Domain-Size Aware Markov Logic Networks

Happy Mittal
IIT Delhi

Ayush Bhardwaj
IIT Delhi

Vibhav Gogate
Univ. of Texas Dallas

Parag Singla
IIT Delhi

Abstract

Several domains in AI need to represent the relational structure as well as model uncertainty. Markov Logic is a powerful formalism which achieves this by attaching weights to formulas in finite first-order logic. Though Markov Logic Networks (MLNs) have been used for a wide variety of applications, a significant challenge remains that weights do not generalize well when training domain sizes are different from those seen during testing. In particular, it has been observed that marginal probabilities tend to extremes in the limit of increasing domain sizes. As the first contribution of our work, we further characterize the distribution and show that marginal probabilities tend to a constant independent of weights and not always to extremes as was previously observed. As our second contribution, we present a principled solution to this problem by defining *Domain-size Aware Markov Logic Networks (DA-MLNs)* which can be seen as re-parameterizing the MLNs after taking domain size into consideration. For some simple but representative MLN formulas, we formally prove that probabilities defined by DA-MLNs are well behaved. On a practical side, DA-MLNs allow us to generalize the weights learned over small-sized training data to much larger domains. Experiments on three different benchmark MLNs show that our approach results in significant performance gains compared to existing methods.

1 Introduction

A number of application domains in AI need to reason about the relational structure of the domain as well as handle uncertainty. The field of Statistical Relational AI [3, 12] achieves this merger by combining the power of logical and statistical representations. Markov Logic [2] is one such popular model which represents the underlying domain as a set of weighted first-order formulas and can be used as a template for constructing features for ground Markov networks. Markov logic has been successfully applied to a variety of AI applications including those in computer vision, NLP, biology and robotics [2].

Despite the applicability of Markov Logic, one significant challenge remains: MLNs suffer from serious generalization issues when training domain sizes are different from those seen during testing. Consider a problem of epidemic prediction in a town. Given the population in a town, and some information about whether each person is sick or not, we would like to predict whether there is an epidemic. The domain is modeled by a single MLN formula $w : sick(x) \Rightarrow epidemic$, where w could be learned from some training data. Figure 1 shows that the probability of epidemic quickly tends to 1 as the domain size increases for $w = 1$. In fact, this holds true for any fixed positive w.

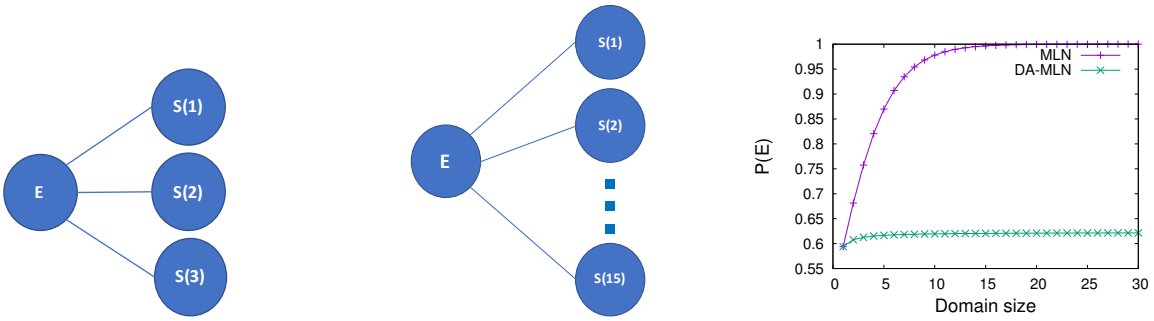
Hence, even if the probability of epidemic is strictly less than 1, the model will predict an epidemic with certainty if the test population is significantly larger than the training population. For illustration and comparison, we also plot (Figure 1c) the marginal probability with increasing domain size for standard MLNs as well our proposed solution for a formula weight $w = 1$. Training a different network for each domain size is clearly not a viable option.

In this paper, our contribution is twofold. First, we mathematically characterize the problem with the MLN representation as the domain size is increased for a given set of weights. Poole et al. [11] have previously shown that probabilities tend to extremes in the limit of increasing domain size. We show that their characterization in incomplete [1]. We show that in certain scenarios, the probabilities may not tend to the extreme but rather converge to a constant which is independent of the formula weights. We clearly separate out the two cases from each other in our work.

As our second contribution, we present a principled solution to the above problem by proposing a re-parameterization of MLNs which also takes into consideration the domain size. We refer to our re-

Proceedings of the 22nd International Conference on Artificial Intelligence and Statistics (AISTATS) 2019, Naha, Okinawa, Japan. PMLR: Volume 89.

[1] we present counterexamples to the propositions in their paper



(a) $|\Delta_x| = 3$, $P(E = 1) \approx 0.75$ (b) $|\Delta_x| = 15$, $P(E = 1) \approx 1$ (c) $|\Delta_x|$ vs $P(E = 1)$

Figure 1: Figure showing effect of increasing domain size of x (denoted by $|\Delta_x|$) on $P(epidemic)$ in MLN formula $w : sick(x) \Rightarrow epidemic$. Let $w = 1$. Figures 1a and 1b show the Ground Network and $P(epidemic)$ for $|\Delta_x| = 3$ and $|\Delta_x| = 15$ respectively. Figure 1c shows the plot of $|\Delta_x|$ vs $P(epidemic)$ in MLNs and DA-MLNs (our approach). For readability, $sick$ and $epidemic$ are abbreviated as S and E respectively.

parameterized MLNs as *Domain-Size Aware Markov Logic Networks (DA-MLNs)*. DA-MLNs are precisely a re-parameterization: there is a one to one correspondence between the parameters of the two models for any given domain size. Further, DA-MLNs allow us to successfully generalize weights across different domain sizes. This is demonstrated by our theoretical analysis on some simple but representative MLN formulas, where we show that the probabilities are well behaved in the limit. Our experiments on three benchmark networks with differing train and test domain sizes show impressive performance gains for DA-MLNs both in terms of AUC (area under the precision-recall curve) as well as data log-likelihood.

Our paper is organized as follows. We present some related work for our problem. Then we present the background on MLNs. This is followed by our characterization of the issue of generalization across varying domain sizes in MLNs. As a solution, we present DA-MLNs and prove their properties. Finally, we present our experiments and then conclude with directions for future work.

2 Related Work

Poole et al. [11] characterized the behavior for some simple classes of MLNs as the variable domain size goes to infinity. They show that in the limit, marginal probability of the query atom goes to extreme (i.e., either 0 or 1) if certain conditions are satisfied. However, in our analysis, we show that their characterization is incomplete. There are scenarios where probabilities do not go to the extreme in the limit but converge to a constant independent of the formula weights. Moreover, their paper only presents a characterization of the problem and does not provide a solution.

Jain et al. [5] propose Adaptive Markov Logic Networks (AMLNs) in which weights are learned over multiple databases of different sizes. The learned weights are approximated using a weighted combination of pre-defined basis functions which in turn are defined over underlying variable domain sizes. The coefficients of these basis functions become the parameters of the model. At test time, the formula weights are obtained by a linear combination of the coefficients and the basis functions (for the test domain sizes). Unfortunately, their approach seems to be limited by a choice of fixed set of basis functions. Further, in our experiments, their model performs poorly, fairing even worse than standard MLNs (see section 7.2).

In a recent piece work, Jaeger and Schulte [4] examine the dependence of marginal probabilities on the domain sizes of the variables for different Statistical Relational Learning (SRL) models. They show that if certain conditions are satisfied, several SRL models define a projective family of distributions in which inference does not depend on the variable domain sizes. Often, the set of required conditions for this to hold can be very restrictive. For instance, for Markov logic, they require that every predicate in a formula should have the same arity. This severely limits the applicability of the model. No solutions are proposed for a general scenario.

Finally, in another recent work, Kuzelka et al. [7] extend the duality between the maximum likelihood and max-entropy models to a relational setting, which they use to map the weights learned on a given training data set size to a different test data size. Their approach looks promising but lack of any experimental evidence questions the efficacy of the proposed model to real-world settings.

As opposed to existing approaches, we propose a very simple solution where we re-parameterize the MLN distribution by making the dependence on the domain size explicit via each ground atom's number of connections. Our approach has a very intuitive justification (see Section 5), comes with provable well behaved probabilities for a simple but representative class of MLN formulas, and most importantly, works surprisingly well in real-world settings,

offering a serious alternative to the standard MLNs.

3 Background

A First order logic (FOL) theory [13] consists of *constant*, *variable*, *function* and *predicate* symbols. We will use a strict subset of FOL which is function free with Herbrand interpretations [13]. A term is a constant or a logical variable. We will denote the logical variable by small English letters. The domain of a logical variable x is denoted by Δ_x. A first-order predicate takes terms as its arguments and defines a Boolean relation between the terms. A propositional predicate is a predicate which does not take any argument. An *atom* is a predicate symbol applied to a tuple of terms. A *literal* is an atom or its negation. A *formula* is defined recursively as follows : (a) A literal is a formula, (b) Negation of a formula f, denoted by $\neg f$ is also a formula, (c) If $f1$ and $f2$ are two formulas, then $f1 \wedge f2$ and $f1 \vee f2$ are also formulas. A constant is also referred to as a ground term. A ground atom is an atom all of whose terms are ground. A ground formula is a formula all of whose atoms have been grounded. In the rest of the paper, we will refer to a logical variable simply as variable for the convenience of notation. Also, we will work with typed theories, where a subset of variables may belong to the same type in which case their domains will be shared. We will use $\mathcal{X}$ to denote the set of all the variables appearing in the formulas of an MLN. We will use $\mathbf{\Delta}_{\mathcal{X}}$ to denote the set of domains of the variables in $\mathcal{X}$.

Definition 1. Markov Logic Network [2]: A Markov Logic Network (MLN) M is a set weighted formulas given as $\{(F_i, w_i)\}_{i=1}^m$, where F_i is a first order formula, and w_i is a real number, the weight of F_i. Given $\mathbf{\Delta}_{\mathcal{X}}$ i.e. domains of all the variables in all the formulas, M induces a Markov Network in which
(a) Each ground atom of M appears as a node, and two nodes have an edge iff they appear in same ground formula in M.
(b) Each grounding (ground formula) of F_i defines a Boolean feature f_{ij}, which is 1 if the ground formula is satisfied, otherwise 0. Weight of feature f_{ij} is w_i, where w_i is weight of F_i.

Intuitively, weight of a formula indicates how likely it is that the formula is true in the world. Higher the weight, higher is the probability of formula being true. In the extreme case, a weight can be infinite, which means a formula is true for every object i.e. the formula becomes pure first order logic formula. Now given an assignment to a set of ground atoms X (evidence), the probability distribution over the remaining ground atoms Y (query) is given by:

$$P_M(Y = y|X = x; w) = \frac{1}{Z_x} \exp\left(\sum_{i=1}^{m} w_i n_i(x, y)\right) \quad (1)$$

Here, m is the number of first-order formulas, w_i is the weight of first order formula F_i, $n_i(x, y)$ is the number of satisfied groundings of F_i under the assignment (x, y), and Z_x is the normalization constant given by $\sum_{y'} \exp\left(\sum_{i=1}^{m} w_i n_i(x, y')\right)$. For notational convenience, wherever clear from the context, we will denote Z_x simply by Z. Let V be a ground atom in M, then marginal probability of V being true can be calculated as :

$$P_M(V = 1) = \frac{Z_{V=1}}{Z_{V=0} + Z_{V=1}} = \frac{1}{1 + \frac{Z_{V=0}}{Z_{V=1}}} \quad (2)$$

where $Z_{V=v}$ denotes normalization constant with V restricted to value v in all assignments. Several algorithms have been proposed for inference and learning of parameters in MLNs [2]. Exact inference is often intractable so approximate algorithms such as those based on sampling are used in practice. Parameters are typically learned by maximizing the log-likelihood of the training data (denoted by $LL_M(w)$), whose gradient with respect to a particular weight w_i can be calculated from equation (1) and given as

$$\frac{\partial}{\partial w_i} LL_M(w) = n_i(x, y) - E[n_i(x, y)]_{P_M(Y|X)} \quad (3)$$

where (x, y) is the assignment of all ground atoms in the data. In the following exposition, we will be interested in understanding how the marginal distribution of a query (ground) atom changes as we vary the domain size of one or more variables in the theory. We will safely assume that all the formula weights are positive; a formula with negative weight can be equivalently replaced by a negated formula with a positive weight of the same magnitude, and a formula with weight zero can be ignored without changing the distribution. Though our description is in terms of MLNs, we believe our ideas can be easily extended to other similar SRL representations such as weighted parfactors [1].

4 Characterizing Issues in MLNs

4.1 Motivation

Let us revisit *epidemic* example in the Figure 1. The problem is modeled by a single MLN formula $w : sick(x) \Rightarrow epidemic$, where w could be learned from some training data. Let Δ_x denote the domain of x and let $|\Delta_x| = n$. $epidemic$ is the query predicate and we are interested in figuring out what happens to $P(epidemic)$ as $n \to \infty$.

In the ground network induced by the above MLN, the predicate $epidemic$ has n neighbors $sick(1), sick(2), \ldots, sick(n)$. Figures 1a and 1b depict the ground networks for $n = 3$ and $n = 15$, respectively. With increasing n, the number of neighbors of the query predicate increases, resulting in extreme probabilities as $n \to \infty$. This holds true independent of the value of the formula weight w. We note that this problem is not directly due to the increasing domain size, but rather due to a large number of connections the query

atom is involved in. For example, if our MLN was instead $w : sick(x) \Rightarrow unhappy(x)$, then increasing the domain size would not affect the probability of $unhappy(x)$. In this section, we are interested in mathematically characterizing this problem with increasing domain size.

In our exposition below, let Q be the query predicate, and let $Q(1)$ denote a grounding of Q whose marginal we are interested in. As a shorthand, we will use $P(Q(1))$ to denote the marginal probability of $Q(1)$ being true. For our current exposition, we will assume that the underlying theory is evidence-free. We will partly relax this assumption in section 6. For all the proofs and propositions marked with (*), detailed proofs can be found in the supplement.

4.2 Earlier work

Poole et al. [11] have earlier characterized the problem of probabilities going to the extreme in the limit. But careful analysis reveals that their exposition was only partially correct i.e. there are cases where the marginal probabilities become a constant independent of the underlying weight, but do not converge to either 0 or 1. Here we restate the proposition given by them, restricted to a single formula:

Proposition 0. *Consider an MLN M with a single formula of the form $w : Q(x) \vee R_1(y_1) \vee \ldots \vee R_k(y_k) \vee P_1 \vee \ldots \vee P_m$. Here, $k \geqslant 1$ and $m \geqslant 0$. $|\Delta_x| = r$ where $r \geqslant 1$ is a fixed constant. Also $|\Delta y_1| = \ldots = |\Delta y_k| = n$. $P_1, \ldots, P_m$ are propositional predicates. Then either $P_M(Q(1))$ is a constant (independent of n), or $\lim_{n\to\infty} P_M(Q(1))$ is either 1 or 0. [11]*

Our *epidemic* MLN is an example which satisfies this proposition, with $|\Delta x| = 1$, $k = 1$, and $m = 0$. We now show that this proposition doesn't always hold. Below we present a counterexample to this proposition.

Counterexample * : Consider an MLN M consisting of one formula $w : Q(x) \vee R(y) \vee P$, where $|\Delta_x| = 1$ and $|\Delta_y| = n$. Then, using Eq 2, we have:

$$P_M(Q(1)) = \frac{2^{n+1}e^{wn}}{(2^n e^{wn} + (1+e^w)^n) + 2^{n+1}e^{wn}} \quad (4)$$

This expression is not independent of n. Further, $\lim_{n\to\infty} P(Q(1)) = \frac{2}{3}$ which shows that $P(Q(1))$ is neither 0 nor 1 in the limit. Hence, this is a counterexample to Proposition 0. For a detailed derivation, see supplement.

4.3 A More Correct Analysis

The primary problem with Poole et al. [11]'s characterization is that they failed to capture the impact of additional propositional predicates in the formula (e.g., P_j's in Proposition 0), which do not let the probabilities to go to the extreme in the limit of increasing domain size. For instance, in the *epidemic* example, there was no propositional predicate (other than the query) and the probabilities went to the extreme. On the other hand, in our counterexample, the presence of the propositional predicate P forced the limiting probability to be different from either 0 or 1. Additionally, Poole et al. miss out on cases when the domain of the query predicate argument (i.e., Δ_x in Prop. 0) also goes to infinity in the limit. Next, we present a more complete (and correct) characterization of this problem.

Proposition 1. *Consider an MLN M with a single formula of the form $w : Q(x) \vee R_1(y_1) \vee \ldots \vee R_k(y_k)$. Here $k \geqslant 1$. Also $|\Delta y_1| = \ldots = |\Delta y_k| = n$, and $|\Delta x| = r$, where $r \geqslant 1$ is some constant. $\lim_{n\to\infty} P_M(Q(1))$ is 1.*

Proof * : We have

$$\frac{Z_{Q(1)=0}}{Z_{Q(1)=1}} = \lim_{n\to\infty} \frac{k-1}{2^n} + \left(\frac{1+e^{-w}}{2}\right)^n 2^{r-1} = 0 \quad (5)$$

Substituting (5) in (2), we get $\lim_{n\to\infty} P_M(Q(1)) = 1$.

Proposition 2. *Consider an MLN M having a single formula of the form $w : Q(x) \vee R(y) \vee P_1 \vee P_2 \ldots \vee P_m$, where $|\Delta_x| = 1$, $|\Delta y| = n$. Then $\lim_{n\to\infty} P_M(Q(1)) = \frac{2^m}{2^{m+1}-1}$*

Proof * : We have

$$\lim_{n\to\infty} \frac{Z_{Q(1)=0}}{Z_{Q(1)=1}} = \left(1 - \frac{1}{2^m}\right) \quad (6)$$

Substituting (6) in (2), $\lim_{n\to\infty} P_M(Q(1)) = \frac{2^m}{2^{m+1}-1}$

Proposition 3. *Consider an MLN M with a single formula of the form $w : Q(x) \vee R(x)$, where $|\Delta x| = |\Delta y| = n$. Then $\lim_{n\to\infty} P_M(Q) = \frac{3}{4}$*

Proof * : We have

$$\lim_{n\to\infty} \frac{Z_{Q(1)=0}}{Z_{Q(1)=1}} = \lim_{n\to\infty} \frac{2^{n-1}e^{n^2 w}}{3 * 2^{n-1}e^{n^2 w}} = \frac{1}{3} \quad (7)$$

Putting Eq (7) in Eq (2), we get $\lim_{n\to\infty} P_M(Q(1)) = \frac{3}{4}$.

In our next proposition, we move to a more general case, where the formula also contains binary predicates.

Proposition 4. * *Consider an MLN M with single formula of the form $w : Q(x) \vee P(x,y) \vee R(y)$. Here $|\Delta x| = r$, where $r \geqslant 1$ is some constant, and $|\Delta y| = n$. Then $\lim_{n\to\infty} P_M(Q(1))$ is 1.*

Interestingly, all the MLNs considered in Propositions (1-4) satisfy the Single Occurrence (SO) Property [9]. Examining whether there is a more general connection between MLN formulas satisfying SO property and probabilities going to extreme (or tending to a constant) is a direction for future work.

In this section, we have presented a detailed characterization of the marginal distribution of the query atom in the limit of increasing domain size. Our description better describes the issues involved compared to earlier work. We note that our analysis is still limited to relatively simple

MLNs having a single formula. [2] Nevertheless, we expect that these issues will only be aggravated as we move to more complex MLNs. Hence, we need to fix the underlying representation if MLNs were to generalize across varying domain sizes. Next, we present a principled solution to this problem based on a re-parameterized distribution.

5 Domain-size Aware Markov Logic Networks (DA-MLNs)

5.1 Intuition

In the previous section, we characterized how the marginal probability of query is affected as the domain size of the variables in the network is increased. In several cases, the probability goes to extreme (Proposition 1, 4) and in others, converges to a constant independent of the formula weight (Propositions 2, 3). Intuitively, as discussed in Section 4.1, the key reason behind this problem is the large number of connections that the query atom gets involved in. The aggregate effect of these connections renders the formula weight inconsequential in determining the marginal probability in the limit of large domain size. Arguably, the solution should also stem from the same phenomenon: if we could somehow make the distribution depend explicitly on the number of connections each atom is involved in, we may be able to counter the effect of increasing domain size on the distribution. To formalize this notion, we start with a few definitions.

5.2 Definitions

Notation: Given a formula F, we will use $Preds(F)$ to denote the set of predicate occurrences in F. By predicate occurrences, we also mean to count the repeated occurrences of the same predicate in a formula. For example, if the formula F is $P(x) \vee P(y)$, $Preds(F)$ will contain two occurrences of P, one for each position that P occurs in F. We will use $Vars(P)$ to denote the set of (logical) variables appearing in P (in the context of formula F).

Definition 2. (NumConnections) Let F be a first order formula. Let $P \in Preds(F)$ be a predicate occurrence in F. As defined earlier, let $Vars(P)$ denote the set of (logical) variables appearing in P. Further, let $Vars(P)^-$ denote the set of logical variables in F not appearing in P, Then, the number of connections c for P in F is defined as $\max\left(1, \prod_{x \in Vars(P)^-} |\Delta_x|\right)$.

Intuitively, the number of connections of a predicate P (in a formula F) is the number of groundings of F that each (any) ground atom corresponding to P is involved in [3].

Example: Given a formula F as $w : P(x, y) \Rightarrow Q(x)$ where $\Delta_x = \Delta_y$, the number of connections for P and Q in F are 1 and $|\Delta_x|$, respectively. Since each predicate in a formula can have different number of connections, we define a connection-vector of a first order formula as follows.

Definition 3. (Connection-vector) Let F be a first-order formula. Let the number of predicate occurrences in F be given by $m = |Preds(F)|$. Let $(P_1, P_2, \cdots, P_m)$ be some ordering over these predicate occurrences (for instance, it can simply be the ordering in which each predicates appears in the formula). Then, the connection vector v for F is defined as $(c_1, c_2, \cdots, c_m)$ where c_j is the number of connections of P_j in F.

For the example formula $w : P(x, y) \Rightarrow Q(x)$ considered above, the connection vector is $(1, |\Delta_x|)$. Intuitively, the magnitude of the connection vector captures the number of connections that predicates in the formula are involved in. We somehow need to aggregate the elements of the connection vector v to get a single number such that the overall strength of the connection of the formula can be captured.

Definition 4. (Scaling-down Factor) Let F be a first-order logic formula. Let v be its connection vector. Let $\Psi : \mathbb{R}^d \rightarrow \mathbb{R}$ denote an aggregation function over the elements of v where where d denotes the size of v. Given the function ψ, we define $s = \Psi(v)$ as the scaling-down factor for formula F.

Several choices for Ψ are possible. In our exposition, we choose Ψ as the $\max$ function. This allows us to prove some of the properties of our resulting formulation in a seamless manner and this also works well in practice. Exploring other alternatives for Ψ function such as $\sum$ is a direction for future work. We are now ready to define our re-parameterized MLNs.

Definition 5. Re-parameterized MLNs: Let M denote an MLN with the set of weighted formulas given by $\{F_i, w_i\}_{i=1}^m$. We define a new parameterization with the set of weighted formulas given by $\{F_i, w_i'\}_{i=1}^m$ where F_i is inherited from M and $w_i' = w_i * s_i$ where s_i is the scaling-down factor for F_i given the variable domains $\mathbf{\Delta}_\mathcal{X}$. We refer to this re-parameterization as *Domain-Size Aware Markov Logic Network* (DA-MLN). The distribution defined by DA-MLN D is given as:

$$P_D(Y = y | X = x; w') = \frac{1}{Z_x} \exp\left(\sum_{i=1}^{m} \frac{w_i'}{s_i} n_i(x, y)\right)$$

where the symbols used are same as in Eq 1. Additionally, s_i denotes the scaling-down factor for F_i. Also $Z_x = \sum_{y'} \exp\left(\sum_{i=1}^m \frac{w_i'}{s_i} n_i(x, y')\right)$ Intuitively, DA-MLNs make the dependence on the number of connections explicit.

Corollary 1. *For any given set of variable domains $\mathbf{\Delta}_\mathcal{X}$, the distribution defined by a DA-MLN D with weighted formulas $\{(F_i, w_i')\}_{i=1}^m$, is same as the distribution defined the MLN with weighted formulas $\{(F_i, w_i)\}_{i=1}^m$, with $w_i = \frac{w_i'}{s_i}$ where s_i denotes the scaling down factor for formula F_i.*

[2] Extending to a more general setting is a future direction.

[3] Each ground atom will be involved in the same number of formula groundings due to the MLN structure.

5.3 Inference and Learning

Since DA-MLNs are nothing but a re-parameterization of MLNs, inference problem in DA-MLNs reduces to one in MLNs, and hence, the same set of algorithms can be employed (see Section 3). Learning in DA-MLNs can be performed by maximizing the log-likelihood (denoted by $LL_D(w')$) of the data, whose gradient with respect to the weight parameter w'_i is given by:

$$\frac{\partial}{\partial w'_i} LL_D(w') = \frac{n_i(x,y)}{s_i} - E\left[\frac{n_i(x,y)}{s_i}\right]_{P_D(Y|X)}$$

This equation is identical to the eq 3, except for the presence of the scaling-down factor s_i. Hence, the same set of algorithms based on gradient descent can be employed.

5.4 Properties

Next, we characterize the behavior of DA-MLNs in the limit of variable domain sizes going to infinity. We will show that at least in some simple but representative cases, the probabilities are well behaved i.e., they are neither extreme nor become independent of the formula weight in the limit.

Proposition 5. *Consider a DA-MLN D with a single formula of the form $w : Q(x) \vee R(y)$. Let $|\Delta_x| = 1$. Further, let $|\Delta_y| = n$. Then,* $\lim_{n\to\infty} P_D(Q(1)) = \frac{1}{1+e^{\frac{-w}{2}}}$.

Proof: In order to compute $P_D(Q(1))$, we need to compute $\frac{Z_{Q(1)=0}}{Z_{Q(1)=1}}$: We have

$$\lim_{n\to\infty} \frac{Z_{Q(1)=0}}{Z_{Q(1)=1}} = \lim_{n\to\infty} \frac{(1+e^{\frac{w}{n}})^n}{e^w * 2^n} = e^{\frac{-w}{2}}$$

So $\lim_{n\to\infty} P_D(Q(1)) = \frac{1}{1+e^{\frac{-w}{2}}}$.

In particular, note that our *epidemic* examples falls in the category of Proposition 5 where $Q(1)$ is epidemic and $\neg R(x)$ is $sick(x)$. Hence, the $P(epidemic)$ won't go to extreme as n goes to ∞. Proposition 5 is the analogue of Proposition 1 (for MLNs) albeit with a slightly restricted form i.e., with $|\Delta_x| = 1$ and a single non query predicate in the formula. Unlike Proposition 1, the limiting probability does not go to extreme in this case. Next, we present a proposition which handles the case of formulas with (non-query) propositional predicates.

Proposition 6. *Consider a DA-MLN D having a single formula of the form $w : Q(x) \vee R(y) \vee P_1 \vee P_2 ... \vee P_m$, where $|\Delta_x| = 1$ and $|\Delta y| = n$. Then* $\lim_{n\to\infty} P_D(Q(1))$ *is a (non-constant) function of w.*

Proof * : We have

$$\lim_{n\to\infty} \frac{Z_{Q(1)=0}}{Z_{Q(1)=1}} = \left(1 - \frac{1}{2^m}\right) + \frac{1}{2^m - 1} e^{-\frac{w}{2}}$$

$$\lim_{n\to\infty} P_D(Q(1) = 1) = \frac{1}{1 + \left(1 - \frac{1}{2^m}\right) + \frac{1}{2^m - 1} e^{-\frac{w}{2}}}$$

It clearly shows that the limiting marginal probability depends on w. Proposition 6 is an analogue of Proposition 2 for MLNs. Unlike Proposition 2, there is a clear dependence on weight in this case. Next, we present the proposition which deals with increasing domain size for the argument of the query predicate (in addition to the other variables).

Proposition 7. * *Consider a DA-MLN D with a single formula of the form $w : Q(x) \vee R(y)$, where $|\Delta x| = |\Delta y| = n$. Then* $\lim_{n\to\infty} P_D(Q(1)) = f(w)$, *where $f(w)$ is a (non constant) function of w .*

We present the analogue of proposition 4 for DA-MLNs.

Proposition 8. * *Consider a DA-MLN D with a single formula of the form $w : Q(x) \vee P(x,y) \vee R(y)$. Here $|\Delta x| = r$, where $r \geqslant 1$ is some constant, and $|\Delta y| = n$. Then* $\lim_{n\to\infty} P_D(Q(1)) = f(w)$, *where $f(w)$ is a (non constant) function of w.*

All of the propositions above clearly demonstrate that query marginals produced under DA-MLNs are well-behaved unlike in MLNs where query marginals either went to extreme or became independent of the formula weight under very similar conditions. This clearly presents DA-MLNs as a strong alternative for MLNs for a variety of reasoning tasks. Proving these nice properties for more complex DA-MLN structures is a direction for future work.

6 Handling Evidence

In this section, we will relax the evidence-free assumption. Our intuition (Section 4.1) is built around the fact that as the domain size of variables appearing in non-query predicates increases, number of neighbors also increases, and hence, probabilities tend to extreme. DA-MLNs can counter this phenomenon by re-parameterizing MLNs by explicitly taking domain size in account. A natural question arises: what happens in the presence of evidence predicates since their groundings will not be part of the network. Interestingly, we show that in MLNs, under certain conditions, even in the presence of evidence, marginals tend to extreme. On the other hand, DA-MLNs still remain well behaved.

Proposition 9. * *Consider an MLN M with single formula of the form $w : Q(x) \vee P(y)$. Here $|\Delta y| = n$. Let $|\Delta x| = 1$. Suppose P is evidence predicate, i.e., all its groundings are given to be true or false. If the ratio of true and false groundings of P remains constant with respect to n, then* $\lim_{n\to\infty} P_M(Q(1)) = 1$.

Now we show that in DA-MLNs, these marginals do not go to extreme.

Proposition 10. * *Consider a DA-MLN D with single formula of the form $w : Q(x) \vee P(y)$. Here $|\Delta y| = n$. Let $|\Delta x| = 1$. Suppose P is evidence predicate. If the ratio of true and false groundings (denoted by r) of P remains*

constant with respect to n*, then* $\lim_{n\to\infty} P_D(Q(1)) = \frac{1}{1+e^{(r-1)w}}$.

Proving these results for a more general class of MLN formulas is a direction for future work.

7 Experiments

The goal of our experiments was to examine whether DA-MLNs can help us generalize better when training and testing domains come from different sizes. In particular, we were interested in a setting where the training domain is much smaller compared to the testing domains - a scenario typically expected in real life setting due to the high cost of obtaining labeled data. To answer these questions, we experimented on three benchmark MLN domains where we kept the size of the training data fixed and varied the size of the testing data. We then compared the performance of three competing algorithms: (a) DA-MLN (current work) (b) (Standard) MLN [2] (c) Adaptive MLN (AMLN) [5]. We compare each algorithm on two different metrics: (1) AUC: area under the precision-recall curve (2) Average CLL (Conditional Log-likelihood): average over log marginal probabilities of query atoms (given evidence).

For our experiments, we used our own implementation of the Alchemy [6] software [4]. For each of our models, we learned the weights using the PSCG [8] algorithm and ran the algorithm until convergence or up to a maximum of 100 iterations. For learning the coefficients of the basis functions in AMLNs, we implemented a utility as described by Jain et al. [5]. For each of the models, the inference was performed using Gibbs sampling with 5000 samples per (query) ground atom in each case. All our experiments were run on servers with 40 cores and up to 256 GB of RAM. We next describe the details of our datasets, methodology and the results of our experimental evaluation.

7.1 Datasets and Methodology

We used three benchmark domains used in the earlier literature : Friends & Smokers (FS) [14], IMDB [10], and WebKB [6].

Friends & Smokers (FS): This dataset contains information about the smoking habits of people, their friendship relationships and whether they suffer from cancer or not. The dataset contains MLN theory contains three predicates: *Smokes(person)*, *Cancer(person)*, *Friends(person,person)*. We generated the actual data to model real-life communities. The entire population of size n was first divided into $\sqrt{n}$ groups. Each group was labeled Smoking or Non-Smoking randomly, with the probability of Smoking group 0.3. The probability of beings friends within the same group was set to 0.8, and the probability of being friends outside the group was set to 0.1. Each person was labeled Smoker and Non-Smoker depending on their group's smoking habits. For the smoking groups, the probability of smoking was set to 0.7. For a non-smoking group, this was set to 0.1. A smoking person was set to have cancer with probability 0.5, and a non-smoking person with probability 0.1. We learned on randomly generated datasets with domain sizes 20, 40, 60, 80, 100, and inferred on randomly generated datasets of sizes varying from 50 to 500.

IMDB: We downloaded this dataset from an online kaggle competition [5]. The dataset contains information about 1000 movies and their casts. This dataset is defined using four predicates: *Actor(person)*, *Director(person)*, *Movie(title, person)*, and *WorkedUnder(person, person)*. For creating varying size data, we randomly chose a subset of directors, picked the movies done by them and the actors who worked in those movies. We learned on 4 randomly generated subsets having 2, 4, 5 and 10 directors, and inferred on randomly generated subsets with directors varying from 10 to 50. We made sure there was no overlap between the training and testing set of directors.

WebKB: The dataset is publicly available for download from the Alchemy website [6] which contains information about pages from various US universities. We worked with a set of about 1300 pages from one of the universities. The dataset is defined using three predicates: Has(page,word), Links(page, page), and Class(page, category). Each webpage consists of words and hyperlinks. Each webpage can belong to a subset of categories: person, student, faculty, professor, department, research project, and course. For creating varying size data, we randomly chose a subset of web pages and the associated entities. We learned on two subsets corresponding to randomly chosen 50 and 100 web pages [7]. We tested on subsets of the data corresponding to randomly chosen webpages (different from training) with the number varying from 50 to 800. Figure 2 (lower half) shows the details of the formulas for each domain. Additionally, we had a unary clause for every predicate. At the bottom of each set of rules, we specify which predicates were treated as query (evidence) during learning and inference, respectively, for each domain.

7.2 Results

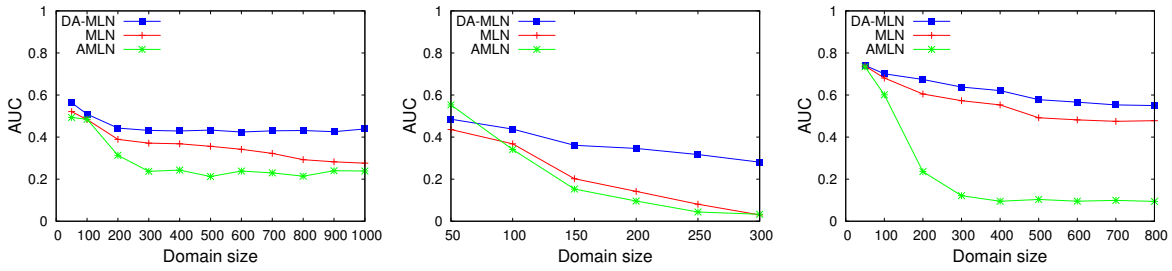
Figures 2a, 2b, and 2c plot the AUCs on FS, IMDB, and WebKB datasets, respectively, as we vary the test data sizes. As expected, the three models result in very similar performance on smaller datasets (which are similar in size to the respective training sets). DA-MLNs clearly outperform the other two algorithms for larger domains with the gain successively increasing with domain size. For WebKB, the difference between MLN and DA-MLN stabilizes after a point. This is due to the fact that in WebKB, we

[4] https://github.com/happy2332/alchemy-java

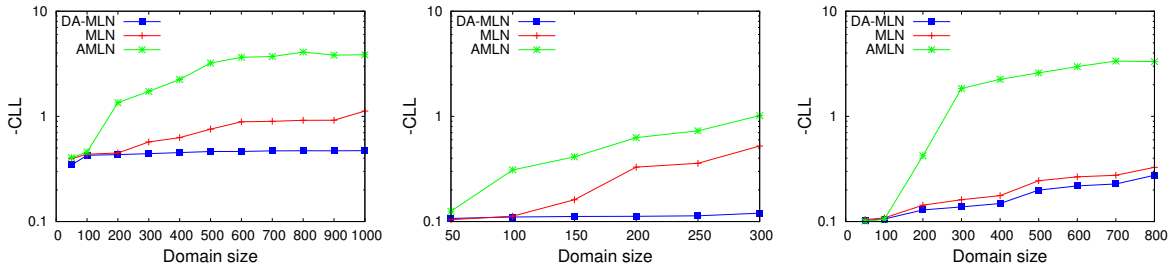
[5] https://www.kaggle.com/PromptCloudHQ/imdb-data/data

[6] http://alchemy.cs.washington.edu

[7] For AMLN, we supplemented with another subset having 75 randomly chosen webpages



(a) Domain Size vs AUC (FS) (b) Domain size vs AUC (IMDB) (c) Domain size vs AUC (WebKB)



(d) Domain Size vs CLL (FS) (e) Domain size vs CLL (IMDB) (f) Domain size vs CLL (WebKB)

FS [14]
Smokes(p) ⇒ Cancer(p)
Smokes(p1) ∧ Friend(p1, p2) ⇒ Smokes(p2)
`(Learning)` Query: All.
`(Inference)` Query: Cancer, Smokes(Random 50%)
WebKB [6]
Has(p,+w) ⇒ Class(p,+c)
¬Has(p,+w) ⇒ Class(p,+c)
Class(p1,+c1) ∧ Links(p1,p2) ⇒ Class(p2,+c2)
`(Learning)` Query: Class
`(Inference)` Query: Class

IMDB [10]
WrkdUndr(p1,p2) ⇒ Act(p1)
WrkdUndr(p1,p2) ⇒ Dir(p2)
Dir(p1) ∧ Act(p2) ∧ Mov(m,p1) ∧ Mov(m,p2) ⇒ WrkdUndr(p2,p1)
Dir(p1) ∧ Act(p2) ∧ Mov(m,p2) ∧ WrkdUndr(p2,p1) ⇒ Mov(m,p1)
Dir(p1) ∧ Act(p2) ∧ Mov(m,p1) ∧ WrkdUndr(p2,p1) ⇒ Mov(m,p2)
Dir(p1) ∧ Act(p2) ⇒ WrkdUndr(p2,p1)
`(Learning)` Query: All.
`(Inference)` Query: All (Random 50%)

Figure 2: Results and Rules for FS, IMDB, and WebKB datasets.

do not see a much change in the number of connections with increasing domain size. For the largest values of domain sizes considered, DA-MLNs outperform MLNs by 0.17 AUC points (FS), 0.25 AUC points (IMDB) and 0.08 AUC points (WebKB). AMLN performs even worse than MLN in each case.

Figures 2d, 2e, and 2f plot the Negative CLLs (in log-scale) on FS, IMDB, and WebKB datasets respectively (lower is better). The general trend is the same as that for AUC, with DA-MLN outperforming both MLN and AMLN, with increasing gain with domain size.

Both these sets of results clearly demonstrate the efficacy of our approach in generalizing the parameters learned over small-sized training data to much larger domains. Though we observe some drop in performance with increasing domain size, it is significantly less compared to other approaches which can perform abysmally, e.g., MLN for IMDB has an AUC of 0.03 at domain size of 300. Whether we can we modify DA-MLN such that there is no drop in performance whatsoever with increasing domain size is a direction for future work.

8 Conclusion and Future Work

In this paper, we have addressed the problem with the standard Markov Logic representation in the limit of increasing domain size. We have shown that in the limit of domain size approaching infinity, the marginal probabilities either tend to the extreme or converge to a constant (independent of formula weight) even for some simple MLN formulas. As a solution, we have proposed a re-parameterization of MLNs, referred to as Domain-size Aware Markov Logic Networks (DA-MLNs). While defining the distribution, they take into the account the number of connections each ground atom has in the ground network. We show that probabilities are well behaved in the limit i.e. they depend on the learned weights in DA-MLNs at least in some simple cases. Experiments on three benchmark domains show the efficacy of DA-MLNs possibly establishing them as a superior alternative to MLNs. Directions for future work include developing the theory around DA-MLNs further, running additional experiments on newer domains, and experimentally comparing with other approaches for which we currently do not have any implementation e.g., [7].

9 Acknowledgements

Happy Mittal is supported by the TCS Research Scholar Program. Parag Singla is supported by the Visvesvaraya Young Faculty Fellowships by Govt. of India and IBM SUR awards. Vibhav Gogate is supported by the National Science Foundation grants IIS-1652835 and IIS-1528037. Both Parag and Vibhav are also supported by the DARPA Explainable Artificial Intelligence (XAI) Program with number N66001-17-2-4032. Any opinions, findings, conclusions or recommendations expressed in this paper are those of the authors and do not necessarily reflect the views or official policies, either expressed or implied, of the funding agencies.

References

[1] R. de Salvo Braz, E. Amir, and D. Roth. Lifted first-order probabilistic inference. In *Proc. of IJCAI-05*, pages 1319–1325, 2005.

[2] P. Domingos and D. Lowd. *Markov Logic: An Interface Layer for Artificial Intelligence*. Synthesis Lectures on Artificial Intelligence and Machine Learning. Morgan & Claypool Publishers, 2009.

[3] L. Getoor and B. Taskar, editors. *Introduction to Statistical Relational Learning*. MIT Press, 2007.

[4] Manfred Jaeger and Oliver Schulte. Inference, learning, and population size: Projectivity for srl models. In *Proc. of IJCAI-18 Wkshp. on Statistical Relational AI*, 2018.

[5] Dominik Jain, Andreas Barthels, and Michael Beetz. Adaptive markov logic networks: Learning statistical relational models with dynamic parameters. In *Proc. of ECAI-10*, pages 937–942, 2010.

[6] S. Kok, M. Sumner, M. Richardson, P. Singla, H. Poon, D. Lowd, J. Wang, and P. Domingos. The Alchemy system for statistical relational AI. Technical report, University of Washington, 2008. http://alchemy.cs.washington.edu.

[7] Ondrej Kuzelka, Yuyi Wang, Jesse Davis, and Steven Schockaert. Relational marginal problems: Theory and estimation. In *Proc. of AAAI-2018*, 2018.

[8] Daniel Lowd and Pedro Domingos. Efficient weight learning for markov logic networks. In *Proc. of PKDD-07*, pages 200–211. Springer, 2007.

[9] H. Mittal, P. Goyal, V. Gogate, and P. Singla. New rules for domain independent lifted MAP inference. In *Proc. of NIPS-14*, pages 649–657, 2014.

[10] Happy Mittal, Shubhankar Suman Singh, Vibhav Gogate, and Parag Singla. Fine grained weight learning in markov logic networks. In *Proc. of IJCAI-16 Wkshp. on Statistical Relational AI*, 2016.

[11] David Poole, David Buchman, Seyed Mehran Kazemi, Kristian Kersting, and Sriraam Natarajan. Population size extrapolation in relational probabilistic modelling. In *Proc. of SUM-14*, pages 292–305. Springer, 2014.

[12] Luc De Raedt, Kristian Kersting, Sriraam Natarajan, and David Poole. Statistical relational artificial intelligence: Logic, probability, and computation. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 10(2):1–189, 2016.

[13] S. J. Russell and P. Norvig. *Artificial Intelligence - A Modern Approach (3rd edition)*. Pearson Education, 2010.

[14] P. Singla and P. Domingos. Lifted first-order belief propagation. In *Proc. of AAAI-08*, pages 1094–1099, 2008.