Advances in Lifted Importance Sampling

Vibhav Gogate
Department of Computer Science
The University of Texas at Dallas
Richardson, TX 75080, USA
vgogate@hlt.utdallas.edu

Abhay Jha
Computer Science and Engineering
University of Washington
Seattle, WA 98195, USA
abhaykj@cs.washington.edu

Deepak Venugopal
Department of Computer Science
The University of Texas at Dallas
Richardson, TX 75080, USA
dxv021000@utdallas.edu

Abstract

We consider lifted importance sampling (LIS), a previously proposed approximate inference algorithm for statistical relational learning (SRL) models. LIS achieves substantial variance reduction over conventional importance sampling by using various lifting rules that take advantage of the symmetry in the relational representation. However, it suffers from two drawbacks. First, it does not take advantage of some important symmetries in the relational representation and may exhibit needlessly high variance on models having these symmetries. Second, it uses an uninformative proposal distribution which adversely affects its accuracy. We propose two improvements to LIS that address these limitations. First, we identify a new symmetry in SRL models and define a lifting rule for taking advantage of this symmetry. The lifting rule reduces the variance of LIS. Second, we propose a new, structured approach for constructing and dynamically updating the proposal distribution via adaptive sampling. We demonstrate experimentally that our new, improved LIS algorithm is substantially more accurate than the LIS algorithm.

Introduction

The emerging field of statistical relational learning (SRL) (Getoor and Taskar 2007) seeks to combine logical and probabilistic representation and reasoning techniques. This combination is essential because many large, complex application domains have both rich relational structure and large amount of uncertainty. Logical representation and reasoning techniques such as first-order logic and theorem proving are good at handling complex, relational structure but have no representation for uncertainty. On the other hand, probabilistic models such as Bayesian networks and Markov networks, and reasoning techniques for them are adept at dealing with uncertainty but cannot handle relational structure.

Over the last decade, several modeling languages that combine logic and probability have been introduced. Notable examples are probabilistic relational models (Friedman et al. 1999), ProbLog (De Raedt, Kimmig, and Toivonen 2007), Bayesian logic (Milch et al. 2005) and Markov logic (Richardson and Domingos 2006). Obviously, for the wide applicability of these languages, we need access to fast, scalable, and accurate inference engines for them. To this end, several exact and approximate inference algorithms have been proposed over the last few years, starting with the pioneering work of (Poole 2003). These algorithms are often called lifted or first-order inference algorithms because they are capable of exploiting both relational (first-order) and probabilistic structure. Notable exact algorithms are first-order variable elimination (Poole 2003) and its extensions (de Salvo Braz 2007; Ng, Lloyd, and Uther 2008), first-order knowledge compilation (Van den Broeck et al. 2011) and probabilistic theorem proving (Gogate and Domingos 2011). Notable approximate algorithms are lifted belief propagation (Singla and Domingos 2008) and lifted importance sampling (Gogate and Domingos 2011).

In this paper, we consider the lifted importance sampling (LIS) algorithm of (Gogate and Domingos 2011). Just as propositional importance sampling (IS) can be understood as a partial exploration of the full propositional search space, LIS can be understood as a partial exploration of the lifted search space. The lifted search space groups similar propositional atoms together and as a result a sample in the lifted space corresponds to multiple samples in the propositional space. Because of an increase in the effective sample size, LIS has smaller variance and therefore higher accuracy than propositional IS. However, LIS, in its current form has two limitations. First, it does not use lifting (relational structure) to the fullest extent and as a result it can be needlessly inefficient and inaccurate on some problems. Second, it uses an uninformative proposal distribution. This is problematic because the accuracy of importance sampling is highly dependent on the quality of the proposal distribution.

To remedy these issues, we improve upon the LIS algorithm in the following ways. First, we propose a new lifting rule that reduces the lifted search space exponentially in many instances. We show how to perform importance sampling in this reduced space and prove that our new sampling algorithm has smaller variance. Second, we propose an adaptive, structured approach for constructing and dynamically updating the proposal distribution. Given an SRL model and evidence, the main idea here is to apply various lifting and probability propagation rules in an approximate manner by relaxing their pre-conditions, yielding a polynomially specifiable proposal distribution. Then, we initialize it to the prior distribution and dynamically update its parameters via adaptive importance sampling (Cheng and

Druzdzel 2000; Ortiz and Kaelbling 2000). We present experiments, comparing the quality of estimation of our advanced LIS scheme with the LIS scheme of (Gogate and Domingos 2011) on SRL models from various domains. Our experiments clearly demonstrate that our advanced algorithm is always superior.

Preliminaries

The language of propositional logic consists of atomic sentences called propositions or atoms, and logical connectives such as $\wedge$ (conjunction), $\vee$ (disjunction), $\neg$ (negation), $\Rightarrow$ (implication) and $\Leftrightarrow$ (equivalence). Each proposition takes values from the binary domain $\{\texttt{True}, \texttt{False}\}$ (or $\{0,1\}$). A propositional formula (sentence) f is an atom, or any complex sentence that can be constructed from atoms using logical connectives. For example, $\texttt{A}$, $\texttt{B}$ and $\texttt{C}$ are propositional atoms and $f = \texttt{A} \vee \neg\texttt{B} \wedge \texttt{C}$ is a propositional formula. A *knowledge base* (KB) is a set of formulas. A *world* is a truth assignment to all atoms in the KB. A world is a *model* of KB, if it evaluates every formula in the KB to $\texttt{True}$.

First-order logic (FOL) generalizes propositional logic by allowing atoms to have internal structure; an atom in FOL is a predicate that represents relations between objects. A predicate consists of a predicate symbol, denoted by Monospace fonts, e.g., `Friends`, `Smokes`, etc., followed by a parenthesized list of arguments called *terms*. A term is a logical variable, denoted by lower case letters such as x, y, z, etc., or a constant, denoted by upper case letters such as X, Y, Z, etc. We assume that each logical variable, e.g., x is typed and takes values over a finite set Δ_x. The language of FOL also includes two quantifiers in addition to the logical connectives: $\forall$ (universal) and $\exists$ (existential). Quantifiers express properties of an entire collection of objects. A formula in first order logic is a predicate (atom), or any complex sentence that can be constructed from atoms using logical connectives and quantifiers. For example, the formula $\forall x\ \texttt{Smokes}(x) \Rightarrow \texttt{Asthma}(x)$ states that all persons who smoke have asthma. $\exists x\ \texttt{Cancer}(x)$ states that there exists a person x who has cancer. A *first-order KB* is a set of first-order formulas.

In this paper, we assume that the formulas are of the form $\forall \mathbf{x}\ f$, where $\mathbf{x}$ is the set of variables in f and f is a conjunction or disjunction of literals; each literal is an atom or its negation. In other words, we assume that every logical variable is universally quantified and there are no existential quantifiers. Therefore, for brevity, we will drop $\forall$ from all formulas. Given variables $\mathbf{x}$ and constants $\mathbf{X}$ from their domain, $f[\mathbf{X}/\mathbf{x}]$ is obtained by substituting every occurrence of variable $x_i \in \mathbf{x}$ in f with $X_i \in \mathbf{X}$. A ground formula is a formula obtained by substituting each of its variable with a constant. A ground KB is a KB containing all possible groundings of all of its formulas. For example, the grounding of a KB containing one formula, $\texttt{Smokes}(x) \Rightarrow \texttt{Asthma}(x)$ where $\Delta_x = \{Ana, Bob\}$, is a KB containing two formulas: $\texttt{Smokes}(Ana) \Rightarrow \texttt{Asthma}(Ana)$ and $\texttt{Smokes}(Bob) \Rightarrow \texttt{Asthma}(Bob)$. A world in FOL is a truth assignment to all atoms in its grounding.

Markov logic (Domingos and Lowd 2009) extends FOL by softening the hard constraints expressed by the formulas. A soft formula or a weighted formula is a pair (f, w) where where f is a formula in FOL and w is a real-number. A Markov logic network (MLN), denoted by $\mathcal{M}$, is a set of weighted formulas (f_i, w_i). It represents the following probability distribution.

$$P_{\mathcal{M}}(\omega) = \frac{1}{Z(\mathcal{M})} \exp\left(\sum_i w_i N(f_i, \omega)\right) \tag{1}$$

where ω is a world, $N(f_i, \omega)$ is the number of groundings of f_i that evaluate to $\texttt{True}$ in the world ω and $Z(\mathcal{M})$ is a normalization constant or the partition function, given by

$$Z(\mathcal{M}) = \sum_{\omega} \exp\left(\sum_i w_i N(f_i, \omega)\right) \tag{2}$$

The main inference task in MLNs is computing the partition function and in this paper we focus on this task. At inference time, we often convert the MLN (or an SRL model) into canonical form, which serves as a basic object for various inference operators. Each inference operator takes one or more SRL models in canonical form as input, manipulates them and outputs an SRL model in canonical form. Notable canonical forms are parfactors (Poole 2003), first-order weighted CNFs with substitution constraints (Gogate and Domingos 2011) and normal forms (Milch et al. 2008; Jha et al. 2010). Although, our method can be easily adapted to work with any canonical form, in this paper, we use the normal form for simplicity of exposition. An important advantage of normal forms is that they do not need access to a constraint solver at inference time. However, in some cases, they can be quite inefficient (Kisynski and Poole 2009).

A *normal* MLN (Jha et al. 2010) is an MLN that satisfies the following properties: (1) There are no constants in any formula and (2) If two distinct atoms with the same predicate symbol have variables x and y in the same position then $\Delta_x = \Delta_y$. For example, consider an MLN having two formulas $(\texttt{Smokes}(x) \Rightarrow \texttt{Asthma}(x), w)$ and $(\texttt{Smokes}(Ana), \infty)$. It is not in normal form. Its normal form has three formulas: $(\texttt{Smokes}(x') \Rightarrow \texttt{Asthma}(x'), w)$, $(\texttt{Smokes1}(y) \Rightarrow \texttt{Asthma1}(y), w)$ and $(\texttt{Smokes1}(y), \infty)$, where $\Delta_x' = \Delta_x \setminus \{Ana\}$ and $\Delta_y = \{Ana\}$.

Lifted Importance Sampling

The main idea in importance sampling (IS) (c.f. (Liu 2001)) is to reformulate the summation problem in Eq. (2) as an expectation problem using a probability distribution Q, called the proposal or the importance distribution. Q should be such that it is easy to generate independent samples from it. Also, in order to apply IS to MLNs, Q should satisfy the constraint: $\exp(\sum_i w_i N(f_i, \omega)) > 0 \Rightarrow Q(\omega) > 0$. Formally, using Q, we can rewrite Eq. (2) as

$$Z(\mathcal{M}) = \sum_{\omega} \exp\left(\sum_i w_i N(f_i, \omega)\right) \frac{Q(\omega)}{Q(\omega)} \tag{3}$$

$$= \mathbb{E}_Q\left[\frac{\exp\left(\sum_i w_i N(f_i, \omega)\right)}{Q(\omega)}\right] \tag{4}$$

where $\mathbb{E}_Q[x]$ denotes the expected value of the random variable x w.r.t. Q. Given N worlds $(\omega^{(1)}, \ldots, \omega^{(N)})$, sampled

Algorithm 1: Lifted Importance Sampling (LIS)

```
Input: A normal MLN $\mathcal{M}$ and a proposal distribution $Q$
Output: An unbiased estimate of the partition function of $\mathcal{M}$
if $\mathcal{M}$ is empty then return $1$
if there exists a decomposer $\mathbf{x}$ then
    Let $x \in \mathbf{x}$ and $X \in \Delta_x$. return $[\text{LIS}(\mathcal{M}[X/\mathbf{x}], Q)]^{|\Delta_x|}$
if there exists a singleton atom $\mathtt{R}(x)$ that does not appear
more than once in the same formula then
    Use $Q$ to sample an integer $i$ from the range $[0, |\Delta_x|]$
    return $\frac{\text{LIS}(\mathcal{M}|\bar{\mathtt{R}}^i, Q)w(i)2^{p(i)}}{Q(i)}\binom{|\Delta_x|}{i}$
Choose an atom $\mathtt{A}$ and sample all of its groundings from $Q$.
Let $\bar{\mathtt{A}}$ be the sampled assignment.
return $\frac{\text{LIS}(\mathcal{M}|\bar{\mathtt{A}}, Q)w(\bar{\mathtt{A}})2^{p(\bar{\mathtt{A}})}}{Q(\bar{\mathtt{A}})}$
```

independently from Q, we can estimate $Z(\mathcal{M})$ using:

$$\widehat{Z}(\mathcal{M}) = \frac{1}{N}\sum_{j=1}^{N}\frac{\exp\left(\sum_i w_i N(f_i, \omega^{(j)})\right)}{Q(\omega^{(j)})} \qquad (5)$$

It is known that $\mathbb{E}[\widehat{Z}(\mathcal{M})] = Z(\mathcal{M})$ (i.e., it is unbiased) and therefore the mean squared error between $\widehat{Z}(\mathcal{M})$ and $Z(\mathcal{M})$ can be reduced by reducing its variance. The variance can be reduced by using a proposal distribution Q that is as close as possible to the distribution $P_{\mathcal{M}}$. Thus, a majority of research on importance sampling is focused on finding a good Q. For more details, see (Liu 2001).

The lifted importance sampling (LIS) algorithm (Gogate and Domingos 2011) reduces the variance of IS by grouping symmetric random variables, sampling just one member from each group and using the sampled member to estimate quantities defined over the group. It uses two lifting rules to identify symmetric variables; we will refer to them as *power rule* and *generalized binomial rule*.

The power rule is based on the concept of a decomposer. Given a normal MLN $\mathcal{M}$, a set of logical variables, denoted by $\mathbf{x}$, is called a *decomposer* if it satisfies the following two conditions: (i) Every atom in $\mathcal{M}$ contains exactly one variable from $\mathbf{x}$, and (ii) For any predicate symbol $\mathtt{R}$, there exists a position s.t. variables from $\mathbf{x}$ only appear at that position in atoms of $\mathtt{R}$. Given a decomposer $\mathbf{x}$, it is easy to show that $Z(\mathcal{M}) = [Z(\mathcal{M}[X/\mathbf{x}])]^{|\Delta_x|}$ where $x \in \mathbf{x}$ and $\mathcal{M}[X/\mathbf{x}]$ is the MLN obtained by substituting all logical variables $\mathbf{x}$ in $\mathcal{M}$ by the same constant $X \in \Delta_x$ and then converting the resulting MLN to a normal MLN. Note that for any two variables x, y in $\mathbf{x}$, $\Delta_x = \Delta_y$ by normality.

The generalized binomial rule is used to sample singleton atoms efficiently. The rule requires that the singleton atom does not appear more than once in the same formula (*self-joins*). Given a normal MLN $\mathcal{M}$ having a singleton atom $\mathtt{R}(x)$ that is not involved in self-joins, we can show that $Z(\mathcal{M}) = \sum_{i=0}^{|\Delta_x|}\binom{|\Delta_x|}{i}Z(\mathcal{M}|\bar{\mathtt{R}}^i)w(i)2^{p(i)}$ where $\bar{\mathtt{R}}^i$ is a truth-assignment to all groundings of $\mathtt{R}$ such that exactly i groundings of $\mathtt{R}$ are set to $\mathtt{True}$ (and the remaining are set to $\mathtt{False}$). $\mathcal{M}|\bar{\mathtt{R}}^i$ is the MLN obtained from $\mathcal{M}$ by performing the following steps in order: (i) Ground all $\mathtt{R}(x)$ and set its groundings to have the same assignment as $\bar{\mathtt{R}}^i$, (ii) Delete all formulas that evaluate to either $\mathtt{True}$ or $\mathtt{False}$, (iii) Delete all groundings of $\mathtt{R}(x)$ and (iv) Convert the resulting MLN to a normal MLN. $w(i)$ is the exponentiated sum of the weights of formulas that evaluate to $\mathtt{True}$ and $p(i)$ is the number of ground atoms that are removed from the MLN as a result of removing formulas (these are essentially don't care propositional atoms which can be assigned to either $\mathtt{True}$ or $\mathtt{False}$).

Algorithm 1 provides a schematic description of LIS. It takes as input a normal MLN $\mathcal{M}$ and a proposal distribution Q. If the MLN is empty, the algorithm returns 1. Otherwise, if there exists a decomposer $\mathbf{x}$, the algorithm recurses on $\mathcal{M}[X/\mathbf{x}]$, raising the result by $|\Delta_x|$ using the power rule. The algorithm then checks if there exists a singleton atom $\mathtt{R}(x)$. If there exists one, then the algorithm samples an integer i from Q and recurses on $\mathcal{M}|\bar{\mathtt{R}}^i$ according to the generalized binomial rule. If all of the above conditions fail, the algorithm selects an atom $\mathtt{A}$, samples all of its groundings from Q and recurses on the MLN obtained by instantiating the sampled assignment $\bar{\mathtt{A}}$ (denoted by $\mathcal{M}|\bar{\mathtt{A}}$). $w(\bar{A})$ denotes the exponentiated sum of the weights of formulas that evaluate to $\mathtt{True}$ because of the assignment $\bar{\mathtt{A}}$ and $p(\bar{\mathtt{A}})$ denotes the number of ground atoms that are removed from the MLN as a result of removing formulas.

A New Lifting Rule

In this section, we illustrate the key idea behind our new lifting rule using a non-trivial MLN having just one weighted formula $f = \mathtt{R}(x, y) \wedge \mathtt{S}(y, z) \wedge \mathtt{T}(z, u)$. Note that none of the existing exact techniques (de Salvo Braz 2007; Gogate and Domingos 2011) that we are aware of can compute $Z(\{(f, w)\})$ in time that is polynomial in the domain sizes of x, y, z and u.

We begin by demonstrating how LIS will estimate the partition function of $\{(f, w)\}$ (see Fig. 1). LIS will first select an atom, either $\mathtt{R}$, $\mathtt{S}$ or $\mathtt{T}$, and check if it can be sampled in a lifted manner. For the given f, this is not possible. Therefore, it will define an importance distribution over all groundings of the selected atom and sample all of its groundings from it. Let us assume that LIS selected $\mathtt{R}$, which has $n_x n_y$ possible groundings, assuming that $|\Delta_x| = n_x$ and $|\Delta_y| = n_y$. Sampling $\mathtt{R}$ has the effect of removing it from all groundings of f, yielding an MLN having possibly $n_x n_y$ formulas of the form $\mathtt{S}(Y_i, z) \wedge \mathtt{T}(z, u)$. Note that some of the formulas in the resulting MLN can be deleted because they will evaluate to $\mathtt{False}$. Also, we can further reduce the representation by merging identical formulas; the weight of the new formula equals the sum of the weights of the merged formulas. Fig. 1(c) shows the reduced MLN obtained by instantiating the sampled assignments of $\mathtt{R}(x, y)$ given in Fig. 1(b). Now LIS can sample the remaining atoms in a lifted manner: $\{z\}$ is a decomposer and after instantiating the decomposer to a value $Z \in \Delta_z$, the remaining atoms become singleton, which can in turn be sampled using the generalized binomial rule.

We now show how we can group instantiations of $\mathtt{R}(x, y)$ yielding an estimate having smaller variance than LIS. Let $\Delta_y = \{Y_1, \ldots, Y_{n_y}\}$ and $\Delta_x = \{X_1, \ldots, X_{n_x}\}$. For $i = 1$ to n_y, let $j_i \in \{0, \ldots, n_x\}$, yielding a vector $(j_1, \ldots, j_{n_y})$. Consider the set of truth-assignments to the groundings of

MLN: $(\mathtt{R}(x,y) \wedge \mathtt{S}(y,z) \wedge \mathtt{T}(z,u), w)$	**Sampled groundings of** $\mathtt{R}(x,y)$:	**Reduced MLN after instantiating** $\mathtt{R}$:
Domains:	$\mathtt{R}(X_1,Y_1), \neg\mathtt{R}(X_1,Y_2), \neg\mathtt{R}(X_1,Y_3)$	$(\mathtt{S}(Y_1,z) \wedge \mathtt{T}(z,u), 2w)$
$\Delta_x = \{X_1,X_2,X_3\}$, $\Delta_y = \{Y_1,Y_2,Y_3\}$	$\mathtt{R}(X_2,Y_1), \mathtt{R}(X_2,Y_2), \neg\mathtt{R}(X_2,Y_3)$	$(\mathtt{S}(Y_2,z) \wedge \mathtt{T}(z,u), 2w)$
$\Delta_z = \{Z_1,Z_2,Z_3\}$, $\Delta_u = \{U_1,U_2,U_3\}$	$\neg\mathtt{R}(X_3,Y_1), \mathtt{R}(X_3,Y_2), \mathtt{R}(X_3,Y_3)$	$(\mathtt{S}(Y_3,z) \wedge \mathtt{T}(z,u), w)$
(a)	(b)	(c)

Figure 1: Illustration of lifted importance sampling. (a) An example MLN. (b) Sampled groundings of $\mathtt{R}(x,y)$. (c) Reduced MLN obtained by instantiating the sampled groundings of $\mathtt{R}$.

```
Lifted sampling of groundings of $\mathtt{R}(x,y)$:
For $i = 1,2,3$ do
    Select an index $j_i$ from $Q_{j_i|j_1,\ldots,j_{i-1}}$
    Randomly set $j_i$ of $\mathtt{R}(x,Y_i)$ to True, remaining to False.
Let the sampled indices be $j_1 = 2$, $j_2 = 2$ and $j_3 = 1$.
```

Figure 2: Illustration of the advanced grouping strategy for lifted importance sampling. Here we sample indices of j_i for each $Y_i \in \Delta_y$. Let the sampled indices j_i be as shown. Then, we will get the same MLN as the one in Fig. 1(c).

$\mathtt{R}(x,y)$ such that exactly j_i of $R(x,Y_i)$ are instantiated to True and the remaining to False. For each such group of truth assignments we have the following reduced MLN $\mathcal{M}_{j_1,\ldots,j_{n_y}} = \bigcup_{i=1}^{n_y}\{(\mathtt{S}(Y_i,z) \wedge \mathtt{T}(z,u), j_i w)\}$. Moreover, there are $\prod_{i=1}^{n_y}\binom{n_x}{j_i}$ members in this group (since for each j_i, there are $\binom{n_x}{j_i}$ ways in which j_i of $\mathtt{R}(x,Y_i)$ can be made True). Therefore, $Z(\mathcal{M})$ can be expressed as a sum over all possible vectors $(j_1,\ldots,j_{n_y})$:

$$Z(\mathcal{M}) = \sum_{j_1=0}^{n_x} \cdots \sum_{j_{n_y}=0}^{n_x} Z(\mathcal{M}_{j_1,\ldots,j_{n_y}}) \prod_{i=1}^{n_y} \binom{n_x}{j_i} \qquad (6)$$

In LIS, we sampled a truth assignment to all groundings of $\mathtt{R}(x,y)$ from a state space of size $O(2^{n_x n_y})$. If we sample from the grouping described above, the state space size reduces exponentially from $O(2^{n_x n_y})$ to $O((n_x+1)^{n_y})$.

Next, we describe how to define an importance distribution Q over this space. From Eq. (6), it is easy to see that we can define it sequentially as $\prod_{i=1}^{n_y} Q_{j_i|j_1,\ldots,j_{i-1}}$ along an order $(Y_1,\ldots,Y_n)$ of Δ_y, where each $Q_{j_i|j_1,\ldots,j_{i-1}}$ gives the conditional probability of sampling the index $j_i \in \{0,\ldots,n_x\}$ given an assignment to all previous indices.

Fig. 2 shows how to use our advanced grouping strategy to generate samples from the MLN given in Fig. 1.

The ideas presented in this section can be generalized using the following *isolated variables* rule in LIS. For a predicate symbol $\mathtt{R}$ of an MLN $\mathcal{M}$, define a logical variable x at position m in its arguments as isolated, if it is exclusive to $\mathtt{R}$ in all formulas containing $\mathtt{R}$. Let $\mathbf{x}$ denote the set of all isolated variables of $\mathtt{R}$ and let $\mathbf{y}$ denote the set of remaining variables in $\mathtt{R}$. Let $\Delta_{\mathbf{y}}$ denote the Cartesian product of the domains of variables in $\mathbf{y}$ and let $\mathbf{Y}_i$ denote the i-th element in $\Delta_{\mathbf{y}}$. Let $\mathcal{M}[\mathtt{R},\mathbf{x}]$ be an MLN obtained from $\mathcal{M}$ by applying the following steps in order: (i) for $i = 1$ to $|\Delta_{\mathbf{y}}|$, sample j_i from a distribution $Q_i(j_i|j_1,\ldots,j_{i-1})$ and set j_i arbitrarily selected groundings of $\mathtt{R}(x,\mathbf{Y}_i)$ to True and the remaining to False, (ii) Delete all formulas that evaluate to either True or False, (iii) Delete all groundings of $\mathtt{R}$ and (iv) Convert the MLN to a normal MLN. Let $w(\mathtt{R})$ be the exponentiated sum of the weights of formulas that evaluate to True and let $p(\mathtt{R})$ be the number of ground atoms that are removed from the MLN as a result of removing formulas. The unbiased estimate of $Z(\mathcal{M})$ is given by:

$$\widehat{Z}(\mathcal{M}) = \widehat{Z}(\mathcal{M}[\mathtt{R},\mathbf{x}])w(\mathtt{R})2^{p(\mathtt{R})} \prod_{i=1}^{|\Delta_{\mathbf{y}}|} \frac{\binom{|\Delta_{\mathbf{x}}|}{j_i}}{Q_i(j_i|j_1,\ldots,j_{i-1})}$$

where $\widehat{Z}(\mathcal{M}[\mathtt{R},\mathbf{x}])$ is the unbiased estimate of $Z(\mathcal{M}[\mathtt{R},\mathbf{x}])$.

Note that in general the isolated variables rule is only applicable to atoms not involved in self-joins. However, if the isolated variables appear in the same position in all instances of $\mathtt{R}$ that are involved in self-joins, we can safely apply it to $\mathtt{R}$. For efficiency reasons, the isolated variables rule should be applied only if neither the power rule nor the generalized binomial rule is applicable. In other words, we should apply the three rules in the following order: power rule, generalized binomial rule and isolated variables rule. In summary,

Theorem 1. *LIS augmented with the isolated variables rule yields an unbiased estimate of the partition function of its input MLN.*

Variance Reduction

Intuitively, the scheme that utilizes the most grouping is likely to have better accuracy because it samples a smaller (sub)space. We formalize this notion using the following grouping lemma:

Lemma 1 (Grouping Lemma). *Let Z be a sum over M numbers grouped into k groups such that all numbers in each group are identical. Let $(m_{1,1},\ldots,m_{1,g_1},\ldots,m_{k,1},\ldots,m_{k,g_k})$ denote an arbitrary ordering of the M numbers such that $\forall\ a,b,c$, $m_{a,b} = m_{a,c}$, where $a \in \{1,\ldots,k\}$, $b,c \in \{1,\ldots,g_a\}$ and g_a is the number of numbers in group a. Let Q be a proposal distribution defined over all the M numbers and R be a proposal distribution defined over the k groups such that $R(i) = \sum_{j=1}^{g_i} Q(m_{i,j})$. Then, the variance of the importance sampling estimate of Z defined with respect to R is smaller than the variance of the estimate of Z defined with respect to Q.*

The proof of Lemma 1 is straight-forward and can be derived easily from first principles. We skip it due to space constraints. Since the isolated variables rule reduces the size of the lifted space by grouping atoms (see Eq. (6)), it follows from the grouping lemma that:

Theorem 2. *LIS augmented with the isolated variables rule has smaller variance than LIS.*

Constructing the Proposal Distribution

As mentioned earlier, the accuracy of any importance sampling algorithm depends on how close the proposal distribution is to the target distribution (the one represented by the MLN). Often, practical constraints dictate that the importance distribution should be polynomially specifiable (i.e., tractable) as well as easy to sample from. To construct such a tractable distribution for MLNs, a natural choice is to use the generalized binomial rule approximately by relaxing the requirement that the atom must be a singleton. For example,

EXAMPLE **1.** Consider our example MLN, $\mathcal{M} = \{(\mathtt{R}(x,y) \wedge \mathtt{S}(y,z) \wedge \mathtt{T}(z,u), w)\}$. Applying the approximate generalized binomial rule to $\mathtt{R}(x,y)$, we can rewrite the partition function as $\sum_{i=0}^{|\Delta_x \times \Delta_y|} Z(\mathcal{M}|\bar{\mathtt{R}}^i)$. Each MLN, $\mathcal{M}|\bar{\mathtt{R}}^i$ is tractable and therefore we can associate a tractable probability distribution, say $Q(\mathcal{M}|\bar{\mathtt{R}}^i)$ with each. The full proposal distribution is $Q(i)Q(\mathcal{M}|\bar{\mathtt{R}}^i)$ where $Q(i)$ is the distribution defined over $|\Delta_x \times \Delta_y|+1$ points, where each i-th point corresponds to setting exactly i groundings of $\mathtt{R}(x,y)$ to `True` and the remaining to `False`.

Although the approximate rule reduces the branching factor (of the search space) from $2^{|\Delta_R|}$ to $|\Delta_R| + 1$ for an atom $\mathtt{R}$, it is still infeasible when the number of atoms is large. In particular, we will assume that the proposal distribution is specified in the product form, i.e., a relational Bayesian network (Jaeger 1997). Formally, given an ordered set of atoms $(\mathtt{R}_1, \ldots, \mathtt{R}_m)$, the proposal distribution is given by $\prod_{i=1}^{m} Q_i(\mathtt{R}_i|\mathtt{R}_1, \ldots, \mathtt{R}_{i-1})$. The space required by this product form will be $O(m[\max_i(|\Delta_{\mathtt{R}_i}|)]^m)$, where $\Delta_{\mathtt{R}_i}$ is the Cartesian product of arguments of $\mathtt{R}_i$. Therefore, in order to achieve polynomial complexity, we make the following conditional independence assumption: $\mathtt{R}_i$ is conditionally independent of all other atoms given k atoms from the set $\{\mathtt{R}_1, \ldots, \mathtt{R}_{i-1}\}$, where k is a constant. Thus, each component of the proposal distribution is of the form: $Q_i(\mathtt{R}_i|pa(\mathtt{R}_i))$ where $pa(\mathtt{R}_i) \subseteq \{\mathtt{R}_1, \ldots, \mathtt{R}_{i-1}\}$ and $|pa(\mathtt{R}_i)| \leq k$. We will refer to $pa(\mathtt{R}_i)$ as the parents of $\mathtt{R}_i$.

Algorithm 2 describes a recursive approach for constructing the proposal distribution using the ideas discussed above. The algorithm takes as input an MLN $\mathcal{M}$, a constant k that limits the parent size for each atom (in our experiments, we used $k = 2$), and the potential parent set $\mathbf{R}$. The algorithm first checks the base condition: if the MLN is empty, it returns a 1. Then, the algorithm checks if there is a decomposer $\mathbf{x}$. If there exists one, the algorithm recurses on the reduced MLN $\mathcal{M}[X/\mathbf{x}]$, where $x \in \mathbf{x}$ and $X \in \Delta_x$, and then exits. Otherwise, the algorithm checks if the MLN can be decomposed into (multiple) independent MLNs (if two MLNs do not share any atoms, they are independent). If it can be decomposed, the algorithm recurses on the independent MLNs and exits. Then the algorithm heuristically selects an atom $\mathtt{R}_i$ and selects k atoms from the potential parent set $\mathbf{R}$ as parents of $\mathtt{R}$. It then constructs the proposal distribution component for $\mathtt{R}$ (described below), adds $\mathtt{R}$ to $\mathbf{R}$, reduces the MLN by removing $\mathtt{R}$ from all formulas and recurses on the reduced MLN.

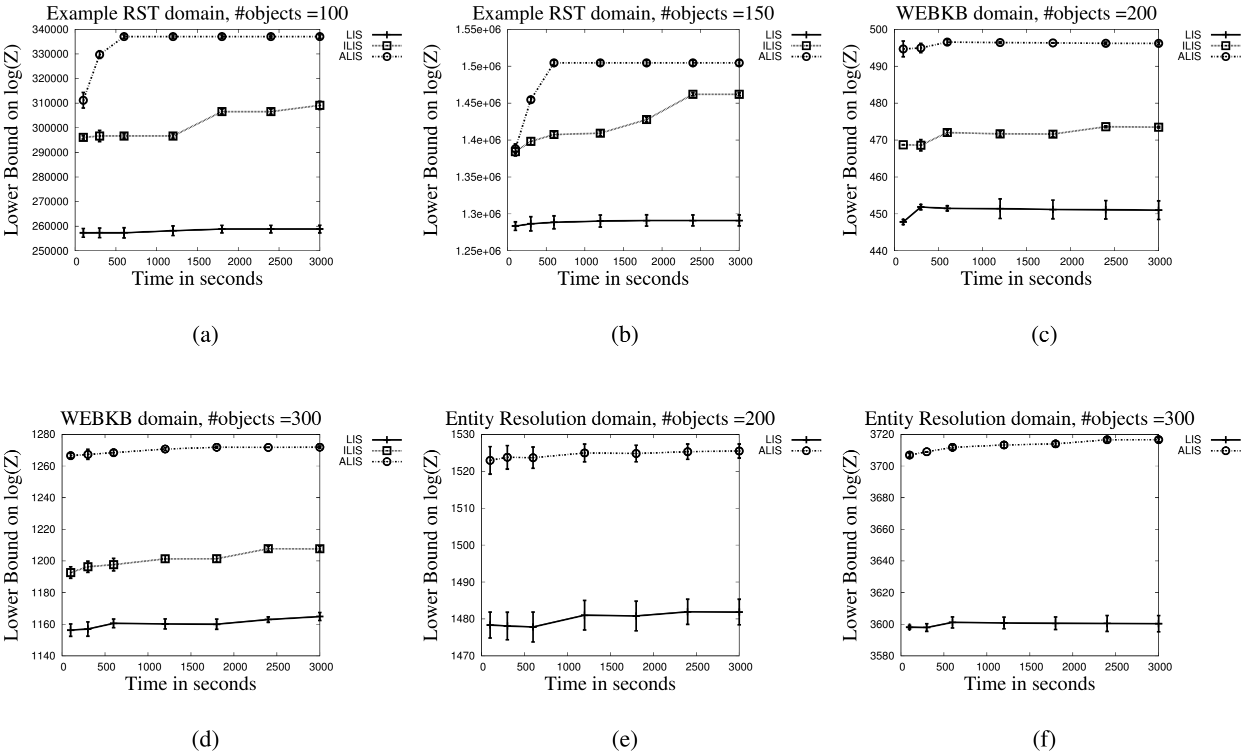
Algorithm 2: Construct Proposal (CP)

```
   Input: An MLN $\mathcal{M}$, an integer $k$ and a set of atoms $\mathbf{R}$
   Output: The structure of the proposal distribution Q
1  if $\mathcal{M}$ is empty then return 1
2  if there exists a decomposer x then
3      Let $x \in \mathbf{x}$ and $X \in \Delta_x$. return CP($\mathcal{M}[X/\mathbf{x}], k, \mathbf{R}$)
4  if $\mathcal{M}$ can be decomposed into $m$ MLNS $\mathcal{M}_1, \ldots, \mathcal{M}_k$ such
   that no two MLNs share any atoms then
5      for $i = 1$ to $m$ do
6          CP($\mathcal{M}_i, k, \mathbf{R}$)
7      return 1
8  Heuristically select an atom R from $\mathcal{M}$
9  Heuristically select $k$ atoms from $\mathbf{R}$ as parents of R
   // Construct Proposal over R
10 for every assignment to the groundings of $pa$(R) index by $i$ do
11     if R contains no isolated variables then
12         Use the approximate generalized binomial rule to
           construct $Q_i$(R)
13     else
14         Use the isolated variables rule to construct $Q_i$(R)
15 Add R to $\mathbf{R}$
16 Ground R and then remove it from all formulas of $\mathcal{M}$
17 return CP($\mathcal{M}, k, \mathbf{R}$)
```

The proposal distribution component for $\mathtt{R}$ is computed as follows. Given an assignment to all groundings of the parents, denoted by $\overline{pa(\mathtt{R})}$ each conditional marginal distribution $Q(\mathtt{R}|\overline{pa(\mathtt{R})})$ is constructed as follows. If $\mathtt{R}$ contains a set $\mathbf{x}$ of isolated variables, we use the following method. Let $\mathbf{y}$ denote the set of variables which are not isolated in $\mathtt{R}$. Note that to effectively utilize the isolated variables rule, we have to sample a number in the range $[0, |\Delta_{\mathbf{x}}|]$, for each value $\mathbf{Y} \in \Delta_{\mathbf{y}}$. We propose to express this distribution using a product of $|\Delta_{\mathbf{y}}|$ marginal distributions, each defined over $|\Delta_{\mathbf{x}}| + 1$ points. Namely, using notation from the previous section, we define $Q_i(j_1, \ldots, j_{|\Delta_{\mathbf{y}}|}) = \prod_{a=1}^{|\Delta_{\mathbf{y}}|} Q_{i,a}(j_a)$. If $\mathtt{R}$ has no isolated atoms then we use the approximate generalized binomial rule and define a distribution over $|\Delta_A| + 1$ points. To limit the number of assignments $\overline{pa(\mathtt{R})}$ (see line 10 of Algorithm 2), we group the assignments to each atom $\mathtt{A} \in pa(\mathtt{R})$ into $|\Delta_{\mathtt{A}}| + 1$ groups, where the j-th group has j groundings of $\mathtt{A}_i$ set to `True` and the remaining to `False`. This helps us polynomially bound the space required by the proposal distribution component at $\mathtt{R}$. In particular, the space complexity of each component is $O(|\Delta_{\mathtt{R}}|(\sum_{\mathtt{A} \in pa(\mathtt{R})} |\Delta_A|))$.

We use the following heuristics to select the atom $\mathtt{R}$: Select any singleton atom. Otherwise, select an atom that participates in most formulas, ties broken randomly. This heuristic is inspired by the max-degree conditioning heuristic which often yields a smaller search space. To select parents of $\mathtt{R}$, we first select atoms, say $\mathbf{R}_1$, that are mentioned in the same formula that $\mathtt{R}$ participates in, followed by atoms which participate in formulas that atoms in $\mathbf{R}_1$ participate in and so on. Again, ties are broken randomly.

Until now, we have described an algorithm that outputs



(a) (b) (c)

(d) (e) (f)

Figure 3: Lower bound on the partition function computed using LIS, ILIS and ALIS as a function of time. (a) The example R, S, T domain with 100 objects. (b) The example R, S, T domain with 150 objects. (c) WEBKB MLN with 200 objects, (d) WEBKB MLN with 300 objects, (e) Entity Resolution MLN with 200 objects and (f) Entity resolution MLN with 300 objects. Note that for each point, we have plotted error bars showing the standard deviation. When the standard deviation is small, the error bars are not visible in the plots.

the structural form of the proposal distribution. To use it in LIS, we have to define its parameters. Moreover, we should define its parameters in such a way that the resulting distribution is as close as possible to the target distribution. In principle, we can use any approximate inference method such as lifted BP (Singla and Domingos 2008) to compute the parameters. However, because of the relatively high time-complexity of lifted BP, this approach is not likely to be cost effective.

Therefore, we use the following adaptive importance sampling approach (Cheng and Druzdzel 2000; Ortiz and Kaelbling 2000) that dynamically updates the proposal distribution Q based on the generated samples. The updating step is performed every l samples. Since the initial proposal distribution, no matter how well chosen, is often very different from the target distribution, dynamic updating can substantially improve the accuracy of importance sampling. The hope is that as more and more samples are drawn, the updated proposal distribution gets closer and closer to the target distribution. We initialize the proposal distribution to the prior distribution Q^0, defined by a collection of components Q_i^0 (for each atom R chosen in Algorithm 2). After every l samples, we update each component Q_i^m using the expression $Q_i^{m+1}(j) = Q_i^m(j) + \alpha(m)(\Pr(j) - Q_i^m(j))$ where $0 \leq \alpha(m) \leq 1$ is the learning rate and $\Pr(j)$ is the estimate of the probability of j based on the last l samples. In our experiments, we set $\alpha(m) = 0.1$ and $l = 10^3$.

Experiments

In this section, we compare the performance of LIS (see Algorithm 1) with two advanced versions: (i) LIS augmented with the new lifting rule and (ii) LIS augmented with the new lifting rule and the adaptive structured method for constructing the proposal distribution described in the previous section. We will call the two new schemes isolated variables' rule LIS (ILIS) and adaptive LIS (ALIS) respectively. Note that both LIS and ILIS use the same proposal distribution as the one used in (Gogate and Domingos 2011) while ALIS uses the structured, adaptive approach described in the previous section. We experimented with three MLNs: the example R, S, T MLN used in this paper, the WEBKB MLN used in (Lowd and Domingos 2007) and the Entity resolution MLN used in (Singla and Domingos 2006). The last two MLNs are publicly available from www.alchemy.cs.washington.edu. We set the weights of each formula in each MLN arbitrarily by sampling a value from the range $(-1, 1)$. For each MLN, we set 10% randomly selected ground atoms as evidence. We varied the number of objects in the domain from 100 to 300.

Because computing the partition function of the MLNs used is not feasible, we use the following approach for evaluating the algorithms. We use the sampling algorithms to compute a probabilistic lower bound on the partition function. The higher the lower bound the better the sampling algorithm. For computing the lower bound, we combine our sampling algorithms with the *Markov inequality based minimum lower bounding scheme* presented in (Gogate, Bidyuk, and Dechter 2007). This lower bounding scheme, see also (Gomes et al. 2007), takes as input a set of unbiased estimates of the partition function and a real number $0 < \alpha < 1$, and outputs a lower bound on the partition function that is correct with probability greater than α. Formally,

Theorem 3. *(Gomes et al. 2007; Gogate, Bidyuk, and Dechter 2007) Let $\widehat{Z}_1, \ldots, \widehat{Z}_m$ be the unbiased estimates of Z computed over m independent runs of an importance sampling algorithm. Let $0 < \alpha < 1$ be a constant and let $\beta = \frac{1}{(1-\alpha)^{1/m}}$. Then $Z_{lb} = \frac{1}{\beta}\left[\min_{i=1}^{m}(\widehat{Z}_m)\right]$ is a lower bound on Z with probability greater than α.*

In our experiments, we set $\alpha = 0.99$ and $m = 7$, namely, we run each sampling algorithm 7 times and each lower bound is correct with probability greater than 0.99.

Figure 3 shows the impact of varying time and number of objects on the performance of the three algorithms. Note that the Entity Resolution MLN has no isolated variables and as a result LIS is equivalent to ILIS. Therefore, for this domain, we only compare LIS with ALIS. Also, note that we are plotting the log partition function as a function of time and therefore the Y-axis is in log-scale. From Figure 3, it is easy to see that ALIS is superior to ILIS which in turn is superior to LIS. Moreover, from the error bars in Figure 3, we see that the variance of ALIS and ILIS is typically smaller than that of LIS.

Summary and Future Work

In this paper, we improved the lifted importance sampling algorithm (LIS) in two ways. First, we proposed a new lifting rule that reduces the variance of its estimates. Second, we proposed a new, structured approach for constructing the proposal distribution and dynamically learning its parameters via adaptive importance sampling. Our experiments on many real-world and artificial domains showed that our new, advanced algorithm is substantially more accurate than LIS.

Future work includes: developing new lifting rules, learning the initial parameters of the proposal distribution using a variational approach, combining improved LIS with exact inference (Rao-Blackwellised sampling), using our algorithm for weight learning, applying our lifting rules to existing relational MCMC approaches (Milch and Russell 2006; Liang, Jordan, and Klein 2010), etc.

Acknowledgements This research was partly funded by the ARO MURI grant W911NF-08-1-0242. The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of ARO or the U.S. Government.

References

Cheng, J., and Druzdzel, M. J. 2000. AIS-BN: An adaptive importance sampling algorithm for evidential reasoning in large Bayesian networks. *Journal of Artificial Intelligence Research* 13:155–188.

De Raedt, L.; Kimmig, A.; and Toivonen, H. 2007. Problog: A probabilistic prolog and its application in link discovery. In *IJCAI*, 2462–2467.

de Salvo Braz, R. 2007. *Lifted First-Order Probabilistic Inference*. Ph.D. Dissertation, University of Illinois, Urbana-Champaign, IL.

Domingos, P., and Lowd, D. 2009. *Markov Logic: An Interface Layer for Artificial Intelligence*. Morgan & Claypool.

Friedman, N.; Getoor, L.; Koller, D.; and Pfeffer, A. 1999. Learning probabilistic relational models. In *IJCAI*, 1300–1307.

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

Gogate, V., and Domingos, P. 2011. Probabilistic theorem proving. In *UAI*, 256–265.

Gogate, V.; Bidyuk, B.; and Dechter, R. 2007. Studies in lower bounding probability of evidence using the Markov inequality. In *UAI*, 141–148.

Gomes, C. P.; Hoffmann, J.; Sabharwal, A.; and Selman, B. 2007. From sampling to model counting. In *IJCAI*, 2293–2299.

Jaeger, M. 1997. Relational Bayesian networks. In *UAI*, 266–273.

Jha, A.; Gogate, V.; Meliou, A.; and Suciu, D. 2010. Lifted inference from the other side: The tractable features. In *NIPS*, 973–981.

Kisynski, J., and Poole, D. 2009. Constraint processing in lifted probabilistic inference. In *UAI*, 293–302.

Liang, P.; Jordan, M. I.; and Klein, D. 2010. Type-based MCMC. In *HLT-NAACL*, 573–581.

Liu, J. 2001. *Monte-Carlo strategies in scientific computing*. Springer-Verlag, New York.

Lowd, D., and Domingos, P. 2007. Efficient weight learning for Markov logic networks. In *EMML-PKDD*, 200–211.

Milch, B., and Russell, S. J. 2006. General-purpose MCMC inference over relational structures. In *UAI*, 349–358.

Milch, B.; Marthi, B.; Russell, S. J.; Sontag, D.; Ong, D. L.; and Kolobov, A. 2005. BLOG: Probabilistic models with unknown objects. In *IJCAI*, 1352–1359.

Milch, B.; Zettlemoyer, L. S.; Kersting, K.; Haimes, M.; and Kaelbling, L. P. 2008. Lifted probabilistic inference with counting formulas. In *AAAI*, 1062–1068.

Ng, K. S.; Lloyd, J. W.; and Uther, W. T. 2008. Probabilistic modelling, inference and learning using logical theories. *Annals of Mathematics and Artificial Intelligence* 54(1-3):159–205.

Ortiz, L. E., and Kaelbling, L. P. 2000. Adaptive importance sampling for estimation in structured domains. In *UAI*, 446–454.

Poole, D. 2003. First-order probabilistic inference. In *IJCAI*, 985–991.

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

Singla, P., and Domingos, P. 2006. Entity resolution with Markov logic. In *ICDM*, 572–582.

Singla, P., and Domingos, P. 2008. Lifted first-order Belief propagation. In *AAAI*, 1094–1099.

Van den Broeck, G.; Taghipour, N.; Meert, W.; Davis, J.; and De Raedt, L. 2011. Lifted probabilistic inference by first-order knowledge compilation. In *IJCAI*, 2178–2185.