

# Attributed Graph Models: Modeling Network Structure with Correlated Attributes

Joseph J. Pfeiffer III<sup>1</sup>, Sebastian Moreno<sup>1</sup>, Timothy La Fond<sup>1</sup>,  
Jennifer Neville<sup>1</sup>, Brian Gallagher<sup>2</sup>

<sup>1</sup>Purdue University, <sup>2</sup>Lawrence Livermore National Laboratory  
{jpfeiffer,smorenoa,tlafond,neville}@purdue.edu, bgallagher@llnl.gov

## ABSTRACT

Online social networks have become ubiquitous to today's society and the study of data from these networks has improved our understanding of the processes by which relationships form. Research in statistical relational learning focuses on methods to exploit correlations among the attributes of linked nodes to predict user characteristics with greater accuracy. Concurrently, research on generative graph models has primarily focused on modeling network structure *without* attributes, producing several models that are able to replicate structural characteristics of networks such as power law degree distributions or community structure. However, there has been little work on how to generate networks with real-world structural properties *and* correlated attributes.

In this work, we present the *Attributed Graph Model* (AGM) framework to *jointly* model network structure and vertex attributes. Our framework learns the attribute correlations in the observed network and exploits a generative graph model, such as the Kronecker Product Graph Model (KPGM) [11] and Chung Lu Graph Model (CL) [2], to compute structural edge probabilities. AGM then combines the attribute correlations with the structural probabilities to sample networks conditioned on attribute values, while keeping the expected edge probabilities and degrees of the input graph model. We outline an efficient method for estimating the parameters of AGM, as well as a sampling method based on *Accept-Reject* sampling to generate edges with correlated attributes. We demonstrate the efficiency and accuracy of our AGM framework on two large real-world networks, showing that AGM scales to networks with hundreds of thousands of vertices, as well as having high attribute correlation.

**Categories and Subject Descriptors:** H.2.8 [Database Management]: Database Applications - *Data Mining*

**General Terms:** Algorithms, theory, rejection sampling

**Keywords:** Network analysis, network modeling, attributed graph models.

## 1. INTRODUCTION

The growth of the internet has created large scale collections of networked data, with online services encouraging social interaction (Facebook, Twitter) and fostering business communication (Email, LinkedIn). By studying the structures and attributes of large scale relational data, researchers have discovered that data often exhibits *correlation* among the attributes of linked individuals. Further work has considered the root cause of this correlation and distinguished between *social influence*, which is the tendency for linked individuals to adopt the characteristics of their neighbors, and *homophily*, where links are created based on the attribute similarities of the individuals [10, 15, 22]. Relational machine learning takes advantage of correlation in networks to *jointly* predict class labels and has been applied to a wide variety of domains to improve prediction accuracy, from web-pages and online social networks to genetics and securities regulators [24, 17, 4].

Concurrently, research on *generative graph models* has focused on statistical processes to generate distributions of graphs, with the goal of understanding how commonly occurring structural features arise, such as power law degree distributions, clustering, or community structure [2, 11, 19, 8, 18]. Models of processes to construct large complex networks can be used to further tailor and improve predictive algorithms, detect anomalies in networks, and test algorithm performance to future network structure. Additionally, since access to large network data for research is often restricted due to security and privacy concerns, accurate generative graph models can be used to produce realistic, yet anonymous, synthetic networks for public study.

While the modeling of attribute correlations across network links has found widespread acceptance in relational machine learning, the converse—modeling network structure given observed vertex attributes—has generally not been considered by researchers focusing on generative graph models. The majority of research in this area has focused on modeling graphs *without* vertex attributes (e.g., [2, 11, 19, 18]), in spite of the fact that many social networks such as Facebook, LinkedIn, and Twitter have associated vertex attributes (e.g., users' interests and affiliations). Notable exceptions which can (potentially) include vertex attributes are the Exponential Random Graph (ERG) Model [20], the Multiplicative Attribute Graph (MAG) Model [7], Latent Space (LS) approaches [6], and Mixed Membership Stochastic Blockmodels (SBM) [1]. The ERG model is flexible enough to include a wide range of attribute and structural information, but this flexibility leads to computational

costs which prohibit its application to networks with greater than a few thousand vertices. The MAG model aims to exploit vertex attributes to better model the structural features found in real world networks. However, this recent work has mainly focused on marginalizing over a set of *latent* (i.e., hidden) vertex attributes, rather than explicitly *learning* the model from observed vertex attributes. This is similar to LS and SBM approaches, which do not approach the problem in terms of structural characteristics such as a power law degree distribution and small diameter, but rather in terms of using latent variables to discover and model communities. Moreover, ERGM and MAG are largely used for *descriptive* analysis, where the structure of a large graph is summarized by a small set of statistics for hypothesis testing. While the ERGM representation can improve our understanding of the structure of a network, it generally makes sampling more difficult [5].

**Motivating Example:** Consider the following scenario that we will refer to throughout this work. Two users in a network (Alice and Bob) have a large number of common friends, which in turn implies a high likelihood that they will become friends. At some point in time, Alice and Bob might meet through a mutual friend. However, if we examine the intrinsic attributes of Alice and Bob, we may find that Alice is conservative while Bob is liberal. Although this does not prevent the two from becoming friends, political views typically correlate across edges in a network. Thus, a model which represents the probability that an edge will form between Alice and Bob should consider both their network structure and their attributes.

**Modeling Structure and Correlated Attributes:** We introduce the *Attributed Graph Model* (AGM) to model joint distributions of edges conditioned on vertex attributes. AGM utilizes an underlying structural graph model to sample possible edges (e.g., the likelihood that Alice and Bob will meet due to their structural properties). However, in contrast to prior work on generative graph models, AGM also models the attribute correlations in a network. It then uses these correlations to estimate the conditional probability a proposed edge should be included in the network given the attributes of the corresponding vertices.

The AGM framework is based on *Accept-Reject* sampling from computational statistics [13], where a *proposal* distribution is used to draw a sample that is either accepted or rejected (probabilistically) depending on the characteristics of the sample. Accepted samples are samples from the true distribution, with acceptance probabilities (loosely) reflecting the distance between the proposing and the true distribution. In AGM, a *proposed* edge is drawn from an underlying *structural* graph model (e.g., Alice and Bob meet at a party). Then the possible edge is *accepted* into the network with some probability, depending on the characteristics of the incident vertices (e.g., Alice and Bob’s political views). The resulting sample of accepted edges is equivalent to a draw from a distribution that has both the desired network structure and attribute correlation. More formally, AGM models and samples from the joint distribution of edges *given* vertex attributes—which could be interpreted as an explicit model of homophily.

The AGM framework is general enough to use in conjunction with many probabilistic generative graph models and does not sacrifice important characteristics of a given struc-

tural model. In particular, we prove AGM’s expected degree distribution equals the degree distribution of the input generative graph model. We implement versions of AGM based on some of the most well known scalable generative models, i.e., *Kronecker Product Graph Model* (KPGM) [11], *Chung Lu* (CL) [2], and *Transitive Chung Lu* (TCL) [18]. As part of our analysis, we show that the KPGM and CL models can be grouped into a single framework, which AGM extends. Moreover, we outline learning and sampling algorithms for AGM that are efficient provided the selected generative graph model is scalable.

The specific contributions of our work include:

- Introduction of a novel framework (AGM) for modeling and sampling networks where vertex attributes are correlated across edges. AGM exploits generative graph models to enable efficient sampling and modeling of a network’s structural characteristics.
- Efficient sampling and methods for AGM which scale to large networks of hundreds of thousands of vertices and multiple attribute correlations.
- Proofs that the AGM model preserves the degree distribution in expectation, in addition to accurately modeling the correlations of vertex attributes.
- Demonstration that AGM can be paired with a number of generative graph models to sample networks with correlation while retaining the structural characteristics of the input graph model.

We begin with a brief review of generative graph models and other related work in Section 2, with further notation and background as needed in Section 3. We then outline our AGM framework in Section 4 and discuss analytical properties in Section 5. We demonstrate the abilities of AGM in Section 6, including generating a network with over 500,000 vertices and multiple correlated attributes. We conclude in Section 7.

## 2. RELATED WORK

Our work connects several prominent areas of the social network literature. First, a primary concern in social network analysis is the *process* which generates edges in the network. The first generative graph model, the Erdős Rényi graph model, generated graphs with equal probability of every edge occurring [3]. However, the structural features of this model failed to match those found in many real world networks, leading to a variety of approaches attempting to match the power law degree distribution of the network ([20, 11, 19, 18]). The majority of approaches attempt to model structural features, ignoring the vertex attributes.

One notable exception to this is the exponential random graph (ERG) model (an extension to the Erdős Rényi model), which models a set of features as a distribution in the exponential family, where the functions are flexible enough to incorporate attributed vertices. The computational cost of this method is at least quadratic in terms of the number of vertices, limiting its applicability in large scale networks. ERG models also currently suffer from several other issues, including degeneracy [5] as well as being inconsistent under anything but dyadic independence [21]. Another notable exception is the multiplicative attributed graph (MAG) model,

which considers vertex attributes in order to match relational structure. While MAG primarily marginalizes over latent attributes to capture network features, we modify MAG to incorporate observed variables, which allows comparison. Both ERG and MAG are generally utilized in a more *descriptive* manner, providing insight into the structural characteristics of the network rather than for generating accurate samples. Scalable models for sampling, such as KPGM [11], CL [2], TCL [18] and Block Two-Level Erdős Rényi [8], currently only consider the structural characteristics of networks, omitting any vertex attributes. Latent Space [6] and Stochastic Blockmodels [1] generally approach the problem of clustering vertices without considering network statistics such as degree distributions and diameter.

Although generative graph models which are conditioned on vertex attributes are relatively rare, finding such behavior in real-world networks is not. Individuals in social networks tend to create links with others who have a common interest (i.e., homophily [15]), which leads to networks where the existence of edges is dependent on the attributes of their endpoints [10, 12]. Thus, modeling of the correlation of attributes in generative models leads to better understanding of link formation. However, our work has goals which are distinct from *link prediction* [4, 24] in that we are modeling a distribution of graphs with similar structure and correlation, not a conditional distribution of unobserved edges given a set of observed edges and vertex attributes.

### 3. NOTATION AND BACKGROUND

A graph  $G = \langle \mathbf{V}, \mathbf{E}, \mathbf{X} \rangle$  is comprised of a set of  $N_v$  vertices  $\mathbf{V}$ , a set of  $N_e$  edges  $\mathbf{E} \subset \mathbf{V} \times \mathbf{V}$ , and a set of  $N_v$   $W$ -dimensional feature vectors  $\mathbf{x}_i \in \mathbf{X}$ . An edge  $(v_i, v_j)$  indicates a *relationship* between the vertices  $v_i$  and  $v_j$ . Given  $\mathbf{E}$ , the degree  $d_i$  of a vertex  $v_i$  is defined by the number of vertices that it is connected to ( $d_i = \sum_{j=1}^{N_v} \mathbb{I}[(v_i, v_j) \in \mathbf{E}]$ ). The  $W$ -dimensional feature vectors  $\mathbf{x}_i \in \mathbf{X}$  are paired with corresponding vertices  $v_i \in \mathbf{V}$  and represent characteristics of  $v_i$ . For example, if  $v_i$  represents a person, the  $w^{th}$  characteristic  $\mathbf{x}_i[w]$  could include attributes such as: IsConservative, IsChristian, IsFemale, etc. In this work we consider only binary attributes, but the algorithms and results hold for more general settings.

#### 3.1 Generative Graph Models

Let  $\mathcal{M}$  be a generative graph model such as: Chung Lu (CL) [2], Transitive Chung Lu (TCL) [18], Kronecker Product Graph Model (KPGM) [11], or Block Two-Level Erdős Rényi [8]. We consider the class of generative graph models that represent the set of possible edges in the graph as binary random variables  $E_{ij}$ . In this case, the event  $E_{ij} = 1$  implies  $(v_i, v_j) \in \mathbf{E}$  (or  $(v_j, v_i) \in \mathbf{E}$  in the case of undirected networks) and the model  $\mathcal{M}$  assigns a *probability* to the variable  $E_{ij}$  given a set of parameters  $\Theta_{\mathcal{M}}$  ( $P(E_{ij} = 1 | \Theta_{\mathcal{M}})$ ). Typically,  $\Theta_{\mathcal{M}}$  is learned from an input graph  $G^o = \langle \mathbf{V}^o, \mathbf{E}^o, \mathbf{X}^o \rangle$  using the model  $\mathcal{M}$ .

As  $\Theta_{\mathcal{M}}$  defines a distribution over graph configurations with respect to the chosen model  $\mathcal{M}$ , a complete set of edges  $\mathbf{E}$  can be drawn (sampled) using  $\Theta_{\mathcal{M}}$ . To generate a new graph  $G = \langle \mathbf{V}, \mathbf{E} \rangle$ , every edge is sampled according to  $Bernoulli(P(E_{ij} = 1 | \Theta_{\mathcal{M}}))$ ; if the draw is a success, the edge  $(v_i, v_j)$  is added to  $\mathbf{E}$ . *Note that these models do not consider attributes  $\mathbf{X}$  that are observed in the input graph.*

---

#### Algorithm 1 AcceptRejectSampling ( $Q, Q'$ )

---

```

1:  $\mathbf{R}(Y) = \frac{Q(Y)}{Q'(Y)}$ 
2:  $\mathbf{A}(Y) = \frac{\mathbf{R}(Y)}{\sup[\mathbf{R}(Y)]}$ 
3:  $S = \emptyset$ 
4: while  $|S| < \text{number of samples}$  do
5:    $u \sim \text{Uniform}(0, 1)$ 
6:    $y \sim Q'(Y)$ 
7:   if  $u < \mathbf{A}(y)$  then
8:      $S = S \cup y$ 
9:   end if
10: end while
11: return  $S$ 

```

---

**Chung Lu Graph Models:** The Chung-Lu (CL) is a generative graph model, a *weighted* version of the Erdős-Rényi model [2]. In its basic form, every edge is sampled proportional to the product of the degrees of its endpoints, where the probabilities of an edge is given by:

$$P_{CL}(E_{ij} = 1 | \Theta_{CL}) = \frac{\theta_{d_i} \theta_{d_j}}{\sum_{v_k \in V} \theta_{d_k}}$$

where  $\Theta_{\mathcal{M}} = [\theta_{d_1}, \dots, \theta_{d_{N_v}}]$  and  $\theta_{d_i} = d_i$ .

This formulation guarantees the *expected* degree of the sampled graph is equal to the degree of the original graph:

$$\mathbb{E}_{CL}[d_i | \Theta_{CL}] = \sum_{v_j \in V} \frac{\theta_{d_i} \theta_{d_j}}{\sum_{v_k \in V} \theta_{d_k}} = \theta_{d_i} \frac{\sum_{v_j \in V} \theta_{d_j}}{\sum_{v_k \in V} \theta_{d_k}} = \theta_{d_i}$$

In [19], the authors noted that generating a network with a random draw on every edge is computationally expensive ( $O(N_v^2)$ ), and proposed drawing from the degree distribution ( $\frac{\theta_{d_i}}{\sum_k \theta_{d_k}}$ ) twice in order to generate an edge, and repeating the process  $N_e$  times.

**Kronecker Product Graph Models:** With Kronecker Product Graph Models (KPGM),  $K$  Kronecker products of a  $b \times b$  initiator matrix of parameters  $\Theta_{\mathcal{M}}$  are used to define the marginal probabilities of edges in the network [11]. For example, the marginal probability of an edge existing is:

$$P_{KP}(E_{ij} = 1 | \Theta_{KP}) = \prod_{k=1}^K \Theta_{KP}(\sigma_{ki}, \sigma_{kj})$$

where  $\sigma_{ki}$  indicates the position of the parameter in the initiator matrix  $\Theta_{\mathcal{M}}$  that is associated with vertex  $(v_i)$  in the  $k^{th}$  Kronecker multiplication. The fast generation algorithm for KPGM draws from the normalized parameter matrix ( $\frac{\Theta_{\mathcal{M}}}{\sum_{ij} \Theta_{\mathcal{M}}}$ )  $K$  times to determine a single edge sample, then repeats the process  $N_e$  times.

#### 3.2 Accept-Reject Sampling

*Accept-Reject* sampling is a framework for generating samples from a desired distribution  $Q$  [13]. For many distributions, direct sampling from  $Q$  is difficult either because direct methods do not exist or are inefficient; however, *proposal* distributions  $Q'$  exist which are easier to sample.

Given distributions  $Q(Y), Q'(Y)$  for a random variable  $Y$ , define the ratio between them for a particular value  $Y = y$ :

$$R(Y = y) = \frac{Q(Y = y)}{Q'(Y = y)} \quad \mathbf{R}(Y) = \frac{Q(Y)}{Q'(Y)}$$

with the set of ratios over the possible values for  $Y$  being  $\mathbf{R}(Y)$ : The *acceptance* probabilities  $A(Y = y)$  (and corresponding set  $\mathbf{A}(Y)$ ) are defined as:

$$A(Y = y) = \frac{R(Y = y)}{\sup[\mathbf{R}(Y)]} \quad \mathbf{A}(Y) = \frac{\mathbf{R}(Y)}{\sup[\mathbf{R}(Y)]}$$

A typical algorithm for accept-reject sampling is given in Algorithm 1. It begins by initially computing  $\mathbf{R}(Y)$ ,  $\mathbf{A}(Y)$ , then proceeds to iteratively *propose* (or draw) samples  $y \sim Q'(Y)$  (lines 4-10). With probability  $A(y)$ , the proposed samples are *accepted* in the set of samples to return ( $S$ ); otherwise they are *rejected*. For intuition, when the distribution  $Q'(Y)$  samples  $y$  too frequently in comparison to  $Q(Y)$ , those samples are generally excluded from the final sample set. The resulting distribution of accepted samples follows  $Q(Y)$ . Accept-Reject sampling can be utilized for discrete or continuous random variables, with the only requirement that the ratios are bounded [13].

## 4. ATTRIBUTED GRAPH MODELS

In this section we outline our proposed *Attributed Generative Model* (AGM) framework. Current scalable graph models (such as TCL and KPGM) draw from the joint distribution of edges given a set of edge parameters  $\Theta_{\mathcal{M}}$ . This could be combined with simple generation of attributes on the vertices, given attribute parameters  $\Theta_X$ , by assuming the vertex attributes are independent of the edges. However, as social networks typically exhibit *homophily* this assumption is generally incorrect, meaning:

$$P(\mathbf{E}|\mathbf{X}, \Theta_{\mathcal{M}})P(\mathbf{X}|\Theta_X) \neq P(\mathbf{E}|\Theta_{\mathcal{M}})P(\mathbf{X}|\Theta_X)$$

Here,  $P(\mathbf{X}|\Theta_X)$  represents a prior distribution for the attributes on the vertices, which can be estimated, for example, using probabilistic graphical models [9]. However, estimation of  $P(\mathbf{E}|\mathbf{X}, \Theta_{\mathcal{M}})$  in large domains remains an open problem. For example, consider again the motivating case from the introduction, where the goal is to model the *Political Views* in an input friendship network  $G^o$ . If we assume independence between attributes and edges (as scalable graph models do), the generated graph will have many fewer friendships generated among two Conservatives (C-C) compared to those that are observed in  $G^o$  (Figure 1.a). If we consider the *ratio* between the attribute configurations on edges observed in the input network against the proposed edges (from the independent model), configurations such as NC-C are overrepresented by the proposed model (Figure 1.b). This motivates our use of accept-reject sampling in AGM—our new framework that considers the dependencies between the attributes and edges of a network in a generative model.

### 4.1 Framework

Our AGM framework incorporates distributions over the attributes ( $P(\mathbf{X}|\Theta_X)$ ) and edges ( $P(\mathbf{E}|\Theta_{\mathcal{M}})$ ). In addition, the AGM approach uses a parameterization  $\Theta_F$  to model the desired attribute correlations across edges in a scalable way—in conditionals of the form  $P(E_{ij} = 1|\mathbf{X}, \Theta_{\mathcal{M}}, \Theta_F)$ . Specifically, we introduce a deterministic function  $f(\mathbf{x}_i, \mathbf{x}_j)$ , which maps tuples of attribute vectors to a single model of correlation across linked edges. The random variables  $E_{ij}$  remain conditionally independent Bernoulli trials, and the

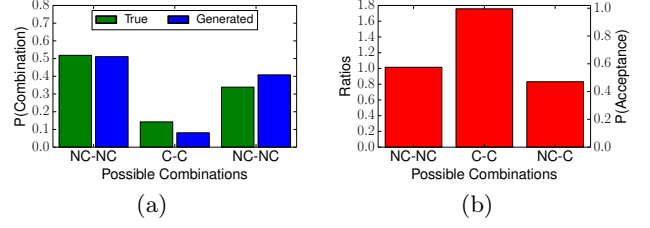


Figure 1: (a) Distributions of Politics across edges (C conservative, NC non conservative), for network  $G^o$  and network generated by  $\mathcal{M}$ . (b) Ratios between these distributions (left y-axis) and acceptance probabilities (right y-axis).

only additional dependence is on the attributes of the incident vertices  $\mathbf{x}_i, \mathbf{x}_j$ . Thus, the edge trials are conditionally independent from one another:

$$P(\mathbf{E}|\mathbf{X}, \Theta_{\mathcal{M}}, \Theta_F) = \prod_{ij} P(E_{ij} = 1|\mathbf{X}, \Theta_{\mathcal{M}}, \Theta_F) \\ = \prod_{ij} P(E_{ij} = 1|f(\mathbf{x}_i, \mathbf{x}_j), \Theta_{\mathcal{M}}, \Theta_F)$$

Let  $P_o(E_{ij} = 1|f(\mathbf{x}_i, \mathbf{x}_j), \Theta_{\mathcal{M}}, \Theta_F)$  be the conditional distribution of an edge in the observed graph given the corresponding attributes on the incident vertices, with  $\Theta_F$  referring to the parameterization estimated from the observed graph. Applying Bayes' Theorem, we have:

$$P_o(E_{ij} = 1|f(\mathbf{x}_i, \mathbf{x}_j), \Theta_{\mathcal{M}}, \Theta_F) \\ = \frac{P_o(f(\mathbf{x}_i, \mathbf{x}_j)|E_{ij} = 1, \Theta_{\mathcal{M}}, \Theta_F) \cdot P_o(E_{ij} = 1|\Theta_{\mathcal{M}}, \Theta_F)}{P_o(f(\mathbf{x}_i, \mathbf{x}_j)|\Theta_{\mathcal{M}}, \Theta_F)} \\ = P_o(E_{ij} = 1|\Theta_{\mathcal{M}}, \Theta_F) \frac{P_o(f(\mathbf{x}_i, \mathbf{x}_j)|E_{ij} = 1, \Theta_{\mathcal{M}}, \Theta_F)}{P_o(f(\mathbf{x}_i, \mathbf{x}_j)|\Theta_{\mathcal{M}}, \Theta_F)}$$

Here we assume that the prior distribution of the edge is defined by our chosen *structural* model  $\mathcal{M}$ , meaning  $P_o(E_{ij} = 1|\Theta_{\mathcal{M}}, \Theta_F) = P_{\mathcal{M}}(E_{ij} = 1|\Theta_{\mathcal{M}})$ , while the posterior distribution accounts for the observed vertex attributes. Unfortunately, it is not simple to derive efficient estimation and sampling methods for the underlying data that reflect the observed edge/attribute correlations. However, there has been considerable work on scalable *structural* generation models (i.e.,  $\mathcal{M}$ ). Thus, we exploit the sampling mechanism from a simpler graph model  $\mathcal{M}$  to approximate the true data distribution observed in  $G^o$ .

We define the ratio between the edge probabilities in the the observed data  $G^o$  and in the graph model  $\mathcal{M}$ :

$$R(f(\mathbf{x}_i, \mathbf{x}_j)|\Theta_{\mathcal{M}}, \Theta_F) = \frac{P_o(E_{ij} = 1|f(\mathbf{x}_i, \mathbf{x}_j), \Theta_{\mathcal{M}}, \Theta_F)}{P_{\mathcal{M}}(E_{ij} = 1|f(\mathbf{x}_i, \mathbf{x}_j), \Theta_{\mathcal{M}}, \Theta_F)} \quad (1)$$

Given estimation and sampling methods for  $\mathcal{M}$ , we can adjust the edge probabilities to recover the distribution for  $G^o$  using  $R(f(\mathbf{x}_i, \mathbf{x}_j)|\Theta_{\mathcal{M}}, \Theta_F)$ :

$$P_o(E_{ij} = 1|f(\mathbf{x}_i, \mathbf{x}_j), \Theta_{\mathcal{M}}, \Theta_F) \\ = P_{\mathcal{M}}(E_{ij} = 1|f(\mathbf{x}_i, \mathbf{x}_j), \Theta_{\mathcal{M}}, \Theta_F) \cdot R(f(\mathbf{x}_i, \mathbf{x}_j)|\Theta_{\mathcal{M}}, \Theta_F) \quad (2)$$

The equation above can be used to adjust for the discrepancies between the probabilities calculated by the model  $\mathcal{M}$  and those that reflect the true data distribution of  $G^o$ . For sparse networks we can utilize a sample graph from  $\mathcal{M}$  and the original graph  $G^o$  to further approximate  $R$  in Equation 2.

LEMMA 1. *Given a target distribution  $P_o$  and a generative graph model  $\mathcal{M}$ , we can model  $P_o$  indirectly using  $P_{\mathcal{M}}$*

and the ratio  $R$  from Eq. 1. Furthermore, when the edge priors are modeled by  $\mathcal{M}$  (i.e.,  $P_o(E_{ij}=1|\Theta_{\mathcal{M}}, \Theta_F) = P_{\mathcal{M}}(E_{ij}=1|\Theta_{\mathcal{M}})$ ) and the graph is sparse, we can approximate  $R$  efficiently with  $\tilde{R} = \frac{P_o(f(\mathbf{x}_i, \mathbf{x}_j)|E_{ij}=1, \Theta_{\mathcal{M}}, \Theta_F)}{P_{\mathcal{M}}(f(\mathbf{x}_i, \mathbf{x}_j)|E_{ij}=1, \Theta_{\mathcal{M}}, \Theta_F)}$ :

$$\begin{aligned} P_o(E_{ij}=1|f(\mathbf{x}_i, \mathbf{x}_j), \Theta_{\mathcal{M}}, \Theta_F) \\ = P_{\mathcal{M}}(E_{ij}=1|\Theta_{\mathcal{M}}) \cdot R(f(\mathbf{x}_i, \mathbf{x}_j)|\Theta_{\mathcal{M}}, \Theta_F) \end{aligned} \quad (3)$$

$$\begin{aligned} \approx P_{\mathcal{M}}(E_{ij}=1|\Theta_{\mathcal{M}}) \cdot \tilde{R}(f(\mathbf{x}_i, \mathbf{x}_j)|\Theta_{\mathcal{M}}, \Theta_F) \\ = P_{\mathcal{M}}(E_{ij}=1|\Theta_{\mathcal{M}}) \cdot \frac{P_o(f(\mathbf{x}_i, \mathbf{x}_j)|E_{ij}=1, \Theta_{\mathcal{M}}, \Theta_F)}{P_{\mathcal{M}}(f(\mathbf{x}_i, \mathbf{x}_j)|E_{ij}=1, \Theta_{\mathcal{M}}, \Theta_F)} \end{aligned} \quad (4)$$

See proof in Appendix A. Estimation and sampling in AGM involves the three probabilities on the last line of Equation 4. From a high level, these can each be explained as follows:

- $P_{\mathcal{M}}(E_{ij}=1|\Theta_{\mathcal{M}})$  is the prior probability of an edge existing according to  $\mathcal{M}$ .
- $P_o(f(\mathbf{x}_i, \mathbf{x}_j)|E_{ij}=1, \Theta_{\mathcal{M}}, \Theta_F)$  represents the correlations observed in the graph  $G^o$ .
- $P_{\mathcal{M}}(f(\mathbf{x}_i, \mathbf{x}_j)|E_{ij}=1, \Theta_{\mathcal{M}}, \Theta_F)$  represents the correlation (randomly) produced by  $\mathcal{M}$ .

Edge samples with attribute configurations that are under-sampled in  $\mathcal{M}$  are given a higher conditional probability, while samples with configurations that are over-sampled in  $\mathcal{M}$  are given lower probability. AGM provides efficient methods for sampling and estimation in each of these three distributions.

## 4.2 Sampling

Ideally, an algorithm would estimate and sample directly from Equation 4. However, as  $N_v^2$  edges can exist in the network, both estimation and sampling from this distribution are prohibitively expensive for large networks. Instead, we draw  $N_e$  samples from a multinomial parameterized by:

$$\begin{aligned} Q(i, j) &= \frac{P_o(E_{ij}=1|f(\mathbf{x}_i, \mathbf{x}_j), \Theta_{\mathcal{M}}, \Theta_F)}{\sum_{k,l} P_o(E_{kl}=1|f(\mathbf{x}_k, \mathbf{x}_l), \Theta_{\mathcal{M}}, \Theta_F)} \\ &\propto P_o(E_{ij}=1|f(\mathbf{x}_i, \mathbf{x}_j), \Theta_{\mathcal{M}}, \Theta_F) \end{aligned}$$

By applying Equation 3 and normalizing,  $Q(i, j)$  is proportional to:

$$\begin{aligned} Q(i, j) &\propto P_{\mathcal{M}}(E_{ij}=1|\Theta_{\mathcal{M}}) \cdot R(f(\mathbf{x}_i, \mathbf{x}_j)|\Theta_{\mathcal{M}}, \Theta_F) \\ &\propto Q'_{\mathcal{M}}(i, j) \cdot A(f(\mathbf{x}_i, \mathbf{x}_j)|\Theta_{\mathcal{M}}, \Theta_F) \end{aligned}$$

where:

$$Q'_{\mathcal{M}}(i, j) = \frac{P_{\mathcal{M}}(E_{ij}=1|\Theta_{\mathcal{M}})}{\sum_{k,l} P_{\mathcal{M}}(E_{kl}=1|\Theta_{\mathcal{M}})} \quad (5)$$

$$A(f(\mathbf{x}_i, \mathbf{x}_j)|\Theta_{\mathcal{M}}, \Theta_F) = \frac{R(f(\mathbf{x}_i, \mathbf{x}_j)|\Theta_{\mathcal{M}}, \Theta_F)}{\sup_{v_l, v_k \in \mathbf{V}} R(f(\mathbf{x}_l, \mathbf{x}_k)|\Theta_{\mathcal{M}}, \Theta_F)}$$

$Q(i, j)$  is therefore proportional to a draw from a proposing matrix  $Q'_{\mathcal{M}}(i, j)$ , moderated by an acceptance probability conditioned on the features  $(A(f(\mathbf{x}_i, \mathbf{x}_j)|\Theta_{\mathcal{M}}, \Theta_F))$ .

The sampling algorithm for AGM is outlined in algorithm 2. The algorithm begins by initializing the nodes and sampling attributes (lines 2-3) and computing a proposing distribution  $Q'_{\mathcal{M}}(i, j)$  from  $\mathcal{M}$  and  $\Theta_{\mathcal{M}}$  (line 4). Then it draws a graph from  $\mathcal{M}$  (line 5) in order to compute the ratios  $R(f(\mathbf{x}_i, \mathbf{x}_j)|\Theta_{\mathcal{M}}, \Theta_F)$  and the corresponding acceptance probabilities (lines 7-8). The main loop (lines 11-17) repeatedly

---

### Algorithm 2 SampleFromAGM ( $\Theta_{\mathcal{M}}, \Theta_{\mathcal{X}}, \Theta_F, G^o$ )

---

```

1: // Draw initial graph and attributes
2:  $\mathbf{V}' = \mathbf{V}$ 
3:  $\mathbf{X}' \sim$  from  $\mathcal{X}$  using  $\Theta_{\mathcal{X}}$ 
4: Calculate  $Q'_{\mathcal{M}}$  in Eq. 5 from  $\mathcal{M}$  and  $\Theta_{\mathcal{M}}$ 
5:  $\mathbf{E}' \sim$  from  $\mathcal{M}$  using  $Q'_{\mathcal{M}}$ 
6: // Compute Acceptance Probabilities
7:  $\mathbf{R}(f(\mathbf{X}, \mathbf{X})) = \frac{P(f(\mathbf{X}^o, \mathbf{X}^o)|\mathbf{E}^o, \Theta_F^o, \Theta_{\mathcal{M}}^o)}{P(f(\mathbf{X}', \mathbf{X}')|\mathbf{E}', \Theta_F', \Theta_{\mathcal{M}}')}$ 
8:  $\mathbf{A}(f(\mathbf{X}, \mathbf{X})) = \frac{R(f(\mathbf{X}, \mathbf{X}))}{\sup[R(f(\mathbf{X}, \mathbf{X}))]}$ 
9: // Reinitialize  $E$  and generate new edges based on  $\mathbf{X}$ 
10:  $\mathbf{E}' = \emptyset$ 
11: while  $|\mathbf{E}'| < |\mathbf{E}^o|$  do
12:    $E'_{ij} \sim \text{multinomial}(Q'_{\mathcal{M}})$ 
13:    $u \sim \text{Uniform}(0,1)$ 
14:   if  $u \leq A(f(\mathbf{x}_i, \mathbf{x}_j))$  then
15:      $\mathbf{E}' = \mathbf{E}' \cup E'_{ij}$ 
16:   end if
17: end while
18: return  $G' = \langle \mathbf{V}', \mathbf{E}', \mathbf{X}' \rangle$ 

```

---

draws a sample from  $Q'_{\mathcal{M}}$  and determines whether to accept it into the graph based on the attributes of the vertices of the proposed edge and the acceptance probabilities (line 14). This loop is repeated until enough edges are inserted into the network.

For certain graph models, the calculation of  $Q'_{\mathcal{M}}$  could be prohibitively expensive, but models such as FCL and KPGM do not need direct computation of  $Q'_{\mathcal{M}}(i, j)$ . Consider the fast Chung Lu (FCL) model, which samples  $(v_i, v_j)$  with probability  $\frac{\theta_{d_i} \theta_{d_j}}{\sum_k \theta_{d_k} \sum_l \theta_{d_l}}$  [18]. This sampling algorithm simplifies to repeated samples from  $Q'_{\mathcal{M}}(i, j)$  as expressed in Equation 5:

$$\frac{\theta_{d_i} \theta_{d_j}}{\sum_k \theta_{d_k} \sum_l \theta_{d_l}} = \frac{\frac{\theta_{d_i} \theta_{d_j}}{2N_e}}{\sum_{k,l} \frac{\theta_{d_k} \theta_{d_l}}{2N_e}} = \frac{P_{CL}(E_{ij}=1|\Theta_{CL})}{\sum_{k,l} P_{CL}(E_{kl}=1|\Theta_{CL})}$$

Furthermore, the KPGM family of models also samples according to Equation 5. To quickly sample KPGMs, each edge is sampled with probability  $\frac{\prod_{k=1}^K \Theta_{KP}(\sigma_{ki}, \sigma_{kj})}{(\sum_{lm} \Theta_{KP})^K}$  [11]. As with FCL, we can rearrange to get:

$$\frac{\prod_{k=1}^K \Theta_{KP}(\sigma_{ki}, \sigma_{kj})}{(\sum_{lm} \Theta_{KP})^K} = \frac{P_{KP}(E_{ij}=1|\Theta_{KP})}{\mathbb{E}[|\mathbf{E}|]} = \frac{P_{KP}(E_{ij}=1|\Theta_{KP})}{\sum_{lm} P_{KP}(E_{lm}=1|\Theta_{KP})}$$

where  $(\sum_{lm} \Theta_{KP})^K = \mathbb{E}[|\mathbf{E}|]$  [16]. Thus, both FCL and KPGM, two popular scalable graph models, iteratively sample  $E_{ij}=1$  from  $Q'_{\mathcal{M}}(i, j)$ , without explicit enumeration of the full  $Q'_{\mathcal{M}}(i, j)$  distribution.

## 4.3 Estimation

Algorithm 3 outlines the framework for learning the parameters required by SampleFromAGM (Algorithm 2). Given a generative model  $\mathcal{M}$ , we assume methods for estimation of parameters  $\Theta_{\mathcal{M}}$  for modeling  $P(\mathbf{E}|\Theta_{\mathcal{M}})$  exist. We assume a model  $P(\mathbf{X}|\Theta_{\mathcal{X}})$  where  $\Theta_{\mathcal{X}}$  can be learned from the vertex attributes, and from which samples  $\mathbf{x} \sim P(\mathbf{X}|\Theta_{\mathcal{X}})$  can be drawn. However, as AGM additionally models correlations in the original network  $P_o(f(\mathbf{X}, \mathbf{X})|\mathbf{E}=1, \Theta_F, \Theta_{\mathcal{M}})$  and model  $P_{\mathcal{M}}(f(\mathbf{X}, \mathbf{X})|E_{ij}=1, \Theta_F, \Theta_{\mathcal{M}})$ , we need to es-

---

**Algorithm 3** LearnAGM ( $\mathcal{M}, \mathcal{X}, G^o$ )

---

- 1: Learn  $\Theta_{\mathcal{M}}$  from  $G^o$  using  $\mathcal{M}$
  - 2: Learn  $\Theta_{\mathcal{X}}$  from  $G^o$  using  $\mathcal{X}$
  - 3: Learn  $\Theta_F$  from  $G^o$
  - 4: **return** ( $\Theta_{\mathcal{M}}, \Theta_{\mathcal{X}}, \Theta_F$ )
- 

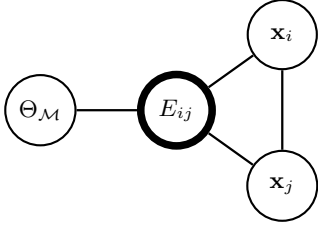


Figure 2: Estimation where attributes are independent of the model parameters given the edges.

timate the parameters  $\Theta_F$  for each. In this subsection, we show how these conditionals can be efficiently estimated.

We begin by making a simplifying assumption about the dependencies between the observed features  $f(\mathbf{x}_i, \mathbf{x}_j)$  and the parameters of the structural graph model ( $\Theta_{\mathcal{M}}$ ), then later demonstrate how to estimate the accept-reject probabilities when this assumption is removed. To start, assume the distribution of the features  $f(\mathbf{x}_i, \mathbf{x}_j)$  is conditionally independent of the graph model parameters  $\Theta_{\mathcal{M}}$ <sup>1</sup>:

$$P(f(\mathbf{x}_i, \mathbf{x}_j) | E_{ij} = 1, \Theta_{\mathcal{M}}, \Theta_F) = P(f(\mathbf{x}_i, \mathbf{x}_j) | E_{ij} = 1, \Theta_F) \quad (6)$$

A graphical representation of this assumption is given in Figure 2. The interpretation is this: if we observe the value of  $E_{ij}$ , then the parameters for the distributions of  $f(\mathbf{x}_i, \mathbf{x}_j)$  do not depend on the generative model  $\mathcal{M}$ . This simplifies our estimation of the distribution, as it removes dependencies on the underlying model  $\mathcal{M}$ . Our assumption of conditional independence helps to improve the efficiency of the algorithm as we can estimate the parameters  $\Theta_F$  using maximum likelihood estimation (MLE).

$$\hat{\Theta}_F = \arg \max_{\Theta_F} \sum_{(v_i, v_j) \in \mathbf{E}} \log P(f(\mathbf{x}_i, \mathbf{x}_j) | E_{ij} = 1, \Theta_F)$$

We will now demonstrate how to estimate the MLE of  $P(f(\mathbf{x}_i, \mathbf{x}_j) | E_{ij} = 1, \Theta_F)$ . First, we will use the correlation of a single binary variable  $w$  across edges as its criteria:

$$f_w(\mathbf{x}_i, \mathbf{x}_j) = \begin{cases} (0, 0) & \text{if } \mathbf{x}_i[w] = 0 \text{ and } \mathbf{x}_j[w] = 0 \\ (1, 1) & \text{if } \mathbf{x}_i[w] = 1 \text{ and } \mathbf{x}_j[w] = 1 \\ (0, 1) & \text{if } \mathbf{x}_i[w] \neq \mathbf{x}_j[w] \end{cases} \quad (7)$$

where  $\mathbf{x}_i(0)$  represents the attribute whose correlation we are trying to encode; for example,  $\mathbf{x}_i(0)$  can be a binary attribute indicating whether the corresponding individual is Conservative or Not Conservative. To maximize the likelihood, we take all  $(v_i, v_j) \in \mathbf{E}$  and count the number of observations of each value the feature can take (in this case,  $\{(0, 0), (0, 1), (1, 1)\}$ ). For example:

$$\hat{\Theta}_{F_w}((0, 0)) = \frac{\sum_{(v_i, v_j) \in \mathbf{E}} \mathbb{I}[(\mathbf{x}_i[w] = 0) \wedge (\mathbf{x}_j[w] = 0)]}{|\mathbf{E}|}$$

For attributes with larger scope  $S$ , the function  $f(\mathbf{x}_i, \mathbf{x}_j)$  makes a mapping over the  $\binom{S+1}{2}$  combinations using the  $S$

<sup>1</sup>These are equally applicable for  $P_o$  and  $P_{\mathcal{M}}$ , so the reference to a specific model is dropped.

possibles values of the characteristic, where  $f(\mathbf{x}_i, \mathbf{x}_j)$  and  $\hat{\Theta}_F$  are given by

$$f_w(\mathbf{x}_i, \mathbf{x}_j) = \begin{cases} (k, k) & \text{if } \mathbf{x}_i[w] = k \wedge \mathbf{x}_j[w] = k \\ (k, l) & \text{if } (\mathbf{x}_i[w] \neq \mathbf{x}_j[w]) \wedge \\ & ((\mathbf{x}_i[w] = k \wedge \mathbf{x}_j[w] = l) \vee \\ & (\mathbf{x}_i[w] = l \wedge \mathbf{x}_j[w] = k)) \end{cases}$$

$$\hat{\Theta}_{F_w}((k, l)) = \frac{\sum_{(v_i, v_j) \in \mathbf{E}} \mathbb{I}[f_w(\mathbf{x}_i, \mathbf{x}_j) = (k, l)]}{|\mathbf{E}|}$$

We can also create an edge function which considers more than a single attribute. We let:

$$f(\mathbf{x}_i, \mathbf{x}_j) = (f_0(\mathbf{x}_i, \mathbf{x}_j), \dots, f_{W-1}(\mathbf{x}_i, \mathbf{x}_j)) \quad (8)$$

meaning the output of  $f(\mathbf{x}_i, \mathbf{x}_j)$  is the multiple pairs of the edge functions  $f_w(\mathbf{x}_i, \mathbf{x}_j)$  defined for the  $W$  different characteristics. For example, when we have two attributes such as Religion and Political Views our corresponding features are  $f(\mathbf{x}_i, \mathbf{x}_j) = (f_0(\mathbf{x}_i, \mathbf{x}_j), f_1(\mathbf{x}_i, \mathbf{x}_j))$ , where  $f_0(\mathbf{x}_i, \mathbf{x}_j)$  refers to the pairing of religious views and  $f_1(\mathbf{x}_i, \mathbf{x}_j)$  to the pairing of political views. Although this edge function has a higher order of magnitude than with single variables, the estimation of  $\hat{\Theta}_F$  can also apply to Equation 8. This allows for modeling a variety of feature functions ( $\hat{\Theta}_F((k_1, l_1), \dots, (k_i, l_i))$ ).

### Removing Conditional Independence Assumption

In Equation 6, we inserted an assumption that the distribution of edge features was independent of the underlying generative graph model  $\mathcal{M}$ . For many generative graph models this is true, such as FCL and KPGM. However, other models are more complicated (e.g., TCL). TCL enforces that the marginal probability of an edge existing in the graph will remain proportional to the product of the degrees [18]. As TCL iteratively lays triangles over an *existing* graph sample, future edge samples are dependent on the previously laid edges in the network. By extension, the samples are dependent on our accept-reject probabilities, as well as our edge function parameters  $\Theta_F$ .

To address this issue, we use the fact that the correct accept-reject probabilities will result in a sampled network  $G'$  where the observed  $f(\mathbf{x}_i, \mathbf{x}_j)$  in  $G'$  equals the observed  $f(\mathbf{x}_i, \mathbf{x}_j)$  in  $G^o$ . Let  $\mathbf{A}^{old}(f(\mathbf{x}_i, \mathbf{x}_j) | \Theta_{\mathcal{M}}, \Theta_F)$  be the initial acceptance probabilities. Define  $\alpha(f(\mathbf{x}_i, \mathbf{x}_j))$  to be the proportion  $P_{\mathcal{M}}(f(\mathbf{x}_i, \mathbf{x}_j) | E_{ij} = 1, \Theta_{\mathcal{M}}, \Theta_F)$  under- or over-samples the desired distribution:

$$P_o(f(\mathbf{x}_i, \mathbf{x}_j) | E_{ij} = 1, \Theta_{\mathcal{M}}, \Theta_F) = \alpha(f(\mathbf{x}_i, \mathbf{x}_j)) \cdot P_{\mathcal{M}}(f(\mathbf{x}_i, \mathbf{x}_j) | E_{ij} = 1, \Theta_{\mathcal{M}}, \Theta_F)$$

Solving for  $\alpha(f(\mathbf{x}_i, \mathbf{x}_j))$  gives:

$$\alpha(f(\mathbf{x}_i, \mathbf{x}_j)) = \frac{P_o(f(\mathbf{x}_i, \mathbf{x}_j) | E_{ij} = 1, \Theta_{\mathcal{M}}, \Theta_F)}{P_{\mathcal{M}}(f(\mathbf{x}_i, \mathbf{x}_j) | E_{ij} = 1, \Theta_{\mathcal{M}}, \Theta_F)}$$

We then update our acceptance probabilities with:

$$\mathbf{A}^{new}(f(\mathbf{x}_i, \mathbf{x}_j) | \Theta_{\mathcal{M}}, \Theta_F) = \alpha(f(\mathbf{x}_i, \mathbf{x}_j)) \cdot \mathbf{A}^{old}(f(\mathbf{x}_i, \mathbf{x}_j) | \Theta_{\mathcal{M}}, \Theta_F)$$

A subsequent graph is then drawn by AGM, but using the updated acceptance rates  $\mathbf{A}^{new}(f(\mathbf{x}_i, \mathbf{x}_j) | \Theta_{\mathcal{M}}, \Theta_F)$ . If AGM previously over-sampled certain edge values, it will adjust and sample them lower. In contrast, if attribute combinations are observed too rarely, AGM will adjust and sample them at a higher rate.

In Algorithm 2, these changes can be implemented by adding another loop around lines 7-17—in which  $\mathbf{A}$  and  $\mathbf{R}$

are updated as described and the edges then drawn again according to the new accept-reject probabilities. We find it takes relatively few iterations of this outer loop to converge on accurate acceptance probabilities.

#### 4.4 Runtime

The benefit of using AGM is the efficiency of the algorithm. Namely, let  $\kappa$  indicate the complexity of sampling from  $Q(i, j) \propto Q'_{\mathcal{M}}(i, j) \cdot A(f(\mathbf{x}_i, \mathbf{x}_j) | \Theta_{\mathcal{M}}, \Theta_F)$ . For example, with FCL  $\kappa = O(1)$ , while for KPGM  $\kappa = O(\log N_v)$ . Since the AGM uses this call once per iteration, its complexity is  $O(\kappa \cdot N_e \cdot \lambda)$ , where  $N_e \cdot \lambda$  corresponds to the total number of iterations to be sampled to obtain a total of  $N_e$  edges and  $\lambda$  is the expected value of the number of trials to get a single edge accepted (a geometric distribution parameterized by the probability a proposed edge is accepted). Then, the total running for FCL is  $O(N_e \cdot \lambda)$  and for KPGM is  $O(N_e \cdot \log N_v \cdot \lambda)$ .

### 5. AGM ANALYTICAL PROPERTIES

We have proposed AGM, a new framework which considers the dependencies between the attributes and edges of a network. Besides its general formulation that can be implemented for a class of generative graph models and its efficient running time, AGM has several important analytical characteristics:

- Theorem 1: AGM approximately draws from the conditional edge distribution  $P_o(\mathbf{E} | \mathbf{X}, \Theta_{\mathcal{M}}, \Theta_F)$ .
- Theorem 2: The expected probability of an edge  $(v_i, v_j)$  in the AGM model is equal to the probability of the edge  $(v_i, v_j)$  in the underlying graph model  $\mathcal{M}$ .
- Corollary 1: The expected degree of a vertex in AGM is equal to its expected degree in  $\mathcal{M}$ .

For clarity, we will discuss these theorems below but defer the proofs to the Appendix. We restate Equation 3 from Section 4.1:

$$P_o(E_{ij} = 1 | f(\mathbf{x}_i, \mathbf{x}_j), \Theta_{\mathcal{M}}, \Theta_F) = P_{\mathcal{M}}(E_{ij} = 1 | \Theta_{\mathcal{M}}) \cdot R(f(\mathbf{x}_i, \mathbf{x}_j) | \Theta_{\mathcal{M}}, \Theta_F)$$

Which is used throughout this section. For simplicity we define:

$$Z = \sum_{i,j}^{N_v, N_v} P_{\mathcal{M}}(E_{ij} = 1 | \Theta_{\mathcal{M}}) \quad C = \sup_{v_k, v_l \in \mathbf{V}} [R(f(\mathbf{x}_i, \mathbf{x}_k) | \Theta_{\mathcal{M}}, \Theta_F)]$$

**Edge Probabilities:** Recall that a draw of  $E_{ij}$  from our proposal distribution  $Q'$  occurs with probability  $P_{\mathcal{M}}(E_{ij} = 1 | \Theta_{\mathcal{M}})/Z$ . In Lemma 2, we show the target conditional distribution  $Q$  (probability of an edge existing *given* the vertex attributes) can be split into a sum of (a) the probability of drawing  $E_{ij} \sim P_{\mathcal{M}}(E_{ij} = 1 | \Theta_{\mathcal{M}})/Z$  and (b) the acceptance probability of  $f(\mathbf{x}_i, \mathbf{x}_j)$  (proof in Appendix B).

LEMMA 2. *For every edge  $(v_i, v_j) \in \mathbf{E}$ :*

$$P_o(E_{ij} = 1 | f(\mathbf{x}_i, \mathbf{x}_j), \Theta_{\mathcal{M}}, \Theta_F) = \sum_1^{Z \cdot C} \left[ \frac{P_{\mathcal{M}}(E_{ij} = 1 | \Theta_{\mathcal{M}})}{Z} \cdot A(f(\mathbf{x}_i, \mathbf{x}_j) | \Theta_{\mathcal{M}}, \Theta_F) \right]$$

This shows the conditional probabilities of the edges can be broken into  $Z \cdot C$  parts, with each part referring to the probability  $(v_i, v_j)$  is drawn and accepted. However, the

probability of an edge existing in the accept-reject process is not the summation of the individual probabilities, but:

$$1 - \left[ 1 - \left( \frac{P_{\mathcal{M}}(E_{ij} = 1 | \Theta_{\mathcal{M}})}{Z} A(f(\mathbf{x}_i, \mathbf{x}_j) | \Theta_{\mathcal{M}}, \Theta_F) \right) \right]^{Z \cdot C}$$

The probability in the square brackets represents the probability of *not* drawing edge  $(v_i, v_j)$  on each iteration. The loop is executed  $Z \cdot C$  times, meaning the quantity on the right is the probability an edge is never sampled. The probability an edge *is* sampled is 1 minus this quantity. We prove the accept-reject process is a good approximation to  $P_o(E_{ij} = 1 | f(\mathbf{x}_i, \mathbf{x}_j), \Theta_{\mathcal{M}}, \Theta_F)$ , as the probability is small, due to the Binomial Approximation [23] (Proof in Appendix C).

THEOREM 1. *For every edge  $(v_i, v_j) \in \mathbf{E}$ :*

$$P_{AGM} := 1 - \left[ 1 - \left[ \frac{P_{\mathcal{M}}(E_{ij} = 1 | \Theta_{\mathcal{M}})}{Z} A(f(\mathbf{x}_i, \mathbf{x}_j) | \Theta_{\mathcal{M}}, \Theta_F) \right] \right]^{Z \cdot C} \approx P_o(E_{ij} = 1 | f(\mathbf{x}_i, \mathbf{x}_j), \Theta_{\mathcal{M}}, \Theta_F)$$

Thus, the AGM sampling formulation provides a good approximation to the true distribution of edges conditioned on the vertex attributes.

**Expected Degrees:** Many generative graph models explicitly model the degree distribution of the network; KPGM has a heavy-tailed degree distribution [11], while the CL family of models has a degree distribution whose expectation is equal to that of the input graph  $G^o$ . We now prove that the expected degree of a vertex with AGM is equal to the expected degree of the vertex as produced by  $\mathcal{M}$ . We begin with Theorem 2, which states that the expected probability of an edge under AGM is equal to the probability of the edge as defined by  $\mathcal{M}$  (Proof in Appendix D).

THEOREM 2. *If the generating distribution  $\mathcal{M}$  is independent from the parameters  $\Theta_F$ , i.e.,  $P_{\mathcal{M}}(E_{ij} = 1 | \Theta_{\mathcal{M}}, \Theta_F) = P_{\mathcal{M}}(E_{ij} = 1 | \Theta_{\mathcal{M}})$ , then*

$$\mathbb{E}_{\mathbf{X}} [P_o(E_{ij} = 1 | f(\mathbf{x}_i, \mathbf{x}_j), \Theta_{\mathcal{M}}, \Theta_F)] = P_{\mathcal{M}}(E_{ij} = 1 | \Theta_{\mathcal{M}})$$

Using Theorem 2, we can show that the expected value of the vertex degree under AGM is equal to the expected value of the vertex degree under  $\mathcal{M}$  (Proof in Appendix E).

COROLLARY 1. *If the generating distribution  $\mathcal{M}$  is independent from the parameters  $\Theta_F$ , i.e.,  $P_{\mathcal{M}}(E_{ij} = 1 | \Theta_{\mathcal{M}}, \Theta_F) = P_{\mathcal{M}}(E_{ij} = 1 | \Theta_{\mathcal{M}})$ , then  $\mathbb{E}_{\mathbf{X}} [d_i] = \mathbb{E}_{\mathcal{M}} [d_i]$ .*

Corollary 1 states that regardless of the generating distribution, if the attributes are independent of the generating distribution we will draw the same degrees. Applying AGM with CL models will provably have the same expected degree distribution as the input graph, while applying AGM with KPGM will retain KPGM's expected degrees.

### 6. EXPERIMENTS

To demonstrate the flexibility of AGM, we use four popular generative graph models as proposing distributions: fast Chung Lu (FCL), transitive Chung Lu (TCL), and the Kronecker Product Graph Model (KPGM) with a  $2 \times 2$  and  $3 \times 3$  initialization matrix. Our experiments will show that the

AGM versions of each of the underlying generative models have the same structure as the structural model, but capture the Pearson correlations of the attributes as well. We implemented learning and generation for FCL and TCL directly, only modifying the generation step for AGM-FCL and AGM-TCL. For the two KPGMs, we utilized the authors' publicly distributed code for learning the parameter matrix<sup>2</sup>, but augmented the generation process to incorporate correlation. We also compare against the Multiplicative Attribute Graph (MAG) model; as MAG is intended for learning latent attributes, we augment the model for usage in this domain to utilize the known correlations<sup>3</sup>.

## 6.1 Datasets

We evaluate our models on two network data sets: the CoRA citations network [14] and Facebook wall postings from Purdue University. For CoRA, we consider the categorical feature "AI" (1 iff the topic of a paper lies in the field of Artificial Intelligence). CoRA contains 11,881 vertices with 31,482 citations between them, and the AI feature is highly correlated across edges. We model the distribution of attributes  $P(\mathbf{X}|\Theta_{\mathcal{X}})$  by maximizing the likelihood of Bernoulli trials; the probability of a label being AI is proportional to the number of AI labels in the CoRA dataset.

The Facebook network has 449,748 user vertices with 1,016,621 wall postings between them. We estimate model parameters from all visible vertices, ignoring instances for which privacy settings prevent us from accessing the information. We consider two attributes: *Religion* (1 iff Religious Views contains the string "christ") and *Political* (1 iff Political Views contains the string "conservative"). Here, we model the distribution of attributes  $P(\mathbf{X}|\Theta_{\mathcal{X}})$  as a bivariate multinomial distribution and use maximum likelihood estimation to estimate the parameters. For each network, the vertex attributes are drawn independently and identically distributed from their respective  $P(\mathbf{X}|\Theta_{\mathcal{M}})$  distributions.

## 6.2 MAG Implementation

Rather than use the normal fitting process which assumes latent attributes, we use the observed attributes to calculate the probability of seeing an edge between particular attributes. This allows us to directly calculate the affinity matrix parameter for the MAG model. On the single-attribute CoRA dataset this calculation is simple, as  $P(E_{ij} = 1|f(\mathbf{x}_i, \mathbf{x}_j))$  can be estimated using the number of edges between vertices with a specific pair of attribute values.

This calculation is not as simple on the Facebook dataset as we must estimate the probabilities for two attributes. Let  $\mathbf{x}_i[0]$  represent the *Political* attribute and  $\mathbf{x}_i[1]$  represent the *Religion* attribute. As MAG treats edge affinities as independent, we decompose  $P(E_{ij} = 1|f(\mathbf{x}_i, \mathbf{x}_j))$  into two independent components:  $P(E_{ij} = 1|f(\mathbf{x}_i[0], \mathbf{x}_j[0]))$  and  $P(E_{ij} = 1|f(\mathbf{x}_i[1], \mathbf{x}_j[1]))$ . Then, for every attribute permutation of two vertices we can estimate  $P(E_{ij} = 1|f(\mathbf{x}_i, \mathbf{x}_j))$  from the observed data and set up a system of equations. Solving this system of equations gives us  $P(E_{ij} = 1|f(\mathbf{x}_i[0], \mathbf{x}_j[0]))$  and  $P(E_{ij} = 1|f(\mathbf{x}_i[1], \mathbf{x}_j[1]))$ , which are the edge affinities. However, as the real data has dependencies between the attributes, there is no exact solution for this system and we must use an approximation

<sup>2</sup>Source available at <http://snap.stanford.edu/>

<sup>3</sup>Source also available at <http://snap.stanford.edu/>

instead. We chose an approximation that kept the affinity for two non-conservative vertices equal to the affinity for two non-religious vertices.

Finally, we must take into account the vertices in the Facebook network with unobserved attributes. These vertices had much lower degrees than observed vertices in the original network. We chose to create a third attribute, observed vs. unobserved, when generating the graph. Vertices labeled unobserved still have their other attributes simulated, but have their edge affinities reduced to the rate of unobserved vertices in the original graph.

## 6.3 AGM Implementations

In order to test the correlations of generated networks, every vertex was assigned attributes drawn from the prior distribution of vertex attributes as computed on the real world network and independent of the other vertices. Tests were run for each of our four generative models, with each generative model proposing edges which are then either accepted or rejected. The end result is a joint sampling of attributes and edges, with the edges having been conditioned on the attributes.

**Edge Functions:** For the CoRA dataset, we have one feature to consider (AI) and we use the edge feature for a single attribute as discussed in Section 4.3, Equation 7. For the Facebook dataset, we have two attributes to consider (Religion and Politics). This corresponds to the edge features discussed in Equation 8, which models the joint conditionals of the two attributes, allowing AGM to model the correlations of each.

## 6.4 Graph Structure

We begin our analysis by determining whether AGM produces graphs which alter the structure of the proposing distributions. First, the *degree distributions* for each dataset are plotted in Figure 3a-b and compared against some of the models (to reduce clutter we omit the simpler FCL and KPGM<sub>2x2</sub> in this part of the analysis). For each of these plots, the x-axis represents vertex degrees, while the y-axis represents the complementary cumulative distribution function (CCDF). For any point on the x-axis, the y-axis is the proportion of vertices with the corresponding degree (on the x-axis) or higher. The degree distribution of CoRA (Figure 3.a) shows that AGM-TCL closely matches the degree distribution of TCL, while AGM-KPGM<sub>3x3</sub> closely matches the degree distribution of KPGM<sub>3x3</sub>. For the Facebook network (Figure 3.b), which has a more complicated edge feature distribution, AGM-TCL and AGM-KPGM<sub>3x3</sub> also match their corresponding proposing distributions (TCL and KPGM<sub>3x3</sub>).

We extend our analysis of the degrees in Table 1, where we show the *KS-Statistic* between the degree distribution of each AGM model and its corresponding generative model (FCL, TCL, KPGM<sub>2x2</sub>, KPGM<sub>3x3</sub>). We see no change between the original model and corresponding AGM distributions, since for all but one test we are unable to reject the null hypothesis that the distributions are equal ( $p = 0.01$ ). TCL is the only rejection, which is due to TCL not having dyadic independence. However, empirically AGM-TCL performs comparably to TCL, meaning we can effectively model degree distributions even when there is edge dependence.

In Figure 4.a-b, we show the local *Clustering Coefficient* distributions, which measure the number of triangles each

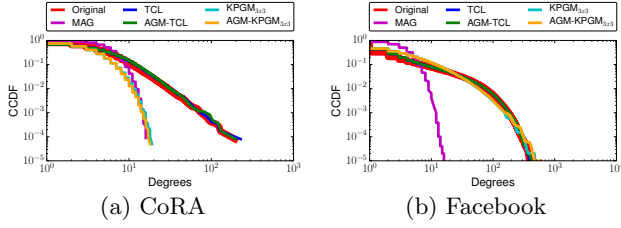


Figure 3: Degree distributions for each network

| Dataset  | AGM KS-Distance (Degree Distribution) |       |                     |                     |
|----------|---------------------------------------|-------|---------------------|---------------------|
|          | FCL                                   | TCL   | KPGM <sub>2x2</sub> | KPGM <sub>3x3</sub> |
| CoRA     | 0.003                                 | 0.021 | 0.004               | 0.009               |
| Facebook | 0.003                                 | 0.002 | 0.004               | 0.004               |

Table 1: KS-Statistic for AGM degree distributions against corresponding proposal distributions.

vertex has compared to the number of triangles the vertex *could have* given its degree. KPGM<sub>3x3</sub> does not explicitly model the clustering coefficients in the network, thus the low clustering the model produces is expected. Further, since its corresponding generative model does not generate networks with high clustering, neither does AGM-KPGM<sub>3x3</sub>. In contrast, TCL was explicitly designed to incorporate transitivity into the generative process by incorporating two step random walks. As TCL proposes a larger number of triangles, the networks produced by AGM-TCL will also have a higher numbers of triangles. More generally, AGM does not interfere with structural characteristics such as degree and clustering that the underlying generative graph model provides, meaning AGM is not limited to a single characterization of structural components.

## 6.5 Feature Correlations

Lastly, we demonstrate how our formulation can capture accurate correlations between the feature instances. We begin by analyzing the correlations of the simpler CoRA network (with a single attribute to model), then move to the more complicated Facebook network with two attributes.

As seen in Table 2, the initial CoRA network contains a high level of correlation (.837), which none of our underlying generative models capture (FCL, TCL, KPGM<sub>2x2</sub>, and KPGM<sub>3x3</sub>). Introducing our AGM framework in conjunction with each one, we see that every AGM version of the models has correlation very close to the original network. Further, recall that each AGM method accomplishes this without disrupting the underlying structural distribution (prior subsection). Thus, AGM is jointly modeling both structural components and the correlation of the attribute.

When MAG is presented just the single attribute found in CoRA it captures the correlation as well but when the number of attributes is expanded MAG begins to break down.

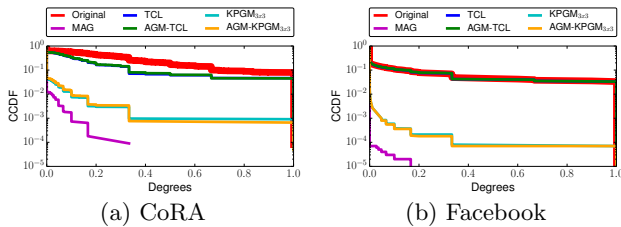


Figure 4: Clustering Coefficients for each network.

| Model                   | Correlations |              |              |              |
|-------------------------|--------------|--------------|--------------|--------------|
|                         | CoRA         | Facebook     |              |              |
|                         |              | R            | P            | RP           |
| Original                | 0.837        | 0.108        | .211         | 0.106        |
| MAG                     | <b>0.835</b> | 0.584        | 0.436        | 0.002        |
| FCL                     | 0.005        | 0.001        | 0.001        | -0.001       |
| AGM-FCL                 | <b>0.835</b> | <b>0.130</b> | <b>0.223</b> | <b>0.095</b> |
| TCL                     | -0.006       | 0.001        | 0.001        | 0.001        |
| AGM-TCL                 | <b>0.856</b> | <b>0.128</b> | <b>0.219</b> | <b>0.093</b> |
| KPGM <sub>2x2</sub>     | -0.002       | 0.001        | -0.002       | 0.001        |
| AGM-KPGM <sub>2x2</sub> | <b>0.839</b> | <b>0.131</b> | <b>0.221</b> | <b>0.095</b> |
| KPGM <sub>3x3</sub>     | -0.004       | 0.001        | -0.001       | 0.001        |
| AGM-KPGM <sub>3x3</sub> | <b>0.841</b> | <b>0.132</b> | <b>0.221</b> | <b>0.092</b> |

Table 2: Correlations for attributes in each dataset. Bold indicates within .05 of the original network correlation.

MAG does not accurately model the joint distribution of edges given vertex attributes and does not model the correct correlations (Table 2).

In contrast, for each underlying proposal distribution, AGM’s augmentation allows the proposal distribution to model the edge correlations. This observation holds for each possible correlated attribute pair: Religion (R), Politics (P), and the correlation of Religion with Politics *across* edges. Again, these correlations are being modeled while the corresponding structural behavior remains unchanged, meaning AGM models both the attributes and structure of the graph.

## 7. CONCLUSIONS

In this work, we have introduced a new framework, the Attributed Graph Model (AGM), which enables conditional sampling of graph structure based on vertex attributes. We have shown that AGM can be combined with several generative graph models, i.e., fast Chung Lu (FCL), transitive Chung Lu (TCL), and Kronecker Product Graph Model (KPGM). AGM has efficient learning and sampling mechanisms that accurately replicate *both* the characteristics of the underlying graph structure and the vertex attribute correlations. Further, we demonstrated empirically that our approach offers improvements compared to the competing Multiplicative Attributed Graph (MAG) model. Notably, our AGM framework enables efficient generation of *large-scale* network structure with *homophily*.

AGM makes a single draw from the distribution of attributes on vertices, followed by a draw from the distribution of edges conditioned on attributes. This process produces a sample from the *joint* distribution of attributes and edges. In future work we will consider models from the area of statistical relational learning, which represent and reason about complex correlations among linked vertices through the joint distribution of vertex attributes given a set of edges. AGM could be combined with these types of models by first drawing a joint sample of edges conditioned on the vertex attributes, followed by drawing a joint sample of vertex attributes given the new edges.

The above search over the joint distribution points to additional interesting problems in *temporal* domains. Namely, the vertices and edges that occur in a timewindow  $t$  are likely to be correlated with the vertices and edges that occur in previous timesteps, meaning we can jointly draw edges and attributes conditioned on prior timesteps. Interesting questions in this scenario would involve assessing *stationarity* of parameters, as well as investigating mechanisms to identify and model periodic behavior.

## Acknowledgements

We thank our anonymous reviewers for helpful feedback. This research is supported by NSF under contract numbers IIS-1149789, CCF-0939370 and IIS-1219015. Additionally this work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements either expressed or implied, of NSF or the U.S. Government.

## APPENDIX

### A. PROOF OF LEMMA 1

We wish to model the conditional probability of an edge existing in the original network using the proposing distribution  $\mathcal{M}$ . This results in a *Ratio* representing how close the two distributions are to each other, which we denote  $R(f(\mathbf{x}_i, \mathbf{x}_j)|\Theta_{\mathcal{M}}, \Theta_F) = \frac{P_o(E_{ij}=1|f(\mathbf{x}_i, \mathbf{x}_j), \Theta_{\mathcal{M}}, \Theta_F)}{P_{\mathcal{M}}(E_{ij}=1|f(\mathbf{x}_i, \mathbf{x}_j), \Theta_{\mathcal{M}}, \Theta_F)}$ :

$$\begin{aligned} P_o(E_{ij} = 1|f(\mathbf{x}_i, \mathbf{x}_j), \Theta_{\mathcal{M}}, \Theta_F) \\ &= P_{\mathcal{M}}(E_{ij} = 1|\Theta_{\mathcal{M}}) R(f(\mathbf{x}_i, \mathbf{x}_j)|\Theta_{\mathcal{M}}, \Theta_F) \\ &= P_{\mathcal{M}}(E_{ij} = 1|\Theta_{\mathcal{M}}) \frac{P_o(E_{ij} = 1|f(\mathbf{x}_i, \mathbf{x}_j), \Theta_{\mathcal{M}}, \Theta_F)}{P_{\mathcal{M}}(E_{ij} = 1|f(\mathbf{x}_i, \mathbf{x}_j), \Theta_{\mathcal{M}}, \Theta_F)} \quad (9) \\ &= P_o(E_{ij} = 1|f(\mathbf{x}_i, \mathbf{x}_j), \Theta_{\mathcal{M}}, \Theta_F) \end{aligned}$$

where  $P_{\mathcal{M}}(E_{ij} = 1|\Theta_{\mathcal{M}}) = P_{\mathcal{M}}(E_{ij} = 1|f(\mathbf{x}_i, \mathbf{x}_j), \Theta_{\mathcal{M}}, \Theta_F)$ . We simplify  $R(f(\mathbf{x}_i, \mathbf{x}_j)|\Theta_{\mathcal{M}}, \Theta_F)$ :

$$\begin{aligned} R(f(\mathbf{x}_i, \mathbf{x}_j)|\Theta_{\mathcal{M}}, \Theta_F) \\ &= \frac{P_o(E_{ij} = 1|\Theta_{\mathcal{M}}, \Theta_F) \frac{P_o(f(\mathbf{x}_i, \mathbf{x}_j)|E_{ij}=1, \Theta_{\mathcal{M}}, \Theta_F)}{P_o(f(\mathbf{x}_i, \mathbf{x}_j)|\Theta_{\mathcal{M}}, \Theta_F)}}{P_{\mathcal{M}}(E_{ij} = 1|\Theta_{\mathcal{M}}) \frac{P_{\mathcal{M}}(f(\mathbf{x}_i, \mathbf{x}_j)|E_{ij}=1, \Theta_{\mathcal{M}}, \Theta_F)}{P_{\mathcal{M}}(f(\mathbf{x}_i, \mathbf{x}_j)|\Theta_{\mathcal{M}}, \Theta_F)}} \\ &= \frac{P_o(f(\mathbf{x}_i, \mathbf{x}_j)|E_{ij} = 1, \Theta_{\mathcal{M}}, \Theta_F)}{P_{\mathcal{M}}(f(\mathbf{x}_i, \mathbf{x}_j)|E_{ij} = 1, \Theta_{\mathcal{M}}, \Theta_F)} \cdot \left[ \frac{P_{\mathcal{M}}(f(\mathbf{x}_i, \mathbf{x}_j)|\Theta_{\mathcal{M}}, \Theta_F)}{P_o(f(\mathbf{x}_i, \mathbf{x}_j)|\Theta_{\mathcal{M}}, \Theta_F)} \right] \quad (10) \end{aligned}$$

Here we used our assumption on the prior to cancel the terms. Consider normalization terms in the brackets<sup>4</sup>:

$$\begin{aligned} P_{\mathcal{M}}(f(\mathbf{x}_i, \mathbf{x}_j)|\Theta_F) &= P_{\mathcal{M}}(E_{ij} = 1)P_{\mathcal{M}}(f(\mathbf{x}_i, \mathbf{x}_j)|E_{ij} = 1, \Theta_F) \\ &\quad + P_{\mathcal{M}}(E_{ij} = 0)P_{\mathcal{M}}(f(\mathbf{x}_i, \mathbf{x}_j)|E_{ij} = 0, \Theta_F) \\ P_o(f(\mathbf{x}_i, \mathbf{x}_j)|\Theta_F) &= P_{\mathcal{M}}(E_{ij} = 1)P_o(f(\mathbf{x}_i, \mathbf{x}_j)|E_{ij} = 1, \Theta_F) \\ &\quad + P_{\mathcal{M}}(E_{ij} = 0)P_o(f(\mathbf{x}_i, \mathbf{x}_j)|E_{ij} = 0, \Theta_F) \end{aligned}$$

For dense matrices this would need to be computed exactly, and for sparse matrices the estimates for  $P(f(\mathbf{x}_i, \mathbf{x}_j)|E_{ij} = 0, \Theta_F)$  can be approximated by sampling. However, for sparse matrices this can be simplified further as  $P_{\mathcal{M}}(E_{ij} = 0)$  dominates the sum for each equation as all edges exist with probability near 0:

$$\begin{aligned} P_{\mathcal{M}}(f(\mathbf{x}_i, \mathbf{x}_j)|\Theta_F) &\approx P_{\mathcal{M}}(E_{ij} = 0)P_{\mathcal{M}}(f(\mathbf{x}_i, \mathbf{x}_j)|E_{ij} = 0, \Theta_F) \\ P_o(f(\mathbf{x}_i, \mathbf{x}_j)|\Theta_F) &\approx P_{\mathcal{M}}(E_{ij} = 0)P_o(f(\mathbf{x}_i, \mathbf{x}_j)|E_{ij} = 0, \Theta_F) \end{aligned}$$

Further,  $P_o(f(\mathbf{x}_i, \mathbf{x}_j)|E_{ij} = 0, \Theta_F)$  and  $P_{\mathcal{M}}(f(\mathbf{x}_i, \mathbf{x}_j)|E_{ij} = 0, \Theta_F)$  define distributions over nearly every possible pair of vertices in  $\mathbf{V} \times \mathbf{V}$ . As  $\mathbf{x}_i \sim P(\mathbf{X}|\Theta_{\mathcal{X}})$  for both distributions,  $P_o(f(\mathbf{x}_i, \mathbf{x}_j)|E_{ij} = 0, \Theta_F) \approx P_{\mathcal{M}}(f(\mathbf{x}_i, \mathbf{x}_j)|E_{ij} = 0, \Theta_F)$ . Thus, in Equation 10 the ratio in brackets is approximately 1. Inserting this result into Equation 9, the conditional is<sup>5</sup>:

<sup>4</sup>For the following discussion we omit  $\Theta_{\mathcal{M}}$  from the equations to reduce clutter, as they appear in each term.

<sup>5</sup>With parameters  $\Theta_{\mathcal{M}}$  reintroduced.

$$\begin{aligned} P_o(E_{ij} = 1|f(\mathbf{x}_i, \mathbf{x}_j), \Theta_{\mathcal{M}}, \Theta_F) \\ &= P_{\mathcal{M}}(E_{ij} = 1|\Theta_{\mathcal{M}}) \cdot R(f(\mathbf{x}_i, \mathbf{x}_j)|\Theta_{\mathcal{M}}, \Theta_F) \\ &\approx P_{\mathcal{M}}(E_{ij} = 1|\Theta_{\mathcal{M}}) \cdot \frac{P_o(f(\mathbf{x}_i, \mathbf{x}_j)|E_{ij} = 1, \Theta_{\mathcal{M}}, \Theta_F)}{P_{\mathcal{M}}(f(\mathbf{x}_i, \mathbf{x}_j)|E_{ij} = 1, \Theta_{\mathcal{M}}, \Theta_F)} \end{aligned}$$

### B. PROOF OF LEMMA 2

We begin by applying Equation 3:

$$\begin{aligned} P_o(E_{ij}|f(\mathbf{x}_i, \mathbf{x}_j), \Theta_{\mathcal{M}}, \Theta_F) \\ &= P_{\mathcal{M}}(E_{ij} = 1|\Theta_{\mathcal{M}}) R(f(\mathbf{x}_i, \mathbf{x}_j)|\Theta_{\mathcal{M}}, \Theta_F) \\ &= \sum_1^Z \left[ \frac{P_{\mathcal{M}}(E_{ij} = 1|\Theta_{\mathcal{M}})}{Z} \cdot R(f(\mathbf{x}_i, \mathbf{x}_j)|\Theta_{\mathcal{M}}, \Theta_F) \right] \\ &= \sum_1^{Z \cdot C} \left[ \frac{P_{\mathcal{M}}(E_{ij} = 1|\Theta_{\mathcal{M}})}{Z} \left( \frac{1}{C} \cdot R(f(\mathbf{x}_i, \mathbf{x}_j)|\Theta_{\mathcal{M}}, \Theta_F) \right) \right] \\ &= \sum_1^{Z \cdot C} \left[ \frac{P_{\mathcal{M}}(E_{ij} = 1|\Theta_{\mathcal{M}})}{Z} \cdot A(f(\mathbf{x}_i, \mathbf{x}_j)|\Theta_{\mathcal{M}}, \Theta_F) \right] \end{aligned}$$

where in the second step we have multiplied every piece of the summation by  $\frac{1}{Z}$  but summed  $Z$  times and in the third step where we again multiply every instance by  $\frac{1}{C}$ , but additionally sum over the quantity  $C$  times.

### C. PROOF OF THEOREM 1

The Binomial Approximation [23] states that for values  $z$  close to 0,  $(1 + z)^\alpha = 1 + \alpha z$ . Here, our individual draws and corresponding accept-reject probabilities are close to 0 for real-world networks, meaning:

$$\begin{aligned} 1 - \left[ 1 - \left[ \frac{P_{\mathcal{M}}(E_{ij} = 1|\Theta_{\mathcal{M}})}{Z} A(f(\mathbf{x}_i, \mathbf{x}_j)|\Theta_{\mathcal{M}}, \Theta_F) \right] \right]^{Z \cdot C} \\ \approx 1 - \left[ 1 - Z \cdot C \cdot \left[ \frac{P_{\mathcal{M}}(E_{ij} = 1|\Theta_{\mathcal{M}}, \Theta_F)}{Z} A(f(\mathbf{x}_i, \mathbf{x}_j)|\Theta_{\mathcal{M}}, \Theta_F) \right] \right] \\ = Z \cdot C \cdot \left[ \frac{P_{\mathcal{M}}(E_{ij} = 1|\Theta_{\mathcal{M}})}{Z} A(f(\mathbf{x}_i, \mathbf{x}_j)|\Theta_{\mathcal{M}}, \Theta_F) \right] \\ = \sum_1^{Z \cdot C} \left[ \frac{P_{\mathcal{M}}(E_{ij} = 1|\Theta_{\mathcal{M}})}{Z} \cdot A(f(\mathbf{x}_i, \mathbf{x}_j)|\Theta_{\mathcal{M}}, \Theta_F) \right] \\ = P_o(E_{ij} = 1|f(\mathbf{x}_i, \mathbf{x}_j), \Theta_{\mathcal{M}}, \Theta_F) \end{aligned}$$

where in the last step we have applied Lemma 1.

### D. PROOF OF THEOREM 2

We marginalize over the combinations of attributes that can exist on the vertices.

$$\begin{aligned} \mathbb{E}_{\mathbf{X}} [P_o(E_{ij} = 1|f(\mathbf{x}_i, \mathbf{x}_j), \Theta_{\mathcal{M}}, \Theta_F)] \\ &= \sum_{\mathbf{x}_i \in \mathbf{X}_i} \sum_{\mathbf{x}_j \in \mathbf{X}_j} P_o(f(\mathbf{x}_i, \mathbf{x}_j)|E_{ij} = 1, \Theta_{\mathcal{M}}, \Theta_F) P_{\mathcal{M}}(E_{ij} = 1|\Theta_{\mathcal{M}}) \\ &= P_{\mathcal{M}}(E_{ij} = 1|\Theta_{\mathcal{M}}) \sum_{\mathbf{x}_i \in \mathbf{X}_i} \sum_{\mathbf{x}_j \in \mathbf{X}_j} P_o(f(\mathbf{x}_i, \mathbf{x}_j)|E_{ij} = 1, \Theta_{\mathcal{M}}, \Theta_F) \\ &= P_{\mathcal{M}}(E_{ij} = 1|\Theta_{\mathcal{M}}) \end{aligned}$$

where in the second step we observed the summation must sum to 1 to be a valid probability distribution.

### E. PROOF OF COROLLARY 1

Apply Theorem 2 and linearity of expectation:

$$\mathbb{E}_{\mathbf{X}} [d_i] = \sum_{v_j} P_{\mathcal{M}}(E_{ij} = 1|\Theta_{\mathcal{M}}) = \mathbb{E}_{\mathcal{M}} [d_i]$$

## 8. REFERENCES

- [1] E. M. Airoldi, D. M. Blei, S. E. Fienberg, and E. P. Xing. Mixed membership stochastic blockmodels. *J. Mach. Learn. Res.*, 9:1981–2014, June 2008.
- [2] F. Chung and L. Lu. The average distances in random graphs with given expected degrees. *Internet Mathematics*, 1, 2002.
- [3] P. Erdos and A. Renyi. On the evolution of random graphs. *Publ. Math. Inst. Hung. Acad. Sci.*, 5:17–61, 1960.
- [4] L. Getoor. *Learning Statistical Models from Relational Data*. PhD thesis, Stanford, 2001.
- [5] M. S. Handcock. Assessing degeneracy in statistical models of social networks.
- [6] P. D. Hoff, A. E. Raftery, and M. S. Handcock. Latent space approaches to social network analysis. *Journal of the American Statistical Association*, 97:1090+, December 2002.
- [7] M. Kim and J. Leskovec. Multiplicative attribute graph model of real-world networks. *Int Mathematics*, 2012.
- [8] T. G. Kolda, A. Pinar, T. Plantenga, and C. Seshadhri. A scalable generative graph model with community structure. arXiv:1302.6636, February 2013. revised March 2013.
- [9] D. Koller and N. Friedman. *Probabilistic Graphical Models: Principles and Techniques - Adaptive Computation and Machine Learning*. The MIT Press, 2009.
- [10] T. La Fond and J. Neville. Randomization tests for distinguishing social influence and homophily effects. WWW '10, 2010.
- [11] J. Leskovec, D. Chakrabarti, J. Kleinberg, C. Faloutsos, and Z. Ghahramani. Kronecker graphs: An approach to modeling networks. *J. Mach. Learn. Res.*, 2010.
- [12] J. Leskovec and E. Horvitz. Planetary-scale views on a large instant-messaging network. *Proceedings of the 17th international conference on World Wide Web*, 2008.
- [13] F. Liang, C. Liu, and R. J. Carrol. *Advanced Markov chain Monte Carlo methods: learning from past samples*. Wiley Series in Computational Statistics. Wiley, New York, NY, 2010.
- [14] A. McCallum, K. Nigam, J. Rennie, and K. Seymore. Automating the construction of internet portals with machine learning. *Information Retrieval*, 3(2):127–163, 2000.
- [15] M. McPherson, L. Smith-Lovin, and J. M. Cook. Birds of a feather: Homophily in social networks. *Annual review of sociology*, pages 415–444, 2001.
- [16] S. Moreno, S. Kirshner, J. Neville, and S. Vishwanathan. Tied kronecker product graph models to capture variance in network populations. In *Allerton'10*, pages 17–61, 2010.
- [17] J. Neville, O. Şimşek, D. Jensen, J. Komoroske, K. Palmer, and H. Goldberg. Using relational knowledge discovery to prevent securities fraud. In *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, KDD '05, pages 449–458, New York, NY, USA, 2005. ACM.
- [18] J. J. Pfeiffer III, T. La Fond, S. Moreno, and J. Neville. Fast generation of large scale social networks while incorporating transitive closures. In *(SocialCom)*, 2012.
- [19] A. Pinar, C. Seshadhri, and T. G. Kolda. The similarity between stochastic kronecker and chung-lu graph models. *CoRR*, abs/1110.4925, 2011.
- [20] G. Robins, P. Pattison, Y. Kalish, and D. Lusher. An introduction to exponential random graph (p\*) models for social networks. *Social Networks*, May 2007.
- [21] C. R. Shalizi and A. Rinaldo. Consistency under sampling of exponential random graph models. *The Annals of Statistics*, 41(2):508–535, 04 2013.
- [22] C. R. Shalizi and A. C. Thomas. Homophily and contagion are generically confounded in observational social network studies. *Sociological Methods and Research*, 40.2:211–239, 2011.
- [23] D. Stirling. *Mathematical Analysis: A Fundamental and Straightforward Approach*. MATHEMATICS AND ITS APPLICATIONS. Prentice Hall, 1987.
- [24] B. Taskar. *Learning structured prediction models: a large margin appr.* PhD thesis, Stanford, 2004.