

Lifted Inference On Transitive Relations

Wen Pu

Department of Computer Science
University of Illinois at Urbana-Champaign
Urbana, IL 61801, USA
wenpu1@illinois.edu

Jaesik Choi

Computational Research Division
Lawrence Berkeley Laboratory
Berkeley, CA, USA
jaesikchoi@lbl.gov

Eyal Amir

Department of Computer Science
University of Illinois at Urbana-Champaign
Urbana, IL, USA
eyal@illinois.edu

Abstract

Lifted inference algorithms are able to boost efficiency through exploiting symmetries of the underlying first-order probabilistic models. Models with transitive relations (e.g., if X and Y are friends and so are Y and Z , then X and Z will likely be friends) are essential in social network analysis. With n elements in a transitive relation model, the computational complexity of exact propositional inference is $O(2^{n(n-1)/2})$, making it intractable for large domains. However, no tractable exact inference on the transitive relations has been reported on the transitive relations. In this paper, we report a novel deterministic approximate lifted inference algorithm, which efficiently solves inference problems on the transitive relations without degenerating input models. We introduce an alternative graph representation for first-order probabilistic models with formulas of homogeneous bivariate predicates. The new representation, which is closely related to exponential-family random graph models, leads to an efficient deterministic approximate lifting algorithm by exploiting the asymptotic properties of the state space. We perform experiments to verify the effectiveness of the proposed algorithm.

1 Introduction

First-order probabilistic representations (e.g., (Nilsson 1986; Poole 2003; Richardson and Domingos 2006; McCallum, Schultz, and Singh 2009)) are powerful tools for combining first-order logic and graphical models, where the former is good at representing complex structural and relational knowledge, the latter shines at capturing uncertainty in the system. Markov logic networks (MLNs) (Richardson and Domingos 2006), for example, are one of the popular representations that fall in this category. In general, a first-order probabilistic representation can be treated as a template for generating graphical models. Applications of these models can be found in natural language processing, social network analysis, etc.

The straight forward approach for carrying out inference on a first-order probabilistic model is to first ground the model into a propositional level graphical model, and then to enlist a typical (either variable elimination or sampling) probabilistic inference algorithm. This naive approach, however, does not take advantage of the compactness of the

first-order representation. Moreover, the propositional procedures are intractable for large grounded probabilistic models.

Lifted inference (Poole 2003; De Salvo Braz, Amir, and Roth 2005; Singla and Domingos 2008) is an endeavor towards lifting the inference from propositional level to the first-order level. In many cases, such lifting can bring a significant gain in performance.

Many existing lifted inference algorithms are based on the idea of grouping exchangeable random variables (Carbonetto et al. 2005; De Salvo Braz, Amir, and Roth 2005; Van den Broeck et al. 2011; Choi and Amir 2012). These algorithms usually require certain degree of isolation among the set of random variables of interests. Consequently, non-trivial interactions will invalidate the lifting strategies (Jaeger and Van den Broeck 2012). One remarkable case is the transitive relation, i.e. $\forall X, Y, Z Fr(X, Y) \wedge Fr(Y, Z) \Rightarrow Fr(X, Z)$, where Fr is short for *Friends*. While very common in many applications, the formula introduces self-join of the predicates with complex constraints, and the exact inference can only be carried out on a fully grounded basis. Several approximate lifted inference techniques have been proposed. The lifted version of (loopy) belief propagation (Singla and Domingos 2008; Kersting, Ahmadi, and Natarajan 2009) construct a compressed model by grouping nodes sending/receiving identical messages. However, the accuracy of loopy belief propagation suffers from the complex interactions between the random variables introduced by the formula, especially when estimating joint distributions. The symmetry of transitive relation in lifted BP may also be broken by introducing a single evidence. (Bhamidi, Bresler, and Sly 2008) shows the mixing time for stochastic sampling methods on transitive models is exponentially slow in low temperature regime. Lifted MCMC (Niepert 2012) exploits the orbital Markov chains to accelerate the sampling process. However its convergence property on transitive models is still unknown.

This paper investigates inference on first-order probabilistic models with transitive relations from a graph perspective. We convert probabilistic logic of homogeneous irreflexive and symmetric bivariate predicates into exponential-family random graph models (ERGMs) (Robins et al. 2007) for deterministic approximate lifted inference. Our deterministic algorithm (based on Edge Count Search, or ECS (Pu, Amir,

and Espelage 2012; Chatterjee and Diaconis 2011)) takes a macroscopic view of lifting: instead of seeking conditional independence with exchangeability, it exploits the concentration of measure in graph space to approximate the equivalent classes of the states that share the same feature vectors. Our method has several benefits over non-deterministic stochastic sampling. One of the benefits is that our algorithm finishes in a specified time given an inference problem without degrading accuracy. We verify the effectiveness and efficiency of the algorithm in the context of social network analysis.

The rest of the paper is organized as follows: Section 2 explains models and notations used in this paper; Section 3 introduces the alternative representation using subgraph features; Section 4 discusses the connection between the new representation and exponential-family random graph models, and reveals the approximate lifted inference algorithm; Section 5 reports experimental results of our approximate algorithm; Section 6 is related work; and Section 7 concludes the paper.

2 Preliminaries

2.1 Markov Logic Networks

An MLN is a set of weighted first-order logic formulas, where each first-order logic formula describes constraints over variables, and the associated weight measures the confidence of each formula. That is, an MLN sentence captures the *degree of belief* (Bacchus 1990; Halpern 1990) of the first-order logic formula.¹

Formally, an MLN is a set of m pairs $(f_i, \theta_i)_{i=1}^m$, where f_i is a First-Order Logic (FOL) formula defined with *constants*, *logic variables* (or *just variables*) and *predicates*, the weight θ_i is a real number (∞ means the formula is a hard constraint). Table 1 presents an example extracted and modified from (Singla and Domingos 2008). Constants represent objects, denoted with lowercase letters (e.g. *alice*, *bob*). Variables are associated with a finite set of constants, denoted using uppercase letters (e.g. X , Y). Predicates (e.g. *Fr*, *Smoke*) apply to constants and variables to form *atoms*, such as $Fr(X, bob)$ and $Smoke(alice)$. A *grounding* of a formula replaces all the variables in the formula with constants. For example, $Smoke(alice) \wedge Fr(alice, bob) \Rightarrow Smoke(bob)$ is a grounding of the third formula in Table 1. In the rest of the paper, we consider only formulas with universal quantifiers, thus may drop the symbol \forall from the context.² An assignment to a grounded atom sets a truth value to the atom (e.g., $Smoke(alice) = \perp$ and $Fr(alice, bob) = \top$).

¹In this paper, we will focus on MLNs due to its simplicity. However, the topics and methods discussed here are also applicable to other first-order probabilistic representations in exponential family under the closed world assumption. An MLN is a set of weighted first-order logic formulas, where each logical formula describes some relational knowledge, and the associated weight measures for our confidence of the formula. Table 1 shows a modified example MLN.

²Regarding to existential quantifiers, there are polynomial time inference algorithms for weighted FOL formulas with typical uses of the existential quantifiers (Kisynski and Poole 2009; Choi, de Salvo Braz, and Bui 2011).

A Markov network derived from an MLN $(f_i, \theta_i)_{i=1}^m$ specifies a probability distribution over all the possible assignments to grounded atoms (i.e. different worlds $w \in \mathcal{W}$):

$$\begin{aligned} p(w; \theta) &= \frac{1}{Z(\theta)} \prod_{i=1}^m \exp(\theta_i N(f_i, w)) \\ &= \frac{1}{Z(\theta)} \exp(\theta^T N(\mathbf{f}, w)) \end{aligned}$$

Here, $N(f_i, w)$ counts the number of groundings of f_i that evaluates to true in world w , $\theta = (\theta_1, \dots, \theta_m)^T$ and $\mathbf{f} = (f_1, \dots, f_m)^T$. $Z(\theta)$ is the normalizing constant or partition function:

$$Z(\theta) = \sum_{w \in \mathcal{W}} \exp(\theta^T N(\mathbf{f}, w)) \quad (1)$$

For example, assume the domain of the MLN in Table 1 is $\{alice, bob, joe, harry\}$, then there are $2^{6+4} = 1024$ different worlds. Notice the fourth and fifth formulas are hard constraints to guarantee the reflexivity and the symmetry for predicate *Fr*. Consider one of the worlds w : *alice* and *bob* smoke; $(alice, bob)$ and $(joe, harry)$ are two pairs of friends. The probability of this world is $p(w) = \frac{1}{Z} \exp(1.4 \times 2 + 4.6 \times 8 + 1.1 \times 12)$. Notice that $Fr(alice, bob)$ and $Fr(bob, alice)$ are different, although under the constraints they always share the same assignment in the same world.

Let g and ψ be a query and an evidence respectively. An inference problem on MLN is to compute the conditional probability $p(g|\psi) = p(g \wedge \psi)/p(\psi)$, which turns into the sum-of-product computation problem. If $\psi = \emptyset$, then $p(g|\psi) = p(g)$ is the marginal of g . In this paper, we focus on the cases that $\psi = \emptyset$ and g is a full assignment to the grounded atoms. Inference with full assignments is useful for learning tasks with fully observed training data:

$$\theta^* = \operatorname{argmax}_{\theta} L(\theta|g) = \operatorname{argmax}_{\theta} \{\theta^T N(\mathbf{f}, g) - \ln Z(\theta)\}$$

In this case, evaluation of $\ln Z(\theta)$ is the main obstacle for inference because the computation may require to evaluate over all the possible assignments \mathcal{W} . To overcome this problem, our linear transformation (presented in Section 2.3) converts an MLN into a Exponential-family Random Graph model (ERGM) is explained in the next section.

2.2 Exponential-family Random Graph Models

An ERGM defines an exponential family distribution over an order- n simple graphs, in which sufficient statistics are subgraph statistics. ERGMs have been actively studied in the context of social network analysis (Handcock et al. 2003), statistics (Hunter, Krivitsky, and Schweinberger 2012) and machine learning (Wyatt, Choudhury, and Bilmes 2008; Guo et al. 2007; Pu, Amir, and Espelage 2012).

Notation-wise, a graph $G = (V, E)$ consists of a set of vertices (or nodes V) and a set of edges E ; $|V|$ and $|E|$ are G 's *order* and *size*. For convenience, we use $v(G)$ and $e(G)$ are respectively the order and the size of the graph G . A graph $H = (V', E')$ is a *subgraph* of G if $V' \subseteq V$ and $E' \subseteq E$. Moreover, H is an *induced subgraph* of G if every pair of V' is connected iff they are connected in G . Two

Table 1: An example MLN

	Feature	Weight	Explanation
1	$\forall X \neg \text{Smoke}(X)$	1.4	Most people don't smoke
2	$\forall X \forall Y \neg \text{Fr}(X, Y)$	4.6	Most people are not friends
3	$\forall X \forall Y \text{Smoke}(X) \wedge \text{Fr}(X, Y) \Rightarrow \text{Smoke}(Y)$	1.1	Friends of people who smoke are likely to smoke
4	$\forall X \text{Fr}(X, X)$	$-\infty$	Anti-reflexive relation constraint
5	$\forall X \forall Y \neg (\text{Fr}(X, Y) \Leftrightarrow \text{Fr}(Y, X))$	$-\infty$	Symmetric relation constraint

graphs $G = (V, E)$ and $H = (V', E')$ are *isomorphic* if there is an one-to-one mapping Ψ from V to V' , such that $(u, v) \in E$ iff $(\Psi(u), \Psi(v)) \in E'$. K_n is the complete order- n graph with $v(K_n) = n$ and $e(K_n) = \binom{n}{2}$. For graph G and H , we define function $t(G, H)$ that counts the number of subgraphs in G that are isomorphic to H .

Formally, let $\{H_1, H_2, \dots, H_r\}$ be the set of subgraph structures of interests and θ' be a r -dimensional vector (or a model coefficient), an ERGM defines a probability for a graph $g \in \mathcal{G}$:

$$p(g; \theta) = \frac{1}{Z'(\theta')} \exp(\theta'^T \phi(g))$$

Here $Z'(\theta')$ is the normalizing constant similar to (1), summing over \mathcal{G} ; $\phi(g)$ is a vector of subgraph statistics of H_1, \dots, H_r , such as density:

$$\phi(g) = \left(\frac{t(g, H_1)}{t(K_n, H_1)}, \frac{t(g, H_2)}{t(K_n, H_2)}, \dots, \frac{t(g, H_r)}{t(K_n, H_r)} \right)$$

Here, $t(K_n, H_i) = \binom{n}{v(H_i)} t(K_{v(H_i)}, H_i)$ is the maximum possible subgraph counts for H_i , and can be treated as a constant if n is fixed. Later we will show that $N(\mathbf{f}, w)$ can be translated into $\phi(g)$ for a certain subset of MLNs.

2.3 Generalizing Counting Elimination

Lifted inference algorithms (e.g., (De Salvo Braz, Amir, and Roth 2005; Milch et al. 2008; Singla and Domingos 2008)) aim to scale up the computation up from propositional level as much as possible to expedite the inference procedures. Those methods avoid redundant computations by exploiting the symmetry or indistinguishable objects of the input models. A popular strategy for exact inference is to introduce a *counting function* for each group of redundant computation, and it has been successfully applied in many lifted inference algorithms, such as (De Salvo Braz, Amir, and Roth 2005; Milch et al. 2008; Singla and Domingos 2008). In this paper, we generalize this strategy for lifting the computation of $\log Z(\theta)$.

For the problems in the form of (1), a counting-function-based approach has the following generalized form:

$$\sum_{w \in \mathcal{W}} \exp(\theta^T N(\mathbf{f}, w)) = \sum_{\mathbf{h} \in \mathcal{H}_{\mathbf{f}}} \#(\mathbf{h}) \exp(\theta^T \mathbf{h}), \quad (2)$$

Where $\mathcal{H}_{\mathbf{f}} = N(\mathbf{f}, \mathcal{W})$ is the domain of sufficient statistics, and $\#(\mathbf{h}) = |\{w | N(\mathbf{f}, w) = \mathbf{h}, w \in \mathcal{W}\}|$ is the counting function that counts the number of (redundant) configurations of a sufficient statistic \mathbf{h} . For first-order probabilistic models, $\mathcal{H}_{\mathbf{f}}$ is usually significantly smaller, thus easier

to enumerate) than \mathcal{W} . Thus, the sufficient statistic and the counting function leads to a significant speedup. One of the key issues is to be able to identify $\mathcal{H}_{\mathbf{f}}$ and evaluate $\#(\mathbf{h})$ efficiently.

For example, consider the following simple example with three MLN sentences on domain \mathcal{D} (De Salvo Braz, Amir, and Roth 2005):

$$\begin{aligned} f_1 &: \text{Pred}(X) \wedge \text{Pred}(Y) & \theta_1 \\ f_2 &: \text{Pred}(X) \wedge \neg \text{Pred}(Y) & \theta_2 \\ f_3 &: \neg \text{Pred}(X) \wedge \neg \text{Pred}(Y) & \theta_3 \end{aligned}$$

The lifted computation of the partition function:

$$\begin{aligned} & \sum_{w \in \mathcal{W}} \exp(\theta_1 N(f_1, w) + \theta_2 N(f_2, w) + \theta_3 N(f_3, w)) \\ &= \sum_{\substack{c_0, c_1 \text{ s.t.} \\ c_0 + c_1 = |\mathcal{D}|}} \binom{|\mathcal{D}|}{c_0} \exp(\theta_1 c_1^2 + \theta_2 c_1 c_0 + \theta_3 c_0^2) \end{aligned}$$

Here, c_0 and c_1 are counting parameters for the number of 0 and 1 in grounded $\text{Pred}(X)$, they subject to the constraint $c_0 + c_1 = |\mathcal{D}|$. All grounded $\text{Pred}(X)$ with the same assignment are exchangeable or interchangeable. Therefore, we can identify $\mathcal{H}_{\mathbf{f}}$ very easily given a counting argument by using the counting arguments, together with the counting function:

$$\mathbf{h} = (c_1^2, c_1 c_0, c_0^2)^T, \quad \#(\mathbf{h}) = \binom{|\mathcal{D}|}{c_0}$$

Because $|\mathcal{H}_{\mathbf{f}}| = |\mathcal{D}|$ while $|\mathcal{W}| = 2^{|\mathcal{D}|}$, this method brings an exponential speedup. However, in general, it is non-trivial to apply this strategy to intermingled variables without extensive grounding or expending the model.

3 Graph Interpretation for Markov Logic

In this section, we introduce a graph interpretation for MLNs, and show its relationship with ERGMs.

Definition 1 (Homogeneous Bivariate MLNs). *A homogeneous Bivariate MLN consist of which logical formulas that are represented by only bivariate predicates of the same type.*

Table 2 shows a homogeneous irreflexive and asymmetric bivariate MLN with transitive relation. Let objects in the domain \mathcal{D} be nodes, and there is an edge between a and b if the ground atom $\text{Fr}(a, b) = 1$, then under the irreflexive relation and symmetric relation constraints (formula 3 and 4 in Table 2), each assignment $w \in \mathcal{W}$ can be treated

Table 2: MLN with transitive relations

	Feature	Weight
f_1	$\neg Fr(X, Y)$	θ_1
f_2	$Fr(X, Y) \wedge Fr(Y, Z) \Rightarrow Fr(X, Z)$	θ_2
f_3	$Fr(X, X)$	$-\infty$
f_4	$\neg(Fr(X, Y) \Leftrightarrow Fr(Y, X))$	$-\infty$

as a unique undirected graph, with order $n = |\mathcal{D}|$. Without ambiguity, we will refer to w as an assignment and a graph interchangeably for the rest of the paper. Accordingly, $N(\mathbf{f}, w)$ also shares a graph interpretation for labeled graph w . It is also possible to map other more complex MLNs to graph representation with the help of dummy nodes and colored edges, but we will focus on this simplest form for the convenience of discussion.

From a graph perspective, it is easy to see the complex interaction introduced by the transitive relation: all the edges in the graph are random variables, and every edge is correlated with its neighbor through the formula; the distance between any two edges is less than two; each edge falls in many loops. For the rest of this section, we will show how to translate the formula counting $N(\mathbf{f}, w)$ into a linear combination of subgraph statistics, and consequently build the connection between MLNs and ERGMs.

3.1 Subgraph Features \mathbf{H}^k

Given a homogeneous bivariate MLN Formula with k logical variable, *subgraph features* \mathbf{H}^k of the formula are graphs in vector form $\mathbf{H}^k = (H_0^k, \dots, H_{r_k}^k)^T$. Here, each H_i^k is a graph of order- k , and for any $0 \leq i, j \leq r_k$, H_i^k and H_j^k are not isomorphic. Let $H_i^k \preceq H_j^k$ if H_i^k is isomorphic to some subgraph of H_j^k , then \preceq defines a partial order on $\{H_i^k\}$. Notice that the subgraph features are in the same form as the subgraphs used to define ERGM in Section 2.2.

We explain how to translate MLNs into ERGMs using subgraph features with the MLN example in Table 2.

Example 2 $f_1 : \neg Fr(X, Y)$. The subgraph features of f_1 is $\mathbf{H}^2 = (H_0^2, H_1^2)^T$, because f_1 has two logical variables: X and Y . Here, H_0^2 is an empty order-2 graph and H_1^2 has two nodes connected by an edge, or K_2 .

$$H_0^2: \bullet \quad \bullet \quad H_1^2: \bullet \text{---} \bullet$$

Here, $N(f_1, H_0^2) = 2$ and $N(f_1, H_1^2) = 0$.³ Given any graph w , we know $t(w, H_0^2) = \binom{n}{2}$, and $t(w, H_1^2)$ is the number of edges in w . Because $H_0^2 \preceq H_1^2$, the counting $t(w, H_0^2)$ includes $t(w, H_1^2)$, therefore we need to exclude the latter when computing their contribution to formula counting. More specifically, $N(f_1, w)$ can be represented as:

$$\begin{aligned} N(f_1, w) &= N(f_1, H_0^2) (t(w, H_0^2) - t(w, H_1^2)) \\ &\quad + N(f_1, H_1^2) t(w, H_1^2) \\ &= 2 \binom{n}{2} - 2t(w, H_1^2) \end{aligned}$$

³Note that \mathbf{H}^k and $N(f_i, \mathbf{H}^k)$ are only computed once regardless of w .

That is, $N(f_1, w)$ counts two times the number of non-connected pairs in w .

Example 3 $f_2 : Fr(X, Y) \wedge Fr(Y, Z) \Rightarrow Fr(X, Z)$. The translation of formula f_2 requires more considerations because it has three logical variables X, Y and Z . In this case, the subgraph features are $\mathbf{H}^3 = (H_0^3, H_1^3, H_2^3, H_3^3)^T$, as illustrated below:

$$H_0^3: \bullet \quad \bullet \quad H_1^3: \bullet \text{---} \bullet \quad H_2^3: \bullet \text{---} \bullet \quad H_3^3: \bullet \text{---} \bullet \text{---} \bullet$$

H_0^3, H_1^3 and H_3^3 satisfies f_2 regardless of assignments from logical variables to nodes. Thus, by considering permutations, $\pi(f_2, H_0^3) = \pi(f_2, H_1^3) = \pi(f_2, H_3^3) = 6$. H_2^3 do not satisfy f_2 when there is no edge between X, Z (i.e., $\neg Fr(X, Z)$). Thus, considering two such cases, $\pi(f_2, H_2^3) = 4$. The shorthand representation is as follows,

$$\pi(f_2, \mathbf{H}^3) = (6, 6, 4, 6)^T$$

where $\mathbf{H}^3 = (H_0^3, H_1^3, H_2^3, H_3^3)^T$.

Now suppose that we induce all order-3 subgraphs W^3 from the graph w , i.e., $|W^3| = \binom{n}{3}$. Let $W_j^3 \subset W^3$ be the subset of induced subgraphs which are isomorphic to H_j^3 . Then, $N(f_2, w)$ can be represented as follows,

$$N(f_2, w) = \sum_{x \in W^3} N(f_2, x) = \sum_{0 \leq j \leq 3} \pi(f_2, H_j^3) |W_j^3|$$

In principle, this step is the generalized counting (Section 2.3). Here, the main computational task is to compute $|W_j^3|$ for all j . We make a step further to represent $|W_j^3|$ by using subgraph features \mathbf{H}^k and graph isomorphic counts $t(w, H^3)$.

$|W_3^3| = t(w, H_3^3)$ is trivial because they both count the number of triangles in w . To compute $|W_2^3|$, we need to exclude the counts of W_3^3 from $t(w, H_2^3)$ since $H_2^3 \preceq H_3^3$ and each occurrence of H_3^3 is counted as $t(H_3^3, H_2^3) = 3$ times subgraphs occurrence of H_2^3 . Therefore $|W_2^3| = t(w, H_2^3) - 3|W_3^3|$. Similarly, we have $|W_1^3| = t(w, H_1^3) - 2|W_2^3| - 3|W_3^3|$ and $|W_0^3| = t(w, H_0^3) - |W_1^3| - |W_2^3| - |W_3^3|$. Put everything together we get:

$$\begin{pmatrix} |W_0^3| \\ |W_1^3| \\ |W_2^3| \\ |W_3^3| \end{pmatrix} = \begin{pmatrix} 1 & -1 & 1 & -1 \\ 0 & 1 & -2 & 3 \\ 0 & 0 & 1 & -3 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} t(w, H_0^3) \\ t(w, H_1^3) \\ t(w, H_2^3) \\ t(w, H_3^3) \end{pmatrix}$$

Note that,

$$\begin{aligned} N(f_2, w) &= \sum_{0 \leq j \leq 3} N(f_2, H_j^3) |W_j^3| \\ &= (N(f_2, \mathbf{H}^3)^T \mathbf{A}) t(w, \mathbf{H}^3) \\ &= (6, 0, -2, 6) t(w, \mathbf{H}^3) \\ &= 6t(w, H_0^3) - 2t(w, H_2^3) + 6t(w, H_3^3) \end{aligned}$$

Here, matrix \mathbf{A} is:

$$\mathbf{A} = \begin{pmatrix} 1 & -1 & 1 & -1 \\ 0 & 1 & -2 & 3 \\ 0 & 0 & 1 & -3 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Because $t(w, H_0^3) = \binom{n}{3}$ is a constant, we can re-write $Z(\theta)$ for the MLN in Table 2 as

$$\begin{aligned} Z(\theta) &= \exp\left(2\binom{n}{2}\theta_1 + 6\binom{n}{3}\theta_2\right) \\ &\sum_{w \in \mathcal{W}} \exp\left(-2\theta_1 t(w, H_1^2) - 2\theta_2 t(w, H_2^3) - 6\theta_2 t(w, H_3^3)\right) \\ &= c(\theta) \sum_{w \in \mathcal{W}} \exp\left(\theta'^T t(w, \mathbf{f}_g)\right) \end{aligned} \quad (3)$$

Here $\theta' = (-2\theta_1, -2\theta_2, -6\theta_2)^T$ and $\mathbf{f}_g = (H_1^2, H_2^3, H_3^3)^T$. Notice (3) has the same form as (1).

In general, any MLN formula of homogeneous bivariate predicates can be represented using subgraph features:

Lemma 1. *Given MLN formula f of homogeneous bivariate predicates with k logical variables, and assignment w , $N(f, w)$ can be represented using subgraph features $\mathbf{H}^k = (H_0^k, \dots, H_{r_k}^k)^T$ as*

$$N(f, w) = N(f, \mathbf{H}^k) \mathbf{A} t(w, \mathbf{H}^k)$$

where matrix \mathbf{A} is defined recursively:

$$\mathbf{A}_{ij} = \begin{cases} 1 & i = j \\ -\sum_{l \neq i} t(H_l^k, H_i^k) \mathbf{A}_{lj} & \text{otherwise} \end{cases} \quad (4)$$

Notice that $t(H_l^k, H_i^k) \neq 0$ iff $H_i^k \preceq H_l^k$, and matrix \mathbf{A} can be computed efficiently using our dynamic programming as shown in Algorithm 1.⁴

3.2 Relationship with Triangle-Free Graph Enumeration

Re-writing (3) with (2) we can get:

$$c(\theta) \sum_{w \in \mathcal{W}} \exp\left(\theta'^T t(w, \mathbf{f}_g)\right) = c(\theta) \sum_{\mathbf{h} \in \mathcal{H}_{\mathbf{f}_g}} \#(\mathbf{h}) \exp\left(\theta'^T \mathbf{h}\right)$$

In this case, $\#(\mathbf{h})$ enumerates all the graphs with the given subgraph configuration \mathbf{h} . The lifting can be achieved by solving this graph enumeration problem. However, there is an easy polynomial time reduction from triangle-free graph enumeration problem. Let h_3 be the count of triangle subgraphs, then

$$\# \text{ of order-}n \text{ triangle-free graphs} = \sum_{\mathbf{h} \in \mathcal{H}_{\mathbf{f}_g}} \#(\mathbf{h}) \delta(h_3 = 0)$$

Here $\delta(h_3 = 0) = 1$ if $h_3 = 0$, otherwise 0. The reduction suggests the lifting is at least as difficult as enumerating triangle-free graphs of n vertices. Unfortunately, there is no known formula or efficient algorithm for the latter problem, and the few known terms for $n \leq 17$ were generated using exhaustive enumeration methods (Sloane and Plouffe 1995; Sloane 2013).

On the other hand, the results on asymptotic enumeration of triangle-free graphs have a long history (Erdős, Kleitman, and Rothschild 1976). The approximations usually rely on asymptotic properties in random graphs and give accurate estimations for large n .

⁴Section 4.1 provides the computational complexity of the dynamic programming algorithm.

4 Approximate Lifting Algorithm

Our ‘‘Approximate Lifting’’ algorithm has two parts: (2) converting an input MLN to an ERGM and (2) applying Edge Count Search (ECS) approximation (Pu, Amir, and Espelage 2012) on the ERGM. The conversion part is based on the fact that Equation (3) of MLN is proportional to the partition function of the ERGMs with $\phi(\mathbf{f}_g)$ as sufficient statistics and $\mathbf{B}\theta'$ as parameters where $\mathbf{B} = \text{diag}(t(K_n, H_1^2), t(K_n, H_2^3), t(K_n, H_3^3))$. The computation of \mathbf{B} and θ' is as shown in Section 3.1.

After the conversion procedure is done, our algorithm computes the log partition function $\ln Z(\theta)$ using the ECS approximation algorithm (Pu, Amir, and Espelage 2012). In ECS approximation, the inference is lifted through applying the generalized counting strategy in Section 2.3. Instead of seeking conditional independence, the algorithm exploits the asymptotic property of subgraph statistics (Nowicki 1989) to approximate the equivalent classes of the states that share the same feature vectors. More specifically, let $\mathcal{H}_\phi = \{\phi(\mathcal{G})\}$ be the space of subgraph densities, we apply the generalized counting function strategy to get:

$$\begin{aligned} \ln Z(\theta) &= \ln \sum_{\mathbf{h} \in \mathcal{H}_\phi} \#(\mathbf{h}) \exp\left(\theta'^T \mathbf{h}\right) \\ &= \ln \sum_{\mathbf{h} \in \mathcal{H}_\phi} \exp\left(\theta'^T \mathbf{h} + \ln \#(\mathbf{h})\right) \\ &\simeq \max_{\mathbf{h} \in \mathcal{H}_\phi} \{\theta'^T \mathbf{h} + \ln \#(\mathbf{h})\} + O(\ln |\mathcal{H}_\phi|) \end{aligned} \quad (5)$$

Here, the first term of (5) is a good approximation with a relatively small number of low order subgraph features because $|\mathcal{H}|$ is in $O(\text{poly}(n))$ (Pu, Amir, and Espelage 2012). Let $\mathcal{G}_u \subset \mathcal{G}$ be the set of order- n graphs with u edges. Confining the counting $\#(\mathbf{h})$ and feature space \mathcal{H}_ϕ within \mathcal{G}_u leads to the following lower bound to (5):

$$\gamma(\theta, u) = \max_{\mathbf{h} \in \mathcal{H}_{\phi, u}} \{\theta'^T \mathbf{h} + \ln \#_u(\mathbf{h})\} \quad (6)$$

The ECS approximation exploits the property that as $n \rightarrow \infty$, subgraph statistics of graphs in \mathcal{G}_u tends to concentration around a single configuration in \mathcal{H}_ϕ , and proposes the following approximation of (6):

$$\tilde{\gamma}(\theta, u) = \theta'^T \rho(u) + \binom{n}{2} H(u / \binom{n}{2})$$

Where $H(x) = -x \ln x - (1-x) \ln(1-x)$, $\rho(u) = (\rho_1(u), \dots, \rho_r(u))^T$ with $\rho_i(u) = (u / \binom{n}{2})^{e(H_i)}$, and u be an edge count between 0 and $\binom{n}{2}$. Given parameter θ' , the variational approximation $\text{ECS}(\theta, \mathbf{H})$ simply picks u that returns the largest lower bound as an estimator of the log partition function.

When the ECS approximation is embedded, Algorithm 1 shows the complete approximate lifting algorithm. Notice that $N(f_i, w)$ and $t(w, \mathbf{H}^k)$ in the algorithm are both functions of w .

4.1 Computational Complexity

During the conversion from an MLN to an ERGM, the algorithm generates subgraph features for a formula with k

Algorithm 1 Approximate Lifting

Input: Homogeneous bivariate MLN $(f_i, \theta_i)_{i=1}^m$ (hard constraints excluded)

Output: an approximation of $\ln Z(\theta)$

for $i = 1 \rightarrow m$ **do**

 For f_i with k logical variables, generate subgraph features \mathbf{H}^k .

 Compute \mathbf{A} by (4) using \mathbf{H}^k .

$N(f_i, w) \leftarrow N(f_i, \mathbf{H}^k) \mathbf{A} t(w, \mathbf{H}^k)$

end for

Substitute $N(\mathbf{f}, w)$ in $\ln Z(\theta)$ with subgraph features \mathbf{H}^k and sum terms with the same subgraphs as in (3).

$c(\theta) \leftarrow$ constant terms.

$\mathbf{H} \leftarrow$ subgraph features w/ nonzero coefficients ($\mathbf{H} \subset \mathbf{H}^k$).

$\theta' \leftarrow$ coefficients of \mathbf{H} .

$\mathbf{B} \leftarrow \text{diag}(t(K_n, H_0), \dots, t(K_n, H_{r_k}))$

$M \leftarrow \text{ECS}(\mathbf{B}\theta', \mathbf{H})$ ▷ Call ECS approximation

return $c(\theta)M$

model in Table 2 to generate social networks from a 16×16 grid of parameters ranging between $(-5.0, -5.0)$ and $(5.0, 5.0)$. Networks of different scales are generated, with n be 10, 20, 30, 40, 60, 80, 100, 120, 140 and 160. Notice that the generated parameters for different n are properly scaled (with B as defined in Section 4) to factor out the impacts of different n .

The exact value log partition function is unknown. Because bridge sampling (Gelman and Meng 1998) provides an unbiased estimator of the log-likelihood for ERGMs, we resort to run the samplers extensively for close estimations of the real log-likelihood as our target. For each n , we estimate the log-likelihood for all the parameters using both approximate lifting and MCMC sampling. We exclude the potential degenerated models (i.e. the empty graph or complete graph is dominating in the model), and compute the relative difference for each pair of estimations: $|(\text{LLK}_{\text{lift}} - \text{LLK}_{\text{mcmc}}) / \text{LLK}_{\text{mcmc}}|$. Figure 1 reports the average relative differences and the variance for each n . We can observe that the average error and variance are almost negligible for $n \geq 40$.

6 Related Work

Lifted inference can scale up statistical inference in first-order probabilistic models such as First-Order Probabilistic Models (Poole 2003), Bayesian Logic (Milch et al. 2008), MLN (Richardson and Domingos 2006) and FACTORIE (McCallum, Schultz, and Singh 2009). Lifted (normally polynomial-time) inference algorithms exploit exchangeability of random variables for variable eliminations (e.g., (Poole 2003; De Salvo Braz, Amir, and Roth 2005; Milch et al. 2008)), message passing (e.g., `singla2008lifted`) and variational inference (e.g., (Carbonetto et al. 2005; Choi and Amir 2012)). Recently, (Domingos and Webb 2012) presents a tractable class of first-order probabilistic models with a domain hierarchy.

Models with transitivity relations (e.g., the Smoke and Friendships problem in MLNs) are active research problems in lifted inference (e.g., (Bui, Huynh, and Riedel 2012; Niepert 2012; Gogate, Jha, and Venugopal 2012; Apse and Brafman 2011)) and social networks (e.g., (Kemp et al. 2006; Pu, Amir, and Espelage 2012; Hunter, Krivitsky, and Schweinberger 2012)). Transitivity relation is common in social network models. However, no tractable exact inference algorithm has been reported (Jaeger and Van den Broeck 2012). Existing approximate lifted inference algorithms that applies to transitive relation includes lifted belief propagation (Singla and Domingos 2008; Kersting, Ahmadi, and Natarajan 2009) and lifted MCMC (Niepert 2012). However, the accuracy of belief propagation algorithms suffer from the complex interactions introduced by the transitive relation, especially on estimation of joint distribution, which is important in learning tasks. The symmetry lifted BP relies on also breaks after introducing a single evidence. (Bhamidi, Bresler, and Sly 2008) shows the mixing time for stochastic sampling on transitive relations can be exponentially slow. Although lifted MCMC use orbital Markov chain to accelerate the sampling, its convergence property on transitive relations is still unknown. The approximate lifting al-

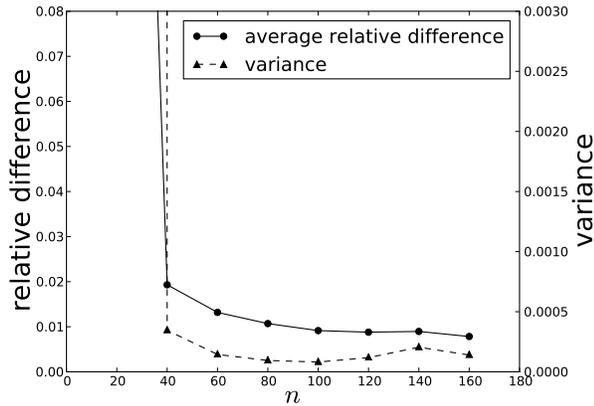


Figure 1: Accuracy of approximate lifting

logical variables in $O(2^{k(k-1)/2})$. Fortunately, k is usually very small (e.g. $k = 3$ for transitive relation) and the subgraph features can be pre-computed, so does $t(H_l^k, H_i^k)$ for all l and i . For each formula, computing \mathbf{A} with dynamic programming can be done in $O(|P_k|^3)$, where $|P_k|$ is the number of feature subgraphs in \mathbf{H}^k .⁵

Assuming the number of generated subgraph features in \mathbf{H} is small, then ECS approximation can be computed in $O(n^2)$, i.e., linear in the size of domain n , and square in the number of random variables, e.g., $Fr(X, Y)$.

5 Experiments

We evaluate the accuracy of the approximation in the task of log-likelihood estimation on synthetic data. We use the

⁵Formula with a large k is not the scope of this paper. In this case, $|P_k|$ is approximately $O(2^{k(k-1)/2}/k!)$ (Harary and Palmer 1973).

gorithm proposed in this paper does not depend on message passing or stochastic sampling, therefore immune to the shortcomings.

The generalized counting function strategy used in this paper is highly related to the concept of “lumping” in lifted MCMC (Niepert 2012). Both techniques try to identify equivalent classes in the state space in theory, while both are not applicable directly in general. (Niepert 2012) uses graph automorphism as an alternative to generate orbital Markov chains instead of lumping, while (Pu, Amir, and Espelage 2012) and this work resort to approximate solution by exploiting the asymptotic properties of the underlying state space.

7 Conclusions

In this paper, we report a novel deterministic approximate lifted inference algorithm for transitive relations. This paper builds a new connection between homogeneous bivariate MLNs and ERGMs. When comparing to exact inference algorithm, our algorithm brings exponential speedups with a reasonable accuracy.

Inference in ERGMs has been widely studied in the context of social network analysis and statistics. Therefore building the connection between the two models also helps to better understand transitive relations and first-order probabilistic models in general.

For future work, we would like to generalize our algorithm to handle non-symmetric relations and more general MLNs.

8 Acknowledgements

The work of Pu and Amir is supported by NSF EAR grant 09-43627, IIS grant 09-17123, IIS grant 09-68552, and a DARPAR grant as part of the Machine Reading Program under AFRL prime contract no. FA8750-09-C-0181. The work of Choi is supported by the Office of Advanced Scientific Computing Research, Office of Science, of the U.S. Department of Energy under Contract No. DE-AC02-05CH11231.

References

- Apfel, U., and Brafman, R. I. 2011. Extended lifted inference with joint formulas. In *UAI*, 11–18.
- Bacchus, F. 1990. *Representing and reasoning with probabilistic knowledge: a logical approach to probabilities*. Cambridge, MA, USA: MIT Press.
- Bhamidi, S.; Bresler, G.; and Sly, A. 2008. Mixing time of exponential random graphs. In *FOCS*, 803–812.
- Bui, H.; Huynh, T.; and Riedel, S. 2012. Automorphism groups of graphical models and lifted variational inference. In *UAI workshop Statistical Relational AI (StaRAI)*.
- Carbonetto, P.; Kisynski, J.; de Freitas, N.; and Poole, D. 2005. Nonparametric bayesian logic. In *UAI*, 85–93.
- Chatterjee, S., and Diaconis, P. 2011. Estimating and understanding exponential random graph models. *Arxiv preprint arxiv:1102.2650*.
- Choi, J., and Amir, E. 2012. Lifted relational variational inference. In *UAI*, 196–206.
- Choi, J.; de Salvo Braz, R.; and Bui, H. H. 2011. Efficient methods for lifted inference with aggregate factors. In *AAAI*, 1030–1036.
- De Salvo Braz, R.; Amir, E.; and Roth, D. 2005. Lifted first-order probabilistic inference. In *IJCAI*, 1319–1325.
- Domingos, P., and Webb, W. A. 2012. A tractable first-order probabilistic logic. In *AAAI*.
- Erdős, P.; Kleitman, D. J.; and Rothschild, B. L. 1976. Asymptotic enumeration of k_n -free graphs. In *Colloquio Internazionale sulle Teorie Combinatorie, (Rome, 1973), Tomo II, Atti dei Convegni Lincei, No. 17*, 19–27. Accad. Naz. Lincei, Rome.
- Gelman, A., and Meng, X. 1998. Simulating normalizing constants: From importance sampling to bridge sampling to path sampling. *Statistical Science* 163–185.
- Gogate, V.; Jha, A. K.; and Venugopal, D. 2012. Advances in lifted importance sampling. In *AAAI*.
- Guo, F.; Hanneke, S.; Fu, W.; and Xing, E. P. 2007. Recovering temporally rewiring networks: A model-based approach. In *Proceedings of the 24th international conference on Machine learning*, 321–328. ACM.
- Halpern, J. Y. 1990. An analysis of first-order logics of probability. *Artificial Intelligence* 46:311–350.
- Handcock, M. S.; Hunter, D. R.; Butts, C. T.; Goodreau, S. M.; and Morris, M. 2003. *statnet: Software tools for the Statistical Modeling of Network Data*. Seattle, WA. Version 2.0.
- Harary, F., and Palmer, E. M. 1973. *Graphical Enumeration*. Academic Press.
- Hunter, D. R.; Krivitsky, P. N.; and Schweinberger, M. 2012. Computational statistical methods for social network models. *Journal of Computational and Graphical Statistics* 21(4):856–882.
- Jaeger, M., and Van den Broeck, G. 2012. Liftability of probabilistic inference: Upper and lower bounds. In *UAI workshop Statistical Relational AI (StaRAI)*.
- Kemp, C.; Tenenbaum, J. B.; Griffiths, T. L.; Yamada, T.; and Ueda, N. 2006. Learning systems of concepts with an infinite relational model. In *AAAI*, 381–388.
- Kersting, K.; Ahmadi, B.; and Natarajan, S. 2009. Counting belief propagation. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, 277–284. AUAI Press.
- Kisynski, J., and Poole, D. 2009. Lifted aggregation in directed first-order probabilistic models. In *IJCAI*, 1922–1929.
- McCallum, A.; Schultz, K.; and Singh, S. 2009. Factorie: Probabilistic programming via imperatively defined factor graphs. In *NIPS*, 1249–1257.
- Milch, B.; Zettlemoyer, L.; Kersting, K.; Haimes, M.; and Kaelbling, L. 2008. Lifted probabilistic inference with counting formulas. In *AAAI*, 1062–1068.
- Niepert, M. 2012. Markov chains on orbits of permutation groups. In *UAI*, 624–633.

- Nilsson, N. J. 1986. Probabilistic logic. *Artif. Intell.* 28(1):71–87.
- Nowicki, K. 1989. Asymptotic normality of graph statistics. *Journal of statistical planning and inference* 21(2):209–222.
- Poole, D. 2003. First-order probabilistic inference. In *IJCAI*, 985–991.
- Pu, W.; Amir, E.; and Espelage, D. 2012. Approximate partition functions for exponential-family random graph models. *NIPS Workshop on Algorithmic and Statistical Approaches for Large Social Networks*.
- Richardson, M., and Domingos, P. 2006. Markov logic networks. *Machine Learning* 62(1-2):107–136.
- Robins, G.; Pattison, P.; Kalish, Y.; and Lusher, D. 2007. An introduction to exponential random graph (p^*) models for social networks. *Social Networks* 29(2):173–191.
- Singla, P., and Domingos, P. 2008. Lifted first-order belief propagation. In *AAAI*, volume 2, 1094–1099.
- Sloane, N. J. A., and Plouffe, S. 1995. *The Encyclopedia of Integer Sequences*. Academic Press.
- Sloane, N. J. A. 2013. The On-Line Encyclopedia of Integer Sequences. <http://oeis.org/A006785>. Triangle-free graphs on n vertices.
- Van den Broeck, G.; Taghipour, N.; Meert, W.; Davis, J.; and De Raedt, L. 2011. Lifted probabilistic inference by first-order knowledge compilation. In *Proceedings of the Twenty-Second international joint conference on Artificial Intelligence-Volume Volume Three*, 2178–2185. AAAI Press.
- Wyatt, D.; Choudhury, T.; and Bilmes, J. 2008. Learning hidden curved exponential family models to infer face-to-face interaction networks from situated speech data. In *AAAI*, 732–738.