From Probabilistic Logic to Probabilistic Argumentation

Nico Potyka[1]

[1]*University of Stuttgart, Universitätsstraße 32, 70569 Stuttgart, Germany*

Abstract

One natural approach to probabilistic logic programming is Nilsson's probabilistic logic. We discuss the basic framework in the propositional setting, some reasoning ideas and extensions that allow handling inconsistent information. We then discuss its relationship to probabilistic epistemic argumentation. Roughly speaking, probabilistic epistemic argumentation is to classical argumentation as Nilsson's probabilistic logic is to classical logic. We discuss similarities and differences between the two and in which scenarios one approach may be better suited than the other.

Keywords

Probabilistic Logic, Probabilistic Epistemic Argumentation, Inconsistency Tolerance, Uncertain Reasoning

1. Introduction

The goal of this paper is to introduce the reader to some interesting relationships between probabilistic programming and probabilistic argumentation. As these terms encompass a variety of different models, let us first clarify what the focus of this paper is. To the best of my knowledge, the term probabilistic programming is not precisely defined. From my personal experience, the common characteristic of the different approaches is that they combine a formal language with probability theory in order to allow expressing uncertainty. Some approaches focus on the programming aspect. For example, Church [1] is a scheme-like probabilistic programming language that allows defining random variables on top of deterministic constructs. Probabilistic programming has also been used to refer to probabilistic extensions of mathematical programming [2]. The formal language here is the language of numerical optimization problems described by an objective function and constraints and uncertainty can be introduced, for example, on the coefficients of constraints. At another end of the spectrum, there are formalisms that combine logical languages and probability theory. One influential example and the main content of this paper is Nilsson's probabilistic logic [3]. Somewhere in between the more logical and the more programming-oriented approaches, there are probabilistic extensions of logic programming languages. One popular example is Problog [4] that extends the logical core of Prolog with uncertainty that can be expressed over facts and rules.

CEUR Workshop Proceedings (CEUR-WS.org)

Our focus here is on Nilsson's probabilistic logic [3]. Nilsson's ideas were not completely new and have been invented before and reinvented later. Some ideas can even be related back to Boole's work [5]. Nevertheless, Nilsson's work probably played a great part in establishing probabilistic logic programming in Artificial Intelligence and inspired several interesting extensions. Therefore, we will just refer to this particular approach to probabilistic logic as Nilsson's probabilistic logic to distinguish it from other approaches. From a high-level perspective, the framework allows enriching logical formulas with probabilities such that a formula with probability 1 corresponds to a classical logical formula, a formula with probability 0 corresponds to the negation of a classical logical formula and a formula with probability strictly between 0 and 1 balances between these extremes. Philosophically, the probabilities are usually interpreted as degrees of belief, that is, as the subjective probability that a formula is a true. Nilsson's idea can be applied to many logical languages including first-order languages. However, the original formalism seems most intuitive in the propositional setting. Later, several relational extensions have been introduced [6, 7, 8] that, roughly speaking, propositionalize first-order languages by using Herbrand semantics similar to classical probabilistic programming approaches. While they do not support the full power of first-order logic, they allow talking about properties of individuals and their relationships and the semantics is quite intuitive in many applications. Conceptually, they are similar to probabilistic models that have been considered in the area of statistical relational learning [9], which is also known as statistical artificial intelligence now. Almost all extensions of Nilsson's approach can be unified again by considering the logical atoms (propositional or ground atoms) as random variables and noticing that the logical constraints are essentially encoded as linear constraints over these random variables [10]. In this paper, we will mostly focus on propositional languages because it keeps the formalism simple. However, once the reader understands the propositional setting, she will not have problems to generalize the ideas to the relational setting or the more general setting of linear constraints over random variables.

Another popular knowledge representation formalism closely related to logic is argumentation. Argumentation can be roughly divided into structured approaches [11] that take account of the logical structure of arguments and abstract approaches [12] that focus merely on the relationships between arguments to determine which arguments can be accepted. Naturally, also many probabilistic extensions of classical argumentation models have been considered [13]. Probabilistic argumentation can, in fact, be seen as another instance of our working definition of probabilistic programming. The formal language is the language of argumentation problems and uncertainty can be introduced, for example, on arguments (degrees of acceptance) or relationships between them. We will focus on probabilistic epistemic argumentation here [14], which, roughly speaking is to classical abstract argumentation as Nilsson's probabilistic logic is to classical logic. The intuitive idea of the basic framework is that the meaning of attacks between arguments is encoded by probabilistic constraints. For example, we may want that the belief in an argument, should bound the belief in arguments that it attacks from above. As we discuss later, the original framework can be related to a very weak propositional probabilistic logic that only allows talking about atomic beliefs. The framework has been extended later by more and expressive constraints and in its current is closely related [15] to some of the more expressive extensions of Nilsson's probabilistic logic.

The remainder of this paper is structured as follows: We give an introduction to Nilsson's

probabilistic logic in Section 2. We start with a simple language in Section 2.1 and explain the basic reasoning problem in Section 2.2. In Section 2.3, we discuss some important extensions of the language. In particular, we explain the idea of relational extensions by means of a larger example. We finish the section with an overview of some ideas to make probabilistic logic inconsistency-tolerant. In Section 3, we give an analogously structured overview of epistemic probabilistic argumentation. We do not assume any familiarity with argumentation and sketch all necessary basics. As we will explain later, we will use some non-standard notation, to highlight the relationships to Nilsson's probabilistic logic. Again, we will start from the basic framework and then discuss basic reasoning problems, extensions of the basic language, give a larger example and explain how to make the framework robust against inconsistencies. Along the way, we will highlight similarities and differences to Nilsson's probabilistic logic and why both frameworks are interesting in their own right. We close the paper with some conclusions and some potential ideas for future research in the area.

2. Nilsson's Probabilistic Logic and Extensions

2.1. The Basic Framework

We start from a classical propositional language defined over a finite set of propositional atoms $\mathscr{A}$ using the logical connectives $\neg, \wedge, \vee, \rightarrow, \leftrightarrow$. An interpretation $I : \mathscr{A} \rightarrow \{0, 1\}$ is a mapping from atoms to truth values, where we associate *false* with 0 and *true* with 1. We let Int denote the set of all interpretations. The interpretation of complex formulas is defined as usual, for example, $I(\neg F) = 1 - I(F)$ and $I(F \wedge G) = I(F) \cdot I(G)$. We say that I satisfies a formula F iff $I(F) = 1$ and call I a model of F in this case. We let $\mathrm{Mod}(F) = \{I \in \mathrm{Int} \mid I \text{ satisfies } F\}$ denote the models of F.

The conceptual idea of Nilsson's probabilistic logic is simple. In the language of probability theory, we understand interpretations as elementary events and associate formulas with events by identifying them with the set of their models. What probabilistic logic adds to probability theory is a simple formal language to describe events. Formally, a probabilistic formula is an expression of the form $(F)[p]$, where F is a formula in our classical logical language and $p \in [0, 1]$ is a probability. Intuitively, $(F)[p]$ says that we believe that F is true with probability p. Probabilistic interpretations are then probability distributions $P : \mathrm{Int} \rightarrow [0, 1]$ (such that $\sum_{I \in \mathrm{Int}} P(I) = 1$) over interpretations. We extend the domain of P to arbitrary formulas by letting $P(F) = \sum_{I \in \mathrm{Mod}(F)} P(I)$. This definition is elegant and simple and has some immediate intuitive consequences. For example, it is easy to check from the definition that

- $P(\top) = 1$ for every tautological formula $\top$,
- $P(\neg F) = 1 - P(F)$,
- $P(F) \leq P(G)$ if F logically entails G (which means that $\mathrm{Mod}(F) \subseteq \mathrm{Mod}(G)$).

A probabilistic interpretation P satisfies a probabilistic formula $(F)[p]$ iff $P(F) = p$. With this, Nilsson's propositional probabilistic logic is already completely defined.

Table 1
Two probabilistic interpretations that satisfy $\mathcal{K}_1$.

a	b	P_1	P_2
0	0	0.5	0
0	1	0	0.5
1	0	0.5	0.5
1	1	0	0

2.2. Basic Reasoning Problems

A probabilistic knowledge base $\mathcal{K}$ is a finite set of probabilistic formulas. There are two fundamental reasoning problems. The *probabilistic satisfiability problem* asks if there is a probabilistic interpretation P that satisfies every probabilistic formula in $\mathcal{K}$. If $\mathcal{K}$ is satisfiable, we are interested in what $\mathcal{K}$ entails about the probability of formulas. It may come as a surprise to the reader that we actually cannot necessarily derive point probabilities. This should be clear for the empty knowledge base as it is satisfied by every probabilistic interpretation. Therefore, for every formula, we can only entail that the probability must be between 0 and 1. However, even in non-trivial cases, our knowledge bases do not necessarily entail point probabilities.

Example 2.1. *Consider the knowledge base $\mathcal{K}_1 = \{(a)[0.5], (a \rightarrow b)[0.5]\}$. Table 1 shows two interpretations. It is easy to check from the definition that they satisfy $\mathcal{K}_1$. However, we have $P_1(b) = 0$ and $P_2(b) = 0.5$. By doing a more sophisticated analysis, one can actually show that if there are two probability distribution that satisfy $\mathcal{K}$ and assigns probabilities p_1 and p_2 to a formula F, then for every $p \in [p_1, p_2]$, there must be another probability distribution P that satisfies F and assigns probability p to F. In other words, the entailed probabilities form an interval. In our example, one can check that $\mathcal{K}$ entails that the probability of b must be in the interval $[0, 0.5]$.*

Since the entailed probabilities form an interval (a point probability is the special case of an interval of length 0), the entailment problem is usually solved by computing the lower and upper bound. This leads to two optimization problems. Given a probabilistic knowledge base $\mathcal{K}$ and a formula F over the same language, the *probabilistic entailment problem* is to solve the following minimization and maximization problem:

$$\min_P / \max_P P(F)$$

$$\text{subject to } P \text{ satisfies } \mathcal{K}.$$

If the lower bound is l and the upper bound is u, we write $\mathcal{K} \vDash (F)[l, u]$. Both the probabilistic satisfiability and the entailment problems are linear optimization problems [3]. However, the number of optimization variables corresponds to the number of interpretations in the propositional language. Therefore, the naive optimization problems grow exponentially with respect to the number of atoms in the language. The probabilistic satisfiability problem is indeed NP-hard even when all formulas in $\mathcal{K}$ correspond to Krom clauses [16]. Recall that a Krom clause is a literal or a disjunction of two literals and that the classical satisfiability problem for Krom clauses (2SAT) is in P. However, it is also interesting to note that the general probabilistic satisfiability problem remains in NP, that is, is NP-complete [16]. A well established method to

deal with probabilistic satisfaction and entailment problems with hundreds of propositional atoms is to apply column generation methods [16, 17].

2.3. Enriching the Language

Since knowledge bases generally entail interval probabilities, a first natural extension is to allow interval probabilities in knowledge bases. Probabilistic formulas have then the form $(F)[l, u]$ and a probabilistic interpretation P satisfies such a formula iff $l \leq P(F) \leq u$. Computationally, nothing changes for the satisfiability and entailment problem. The optimization problems remain exponentially large linear optimization problems.

Another important extension are conditional probabilities. Sometimes people tend to apply probabilistic implications instead, but it is important to note that they do not have the same meaning. $(F \rightarrow G)[p]$ says that the probability that the statement $F \rightarrow G$ is true is p. It does not speak about the probability of G under the assumption that F is true. The latter is indeed what conditional probability captures. Probabilistic conditionals are written in the form $(G \mid F)[l, u]$ and can be read as 'the conditional probability of G given F is between l and u'. Roughly speaking, satisfaction is defined by the usual definition of conditional probability. That is, P satisfies $(G \mid F)[l, u]$ if $l \leq \frac{P(F \wedge G)}{P(F)} \leq u$. However, this expression is undefined if $P(F) = 0$. Therefore, the precise definition is usually that P satisfies $(G \mid F)[l, u]$ iff $l \cdot P(F) \leq P(F \wedge G) \leq u \cdot P(F)$, which is obtained from the original inequality by multiplying all terms by $P(F)$. This form has the additional advantage that it again corresponds to linear inequalities, which means that the probabilistic satisfiability and entailment problem again maintain the original structure. One may try to additionally enforce the condition $P(F) > 0$ and there is some philosophical discussion about potential advantages and disadvantages of this stricter definition of satisfaction. However, one can show that, in most cases, it does not make a big difference for the entailed probabilities [18]. Let us highlight again that probabilistic implications are different from probabilistic conditionals. To see formally that $(G \mid F)[l, u]$ is different from $(F \rightarrow G)[l, u]$, just notice that the satisfaction condition for $(F \rightarrow G)[l, u]$ is $l \leq P(F \rightarrow G) \leq u$, or equivalently, $l \leq P(\neg F \vee G) \leq u$, which is clearly different from the condition $l \cdot P(F) \leq P(F \wedge G) \leq u \cdot P(F)$ for $(G \mid F)[l, u]$.

When allowing conditionals in the knowledge base, it is natural to also allow conditionals in queries. The entailment problem then, more generally, becomes

$$\min_P / \max_P P(G \mid F)$$

$$\text{subject to } P \text{ satisfies } \mathcal{K}.$$

For $F \equiv \top$, one obtains the original problem. While conditional probabilities lead to a non-linear objective function of the optimization problem, fractional programming techniques can be applied to transform the problem into an equivalent linear optimization problem again [19, 5].

In the following, we can simply assume that probabilistic knowledge bases contain only conditionals of the form $(G \mid F)[l, u]$. For $l = u$, we get the special case of point probabilities and for $F \equiv \top$, we get the special case of unconditional probabilistic formulas. Implementations benefit from treating these special cases separately, but the algorithmic complexity will change only by a constant.

Another interesting extension is to move to more expressive logical languages. A discussion of this topic is out of scope here, but we can illustrate some key ideas in an example. Figure 1 shows a small relational knowledge base that contains some general beliefs about relationships between some animals and domain knowledge about the bird *tweety*, the cat *sylvester* and the dog *hector*.

//Animal types are disjoint
$(bird(X) \wedge cat(X))[0]$,
$(bird(X) \wedge dog(X))[0]$,
$(cat(X) \wedge dog(X))[0]$,

//Type of some individuals
$(bird(tweety))[1]$,
$(cat(sylvester))[1]$,
$(dog(hector))[1]$

//largerThan is asymmetric (>)
$(largerThan(X,Y) \wedge largerThan(Y,X))[0]$,

//attacks is irreflexive
$(attacks(X,X))[0]$

//cats are usually larger than birds
$(largerThan(X,Y) \mid cat(X) \wedge bird(Y))[0.9]$,

//cats often attack birds
$(attacks(X,Y) \mid cat(X) \wedge bird(Y))[0.8]$,

//animals are unlikely to attack larger animals
$(attacks(X,Y) \mid largerThan(Y,X))[0.1]$.

Figure 1: Example of a relational knowledge base.

There are different ways to define the semantics of such knowledge bases, but one simple way is to transform it into a propositional knowledge base [6, 7]. To this end, we consider all ground atoms over a finite set of constants and use them as the propositional atoms of our language. In our example, we simply use the constants from the knowledge base, namely the set $\{tweety, sylvester, hector\}$. For every predicate symbol, we then create a propositional atom for every possible instantiation of the arguments. For every relational conditional, we create a propositional conditional for every ground instance. For example, $(bird(X) \wedge cat(X))[0]$ will be replaced with the three instantiations $(bird(tweety) \wedge cat(tweety))[0]$, $(bird(sylvester) \wedge cat(sylvester))[0]$ and $(bird(hector) \wedge cat(hector))[0]$. In this way, we end up with a propositional knowledge base that we can treat as before. The number of interpretations grows quickly in relational knowledge bases, but we can often exploit deterministic knowledge in order to keep the number of interpretations manageable [20]. For example, $(bird(tweety) \wedge cat(tweety))[0]$ tells us that every interpretation that interprets both $bird(tweety)$ and $cat(tweety)$ as true necessarily has probability 0 and therefore can be ignored. Some results that we can infer from the knowledge base in Figure 1 are the following:

- $\mathcal{K} \vDash (attacks(tweety, sylvester))[0.09, 0.19]$,
- $\mathcal{K} \vDash (attacks(tweety, tweety))[0]$,
- $\mathcal{K} \vDash (attacks(tweety, hector))[0, 1]$,
- $\mathcal{K} \vDash (attacks(sylvester, tweety))[0.8, 0.8]$.

The first example shows again that we may entail interval probabilities even if all conditionals in the knowledge base contain point probabilities. The rough intuitive reasoning in this example is as follows: We know that *tweety* is a bird and that *sylvester* is a cat. Therefore, there is a 90% chance that *sylvester* is larger than *tweety*. In this case, there is only a 10% chance that *tweety* attacks *sylvester*. This makes up for the lower bound 0.09. However, there is also a 10% chance that *sylvester* is not larger than *tweety*. In this case, we can say nothing about the probability that *tweety* attacks *sylvester*, which basically makes up for the 10% uncertainty that are reflected in the upper bound 0.19. The probabilities in this example have been computed with the Java library *Log4KR* that can be downloaded as part of the *Kreator* project[1]. The documentation of the library can be found in the appendix of [20].

Finally, let us mention that the language can be extended further indefinitely. For example, one can take the logical closure of probabilistic formulas (probabilistic formulas are then the atoms of a boolean language) or assign probabilities to probabilistic formulas that express the degree of belief that the degree of belief in the original formula is correct (second-order probabilities). This ideas can be applied repeatedly, so that an infinite number of possible extensions of the basic language exists. However, the computational problems also become significantly harder, so that we will not discuss these extensions here. A systematic investigation and classification of such extensions can be found in [21].

2.4. Dealing with Inconsistencies

When a classical knowledge base is inconsistent, it entails everything. This is one of the few properties that do not generalize from classical to Nilsson's probabilistic logic immediately. When a probabilistic knowledge base is inconsistent, there is no interpretation that satisfies the knowledge base. Therefore, the optimization problem corresponding to the probabilistic entailment problem simply has no solution. One may close the definition gap by just saying that it entails arbitrary conditionals in this case to maintain consistency with classical logic. In practice, it does not make a big difference because the entailment results would be meaningless in this case. What seems more important is a way to deal with inconsistent knowledge bases. Like in classical logic, one could try to repair the knowledge base or define new reasoning procedures that can derive non-trivial results from inconsistent knowledge bases. Some examples for approaches to repair inconsistent probabilistic knowledge bases can be found in [22, 23, 24]. Some inconsistency-tolerant reasoning approaches can be found in [25, 26].

We discuss the inconsistency-tolerant reasoning approach in [26] in a little bit more detail because it allows generalizing many reasoning problems (including the probabilistic entailment problem) to inconsistent knowledge bases without changing their asymptotic complexity. The conceptual idea is very simple. The probabilistic entailment problem is undefined when a knowledge base is unsatisfiable. To overcome the problem, one can consider probability distributions that satisfy the knowledge base 'as well as possible'. One way to do this is to consider probability distributions that violate the knowledge base in a minimal way. The violation can be measured with respect to the linear constraints that encode the meaning of conditionals in the knowledge base [27]. It is interesting to note that the problem of computing the degree

---

[1]https://sourceforge.net/projects/kreator-ide/

of violation can be solved by a linear optimization problem with almost the same structure as the probabilistic satisfiability problem. Once the degree of violation has been computed, a generalized probabilistic entailment problem can be solved by a linear optimization problem with almost the same structure as the probabilistic entailment problem. In particular, if the knowledge base is consistent, the generalized probabilistic entailment results are equal to the standard probabilistic entailment results. Furthermore, the generalized probabilistic entailment problem has a continuity property that guarantees that when a knowledge base is close to a consistent knowledge base, the generalized reasoning results will be close to the reasoning results under the consistent knowledge base [10]. Hence, by replacing probabilistic satisfiability tests with a computation of minimal violation, and replacing probabilistic entailment with generalized probabilistic entailment, Nilsson's probabilistic logic can be made robust against inconsistencies without making the computational problems harder. In a similar way, it can be extended to stratified knowledge bases where conditionals partitioned with respect to priorities [28].

3. Probabilistic Epistemic Argumentation and Extensions

3.1. The Basic Framework

While Nilsson's probabilistic logic extends classical logic, probabilistic epistemic argumentation [14] extends classical abstract argumentation [12]. While classical logic basically asks for the consequences of a set of formulas, classical abstract argumentation basically asks for the acceptable arguments among arguments that may attack each other. In general, there is not a unique answer, but no or multiple sets of arguments may be acceptable. Due to this characteristic, probabilistic epistemic argumentation does not generalize classical argumentation as smoothly as probabilistic logic generalizes classical logic. Probabilistic epistemic argumentation is similar to probabilistic logic in the sense that the goal is to assign a unique degree of belief to every argument (as opposed to having multiple possible answers). The degree of belief is again represented by a probability interval (including the special case of point probabilities).

Instead of a set of propositional atoms, we now have a set of abstract arguments that we again denote by $\mathscr{A}$. For our purposes, an argument is simply something that can be accepted or rejected. We refer to [29] for a more sophisticated discussion. Instead of having conditionals that express relationships between propositional atoms, we now have edges that express relationships between arguments. In the most basic framework, there are only attack edges. Intuitively, if a attacks b, then accepting a should imply that we reject b. An argumentation framework can then be defined as a pair $(\mathscr{A}, \text{Att})$ consisting of a set of arguments $\mathscr{A}$ and a set $\text{Att} \subseteq \mathscr{A} \times \mathscr{A}$ of attacks between them. With a slight abuse of notation, we let $\text{Att}(a) = \{b \in \mathscr{A} \mid (b, a) \in \text{Att}\}$ denote the attackers of argument a.

The goal of probabilistic epistemic argumentation is to add a degree of belief to every argument. To this end, we proceed similar to probabilistic logic. We start again by defining interpretations $I : \mathscr{A} \leftarrow \{0, 1\}$ that can accept (1) or reject (0) arguments. In the context of abstract argumentation, it is more common to consider sets of accepted arguments instead, but we use interpretations here to highlight the similarity. Note that there is a 1-1-correspondence between the two because every interpretation can be associated with the set of arguments that

it accepts and vice versa. We let again Int denote the set of all interpretations and consider probability distributions $P : \text{Int} \to [0, 1]$ (such that $\sum_{I \in \text{Int}} P(I) = 1$) over interpretations. The question is now, when does a probabilistic interpretation P satisfy an argumentation framework $(\mathcal{A}, \text{Att})$? To answer this question, probabilistic epistemic argumentation allows defining the meaning of attack edges by constraints over probability distributions. While many different options exist, we present only a small selection from [30] here. We say that P respects

Rationality if $(a, b) \in \text{Att}$ and $P(a) > 0.5$ implies that $P(b) \leq 0.5$,

Coherence if $(a, b) \in \text{Att}$ implies that $P(b) \leq 1 - P(a)$,

Foundedness if $\text{Att}(a) = \varnothing$ implies $P(a) = 1$,

Optimism if $P(a) \geq 1 - \sum_{b \in \text{Att}(a)} P(b)$.

To understand the intuition of these constraints, it is important to know that probability 0.5 is associated with indifference, whereas probabilities greater than 0.5 tend towards acceptance and probabilities smaller than 0.5 tend towards rejection. Intuitively, the constraints generalize ideas from classical abstract argumentation. Rationality states that if a attacks b and we accept a, then we should not accept b. Coherence encodes the same idea in a more continuous way. As our belief in a increases, our belief in b must decrease. Foundedness generalizes Dung's idea that arguments can be accepted when it can be successfully argued against their attackers [12]. In the context of an unattacked argument, it simply means that the argument can be accepted. Foundedness demands the belief in such an argument is 1. Optimism respects this idea, but may also give a non-trivial lower bound for attacked arguments. The lower bound linearly decreases with respect to the belief in the attackers of an argument.

A probabilistic epistemic argumentation framework can now be seen as a triple $(\mathcal{A}, \text{Att}, \mathcal{C})$ consisting of an argumentation framework and a collection of constraints like Coherence and Optimism. A probability distribution P satisfies the framework iff it satisfies all constraints in $\mathcal{C}$.

3.2. Basic Reasoning Problems

In order to compute degrees of belief, we can proceed analogously to probabilistic logic. Among all propbability distributions that satisfy a probabilistic epistemic argumentation framework $(\mathcal{A}, \text{Att}, \mathcal{C})$, we minimize and maximize the probability of an argument to obtain the lower and the upper bound for the belief, respectively. As these beliefs can only be computed if the argumentation framework is satisfiable, it also makes sense to consider a satisfiability problem. Formally, the *probabilistic epistemic satisfiability problem* asks if a given probabilistic epistemic argumentation framework is satisfiable. The *probabilistic epistemic entailment problem* takes an argument a as additional input and asks for the solutions of the following minimization and maximization problem:

$$\min_P / \max_P P(a)$$
$$\text{subject to } P \text{ satisfies } (\mathcal{A}, \text{Att}, \mathcal{C}).$$

The difficulty of the probabilistic epistemic entailment problem depends on the nature of the constraints contained in $\mathcal{C}$. The constraints proposed in [30] are almost all linear. The only

exception is the Rationality constraint. Recall that it states that a probability distribution must satisfy $P(b) \leq 0.5$ whenever $(a, b) \in \text{Att}$ and $P(a) > 0.5$. This is equivalent to saying that for all $(a, b) \in \text{Att}$, we must have $P(a) \leq 0.5$ or $P(b) \leq 0.5$, where the or is the logical inclusive or. This is a disjunction of two linear constraints that is computationally difficult to handle. However, all other constraints in [30] can be brought into the form

$$\sum_{i=1}^{k} c_i \cdot P(a_i) \leq c_0,$$

where $a_1, \ldots, a_k \in \mathscr{A}$ are arguments and $c_0, \ldots, c_n \in \mathbb{R}$. Note that this form also encompasses equalities because an equality $\sum_{i=1}^{k} c_i \cdot P(a_i) = c_0$ is equivalent to two inequalities $\sum_{i=1}^{k} c_i \cdot P(a_i) \leq c_0$ and $\sum_{i=1}^{k} c_i \cdot P(a_i) \geq c_0$. Constraints of this general form are called linear atomic constraints and one can show that the probabilistic epistemic satisfiability and entailment problems can be solved in linear time when only linear atomic constraints are involved [31]. Intuitively, the reason for this is that the constraints only involve probabilities over atomic arguments. Therefore, one can show that probability distributions $P : \text{Int} \rightarrow [0, 1]$ can be replaced with *probability labellings* $P : \mathscr{A} \rightarrow [0, 1]$ that assign probabilities to arguments directly without changing the semantics. Note, in particular, that the probabilistic epistemic satisfiability problem is indeed simpler than the probabilistic satisfiability problem over Krom clauses that is NP-hard as we discussed before. For the same reason that probabilistic satisfiability over Krom clauses is NP-hard, allowing the conjunction or disjunction of two arguments in constraints immediately renders the probabilistic epistemic satisfiability problem NP-hard [31].

3.3. Enriching the Language

A first natural extension of the basic framework is to not only consider attack, but also support edges. This corresponds to the idea of bipolar abstract argumentation [32, 33]. A bipolar abstract argumentation framework is a triple $(\mathscr{A}, \text{Att}, \text{Sup})$ that adds an additional support relation $\text{Sup} \subseteq \mathscr{A} \times \mathscr{A}$ of attacks between them. As before, with a slight abuse of notation, we let $\text{Sup}(a) = \{b \in \mathscr{A} \mid (b, a) \in \text{Sup}\}$ denote the supporters of argument a. There is some philosophical discussion about how support should be defined, but one natural idea is to consider it as the opposite of attack [34]. Whereas an attacker weakens the attacked argument, a supporter should strengthen it. One may then introduce symmetric constraints for support. For example, we could say that P respects

S-Rationality if $(a, b) \in \text{Sup}$ and $P(a) > 0.5$ implies that $P(b) > 0.5$,

S-Coherence if $(a, b) \in \text{Sup}$ implies that $P(b) \geq P(a)$.

The ideas are symmetrical to Rationality and Coherence for attacks. S-Rationality says, intuitively, that if a supports b and we accept a, then we must accept b as well. S-Coherence again encodes a similar idea in a continuous way. As the belief in a increases, our belief in b must increase as well. Similar to the attack constraints, S-Rationality is a difficult non-linear constraint that corresponds to a discjunction of two constraints. S-Coherence is a linear atomic constraint. As we discussed before, the probabilistic epistemic satisfiability and entailment problems can be solved in polynomial time for linear atomic constraints.
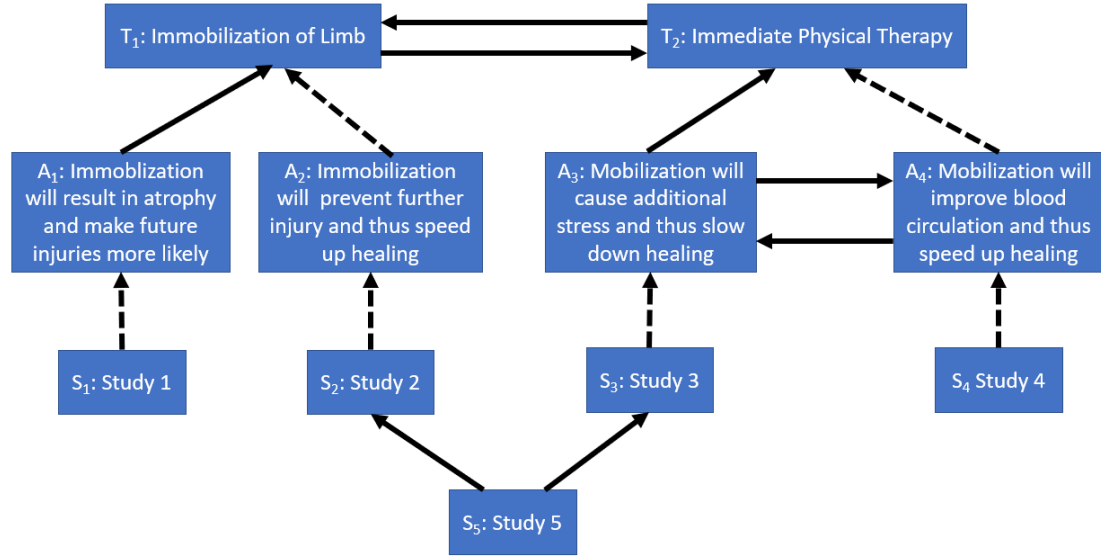
Figure 2: A simple medical decision support scenario.

The most general extension of probabilistic epistemic argumentation has been discussed in [15] recently. Constraints can now involve probabilities over arguments connected by logical connectives. This basic form corresponds to probabilities over propositional formulas (where arguments take the role of propositional atoms) like in Nilsson's probabilistic logic. These probability statements can be combined linearly and used in constraints with (in-)equality and strict inequality. The former basically corresponds to the basic constraints considered in probabilistic logic. Probabilistic formulas over arguments can also be connected via logical connectives analogous to the idea of taking the logical closure of probabilistic constraints. In this most general form, the framework of probabilistic epistemic argumentation is basically equivalent to probabilistic logic. A natural question is then, do we really need probabilistic epistemic argumentation? Personally, I would argue that the answer is yes. This is for the same reason that we need probabilistic logic, even though we have the more general probability theory. What Nilsson's probabilistic logic adds to probability theory is a formal language to conveniently describe events at a higher level of abstraction. Probabilistic epistemic argumentation can potentially add another layer of abstraction. Instead of having to define a knowledge base with probabilistic formulas, the user can build up an argumentation graph. This, of course, requires that building blocks in the graph can be associated with constraints automatically. It therefore seems worthwhile to invest more time in associating argumentation structure with more complex probabilistic logical constraints similar to what happened in the basic framework with atomic constraints like Coherence and Foundedness.

To illustrate the idea, let us consider the argumentation graph in Figure 2 from [31], where solid edges denote attack and dashed edges denote support relations between arguments. The scenario is that we want to decide between two alternatives for treating a muscle injury. Treatment T_1 involves immobilizing the limb for some weeks before starting physical therapy.

$\mathscr{A}$	$\emptyset$	$\mathscr{C}_0$	$\mathscr{C}_1$	$\mathscr{C}_2$	$\mathscr{C}_3$	$\mathscr{C}_4$	$\mathscr{C}_5$	$\mathscr{C}_6$
T_1	$[0,1]$	0.5	0	1	$[0,1]$	0	$[0,1]$	$[0,0.25]$
T_2	$[0,1]$	0.5	$[0,1]$	0	0	1	$[0,1]$	$[0.75,1]$
A_1	$[0,1]$	0.5	1	0	$[0,1]$	$[0,1]$	$[0,1]$	$[0.75,1]$
A_2	$[0,1]$	0.5	0	1	$[0,1]$	0	$[0,1]$	$[0,0.25]$
A_3	$[0,1]$	0.5	$[0,1]$	$[0,1]$	1	0	$[0,1]$	$[0,0.25]$
A_4	$[0,1]$	0.5	$[0,1]$	0	0	1	$[0,1]$	$[0.75,1]$
S_1	$[0,1]$	0.5	1	0	$[0,1]$	$[0,1]$	$[0,1]$	$[0.75,1]$
S_2	$[0,1]$	0.5	0	1	$[0,1]$	0	0	$[0,0.25]$
S_3	$[0,1]$	0.5	$[0,1]$	$[0,1]$	1	0	0	$[0,0.25]$
S_4	$[0,1]$	0.5	$[0,1]$	0	0	1	$[0,1]$	$[0.75,1]$
S_5	$[0,1]$	0.5	$[0,1]$	0	0	$[0,1]$	1	$[0.75,1]$

Table 2
Some entailment results for the BAF in Figure 2 using Coherence and S-Coherence for encoding the meaning of edges and additional constraints on the probabilities of arguments. The directly constrained arguments are highlighted in red.

Treatment T_2 starts physical therapy immediately. The arguments $A_1, \dots A_4$ state reasons for or against the two treatments. Arguments $S_1, \dots, S_5$ represent studies that, in turn, support or attack $A_1, \dots A_4$ or, in case of S_5, attack other studies. We use Coherence and S-Coherence to encode the meaning of attack and support edges. For the associated optimization problems, we basically add one linear constraint for every edge. Table 2 shows some probabilistic epistemic entailment results when adding additional constraints on the belief in arguments (shown in red in the respective columns). For example, in the second column, we assume that we are indifferent about the studies (probability 0.5). As a result, we are also indifferent about the treatment. In the fourth column, we assume that we fully accept the second study (probability 1). This enforces that we fully accept treatment T_1 and reject treatment T_2. The probabilities in this example have been computed with the Java library *Attractor*[2] that supports polynomial-time reasoning over linear atomic constraints.

Some ideas for more sophisticated general-purpose constraints and general design templates for probabilistic epistemic argumentation graphs can be found in [35].

3.4. Dealing with Inconsistencies

Given the close relationship between probabilistic logic and probabilistic epistemic argumentation, one can apply the same ideas to deal with inconsistencies. One inconsistency-tolerant version of the probabilistic epistemic entailment problem has been discussed in [30]. However, as opposed to the generalized probabilistic entailment problem that we discussed before, the formulation makes the optimization problems actually harder. It should be possible, however, to use the same ideas like in [26] to get an inconsistency-tolerant version that maintains the linear structure of the optimization problems. In particular, the complexity for this generalization should remain polynomial for linear atomic constraints. Investigating this direction further, is one interesting direction for future work in this area.

---

[2]https://sourceforge.net/projects/attractorproject/

4. Conclusions and Some Potential Research Directions

We discussed relationships between Nilsson's probabilistic logic and probabilistic epistemic argumentation. While both are closely related, there are two main differences worth highlighting. First, the atomic fragment that is not of considerable interest in the logical setting, is interesting from an abstract argumentation perspective and allows solving simple reasoning problems in a quite intuitive and computationally efficient way. Second, argumentation graphs allow modelling reasoning problems in a way that can be more intuitive for domain experts without logical background. Hence, while probabilistic logic adds a formal language to describe events to probability theory, probabilistic epistemic argumentation adds a graphical language to describe probabilistic reasoning problems. From a probabilistic programming perspective, one could say that probabilistic logic turns probability theory into a logical programming language while probabilistic epistemic argumentation turns it into a graphical programming language.

The question which research directions are most interesting is, of course, very subjective. In my opinion, the main difference between probabilistic logic and the more expressive (and computationally expensive) forms of probabilistic epistemic argumentation is that users can potentially model reasoning problems in a more intuitive graphical way. Therefore, one interesting direction for future research in probabilistic epistemic argumentation is to look deeper into ways how to automatically extract meaningful probabilistic constraints from argumentation graphs.

It seems indeed that the logical formalism of probabilistic epistemic argumentation has now become equivalent to extensions of Nilsson's probabilistic logic that have been studied before. Therefore, it may be worthwhile to take a step back and to unify the results in the two areas. The framework of linear probabilistic knowledge bases [10] has been used before to unify different probabilistic logical languages and can be extended easily to also capture probabilistic argumentation problems with linear constraints. This is more than a technical exercise because it allows implementing general-purpose reasoners that can be used and optimized for multiple formalisms simultaneously. For example, the Java library *Log4KR* that we mentioned before transforms knowledge bases in different propositional and relational dialects of probabilistic logic into linear constraints and solves them in a uniform way. By adding probabilistic epistemic argumentation to this framework, satisfiability tests, entailment algorithms and generalized (inconsistency-tolerant) entailment algorithms for probabilistic logic become immediately available for probabilistic epistemic argumentation.

We did not have space to talk much about relational probabilistic logics, but they are, of course, a very interesting topic in their own right. The main challenge when using Herbrand semantics is that the number of ground atoms grows rapidly. Lifted inference approaches [36] try exploiting symmetries to control the blowup. While some similar ideas have been considered for probabilistic logics [37, 38], it seems worthwhile revisiting the state-of-the-art and to evaluate which recent ideas can be transferred to probabilistic logic. It seems also interesting to investigate to which extent relational probabilistic logics can be applied to add expressiveness to probabilistic epistemic argumentation. For example, one could associate arguments with constants, consider a unary predicate *accept* that expresses acceptance of an argument and binary predicates like *attack* and *support* for edges to allow both uncertainty about arguments and relationships between them.

References

[1] N. Goodman, V. Mansinghka, D. M. Roy, K. Bonawitz, J. B. Tenenbaum, Church: a language for generative models, in: Conference on Uncertainty in Artificial Intelligence (UAI), 2008, pp. 220–229.

[2] S. Vajda, Probabilistic programming, Academic Press, 1972.

[3] N. J. Nilsson, Probabilistic logic, Artificial intelligence 28 (1986) 71–87.

[4] L. De Raedt, A. Kimmig, H. Toivonen, Problog: A probabilistic prolog and its application in link discovery., in: International Joint Conference on Artificial Intelligence (IJCAI), 2007, pp. 2462–2467.

[5] T. Hailperin, Boole's logic and probability: a critical exposition from the standpoint of contemporary algebra, logic and probability theory, Elsevier, 1986.

[6] T. Lukasiewicz, Probabilistic logic programming., in: European Conference on Artificial Intelligence (ECAI), 1998, pp. 388–392.

[7] J. Fisseler, First-order probabilistic conditional logic and maximum entropy, Logic Journal of IGPL 20 (2012) 796–830.

[8] G. Kern-Isberner, M. Thimm, Novel semantical approaches to relational probabilistic conditionals, in: International Conference on the Principles of Knowledge Representation and Reasoning (KR), 2010.

[9] D. Koller, N. Friedman, S. Džeroski, C. Sutton, A. McCallum, A. Pfeffer, P. Abbeel, M.-F. Wong, C. Meek, J. Neville, et al., Introduction to statistical relational learning, MIT press, 2007.

[10] N. Potyka, M. Thimm, Inconsistency-tolerant reasoning over linear probabilistic knowledge bases, International Journal of Approximate Reasoning 88 (2017) 209–236.

[11] P. Besnard, A. Garcia, A. Hunter, S. Modgil, H. Prakken, G. Simari, F. Toni, Introduction to structured argumentation, Argument & Computation 5 (2014) 1–4.

[12] P. M. Dung, On the acceptability of arguments and its fundamental role in nonmonotonic reasoning and logic programming., in: International Joint Conference on Artificial Intelligence (IJCAI), volume 93, 1993, pp. 852–857.

[13] A. Hunter, S. Polberg, N. Potyka, T. Rienstra, M. Thimm, Probabilistic argumentation: A survey, in: D. Gabbay, M. Giacomin, G. R. Simari, M. Thimm (Eds.), Handbook of Formal Argumentation, volume 2, College Publications, 2021.

[14] A. Hunter, M. Thimm, Probabilistic reasoning with abstract argumentation frameworks, Journal of Artificial Intelligence Research 59 (2017) 565–611.

[15] A. Hunter, S. Polberg, M. Thimm, Epistemic graphs for representing and reasoning with positive and negative influences of arguments, Artificial Intelligence 281 (2020) 103236.

[16] G. Georgakopoulos, D. Kavvadias, C. H. Papadimitriou, Probabilistic satisfiability, Journal of complexity 4 (1988) 1–11.

[17] P. Hansen, B. Jaumard, Probabilistic satisfiability, in: Handbook of Defeasible Reasoning and Uncertainty Management Systems, Springer, 2000, pp. 321–367.

[18] N. Potyka, Relationships between semantics for relational probabilistic conditional logics, Computational Models of Rationality (2016) 332–347.

[19] A. Charnes, W. W. Cooper, Programming with linear fractional functionals, Naval Research logistics quarterly 9 (1962) 181–186.

[20] N. Potyka, Solving reasoning problems for probabilistic conditional logics with consistent and inconsistent information, Ph.D. thesis, FernUniversität in Hagen, 2016.

[21] G. De Bona, F. G. Cozman, M. Finger, Towards classifying propositional probabilistic logics, Journal of Applied Logic 12 (2014) 349–368.

[22] M. Finthammer, G. Kern-Isberner, M. Ritterskamp, Resolving inconsistencies in probabilistic knowledge bases, in: German Conference on Artificial Intelligence (KI), Springer, 2007, pp. 114–128.

[23] D. Picado-Muiño, Measuring and Repairing Inconsistency in Probabilistic Knowledge Bases, International Journal of Approximate Reasoning (2011).

[24] M. Thimm, Inconsistency measures for probabilistic logics, Artificial Intelligence 197 (2013) 1–24.

[25] L. Daniel, Paraconsistent probabilistic reasoning: applied to scenario recognition and voting theory, Ph.D. thesis, École Nationale Supérieure des Mines de Paris, 2010.

[26] N. Potyka, M. Thimm, Probabilistic reasoning with inconsistent beliefs using inconsistency measures, in: Twenty-Fourth International Joint Conference on Artificial Intelligence, 2015.

[27] N. Potyka, Linear programs for measuring inconsistency in probabilistic logics, in: International Conference on the Principles of Knowledge Representation and Reasoning (KR), 2014.

[28] N. Potyka, Reasoning over linear probabilistic knowledge bases with priorities, in: International Joint Conference on Artificial Intelligence (IJCAI), Springer, 2015, pp. 121–136.

[29] P. Baroni, M. Caminada, M. Giacomin, An introduction to argumentation semantics, The knowledge engineering review 26 (2011) 365–410.

[30] A. Hunter, M. Thimm, On partial information and contradictions in probabilistic abstract argumentation, in: International Conference on the Principles of Knowledge Representation and Reasoning (KR), 2016.

[31] N. Potyka, A polynomial-time fragment of epistemic probabilistic argumentation, International Journal of Approximate Reasoning 115 (2019) 265–289.

[32] L. Amgoud, C. Cayrol, M.-C. Lagasquie-Schiex, P. Livet, On bipolarity in argumentation frameworks, International Journal of Intelligent Systems 23 (2008) 1062–1093.

[33] N. Oren, T. J. Norman, Semantics for evidence-based argumentation, in: International Conference on Computational Models of Argument (COMMA), IOS Press, 2008, pp. 276–284.

[34] N. Potyka, Bipolar Abstract Argumentation with Dual Attacks and Supports, in: International Conference on Principles of Knowledge Representation and Reasoning (KR), 2020, pp. 677–686.

[35] I. Ibs, N. Potyka, Explainable automated reasoning in law using probabilistic epistemic argumentation, in: Workshop on Models of Legal Reasoning (MLR), 2020.

[36] K. Kersting, Lifted probabilistic inference., in: European Conference on Artificial Intelligence (ECAI), 2012, pp. 33–38.

[37] M. Thimm, Probabilistic reasoning with incomplete and inconsistent beliefs, AKA, 2012.

[38] C. Beierle, N. Potyka, J. Baudisch, M. Finthammer, Towards lifted inference under maximum entropy for probabilistic relational fo-pcl knowledge bases, in: European Conference on Symbolic and Quantitative Approaches to Reasoning and Uncertainty (ECSQARU), Springer, 2015, pp. 506–516.