LIFTED PROBABILISTIC RELATIONAL INFERENCE FOR UNCERTAIN NETWORKS

BY

WEN PU

DISSERTATION

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Computer Science
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2014

Urbana, Illinois

Doctoral Committee:

Associate Professor Eyal Amir, Chair, Director of Research
Professor Dan Roth
Professor Gerald DeJong
Professor David Hunter, Pennsylvania State University

Abstract

Probabilistic Relational Graphical Model (PRGM) is a popular tool for modeling uncertain relational knowledge, of which the set of uncertain relational knowledge is usually assumed to be independent with the domain of the application. One common application of PRGM is to model complex networks using structural features. Efficient and accurate inference algorithms that can handle models with non-trivial structural features (e.g., transitive relations) are important for applications of this kind. In this thesis, (1) we provide new algorithm for efficient and accurate inference on PRGMs with structural features; (2) we show a counter example to the domain-independence assumption of PRGM.

A PRGM is a set of uncertain relational knowledge, which translates to Probabilistic Graphical Models (PGM) on different domains of discourse. Lifted inference and domain-independence assumption are two important concepts for PRGM. Domain-independence assumption separates the uncertain relational knowledge of a PRGM from its domains of application, therefore distinguishes PRGM from propositional PGM. Lifted inference techniques try to speedup inference on PRGM by lifting the computation from propositional level to relational level. However, these techniques are not designed to handle complex structural features, therefore lack efficiency and accuracy in the presence of these features.

In this thesis, we propose a deterministic approximate inference algorithm

for Exponential Random Graph Model (ERGM) – a family of statistical models, which are closely related to PRGM. An ERGM defines a probabilistic distribution of all graphs of n nodes using a set of subgraph statistics. The main insight enabling this advance is that subgraph statistics are sufficient to derive a lower bound for partition functions of ERGM when the model of interests is not dominated by a few graphs. We then show that a class of PRGMs with structural features can be converted to ERGM, which leads to an approximate lifted inference algorithm for PRGM. Theoretical and experimental results show that the proposed algorithms are scalable, stable, and precise enough for inference tasks.

Lastly, we show a counter example of the domain-independence assumption. In general, PRGM parameters fitted to one network data cannot be extrapolated to other networks of different sizes.

To My Family And Friends

Acknowledgments

This thesis would not exist without support of many people. First and foremost, I would like to thank my advisor, Dr. Eyal Amir, for his invaluable advice and generous support during my thesis study. He inspired me to pursue AI research, and encouraged me whenever I met difficulties throughout the years. He always treats his students as friends instead of acting as a supervisor, which made my life as PhD student much less stressful. I felt lucky to have him as my academic mentor. I would also like to thank my committee members, Dr. Dan Roth, Dr. Gerald DeJong and Dr. David Hunter, who spent considerable time reviewing my thesis and gave valuable feedback. Their encouragements and constructive comments are essential for the completion of this thesis.

I would like to give special thanks to Jaesik Choi. As an excellent researcher, his passion for research has always been a great source of inspiration for me. As a close collaborator, his insightful observations and comments have substantially advanced the results of this thesis. He taught me a lot about how to conduct research and how to write good research papers. Thank you!

I would like to thank Juan Mancilla, Codruta Girlea. They are the most frequent audience of my new research progresses. I appreciate those valuable first-hand feedbacks, which contribute substantially to my thesis and my presentation. It was a great pleasure to work with you and wish you best

luck!

I would also like to thank my other friends and colleagues at Illinois for their helps over the years: Hang Chen, Abner Guzman-Rivera, Mark Richards, Leonardo Bobadilla, Yonatan Bisk, Daphne Tsatsoulis, Mianwei Zhou and Hsiang-Yeh Hwang.

Over the years, the very kind and supportive department staffs have been extremely helpful. Especially, Donna Coleman, Mary Beth Kelley and Rhonda McElroy, thank you!

During the past seven months, I have been working full-time at LinkedIn. I had to squeeze every minute outside of work to write my thesis, and it was an exhaustive experience. I would like to thank my colleagues at LinkedIn who made it as easy for me as possible. Especially, I would like to thank my manager Anmol Basin and Kun Liu for their support.

Last but not least, I would like to thank my family for their support over my lengthy PhD study. Besides my parents who consistently gave me emotional support at the other end of the earth, I dedicate my appreciation to my lovely wife, Weishu Zhang. She has always been extremely supportive and considerate. Every time I felt frustrated, she was always able to figure out a way to get me back on track. You are the best!

Throughout my studies, I was supported by grants given to my advisor. I would like to acknowledge the support of NSF EAR grant 09-43627, NSF IIS grant 09-17123, NSF IIS grant 09-68552, and a DARPAR grant as part of the Machine Reading Program under AFRL prime contract no. FA8750-09-C-0181. I would also like to acknowledge the support of University of Illinois and Argonne National Laboratory.

Table of Contents

List of Abbreviations

BP	Belief Propagation
CBP	Counting Belief Propagation
CNF	Conjunctive Normal Forms
ECS	Edge Count Search Approximation
ERGM	Exponential Random Graph Model
FOVE	First-Order Variable Elimination
KB	Knowledge Base
LBP	Lifted Belief Propagation
MCMC	Markov Chain Monte Carlo
MLE	Maximum Likelihood Estimation
MLN	Markov Logic Network
PGM	Probabilistic Graphical Model
PRGM	Probabilistic Relational Graphical Model

Chapter 1

Introduction

Probabilistic graphical models (PGM) [Pearl, 1988; Wainwright and Jordan, 2008] are powerful tools for modeling complex real-world systems with uncertainties. A PGM defines a joint probability distribution over a set of propositional knowledge by representing the conditional dependence structures of the random variables using graph notations. Each node of a PGM corresponds to a single random variable, and queries about a set of random variables can be answered by conducting inference on the model at the propositional level.

Probabilistic relational graphical model (PRGM)[1] [Halpern, 1990; Koller, 1999; Ng and Subrahmanian, 1992; Pfeffer et al., 1999; Poole, 2003; Richardson and Domingos, 2006] extends PGM by adding the ability of explicitly modeling uncertain relational knowledge using first-order logic. Given a domain of objects (e.g., a group of people) and a set of uncertain relational knowledge (e.g., friendship, smoking habits), a PRGM provides a concise yet expressive representation that acts as a template for generating an equivalent PGM. In PRGM applications, relational knowledge is usually assumed to be domain-independent. Under this assumption, one can expect learning the relational knowledge from the training data of one domain , and apply the

[1]The concept of PRGM in this thesis is different from *probabilistic relational model* (PRM) [Friedman et al., 1999]. Here PRGM refers to a general class of models that combine relational logic and probabilistic graphical models, while PRM is one specific implementation of PRGM.

model to test data on a different domain.

For example, it is well known that smoking may leads to cancer, no matter who smokes. Using the language of first-order logic, we can write:

$$\forall \mathtt{x}\ \mathtt{Smoke}(\mathtt{x}) \Rightarrow \mathtt{Cancer}(\mathtt{x})$$

However, the logic formula does not capture the uncertainty of the original statement. In PRGM, we are able to explicitly model our belief of the "truthness" of a logic formula. In the language of Markov logic network [Richardson and Domingos, 2006], we may say:

$$\forall \mathtt{x}\ \mathtt{Smoke}(\mathtt{x}) \Rightarrow \mathtt{Cancer}(\mathtt{x}) \quad 0.05$$

Here 0.05 is a real number that represents our confidence to the "truthness" of the relational knowledge. Given a list of weighted relational knowledge and a domain of objects $\{\mathtt{Alice}, \mathtt{Bob}, \mathtt{Chris}, \ldots\}$, a PRGM is able to define a distribution over the set of propositional knowledge {`Smoke`(`Alice`), `Smoke`(`Bog`), `Cancer`(`Bob`), ...}. Under the domain-independence assumption, we can conveniently apply the set of uncertain relational knowledge to a different group of people.

Given the expressiveness of PRGMs, practitioners are tempted to model complex relational knowledge in large domains. This practice usually results in very large PGMs, which post challenge for developing efficient inference algorithms. Lifted inference [Poole, 2003] in PRGMs is an endeavor towards lifting the inference from propositional level to relational level by leverage the structure of the model. These techniques are often able to significantly reduce the computation complexity without compromising accuracy. However, the

applicability and efficiency of lifted inference algorithms usually depend on the relational knowledge in the model.

The presence of relational knowledge with complex interactions between the logic variables often causes difficulty for applying any lifting strategy effectively. One remarkable class of relational knowledge is network structural features. For example, we would like to model the transitivity effects of relations in networks in many applications:

$$\forall \mathtt{x}, \mathtt{y}, \mathtt{z} \quad \mathtt{Friend}(\mathtt{x}, \mathtt{y}) \wedge \mathtt{Friend}(\mathtt{y}, \mathtt{z}) \Rightarrow \mathtt{Friend}(\mathtt{x}, \mathtt{z})$$

However, besides extensive study on lifted inference algorithms, the liftability for PRGMs with relational knowledge of this type is still unclear [Jaeger and Van den Broeck, 2012]. Although general-purpose approximate lifted inference algorithms [Niepert, 2012b; Singla and Domingos, 2008] are able to take any relational formula as input, their behaviors on this particular class of relational features are not clear yet. In fact, a more fundamental problem is yet to be answered: whether these network structural features can provide the desired statistical properties?

This thesis investigates network structural features in PRGMs and how to scale up the inference from a probabilistic graph-theoretic perspective. We first study exponential random graph model (ERGM) [Hunter et al., 2012; Lusher et al., 2012; Snijders, 2002], a family of statistical models popular in social network analysis applications. We propose a deterministic approximate inference algorithm for ERGM. We also show that the ERGMs fitted to one network data cannot be extrapolated to networks of different sizes. Then we show that a class of PRGMs with network structural features can be essentially translate into equivalent ERGMs, which leads to a new approxi-

mate lifted inference algorithm. Lastly, the relationship between PRGM and ERGM also reveals that the popular domain-independence assumption for PRGM is not always true, therefore prompting the limitation on the expressiveness of PRGM.

The thesis is organized as follow: Chapter 2 reviews the background or PRGM and its inference; Chapter 3 to Chapter 5 is the main contributions; Chapter 6 concludes the thesis. The rest of this chapter summarizes the main results.

1.1 Learning Exponential Random Graph Models

Exponential random graph model (ERGM) [Lusher et al., 2012] is a family of statistical models commonly used by social network analysis practitioners. An ERGM defines a probabilistic distribution over all graphs of n nodes. Besides conventional node-wise attributes, features of an ERGM may include subgraph statistics (e.g., number of edges, triangles, and k-stars) Robins et al. [2007]. The model is able to captures the correlation between network substructures explicitly, which enables many interesting inference tasks Simpson et al. [2011]; Wyatt et al. [2008].

Learning ERGMs from data is achieved through maximum likelihood estimation (MLE). Unfortunately, such learning is hard even for networks of modest size (e.g., 40 nodes) because calculating normalizing constants (partition functions) precisely for such models is intractable. For this reason, most current techniques involve stochastic sampling Handcock et al. [2003]; van Duijn et al. [2009]. This often results in slow mixing time, and practical difficulties well known as "degeneracy" or "near degeneracy" phenomenon, in

which the learned model tends to generate either empty or complete graphs Bhamidi et al. [2008]; Handcock [2003]; Hunter et al. [2012]; Snijders [2002]. In Chapter 3, we propose a new deterministic approximation to the log partition functions. Compared to sampling based method, the proposed algorithm is able to produce more reliable estimations. Analysis of the approximation also shows that the behavior of the same ERGM may change for networks of different sizes.

Specifically, we present a quadratic time (or linear time w.r.t. the number of random variables) deterministic approximation to the log partition function of ERGMs. Asymptotic properties of the subgraph statistics space enable this new approximation. The approximation works as follows: Given (coefficient) parameters θ of an ERGM, find that edge-count u (between 0 and $\binom{n}{2}$) that maximizes $\tilde{\gamma}(\theta, u) = \theta^T \rho(u) + C(n, u)$ (See (3.13) for definition), where $\rho(u)$ is a vector of subgraph statistics approximated for graphs with u edges and function $C(n, u)$ approximates the logarithm of the number of graphs with subgraph statistics close to $\rho(u)$. Once the maximizing u is found, we estimate the log partition function $\ln Z(\theta)$ by $\tilde{\gamma}(\theta, u)$. The approximation works because this $\rho(u)$ captures the subgraph statistics of a large (asymptotically) mass of graphs of n nodes. So, in a sense, many graphs look similar from a subgraph statistics perspective. We show that the new method performs well experimentally comparing to existing sampling methods Gelman and Meng [1998]; Handcock et al. [2003]. Experimental results show that the new algorithm yields reliable approximations when the size of the network is larger than 30.

Our asymptotic analysis on a class of lower bounds to $\ln Z(\theta)$ shows that the parameter θ needs to be in $O(n^2)$ to be relevant for large n, which suggests that fixed θ leads to different models when n changes. This result is

consistent with [Schweinberger, 2011] and more recently [Shalizi and Rinaldo, 2013].

1.2 Approximate Lifting for Structural Relational Knowledge

Exact lifted inference in the presence of structural relational knowledge, such as transitive relations, is still an open problem [Jaeger and Van den Broeck, 2012]. This thesis approaches this problem through a probabilistic graph-theoretic approach by building connection between ERGM and PRGM.

ERGMs and PRGMs are both tools for modeling structural relational knowledge. However, they have very different formulations: ERGMs explicitly use subgraph statistics as features, while PRGMs use weighted first-order logic formulas. In Chapter 4, we show that PRGMs of structural features can be translated into equivalent ERGMs using an efficient dynamic programming algorithm.

This relationship between PRGM and ERGM enables us to leverage ECS approximation (see Chapter 3) to derive a deterministic approximate inference algorithm for PRGMs of structural relational knowledge. We show that the proposed algorithm is essentially a generalization of the idea behind counting elimination in first-order variable elimination [De Salvo Braz et al., 2005], one of the exact lifted inference algorithms.

The proposed algorithm 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 existing approximate lifted inference algorithms: it is a deter-

ministic algorithm that runs in quadratic time with respect to domain size n, and the approximate function converges to the actual function asymptotically as $n \to \infty$.

1.3 Limited Expressiveness of Structural Features

PRGM is well known for its expressiveness: Given a domain of interests and a set of weighted relational knowledge, a PRGM serves as a template for generating a probabilistic distributions over all the relevant propositional knowledge. Ideally in a PRGM, we assume the uncertain relational knowledge is independent with the domain of interests. One typical scenario under this domain-independence assumption is that we learn the relational knowledge and their weights from a small observable domain (e.g., through expensive data collection process, such as survey), and then apply the learned model to a large domain (e.g., population of the whole city). However, we show in Chapter 5, this seemingly safe assumption does not always hold.

The relationship between PRGM and ERGM revealed in Chapter 4 suggests that the PRGM of structural relational knowledge essentially inherits all the undesirable statistical properties of ERGM; therefore a PRGM fitted to one data set does not apply to other domains of different sizes.

Chapter 2

Background

2.1 Probabilistic Relational Graphical Models

A *probabilistic graphical model* (PGM) defines a joint probability distribution over a set of propositional knowledge, and a variety of inference tasks can be performed, such as statistical learning, explanation and prediction. However, a PGM is fixed on a set of given propositional knowledge, therefore is not capable of modeling knowledge of which domain is not preset. For example, a PGM built on a specific social network is not applicable to other networks because a different set of random variables is involved, even though these networks may share similar relational characteristics.

Probabilistic relational graphical model (PRGM) is a family of models that combine relational logic and PGM. Instead of modeling propositional knowledge directly, a PRGM explicitly models a domain of entities and a set of relations among them. Many PRGM languages have been proposed and well studied [Halpern, 1990; Koller, 1999; Ng and Subrahmanian, 1992; Pfeffer et al., 1999; Poole, 2003; Richardson and Domingos, 2006]. Although different languages emphasize on different applications and vary on techniques used, they all target on modeling the probabilistic semantics of the relational knowledge directly. In this thesis, we focus on Markov Logic Net-

works [Richardson and Domingos, 2006] due to its simplicity, but the topics and techniques discussed are also applicable to other PRGMs, given that the grounded models is in exponential family, which is true in most applications. For the rest of the section, we review Markov random field, relational logic, and Markov logic networks.

2.1.1 Markov Random Field

A *Markov random field* (MRF), also known as *undirected graphical model*, defines a probability distribution which factorizes as a set of functions based on the cliques of an undirected graph representation of the random variables $X = (X_1, X_2, \ldots, X_m) \in \mathcal{X}$ [Pearl, 1988; Wainwright and Jordan, 2008]. X_i is a random variable of the state space $\mathcal{X}_i$, which can be continuous or discrete. In this thesis, we focus on binary random variables, i.e., $\mathcal{X}_i = \{\top, \bot\}$. Let (V, E) be an undirected graph where $s \in V$ has a one-to-one mapping to $X_s \in X$, a clique $C \subset V$ is a fully connected subset of V in (V, E). Each clique C is associated with a *compatibility function* $\psi_C : (\otimes_{s \in C} \mathcal{X}_s) \rightarrow \mathcal{R}^+$, where $\otimes_{s \in C} \mathcal{X}_s$ is the Cartesian product of state spaces $\mathcal{X}_C = \{\mathcal{X}_s | s \in C\}$.

Formally, the distribution of X can be defined as:

$$p(X) = \frac{1}{Z} \prod_{C \in \mathcal{C}} \psi_C(X_C) \tag{2.1}$$

$$\text{where} \quad Z = \sum_{x \in \mathcal{X}} \prod_{C \in \mathcal{C}} \psi_C(x_C)$$

Here Z is a normalizing constant to make sure the given distribution function returns valid probabilities, and $\mathcal{C}$ is the set of maximal cliques. The definition of ψ_C is therefore local with respect to X_C for each $C \in \mathcal{C}$.

Figure 2.1 illustrates a MRF on $m = 6$ random variables. The graphical

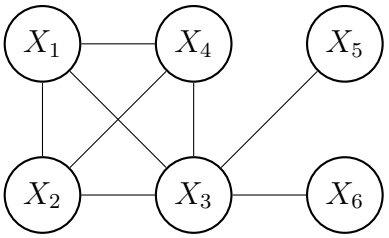


Figure 2.1: Illustriation of a simple Markov random field with m=6 random variables. There are three maximal cliques: $\{X_1, X_2, X_3, X_4\}$, $\{X_3, X_5\}$ and $\{X_3, X_6\}$. $\{X_3\}$ forms a Markov blanket for $\{X_1, X_2, X_4\}$, $\{X_5\}$ and $\{X_6\}$.

representation of a MRF uniquely specifies the conditional independence between subsets of random variables. Given a subset of the random variables A, the set of random variables B adjacent to S forms a *Markov blanket* of A [Pearl, 1988], so that A is conditionally independent with any other random variables given B. For example, once X_3 is fixed, any two of $\{X_1, X_2, X_4\}$, $\{X_5\}$ and $\{X_6\}$ are independent with each other or the combination of the rest. All the probability distributions that satisfy this specification can be represented by the MRF by choosing a proper set of compatibility functions [Wainwright and Jordan, 2008].

Given observations of X and a set of functions $\phi = (\phi_1, \phi_2, \ldots, \phi_r)$ where $\phi_i : X_{C_i} \to \mathcal{R}$ is a function defined on clique $C_i \in \mathcal{C}$, we would like to identify a probability distribution p (i.e., find a proper compatibility function ψ), so that the empirical expectation $\hat{\mu}_\phi$ of $\phi(X)$ equals to $E_p(\phi(X))$:

$$E_p(\phi(X)) = \hat{\mu}_\phi$$

In general, there are many distributions meet this requirement. Therefore, the problem is under-determined. In this case, maximum entropy principle is usually employed to pick the one from the family of qualified distributions that maximize the Shannon entropy, so that it has the maximal uncertainty

under the conditional independence assumptions specified by the graph representation. The optimal solution p^* leads to an exponential family distribution [Wainwright and Jordan, 2008]:

$$p_\theta(X) \propto \exp\{\theta^T \phi(X)\} = \exp\left\{\sum_{i=1}^{r} \theta_i \phi_i(X)\right\} = \prod_{i=1}^{r} \exp\left\{\theta_i \phi_i(X)\right\}$$

where $\theta = (\theta_1, \theta_2, \ldots, \theta_r) \in \mathcal{R}^r$ is a vector of model parameters. A natural choice is $\psi_{C_i}(X_{C_i}) = \exp\{\theta_i \phi_i(X_{C_i})\}$. For simplicity, we drop the subscript C_i from $\phi_i(X_{C_i})$ for the rest of the thesis and write $\phi_i(X)$ instead. The probability mass function of an exponential family Markov random field is:

$$p_\theta(X) = \frac{1}{Z(\theta)} \exp\left\{\theta^T \phi(X)\right\} \qquad \text{where} \quad Z(\theta) = \sum_{X \in \mathcal{X}} \exp\{\theta^T \phi(X)\} \quad (2.2)$$

Once the probability distribution p is specified, it can be used to compute the conditional distribution of random variables given a set of observations or evidence. Specifically, for $X', E \subseteq X$ and $X' \cap E = \emptyset$, we are interested in computing the *conditional distribution* $p_\theta(X'|E)$. The problem of computing *marginal distribution* $p_\theta(X')$ is a special case of computing conditional distribution for which $E = \emptyset$. For a given observation $X' = x$, $\log p_\theta(X' = x)$ is the *log-likelihood* of state x. Sometimes we are more interested in the *most probable explanation* (i.e., the mode) of the model instead of the distribution: $\text{argmax}_{x \in \mathcal{X}}\, p_\theta(x)$.

Another important inference task for MRF is to fit the model to a given data set, or *learning the parameters*. Assume $\{x^{(1)}, x^{(2)}, \ldots, x^{(n)}\}$ are n i.i.d samples from $p_\theta(X)$, the objective of the learning task is to find the optimal θ^* that maximize the log-likelihood of the samples:

$$
\begin{aligned}
\theta^* =& \operatorname*{argmax}_{\theta \in \Theta} \left\{ \log \prod_{i=1}^{n} p_\theta(x^{(i)}) \right\} \\
=& \operatorname*{argmax}_{\theta \in \Theta} \left\{ \sum_{i=1}^{n} \log p_\theta(x^{(i)}) \right\} \\
=& \operatorname*{argmax}_{\theta \in \Theta} \left\{ \theta^T \sum_{i=1}^{n} x^{(i)} + n \log Z(\theta) \right\} \\
=& \operatorname*{argmax}_{\theta \in \Theta} \left\{ \theta^T \bar{x} + \log Z(\theta) \right\} \qquad \text{where } \bar{x} = \sum_{i=1}^{n} x^{(i)}/n
\end{aligned} \tag{2.3}
$$

For this thesis, we will focus on computing log-likelihood function and the learning task. In Section 2.2, we will review some common inference algorithms for MRFs.

2.1.2 First-Order Logic

The expressiveness of propositional logic is limited to representing propositions. For example, we can represent the following statement in propositional logic conveniently: "if Alice smokes, Alice and Bob are friends, then Bob also smokes". However, this statement does not tell us anything about smoking habits and friendship beyond the case of Alice and Bob. First-order logic [Genesereth and Nilsson, 1987; Russell et al., 1995] solves this problem by providing a more powerful language that enables us to represent relational knowledge like "if `x` smokes, `x` and `y` are friends, then `y` smokes", where `x` and `y` can be anyone.

A first-order logic formula contains three types of non-logical symbols: *object constant*, *predicate* and *function*. Objects constants form a *domain of discourse*. For example, a domain of people may contain object constants like "`Alice`", "`Bob`" and "`Chris`". A predicate symbol, usually associated with

an arity k, represents some relation between k objects (e.g., "`Friend(Alice, Bob)`" is a binary relation between `Alice` and `Bob`, "`Smoke(Bob)`" is a unary relation on `Bob`). Predicates with 0-arity are treated as propositions. A function symbol of arity k maps a tuple of k objects to an object in the domain (e.g., "`MotherOf(x)`" is a unary function). We use *logic variables*, such as `x` and `y`, to represent any of the objects in the domain instead fixed object constants.

The logical symbols in first-order logic formula we considered are those in propositional logic plus *quantifiers*. An *atomic formula*, or *atom* for short, is a predicate applies to a tuple of objects or logic variables that match its arity, such as "`Friend(Alice, y)`". A *logical formula* is defined recursively on formulas: (1) an atom is a logical formula. (2) Let `F1` and `F2` be two formulas, then the negation $\neg$`F1` (true if `F1` is false), conjunction `F1` $\wedge$ `F2` (true if both `F1` and `F2` are true), disjunction `F1` $\vee$ `F2` (true if `F1` or `F2` is true), implication `F1` $\Rightarrow$ `F2` (true if `F1` is true implies `F2` is true) and equivalence `F1` $\Leftrightarrow$ `F2` (true if `F1` and `F2` are both true or both false) are all formulas. (3) A *quantified formula* is a formula with logic variables and one of the two *quantifiers*: universal qualified formula $\forall$`xF1` is true if `F1` is true for all objects in place of `x` in `F1`; existential qualified formula $\exists$`xF1` is true if `F1` is true for at least one of the objects in pace of `x` in `F1`. We say an atom or formula is *grounded* if there is no free logic variable. For example, `Friend(Alice, Bob)` and $\forall$`x` `Friend(Alice, x)` $\wedge$ `Smoke(Alice)` are fully grounded, while $\forall$`x` `Friend(x, y)` is not.

A *relational knowledge base* (KB) is a conjunction of a set of first-order logic formulas. The semantics of a KB is obtained through an *interpretation* of the first-order language, which specifies the domain of discourse D and the meaning of all non-logical constants, or a *first-order structure*. For example, a

domain of people could be $D = \{\texttt{Alice}, \texttt{Bob}, \texttt{Chris}\}$, and $\texttt{Friend}(\texttt{x}, \texttt{y}) = \top$ means $\texttt{x}$ and $\texttt{y}$ are friends. A *possible world*, or *Herbrand interpretation*, assigns truth values to all the atoms in the fully grounded KB. Given D is finite, the number of possible worlds is limited. For this thesis, we focus on interpretations with finite D.

A logic inference task is to determine whether a formula $\texttt{F}$ can be entailed from the given KB (i.e., $\texttt{KB} \models \texttt{F}$), and has been widely studied in mathematical logic [Shoenfield, 1967] and artificial intelligence [Russell et al., 1995]. Automated inference procedures usually start by converting formulas in KB into *conjunctive normal forms* (CNF), and prove the entailment by refuting $\texttt{KB} \cup \neg\texttt{F}$ using a list of inference rules. First-order logic is complete in the sense that if $\texttt{KB}$ entails $\texttt{F}$ then there is a finite proof (i.e., $\texttt{KB} \models \texttt{F} \Rightarrow \texttt{KB} \vdash \texttt{F}$); However, there is no guarantee that the inference procedures will halt if $\texttt{KB} \not\models \texttt{F}$.

Besides the semi-decidability of general first-order inference, one major disadvantage of first-order logic for artificial intelligence applications is its limitation on representing uncertainty. A state of the system (i.e., a world) is either possible by satisfying all the formulas, or impossible by violating anyone of them. It is not straightforward to represent the concept that a certain world is highly likely. Nilsson [1986] and Halpern [1990] first studied the semantics for combining logic and probability, and leads to the development of many hybrid languages, including PRGMs. In next section, we introduce a popular general purpose PRGM variation built on first-order KB.

2.1.3 Markov Logic Networks

Halpern [1990] discussed two different semantics for combining first-order logic and probability. The first is to embed the probabilistic information into

the domain, so that we can make statistical statements like "The chance of a randomly chosen American is a smoker is 20%"; the second is to assign probability to a possible world describing our degree of belief on a statement like "The probability of Alice is a smoker is 20%". Most aforementioned PRGMs fall in the second category, including Markov Logic Networks.

A *Markov logic network* (MLN) [Richardson and Domingos, 2006] is an augmented first-order KB on a finite domain, where each formula in the KB is associated with a weight to signify our confidence of the truthfulness of the statement. Unlike first-order logic, which rules out worlds that could not satisfy all formulas in the KB, an MLN specifies a probability distribution over all the possible worlds. A world that satisfies many formulas of strongly confidence is more probable than a world that violates many of those formulas.

Definition 1. *[Richardson and Domingos, 2006] A Markov logic network L is a list of pairs $(f_i, \theta_i)_{i=1}^{r}$, where f_i is a first-order logic formula and its weight w_i is a real number. Given domain $\mathcal{D}$, it defines a Markov random field $M_{L,D}$ as follows:*

1. *Each ground atom is a binary variable in $M_{L,D}$;*
2. *Each grounding of formula f_i forms a clique among involved ground atoms in $M_{L,D}$. It corresponds to a feature function of its ground atoms that is θ_i if the ground formula evaluates to true ($\top$), or 0 if the ground formula evaluates to false ($\bot$).*

Let $x \in \mathcal{X}$ be an assignment to all the ground atoms in $M_{L,D}$ (i.e., a possible world) and C_i be the set of cliques induced from formula f_i, we use the notation $f_i^{(j)}(x)$ to represent the feature function for the j-th grounding of formula f_i (i.e., the j-th clique in C_i):

$$f_i^{(j)}(x) = \begin{cases} \theta_i & \text{the } j\text{-th grounding of } f_i \text{ in } x \text{ is true} \\ 0 & \text{otherwise} \end{cases}$$

The probability mass function for $x \in \mathcal{X}$ in $M_{L,D}$ is:

$$\begin{aligned} p_\theta(X = x) =& \frac{1}{Z(\theta)} \prod_{i=1}^{r} \prod_{j=1}^{|C_i|} \exp\left\{ f_i^{(j)}(x) \right\} \\ =& \frac{1}{Z(\theta)} \exp\left\{ \sum_{i=1}^{r} \sum_{j=1}^{|C_i|} f_i^{(j)}(x) \right\} \\ =& \frac{1}{Z(\theta)} \exp\left\{ \sum_{i=1}^{r} \theta_i N(f_i, x) \right\} \\ =& \frac{1}{Z(\theta)} \exp\left\{ \theta^T N(\mathbf{f}, x) \right\} \end{aligned} \tag{2.4}$$

Here $N(f_i, x) = \sum_{j=1}^{|C_i|} f_i^{(j)}(x)/\theta_i$ counts the number of groundings of f_i in x that are true; $\mathbf{f} = (f_1, f_2, \ldots, f_r)^T$ is a convenience notation for the list of formulas, and we define the notation $N(\mathbf{f}, x)$ as follow:

$$N(\mathbf{f}, x) = \begin{pmatrix} N(f_1, x) \\ N(f_2, x) \\ \vdots \\ N(f_r, x) \end{pmatrix}$$

$Z(\theta)$ is the normalizing constant:

$$Z(\theta) = \sum_{x \in \mathcal{X}} \exp\left\{ \theta^T N(\mathbf{f}, x) \right\} \tag{2.5}$$

Notice that Eq (2.4) has the same form as Eq (2.2) given $\phi(x) = N(\mathbf{f}, x)$. The sufficient statistics of x for $M_{L,D}$ are the raw counts of satisfied ground formulas in L. For formula f_i, a larger θ_i in general suggests we have stronger confidence on the truthfulness of f_i; therefore the weighted formulas serve as some soft constraints in the model.

It is also possible to set hard constraints in the framework of Definition 1 by setting $\theta_i = -\infty$. In this case, Eq (2.4) suggests any non-zero $N(f_i, x)$ will lead to $p(X = x) = 0$.

The grounding of formulas with universal quantifier in $M_{L,D}$ is straightforward, we simply enumerate through all possible object constants combinations from $\mathcal{D}$, and each ground formula corresponds to a clique. The grounding of formulas with existential quantifier involves a huge clique of all ground atoms of the formula. In the rest of the paper, we focus on formulas with universal quantifiers, which are sufficient for common applications of modeling uncertain networks [1]. We treat free logic variables in formulas as being quantified by universal quantifiers at the outmost level.

Table 2.1 shows a modified example MLN from [Singla and Domingos, 2008]. It has six formulas, of which two are hard constraints (f_5 and f_6). Given the domain $\mathcal{D} = \{\texttt{Alice}, \texttt{Bob}, \texttt{Chris}\}$, the resulting MRF of the MLN will have nine ground atoms (i.e., binary random variables): { `Smoke(Alice)`, `Smoke(Bob)`, `Smoke(Chris)`, `Cancer(Alice)`, `Cancer(Bob)`,`Cancer(Chris)`, `Fr(Alice, Bob)`, `Fr(Bob, Chris)`, `Fr(Alice, Chris)` }[2], therefore $|\mathcal{X}| = 2^9 = 514$ possible worlds. Notice that hard constraint f_5 excluded reflective friendship (e.g., `Fr(Alice, Alice)`); hard constraint f_6 excluded the neces-

[1] Regarding to existential quantifiers, [Kisynski and Poole, 2009] and [Choi et al., 2011] discussed polynomial time inference algorithms for weighted FOL formulas with typical use cases.

[2] We use the shortened `Fr(x,y)` to denote `Friend(x, y)`.

sity of having both `Fr(x, y)` and `Fr(y, x)`, since `Fr(x, y)` = `Fr(y, x)` for all `x` and `y`.

Figure 2.2 illustrates the graphical representation of the MRF from the example MLN in Table 2.1 on our toy domain $\mathcal{D}$. Consider the following world: x={`Smoke(Alice)` $= \top$, `Smoke(Bob)` $= \top$, `Fr(Alice, Bob)` $= \top$, `Cancer(Bob)` $= \top$} and other ground atoms have value $\perp$, the sufficient statistics are:

$$N(\mathbf{f}, x) = \begin{pmatrix} N(f_1, x) \\ N(f_2, x) \\ N(f_3, x) \\ N(f_4, x) \end{pmatrix} = \begin{pmatrix} 1 \\ 2 \\ 5 \\ 2 \end{pmatrix}$$

For f_3, there are six combinations of `x` and `y`, but only one of them (`x=Chris, y=Alice`) evaluates to $\perp$. Therefore, the probability of this world is $p(x) = \frac{1}{Z} \exp(1.4 \times 1 + 4.6 \times 2 + 1.2 \times 5 + 1.1 \times 2)$.

Given $M_{L,D}$, MLN is able to answer queries about ground atoms as discussed in Section 2.1.1. In fact, MLN supports the more general relational queries. For example, query $p(\texttt{Cancer(x)}|\texttt{Smoke(x)}; L, \mathcal{D})$ answers the cancer rate for smokers given the MLN L and domain $\mathcal{D}$.

More specifically, for the two first-order logic formulas $\texttt{F}_1$ and $\texttt{F}_2$, we are interested in computing the conditional distribution:

Table 2.1: A simple MLN on smoking habits, friendship and cancer.

	Feature	Weight	Explanation
1	$\neg\texttt{Smoke}(\texttt{x})$	1.4	Most people don't smoke
2	$\neg\texttt{Friend}(\texttt{x},\texttt{y})$	4.6	Most people are not friends
3	$\texttt{Smoke}(\texttt{x}) \wedge \texttt{Friend}(\texttt{x},\texttt{y}) \Rightarrow \texttt{Smoke}(\texttt{y})$	1.2	Friends of smokers are likely to smoke
4	$\texttt{Smoke}(\texttt{x}) \Rightarrow \texttt{Cancer}(\texttt{x})$	1.1	Smoking is likely to cause lung cancer
5	$\texttt{Friend}(\texttt{x},\texttt{x})$	$-\infty^*$	Friendship relation is anti-reflexive
6	$\neg(\texttt{Friend}(\texttt{x},\texttt{y}) \Leftrightarrow \texttt{Friend}(\texttt{y},\texttt{x}))$	$-\infty^*$	Friendship relation is symmetric

* $-\infty$ means the formula is a hard constraints, which should never be true.

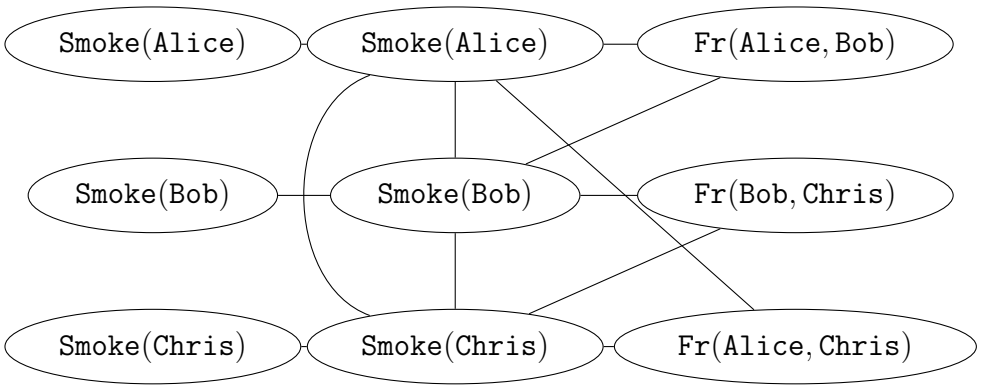


Figure 2.2: Markov random field constructed from MLN in table 2.1 on domain $\mathcal{D} = \{\texttt{Alice}, \texttt{Bob}, \texttt{Chris}\}$

$$
\begin{aligned}
p(\texttt{F}_1|\texttt{F}_2; L, \mathcal{D}) =& p(\texttt{F}_1|\texttt{F}_2; M_{L,D}) \\
=& \frac{p(\texttt{F}_1, \texttt{F}_2; M_{L,D})}{p(\texttt{F}_2; M_{L,D})} \\
=& \frac{\sum_{x \in \mathcal{X}_{F_1} \cap \mathcal{X}_{F_2}} p_\theta(x)}{\sum_{x \in \mathcal{X}_{F_2}} p_\theta(x)}
\end{aligned}
$$

To perform inference task on an MLN, one can always first convert it into a MRF, and carry the inference using conventional inference algorithms. However, the size of the MRF generated from relational models like MLN usually scales up dramatically as the size of domain increases, and more

efficient algorithms that exploit the relational structure of the model have been studied.

2.2 Inference

In this section, we review existing popular inference algorithms for MLN. We first discuss propositional level inference algorithms in the context of MRF in Section 2.2.1, since an MLN can be treated as a template for generating MRFs and these algorithms can be directly used to answer MLN queries. We then discuss the algorithms that exploit the first-order semantics of MLN in Section 2.2.2. We review learning algorithms for MLN in Section 2.2.3.

2.2.1 Inference on Markov Random Fields

Inference on general probabilistic graphical models is NP-hard [Cooper, 1990; Roth, 1996]. For MRF, the difficulty of inference roots in the summation of products of the compatibility functions over a huge state space. Assume there are r random variables, or nodes, in the MFR $\{X_1, X_2, \ldots, X_m, X_{m+1}, \ldots, X_r\}$, if we want to compute the marginal distribution $p_\theta(X_1, \ldots, X_m)$ for random variables $\{X_1, X_2, \ldots, X_m\}$, we need to compute the summation

$$\sum_{X_{m+1}, \ldots, X_r} p_\theta(X_1, \ldots, X_r)$$

For which a naïve algorithm runs in $O(2^{(m-r)})$, not to mention the partition function. If we are interested in computing the log-likelihood function, for example in the EM algorithm, the evaluation of partition function (2.5) is inevitable and leads to $O(2^r)$ summations.

Certain conditional independence structures specified by MRF may be

exploited for efficient inference algorithms. For example, efficient exact inference algorithms (e.g., Variable Elimination [Zhang and Poole, 1994] and Belief Propagation [Pearl, 1982]) are available for acyclic MRFs (i.e., trees). An MRF with cycles can be converted into a *junction tree* [Lauritzen and Spiegelhalter, 1988; Pearl, 1988] through tree decomposition [Robertson and Seymour, 1984], and we may apply tree-based inference algorithms. However, the complexity of these algorithms on junction is in exponential of the tree-width. The tree-width of a graph depends on its intrinsic complexity, yet finding the optimal tree-width for a given graph has been shown to be NP-hard [Arnborg et al., 1987] and we usually settle with sub-optimal tree-decomposition resulting from approximation algorithms [Amir, 2010; Becker and Geiger, 1996].

Approximate inference algorithms on MRF fall in two categories: *variational methods* [Jordan et al., 1998; Murphy et al., 1999; Wainwright and Jordan, 2008; Yedidia et al., 2003] and *Markov Chain Monte Carlo* [Andrieu et al., 2003; Gelman and Meng, 1998]. We review some of the popular algorithms.

Loopy Belief Propagation

Belief propagation (BP) [Pearl, 1988] is an exact inference algorithm on MRFs with a tree topology. Assume the graph topology (V, E) of a binary MRF is a tree, then the cliques in (V, E) are simply nodes and edges, therefore the pmf of the MRF can be factorized as follow:

$$
\begin{aligned}
p_\theta(X_1, X_2, \ldots, X_m) =& \frac{1}{Z(\theta)} \exp\left\{ \sum_{s\in V} \theta_s \phi_s(X_s) + \sum_{(s,t)\in E} \theta_{st}\phi_{st}(X_s, X_t) \right\} \\
=& \frac{1}{Z(\theta)} \prod_{s\in V} \exp\{\theta_s\phi_s(X_s)\} \prod_{(s,t)\in E} \exp\{\theta_{st}\phi_{st}(X_s, X_t)\}
\end{aligned}
$$

We are interested in computing the marginal distribution of X_s of node $s \in V$:

$$
b_s(X_s) = \sum_{\{X_i | i\neq s\}} p_\theta(X_1, X_2, \ldots, X_m) \tag{2.6}
$$

$$
\propto \sum_{\{X_i|i\neq s\}} \prod_{u\in V} \exp\{\theta_u\phi_u(X_u)\} \prod_{(u,v)\in E} \exp\{\theta_{uv}\phi_{uv}(X_u, X_v)\} \tag{2.7}
$$

The computation of $b_s(X_s)$ is the normalized sum-product of a series of factors. Let $N(s) = \{t|(s,t) \in E\}$ be the set of neighbors for node s, and each node $t \in N(s)$ corresponds to the sub tree (V_t, E_t), then all factors in Eq (2.7) that involves X_s are only associated with $N(s)$, therefore we have:

$$
\begin{aligned}
b_s(X_s) \propto& \exp\{\theta_s\phi_s(X_s)\} \prod_{t\in N(s)} M_{ts}(X_s) \\
\text{where} \quad M_{ts}(X_s) =& \sum_{X_{V_t}} \exp\{\theta_{st}\phi_{st}(X_s, X_t)\} p_t(X_{V_t}) \\
=& \sum_{X_t} \exp\{\theta_{st}\phi_{st}(X_s, X_t)\} \sum_{\{X_i|i\neq t, i\in V_t\}} p_t(X_{V_t}) \\
\text{and} \quad p_t(X_{v_t}) \propto& \prod_{u\in V_t} \exp\{\theta_u\phi_u(X_u)\} \prod_{(u,v)\in E_t} \exp\{\theta_{uv}\phi_{uv}(X_u, X_v)\}
\end{aligned}
$$

Notice that $\sum_{\{X_i|i\neq t, i\in V_t\}} p_t(X_{V_t})$ is again a sum-product just like Eq (2.7),

therefore the computation can be carried out recursively. However, instead of repeating the computation for each node in V, we can exploit the fact that $b_s(X_s)$ only depends on $M_{ts}(X_s), \forall t \in N(s)$ and focus on the computation of the "message" functions, which can be defined recursively:

$$M_{ts}(X_s) \leftarrow c \sum_{X_t} \left\{ \exp\left\{\theta_t \phi_t(X_t) + \theta_{st}\phi_{st}(X_s, X_t)\right\} \prod_{u \in N(t)/s} M_{ut}(X_t) \right\} \quad (2.8)$$

Here c is some normalizing constant, and $M_{st}(X_s)$ is the message s sends to t on X_s. The nodal or pairwise marginals can be represented using these message functions:

$$\begin{aligned} b_s(X_s) &= c' \exp\{\theta_s \phi_s(X_s)\} \prod_{t \in N(s)} M_{ts}(X_s) \\ b_{st}(X_s, X_t) &= c'' \exp\left\{\theta_t \phi_t(X_t) + \theta_{st}\phi_{st}(X_s, X_t)\right\} \\ &\quad \times \prod_{u \in N(s)/t} M_{us}(X_s) \prod_{u \in N(t)/s} M_{ut}(X_t) \end{aligned}$$

The joint distribution $p_\theta(X)$ of tree-structured (V, E) can be represented in terms of $b_s(X_s)$ and $b_{st}(X_s, X_t)$:

$$p_\theta(X) = \prod_{s \in V} b_s(X_s) \prod_{(s,t) \in E} \frac{b_{st}(X_s, X_t)}{b_s(X_s) b_t(X_t)}$$

The log partition function $\log Z(\theta)$ for Eq (2.6) is the entropy of $p_\theta(X)$ [Wainwright and Jordan, 2008], therefore we have

$$\ln Z(\theta) = H(p_\theta) = -\sum_{s\in V}\sum_{X_s} b_s(X_s)\log b_{s(X_s)} - \sum_{(s,t)\in E}\sum_{X_s,X_t} b_{st}(X_s,X_t)\log\frac{b_{st}(X_s,X_t)}{b_s(X_s)b_t(X_t)} \tag{2.9}$$

BP algorithm gives exact solution for MRF of tree topology, which does not hold for general MRFs [Wainwright and Jordan, 2008]. However, the algorithm does not stop us from applying it to MRF with cycles. *Loopy Belief Propagation* (LBP) [Murphy et al., 1999; Yedidia et al., 2003] applies Eq (2.8) to any MRF, and conducts the inference by pretending they are trees. There is no guarantee on the convergence of LBP or the accuracy of approximation, yet it has been successfully used in many applications. It is expected to work well only when the MRF is sparse, i.e., close to trees. For MRFs with cycles, Eq (2.9) is known as *Bethe entropy* or *Bethe approximation* [Wainwright and Jordan, 2008].

Mean Field Algorithm

Mean field algorithm [Jordan et al., 1999; Xing et al., 2002] is another popular variational inference algorithm. The naïve mean field method use a tractable product distribution q to approximate the original distribution:

$$p_\theta(X_1,\ldots,X_m) \approx q(X_1,\ldots,X_m) = \prod_{s\in V} q_s(X_s)$$

Here $q_s(X_s)$ is a variational distribution of X_s. By minimizes the KL-

divergence $D(q||p)$, we can derive the following update rule for $q_s(X_s)$:

$$q_s(X_s) \ \propto \ \exp\{\theta_s\phi_s(X_s) + \sum_{t\in N(s)} E_{q_t}(\theta_{st}\phi_{st}(X_s, X_t))\}$$

The deterministic procedure will converge to some local optimal in terms of the KL-divergence, but not to the real distribution. Mean field algorithm may also be treated as a message passing algorithm, in which distribution $q_s(X_s)$ is the message function being passed around. The product distribution assumption suggests that the approximation works better if the correlations between random variables are weaker.

Gibbs Sampling

Given a set of i.i.d. samples $\{x^{(i)}\}_{i=1}^N$ from distribution $p(X)$, the Monte Carlo integration $I_N(f)$ on function $f : \mathcal{X} \to \mathcal{R}$ converges to the expectation $E_p(f(X))$ as $N \to \infty$:

$$I_N(f) = \frac{1}{N}\sum_{i=1}^{N} f(x^i) \ \xrightarrow{N\to\infty} \ E_p(f(X)) = \int_{\mathcal{X}} f(x)p(x)dx$$

Using the Monte Carlo integral, we can conveniently approximate various inference tasks once given the i.i.d. samples from the distribution of interests.

For MRF, the difficulty of drawing samples from $p(X)$ resides in the unknown partition function $Z(\theta)$. Alternatively, a MCMC method constructs a Markov chain on $x^{(i)}$:

$$p(x^{(i)}|x^{(i-1)}, \ldots, x^{(1)}) = T(x^{(i)}|x^{(i-1)})$$

If the stochastic transition matrix $T(x^{(i)}|x^{(i-1)})$ is *irreducible*, *aperiodic* and

has the detailed balance condition:

$$p(x^{(i)})T(x^{(i-1)}|x^{(i)}) = p(x^{(i-1)})T(x^{(i)}|x^{(i-1)})$$

then theoretically the chain will converge to the invariant distribution $p_\theta(X)$. The most important part of a MCMC method is the design of the transition matrix.

Gibbs sampler [Geman and Geman, 1984] is a popular MCMC method for sampling from MRFs. The algorithm is a special case of Metropolis-Hastings method [Hastings, 1970; Metropolis et al., 1953], of which a new sample is never rejected. The transition matrix exploits the conditional independence specified in the graphical structure, and sample of a random variable X_j only depends on its *Markov blanket* $X_{N(j)}$:

$$T(X^{(i)}|X^{(i-1)}) = \begin{cases} p(X_j^{(i)}|X_{N(j)}^{(i-1)}) & \text{if } X_{N(j)}^{(i)} = X_{N(j)}^{(i-1)} \\ 0 & \text{otherwise} \end{cases}$$

The time for a Markov chain to converge is called *mixing time*, and a quick mixing time (usually in polynomial time) is essential for the sampler to be practical.

MCMC methods have been widely used in many applications. It provides a general approach for inference problems in a broad range of probabilistic models. The effectiveness of MCMC methods depends on the design of a fast mixing Markov chain, which in many cases are non-trivial. One can always apply these algorithms even if fast mixing is not guaranteed. However, the slow convergence of the algorithm may lead to wrong solution.

2.2.2 Lifted Inference

Lifted inference is an umbrella term for the group of inference algorithms that exploit the symmetry of PRGM to speed up the computation. We briefly review some of the notable methods.

First-Order Variable Elimination

First-order variable elimination (FOVE) [De Salvo Braz et al., 2005] is an exact lifted inference algorithm. The concept of lifted variable elimination was first introduced by Poole [2003]. Poole shows that the variable elimination algorithm [Zhang and Poole, 1994] can be adapted to work with certain first-order representations directly without propositionalization. Specifically, assuming our target is to compute $\sum_{X_1,\ldots,X_n} \prod_{i=1}^{n} \psi(X_i, Y)$. Here $\{X_i\}_{i=1}^{n}$ are ground atoms of the same type (i.e., generated from the same predicate), Y is another atom (not necessarily grounded), and $\psi(X_i, Y)$ is some factor defined on the two types of atoms. If the ground atoms $\{X_i\}_{i=1}^{n}$ are interchangeable, one can invert the order of summation and product:

$$\sum_{X_1,\ldots,X_n} \prod_{i=1}^{n} \psi(X_i, Y) = \prod_{i=1}^{n} \sum_{X_i} \psi(X_i, Y) = \prod_{i=1}^{n} \psi'(Y)$$

The performance gain of the procedure stems from reducing the number of summations over the whole space of $\{X_i\}_{i=1}^{n}$ to X_i.

FOVE [De Salvo Braz et al., 2005] includes the procedure and refers it as *inversion elimination*, and another lifted variable elimination procedure called with *counting elimination*. The latter is useful for eliminating factors on atoms of the same type (e.g., $\psi(\texttt{Smoke(x)}, \texttt{Smoke(y)})$). The procedure

works as follow:

$$\sum_{X_1,\ldots,X_n} \prod_{i=1}^{n} \prod_{j=1}^{n} \psi(X_i, X_j) = \sum_{i=0}^{n} \binom{n}{i} \psi(0,0)^{i^2} \psi(0,1)^{i(n-i)} \psi(1,0)^{(n-i)i} \psi(1,1)^{(n-i)^2}$$

The key for counting elimination is to identify sum-product problems of this specific form and apply the procedure to accelerate the elimination. This specific example reduced the complexity from $O(2^n)$ to $O(n)$.

De Salvo Braz et al. [2005] introduces strategies to apply these two lifted inference procedures to more general PRGMs by shattering or partially grounding. Milch et al. [2008] further extends FOVE to handle factors with counting formula. However, the performance of the algorithm depends on the applicability of these rules, which, as expected, are too restrictive for many complex models.

Lifted Belief Propagation

Lifted belief propagation (Lifted BP) [Singla and Domingos, 2008] and the more general *counting belief propagation* (CBP) [Kersting et al., 2009] simulate the process of (loopy) belief propagation algorithm on relational graphical models, and identifies random variables and factors that send/receive identical messages during the same iteration. These random variables and factors can be grouped to form a *lifted network* of *super nodes* and *super features*. A slightly modified belief propagation procedure is able to produce the same result as propositional BP on this new graphical representation while saving the redundant computation on identical messages.

Lifted BP and CBP share the same disadvantage of propositional BP

in terms of the accuracy. The compactness of the lifted network suffers from the symmetry breaking (e.g., by new evidence). The resulting lifted network may be the same as the original ground network in extreme cases. Moreover, the messages passing from variables to factors are products of all but one incoming messages from all participating factors, which may cause serious numerical difficulty for dense graphical models that are common when modeling uncertain networks.

Bisimulation-based Lifting

Sen et al. [2008, 2009] propose to explicitly construct symmetrical groups from graphical models using the graph-theoretic concept of bisimulation. The algorithm simulates the process of variable elimination to construct a RV-Elim graph, and reduce the elimination steps that are guaranteed to generate the same output. It is a generalization of the inversion elimination step in FOVE and is able to detect more symmetrical structures. The key for bisimulation-based lifting is to identify *shared factors*, which require two factors share the same domain. However, transitive relations induce complex constraints to the factor domain, and consequently shared factors do not exist.

There are three approximations based on relaxed conditions on shared factors: *approximate bisimulation*, *factor binning* and *mini-bucket scheme*. Approximate bisimulation does not require two nodes have exactly the identical parent elimination path in the RV-Elim graph to be combined into a block. Instead, only the same length-k parent path is enforced. For transitive relations (or self-joins in general), there are only two levels, in which k=1 is equivalent to exact bisimulation. Factor binning also only works for models with more than two levels, in hope that after eliminating one step will result

in similar intermediate nodes in RV-Elim. The effectiveness of mini-bucket scheme is very sensitive to the treewidth of the graphical model, which makes it unsuitable for dense graphs.

Lifted MCMC

Niepert [2012a,b] propose to lift the MCMC by constructing *orbital Markov chains*. An orbital Markov chain operates on a symmetry-induced partition of the space of joint variable assignments instead of the much larger original state space (i.e., all possible worlds) therefore takes less computation. Automorphism groups of graphical models are constructed to approximate the ideal orbital Markov chain, which is in general intractable.

In principle, Lifted MCMC samples from a much smaller state space than propositional MCMC. It has the potential of reducing the mixing time significantly based on the model's level of symmetry. However, similar to its propositional cousin, lifted MCMC has many parameters need to be fine-tuned, and no theoretical guarantee is known on the mixing time.

First-Order Knowledge Compilation

Van den Broeck [2011] and Gogate and Domingos [2011] exploited the *liftability* of a relational model through first-order knowledge compilation. They proved that 2-WFOMC (weighted first-order model counting with up to 2 logical variables per formula) is domain liftable.

First-Order knowledge compilation provides a theoretical framework for identifying the liftability of a PRGM. However, only a very restrictive class of models was identified in existing work. The liftability, either exact or approximate, is still unknown form many popular models.

2.2.3 Learning

Learning the weight for MLN formulas is equivalent to learning in MRF, which corresponds to the ground network of the MLN. The target is to find the θ that maximizes the log-likelihood of data samples, as Eq (2.3). The difficulty for learning MRF resides in computing the log partition function $\log Z(\theta)$ or its derivative.

The most commonly used method is Markov chain Monte Carlo maximum likelihood estimator (MCMC-MLE or MC-MLE) [Geyer and Thompson, 1992]. Geyer and Thompson showed that MCMC-MLE is a consistent estimator for the derivative of log partition function, therefore an iterative EM algorithm can be used to estimate θ^*. The mixing time for the Markov chain depends on the model we are sampling from, and the effectiveness of MCMC-MLE on different models may be completely different.

Another popular approximation is to optimize the pseudo-likelihood [Besag, 1975] of the data instead of the real likelihood. The pseudo-likelihood, quite similar to the mean field method, uses a product distribution to approximate the real likelihood. Each factor in the product distribution is the likelihood of a single random variable conditioned on its Markov blanket[3].

[3]We assume the training data is fully observable.

Chapter 3

Learning Exponential Random Graph Models

3.1 Overview

Exponential random graphs models (ERGM) are common, simple statistical models for social network and other uncertain network structures. Unfortunately, inference and learning with them is hard even for small networks because their partition functions are intractable to compute precisely. ERGM practitioners usually resort to stochastic sampling methods. However, a variety of pathological behaviors, known as degeneracy, have been observed while fitting ERGMs to network data, causing difficulties in their applications. In this chapter, we introduce a new quadratic time deterministic approximation to the partition functions. Our main insight enabling this advance is that subgraph statistics are sufficient to derive a lower bound for partition functions when the model of interests is not dominated by a few graphs. The proposed method differs from existing methods in the way it exploits asymptotic properties of subgraph statistics. In comparison to current Monte Carlo simulation based methods, the new method is scalable, stable, and precise enough for inference tasks. Moreover, the derived lower bound reveals that an ERGM fitted with one network cannot be extrapolated to networks of different sizes.

3.2 Exponential Random Graph Models

Exponential random graph model (ERGM) [Lusher et al., 2012; Robins et al., 2007] is a family of statistical models commonly used by social network analysis practitioners. An ERGM defines a probabilistic distribution over all graphs of n nodes. Besides conventional node-wise attributes, features of an ERGM may include subgraph statistics (e.g., number of edges, triangles, and k-stars) [Robins et al., 2007]. The model captures the correlation between network sub-structures, which enables many interesting inference tasks [Simpson et al., 2011; Wyatt et al., 2008]. For example, we can tell whether transitivity is prominent in a network by fitting an ERG model with related subgraphs as features, such as triangles. Applications of ERG models can be found in social network analysis [Contractor et al., 2006; Goodreau et al., 2009; Wyatt et al., 2008, 2010] and cognition research [Simpson et al., 2011]. As we will show in Chapter 4, ERGMs are closely related PRGMs in terms of modeling uncertain networks.

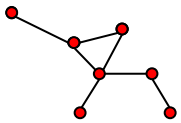
Fixing the number of nodes n, an ERGM defines a probabilistic distribution over all graphs with n nodes. More specifically, the probabilistic mass function for graph $g \in \mathcal{G}$ is:

$$p_\theta(g) = \frac{1}{Z(\theta)} \exp\left(\theta^T \phi(g)\right) \tag{3.1}$$

where $\phi(g)$ is the feature vector for graph $g \in \mathcal{G}$, parameter θ is a real vector. Partition function $Z(\theta)$ is a normalizing constant, which sums the potentials over $\mathcal{G}$:

$$Z(\theta) = \sum_{g \in \mathcal{G}} \exp\left(\theta^T \phi(g)\right) \tag{3.2}$$

The feature vector $\phi(g)$ is an array of functions on graph g that capture



feature	count	density
edge	7	0.333 (7/21)
triangle	1	0.029 (1/35)
2-star	11	0.105 (11/105)
3-star	5	0.036 (5/140)
rectangle	0	0 (0/70)

Figure 3.1: An example network of order $n = 7$ (In ERGM, edges are random variables). Table on the right shows the sufficient statistics (densities) for an ERGM with edge, triangle, 2/3-stars and rectangle as features.

the network properties of interests, such as network structural statistics and nodal attributes. Lusher et al. [2012] gives an extensive discussion on the features may be used. Here, what makes ERGM interesting for network modeling is the inclusion of network structural statistics (i.e., subgraph statistics).

In this work, we focus on undirected graphs and subgraph statistics features for simplicity. Specifically, for a set of subgraph structures of interests $\{L_1, \ldots, L_r\}$, the feature vector of undirected graph g can be defined with subgraph densities as below:

$$\phi(g) = \left(\frac{t(g, L_1)}{t(K_n, L_1)}, \frac{t(g, L_2)}{t(K_n, L_2)}, \ldots, \frac{t(g, L_r)}{t(K_n, L_r)} \right) \tag{3.3}$$

Here $t(g, L_i)$ counts the number of subgraphs in g that are isomorphic to L_i; K_n is the order-n complete graph, therefore $t(K_n, L_i) = \binom{n}{v_i} t(K_{v_i}, L_i)$ is a constant for any L_i of order v_i. Notice that the simplest subgraph K_2, or edge, is almost always included as a feature in ERGMs. Its role in the model is similar to that of the intercept term in most linear regression models [Hunter, 2007; Robins et al., 2007]. For the rest of the thesis, we assume K_2 is always included in the feature subgraphs. The edge statistics here play a role similar to the intercept in logistic regression [Robins et al., 2007].

Example: Figure 3.1 illustrates a simple example network of order 7. It has seven edges, one triangle, eleven 2-stars, five 3-stars and no rectangle.

The third column shows the subgraph densities of the network. For example, the 7-node labeled graph can have at most $\binom{7}{3} \times 1 = 35$ triangles, therefore the triangle density is $1/35 \simeq 0.029$.

Learning ERGMs from data is achieved through Maximum Likelihood Estimation (MLE). Given a network g, the MLE of parameter vector θ is:

$$\begin{aligned} \theta^* &= \operatorname*{argmax}_{\theta} \ell(\theta|g) \\ &= \operatorname*{argmax}_{\theta} \left\{ \theta^T \phi(g) - \ln Z(\theta) \right\} \end{aligned} \tag{3.4}$$

Here $\ell(\theta|g) = \log p_\theta(g)$ is the log-likelihood. In this thesis, we are interested in approximating the log partition function $\ln Z(\theta)$.

Unfortunately, such learning is hard even for networks of modest size (e.g., 40 nodes) because calculating normalizing constants (*partition functions*) precisely for such models is intractable. For this reason, most current techniques involve stochastic sampling [Handcock et al., 2003; van Duijn et al., 2009]. This often results in slow mixing time, and difficulties well known as "degeneracy" or "near degeneracy" phenomenon, in which the learned model tends to generate either empty or complete graphs [Bhamidi et al., 2008; Handcock, 2003; Hunter et al., 2012; Schweinberger, 2011; Snijders, 2002].

3.3 Approximating Log Partition Functions

In this section, we introduce a deterministic approximation to the log partition function $\ln Z(\theta)$. Figure 3.3 summarizes the approach we take.

$$\ln Z(\theta) \quad \xrightarrow[\text{ⓐ}]{\geq} \quad \max_u \gamma(\theta, u) \quad \xrightarrow[\text{ⓑ}]{\approx} \quad \max_u \tilde{\gamma}(\theta, u)$$

Figure 3.2: The algorithm has two approximations: ⓐ $\gamma(\theta, u)$ is an edge-count-u induced lower bound for $\ln Z(\theta)$, Lemma 3 shows the error is bounded in $O(\ln n)$; ⓑ We propose $\tilde{\gamma}(\theta, u)$ as an approximation to the unknown $\gamma(\theta, u)$, following Lemma 4 and Lemma 5.

3.3.1 Graph counting in the feature space

We introduce the key concept of graph counting function for the feature space of ERGM. Let $\phi : \mathcal{G} \to \mathcal{H}$ be the function that maps a graph in $g \in \mathcal{G}$ to the subgraph density space $\mathcal{H}$ of the graphs. For $\mathbf{h} \in \mathcal{H}$, we define *counting function* $\#(\mathbf{h}) = |\mathcal{G}_h|$ where $\mathcal{G}_h = \{g \in \mathcal{G} | \phi(g) = \mathbf{h}\}|$, i.e., the number of graphs in $\mathcal{G}$ having $\mathbf{h}$ as subgraph densities. We rewrite the partition function (3.2) into a compact form using counting function[1]:

$$Z(\theta) = \sum_{\mathbf{h} \in \mathcal{H}} \#(\mathbf{h}) \exp\left(\theta^T \mathbf{h}\right) = \sum_{\mathbf{h} \in \mathcal{H}} \exp\left(\theta^T \mathbf{h} + \ln \#(\mathbf{h})\right) \tag{3.5}$$

Notice that when $\theta = 0$, each term in (3.5) simply counts the graphs with given subgraph configuration, and the normalizing constant becomes the total number of graphs $|\mathcal{G}|$. Later we will show how the graph counting interpretation helps in computing $\ln Z(\theta)$.

Let $L_1, L_2, \ldots, L_r$ be simple graphs of interests and v_i be the number of nodes for L_i. The following lemma provides an upper bound to $|\mathcal{H}|$. Under the assumption $\forall i,\ n \gg v_i$ and $n \gg r$, the lemma establishes reasonable error bounds for several arguments in the rest of the paper:

Lemma 1. *For* $v* = \max\{v_1, \ldots, v_r\}$*, it holds that* $\ln |\mathcal{H}| \leq r v^* \ln n$.

Proof. Subgraph count for L_i in any g is bounded by $0 \leq t(g, L_i) \leq t(K_n, L_i) \leq$

[1]Note that all isomorphic graphs have the same subgraph densities, but the reverse is not true.

$\binom{n}{v_i} v_i!$, therefore

$$\begin{aligned}\ln|\mathcal{H}| &\leq \ln \prod_{i=1}^{r} t(K_n, L_i) \leq \ln\left[\prod_{i=1}^{r}\binom{n}{v_i} v_i!\right] \\ &\leq r \ln\left[\binom{n}{v^*} v^*!\right] = r \ln \frac{n!}{(n-v^*)!} \leq r v^* \ln n\end{aligned}$$

□

3.3.2 Approximate Log-Sum-of-Exponentials

In this section, we introduce a widely used computational trick. Given some set $\mathbf{S}$, and any function $f : \mathbf{S} \to \mathbf{R}$, formula of the form $\ln \sum_{x \in S} \exp f(x)$ can be approximated by $\max_{x \in \{\mathbf{S}\}} f(x)$ if $|S|$ is small. Specifically, we have the following upper and lower bounds:

Lemma 2. *Let f be a function on S and $x^* = \operatorname{argmax}_{x \in S} f(x)$, it holds that:*

$$f(x^*) \leq \ln \sum_{x \in S} \exp f(x) \leq f(x^*) + \ln|S|$$

Proof. On the lower bound: $f(x^*) = \ln \exp f(x^*) \leq \ln \sum_{x \in S} \exp f(x)$. On the upper bound: $\ln \sum_{x \in S} \exp f(x) \leq \ln|S| \exp f(x^*) = f(x^*) + \ln|S|$. □

Direct application of Lemma 2 to $\ln Z(\theta)$ yields a sloppy approximation because the huge size of $\mathcal{G}$. Thanks to Lemma 1, the following approximation to (3.5) has a much tighter error bound[2]:

$$\ln Z(\theta) = \max_{\mathbf{h} \in \mathcal{H}} \{\theta^T \mathbf{h} + \ln \#(\mathbf{h})\} + O(\ln n) \tag{3.6}$$

In next section, we discuss how to estimate the first term of (3.6).

[2]We will show later that $\ln Z(\theta)$ is in $O(n^2)$.

3.3.3 Edge-Count Induced Lower Bounds

In this section, we first introduce a edge-count induced lower bound $\gamma(\theta, u)$ to $\ln Z(\theta)$ using (3.6), then we develop an estimator this bound.

Let $\mathcal{G}_u \subset \mathcal{G}$ be the set of graphs with u edges, $\mathcal{H}_u \subset \mathcal{H}$ be the set of subgraph statistics induced by $\mathcal{G}_u$, and $\#_u(\mathbf{h})$ be the restricted counting function which only counts graphs in $\mathcal{G}_u$, i,e. $\#_u(\mathbf{h}) = |\{g \in \mathcal{G}_u | \phi(g) = \mathbf{h}\}|$. For any θ and u, we can define lower bound $\gamma(\theta, u)$ to $\ln Z(\theta)$ using (3.6):

$$\gamma(\theta, u) = \max_{\mathbf{h} \in \mathcal{H}_u} \{\theta^T \mathbf{h} + \ln \#_u(\mathbf{h})\} \leq \max_{\mathbf{h} \in \mathcal{H}} \{\theta^T \mathbf{h} + \ln \#(\mathbf{h})\} \tag{3.7}$$

Notice that the equality holds when K_2 is a feature subgraph and $u = \text{argmax}_u \gamma(\theta, u)$, because in this case $\mathcal{H}_u \cap \mathcal{H}_{u'} = \emptyset$ if $u' \neq u$, therefore $\{\mathcal{H}_u\}$ is a partition of $\mathcal{H}$. Specifically, we have the following lemma:

Lemma 3. *Given K_2 is included in subgraph features, the following equation holds:*

$$\begin{aligned} \max_u \{\gamma(\theta, u)\} &= \max_u \left\{ \max_{\mathbf{h} \in \mathcal{H}_u} \{\theta^T \mathbf{h} + \ln \#_u(\mathbf{h})\} \right\} \\ &= \max_{\mathbf{h} \in \mathcal{H}} \{\theta^T \mathbf{h} + \ln \#(\mathbf{h})\} = \ln Z(\theta) - O(\ln n) \end{aligned} \tag{3.8}$$

Lemma 3 shows that $\max_u \{\gamma(\theta, u)\}$ is a reasonably tight lower bound of $\ln Z(\theta)$. However, $\gamma(\theta, u)$ is still unknown. For the rest of the section, we develop an approximation of $\gamma(\theta, u)$ by exploiting the asymptotic property of $\#_u(\mathbf{h})$ in $\mathcal{G}_u$.

Let $\mathbf{h}'(\theta, u)$ and $\mathbf{h}^*(u)$ be the optimum of $\gamma(\theta, u)$ and maximizer of $\#_u(\mathbf{h})$

respectively:

$$\mathbf{h}'(\theta, u) = \operatorname*{argmax}_{\mathbf{h} \in \mathcal{H}_u} \{\theta^T \mathbf{h} + \ln \#_u(\mathbf{h})\}$$

$$\mathbf{h}^*(u) = \operatorname*{argmax}_{\mathbf{h} \in \mathcal{H}_u} \{\ln \#_u(\mathbf{h})\}$$

Then the following bounds of $\gamma(\theta, u)$ hold for all θ and u:

$$\begin{aligned} & \theta^T \mathbf{h}^*(u) + \ln \#_u(\mathbf{h}^*(u)) \\ \leq\ & \gamma(\theta, u) = \theta^T \mathbf{h}'(\theta, u) + \ln \#_u(\mathbf{h}'(\theta, u)) \\ \leq\ & \theta^T \mathbf{h}'(\theta, u) + \ln \#_u(\mathbf{h}^*(u)) \end{aligned} \tag{3.9}$$

The gap between the upper and lower bounds in (3.9) is a linear term $\theta^T(\mathbf{h}'(\theta, u) - \mathbf{h}^*(u))$. We argue that $\mathbf{h}^*$ can be used to approximate $\mathbf{h}'$ in terms of estimating the log partition function $\ln Z(\theta)$ if the model of interests is not dominated by a few graphs[3]. Compared to $\mathbf{h}'(\theta, u)$ and $\ln \#_u(\mathbf{h}'(\theta, u))$, $\mathbf{h}^*$ and $\ln \#_u(\mathbf{h}^*(u))$ are much easier to estimate, therefore lead to a feasible approximation to $\gamma(\theta, u)$.

Estimating $\mathbf{h}^*(u)$

$\mathbf{h}^*(u)$ maximizes the counting function $\#_u(\mathbf{h})$ on $\mathcal{G}_u$; therefore it is the mode of $\phi(g)$ for $g \in \mathcal{G}_u$. If we define a uniform distribution on $\mathcal{G}_u$, then $\mathbf{h}^*(u)$ is the mode of $\phi(\mathcal{G}_u)$.

The process of drawing graphs randomly from $\mathcal{G}_u$ is known as Erdös-Rényi (ER) random graphs model $G(n, M)$ [Erdös and Rényi, 1960]. Here n is the number of nodes in the graph and $M = u$ is the number of edges. An alternative (and more popular) definition of ER model is $G(n, p)$ [Gilbert, 1959], in

[3]In the cases where a few graphs are dominating the model, $\mathbf{h}'(\theta, u)$ will sway away from $\mathbf{h}^*(u)$.

which an order-n graph is constructed by drawing each edge independently with probability p.

Nowicki [Nowicki, 1989] proved that that $\phi(g)$ is asymptotically normally distributed for $g \in G(n,p)$. Since we partition the graph space with on edge count, the following lemma simply extends Nowicki's theorem from $G(n,p)$ to $G(n,M)$ over $\mathcal{G}_u$ using Chebyshev's inequality:

Lemma 4. *Let s_i be the edge count of L_i, define function $\rho_i(u) = (u/\binom{n}{2})^{s_i}$. Given any edge density μ, write the edge count $u = \binom{n}{2}\mu$ as a function of n. Then for any real vector $\mathbf{a} = (a_1, a_2, \ldots, a_r)^T$ and random graph*

$g \in G(n, M = u)$, the following holds as $n \to \infty$:

$$P\left(\left|\mathbf{a}^T(\phi(g) - \rho(u))\right| \geq \frac{1}{cn}\right) \to 0 \tag{3.10}$$

where $\rho(u) = (\rho_1(u), \ldots, \rho_r(u))^T$ and c is some constant.

The proof is available in Appendix A. Notice here $\rho_i(u)$ is the expected density of L_i in $G(n, p = u/\binom{n}{2})$. Lemma 4 suggests that the subgraph densities for most graphs of $\mathcal{G}_u$ are close to $\rho(u)$. In a sense, graphs in $\mathcal{G}_u$ forms a cluster in terms of the subgraph statistics. Figure 3.3 illustrates the property using order 12 unlabeled graphs [Brouwer]. Based on this property, we propose using $\rho(u)$ to approximate $\mathbf{h}^*(u)$ for large n.

Estimating $\ln \#_u(\mathbf{h}^*(u))$

Lemma 4 also hints using $|\mathcal{G}_u|$ to approximate $\#_u(\mathbf{h}^*(u))$ as $\phi(g)$ concentrates. With the help of Lemma 1, it turns out to be a very good estimation:

Lemma 5. *Given edge count u, it holds that*

$$\ln \#_u(\mathbf{h}^*(u)) = \binom{n}{2} H(u/\binom{n}{2}) - O(\ln n)$$

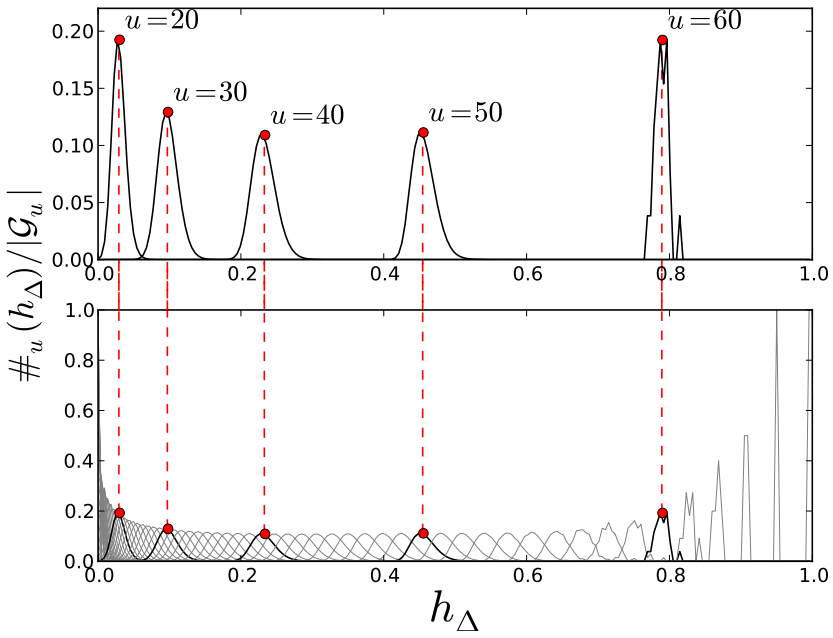


Figure 3.3: Concentration of triangle density h_Δ conditioned on the number of edges u for unlabeled graphs ($n = 12$). In this case, there are $\binom{12}{2} = 67$ possible edge counts. Y-axis measures the counting function $\#_u(h_\Delta)$ normalized by $|\mathcal{G}_u|$. Lower plot illustrates all 67 distributions; upper plot shows a subset for $u \in \{20, 30, 40, 50, 60\}$.

where $H(x) = -x \ln x - (1-x)\ln(1-x)$.

Proof. Let $\mathcal{G}_u$ be the set of graphs with edge count u, since $\mathbf{h}^*(u)$ is the maximizer, we have $\#(\mathbf{h}^*(u)) \geq |\mathcal{G}_u|/|\mathcal{H}|$. Since $\#(\mathbf{h}^*(u)) \leq |G_u|$, we have:

$$\ln|\mathcal{G}_u| - \ln|\mathcal{H}| \leq \ln \#(\mathbf{h}^*(u)) \leq \ln|\mathcal{G}_u| \tag{3.11}$$

Apply Stirling's approximation on $\ln|\mathcal{G}_u|$:

$$\begin{aligned} \ln|\mathcal{G}_u| = \ln\binom{\binom{n}{2}}{u} &\simeq (\binom{n}{2} - u)\ln\frac{\binom{n}{2}}{\binom{n}{2}-u} + u\ln\frac{\binom{n}{2}}{u} \\ &= \binom{n}{2} H(u/\binom{n}{2}) \end{aligned} \tag{3.12}$$

Together with Lemma 1, the claim follows. □

Estimating Lower Bound $\gamma(\theta, u)$

Given estimations $\mathbf{h}^*(u) \simeq \rho(u)$ and $\ln \#_u(\mathbf{h}^*(u)) \simeq \binom{n}{2} H\left(u/\binom{n}{2}\right)$, the simple approximation to $\gamma(\theta, u)$ follows immediately:

$$\tilde{\gamma}(\theta, u) = \theta^T \rho(u) + \binom{n}{2} H(u/\binom{n}{2}) \tag{3.13}$$

Here $\rho(\cdot)$ and $H(\cdot)$ are defined in Lemma 4 and Lemma 5 respectively. Lemma 4 hints that when $\mathbf{h}'(\theta, u)$ deviates away from $\mathbf{h}^*(u)$, $\ln \#_u(\mathbf{h}')$ will diminish rapidly as $\theta^T \mathbf{h}'(\theta, u)$ increases linearly. When the gradient of the linear term is small, $\mathbf{h}^*(u)$ tends to be a good approximation of $\mathbf{h}'(\theta, u)$. If the gradient is steep, $\mathbf{h}'(\theta, u)$ will lean towards the extreme entry in $\mathcal{H}_u$ that maximizes the linear term but leads to a minimal $\#_u(\mathbf{h}'(\theta, u))$, i.e., only one graph (or a few graphs) in $\mathcal{G}_u$ has feature vector $\mathbf{h}'(\theta, u)$, but it dominates all other graphs.

3.3.4 Approximation Algorithm

The estimation of edge-count induced lower bound (3.13) immediately leads to an approximation of $\ln Z(\theta)$: **Edge Count Search (ECS) approximation:**

$$\mathrm{ECS}(\theta, n) = \max_{0 \le u \le \binom{n}{2}} \left\{ \theta^T \rho(u) + \binom{n}{2} H(u/\binom{n}{2}) \right\} \tag{3.14}$$

$\rho(\cdot)$ and $H(\cdot)$ are defined in Lemma 4 and Lemma 5 respectively. Algorithm 1 reports a straightforward implementation of (3.14), which simply searches through all u to maximize $\tilde{\gamma}(\theta, u)$. Notice that the algorithm requires no extra parameters, which makes the ECS approximation very easy to apply compared to current MCMC sampling methods.

Assume the number of subgraph features $r \ll n$, the time complexity of Algorithm 1 is in $O(n^2)$, which is linear in terms of the number of random variables (i.e., edges) of the model, and quadratic in terms of the order of the network.

A straightforward approximation to the log-likelihood function $l(g|\theta)$ is to replace $\ln Z(\theta)$ with $ECS(\theta, n)$:

$$\ell_{ECS}(\theta \mid g) = \theta^T \phi(g) - ECS(\theta, n)$$

The decision of approximating $\mathbf{h}'$ with $\mathbf{h}^*$ in Section 3.3.3 leads to a simple algorithm. However, it complicates the error bound analysis, which seems to be beyond the scope of the paper. As $n \to \infty$, ECS approximation (3.14) converges to another closely related approximation proposed by Chatterjee and Diaconis [Chatterjee and Diaconis, 2011], who show that for certain θ the approximation converges to the true log partition functions. In next

section, we resort to experiments to verify the effectiveness of the proposed algorithm.

Algorithm 1 Our new ECS Approximation to the log partition function $\ln Z(\theta)$

Input: model parameter θ and number of nodes n

Output: ECS – the estimation of $\ln Z(\theta)$

Initialize $ECS \leftarrow -\infty$

for $u \leftarrow 0$ **to** $n(n-1)/2$ **do**

$\tilde{\gamma}(\theta, u) \leftarrow \theta^T \rho(u) + \binom{n}{2} H(u/\binom{n}{2})$

$ECS \leftarrow \max\{\tilde{\gamma}(\theta, u), ECS\}$

end for

3.4 Limited Expressiveness

We often expect to apply the ERGM learned from one network sample to other networks for predictions or statistical tests. For example, learning the model using a sub-network and apply the learned parameters to the complete network. However, analysis in Section 3.3.3 suggests that this may not be possible for existing ERGM specifications.

To see this, let $u^* = \operatorname{argmax}_u \gamma(\theta, u)$, we show that as $n \to \infty$, any fixed θ becomes irrelevant for $\max_u \gamma(\theta, u)$, since $u^*/\binom{n}{2}$ converges towards $1/2$, which implies $\mathbf{h}'(\theta, u^*)$ converges towards $\rho(u^*)$. Specifically, we have the following theorem:

Theorem 1. *Let* $u^* = \operatorname{argmax}_u \gamma(\theta, u)$ *with* θ *fixed,* $\mathbf{h}'(\theta, u^*)$ *converges to* $\rho(u^*)$ *asymptotically as* $n \to \infty$.

Proof. By definition $\mathbf{h}'(\theta, u^*)$ are densities ranging in $[0, 1]$, therefore the product $|\theta^T \mathbf{h}'(\theta, u^*)|$ is bounded by constant $\sum_i |\theta_i|$.

Let $\alpha = \max_u H(u/\binom{n}{2}) \approx H(1/2)$, Lemma 5 shows that

$$\ln \#_{u^*}(\mathbf{h}^*(u^*)) = \binom{n}{2}\alpha - O(\ln n) \approx \emptyset(n^2) \tag{3.15}$$

Because Eq (3.9) implies

$$\ln \#_{u^*}(\mathbf{h}^*(u^*)) - \sum_i |\theta_i| \leq \ln \#_{u^*}(\mathbf{h}'(\theta, u^*)) \leq \ln \#_{u^*}(\mathbf{h}^*(u^*))$$

as $n \to \infty$ we have:

$$\ln \#_{u^*}(\mathbf{h}'(\theta, u^*)) \to \ln \#_{u^*}(\mathbf{h}^*(u^*))$$

Let $\mathbf{b}$ be some real vector, assume there is some $\epsilon > 0$ so that as $n \to \infty$ we have:

$$|\mathbf{b}^T(\mathbf{h}'(\theta, u) - \rho(u^*))| \geq \epsilon$$

In this case, Lemma 4 implies $\ln \#_{u^*}(\mathbf{h}'(\theta, u^*)) \to 0$. However, given that $\ln \#_{u^*}(\mathbf{h}^*(u^*))$ is in $\emptyset(n^2)$, it contradicts with the definition of $\mathbf{h}'(\theta, u)$. Therefore, $\mathbf{h}'(\theta, u) \to \rho(u^*)$. □

The result of Theorem 1 implies that the effects of any fixed θ will diminish to a set of single dimensional functions $\rho(u^*)$ as n increases, and $\ln Z(\theta)$ converges to $ECS(\theta, n)$ as $n \to \infty$. The shifting of model behavior for different n is closely related to the instability of ERGM sufficient statistics [Schweinberger, 2011], and more recently the result of ERGM's inconsistency under sampling [Shalizi and Rinaldo, 2013]. The latter suggests the expressiveness of an ERGM is limited to networks of the same order.

The proof of Theorem 1 implies that θ in $O(n^2)$ is a necessary condition for

non-degenerate parameterization (Eq (3.15)). The question remains whether using raw subgraph counts as features, instead of subgraph densities defined in Eq (3.3), will help? Unfortunately, the answer is negative, as we will soon discuss in Section 5.2.

Although it is not likely to fix the fundamental limitation, this condition is useful for scoping point-wise MLE. Specifically, we can rewrite θ as a function of n and meta-parameter vector ξ:

$$\theta(n, \xi) = \binom{n}{2}\xi \tag{3.16}$$

Eq (3.16) is helpful for identifying reasonable parameter range.

3.5 Handling Complex Features

So far, our algorithms have covered simple subgraph features, such as triangles and k-stars. These features have been widely used in many network-modeling tasks. However, experimental results [Handcock, 2003] have shown that MCMC-based parameter estimates using these features often yield ill-behaved models, which leads to degenerate graph distributions. To counter this empirical difficulty, Snijders et al. [2006] introduced a series of new feature specifications. The new specifications follow the pattern of combining a whole family of subgraphs into one parameterized feature, which sums over the geometrically weighted statistics of the subgraphs with alternating signs. Empirically, ERGMs with these new features have shown better statistical properties then using simple features along when the inference is carried out using MCMC-based methods [Hunter, 2007; Snijders et al., 2006].

In this section, we show how to convert these complex features into simple

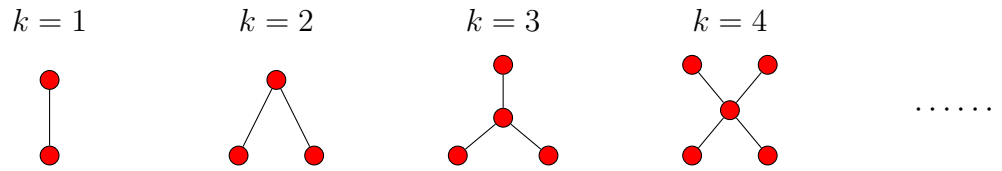


Figure 3.4: Nodal degree statistics are equivalent to k-stars

subgraph features that fit into Algorithm 1. We use *geometrically weighted degrees* as example, but the same procedure also applies to other complex features, such as *alternating k-triangles* or *alternating independent two-paths*.

For network of size n, the maximum degree is $n-1$. Let $d_k(g)$ be the number of nodes with degree k, the geometrically weighted degrees can be formally defined as:

$$\mathrm{GWD}_\alpha(g) = \sum_{k=0}^{n-1} \exp\{-\alpha k\} d_k(g)$$

where α is a control parameter.

Let L_k be k-stars (as shown in Figure 3.4), we can rewrite $\mathrm{GWD}_\alpha(g)$ using subgraph statistics as $\mathrm{GWD}_\lambda(g)$:

$$\begin{aligned}\mathrm{GWD}_\lambda(g) &= t(g, L_2) - \frac{t(g, L_3)}{\lambda} + \frac{t(g, L_4)}{\lambda^2} - \cdots + (-1)^{n-2}\frac{t(g, L_{n-1})}{\lambda^{n-3}} \\ &= \sum_{k=2}^{n-1}(-1)^k \frac{t(g, L_k)}{\lambda^{k-2}} \end{aligned} \tag{3.17}$$

It is easy to show that when $\lambda = e^\alpha/(e^\alpha - 1)$ we have:

$$\mathrm{GWD}_\lambda(g) = \lambda^2 \mathrm{GWD}_\alpha(g) + 2\lambda t(L_1) - n\lambda^2$$

Assuming θ_{GWD} is the coefficient corresponds to feature $\mathrm{GWD}_\lambda(g)$, we would like to represent the term $\theta_{GWD}\mathrm{GWD}_\lambda(g)$ using k-star subgraph fea-

tures $\phi(g)$ (Eq (3.3)) so that:

$$\theta^T \phi(g) = \theta_{GWD} \mathrm{GWD}_\lambda(g)$$

By substituting $\phi(g)$ and $\mathrm{GWD}_\lambda(g)$ with Eq (3.3) and Eq (3.17) respectively, we have:

$$\begin{aligned}
& \theta^T \phi(g) = \theta_{GWD} \sum_{k=2}^{n-1} (-1)^k \frac{t(g, L_k)}{\lambda^{k-2}} \\
\Rightarrow \quad & \frac{\theta_k t(g, L_k)}{t(K_n, L_k)} = \frac{(-1)^k t(g, L_k)}{\lambda^{k-2}} \theta_{GWD} \qquad (2 \leq k \leq n-1) \\
\Rightarrow \quad & \theta_k = \frac{(-1)^k}{\lambda^{k-2}} t(K_n, L_k) \theta_{GWD} \qquad (3.18)
\end{aligned}$$

Given Eq (3.18), we can easily convert the geographically weighted degrees in an ERGM into a series of k-stars. Notice that the maximum subgraph count $t(K_n, L_k)$ in Eq (3.18) converts the raw subgraph counts into subgraph densities as defined in Eq (3.3). Because the difference between $t(K_n, L_k)$ for different k can be huge, the compound effect of $K(K_n, L_k)$ and λ is hard to interpret.

3.6 Experimental Results

In this section, we evaluate ECS approximation using two tasks: estimating log-likelihood and MLE.

ECS approximation only makes sense for large n because of the underlying asymptotic properties. However, computing $\ln Z(\theta)$ exactly is not practical for $n > 8$. Instead, we resort to comparing the output of ECS with MCMC sampling algorithm commonly used in ERGMs: Bridge Sampling (BR) [Gelman and Meng, 1998; Handcock et al., 2003; Hunter et al., 2012].

The sampling algorithm is implemented in the *statnet* package [Handcock et al., 2003], and has been widely used. *statnet* also provides routines for sampling graphs directly from a given ERGM, which we used for generating synthetic data set. We implement the ECS approximation and MLE-ECS in Python. We use the common triad model (edge, 2-star, triangle) for the experiments on synthetic. For case study on Kapferer data, we also implemented the edge + GWD model.

Slow mixing time and degeneracy has long been troubling stochastic sampling methods on ERGM [Bhamidi et al., 2008; Handcock, 2003], and frequently result in unrealistic log-likelihood estimations. To identify some of these exceptions during experiments, we introduce the following trivial upper bound to the log-likelihood function (UB test):

$$\ell(\theta|g) \leq \theta^T \phi(g) - \max\{0, \sum_{i=1}^{r} \theta_i\} \tag{3.19}$$

The bound holds for any θ and g, because $\ln Z(\theta)$ is larger than the log potentials of empty graph, which is 0, and of complete graph, which is $\sum_{i=1}^{r} \theta_i$. By design, ECS will never generate log-likelihood estimates that exceed this upper bound, because for any θ, we have $\gamma(\theta, 0) = \tilde{\gamma}(\theta, 0)$ and $\gamma(\theta, \binom{n}{2}) = \tilde{\gamma}(\theta, \binom{n}{2})$. We apply this test to all log-likelihoods estimated using sampling.

3.6.1 Estimating log-likelihood functions

In this experiment, we sample synthetic networks from triad models of various parameters and compare the likelihood of the sample. As discussed in Section 3.4, we first generate a $6 \times 6 \times 6$ grid of ξ ranging from (-5.0, -5.0, -5.0) to (5.0, 5.0, 5.0), then generate θ using $\theta(n, \xi) = \binom{n}{2}\xi$. After dropping ξ in which all

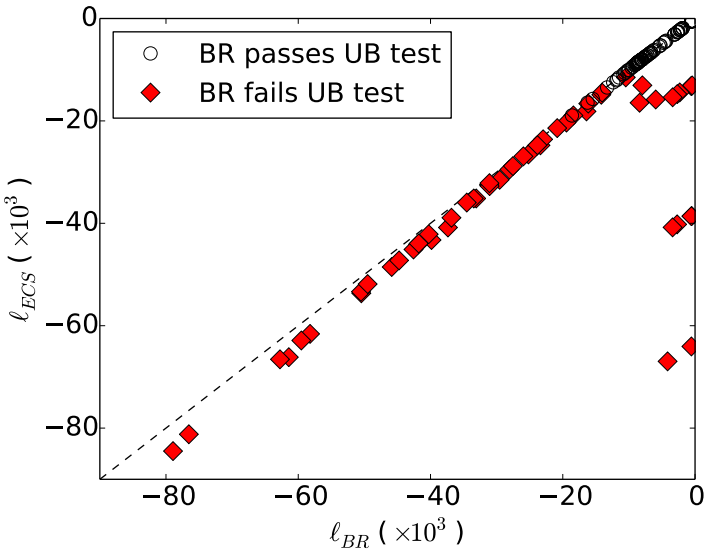


Figure 3.5: Scatter plot of log-likelihood estimations for ECS and BR on networks of $n = 160$. Many BR estimations fail UB test (3.19). Otherwise, ECS and BR estimations are very close (top right).

three numbers have the same sign, we ended up with 162 different ξs. Then for each ξ, we generate θ for different $n \in \{30, 40, 60, 80, 100, 120, 140, 160\}$. For each θ, one graph is sampled. The total number of graphs is 1,296. We estimate the log-likelihood for each sampled network using both Bridge Sampling and ECS.

Figure 3.5 reports the scatter plot of the results of both methods for $n = 160$. Points close to the dashed line suggest ECS and BR produce similar results; Points far away from the dashed line suggests the estimation results are very different. For estimations of BR, we also check whether it could pass the UB test (3.19). If the estimation exceeds the upper bound, we mark the data point with cross ($\times$); Otherwise, we mark with blue circle.

From 3.5 we can tell that when BR estimation passes the UB test, the difference between ECS and BR results are almost negligible. However, there was a significant portion (about 30%) of BR estimations failed the UB test.

We also compare the relative difference between BR and ECS results

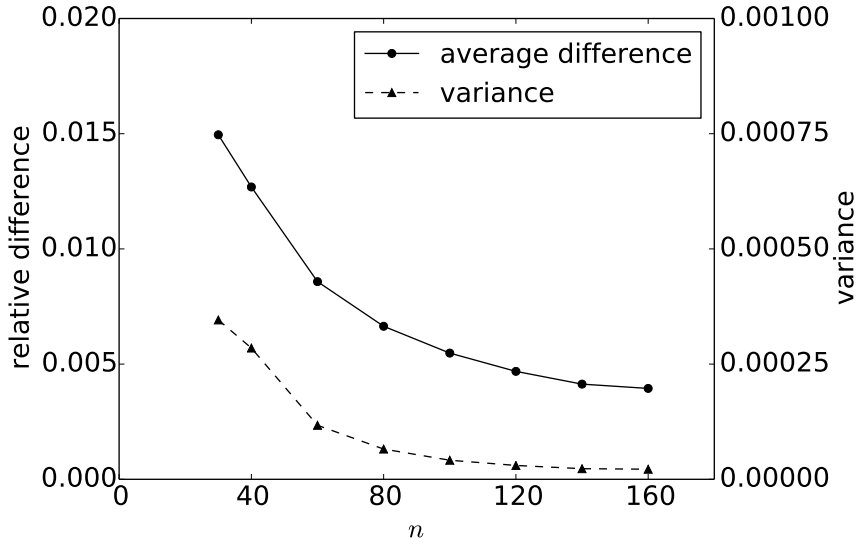


Figure 3.6: Relative difference between the estimations of ECS and BR for different n, given BR estimation passes the UB test (3.19). x-axis is the order of the network, y-axis measures the mean and variance of relative differences.

when BR result passes UB test: reldiff $= |(\ell_{ECS} - \ell_{Bridge})/\ell_{Bridge}|$. Figure 4.2 reports the mean and variance of the relative difference for different n. The plot shows both the mean and variance decrease as n increases.

3.6.2 MLE estimation

In MLE estimation, we leverage point-wise MLE by searching through a range of parameters (ECS-MLE). We first perform a case study on a real world social network data set, and then evaluate ECS-MLE on synthetic data set.

Case Study

The *kapferer* network [4] consists of 43 nodes and 190 edges. It is a simplified representation of social interactions happened in a tailor shop in Zambia [Kapferer, 1972]. We evaluate the quality of the fitted models by comparing

[4]The dataset is included in *statnet* package.

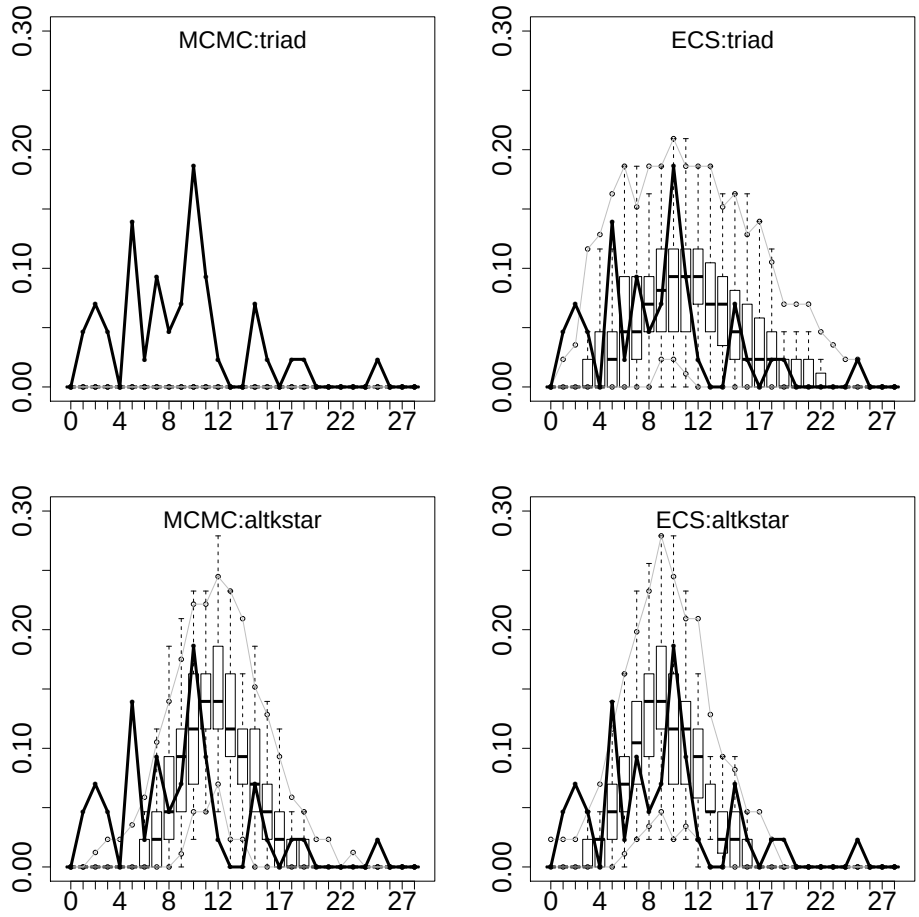


Figure 3.7: Node degree distributions for Kapferer network (solid line) and 100 simulated networks from each of the four fitted ERGMs (error bars). x-axis is the degree of nodes ($x \in [0, 1, 2, \ldots, 43]$) and y-axis is the proportion of nodes have the given degree ($y \in [0, 1]$). The models we used are *triad* (edge + kstar(2) + triangle, first row) and *altkstar* (edge + altkstar(1.5), second row). MCMC-MLE on triad model results in degeneracy, with all simulated networks being complete. In contrast, ECS-MLE fitted a reasonably good triad. On the other hand, altkstar models are known to be more friendly to MCMC-MLE, and both methods are able to fit reasonable altkstar models.

the node degree distribution between the original network and other networks generated from the trained model [Hunter et al., 2008]. Figure 3.7 reports the results, where the solid line illustrates the distribution of original *kapferer* network and the error bars are generated based on 100 samples generated from the fitted models.

The first model we used is the triad mode, which is notorious for degeneracy [Hunter et al., 2008, 2012]. This phenomenon is captured in the first plot of Figure 3.7, of which all nodes of sampled networks from the MCMC-MLE learned model have zero degrees. On the other hand, ECS-MLE successfully learned a reasonably good model, as shown in the second plot of Figure 3.7.

The second model we used is edges + altkstar($\lambda = 1.5$) (see Section 3.5). MCMC-MLE is in general more stable when fitting models with altkstar features, therefore we expect to be able to fit non-degenerate models. The third and fourth plots of Figure 3.7 show that both ECS-MLE and MCMC-MLE are able to fit reasonable models, while ECS-MLE result is slightly better.

To make sure the convergence of MCMC-MLE, we perform convergence diagnosis for both models. Figure 3.8 reports the convergence diagnostic plots for MCMC-MLE on edge+altkstar(1.5) model. The plots on the left show that the statistics of both edges and alternative k-stars for simulated networks converge to stable distributions. The plots on the right are the histograms for both statistics. Figure 3.9 reports the diagnosis for triad model. Although the plots indicate the Markov chain converged, it clear shows that the model is degenerated.

Figure 3.11 shows the simulated networks from models learned with ECS-MLE and MCMC-MLE respectively. We set the starting state of the simulation as a randomly sampled network with 43 nodes, and set edge density to 0.5. The burn-in for the simulation is set to 50000.

Synthetic Data Set

To create synthetic data set, we first generated a $6 \times 6 \times 6$ grid of ξ ranging from $(-3.0, -3.0, -3.0)$ to $(3.0, 3.0, 3.0)$. We then fix $n = 60$ and generate θ

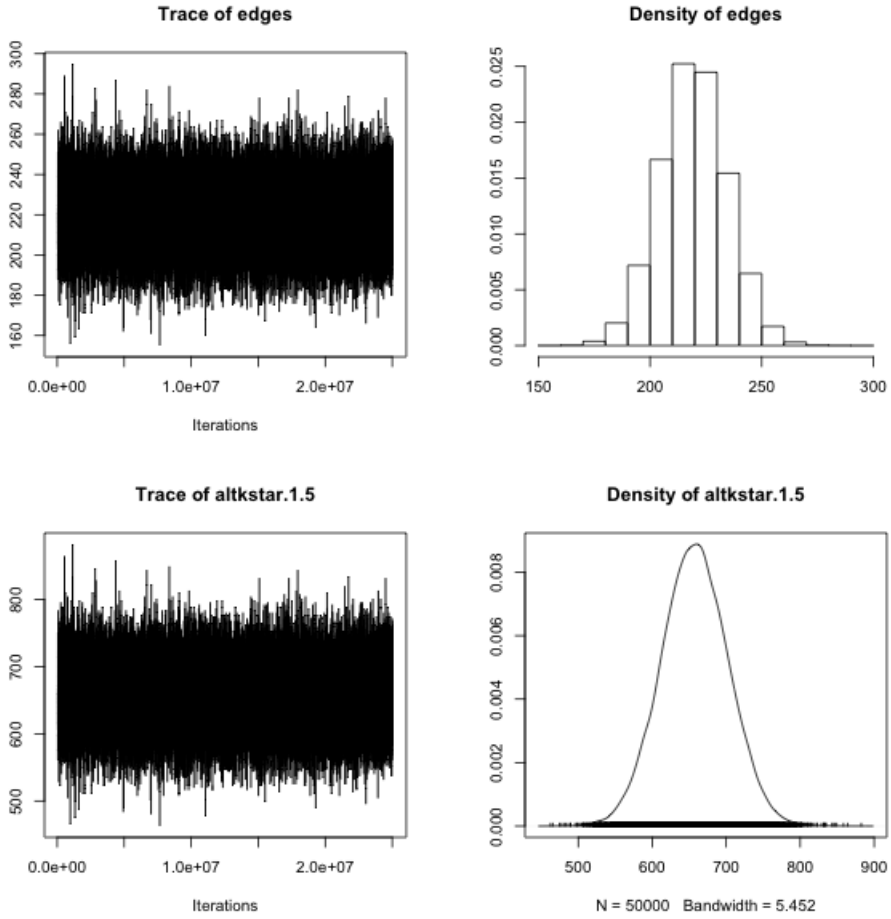


Figure 3.8: MCMC convergence diagnostic for edge+altkstar(1.5) model fitted using Kapferer2 network data. The left plots show that the Markov chain converges to a stable distribution. The right plots show the histograms of the two statistics: densities of edge and alternating k-stars

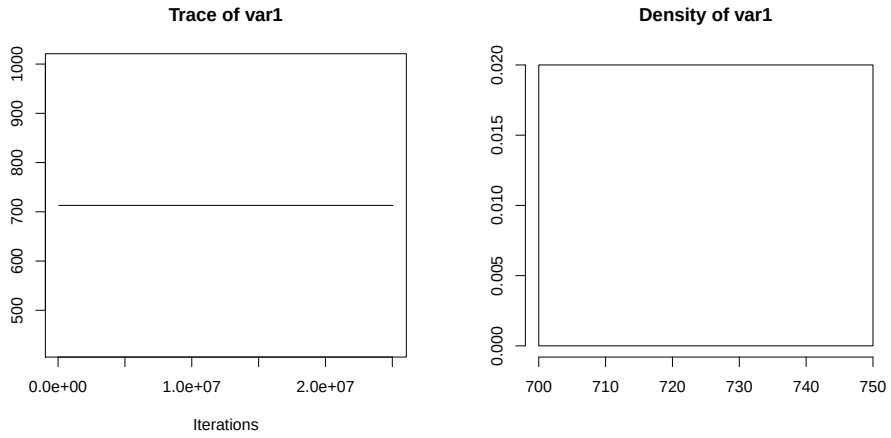


Figure 3.9: MCMC convergence diagnostic for the triad model fitted using Kapferer2 network data. The straight line in the left plot shows that the samples converge to some constant, clearly indicates the model is degenerated.

for each ξ. For each θ, one network is sampled using triad model. Then we fit the triad model with the sampled network using both MCMC-MLE and ECS-MLE. For ECS-MLE, we performed grid search in a slightly enlarged parameter space with finer granularity.

To evaluate, we estimate the log-likelihood of the network on both fitted models using Bridge Sampling. Figure 3.10 reports the scatter plot for the estimated log-likelihood for both ECS-MLE and MCMC-MLE. It shows that ECS-MLE is able to produce estimations as good as MCMC-MLE in most case. Notice that in many cases, as shown in Section 3.6.1, the sampling algorithm is very prone to generate unrealistic estimations.

3.7 Related Work

Social network structural modeling has been actively studied in machine learning community. Latent variable models, such as matrix factorization [Hoff, 2008], block modeling [Airoldi et al., 2008; Ho et al., 2012; Kemp

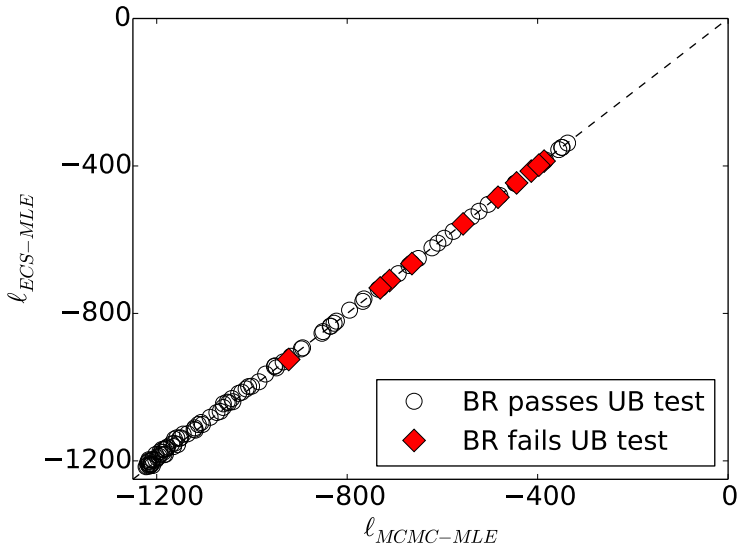


Figure 3.10: Scatter plot of adjusted log-likelihood for MCMC-MLE and ECS-MLE estimations on networks of $n = 60$. ECS-MLE outperforms MCMC-MLE in all trials (all points are on the left of the dashed line). BR failed UB test (3.19) in many trials that MCMC-MLE and MCS-MLE significantly disagree.

et al., 2006b] and others [Lloyd et al., 2012; Miller et al., 2009], represent the relational data with latent variables. Among those, Ho et al. [2012] proposed triangular motifs as network representation, which is closely related to ERGM's subgraph features.

Computing normalizing constants for complex and high-dimensional models, such as ERGMs, is intractable. Markov chain Monte Carlo simulations are arguably among the most effective methods. Gelman and Meng [1998] proposed the path sampling formulation to unify acceptance ratio method and thermodynamic integration from theoretical physics for estimating the (ratios of) normalizing constants. Annealed importance sampling (AIS) [Neal, 2001], which is popular in deep learning literature [Salakhutdinov and Murray, 2008], can also be viewed as one form of thermodynamic integration. Although effective in many applications, Bhamidi et al. [2008] shows that the mixing time for any local Markov chain in low temperature regimes

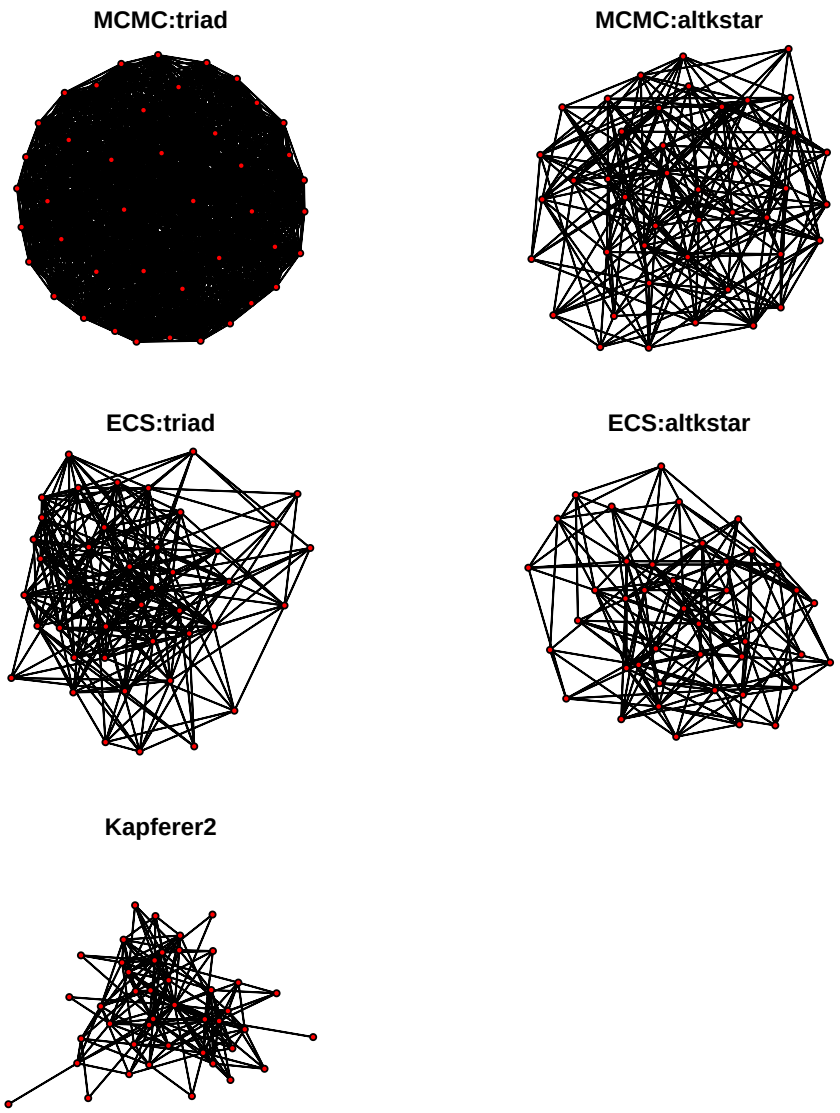


Figure 3.11: Networks simulated from the four models fitted on Kapferer2 network. Networks in the first row were fitted using MCMC-MLE, networks in the second row were fitted using ECS-MLE, and the original Kapferer network is plotted in the third row for reference.

of ERGMs is exponentially slow, rendering these methods computationally intractable in many cases. In comparison, ECS approximation is deterministic, therefore avoids the sampling completely.

ECS approximation is a variational inference algorithm. In this category, there are many other techniques, such as pseudo-log-likelihood [Strauss and Ikeda, 1990], mean field approximation and Bethe approximation [Wainwright and Jordan, 2008]. In the context of ERGM, these methods have been reported to be inferior to sampling based methods [van Duijn et al., 2009], and are usually used to generated initial states for sampling based algorithms [Hunter et al., 2012]. ECS distinguishes from others by exploiting the asymptotic property in the feature space of the model. This macroscopic view goes beyond the conditional independence in local structures of the model, and may be more effective for complex high-dimensional models like ERGMs.

ECS approximation is closely related to the work of Chatterjee and Diaconis [2011]. They apply large derivation principle results on Erdös-Rényi model to derive an analytic approximation to the log-likelihood function of ERGM with non-negative/non-positive parameters for subgraphs (with the exception of K_2). In fact, as $n \to \infty$, (3.14) converges to their result. Compared to Chatterjee and Diaconis, ECS approximation relies on much weaker conditions, therefore more flexible from the algorithmic perspective.

Lots of efforts have been put on understanding ERGM model, especially the degeneracy phenomenon. Rinaldo et al. [2009] experimentally characterized the behavior of degeneracy. Schweinberger [2011] suggests that subgraph counts are not stable sufficient statistics, which lead to the degeneracy of MCMC. Recently, Shalizi and Rinaldo [2013] proved that ERGM is inconsistent under sampling; therefore a model learned from our sub-network could not be extrapolated to the whole network. Our result shows the sim-

ilar expressiveness limitation for ERGM from a computational perspective, yet our experiments show that degeneracy may be avoidable through more stable inference algorithms.

3.8 Conclusion

In this work, we propose a new deterministic approximation to the log partition functions of ERGMs. Computing the partition functions (or the ratio of them) is essential in learning ERGMs. Our experimental results show the new method is able to overcome some of the stability issues faced by sampling based methods without losing accuracy. The new algorithm does not depend on extra parameters, making it easy to implement and apply compared to sampling.

ERGM is popular for social network analysis applications due to its simplicity and flexibility. The model itself, however, is yet to be fully understood. We show from a computational perspective that the behavior of an ERGM varies for networks of different sizes. In the future we would like to explore the implications of the difficulty and possible remedies.

Chapter 4

Approximate Lifting For Structural Relational Knowledge

4.1 Overview

Probabilistic relational graphical models (PRGMs) [McCallum et al., 2009; Nilsson, 1986; Poole, 2003; Richardson and Domingos, 2006] are powerful tools for combining first-order logic and graphical models, where the former is good at representing complex relational knowledge, the latter shines at capturing uncertainty in the system. Markov logic network (MLN) [Richardson and Domingos, 2006], for example, is one of the popular representations that fall in this category. In general, a PRGM can be treated as a template for generating probabilistic graphical models. Applications of these models can be found in natural language processing, social network analysis, etc. More details about PRGM can be found in Section 2.1.

The straightforward approach for carrying out inference on a PRGM is to first ground the model into a propositional graphical model, and then enlist propositional probabilistic inference algorithms for heavy lifting. In many cases, the size of the grounded propositional model is exponential in the size of the domain of interests; therefore this naïve approach is usually incapable of handling domains of even moderate size. Lifted inference [De Salvo Braz et al., 2005; Poole, 2003; Singla and Domingos, 2008] is an endeavor towards lifting the inference from propositional level to first-order level. Such lifting can often brings a significant performance gain. A survey of existing lifted

inference algorithms can be found in Section 2.2.

Many existing lifted inference algorithms root in the idea of grouping exchangeable random variables [Carbonetto et al., 2005; Choi and Amir, 2012; De Salvo Braz et al., 2005; Van den Broeck, 2011]. However, the exchangeability usually requires certain degree of isolation among the random variables of interests, which limits the applicability of these algorithms.

The logic formulas of structural relational knowledge often involve non-trivial interactions among the predicates and logic variables, which often leads to the non-exchangeable random variables. One remarkable case is the transitive relation:

$$\forall \mathtt{x}, \mathtt{y}, \mathtt{z}\ \mathtt{Fr}(\mathtt{x}, \mathtt{y}) \wedge \mathtt{Fr}(\mathtt{x}, \mathtt{z}) \Rightarrow \mathtt{Fr}(\mathtt{x}, \mathtt{z}) \tag{4.1}$$

In network analysis, the transitive relation is commonly included in relational knowledge base for modeling network transitivity, yet no exact lifted inference algorithm is known to work in its presence [Jaeger and Van den Broeck, 2012]. Several approximate lifted inference algorithms have been proposed [Kersting et al., 2009; Niepert, 2012a; Singla and Domingos, 2008], which are able to take transitive relations as input. However, there is no guarantee on the quality of the approximation.

This work takes a graph-theoretic perspective to investigate lifted inference on PRGM with structural relational knowledge. We first build connection between exponential-family random graph models (ERGMs) [Robins et al., 2007], showing how certain PRGMs with transitive relation can be converted into ERGMs. Then we leverage ECS approximation (see Chapter 3) to derive a deterministic approximate lifting algorithm. The new algorithm takes a macroscopic view towards lifting. It generalizes the idea of counting

elimination [De Salvo Braz et al., 2005] to focus on the statistics of states sharing the same feature vectors. 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.

For the rest of the chapter, we first define the problem in Section 4.2, and then we discuss the generalization of counting elimination in Section 3.3.1, which gives a high level view of the method. We show how to convert a special class of MLN into ERGMs in Section 4.4, and how to leverage ECS approximation to perform lifted inference on transitive relations in Section 4.5. Lastly, Section 4.8 concludes the chapter.

4.2 Problem Definition

In this thesis, we focus on lifting the inference for Markov logic networks (MLN) with transitive relation formula. Given a set of logic sentence and weight pairs $(f_i, \theta_i)_{i=1}^m$, let $\theta = (\theta_1, \ldots, \theta_m)^T$ and $\mathbf{f} = (f_1, \ldots, f_m)^T$, an MLN specifies a probability distribution of the possible assignment x to all the ground atoms in the network:

$$p_\theta(x) = \frac{1}{Z(\theta)} \exp\{\theta^T N(\mathbf{f}, x)\}$$

Here $N(f_i, x)$ counts the number of groundings of logic formula f_i that evaluates to true in assignment x. $Z(\theta)$ is the normalizing constant (a.k.a. partition function):

$$Z(\theta) = \sum_{x \in \mathcal{X}} \exp\{\theta^T N(\mathbf{f}, \mathbf{x})\} \tag{4.2}$$

Inference on MLN is difficult due to computing the sum-product problem over a huge state space. Particularly, to evaluate the log-likelihood for some assignment x:

$$\ell(\theta|x) = \theta^T N(\mathbf{f}, x) - \ln Z(\theta)$$

Given a training data set $\{x^{(1)}, x^{(2)}, \ldots, x^{(n)}\}$, the parameter vector θ can be fitted using maximum log-likelihood estimation (MLE):

$$\begin{aligned}\theta^* &= \text{argmax}_\theta \sum_{i=1}^{n} \ell(\theta|x^{(i)}) \\ &= \text{argmax}_\theta \left\{ \theta^T \sum_{i=1}^{n} N(\mathbf{f}, x^{(i)})/n - \ln Z(\theta) \right\}\end{aligned}$$

In both cases, being able to efficiently compute $\ln Z(\theta)$ is essential for inference task. In this chapter, we focus on lifting the computation of $\ln Z(\theta)$.

For simplicity, we consider a special class of MLNs, *Homogeneous Bivariate MLN*. In this case, we can focus on the complication of lifting caused by transitive relations.

Definition 2 (Homogeneous Bivariate MLNs)**.** *A Homogeneous Bivariate MLN consist of logical formulas with only one predicate of arity 2.*

For this work, we only focus on formulas with one bivariate predicate. Note that in first-order logic, every formula can be rewritten into one with only bivariate predicates. More background information on MLN and its inference can be found in Chapter 2.

4.3 Generalizing Counting Elimination

To get an insight of the approach we take, we first revisit an existing lifted inference method. Lifted inference algorithms (e.g., [De Salvo Braz et al., 2005; Milch et al., 2008; Singla and Domingos, 2008]) aim to scale up the inference tasks by lifting the computation from propositional level to relational level. These methods take quite different approaches and are applicable to different models, but the ideas are similar: grouping multiple redundant computations into fewer steps while maintaining the same results as propositional inference. One strategy is to introduce a *counting function* for each group of redundant computation, and it has been successfully applied in many lifted inference algorithms [De Salvo Braz et al., 2005; Milch et al., 2008; Singla and Domingos, 2008]. In this section, we generalize this strategy for lifting the computation of $\log Z(\theta)$.

For problems in the form of (4.2), a counting-function-based approach has the following generalized form:

$$\sum_{x \in \mathcal{X}} \exp\left(\theta^T N(\mathbf{f}, x)\right) = \sum_{\mathbf{h} \in \mathcal{H}_\mathbf{f}} \#(h) \exp\left(\theta^T \mathbf{h}\right), \tag{4.3}$$

Where $\mathcal{H}_\mathbf{f} = N(\mathbf{f}, \mathcal{X})$ is the domain of sufficient statistics, and $\#(\mathbf{h}) = |\{x | N(\mathbf{f}, x) = \mathbf{h},\ x \in \mathcal{X}\}|$ is the counting function that counts the number of (redundant) configurations of a sufficient statistic $\mathbf{h}$. For relational probabilistic models, $|\mathcal{H}_\mathbf{f}|$ is usually significantly smaller than $|\mathcal{X}|$, therefore it is much easier to enumerate $\mathcal{H}_\mathbf{f}$. If we are able to find a proper set of configurations $\mathcal{H}_\mathbf{f}$ and an efficient way to evaluate $\#(\mathbf{h})$, the counting function will lead to significant speedup.

For example, consider the following simple example with three MLN sen-

tences on domain $\mathcal{D}$:

$$
\begin{aligned}
f_1: &\quad \texttt{Pred(x)} \wedge \texttt{Pred(y)} && \theta_1 \\
f_2: &\quad \texttt{Pred(x)} \wedge \neg\texttt{Pred(y)} && \theta_2 \\
f_3: &\quad \neg\texttt{Pred(x)} \wedge \neg\texttt{Pred(y)} && \theta_3
\end{aligned}
$$

De Salvo Braz et al. [2005] proposed the following counting function:

$$
\begin{aligned}
&\sum_{x\in\mathcal{X}} \exp\left(\theta_1 N(f_1,x) + \theta_2 N(f_2,x) + \theta_3 N(f_3,x)\right) \\
&= \sum_{\substack{c_0,c_1 s.t. \\ c_0+c_1=|\mathcal{D}|}} \binom{|\mathcal{D}|}{c_0} \exp\left(\theta_1 c_1^2 + \theta_2 c_1 c_0 + \theta_3 c_0^2\right) \qquad (4.4)
\end{aligned}
$$

In the new formulation, c_0 and c_1 are counting parameters for the number of 0 and 1 in grounded $\texttt{Pred(x)}$, they subject to the constraint $c_0 + c_1 = |\mathcal{D}|$. All grounded $\texttt{Pred(x)}$ with the same assignment are exchangeable. Here configurations in $\mathcal{H}_{\mathbf{f}}$ and the corresponding counting function can be easily represented using counting parameter c_0 and c_1:

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

Since $|\mathcal{H}_{\mathbf{f}}| = |\mathcal{D}|$ is significantly smaller than $|\mathcal{X}| = 2^{|\mathcal{D}|}$, Eq (4.4) brings an exponential speedup.

However, it is non-trivial to apply this counting parameter strategy to models with intermingled random variables without extensive grounding or expending the model [De Salvo Braz et al., 2005]. In this specific example, the availability of counting parameters is a result of completely non-constrained logic variables $\texttt{x}$ and $\texttt{y}$. For transitive relation (4.1), there is no obvious way for constructing counting parameters.

For the rest of the chapter, we take a graph interpretation for MLN, which eventually leads to an approximation of $\mathcal{H}_{\mathbf{f}}$ and $\#(\mathbf{h})$, so that Eq (4.3) can be applied for lifting the computation of $\ln Z(\theta)$.

4.4 Graph Interpretation for Markov Logic

In this section, we introduce a graph interpretation for homogeneous bivariate MLN, which reveals the relationship between MLN and exponential random graph model (ERGM, see Section 3.2 for details). Through out this chapter, we use the homogeneous bivariate MLN in Table 4.1 as our running example. It is irreflexive and asymmetric, and has the transitive relation (formula f_2).

Let objects in the domain $\mathcal{D}$ be nodes, and there is an edge between a and b if the ground atom $\texttt{Fr}(\texttt{a}, \texttt{b}) = 1$. Under the irreflexive relation and symmetric relation constraints (formula f_3 and f_4 in Table 4.1), each assignment $x \in \mathcal{X}$ to the grounded atoms can be treated as a unique undirected graph, with $n = |\mathcal{D}|$ nodes. Without ambiguity, we will refer to x as an MLN assignment and a graph interchangeably for the rest of the paper. Mapping more complex MLNs to graph representation is beyond the scope of the thesis, but it is possible by introducing dummy nodes and colored edges.

From a graph-theoretic perspective, it is easy to see the complex interactions introduced by the transitive relation: all the edges in the graph are random variables, and every edge is correlated with its neighbors through the formula; the edge distance between any two edges is at most one; each edge is associated with $n - 2$ groundings of transitive relation f_2. For the rest of the section, we show how to convert the feature vector $N(\mathbf{f}, x)$ to a linear combination of a set of subgraph statistics, which consequently builds the connection between MLN and ERGM.

Table 4.1: MLN with transitive relations

	Feature	Weight
f_1	$\neg\mathtt{Fr}(\mathtt{x},\mathtt{y})$	θ_1
f_2	$\mathtt{Fr}(\mathtt{x},\mathtt{y}) \wedge \mathtt{Fr}(\mathtt{y},\mathtt{z}) \Rightarrow \mathtt{Fr}(\mathtt{x},\mathtt{z})$	θ_2
f_3	$\mathtt{Fr}(\mathtt{x},\mathtt{x})$	$-\infty$
f_4	$\neg(\mathtt{Fr}(\mathtt{x},\mathtt{y}) \Leftrightarrow \mathtt{Fr}(\mathtt{y},\mathtt{x}))$	$-\infty$

4.4.1 Subgraph Features $\mathbf{L}^k$

Given a homogeneous bivariate MLN formula f with k logical variables, *subgraph features* $\mathbf{L}^k$ for f are graphs in vector notation $\mathbf{L}^k = (L_0^k, \ldots, L_{r_k}^k)^T$. Here, L_i^k is a graph of order-k, and L_i^k and L_j^k are not isomorphic for any $0 \leq i, j \leq r_k$. We say $L_i^k \preceq L_i^k$ if L_i^k is isomorphic to some subgraph of L_j^k. Note that $\preceq$ defines a partial order on $\{L_i^k\}$. For a homogeneous bivariate MLN $\{f_i, \theta_i\}_{i=1}^r$, we define *effective subgraph features* $\mathbf{f}_g$ as the set of all subgraph features that are not constant (i.e., empty graphs) and have non-zero coefficients.

For formula f and its subgraph feature L_i^k, we define $\pi(f, L_i^k)$ to be the number of times f is true in all order-k graphs that are isomorphic to L_i^k. We say $x \cong L_i^k$ if x is isomorphic to L_i^k, then we have

$$\pi(f, L_i^k) = \sum_{x \cong L_i^k} N(f, x) \tag{4.5}$$

also in vector format:

$$\pi(f, \mathbf{L}^k) = (\pi(f, L_0^k), \pi(f, L_1^k), \ldots, \pi(f, L_{r_k}^k))^T \tag{4.6}$$

Note that $\pi(f, \mathbf{L}^k)$ is defined using $N(f, x)$, but here the order of x is fixed to k. When k is small, which is usually the case, the computation cost of

$\pi(f, \mathbf{L}^k)$ is negligible.

For the rest of the section, we explain how to convert MLNs to ERGMs (see Section 3.2) using subgraph features on the example MLN in Table 4.1.

Converting $f_1 : \neg\texttt{Fr}(\texttt{x}, \texttt{y})$.

The subgraph features of f_1 are $\mathbf{L}^2 = (L_0^2, L_1^2)^T$; because f_1 has two logical variables: $\texttt{x}$ and $\texttt{y}$. Here, L_0^2 is an empty order-2 graph and L_1^2 has two nodes connected by an edge, or K_2, as illustrated below:

Here $\pi(f, L_i^2)$ counts the number of times f is true in all order-2 graphs that are isomorphic to L_i^2.[1] For L_0^2 and L_1^2 we have $\pi(f_1, L_0^2) = 2$ and $\pi(f_1, L_1^2) = 0$. Given any graph x, we know $t(x, L_0^2) = \binom{n}{2}$, and $t(x, L_0^2)$ is the number of edges in x. Because $L_0^2 \preceq L_1^2$, the counting $t(x, L_0^2)$ includes $t(x, L_1^2)$, therefore we need to exclude the latter when computing their contribution to formula counting. More specifically, $N(f_1, x)$ can be represented as:

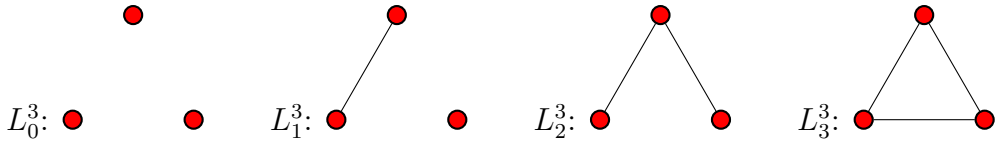
$$\begin{aligned} N(f_1, x) =& \pi(f_1, L_0^2) \left(t(x, L_0^2) - t(x, L_1^2)\right) \\ &+ \pi(f_1, L_1^2) t(x, L_1^2) \\ =& 2\binom{n}{2} - 2t(x, L_1^2) \end{aligned}$$

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

Converting $f_2 : \texttt{Fr}(\texttt{x}, \texttt{y}) \wedge \texttt{Fr}(\texttt{y}, \texttt{z}) \Rightarrow \texttt{Fr}(\texttt{x}, \texttt{z})$.

The translation of formula f_2 requires more consideration because it has three logical variables: $\texttt{x}$, $\texttt{y}$ and $\texttt{z}$. In this case, the subgraph features are $\mathbf{L}^3 = (L_0^3, L_1^3, L_2^3, L_3^3)^T$, as illustrated below:

[1] Note that $\mathbf{L}^k$ and $\pi(f_i, \mathbf{L}^k)$ are only computed once regardless of x.



Graphs isomorphic to L_0^3, L_1^3 or L_3^3 satisfy f_2 regardless of assignments from logical variables to nodes. After counting all permutations, we have $\pi(f_2, L_0^3) = \pi(f_2, L_1^3) = \pi(f_2, L_3^3) = 6$. Graphs isomorphic to L_2^3 do not satisfy f_2 when there is no edge between `x` and `z` (i.e., $\neg\texttt{Fr}(\texttt{x}, \texttt{z})$). Excluding these two cases, we get $\pi(f_2, L_2^3) = 4$. Put them together, we have:

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

Here $\mathbf{L}^3 = (L_0^3, L_1^3, L_2^3, L_3^3)^T$.

Let W^3 be the set of all induced order-3 subgraphs of the graph x, so we have $|W^3| = \binom{n}{3}$. Let $W_j^3 \subset W^3$ be the subset that are isomorphic to L_j^3, then we can rewrite $N(f_2, x)$ as follow:

$$N(f_2, x) = \sum_{y \in W^3} N(f_2, y) = \sum_{0 \le j \le 3} \pi(f_2, L_j^3)|W_j^3| \tag{4.7}$$

The problem here is to compute $|W_j^3|$ using subgraph features $\mathbf{L}^3$, and graph isomorphic counts $t(x, \mathbf{L}^3)$ (see Section 3.2).

$|W_3^3| = t(x, L_3^3)$ is trivial because they both count the same thing: number of triangles in x. To compute $|W_2^3|$, we need to exclude the counts of W_3^3 from $t(x, L_2^3)$ since $L_2^3 \preceq L_3^3$ and each occurrence of L_3^3 is counted as $t(L_3^3, L_2^3) = 3$ times occurrences of L_2^3 in x, which leads to $|W_2^3| = t(x, L_2^3) - 3|W_3^3|$. Similarly, we have $|W_1^3| = t(x, L_1^3) - 2|W_2^3| - 3|W_3^3|$ and $|W_0^3| = t(x, L_0^3) -$

$|W_1^3| - |W_2^3| - |W_3^3|$. Put everything together we get:

$$\begin{pmatrix} |W_0^3| \\ |W_1^3| \\ |W_2^3| \\ |W_3^3| \end{pmatrix} = \begin{pmatrix} 1 & -1 & 1 & -1 \\ 0 & 1 & -2 & 3 \\ 0 & 0 & 1 & -3 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} t(x, L_0^3) \\ t(x, L_1^3) \\ t(x, L_2^3) \\ t(x, L_3^3) \end{pmatrix}$$

Therefore we can rewrite Eq (4.7) as follow:

$$\begin{aligned} N(f_2, x) &= \sum_{0 \le j \le 3} \pi(f_2, L_j^3)|W_j^3| \\ &= \left(\pi(f_2, \mathbf{L}^3)^T \mathbf{A}\right)\ t(x, \mathbf{L}^3) \\ &= (6,\ 0,\ -2,\ 6)\ t(x, \mathbf{L}^3) \\ &= 6t(x, L_0^3) - 2t(x, L_2^3) + 6t(x, L_3^3) \end{aligned}$$

Here $\mathbf{A}$ is a matrix:

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

Figure 4.1 summarizes the conversion process of $N(f_2, x)$.

Rewriting $Z(\theta)$

Given the subgraph statistics representation of $N(f_1, x)$ and $N(f_2, x)$, we are able to rewrite $Z(\theta)$ for the MLN in Table 4.1 below (note that $t(x, L_0^3) = \binom{n}{3}$

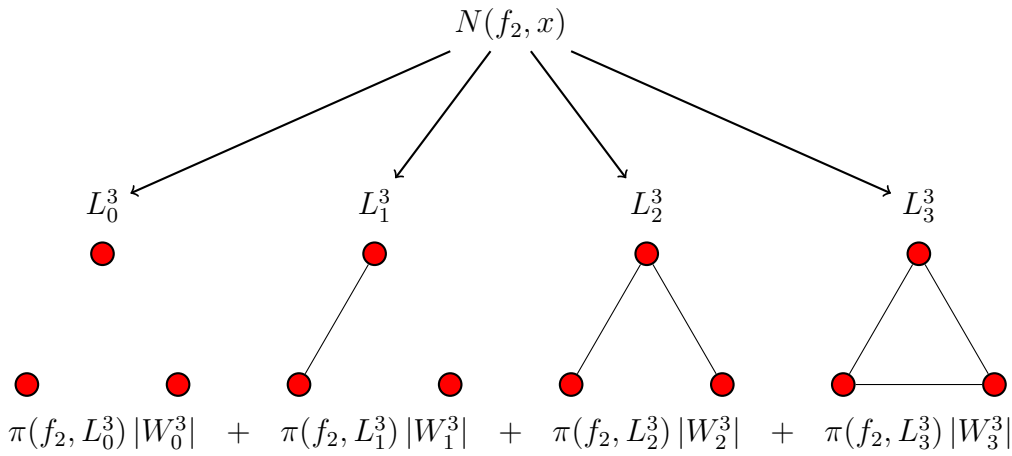


Figure 4.1: Converting MLN feature representation $N(f_2, x)$ into a function of subgraph statistics of x through subgraph features $\mathbf{L}^3$. Here $|W_j^3|$ are a linear functions of subgraph statistics, which can be computed by Theorem 2.

is a constant):

$$
\begin{aligned}
& Z(\theta) \\
=& \sum_{x \in \mathcal{X}} \exp\left\{\theta_1 \left(2\binom{n}{2} - 2t(x, L_1^2)\right) + \theta_2 \left(6\binom{n}{3} - 2t(x, L_2^3) + 6t(x, L_3^3)\right)\right\} \\
=& \exp\left\{2\binom{n}{2}\theta_1 + 6\binom{n}{3}\theta_2\right\} \\
& \sum_{x \in \mathcal{X}} \exp\left\{-2\theta_1 t(x, L_1^2) - 2\theta_2 t(x, L_2^3) - 6\theta_2 t(x, L_3^3)\right\} \\
=& c(\theta) \sum_{x \in \mathcal{X}} \exp\left(\theta'^T t(x, \mathbf{f}_g)\right) \qquad (4.8)
\end{aligned}
$$

Here effective subgraph features $\mathbf{f}_g$ are $(L_1^2, L_2^3, L_3^3)^T$, and the corresponding coefficients are $\theta' = (-2\theta_1, -2\theta_2, -6\theta_2)^T$.

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

Theorem 2. *Given homogeneous bivariate MLN formula f with k logical variables, and assignment $x \in \mathcal{X}$, $N(f, x)$ can be represented using subgraph*

features $\mathbf{L}^k = (L_0^k, \ldots, L_{r_k}^k)^T$ *as*

$$N(f, x) = \pi(f, \mathbf{L}^k)\mathbf{A}\ t(x, \mathbf{L}^k)$$

where $\pi(f, \mathbf{L}^k)$ *is as defined in Eq* (4.6)*, and matrix* $\mathbf{A}^f$ *is defined recursively:*

$$\mathbf{A}_{ij}^f = \begin{cases} 1 & i = j \\ -\sum_{i|l \neq i} t(L_l^k, L_i^k)\mathbf{A}_{lj}^f & otherwise \end{cases} \tag{4.9}$$

Note that $t(L_l^k, L_i^k) \neq 0$ iff $L_i^k \preceq L_l^k$, and can be pre-computed for any formula f. Eq (4.9) suggests an efficient dynamic programming algorithm for computing matrix $\mathbf{A}^f$.[2]

Theorem 2 shows that $N(f, x)$ for any homogeneous bivariate MLN formula f can be represented using subgraph statistics of subgraph features $\mathbf{L}^k$, assuming f has k logic variables. This conversion enables us to first convert an input MLN to ERGM, and then leverages ECS approximation (see Section 3.3.4) for inference.

Note that the ERGM features in Eq (3.3) are subgraph densities. To complete the conversion, it is necessary to scale each subgraph statistics accordingly. Let $\mathbf{f}_g = (L_1, L_2, \ldots, L_r)^T$ be effective subgraph features, we simply multiply their subgraph counts with the following diagonal matrix:

$$\mathbf{B} = \text{diag}\left(t(K_n, L_1), t(K_n, L_2), \ldots, t(K_n, L_r)\right) \tag{4.10}$$

For example, the diagonal matrix for Eq (4.8) is:

$$\mathbf{B} = \text{diag}(t(K_n, L_0^2), t(K_n, L_2^3), t(K_n, L_3^3))$$

[2]Section 4.5.1 provides the computational complexity of the dynamic programming algorithm.

We can rewrite Eq (4.8) in terms of $\phi_{\mathbf{f}_g}(x)$:

$$Z(\theta) = c(\theta) \sum_{x \in \mathcal{X}} \exp\left(\theta'^T \mathbf{B} \phi_{\mathbf{f}_g}(x)\right) \tag{4.11}$$

Here $c(\theta)$ is some linear function of θ when n is fixed; θ' is a linear transformation of θ; $\phi_{\mathbf{f}_g}(x)$ is the subgraph density vector for $\mathbf{L}_k$, as define in Eq (3.3):

$$\phi_{\mathbf{f}_g}(x) = \left(\frac{t(x, L_1)}{t(K_n, L_1)}, \frac{t(x, L_2)}{t(K_n, L_2)}, \ldots, \frac{t(x, L_r)}{t(K_n, L_r)} \right)$$

Eq (4.11) illustrates a direct mapping from homogeneous bivariate MLN partition function to ERGM partition function.

4.4.2 Relationship Between Exact Lifting For Transitive Relations and Triangle-Free Graph Enumeration

So far, no complexity result of lifting inference on transitive relation is available [Jaeger and Van den Broeck, 2012]. In this section, we show that the problem is at least as hard as enumerating triangle-free graphs. The latter is a well-known counting problem, which no efficient algorithm has been found yet. However, accurate approximate solution is available.

By rewriting (4.8) with (4.3), we can get:

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

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

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

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, 2013; Sloane and Plouffe, 1995].

On the other hand, the results on asymptotic enumeration of triangle-free graphs have a long history [Erdös et al., 1976]. The approximations usually rely on asymptotic properties in random graphs and give accurate estimations for large n.

4.5 Approximate Lifting Algorithm

By converting homogeneous bivariate MLNs into equivalent ERGMs, we are able to leverage ECS approximation 3.3.4 to estimate the log partition functions $\ln Z(\theta)$. In fact, ECS approximation can be viewed as an approximation to the generalized counting elimination in Eq (4.3).

Algorithm 2 shows the complete approximate lifting algorithm, which use ECS approximation as a sub-routine. Notice that $N(f_i, x)$ and $t(x, \mathbf{L}^k)$ in the algorithm are both functions of x.

Algorithm 2 Approximate Lifting

Input: Homogeneous bivariate MLN $(f_i, \theta_i)_{i=1}^m$ (hard constraints excluded)
Output: an approximation of $\ln Z(\theta)$
for $i = 1 \to m$ **do**
  For f_i with k logical variables, generate subgraph features $\mathbf{L}^k$.
  Compute $\mathbf{A}$ by (4.9) using $\mathbf{L}^k$.
  $N(f_i, w) \Leftarrow \phi(f_i, \mathbf{L}^k)\mathbf{A}\ t(w, \mathbf{L}^k)$
end for
Substitute $N(\mathbf{f}, w)$ in $\ln Z(\theta)$ with subgraph features $\mathbf{L}^k$ and sum terms with the same subgraphs as in (4.8).
$c(\theta) \Leftarrow$ constant terms.
$\mathbf{L} \Leftarrow$ subgraph features w/ nonzero coefficients ($\mathbf{L} \subset \mathbf{L}^k$).
$\theta' \Leftarrow$ coefficients of $\mathbf{L}$.
$\mathbf{B} \Leftarrow \text{diag}(t(K_n, L_0), \ldots, t(K_n, L_{r_k}))$
$M \Leftarrow \text{ECS}(\mathbf{B}\theta', \mathbf{L})$ {Call ECS approximation}
return $c(\theta)M$

4.5.1 Computational Complexity

During the conversion from an MLN to an ERGM, the algorithm generates subgraph features for a formula with k logical variables in $O\left(2^{k(k-1)/2}\right)$. Fortunately, k is usually very small (e.g., $k = 3$ for transitive relation) and the subgraph features can be pre-computed, so does $t(L_l^k, L_i^k)$ 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 $\mathbf{L}^k$.[3]

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

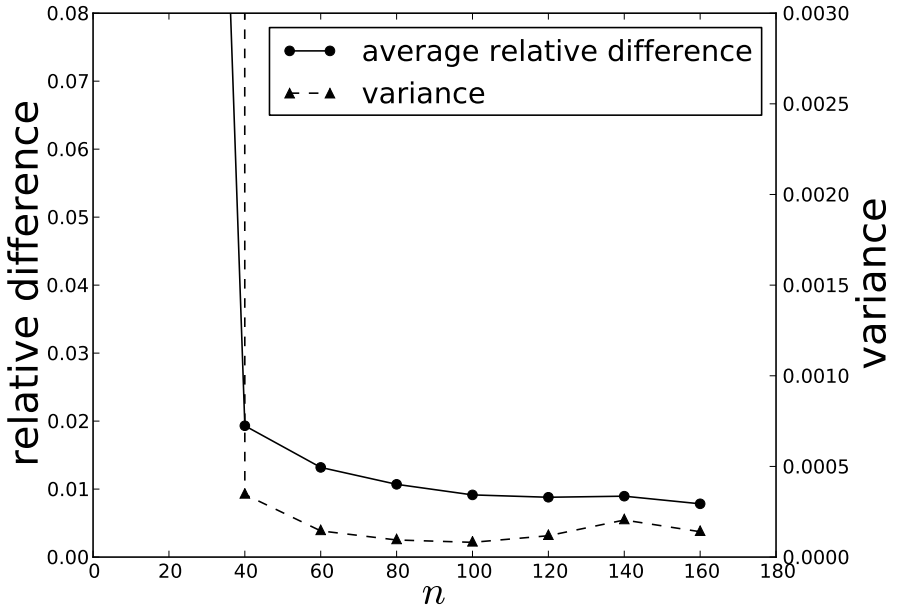


Figure 4.2: Accuracy of Approximate Lifting. The relative difference between Approximate Lifting and Bridge Sampling (if Bridge Sampling result passes the upper bound test Eq (3.19)) is almost negligible when n is large.

4.6 Experiments

We evaluate the accuracy of the approximation in the task of log-likelihood estimation on synthetic data. We use the model in Table 4.1 to generate social networks from a 16×16 grid of meta-parameters ranging between $(-5.0, -5.0)$ and $(5.0, 5.0)$. Networks of different sizes 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 defined in Eq (4.10) and Eq (3.16)) to factor out the impacts of different n.

The exact value of log partition function is unknown; therefore we resort to comparing Approximate Lifting with Bridge Sampling Gelman and Meng [1998]. For each n, we estimate the log-likelihood for all the parame-

[3]Formula with a large k is not the scope of this thesis. In this case, $|P_k|$ is approximately $O\left(2^{k(k-1)/2}/k!\right)$ [Harary and Palmer, 1973].

ters using both Approximate Lifting and Bridge Sampling. We exclude the potential degenerated results of Bridge Sampling using the upper bound test (Eq (3.19)), and compute the relative difference between the two methods for the rest of the estimations: $|(\ell_{\text{lift}} - \ell_{\text{mcmc}})/\ell_{\text{mcmc}}|$. Figure 4.2 reports the average relative differences and the variance for each n. We observe that the average error and variance are almost negligible for $n \geq 40$.

4.6.1 Notes on Lifted Belief Propagation

Lifted BP [Singla and Domingos, 2008], and the more general counting BP [Kersting et al., 2009], are the most popular approximate lifted inference algorithms. However, belief propagation algorithms suffer severe numerical instability for the MLN in Table 4.1 besides the inherent limitations of Bethe approximation.

To see this, we examine the messages passed from each variable (edge) to the factors (transitive relation) during BP. For example, the message passed from a variable to the first argument of transitive relation:

$$M_{v\to f_1}(X_v) \propto \exp\{\theta_1(1-X_v)\} \cdot M_{f\to v_1}^{\binom{n}{3}-1}(X_v) \cdot M_{f\to v_2}^{\binom{n}{3}}(X_v) \cdot M_{f\to v_3}^{\binom{n}{3}}(X_v) \tag{4.12}$$

Here $M_{f-v_i}(X_v)$ is the message sent from the factors where variable X_v is the i-th argument to X_v. The problem of Eq (4.12) is that the density of the graphical model leads to very large exponents for the three unique messages in the BP algorithm, therefore rendering $M_{v\to f_1}(X_v)$ extreme sensitive to any small changes to the three unique messages in the formula. Even after we carefully handled the numerical stability problem, any tiny difference between $M_{v\to f_1}(0)$ and $M_{v\to f_1}(1)$ will eventually leads to the inferred model

being dominated by two extreme cases: empty graph or complete graph (i.e., $b(X_v = 0) = 1$ or $b(X_v = 1) = 1$).

4.7 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 et al., 2009]. Lifted (normally polynomial-time) inference algorithms exploit exchangeability of random variables for variable eliminations (e.g., [De Salvo Braz et al., 2005; Milch et al., 2008; Poole, 2003]), message passing (e.g., [Singla and Domingos, 2008]) 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., [Apsel and Brafman, 2011; Bui et al., 2012; Gogate et al., 2012; Niepert, 2012a]) and social networks (e.g., [Hunter et al., 2012; Kemp et al., 2006a; Pu et al., 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 [Kersting et al., 2009; Singla and Domingos, 2008] and lifted MCMC [Niepert, 2012a]. 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 any evidence. Bhamidi et al. [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 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, 2012a]. Both techniques try to identify equivalent classes in the state space in theory, while both are not applicable directly in general. Niepert [2012a] uses graph automorphism as an alternative to generate orbital Markov chains instead of lumping, while Pu et al. [2012] and this work resort to approximate solution by exploiting the asymptotic properties of the underlying state space.

4.8 Conclusion

In this chapter, we show that a special class of MLN modeling structural relational knowledge can be converted into equivalent ERGM, and ECS approximation discussed in Section 3 can be leveraged to perform approximate lifted inference.

The conversion is done through introducing a set of subgraph features, of which the subgraph statistics (i.e., ERGM features) can be used to represent formula counting in MLN. The conversion can be performed efficiently using a dynamic programming for formulas with limited number of logic variables. The relationship between MLN and ERGM also reveals that ECS can be treated as an approximation to a generalized form of counting elimination.

For future work, we would like to extend the algorithm to cover more general MLN classes.

Chapter 5

Limited Expressiveness of Probabilistic Relational Graphical Models

5.1 Overview

Probabilistic relational graphical models (PRGM) are popular tools for explicitly modeling uncertain relational knowledge. PRGMs, such as MLN [Domingos and Lowd, 2009], enable researchers and practitioners to create probabilistic models on large complex uncertain systems easily. The flexibility and expressiveness of PRGM over propositional probabilistic graphical model stem from the domain-independence assumption: the relational knowledge of the model is independent with the domain of application. In this chapter, we show that this seemingly natural assumption does not always hold.

In most PRGM applications, the uncertain relational knowledge is often assumed to be independent with the domain of interests. This implicit assumption differentiates PRGM from propositional level PGMs by separating the model from domain; therefore enable researchers to easily transfer knowledge learned from one domain to another different domain as far as the two domains share similar structures. For example, Richardson and Domingos [2006] divided the UW-CSE professor-student dataset of the CSE department at University of Washington into 5 different groups based on their research areas, and performed leave-one-out testing by areas. In this case, a domain is a set of professors and students. The weights of the MLNs are trained

using data observed in four research areas, and testing is conducted on the last research area.

However, this important assumption does not always hold. We use the simple example MLN in Table 4.1 to demonstrate:

	Feature	Weight
f_1	$\neg\texttt{Fr}(\texttt{x},\texttt{y})$	θ_1
f_2	$\texttt{Fr}(\texttt{x},\texttt{y}) \wedge \texttt{Fr}(\texttt{y},\texttt{z}) \Rightarrow \texttt{Fr}(\texttt{x},\texttt{z})$	θ_2
f_3	$\texttt{Fr}(\texttt{x},\texttt{x})$	$-\infty$
f_4	$\neg(\texttt{Fr}(\texttt{x},\texttt{y}) \Leftrightarrow \texttt{Fr}(\texttt{y},\texttt{x}))$	$-\infty$

We show that the behavior of the model will change dramatically for domains of different sizes n even θ_1 and θ_2 are fixed. This result is a natural extension of Theorem 1 and Theorem 2. It is consistent with Shalizi and Rinaldo [2013]'s prediction that the same ERGM with non-trivial subgraph features has different behaviors for networks of different sizes.

For the rest of this chapter, we first re-visit the limited expressiveness for ERGMs in Section 5.2 to examine the effects of different feature specifications, then we discuss its implications for MLN in Section 5.3; Section 5.4 is related work; Section 5.5 concludes the chapter.

5.2 Limited Expressiveness for ERGM Revisited

Theorem 1 (Section 3.4) implies that the effects of any fixed ERGM parameter θ diminishes as the number of nodes in the graph n increases. The proof of Theorem 1 relies on the fact that the features we defined in Eq (3.3) are subgraph densities. Remember that $\ln Z(\theta)$ can be approximated by the sum

of a linear term and a log counting function:

$$\theta^T \mathbf{h}'(\theta, u^*) + \ln \#_{u^*} (\mathbf{h}'(\theta, u^*)) \tag{5.1}$$

Constant $|\sum_i |\theta_i||$ bound the first term, but the second term is in $O(n^2)$; therefore the optimal choice of $\mathbf{h}'$ will be dominated by the second term as $n \to \infty$.

The question remains whether the effect of varying model behavior exists if we use raw graph counts as features, because in this case the first term will be a function of n. For the rest of the section, we show that using raw subgraph counts as features does not change the fact that ERGM model behavior changes for networks of different sizes.

5.2.1 Using Subgraph Counts As Features

Lemma 3 shows that the sum of the summation in Eq (5.1) is a good approximation to $\ln Z(\theta)$ with bounded error $O(\ln n)$. What if we use the raw subgraph counts as features, instead of subgraph densities?

$$\phi_c(g) = (t(g, L_1), t(g, L_2), \ldots, t(g, L_r)) \tag{5.2}$$

Unfortunately, the adoption of $\phi_c(g)$ does not solve the problem. To see this, consider the simple ERGM with two subgraph features: edges L_- and triangles $L_\triangle$. Note that there is a one-to-one mapping between subgraph count feature and subgraph density feature, therefore the new feature specification does not affect $\ln \#_u(\mathbf{h'}_c(\theta, u))$, and our analysis focuses on $\theta^T \mathbf{h'}_c(\theta, u)$.

Here $\mathbf{h'}_c(\theta, u)$ is the optimum of $\gamma(\theta, u)$ with raw subgraph counts as features:

$$\mathbf{h'}_c(\theta, u) = \text{argmax}_{\mathbf{h}_c \in \phi_c(\mathcal{G}_u)} \left\{ \theta^T \mathbf{h}_c + \ln \#_u(\mathbf{h}_c) \right\}$$

We start with the subgraph density features for graph $g \in \mathcal{G}$: (see Eq (3.3))

$$\begin{aligned} \phi(g) =& (\phi_-(g), \phi_\triangle(g)) \\ =& \left(\frac{t(g, L_-)}{t(K_n, L_-)}, \frac{t(g, L_\triangle)}{t(K_n, L_\triangle)} \right) \\ =& \left(\frac{t(g, L_-)}{\binom{n}{2}}, \frac{t(g, L_\triangle)}{\binom{n}{3}} \right) \end{aligned}$$

If we only use edge count $t(g, L_-)$ as feature, and θ_- is a fixed coefficient:

$$\theta_- \, t(g, L_-) = \binom{n}{2} \theta_- \, \frac{t(g, L_-)}{t(K_n, L_-)} = \binom{n}{2} \theta_- \, \phi_-(g) \approx O(n^2)$$

However, for triangle count $t(g, L_\triangle)$ and the fixed coefficient $\theta_\triangle$, we have:

$$\theta_\triangle \, t(g, L_\triangle) = \binom{n}{3} \theta_\triangle \, \frac{t(g, L_\triangle)}{t(K_n, L_\triangle)} = \binom{n}{3} \theta_\triangle \, \phi_\triangle(g) \approx O(n^3)$$

In this case, the triangle count feature in $\mathbf{h'}_c(\theta, u)$ will quickly dominate in Eq (3.8), therefore leads to the domination of either complete graph (if $\theta_\triangle > 0$) or empty graph (if $\theta_\triangle < 0$) as $n \to \infty$.

This example illustrates that the unfavorable behavior exists no matter we use subgraph densities or raw subgraph counts as features. It essentially limits ERGMs with non-trivial subgraph features to graphs of fixed size. More generally, we have the following necessary condition for non-trivial ERGM models:

Lemma 6. *As $n \to \infty$, $\theta^T \mathbf{h}'(g)$ in $O(n^2)$ is a necessary condition for non-trivial ERGM.*

The proof follows from the fact that the second term in Eq (5.1) is in $O(n^2)$.

An interesting question to ask is whether defines $\theta = \binom{n}{2}\xi$ as a function of n and meta-parameter ξ help. The argument of Theorem 1 is no longer valid in this case. In fact, Chatterjee and Diaconis [2011] proved that under this definition of θ, ECS approximation (3.14) converges to $\ln Z(\theta)$ as $n \to \infty$ for certain configuration of θ.

This result is not new in ERGM literatures [Schweinberger, 2011; Shalizi and Rinaldo, 2013], but it has not been discussed in the context of MLN to the best of our knowledge.

5.3 Limited Expressiveness for MLN

The limitation of ERGM's expressiveness discussed in Section 5.2 has immediate implication on MLN through the relationship of the two we studied in Section 4.4.

We use the example in Table 4.1 to demonstrate that the relational knowledge in current definition of MLN is not domain-independent. Excluding the two hard constraints, the example MLN has two weighted formulas:

$$\begin{aligned} f_1 : &\quad \neg \texttt{Fr}(\texttt{x}, \texttt{y}) && \theta_1 \\ f_2 : &\quad \texttt{Fr}(\texttt{x}, \texttt{y}) \wedge \texttt{Fr}(\texttt{y}, \texttt{z}) \Rightarrow \texttt{Fr}(\texttt{x}, \texttt{z}) && \theta_2 \end{aligned}$$

Theorem 2 shows that the partition function of the MLN can be rewritten

using subgraph statistics (Eq (4.8)):

$$Z(\theta) = \exp\left\{2\binom{n}{2}\theta_1 + 6\binom{n}{3}\theta_2\right\} \sum_{x\in\mathcal{X}} \exp\left(\theta'^T t(x, \mathbf{f}_g)\right)$$

Here $\mathbf{f}_g$ are subgraph graphs in convenience vector representation $(L_1^2, L_2^3, L_3^3)^T$, and the coefficients are $\theta' = (-2\theta_1, -2\theta_2, -6\theta_2)^T$. Apply the same technique of Eq (3.6) to the log partition function we have:

$$\begin{aligned}\ln Z(\theta) &= 2\binom{n}{2}\theta_1 + 6\binom{n}{3}\theta_2 + \ln \sum_{x\in\mathcal{X}} \exp\left(\theta'^T t(x, \mathbf{f}_g)\right) \\ &\approx 2\binom{n}{2}\theta_1 + 6\binom{n}{3}\theta_2 + \max_{\mathbf{h}\in t(H,\mathbf{f}_g)} \left\{\theta'^T \mathbf{h} + \ln \#(\mathbf{h})\right\} + O(\ln n)\end{aligned}$$

Following similar analysis as Section 5.2, it is easy to see that the triangle and 2-star count in $\mathbf{h}$ is in $\Theta(n^3)$ and will be dominating the log counting function $\ln \#(\mathbf{h})$ as $n \to \infty$. In other words, as n increases, the model will quickly bias towards complete graph (if $\theta_2 < 0$) or empty graph (if $\theta_2 > 0$) even though the parameter pair (θ_1, θ_2) remains the same.

Theorem 3. *The domain-independence assumption of MLN does not always hold.*

5.4 Related Work

Schweinberger [2011] studied the varying behavior of ERGM for different n. The author introduced the concept of unstable sufficient statistics, and discussed its relationship with model degeneracy. Our result is derived from a computational perspective therefore more intuitive.

Similar limitation of ERGM is also discussed by Shalizi and Rinaldo [2013]. The authors introduced the concept of projective family distribu-

tions, and provided necessary conditions for projectivity. It was shown that ERGMs with non-trivial subgraph statistics, such as triangle and k-stars, are non-projective. As a result, any fixed model trained on one network cannot be extrapolated to networks of different size n.

5.5 Conclusion

Domain-independence is generally assumed to be true from the inception of PRGMs, such as MLN. The assumption allows us to separate a set of uncertain relational knowledge from the domain of applications, therefore distinguishes itself from propositional PGMs. In this chapter, we illustrated a counter example in which this assumption does not hold. The lack of domain-independence assumption severely limits the expressiveness and application of PRGMs.

Besides the counter example, it is still unclear which subset of model specifications may break the assumption. However, [Shalizi and Rinaldo, 2013] predicted that ERGMs with non-trivial subgraph statistics are non-projective, which is likely to be the case for MLN. We would like to explore a general theorem in future work.

Chapter 6

Summary and Future Work

This thesis studies efficient lifted inference on Probabilistic Relational Graphical Model (PRGM) with structural features, and the closely related Exponential Random Graph Model (ERGM). The contribution is three-folds: (1) We propose an efficient deterministic approximate inference algorithm for ERGM (Chapter 3); (2) We show that certain class of PRGM can be converted to ERGM, and the relationship leads to efficient approximate lifted inference algorithm (Chapter 4); (3) We show that ERGM and PRGM have limited expressiveness when modeling networks of different sizes, which in turn rejects the common domain-independence assumption of PRGM (Chapter 5).

6.1 Summary of Contributions

This thesis provides the following contributions:

- It introduces a new quadratic time deterministic approximation (ECS approximation) to the partition functions of ERGM. Our main insight enabling this advance is that subgraph statistics are sufficient to derive a lower bound for partition functions when the model of interests is not dominated by a few graphs. In comparison to Monte Carlo simulation based methods, the new method is scalable, stable, and precise enough for inference tasks. [Pu et al., 2012, 2013b]

- It shows that ERGM and PRGM are closely related to each other in terms of representing uncertain structural relational knowledge, and provides an efficient conversion algorithm. The connection reveals that the ECS approximation is an approximate solution to a generalized form counting elimination, which can also be used for approximate lifted inference on PRGMs. [Pu et al., 2013a]

- It re-visits the domain-independence assumption of PRGM, and shows that the assumption does not always hold. The observation is based on analyzing the asymptotic behavior of a bounded approximation of the log partition function of MLN.

6.2 Future Work

This thesis enables several future research directions:

- The ECS approximation introduced in Chapter 3 assumes the graphs are undirected, and only subgraph statistics features are allowed. Challenges remain to extend the algorithm to support directed graphs and other non-subgraph features.

- Current ECS approximation algorithm Eq (3.14) involves a naïve exhaustive search process, which leads to the quadratic time complexity. How to solve the optimization problem on the non-convex one-dimensional target function efficiently is essential for scaling up the algorithm.

- The relationship between ERGM and MLN is helpful for us to better understand both models. The conversion from MLN to ERGM in

Chapter 4 is limited to homogeneous bivariate MLNs. However, more general conversion algorithms are possible.

- The generalized counting elimination formulation discussed in Section 4.3 and 4.4.2 prompts the question: Does efficient exact lifting exist? To the best of our knowledge, no complexity result for enumerating triangle-free graphs exists.

- The rejection of the domain-independence assumption for general PRGM severely restricted its application. However, the assumption still holds for certain subset of relational knowledge. Verification of domain-independence is an interesting topic to explore in the future.

Appendix A

Appendix

A.1 Proof of Lemma 4

Before the proof of Lemma 4, we need some preparations. [Nowicki, 1989] proved that a vector of subgraph counts in $G(n,p)$ is asymptotically normally distributed with a degenerated co-variance matrix with rank 1, as the order of the graph $n \to \infty$. In other words, the subgraph counts are asymptotically linearly dependent on each other. Formally, let $\phi(g') = \{\phi_1(g'), \phi_2(g'), \ldots, \phi_r(g')\}$ be the densities of subgraphs $L_1, L_2, \ldots, L_r$ (i.e., $\phi_i(g') = \frac{t(g', L_i)}{t(K_n, L_i)}$) for $g' \in G(n,p)$, the sizes (number of edges) of these subgraphs are $s_1, s_2, \ldots, s_r$, and $u \sim \text{Bin}(\binom{n}{2}, p)$ is the edge count of g', we have the following theorem:

Theorem 4. *[Nowicki, 1989] For $g' \in G(n,p)$, and real vector $\mathbf{a} = (a_1, a_2, \ldots, a_r)^T$, the following asymptotic property holds:*

$$n^2 E\left[\mathbf{a}^T\Big(\phi(g') - \rho(u,p)\Big)\right]^2 \to 0 \tag{A.1}$$

where $\rho(u,p) = (\rho_1(u,p), \ldots, \rho_r(u,p))$, and $\rho_i(u,p) = s_i p^{s_i - 1} \cdot \frac{u}{\binom{n}{2}} - (s_i - 1)p^{s_i}$.

In theorem 4, if we set $p = u/\binom{n}{2}$, then $\rho_i(u, u/\binom{n}{2}) = \left(\frac{u}{\binom{n}{2}}\right)^{s_i}$, which becomes the expected density of L_i in $G(n, p = u/\binom{n}{2})$.

Next step is to extend the above property from $G(n,p)$ to $G(n,M)$.

Corollary 1. *For $g \in G(n, M = u)$, as $n \to \infty$, it holds that*

$$n^2 E_u \left[\mathbf{a}^T \Big(\phi(g) - \rho(u) \Big) \right]^2 \to 0$$

where $\rho_i(u) = \left(u / \binom{n}{2} \right)^{s_i}$

Proof. Following theorem 4, let $\rho_i(u) = \rho_i(u, u/\binom{n}{2})$, as $n \to \infty$, the following holds for $g' \in G(n, p = (u/\binom{n}{2}))$:

$$\begin{aligned} & n^2 E \left[\mathbf{a}^T \Big(\phi(g') - \rho(u) \Big) \right]^2 \to 0 \\ \Rightarrow & n^2 E \left[E_u \big[\mathbf{a}^T \Big(\phi(g') - \rho(u) \Big) \ \big| u \big]^2 \right] \to 0 \\ \Rightarrow & n^2 \sum_u p(u) \, E_u \big[\mathbf{a}^T \Big(\phi(g') - \rho(u) \Big) \ \big| u \big]^2 \to 0 \end{aligned}$$

Because $\sum_u p(u) = 1$ and $p(u) > 0$, the claim holds. □

Let c be some positive constant, apply Chebyshev's inequality to the linear combination $\mathbf{a}^T \phi(g)$, we get:

$$\begin{aligned} & P \left(\left| \mathbf{a}^T \left(\phi(g) - E_u \left(\phi(g) \right) \right) \right| \geq \frac{1}{2cn} \right) \\ & \qquad \leq 4c^2 n^2 \mathrm{Var}(\mathbf{a}^T \phi(g)) \end{aligned} \tag{A.2}$$

Now we start to prove Lemma 4.

Proof of Lemma 4. We first define function $\varepsilon(u)$:

$$\varepsilon(u) = \mathbf{a}^T (E_u(\phi(g)) - \rho(u)) \tag{A.3}$$

As we know

$$\begin{aligned} E_u\left(\mathbf{a}^T(\phi(g)-\rho(u))\right)^2 \geq & E_u\left(\mathbf{a}^T\left(\phi(g)-E_u(\phi(g))\right)\right)^2 \\ = & \mathrm{Var}(\mathbf{a}^T\phi(g)) \end{aligned} \tag{A.4}$$

The equality holds if and only if $\varepsilon(u)=0$. We can get the following property after applying it to corollary 1: as $n\to\infty$

$$n^2\mathrm{Var}\left(\mathbf{a}^T\phi(g)\right)\to 0 \tag{A.5}$$

Therefore, as $n\to\infty$

$$\begin{aligned} & n^2E_u\left(\mathbf{a}^T\left(\phi(g)-\rho(u)\right)-\varepsilon(u)\right)^2\to 0 \\ \Rightarrow & n^2E_u\left(\mathbf{a}^T\left(\phi(g)-\rho(u)\right)\right)^2-n^2\varepsilon(u)^2\to 0 \\ \Rightarrow & |\varepsilon(u)|<\frac{1}{2n} \end{aligned} \tag{A.6}$$

The last step used corollary 1.

We slack (A.2) using (A.4), and rewrite the inner expectation term using (A.3):

$$\begin{aligned} & P\left(\left|\mathbf{a}^T\left(\phi(g)-\rho(u)\right)-\varepsilon(u)\right|\geq\frac{1}{2cn}\right) \\ & \qquad \leq 4c^2n^2E_u\left(\mathbf{a}^T\left(\phi(g)-\rho(u)\right)\right)^2 \end{aligned} \tag{A.7}$$

Using (A.6), we can get

$$\begin{aligned} & P\left(\left|\mathbf{a}^T\left(\phi(g)-\rho(u)\right)-\varepsilon(u)\right|\geq\frac{1}{2cn}\right) \\ & \qquad \geq P\left(\left|\mathbf{a}^T\left(\phi(g)-\rho(u)\right)\right|\geq\frac{1}{cn}\right) \end{aligned}$$

Therefore, apply corollary 1, as $n \to \infty$, we get

$$P\left(\left|\mathbf{a}^T\left(\phi(g) - \rho(u)\right)\right| \geq \frac{1}{cn}\right) \to 0$$

□

References

E. M. Airoldi, D. M. Blei, S. E. Fienberg, and E. P. Xing. Mixed membership stochastic blockmodels. *The Journal of Machine Learning Research*, 9: 1981–2014, 2008.

E. Amir. Approximation algorithms for treewidth. *Algorithmica*, 56(4):448–479, 2010.

C. Andrieu, N. De Freitas, A. Doucet, and M. I. Jordan. An introduction to mcmc for machine learning. *Machine learning*, 50(1-2):5–43, 2003.

U. Apsel and R. I. Brafman. Extended lifted inference with joint formulas. In *UAI*, pages 11–18, 2011.

S. Arnborg, D. G. Corneil, and A. Proskurowski. Complexity of finding embeddings in ak-tree. *SIAM Journal on Algebraic Discrete Methods*, 8 (2):277–284, 1987.

A. Becker and D. Geiger. A sufficiently fast algorithm for finding close to optimal junction trees. In *Proceedings of the twelfth international conference on uncertainty in artificial intelligence*, pages 81–89. Morgan Kaufmann Publishers Inc., 1996.

J. Besag. Statistical analysis of non-lattice data. *The statistician*, pages 179–195, 1975.

S. Bhamidi, G. Bresler, and A. Sly. Mixing time of exponential random graphs. In *FOCS*, pages 803–812, 2008.

A. E. Brouwer. Number of unlabelled graphs with given number of triangles. `http://www.win.tue.nl/~aeb/graphs/cospectral/triangles.html`. Accessed: 2012-09-30.

H. Bui, T. Huynh, and S. Riedel. Automorphism groups of graphical models and lifted variational inference. In *UAI workshop Statistical Relational AI (StaRAI)*, 2012.

P. Carbonetto, J. Kisynski, N. de Freitas, and D. Poole. Nonparametric bayesian logic. In *UAI*, pages 85–93, 2005.

S. Chatterjee and P. Diaconis. Estimating and understanding exponential random graph models. *Arxiv preprint arxiv:1102.2650*, 2011.

J. Choi and E. Amir. Lifted relational variational inference. In *UAI*, pages 196–206, 2012.

J. Choi, R. de Salvo Braz, and H. H. Bui. Efficient methods for lifted inference with aggregate factors. In *AAAI*, pages 1030–1036, 2011.

N. Contractor, S. Wasserman, and K. Faust. Testing multitheoretical, multilevel hypotheses about organizational networks: An analytic framework and empirical example. *The Academy of Management Review*, 31(3):681–703, 2006.

G. F. Cooper. The computational complexity of probabilistic inference using bayesian belief networks. *Artificial intelligence*, 42(2):393–405, 1990.

R. De Salvo Braz, E. Amir, and D. Roth. Lifted first-order probabilistic inference. In *IJCAI*, pages 1319–1325, 2005.

P. Domingos and D. Lowd. Markov logic: An interface layer for artificial intelligence. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 3(1):1–155, 2009.

P. Domingos and W. A. Webb. A tractable first-order probabilistic logic. In *AAAI*, 2012.

P. Erdös and A. Rényi. On the evolution of random graphs. *Magyar Tud. Akad. Mat. Kutató Int. Közl*, 5:17–61, 1960.

P. Erdös, D. J. Kleitman, and B. L. Rothschild. Asymptotic enumeration of k_n-free graphs. In *Colloquio Internazionale sulle Teorie Combinatorie, (Rome, 1973), Tomo II, Atti dei Convegni Lincei, No. 17*, pages 19–27. Accad. Naz. Lincei, Rome, 1976.

N. Friedman, L. Getoor, D. Koller, and A. Pfeffer. Learning probabilistic relational models. In *IJCAI*, volume 99, pages 1300–1309, 1999.

A. Gelman and X. Meng. Simulating normalizing constants: From importance sampling to bridge sampling to path sampling. *Statistical Science*, pages 163–185, 1998.

S. Geman and D. Geman. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, (6):721–741, 1984.

M. R. Genesereth and N. J. Nilsson. *Logical foundations of artificial intelligence*, volume 9. Morgan Kaufmann Los Altos, 1987.

C. Geyer and E. Thompson. Constrained monte carlo maximum likelihood for dependent data. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 657–699, 1992.

E. Gilbert. Random graphs. *The Annals of Mathematical Statistics*, pages 1141–1144, 1959.

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

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

S. Goodreau, J. Kitts, and M. Morris. Birds of a feather, or friend of a friend? using exponential random graph models to investigate adolescent social networks*. *Demography*, 46(1):103–125, 2009.

J. Y. Halpern. An analysis of first-order logics of probability. *Artificial intelligence*, 46(3):311–350, 1990.

M. Handcock. Assessing degeneracy in statistical models of social networks, 2003.

M. S. Handcock, D. R. Hunter, C. T. Butts, S. M. Goodreau, and M. Morris. *statnet: Software tools for the Statistical Modeling of Network Data*. Seattle, WA, 2003. URL `http://statnetproject.org`. Version 2.0.

F. Harary and E. M. Palmer. *Graphical Enumeration*. Academic Press, 1973.

W. K. Hastings. Monte carlo sampling methods using markov chains and their applications. *Biometrika*, 57(1):97–109, 1970.

Q. Ho, J. Yin, and E. Xing. On triangular versus edge representations – towards scalable modeling of networks. In *Advances in Neural Information Processing Systems (NIPS)*, 2012.

P. Hoff. Modeling homophily and stochastic equivalence in symmetric relational data. In *Advances in Neural Information Processing Systems (NIPS)*, 2008.

D. Hunter, S. Goodreau, and M. Handcock. Goodness of fit of social network models. *Journal of the American Statistical Association*, 103(481):248–258, 2008.

D. R. Hunter. Curved exponential family models for social networks. *Social networks*, 29(2):216–230, 2007.

D. R. Hunter, P. N. Krivitsky, and M. Schweinberger. Computational statistical methods for social network models. *Journal of Computational and Graphical Statistics*, 21(4):856–882, 2012. doi: 10.1080/10618600.2012.732921.

M. Jaeger and G. Van den Broeck. Liftability of probabilistic inference: Upper and lower bounds. In *UAI workshop Statistical Relational AI (StaRAI)*, 2012.

M. Jordan, Z. Ghahramani, T. Jaakkola, and L. Saul. An introduction to variational methods for graphical models. *Machine learning*, 37(2):183–233, 1999.

M. I. Jordan, Z. Ghahramani, T. S. Jaakkola, and L. K. Saul. *An introduction to variational methods for graphical models*. Springer, 1998.

B. Kapferer. *Strategy and transaction in an African factory: African workers and Indian management in a Zambian town*, volume 10. Manchester University Press ND, 1972.

C. Kemp, J. B. Tenenbaum, T. L. Griffiths, T. Yamada, and N. Ueda. Learning systems of concepts with an infinite relational model. In *AAAI*, pages 381–388, 2006a.

C. Kemp, J. B. Tenenbaum, T. L. Griffiths, T. Yamada, and N. Ueda. Learning systems of concepts with an infinite relational model. In *Proceedings of AAAI Conference*, 2006b.

K. Kersting, B. Ahmadi, and S. Natarajan. Counting belief propagation. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, pages 277–284. AUAI Press, 2009.

J. Kisynski and D. Poole. Lifted aggregation in directed first-order probabilistic models. In *IJCAI*, pages 1922–1929, 2009.

D. Koller. Probabilistic relational models. In *Inductive logic programming*, pages 3–13. Springer, 1999.

S. L. Lauritzen and D. J. Spiegelhalter. Local computations with probabilities on graphical structures and their application to expert systems. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 157–224, 1988.

J. R. Lloyd, P. Orbanz, Z. Ghahramani, and D. M. Roy. Random function priors for exchangeable arrays with applications to graphs and relational data. In *Advances in Neural Information Processing Systems (NIPS)*, 2012.

D. Lusher, J. Koskinen, and G. Robins. *Exponential Random Graph Models for Social Networks: Theory, Methods, and Applications*. Cambridge University Press, 2012.

A. McCallum, K. Schultz, and S. Singh. Factorie: Probabilistic programming via imperatively defined factor graphs. In *NIPS*, pages 1249–1257, 2009.

N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller. Equation of state calculations by fast computing machines. *The journal of chemical physics*, 21:1087, 1953.

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

K. Miller, T. Griffiths, and M. Jordan. Nonparametric latent feature models for link prediction. *Advances in Neural Information Processing Systems (NIPS)*, 2009.

K. Murphy, Y. Weiss, and M. Jordan. Loopy belief propagation for approximate inference: An empirical study. In *UAI*, pages 467–475, 1999.

R. M. Neal. Annealed importance sampling. *Statistics and Computing*, 11 (2):125–139, 2001.

R. Ng and V. S. Subrahmanian. Probabilistic logic programming. *Information and computation*, 101(2):150–201, 1992.

M. Niepert. Markov chains on orbits of permutation groups. In *Proceedings of the Twenty-Eighth Conference on Uncertainty in Artificial Intelligence*, pages 624–633. AUAI Press, 2012a.

M. Niepert. Lifted probabilistic inference: An mcmc perspective. In *UAI workshop Statistical Relational AI (StaRAI)*, 2012b.

N. J. Nilsson. Probabilistic logic. *Artificial intelligence*, 28(1):71–87, 1986.

K. Nowicki. Asymptotic normality of graph statistics. *Journal of statistical planning and inference*, 21(2):209–222, 1989.

J. Pearl. Reverend bayes on inference engines: A distributed hierarchical approach. In *AAAI*, pages 133–136, 1982.

J. Pearl. *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Morgan Kaufmann, 1988.

A. Pfeffer, D. Koller, B. Milch, and K. T. Takusagawa. Spook: A system for probabilistic object-oriented knowledge representation. In *Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence*, pages 541–550. Morgan Kaufmann Publishers Inc., 1999.

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

W. Pu, E. Amir, and D. Espelage. Approximate partition functions for exponential-family random graph models. *NIPS Workshop on Algorithmic and Statistical Approaches for Large Social Networks*, 2012.

W. Pu, J. Choi, and E. Amir. Lifted inference on transitive relations. In *Workshops at the Twenty-Seventh AAAI Conference on Artificial Intelligence*, 2013a.

W. Pu, J. Choi, E. Amir, and D. Espelage. Learning exponential random graph models. Technical report, Department of Computer Science, University of Illinois at Urbana-Champaign, 2013b.

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

A. Rinaldo, S. E. Fienberg, and Y. Zhou. On the geometry of discrete exponential families with application to exponential random graph models. *Electronic Journal of Statistics*, 3:446–484, 2009.

N. Robertson and P. D. Seymour. Graph minors. iii. planar tree-width. *Journal of Combinatorial Theory, Series B*, 36(1):49–64, 1984.

G. Robins, P. Pattison, Y. Kalish, and D. Lusher. An introduction to exponential random graph (p*) models for social networks. *Social Networks*, 29(2):173–191, 2007.

D. Roth. On the hardness of approximate reasoning. *Artificial Intelligence*, 82(1):273–302, 1996.

S. J. Russell, P. Norvig, J. F. Canny, J. M. Malik, and D. D. Edwards. *Artificial intelligence: a modern approach*, volume 74. Prentice hall Englewood Cliffs, 1995.

R. Salakhutdinov and I. Murray. On the quantitative analysis of deep belief networks. In *Proceedings of the 25th ICML*, 2008.

M. Schweinberger. Instability, sensitivity, and degeneracy of discrete exponential families. *Journal of the American Statistical Association*, 106(496): 1361–1370, 2011.

P. Sen, A. Deshpande, and L. Getoor. Exploiting shared correlations in probabilistic databases. *Proceedings of the VLDB Endowment*, 1(1):809–820, 2008.

P. Sen, A. Deshpande, and L. Getoor. Bisimulation-based approximate lifted inference. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, pages 496–505. AUAI Press, 2009.

C. R. Shalizi and A. Rinaldo. Consistency under sampling of exponential random graph models. *The Annals of Statistics*, 41(2):508–535, 2013.

J. R. Shoenfield. *Mathematical logic*, volume 21. Addison-Wesley Reading, 1967.

S. Simpson, S. Hayasaka, and P. Laurienti. Exponential random graph modeling for complex brain networks. *PloS one*, 6(5):e20039, 2011.

P. Singla and P. Domingos. Lifted first-order belief propagation. In *AAAI*, volume 2, pages 1094–1099, 2008.

N. J. A. Sloane. The On-Line Encyclopedia of Integer Sequences. http://oeis.org/A006785, 2013. Triangle-free graphs on n vertices.

N. J. A. Sloane and S. Plouffe. *The Encyclopedia of Integer Sequences*. Academic Press, 1995.

T. Snijders. Markov chain monte carlo estimation of exponential random graph models. *Journal of Social Structure*, 3(2):1–40, 2002.

T. Snijders, P. Pattison, G. Robins, and M. Handcock. New specifications for exponential random graph models. *Sociological Methodology*, 36(1):99–153, 2006.

D. Strauss and M. Ikeda. Pseudolikelihood estimation for social networks. *Journal of the American Statistical Association*, pages 204–212, 1990.

G. Van den Broeck. On the completeness of first-order knowledge compilation for lifted probabilistic inference. *Advances in Neural Information Processing Systems 24*, pages 1386–1394, 2011.

M. van Duijn, K. Gile, and M. Handcock. A framework for the comparison of maximum pseudo-likelihood and maximum likelihood estimation of exponential family random graph models. *Social Networks*, 31(1):52–62, 2009.

M. Wainwright and M. Jordan. Graphical models, exponential families, and variational inference. *Foundations and Trends in Machine Learning*, 1 (1-2):1–305, 2008.

D. Wyatt, T. Choudhury, and J. Bilmes. Learning hidden curved exponential family models to infer face-to-face interaction networks from situated speech data. In *AAAI*, pages 732–738, 2008.

D. Wyatt, T. Choudhury, and J. Bilmes. Discovering long range properties of social networks with multi-valued time-inhomogeneous models. In *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence*, pages 630–636, 2010.

E. P. Xing, M. I. Jordan, and S. Russell. A generalized mean field algorithm for variational inference in exponential families. In *Proceedings of the Nineteenth conference on Uncertainty in Artificial Intelligence*, pages 583–591. Morgan Kaufmann Publishers Inc., 2002.

J. S. Yedidia, W. T. Freeman, and Y. Weiss. Understanding belief propagation and its generalizations. *Exploring artificial intelligence in the new millennium*, 8:236–239, 2003.

N. L. Zhang and D. Poole. A simple approach to bayesian network computations. In *Proceedings of the 10th Canadian conference on artificial intelligence*, Banff, Alberta, Canada, 1994.