ECAI 2014
T. Schaub et al. (Eds.)

doi:10.3233/978-1-61499-419-0-621

Imprecise Probabilistic Horn Clause Logic[1]

Steffen Michels and **Arjen Hommersom** and **Peter J.F. Lucas** and **Marina Velikova**[2]

Abstract. Approaches for extending logic to deal with uncertainty immanent to many real-world problems are often on the one side purely qualitative, such as *modal logics*, or on the other side quantitative, such as *probabilistic logics*. Research on combinations of qualitative and quantitative extensions to logic which put qualitative constraints on probability distributions, has mainly remained theoretical until now. In this paper, we propose a practically useful logic, which supports qualitative as well as quantitative uncertainty and can be extended with modalities with varying level of quantitative precision. This language has a solid semantic foundation based on *imprecise probability theory*. While in general imprecise probabilistic inference is much harder than the precise case, this is the first expressive imprecise probabilistic formalism for which probabilistic inference is shown to be as hard as corresponding precise probabilistic problems. A second contribution of this paper is an inference algorithm for this language based on the translation to a *weighted model counting* (WMC) problem, an approach also taken by state-of-the-art probabilistic inference methods for precise problems.

1 INTRODUCTION

The use of knowledge representation and reasoning methods to cope with the uncertainty that comes with real-world problems has become a crucial element in the development of intelligent systems. The way in which this uncertainty presents itself varies from problem to problem; in some cases precise probabilistic information is available, whereas in other cases the best that can be achieved is qualitative uncertainty.

Probabilistic logics, which use the structure of a logic theory to define a probability distributions, are the state-of-the-art for knowledge representation when precise probabilistic information is available. *Alethic modal logics*, which extend logic with purely qualitative modalities, such as that something is *possible*, allow dealing with uncertainty that is merely qualitative in nature. Alternatively, one can handle lack of precise probabilistic information by putting constraints on a probability distribution, as in the *probabilistic belief logic* of Bacchus [1].

In this paper we propose a new language that integrates these different approaches. It supports qualitative as well as quantitative uncertainty and can also be extended with modalities with varying level of quantitative precision. This is achieved by the well-defined semantic basis of *imprecise probability theory* [18], specifically interval probabilities. This semantics allows to exactly define the semantics of qualitative modalities in terms of probability intervals and also provides precise probabilistic logic as a special case. The language is designed with particular guarantees for computational tractability in mind.

In general, computing marginal probabilities of imprecise probability distributions is more complex than inference for their precise counterparts. For instance, it is known that inference in *credal networks* [5] is much harder ($\mathrm{NP}^{\mathrm{PP}}$-hard) than for the corresponding precise case of *Bayesian networks* [15]. However, as we claim our language to be practically useful, we designed an expressive language in such a way that inference has the same complexity as corresponding precise probabilistic inference problems, making it the first imprecise probabilistic language that offers such guarantees. This is the best we can hope for, as the language supports precise probabilities as a special case. While inference is still NP-hard in general, one can often make use of the problem's structure to perform more efficient inference. For example, it is well-known that inference is linear for problems corresponding to singly-connected Bayesian networks with a bounded indegree.

We further propose a concrete inference mechanism based on the translation to a *weighted model counting* (WMC) problem, an approach also taken by state-of-the-art probabilistic inference methods for precise problems. The approach has successfully been used in the context of probabilistic logic programming [7] and can exploit local structure, such as *determinism*, which is often present in logic theories. We have already shown in previous work that WMC can be used to compute marginals for certain classes of credal sets, which we used to approximate continuous distributions [11].

The paper is structured as follows. We first provide some background in Section 2. Then the language is defined formally (Section 3) and the inference approach is discussed (Section 4). Finally, related work is discussed in Section 5 and Section 6 concludes the paper.

2 BACKGROUND

We first give some background and focus on properties and limitations of existing languages.

2.1 Logic programming

As the work described in this paper builds upon probabilistic logic programming, we will first introduce some basic logic programming (LP) concepts. The idea of LP is to use predicate logic as programming language [10], with programs consisting of *rules*, in this paper *Horn clauses*. They are (implicitly universally quantified) expressions of the form: $h \leftarrow b_1, \ldots, b_n$, where h is called the *head* and $b_1, \ldots, b_n$ is called the *body* of the rule, representing a conjunction. The head h and elements of the body b_i, $1 \leq i \leq n$ are *atoms*, i.e. expressions of the form $p(t_1, \ldots, t_m)$ with p a *predicate*, and $t_1, \ldots, t_n$ are terms.

[1] This publication was supported by the Dutch national program COMMIT. The research work was carried out as part of the Metis project under the responsibility of the TNO-Embedded Systems Innovation, with Thales Nederland B.V. as the carrying industrial partner.
[2] Radboud University Nijmegen, The Netherlands
email: {s.michels,arjenh,peterl,marinav}@science.ru.nl

In the remainder of this paper, we assume the traditional least model semantics of LP. This semantics implies the closed world assumption (CWA), which states that statements that do not follow from the rules are false.

Example 1. *To illustrate the different formalisms and their properties, we make use of the following running example originating from the maritime safety and security domain, in which we already successfully applied precise probabilistic logics [12].*

Suppose we want to model in which cases a vessel is an environmental hazard and may for instance not enter certain restricted areas. One reason for a vessel being an environmental hazard is that it has some chemical substances loaded. We could model this using LP as:

$$env\_hazard \leftarrow chemicals$$

The problem is that the model actually expresses that in case we know the vessel has chemicals loaded it certainly is an environmental hazard and otherwise it is certainly not, using the CWA. Clearly, there are vessels with chemicals which are no environmental hazard, for instance because the amount is not significant, and ships without chemicals on board which still are an environmental hazard.

2.2 Modal logic

The idea of modal logics is to lift the restrictions that propositional statements are certainly true or false, by including operators that express modalities. Several practical implementations of programming languages based on modal logic are available [14]. There are different ways to define and interpret those modalities, but we restrict to classical *alethic* modalities, which express that something is possibly ($\Diamond$) or necessarily true ($\Box$).

Example 2. *Example 1 can be made more precise using modal operators. For example, we could model the fact that having chemicals loaded makes it possible that the vessel is an environmental hazard with:*

$$\Diamond env\_hazard \leftarrow chemicals$$

However, using the CWA, this rule alone implies that if the vessel has no chemicals on board, then it is not an environmental hazard. While we could try to sum up all the reasons for a vessel being an environmental hazard, it is a reasonable assumption that in reality we can never observe or even know all of those reasons. A possible solution is adding the rule which states that it is always possibly true that a vessel is an environmental hazard:

$$\Diamond env\_hazard \leftarrow \top$$

where $\top$ *denotes* true. *This is not useful in practice, since these rules imply that* env_hazard *is possible whether or not the vessel is carrying chemicals. The knowledge that chemicals are a risk factor, increasing the likelihood of environmental hazard, cannot be expressed in the language.*

Generally, modal logics can only be used to represent and reason about qualitative uncertainty, whereas the available quantitative knowledge cannot be used.

2.3 Probabilistic logic programming

Probability theory offers an alternative widely used and well-founded basis for representing and reasoning with uncertainty. Combining logic with probability theory is a subject gaining increasing interest and various approaches and their implementations have been developed. We focus on logic programming extended with probabilities associated to atoms or rules, for which efficient inference mechanism are available (see e.g. [7]).

Example 3. *To include degrees of uncertainty, we extend Example 2 with probabilities as follows:*

$$0.1\colon env\_hazard \leftarrow \top$$
$$0.4\colon env\_hazard \leftarrow chemicals$$

This states that the fact that if a vessel has chemicals loaded, then it will cause an environmental hazard with a probability of 0.4. *The first rule represents other causes we do not model explicitly with a low probability.*

This basic language serves as a basis for the work described in this paper. The semantics of this language is a variant of Sato's *distribution semantics* [16], which we introduce next.

First, if $\mathbf{R}$ is a set of probability-rule pairs, then to each subset $\mathbf{S} \subseteq \mathbf{R}$, a probability is assigned:

$$P_{\mathbf{S}} \stackrel{\text{def}}{=} \prod_{p\colon (h\leftarrow b_1,\ldots,b_n)\in\mathbf{S}} p \prod_{p\colon (h\leftarrow b_1,\ldots,b_n)\in\mathbf{R}\setminus\mathbf{S}} 1-p \qquad (1)$$

In this paper, we will assume that there are a finite number of ground terms, which means that we can look upon each program as a propositional one by replacing all variables by all ground terms. As a result, there will be a finite number of subsets; however, the approach can be easily generalised to the full first-order case with an infinite number of constants, as shown in the original distribution semantics [16].

For each such subset we can determine whether a query q holds ($\mathbf{S} \models q$) under the least model semantics of logic programming. Then, the probability of a query q given the rules of a program $\mathbf{R}$ is defined as the sum of the probabilities of all rule subsets for which the query can be derived:

$$P_{\mathbf{R}}(q) \stackrel{\text{def}}{=} \sum_{\substack{\mathbf{S}\models q \\ \mathbf{S}\subseteq\mathbf{R}}} P_{\mathbf{S}} \qquad (2)$$

Example 4. *The query* env_hazard *can be derived for the following subsets of rules in Example 3 assuming* $chemicals$ *is true:*

$$\mathbf{S}_1 = \{\ 0.1\colon env\_hazard \leftarrow \top\}$$
$$\mathbf{S}_2 = \{\ 0.4\colon env\_hazard \leftarrow chemicals\ \}$$
$$\mathbf{S}_3 = \{\ 0.1\colon env\_hazard \leftarrow \top,\ 0.4\colon env\_hazard \leftarrow chemicals\ \}$$

We have the following probabilities: $P_{\mathbf{S}_1} = 0.1 \cdot (1-0.4) = 0.06$, $P_{\mathbf{S}_2} = 0.36$ *and* $P_{\mathbf{S}_3} = 0.04$. *Therefore* $P_{\mathbf{R}}(env\_hazard) = 0.06 + 0.36 + 0.04 = 0.46$.

A limitation of such probabilistic approaches is that they require the precise quantification of likelihoods. This is often infeasible, for instance in domains dealing with very rare events. For instance, estimations of the probability that a vessel without chemicals on board is an environmental hazard are unreliable, as there are few of such cases. The consequence is that predictions suggest more precision than can be provided by the knowledge available, which may lead to wrong decisions.

2.4 Imprecise probabilities

Imprecise probability theory is a generalisation of probability theory, which allows to express different levels of ignorance regarding the likelihoods of events. There are different approaches with varying expressiveness [18]. In this paper we follow the approach of using sets of probability distributions, called *credal sets*. In the remainder, we only deal with binary logic statements and convex credal sets, which makes it possible to denote and define credal sets in terms of probability intervals. For instance, this allows to express that the probability that a ship is an environmental hazard is in the interval $[0.3, 0.6]$. In this way, we can differentiate between cases in which the best decision can be made based on the available knowledge and cases more knowledge is required to take the optimal decision.

3 IMPRECISE PROBABILISTIC HORN CLAUSE LOGIC

We introduce the idea of the language and then formally develop its semantics.

3.1 Basic idea

The basic idea of the language is to extend logic programming with probability interval annotations, such as probabilistic logic programming extends logic programming with point probabilities. Our language supports two possible interpretations of rules as $\mathbf{p}\colon h \leftarrow b_1, \ldots, b_n$ with $\mathbf{p}$ a probability interval, which can be closed, open, or half-closed. We refer to those different interpretations as different kinds of *imprecisions*, *rule-imprecisions* and *head-imprecisions*, which come into play when there are multiple rules with the same head. Formally, we say an imprecise probabilistic horn clause logic (IPHL) program $\mathbf{P} = (\mathbf{R}_R, \mathbf{R}_H)$, where $\mathbf{R}_R$ models rule-imprecisions and $\mathbf{R}_H$ models head-imprecisions. The heads occurring in $\mathbf{R}_R$ and $\mathbf{R}_H$ are disjoint.

Rules in $\mathbf{R}_R$ are denoted by $\mathbf{p}\colon (h \leftarrow b_1, \ldots, b_n)$ with $\mathbf{p}$ a probability interval. The interpretation of these rules is that the probability that $b_1, \ldots, b_n$ leads to h is in $\mathbf{p}$, which corresponds to the semantics of precise probabilistic programming as given in Section 2.3. Multiple rules with the same head are combined using a noisy-OR operator for probability intervals.

Example 5. *Consider an imprecise version of Example 3:*

$$[0.05, 0.15]\colon (env\_hazard \leftarrow \top)$$
$$[0.4, 0.6]\colon (env\_hazard \leftarrow chemicals)$$

This means that carrying chemicals causes a vessel to be an environmental hazard with probability between 0.4 *and* 0.6*, while it is unlikely that other reasons cause a ship to be an environmental hazard.*

In case a vessel has no chemicals on board the probability of it being an environmental hazard is between 0.05 *and* 0.15*. Otherwise, we consider the probabilities one gets for all possible choices of probabilities from the intervals given the semantics of point probabilities (Section 2.3). These probabilities are within the interval* $[0.43, 0.66]$.

Rules in $\mathbf{R}_H$ are denoted by $(\mathbf{p}\colon h) \leftarrow b_1, \ldots, b_n$. As indicated by the brackets, in this case, the probability interval only applies to the head. The rules are interpreted as follows: in case $b_1, \ldots, b_n$ holds the probability of h is in $\mathbf{p}$. This means that rules do not represent independent causes for the head, but conditions under which the knowledge about the head's probability becomes more precise. If no rule applies, there is complete ignorance about the head's probability, i.e. it is in $[0.0, 1.0]$. While this interpretation makes no sense for the precise case, it can conveniently express certain kinds of imprecise knowledge. Note that this kind of rules can lead to inconsistent definitions, while the former kind cannot.

Example 6. *Suppose we have statistical knowledge about tankers, for example because all tankers have to register their cargo due to safety regulations, and can estimate the likelihood of tankers having chemicals loaded as* 0.3*. For a random vessel, we do not have that information, for instance because we do not have data about all ships, and only express that it is possibly carrying chemicals, we interpret as the probability interval* $(0.0, 1.0]$*, i.e. the probability is greater* 0.0*. We can model this with head-imprecisions:*

$$\big((0.0, 1.0]\colon chemicals\big) \leftarrow \top$$
$$\big([0.3, 0.3]\colon chemicals\big) \leftarrow tanker$$

Given the rules above, if we do not know the ship is a tanker, its probability of having chemicals on board is in $(0.0, 1.0]$*. In the case it is a tanker it is in* $(0.0, 1.0]$ *and in* $[0.3, 0.3]$*, which means it is in* $(0.0, 1.0] \cap [0.3, 0.3] = [0.3, 0.3]$.

3.2 Qualitative interpretation

Given the basic language defined above, we can give various intervals a qualitative interpretation. For example, special cases of intervals include determinism ($[0.0, 0.0]$ and $[1.0, 1.0]$), complete ignorance as in 3-valued logics ($[0.0, 1.0]$) and precise probabilities ($[p, p]$ with p some probability).

IPHL can also be seen as a basic language to define various modalities in terms of probability intervals. For example, notice that the interval $[1.0, 1.0]$ corresponds to the modality that something is necessarily true: $\Box$. Furthermore, the modality $\Diamond$ expressing that some statement is possible, in its least strict interpretation, means that the probability is greater than 0.0, i.e. it is in the interval $(0.0, 1.0]$. Analogously one could introduce the modality $\Diamond\neg$ as the interval $[0.0, 1.0)$. One could however more carefully only consider statements possible in case their probability is above a certain threshold and define for instance $\Diamond_{0.05}$ as $(0.05, 1.0]$. One could also introduce linguistic modalities, for instance $unlikely$ as $[0.05, 0.15]$. Note that in order to differentiate between complete ignorance ($[0.0, 1.0]$) and possibility ($(0.0, 1.0]$) the distinction between open and closed intervals is necessary, which cannot be made in common imprecise formalisms as *credal networks* [5].

Similar to a qualitative specification of the IPHL program, marginal probability intervals of arbitrary atoms can also be given a qualitative interpretation, which implies that the language supports qualitative reasoning as a special case. For instance, a probability interval of $[1.0, 1.0]$ means a particular statement is necessarily true and a probability greater 0.0 implies that the statement is possible. Furthermore, given the lowerbound of a probability interval, we may conclude whether or not the probability is possibly or necessarily larger than a given threshold. Finally, one can also qualitatively compare the likelihood of two statements, for instance in case the probability intervals of two statements are disjoint, one can determine which one is more likely than the other.

3.3 Semantics

In accordance with probabilistic logics programming, the semantics of IPHL programs is defined in terms of the marginal probability

intervals of arbitrary query atoms. This semantics is defined incrementally, by first extending the semantics for precise logic programs given in Section 2.3 for rules with rule-imprecision. Then, we also show how to deal with head-imprecisions.

3.3.1 Rules with rule-imprecision

We first develop a semantics assuming the program only consists of rules with rule-imprecision. The semantics of rules with rule-imprecision is defined in terms of a set of programs obtained by replacing the intervals with point probabilities. Let $\mathcal{R}$ be the set of all programs where each rule $\mathbf{p}\colon (h \leftarrow b_1, \ldots, b_n)$ from $\mathbf{R}_R$ is replaced by $p\colon (h \leftarrow b_1, \ldots, b_n)$ such that $p \in \mathbf{p}$. In other words, we consider all programs for all possible choices of probabilities for each interval.

Example 7. *The precise program of Example 3 is one example of an element of $\mathcal{R}$ given the imprecise program of Example 5.*

We can then define the probability range of a query q given a program as:

$$P(q) \stackrel{\text{def}}{\in} \{P_{\mathbf{R}}(q) \mid \mathbf{R} \in \mathcal{R}\} \tag{3}$$

where $P_{\mathbf{R}}(q)$ is defined as in (2). This set of probabilities is convex and can therefore be expressed as interval.

Example 8. *For Example 5 we get $P(env\_hazard) \in [0.05, 0.15]$, since there is no rule with head $chemicals$ its probability is 0.0 and the second rule never applies. In case we assume we know the vessel is carrying chemical and add $[1.0, 1.0]\colon (chemicals \leftarrow \top)$ to the rules, the probability is in $[0.43, 0.66]$.*

3.3.2 Rules with head-imprecision

Next, we extend the semantics with head-imprecisions. Probability intervals on heads have a different characteristic as probability intervals on rules. Head-imprecisions can be looked upon as constraints that exclude programs for which the probability distribution does not obey the specified bounds. Therefore, we first generate rules allowing for all possible probabilities and then enforce the constraints on the set of programs.

Example 9. *As example we use the following program, which is a combination of the rules of Examples 5 and 6:*

$$\begin{aligned}
\mathbf{R}_R = \{ \quad & [0.05, 0.15]\colon (env_hazard \leftarrow \top) \\
& [0.4\ , 0.6\ ]\colon (env_hazard \leftarrow chemicals) \quad \} \\
\mathbf{R}_H = \{ \quad & \big((0.0\ , 1.0\ ]\colon chemicals\big) \leftarrow \top \\
& \big([0.3\ , 0.3\ ]\colon chemicals\big) \leftarrow tanker \quad \}
\end{aligned}$$

To allow all possible probabilities for rules with head-imprecision, we add for each head h occurring in the rules $\mathbf{R}_H$ a rule $[0.0, 1.0]\colon (h \leftarrow \top)$ to $\mathbf{R}_R$ and call the resulting set of rules $\mathbf{R}'_R$.

Example 10. *For the above example we get the following transformed set of rules:*

$$\begin{aligned}
\mathbf{R}'_R = \{ \quad & [0.05, 0.15]\colon (env_hazard \leftarrow \top) \\
& (0.4\ , 0.6\ ]\colon (env_hazard \leftarrow chemicals) \\
& [0.0\ , 1.0\ ]\colon (chemicals \leftarrow \top) \quad \}
\end{aligned}$$

We can now consider a set of programs $\mathcal{R}$ with point probabilities generated by $\mathbf{R}'_R$, under the semantics of rule-imprecision. To incorporate the constraints given by the probability intervals on heads, we define a set of programs $\mathcal{R}'$, by including only those programs obeying all constraints given by $\mathbf{R}_H$:

$$\mathcal{R}' \stackrel{\text{def}}{=} \{\mathbf{R} \in \mathcal{R} \mid \forall R \in \mathbf{R}_H\colon obeys(\mathbf{R}, R)\} \tag{4}$$

where $obeys\big(\mathbf{R}, (\mathbf{p}\colon h) \leftarrow b_1, \ldots, b_n\big) = P_{\mathbf{R}}(h \mid b_1, \ldots, b_n) \in \mathbf{p}$.

Example 11. *Consider the rules with head-imprecision of Example 9. The restricted set of rules $\mathcal{R}'$ given those constraints is:*

$$\begin{aligned}
\mathcal{R}' = \{\mathbf{R} \in \mathcal{R} \mid & P_{\mathbf{R}}(chemicals \mid tanker) = 0.3 \wedge \\
& 0 < P_{\mathbf{R}}(chemicals) \leq 1\}
\end{aligned}$$

In case $\mathcal{R}'$ becomes the empty set, the entire program is called *inconsistent*. Otherwise, the probability of a query q is defined as in Eq. (3) using $\mathcal{R}'$ instead of $\mathcal{R}$.

4 INFERENCE

In this section, an inference mechanism is introduced which translates the problem of computing a bound to a precise probabilistic inference problem without changing the complexity of the problem. We only deal with exact inference, as approximate inference could change the qualitative nature of the answer. For instance, there is a qualitative difference of a probability greater zero and greater or equal to zero, which cannot be made by sampling algorithms. Similarly to the semantics, we introduce the inference approach incrementally, starting with point probabilities. Therefore, first, we briefly discuss probabilistic inference by WMC.

4.1 Probabilistic Inference by Weighted Model Counting

Various inference approaches have been proposed to exploit the structure of probabilistic inference problems. We focus on performing probabilistic inference by translation to a WMC problem, which has been shown to be an efficient inference method for probabilistic logic programming [7]. In contrast to other approaches, WMC not only exploits topological structure, but also local structure as *determinism* and *context-specific independence*.

The problem of model counting is to find the number of models of a propositional knowledge base. WMC is a straightforward generalisation of the problem, where each model has a weight. Those weights are defined in terms of weights attached to the literals. The weight of a model is the product of the weights of all literals included. In the following we denote the weight of a literal l with $W(l)$ and the weighted model count of a weighted knowledge base Δ with $WMC(\Delta)$.

4.2 Rules with point probabilities

WMC is defined for propositional knowledge bases without the CWA assumption. We therefore translate the rules to propositional logic and make the heads equivalent to the combination of all rules defining it to capture the CWA. This is known as *Clark's completion* [4] and a similar approach for probabilistic inference it taken by *Problog* [7].

Probabilities are added by introducing auxiliary atoms for each rule. So in the first step each rule $R = p\colon (h \leftarrow b_1, \ldots, b_n)$ is

Algorithm 1: Imprecise inference (lower bound)

Input: Query q and IPHL program $(\mathbf{R}_R, \mathbf{R}_H)$
Result: The lower probability bound of q

1 $\mathbf{R} = \emptyset$
2 **for** $\big(\mathbf{p}\colon (h \leftarrow b_1, \ldots, b_n)\big) \in \mathbf{R}_R$
3 $\quad \mathbf{R} \leftarrow \big(lower(\mathbf{p})\colon (h \leftarrow b_1, \ldots, b_n)\big)$
4 **for** *all heads* h *in* $\mathbf{R}_H$
5 $\quad$ **for** $\{b_1, \ldots, b_n\} \subseteq \mathbf{B}$*, with* $\mathbf{B}$ *all body-atoms defining* h
6 $\quad\quad \mathbf{R} \leftarrow \big(subset\_lower_h(b_1, \ldots, b_n)\colon (h \leftarrow b_1, \ldots, b_n)\big)$
7 **return** $WMC(\Delta_{\mathbf{R}} \wedge q)$

translated to $h \leftarrow aux_R, b_1, \ldots, b_n$, before *Clark's completion* is performed. The weight p is assigned to aux_R and $1 - p$ to its negation. All other literals get weight 1. We denote the resulting weighted knowledge base with $\Delta_{\mathbf{R}}$. The probability can then be computed as $P_{\mathbf{R}}(q) = WMC(\Delta_{\mathbf{R}} \wedge q)$.

Example 12. *Consider the rules of Example 3 and the assumption the vessel is carrying chemicals (*$1.0\colon chemicals \leftarrow \top$*). The resulting weighted knowledge base* $\Delta_{\mathbf{R}}$ *is:*

$$\big(env\_hazard \leftrightarrow aux_1 \vee (aux_2 \wedge chemicals)\big) \wedge chemicals$$

Note that for brevity we added $chemicals$ *as fact instead of making it equivalent to an auxiliary literal with weight* 1.0*. The weights are:* $W(aux_1) = 0.1$*,* $W(\neg aux_1) = 0.9$*,* $W(aux_2) = 0.4$*,* $W(\neg aux_2) = 0.6$*,* $W(env\_hazard) = W(\neg env\_hazard) = 1.0$*,* $W(chemicals) = 1.0$ *and* $W(\neg chemicals) = 0.0$*.*

To compute the probability of env_hazard*, we consider the models of the knowledge base in which* env_hazard *holds, with corresponding weights:*

$$0.04\colon env\_hazard,\ aux_1,\ aux_2, chemicals$$
$$0.06\colon env\_hazard,\ aux_1, \neg aux_2, chemicals$$
$$0.36\colon env\_hazard, \neg aux_1,\ aux_2, chemicals$$

The sum of those weights is 0.46*, which corresponds to the probability according to the semantic definition as illustrated in Example 4.*

4.3 Imprecise inference

Given the fact that we only make use of intervals, the set of probabilities of a query is always convex. We can therefore represent the semantically infinite set of programs $\mathcal{R}$ by its extreme points. We denote the lower and upper bounds of the result probability as $\underline{P}(q)$ and $\overline{P}(q)$ respectively. The basic idea of the inference algorithm is to translate the problem to a precise probabilistic inference problem, for both bounds. An algorithm for the lower bound is given in Algorithm 1.

The fact that we use horn clauses, thus bodies consist of positive atoms only, makes it possible to locally determine the extreme points of each rule independent of the query. In fact, taking the minimum or maximum probability for all rules determines the minimal or maximal probability for all possible queries, respectively. This is the key insight which makes the inference problem tractable: without the restriction to horn clauses, the combination of all rules' extreme points would have to be considered. This requires an exponential number of precise probabilistic programs which heavily increases the complexity of the inference problem. Therefore, for the rules with rule-imprecision the translation is straightforward: for each rule we just use its lower bound probability (Lines 2, 3).

In order to represent open as well as closed intervals, we make use of a calculus for *hyperreal numbers* [8]. For instance, the interval $(0.2, 0.7)$ can be represented by the extreme points $0.2 + \mathrm{d}$ and $0.7 - \mathrm{d}$, where d represents an infinitesimal number. WMC can straightforwardly be extended with hyperreal weights, by making use of addition and multiplication as defined for the hyperreal calculus. In Algorithm 1, the function $lower(\mathbf{p})$, is defined as $\min \mathbf{p}$ in case the lower bound is closed and $\sup \mathbf{p} + \mathrm{d}$ otherwise.

Example 13. *Consider the rules of Example 5. For each query the lower bound of the probability is the probability given the following program:*

$$0.05\colon (env\_hazard \leftarrow \top)$$
$$0.4\ \colon (env\_hazard \leftarrow chemicals)$$

Theorem 1. *For any IPHL program* $\mathbf{P} = (\mathbf{R}_R, \varnothing)$*, the computational complexity of computing any query is the same as for a probabilistic logic program consisting of* $\mathbf{R}_R$ *where all intervals are replaced by point probabilities.*

The theorem obviously holds as replacing imprecisions with point probabilities is a simple linear transformation. In practice, inference could even be less expensive if the extremes of some intervals are 0.0 and 1.0, introducing additional determinism, which can be exploited by WMC algorithms.

The translation of rules with head-imprecision is more involved (Lines 4 – 6). For each head h (Line 4) we have to consider all cases of truth assignments to the atoms in the bodies of the rules defining h (Line 5). Each of those cases corresponds to a subset $\mathbf{B}$ of those atoms. We can define the probability interval in which h must be in that case by the intersection of all intervals of rules which apply given the set of atoms:

$$\mathbf{P}_h(\mathbf{B}) = \bigcap_{\substack{\{b_1,\ldots,b_n\} \subseteq \mathbf{B} \\ (\mathbf{p}\colon h) \leftarrow b_1,\ldots,b_n \in \mathbf{R}_h}} \mathbf{p} \tag{5}$$

where $\mathbf{R}_h$ are all rules with head h. $\mathbf{P}_h(\mathbf{B})$ is never the empty set for consistent programs.

Naively translating to rules with all possible $\mathbf{B}$ as body and using the lower bounds of such intervals, i.e. $lower(\mathbf{P}_h(\mathbf{B}))\colon (h \leftarrow \mathbf{B})$, results in incorrect probabilities, since all rules with a subset of $\mathbf{B}$ as body also contribute to the probability of h in case $\mathbf{B}$ holds. To solve that problem we make use of the property that with increasing cardinality of $\mathbf{B}$, the number of satisfied bodies also increases. Therefore with increasing cardinality of $\mathbf{B}$ the probability is restricted to a more tight interval, as it is restricted to the intersection of all such rules' intervals. This implies that the lower bound monotonically increases with the cardinality of $\mathbf{B}$.

For the correct transformation of the probabilities we use in Line 6 of the algorithm the function $subset\_lower$, which computes the correct probabilities for each subset $\mathbf{B}$. The idea is to consider the probability already given by the rules corresponding to proper subsets of $\mathbf{B}$ and only add as much probability mass as is needed to get the desired lower bound of the probability $\mathbf{P}_h(\mathbf{B})$:

$$subset\_lower_h(\mathbf{B}) = 1 - \frac{1 - lower(\mathbf{P}_h(\mathbf{B}))}{\prod_{\mathbf{B}' \subset \mathbf{B}} 1 - subset\_lower_h(\mathbf{B}')} \tag{6}$$

where the denominator is 1 for the empty set.

Theorem 2. *For any IPHL* $\mathbf{P}$ *and any query q, Algorithm 1 computes the correct lower probability bound of q, as defined by* (3) *and* (4)*.*

The proof of this theorem is not provided due to lack of space.

Example 14. *The rules for chemicals in Example 9 are translated to:*

$$(\mathrm{d}\colon \mathit{chemicals}) \leftarrow \top$$
$$(1-(0.7/(1-\mathrm{d}))\colon \mathit{chemicals}) \leftarrow \mathit{tanker}$$

Note that in case tanker holds, the probability of chemicals according to (2) *is:* $(1-(0.7/(1-\mathrm{d})))\mathrm{d}+(0.7/(1-\mathrm{d}))\mathrm{d}+(1-(0.7/(1-\mathrm{d})))(1-\mathrm{d}) = 0.3$*, which is equal to the lower bound for that case defined by* $\mathbf{R}_H$.

A similar transformation of rules with head-imprecision is incorrect for the upper bound, because it decreases with larger cardinality of $\mathbf{B}$ and additional rules of which the body holds can only increase, but not decrease, the probability. This problem is solved by actually computing the lower bound of the query's negation and computing the upper bound of the original query as $\overline{P}(q) = 1 - \underline{P}(\neg q)$. To compute the lower bound of the negation we transform the knowledge base, such that each atom actually represents its negation. This can be achieved by just swapping $\wedge$ and $\vee$ in *Clark's completion* and use as weight for auxiliary atoms $1 - \mathit{upper}(\mathbf{p})$.

To characterize the computational complexity of a full IPHL program, it makes no sense to speak about a corresponding precise probabilistic logic program for IPHLs with head-impressions as in Theorem 1, since it makes no sense to replace all probabilities with point probabilities for such rules. However, we still get the following guarantee in terms of complexity compared to programs with a similar structure.

Theorem 3. *Inference for a IPHL programs has the same complexity in terms of the treewidth, as a corresponding precise probabilistic logic program, where each head is defined in terms of the same body atoms as in the IPHL program. That is, the complexity is* $O(n2^w)$*, where* n *is the number of variables and* w *the program CNF's treewidth.*

Proof. This follows from the complexity of WMC [3] and the fact that the translation in Algorithm 1 does not change the program's treewidth. □

5 RELATED WORK

Several approaches have been proposed to combine probability theory and qualitative modalities. One example is the logic of Bacchus [1], which makes it possible to put constraints on probabilities as 'the agent believes φ with probability greater than 0.5'. The connection to imprecise probability theory is not made in this work. There are also approaches allowing for higher-order probabilistic statements [9], such as 'the probability that the probability of φ is larger 0.5 is 0.9'. This work is mainly theoretical in nature and efficient inference mechanisms are not provided.

The epistemic logic approaches of Milch and Koller [13] and Shirazi and Amir [17], deal with beliefs of multiple agents and also higher-order beliefs about beliefs, and therefore have a different goal as our approach. They provide mechanisms for exact inference, based on Bayesian networks. However, this probabilistic inference is only a subroutine of the complete inference method whereas the complete inference is strictly more expensive than ordinary probabilistic inference.

There is some work on inference for imprecise formalisms such as *locally defined credal networks* (LDCNs) [5]. All exact approaches, such as [2, 6], suffer from the worst-case complexity of the problem. We do not discuss approximate methods here, since they are not suited for qualitative inference, as we discussed before.

6 CONCLUSIONS

We introduced an imprecise probabilistic horn clause logic language, which makes it possible to express and unambiguously define qualitative statements with varying level of precision, with as special cases complete ignorance, point probabilities and determinism. This is made possible by a solid semantic foundation based on imprecise probability theory. We have furthermore shown that it is possible to provide inference for an imprecise language, which is as expensive as its precise counterpart, while in general imprecise inference problems are more complex. Finally, the approach shows that it is possible to employ state-of-the-art probabilistic inference methods for imprecise problems.

REFERENCES

[1] F. Bacchus, *Representing and Reasoning with Probabilistic Knowlege: A Logical Approach to Probabilities*, MIT Press, Cambridge, MA, 1990.

[2] Andrés Cano, José E Cano, and Serafín Moral, 'Convex sets of probabilities propagation by simulated annealing', in *In Proceedings of the Fith International Conference IPMU'94*. Citeseer, (1994).

[3] Mark Chavira and Adnan Darwiche, 'On probabilistic inference by weighted model counting', *Artif. Intell.*, **172**(6-7), 772–799, (2008).

[4] K.L. Clark, 'Negation as failure', *Logic and Data Bases*, 293–322, (1978).

[5] Fabio G Cozman, 'Credal networks', *Artificial Intelligence*, **120**(2), 199–233, (2000).

[6] Enrico Fagiuoli and Marco Zaffalon, '2u: an exact interval propagation algorithm for polytrees with binary variables', *Artificial Intelligence*, **106**(1), 77–107, (1998).

[7] Daan Fierens, Guy Van den Broeck, Joris Renkens, Dimitar Shterionov, Bernd Gutmann, Ingo Thon, Gerda Janssens, and Luc De Raedt, 'Inference and learning in probabilistic logic programs using weighted boolean formulas', *arXiv preprint arXiv:1304.6810*, (2013).

[8] Robert Goldblatt, *Lectures on the hyperreals: an introduction to nonstandard analysis*, volume 188, Springer, 1998.

[9] Aviad Heifetz and Philippe Mongin, 'The modal logic of probability', in *Proceedings of the 7th conference on Theoretical aspects of rationality and knowledge*, pp. 175–185. Morgan Kaufmann Publishers Inc., (1998).

[10] J.W. Lloyd, *Foundations of Logic Programming, 2nd Edition*, Springer, 1987.

[11] Steffen Michels, Arjen Hommersom, Peter J. F. Lucas, Marina Velikova, and Pieter W. M. Koopman, 'Inference for a new probabilistic constraint logic', in *IJCAI*, ed., Francesca Rossi. IJCAI/AAAI, (2013).

[12] Steffen Michels, Marina Velikova, Arjen Hommersom, and Peter JF Lucas, 'A decision support model for uncertainty reasoning in safety and security tasks', in *Systems, Man, and Cybernetics (SMC), 2013 IEEE International Conference on*, pp. 663–668. IEEE, (2013).

[13] Brian Milch and Daphne Koller, 'Probabilistic models for agents' beliefs and decisions', in *Proceedings of the Sixteenth conference on Uncertainty in artificial intelligence*, pp. 389–396. Morgan Kaufmann Publishers Inc., (2000).

[14] Mehmet A Orgun and Wanli Ma, 'An overview of temporal and modal logic programming', in *Temporal logic*, 445–479, Springer, (1994).

[15] Judea Pearl, *Probabilistic reasoning in intelligent systems: networks of plausible inference*, Morgan Kaufmann, 1988.

[16] Taisuke Sato, 'A statistical learning method for logic programs with distribution semantics', in *ICLP*, pp. 715–729, (1995).

[17] Afsaneh Shirazi and Eyal Amir, 'Probabilistic modal logic', in *Proceedings of the national conference on artificial intelligence*, volume 22, p. 489. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, (2007).

[18] Peter Walley, 'Towards a unified theory of imprecise probability', *International Journal of Approximate Reasoning*, **24**(2), 125–148, (2000).