Non-parametric Domain Approximation for Scalable Gibbs Sampling in MLNs

Abstract

MLNs utilize relational structures that are ubiquitous in real-world situations to represent large probabilistic graphical models compactly. However, as is now well-known, inference complexity is one of the main bottlenecks in MLNs. Recently, several approaches have been proposed that exploit approximate symmetries in the MLN to reduce inference complexity. These approaches approximate large domains containing many objects with much smaller domains of *meta-objects* (or cluster-centers), so that inference is considerably faster and more scalable. However, a drawback in most of these approaches is that it is typically very hard to tune the parameters (e.g., number of clusters) such that inference is both efficient and accurate. Here, we propose a novel non-parametric approach that trades-off solution quality with efficiency to automatically learn the optimal domain approximation. Further, we show how to perform Gibbs sampling effectively in a domain-approximated MLN by adapting the sampler according to the approximation. Our results on several benchmarks show that our approach is scalable, accurate and converges faster than existing methods.

1 Introduction

Markov Logic Networks (MLNs) offer a convenient way to express uncertain domain knowledge in the form of weighted first-order formulas. However, probabilistic inference in MLNs is well-known to be a notoriously challenging problem since the size of the Markov network underlying an MLN (*ground* Markov network) typically grows at an exponential rate as we increase the number of real-world objects that the MLN is defined over. Therefore, for MLNs to be practically applicable, controlling inference complexity in large domains is essential.

Efficient inference for MLNs and Statistical Relational Models, in general, has received a great deal of attention from the research community. In particular, the idea of *lifting* inference over the domain of the MLN such that we can perform inference over groups of objects instead of individual objects has been widely explored over the last few years. The main idea in lifted inference is to reduce complexity by taking advantage of exchangeable variables in the model. Several exact and approximate inference methods have been proposed over the past few years starting with the work by Poole [18], including, FOVE [4], WFOMC [24], Probabilistic Theorem Proving (PTP) [7] and lifted inference with soft evidence [3]. Popular approximate lifted inference methods include [14, 15, 20, 10, 8, 16, 27, 2, 1].

More recently, it has been understood that lifting even approximate inference techniques is typically insufficient to ensure scalability in MLNs. The key problem with most traditional lifting techniques is their over-reliance on exact symmetries which are either not present in real-world situations or are known to be broken fairly easily in the presence of evidence [25]. Therefore, a new class of methods that utilize approximate symmetries thereby forgoing strong theoretical guarantees in lieu of scalability have been proposed making MLNs much more attractive from a practical perspective. In this work, we develop one such novel approach by learning the lifting strategy automatically using non-parametric clustering and integrating a Gibbs sampler that adapts itself with the learned strategy.

The idea of clustering as a pre-processing step to reduce inference complexity has been proposed in Broeck and Darwiche [25], Venugopal and Gogate [28] and more recently explored by Sarkhel et al. [19] in the context of MAP inference. Specifically, the key idea in these approaches is to pre-process the domains in the original MLN and generate new, smaller domains consisting of meta-objects that implicitly represent a cluster of objects in the original MLN. However, the main problem with existing methods is that it is very hard to tune the parameters (e.g., number of clus-

ters) such that we select the optimal clustering for an MLN balancing accuracy with complexity of inference. For example, consider a simple MLN with just one unit clause, $\mathtt{R}(x)\ w$, then, it turns out that given any evidence, the optimal number of clusters required to represent the complete domain of x accurately is equal to 3. Specifically, we form one cluster with all atoms for which the predicate R is known to be true, one cluster that contains all atoms for which R is known to be false and a third cluster of the remaining objects. On the other hand, for a more complex MLN, such as, $\mathtt{R}(x, y) \wedge \mathtt{R}(y, z) \Rightarrow \mathtt{R}(z, x)\ w$, choosing the correct number of clusters is not obvious. In this paper, we develop a novel method that automatically finds the clustering that is in some sense optimal. Specifically, we make the following contributions.

1. We develop a fully non-parametric approach to approximate the domain in an MLN with a new domain of meta-objects that correspond to the optimal clustering for that domain.
2. We integrate our clustering method with a Gibbs sampler that adapts itself based on the domain approximation in order to minimize sampling errors.

We perform an evaluation of our approach in terms of both accuracy and convergence on benchmarks chosen from Alchemy [11]. Our results clearly illustrate that our approach is scalable, yields accurate results and importantly converges quickly on large models.

2 Background and Preliminaries

2.1 First-Order Logic

We assume a strict subset of first-order logic, called finite Herbrand logic. Thus, we assume that we have no function constants and finitely many object constants. A first-order knowledge base (KB) is a set of first-order formulas. A formula in first-order logic is made up of quantifiers ($\forall$ and $\exists$), logical variables, constants, predicates and logical connectives ($\vee$, $\wedge$, $\neg$, $\Rightarrow$, and $\Leftrightarrow$). We denote logical variables by lower case letters (e.g., x, y, z, etc.) and constants by strings that begin with an upper case letter (e.g., A, Ana, Bob, etc.). Constants model objects in the real-world domain. A predicate is a relation that takes a specific number of arguments (called its arity) as input and outputs either True (synonymous with 1) or False (synonymous with 0). A term is either a logical variable or a constant. We denote predicates by strings in typewriter font (e.g., R, S, Smokes, etc.) followed by a parenthesized list of terms.

A first-order formula is recursively defined as follows: (i) An atomic formula is a predicate; (ii) Negation of an atomic formula is a formula; (iii) If f and g are formulas then connecting them by binary connectives such as $\wedge$ and $\vee$ yields a formula; and (iv) If f is a formula and x is a logical variable then $\forall x f$ and $\exists x f$ are formulas.

We assume that each argument of each predicate is typed and can only be assigned to a fixed subset of constants. By extension, each logical variable in each formula is also typed. We further assume that all first-order formulas are disjunctive (clauses), have no free logical variables (namely, each logical variable is quantified), have only universally quantified logical variables (CNF), and have no constants. Note that all first-order formulas can be easily converted to this form. A ground atom is an atom that contains no logical variables. A possible world, denoted by ω, is a truth assignment to all possible ground atoms that can be formed from the constants and the predicates.

2.2 Markov Logic Networks

Markov logic networks (MLNs) combine Markov networks and first-order logic. Formally, an MLN is a set of pairs (f_i, θ_i) where f_i is a formula in first-order logic and θ_i is a real number. Given a set of constants, an MLN represents a ground Markov network, defined as follows. We have one binary random variable in the Markov network for each possible ground atom. We have one propositional feature for each possible grounding of each first-order formula. The weight associated with the feature is the weight attached to the corresponding formula. The ground Markov network represents the following probability distribution:

$$\Pr(\omega) = \frac{1}{Z} \exp\left(\sum_i \theta_i N_{f_i}(\omega)\right) \tag{1}$$

where $N_{f_i}(\omega)$ is the number of groundings of f_i that evaluate to True given ω.

Important inference queries in MLNs are computing the partition function, finding the marginal probability of an atom given evidence (an assignment to a subset of variables) and finding the most probable assignment to all atoms given evidence (MAP inference). Here, we focus on the marginal inference problem.

Gibbs Sampling

Gibbs sampling [6] is one of the most widely used MCMC algorithms to date. Gibbs sampling changes one variable at a time by sampling that variable from its conditional distribution given all other variables as described below.

Given a set of n variables $X_1 \ \ldots \ X_n$, the Gibbs sampling algorithm begins with a random assignment $\overline{\mathbf{x}}^{(0)}$ to all variables. Then, for $t = 1, \ldots, T$, it performs the following steps (each step is called a Gibbs iteration). Let $(X_1, \ldots, X_n)$ be an arbitrary ordering of variables in $\mathcal{M}$. Then, for $i = 1$ to n, it generates a new value $\overline{x}_i^{(t)}$ for X_i by sampling a value from the distribution $P(X_i|\overline{\mathbf{x}}_{-i}^{(t)})$ where $\overline{\mathbf{x}}_{-i}^{(t)} = (\overline{x}_1^t, \ldots, \overline{x}_{i-1}^t, \overline{x}_{i+1}^{(t-1)}, \ldots, \overline{x}_n^{(t-1)})$.

Gibbs sampling is typically used to estimate the marginal probabilities. Typically, the sampler is allowed to run for some time (called the *burn-in* time) to allow it to *mix* which ensures that it forgets its initialization, and after T samples from a mixed Gibbs sampler are generated, the 1-variable marginal probabilities can be estimated using the following equation.

$$\widehat{P}_T(\overline{x}_i) = \frac{1}{T}\sum_{t=1}^{T} P(\overline{x}_i|\overline{\mathbf{x}}_{-i}^{(t)}) \quad (2)$$

DP-Means

DP-Means [12] is a non-parametric clustering method that unifies K-means clustering with Bayesian non-parametric models. Specifically, Kullis and Jordan showed that modifying the K-means objective with a penalty term is asymptotically equivalent to performing Gibbs sampling in a Dirichlet-process Mixture Model to infer the right number of clusters. The modified objective is as follows:

$$\min_{\ell_c} \sum_{c=1}^{k} \sum_{x\in\ell_c} ||x-\mu_c||^2+\lambda k \quad \text{where } \mu_c = \frac{1}{|\ell_c|}\sum_{x\in\ell_c} x \quad (3)$$

To solve the modified K-means objective, DP-Means creates new clusters only when points are sufficiently far off from existing clusters. For completeness sake, we restate the key aspects of the algorithm in Algorithm 1.

Input: $x_1, \ldots x_n; \lambda$
Output: clustering: $\ell_1 \ldots \ell_k$
while *converged=false* **do**
  for *each input point* x_i **do**
    m = Compute minimum distance of x_i w.r.t all current cluster centers
    if $m > \lambda$ **then**
      Create new cluster and assign x_i to new cluster
    end
    else
      Assign x_i to its closest cluster
    end
  end
end

Algorithm 1: DP-Means

Kulis and Jordan showed that Algorithm 1 converges to a local optimal solution. Depending on the value of λ, we would converge to solutions that place more (or less) emphasis on reducing the overall number of clusters.

3 Non-Parametric Domain Approximation

It is now quite widely understood that in order to scale up inference in MLNs, one needs to perform *domain lifting* [23], i.e., take advantage of symmetries or exchangeability of variables in the MLN [17] to perform efficient inference over groups of objects in the MLN. Following a similar vein, the idea behind approximate domain lifting is to relax the notion of symmetries or exchangeable variables such that domain lifting is applicable to a much larger class of MLNs. One way to find such symmetries is to treat the problem of domain lifting as an unsupervised machine learning problem and use clustering algorithms to learn symmetries based on the structure of the MLN, the given inference query, and evidence. Specifically, for marginal inference, ideally, we would like to cluster together all ground atoms that have similar marginal probabilities. This would then allow us to treat all atoms in the cluster uniformly without having to explicitly compute the marginal probabilities separately for every atom in the cluster. However, it should be noted that clustering at the ground atom level is a non-trivial problem and one that is computationally expensive since the number of ground atoms may themselves be extremely large in MLNs that encode application domains such as Natural Language Understanding.

Formulas:
$\mathtt{R}(x) \vee \mathtt{S}(x,y), w$
Original Domains:
$\Delta_x = \{A_1, B_1, C_1, D_1\}$
$\Delta_y = \{A_2, B_2, C_2, D_2\}$
Domain Approximation:
$\Delta'_x = \{\mu_1, \mu_2\}$
$\Delta'_y = \{\mu_3, \mu_\}$
(a)

Meta-Objects:
$\mu_1 = \{A_1, B_1\}$; $\mu_2 = \{C_1, D_1\}$
$\mu_3 = \{A_2, B_2\}$; and $\mu_4 = \{C_2, D_2\}$
(b)

Meta-Atoms:
$\mathtt{R}_1(\mu_1) = \{\mathtt{R}(A_1), \mathtt{R}(B_1)\}$
$\mathtt{R}_2(\mu_2) = \{\mathtt{R}(C_1), \mathtt{R}(D_1)\}$
$\mathtt{S}_1(\mu_1, \mu_2) = \{\mathtt{S}(A_1, C_1), \mathtt{S}(A_1, D_1),$
$\mathtt{S}(B_1, C_1), \mathtt{S}(B_1, D_1)\}$
...
(c)

Figure 1: (a) an example MLN $\mathcal{M}$ and a possible domain approximation for the original domain of $\mathcal{M}$. $\mathcal{M}'$ contains meta-objects and meta-atoms, i.e., objects that represent multiple objects in the original domain and atoms that represent multiple atoms in $\mathcal{M}$ as shown in (b) and (c)

As an alternative to clustering at the level of ground atoms, Venugopal and Gogate [28], and Broeck and Darwiche [25] proposed clustering approaches at the object-level. That is, given a set of domains $\mathcal{D} = \{D_1, \ldots D_M\}$, where each D_j is a set of real-world objects that can be instantiated in

$\mathcal{M}$, and evidence $\mathbf{E}$, we cluster each domain in $\mathcal{D}$ independently and replace the set of objects with *meta-objects*, i.e., the set of cluster-centers, to generate a new domain $\mathcal{D}' = \{D'_1, \ldots D'_k\}$, where each $|D'_j| << |D_j|$. Replacing the domain in $\mathcal{M}$ with $\mathcal{D}'$ yields a new MLN $\mathcal{M}'$ which we refer to as the *domain-approximated* version of $\mathcal{M}$. In $\mathcal{M}'$, each ground atom is now a *meta-atom* since it implicitly represents a set of ground atoms in $\mathcal{M}$. An example MLN and its domain approximation is shown in Fig 1.

Clearly, choosing the right domain approximation is crucial to ensuring the quality of inference results. Therefore, the key question that we wish to answer here is: Given $\mathcal{M}$, $\mathcal{D}$ and $\mathbf{E}$, how do we choose D'_1, $D'_2 \ldots D'_k$ to obtain $\mathcal{M}'$ that is in some sense optimal?

3.1 Problem Formulation

We learn the approximate domains for an MLN using a non-parametric approach. Specifically, we use the DP-means algorithm to find the optimal clustering. Note that other notable alternatives for non-parametric clustering exist, such as Dirichlet Process Mixture Models, which uses the Bayesian non-parametric framework for learning clusters without fixing them apriori. However, it turns out that DP-means is a much simpler, more scalable approach and seamlessly integrates Bayesian non-parametrics with the classical and universally popular K-means clustering algorithm which makes it an ideal model for our problem.

Specifically, we formulate the non-parametric domain approximation problem for a given MLN as follows:

$$\min_{\{\ell_{cj}\}_{c=1;j=1}^{|D'_j|;M}} \sum_{j=1}^{M} \sum_{c=1}^{|D'_j|} \sum_{\mathbf{x} \in \ell_{cj}} ||\mathbf{x} - \mu_{cj}||^2 + \lambda |D'_j| \quad (4)$$

where $\mu_{cj} = \frac{1}{|\ell_{cj}|} \sum_{\mathbf{x} \in \ell_{cj}} \mathbf{x}$, λ is a parameter that controls the number of clusters created for each domain in $\mathcal{M}$. $||\mathbf{x} - \mu_{cj}||^2$ is the Euclidean distance between the cluster center μ_{cj} and $\mathbf{x}$.

Given a constant λ, it is easy to see that we can decompose Eq. (4) into M independent objective functions and optimize each objective independently. This will yield the approximate domains for the input MLN $\mathcal{M}$. However, the challenging task is to automatically tune the parameter λ such that $\mathcal{M}'$ is in some way a "good" approximation of $\mathcal{M}$. We next describe an approach to quantify the error made by $\mathcal{M}'$ in approximating $\mathcal{M}$ and incorporate this error to automatically tune λ in Eq. (4).

3.2 Domain Approximation Error

Let $\mathcal{M}$ be the original MLN and $\mathcal{M}'$ be the MLN obtained after approximating each domain in $\mathcal{M}$. Clearly, the distributions $P_{\mathcal{M}}$ and $P_{\mathcal{M}'}$ are defined over spaces with different cardinalities since they have a different number of possible ground atoms. It turns out computing a valid distance metric that can directly compare such distributions is extremely challenging and is shown to be NP-hard [29]. Thus, we need to design approximations that can reasonably compare $P_{\mathcal{M}}$ and $P_{\mathcal{M}'}$.

Consider a single meta-atom in $\mathcal{M}'$, X, which corresponds to a set of ground atoms in $\mathcal{M}$ which we denote by $\mathbf{X}$. Thus, to map $P'_{\mathcal{M}}$ to $P_{\mathcal{M}}$, we need to map a 0/1 assignment of X to a vector of 0/1 assignments to $\mathbf{X}$. Clearly, there are $2^{|\mathbf{X}|}$ different ways to define this mapping. For each mapping, we will end up with a different approximation to $P_{\mathcal{M}}$. If we fix a specific mapping ρ, clearly, we can convert any sample $\mathbf{x}$ drawn from $P_{\mathcal{M}'}$ to a set of samples $\rho(\mathbf{x})$ in $P_{\mathcal{M}}$. In this case, the (un-normalized) probability $P_{\mathcal{M}'}(\mathbf{x})$ can be computed by summing over the (un-normalized) probabilities of $\rho(\mathbf{x})$ as follows:

$$P_{\mathcal{M}'}(\mathbf{x}) = \sum_{\mathbf{y} \in \rho(\mathbf{x})} P_{\mathcal{M}}(\mathbf{y})$$

However, computing the above probability is clearly infeasible since it involves a summation over the probabilities in the original space which can be very large and is precisely the reason to perform domain-approximation in the first place. Instead, if we choose ρ to be a one-to-one mapping, we map each sample in $P_{\mathcal{M}'}$ to exactly one sample in $P_{\mathcal{M}}$. In other words we assume that all other samples that $\mathbf{x}$ can be mapped to have negligible probabilities. Using this assumption, the above equation now reduces to

$$P_{\mathcal{M}'}(\mathbf{x}) \approx P_{\mathcal{M}}(\rho(\mathbf{x}))$$

Even with the above approximation of one-to-one mapping, computing $P_{\mathcal{M}'}(\mathbf{x})$ may be hard since we additionally require that $P_{\mathcal{M}}(\rho(\mathbf{x})$ should be sufficiently easy to compute. That is, given $\mathbf{x}$, we should be able to compute $P_{\mathcal{M}}(\rho(\mathbf{x})$ in bounded time/space. Unfortunately, for an arbitrary ρ, this problem requires computing the counts of satisfied formulas in a sample as its sub-step and is thus $\#P$-complete [22]. However, consider a special ρ, namely, given a sample from $\mathcal{M}'$ with meta-atom X assigned to x, we assign the same value x to all atoms in $\mathcal{M}$ that X corresponds to. We refer to such a mapping as a *uniform assignment* mapping and under this mapping, it turns out that marginal probabilities in $P_{\mathcal{M}}$ and $P'_{\mathcal{M}}$ have a direct relationship. Specifically,

Theorem 1. *Given an MLN $\mathcal{M}$ and its domain-reduced approximation $\mathcal{M}'$, under the assumption of uniform assignment mapping, for any ground atom X in $\mathcal{M}'$, $P_{\mathcal{M}'}(X) = P_{\mathcal{M}}(X')$, where $X' \in \mathbf{X}$.*

Proof. Let ω' be a world in $\mathcal{M}'$ and ω be a world in $\mathcal{M}$ obtained by the mapping function ρ.

$$P(\omega') \propto exp(\sum_i N_i(\omega')\theta_i)$$

$$P(\omega) \propto exp(\sum_i N_i(\omega)\theta_i)$$

Since we assume that ρ is a uniform assignment mapping, we have that for any formula f_i, $N_i(\omega') \propto N_i(\omega)$. Therefore, $P(\omega') \propto P(\omega)$. By extension, the partition function $Z(\mathcal{M}') \propto Z(\mathcal{M})$. Since marginal probabilities are simply ratios of partition functions, the result of the theorem holds. □

The above theorem means that under the assumption of uniform assignment mapping, we can perform marginal inference as follows. We approximate the domains in $\mathcal{M}$ and without changing the weights or formulas in $\mathcal{M}$, we can simply replace the original domain by its domain-approximation to yield $\mathcal{M}'$. We then generate samples from $\mathcal{M}'$ and estimate the marginal probabilities from the generated samples for each meta-atom X. We can finally compute the marginal probabilities in $\mathcal{M}$ by using $P(X)$ for all atoms in $\mathcal{M}$ that meta-atom X represents.

3.3 Adaptive Gibbs Sampling

In the presence of evidence, Theorem 1 no longer holds since we need to first translate the evidence $\mathbf{E}$ observed for $\mathcal{M}$ to $\mathcal{M}'$. Depending on this translation, $P_{\mathcal{M}}(\cdot|\mathbf{E})$ may be very different from $P_{\mathcal{M}'}(\cdot|\mathbf{E}')$ even under the assumption of uniform assignment mapping. Previous approaches such as [28] have proposed the transformation of evidence based on majority voting. Specifically, if X is a meta-atom in $\mathcal{M}'$ that corresponds to $\mathbf{X}$ in $\mathcal{M}$, X is assigned as true evidence in $\mathcal{M}'$ if the number of true evidence atoms in $\mathbf{X}$ outweighs the number of false or unknown atoms. However, consider sampling from $P_{\mathcal{M}'}(\cdot|\mathbf{E}')$, where $\mathbf{E}'$ has been generated by applying the aforementioned majority voting. Unless every meta-atom in $\mathcal{M}'$ represents a set of atoms that are all either evidence or all non-evidence atoms, each sample derived from $P_{\mathcal{M}'}(\cdot|\mathbf{E}')$ when mapped to $\mathcal{M}$ using a uniform assignment mapping, will have inconsistencies. For example, let X, Y, Z, U be the ground atoms in $\mathcal{M}$ and let $C1 = X, Y, Z$ and $C2 = U$ be the meta-atoms of $\mathcal{M}'$. If $X = 1$ is given as evidence to $\mathcal{M}$, then no evidence is set in $\mathcal{M}'$. Therefore, in every sample generated from $\mathcal{M}'$ is inconsistent with evidence $X = 1$. Similarly, if $Y = 1$ is added as evidence, then $C1 = 1$, and no samples with $Z = 0$ are generated. Our main idea is to reduce the expected number of inconsistencies when we generate samples from $\mathcal{M}'$ via Gibbs sampling.

In each iteration of Gibbs sampling, we pick a meta-atom, say X in $\mathcal{M}'$ and sample an assignment to this meta-atom from the conditional probability distribution $P_{\mathcal{M}'}(X|X_{-i})$, where X_{-i} is the set of all meta-atoms other than X. The choice of which meta-atom to sample in each iteration is according to a distribution of selection probabilities. Typically, in random-scan Gibbs sampling, this selection probability is a uniform distribution over the non-evidence atoms. However, in general, we can select the selection probabilities to be non-uniform, i.e. a probability α_i for atom X_i. It has also been shown that as long as the selection probabilities are not continuously updated (also called vanishing adaptation), we can show that the Markov chain remains ergodic (cf. [9])

We now define the expected error in a sample drawn from the Gibbs sampler as follows. Let $X_1 \ldots X_K$ be the meta-atoms in $\mathcal{M}'$ and let $\mathbf{X}_1 \ldots \mathbf{X}_K$ be sets of atoms in $\mathcal{M}$ such that X_i represents atoms $\mathbf{X}_i$. If $\mathbf{X}_i$ consists of both evidence and non-evidence atoms and we sample X, then the sample generated may be erroneous on all evidence atoms in $\mathbf{X}_i$. But if we do not sample X, then we lose out on sampling all the non-evidence atoms in $\mathbf{X}$. The overall expected Gibbs sampling error is given by

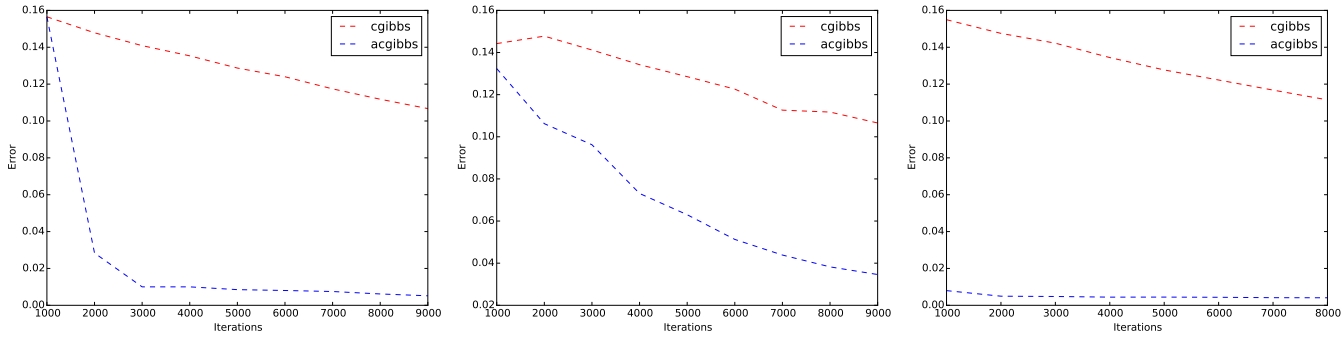
$$E_G = \sum_{i=1}^{K} \alpha_i \sum_{X' \in \mathbf{X}_i} \mathbb{I}(X') + (1-\alpha_i)(|X_i| - \sum_{X' \in \mathbf{X}_i} \mathbb{I}(X'))$$

where α_i is the selection probability for meta-atom X_i and $\mathbb{I}(X') = 1$ if X' is an evidence atom and 0 otherwise.

In order to minimize E_G, we would want to choose the selection probabilities $\alpha_i, \ldots \alpha_K$ that minimizes E_G. This can be shown to be equal to $\alpha_i = 1 - \frac{1}{\sum_{X' \in \mathbf{X}_i} \mathbb{I}(X')}$. That is, we want to sample X_i with a selection probability that is inversely proportional to the number of evidence atoms in $\mathbf{X}_i$. This means that if the number of evidence atoms in $\mathbf{X}$ is large, we sample X relatively few times and thus fewer samples will be inconsistent with the original evidence. If all the atoms in $\mathbf{X}$ are evidence, then X is never sampled at all. By substituting $\alpha_i = 1 - \frac{1}{\sum_{X' \in \mathbf{X}_i} \mathbb{I}(X')}$, we can simplify the sampling error as,

$$E_G = |\mathbf{E}| + \sum_{i=1}^{k} \left(\frac{|\mathbf{X}_i| - \sum_{X' \in \mathbf{X}_i} \mathbb{I}(X')}{\sum_{X' \in \mathbf{X}_i} \mathbb{I}(X')} \right) - 2K \quad (5)$$

Next, with the help of a simple example MLN, we illustrate that adapting the selection probabilities of meta-atoms based on the clustering is likely to yield more accurate estimates as opposed to randomly choosing a meta-atom to sample. Here, we considered a simple MLN with three formulas $w_1; \mathtt{R}(x) \vee \mathtt{S}(x)$, $w_2; \mathtt{R}(x)$ and $w_2; \mathtt{S}(x)$ where x has 10000 objects. We introduced 25% random evidence on the ground atoms of `R` and `S`. We then randomly divided the objects in x into K clusters to approximate its domain. Thus, there are $2K$ meta-atoms with each meta-atom representing a variable number of original atoms with varying evidence. We compared the performance of Gibbs sampling with random selection probabilities which we refer to as `cgibbs` and our approach where the selection probabilities are tied to the clustering, which we refer to as `acgibbs`. Specifically, we compared the average error between marginal probabilities output by `cgibbs` and `acgibbs` with the true marginal probabilities for the MLN



(a) $K = 1000$ (b) $K = 100$ (c) $K = 10$

Figure 2: Illustrating the effect of adapting the selection probabilities in Gibbs sampling based on the clustering. The results are shown for $w_1; \mathtt{R}(x) \vee \mathtt{S}(x)$, $w_2; \mathtt{R}(x)$ and $w_2; \mathtt{S}(x)$ with a domain of 10000 objects and randomly generated evidence. K signifies the number of clusters into which the domain objects were divided. `cgibbs` denotes Gibbs sampling without adapting the selection probability and `acgibbs` denotes Gibbs sampling with adapting the selection probability. The curves are shown as the average mean square error between the true marginal probabilities and the marginal probabilities computed by the samples.

(which we could easily compute due to our choice of the specific MLN structure). The results are shown in Fig. 2 for varying number of clusters (K). As seen in the figure, `acgibbs` generates much more accurate estimates of the marginal probability by reducing the number of inconsistent samples that were generated. The effect is even more pronounced when we have fewer but larger clusters, i.e., Fig. 2 (c), where since K is small, each meta-atom therefore represents a large number of atoms. Here, using random scan Gibbs is much worse since the number of inconsistencies in each sample is extremely large yielding to less accurate estimates of marginal probabilities.

Gibbs Sampling Efficiency : If the Gibbs sampler contains N variables, it roughly takes N steps to change the complete state of the sampler. Thus, suppose $\mathcal{M}$ has a million non-evidence atoms, we need to at least perform a million sampling operations to touch every variable in the model once which is prohibitively expensive. Further, the *mixing* time of the Gibbs sampler is roughly proportional to the number of variables in the model [13]. Thus, we would like our Gibbs sampler to have a bounded number of variables for efficiency and fast mixing. Given $\mathcal{M}'$ which is a domain-approximated version of $\mathcal{M}$, we define the sampling efficiency (S_G) of $\mathcal{M}'$ to be proportional the total number of meta-atoms in $\mathcal{M}'$.

3.4 Computing the Optimal Clusters

We now re-formulate the minimization problem in Eq. (4) by incorporating the sampling error (E_G) and the sampling efficiency (S_G). That is, given constants β_1 and β_2, our clustering problem is defined as,

$$\min_{\{\ell_{cj}\}_{c=1;j=1}^{|D'_j|;M};\lambda} \sum_{j=1}^{M} \sum_{c=1}^{|D'_j|} \sum_{\mathbf{x} \in \ell_{cj}} ||\mathbf{x} - \mu_{cj}||^2 + \lambda |D'_j| \qquad (6)$$

where $E_G(\lambda) < \beta_1$ and $S_G(\lambda) < \beta_2$

Note that both the sampling error and efficiency depend upon the parameter λ. That is, if λ is large, then we penalize the number of clusters a lot more, therefore, our solution will yield very few meta-atoms which results in a large $E_G(\lambda)$ but small $S_G(\lambda)$. Similarly, smaller λ will yield solutions that has small $E_G(\lambda)$ but large $S_G(\lambda)$. Jointly optimizing $\{\ell_{cj}\}_{c=1;j=1}^{|D'_j|;M}; \lambda$ is challenging because we need to consider every possible clustering with every possible parameter λ. Instead, we use a co-ordinate descent type of approach where we pick a λ and find the best clusters, and then fix the clustering and pick the next best λ.

The algorithm for non-parametric clustering of domains is shown in Algorithm 2. Algorithm 2 starts by fixing λ to a large constant and computes the optimal clustering for the domains of the input MLN, at this value of λ. The DP-Means algorithm is used a sub-step to compute optimal clustering for a given λ. Once, we compute the optimal clustering of the domains, we evaluate the sampling error and efficiency based on the new MLN generated from the clusters. If we satisfy the constraints on sampling error bound and efficiency, then the algorithm has reached a local optima and we output the domain-approximated MLN. However, if we fail to satisfy the constraints, we check if $S_G(\lambda)$ is greater than β_2 in which case, we cannot find a solution, else we reduce λ by ϵ to improve $E_G(\lambda)$ and $S_G(\lambda)$.

Input: $\mathcal{M}$, β_1, β_2, ϵ
Output: $\mathcal{M}'$
$\lambda = 10000$
while *converged=false* **do**
  `// Find the clustering for a fixed` λ
  for *each domain* D_i *in* $\mathcal{M}$ **do**
    $D_i' =$ DP-Means(D_i)
  end
  $\mathcal{M}' =$ Replace-Domains($\mathcal{M}$, $D_1, \ldots$)
  $E_G(\lambda) =$ Evaluate Eq (5) for $\mathcal{M}'$
  $S_G(\lambda) =$ Number of meta-atoms in $\mathcal{M}'$
  if $E_G(\lambda) < \beta_1$ *and* $S_G(\lambda) < \beta_2$ **then**
    converged=true
    return $\mathcal{M}'$
  end
  else if $S_G(\lambda) > \beta_2$ **then**
    `// Could not find solution`
    return $\mathcal{M}'$
  end
  else
    `// Reduce` λ
    $\lambda = \epsilon\lambda$
  end
end

Algorithm 2: NP-Cluster

4 Related Work

In recent years many methods have been proposed that uses symmetries for improving the scalability both in exact inference [18, 4, 7, 24, 3] as well as approximate inference [20, 10, 8, 16, 27, 2]. Niepert and Broeck [17] recently showed that most of the earlier work on lifted inference can be connected to the concept of exploiting *finite partial exchangeability* in statistics that allows one to perform inference over groups of exchangeable variables efficiently. However, lifted inference that only looks for exact exchangeability tends to work with limited classes of MLNs as shown in [25]. To address this issue, Broeck and Darwiche [25] proposed *over symmetric approximations*, i.e., adding symmetries to the MLN to make inference more efficient. Venugopal and Gogate [28] proposed the use of unsupervised machine learning methods for approximately inducing these over-symmetries and hence approximately lifting the MLN. Recently, Belief propagation and MAP inference algorithms that exploit approximate symmetries were proposed in [21, 19]. The most closely related work to ours is the work by Broeck and Niepert [26] who developed a Metropolis-Hastings sampler by utilizing over-symmetric approximations of MLNs as a proposal distribution. However, unlike our approach where the clustering and sampling is more tightly integrated, Broeck and Niepert did not focus on learning the ideal clusters for their type of sampling.

Dataset	#Clauses	#Atoms	#Parameters
WebKB	892 million	20 million	64
Protein	408 million	3.3 million	211
ER	1.7 trillion	5.5 million	15

Table 1: Dataset sizes.

5 Experiments

We evaluate our approach, which we refer to as `acgibbs`, using three benchmark MLNs obtained from the Alchemy [11] website: Webkb MLN that models the relations between web-page links and topics in the webpage, Protein MLN that models the interaction between proteins, and the ER MLN that is used for entity resolution in NLP. The details of these benchmarks are shown in Table 1. We evaluate our approach along two dimensions: accuracy in estimating the marginal probabilities and convergence of the Markov Chain. We compare our results with regular Gibbs sampling (`Gibbs`) and the approach proposed in Venugopal and Gogate [28] (`cgibbs`), where they use clustering algorithms such as KMeans to derive an approximate MLN and then sample this MLN using regular Gibbs sampling.

For `cgibbs`, since we explicitly need to set the number of clusters for each domain, we set this to be 10% of the original domain-size. For `acgibbs`, for an approximate MLN, $\mathcal{M}'$, we set the threshold β_1 as 0.01% of the number of meta-atoms in $\mathcal{M}'$ and β_2 as 10K. For fairness, in both `cgibbs` and `acgibbs`, we used the same features as specified in [28]for computing the distances.

5.1 Accuracy

We compare the accuracy of `cgibbs` and `acgibbs` using the following approach. We assume that the `gibbs` algorithm outputs the true marginal probabilities and compare the results of `cgibbs` and `acgibbs` with that of `gibbs`. We measure the average Hellinger distance between the marginal probabilities of the query atoms output by `gibbs` with the probabilities output by `acgibbs` and `cgibbs`. We considered all ground atoms not set as evidence to be the query atoms. We measured accuracy on small MLNs since for larger MLNs the output of `gibbs` is not reliable. Specifically, we subsampled the true domain of the benchmarks and derived smaller MLNs out of the original benchmarks. Further, we also evaluated the performance of the algorithms in the presence of low as well as high evidence. Fig. 3 shows the results that we obtained for our benchmarks. As seen here, for the protein benchmark, `acgibbs` performs much better than `cgibbs`. For the Webkb benchmark, the accuracy of `acgibbs` is again better than `cgibbs`. For the ER benchmark, for the low evidence case, `acgibbs` was slightly worse than `cgibbs` but for the high evidence case, `acgibbs` was much bet-

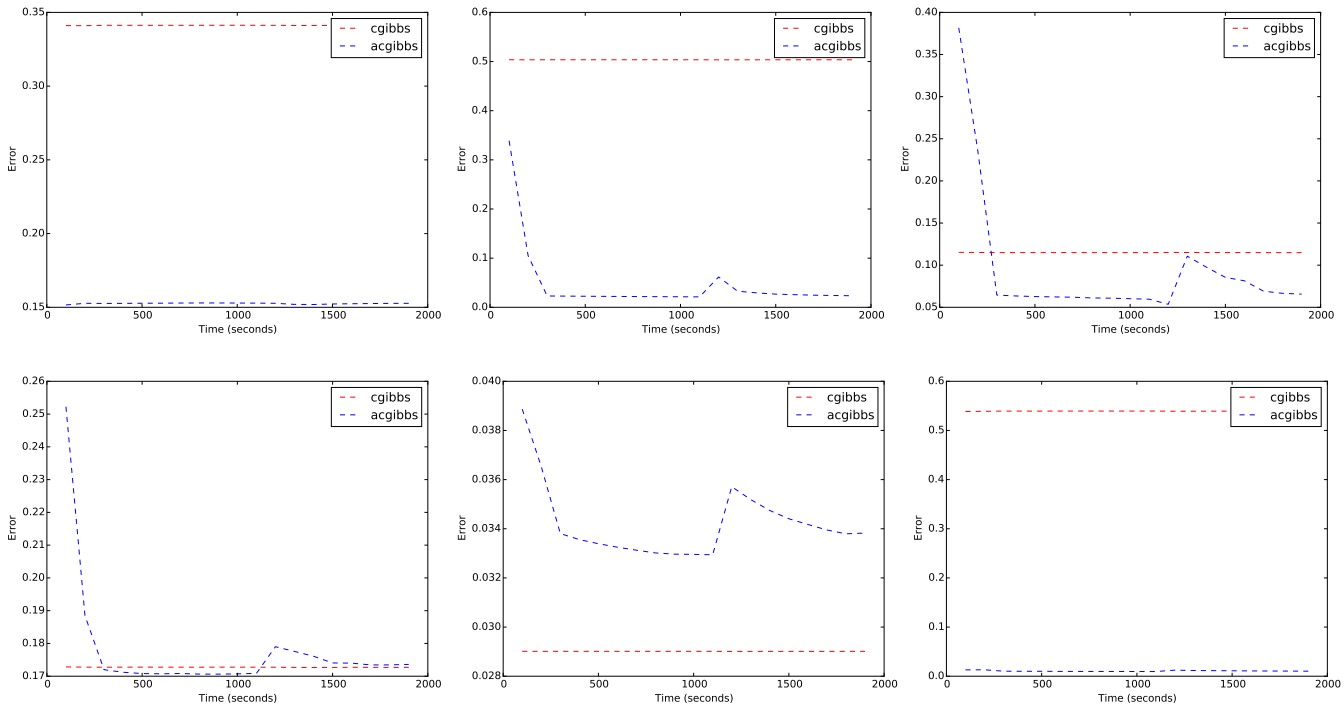


Figure 3: Accuracy Plots for various benchmarks. The domain size of each benchmark was reduced to 10% of its original size to ensure that we get better estimates with `gibbs`. The plots are shown for the average Hellinger distance between the marginal probabilities generated by `gibbs` and that generated by `cgibbs` and `acgibbs` respectively. (a) and (b) correspond to the Protein benchmark with 25% and 50% evidence respectively. Similarly (c) and (d) correspond to the Webkb benchmark, and (e), (f) correspond to the ER benchmark.

ter than `cgibbs`. One hypothesis for this behavior is that perhaps `gibbs` is not very accurate on this benchmark as shown by the poor convergence property that it exhibits for this benchmark (see next section). Overall, `acgibbs` was seen to be much more accurate than `cgibbs` in our evaluation.

5.2 Convergence

We compare the mixing time of `gibbs`, `cgibbs` and `acgibbs` based on the Gelman-Rubin (G-R) Statistic [5]. For a well-mixed sampler, the G-R statistic should ideally decrease over time illustrating that the MCMC chain has mixed. To compute the G-R statistics, we set up 5 Gibbs samplers from random initialization points and measure the within chain and across chain variances for the marginal probabilities for 1000 randomly chosen query ground atoms. We compute the G-R statistics for each of the 1000 query atoms and measure the mean G-R statistic. Fig. 4 shows our results on the benchmarks. Note that, here we consider larger sized MLNs, i.e., we evaluate on the full benchmark without subsampling the MLN domain. As seen here, for the protein benchmark `cgibbs` and `acgibbs` mix much faster than `gibbs` which has an upward trajectory for the G-R statistic indicating that it has not mixed at all. Among `cgibbs` and `acgibbs`, `acgibbs` mixes faster than `cgibbs` because the selection probability ensures that we spend more resources on sampling meta-atoms which are less deterministic (there is less evidence on the atoms that the meta-atom represents) as compared to meta-atoms which are more deterministic (more or less all the atoms represented by the meta-atom are evidence atoms). For the Webkb benchmark, the results look similar with `gibbs` not mixing and `acgibbs` mixing faster than `gibbs`. For the ER case, the curve for `gibbs` stays flat at almost 0. Due to the large size of the benchmark, `gibbs` is very slow in its iterations and most of the query atoms remain un-sampled and thus only retain their initialization values. Even `cgibbs` and `acgibbs` though clearly better than `gibbs`, have slower mixing times as compared to the other two benchmarks.

6 Conclusion

Exploiting approximate symmetries has been recognized as a practical approach to obtain scalable inference algorithms in MLNs. Several inference methods that take advantage of approximate symmetries have been proposed over the last couple of years. However, a major disadvantage of existing

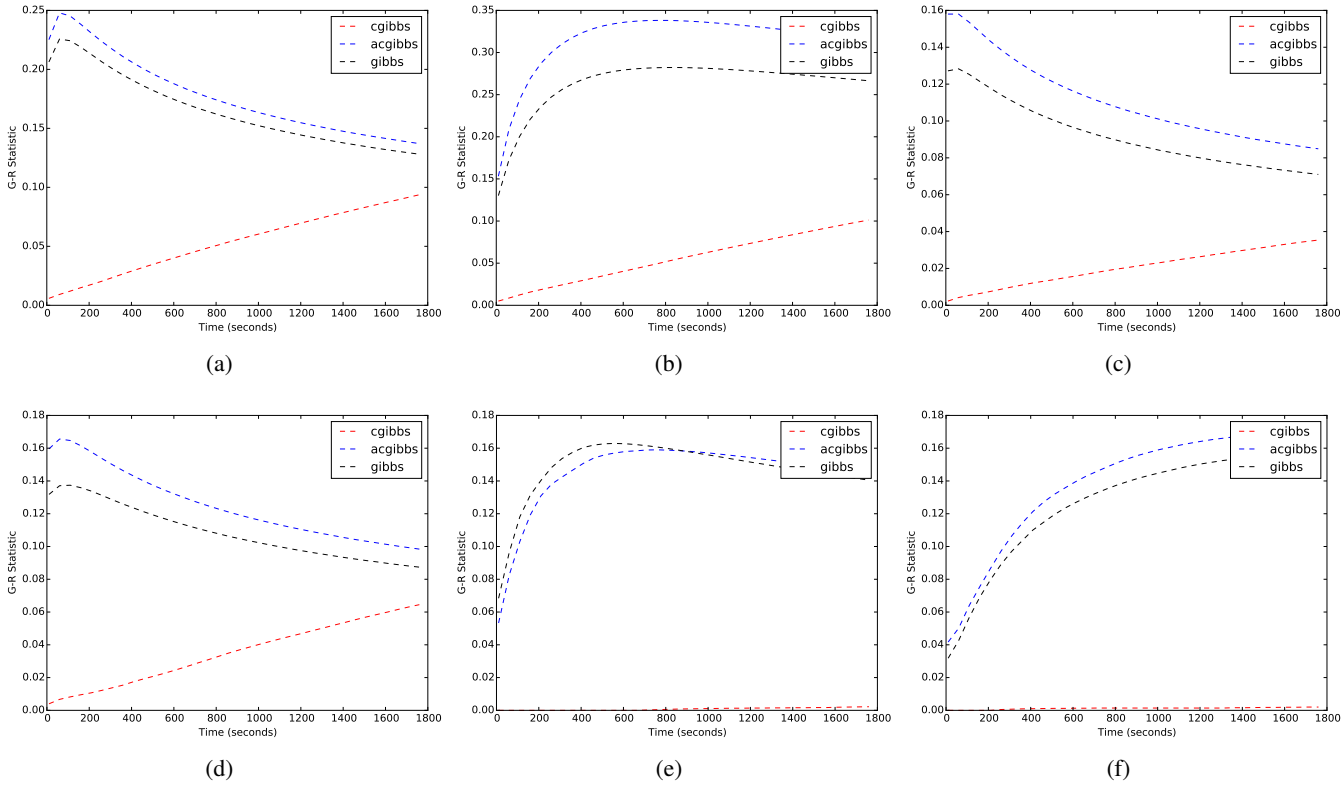


Figure 4: Convergence Plots for various benchmarks. The plots are shown for the average Gelman-Rubin statistic computed using 5 runs of the algorithms across 1000 randomly chosen estimates for the marginal probabilities. (a) and (b) correspond to the Protein benchmark with 25% and 50% evidence respectively. Similarly (c) and (d) correspond to the Webkb benchmark, and (e), (f) correspond to the ER benchmark.

methods is that it is quite difficult to manually tune the parameters in these approaches to obtain accurate inference results. Here, we proposed a non-parametric approach that approximates the domains of an MLN and can systematically trade-off accuracy with efficiency. Further, we integrated our approach with a Gibbs sampling algorithm that adapts itself based on the domain-approximation to generate higher quality samples. Our results on benchmarks showed the promise of our approach in terms of scalability, accuracy and convergence.

In future, we would like to explore more complex mapping function from meta-atoms to the original atoms and develop more sophisticated algorithms and guarantees for such functions. We will also explore the possibility of applying our approach for MAP inference as well.

References

[1] Babak Ahmadi, Kristian Kersting, Martin Mladenov, and Sriraam Natarajan. Exploiting symmetries for scaling loopy belief propagation and relational training. *Machine Learning*, 92(1):91–132, 2013.

[2] Hung Bui, Tuyen Huynh, and Sebastian Riedel. Automorphism Groups of Graphical Models and Lifted Variational Inference. In *Proceedings of the 29th Conference on Uncertainty in Artificial Intelligence*, pages 132–141. AUAI Press, 2013.

[3] Hung B Bui, Tuyen N Huynh, and Rodrigo de Salvo Braz. Exact lifted inference with distinct soft evidence on every object. 2012.

[4] R. de Salvo Braz. *Lifted First-Order Probabilistic Inference*. PhD thesis, University of Illinois, Urbana-Champaign, IL, 2007.

[5] A. Gelman and D. B. Rubin. Inference from Iterative Simulation Using Multiple Sequences. *Statistical Science*, 7(4): 457–472, 1992.

[6] S. Geman and D. Geman. Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6:721–741, 1984.

[7] V. Gogate and P. Domingos. Probabilistic Theorem Proving. In *Proceedings of the 27th Conference on Uncertainty in Artificial Intelligence*, pages 256–265. AUAI Press, 2011.

[8] V. Gogate, A. Jha, and D. Venugopal. Advances in Lifted Importance Sampling. In *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence*, 2012.

[9] J. Gonzalez, Y. Low, A. Gretton, and C. Guestrin. Parallel gibbs sampling: From colored fields to thin junction trees. In *In Artificial Intelligence and Statistics*, May 2011.

[10] K. Kersting, B. Ahmadi, and S. Natarajan. Counting Belief Propagation. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, pages 277–284, 2009.

[11] S. Kok, M. Sumner, M. Richardson, P. Singla, H. Poon, and P. Domingos. The Alchemy System for Statistical Relational AI. Technical report, Department of Computer Science and Engineering, University of Washington, Seattle, WA, 2006. http://alchemy.cs.washington.edu.

[12] Brian Kulis and Michael I Jordan. Revisiting k-means: New algorithms via bayesian nonparametrics. *ICML*, 2012.

[13] J. S. Liu. *Monte Carlo Strategies in Scientific Computing*. Springer Publishing Company, Incorporated, 2001.

[14] B. Milch and S. J. Russell. General-Purpose MCMC Inference over Relational Structures. In *Proceedings of the Twenty-Second Conference on Uncertainty in Artificial Intelligence*, pages 349–358, 2006.

[15] B. Milch, L. S. Zettlemoyer, K. Kersting, M. Haimes, and L. P. Kaelbling. Lifted Probabilistic Inference with Counting Formulas. In *Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence*, pages 1062–1068, 2008.

[16] Mathias Niepert. Markov chains on orbits of permutation groups. In *UAI*, pages 624–633. AUAI Press, 2012.

[17] Mathias Niepert and Guy Van den Broeck. Tractability through exchangeability: A new perspective on efficient probabilistic inference. In *Proceedings of the 28th AAAI Conference on Artificial Intelligence, AAAI Conference on Artificial Intelligence*, 2014.

[18] D. Poole. First-Order Probabilistic Inference. In *Proceedings of the 18th International Joint Conference on Artificial Intelligence*, pages 985–991, Acapulco, Mexico, 2003. Morgan Kaufmann.

[19] Somdeb Sarkhel, Parag Singla, and Vibhav Gogate. Fast lifted map inference via partitioning. In *Advances in Neural Information Processing Systems*, pages 3222–3230, 2015.

[20] P. Singla and P. Domingos. Lifted First-Order Belief Propagation. In *Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence*, pages 1094–1099, Chicago, IL, 2008. AAAI Press.

[21] Parag Singla, Aniruddh Nath, and Pedro Domingos. Approximate Lifting Techniques for Belief Propagation. pages 2497–2504, 2014.

[22] Dan Suciu, Dan Olteanu, Christopher Ré, and Christoph Koch. *Probabilistic Databases*. Morgan & Claypool, 2011.

[23] G. Van den Broeck. On the Completeness of First-Order Knowledge Compilation for Lifted Probabilistic Inference. In John Shawe-Taylor, Richard S. Zemel, Peter L. Bartlett, Fernando C. N. Pereira, and Kilian Q. Weinberger, editors, *Proceedings of the 25th Annual Conference on Neural Information Processing Systems (NIPS)*, pages 1386–1394, 2011.

[24] G. Van den Broeck, N. Taghipour, W. Meert, J. Davis, and L. De Raedt. Lifted Probabilistic Inference by First-Order Knowledge Compilation. In *Proceedings of the 22nd International Joint Conference on Artificial Intelligence*, pages 2178–2185, 2011.

[25] Guy van den Broeck and Adnan Darwiche. On the complexity and approximation of binary evidence in lifted inference. In *Advances in Neural Information Processing Systems 26*, pages 2868–2876, 2013.

[26] Guy Van den Broeck and Mathias Niepert. Lifted probabilistic inference for asymmetric graphical models. In *Proceedings of the 29th Conference on Artificial Intelligence (AAAI)*, 2015.

[27] D. Venugopal and V. Gogate. On lifting the gibbs sampling algorithm. In *Proceedings of the 26th Annual Conference on Neural Information Processing Systems (NIPS)*, pages 1664–1672, 2012.

[28] Deepak Venugopal and Vibhav Gogate. Evidence-based clustering for scalable inference in markov logic. In *ECML PKDD*, 2014.

[29] Mathukumalli Vidyasagar. A metric between probability distributions on finite sets of different cardinalities and applications to order reduction. *IEEE Trans. Automat. Contr.*, 57(10):2464–2477, 2012.