

# Exact Lifted Inference with Distinct Soft Evidence on Every Object

Hung H. Bui and Tuyen N. Huynh and Rodrigo de Salvo Braz

Artificial Intelligence Center  
SRI International  
Menlo Park, CA 94025, USA  
{bui,huynh,braz}@ai.sri.com

## Abstract

The presence of non-symmetric evidence has been a barrier for the application of lifted inference since the evidence destroys the symmetry of the first-order probabilistic model. In the extreme case, if distinct soft evidence is obtained about each individual object in the domain then, often, all current exact lifted inference methods reduce to traditional inference at the ground level. However, it is of interest to ask whether the symmetry of the model itself before evidence is obtained can be exploited. We present new results showing that this is, in fact, possible. In particular, we show that both exact maximum a posteriori (MAP) and marginal inference can be lifted for the case of distinct soft evidence on a unary Markov Logic predicate. Our methods result in efficient procedures for MAP and marginal inference for a class of graphical models previously thought to be intractable.

## 1 Introduction

First-order probabilistic models (FOPMs) (Richardson and Domingos 2006; Getoor et al. 2001; Ngo and Haddawy 1995; Poole 1993) are graphical models specified in a relational manner (in other words, a compact, intensional, or template-based manner), as opposed to the traditional representation in which every random variable and dependency is listed explicitly. FOPMs are an intuitive and compact representation, and preserve exploitable domain structure by directly indicating dependencies and parameters shared across random variables.

Inference on FOPMs can be done by grounding them to a regular graphical model and using standard graphical model algorithms. A much more efficient alternative is to perform *lifted inference* (Gogate and Domingos 2011; Jha et al. 2010; Milch et al. 2008; Singla and Domingos 2008; de Salvo Braz, Amir, and Roth 2005; Poole 2003), in which the first-order structure is used and preserved by the algorithm during inference, and in which each computation step corresponds to many computation steps performed by a regular algorithm (in a way analogous to first-order logic (FOL) resolution as opposed to propositional resolution).

Traditional lifted inference operates by identifying sets of individuals (the values which parameterize, or index, random variables) that are symmetrical given the model, that

is, about which we have exactly the same knowledge. This renders them interchangeable, and computations performed for one of them hold for *all* of them. The time complexity of lifted inference can be polynomial or even constant on the size of such symmetrical individual sets, while inference on the grounded model is exponential in it.

One obstacle to the application of lifted inference algorithms presented to date is the presence of distinct soft evidence on every individual object in the domain of the first-order language. This introduces unique knowledge about each individual and breaks the symmetry of the model, forcing inference to be performed on the ground level, typically with exponential time complexity. This paper considers the simplest form of distinct evidence: those that arise when there is distinct soft evidence on every grounding instance of a single unary predicate. The main contribution of this paper is a technique (LIDE, for Lifted Inference with Distinct Evidence) for efficient exact lifted inference in this case. Note that distinct evidence even in this simplest form would typically cause existing lifted inference methods to revert to ground inference.

LIDE works in the following manner. Instead of shattering (that is, breaking symmetries in) the original model with distinct soft evidence, as previous lifted inference methods do, LIDE first applies lifted inference to the model without the problematic soft evidence. Because the model is unshattered in this step, this calculation can take full advantage of existing lifted inference methods. LIDE then uses the marginal probability of the previous step as a prior, and along with the distinct evidence, computes the final posterior probability. To achieve efficient computation in this key step ( $O(n \ln n)$  for MAP and  $O(n^2)$  for marginal inference), we exploit the symmetry of the partition function of the probability model, a notion of symmetry previously not exploited in current lifted inference methods. LIDE thus acts as an efficient wrapper around another lifted inference method, shielding it from the problematic soft evidence. Our experiments show that LIDE can perform *exact* lifted inference under the presence of distinct soft evidence much faster than (lifted) belief propagation (which is a very efficient *approximate* method), without suffering from non-convergence issues or approximation errors.

We begin by providing some background on FOPMs and lifted inference. Then we show how to reduce certain

queries on a Markov Logic Network (MLN) with distinct soft evidence on each individual to a calculation that takes advantage of the symmetry in the original MLN (without the evidence). We discuss maximum a posteriori (MAP), partition function and marginal variants of inference, ending with the presentation of empirical results.

## 2 Background

### 2.1 First-order Probabilistic Models

A **factor**  $f$  is a pair  $(A_f, \phi_f)$  where  $A_f$  is a tuple of random variables and  $\phi_f$  is a **potential function** from the range of  $A_f$  to the non-negative real numbers. Given a **valuation**  $v$  of random variables (**rvs**), the **potential** of  $f$  on  $v$  is  $\phi_f(v) = \phi_f(v(A_f))$ .

The joint probability defined by a set  $F$  of factors on a valuation  $v$  of random variables is the normalization of  $\prod_{f \in F} \phi_f(v)$ . If each factor in  $F$  is a conditional probability of a child random variable given the value of its parent random variables, and there are no directed cycles in the graph formed by directed edges from parents to children, then the model defines a Bayesian network. Otherwise, it is an undirected model.

We can have parameterized (indexed) random variables by using **predicates**, which are functions mapping parameter values (indices) to random variables. A **relational atom** is an application of a predicate, possibly with free variables. For example, a predicate *Friends* is used in atoms *Friends(x, y)*, *Friends(x, Bob)* and *Friends(John, Bob)*, where  $x$  and  $y$  are free variables and *John* and *Bob* possible parameter values. *Friends(John, Bob)* is a **ground atom** and directly corresponds to a random variable.

A **parfactor** is a tuple  $(L, C, A, \phi)$  composed of a set of parameters (also called *logical variables*)  $L$ , a constraint  $C$  on  $L$ , a tuple of atoms  $A$ , and a potential function  $\phi$ . Let a **substitution**  $\theta$  be an assignment to  $L$  and  $A\theta$  the relational atom (possibly ground) resulting from replacing logical variables by their values in  $\theta$ . A pactor  $g$  stands for the set of factors  $gr(g)$  with elements  $(A\theta, \phi)$  for every assignment  $\theta$  to the parameters  $L$  that satisfies the constraint  $C$ .

A First-order Probabilistic Model (**FOPM**) is a compact, or intensional, representation of a graphical model. It is composed by a **domain**, which is the set of possible parameter values (referred to as **domain objects**) and a set of pactors. The corresponding graphical model is the one defined by all instantiated factors. The joint probability of a valuation  $v$  according to a set of pactors  $G$  is

$$P(v) = 1/Z \prod_{g \in G} \prod_{f \in gr(g)} \phi_f(v), \quad (1)$$

where  $Z$  is a normalization constant.

*Example:* the following is an example of an FOPM in which the prior of an epidemic is 0.1 and the probability of each individual getting sick is 0.4 in the case of an epidemic and 0.1 otherwise, but for an individual *Bob*, who is known

to be sick:

$$\begin{aligned} &(\emptyset, \top, (\textit{Epidemic}), P(\textit{Epidemic}) = 0.1) \\ &(\{x\}, x \neq \textit{Bob}, (\textit{Sick}(x), \textit{Epidemic}), \\ &P(\textit{Sick}(x)|\textit{Epidemic}) = \begin{cases} 0.4, & \text{if Epidemic} \\ 0.1, & \text{otherwise} \end{cases} ) \\ &(\emptyset, \top, (\textit{Sick}(\textit{Bob})), P(\textit{Sick}(\textit{Bob})) = 1) \end{aligned}$$

where  $\top$  is the tautological constraint.

The formalism above subsumes that of MLNs (Richardson and Domingos 2006). An MLN with a set of formulas  $\{F_i\}_i$  with weights  $\{w_i\}_i$ , logical variables  $\{L_i\}_i$  and random variables  $\{RV_i\}_i$  for each  $i$ -th clause is equivalent to the FOPM  $\{(L_i, \top, RV_i, e^{w_i F_i})\}_i$ .

We use the “Friends & Smokes” MLN (Singla and Domingos 2008) as a running example in this paper:

$$\begin{aligned} 1.4 : & \neg \textit{Smokes}(x) \\ 2.3 : & \neg \textit{Cancer}(x) \\ 4.6 : & \neg \textit{Friends}(x, y) \\ 1.5 : & \textit{Smokes}(x) \Rightarrow \textit{Cancer}(x) \\ 1.1 : & \textit{Smokes}(x) \wedge \textit{Friends}(x, y) \Rightarrow \textit{Smokes}(y) \end{aligned}$$

### 2.2 Lifted Inference and Counting Formulas

Lifted inference (Gogate and Domingos 2011; Jha et al. 2010; Milch et al. 2008; Singla and Domingos 2008; de Salvo Braz, Amir, and Roth 2005; Poole 2003) is a set of techniques for performing inference on the FOPM representation without simply reducing the FOPMs to propositional (ground) graphical models first. In the epidemic example model, lifted inference can exactly compute the marginal probability of an epidemic given that 100 people out of a million are sick *without* instantiating a million random variables, and in time *constant* in the number of people in the problem.

All of these methods, with the exception of (Kersting et al. 2010), are guaranteed to revert back to ground inference under the presence of the kind of distinct soft evidence considered in this paper. To see why, suppose that from the Friends and Smokes MLN, we obtain the soft evidence  $w_i : \textit{Cancer}(P_i)$ ,  $i = 1 \dots n$ , where  $w_i$  are distinct real numbers. The exact lifted inference methods (Gogate and Domingos 2011; Milch et al. 2008; de Salvo Braz, Amir, and Roth 2005; Poole 2003) would treat each individual  $P_i$  differently, leading to the grounding of the whole model. Lifted BP (Singla and Domingos 2008) and Counting BP (Kersting, Ahmadi, and Natarajan 2009) both treat the distinct soft evidence as distinct factors, leading to distinct computation trees for every individual messages; thus these methods also revert to ground inference. While it is plausible that a quantization-based approximation such as (Kersting et al. 2010) can find a grouping of variables for lifting in this case, distinct soft evidence would pose a serious challenge for this approach, especially if the strengths of the soft evidence are uniformly distributed.

Since counting formulas play an important role in the LIDE method, we describe the method in more details. C-

FOVE (Milch et al. 2008) introduced the notion of **counting formulas** to lifted inference. Counting formulas are random variables representing the histogram of values in a set of symmetric random variables. In the “Friends & Smokes” MLN example,  $\{Cancer(x)\}_x$  is a set of symmetric random variables (in the absence of evidence distinguishing them from each other), and  $\#_x[Cancer(x)]$  is a counting formula ranging over histograms of how many individuals have cancer and how many do not.

Imagine that we want to compute the joint marginal probability of  $\{Cancer(x)\}_x$ . Because the fifth clause in the model connects  $Smokes(x)$  and  $Smokes(y)$ , the instances of  $\{Cancer(x)\}_x$  become connected and their marginal cannot be provided in a factorized form with a factor per  $x$ . However, this marginal can be provided as a function of the counting formula  $\#_x[Cancer(x)]$ , because it only depends on *how many* individuals have cancer, not *which* ones (since, after all, the individuals in this set are interchangeable). This allows a much more compact representation than a function on the exponentially many assignments on  $\{Cancer(x)\}_x$ .

### 3 Lifted Inference With Distinct Evidence (LIDE)

Consider an MLN  $M_0$  and a unary predicate  $q$  that appears in  $M_0$ , where each instance  $q(x)$  of  $q$  is a binary ( $\{0, 1\}$ -valued) random variable. Consider also soft evidence on every instance  $q(x)$  of  $q$  in the form of a weighted formula<sup>1</sup>  $w_x : q(x)$  where the weights  $w_x$  are distinct, which together with  $M_0$  form an MLN  $M$ . With traditional lifted inference, the symmetry of the model is destroyed by the sentences on individual random variables, which causes existing lifted inference methods to revert back to ground inference. For example, we may have  $M_0$  equal to the MLN in the “Friends & Smokes” example, and  $M$  equal to  $M_0$  plus the distinct soft evidence on every instance of  $Cancer(x)$  (for example, the result of a screening test providing a probability of cancer on an individual). We want to take the prior knowledge encoded in  $M_0$  into account in order to assess the posterior probability of each individual actually having cancer. It is clear that current exact (and some approximate) lifted inference methods would reduce to ground inference in the presence of this type of evidence.

The main contribution of this paper is to show how the symmetry present in  $M_0$  can still be exploited in this case. The method is based on obtaining the prior probability on the counting formula of the random variables on which one has evidence (in this case,  $\{Cancer(x)\}_x$ ). Therefore, this prior is a symmetric function on that set of random variables. Because this prior probability is computed before the evidence is taken into account, it can be obtained with lifted inference from  $M_0$  in time polynomial in the size of the symmetric sets of random variables. Only then is the evidence taken into account, in time also polynomial, exploiting the fact that the prior computed from  $M_0$  is a symmetric function.

<sup>1</sup>This set of evidence can also be modeled as a set of factors  $\{\phi_x(q(x))\}$

### 3.1 Symmetry of Ground Instances of MLN Unary Predicates

Here, we study the symmetry of vectors of random variables resulting from grounding a unary predicate in a given MLN. We start with a formal definition of symmetric functions.

**Definition 1.** A  $n$ -variable function  $f(t_1, \dots, t_n)$  is symmetric if for all permutation  $\pi \in \mathbb{S}_n$ , the set of all permutations on  $n$  items, permuting the variables of  $f$  by  $\pi$  does not change the output value, that is,  $f(t_1, \dots, t_n) = f(t_{\pi(1)} \dots, t_{\pi(n)})$ .

In the case where the variables  $t_i$  of a symmetric function  $f$  are binary, it is clear that the value of  $f$  depends only on the number of 1 values in the vector  $t$ . Thus,  $f$  can be summarized precisely by  $n + 1$  values  $c_k$ ,  $k = 0, \dots, n$ , where  $c_k = f(t)$  for any  $t$  such that  $\|t\|_1 = k$ . The set of  $c_k$  is termed the *counting* representation of the symmetric function  $f$ .

Given the MLN  $M_0$ , let  $P_0$  be the distribution defined by  $M_0$ , and let  $q$  be a unary predicate in  $M_0$ . For ease of exposition, we assume that all individuals of  $M_0$  are from a single domain (type)  $\mathcal{D}$ , but this requirement is not essential and can be relaxed. Assume that the MLN  $M_0$  mentions a set of constants  $\mathcal{D}_0 \subset \mathcal{D}$ , and let  $\mathcal{D}_* = \mathcal{D} \setminus \mathcal{D}_0 = \{d_1, \dots, d_n\}$  be the set of individuals in the domain that do not appear in the MLN. Since  $M_0$  mentions no specific individual in  $\mathcal{D}_*$ , all individuals in it are exchangeable. Therefore, we can expect that the random variables  $(q(d_1) \dots q(d_n))$  are exchangeable under  $\Pr(\cdot | M_0)$  as stated formally in the following theorem.

**Theorem 1.** Let  $\mathcal{D}_* = \{d_1, \dots, d_n\}$  be the set of individuals that do not appear as constants in the MLN  $M_0$  and let  $q$  be a unary predicate in  $M_0$ . Then,  $P_0(\cdot) = \Pr(q(d_1) \dots q(d_n) | M_0)$  is a symmetric function of  $n$  variables. Equivalently, the random vector  $(q(d_1) \dots q(d_n))$  is exchangeable under  $P_0$ .

The proof of this theorem can be found in the appendix. While the above result might seem intuitive, it does not hold for the case when the arity of  $q$  is strictly larger than one.

Given the application of lifted inference techniques on  $M_0$  alone, we can obtain a potential function on the counting formula  $\#_{x \in \mathcal{D}_*}[q(x)]$ , which is a counting representation of the symmetric function  $P_0(q(d_1) \dots q(d_n))$ . In the next subsection, we show how to take into account the distinct evidence on each instance of  $q(x)$ , where  $x$  ranges over the individuals in  $\mathcal{D}_*$ .

### 3.2 Posterior of Exchangeable Binary Random Variables

To proceed, we make several simplifications of the notations:  $q(d_i)$  will be referred to as  $q_i$ , and the potential representing the evidence  $\phi_{d_i}(q(d_i))$  simply as  $\phi_i(q_i)$ . The problem of incorporating the evidence concerns the posterior of the random vector  $q$ , which can be written in the form  $P(q_1, \dots, q_n) = \frac{1}{Z} F(q_1 \dots q_n) \prod_{i=1}^n \phi_i(q_i)$  where  $F(q_1 \dots q_n)$  denote a symmetric function on  $q = (q_1, \dots, q_n)$  (of which  $P_0$  is one possible example). In particular, we would like to compute the MAP assignment  $q$ ,

the normalizing term  $Z$ , and the marginal probability  $P(q_i)$  for some  $i$ . Note that  $q$  can be thought of as having an exchangeable prior distribution whose posterior needs to be computed.

We will assume that  $\phi_i(0) > 0$  for ease of exposition. Let  $\alpha_i = \frac{\phi_i(1)}{\phi_i(0)}$ . Note that  $\phi_i(q_i) = \alpha_i^{q_i} \phi_i(0)$ , so

$$\begin{aligned} P(q) &= \frac{1}{Z} F(q_1 \dots q_n) \prod_{i=1}^n \alpha_i^{q_i} \phi_i(0) \\ &= \frac{1}{Z} F(q_1 \dots q_n) \prod_{i=1}^n \alpha_i^{q_i} \left( \prod_{i=1}^n \phi_i(0) \right) \\ &= \frac{\Phi}{Z} F(q_1 \dots q_n) \prod_{i=1}^n \alpha_i^{q_i} \end{aligned}$$

where  $\Phi = \prod_{i=1}^n \phi_i(0)$  does not depend on  $q$ .

The key fact allowing efficient computation is that  $F$  is a symmetric function on a vector of binary variables. We know that  $F$  depends only on the number of values 1 in  $q$ , so let  $c_k = F(q)$  where  $\|q\|_1 = k$ . Then, for vector  $q$  with  $k$  values 1 and indices  $i_1, \dots, i_k$  such that  $q_{i_1} \dots q_{i_k}$  are 1 and the rest of  $q$ 's values are 0, we have

$$P(q) = \frac{\Phi}{Z} c_k \prod_{j=1}^k \alpha_{i_j} \quad (2)$$

### 3.3 MAP Inference

We now show how to compute the MAP assignment of the set of all groundings of the unary predicate  $q$  given the evidence. Since  $\alpha$  depends only on the evidence  $\{\phi_i\}$ , we can sort it in decreasing order in advance into another vector  $\alpha' = \alpha_{\pi(1)} \dots \alpha_{\pi(n)}$ . Then with the help of Eq. (2)

$$\begin{aligned} \max_q P(q) &= \max_k \max_{q: \|q\|_1=k} P(q) \\ &= \max_k \max_{q: \|q\|_1=k} \frac{\Phi}{Z} c_k \prod_{j=1}^k \alpha_{i_j} = \max_k \frac{\Phi}{Z} c_k \max_{q: \|q\|_1=k} \prod_{j=1}^k \alpha_{i_j} \end{aligned}$$

The second maximization allows us to choose any vector  $q$  with  $\|q\|_1 = k$ , and is trivially solved by picking  $q$  that selects the largest  $k$  values in  $\alpha$ , which are the  $k$  first values of  $\alpha'$ , that is,  $q$  such that  $q_i = 1$  if  $i \in \{\pi(1), \dots, \pi(k)\}$  and 0 otherwise. This constant-time maximization is then repeated for each value of  $k$ , as shown in Algorithm 1. The complexity of this algorithm is  $O(n \log n)$  due to the sorting step.

### 3.4 Compute $Z$

By the definition of the normalization term,  $Z = \Phi \sum_{q_1 \dots q_n} F(q) \prod_{i=1}^n \alpha_i^{q_i}$ . We first observe that  $Z$  is a multivariate polynomial of the variables  $\alpha_1 \dots \alpha_n$ . Furthermore,  $Z$  as a function of  $\alpha_1 \dots \alpha_n$  is also symmetric; thus, it is a symmetric polynomial in  $\alpha_1 \dots \alpha_n$ . Symmetric polynomials are known to have nice properties that lead to efficient

---

#### Algorithm 1 MAP Inference

Find  $\max_q P(q) = \frac{1}{Z} F(q) \prod_{i=1}^n \phi_i(q_i)$

Input:  $c_k = F(q)$ , *s.t.*  $\|q\|_1 = k$ ;  $\alpha_i = \frac{\phi_i(1)}{\phi_i(0)}$

---

```

1:  $(\alpha', \text{order}) \leftarrow \text{sort}(\alpha_1, \dots, \alpha_n)$  in descending order
2:  $\text{proda}' \leftarrow 1$ ;  $\text{maxval} \leftarrow c_0$ ;  $k \leftarrow 0$ 
3: for  $i=1, \dots, n$  do
4:    $\text{proda}' \leftarrow \text{proda}' \times \alpha'_i$ 
5:   if  $c_i \times \text{proda}' > \text{maxval}$  then
6:      $\text{maxval} \leftarrow c_i \times \text{proda}'$ 
7:      $k = i$ 
8:   end if
9: end for
10: return  $q$  where
11:  $q_i = \begin{cases} 1 & \text{if } \text{order}(i) \leq k \\ 0 & \text{otherwise} \end{cases}$ 

```

---

computation.<sup>2</sup> In particular, the fundamental theorem of symmetric polynomials asserts that any symmetric polynomial can be expressed in terms of a small number of building units called elementary symmetric polynomials (Fine and Rosenberger 1997).

**Definition 2.** For  $0 \leq k \leq n$ , the  $k$ -th order elementary symmetric polynomial of  $n$  variables  $\alpha_1 \dots \alpha_n$ , denoted by  $e_k(\alpha_1 \dots \alpha_n)$ , is the sum of all products of distinct  $k$  elements of  $\alpha$

$$e_k(\alpha) = \sum_{1 \leq i_1 < \dots < i_k \leq n} \alpha_{i_1} \dots \alpha_{i_k}$$

when  $k = 0$ ,  $e_0(\alpha) = 1$ .

The expression of  $Z$  in terms of the elementary symmetric polynomials in this case turns out to be very simple as the following theorem shows.

**Theorem 2.**  $Z = \Phi \sum_{k=0}^n c_k e_k(\alpha)$  where  $c_k = F(q)$ ,  $q$  is such that  $\|q\|_1 = k$  (i.e.,  $\{c_k\}$  are the counting representation of  $F$ ).

*Proof.* Starting from  $Z = \Phi \sum_{q_1 \dots q_n} F(q) \prod_{i=1}^n \alpha_i^{q_i}$ , by breaking the summation over all  $q$ 's into  $n+1$  groups where in each group,  $\|q\|_1 = k$ ,  $k = 0 \dots n$  we obtain

$$Z = \Phi \sum_{k=0}^n c_k \left( \sum_{q: \|q\|_1=k} \prod_{i=1}^n \alpha_i^{q_i} \right)$$

Note that  $\sum_{q: \|q\|_1=k} \prod_{i=1}^n \alpha_i^{q_i}$  is the sum of all products of  $k$  distinct elements of  $(\alpha_1 \dots \alpha_n)$ , and this by definition is  $e_k(\alpha)$ .  $\square$

It remains to show that the elementary symmetric polynomials can be computed efficiently. One way to achieve this is via Newton's identity (Mead 1992), which yields a recursive method of computing all elementary symmetric polynomials up to  $n$ -th order in  $O(n^2)$ .

---

<sup>2</sup>(Jha et al. 2010) exploited the fact that  $Z$  is a multivariate polynomial. Our work however also exploits the symmetry of  $Z$ .

---

**Algorithm 2** Computing  $Z$ Compute  $Z = \sum_q P(q) = \sum_q F(q) \prod_{i=1}^n \phi_i(q_i)$ Input:  $c_k = F(q)$ , s.t.  $\|q\|_1 = k$ ;  $\phi_i$ 


---

```

1:  $Z \leftarrow 0$ 
2:  $\Phi \leftarrow 1$ 
3: for  $i = 1, \dots, n$  do
4:    $\alpha_i \leftarrow \phi_i(1)/\phi_i(0)$ 
5:    $\Phi \leftarrow \Phi \times \phi_i(0)$ 
6: end for
7:  $e_0(\alpha) \leftarrow 1$ 
8: for  $k = 1, \dots, n$  do
9:    $p_k(\alpha) \leftarrow 0$ 
10:  for  $i = 1, \dots, n$  do
11:     $p_k(\alpha) \leftarrow p_k(\alpha) + \alpha_i^k$ 
12:  end for
13:   $e_k(\alpha) \leftarrow 0$ 
14:  for  $i = 1, \dots, k$  do
15:     $e_k(\alpha) \leftarrow e_k(\alpha) + (-1)^{i-1} \times e_{k-i}(\alpha) \times p_i(\alpha)$ 
16:  end for
17:   $e_k(\alpha) \leftarrow e_k(\alpha)/k$ 
18:   $Z \leftarrow Z + c_k \times e_k(\alpha)$ 
19: end for
20:  $Z \leftarrow Z \times \Phi$ 
21: return  $Z$ 

```

---

**Proposition 1.** (Newton’s identities) Let  $p_k(\alpha_1 \dots \alpha_n) = \sum_{i=1}^n \alpha_i^k$ . Then

$$e_k(\alpha) = \frac{1}{k} \sum_{i=1}^k (-1)^{i-1} e_{k-i}(\alpha) p_i(\alpha)$$

The overall algorithm for computing the normalization term  $Z$  is given in Algorithm 2. The complexity of this algorithm (not taking into account the complexity of computing  $c_k$ ) is  $O(n^2)$ .

### 3.5 Marginal Inference

As usual, the marginals  $P(q_i)$  can be computed in a way similar to the computation of the normalization term  $Z$ , as the following theorem shows.

**Theorem 3.** Let  $\alpha^*$  be the vector such that  $\alpha_i^* = 0$  and  $\alpha_j^* = \alpha_j$  for every  $j \neq i$ . Then

$$P(q_i = 0) = \frac{\sum_{k=0}^n c_k e_k(\alpha^*)}{\sum_{k=0}^n c_k e_k(\alpha)}$$

*Proof.* Basic manipulation of  $P(q_i = 0)$  gives

$$\begin{aligned} P(q_i = 0) &= \sum_{q_1 \dots q_n} P(q) \delta(q_i = 0) \\ &= \frac{1}{Z} \sum_{q_1 \dots q_n} F(q) \delta(q_i = 0) \prod_{j=1}^n \phi_j(q_j) \\ &= \frac{1}{Z} \sum_{q_1 \dots q_n} F(q) \prod_{j=1}^n \phi_j^*(q_j) \end{aligned}$$

where  $\delta(\cdot)$  is the indicator function and  $\phi_i^*(q_i) = \phi_i(q_i) \delta(q_i = 0)$  and  $\phi_j^*(q_j) = \phi_j(q_j)$  for every  $j \neq i$ .

Observe that the term  $\sum_{q_1 \dots q_n} F(q) \prod_{j=1}^n \phi_j^*(q_j)$  is just another normalization term  $Z^*$  that, by theorem 2,

$$Z^* = \prod_{j=1}^n \phi_j^*(0) \sum_{k=0}^n c_k e_k(\alpha^*) = \Phi \sum_{k=0}^n c_k e_k(\alpha^*)$$

where  $\alpha_j^* = \frac{\phi_j^*(1)}{\phi_j^*(0)} = \alpha_j$  for all  $j \neq i$  and  $\alpha_i^* = 0$ .  $\square$

## 4 Experiments

We ran experiments on the “Friends & Smokes” MLN described earlier. The task is to compute the marginal probability of having cancer of each individual given the cancer screening test results of the entire population as soft evidence. We encoded this soft evidence by introducing a unit clause  $w_i : \text{Cancer}(P_i)$  for each person in the population; each weight  $w_i$  was uniformly sampled from the interval  $[0,2]$ . As discussed above, all existing exact lifted inference methods reduce to ground inference with this type of evidence. Exact inference on the grounded model is intractable in this case due to the high tree-width of the grounded model. In particular, exact junction tree ran out of memory when the number of individuals  $n = 33$ . On the other hand, Lifted BP and Counting BP reduce to ground BP.<sup>3</sup> Thus, we compared the performance of our exact lifting algorithm (LIDE) with Belief Propagation (BP) (Pearl 1988). For LIDE, we used a minor extension of C-FOVE (Milch et al. 2008) to compute the  $c_k$ ’s, which are the unnormalized probabilities of exactly  $k$  people having cancer before taking into account soft evidence. For BP, we used the implementation in libDAI (Mooij 2010) with the following setting: parallel updates, max iterations = 500, convergence threshold = 1e-9, no damping. All the experiments were conducted on a personal computer with 2.7 GHz Intel Core i7 and 8 GB memory.

We first ran an experiment with the original “Friends & Smokes” MLN. The running time of each algorithm is limited to 900 seconds. Figure 1 shows the running times of our exact lifting algorithm and BP when the number of persons is increased. For this MLN, BP surprisingly converged (in less than 10 iterations) to the right marginals but it ran much slower than LIDE. For example, in the case of 800 people, our algorithm took 132 seconds while BP took 643.2 seconds (the time for grounding the network is not included in the BP running times). As a result, within the time limit, BP can run only up to the case of 800 people while LIDE can run up to the case of 1500 people. Memory usage is another bottleneck for BP since storing the ground network takes  $O(n^2)$  space. Since LIDE does not ground the network, it uses only  $O(n)$  space.

The fifth clause in the original “Friends & Smokes” MLN leads to attractive potentials on the pairs  $\text{Smoke}(x), \text{Smoke}(y)$ , which is considered an easy case for BP (Wainwright, Jaakkola, and Willsky 2003). Thus, we conducted another experiment on the modified “Friends & Smokes” MLN where the weight of the fifth clause is negated (i.e., set to  $-1.1$ ), which simulates the

<sup>3</sup>In fact, BP is faster than Lifted or Counting BP since it does not waste time attempting to find a lifted network

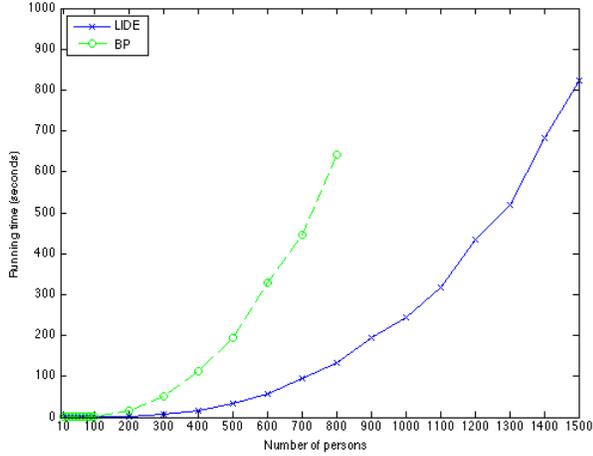


Figure 1: Running time on the original “Friends & Smokes” MLN

*mixed* scenario in (Wainwright, Jaakkola, and Willsky 2003). For this case, we use BP with damping where the damping parameter is set to 0.1. Figure 2 shows the running times of LIDE and BP with damping. On this modified MLN, the running times of LIDE are still the same as those on the original MLN since its computational complexity does not depend on the type of potentials. In contrast, even with damping, BP still requires considerably more iterations to converge, and the number of iterations requires increases as the number of individuals  $n$  grows. BP did not converge when  $n = 200$ .

It is worth noting that computing  $c_k$  (that is, the computation before taking evidence into account) dominates the time taken by our method, taking much longer than the step incorporating evidence. This is significant since this can be seen as a pre-processing step that can be re-used for multiple sets of evidence given a fixed  $M_0$ .

## 5 Conclusion

We have shown how to leverage the symmetry of an FOPM in order to perform lifted inference before symmetry is broken by evidence at the level of each individual in a domain. This is done by using available lifted inference methods on the model and obtaining compact functions on potentially large sets of symmetric random variables, and only then using this function in conjunction with evidence in order to compute posterior probabilities. This is significant because in real applications one typically has an FOPM specified at an abstract level, with much symmetry, and evidence coming from large datasets that break that symmetry; this typical scenario renders lifted inference methods proposed to date ineffective. This paper opens a new line of research in lifted inference in which the symmetric parts of a model are processed separately from the asymmetrical parts. Further directions include investigation on more general forms of evidence, including evidence on multiple unary predicates

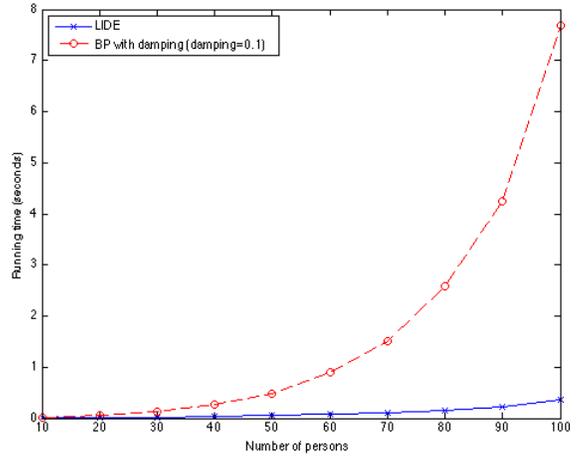


Figure 2: Running time on the modified “Friends & Smokes” MLN

and on non-unary predicates.

## 6 Acknowledgements

We thank the reviewers for their valuable comments that help improve the manuscript. The authors gratefully acknowledge the support of the Defense Advanced Research Projects Agency (DARPA) Machine Reading Program under Air Force Research Laboratory (AFRL) prime contract no. FA8750-09-C-0181. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the view of DARPA, AFRL, or the US government.

Distribution Statement “A” (Approved for Public Release, Distribution Unlimited).

## Appendix

Proof of Theorem 1.

The key element of the proof involves an operation akin to renaming individuals in the domain  $\mathcal{D}$  of the MLN. Let  $\pi$  be a permutation over elements of  $\mathcal{D}$ , that is,  $\pi \in \mathbb{S}(\mathcal{D})$ . The permutation  $\pi$  has a renaming effect in that an individual  $a \in \mathcal{D}$  can now be renamed to  $\pi(a) \in \mathcal{D}$ . Applying this renaming operation to a ground atom  $g = r(b_1, \dots, b_m)$  we obtain  $\pi(g) = r(\pi(b_1), \dots, \pi(b_m))$ . Similarly, for any Herbrand model  $\omega$  MLN,  $\pi(\omega)$  is the set of all ground atoms in  $\omega$  after being renamed by  $\pi$ .

**Lemma 1.** *Let  $F$  be an MLN formula with variables  $x_1, \dots, x_n$  and constants  $b_1, \dots, b_m$ . Denote  $F[t_1 \dots t_n]$  the result of substituting  $t_i$  for  $x_i$  in  $F$ . If  $\pi(b_i) = b_i$  for all constants  $b_i$  then*

$$\omega \models F[a_1 \dots a_n] \iff \omega' \models F[a'_1 \dots a'_m]$$

where  $a_i \in \mathcal{D}_*$ ,  $\omega' = \pi(\omega)$ ,  $a'_i = \pi(a_i)$ .

*Proof.* The proof is straightforward using an inductive argument on the length of  $F$ .

The base case assumes that  $F$  is an atomic formula. In this case  $\omega \models F[a_1 \dots a_n]$  simplifies to  $F[a_1 \dots a_n] \in \omega$ . Renaming  $F[a_1 \dots a_n]$  by  $\pi$ , we obtain  $F[a'_1 \dots a'_m]$  since  $\pi$  does not change any of the constant symbols in  $F$ . Thus, by definition of  $\pi(\omega)$ ,  $F[a_1 \dots a_n] \in \omega$  and  $F[a'_1 \dots a'_m] \in \omega'$  are equivalent. The same argument also holds if  $F$  is a negation of an atomic formula.

The induction argument takes care of logical connectives. As an example let  $F = F_1 \wedge F_2$ ; then, by induction

$$\omega \models F_i[a_1 \dots a_n] \iff \omega' \models F_i[a'_1 \dots a'_n]$$

$$\text{so } \omega \models F_1[a_1 \dots a_n] \wedge F_2[a_1 \dots a_n] \iff \omega' \models F_1[a'_1 \dots a'_n] \wedge F_2[a'_1 \dots a'_n]. \quad \square$$

**Lemma 2.** *Given an MLN  $M_0$ , let  $\pi$  be a renaming permutation that fixes all constants in  $M_0$ . Then, for any Herbrand model  $\omega$*

$$\Pr(\omega \mid M_0) = \Pr(\pi(\omega) \mid M_0)$$

*Proof.* For any formula  $F$  of the MLN  $M_0$ , let  $G(F, \omega)$  be the set of groundings of  $F$  that are in  $\omega$  (that is, the set of true groundings of  $F$ ). It is sufficient to show that  $|G(F, \omega)| = |G(F, \omega')|$  where  $\omega' = \pi(\omega)$ . To do this, we establish a bijective mapping between members of the two sets. Let  $F[a_1 \dots a_n]$  be a true grounding in  $\omega$ . This mapping returns  $F[a'_1 \dots a'_n]$ , which, by the above lemma, is a true grounding in  $\omega'$ . Observe that this mapping is a bijection since its inverse can be obtained via the renaming operation  $\pi^{-1}$ .  $\square$

We now return to the main proof of Theorem 1.

*Proof.* We first encode the random vector  $(q(d_1) \dots q(d_n))$  as a ground formula  $Q = Q_1 \wedge \dots \wedge Q_n$  where  $Q_i = q(d_i)$  if  $q(d_i) = 1$  and  $Q_i = \neg q(d_i)$  if  $q(d_i) = 0$ . Observe that  $P_0(q(d_1) \dots q(d_n)) = \Pr(Q) = \sum_{\omega \models Q} \Pr(\omega)$ . Let  $\pi$  be any renaming permutation that fixes all constants in  $M_0$ . By the above two lemmas,  $\omega \models Q \iff \pi(\omega) \models \pi(Q)$  and  $\Pr(\omega) = \Pr(\pi(\omega))$ . Simple arithmetic then leads to  $\Pr(Q) = \Pr(\pi(Q))$ . Finally, observe that  $\pi(Q)$  is a formula encoding of the vector  $(q(\pi(d_1)) \dots q(\pi(d_n)))$ . Thus, we have proved that  $P_0(q(d_1) \dots q(d_n)) = P_0(q(\pi(d_1)) \dots q(\pi(d_n)))$ . The only requirement for  $\pi$  is that it fixes all constants in  $M_0$ . Thus,  $\pi$  can permute  $d_1 \dots d_n$  in any possible way.  $\square$

## References

de Salvo Braz, R.; Amir, E.; and Roth, D. 2005. Lifted first-order probabilistic inference. In *Proceedings of IJCAI-05, 19th International Joint Conference on Artificial Intelligence*.

Fine, B., and Rosenberger, G. 1997. *The Fundamental Theorem of Algebra*. Springer.

Getoor, L.; Friedman, N.; Koller, D.; and Pfeffer, A. 2001. Learning probabilistic relational models. In Dzeroski, S., and Lavrac, N., eds., *Relational Data Mining*. Springer-Verlag. 307–335.

Gogate, V., and Domingos, P. 2011. Probabilistic theorem proving. In *Proceedings of the Twenty-Seventh Annual Conference on Uncertainty in Artificial Intelligence (UAI-11)*, 256–265. Corvallis, Oregon: AUAI Press.

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

Kersting, K.; Ahmadi, B.; and Natarajan, S. 2009. Counting belief propagation. In A. Ng, J. B., ed., *Proceedings of the 25th Conference on Uncertainty in Artificial Intelligence (UAI-09)*.

Kersting, K.; Massaoudi, Y. E.; Hadji, F.; and Ahmadi, B. 2010. Informed lifting for message-passing. In *AAAI*.

Mead, D. G. 1992. Newton’s identities. *The American Mathematical Monthly* 99(8):pp. 749–751.

Milch, B.; Zettlemoyer, L.; Kersting, K.; Haimes, M.; and Kaelbling, L. P. 2008. Lifted probabilistic inference with counting formulas. In *Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence (AAAI-2008)*, 1062–1068.

Mooij, J. M. 2010. libDAI: A free and open source C++ library for discrete approximate inference in graphical models. *Journal of Machine Learning Research* 11:2169–2173.

Ngo, L., and Haddawy, P. 1995. Probabilistic logic programming and Bayesian networks. In *Asian Computing Science Conference*, 286–300.

Pearl, J. 1988. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, San Mateo (Calif.).

Poole, D. 1993. Probabilistic Horn abduction and Bayesian networks. *Artificial Intelligence* 64(1):81–129.

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

Richardson, M., and Domingos, P. 2006. Markov logic networks. *Machine Learning* 62(1-2):107–136.

Singla, P., and Domingos, P. 2008. Lifted first-order belief propagation. In *Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence (AAAI-2008)*, 1094–1099.

Wainwright, M. J.; Jaakkola, T.; and Willsky, A. S. 2003. Tree-based reparameterization framework for analysis of sum-product and related algorithms. *IEEE Transactions on Information Theory* 49(5):1120–1146.