Relational Logistic Regression*

Seyed Mehran Kazemi, David Buchman, Kristian Kersting, Sriraam Natarajan, and David Poole
cs.ubc.ca/~smkazemi/ cs.ubc.ca/~davidbuc/
www-ai.cs.uni-dortmund.de/PERSONAL/kersting.html
homes.soic.indiana.edu/natarasr/ cs.ubc.ca/~poole/

Abstract

Logistic regression is a commonly used representation for aggregators in Bayesian belief networks when a child has multiple parents. In this paper we consider extending logistic regression to relational models, where we want to model varying populations and interactions among parents. In this paper, we first examine the representational problems caused by population variation. We show how these problems arise even in simple cases with a single parametrized parent, and propose a linear relational logistic regression which we show can represent arbitrary linear (in population size) decision thresholds, whereas the traditional logistic regression cannot. Then we examine representing interactions among the parents of a child node, and representing non-linear dependency on population size. We propose a multi-parent relational logistic regression which can represent interactions among parents and arbitrary polynomial decision thresholds. Finally, we show how other well-known aggregators can be represented using this relational logistic regression.

Introduction

Relational probabilistic models are models where there are probabilities about relations among individuals that can be specified independently of the actual individuals, and where the individuals are exchangeable; before we know anything about the individuals, they are treated identically. One of the features of relational probabilistic models is that the predictions of the model may depend on the number of individuals (the population size) (Poole et al. 2012). Sometimes, this dependence is desirable; in other cases, model weights may need to change (Jian, Bernhard, and Beetz 2007; Jian, Barthels, and Beetz 2009). In either case, it is important to understand how the predictions change with population size.

Varying population sizes are quite common. They can appear in a number of ways including:

- The actual population may be arbitrary. For example, in considering the probability of someone committing a crime (which depends on how many other people could have committed the crime) (Poole 2003) we could consider the population to be the population of the neighbourhood, the population of the city, the population of the country, or the population of the whole world. It would be good to have a model that does not depend on this arbitrary decision. We would like to be able to compare models which involve different choices.
- The population can change. For example, the number of people in a neighbourhood or in a school class may change. We would like a model to make reasonable predictions as the population changes. We would also like to be able to apply a model learned at one or a number of population sizes to different population sizes. For example, models from drug studies are acquired from very limited populations but are applied much more generally.
- The relevant populations can be different for each individual. For example, the happiness of a person may depend on how many of her friends are kind (and how many are not kind). The set of friends is different for each individual. We would like a model that makes reasonable predictions for diverse numbers of friends.

In this paper, we consider applying standard logistic regression to relational domains and tasks and investigate how varying populations can cause a problem for logistic regression. Then we propose *single-parent linear relational logistic regression* which solves this problem with standard logistic regression by taking the population growth into account. This representation is, however, only able to model linear function dependencies of the child on its parents' population sizes. Also when used for multiple parents, it cannot model the interactions among the parents. We examine these two limitations and propose a general relational logistic regression which we prove can represent arbitrary Boolean formulae among the parents as well as every polynomial dependency of the child node on its parents' population sizes. We also show how other well-known aggregators can be represented using our polynomial relational logistic regression.

Our model assumes all the parent variables are categorical and the child variable is Boolean. Extending the model to multi-valued child variables and continuous parent variables (as done by Mitchell (2010) for non-relational models) is left as a future work.

*This work was supported in part by the Institute for Computing Information and Cognitive Systems (ICICS) at UBC, NSERC, MITACS and the German Science Foundation (DFG), KE 1686/2-1.

Background

Bayesian Belief Networks

Suppose we have a set of random variables $\{X_1,\ldots,X_n\}$. A **Bayesian network (BN)** or **belief network** (Pearl 1988) is an acyclic directed graph where the random variables are the nodes, and the arcs represent interdependence between the random variables. Each variable is independent of its non-descendants given values for its parents. Thus, if X_i is not an ancestor of X_j, then $P(X_i \mid parents(X_i), X_j) = P(X_i \mid parents(X_i))$. The joint probability of the random variables can be factorized as:

$$P(X_1,X_2,...,X_n) = \prod_{i=1}^{n} P(X_i \mid parents(X_i))$$

One way to represent a conditional probability distribution $P(X_i \mid parents(X_i))$ is in terms of a table. Such a tabular representation for a random variable increases exponentially in size with the number of parents. For instance, a Boolean child having 10 Boolean parents requires $2^{10} = 1024$ numbers to specify the conditional probability. A compact alternative to a table is an **aggregation** operator, or aggregator, that specifies a function of how the distribution of a variable depends on the values of its parents. Examples for common aggregators include OR, AND, as well as "noisy-OR" and "noisy-AND". These can be specified much more compactly than as a table.

Logistic Regression

Suppose a Boolean random variable Q is a child of the numerical random variables $\{X_1,X_2,\ldots,X_n\}$. Logistic regression is an aggregation operator defined as:

$$P(q \mid X_1,\ldots,X_n) = \text{sigmoid}(w_0 + \sum_i w_i X_i) \qquad (1)$$

where $q \equiv$ "$Q = True$" and $\text{sigmoid}(x) = 1/(1+e^{-x})$. It follows that $P(q \mid X_1,\ldots,X_n) > 0.5$ iff $w_0 + \sum_i w_i X_i > 0$.

The space of assignments to the w's so that $w_0 + \sum_i w_i X_i = 0$ is called the **decision threshold**, as it is the boundary of where $P(q \mid X_1,\ldots,X_n)$ changes between being closer to 0 and being closer to 1. Logistic regression provides a soft threshold, in that it changes from close to 0 to close to 1 in a continuous manner. How fast it changes can be adjusted by multiplying all weights by a positive constant.

The Factorization Perspective

A simple and general formulation of logistic regression can be defined using a multiplicative factorization of the conditional probability. (1) then becomes a special case, which is equivalent to the general case when variables are binary and probabilities are positive (non-zero).

We define a **general logistic regression** for Q with parents $X_1,\ldots,X_n$ (all variables here may be discrete or continuous) to be when $P(Q \mid X_1,\ldots,X_n)$ can be factored into a product of non-negative pairwise factors and a non-negative factor for Q:

$$P(Q \mid X_1,\ldots,X_n) \propto f_0(Q) \prod_{i=1}^{n} f_i(Q,X_i)$$

where $\propto$ *(proportional-to)* means it is normalized separately for each assignment to the parents. This differs from the normalization for joint distributions (as used in undirected models), where there is a single normalizing constant. Here the constraint that causes the normalization is $\forall X_1,\ldots,X_n : \sum_Q P(Q \mid X_1,\ldots,X_n) = 1$, whereas for joint distributions, the normalization is to satisfy the constraint $\sum_{Q,X_1,\ldots,X_n} P(Q,X_1,\ldots,X_n) = 1$.

If Q is binary, then:

$$P(q \mid X_1,\ldots,X_n) = \frac{f_0(q)\prod_{i=1}^{n} f_i(q,X_i)}{f_0(q)\prod_{i=1}^{n} f_i(q,X_i) + f_0(\neg q)\prod_{i=1}^{n} f_i(\neg q,X_i)}$$

If all factors are positive, we can divide and then use the identity $y = e^{\ln y}$:

$$\begin{aligned} P(q \mid X_1,\ldots,X_n) &= \frac{1}{1 + \frac{f_0(\neg q)}{f_0(\ \ q)} \prod_{i=1}^{n} \frac{f_i(\neg q,X_i)}{f_i(\ \ q,X_i)}} \\ &= \frac{1}{1 + exp\left(\ln \frac{f_0(\neg q)}{f_0(\ \ q)} + \sum_{i=1}^{n} \ln \frac{f_i(\neg q,X_i)}{f_i(\ \ q,X_i)}\right)} \\ &= \text{sigmoid}\left(\ln \frac{f_0(\ \ q)}{f_0(\neg q)} + \sum_{i=1}^{n} \ln \frac{f_i(\ \ q,X_i)}{f_i(\neg q,X_i)}\right). \end{aligned}$$

When the $\ln \frac{f_i(\ \ q,X_i)}{f_i(\neg q,X_i)}$ are linear functions w.r.t. X_i, it is possible to find values for all w's such that this can be represented by Eq. (1). This is always possible when the parents are binary.

The idea of *relational* logistic regression is to extend logistic regression to relational models, by allowing weighted logical formulae to represent the factors in the factorization of a conditional probability.

Relational Models

Relational probabilistic models (Getoor and Taskar 2007) or template based models (Koller and Friedman 2009) extend Bayesian or Markov networks by adding the concepts of individuals (objects, entities, things), relations among individuals (including properties, which are relations of a single individual) and by allowing for probabilistic dependencies among these relations. In these models, individuals about which we have the same information are exchangeable, meaning that, given no evidence to distinguish them, they should be treated identically. We provide some basic definitions and terminologies in these models which are used in the rest of the paper.

A **population** is a set of **individuals**. A population corresponds to a domain in logic. The **population size** is the cardinality of the population which can be any non-negative integer.

A **logical variable** is written in lower case. Each logical variable is typed with a population; we use $|x|$ for the size of the population associated with a logical variable x. Constants, denoting individuals, start with an upper case letter.

A **parametrized random variable (PRV)** is of the form $F(t_1,\ldots,t_k)$ where F is a k-ary functor (a function symbol or a predicate) and each t_i is a logical variable or a constant. Each functor has a range, which is $\{True, False\}$ for predicate symbols. A PRV represents a set of random variables,

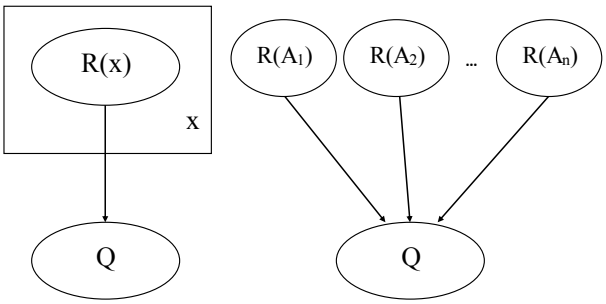

Figure 1: logistic regression (with i.i.d. priors for the $R(x)$). The left side is the relational model in plate notation and on the right is the groundings for the population $\{A_1, A_2, \ldots, A_n\}$.

one for each assignment of individuals to its logical variables. The range of the functor becomes the range of each random variable.

A **relational belief network** is an acyclic directed graph where the nodes are PRVs. A **grounding** of a relational belief network with respect to a population for each logical variable is a belief network created by replacing each PRV with the set of random variables it represents, while preserving the structure.

A **formula** is made up of assignments of values to PRVs with logical connectives. For a Boolean PRV $R(x)$, we represent $R(x) = \mathit{True}$ by $R(x)$ and $R(x) = \mathit{False}$ by $\neg R(x)$.

When using a single population, we write the population as $A_1 \ldots A_n$, where n is the population size, and use $R_1 \ldots R_n$ as short for $R(A_1) \ldots R(A_n)$. We also use n_{val} for the number of individuals x for which $R(x) = val$. When $R(x)$ is binary, we use the shortened $n_T = n_{\mathit{True}}$ and $n_F = n_{\mathit{False}}$.

Relational Logistic Regression

While aggregation is optional in non-relational models, it is necessary in directed relational models whenever the parent of a PRV contains extra logical variables. For example, suppose Boolean PRV Q is a child of the Boolean PRV $R(x)$, which contains an extra logical variable, x, as in Figure 1. In the grounding, Q is connected to n instances of $R(x)$, where n is the population size of x. For the model to be defined before n is known, it needs to be applicable for all values of n.

Common ways to aggregate the parents in relational domains, e.g. (Horsch and Poole 1990; Friedman et al. 1999; Neville et al. 2005; Perlish and Provost 2006; Kisynski and Poole 2009; Natarajan et al. 2010), include logical operators such as *OR*, *AND*, *noisy-OR*, *noisy-AND*, as well as ways to combine probabilities.

Logistic regression, as described above, may also be used for relational models. Since the individuals in a relational model are exchangeable, w_i must be identical for all parents R_i (this is known as parameter-sharing or weight-tying), so Eq. (1) becomes:

$$P(q \mid R_1, \ldots, R_n) = \text{sigmoid}(w_0 + w_1 \sum_i R_i). \qquad (2)$$

Consider what happens with a relational model when n is not fixed.

Example 1. Suppose we want to represent "Q is True if and only if R is True for 5 or more individuals", i.e., $q \equiv \left(|\{i : R_i = \mathit{True}\}| \geq 5\right)$ or $q \equiv (n_T \geq 5)$, using a logistic regression model $(P(q) \geq 0.5) \equiv (w_0 + w_1 \sum_i R_i \geq 0)$, which we fit for a population of 10. Consider what this model represents when the population size is 20.

If $R = \mathit{False}$ is represented by 0 and $R = \mathit{True}$ by 1, this model will have $Q = \mathit{True}$ when R is true for 5 or more individuals out of the 20. It is easy to see this, as $\sum_i R_i$ only depends on the number of individuals for which R is *True*.

However, if $R = \mathit{False}$ is represented by -1 and $R = \mathit{True}$ by 1, this model will have $Q = \mathit{True}$ when R is *True* for 10 or more individuals out of the 20. The sum $\sum_i R_i$ depends on how many more individuals have *R True* than have *R False*.

If $R = \mathit{True}$ is represented by 0 and $R = \mathit{False}$ by any other value, this model will have $Q = \mathit{True}$ when R is *True* for 15 or more individuals out of the 20. The sum $\sum_i R_i$ depends on how many individuals have *R False*.

While the choice of representation for *True* and *False* was arbitrary in the non-relational case, in the relational case different parametrizations can result in different decision thresholds as a function of the population. The following table gives five numerical representations for *False* and *True*, with corresponding parameter settings (w_0 and w_1), such that all regressions represent the same conditional distribution for $n = 10$. However, for $n = 20$, the predictions are different:

False	True	w_0	w_1	Prediction for n = 20
0	1	-4.5	1	$Q \equiv (n_T \geq 5)$
-1	1	0.5	0.5	$Q \equiv (n_T \geq 10)$
-1	0	5.5	1	$Q \equiv (n_T \geq 15)$
-1	2	$-\frac{7}{6}$	$\frac{1}{3}$	$Q \equiv (n_T \geq 8)$
1	2	-14.5	1	$Q \equiv (n_T \geq 0)$

The decision thresholds in all of these are linear functions of population size. It is straightforward to prove the following proposition:

Proposition 1. *Let $R = \mathit{False}$ be represented by the number α and $R = \mathit{True}$ by $\beta \neq \alpha$. Then, for fixed w_0 and w_1 (e.g., learned for one specific population size), the decision threshold for a population of size n is*

$$\frac{w_0}{w_1(\alpha - \beta)} + \frac{\alpha}{\alpha - \beta} n.$$

What is important about this proposition is that the way the decision threshold changes with the population size n, i.e., the coefficient $\frac{\alpha}{\alpha - \beta}$, does not depend on data (which affects the weights w_0 and w_1), but only on the arbitrary choice of the numerical representation of R.

Thus, Eq. (2) with a specific numeric representation of *True* and *False* is only able to model one of the dependencies

of how predictions depend on population size, and so cannot properly fit data that does not adhere to that dependence.

We need an additional degree of freedom to get a relational model that can model any linear dependency on n, regardless of the numerical representation chosen.

Definition 1. Let Q be a Boolean PRV with a single parent $R(x)$, where x is the set of logical variables in R that are not in Q (so we need to aggregate over x). A (single-parent, linear) **relational logistic regression (RLR)** for Q with parents $R(x)$ is of the form:

$$P(q \mid R(A_1),\ldots,R(A_n)) = \text{sigmoid}\left(w_0 + w_1 \sum_i R_i + w_2 \sum_i (1-R_i)\right) \quad (3)$$

where R_i is short for $R(A_i)$, and is treated as 1 when it is *True* and 0 when it is *False*. Note that $\sum_i R_i$ is the number of individuals for which R is *True* ($= n_T$) and $\sum_i (1-R_i)$ is the number of individuals for which R is *False* ($= n_F$).

An alternative but equivalent parametrization is:

$$P(q \mid R(A_1),\ldots,R(A_n)) = \text{sigmoid}(w_0 + w_2 \sum_i 1 + w_3 \sum_i R_i) \quad (4)$$

where 1 is a function that has value 1 for every individual, so $\sum_i 1 = n$. The mapping between these parametrizations is $w_3 = w_1 - w_2$; w_0 and w_2 are the same.

Proposition 2. *Let $R = False$ be represented by α and $R = True$ by $\beta \neq \alpha$. Then, for fixed w_0, w_2 and w_3 in Eq. (4), the decision threshold for a population of size n is*

$$\frac{w_0}{w_3(\alpha-\beta)} + \frac{\alpha + w_2/w_3}{\alpha-\beta} n.$$

Proposition 2 implies that the way the decision threshold in a single-parent linear RLR grows with the population size n, i.e. the coefficient $\frac{\alpha+w_2/w_3}{\alpha-\beta}$, depends on the weights. Moreover, for fixed α and β, any linear function of population can be modeled by varying the weights. This was not true for the traditional logistic regression.

For the rest of this paper, when we embed logical formulae in arithmetic expressions, we take *True* formulae to represent 1, and *False* formulae to represent 0. Thus $\sum_L F$ is the number of assignments to the variables L for which formula F is *True*.

Interactions Among Parents

The RLR proposed in Definition 1 can be extended to multiple (parametrized) parents by having a different pair of weights ((w_1, w_2) or (w_2, w_3)) for each parent PRV. This is similar to the non-relational logistic regression, where each parent has a (single) different weight. However, there are cases where we want to model the interactions among the parents.

Example 2. Suppose we want to model whether someone being happy depends on the number of their friends that are kind. We assume the variable $Happy(x)$ has as parents $Friend(y,x)$ and $Kind(y)$. Note that the number of friends for each person can be different.

Consider the following hypotheses:

(a) A person is happy as long as they have 5 or more friends who are kind.

$$happy(x) \equiv |\{y : Friend(y,x) \wedge Kind(y)\}| \geq 5$$

(b) A person is happy if half or more of their friends are kind.

$$happy(x) \equiv |\{y : Friend(y,x) \wedge Kind(y)\}| \geq |\{y : Friend(y,x) \wedge \neg Kind(y)\}|$$

(c) A person is happy as long as fewer than 5 of their friends are not kind.

$$happy(x) \equiv |\{y : Friend(y,x) \wedge \neg Kind(y)\}| < 5$$

These three hypotheses coincide for people with 10 friends, but make different predictions for people with 20 friends.

All three hypotheses are based on the interaction between the two parents. Linear RLR considers each parent separately from the others, and so cannot model these interactions without introducing new relations. In order to model such aggregators, we need to be able to count the number of instances of a formula that are *True* in an assignment to the parents. We can use the following extended RLR to model these cases:

$$P(happy(x) \mid \Pi) = \text{sigmoid}\Big(w_0 + w_1 \sum_y Friend(y,x) \wedge \quad Kind(y) + w_2 \sum_y Friend(y,x) \wedge \neg Kind(y) \Big) \quad (5)$$

where Π is a complete assignment of *friend* and *kind* to the individuals, and the right hand side is summing over the propositions in Π for each individual. To model each of the above three cases, we can set w_0, w_1, and w_2 in Eq. (5) as follows:

(a) Let $w_0 = -4.5$, $w_1 = 1$, $w_2 = 0$
(b) Let $w_0 = 0.5$, $w_1 = 1$, $w_2 = -1$
(c) Let $w_0 = 5.5$, $w_1 = 0$, $w_2 = -1$

Going from Eq. (3) to Eq. (4) allowed us to only model the positive cases in linear single-parent RLR. We can use a similar construction for more general cases:

Example 3. Suppose a PRV Q is a child of PRVs $R(x)$ and $S(x)$. We want to represent "Q is *True* if and only if there are more than t individuals for x for which $R(x) \wedge \neg S(x)$." As in Example 2, we need to count the number of instances of a formula that are *True* in an assignment to the parents. It turns out that in this case $R(x) \wedge S(x)$ is the only non-atomic formula required to model the interactions between the two parents, because other conjunctive interactions can be represented using this count as follows:

$$\sum_x R(x) \wedge \neg S(x) = \sum_x R(x) - \sum_x R(x) \wedge S(x)$$
$$\sum_x \neg R(x) \wedge S(x) = \sum_x S(x) - \sum_x R(x) \wedge S(x)$$
$$\sum_x \neg R(x) \wedge \neg S(x) = |x| - \sum_x R(x) - \sum_x S(x) + \sum_x R(x) \wedge S(x)$$

with $|x| = \sum_x True$.

Example 3 shows that the positive conjunction of the interacting parents is the only formula required to compute arbitrary conjunctions of the two parents. In more complicated cases, however, subtle changes to the representation may be required.

Example 4. Suppose a PRV Q is a child of PRVs $R(x,y)$ and $S(x,z)$. Suppose we want to represent "Q is *True* if and only if we have $R(x,y) \wedge \neg S(x,z)$ for more than 5 triples $\langle x,y,z \rangle$". If we only count the number of instances of $(R \wedge T)(x,y,z)$ that are *True* given the assignment to the parents and do the same as in Example 3, we would only count the number of pairs $\langle x,y \rangle$ for which $R(x,y)$ is *True*. However, we need the number of triples $\langle x,y,z \rangle$ for which $R(x,y)$ is *True*. We thus need to use $\sum_{x,y,z} R(x,y)$ as the number of assignments to x, y and z, for which $R(x,y)$ is *True*, as follows:

$$\sum_{x,y,z} R(x,y) \wedge \neg S(x,z) = \sum_{x,y,z} R(x,y) - \sum_{x,y,z} R(x,y) \wedge S(x,y,z)$$

So as part of the representation, we need to incude the set of logical variables, and not just a weighted formula.

Non-Linear Decision Thresholds

Examples 3 and 4 suggest how to model interactions among the parents. Now consider the case where the child PRV is a non-linear function of its parents' population sizes. For instance, if the individuals are the nodes in a dense graph, some properties of arcs grow with the square of the population of nodes.

Markov logic networks (MLNs) (Richardson and Domingos 2006; Domingos et al. 2008) define probability distributions over worlds (complete assignments to the ground model) in terms of weighted formulae. The probability of a world is proportional to the exponential of the sum of the weights of the instances of the formulae that are *True* in the world. The probability of any formula is obtained by summing over the worlds in which the formula is *True*. MLNs can also be adapted to define conditional distributions. The following example shows a case where a non-linear conditional distribution is modeled by MLNs.

Example 5. Consider the MLN for PRVs Q and $R(x)$, consisting of a single formula $Q \wedge R(x) \wedge R(y)$ with weight w, where y represents the same population as x. The probability of q given observations of $R(A_i)$ for all A_i has a quadratic decision threshold:

$$P(q \mid R(A_1), \ldots, R(A_n)) = \text{sigmoid}(w\, {n_T}^2).$$

A similar method can be used by RLR to model non-linear decision thresholds. Consider the following example:

Example 6. Suppose a PRV Q is a child of the PRV $R(x)$, and we want to represent "Q is *True* if and only if ${n_T}^2 > n_F$". This dependency can be represented using the single-parent linear RLR by introducing a new logical variable x' with the same population as x and treating $R(x')$ as if it were a separate parent of Q. Then we can use the interaction between $R(x)$ and $R(x')$ to represent the model in this example as:

$$\sum_{x,x'} R(x) \wedge R(x') - \sum_{x} \mathit{True} + \sum_{x} R(x).$$

General Relational Logistic Regression

The previous examples show the potential for using RLR as an aggregator for relational models. We need a language for representing aggregation in relational models in which we can address the problems mentioned. We propose a generalized form of RLR which works for multi-parent cases and can model polynomial decision thresholds.

Definition 2. A **weighted parent formula (WPF)** for a PRV $Q(x)$, where x is a set of logical variables, is a triple $\langle L, F, w \rangle$ where L is a set of logical variables for which $L \cap x = \{\}$, F is a Boolean formula of parent PRVs of Q such that each logical variable in F either appears in Q or is in L, and w is a weight. Only those logical variables that do not appear in Q are allowed to be substituted in F.

Definition 3. Let $Q(x)$ be a Boolean PRV with parents $R_i(x_i)$, where x_i is the set of logical variables in R_i. A (multi-parent, polynomial) **relational logistic regression (RLR)** for Q with parents $R_i(x_i)$ is defined using a set of WPFs as:

$$P(q(X) \mid \Pi) = \text{sigmoid}\Big(\sum_{\langle L,F,w \rangle} w \sum_{L} F_{\Pi, x \to X} \Big)$$

where Π represents the assigned values to parents of Q, X represents an assignment of an individual to each logical variable in x, and $F_{\Pi, x \to X}$ is formula F with each logical variable x in it being replaced according to X, and evaluated in Π. (The first summation is over the set of WPFs; the second summation is over the tuples of L. Note that $\sum_{\{\}}$ sums over a single instance.)

Since the logical variables that appear in $Q(x)$ are fixed in the formulae and not allowed to be substituted according to Definition 2, in the rest of the paper we only focus on those logical variables of the parents that do not appear in $Q(x)$.

The single-parent linear RLR (Definition 1) is a subset of Definition 3, because the terms of Eq. (4) can be modeled by WPFs:

- w_0 can be represented by $\langle \{\}, \mathit{True}, w_0 \rangle$
- $w_2 \sum_i 1$ can be represented by $\langle \{x\}, \mathit{True}, w_2 \rangle$
- $w_3 \sum_i R_i$ can be represented by $\langle \{x\}, R(x), w_3 \rangle$

RLR then sums these WPFs, resulting in:

$$\begin{aligned} P(q \mid \Pi) &= \text{sigmoid}\Big(w_0 \sum_{\{\}} \mathit{True} + w_1 \sum_{\{x\}} \mathit{True} + w_2 \sum_{\{x\}} R(x)\Big) \\ &= \text{sigmoid}\Big(w_0 + w_2 n + w_3 \sum_i R_i\Big). \end{aligned}$$

Example 7. Consider the problem introduced in Example 4. Using general RLR (Definition 3), we can model the conditional probability of Q using the following WPFs:

$$\langle \{\}, \mathit{True}, w_0 \rangle$$
$$\langle \{x,y,z\}, R(x,y) \wedge \neg S(y,z), w_1 \rangle$$

Or alternatively:

$$\langle \{\}, \mathit{True}, w_0 \rangle$$
$$\langle \{x,y,z\}, R(x,y), w_1 \rangle$$
$$\langle \{x,y,z\}, R(x,y) \wedge S(y,z), -w_1 \rangle$$

Canonical Forms for RLR

While in Definition 2 we allow for any Boolean formula of parents, we can prove that a positive conjunctive form is sufficient to model all the Boolean interactions among parents. A Boolean interaction is one that can be expressed in terms of logical connectives of values to the parents.

Proposition 3. *Let Q be a Boolean PRV with parents $R_i(x_i)$, where x_i is a set of logical variables in R_i which are not in Q. Using only positive conjunctive form formulae in the WPFs for Q, all Boolean interactions between the parents can be modeled by RLR.*

Proof. We show how to model every conjunctive form interaction between the parents. Other interactions (such as a disjunctive interaction) can be modeled by a set of conjunctives.

For a subset M of parents of Q, we prove by induction on the number of negations $j \leq |M|$, that every conjunctive form interaction between the parents having j negations can be modeled by a set of WPFs.

For $j = 0$, the formula F is in a positive conjunction form and the proposition holds (even if $M = \{\}$). Assume the proposition holds for $j < |M|$. For $j+1$, let $R_i(x_i)$ be one of the negated parents. Removing $\neg R_i(x_i)$ from the formula F gives a new formula F_1 with j negated parents. According to our assumption for j negations, there exists a set S_1 of WPFs that models F_1. Replacing $\neg R_i(x_i)$ in F by $R_i(x_i)$ gives a new formula F_2 with j negated parents, which can be modeled by some set S_2 of WPFs. Let S_1' represent the WPFs in S_1 where each set of logical variables L in each WPFs is replaced by $L \cup x_i$. F can then be modeled by combining S_1' and S_2 and negating the weights associated with S_2. The reason why this is correct can be seen in Example 4. □

Proposition 3 suggests using only positive conjunctive formuale in WPFs. Proposition 4 proves that positive disjunctive RLR has the same representational power as positive conjunctive RLR. Therefore, all propositions proved for positive conjunctive RLR in the rest of the paper also hold for positive disjunctive RLR.

Proposition 4. *A conditional distribution $P(Q \mid R_i(x_i))$ can be expressed by a positive disjunctive RLR if and only if it can be expressed by a positive conjunctive RLR.*

Proof. First, suppose $P(Q \mid R_i(x_i))$ can be expressed by a positive disjunctive RLR. We can write a disjunctive formula as a negated conjunctive formula. So we change all the disjunctive formulae in the WPFs for $P(Q \mid R_i(x_i))$ to negated conjunctive formulae. A negated conjunctive formula in WPF $\langle L, \neg F, w \rangle$ can be modeled by two conjunctive WPFs $\langle L, T, w \rangle$ and $\langle L, F, -w \rangle$. The latter WPF consists of negated parents but we know from Proposition 3 that we can model it by a set of positive conjunctive WPFs. Consequently, $P(Q \mid R_i(x_i))$ can be also expressed by a positive conjunctive RLR.

Now, suppose the conditional distribution can be expressed by a positive conjunctive RLR definition of $P(Q \mid R_i(x_i))$. While Proposition 3 is written for positive conjunctive RLR, it is straight forward to see that it also holds for negative conjunctive RLR. This means that we can express Q by WPFs having negative conjunctive formulae. We can represent each of these formulae in a negated positive disjunctive form. We also mentioned that a negated WPF $\langle L, \neg F, w \rangle$ can be expressed by two WPFs $\langle L, T, w \rangle$ and $\langle L, F, -w \rangle$. The former does not contain any parent and the latter is in positive disjunctive form. Consequently, $P(Q \mid R_i(x_i))$ can be also expressed by a positive disjunctive RLR. □

Buchman et al. (2012) looked at canonical representations for probability distributions with binary variables in the non-relational case. Our positive conjunctive canonical form corresponds to their "canonical parametrization" with a "reference state" *True* (i.e., in which all variables are assigned *True*), and our positive disjunctive canonical form has a connection to using a "reference state" *False*. Their "spectral representation" would correspond to a third positive canonical form for RLR, in terms of **XORs** (i.e., parity functions).

Polynomial Decision Thresholds

We can also model polynomial decision thresholds using RLR. The following example is a case where the child PRV depends on $|x|^2$.

Example 8. Suppose Q is a Boolean PRV with a parent $R(x)$, where x is a set of logical variables in R which are not in Q. By having a WPF $\langle \{x, x'\}, R(x) \wedge R(x'), w \rangle$ for Q where x' is typed with the same population as x, the conditional probability of Q depends on the square of the number of assignments to x for which $R(x)$ is *True*.

Example 8 represents a case where the conditional probability of a child PRV is a non-linear function of its parent's population size. We can prove that by using only positive conjunctive formulae in the WPFs of a child PRV, we can model any polynomial decision threshold. First we prove this for the single-parent case and then for the general case of multi-parents. We assume in the following propositions that Q is a Boolean PRV and $R_i(x_i)$ are its parents where x_i is the set of logical variables in R_i which are not in Q. We also use x_i' to refer to a new logical variable typed with the same population as x_i.

Proposition 5. *A positive conjunctive RLR definition of $P(Q \mid R(x))$ (single-parent case) can represent any decision threshold that is a polynomial of terms each indicating a number of (tuples of) individuals for which R(x) is True or False.*

Proof. Based on Proposition 3 we know that a WPF having any Boolean formula can be written as a set of WPFs each having a positive conjunctive formula. Therefore, in this proof we disregard using only positive conjunctives. The final set of WPFs can be then represented by a set of positive conjunctive WPFs using Proposition 3.

Each term of the polynomial in the single-parent case is of the form $w(\prod_i |y_i|^{d_i}) {n_T}^{\alpha} {n_F}^{\beta}$, where n_T and n_F denote the number of individuals for which $R(x)$ is *True* or *False* respectively, $y_i \in x$ represents the logical variables in x, α, β and d_is are non-negative integers, and w is the weight of the

term. First we prove by induction that for any j, there is a WPF that can build the term ${n_T}^{\alpha}{n_F}^{\beta}$ where $\alpha+\beta=j$, $\alpha \geq 0$ and $\beta \geq 0$.

For $j=0$, ${n_T}^{\alpha}{n_F}^{\beta}=1$. We can trivially build this by WPF $\langle\{\},True,w\rangle$. Assuming it is correct for j, we prove it for $j+1$. For $j+1$, either $\alpha>0$ or $\beta>0$. If $\alpha>0$, using our assumption for j, we can have a WPF $\langle L,F,w\rangle$ which builds the term ${n_T}^{\alpha-1}{n_F}^{\beta}$. So the WPF $\langle L\cup x',F\wedge R(x'),w\rangle$ builds the term ${n_T}^{\alpha}{n_F}^{\beta}$ because the first WPF was $True\ {n_T}^{\alpha-1}{n_F}^{\beta}$ times and now we count it n_T more times because $R(x')$ is $True\ n_T$ times.

If $\alpha=0$ and $\beta>0$, we can have a WPF $\langle L,F,w\rangle$ which builds the term ${n_T}^{\alpha}{n_F}^{\beta-1}$. By the same reasoning as in previous case, we can see that the WPF $\langle L\cup\{x'\},F\wedge\neg R(x'),w\rangle$ produces the term ${n_T}^{\alpha}{n_F}^{\beta}$.

In order to include the population size of logical variables y_i, where $y_i \in x$, and generate the term $(\prod_i |y_i|^{d_i}){n_T}^{\alpha}{n_F}^{\beta}$, we only add d_i extra logical variables y_i' to the set of logical variables of the WPF that generates ${n_T}^{\alpha}{n_F}^{\beta}$. Then we set the weight of this WPF to w to generate the desired term. □

Conclusion. We can conclude from this proposition that a term $w(\prod_i |y_i|^{d_i}){n_T}^{\alpha}{n_F}^{\beta}$ can be generated by having a WPF with its formula consisting of n_T instances of $R(x')$ and n_F instances of $\neg R(x')$, adding d_i of each logical variable y_i to the set of logical variables, and setting the weight of WPF to w. We will use this conclusion for proving the proposition in multi-parent case.

Example 9. Suppose we want to model the case where Q is *True* if ${n_T}^2 \geq 2n_F+5$. In this case, we need to model the sigmoid of the polynomial ${n_T}^2-2n_F-4.5$. The reason for using 4.5 instead of 5 is to make the polynomial positive when ${n_T}^2=2n_F+5$. The following WPFs are used by RLR to model this polynomial. The first one generates the term ${n_T}^2$, the second one generate $-2n_F$ and the third one generates -4.5.

$$\langle\{x,x'\},R(x)\wedge R(x'),1\rangle$$
$$\langle\{x\},\neg R(x),-2\rangle$$
$$\langle\{\},True,-4.5\rangle$$

Note that the second WPF above can be written in positive form by using the following two WPFs:

$$\langle\{x\},True,-2\rangle$$
$$\langle\{x\},R(x),2\rangle$$

Using the conclusion following Proposition 5, we now extend Proposition 5 to the multi-parent case:

Proposition 6. *A positive conjunctive RLR definition of $P(Q \mid R_i(x_i))$ (multi-parent case) can represent any decision threshold that is a polynomial of terms each indicating the number of (tuples of) individuals for which a Boolean function of parents hold.*

Proof. Let $G_1,G_2,\ldots,G_t$ represent the desired Boolean functions of parents for our model. Also let $n_{(i)}$ denote the number of individuals for which G_i is *True*. Each term of the polynomial in the multi-parent case is then of the form: w * *(a polynomial of population sizes)* * $n_{(1)}^{\alpha_1}n_{(2)}^{\alpha_2}\ldots n_{(t)}^{\alpha_t}$. We demonstrate how we can generate any term $n_{(1)}^{\alpha_1}n_{(2)}^{\alpha_2}\ldots n_{(t)}^{\alpha_t}$. The inclusion of population size of logical variables and the weight w for each term is the same as in Proposition 5.

The conclusion of Proposition 5 can be easily generalized to work for any Boolean formula G_i instead of R. We only need to include a conjunction of α_i instances of G_i with different logical variables representing the same population in each instance. We use this generalization in our proof.

For each $n_{(i)}^{\alpha_i}$, we can use the generalization of conclusion of the Proposition 5 to come up with a WPF $\langle L_i,F_i,1\rangle$ which generates this term. Similar to the reasoning for single-parent case, we can see that the WPF $\left\langle\{\cup_{i=1}^{t}L_i,F_1\wedge F_2\wedge\cdots\wedge F_t,w\right\rangle$ generates the term $n_{(1)}^{\alpha_1}n_{(2)}^{\alpha_2}\ldots n_{(t)}^{\alpha_t}$. We can then use Proposition 3 to write this WPF using only positive conjunctive WPFs. □

Example 10. Suppose we want to model the case where Q is *True* if the square of number of individuals for which $R_1(x_1)=True$ multiplied by the number of individuals for which $R_2(x_2)=True$ is less than five times the whole number of *False* individuals in $R_1(x_1)$ and $R_2(x_2)$. In this case, we define $G_1=\neg R_1(x_1)$, $G_2=\neg R_2(x_2)$ and $G_3=R_1(x_1)\wedge R_1(x_1')\wedge R_2(x_2)$ we need to model the sigmoid of the polynomial $5n_{(1)}+5n_{(2)}-n_{(3)}-0.5$. The reason why we use -0.5 in the polynomial is that we want the polynomial to be negative when $5n_{(1)}+5n_{(2)}=n_{(3)}$. The following WPFs are used by RLR to model this polynomial where the first formula generates the term $5n_{(1)}$, the second generates $5n_{(2)}$, the third generates $-n_{(3)}$ and the fourth generates -0.5.

$$\langle\{x_1\},\neg R_1(x_1),5\rangle$$
$$\langle\{x_2\},\neg R_2(x_2),5\rangle$$
$$\langle\{x_1,x_1',x_2\},R_1(x_1)\wedge R_1(x_1')\wedge R_2(x_2),-1\rangle$$
$$\langle\{\},True,-0.5\rangle$$

Note that the first and second WPF above can be written in positive form in the same way as in Example 9.

Proposition 7 proves the converse of Proposition 6:

Proposition 7. *Any decision threshold that can be represented by a positive conjunctive RLR definition of $P(Q \mid R_i(x_i))$ is a polynomial of terms each indicating the number of (tuples of) individuals for which a Boolean function of parents hold.*

Proof. We prove that every WPF for Q can only generate a term of the polynomial. Since RLR sums over these terms, it will always represent the polynomial of these terms.

Similar to Proposition 6, let $G_1,G_2,\ldots,G_t$ represent the desired Boolean functions of parents and let $n_{(i)}$ denote the number of individuals for which G_i is *True*. A positive conjunctive formula in a WPF can consist of α_1 instances of G_1, α_2 instances of G_2, …, α_t instances of G_t. Based on Proposition 6, we know that this formula is $True\ n_{(1)}^{\alpha_1}n_{(2)}^{\alpha_2}\ldots n_{(t)}^{\alpha_t}$ times. The WPF can contain more logical variables in its set of logical variables than the ones in its formula. This, however, will only cause the above term to be multiplied by the

population size of the logical variable generating a term of the polynomial described in Proposition 6. Therefore, each of the WPFs can only generate a term of the polynomial which means that positive conjunctive RLR can only represent the sigmoid of this polynomial. □

Approximating Other Aggregators Using RLR

We can model other well-known aggregators using positive conjunctive RLR. In most cases, however, this is only an approximation because the sigmoid function reaches 0 or 1 only asymptotically and we cannot choose infinitely large numbers. We can, however, get arbitrarily close to 0 or 1 by choosing arbitrarily large weights. In the rest of this section, we use M to refer to a number which can be set sufficiently large to receive the desired level of approximation. n_{val} is the number of individuals x for which $R(x) = val$, when R is not Boolean.

OR. To model OR in RLR, we use the WPFs:

$$\langle\{\}, True, -M\rangle$$
$$\langle\{x\}, S(x), 2M\rangle$$

for which $P(q \mid S(x)) = \text{sigmoid}(-M + 2Mk)$. We can see that if none of the individuals are *True* (i.e. $n_T = 0$), the value inside the sigmoid is $-M$ which is a negative number thus the probability is close to 0. If even one individual is *True* (i.e. $n_T \geq 1$), the value inside the sigmoid becomes positive and the probability becomes closer to 1. In both cases, the value inside the sigmoid is a linear function of M. Increasing M pushes the probability closer to 0 or to 1 and the approximation becomes more accurate.

AND. AND can be modeled similarly to OR, but with the WPFs

$$\langle\{\}, True, M\rangle$$
$$\langle\{x\}, True, -2M\rangle$$
$$\langle\{x\}, S(x), 2M\rangle$$

for which $P(q \mid S(x)) = \text{sigmoid}(M - 2Mn_F)$. When $n_F = 0$, the value inside the sigmoid is $M > 0$, so the probability is closer to 1. When $n_F \geq 1$, the value inside the sigmoid becomes negative and the probability becomes closer to 0. Like OR, accuracy increases with M. Note that the number of WPFs for representing *OR* and *AND* is fixed.

Noisy-OR and noisy-AND. Figure 2 (a) represents how noisy-OR and noisy-AND can be modeled for the network in Figure 1. In this figure, $R(x)$ represents the values of the individuals being combined and $N(x)$ represents the noise probability. For noisy-OR, $S(x) \equiv R(x) \wedge N(x)$, and Q is the OR aggregator of $S(x)$. For noisy-AND, $S(x) \equiv R(x) \vee N(x)$, and Q is the AND aggregator of $S(x)$.

Mean $>$ t. We can model "Q is *True* if $\text{mean}(R(x)) > t$" using the following WPFs (*val* and t are numeric constants):

$$\langle\{\}, True, -M\rangle$$
$$\langle\{x\}, R(x) = val, M^2(val - t)\rangle \quad \text{for each } val \in \text{range}(R)$$

for which

$$P(q \mid R(x)) = \text{sigmoid}(-M + M^2 \sum_{val \in \text{range}(R)} n_{val}(val - t))$$
$$= \text{sigmoid}(-M + M^2(sum - nt))$$

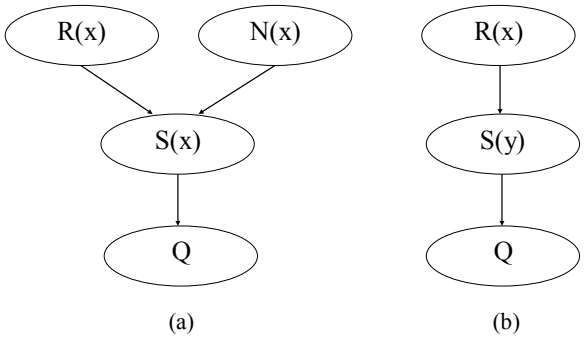

Figure 2: (a) The model for the noisy-OR and noisy-AND aggregators (b) The model for the mode aggregator

where $n = |x|$ and *sum* represents the sum of the values of the individuals. When $mean = \frac{sum}{n} > t$, the value inside the sigmoid is positive and the probability is close to 1. Otherwise, the value inside the sigmoid is negative and the probability is close to 0. Note that M should be greater than $|\frac{1}{sum - nt}|$ to generate a number greater than 1 when multiplied by $(sum - nt)$. Otherwise, it may occur that $sum - nt > 0$ but $M^2(sum - nt) \leq M$ which makes the sigmoid produce a number close to 0. Also note that the number of required WPFs grows with the number of values that the parent can take.

More than t *True*s. "Q is *True* if R is *True* for more than t individuals" can be modeled using the WPFs:

$$\langle\{\}, True, -2Mt - M\rangle$$
$$\langle\{x\}, R(x), 2M\rangle$$

giving $P(q \mid R(x)) = \text{sigmoid}(-2Mt - M + 2Mn_T)$ and the value inside the sigmoid is positive if $n_T > t$. The number of WPFs required is fixed.

More than t% *True*s. "Q is *True* if R is *True* for more than t percent of the individuals" is a special case of the aggregator "mean $> \frac{t}{100}$" when we treat *False* values as 0 and *True* values as 1. This directly provides the WPFs:

$$\langle\{\}, True, -M\rangle$$
$$\left\langle\{x\}, \neg R(x), M^2(0 - \tfrac{t}{100})\right\rangle$$
$$\left\langle\{x\}, \quad R(x), M^2(1 - \tfrac{t}{100})\right\rangle$$

while requiring $M > |\frac{1}{n_T - nt/100}|$, where n is the populations size of x. Note that we can use Proposition 3 to replace the second WPF with two WPFs having positive conjunctive formulae. Unlike the aggregator "mean $> \ldots$", here the number of WPFs is fixed.

Max $>$ t. We can model "Q is *True* if $\max(R(x)) > t$" using the following WPFs:

$$\langle\{\}, True, -M\rangle$$
$$\langle\{x\}, R(x) = val, 2M\rangle \quad \text{for each } val > t,\ val \in \text{range}(R)$$

thus $P(q \mid R(x)) = \text{sigmoid}(-M + 2M \sum_{val > t \in \text{range}(R)} n_{val})$. The value inside the sigmoid is positive if there is an individual having a value greater then t (i.e. $\exists val > t \in \text{range}(R)$:

$n_{val} > 0$). Note that the number of WPFs required grows with the number of values greater than t that the parents can take.

Max. For binary parents, the "max" aggregator is identical to "OR". Otherwise, $\text{range}(Q) = \text{range}(R(x))$. The "max" aggregator can be modeled using a 2-level structure. First, for every $t \in \text{range}(R(x))$, create a separate "max $\geq t$" aggregator, with $R(x)$ as its parents. Then, define the child Q, with all the "max $\geq t$" aggregators as its parents. Q can compute $\max R(x)$ given its parents. Note that while $|parents(Q)| = |\text{range}(R(x))|$ may be arbitrarily large, $|parents(Q)|$ does not change with population size, hence it is possible to use non-relational constructs (e.g., a table) for its implementation.

Mode = t. To model "Q is *True* if $\text{mode}(R(x)) = t$", we first add another PRV $S(y)$ to the network as in Figure 2 (b) where y represents the range of the values for $R(x)$. Then for each individual $S(C)$ of $S(y)$, we use the following WPFs for which $P(s(C) \mid R(x)) = \text{sigmoid}(M - 2M(n_C - n_t))$ and the value inside the sigmoid is positive if $n_t \geq n_C$. Note that the number of WPFs required grows with the number of values that the parent can take.

$$\langle \{\}, True, M \rangle$$
$$\langle \{x\}, R(x) = C, -2M \rangle$$
$$\langle \{x\}, R(x) = t, 2M \rangle$$

Then Q must be *True* if all the individuals in S are *True*. This is because a *False* value for an individual of S means that this individual has occurred more than t and t is not the mode. Therefore, we can use WPFs similar to the ones we used for AND:

$$\langle \{\}, True, M \rangle$$
$$\langle \{y\}, True, -2M \rangle$$
$$\langle \{y\}, S(y), 2M \rangle$$

Mode & Majority. For binary parents, the "mode" aggregator is also called "majority", and can be modeled with the "more than t% Trues" aggregator, with $t = 50$. Otherwise, $\text{range}(Q) = \text{range}(R(x))$, and we can use the same approach as for "max", by having $|\text{range}(R(x))|$ separate "mode $= t$" aggregators, with Q as their child.

Beyond Polynomial Decision Thresholds

Proposition 7 showed that any conditional probability that can be expressed using a positive conjunctive RLR definition of $P(Q \mid R_i(x_i))$ is the sigmoid of a polynomial of the number of *True* and *False* individuals in each parent $R_i(x_i)$. However, given that the decision thresholds are only defined for integral counts, some of the apparently non-polynomial decision thresholds are equivalent to a polynomial and so can be modeled using RLR.

Example 11. Suppose we want to model $Q \equiv \left(\left\lceil \sqrt{n_T} \right\rceil < n_F \right)$. This is a non-polynomial decision threshold, but since n_T and n_F are integers, it is equivalent to the polynomial decision threshold $n_T - (n_F - 1)^2 \leq 0$ which can be formulated using RLR.

Example 12. Suppose we want to model $Q \equiv (2^{n_T} > 3^{n_F})$. This is, however, equivalent to the polynomial form $Q \equiv (n_T \log 2 - n_F \log 3 > 0)$ and can be formulated in positive conjunctive RLR using the WPFs:

$$\langle \{x\}, True, -\log 3 \rangle$$
$$\langle \{x\}, R(x), \quad \log 3 + \log 2 \rangle$$

There are, however, non-polynomial decision thresholds that cannot be converted into a polynomial one and RLR is not able to formulate them.

Example 13. Suppose we want to model $Q \equiv (2^{n_T} > n_F)$. This cannot be converted to a polynomial form and RLR cannot formulate it.

Finding a parametrization that allows to model any non-polynomial decision threshold remains an open problem.

Conclusion

Today's data and models are complex, composed of objects and relations, and noisy. Hence it is not surprising that relational probabilistic knowledge representation currently receives a lot of attention. However, relational probabilistic modeling is not an easy task and raises several novel issues when it comes to knowledge representation:

- What assumptions are we making? Why should we choose one representation over another?
- We may learn a model for some population size(s), and want to apply it to other population sizes. We want to make assumptions explicit and know the consequences of these assumptions.
- If one model fits some data, it is important to understand why it fits the data better.

In this paper, we provided answers to these questions for the case of the logistic regression model. The introduction of the relational logistic regression (RLR) family from first principle is already a major contribution. Based on it, we have investigated the dependence on population size for different variants and have demonstrated that already for simple and well-understood (at the non-relational level) models, there are complex interactions of the parameters with population size. Future work includes inference and learning for these models and understanding the relationship to other models such as undirected models like MLNs. Exploring these directions is important since determining which models to use is more than fitting the models to data; we need to understand what we are representing.

References

Buchman, D.; Schmidt, M.; Mohamed, S.; Poole, D.; and Freitas, N. D. 2012. On sparse, spectral and other parameterizations of binary probabilistic models. In *AISTATS 2012-15th International Conference on Artificial Intelligence and Statistics*.

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.

Friedman, N.; Getoor, L.; Koller, D.; and Pfeffer, A. 1999. Learning probabilistic relational models. In *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence (IJCAI-99)*, volume 99, 1300–1309. Stockholm, Sweden: Morgan Kaufman.

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

Horsch, M., and Poole, D. 1990. A dynamic approach to probability inference using bayesian networks. In *Proc. sixth Conference on Uncertainty in AI*, 155–161.

Jian, D.; Barthels, A.; and Beetz, M. 2009. Adaptive Markov logic networks: Learning statistical relational models with dynamic parameters. In *9th European Conference on Artificial Intelligence (ECAI)*, 937–942.

Jian, D.; Bernhard, K.; and Beetz, M. 2007. Extending Markov logic to model probability distributions in relational domains. In *KI*, 129–143.

Kisynski, J., and Poole, D. 2009. Lifted aggregation in directed first-order probabilistic models. In *Twenty-first International Joint Conference on Artificial Intelligence*, 1922–1929.

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

Mitchell, T. 2010. Generative and discriminative classifiers: naive Bayes and logistic regression. http://www.cs.cmu.edu/ tom/mlbook/NBayesLogReg.pdf.

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)*.

Neville, J.; Simsek, O.; Jensen, D.; Komoroske, J.; Palmer, K.; and Goldberg, H. 2005. Using relational knowledge discovery to prevent securities fraud. In *Proceedings of the 11th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. MIT Press.

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

Perlish, C., and Provost, F. 2006. Distribution-based aggregation for relational learning with identifier attributes. *Machine Learning* 62:65–105.

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. 2003. First-order probabilistic inference. In *Proceedings of the 18th International Joint Conference on Artificial Intelligence (IJCAI-03)*, 985–991.

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