*An Asymptotic Analysis of Probabilistic Logic Programming, with Implications for Expressing Projective Families of Distributions**

FELIX Q. WEITKÄMPER

Institut für Informatik, LMU München, Oettingenstr. 67, 80538 München
(*e-mail:* felix.weitkaemper@lmu.de)

Abstract

Probabilistic logic programming is a major part of statistical relational artificial intelligence, where approaches from logic and probability are brought together to reason about and learn from relational domains in a setting of uncertainty. However, the behaviour of statistical relational representations across variable domain sizes is complex, and scaling inference and learning to large domains remains a significant challenge. In recent years, connections have emerged between domain size dependence, lifted inference and learning from sampled subpopulations. The asymptotic behaviour of statistical relational representations has come under scrutiny, and projectivity was investigated as the strongest form of domain size dependence, in which query marginals are completely independent of the domain size. In this contribution we show that every probabilistic logic program under the distribution semantics is asymptotically equivalent to an acyclic probabilistic logic program consisting only of determinate clauses over probabilistic facts. We conclude that every probabilistic logic program inducing a projective family of distributions is in fact everywhere equivalent to a program from this fragment, and we investigate the consequences for the projective families of distributions expressible by probabilistic logic programs.

KEYWORDS: probabilistic logic programming, asymptotic quantifier elimination, determinate logic programs, projective families of distributions, finite model theory, distribution semantics

1 Introduction: Projectivity and statistical relational artificial intelligence

Statistical relational artificial intelligence has emerged over the last 25 years as a means to specify statistical models for relational data. Since then, many different frameworks have been developed under this heading, which can broadly be classified into those who extend logic programming to incorporate probabilistic information (probabilistic logic programming under the distribution semantics) and those who specify an abstract template for probabilistic graphical models (sometimes known as knowledge-based model construction).

Both classes share the distinction between a general model (a template or a probabilistic logic program with variables) and a specific domain used to ground the model. Ideally, the model would be specified abstractly and independently of a specific domain, even though a specific domain may well have been involved in learning the model from data.

However, a significant hurdle is the generally hard to predict or undesirable behaviour of the model when applied to domains of different sizes. This extrapolation problem has received much attention in the past years (Poole *et al.* 2014; Jaeger and Schulte 2020a). Recently Jaeger and Schulte (2018; 2020a) have identified *projectivity* as a strong form of good scaling behaviour: in a projective model, the probability of a given property holding for a given object in the domain is completely independent of the domain size. However, the examples of Poole *et al.* (2014) show that projectivity cannot be hoped for in general statistical relational models, and Jaeger and Schulte (2018) identify very restrictive fragments of common statistical relational frameworks as projective.

The question remains, however, whether those fragments completely capture the projective families of distributions expressible by a statistical relational representation. We will show in this contribution that in the case of probabilistic logic programming under the distribution semantics, this is true, as every projective probabilistic logic program is equivalent to a determinate acyclic probabilistic logic program.

Our method will show that, moreover, every probabilistic logic program is asymptotically equivalent to an acyclic determinate probabilistic logic program. This result is of some independent interest, as it shows that the probabilities of queries expressed by a logic program converge as domain size increases. Moreover, the asymptotic equivalence provides an explicit representation using which the asymptotic query probabilities can be computed.

This will be an application of an asymptotic quantifier elimination result for probabilistic logic programming derived from classical finite model theory, namely from the study of the asymptotic theory of first-order and least fixed point (LFP) logic in the 1980s (particularly 0–1 laws, applied in the form of Blass *et al.* (1985)).

This application is also methodologically interesting as it opens another way in which classical logic can contribute to cutting-edge problems in learning and reasoning. That the theory developed around 0–1 laws would be a natural candidate for such investigations may not surprise, as it is highly developed and is itself in the spirit of "finite probabilistic model theory" (Cozman and Mauá 2019, Section 7), and one might hope for more cross-fertilisation between the two fields in the future.

1.1 Outline of the paper

We will first formally introduce the framework of families of distributions and the notion of projectivity that we will refer to throughout.

In the following section, we present the abstract distribution semantics, which bridges the gap between the tools from finite model theory and the semantics of probabilistic logic programming. We also discuss asymptotic quantifier elimination and introduce the main classical results from finite model theory.

We introduce LFP logic, an adequate representation for (probabilistic) logic programs. We then give the necessary background on the asymptotic behaviour of LFP logic. We

harness the relationship between probabilistic logic programming and LFP distributions to show that every probabilistic logic program is asymptotically equivalent to an acyclic determinate probabilistic logic program.

In the following section, we will apply this analysis to study the projective families of distributions expressible in probabilistic logic programming. We see that every projective logic program is actually everywhere equivalent to an acyclic determinate logic program, and we derive some properties for the projective distributions expressible in this way. For the case of a unary vocabulary, we show that only very few projective families of distributions are expressible in probabilistic logic programming, and we give a concrete example to highlight that point.

Finally, we conclude the paper with a brief discussion of the complexity of asymptotic quantifier elimination and some impulses for further research.

Proofs to all the statements made here can be found in Appendix A in the supplementary material.

1.2 *Notation*

An introduction to the terminology of first-order logic (FOL) used in this paper can be found in Appendix B.1, in the supplementary material. We just point out here that we use $\mathfrak{P}(A)$ to indicate the power set of a set A and $\vec{x}$ as a shorthand for a finite tuple $x_1, \ldots, x_n$ of arbitrary length.

1.3 *Projectivity*

We will introduce projective families of distributions in accordance with Jaeger and Schulte (2018; 2020a), where one can find a much more detailed exposition of the terms and their background. As we are interested in statistical relational representations as a means of abstracting away from a given ground model, we will refer to families of distributions with varying domain sizes.

Definition 1. A *family of distributions* for a relational vocabulary $\mathcal{S}$ is a sequence $\left(Q^{(n)}\right)_{n \in \mathbb{N}}$ of probability distributions on the sets Ω_n of all $\mathcal{S}$-structures with domain $\{1, \ldots, n\} \subseteq \mathbb{N}$.

Definition 2. A family of distributions is called *exchangeable* if every $Q^{(n)}$ is invariant under $\mathcal{S}$-isomorphism.

It is called *projective* if, in addition, for all $m < n \in \mathbb{N}$ and all $\omega \in \Omega_m$ the following holds:

$$Q^{(m)}(\{\omega\}) = Q^{(n)}\left(\{\omega' \in \Omega_n | \omega \text{ is the substructure of } \omega' \text{ with domain } \{1, \ldots, n\} \}\right).$$

Projectivity encapsulates a strong form of domain size independence. Consider, for instance, the query $R(x)$, where R is a relation symbol in $\mathcal{S}$. Then in an exchangeable family of distributions, the unconditional probability of $R(x)$ holding in a world is independent of the precise interpretation of x, and depends only on the domain size. If the family of distributions is projective, then the probability of $R(x)$ is independent even of the domain size. As an immediate consequence, this implies that the computational complexity of quantifier-free queries is constant with domain size, since queries can always be evaluated in a domain consisting just of the terms mentioned in the query itself.

Projectivity also has important consequences for the statistical consistency of learning from randomly sampled subsets (Jaeger and Schulte 2018).

An important class of examples of projective families of distributions are those in which $R(a)$ is independent of $P(b)$ for all R, P, a, b. For instance, consider a vocabulary $\mathcal{S}$ with unary relations P and R, and a family of distributions in which for every domain element a, $P(a)$ and $R(a)$ are determined independently with probabilities p and r respectively. Then the probability that a subset A of a domain D has $\mathcal{S}$-structure M is given by

$$p^{|a \in M|M \models P(a)|} \cdot (1-p)^{|a \in M|M \models \neg P(a)|} \cdot r^{|a \in M|M \models R(a)|} \cdot (1-r)^{|a \in M|M \models \neg R(a)|},$$

regardless of the size of D.

The work of Jaeger and Schulte (2020a) provides a complete characterisation of projective families of distributions in terms of exchangeable arrays (AHK representations). However, it is not clear how this representation translates to the statistical relational formalisms currently in use, such as probabilistic logic programming. We will see below that there are indeed projective families of distributions that are not expressible by a probabilistic logic program. Furthermore, Jaeger and Schulte (2020a) claimed in Proposition 7.1 of their paper an independence property for the subclass of AHK- distributions. While this proposition proved to be incorrect for the class of AHK- distributions (Jaeger and Schulte 2020b), we will see here that for a projective family of distributions induced by a probabilistic logic program, the independence property holds.

In the remainder of this paper, we will investigate the interplay between the asymptotic behaviour of logical theories as they have been studied in finite model theory and the families of distributions that are induced by them. We therefore introduce a notion of asymptotic equivalence of families of distributions.

2 Abstract distribution semantics

As a bridge between classical notions from finite model theory and probabilistic logic programming, we introduce the abstract distribution semantics. It builds on the *relational Bayesian network specifications* of Cozman and Maua (2019), which combine random and independent root predicates with non-root predicates that are defined by first-order formulas. Here we streamline and generalise this idea to a unified framework that we call the *abstract distribution semantics*. In particular, we will generalise away from FOL to a general *logical language:*

Definition 3. Let $\mathcal{R}$ be a vocabulary. Then a *logical language $L(\mathcal{R})$* consists of a collection of functions φ which take an $\mathcal{R}$-structure M and returns a subset of M^n for some $n \in \mathbb{N}$ (called the *arity* of φ). In analogy to the formulas of first-order logic, we refer to those functions as *$L(\mathcal{R})$-formulas* and write $M \models \varphi(\vec{a})$ whenever $\vec{a} \in \varphi(M)$.

The archetype of a logical language is the first-order predicate calculus, where an R-formula φ defines a function $\varphi(M) := \{a \in M | M \models \varphi(a)\}$ and $\models$ is used in the sense of ordinary first-order logic. The concept as defined here is sufficiently general to accommodate many other choices, however, and we will later apply it to least fixed point logic in particular.

Definition 4. Let $\mathcal{S}$ be a relational vocabulary, $\mathcal{R} \subseteq \mathcal{S}$, and let $L(\mathcal{R})$ be a logical language over $\mathcal{R}$. Then an *abstract L-distribution over $\mathcal{R}$ (with vocabulary $\mathcal{S}$)* consists of the following data:

For every $R \in \mathcal{R}$ a number $q_R \in \mathbb{Q} \cap [0,1]$.

For every $R \in \mathcal{S} \backslash \mathcal{R}$, an $L(\mathcal{R})$-formula ϕ_R of the same arity as R.

In the following we will assume that all vocabularies are finite. The semantics of an abstract distribution is only defined relative to a domain D, which we will also assume to be finite. The formal definition is as follows:

Definition 5. Let $L(\mathcal{R})$ be a logical language over $\mathcal{R}$ and let D be a finite set. Let T be an abstract L-distribution over $\mathcal{R}$. Let Ω_D be the set of all $\mathcal{R}$-structures with domain D.

Then the *probability distribution on Ω_D induced by T*, written $Q_T^{(D)}$, is defined as follows:

For all $\omega \in \Omega_D$, if $\exists_{\vec{a} \in \vec{D}} \exists_{R \in \mathcal{S} \backslash \mathcal{R}} : R(\vec{a}) \not\Leftrightarrow \phi_R(\vec{a})$, then $Q_T^{(D)}(\{\omega\}) := 0$

Otherwise, $Q_T^{(D)}(\{\omega\}) := \prod_{R \in \mathcal{R}} (q_R^{|\{\vec{a} \in \vec{D} | R(\vec{a})\}|}) \times \prod_{R \in \mathcal{R}} (1 - q_R)^{|\{\vec{a} \in \vec{D} | \neg R(\vec{a})\}|}$

In other words, all the relations in $\mathcal{R}$ are independent with probability q_R and the relations in $\mathcal{S} \backslash \mathcal{R}$ are defined deterministically by the $L(\mathcal{R})$-formulas ϕ_R. We will illustrate that with an example.

Example 1. Let $\mathcal{R} = \{R, P\}$, $\mathcal{S} = \{R, P, S\}$, for a unary relation R a binary relation P and a unary relation S. Then an abstract distribution over (R) has numbers q_R and q_P which encode probabilities. Consider the FOL distribution T with $\varphi_S = \exists_y (R(x) \land P(x,y))$. For any domain D, $Q_T^{(D)}$ is obtained by making an independent choice of $R(a)$ or $\neg R(a)$ for every $a \in D$, with a q_R probability of $R(a)$. Similarly, an independent choice of $P(a,b)$ or $\neg P(a,b)$ is made for every pair (a,b) from D^2, with a q_P probability of $P(a,b)$. Then, for any possible $\mathcal{R}$-structure, the interpretation of S is determined by $\forall_x S(x) \leftrightarrow \varphi_S(x)$. The resulting family of distributions is not projective, since the probability of $Q(a)$ increases with the size of the domain as more possible candidates b for $P(a,b)$ are added.

As this example has shown, abstract FOL distributions do not necessarily give rise to projective families. If the φ are all given by quantifier-free formulas, however, then the induced families of distributions are indeed projective. We call such abstract $L(\mathcal{R})$ distributions, in which $L(\mathcal{R})$ is the class of quantifier-free FOL formulas over $\mathcal{R}$, *quantifier-free distributions*.

Proposition 6. *Every abstract quantifier-free distribution induces a projective family of distributions.*

Quantifier-free distributions also hold a special role in model-theoretic analysis. In particular, *asymptotic quantifier elimination* has been shown for various logics of interest to artificial intelligence.

2.1 Asymptotic quantifier elimination

We introduce our notion of asymptotic equivalence for families of distributions:

Definition 7. Two families of distributions $(Q^{(n)})$ and $(Q'^{(n)})$ are *asymptotically equivalent* if $\lim_{n \to \infty} \sup_{A \subseteq \Omega_n} |Q^{(n)}(A) - Q'^{(n)}(A)| = 0$

Remark. In measure theoretic terms, the families of distributions $(Q^{(n)})$ and $(Q'^{(n)})$ are asymptotically equivalent if and only if the limit of the total variation difference between them is 0.

We extend the notion to abstract distributions by calling abstract distributions asymptotically equivalent if they induce asymptotically equivalent families of distributions. This gives us the following setting for asymptotic quantifier elimination:

Definition 8. Let $L(\mathcal{R})$ be an extension of the class of quantifier-free $\mathcal{R}$-formulas. Then $L(\mathcal{R})$ *has asymptotic quantifier elimination* if every abstract $L(\mathcal{R})$ distribution is asymptotically equivalent to a quantifier-free distribution over $L(\mathcal{R})$.

It is well-known that first-order logic has asymptotic quantifier elimination.

Indeed, the asymptotic theory of relational first-order logic can be summarised as follows (Ebbinghaus and Flum 2006, Chapter 4):

Definition 9. Let $\mathcal{R}$ be a relational vocabulary. Then the first-order theory $\mathrm{RANDOM}(\mathcal{R})$ is given by all axioms of the following form, called *extension axioms over* $\mathcal{R}$:

$$\forall_{v_1,\ldots,v_r} \left(\bigwedge_{1 \leq i < j \leq r} v_i \neq v_j \to \exists_{v_{r+1}} \left(\bigwedge_{1 \leq i \leq r} v_i \neq v_{r+1} \wedge \bigwedge_{\varphi \in \Phi} \varphi \wedge \bigwedge_{\varphi \in \Delta_{r+1} \setminus \Phi} \neg\varphi \right) \right),$$

where $r \in \mathbb{N}$ and Φ is a subset of

$$\Delta_{r+1} := \{R(\vec{x}) | R \in \mathcal{R}, \ \vec{x} \text{ a tuple from } \{v_1, \ldots, v_{r+1}\} \text{ containing } v_{r+1}\}.$$

Fact 10. $\mathrm{RANDOM}(\mathcal{R})$ *eliminates quantifiers, that is, for each formula* $\varphi(\vec{x})$ *there is a quantifier-free formula* $\varphi'(\vec{x})$ *such that* $\mathrm{RANDOM}(\mathcal{R}) \vdash \forall_{\vec{x}}(\varphi(\vec{x}) \leftrightarrow \varphi'(\vec{x}))$.

It is sometimes helpful to characterise this quantifier-free formula somewhat more explicitly:

Proposition 11. *Let* $\varphi(\vec{x})$ *be a formula of first-order logic. Then:*

1. $\varphi'(\vec{x})$ *as in Fact 10 can be chosen such that only those relation symbols occur in* φ' *that occur in* φ.
2. *If every atomic subformula of* φ *contains at least one free variable not in* $\vec{x}$, *and no relation symbol occurs with different variables in different literals, then either* $\mathrm{RANDOM}(\mathcal{R}) \vdash \forall_{\vec{x}}\varphi(\vec{x})$ *or* $\mathrm{RANDOM}(\mathcal{R}) \vdash \forall_{\vec{x}}\neg\varphi(\vec{x})$.

The importance of $\mathrm{RANDOM}(\mathcal{R})$ comes from its role as the asymptotic limit of the class of all $\mathcal{R}$-structures. In fact, it axiomatises the limit theory of $\mathcal{R}$-structures even when the individual probabilities of relational atoms are given by q_R rather than $\frac{1}{2}$:

Fact 12. $\lim_{n \to \infty} Q_T^{(n)}(\varphi) = 1$ *for all abstract distributions* T *over* $\mathcal{R}$ *and all extension axioms* φ *over* $\mathcal{R}$.

Corollary 13. *First-order logic has asymptotic quantifier elimination.*

3 Probabilistic logic programs as LFP distributions

We will now proceed briefly to discuss fixed point logics. Our presentation follows the book by Ebbinghaus and Flum (2006, Chapter 8), to which we refer the reader for a more detailed exposition. We begin by introducing the syntax.

As atomic second-order formulas occur, as subformulas of LFP formulas, we will introduce second-order variables.

Definition 14. Assume an infinite set of second-order variables, indicated customarily by upper-case letters from the end of the alphabet, each annotated with a natural number arity. Then an *atomic second-order formula φ* is either a (first-order) atomic formula, or an expression of the form $X(t_1, \ldots, t_n)$, where X is a second-order variable of arity n and $t_1, \ldots, t_n$ are constants or (first-order) variables.

We now proceed to least fixed point formulas.

Definition 15. A formula φ is called *positive in a variable x* if x is in the scope of an even number of negation symbols in φ.

A *formula in least fixed point logic* or *LFP formula* over a vocabulary $\mathcal{R}$ is defined inductively as follows:

1. Any atomic second-order formula is an LFP formula.
2. If φ is an LFP formula, then so is $\neg\varphi$.
3. If φ and ψ are LFP formulas, then so is $\varphi \vee \psi$
4. If φ is an LFP formula, then so is $\exists x \varphi$ for a first-order variable x.
5. If φ is an LFP formula, then so is $[\text{LFP}_{\vec{x}, X} \varphi] \vec{t}$, where φ is positive in the second-order variable X and the lengths of the string of first-order variables $\vec{x}$ and the string of terms $\vec{t}$ coincide with the arity of X.

An occurrence of a second-order variable X is *bound* if it is in the scope of an LFP quantifier $\text{LFP}_{\vec{x}, X}$ and *free* otherwise.

Fixed point semantics have been used extensively in (logic) programming theory (Fitting 2002), and we will exploit this when relating the model theory of LFP to probabilistic logic programming below.

We first associate an operator with each LFP formula φ:

Definition 16. Let $\varphi(\vec{x}, \vec{u}, X, \vec{Y})$ be an LFP formula, with the length of $\vec{x}$ equal to the arity of X, and let ω be an $\mathcal{R}$-structure with domain D. Let $\vec{b}$ and $\vec{S}$ be an interpretation of $\vec{u}$ and $\vec{Y}$ respectively. Then we define the operator $F^\varphi : \mathfrak{P}(D^k) \to \mathfrak{P}(D^k)$ as follows:

$$F^\varphi(R) := \left\{ \vec{a} \in D^k | \omega \models \varphi(\vec{a}, \vec{b}, R, \vec{S}) \right\}.$$

Since we have restricted Rule 5 in Definition 15 to positive formulas, F^φ is monotone for all φ (i.e. $R \subseteq F^\varphi(R)$ for all $R \subseteq D^k$). Therefore we have:

Fact 17. *For every LFP formula $\varphi(\vec{x}, \vec{u}, X, \vec{Y})$ and every $\mathcal{R}$-structure on a domain D and interpretation of variables as in Definition 16, there is a relation $R \subseteq D^k$ such that $R = F^\varphi(R)$ and that for all R' with $R' = F^\varphi(R')$ we have $R \subseteq R'$.*

Definition 18. We call the R from Fact 17 the *least fixed point* of $\varphi(\vec{x}, \vec{u}, X, \vec{Y})$

Now we are ready to define the semantics of least fixed point logic:

Definition 19. By induction on the definition of an LFP formula, we define when an LFP formula $\varphi(\vec{X}, \vec{x})$ is said to *hold* in an $\mathcal{R}$-structure ω for a tuple $\vec{a}$ from the domain of ω and relations $\vec{A}$ of the correct arity:

The first-order connectives and quantifiers $\neg$, $\vee$ and $\exists$ as well as $\wedge$ and $\forall$ defined from them in the usual way are given the usual semantics.

An atomic second-order formula $X(\vec{x}, \vec{c})$ holds if and only if $(\vec{a}, \vec{c_\omega}) \in A$.

$[\text{LFP}_{\vec{x}, X} \varphi] \vec{t}$ holds if and only if $\vec{a}$ is in the least fixed point of $F^{\varphi(\vec{x}, X)}$.

3.1 Probabilistic logic programming

Our discussion on probabilistic logic programming employs the simplification proposed by Riguzzi and Swift (2018) and considers a probabilistic logic program as a stratified Datalog program over probabilistic facts. This *distribution semantics* covers several different equally expressive formalisms (Riguzzi and Swift 2018; Raedt and Kimmig 2015). Note that in particular, probabilistic logic programs as used here do not involve function symbols, unstratified negation or higher-order constructs.

See Appendix B.2 in the supplementary material or the book by Ebbinghaus and Flum (2006, Chapter 9) for an introduction to the syntax and semantics of stratified Datalog programs in line with this paper.

We will use the notation $(\Pi, P)\vec{t}$ for an intensional symbol P of a stratified logic program Π to mean that "the program Π proves $P\vec{t}$".

Definition 20. A *probabilistic logic program* consists of probabilistic facts and deterministic rules, where the deterministic part is a stratified Datalog program. We will consider it in our framework of abstract distribution semantics as follows:

$\mathcal{R}$ is given by relation symbols R' for every probabilistic fact $p_R :: R(\vec{x})$, with $q_{R'} := p_R$. Their arity is just the arity of R.

$\mathcal{S}$ is given by the vocabulary of the probabilistic logic program and additionally the R' in $\mathcal{R}$.

Let Π be the stratified Datalog program obtained by prefixing the program $\{R'(\vec{x}) \leftarrow R(\vec{x}) | R' \in \mathcal{R}\}$ to the deterministic rules of the probabilistic logic program.

Then ϕ_P for a $P \in \mathcal{S} \backslash \mathcal{R}$ is given by $(\Pi, P)\vec{x}$.

The distribution semantics for probabilistic logic programming is related to the LFP distribution semantics introduced above through the following fact (Ebbinghaus and Flum 2006, Theorem 9.1.1):

Fact 21. *For every stratifiable Datalog formula $(\Pi, P)\vec{x}$ as above, there exists an* LFP *formula $\varphi(\vec{x})$ over the extensional vocabulary $\mathcal{R}$ of Π such that for every $\mathcal{R}$-structure ω and every tuple $\vec{a}$ of elements of ω of the same length as $\vec{x}$, $\omega \models \varphi(\vec{a})$ if and only if $\omega \models (\Pi, P)\vec{a}$.*

Remark. In fact, it suffices to consider formulas in the so-called bounded fixed point logic, whose expressiveness lies between first-order logic and least fixed point logic (Ebbinghaus and Flum 2006).

Notation. Although we have allowed second-order variables in the inductive definitions above, we will assume from now on unless mentioned otherwise that LFP formulas do not have free second-order variables.

3.2 *Asymptotic quantifier elimination for probabilistic logic programming*

We discuss the asymptotic reduction of LFP to FOL by Blass *et al.* (1985) and conclude that abstract LFP distributions and therefore probabilistic logic programs have asymptotic quantifier elimination.

The main theorem of Blass *et al.* (1985) shows that RANDOM($\mathcal{R}$) not only eliminates classical quantifiers, but also LFP quantifiers:

Fact 22. *Let $\varphi(\vec{x})$ be an* LFP *formula over $\mathcal{R}$. Then there is a finite subset G of* RANDOM($\mathcal{R}$) *and a quantifier-free formula $\varphi'(\vec{x})$ such that $G \vdash \forall_{\vec{x}}\varphi(\vec{x}) \leftrightarrow \varphi'(\vec{x})$.*

Putting this together, we can derive the following:

Theorem 23. *Least fixed point logic has asymptotic quantifier elimination.*

To obtain a characterisation within probabilistic logic programming, however, we need to translate quantifier-free first-order formulas back to stratifiable Datalog.

In fact, they can be mapped to a subset of stratified Datalog that is well-known from logic programming:

Definition 24. A Datalog program, Datalog formula or probabilistic logic program is called *determinate* if every variable occurring in the body of a clause also occurs in the head of that clause.

Example 2. Examples of determinate clauses in this sense are $R(x) :\!- P(x)$ or $Q(x,y) :\!- R(x)$. Indeterminate clauses include $R(x) :\!- P(y)$ or $R(x) :\!- Q(x,y)$.

Determinacy corresponds exactly to the fragment of probabilistic logic programs identified as projective by Jaeger and Schulte (2018, Proposition 4.3).

Indeed, Ebbinghaus and Flum's (2006) proof of their Theorem 9.1.1 shows:

Fact 25. *Every quantifier-free first-order formula is equivalent to an acyclic determinate stratified Datalog formula.*

Therefore, we can conclude from Proposition 6:

Proposition 26. *Every determinate probabilistic logic program is projective.*

We now turn to the main result of this subsection.

Theorem 27. *Every probabilistic logic program is asymptotically equivalent to an acyclic determinate probabilistic logic program.*

4 Projective probabilistic logic programs

As an application of our results, we investigate the projective families of distributions that are expressible by probabilistic logic programs.

The key is the following observation:

Proposition 28. *Two projective families of distributions are asymptotically equivalent if and only if they are equal.*

As modelling in the distribution semantics often involves introducing auxiliary predicates, the family of distributions we want to model will usually be defined on a smaller

vocabulary than the abstract distribution (or probabilistic logic program) itself. We therefore note here that asymptotic equivalence is preserved under reduct. First we clarify how we build reducts of distributions in the first place:

Definition 29. Let $Q^{(n)}$ be a distribution over a vocabulary $\mathcal{S}$. Then its *reduct* $Q^{(n)}_{\mathcal{S}'}$ *to a subvocabulary* $\mathcal{S}' \subseteq \mathcal{S}$ is defined such that for any world $\omega \in \Omega^{\mathcal{S}'}_n$, $Q^{(n)}_{\mathcal{S}'}(\omega) := Q^{(n)}(\{\omega' \in \Omega^{\mathcal{S}}_n | \omega'_{\mathcal{S}'} = \omega\})$.

Remark. $Q^{(n)}_{\mathcal{T}}$ is the pushforward measure of $Q^{(n)}$ with respect to the reduct projection from $\Omega^{\mathcal{S}}_n \to \Omega^{\mathcal{T}}_n$.

We can now formulate preservation of asymptotic equivalence under reducts:

Proposition 30. *The reducts of asymptotically equivalent families of distributions are themselves asymptotically equivalent.*

In combination, we obtain:

Theorem 31. *Let L be a logical language with asymptotic quantifier elimination that extends quantifier-free first-order logic. Let $\mathcal{R} \subseteq \mathcal{S}$ be vocabularies, and let $\mathcal{S}' \subseteq \mathcal{S}$. Furthermore let T be an L-distribution over $\mathcal{R}$ with vocabulary $\mathcal{S}$. Lastly, let $(Q^{(n)})$ be the family of distributions induced by T.*

Then the following holds: If $Q^{(n)}_{\mathcal{S}'}$ is projective, then there is a quantifier-free distribution T_q over $\mathcal{R}$ with vocabulary $\mathcal{S}$ such that $Q^{(n)}_{\mathcal{S}'}$ is the reduct of the family of distributions induced by T_q to $\mathcal{S}'$.

In particular, a projective family of distributions that can be expressed in probabilistic logic programming at all can in fact be expressed already by a determinate probabilistic logic program.

5 Implications and discussion

The results have immediate consequences for the expressiveness of probabilistic logic programming.

We first discuss a particularly striking observation:

5.1 Asymptotic loss of information

Very insightful is the case of a probabilistic *rule*, that is, a clausal formula annotated with a probability. Because of its intuitive appeal, this is a widely used syntactic element of probabilistic logic programming languages such as Problog, and its semantics is defined by introducing a new probabilistic fact to model the uncertainty of the rule. More precisely:

$$p :: R(\vec{x}) :\!- Q_1(\vec{x}_1, \vec{y}_1), \ldots, Q_n(\vec{x}_n, \vec{y}_n),$$

(where $\vec{x}$ are the variables appearing in R, $\vec{x_i} \subseteq \vec{x}$) is interpreted as

$$p :: I(\vec{x}, \vec{y}); R(\vec{x}) :\!- Q_1(\vec{x}_1, \vec{y}_1), \ldots, Q_n(\vec{x}_n, \vec{y}_n), I(\vec{x}, \vec{y}),$$

(where $\vec{y} := \bigcup \vec{y}_i$).

It is now easy to see from Proposition 11 that in the asymptotic quantifier-free representation of this probabilistic rule, I will no longer occur, since it originally occurred

implicitly quantified in the body of the clause. However, I was the only connection between the probability annotation of the rule and its semantics! Therefore, the asymptotic probability of $R(\vec{x})$ is independent of the probability assigned to any non-determinate rule with $R(\vec{x})$ as its head.

5.2 Expressing projective families of distributions

Our results also show how few of the projective families of distributions can be expressed in those formalisms. This confirms the suspicion voiced in by Jaeger and Schulte (2020a) that despite the ostensible similarities between languages such as independent choice logic, which are based on the distribution semantics, and the array representation introduced by Jaeger and Schulte (2020a), a direct application of techniques from probabilistic logic programming to general projective families of distributions might prove challenging.

We start by displaying some properties shared by the projective distributions induced by a probabilistic logic program.

Definition 32. A projective family of distributions has the *Independence Property* or *IP* if for all $\mathcal{S}$-formulas $\varphi(x_1, \ldots x_n)$ and $\psi(x_1, \ldots x_m)$ the events $\{1, \ldots, n\} \models \varphi$ and $\{n+1, \ldots, n+m\} \models \psi$ are independent under $Q^{(n+m)}$. A projective family of distributions $(Q^{(n)})$ of $\mathcal{S}$-structures has the *Conditional Independence Property* or *CIP* if for all n and all quantifier-free $\mathcal{S}$-formulas $\varphi(x_1, \ldots x_n)$ and every $\mathcal{S}$-structure ω on a domain with $n-1$ elements, the events $\{1, \ldots, n\} \models \varphi$ and $\{1, \ldots, n+1\} \setminus \{n\} \models \varphi$ are conditionally independent over $\{1, \ldots, n-1\} \models \omega$ under $Q^{(n+1)}$.

IP has been studied extensively in the field of pure inductive logic (Paris and Vencovská 2015), while CIP is a generalisation of the property that Jaeger and Schulte (2020a) claimed in their Proposition 7.1 for AHK- distributions, to arbitrary quantifier-free formulas rather than worlds.

Example 3. Consider the quantifier-free abstract distribution with a probabilistic fact $R(x)$ with associated probability p and a binary predicate $P(x, y)$ with definition $\phi_P = x = y \lor R(x)$. Then its induced family of distributions satisfies CIP and IP. However, the reduct to the vocabulary $\{P\}$ does not satisfy CIP; indeed, consider the domain with elements $\{1, 2, 3\}$. Then there is just one $\{P\}$-structure ω with domain $\{1\}$ that has probability 1, namely the world where $P(1,1)$ is true. Consider the events $P(1,3)$ and $P(1,2)$. They are not independent, since in fact $P(1,2)$ iff $R(1)$ iff $P(1,3)$. Since there is just one possible $\{P\}$-structure ω on $\{1\}$, conditioning on ω does not alter the probabilities.

Proposition 33. *Let $(Q^{(n)})$ be a projective family of distributions induced by a quantifier-free abstract distribution. Then $(Q^{(n)})$ satisfies CIP. If it does not have any nullary relation symbols, it also satisfies IP.*

As mentioned above, one often expands the vocabulary of interest when modelling in the distribution semantics. It is worth noting, therefore, that IP is trivially transferred to reducts, while CIP is not (see Example 3 above). We can view our results as positive or negative, depending on our viewpoint. We will begin with the positive formulation:

Corollary 34. *If a projective family of distributions is induced by a probabilistic logic program, it satisifies CIP.*

As CIP is a generalisation of the property claimed by Jaeger and Schulte (2020a) in their Proposition 7.1, this shows that while the class of AHK- representations does not satisfy this property (see the discussion in the appendix to Jaeger and Schulte's corrected version (2020b)), every projective family of distributions induced by a probabilistic logic program does.

Since CIP does not transfer to reducts, however, we look towards IP for a property that all projective families of distributions expressible in probabilistic logic programming satisfy.

Corollary 35. *Let $\mathcal{S}' \subseteq \mathcal{S}$ be relational vocabularies without nullary relation symbols. Then for every probabilistic logic program with vocabulary $\mathcal{S}$, if the reduct $(Q_{\mathcal{S}'}^{(n)})$ is projective, $(Q_{\mathcal{S}'}^{(n)})$ satisfies IP.*

If we allow nullary relations in $\mathcal{S}$, we obtain finite sums of distributions with IP instead.

Proposition 36. *Let $\mathcal{S}' \subseteq \mathcal{S}$ be relational vocabularies, possibly with nullary relation symbols. Then for every probabilistic logic program with vocabulary $\mathcal{S}$, if the reduct $(Q_{\mathcal{S}'}^{(n)})$ is projective, $(Q_{\mathcal{S}'}^{(n)})$ is a finite sum of distributions satisfying IP.*

It is natural to ask how strong the condition imposed by the previous results is, bringing us to the negative part of our results. As a special case, we consider a unary vocabulary $\mathcal{S}'$, which only has unary relation symbols, since the projective families of distributions are very well understood for such vocabularies.

Here, *de Finetti's Representation Theorem* (Paris and Vencovská 2015, Chapter 9) says that the projective families of distributions in a unary vocabulary are precisely the *potentially infinite* combinations of those that satisfy IP, while those projective families of distributions expressible in probabilistic logic programs are merely the *finite* combinations of those satisfying IP; so, in some sense "almost all" projective families of distributions in unary vocabularies cannot be expressed in probabilistic logic programming.

As a concrete example, we show that already in the very limited vocabulary of a single unary relation symbol R, there is no probabilistic logic program that induces the distribution that is uniform on isomorphism classes of structures:

Definition 37. Let $\mathcal{S} := \{R\}$ consist of one unary predicate, and let $\mathfrak{m}^*$ be the family of distributions on $\mathcal{S}$-structures defined by

$\mathfrak{m}^*(\{\omega\}) := \frac{1}{|D| * N_\omega}$ for a world $\omega \in \Omega_D$, where $N_\omega := |\{\omega' \in \Omega_D | \omega \cong \omega'\}|$.

This gives each isomorphism type of structures equal weight, and then within each isomorphism type every world is given equal weight too.

$\mathfrak{m}^*$ is an important probability measure for two reasons; it plays a special role in finite model theory since the so-called unlabeled 0-1 laws are introduced with respect to this measure. Furthermore, it was introduced explicitly by Carnap (1950; 1952) as a candidate measure for formalising inductive reasoning, as part of the so-called *continuum of inductive methods*. Paris and Vencovská (2015) provide a modern exposition of Carnap's theory.

$\mathfrak{m}^*$ is easily seen to be exchangeable; it is also projective, and in fact an elementary calculation shows that for any domain D and any $\{a_1, \ldots a_{n+1}\} \subseteq D$,

$$\mathfrak{m}^* \left(R(a_{n+1}) | \{R(a_i)\}_{i \in I \subseteq \{1, \ldots, n\}} \cup \{\neg R(a_i)\}_{i \in \{1, \ldots, n\} \setminus I} \right) = \frac{|I| + 1}{n + 1}, \qquad (5.1)$$

(see any of the sources above for a derivation).

Proposition 38. *Let $\mathcal{S}'$ be a finite vocabulary extending $\mathcal{S}$ from Definition 37. Then there is no probabilistic logic program with vocabulary $\mathcal{S}'$ such that the reduct of the induced family of distributions to $\mathcal{S}$ is equal to $\mathfrak{m}^*$.*

5.3 Complexity results

Since the theory of random structures is decidable, the asymptotic quantifier results in this paper provide us with an algorithmic procedure for determining an asymptotically equivalent acyclic determinate program for any given probabilistic logic program, and by extension a procedure for determining the asymptotic probabilities of quantifier-free queries. What can we say about the complexity of this procedure? Since the operation takes a non-ground probabilistic logic program as input and computes another probabilistic logic program, the notion of data complexity does not make sense in this context. Instead, program complexity is an appropriate measure.

In our context, the input program could be measured in different ways. Since our analysis is based on the setting of abstract distributions, we will be considering as our input abstract distributions obtained from (stratified) probabilistic logic programs . We will furthermore fix our vocabularies $\mathcal{R}$ and $\mathcal{S}$. Since the transformation acts on each ϕ_R in turn and independently, it suffices to consider the individual ϕ_R as input. It is natural to ask about complexity in the *length* of ϕ_R.

In fact, one can extract upper and lower bounds from the work of Blass *et al.* (1985), who build on the work of Grandjean (1983) for analysing the complexity of their asymptotic results. The task of determining whether the probability of a first-order sentence converges to 0 or 1 with increasing domain size, which is a special case of our transformation, is complete in PSPACE (Blass *et al.* 1985, Theorem 1.4). Therefore the program transformation is certainly PSPACE-hard. On the other hand, asymptotic elimination of quantifiers in LFP logic is complete in EXPTIME (Blass *et al.* 1985, Theorems 4.1 and 4.3), so the program transformation is certainly in EXPTIME.

In order to specify further, we note that for abstract first-order distributions, which correspond to acyclic probabilistic logic programs, the transformation can be performed in PSPACE:

Let R be of arity n. Then enumerate the (finitely many) quantifier-free n-types (φ_i) in $\mathcal{R}$. Now for any ϕ_R of arity n we can check successively in polynomial space in the length of ϕ_R, whether the probability of $\varphi_i \to \phi_R$ converges to 0 or 1. Then ϕ_R is asymptotically equivalent to the conjunction of those quantifier-free n-types for which 1 is returned.

In the general case of LFP logic, Blass *et al.* (1985) show that the problem of finding an asymptotically equivalent first-order sentence is EXPTIME complete. However, to represent stratified Datalog, only the fragment known as *bounded* or *stratified* LFP logic is required (Ebbinghaus and Flum 2006, Sections 8.7 and 9.1). Therefore, the complexity class of the program transformation of stratified probabilistic logic programs corresponds

to the complexity of the asymptotic theory of *bounded* fixed point logic, which to the best of our knowledge is still open.

6 Conclusion and further Work

By introducing the formalism of abstract distributions, we have related the asymptotic analysis of finite model theory to the distribution semantics underlying probabilistic logic programming. Thereby, we have shown that every probabilistic logic program is asymptotically equivalent to an acyclic determinate logic program. In particular, this representation provides us with an algorithm to evaluate the asymptotic probabilities of quantifier-free queries with respect to a probabilistic logic program. We have also seen that the asymptotic representation of a probabilistic logic program invoking probabilistic rules is in fact independent of the probability with which the rule is annotated. We applied our asymptotic results to study the projective families of distributions that can be expressed in probabilistic logic programming. We saw that they have certain independence properties, and that in particular the families of distributions induced on the entire vocabulary satisfy the conditional independence property. We also see that at least in the case of a unary vocabulary, only a minority of projective families of distributions can be represented, excluding important example such as Carnap's family of distributions $\mathfrak{m}^*$.

6.1 Further work

The analysis presented here suggests several strands of further research.

While some widely used directed frameworks can be subsumed under the probabilistic logic programming paradigm, undirected models such as Markov logic networks (MLNs) seem to require a different approach. Indeed, the projective fragment of MLNs isolated by Jaeger and Schulte (2018) is particularly restrictive, since it only allows formulas in which every literal has the same variables. Those are precisely the σ-determinate formulas discussed by Domingos and Singla (2007); cf. also the parametric classes of finite model theory (Ebbinghaus and Flum 2006, Section 4.2). It might therefore be expected that if an analogous result to Theorem 27 holds for MLNs, they could express even fewer projective families of distributions than probabilistic logic programs.

Beyond the FOL or LFP expressions used in current probabilistic logic programming, another direction is to explore languages with more expressive power. Candidates for this are for instance Keisler's (1985) logic with probability quantifiers or Koponen's (2020) conditional probability logic. Appropriate asymptotic quantifier elimination results have been shown in both settings (Koponen 2020; Keisler and Lotfallah 2009), allowing an immediate application of our results there.

The asymptotic quantifier elimination presented here excludes higher-order programming constructs from our probabilistic logic programs. Investigating the asymptotic theory of impredicative programs under a formalised semantics such as that presented by Bry (2020) could have direct consequences for the expressiveness of such more general probabilistic logic programs.

Finally, the failure of the classical paradigm under investigation to express general projective families of distributions suggests one may have to look beyond the current

methods and statistical relational frameworks to address the challenge of learning and inference for general projective families of distributions issued by Jaeger and Schulte (2020a).

Supplementary material

To view supplementary material for this article, please visit http://dx.doi.org/10.1017/S1471068421000314.

References

BLASS, A., GUREVICH, Y. AND KOZEN, D. 1985. A zero-one law for logic with a fixed-point operator. *Information and Control 67,* 1–3, 70–90.

BRY, F. 2020. In praise of impredicativity: A contribution to the formalization of meta-programming. *Theory and Practice of Logic Programming 20,* 1, 99–146.

CARNAP, R. 1950. *Logical Foundations of Probability.* University of Chicago Press.

CARNAP, R. 1952. *The Continuum of Inductive Methods.* University of Chicago Press.

COZMAN, F. G. AND MAUÁ, D. D. 2019. The finite model theory of bayesian network specifications: Descriptive complexity and zero/one laws. *International Journal of Approximate Reasoning 110,* 107–126.

DOMINGOS, P. M. AND SINGLA, P. 2007. Markov logic in infinite domains. In *Probabilistic, Logical and Relational Learning - A Further Synthesis, 15.04. – 20.04.2007,* L. D. Raedt, T. G. Dietterich, L. Getoor, K. Kersting, and S. Muggleton, Eds. Dagstuhl Seminar Proceedings, vol. 07161. Internationales Begegnungs- und Forschungszentrum fuer Informatik (IBFI), Schloss Dagstuhl, Germany.

EBBINGHAUS, H. AND FLUM, J. 2006. *Finite model theory,* 2nd ed. Springer Monographs in Mathematics. Springer.

FITTING, M. 2002. Fixpoint semantics for logic programming a survey. *Theoretical Computer Science 278,* 1-2, 25–51.

GRANDJEAN, E. 1983. Complexity of the first-order theory of almost all finite structures. *Information and Control 57,* 2/3, 180–204.

JAEGER, M. AND SCHULTE, O. 2018. Inference, learning, and population size: Projectivity for SRL models. In *Eighth International Workshop on Statistical Relational AI (StarAI).*

JAEGER, M. AND SCHULTE, O. 2020a. A complete characterization of projectivity for statistical relational models. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI 2020,* C. Bessiere, Ed. ijcai.org, 4283–4290.

JAEGER, M. AND SCHULTE, O. 2020b. A complete characterization of projectivity for statistical relational models, version 2. *CoRR abs/2004.10984v2.*

KEISLER, H. J. 1985. Probability quantifiers. In *Model-Theoretic Logics.* Perspect. Math. Logic. Springer, New York, 509–556.

KEISLER, H. J. AND LOTFALLAH, W. B. 2009. Almost everywhere elimination of probability quantifiers. *Journal of Symbolic Logic 74,* 4, 1121–1142.

KOPONEN, V. 2020. Conditional probability logic, lifted Bayesian networks, and almost sure quantifier elimination. *Theoretical Computer Science 848,* 1–27.

PARIS, J. AND VENCOVSKÁ, A. 2015. *Pure Inductive Logic.* Perspectives in Logic. Association for Symbolic Logic, Ithaca, NY; Cambridge University Press, Cambridge.

POOLE, D., BUCHMAN, D., KAZEMI, S. M., KERSTING, K. AND NATARAJAN, S. 2014. Population size extrapolation in relational probabilistic modelling. In *Scalable Uncertainty Management – 8th International Conference, SUM 2014, Oxford, UK, September 15-17, 2014.*

Proceedings, U. Straccia and A. Calì, Eds. Lecture Notes in Computer Science, vol. 8720. Springer, 292–305.

RAEDT, L. D. AND KIMMIG, A. 2015. Probabilistic (logic) programming concepts. *Machine Learning 100,* 1, 5–47.

RIGUZZI, F. AND SWIFT, T. 2018. A survey of probabilistic logic programming. In *Declarative Logic Programming: Theory, Systems, and Applications*, M. Kifer and Y. A. Liu, Eds. ACM/Morgan & Claypool, 185–228.