in A^I \backslash A^F \text{ and } Tgt(r_s) = a\}} P_s(r_s)
\]

It is possible for \( a \) to have multiple supports from the parent EAF. In this case, the presence of any of these could lead to \( a \) being present within \( A^I \), and we must therefore consider the probability of at least one of these appearing, thus:

\[
P(a|A^F) = 1 - \prod_{r_s \in \text{Sups}(a, I, A^F, A^F')} (1 - P_s(r_s)) \tag{6}
\]

Equation 6 provides the probability of a single new argument appearing given the arguments in the parent appear. The probability of a set of new arguments appearing, \( P((A^I \backslash A^F)|A^F) \), is, therefore:

\[
\prod_{a \in A^I \backslash A^F} \left( 1 - \left( \prod_{r_s \in \text{Sups}(a, I, A^F, A^F')} (1 - P_s(r_s)) \right) \right) \tag{7}
\]

Equation 7, together with 4, can now be used within equation 2 to compute the likelihood that a specific EAF is induced, given the likelihood that its parent is induced.

Algorithm 1 takes a PrEAF, an EAF induced from this PrEAF, the arguments found in its parent and parent’s parent (grandparent), as well as the likelihood of the parent’s arguments being induced (\( P(A^F) \)). It then computes Equations 3 (Lines 2–5) and 7 (Lines 8–15), before computing \( P_I^\text{PrEAF} \) for \( I \). The algorithm is called for each of \( I \)'s children, with the set of PrEAF, \( P_I^\text{PrEAF} \) pairs being returned.

Given a finite PrEAF, this algorithm will always terminate, as any argument within a parent will not be visited when a child is created as long as the PrEAF has no support loops.

Computing the likelihood of a set of arguments within a PrEAF being justified under some EAF semantics is then implemented by Algorithm 2. This algorithm requires a semantics function, which takes in a set of arguments and an EAF, and returns \( \top \) if the arguments are justified according to the semantics; for example, this function could check whether the argument appears within one, or all of the preferred extensions of the EAF. The algorithm also takes in a PrEAF, and the arguments whose justification is being tested. It operates by considering every induced EAF of the PrEAF, and its probability of existence, obtained from Algorithm 1, and tests whether the arguments in question
Algorithm 1 Expanding an EAF

Require: $PrEAF = \langle A, R_s, R_d, P_s \rangle$: a PrEAF
Require: $I = \langle A^I, R^I_s, R^I_d \rangle$: an inducible EAF of $PrEAF$
Require: $A^F$: Arguments present in the parent of $I$
Require: $A^{F'}$: Arguments present in the grandparent of $I$
Require: $P^F \in [0..1]$: $P(A^F)$ for $F$ (c.f. Eqn 4)

1: function Expand($PrEAF, I, A^F, A^{F'}, P^F$)
2:     $P_1 \leftarrow 1$
3:     for all $r_s \in R_s \setminus R^I_s$ where $\text{Src}(r_s) \subseteq A^I$ do
4:         $P_1 \leftarrow P_1 \times (1 - P_s(r_s))$
5:     end for
6:     $P_2 \leftarrow 1$
7:     if $I \neq \langle \{\}, \emptyset, \emptyset \rangle$ then
8:         for all $a \in A^I \setminus A^F$ do
9:             $P_3 \leftarrow 1$
10:            for all $r_s \in \text{Sups}(a, I, A^F, A^{F'})$ do
11:                $P_3 \leftarrow P_3 \times (1 - P_s(r_s))$
12:            end for
13:            $P_2 \leftarrow P_2 \times (1 - P_3)$
14:     end for
15:     end if
16:     $P' \leftarrow P^F \times P_2$
17:     $P_{PrEAF} \leftarrow P' \times P_3$
18:     out = $\{I, P_{PrEAF}\}$
19:     for all Children $C$ of $I$ s.t. $C = \langle A^C, R^C_s, R^C_d \rangle$ do
20:         out = out $\cup$ Expand($PrEAF, C, A^I, A^F, P^F$)
21:     end for
22:     return out
23: end function

are justified for this EAF. This algorithm returns the total likelihood of such a justification.

Note that the initial parameters passed to the Expand() function contain the empty sets for the parent and grandparent arguments. The Sups function requires both of these to function. The self supporting nature of $\eta$ means that the initial likelihood $P^F$ must be 1.

PrAFs, as described in [9], assign probabilities to arguments and defeats. However, later work [12], associates probabilities with the arguments alone. For simplicity, we followed the latter approach in describing PrEAFs and our algorithms. Including probabilities over defeats, however, is trivial. We can introduce an extra set of probabilities $P_d$ within our PrEAF, which identify the likelihood of a defeat existing, assuming that its source and target exist within the EAF. We may then utilise the PrAF semantics function, which returns a probability rather than $\top$ or $\bot$ within $\xi^S$ in Algorithm 2, replacing lines 5–7 with the line

$$ P \leftarrow P + L \times \xi^S(A, I) $$
Algorithm 2 Computing the probability-based semantics

Require: A semantics function $\xi^S: 2^A \times EAF \rightarrow [\top, \bot]$
Require: $PrEAF = \langle A, R_s, R_d, P_s \rangle$: a PrEAF
Require: $A$: a set of arguments

1: function $\text{Compute}(sem, PrEAF, A)$
2: $P_{sem} \leftarrow 0$
3: $E \leftarrow \text{Expand}(PrEAF, \langle \{\eta\}, \emptyset, \emptyset, \emptyset, \emptyset \rangle, \emptyset, \emptyset, \emptyset)$
4: for all $(I, P_{I,PrEAF}) \in E$ do
5: if $\xi^S(A, I) = \top$ then
6: $P_{sem} \leftarrow P_{sem} + P_{I,PrEAF}$
7: end if
8: end for
9: return $P_{sem}$
10: end function

Unlike PrAFs, PrEAFs assign probabilities to the support relations. Nevertheless, any PrAF can be encoded as a PrEAF by requiring every argument (except $\eta$) to be supported only by $\eta$, and the probability over the support between $\eta$ and this argument is equivalent to the probability over the argument in PrAF. This, together with the modification suggested above allows us to represent a PrAF as an PrEAF.

Finally, regarding the complexity of our algorithms, consider a PrEAF with $n + 1$ arguments $\{\eta, a_1, ..., a_n\}$. If every argument $a_i$ is supported by $\eta$, then the number of inducible EAFs is $2^{n+1}$, resulting in exponential time and space complexity when computing argument probabilities. On the other hand, a linear support relation, i.e., $R_s = \{\{\eta\}R_s a_1, \{a_1\}R_s a_2, ..., \{a_{n-1}\}R_s a_n\}$, means that we can determine argument likelihoods in $\Theta(n + 1)$ time, assuming $\xi^S$ is $O(n)$.

4.3 Example

To demonstrate our framework, consider two of the inducible EAFs from Figure 2. First, we examine the inducible EAF with arguments $\eta, a$ and $b$. The parent of this inducible EAF is $\langle \{\eta, a\}, \langle\{\eta\}, a\rangle, \emptyset, \emptyset \rangle$. The probability of arguments $\eta$ and $a$ appearing $P(\{\eta, a\})$ is 0.8. (Note that this probability is different from the probability of $\langle\{\eta\}, a\rangle, \emptyset$ being induced.) Applying Equations 4 and 7, we have the probability of $\eta, a$ and $b$ existing:

$$P(\{\eta, a, b\}) = P(\{\eta, a\}) \times (1 - (1 - P_s(\{\{a\}, b\}))) = 0.32$$

The probability of argument $c$ not existing given $\eta, a$ and $b$ exist is $P(\lnot c | \{\eta, a, b\}) = 1 - 0.7 = 0.3$. Therefore the probability of the EAF with arguments $\eta, a$ and $b$ being is induced is:

$$P_{\text{PrEAF}}(\{\eta, a, b\}, \langle\{\eta\}, a\rangle, \langle\{a\}, b\rangle, \emptyset) = P(\{\eta, a, b\}) \times P(\lnot c | \{\eta, a, b\}) = 0.096$$
Next, consider the $\eta$-only EAF. By Equation 4 and Equation 3, we have:

\[
P(\{\eta\}) = 1
\]
\[
P(\{a,d\}|\{\eta\}) = (1 - P_s(\{\eta, a\})) \times (1 - P_s(\{\eta, b\})) = 0.08
\]

Finally, we have:

\[
P^{f}_{\text{PrEAF}}(\{\eta\}, \emptyset, \emptyset) = P(\{\eta\}) \times P(\{a,d\}|\{\eta\}) = 0.08
\]

5 Discussion

PrEAFs extend PrAFs by allowing conditional probabilities over arguments. Dependencies between arguments are explicitly modelled in a PrEAF. In contrast, PrAFS do not provide sufficient information to handle situations where such dependencies occur, forcing the probabilistic independence assumption, which is justified in some, but not all situations [7].

We begin by surveying some of the most relevant work combining argumentation and probability. In [6], Hunter assigns probabilities to a logical language and interprets them as degrees of beliefs. The probability of an argument is derived from the probabilities of its premises. [6] focuses on coherent probability distributions over the logical language. Such a distribution results in a rational (with respect to the structure of the argument graph) probability distribution over arguments. This work provides valuable insights on the relationship between the probability distribution over the logical language and the distribution over the arguments built from the language. The work of Rienstra [20] is very similar in spirit to ours. Rienstra assigns probabilities to rules in a simplified version of the ASPIC+ framework [18], and describes the likelihood that some set of arguments is justified. This approach can be viewed as an instantiation of our PrEAF where supports are captured by strict and defeasible rules. Efficient methods for computing the semantics are not provided, however. Dung and Thang [4] investigate probabilistic argumentation in jury-based disputes. At an abstract level, their approach is similar to PrAFs. To generate abstract arguments with probabilities, they propose a framework based on assumption-based argumentation. Essentially, they allow a subset of inference rules to be probabilistic rules, representing the probability that the conclusion holds given that the premises hold. However, they do not allow the premises of a probabilistic rule to contain other probabilistic rules, which means a probabilistic argument can only depend on non-probabilistic arguments. Riveret et al. [22] study winning chances in dialogue games which are evaluated using Dung’s grounded semantics. Probabilities are assigned to premises and the independence of premises is assumed. Given these probabilities, the likelihood of winning a dialogue is computed, assuming that grounded semantics are used when determining argument justification status. Other works investigate probability and argumentation outside Dung-style argumentation. For example, Kohalas [8] quantifies the reliability of supporting arguments by probabilities in assumption-based reasoning. Pfeifer [16] studies
the conditional probability between premises and conclusions with an emphasis on the strength of arguments.

One question that repeatedly arises in probabilistic argumentation frameworks is with regards to the interpretation of probabilities. [6], for example, views these as the degree of truth of propositions. In such interpretations, probabilities are directly related to the veracity of an argument, or encode a measure of the likelihood of an argument being accepted. They therefore represent some form of argument strength, and can therefore be linked with weighted argumentation frameworks [5]. Unlike these approaches, PrAFs and PrEAFs take an epistemic view of likelihood — they view an argument’s probability as the likelihood of an agent being aware of an argument, or more generally, as the probability that an argument exists in some context. Unlike the weight based approaches discussed above, we assume that all arguments have equal persuasive strength.

It is clear that an agent aware of some argument must also be aware of some chain of arguments supporting the original argument, and that knowledge of the final argument is therefore conditional on knowledge of the sub-arguments. PrAFs are unable to capture this conditionality, and PrEAFs overcome this critical deficiency.

To further justify our interpretation of probabilities adopted in PrAFs and PrEAFs, we compare them with probabilistic logic [11]. In classical deductive logic, sentences are either true or false, and the entailment relation says what must be true given some other sentences are true. Probabilistic logic extends this concept by associating sentences with probabilistic degrees between truth and falsehood. The logic then provides a calculus enabling probabilistic degrees of truth over the entailed sentences to be computed, representing the probabilities of the entailed sentences being true given the probabilities of the premises being true. PrAFs and PrEAFs share a similar intuition. Recall the idea of the argumentation-based semantics: an extension of an argument framework is a set of arguments that are justified given the arguments currently present in the argument framework and their interactions. In other words, it is the presence of the arguments and their interactions in an argument framework that justifies the arguments in the extensions. Building upon this foundation, PrAFs and PrEAFs assign probabilities to arguments as degrees between their presence and absence in an argument framework, and then derive probabilistic degrees of justification of arguments, captured by the probability-based semantics. Therefore, the approach proposed in [9] extends Dung’s abstract argumentation framework by probabilistically quantifying the justification status, and the work presented in the current paper further develops this approach by allowing conditional probabilities. This last step of development is of critical importance as it allows structured argumentation to be modelled, making it more applicable to real application domains.

Another question appears with regards to the source of such probabilities. Dung and Thang’s work [4] provide a good example where the probabilities over arguments are interpreted as the degrees that some juror accepts the arguments as acceptable, defeasible proofs.
One potential application of our work lies in the modelling of a dialogue participant’s knowledge. For example, consider a multi-agent system where agents interact through dialogue. In order to achieve their goals, agents must often act strategically in such dialogues [19]. In such situations, an opponent model can be used to determine which arguments should be advanced by an agent, based on what the agent believes the opponent knows [15, 12]. Recent work [21] has extended the basic opponent modelling approach to deal with uncertainty regarding the opponent’s knowledge through the use of a probability distribution over opponent models. However, no consideration was given as to the source of these distributions. We are currently investigating how PrEAFs can be used to generate probabilistic opponent models, taking into account the requirement that the opponent be aware of an argument’s supporting arguments.

We are pursuing several additional avenues of future work. First, EAFs are unable to model undercutting attacks. Instantiating a PrEAF-like approach using Modgil’s extended argumentation frameworks [10] would enable us to capture these types of inter-argument relations. Second, as mentioned in Section 4, the performance of our algorithms varies according to argumentation graph topologies. An empirical evaluation of our approaches would clearly be beneficial. [14] have demonstrated that EAF semantics can be obtained from a DAF. Applying this line of work to our probabilistic semantics, we may introduce conditional probabilities between arguments on DAFs.

6 Conclusion

In this paper we have presented probabilistic argumentation frameworks, in which the presence of an argument can depend on whether other arguments appear. Such dependencies occur in many situations, the most obvious being in sub-argument support. Existing models, such as PrAFs, are unable to model such dependencies, severely restricting their utility. Our approach allows us to model arguments that probabilistically depend on a conjunction or a disjunction of other arguments, and thus this framework can be instantiated using different logical argumentation models. Furthermore, we can represent PrAFs as PrEAFs, and are thus able to model a richer set of domains.

References


