Computational Models of Argument
F. Toni et al. (Eds.)

doi:10.3233/FAIA220149

Rule-PSAT: Relaxing Rule Constraints in Probabilistic Assumption-Based Argumentation

Xiuyi Fan
School of Computer Science and Engineering
Nanyang Technological University
Singapore

Abstract. Probabilistic rules are at the core of probabilistic structured argumentation. With a language $\mathcal{L}$, probabilistic rules describe conditional probabilities $\Pr(\sigma_0|\sigma_1,\ldots,\sigma_k)$ of deducing some sentences $\sigma_0 \in \mathcal{L}$ from others $\sigma_1,\ldots,\sigma_k \in \mathcal{L}$ by means of prescribing rules $\sigma_0 \leftarrow \sigma_1,\ldots,\sigma_k$ with head σ_0 and body $\sigma_1,\ldots,\sigma_k$. In Probabilistic Assumption-based Argumentation (PABA), a few constraints are imposed on the form of probabilistic rules. Namely, (1) probabilistic rules in a PABA framework must be acyclic, and (2) if two rules have the same head, then the body of one rule must be the subset of the other. In this work, we show that both constraints can be relaxed by introducing the concept of *Rule Probabilistic Satisfiability (Rule-PSAT)* and solving the underlying joint probability distribution on all sentences in $\mathcal{L}$. A linear programming approach is presented for solving Rule-PSAT and computing sentence probabilities from joint probability distributions.

Keywords. Probabilistic Argumentation, Probabilistic Satisfiability

1. Introduction

Probabilistic Assumption-based Argumentation (PABA) [1] provides a probabilistic extension to the Assumption-based Argumentation (ABA) framework [2] by allowing probabilistic rules in argument construction. As a form of probabilistic structured argumentation (along with p-ASPIC [3] and probabilistic argumentation with logic [4,5]) PABA was shown to admit as instances several other probabilistic argumentation approaches and with an implementation engine developed [6], and complexity results studied in [7].

A few design choices have been made in PABA to ensure its semantics and inference approaches sound. Namely:

1. if there are two rules with the same head having different probabilities, then the body of one rule must be the subset of the other (Definition 2.1, [1]);
2. there is no infinite path starting from a probabilistic parameter in its dependency graph in a PABA framework (Lemma 2.1, [1]).

Constraint 1 specifies that a probability sentence can only be deduced from at most one set of antecedents; whereas Constraint 2 specifies that paths leading to probability sentences must be acyclic. In this work, we show that both constraints can be relaxed by considering *Probabilistic Satisfiability* [8]. We see that the two constraints given by [1] are design choices to ensure probabilistic satisfiability. However, as we illustrate in this work, without these two constraints, there are cases where probabilistic satisfiability can still hold with well-understood inference processes available. In other words, we show that there is no intrinsic reason to disallow multiple rules with the same heads and cyclic graphs when constructing probabilistic extensions to ABA. Thus, this work provides a generalisation to the probabilistic rules given in PABA with a sound inference method for sentence probability calculation.

The rest of this paper is organised as the follows. Section 2 reviews two concepts introduced in the literature that are needed in this work. Section 3 introduces of *Rule-PSAT* that describe probabilistic consistency. Section 4 presents an inference approach for reasoning Rule-PSAT. Section 5 compares our work with Nilsson's probabilistic logic / satisfiability in detail. We conclude in Section 6.

2. Background

In this work, we need two notions, *deduction for a sentence*, and *complete conjunction set* of a language introduced in the literature as follows.

Given a language $\mathcal{L}$, and a set of rules $\mathcal{R}$ built with sentences in $\mathcal{L}$, a *deduction* [2] for $\sigma \in \mathcal{L}$ supported by $S \subseteq \mathcal{L}$ and $R \subseteq \mathcal{R}$ denoted $S \vdash^R \sigma$ is a finite tree with

- nodes labelled by sentences in $\mathcal{L}$ or by a special symbol τ that is not in $\mathcal{L}$,
- the root labelled by σ,
- leaves either labelled by τ or sentences in S,
- non-leaves labelled by σ' with, as children, the elements of the body of some rule in $\mathcal{R}$ with head σ', and R the set of all such rules.

Deduction is a fundamental concept in rule-based systems. We will refer to it in Section 3.

Given a language $\mathcal{L}$ with n sentences, the *Complete Conjunction Set (CC Set)* [9] of $\mathcal{L}$ is the set of 2^n conjunction of sentences such that each conjunction contains n distinct sentences. For instance, for $\mathcal{L} = \{\sigma_0, \sigma_1\}$, the CC set of $\mathcal{L} = \{\neg\sigma_0 \wedge \neg\sigma_1, \neg\sigma_0 \wedge \sigma_1, \sigma_0 \wedge \neg\sigma_1, \sigma_0 \wedge \sigma_1\}$. As we will discuss in the next section, a CC set defines the universe of all possible worlds given by the language.

3. Rule-PSAT

We start by introducing the core representation of this work, namely the notion of a *probabilistic rule (p-rule)*, as follows.

Definition 1. Given a language $\mathcal{L}$, a *probabilistic rule (p-rule)* is

$$\sigma_0 \leftarrow \sigma_1, \ldots, \sigma_k : [\theta]$$

for $k \geq 0, \sigma_i \in \mathcal{L}, 0 \leq \theta \leq 1$. σ_0 is referred to as the *head* of the p-rule, $\sigma_1, \ldots, \sigma_k$ the *body*, and θ the *probability*.

Given a language $\mathcal{L}$ and a set of p-rules $\mathcal{R}$, we say that $\mathcal{R}$ is *defined over* $\mathcal{L}$ iff all sentences in p-rules in $\mathcal{R}$ are in $\mathcal{L}$.

The rule in Definition 1 states that the probability of σ_0, when $\sigma_1 \ldots \sigma_k$ all hold, is θ. In other words, this rule states that $\Pr(\sigma_0|\sigma_1, \ldots, \sigma_k) = \theta$. Note that this is the same interpretation of probabilistic rules introduced in [1].

To introduce Rule-PSAT, we need to consider the set of sentences that are "deducible". This is constructed with the notion of deduction as follows.[1]

Definition 2. Given a language $\mathcal{L}$ and a set of p-rules $\mathcal{R}$ defined over $\mathcal{L}$, the *deducible set* $\mathcal{L}_0 = \{\sigma \in \mathcal{L} | \emptyset \vdash^R \sigma$, where $R \subseteq \mathcal{R}\}$.

The *deducible rules* $\mathcal{R}_0 = \{\sigma_0 \leftarrow \sigma_1, \ldots, \sigma_k : [\cdot] \in \mathcal{R} | \sigma_i \in \mathcal{L}_0, i = 0 \ldots k\}$.

We illustrate deducible set and rules in Example 1.

Example 1. Let $\mathcal{L} = \{\sigma_0, \sigma_1, \sigma_2, \sigma_3\}$, $\mathcal{R} = \{\sigma_0 \leftarrow \sigma_1 : [\alpha]; \sigma_1 \leftarrow: [\beta]; \sigma_2 \leftarrow \sigma_3 : [\gamma]\}$. We have $\mathcal{L}_0 = \{\sigma_0, \sigma_1\}$ and $R_0 = \{\sigma_0 \leftarrow \sigma_1 : [\alpha]; \sigma_1 \leftarrow: [\beta]\}$.

Definition 3. Given a language $\mathcal{L}$ and a set of p-rules $\mathcal{R}$, let Ω be the CC set of $\mathcal{L}_0$. A function $\pi : \Omega \to [0, 1]$ is a *consistent probability distribution* with respect to $\mathcal{R}$ on $\mathcal{L}$ for Ω iff:[2]

1. For all $\omega_i \in \Omega$,

$$0 \leq \pi(\omega_i) \leq 1. \tag{1}$$

2. It holds that:

$$\sum_{\omega_i \in \Omega} \pi(\omega_i) = 1. \tag{2}$$

3. For each p-rule $\sigma_0 \leftarrow: [\theta] \in \mathcal{R}_0$, it holds that:

$$\sum_{\omega_i \in \Omega, \omega_i \models \sigma_0} \pi(\omega_i) = \theta. \tag{3}$$

4. For each p-rule $\sigma_0 \leftarrow \sigma_1, \ldots, \sigma_k : [\theta] \in \mathcal{R}_0$, $(k > 0)$, it holds that:

$$\frac{\sum_{\omega_i \in \Omega, \omega_i \models \sigma_0 \wedge \ldots \wedge \sigma_k} \pi(\omega_i)}{\sum_{\omega_i \in \Omega, \omega_i \models \sigma_1 \wedge \ldots \wedge \sigma_k} \pi(\omega_i)} = \theta. \tag{4}$$

[1] We use the notion of "deduction" with the symbol "$\vdash^R$" introduced in Section 2 without modification by treating p-rules as non-probabilistic rules in this context.

[2] In this work, symbols $\neg$, $\wedge$, and $\models$ take their standard meaning as in classical logic.

Our notion of consistency as given in Definition 3 consists of two parts. Equations 1 and 2 assert π being a probability distribution over the CC set of $\mathcal{L}_0$; whereas equations 3 and 4 assert that each p-rule should be viewed as defining conditional probabilities for which the probability of the head of the p-rule conditioned on the body is the probability. In particular, Equation 3 can be viewed as a special case of 4 as when the body is empty, the head is conditioned on the universe. In other words, for p-rule $\sigma_0 \leftarrow: [\theta]$, we assert $\Pr(\sigma_0) = \theta$ with Equation 3; for $\sigma_0 \leftarrow \sigma_1, \ldots, \sigma_k : [\theta]$, we assert $\Pr(\sigma_0|\sigma_1, \ldots, \sigma_k) = \theta$ with Equation 4.

Example 2. (Example 1 continued.) $\Omega = \{\neg\sigma_0 \wedge \neg\sigma_1, \sigma_0 \wedge \neg\sigma_1, \neg\sigma_0 \wedge \sigma_1, \sigma_0 \wedge \sigma_1\}$.
From $\sigma_0 \leftarrow \sigma_1 : [\alpha]$, we have

$$\frac{\pi(\sigma_0 \wedge \sigma_1)}{\pi(\neg\sigma_0 \wedge \sigma_1) + \pi(\sigma_0 \wedge \sigma_1)} = \alpha. \tag{5}$$

From $\sigma_1 \leftarrow: [\beta]$, we have

$$\pi(\neg\sigma_0 \wedge \sigma_1) + \pi(\sigma_0 \wedge \sigma_1) = \beta. \tag{6}$$

π is a consistent probability distribution iff Equations 5 and 6 hold, as well as

$$\pi(\neg\sigma_0 \wedge \neg\sigma_1) + \pi(\sigma_0 \wedge \neg\sigma_1) + \pi(\neg\sigma_0 \wedge \sigma_1) + \pi(\sigma_0 \wedge \sigma_1) = 1, \tag{7}$$

and

$$0 \leq \pi(\neg\sigma_0 \wedge \neg\sigma_1), \pi(\sigma_0 \wedge \neg\sigma_1), \pi(\neg\sigma_0 \wedge \sigma_1), \pi(\sigma_0 \wedge \sigma_1) \leq 1. \tag{8}$$

With consistency defined, we are ready to define Rule-PSAT as follows.

Definition 4. The *Rule Probabilistic Satisfiability (Rule-PSAT)* problem is to determine for a set of p-rules $\mathcal{R}$ on a language $\mathcal{L}$, whether there exists a consistent probability distribution for the CC set of $\mathcal{L}_0$ with respect to $\mathcal{R}$.

Example 3. (Example 2 continued.) To test whether $\mathcal{R}$ is Rule-PSAT on $\mathcal{L}$, we need to solve Equations 5-8 for π as $\mathcal{R}$ is Rule-PSAT iff a solution exists. It is easy to see that this is the case as:

$$\begin{aligned} \pi(\sigma_0 \wedge \sigma_1) &= \alpha\beta \\ \pi(\neg\sigma_0 \wedge \sigma_1) &= \beta - \alpha\beta \\ \pi(\sigma_0 \wedge \neg\sigma_1) + \pi(\neg\sigma_0 \wedge \neg\sigma_1) &= 1 - \beta \end{aligned}$$

Since $0 \leq \alpha, \beta \leq 1$, we have $0 \leq \pi(\sigma_0 \wedge \sigma_1), \pi(\neg\sigma_0 \wedge \sigma_1) \leq 1$. We can let $\pi(\sigma_0 \wedge \neg\sigma_1) = 0$, $\pi(\neg\sigma_0 \wedge \neg\sigma_1) = 1 - \beta$ and obtain one solution for π. As the system is under-specified, we have infinitely many solutions to $\pi(\sigma_0 \wedge \neg\sigma_1)$ and $\pi(\neg\sigma_0 \wedge \neg\sigma_1)$ in the range of $[0,\, 1 - \beta]$.

The next example gives a p-rule set that is not Rule-PSAT.

Example 4. Let $\mathcal{R}$ contain three p-rules: $\sigma_0 \leftarrow \sigma_1 : [0.9], \sigma_0 \leftarrow: [0.8], \sigma_1 \leftarrow: [0.9]$. From $\sigma_0 \leftarrow \sigma_1 : [0.9]$ and Equation 4, we have

$$\frac{\pi(\sigma_0 \wedge \sigma_1)}{\pi(\sigma_0 \wedge \sigma_1) + \pi(\neg\sigma_0 \wedge \sigma_1)} = 0.9. \tag{9}$$

From $\sigma_1 \leftarrow: [0.9]$, we have

$$\pi(\sigma_0 \wedge \sigma_1) + \pi(\neg\sigma_0 \wedge \sigma_1) = 0.9. \tag{10}$$

Substitute (10) in (9), we have $\pi(\sigma_0 \wedge \sigma_1) = 0.81$.

From $\sigma_0 \leftarrow: [0.8]$, we have

$$\pi(\sigma_0 \wedge \sigma_1) + \pi(\sigma_0 \wedge \neg\sigma_1) = 0.8.$$

Thus, $\pi(\sigma_0 \wedge \neg\sigma_1) = -0.01$, which does not satisfy $0 \leq \pi(\omega_i) \leq 1$.

From a Rule-PSAT solution, which characterises a probability distribution over the CC set, one can compute sentence probabilities by marginalising over sentences. In other words, we can compute the probability of sentences by summing up $\pi(\omega_i)$.

Given a language $\mathcal{L}$ and a set of p-rules $\mathcal{R}$, if there is a consistent probability distribution π for Ω, the CC set of $\mathcal{L}_0$, with respect to $\mathcal{R}$, then for any $\sigma \in \mathcal{L}_0$, its probability $\Pr(\sigma)$ is:

$$\Pr(\sigma) = \sum_{\omega_i \in \Omega, \omega_i \models \sigma} \pi(\omega_i). \tag{11}$$

Clearly, from Definition 3, we see that $0 \leq \Pr(\sigma) \leq 1$ and $\Pr(\sigma) + \Pr(\neg\sigma) = 1$. (Note that although Equation 11 is similiar to 3, 11 refers to all sentences $\sigma \in \mathcal{L}_0$, whereas 3 refers to sentences that are heads of rules with an empty body.)

Example 5. (Example 3 continued.) Taking π shown in Example 3, (with $\pi(\sigma_0 \wedge \neg\sigma_1) = 0$) the probabilities of σ_0 and σ_1 can be computed as follows.

$$\Pr(\sigma_0) = \pi(\sigma_0 \wedge \sigma_1) + \pi(\sigma_0 \wedge \neg\sigma_1) = \alpha\beta$$
$$\Pr(\sigma_1) = \pi(\neg\sigma_0 \wedge \sigma_1) + \pi(\sigma_0 \wedge \sigma_1) = \beta$$

If a set of p-rules $\mathcal{R}$ is satisfiable, then the range of the probability of any sentence in $\mathcal{L}_0$ can be found with mathematical optimisation. The upper and the lower bounds of the probability of a sentence $\sigma \in \mathcal{L}_0$ can be found by maximising and minimising the RHS of Equation 11 subject to Equations 1-4, respectively.

Example 6. (Example 5 continued.) To compute the upper and lower bounds of $\Pr(\sigma_0)$, we maximise and minimise $\Pr(\sigma_0) = \pi(\sigma_0 \wedge \sigma_1) + \pi(\sigma_0 \wedge \neg\sigma_1)$, respectively. We see that $\Pr(\sigma_0)$ is at its max when $\pi(\sigma_0 \wedge \neg\sigma_1)$ is. Since $0 \leq \pi(\sigma_0 \wedge \neg\sigma_1) \leq 1-\beta$, we have the upper bound of $\Pr(\sigma_0)$ taking its value $\alpha\beta + 1 - \beta$. Similarly, $\Pr(\sigma_0)$ takes its min value when $\pi(\sigma_0 \wedge \neg\sigma_1) = 0$. Thus, the lower bound of $\Pr(\sigma_0)$ is $\alpha\beta$.

Note that there is no restriction imposed on the form of p-rules other than the ones given in Definition 1, as illustrated in the next two examples (Examples 7 and 8) - a set of p-rules can be consistent even if there are rules in this set forming cycles or having two rules with the same head.

Example 7. Consider a set of p-rules $\mathcal{R} = \{\sigma_0 \leftarrow \sigma_1 : [0.7], \sigma_1 \leftarrow \sigma_0 : [0.6], \sigma_1 \leftarrow: [0.5]\}$. We can see that there are infinitely many different (finite) deductions for both σ_0 and σ_1 due to the cycle formed by *deduce* σ_0 *from* σ_1 and *deduce* σ_1 *from* σ_0. However, we can still compute a (unique) solution for π over the CC set of $\{\sigma_0, \sigma_1\}$. Using Equations 2 to 4, we have:

$$\begin{aligned}
0.7 &= \pi(\sigma_0 \wedge \sigma_1)/(\pi(\sigma_0 \wedge \sigma_1) + \pi(\neg\sigma_0 \wedge \sigma_1)),\\
0.6 &= \pi(\sigma_0 \wedge \sigma_1)/(\pi(\sigma_0 \wedge \sigma_1) + \pi(\sigma_0 \wedge \neg\sigma_1)),\\
0.5 &= \pi(\sigma_0 \wedge \sigma_1) + \pi(\neg\sigma_0 \wedge \sigma_1),\\
1 &= \pi(\neg\sigma_0 \wedge \neg\sigma_1) + \pi(\neg\sigma_0 \wedge \sigma_1) + \pi(\sigma_0 \wedge \neg\sigma_1) + \pi(\sigma_0 \wedge \sigma_1).
\end{aligned}$$

Solutions found are: $\pi(\neg\sigma_0 \wedge \neg\sigma_1) = 0.27$, $\pi(\sigma_0 \wedge \neg\sigma_1) = 0.23$, $\pi(\neg\sigma_0 \wedge \sigma_1) = 0.15$, $\pi(\sigma_0 \wedge \sigma_1) = 0.35$.

Example 8. Consider a set of p-rules $\mathcal{R} = \{\sigma_0 \leftarrow \sigma_1 : [0.6], \sigma_0 \leftarrow \sigma_2 : [0.5], \sigma_1 \leftarrow: [0.7], \sigma_2 \leftarrow: [0.6]\}$, There are two p-rules with head σ_0. They have different bodies and probabilities. We set up equations as follows.[3]

$$\begin{aligned}
0.6 &= (\pi(111) + \pi(110))/(\pi(010) + \pi(011) + \pi(110) + \pi(111)),\\
0.5 &= (\pi(101) + \pi(111))/(\pi(001) + \pi(011) + \pi(101) + \pi(111)),\\
0.7 &= \pi(010) + \pi(011) + \pi(110) + \pi(111),\\
0.6 &= \pi(001) + \pi(011) + \pi(101) + \pi(111),\\
1 &= \pi(000) + \pi(001) + \pi(010) + \pi(011) + \pi(100) + \pi(101) + \pi(110) + \pi(111).
\end{aligned}$$

Solve these, a solution maximising $\Pr(\sigma_0)$ is follows: $(\Pr(\sigma_0) = 0.7)$

$$\begin{array}{llll}
\pi(000) = 0, & \pi(001) = 0.02, & \pi(010) = 0, & \pi(011) = 0.28,\\
\pi(100) = 0.15, & \pi(101) = 0.13, & \pi(110) = 0.25, & \pi(111) = 0.17.
\end{array}$$

A solution minimising $\Pr(\sigma_0)$ is: $(\Pr(\sigma_0) = 0.42)$

$$\begin{array}{llll}
\pi(000) = 0.14, & \pi(001) = 0.16, & \pi(010) = 0.14, & \pi(011) = 0.14,\\
\pi(100) = 0, & \pi(101) = 0, & \pi(110) = 0.12, & \pi(111) = 0.3.
\end{array}$$

[3] To simplify the presentation, Boolean values are used as shorthand for the sentences. E.g., 111, 011, and 001 denote $\sigma_0 \wedge \sigma_1 \wedge \sigma_2$, $\neg\sigma_0 \wedge \sigma_1 \wedge \sigma_2$, and $\neg\sigma_0 \wedge \neg\sigma_1 \wedge \sigma_2$, respectively.

4. Solve Rule-PSAT

Given a set of p-rules $\mathcal{R} = \{\rho_1, \ldots, \rho_m\}$ constructed on some language $\mathcal{L}$ such that $\mathcal{L}_0$ contains n sentences, to test whether $\mathcal{R}$ is Rule-PSAT, we set up a linear system

$$A\Pi = B, \tag{12}$$

where A is an $(m+1) \times 2^n$ matrix, $\Pi = [\pi(\omega_1), \ldots, \pi(\omega_{2^n})]^T$, B an $(m+1) \times 1$ matrix.[4] We construct A and B in a way such that R is Rule-PSAT iff Π has a solution in $[0,1]^{2^n}$, as follows.

For each rule $\rho_i \in \mathcal{R}$, if $\rho_i = \sigma_0 \leftarrow: [\theta]$ has an empty body, then

$$A[i,j] = \begin{cases} 1, & \text{if } \omega_j \models \sigma_0; \\ 0, & \text{otherwise;} \end{cases} \tag{13}$$

and

$$B[i] = \theta. \tag{14}$$

Otherwise, $\rho_i = \sigma_0 \leftarrow \sigma_1, \ldots, \sigma_k : [\theta]$, then

$$A[i,j] = \begin{cases} \theta - 1, & \text{if } \omega_j \models \sigma_0 \wedge \sigma_1 \wedge \ldots \wedge \sigma_k; \\ \theta, & \text{if } \omega_j \models \neg\sigma_0 \wedge \sigma_1 \wedge \ldots \wedge \sigma_k; \\ 0, & \text{otherwise;} \end{cases} \tag{15}$$

and

$$B[i] = 0. \tag{16}$$

Row $m+1$ in A and B are $1 \ldots 1$ and 1, respectively.

Example 9. (Example 6 continued.) Let $\rho_0 = \sigma_0 \leftarrow \sigma_1 : [\alpha]$, $\rho_1 = \sigma_1 \leftarrow: [\beta]$. Here, $m = 2$, $n = 2$. From Equations 12 to 16, we have

$$A = \begin{bmatrix} 0 & \alpha & 0 & \alpha - 1 \\ 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix},$$

$\Pi = [\pi(\neg\sigma_0 \wedge \neg\sigma_1), \pi(\neg\sigma_0 \wedge \sigma_1), \pi(\sigma_0 \wedge \neg\sigma_1), \pi(\sigma_0 \wedge \sigma_1)]^T$, and $B = [0, \beta, 1]^T$. It is easy to see that Π has solutions as shown in Example 3.

Theorem 1. Given a set of p-rules $\mathcal{R}$ on some language $\mathcal{L}$, $\mathcal{R}$ is Rule-PSAT iff Equation 12 has a solution for Π in $[0,1]^{2^n}$.

---

[4]We let $\{\omega_1, \ldots, \omega_{2^n}\}$ be the CC set of $\mathcal{L}_0$. We consider elements in this set being ordered with their Boolean values. E.g., for $\mathcal{L}_0 = \{\sigma_0, \sigma_1\}$, the four elements in the CC set are ordered such that $\{\omega_1 = \neg\sigma_0 \wedge \neg\sigma_1, \omega_2 = \neg\sigma_0 \wedge \sigma_1, \omega_3 = \sigma_0 \wedge \neg\sigma_1, \omega_4 = \sigma_0 \wedge \sigma_1\}$.

Table 1.: Performance Demonstration for Solving Rule-PSAT with the Python *scipy linprog* Library with an Interior-point Method.

Number of Sentences	6	7	8	9	10	11	12
Run Time (s)	0.014	0.022	0.041	0.11	0.55	3.32	22.38

Proof. (Sketch.) Equations 1 to 4 are satisfied by a Π solution in $[0,1]^{2^n}$ as follows.

1. If $\Pi \in [0,1]^{2^n}$, then $0 \leq \pi(\omega_i) \leq 1$ for all ω_i.
2. Since Row $m+1$ in A and B are 1s, we have the sum of all $\pi(\omega_i)$ being 1.
3. For each p-rule $\sigma_0 \leftarrow\!:\![\theta]$, Equations 13 and 14 ensure that Equation 3 is satisfied.
4. For each p-rule $\sigma_0 \leftarrow \sigma_1, \ldots, \sigma_k\!:\![\theta]$, Equation 15 and 16 ensure that Equation 4 is satisfied with simple algebra.

Thus, we see that Equation 12, $A\Pi = B$, is nothing but a linear system representation of Equations 1-4, which characterise probability distributions over the CC set of $\mathcal{L}_0$ with conditionals. □

Table 1 demonstrates the performance of a Python implementation of the linear system approach for solving Rule-PSAT introduced in this section. The implementation is built with the open source *scipy linprog* library[5], using an interior-point method. We observe that the average running time grows exponentially as the number of sentences in a set of p-rules. This is expected as the size of the CC set is 2^n (n the number of sentences in $\mathcal{L}_0$); and interior-point method has a super-linear complexity [10]. P-rules in this experiment are randomly generated with maximum length of rule body 4, the average time of 10 runs for each configuration is reported.

5. Discussion

Many works have been published on probabilistic argumentation in recent years, e.g. [11,12,13,14,15,16,17,4]. With very few exceptions, notably [4,5], existing works are predominantly defined with abstract argumentation, having probability distributions defined over argumentation graphs. In [4,5], arguments are constructed with probabilistic logic. As probabilistic rules are also used to construct arguments, we compare our work with probabilistic logic.

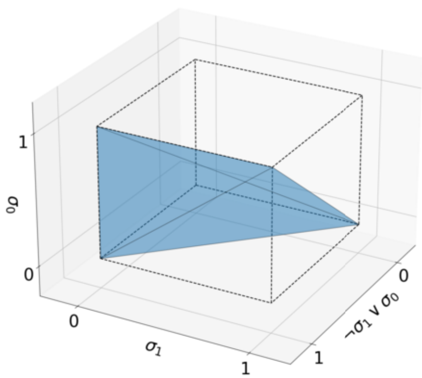
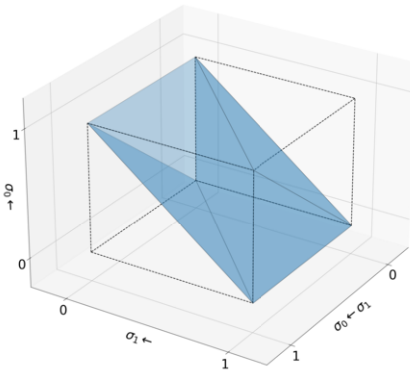
Nilsson [8] introduces Probabilistic Satisfiability with probabilistic logic, considering knowledge bases in Conjunctive Normal Form. A modus ponens example,[6]

$$\text{If } \sigma_1, \text{ then } \sigma_0.\ \sigma_1.\ \text{Therefore, } \sigma_0.$$

[5] `https://docs.scipy.org/doc/scipy/reference/generated/scipy.optimize.linprog.html`

[6] This example is used in [8]. The figure on the left hand side of Table 2 is a reproduction of Figure 2 in [8].

Table 2.: Comparison of Consistent Probability Regions between Nilsson's Probabilistic Logic and Probabilistic Rules on an modus ponens instance.

Probabilistic Logic	Probabilistic Rule
$\neg\sigma_1 \vee \sigma_0 : [\alpha], \quad \sigma_1 : [\beta], \quad \sigma_0 : [\gamma].$	$\sigma_0 \leftarrow \sigma_1 : [\alpha]$, $\sigma_1 \leftarrow: [\beta]$, $\sigma_0 \leftarrow: [\gamma]$.
	

is shown in Table 2. The probabilities of the conditional claim is α, the antecedent β and the consequent γ. With Nilsson's probabilistic logic, this is interpreted as:

$$\neg\sigma_1 \vee \sigma_0 : [\alpha], \quad \sigma_1 : [\beta], \quad \sigma_0 : [\gamma],$$

which gives rise to equations

$$\pi(\neg\sigma_1 \wedge \sigma_0) + \pi(\sigma_1 \wedge \sigma_0) + \pi(\neg\sigma_1 \wedge \neg\sigma_0) = \alpha, \tag{17}$$

$$\pi(\sigma_1 \wedge \sigma_0) + \pi(\sigma_1 \wedge \neg\sigma_0) = \beta, \tag{18}$$

$$\pi(\sigma_1 \wedge \sigma_0) + \pi(\neg\sigma_1 \wedge \sigma_0) = \gamma. \tag{19}$$

With probabilistic rules discussed in this work, the interpretation to modus ponens is the three p-rules as follows.

$$\sigma_0 \leftarrow \sigma_1 : [\alpha], \quad \sigma_1 \leftarrow: [\beta], \quad \sigma_0 \leftarrow: [\gamma],$$

which gives rise to equations 5, 18 and 19. The two shaded polyhedrons shown in Table 2 illustrate probabilistic consistent regions for α, β and γ, with probabilistic logic and probabilistic rule, respectively, as defined by their corresponding equations together with equations 1 and 2. The consistent region in the probabilistic logic case is a tetrahedron, with vertices (0,0,1), (1,0,0), (1,1,0) and (1,1,1). The consistent region in the probabilistic rule case is an octahedron, with vertices (0,0,0), (0,0,1), (0,1,0), (1,0,0), (1,1,0) and (1,1,1). It is argued in [18] that the conditional probability interpretation to modus ponens is more reasonable than the probabilistic logic interpretation in practical settings.

The principal benefit of this analysis comes from observing that both methods are nothing but imposing constraints on the feasible regions of the spaces defined by clauses (in the case of probabilistic logic) or p-rules (in the case of probabilistic rules). In this sense, reasoning on such probability and logic combined forms is about identifying feasible regions determined by solutions to Π in $A\Pi = B$.[7]

It is recognised that solving Π with large matrix A is difficult. The size of A is exponential to the number of sentences in the language; and linear programming methods are super-linear to the size of the CC set (as we illustrate in Section 4). Nilsson suggests that partition could be considered on B so Π can be solved with divide-and-conquer techniques. However, as [19] show that PSAT is NP-complete in its general form, it becomes more plausible to consider important and/or practically useful instances of the generic PSAT problem where reasoning does not rely on exact solution to the probability distribution on the CC set.

A few such instances are considered in the literature. For example, Williamson [20] discusses the case when sentences are *disjoint*. In such cases, for any two sentences σ_0, σ_1 in a language, we have

$$\Pr(\sigma_0 \wedge \sigma_1) = 0.$$

Henderson et.al [9,21,22] discuss the case when sentences are *independent*. In such cases, for any two sentences σ_0, σ_1 in a language,

$$\Pr(\sigma_0 \wedge \sigma_1) = \Pr(\sigma_0) \Pr(\sigma_1).$$

Both settings can greatly reduce the complexity of reasoning as one does not need to explicitly consider joint probabilities amongst sentences and thus one can work in the space defined by n sentences in the language, instead of considering solutions in the 2^n space formed by the CC set.

However, we believe neither of the two is suitable in probabilistic rule settings such as the one discussed in this work, as they both trivialise conditional probabilities. In other words, as $\Pr(\sigma_0|\sigma_1) = \Pr(\sigma_0 \wedge \sigma_1)/\Pr(\sigma_1)$ by the definition of conditional probability, assuming $\Pr(\sigma_0 \wedge \sigma_1) = 0$ makes $\Pr(\sigma_0|\sigma_1) = 0$; whereas assuming $\Pr(\sigma_0 \wedge \sigma_1) = \Pr(\sigma_0)\Pr(\sigma_1)$ makes $\Pr(\sigma_0|\sigma_1) = \Pr(\sigma_0)$. Effectively, these two assumptions make us commit to

$$\sigma \leftarrow \_ : [0] \text{ and } \sigma \leftarrow \_ : [\Pr(\sigma)]^{8}$$

for all p-rules over all sentences σ, respectively. Since neither of the two seems realistic in practical settings, we stay with solving the joint distribution on the CC set for computing sentence probabilities.[9]

[7] Constructions of A differ between Nilsson's probabilistic logic and this work. However, both are designed for solving the full joint probability distribution over the CC set.

[8] Here, _ stands for an anonymous variable as in Prolog.

[9] On this note, PABA also forces independence between their probabilistic parameters by having $\Pr(\omega) = \prod_{(Q,[a:x]) \in GE_\omega} x$, where $(Q, [a : x])$ is a deduction for a and GE_ω is the set of grounded extensions containing all possible worlds, in which a possible world ω is an element in the CC set of all probabilistic parameters (Definition 2.1 and Lemma 2.1 in [1]). However, as PABA also supports non-probabilistic rules and assumptions, the independence assumption is not imposed on all sentences in a PABA framework as some of them are not probabilistic.

6. Conclusion

In this work, we introduce a generalisation to probabilistic rules (p-rule) used in Probabilistic Assumption-based Argumentation. We show that by introducing Rule Probabilistic Satisfiability, we can accommodate probabilistic rules forming cycles and allow multiple rules with the same head but different bodies in the same p-rule set. A reasoning method using linear programming is introduced with a software implementation developed. This work can be viewed as a building block for probabilistic structured argumentation frameworks that use rules to construct arguments. Future work will focus on (more) efficient reasoning and / or approximation approaches.

Acknowledgements

I thank Dr Kristijonas Čyras for insightful discussions on PABA. I also thank the anonymous referees for their helpful suggestions.

References

[1] Dung PM, Thang PM. Towards (Probabilistic) Argumentation for Jury-based Dispute Resolution. In: Baroni P, Cerutti F, Giacomin M, Simari GR, editors. Computational Models of Argument: Proceedings of COMMA 2010, Desenzano del Garda, Italy, September 8-10, 2010. vol. 216 of Frontiers in Artificial Intelligence and Applications. IOS Press; 2010. p. 171-82. Available from: https://doi.org/10.3233/978-1-60750-619-5-171.

[2] Čyras K, Fan X, Schulz C, Toni F. Assumption-Based Argumentation: Disputes, Explanations, Preferences. IfCoLog JLTA. 2017;4(8).

[3] Rienstra T. Towards a Probabilistic Dung-style Argumentation System. In: Ossowski S, Toni F, Vouros GA, editors. Proceedings of the First International Conference on Agreement Technologies, AT 2012, Dubrovnik, Croatia, October 15-16, 2012. vol. 918 of CEUR Workshop Proceedings. CEUR-WS.org; 2012. p. 138-52. Available from: http://ceur-ws.org/Vol-918/111110138.pdf.

[4] Hunter A. Reasoning with Inconsistent Knowledge using the Epistemic Approach to Probabilistic Argumentation. In: Proceedings of the 17th International Conference on Principles of Knowledge Representation and Reasoning, KR 2020, Rhodes, Greece, September 12-18, 2020; 2020. p. 496-505. Available from: https://doi.org/10.24963/kr.2020/50.

[5] Hunter A. Argument Strength in Probabilistic Argumentation Using Confirmation Theory. In: Vejnarová J, Wilson N, editors. Symbolic and Quantitative Approaches to Reasoning with Uncertainty - 16th European Conference, ECSQARU 2021, Prague, Czech Republic, September 21-24, 2021, Proceedings. vol. 12897 of Lecture Notes in Computer Science. Springer; 2021. p. 74-88.

[6] Hung ND. Inference procedures and engine for probabilistic argumentation. International Journal of Approximate Reasoning. 2017;90:163-91. Available from: https://doi.org/10.1016/j.ijar.2017.07.008.

[7] Čyras K, Heinrich Q, Toni F. Computational complexity of flat and generic Assumption-Based Argumentation, with and without probabilities. Artificial Intelligence. 2021;293:103449. Available from: https://doi.org/10.1016/j.artint.2020.103449.

[8] Nilsson NJ. Probabilistic Logic. Artificial Intelligence. 1986;28(1):71-87. Available from: https://doi.org/10.1016/0004-3702(86)90031-7.

[9] Henderson TC, Simmons R, Serbinowski B, Cline M, Sacharny D, Fan X, et al. Probabilistic sentence satisfiability: An approach to PSAT. Artificial Intelligence. 2020;278. Available from: https://doi.org/10.1016/j.artint.2019.103199.

[10] Potra FA, Wright SJ. Interior-point methods. Journal of Computational and Applied Mathematics. 2000;124(1):281-302. Available from: https://www.sciencedirect.com/science/article/pii/S0377042700004337.

[11] Kohlas J. Probabilistic argumentation systems: A new way to combine logic with probability. Journal of Applied Logic. 2003;1(3-4):225-53. Available from: https://doi.org/10.1016/S1570-8683(03)00014-4.

[12] Hunter A. Some Foundations for Probabilistic Abstract Argumentation. In: Proc. of COMMA. vol. 245. IOS Press; 2012. p. 117-28. Available from: https://doi.org/10.3233/978-1-61499-111-3-117.

[13] Thimm M. A Probabilistic Semantics for abstract Argumentation. In: Raedt LD, Bessiere C, Dubois D, Doherty P, Frasconi P, Heintz F, et al., editors. ECAI 2012 - 20th European Conference on Artificial Intelligence. Including Prestigious Applications of Artificial Intelligence (PAIS-2012) System Demonstrations Track, Montpellier, France, August 27-31 , 2012. vol. 242 of Frontiers in Artificial Intelligence and Applications. IOS Press; 2012. p. 750-5. Available from: https://doi.org/10.3233/978-1-61499-098-7-750.

[14] Gabbay DM, Rodrigues O. Probabilistic Argumentation: An Equational Approach. Logica Universalis. 2015;9(3):345-82. Available from: https://doi.org/10.1007/s11787-015-0120-1.

[15] Hunter A, Thimm M. Probabilistic Reasoning with Abstract Argumentation Frameworks. Journal of Artificial Intelligence Research. 2017;59:565-611. Available from: https://doi.org/10.1613/jair.5393.

[16] Fazzinga B, Flesca S, Furfaro F. Complexity of fundamental problems in probabilistic abstract argumentation: Beyond independence. Artif Intell. 2019;268:1-29. Available from: https://doi.org/10.1016/j.artint.2018.11.003.

[17] Mantadelis T, Bistarelli S. Probabilistic abstract argumentation frameworks, a possible world view. Int J Approx Reason. 2020;119:204-19. Available from: https://doi.org/10.1016/j.ijar.2019.12.006.

[18] Nilsson N. Probabilistic Logic Revisited. Artificial Intelligence. 1993;59(1-2):39-42. Available from: https://doi.org/10.1016/0004-3702(93)90167-A.

[19] Georgakopoulos GF, Kavvadias DJ, Papadimitriou CH. Probabilistic satisfiability. Journal of Complexity. 1988;4(1):1-11. Available from: https://doi.org/10.1016/0885-064X(88)90006-4.

[20] Williamson J. Probability Logic. In: Gabbay D, Johnson R, Ohlbach HJ, Woods J, editors. Handbook of the Logic of Argument and Inference: the Turn Toward the Practical. Elsevier; 2002. p. 397-424.

[21] Henderson T, Simmons R, Serbinowski B, Fan X, Mitiche A, Cline M. Probabilistic Logic for Intelligent Systems. In: Strand M, Dillmann R, Menegatti E, Ghidoni S, editors. Intelligent Autonomous Systems 15. vol. 867 of Advances in Intelligent Systems and Computing. Cham: Springer International Publishing; 2018. p. 129-41. Available from: https://doi.org/10.1007/978-3-030-01370-7_11.

[22] Henderson TC, Simmons R, Sacharny D, Mitiche A, Fan X. A probabilistic logic for multi-source heterogeneous information fusion. In: 2017 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems, MFI 2017, Daegu, Korea (South), November 16-18, 2017. IEEE; 2017. p. 530-5. Available from: https://doi.org/10.1109/MFI.2017.8170375.