Representing Aggregators in Relational Probabilistic Models

David Buchman[*]
Department of Computer Science
University of British Columbia
Vancouver, BC, Canada
davidbuc@cs.ubc.ca

David Poole[†]
Department of Computer Science
University of British Columbia
Vancouver, BC, Canada
poole@cs.ubc.ca

Abstract

We consider the problem of, given a probabilistic model on a set of random variables, how to add a new variable that depends on the other variables, without changing the original distribution. In particular, we consider relational models (such as Markov logic networks (MLNs)), where we cannot directly define conditional probabilities. In relational models, there may be an unbounded number of parents in the grounding, and conditional distributions need to be defined in terms of aggregators. The question we ask is whether and when it is possible to represent conditional probabilities at all in various relational models. Some aggregators have been shown to be representable by MLNs, by adding auxiliary variables; however it was unknown whether they could be defined without auxiliary variables. For other aggregators, it was not known whether they can be represented by MLNs at all. We obtained surprisingly strong negative results on the capability of flexible undirected relational models such as MLNs to represent aggregators without affecting the original model's distribution. We provide a map of what aspects of the models, including the use of auxiliary variables and quantifiers, result in the ability to represent various aggregators. In addition, we provide proof techniques which can be used to facilitate future theoretic results on relational models, and demonstrate them on relational logistic regression (RLR).

1 Introduction

Probabilistic graphical models (PGMs) (Pearl 1988; Koller and Friedman 2009), which are perhaps better called factored joint probability models, have found various applications in Computer Science and other sciences. PGMs may be described as a collection of nonnegative functions called "factors" operating on subsets of the variables. The joint distribution is then defined as the normalized product of the factors. PGMs are commonly classified as either directed or undirected graphical models.

One may wish to extend a given PGM by adding new dependent variables. Adding a **dependent** variable to a model means that the new variable depends probabilistically on variables in the model, but adding the new variable has no effect on the original distribution if it is not observed.

Adding dependent variables is straightforward in the context of directed models, where the newly added dependent variable is called a "leaf." In fact, a directed model can be described as the result of iteratively adding dependent variables. However, we may want to add a dependent variable to an undirected model. This can be done by adding a factor connecting the new variable and the variables it depends on; however, model constraints (such as a maximal factor size) may prevent adding such a factor.

One scenario for adding variables to preexisting models as dependent variables, is the addition of (probabilistic or deterministic) summary variables. Another scenario is causality. Adding a variable representing an "effect" should not change the distribution of its "causes."

Other reasons for adding dependent variables may be that retraining the model with the additional variables may be computationally expensive, human-resource expensive (the model may have been built using expert knowledge), or outright impossible (for example, due to privacy concerns, we might not have access to individual health records, but only to probabilistic models based on them).

Dependent variables also have computational advantages during inference: unobserved dependent variables can be iteratively removed, in the reverse order to the order they were added. This is true even in *undirected* models, when some variables can be characterized as "dependent."

Recently, there is growing interest in probabilistic models for the relational setting. Relational probabilistic models (Getoor and Taskar 2007; De Raedt et al. 2008) extend probabilistic graphical models to include the notions of individuals, logical variables, and relations. These can be used for probabilistic modeling of relational structures. These models can be directed, typically building on Bayesian networks, or undirected, typically building on Markov networks.

Here we consider undirected models (Taskar, Abbeel, and Koller 2002; Richardson and Domingos 2006; Domingos et al. 2008), and use MLNs as the prototypical undirected relational probabilistic models, as these are general enough to include other models as special cases. We present significant negative results on the possibility to add dependent variables to these models.

[*] www.cs.ubc.ca/∼davidbuc
[†] www.cs.ubc.ca/∼poole

2 Background

Consider a set of variables $\mathbf{x} = \{x_1, \ldots, x_n\}$. A **factor** $f(\mathbf{x}_f) \geq 0$ is a nonnegative function over a subset $\mathbf{x}_f \subseteq \mathbf{x}$, called its **scope**. $|\mathbf{x}_f|$, the number of variables in the scope, is called the **factor size**. A factor $f(\mathbf{x}_f) > 0$ is called **positive**. A **probabilistic graphical model (PGM)** is a pair $\mathcal{M} = (\mathbf{x}, \mathcal{F})$, where $\mathbf{x}$ is a set of variables and $\mathcal{F}$ is a set of factors over subsets of $\mathbf{x}$. The **joint distribution** is the normalized product of the factors: $\mathrm{P}_{\mathcal{M}}(\mathbf{x}) := \frac{1}{Z} \prod_{f \in \mathcal{F}} f(\mathbf{x}_f)$, where Z is the **partition function**. $\mathcal{M}$ is **positive** if $\forall \mathbf{x}, \mathrm{P}_{\mathcal{M}}(\mathbf{x}) > 0$.

2.1 Dependent Variables

The concept of "dependent variables" cannot be defined given only a joint distribution. One has to compare two models, one with the variable and one without:

Definition 1. *Consider a probabilistic model $\mathcal{M}_0$ representing a joint probability distribution $\mathrm{P}_{\mathcal{M}_0}(\mathbf{x})$, which we alter resulting in $\mathcal{M}_1$, representing $\mathrm{P}_{\mathcal{M}_1}(\mathbf{x}, x')$. If $\sum_{x'} \mathrm{P}_{\mathcal{M}_1}(\mathbf{x}, x') = \mathrm{P}_{\mathcal{M}_1}(\mathbf{x}) = \mathrm{P}_{\mathcal{M}_0}(\mathbf{x})$, i.e., if the marginal distribution over the original variables does not change, then we call this alteration an **addition of a dependent variable** x', with the CPD $\mathrm{P}_{\mathcal{M}_1}(x' \mid \mathbf{x})$.*

Note that x' does not have to actually depend on other variables; just as in a Bayes net, variables do not have to actually depend on their parents.

Modeling dependent variables is a different concept than modeling conditional probability distributions (CPDs). Consider an undirected model with variables $\mathbf{x}$, to which we add a new variable x' and associated factors. The factors involving x' define the CPD $\mathrm{P}(x' \mid \mathbf{x})$. However, if x' is not observed and we sum it out of the model (Zhang and Poole 1994; Dechter 1996), we may get a new factor which alters the distribution of the original model. When this happens, we cannot say that x' was "added as a dependent variable."

Example 1. *Consider a model representing the uniform distribution over $x_1, x_2 \in \{0, 1\}$, which we want to extend by adding $x' \in \{0, 1\}$ with $\mathrm{P}(x' \mid x_1, x_2) \propto (1 + x_1 x')(1 + x_2 x')$. Consider adding the factors $f_1(x_1, x') = 1 + x_1 x' = \left[\begin{smallmatrix} 1 & 1 \\ 1 & 2 \end{smallmatrix}\right]$ and $f_2(x_2, x') = 1 + x_2 x' = \left[\begin{smallmatrix} 1 & 1 \\ 1 & 2 \end{smallmatrix}\right]$. The new model represents the CPD $\mathrm{P}(x' \mid x_1, x_2)$ correctly; however, x_1 and x_2 are now correlated. We have altered the original distribution $\mathrm{P}(x_1, x_2)$, hence x' was not "added as a dependent variable." However, also adding $f_3(x_1, x_2) = \left[\begin{smallmatrix} 15 & 10 \\ 10 & 6 \end{smallmatrix}\right]$ uncouples x_1 and x_2, resulting in the addition of x' as a dependent variable. Note that $f_1(x_1, x') f_2(x_2, x') f_3(x_1, x_2) = \text{constant} \cdot \mathrm{P}(x' \mid x_1, x_2)$, hence we "decomposed" $\mathrm{P}(x' \mid x_1, x_2)$ to pairwise factors.*

2.2 Relational Probabilistic Models

The relational setting is based on **populations**, each of which is a set of **individuals**, which might not be specified in advance. The relational setting introduces **parametrized random variables (PRVs)**, which are random variables (RVs) parametrized by **logical variables**. Each logical variable can refer to any individual from a specific population. A specification of the sets of individuals in each population defines a **grounding**, where every PRV defines a multi-dimensional array of random variables.

When defining probabilistic models for the relational setting, it is common to introduce **parametrized factors (PFs)**, which are nonnegative functions whose scope includes PRVs parametrized by logical variables. In each grounding, each PF is instantiated once for each combination of choices of individuals by the logical variables used in its definition. A set of PFs is therefore a compact description of a PGM for every possible grounding.

Markov logic networks (MLNs) (Richardson and Domingos 2006; Domingos et al. 2008) is a commonly used framework for defining relational probabilistic models. MLNs use binary PRVs assuming the values "false" and "true" (we use 0 and 1), and express PFs using weighted logic formulae with log-probability notation. For example, a weighted MLN formula $\langle w, A(x) \vee A(y) \rangle$ represents the PF $f\big(A(x), A(y)\big) = e^{w \mathbb{I}_{A(x) \vee A(y)}}$, where $\mathbb{I}$ is the indicator function: $\mathbb{I}_{false} = 0$, $\mathbb{I}_{true} = 1$. In order to represent hard constraints (zero probabilities), formulae with infinite weights are sometimes allowed. A weighted MLN formula $\langle \infty, \psi \rangle$ (or, more precisely, $\langle -\infty, \neg\psi \rangle$) is equivalent to a PF defined by the indicator function $\mathbb{I}_{\psi}$. MLN formulae that contain quantifiers, e.g. $\exists x A(x)$, or the equality operator between logical variables, cannot readily be expressed using PFs as defined here. Note that normalization is defined per grounding, so Z depends on the population sizes.

Definition 2. *Consider a relational probabilistic model $\mathbb{M}_0$ with PRVs $\mathbf{X}$ which we alter resulting in $\mathbb{M}_1$ with PRVs $\mathbf{X} \cup \{X'\}$. If, for every grounding, the distribution over $\mathbb{M}_0$'s ground variables remains the same, then we call this alteration an **addition of a dependent PRV** X'.*

2.3 Aggregators

Directed relational models give rise to **aggregators**: random variables whose set of parents in the grounding depends on population sizes, and so may be unbounded. The simplest such case is an unparametrized (i.e., grounded to a single variable) binary PRV B that depends on a single binary PRV $A(x)$. Surprisingly, we obtained strong negative results even for this simple case, and therefore have not considered more complex ones. We use n for the relevant population size, $A_1, \ldots, A_n$ or $A_{1..n}$ for the ground variables, and n_0 and n_1 for the **counts**, i.e., the number of variables A_i assigned 0 and 1 respectively, with $n = n_0 + n_1$. Due to exchangeability (the identities of the individuals do not matter), the distribution over B in the grounding only depends on the counts, so we may define aggregators as functions from counts to distributions over B: $\mathrm{P}(B \mid A_{1..n}) = \mathrm{P}(B \mid n_0, n_1)$.

Several such unbounded-sized aggregators have gained popularity in the literature. A few common ones for binary variables are listed in Table 1 (left & center). Deterministic AND and OR correspond to noisy AND and OR with $\alpha = 0$ (taking $0^0 = 1$). A "random mux"[1] (or "average"),

[1] A multiplexer, or mux, is an electronic device that receives n inputs and a $\log_2 n$-bit "select" signal. The input indexed by "select" is sent as the mux's output. We call this aggregator a "random

Table 1: Our Results for Common Binary Aggregators.

	Aggregator	$\mathrm{P}(B = 1 \mid n_0, n_1)$	Parameters	Thm 2	Thm 4
Deterministic:	AND	$\mathbb{I}_{n_0=0}$		no	yes
	OR	$\mathbb{I}_{n_1\geq 1}$		no	yes
	At least t 1's	$\mathbb{I}_{n_1\geq t}$	$t \geq 0$	no	yes
	XOR	$\frac{1+(-1)^{n_1}}{2}$		no	no
	At least t% 1's	$\mathbb{I}_{\ n_1/(n_0+n_1)\ \geq\ t}$	$0 < t < 1$	no	no
"Hybrid":	Majority	$\frac{1}{2}\mathbb{I}_{n_1=n_0} + \mathbb{I}_{n_1>n_0}$		no	no
Probabilistic:	Noisy AND	α^{n_0}	$\alpha \in [0,1]$	no	approximated
	Noisy OR (Pearl 1988, §4.3.2)	$1-\alpha^{n_1}$	$\alpha \in [0,1]$	no	approximated
	Random mux (or "average")	$\frac{n_1}{n_0+n_1}$		no	no
	Logistic regression	$1/\left(1+e^{-(w+w_0 n_0+w_1 n_1)}\right)$	w, w_0, w_1	no	no(approx if $w_0 w_1 \geq 0$)
	Relational logistic regr. (RLR)	$1/\left(1+e^{-\sum_{\langle L,F,w\rangle} w \sum_L F_{\Pi,x\to X}}\right)$	$\{\langle L, F, w\rangle\}$	no	no(approx if ...)

can be described as a variable that chooses one parent by random and copies its value. Relational logistic regression (RLR) (Kazemi et al. 2014) extends logistic regression to multiple parents and complex interactions.

Our results make heavy use of "saturated" aggregators:

Definition 3. *An aggregator* $\mathrm{P}(B \mid A_{1..n})$ *is* ***S-saturated*** *for an integer* $S \geq 0$ *if, for any given* n, $\mathrm{P}(B \mid A_{1..n})$ *is equal for all assignments for which* $n_0, n_1 \geq S$. *An aggregator is* ***saturated*** *if it is* S*-saturated for some* S.

In other words, when B has S parents that are 0 and S that are 1, it is indifferent to the values of the rest. (It may, however, depend on the population's *size*.) An aggregator is S-saturated if and only if it can be written as:

$$\mathrm{P}(B \mid A_{1..n}) = \mathrm{P}(B \mid \min(n_0, S),\ \min(n_1, S),\ n).$$

2.4 Literature on Aggregators in Undirected Models

We set out to explore, how a new aggregator PRV B may be added to an existing MLN as a dependent PRV, i.e., *without altering the original distribution*. Table 2 describes how our results relate to the literature. A notable positive result was given by Natarajan et al. (2010), who showed how a class of aggregators may be represented using MLNs. In order to represent aggregators, they introduced $\Theta(n)$ auxiliary variables, and used a few special constructs: hard constraints, expressed using infinite weights; the equality operator ("="); and existentially quantified formulae (e.g., $\forall x A(x)$ and $B \wedge \exists x \exists y (x \neq y) \wedge A(x) \wedge A(y)$).

It was not at all clear whether it was mandatory to use all of these constructs to model dependent aggregators. In particular, the auxiliary variables are likely to considerably slow down inference (Natarajan et al. 2010). We thus adopted the same setting, but without introducing auxiliary variables. We found (Theorem 4) that without introducing auxiliary variables, only saturated aggregators can be modeled. Given "enough" 0s and "enough" 1s among $A_{1..n}$, a dependent variable B becomes indifferent to the values of the rest. We thus characterized which aggregators can be modeled without auxiliary variables, which can be approximated arbitrarily well, and which cannot even be approximated. This is a strong negative result, implying that despite the very flexible model, only a restricted form of aggregators can potentially be modeled or approximated, thus pointing out the cruciality of auxiliary variables to the modeling, and characterizing which aggregators require them.

We also investigated the same setting, but when also prohibiting quantifiers. We discovered (Theorem 2) that this only allows "0-saturated" aggregators, where B is independent of the other variables in all possible ground models, but $\mathrm{P}(B)$ may be a different constant in different groundings. This is a strong negative result, since in practice aggregators are used to represent a dependency on the values of their parents, and none can be modeled in this setting. This is surprising, as the setting is still very flexible, allowing for an arbitrarily large number of formulae, arbitrarily large number of free variables, "=", and hard constraints.

Poole et al. (2012) looked at a simpler setting, and showed that no aggregators could be represented in MLNs where formulae only involve a single instance of $A(x)$, i.e., without pairwise factors among the parents. We show aggregators cannot be represented even with pairwise factors, and even with factors over arbitrarily many variables (Theorem 2).

Empirical work on the non-relational case has demonstrated benefit in modeling interactions among more than two variables in undirected models (Dahinden et al. 2007; Schmidt and Murphy 2010; Buchman et al. 2012).

Relational logistic regression (RLR) (Kazemi et al. 2014) can be described as the directed counterpart of MLNs. RLR's definition prevents "side effects" when adding variables, making it easier to model aggregators. Recently, Poole et al. (2014) examined the dependency of variables on population size and model parameters for simple relational models, both undirected (MLN) and directed (RLR).

Jain, Barthels, and Beetz (2010) defined "adaptive MLNs" (AMLNs) as models extending MLNs by allowing formula weights to depend on attributes of the grounding, such as population sizes. They did not address the question of representing aggregators as dependent variables in these more flexible models, which is still an open problem.

3 DI-MLNs: A Normal Form for MLNs

Two of our main results, Theorems 2 and 4, are negative results. Proving them requires proving a property ("cannot

mux" because it can be seen as a mux with a uniform distribution over "select."

Table 2: Our Results vs. the Literature.

Paper:		Poole et al. (2012)	Thm 2	Thm 4	Natarajan et al. (2010)
Model:	Basic Model:	MLN	MLN	MLN	MLN
	Population size:	unbounded	unbounded	unbounded	unbounded
	Hard constraints allowed?	yes	yes	yes	yes
	# free vars per formula:	$\mathbf{0-1}$	$0-\infty$	$0-\infty$	$0-\infty$
	"=" allowed?	**no** (1 lo. var)	yes	yes	yes
	$\exists$, $\forall$ allowed?	**no**	**no**	**yes**	**yes**
	Weights depend on n?	no	no	no	no
Auxiliary vars:		0	0	0	$\boldsymbol{\Theta(n)}$
Summary:	Maximal factor scope:	$\{B, A_i\}$	**fixed, arbitrarily large**	$\{B, A_{1..n}\}$	$\{B, A_{1..n}$, **aux vars**$\}$
	Flexibility:	**very limited flex.**	**flexible**	**highly flexible**	**most flexible**
Result:	Result type[1]:	negative	negative	negative	**positive**
	Which aggregators can be added as dependent PRVs?	none[2]	**none**[2]	**only saturated aggregators**	at least some

[1] A positive result (e.g., "some") is stronger when allowing less flexibility. A negative result (e.g., "none" or "only saturated") is stronger when allowing more flexibility.
[2] Some dependencies on n itself, but not on $A_{1..n}$, can be modeled.

represent...") for all possible MLNs. This is very difficult, as MLNs are arbitrarily large weighted sets of arbitrarily nested first-order logic formulae. To our best knowledge, our results are the first that manage to prove negative representation results for general MLNs.

To tackle the complexity of MLNs, we first suggest a simpler, yet equivalent, model. We reduce semantic complexity by having all logical variables be different (thus proofs need to check fewer cases). We reduce syntactic complexity by avoiding the (now unneeded) "=" among logical variables.

We use $v(\varphi)$ for φ's free variables, and $\varphi_{\psi\to\psi'}$ for φ after replacing all occurrences of ψ with ψ'. Note that $x \neq y$ can be seen as short for $\neg(x = y)$, and $\forall x$ as short for $\neg\exists x\neg$.

Definition 4. *We define the **distinct-individual quantifiers (DI-quantifiers)** $\exists_{\neq}$ **("exists another")** and $\forall_{\neq}$ **("for all other")** similarly to $\exists$ and $\forall$, with the additional requirement that the logical variable be different from all other logical variables in its scope, i.e., $\exists_{\neq}x\gamma := \exists x\big(\gamma \wedge \bigwedge_{y\in Y}(x \neq y)\big)$ and $\forall_{\neq}x\gamma \;:=\; \neg\exists_{\neq}x\neg\gamma \;\equiv\; \forall x\big(\big(\bigwedge_{y\in Y}(x \neq y)\big) \to \gamma\big)$, where Y represents the variables accessible in x's scope.*

Definition 5. *A **Distinct-Individuals formula (DI-formula)** is a formula with no $\exists$, $\forall$ or "=", but which may contain $\exists_{\neq}$ and $\forall_{\neq}$.*

An unquantified DI-formula is therefore an unquantified formula with no "=".

Definition 6. *A **Distinct-Individuals MLN (DI-MLN)** is a model similar to MLNs except for: **(1)** Its formulae are DI-formulae. **(2)** Its formulae are only instantiated when all free variables are assigned distinct individuals.*

Theorem 1. ***(1)** MLNs and DI-MLNs are equally expressive. **(2)** Unquantified MLNs and unquantified DI-MLNs are equally expressive.*

Proof. $\forall$ and $\forall_{\neq}$ can be seen as syntactic sugar. $\exists$ and $\exists_{\neq}$ are interchangeable: each occurrence of $\exists_{\neq}x\gamma$ can be replaced by $\exists x\big(\gamma \wedge \bigwedge_{y\in Y}(x \neq y)\big)$, and each occurrence of $\exists x\gamma$ can be replaced by $(\exists_{\neq}x\gamma) \vee \bigvee_{y\in Y}\gamma_{x\to y}$.

When converting a DI-MLN to an equivalent MLN, convert each $\exists_{\neq}$ to a $\exists$, and then replace each weighted formula $\langle w, \varphi\rangle$ with $\langle\ w\ ,\ \varphi \wedge \bigwedge_{u,v\in v(\varphi)}(u \neq v)\ \rangle$. This prevents φ from having an effect when instantiated with some free variables being equal.

When converting an MLN to an equivalent DI-MLN, convert each $\exists$ to $\exists_{\neq}$, and then, for every quantified variable x, replace all occurrences of $x = \cdot$ and $\cdot = x$ with $false$. Consider a weighted formula $\langle w, \varphi\rangle$. "=" now only appears in φ among free variables. Every instantiation of φ defines a partition among $v(\varphi)$, in which x and y are in the same partition iff $x = y$. We can remove $\langle w, \varphi\rangle$ and, for every partition p, add a new weighted formula $\langle w, \phi_p \wedge \varphi\rangle$, where ϕ_p is the formula representing the partition. (For example, if $v(\varphi) = \{x, y, z\}$, the 5 possible partitions can be represented by: $x \neq y \neq z \neq x$, $x = y \neq z$, $x \neq y = z$, $y \neq z = x$, $x = y = z$.) Each formula $\phi_p \wedge \varphi$ can be simplified. First, all occurrences of "=" in φ can be evaluated according to ϕ_p. Second, whenever ϕ_p determines that two free variables are equal, one can be substituted for the other, e.g. $(x \neq y = z) \wedge (A_x \vee A_y \vee A_z)$ is replaced with $(x \neq y = y) \wedge (A_x \vee A_y \vee A_y)$. The remaining free variables are all distinct, so the $\phi_p\wedge$ prefix can be dropped, because a DI-MLN only instantiates formulae when all free variables are distinct. The remaining part of the formula contains no "=". □

4 Aggregators in Unquantified MLNs

Consider adding B as a dependent PRV to a relational model. In the grounding, the size of the factor $\mathrm{P}(B \mid A_{1..n})$ increases with population size. Many common undirected relational models, such as unquantified MLNs, can only model fixed-sized factors. However, Example 1 hints it might be possible to add dependent variables by decomposing $\mathrm{P}(B \mid A_{1..n})$ to smaller, fixed-sized (parametrized) factors. Surprisingly, this cannot be done:

Theorem 2. *Consider a positive MLN involving a single PRV $A(x)$, and assume that adding a new unparametrized*

variable B with a set $\mathbb{F}$ of weighted unquantified *formulae adds B as a dependent PRV. Then B is independent of its parents, $A_{1..n}$, in all possible groundings.*

Therefore, no "useful" aggregator (e.g., none of Table 1) can be added as a dependent PRV. The theorem is proved at the end of this section.

Note that $\mathrm{P}(B)$ may still be different in different groundings. B cannot depend on the values of the parents, but it can depend on their *number*. This degenerate form of aggregator (this is a 0-saturated aggregator) can be expressed as:

$$\mathrm{P}(B \mid A_{1..n}) = \mathrm{P}(B \mid n).$$

Theorems 2 and 4 require the original MLN to be positive. This is because we want the aggregator's representation, $\mathbb{F}$, to correctly model the aggregator on all possible inputs.

Example 2. *The MLN $\{\langle -\infty, A(x)\rangle\}$ only assigns positive probability to the configuration $(A_1 = 0, \ldots, A_n = 0)$. An aggregator then only needs to be correctly modeled for this single configuration. XOR can then be represented by $\mathbb{F} = \{\langle -\infty, B\rangle\}$, which represents $\mathrm{P}(B = 0) = 1$.*

Such "representations" are not useful, which is why we require all input configurations to have positive probability.

Note that $\mathbb{F}$ *may* contain hard constraints, which are needed for representing deterministic aggregators.

4.1 A Parametric Representation

In order to prove negative properties for MLNs, we first need to represent them in a simplified form. Consider an *unquantified* MLN $\mathbb{M}$. Theorem 1 implies we can treat $\mathbb{M}$ as an unquantified DI-MLN, thus simplifying both its syntax (no "=") and semantics. The second step is to cast a DI-MLN $\mathbb{M}$ into a simplified, parametric form.

In any grounding, $\mathbb{M}$ defines a set $\mathcal{F}$ of grounded factors, and the unnormalized probability is defined as their product.

Consider a weighted unquantified DI-formula $\langle w, \varphi\rangle \in \mathbb{M}$ with k free (logical) variables $y_1, \ldots, y_k$. Each free variable y_j refers to one of the n individuals: $y_j \in \{1, \ldots, n\}$. The formula φ can be represented as a non-negative PF $f_k(B, A_{y_1}, \ldots, A_{y_k})$ that is instantiated once for every *distinct* assignment of k individuals to $y_1, \ldots, y_k$. φ is thus instantiated $\binom{n}{k}$ times (with $\binom{n}{k} = 0$ for $k > n$). A hard constraint is represented as a PF that includes zeros. We may multiply PFs with the same value of k, thus $\mathbb{M}$ can be represented as a finite set of PFs: $\{f_0, \ldots, f_K\}$. (We cannot multiply PFs with different k's, because of the way k affects the number of factor instantiations, $\binom{n}{k}$.)

Consider $f_2(B, A_{y_1}, A_{y_2})$. Both $f_2(B, 0, 1)$ and $f_2(B, 1, 0)$ are raised to the same power in $\mathrm{P}(A_{1..n}, B)$. The individual values of $f_2(B, 0, 1)$ and $f_2(B, 1, 0)$ do not matter; only their product matters. We may therefore set $f_2(B, 0, 1)$ to their product, and set $f_2(B, 1, 0) = 1$, thus practically eliminating it from the model. In general, it is enough to only model a PF f_k for entries with the form $f_k(B, \underbrace{0, \ldots, 0}_{k_0}, \underbrace{1, \ldots, 1}_{k_1})$. Denoting the numbers of consecutive 0s and 1s by k_0 and k_1, we represent this entry by the parameter $\theta_{Bk_0k_1}$. f_k can thus be fully specified by the set of parameters $\{\theta_{Bk_0k_1} \geq 0 : B \in \{0,1\}, k_0, k_1 \in \{0, 1, \ldots, k\}, k_0 + k_1 = k\}$, and $\mathbb{M}$, represented as $\{f_0, \ldots, f_K\}$, can be fully specified by $\{\theta_{Bk_0k_1} \geq 0 : B \in \{0,1\}, k_0, k_1 \in \{0, 1, \ldots, K\}, k_0 + k_1 \leq K\}$.

Consider a specific assignment to $A_{1..n}$ and B. Each parameter $\theta_{Bk_0k_1}$ is used in the PF f_k (where $k = k_0 + k_1$) which is instantiated $\binom{n}{k}$ times, but takes effect only in factor instantiations in which the first k_0 of the A_i variables involved are assigned 0 and the other k_1 are assigned 1. There are $\binom{n_0}{k_0}\binom{n_1}{k_1}$ such factor instantiations, so

$$\prod_{f \in \mathcal{F}} f(A_{1..n}, B) = \prod_{k_0=0}^{K} \prod_{k_1=0}^{K-k_0} \theta_{Bk_0k_1}^{\left(\binom{n_0}{k_0}\binom{n_1}{k_1}\right)} \geq 0 \quad (1)$$

4.2 Proofs

Lemma 1. *Consider adding a new unparametrized variable B with a set $\mathbb{F}$ of weighted unquantified formulae to a positive MLN involving a single PRV $A(x)$. In any grounding of the MLN, $\mathbb{F}$ results in an addition of a set $\mathcal{F}_\mathbb{F}$ of grounded factors. If we marginalize out B, we would get the factor $g(A_{1..n})$ coupling $A_{1..n}$:*

$$g(A_{1..n}) := \sum_{B \in \{0,1\}} \prod_{f \in \mathcal{F}_\mathbb{F}} f(A_{1..n}, B) \quad (2)$$

Let $S \geq 0$ be an arbitrary constant. Assume, for any n, $g(A_{1..n})$ is equal for all assignments $A_{1..n}$ for which $n_0, n_1 \geq S$, and is positive:

$$\forall n, \forall A_1, \ldots, A_n, A'_1, \ldots, A'_n \text{ such that } n_0, n_1, n'_0, n'_1 \geq S,$$
$$g(A_{1..n}) = g(A'_{1..n}) > 0 \quad (3)$$

Then $\mathrm{P}(B \mid A_{1..n})$ is S-saturated.

Proof. The proof only needs to analyze $\mathbb{F}$, which represents the aggregator, and not the initial or final MLNs. An MLN is a weighted set of logic formulae; and $\mathbb{F}$, which is a weighted set of logic formulae, can be treated as an MLN. (If the initial MLN contains no formulae, then the final MLN is $\mathbb{F}$.) We now treat $\mathbb{F}$ as an unquantified MLN. The rest of the proof works out the consequences of assumption (3).

Assuming (3), we can name this positive constant $g(n)$:

$$g(n) := g(A_{1..n}) \quad (\text{for any } A_{1..n} \text{ such that } n_0, n_1 \geq S) \quad (4)$$

and using (2) and the parametric representation (1) for $\mathbb{F}$:

$$U_B(n_0, n_1) := \prod_{f \in \mathcal{F}_\mathbb{F}} f(A_{1..n}, B) = \prod_{k_0=0}^{K} \prod_{k_1=0}^{K-k_0} \theta_{Bk_0k_1}^{\left(\binom{n_0}{k_0}\binom{n_1}{k_1}\right)} \geq 0$$

$$g(n) = \sum_{B \in \{0,1\}} U_B(n_0, n_1) \qquad \text{for } n_0, n_1 \geq S \quad (5)$$

Note that the sum, $\sum_B U_B(n_0, n_1)$, does not depend on n_0 and n_1, rather only on $n = n_0 + n_1$.

An Infinite-Limit Proof Technique An MLN can be seen as an infinite set of grounded graphical models (for different population sizes) with tied weights. We found the weight-tying is a constraint that can be exploited analytically, by taking limits of $n \to \infty$. The proofs of Lemmas 2 and 3, which are in the supplemental appendix available from the authors' web sites, are based on this core idea.

Using Limits Consider dividing a PF f_k by a positive constant c_k, i.e. $\theta^{new}_{Bk_0k_1} \leftarrow \theta^{old}_{Bk_0k_1}/c_k$ for all parameters $\theta_{Bk_0k_1}$ for which $k_0 + k_1 = k$. This division does not change the MLN's distribution; it only changes the normalizing term $Z(n)$ and it changes $g(n)$.

Lemma 2. *There exist constants $c_0, \ldots, c_K > 0$ such that dividing each f_k by c_k leads to:*

$$\lim_{n\to\infty} g(n) = 1 \tag{6}$$

Lemma 3. *Assume the factors were scaled according to Lemma 2, so* (6) *holds. Define a sequence of population sizes and assignments to $A_{1..n}$:*

$$n_0(t) = 2^t, \quad n_1(t) = t \quad \longrightarrow \quad n(t) = 2^t + t$$

and define $U_B(t) := U_B(\, n_0(t),\, n_1(t)\,)$. Then, for any $b \in \{0,1\}$, either $U_b(t) = \theta_{b00}$ or $\lim_{t\to\infty} U_b(t) = 0$.

Assume the factors were scaled according to Lemma 2, so (6) holds. Writing (5) for the configurations defined by t, i.e. $n_0(t)$ and $n_1(t)$, gives $U_0(t) + U_1(t) = g(n(t)) = g(2^t + t)$. Taking the limit and using (6) gives $\lim_{t\to\infty}\big(U_0(t) + U_1(t)\big) = 1$. Using Lemma 3, and since $\lim_{t\to\infty} U_0(t)$ and $\lim_{t\to\infty} U_1(t)$ cannot simultaneously be 0, at least one must be a positive constant: $\exists b \in \{0,1\}, U_b(n_0, n_1) = \theta_{b00} > 0$. Factors from the original MLN do not involve B, so $\mathrm{P}(B \mid A_{1..n})$ is proportional to the product of the factors in $\mathcal{F}_{\mathbb{F}}$: $\mathrm{P}(B \mid A_{1..n}) \propto \prod_{f\in\mathcal{F}_{\mathbb{F}}} f(A_{1..n}, B) = U_B(n_0, n_1)$. So, for $n_0, n_1 \geq S$:

$$\mathrm{P}(B = b \mid A_{1..n}) = \frac{U_b(n_0, n_1)}{U_0(n_0, n_1) + U_1(n_0, n_1)} = \frac{\theta_{b00}}{g(n)}$$

B therefore does not depend on $A_{1..n}$ when $n_0, n_1 \geq S$, so $\mathrm{P}(B \mid A_{1..n})$ is S-saturated. □

Proof of Theorem 2. Since B is added as a dependent variable, if we were to marginalize B out, we would not get a factor coupling $A_{1..n}$ (here we implicitly assumed the original MLN is positive). In other words, $g(A_{1..n})$ must be a positive constant ($g(A_{1..n})$ may depend on n – but not on $A_{1..n}$). Lemma 1 with $S = 0$ gives that $\mathrm{P}(B \mid A_{1..n})$ is 0-saturated, therefore B is independent of $A_{1..n}$. □

5 Aggregators in Quantified MLNs

Quantifiers can create factors over all ground variables. Surprisingly, quantifiers, even when nested, only increase expressiveness in a very limited manner, by only allowing "exceptions" when $n_0 <$ threshold or $n_1 <$ threshold:

Theorem 3. *For every DI-formula φ using the PRVs $\{A(x), B\}$ there is an unquantified DI-formula φ_{unq} and an integer S_φ, such that $\varphi_{unq} \equiv \varphi$ whenever $n_0, n_1 \geq S_\varphi$.*

Theorem 3 can be extended to regular (non-DI) formulae.

Example 3. $\big(\forall_{\neq} x A(x)\big)_{unq} \equiv false$ *and* $S_{(\forall_{\neq} xA(x))} = 1$, *because* $(\forall_{\neq} xA(x)) \equiv false$ *whenever* $n_0, n_1 \geq 1$.

An additional example is in the supplemental appendix.

Proof. We prove by induction on the structure of φ. For $\varphi \in \{true, false, B, A(x)\}$, we may pick $\varphi_{unq} = \varphi$ with $S_\varphi = 0$. We may also pick $(\neg\varphi)_{unq} = \neg(\varphi_{unq})$ with $S_{\neg\varphi} = S_\varphi$, $(\varphi \vee \psi)_{unq} = (\varphi_{unq}) \vee (\psi_{unq})$ with $S_{\varphi\vee\psi} = \max(S_\varphi, S_\psi)$, and similarly for $\varphi \wedge \psi$. $\forall_{\neq} x\varphi$ may be treated as $\neg\exists_{\neq} x\neg\varphi$.

Finally, pick $(\exists_{\neq} x\varphi)_{unq} = (\varphi_{A(x)\to true})_{unq} \vee (\varphi_{A(x)\to false})_{unq}$ and $S_{(\exists_{\neq} x\varphi)} = \max(S_\varphi, |v(\varphi)| + 1)$. x appears in φ only as part of $A(x)$, so $\exists_{\neq} x\varphi \equiv \exists_{\neq} x\big(\big(A(x) \wedge \varphi_{A(x)\to true}\big) \vee \big(\neg A(x) \wedge \varphi_{A(x)\to false}\big)\big)$. Assume $n_0, n_1 \geq S_{(\exists_{\neq} x\varphi)}$. Since $n_0, n_1 \geq |v(\varphi)| + 1$, there is at least one individual not bound by $v(\varphi)$ for which $A(x) = 1$, and at least one for which $A(x) = 0$, so $\exists_{\neq} x\varphi \equiv \varphi_{A(x)\to true} \vee \varphi_{A(x)\to false}$. And since $n_0, n_1 \geq S_\varphi$, $\exists_{\neq} x\varphi \equiv (\varphi_{A(x)\to true})_{unq} \vee (\varphi_{A(x)\to false})_{unq} \equiv (\exists_{\neq} x\varphi)_{unq}$. □

We now extend Theorem 2 to support quantifiers in $\mathbb{F}$:

Theorem 4. *Consider a positive MLN involving a single PRV $A(x)$, and assume that adding a new unparametrized variable B with a set $\mathbb{F}$ of weighted formulae adds B as a dependent PRV. Then $\mathrm{P}(B \,|A_{1..n})$ is a saturated aggregator.*

Theorem 4 shows that even in a very general and flexible setting, only a special class ("saturated") of aggregators can be added as dependent PRVs. However, some aggregators may be approximated to arbitrary precision by saturated aggregators with sufficiently high values for S. (This may, however, necessitate formulae with deeper nested existential quantifiers, as can be hinted by the proof of the theorem.) The rightmost columns of Table 1 show what our results imply for the aggregators: whether they can be added as dependent PRVs, can be approximated with arbitrary precision, or cannot even be approximated.

Example 4. *Random mux cannot be approximated by a saturated aggregator. Any given saturated aggregator is S-saturated for some S. For $n = 10S$, $\mathrm{P}_{r.mux}(B{=}1 \mid n_0{=}9S, n_1{=}S) = 0.1$ and $\mathrm{P}_{r.mux}(B{=}1 \mid n_0{=}S, n_1{=}9S) = 0.9$, but the given aggregator cannot distinguish between these two inputs.*

Proof of Theorem 4. We start similarly to the proof of Theorem 2. B is added as a dependent variable, so if we marginalize it out, we would not get a factor coupling $A_{1..n}$. $g(A_{1..n})$ must therefore be a positive constant (which may depend on n). However, we cannot use Lemma 1, as it relies on $\mathbb{F}$ being unquantified in order to derive the parametric representation.

Form $\mathbb{F}_{unq}$ by replacing every formula $\varphi \in \mathbb{F}$ with the unquantified formula φ_{unq} guaranteed by Theorem 3, and let $S_{\mathbb{F}} := \max_{\varphi\in\mathbb{F}} S_\varphi$. Let $\mathbb{M}_{\mathbb{F}}$ be the MLN with the addition of $\mathbb{F}$, and $\mathbb{M}_{\mathbb{F}_{unq}}$ be the MLN with the addition of $\mathbb{F}_{unq}$ instead. For any joint assignment in which $n_0, n_1 \geq S_{\mathbb{F}}$, $\mathbb{M}_{\mathbb{F}}$ and $\mathbb{M}_{\mathbb{F}_{unq}}$ are indistinguishable: their unnormalized joint probabilities are the same, and so are their $g(A_{1..n})$'s. Since, for any n, $g_{\mathbb{F}}(A_{1..n})$ is a positive constant, $g_{\mathbb{F}_{unq}}(A_{1..n})$ is the same positive constant for all assignments in which $n_0, n_1 \geq S_{\mathbb{F}}$. Applying Lemma 1 with $S = S_{\mathbb{F}}$ to $\mathbb{F}_{unq}$ tells us that $\mathrm{P}_{\mathbb{M}_{\mathbb{F}_{unq}}}(B \mid A_{1..n})$ is $S_{\mathbb{F}}$-saturated. $\mathbb{M}_{\mathbb{F}}$ is indistinguishable

from $\mathbb{M}_{\mathbb{F}_{unq}}$ when $n_0, n_1 \geq S_{\mathbb{F}}$, so $\mathrm{P}_{\mathbb{M}_{\mathbb{F}}}(B \mid A_{1..n})$ is also $S_{\mathbb{F}}$-saturated. □

Note that the conditions for applying Theorem 2 to $\mathbb{F}_{unq}$ are not met: while adding $\mathbb{F}$ adds B as a dependent PRV, there is no similar guarantee for $\mathbb{F}_{unq}$.

6 Relational Logistic Regression Aggregators

The mathematical tools we introduced also promise to facilitate further theoretical results.

As an example, (unquantified) RLR aggregators were found to represent a "sigmoid of a polynomial of counts" (Kazemi et al. 2014). Using our tools it is easy to determine that adding quantifiers to RLR only allows to define "exceptions" for $n_0 <$ threshold and $n_1 <$ threshold:

Theorem 5. *RLR with quantifiers for* $\mathrm{P}(B \mid A_{1..n})$ *represents a "sigmoid of a polynomial of counts" when* $n_0, n_1 \geq$ *(model-specific) constant threshold.*

Proof. Replace every RLR formula φ with the unquantified formula guaranteed by Theorem 3. The modified model is an unquantified RLR, representing a sigmoid of a polynomial of counts. For $n_0, n_1 \geq \max_\varphi S_\varphi$, the new RLR is identical to the original RLR. □

7 Conclusions

We have defined a conceptually simple extension method for probabilistic models, namely adding dependent variables, and pointed to situations where this method may be desirable. Our main results lie in the relational setting, where we investigated how aggregator variables may be added to Markov logic networks (MLNs) as dependent variables, without modifying the original distribution, only extending it, and without bounding the population size. We showed that without introducing auxiliary variables, which may slow down inference, only the limited class of "saturated" aggregators can be added. This result shows which aggregators are suitable for precise or approximate representation without auxiliary variables, and which are inherently unsuitable, informing the modeler they must either use auxiliary variables, or use a model more flexible than MLNs. We also considered avoiding quantified formulae as well, only to find that then aggregators cannot depend on the values of their parents at all. These are negative results about the representational power of MLNs.

While Markov models can represent any distribution (in the worst case, using a factor over all variables), this is not true in the relational case. Different models may have advantages for representing distributions based on different independence assumptions. We hope our results would stimulate future research on the choice of relational models appropriate for particular applications, and into the relative representational advantages of different relational models.

References

Buchman, D.; Schmidt, M. W.; Mohamed, S.; Poole, D.; and de Freitas, N. 2012. On sparse, spectral and other parameterizations of binary probabilistic models. *Journal of Machine Learning Research - Proceedings Track* 22:173–181.

Dahinden, C.; Parmigiani, G.; Emerick, M. C.; and Peter Bühlmann, P. 2007. Penalized likelihood for sparse contingency tables with an application to full-length cDNA libraries. *BMC Bioinformatics* 8:476.

De Raedt, L.; Frasconi, P.; Kersting, K.; and Muggleton, S. H., eds. 2008. *Probabilistic Inductive Logic Programming*. Springer.

Dechter, R. 1996. Bucket elimination: A unifying framework for probabilistic inference. In Horvitz, E., and Jensen, F., eds., *UAI-96*, 211–219.

Domingos, P.; Kok, S.; Lowd, D.; Poon, H.; Richardson, M.; and Singla, P. 2008. Markov logic. In Raedt, L. D.; Frasconi, P.; Kersting, K.; and Muggleton, S., eds., *Probabilistic Inductive Logic Programming*. New York: Springer. 92–117.

Getoor, L., and Taskar, B., eds. 2007. *Introduction to Statistical Relational Learning*. Cambridge, MA: MIT Press.

Jain, D.; Barthels, A.; and Beetz, M. 2010. Adaptive Markov logic networks: Learning statistical relational models with dynamic parameters. In *19th European Conference on Artificial Intelligence (ECAI)*, 937–942.

Kazemi, S. M.; Buchman, D.; Kersting, K.; Natarajan, S.; and Poole, D. 2014. Relational logistic regression. In *14th International Conference on Principles of Knowledge Representation and Reasoning (KR 2014)*.

Koller, D., and Friedman, N. 2009. *Probabilistic Graphical Models – Principles and Techniques*. MIT Press.

Natarajan, S.; Khot, T.; Lowd, D.; Tadepalli, P.; and Kersting, K. 2010. Exploiting causal independence in Markov logic networks: Combining undirected and directed models. In *European Conference on Machine Learning (ECML)*.

Pearl, J. 1988. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. San Mateo, CA: Morgan Kaufmann.

Poole, D.; Buchman, D.; Natarajan, S.; and Kersting, K. 2012. Aggregation and population growth: The relational logistic regression and Markov logic cases. In *Proc. UAI-2012 Workshop on Statistical Relational AI*.

Poole, D.; Buchman, D.; Kazemi, S. M.; Kersting, K.; and Natarajan, S. 2014. Population size extrapolation in relational probabilistic modelling. In *Scalable Uncertainty Management*. Springer. 292–305.

Richardson, M., and Domingos, P. 2006. Markov logic networks. *Machine Learning* 62:107–136.

Schmidt, M., and Murphy, K. P. 2010. Convex structure learning in log-linear models: Beyond pairwise potentials. *Artificial Intelligence and Statistics*.

Taskar, B.; Abbeel, P.; and Koller, D. 2002. Discriminative probabilistic models for relational data. In Darwiche, A., and Friedman, N., eds., *Proceedings of the Eighteenth Conference on Uncertainty in Artificial Intelligence (UAI-02)*, 485–492.

Zhang, N. L., and Poole, D. 1994. A simple approach to Bayesian network computations. In *Proc. 10th Canadian Conference on AI*, 171–178.