

Population Size Extrapolation in Relational Probabilistic Modelling^{*}

David Poole¹, David Buchman¹, Seyed Mehran Kazemi¹, Kristian Kersting²,
and Sriraam Natarajan³

¹ University of British Columbia, {poole,davidbuc,smkazemi}@cs.ubc.ca,
www.cs.ubc.ca/~{poole,davidbuc,smkazemi}/

² Technical University of Dortmund,

<http://www-ai.cs.uni-dortmund.de/PERSONAL/kersting.html>

³ Indiana University, <http://homes.soic.indiana.edu/natarasr/>

Abstract. When building probabilistic relational models it is often difficult to determine what formulae or factors to include in a model. Different models make quite different predictions about how probabilities are affected by population size. We show some general patterns that hold in some classes of models for all numerical parametrizations. Given a data set, it is often easy to plot the dependence of probabilities on population size, which, together with prior knowledge, can be used to rule out classes of models, where just assessing or fitting numerical parameters will be misleading. In this paper we analyze the dependence on population for relational undirected models (in particular Markov logic networks) and relational directed models (for relational logistic regression). Finally we show how probabilities for real data sets depend on the population size.

1 Introduction

Relational probabilistic models [4, 17] or template-based models [10] represent the probabilistic dependencies between relations of individuals. In these models, individuals about which we have the same information are exchangeable (i.e. the individuals are treated identically when we have no evidence to distinguish them) and the probabilities are about relations among individuals, which can be specified independently of actual individuals.

In a relational probabilistic model, the predictions of the model may depend on the number of individuals (the population size). For instance, whether someone enjoys a party or not may depend on the number of people they know at that party, and each person at a party may know a different number of people.

Even simple models make strong predictions about the effect of population size on probabilities. If we want to extrapolate from data (as opposed to interpolating), it is important to know how the models handle changes in population size. Extrapolating from small sample sizes to large ones can be very presumptuous, e.g., people act very differently in small groups than in mobs. The structure

^{*} Parts of this paper appeared in the UAI-2012 StarAI workshop [16].

of the model reflects implicit prior knowledge and assumptions, which are important to understand. We advocate that we should choose from the models where the extrapolation is reasonable given the data and prior knowledge.

We consider two classes of relational models, undirected models exemplified by Markov logic networks (MLNs) [18, 2], and directed models with aggregators exemplified by relational logistic regression (RLR) [9], the directed analogue of MLNs.

This work is complementary to the work of Jain et al. [8, 7], who allow weights to vary with the population. Varying weights may be necessary for a particular domain, but from a modeling perspective it is first important to understand what happens when weights are not varied. This paper mainly considers what happens as the population varies, rather than just the limiting probabilities [6].

In the rest of the paper, we first introduce some basic definitions and describe MLNs and RLR. Then we consider a simple model and explain how RLR models and MLNs are influenced by population size and how they behave differently even for this simple model. We then expand these results to more complicated cases, and give some general theoretical results, some empirical data and many open problems.

2 Some Basic Definitions

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. For this paper we assume the populations are disjoint; each individual is only in one population. When there is a single population, we use n for the population size, and write the population as $A_1 \dots A_n$.

Each **logical variable**, written in lower case, is typed with a population. $pop(x)$ is the population associated with the logical variable x , and $|x| = |pop(x)|$. Constants, denoting individuals, start with an upper case letter. We assume there is a constant for each individual, and there is no uncertainty about the identity of the individuals.

A **parametrized random variable (PRV)** is of the form $F(t_1, \dots, t_k)$ where F is a k -ary predicate symbol and each t_i is a logical variable or a constant. For example, $At(x, y)$, $At(x, Home)$, $At(Sam, Home)$ are PRVs. The range of the random variables is $\{False, True\}$. (It is possible to have PRVs with more general domains, but the points of the paper can already be made in this simpler setting.) A **ground random variable** is a PRV where all t_i are constants.

An **atom** is an assignment of a value to a PRV. For example, $At(x, Home) = True$ is an atom. We will write assignments in lower case; $R(x) = True$ is written as $r(x)$, and $R(x) = False$ is written as $\neg r(x)$. A **formula** is made up of atoms with logical connectives (we ignore quantification in this paper.) An **instance of a formula** is obtained by replacing logical variables with constants.

A **world** is an assignment of a value to each ground random variable. The number of worlds is exponential in the number of ground random variables.

3 Markov Logic Networks and Relational Logistic Regression

Markov logic networks (MLNs) [18, 2] and relational logistic regression (RLR) [9] are defined in terms of weighted formulae. In MLNs the formulae are used to define joint probability distributions. In RLR the formulae are used to define conditional probabilities.

A **weighted formula** (WF) is a triple $\langle L, F, w \rangle$ where L is a set of logical variables, F is a formula where all of the free logical variables in F are in L , and w is a real-valued weight.

An MLN is a set of weighted formulae⁴, where the probability of any world is proportional to the exponent of the sum of the weights of the instances of the formulae that are true in the world.

RLR is a form of aggregation, defining conditional probabilities in terms of weighted formulae. We assume a directed acyclic graph on PRVs (where the PRVs of different nodes do not unify), which defines a Bayesian network on the corresponding ground random variables. For each PRV, there are weighted formulae involving an instance of that PRV and PRVs involving instances of (a subset of) the parent PRVs. The conditional probability of each ground random variable given an assignment of values to each of its parent ground random variables is proportional⁵ to the exponential of the sum of the weights of the instances of the formulae that are true for that assignment.

Example 1. Suppose we have the weighted formulae:

$$\begin{aligned} &\langle \{\}, q, \alpha_0 \rangle \\ &\langle \{x\}, q \wedge \neg r(x), \alpha_1 \rangle \\ &\langle \{x\}, q \wedge r(x), \alpha_2 \rangle \\ &\langle \{x\}, r(x), \alpha_3 \rangle \end{aligned}$$

Treating this as an MLN, if the truth value for $r(x)$ for every individual x is observed:

$$P(q \mid obs) = \text{sigmoid}(\alpha_0 + n_F \alpha_1 + n_T \alpha_2) \quad (1)$$

where obs has $R(x)$ true for n_T individuals, and false for n_F individuals out of a population of $n = n_F + n_T$ individuals. $\text{sigmoid}(x)$ is $1/(1 + e^{-x})$.

Note that, in the MLN, α_3 is not required for representing the conditional probability (because it cancels out), but can be used to affect $P(r(A_i))$.

⁴ MLNs typically do not explicitly include the set of logical variables as part of the weighted formulae, but use the free variables in F . If one wanted to add an extra logical variable, x , one could conjoin $true(x)$ to F where $true$ is a property that is true for all individuals.

⁵ In MLNs there is a single normalizing constant, guaranteeing the probabilities of the worlds sum to 1. In RLR, normalization is done separately for each possible assignment to the parents.

In [9], the sigmoid, as in Equation (1), is used as the definition of RLR. ([9] assumed all formulae were conjoined with $q \wedge$, and omitted $q \wedge$ from the formulae.) When not all $R(A_i)$ are observed, RLR uses Equation (1) for the conditional probability of q given each combination of assignments to the $R(x)$, and requires a separate model for the probability of the $R(x)$.

In summary: RLR uses the weighted formulae to define the conditional probabilities, and MLNs use them to define the joint probability distribution.

Example 2. Suppose people want to go to a party, and the party is fun for them if they know at least one social person in the party. In this case, a PRV $funFor(x)$ is a child of PRVs $knows(x, y)$ and $social(y)$. The following weighted formulae can be used to model the dependence of $funFor(x)$ on its parents:

$$\begin{aligned} &\langle \{x\}, funFor(x), -5 \rangle \\ &\langle \{x, y\}, funFor(x) \wedge knows(x, y) \wedge social(y), 10 \rangle \end{aligned}$$

RLR sums over the above weighted formulae and takes the *sigmoid*, giving:

$$P(funFor(x) | \Pi) = sigmoid(sum), \quad \text{where } sum = -5 + 10n_T$$

where, for each x , Π is an assignment of values to $knows(x, y)$ and $social(y)$, and n_T represents the number of individuals y for which $knows(x, y) \wedge social(y)$ is *True* in Π . When $n_T = 0$, $sum < 0$ and the probability is closer to 0; when $n_T > 0$, $sum > 0$ and the probability is closer to 1.

Example 3. This example is similar to Example 1, but uses only positive conjunctions⁶, and also involves multiple logical variables of the same population.

$$\begin{aligned} &\langle \{\}, q, \alpha_0 \rangle \\ &\langle \{x\}, q \wedge true(x), \alpha_1 \rangle \\ &\langle \{x\}, q \wedge r(x), \alpha_2 \rangle \\ &\langle \{x\}, true(x), \alpha_3 \rangle \\ &\langle \{x\}, r(x), \alpha_4 \rangle \\ &\langle \{x, y\}, q \wedge true(x) \wedge true(y), \alpha_5 \rangle \\ &\langle \{x, y\}, q \wedge r(x) \wedge true(y), \alpha_6 \rangle \\ &\langle \{x, y\}, q \wedge r(x) \wedge r(y), \alpha_7 \rangle \end{aligned}$$

In RLR and in MLN, if all $R(A_i)$ are observed:

$$P(q | obs) = sigmoid(\alpha_0 + n\alpha_1 + n_T\alpha_2 + n^2\alpha_5 + n_Tn\alpha_6 + n_T^2\alpha_7)$$

where obs has $R(x)$ true for n_T individuals, and false for n_F individuals out of a population of n . The use of two logical variables (x, y) of the same population gives a squared dependency in the population.

⁶ Here $true(x)$ is true of every x . This notation is redundant. If you want the traditional MLN notation, you can remove the explicit set of logical variables and keep the $true(\cdot)$ relations. If you are happy with the explicit logical variables, you can remove the $true(\cdot)$ predicates. Removing both is incorrect. Keeping both is harmless. Formulae that involve negation are redundant; any set of weighted formulae involving negation can be replaced by weighted formulae that don't involve negation [9].

4 Three Elementary Models

Consider the simplest case of aggregating over populations, with a PRV Q connected to a PRV $R(x)$ containing an extra logical variable, x , as in Figure 1. In the grounding, Q is connected to $n = |\text{pop}(x)|$ instances of $R(x)$. We assume the model is defined before n is known; it is applicable for all values of n .

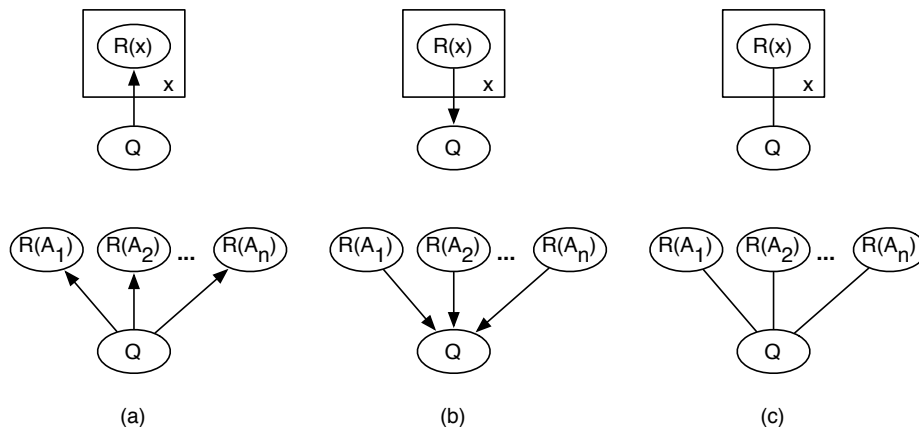


Fig. 1. Running example as (a) naïve Bayes (b) logistic regression with independent priors for each $R(x)$ and (c) Markov network. On the top are the networks using plate notation, where plates [1], drawn as rectangles, correspond to logical variables. On the bottom are the groundings for the population $\{A_1, A_2, \dots, A_n\}$.

For this situation, Fig. 1(c) shows an undirected model with a factor for Q and a pairwise factor for Q with each individual. Fig. 1(a) shows a directed model where $R(x)$ is a child of Q . In the grounding it produces a naïve Bayes model with a factor for $P(Q)$ and a separate factor for $P(R(A_i) | Q)$ for each individual. In both of these models the joint probability is the product of factors. In terms of MLNs and RLR, factors corresponds to weighted formulae.

The naïve Bayes model of Figure 1(a) is an instance of the Markov network of 1(c). Every naïve Bayes model can be represented by a Markov network, but the converse is not true. In some sense the naïve Bayes model is the Markov network with the constraint that the factors represent conditional probabilities (sum to 1, given Q).

For a directed model with $R(x)$ as a parent of Q (Fig. 1(b)), Q has an unbounded number of parents in the grounding, so we need some way to aggregate the parents. Common ways to aggregate in relational domains, e.g. [5, 3, 12, 14, 11], include logical operators such as *noisy-or*, *noisy-and*, as well as ways to combine probabilities. This requirement for aggregation occurs in a directed model whenever a parent contains an extra logical variable.

While it may seem that these models are syntactic variants, the models involve very different independence assumptions [13]:

- In the naïve Bayes and the MLN (Figure 1(a) and 1(c)), the variables $R(x)$ and $R(y)$ (for $x \neq y$) are independent given Q , and dependent not given Q .
- In the directed model with aggregation (Figure 1(b)) the variables $R(x)$ and $R(y)$ (for $x \neq y$) are dependent given Q , and independent not given Q .

These dependencies do not depend on what aggregation is used for the directed model. For the rest of this paper we assume that RLR is used as the aggregator. Note that RLR can use the same formulae as the MLN, in which case, when all $R(A_i)$ are observed, the posterior probability of Q would be the same in the MLN and RLR models; however, the posterior probabilities of Q are different when not all of the $R(A_i)$ are observed.

The difference in the dependency structure means that we cannot represent a logistic regression model where the $R(A_i)$ are dependent when Q is observed using an MLN, because in such an MLN the $R(A_i)$ are independent given Q . It is an open problem whether introducing new formulae that involve multiple individuals may allow an MLN to represent the regression model. Similarly, an RLR model cannot represent the MLN where the $R(A_i)$'s are dependent not given Q , without introducing other relations or dependencies among the variables. It is an open question as to whether any finite set of formulae is adequate to make them able to represent the same distributions.

5 Effects of Population Sizes

In this section we investigate the behaviour of MLNs and RLR as the population size n varies.

5.1 A Comparison of MLN, RLR and MF for the Simplest Case

We now compare MLN, RLR, and a simple mean-field (MF) approximation of RLR, for the elementary models in Figure 1. For MLN (Figure 1 (c)), we use the MLN parametrization of Example 1 as the joint distribution. For RLR (Figure 1 (b)), we use p_r as the i.i.d. prior probability of each $r(x)$, and use the RLR parametrization of Example 1 for $P(q | R(A_1), \dots, R(A_n))$. (Note that $P(r(x)) = p_r$ can be represented by RLR model for $R(x)$ using the single formula $\langle \{x\}, r(x), \alpha_3 \rangle$, where $\text{sigmoid}(\alpha_3) = p_r$.) We can now sum out the unobserved variables $R(x)$, and get $P(q | n)$. The dependency of $P(q)$ on n is an effect of population size.

For the MLN, when Q is conditioned on, the graph is disconnected, with each component $R(x)$ having the same probability. So to compute $P_{MLN}(q | n)$, we can compute the probability of one of them and raise it to the power of n [15]:

$$P_{MLN}(q | n) = \text{sigmoid}(\alpha_0 + n \log(e^{\alpha_2} + e^{\alpha_1 - \alpha_3})) \quad (2)$$

Note this is a logistic function (the sigmoid of a linear function) of n and α_0 , but not a logistic function of the other parameters.

For the RLR model, summing out the unobserved variables $R(x)$ gives:

$$P_{RLR}(q | n) = \sum_{i=0}^n \binom{n}{i} \text{sigmoid}(\alpha_0 + i\alpha_1 + (n-i)\alpha_2)(1-p_r)^i p_r^{n-i}$$

where i is the number of individuals for which $R(x)$ is false. This inference is an instance of first-order variable elimination [19].

Finally, the simple mean-field approximation to the RLR model is:

$$P_{MF}(q | n) = \text{sigmoid}(\alpha_0 + np_r\alpha_1 + n(1-p_r)\alpha_2)$$

Note that np_r is the expected number of $R(x)$'s that are true, and $n(1-p_r)$ is the expected number of $R(x)$'s that are false.

Example 4. Fig. 2 compares $P(q | n)$ for RLR, MLN and the mean-field approximation of RLR, using $\alpha_0 = -4.5$, $\alpha_1 = 1$, $\alpha_2 = -1$, and $p_r = 0.7$ (thus $P_{MF}(q | n) = \text{sigmoid}(-4.5 + 0.4n)$). The MLN uses $\alpha_3 = 2.82$, chosen to give it the same probability as the RLR for $n = 1$.

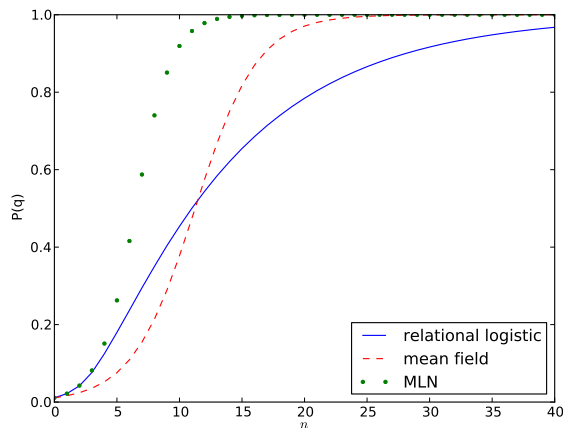


Fig. 2. $P(q | n)$ in Example 4.

$P_{MLN}(q | n)$ is a logistic function (the sigmoid of a linear function) of n , and so is monotonic with n . It might be conjectured that the MLN and RLR models are qualitatively similar. It is therefore intuitive to make the following conjecture:

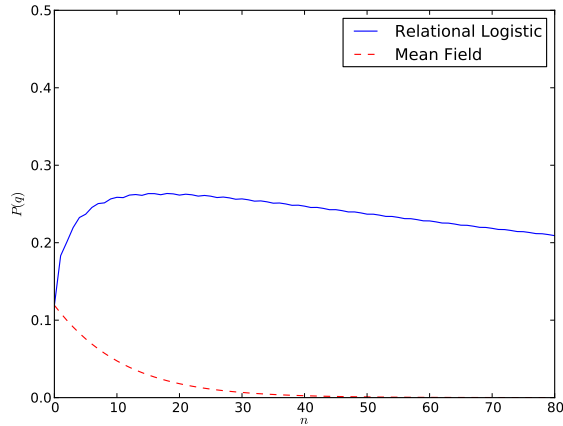


Fig. 3. $P(q | n)$ in Example 5

Conjecture 1. $P_{RLR}(q | n)$ (in the RLR model for Fig. 1 (b)) is monotonic in n .

It turns out that this conjecture is false.

Example 5. Fig. 3 demonstrates the setting: $\alpha_0 = -2$, $\alpha_1 = 2$, $\alpha_2 = -1$, $p_r = 0.3$. Whereas the mean-field approximation of RLR, $P_{MF}(q | n) = \text{sigmoid}(-2 - 0.1n)$, is monotonic, $P_{RLR}(q | n)$ is not, having a maximum at $n = 18$. (Example 6 shows $P_{MLN}(q | n)$ for this setting.)

5.2 Phase Transitions in MLNs

A phase transition in physics arises when a value flips from one state to another. In this section we show how a probability can flip from one value to another (e.g, close to 1 or close to 0) as either a parameter varies or a population varies. These interact, as rate of change can depend on the population and on parameter values.

One of the properties of the directed model of Figure 1(b) is that $P_{RLR}(R(A_i) | n)$ does not depend on n and can be given as input to the model. In MLNs, however, $P_{MLN}(R(A_i) | n)$ depends on n , except for the special case of a naïve Bayes model represented using an MLN. We show that for some MLNs, there is a phase transition where $P_{MLN}(R(A_i) | n)$ cannot be arbitrarily set in the limit as the population increases.

Example 6. Consider the same parametrization as Example 5, and the mapping to MLNs given in Example 1. Under this mapping, the MLN and the RLR both represent the same conditional probability $P(q | R(A_1), \dots, R(A_n))$. To fully

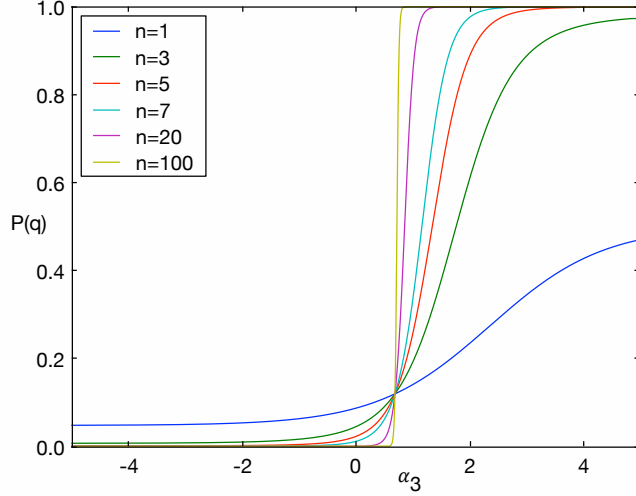


Fig. 4. $P_{MLN}(q | \alpha_3)$ in an MLN for various population sizes n , for Example 6.

specify the model, RLR requires p_r , representing $P(r(x))$ for all x . The MLN requires α_3 .

Fig. 4 shows $P_{MLN}(q | \alpha_3)$ for different population sizes n . All of these slopes are logistic functions. As n increases the slope becomes steeper.

There is a phase transition at approximately $\alpha_3 = 0.7$. For $\alpha_3 < 0.7$, $P_{MLN}(q | n)$ decreases with n , and for $\alpha_3 > 0.7$, $P_{MLN}(q | n)$ increases with n . At the phase transition point, $P_{MLN}(q | n)$ does not depend on n . The phase transition occurs when the coefficient of n in Equation (2) is 0.

Fig. 5 shows $P_{MLN}(r(A_1) | \alpha_3)$ for different population sizes n ($P_{MLN}(r(A_i))$ is identical for all individuals A_i). Similarly to Figure 4, the slope becomes steeper with increasing n 's.

Notice the way the parameter α_3 affects $P_{MLN}(q)$ or $P_{MLN}(r(A_i))$ depends on n . We cannot set the parameters so that the MLN represents arbitrary values for $P_{MLN}(r(A_i))$ as the population varies, as we show:

At the phase transition, there is an approximately vertical line segment for large populations. The corresponding probabilities for $r(A_1)$ cannot be represented in the limit $n \rightarrow \infty$. In the limit, $P_{MLN}(q | n)$ approaches either 0 or 1 (or is not affected by n). Suppose in the limit $P_{MLN}(q | n) \rightarrow 1$ and we tried to adjust α_3 to fit $P_{MLN}(r(A_1) | n) = 0.3$ when $P_{MLN}(q | n) = 1$. The new value found for α_3 implies that $P_{MLN}(q | n) \rightarrow 0$ in the limit. Similarly, suppose $P_{MLN}(q | n) \rightarrow 0$ and we tried to adjust α_3 to fit $P_{MLN}(r(A_1) | n) = 0.3$ when $P_{MLN}(q | n) = 0$, the new value found for α_3 implies that $P_{MLN}(q | n) \rightarrow 1$. Thus α_3 cannot be set to make $P_{MLN}(r(A_1) | n) \rightarrow 0.3$ as $n \rightarrow \infty$.

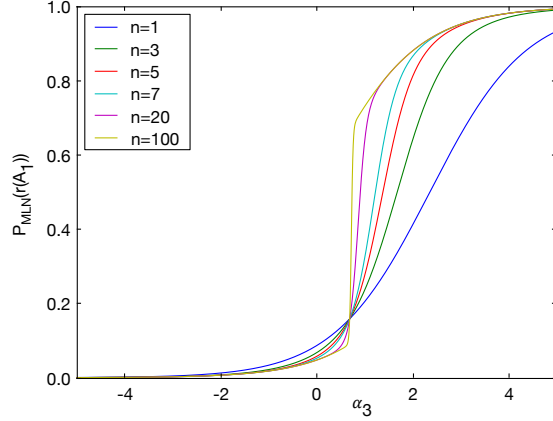


Fig. 5. $P_{MLN}(r(A_1) | \alpha_3)$ in an MLN for various population sizes n , for Example 6.

Fig. 6 shows how $P(q)$ and $P(r(A_1))$ vary with population size for two different parameterizations, $\alpha_3 = 0.66$ and 0.73 . The monotonically increasing lines are for $\alpha_3 = 0.73$ and the decreasing lines are for $\alpha_3 = 0.66$. As α_3 gets closer to the phase transition, the graphs approach the extremes at a slower rate.

5.3 Behavior of MLNs on More General Cases

In general it is a complex inference problem to determine the probability of a random variable as a function of n . However, we can characterize some of the cases where the probability is bounded away from 0 and 1, or approaches 0 or 1 in the limit as a population approaches infinity.

Proposition 1. *Consider an MLN with finite weights. Let n be the size of some population and V be a ground random variable. If the number of formula instantiations that depend on V 's value is independent of n , then $P_{MLN}(V | n)$ is bounded away from 0 and 1, i.e., exists $c > 0$ such that $0 < c \leq P_{MLN}(V | n) \leq 1 - c < 1$ for all n 's.*

Proof. The number of such formula instantiations was guaranteed to be fixed (independent of n). The weights are finite, so each such contribution is bounded. Define the neighbours of V to be the grounding of the other PRVs in the weighted formulae that V appears in. Let c be the minimum of the conditional probability of V given its neighbours, and $\neg V$ given its neighbours. This c has the property specified in the proposition, as $P(V | n)$ is a linear interpolation of the probabilities of V given its neighbours. \square

Proposition 2. *Consider an MLN with finite weights. Let pop be some population, $n = |pop|$, V be any PRV, and V' be any ground instance of V . If V'*

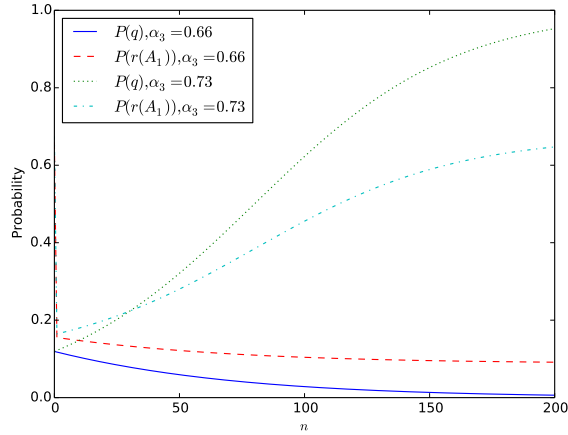


Fig. 6. $P_{MLN}(q | n)$ and $P_{MLN}(r(A_1) | n)$ for $\alpha_3 = 0.66$ and 0.73 , for Example 6.

does not unify with a PRV that is in a weighted formula with another PRV that has an extra logical variable typed with pop, then $P_{MLN}(V' | n)$ is bounded away from 0 and 1.

Proof. In this case V' has a fixed number of neighbours in the grounding as n varies, and there are a fixed number of formula instantiations that depend on V' 's value. Therefore, Proposition 1 guarantees $P_{MLN}(V')$ is bounded away from 0 and 1. \square

Proposition 3. Consider an MLN with finite weights. If PRV V is in a formula with PRV R that includes a logical variable of a population of size n that does not appear in V , and for any such R , R does not unify with a PRV in other formulae or with an instance of itself in that formula, then either $P_{MLN}(V | n)$ is a constant (independent of n), or $\lim_{n \rightarrow \infty} P_{MLN}(V)$ is either 1 or 0.

Proof. Such cases are locally isomorphic to the simple case analyzed earlier. \square

It is an open problem to characterize other cases of what happens in the limit.

5.4 Real Data and Prior Knowledge

Figure 7 show $P(25 < Age(p) < 45 | n)$ for a person p , given the number n of movies they rated, for the Movielens 1M dataset (<http://grouplens.org/datasets/movielens/>), averaged over all people. This is calculated by bucketing over n , with at least 20 people in each bucket.

When trying to fit models to such data, we first need to choose what model class to use. We might want to not only fit to the data, but to fit what we expect

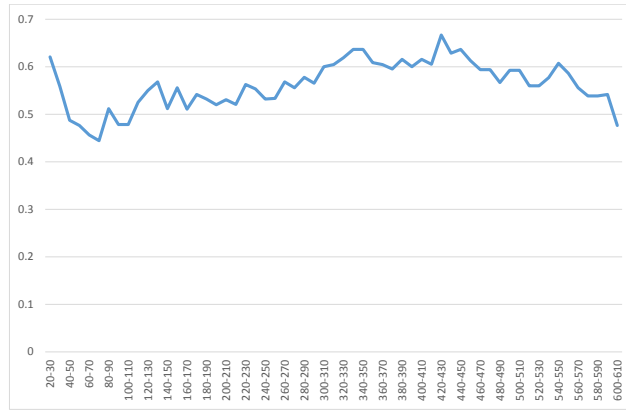


Fig. 7. Observed $P(25 < \text{Age}(p) < 45 \mid n)$ from the MovieLens dataset.

in the limit. We can design the structure of the model to either go to 0 or 1 in the limit or to be bounded away from 0 and 1. In this particular example, we would not expect the probability to go to 0 or 1, and we would also not expect the age to be independent of the number of movies a person has rated (the population size n for each person). So in the model we would not just have weighted formulae that contain $\text{Age}(\text{person})$ and $\text{Rated}(\text{person}, \text{movie})$, for if we did, by Proposition 3, either the age does not depend on the number of movies rated or the Age becomes deterministic (is 1 or 0) in the limit. This does not preclude more complicated formulae, but a preference for simpler models might.

5.5 Fitting Polynomials

In Example 3, $P(q \mid n)$ is a (sigmoid of a) degree-2 polynomial of n . One might innocently write weighted formulae like in Example 3 without realizing the implications of such statements and get very surprising results. In this section we show by example what can happen unexpectedly.

Consider fitting a degree-2 polynomial to data in which the population size n is in the range $0 \leq n \leq 50$. Suppose we find that the closest fit is $0.01n^2 - n + 16$. Suppose in another run, we fit $-0.01n^2 - 0.2n + 8$. Figure 8 plots these, but in the range $0 \leq n \leq 100$. The polynomials are very close in the training range, but the first polynomial goes up soon after, even though we have no evidence of this in the data set.

This is not an isolated occurrence. A degree- k polynomial may have up to $k-1$ points where it changes between increasing and decreasing. If the polynomial we fit has one or more of these points beyond the region of the training set, we are likely to get very unintuitive predictions.

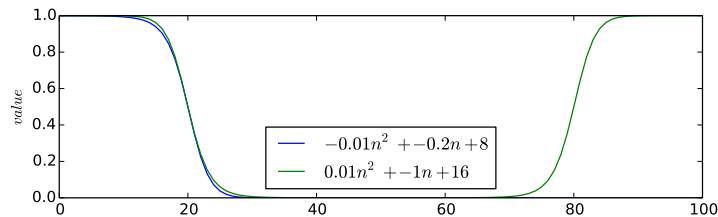


Fig. 8. Sigmoids of polynomials of n . The population size, n , is on the x-axis.

The sign of the coefficient of the leading power in the polynomial determines whether the probability approaches 0 or 1. However, this is often difficult to determine, particularly if we are close to phase transitions.

6 Conclusion

In this paper we investigated the dependence on population size for relational models. Even for simple models that are well understood at the non-relational level, there are complex interactions of the parameters with population size. The results of this paper are important for a number of reasons:

- If we learn a model for some population sizes and apply it to other population sizes, it is important to know what the model implies about such extrapolation of population sizes. Here we have shown some cases where the details of the model makes particular predictions about the extrapolation.
- We want to know the effect of choosing particular formulae. What assumptions are we making? For example, adding an adding an extra variable to a formula adds a dependency on population size.
- If one model fits some data better than another, it is important to understand why. We have investigated the effects of some design decisions for directed and undirected models.
- If we want to extrapolate from data, how can prior information affect the formulae used. The prior information we have considered is what how should the probability change as the population grows.

The other message is that undirected models such as MLNs are different to directed models, such as those that use RLR. It is important to understand these differences if we are to choose an appropriate model for a domain. In particular, when fitting a model to data, we should consider both models, and not assume that one works better than the other independently of the domain.

This paper has exposed more questions than it has answered. Determining dependencies on population sizes for more complicated models is an open question, which may allow a modeler to rule out some models for their specific application. Ideally, we would like ways to generate qualitative descriptions about the model from the model's formulae.

References

1. Buntine, W.L.: Operations for learning with graphical models. arXiv preprint cs/9412102 (1994)
2. Domingos, P., Kok, S., Lowd, D., Poon, H., Richardson, M.: Markov Logic. In Raedt, L.D., Frasconi, P., Kersting, K. and Muggleton, S., eds., Probabilistic Inductive Logic Programming. Springer (2008)
3. Friedman, N., Getoor, L., Koller, D., Pfeffer, A.: Learning probabilistic relational models. In: Proc. IJCAI-99. pp. 1300–1309 (1999)
4. Getoor, L., Taskar, B.: Introduction to Statistical Relational Learning. MIT Press, Cambridge, MA (2007)
5. Horsch, M., Poole, D.: A dynamic approach to probabilistic inference using Bayesian networks. In: Proc. Sixth Conference on Uncertainty in AI. pp. 155–161 (1990)
6. Jaeger, M.: Convergence results for relational Bayesian networks. In: Proceedings of LICS-98 (1998)
7. Jain, D., Barthels, A., Beetz, M.: Adaptive Markov logic networks: Learning statistical relational models with dynamic parameters. In 9th European Conference on Artificial Intelligence (ECAI) pp. 937–942 (2010)
8. Jain, D., Kirchlechner, B., Beetz, M.: Extending Markov logic to model probability distributions in relational domains. In KI pp. 129–143 (2007)
9. Kazemi, S.M., Buchman, D., Kersting, K., Natarajan, S., Poole, D.: Relational logistic regression. In: Proc. 14th International Conference on Principles of Knowledge Representation and Reasoning (KR-2014) (2014)
10. Koller, D., Friedman, N.: Probabilistic Graphical Models: Principles and Techniques. MIT Press, Cambridge, MA (2009)
11. Natarajan, S., Khot, T., Lowd, D., Kersting, K., Tadepalli, P., Shavlik, J.: Exploiting causal independence in Markov logic networks: Combining undirected and directed models. In: European Conference on Machine Learning (ECML) (2010)
12. Neville, J., Simsek, O., Jensen, D., Komoroske, J., Palmer, K., Goldberg, H.: Using relational knowledge discovery to prevent securities fraud. In: Proceedings of the 11th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM Press (2005)
13. Pearl, J.: Probabilistic Reasoning in Intelligent Systems: Networks of Plausible inference. San Mateo, CA: Morgan Kaufmann (1988)
14. Perlich, C., Provost, F.: Distribution-based aggregation for relational learning with identifier attributes. Machine Learning 62(1-2), 65–105 (2006)
15. Poole, D.: First-order probabilistic inference. In: Proceedings of the 18th International Joint Conference on Artificial Intelligence (IJCAI-03). pp. 985–991. Acapulco (2003)
16. Poole, D., Buchman, D., Natarajan, S., Kersting, K.: Aggregation and population growth: The relational logistic regression and Markov logic cases. In: UAI-2012 Workshop on Statistical Relational AI (2012)
17. de Raedt, L., Frasconi, P., Kersting, K., Muggleton, S.: Probabilistic Inductive Logic Programming: Theory and Applications. Springer-Verlag (2008)
18. Richardson, M., Domingos, P.: Markov logic networks. Machine Learning 42, 107–136 (2006)
19. de Salvo Braz, R., Amir, E., Roth, D.: Lifted first-order probabilistic inference. In Getoor, L. and Taskar, B., eds. Introduction to Statistical Relational Learning. MIT Press (2007)