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 straightforward 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. $\forall X, Y, Z \ Fr(X, Y) \land Fr(Y, Z) \Rightarrow Fr(X, Z)$, 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,
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.

Formally, an MLN is a set of \( m \) pairs \( \{ f_i, \theta_i \}^m_{i=1} \), where \( f_i \) is a First-Order Logic (FOL) formula defined with constants, logic variables (or just variables) and predicates, the weight \( \theta_i \) is a real number (\( \infty \) 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, \( Smoke(alice) \land Fr(alice, bob) \Rightarrow Smoke(bob) \) 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. An assignment to a grounded atom sets a truth value to the atom (e.g., Smoke(alice) = \( \perp \) and \( Fr(alice, bob) = \top \)).

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

\[
p(w; \theta) = \frac{1}{Z(\theta)} \prod_{i=1}^{m} \exp(\theta_i N(f_i, w))
\]

\[
= \frac{1}{Z(\theta)} \exp(\theta^T N(f, w))
\]

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

\[
Z(\theta) = \sum_{w \in W} \exp(\theta^T N(f, w)) \tag{1}
\]

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 \( p(w) = \frac{1}{1 \times 4 \times 2 \times 4.6 \times 8 \times 1.1 \times 12} \). 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 \( \psi \) be a query and an evidence respectively. An inference problem on MLN is to compute the conditional probability \( p(g|\psi) = p(g \land \psi)/p(\psi) \), 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:

\[
\theta^* = \arg \max_{\theta} L(\theta|g) = \arg \max_{\theta} \{ \theta^T N(f, g) - \ln Z(\theta) \}
\]

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 \( W \). 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
graphs \( G = (V, E) \) and \( H = (V', E') \) are isomorphic if there is an one-to-one mapping \( \Psi \) 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 \( \{H_1, H_2, \ldots, H_r\} \) be the set of subgraph structures of interests and \( \theta' \) be a \( r \)-dimensional vector (or a model coefficient), an ERGM defines a probability for a graph \( g \in G \):

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

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

\[
\phi(g) = \left( \frac{t(g, H_1)}{t(K_n, H_1)}, \frac{t(g, H_2)}{t(K_n, H_2)}, \ldots, \frac{t(g, H_r)}{t(K_n, H_r)} \right)
\]

Here, \( t(K_n, H_i) = \binom{n}{\Psi(H_i)} \) 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 \( N(f, w) \) can be translated into \( \phi(g) \) for a certain assignment \( \theta \).

### 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:

\[
\sum_{w \in \mathcal{W}} \exp(\theta^T N(f, w)) = \sum_{h \in \mathcal{H}_F} \#(h) \exp(\theta^T h),
\]

where \( \mathcal{H}_F = N(f, \mathcal{W}) \) is the domain of sufficient statistics, and \( \#(h) = \{w \vert N(f, w) = h, w \in \mathcal{W} \} \) is the counting function that counts the number of (redundant) configurations of a sufficient statistic \( h \). For first-order probabilistic models, \( \mathcal{H}_F \) 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 \( \mathcal{H}_F \) and evaluate \( \#(h) \) efficiently.

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

\[
f_1 : \quad \text{Pred}(X) \land \text{Pred}(Y) \quad \theta_1
\]
\[
f_2 : \quad \text{Pred}(X) \land \neg\text{Pred}(Y) \quad \theta_2
\]
\[
f_3 : \quad \neg\text{Pred}(X) \land \neg\text{Pred}(Y) \quad \theta_3
\]

The lifted computation of the partition function:

\[
\sum_{w \in \mathcal{W}} \exp(\theta_1 N(f_1, w) + \theta_2 N(f_2, w) + \theta_3 N(f_3, w))
\]

\[
= \sum_{c_0, c_1 \geq 1} \binom{|D|}{c_0} \exp(\theta_1 c_1^2 + \theta_2 c_1 c_0 + \theta_3 c_0^2)
\]

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 \( c_0 + c_1 = |D| \). All grounded \( \text{Pred}(X) \) with the same assignment are exchangeable or interchangeable. Therefore, we can identify \( \mathcal{H}_F \) very easily given a counting argument by using the counting arguments, together with the counting function:

\[
h = (c_1^2, c_1 c_0, c_0^2)^T, \quad \#(h) = \binom{|D|}{c_0}
\]

Because \( |\mathcal{H}_F| = |D| \) while \( |\mathcal{W}| = 2^{|D|} \), 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 \( D \) be nodes, and there is an edge between \( a \) and \( b \) if the ground atom \( F'r(a, b) = 1 \), 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>
<thead>
<tr>
<th>Feature</th>
<th>Weight</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \forall X \neg \text{Smoke}(X) )</td>
<td>1.4</td>
<td>Most people don’t smoke</td>
</tr>
<tr>
<td>( \forall X \forall Y \neg \text{Fr}(X, Y) )</td>
<td>4.6</td>
<td>Most people are not friends</td>
</tr>
<tr>
<td>( \forall X \forall Y \text{Smoke}(X) \land \text{Fr}(X, Y) \Rightarrow \text{Smoke}(Y) )</td>
<td>1.1</td>
<td>Friends of people who smoke are likely to smoke</td>
</tr>
<tr>
<td>( \forall X \exists Y \text{Fr}(X, X) )</td>
<td>(-\infty)</td>
<td>Anti-reflexive relation constraint</td>
</tr>
<tr>
<td>( \forall X \forall Y \neg(\text{Fr}(X, Y) \Leftrightarrow \text{Fr}(Y, X)) )</td>
<td>(-\infty)</td>
<td>Symmetric relation constraint</td>
</tr>
</tbody>
</table>
as a unique undirected graph, with order \( n = |D| \). Without ambiguity, we will refer to \( w \) as an assignment and a graph interchangeably for the rest of the paper. Accordingly, \( N(f, w) \) 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 \( N(f, w) \) into a linear combination of subgraph statistics, and consequently build the connection between MLNs and ERGMs.

### 3.1 Subgraph Features \( H^k \)

Given a homogeneous bivariate MLN Formula with \( k \) logical variables, subgraph features \( H^k \) of the formula are graphs in vector form \( H^k = (H^k_0, \ldots, H^k_r)^T \). Here, each \( H^k_0 \) is a graph of order-\( k \), and for any \( 0 \leq i, j \leq r_k \), \( H^k_i \) and \( H^k_j \) are not isomorphic. Let \( H^k_i \leq H^k_j \) if \( H^k_i \) is isomorphic to some subgraph of \( H^k_j \), then \( \leq \) defines a partial order on \( \{ H^k_i \} \). 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 \( H^2 = (H^2_0, H^2_2)^T \), because \( f_1 \) has two logical variables: \( X \) and \( Y \). Here, \( H^2_0 \) is an empty order-2 graph and \( H^2_2 \) has two nodes connected by an edge, or \( K_2 \).

\[
H^2_0 : \quad \bullet \quad \rightarrow
\]

Here, \( N(f_1, H^2_0) = 2 \) and \( N(f_1, H^2_2) = 0 \). Given any graph \( w \), we know \( t(w, H^2_0) = (\binom{n}{2}) \), and \( t(w, H^2_2) \) is the number of edges in \( w \). Because \( H^2_0 \leq H^2_2 \), the counting \( t(w, H^2_0) \) includes \( t(w, H^2_2) \), therefore we need to exclude the latter when computing their contribution to formula counting. More specifically, \( N(f_1, w) \) can be represented as:

\[
N(f_1, w) = N(f_1, H^2_0) \left( t(w, H^2_0) - t(w, H^2_2) \right)
+ N(f_1, H^2_2) \left( t(w, H^2_2) \right)
= \binom{n}{2} - 2t(w, H^2_2)
\]

\(^3\)Note that \( H^k \) and \( N(f_1, H^k) \) 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** \( f_2 : Fr(X, Y) \land Fr(Y, Z) \Rightarrow Fr(X, Z) \). 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 \( H^3 = (H^3_0, H^3_3, H^3_2, H^3_1)^T \), as illustrated below:

\[
\begin{align*}
&H^3_0: \quad \bullet \quad \bullet \quad \bullet \\
&H^3_3: \quad \rightarrow \\
&H^3_2: \quad \rightarrow \\
&H^3_1: \quad \rightarrow
\end{align*}
\]

\( H^3_0, H^3_2 \) and \( H^3_3 \) satisfies \( f_2 \) regardless of assignments from logical variables to nodes. Thus, by considering permutations, \( \pi(f_2, H^3_0) = \pi(f_2, H^3_1) = \pi(f_2, H^3_2) = 6 \), \( H^3_2 \) do not satisfy \( f_2 \) when there is no edge between \( X, Z \) (i.e., \( \neg Fr(X, Z) \)). Thus, considering two such cases, \( \pi(f_2, H^3_2) = 4 \). The shorthand representation is as follows,

\[
\pi(f_2, H^3) = (6, 6, 4, 6)^T
\]

where \( H^3 = (H^3_0, H^3_3, H^3_2, H^3_1)^T \).

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

\[
N(f_2, w) = \sum_{x \in W^3} \sum_{0 \leq j \leq 3} \pi(f_2, H^3_j)|W^3_j|
\]

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

\[
|W^3_j| = t(w, H^3_j)
\]

is trivial because they both count the number of triangles in \( w \). To compute \( |W^3_j| \), we need to exclude the counts of \( W^3_j \) from \( t(w, H^3_j) \) since \( H^3_j \not\leq H^3 \) and each occurrence of \( H^3_j \) is counted as \( t(w, H^3_j) - 3|W^3_j| \). Similarly, we have \( |W^3_j| = t(w, H^3_0) - 3|W^3_j| \) and \( |W^3_j| = t(w, H^3_2) - 3|W^3_j| \). Put everything together we get:

\[
\begin{pmatrix}
|W^3_0| \\
|W^3_1| \\
|W^3_2| \\
|W^3_3|
\end{pmatrix} =
\begin{pmatrix}
1 & -1 & 1 & -1 \\
0 & 1 & -2 & 3 \\
0 & 0 & 1 & -3 \\
0 & 0 & 0 & 1
\end{pmatrix}
\begin{pmatrix}
t(w, H^3_0) \\
t(w, H^3_1) \\
t(w, H^3_2) \\
t(w, H^3_3)
\end{pmatrix}
\]

Note that,

\[
N(f_2, w) = \sum_{0 \leq j \leq 3} N(f_2, H^3_j)|W^3_j|
= (N(f_2, H^3)^T A) (t(w, H^3))
= (6, 0, -2, 6) t(w, H^3)
= 6t(w, H^3_0) - 2t(w, H^3_2) + 6t(w, H^3_3)
\]

Here, matrix \( A \) is:

\[
A =
\begin{pmatrix}
1 & -1 & 1 & -1 \\
0 & 1 & -2 & 3 \\
0 & 0 & 1 & -3 \\
0 & 0 & 0 & 1
\end{pmatrix}
\]
Because \( t(w, H^k) = \binom{n}{3} \) is a constant, we can re-write \( Z(\theta) \) for the MLN in Table 2 as
\[
Z(\theta) = \exp \left( 2 \frac{\binom{n}{2}}{2} \theta_1 + 6 \frac{\binom{n}{3}}{3} \theta_2 \right) \\
\sum_{w \in \mathcal{W}} \exp \left( -2\theta_1 t(w, H^2_1) - 2\theta_2 t(w, H^2_2) - 6\theta_2 t(w, H^3_3) \right)
= c(\theta) \sum_{w \in \mathcal{W}} \exp \left( \theta'^T t(w, f_g) \right) \tag{3}
\]
Here \( \theta' = (-2\theta_1, -2\theta_2, -6\theta_2)^T \) and \( f_g = (H^2_1, H^2_2, H^3_3)^T \). 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 \( H^k = (H^k_0, \ldots, H^k_k)^T \) as
\[
N(f, w) = N(f, H^k) t(w, H^k)
\]
where matrix \( A \) is defined recursively:
\[
A_{ij} = \begin{cases} 
1 & i = j \\
- \prod_{l \neq i} A_{lh} & \text{otherwise}
\end{cases}
\tag{4}
\]
Notice that \( t(H^k_0, H^k_1) \neq 0 \) if \( H^k_0 \leq H^k_1 \), and matrix \( A \) can be computed efficiently using our dynamic programming as shown in Algorithm 1.

### 3.2 Relationship with Triangle-Free Graph Enumeration

Re-writing (3) with (2) we can get:
\[
c(\theta) \sum_{w \in \mathcal{W}} \exp \left( \theta'^T t(w, f_g) \right) = c(\theta) \sum_{h \in \mathcal{H}_u} \#(h) \exp \left( \theta'^T h \right)
\]
In this case, \( \#(h) \) enumerates all the graphs with the given subgraph configuration \( h \). 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
\[
\# \text{ of order-} n \text{ triangle-free graphs} = \sum_{h \in \mathcal{H}_u} \#(h) \delta(h_3 = 0)
\]
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 \( n < 17 \) 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 \( n \).

### 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 \( \phi(f_g) \) as sufficient statistics and \( B^\theta \) as parameters where \( B = \text{diag}(t(K_n, H^2_1), t(K_n, H^3_2), t(K_n, H^3_3)) \). The computation of \( B \) and \( \theta' \) 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 exploit 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 \( \mathcal{H}_\phi = \{ \phi(G) \} \) be the space of subgraph densities, we apply the generalized counting function strategy to get:
\[
\ln Z(\theta) = \ln \sum_{h \in \mathcal{H}_\phi} \#(h) \exp (\theta'^T h) \\
= \ln \sum_{h \in \mathcal{H}_\phi} \exp (\theta'^T h + \ln \#(h)) \\
\simeq \max_{h \in \mathcal{H}_\phi} \{ \theta'^T h + \ln \#(h) \} + O(\ln |\mathcal{H}_\phi|) \tag{5}
\]
Here, the first term of (5) is a good approximation with a relatively small number of low order subgraph features because \( |H| \) 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 \( \#(h) \) and feature space \( \mathcal{H}_\phi \) within \( \mathcal{G}_u \) leads to the following lower bound to (5):
\[
\gamma(\theta, u) = \max_{h \in \mathcal{H}_u} \{ \theta'^T h + \ln \#_u(h) \} \tag{6}
\]
The ECS approximation exploits the property that as \( n \to \infty \), subgraph statistics of graphs in \( \mathcal{G}_u \) tends to concentration around a single configuration in \( \mathcal{H}_\phi \), and proposes the following approximation of (6):
\[
\gamma(\theta, u) = \theta'^T \rho(u) + \binom{n}{2} H(u) H(u'/2)
\]
Where \( H(x) = -x \ln x - (1 - x) \ln(1 - x) \), \( \rho(u) = (\rho_1(u), \ldots, \rho_r(u))^T \) with \( \rho_i(u) = \binom{n}{i} e^{(H_i)} \), and \( u \) be an edge count between 0 and \( \binom{n}{2} \). Given parameter \( \theta' \), the variational approximation ECS(\( \theta, H \)) 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 \( t(w, H^k) \) 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 \( H_k^i \).

Compute \( A \) by (4) using \( H_k \).

\[ N(f_i, w) \leftarrow N(f_i, H_k^i) \cdot A \cdot t(w, H_k) \]

end for

Substitute \( N(f, w) \) in \( \ln Z(\theta) \) with subgraph features \( H_k \) and sum terms with the same subgraphs as in (3).

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

\( H \leftarrow \) subgraph features \( H_k^i \) with nonzero coefficients (\( H \subseteq H_k \)).

\( \theta' \leftarrow \) coefficients of \( H \).

\( B \leftarrow \text{diag}(t(K_n, H_0), \ldots, t(K_n, H_{r_k})) \)

\( M \leftarrow \text{ECS}(B\theta', H) \quad \triangleright \text{Call ECS approximation} \)

return \( c(\theta)M \)

---

![Figure 1: Accuracy of approximate lifting](image)

**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, 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; Apsel 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-
algorithm 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


Gogate, V.; Jha, A. K.; and Venugopal, D. 2012. Advances in lifted importance sampling. In AAAI.


