

State-Dependent Kernel Selection for Conditional Sampling of Graphs

James A. Scott and Axel Gandy

Abstract

This article introduces new efficient algorithms for two problems: sampling conditional on vertex degrees in unweighted graphs, and conditional on vertex strengths in weighted graphs. The resulting conditional distributions provide the basis for exact tests on social networks and two-way contingency tables. The algorithms are able to sample conditional on the presence or absence of an arbitrary set of edges. Existing samplers based on MCMC or sequential importance sampling are generally not scalable; their efficiency can degrade in large graphs with complex patterns of known edges. MCMC methods usually require explicit computation of a Markov basis to navigate the state space; this is computationally intensive even for small graphs. Our samplers do not require a Markov basis, and are efficient both in sparse and dense settings. The key idea is to carefully select a Markov kernel on the basis of the current state of the chain. We demonstrate the utility of our methods on a real network and contingency table. Supplementary materials for this article are available online.

Keywords: Contingency Table, Degree Sequence, Exact Test, Markov Chain Monte Carlo, Random Network

James A. Scott is PhD Candidate, Department of Mathematics, Imperial College London, SW7 2AZ (e-mail: james.scott15@imperial.ac.uk). Professor Axel Gandy is Chair in Statistics, Department of Mathematics, Imperial College London, SW7 2AZ (e-mail: a.gandy@imperial.ac.uk).

1 INTRODUCTION

1.1 Background

Inference on graphs conditional on vertex-level data arises in sociology (Holland and Leinhardt, 1981), psychology (Rasch, 1960), community ecology (Connor and Simberloff, 1979) and categorical data analysis (Agresti, 1992). Testing in this setting can be based on asymptotic results. However, these approximations can be poor in sparse graphs. An alternative approach is to use sampling to approximate the distribution of test statistics. This leads to two difficult problems: sampling graphs with given degrees and sampling weighted graphs with given strengths. Researchers often additionally need to condition on the presence or absence of certain edges in the graphs.

Several existing methods construct Markov chains in this setting. Unfortunately, if the null distribution conditions on known edges, it is difficult to construct a connected Markov chain on the relevant state space. Existing methods either specialize to particular patterns of known edges, or in the general case, use techniques from computational algebra to compute a Markov basis (Diaconis and Sturmfels, 1998; Aoki and Takemura, 2005; Rapallo, 2006). These methods are computationally intensive and are impractical for graphs with more than a few vertices.

We propose new MCMC methods that use *state-dependent mixing* of Markov kernels. The idea is to intelligently select a ‘good’ kernel for the current state of the chain. This technique allows us to construct samplers that require little tuning to the problem at hand, and do not require computation of a Markov basis. The samplers are irreducible in the face of arbitrary patterns of known edges and are efficient both in sparse and dense graphs.

The first focus of this article is on sampling unweighted graphs with prescribed vertex degrees. This problem arises in carrying out exact tests. Consider, for example, social network analysis. A social network equipped with a dichotomous relation can be expressed as a digraph. Vertices represent actors, with edges representing the applicability of the relation between actors. Frequently researchers are interested in testing the presence of reciprocity in the network; defined loosely as a preference for mutual dyads in the digraph. Holland and Leinhardt (1981) introduce an exponential family model under which the UMPU test for reciprocity conditions on the observed degree sequences. Conditioning removes nuisance parameters from the null, and the resulting

distribution is then uniform on the reference set.

The complex interactions that result from conditioning render analytic analysis of the null distribution difficult or impossible. Efforts have been made to develop recursive formulas to enumerate all graphs in the reference set (Wasserman and Faust, 1994), however these are impractical for even moderately sized graphs. If we can sample graphs uniformly then we can approximate the null distribution of a test statistic. Thus, the literature has focused on simulation, whose methods can broadly be divided into two camps; Markov Chain Monte Carlo (MCMC) (Rao et al., 1996; Roberts, 2000; Milo et al., 2002; McDonald et al., 2007; Verhelst, 2008) and sequential importance sampling (SIS) (Snijders, 1991; Zhang and Chen, 2013; Chen et al., 2005; Bayati et al., 2010).

Sampling binary tables with given margins is equivalent to sampling undirected bipartite graphs with given vertex degrees. This is applied in community ecology to test for patterns in co-occurrence tables, and in psychometrics to test the Rasch hypothesis (see Gustafsson, 1980). Thus, there exists a substantial parallel literature along these lines.

Most MCMC algorithms proposed for sampling graphs are adaptations of methods proposed for zero-one tables. Typically, they use a combination of ‘switch’ moves (Ryser, 1963) and additional moves to maintain irreducibility in the face of structural zeros. Rao et al. (1996) and McDonald et al. (2007) consider ‘compact alternating hexagon’ and ‘hexad’ updates respectively. Most proposed methods suffer from poor mixing in unbalanced matrices, rendering them impractical for moderate to large graphs. Additionally, they are not extensible to arbitrary known edges. SIS methods build the graph sequentially, at each iteration choosing a candidate edge with probability proportional to the vertex degrees. Early methods for this application include (Snijders, 1991; Chen et al., 2005). Most of these samplers get stuck, and the probability of restarting approaches 1 as the degree sequences grow. Bezáková et al. (2012) provide examples where such algorithms are slow. More recent methods avoid the issue of restarting and often come with better theoretical guarantees (Bayati et al., 2010; Blitzstein and Diaconis, 2011; Zhang and Chen, 2013). Our approach to this problem is to construct an MCMC sampler using a symmetric decomposition of Markov kernels; this is a concept defined in Section 2.2.

The second focus of this article is on sampling integer-weighted graphs with prescribed vertex strengths. This can be used to conduct network tomography in the case of a star network topology.

However, the motivating application is approximating the null distribution for evaluating exact tests on two-way contingency tables. This is a classical problem in statistics which is important because standard asymptotics justifying approximate tests (notably Pearson’s χ^2 test of independence) do not hold for tables with cells with low expected frequencies (see Agresti, 2013).

In conditional tests of independence one is interested in sampling tables with given margins. This corresponds to sampling integer-weighted bipartite graphs conditional on vertex strengths. Diaconis and Sturmfels (1998) proposed a simple ‘switch’ Markov chain to sample from such tables. We describe this in more detail in Section 2. It suffers slow mixing in sparse tables.

Diaconis and Sturmfels (1998) also proposed an algebraic algorithm to construct a connected Markov chain in the context of incomplete tables. Other MCMC methods proposed to sample incomplete tables also rely on computing a Markov basis (Aoki and Takemura, 2005; Rapallo, 2006). The computational cost of this is exponential in the size of the table. Additionally, the computation is example specific; i.e. a new basis must be computed for each pattern of structural zeros considered. Chen et al. (2005) introduced the first SIS method for uniform sampling of contingency tables with given marginals. Chen (2007) extended this to incomplete tables. Eisinger and Chen (2017) develop a sampler with improved efficiency, particularly in sparse graphs. We propose an auxiliary variable MCMC sampler which overcomes many of the aforementioned limitations and compare the sampler to SIS approaches in Section 6.

This article begins by setting notation in Section 1.2. Section 2 introduces state-dependent kernel selection, and presents practical strategies for ensuring chains using this technique have the correct invariant distribution. Sections 3 and 4 propose samplers in the unweighted and weighted graph settings respectively. We present a detailed simulation study in Section 5, and apply our samplers to real data in Section 6. Finally, we conclude in Section 7. All proofs can be found in the appendix.

1.2 Notation

An undirected graph $G := (V, E)$ is a pair with V being a labelled vertex set and E a collection of distinct unordered pairs of vertices. If the graph is directed then E consists of distinct ordered vertex pairs. An integer-weighted graph is a triple $G := (V, E, c)$. The function $c: V \times V \rightarrow \mathbb{N}_0$ assigns a

non-zero weight to each $uv \in E$, and 0 to each $uv \notin E$. If the context requires clarification, we use $V(G)$, $E(G)$ and c_G to denote the objects belonging to G .

The in- and out-degrees of a vertex are the number of edges to and from the vertex respectively. The in- and out-strengths of a vertex of a weighted graph are the total weight of edges to and from the vertex respectively. If the graph is undirected, there is no distinction between in and out, so we simply use the terms degree and strength of a vertex. Two undirected graphs with the same vertex set have the same degree (strength) sequence if every vertex has the same degree (strength) in each graph. We use the same terminology for directed graphs, where both the in and out values must be equal for every vertex.

2 STATE-DEPENDENT KERNEL SELECTION

Before discussing state-dependent kernel selection in general terms, we give a concrete example. Let $r := (r_1, \dots, r_I)$ and $c := (c_1, \dots, c_J)$ be non-negative integer vectors, and let \mathcal{X} denote the set of all $I \times J$ non-negative integer matrices such that the row and column marginals equal r and c respectively. Assume \mathcal{X} is non-empty. The task is to construct a Markov chain ergodic with respect to the uniform distribution on \mathcal{X} . Diaconis and Sturmfels (1998) describe a simple Markov chain for this purpose. Given X_n , pick a pair of rows and a pair of columns uniformly at random. The chain proceeds by sampling from the conditional distribution of the delineated subtable given all other entries. An update takes the form

$$\begin{array}{cc} +\Delta & -\Delta \\ -\Delta & +\Delta \end{array}$$

for Δ sampled uniformly from integers which do not induce negative values in the subtable.

A Markov chain on \mathcal{X} is completely characterized by its kernel Q , a regular conditional distribution, where $Q(x, A) := \Pr[X_{n+1} \in A \mid X_n = x]$ for A measurable. In this example Q can be viewed as randomly selecting from a set of other kernels. Indeed, let \mathcal{Z} be the collection of indices of all 2×2 sub-arrays of $I \times J$ tables. The Gibbs update on each $z \in \mathcal{Z}$ defines a kernel K_z on $(\mathcal{X}, \mathcal{B})$. A scan order is a method of choosing a particular kernel from this collection. The aforementioned chain is an example of *random scan*, where kernels are chosen irrespective of the

current state. The chain's kernel is $Q := \sum_z K_z / |\mathcal{Z}|$.

The chain suffers poor mixing in sparse matrices as Δ is often degenerate at 0. As we will see, we can use a state-dependent scan order to improve mixing whilst maintaining ergodicity.

2.1 General Setup

State-dependent kernel selection can be defined in general terms as follows. Let $(\mathcal{Z}, \mathcal{F})$ and $(\mathcal{X}, \mathcal{B})$ be Borel spaces. \mathcal{X} is the state space and \mathcal{Z} is the index set of $K := \{K_z : z \in \mathcal{Z}\}$, a collection of kernels on $(\mathcal{X}, \mathcal{B})$. We assume throughout that the map $(z, x) \mapsto K_z(x, B)$ is jointly measurable for each B . The kernel selection mechanism is defined via a set $w := \{w_x : x \in \mathcal{X}\}$ where each w_x is a probability measure on \mathcal{F} and the map $x \mapsto w_x(F)$ is measurable for each F . A set satisfying these requirements is often referred to as a *probability kernel* from $(\mathcal{X}, \mathcal{B})$ to $(\mathcal{Z}, \mathcal{F})$. If the current state is x , the chain proceeds to sample a kernel K_z according to w_x , and then samples the next state from $K_z(x, \cdot)$. The kernel of this chain is defined through

$$Q(x, \cdot) := \int K_z(x, \cdot) w_x(dz) \text{ for all } x \in \mathcal{X}. \quad (1)$$

If (1) holds then we call (K, w) a decomposition of the kernel Q . The decomposition of a kernel is not necessarily unique. Any kernel Q has an ‘identity’ decomposition, given by $(\{K_1\}, w)$ with $K_1 = Q$ and $w_x(\{1\}) = 1$ for all $x \in \mathcal{X}$. In Sections 2.2 and 2.3 we give techniques for constructing kernels with a desired invariant distribution π using decompositions. These strategies are then used to develop the samplers in Sections 3 and 4 respectively.

2.2 Using a Symmetric Decomposition

A decomposition (K, w) where each K_z is π -reversible does not imply that Q is π -reversible. One notable exception to this is when w is the random scan order, where each w_x is the uniform distribution on \mathcal{Z} . We now define a class of decompositions, which we call symmetric decompositions, for which the resulting Q will be π -reversible. Loosely speaking, it requires that if a state x' is reachable in one step from a state x via a kernel K_z then the likelihood that K_z is selected from state x is the same as in state x' .

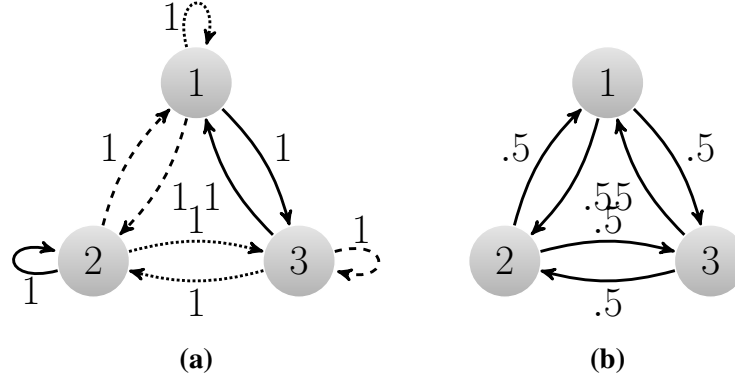


Figure 1: Depiction of the kernel decomposition outlined in Section 2.2. (a) transition probabilities for K_1 (dotted), K_2 (solid) and K_3 (dashed). (b) transitions probabilities for Q .

Definition 2.1 (Symmetric Decomposition). *A decomposition (K, w) is symmetric if there exist a σ -finite measure μ and for every x densities $f_x = dw_x / d\mu$ such that for each z and x , $f_x(z) = f_{x'}(z)$ for $K_z(x, \cdot)$ -almost every x' .*

Any state-independent kernel selection is symmetric: for example, random scan and the ‘identity’ decomposition. As an example of a non-trivial decomposition, consider a three-state state space, as depicted in Figure 1. The left figure defines three kernels on this space. A naive chain might pick from these uniformly, irrespective of the current state. However if the chain is in state i , then K_i cannot change the state. A faster mixing chain Q randomly selects between the other two kernels so that the state changes. This (state-dependent) strategy has a symmetric decomposition. Each of the three kernels shown in Figure 1 (a) is reversible with respect to the uniform distribution and by Lemma 2.2 so is Q .

Lemma 2.2. *Q is π -reversible if it has a symmetric decomposition (K, w) where every $K_z \in K$ is π -reversible.*

The reverse of Lemma 2.2 holds trivially through the identity decomposition. This method is used to support the validity of the sampler developed in Section 3.

2.3 Kernel Selection as an Auxiliary Variable

Here we present an alternative way of constructing a π -invariant chain. The technique described is used in Section 4. Suppose we have a set of statistics $\{T_z : z \in \mathcal{Z}\}$ on \mathcal{X} . Defining a selection

law w , we can construct a chain whose updates keep part of the state fixed. At each iteration this chain proceeds by selecting a feature T_z and changing the current state using some kernel that is π -invariant and keeps T_z fixed. If w does not depend on x this can be viewed as a Gibbs sampler with ‘generalized’ conditioning statistics. Intuitively, however, a state-dependent w may lead to better mixing. Unfortunately in this case the chain will not generally be π -invariant.

We can maintain π -invariance by treating kernel selection as an auxiliary variable. Consider the product space $(\mathcal{Z} \times \mathcal{X}, \mathcal{F} \otimes \mathcal{B})$. The iterated integrals

$$\tilde{\pi}(f) := \int \int f(z, x) w_x(dz) \pi(dx), \quad (2)$$

for all non-negative measurable f define a distribution on $\mathcal{F} \otimes \mathcal{B}$. $\tilde{\pi}$ is the joint distribution for the coordinates $Z(z, x) = z$ and $X(z, x) = x$, while w is the conditional distribution of Z given X and π is the marginal of X .

We now construct a chain on the extended space that is $\tilde{\pi}$ -invariant. This will imply the marginal chain of interest is π -invariant. From the current state (z, x) , the chain first samples $z' \sim w_x$. Then we sample x' using a kernel $K_{z'}$ which keeps both $T_{z'}$ and Z fixed and is $\tilde{\pi}$ -invariant. An obvious choice for each K_z is

$$K_z(x, \cdot) := \Pr[X \in \cdot \mid T_z(X) = T_z(x), Z = z], \quad (3)$$

assuming, of course, that we can sample directly from this distribution. Otherwise if the density of (3) is known up to a normalizing constant we could use Metropolis-Hastings with proposals that keep T_z and Z fixed.

3 SAMPLING UNWEIGHTED GRAPHS

Sampling unweighted graphs conditional on vertex degrees arises in many disciplines. In exponential random graph models, the degrees are often sufficient statistics for nuisance parameters in the null distribution (Snijders, 1991). Other applications include analysis of co-occurrence tables in ecology, and testing the Rasch model in psychometrics (Gustafsson, 1980).

We formalize the sampling problem as follows. Let G_0 be a given directed or undirected graph with a finite vertex set V . Let \mathcal{F} be a subset of possible edges of a graph with vertex set V . Let

\mathcal{G} be the set of all graphs G with the same vertex set and degree sequence as G_0 , and additionally satisfying $E(G) \cap \mathcal{F} = E(G_0) \cap \mathcal{F}$. Our goal is to sample from the uniform distribution π on \mathcal{G} .

Intuitively, \mathcal{F} represents edges known *by design* to be present or absent. Given vertices u and v , if uv belongs to \mathcal{F} then uv is either present in all graphs in \mathcal{G} , or in none. We stress *by design* because the constraints imposed by the degree sequence and \mathcal{F} may imply that further edges are present or absent in all graphs of \mathcal{G} . We call this set $\tilde{\mathcal{F}}$ the set of known edges, and formally define it as

$$\tilde{\mathcal{F}} = \{\text{possible edges } uv : uv \in G_0 \Leftrightarrow (uv \in G \text{ for all } G \in \mathcal{G})\}.$$

We show in Section 3.2 a method of computing $\tilde{\mathcal{F}}$.

Algorithm 1 gives one step of the sampler we propose. It needs two ‘neighborhood’ sets associated to each vertex in a graph G . The set $N_G(u)$ are the in-neighbors of u , excluding any vertex v for which the edge vu is known. $M_G(u)$ is the set of all vertices v which are not out-neighbors of u , and for which the absence of uv is not known. These are defined as

$$N_G(u) := \{v \in V : vu \in E(G), vu \notin \tilde{\mathcal{F}}\},$$

$$M_G(u) := \{v \in V : uv \notin E(G), uv \notin \tilde{\mathcal{F}}\}.$$

Here is a sketch of one run of Algorithm 1. Let $n = 0$, and sample a_0 uniformly from the set of all vertices v for which $N_G(v)$ is non-empty. Sample a_1 uniformly from $N_G(a_0)$, then sample a_2 uniformly from $M_G(a_1)$. Replace a_1a_0 in $E(G)$ with a_1a_2 . Letting $n = n + 2$, iterate this procedure, however in each subsequent step a_{n+1} cannot be a_{n-1} ; this prevents the sampler adding the edge $a_{n+1}a_{n+2}$, and removing it in the next iteration, and should improve state space exploration. Iterate until a_n is a_0 , at which point all degrees have been maintained. The computational cost of Algorithm 1 is proportional to the random length of the sampled vertex sequence a . This does not imply that longer sequences are worse; they tend to reduce the correlation between the current and next state of the chain. Figure 2 shows a simple example step.

3.1 Properties of Algorithm 1

Let a^r denote the reverse of a finite sequence a . Given a graph, let $u_1v_1 \leftrightarrow u_2v_2$ denote the operation of replacing edge u_1v_1 with edge u_2v_2 . We will refer to this operation as a *swap*. We call

Algorithm 1 One Update of the Unweighted Graph Sampler (UGS)

Require: $G, \tilde{\mathcal{F}}$

```

1:  $a_{-1} \leftarrow *$ 
2:  $a_0 \sim U(\{v \in V : N_G(v) \neq \emptyset\})$ 
3:  $n \leftarrow 0$ 
4: repeat
5:    $a_{n+1} \sim U(N_G(a_n) \setminus \{a_{n-1}\})$ 
6:    $a_{n+2} \sim U(M_G(a_{n+1}))$ 
7:    $E(G) \leftarrow E(G) \setminus \{a_{n+1}a_n\}$ 
8:    $E(G) \leftarrow E(G) \cup \{a_{n+1}a_{n+2}\}$ 
9:    $n \leftarrow n + 2$ 
10: until  $a_n = a_0$ 
return  $G$ 

```

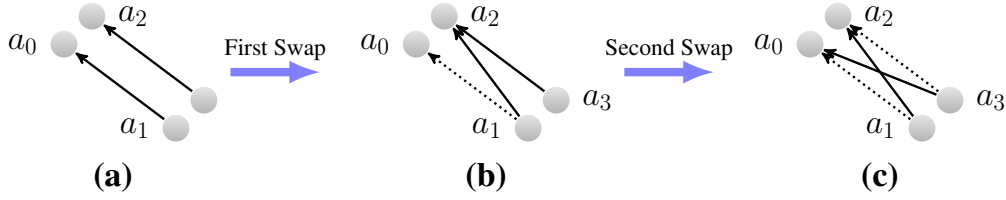


Figure 2: One iteration of Algorithm 1 with two iterations in the inner loop. (a) and (b) graph and quantities prior to the first and second edge swaps respectively. (c) the returned graph. Dashed edges are edges removed through the sampling step.

$u_1v_1 \leftrightarrow u_2v_2$ *viable* if and only if u_1v_1 is an edge, u_2v_2 is not an edge and both u_1v_1 and u_2v_2 are not in $\tilde{\mathcal{F}}$.

A single iteration of Algorithm 1 samples a random sequence of vertices a . Proposition 3.1 implies that the expected length of this is finite, so that a takes the form $a_0a_1\dots a_k a_0$ for some k odd. Let \mathcal{A} be the collection of sequences taking this form.

Proposition 3.1. *For any input graph $G \in \mathcal{G}$ and any \mathcal{F} , the expected length of the vertex sequence a formed by Algorithm 1 is finite.*

Let two sequences be *equivalent* if and only if they are either identical or they are each others' reverse. We let \mathcal{Z} be the quotient set of \mathcal{A} by this equivalence relation.

We will associate each class $z \in \mathcal{Z}$ with a kernel on \mathcal{G} . Fix any z and let a be a representative of z . Consider the following Markov chain on \mathcal{G} . From the current state, attempt to iteratively perform $a_1a_0 \leftrightarrow a_1a_2$, $a_3a_2 \leftrightarrow a_3a_4$, \dots , $a_k a_{k-1} \leftrightarrow a_k a_0$ to obtain the next state. We say this move is *viable* if and only if all of the swaps are viable when applied iteratively. We refer to this sequence of swaps

as the *swaps corresponding to a* . If the swaps are not viable, attempt the swaps corresponding to a^r . If neither swap sequence is viable, then the next state of the chain is unchanged. We define K_z as the kernel of this chain. Remark 3.2 implies that K_z is well-defined; specifically, the definition is independent of the chosen representative of z .

Remark 3.2. *If the sequences a and a^r are distinct and the swaps corresponding to a are viable, then the swaps corresponding to a^r are not viable.*

Let K be the collection of these kernels. The conditional distribution w on \mathcal{Z} is defined implicitly by the law of a given through Algorithm 1. Formally, the sampler selects K_z by sampling either a or a^r belonging to z . Lemma 3.3 implies that Q is π -reversible on \mathcal{G} .

Lemma 3.3. *(K, w) is a symmetric decomposition of Q , and each $K_z \in K$ is π -reversible.*

In practice we consider a lazy version of the chain, which ensures aperiodicity. Fix some small $\alpha \in (0, 1)$ and define $\tilde{Q} := (1 - \alpha)Q + \alpha I$ where I is the identity kernel. Proposition 3.4 follows by Lemma 3.3 and through additionally showing that the chain is connected.

Proposition 3.4. *A Markov chain with kernel \tilde{Q} has limiting distribution π .*

3.2 Identifying all Known Edges/Non-Edges

We show how to determine $\tilde{\mathcal{F}}$ from \mathcal{F} and the degree sequence using an auxiliary graph. Given any $G \in \mathcal{G}$ with n vertices labelled $1, \dots, n$ we construct an auxiliary bipartite digraph $B := (U, V, E)$. Let $U = \{u_i\}$ and $V = \{v_i\}$ for $i = 1, \dots, n$. We define the graph's edge set as

$$E(B) := \{v_j u_i : ij \in E(G) \setminus \mathcal{F}\} \cup \{u_i v_j : ij \notin E(G) \cup \mathcal{F}\}.$$

Figure 3 shows an example of one such graph.

Let $B_{\mathcal{G}}$ denote the collection of all graphs generated this way from the set \mathcal{G} . Proposition 3.5 shows that we can identify $\tilde{\mathcal{F}}$ prior to sampling by identifying all strongly connected components of any graph in $B_{\mathcal{G}}$. This can be done using a depth-first search on B_G , followed by another depth-first search on the transposed graph.

Proposition 3.5. *Fix any graph $B \in B_{\mathcal{G}}$. The vertex pair ij belongs to $\tilde{\mathcal{F}}$ if and only if there is no edge incident to both u_i and v_j , or if u_i and v_j belong to different strongly connected components.*

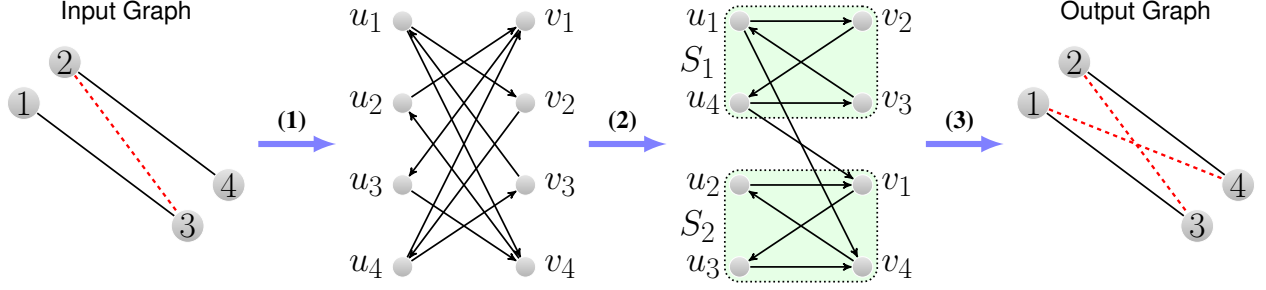


Figure 3: Algorithm for computing $\tilde{\mathcal{F}}$, shown for an undirected graph with four vertices. Input graph G is on the left, where the dashed line represents a prohibited edge (i.e. $\mathcal{F} = \{23\}$). In stage (1) B_G is constructed. In (2) the components S_1 and S_2 are computed, and after (3) we observe $\tilde{\mathcal{F}} = \{23, 14\}$.

The complexity of this preprocessing procedure is $\Theta(2n + n^2)$ (Cormen, Thomas H and Leiserson, Charles E and Rivest, Ronald L and Stein, 2009, chap. 22). It is difficult to formally compare this to the cost of sampling. Empirically the average cost of each iteration of Algorithm 1 appears to grow roughly linearly in n , while the number of iterations needed for sampling grows super-linearly in n . Thus in practice the cost of sampling dominates the edge identification procedure.

4 SAMPLING WEIGHTED GRAPHS

Sampling weighted graphs with given vertex strengths arises in the analysis of two-way contingency tables. In this context sampling is used to approximate the null distribution of test statistics (see Agresti, 1992). The general problem is stated as follows. Let G_0 be a given integer-weighted, directed or undirected graph with a finite vertex set V . Let \mathcal{F} be a subset of possible edges of a graph with vertex set V . Let \mathcal{G} be the set of all weighted graphs with the same vertex set and strength sequence as G_0 , and additionally assigning weight $c_{G_0}(uv)$ to each $uv \in \mathcal{F}$. Our goal is to sample from π , the uniform distribution on \mathcal{G} . We use the auxiliary variable method proposed in Section 2.3. The method first requires defining a set of conditioning statistics $\{T_z : z \in \mathcal{Z}\}$ and a selection law w over them. This is the focus of Sections 4.1 and 4.2. We derive the kernel set K in Section 4.3.

4.1 Conditioning Statistics

We start by specifying the conditioning statistics T_z for a given $z \in \mathcal{Z}$. Define \mathcal{A} and \mathcal{Z} as in Section 3.1, and associate each $a = a_0 a_1 \dots a_k a_0 \in z$ with a vector of vertex pairs

$$e(a) := (a_1 a_0, a_1 a_2, a_3 a_2, a_3 a_4, \dots, a_k a_{k-1}, a_k a_0),$$

of length $k + 1$. We refer to z as *valid* if it satisfies two conditions. Firstly vertex pairs in $e(a)$ must be distinct and secondly they must not be in \mathcal{F} . If these properties hold for $e(a)$ then they hold for $e(a^r)$, and so it suffices that they hold for any $a \in z$. If z is invalid we condition on the whole graph so that no update can occur (i.e. $T_z(G) := G$). Otherwise fix any $a \in z$ and define $T_z(G) := \{c(uv) : uv \notin e(a)\}$. This quantity does not depend on which a is chosen because $uv \in e(a)$ if and only if $uv \in e(a^r)$. This statistic conditions on the weight of all edges outside $e(a)$, so that we only update along $e(a)$.

4.2 Selection Law

Before defining the selection law w , we provide intuition as to which z we wish to sample. From the current state G we intuitively wish to select z that allow us to change the state space. This translates to avoiding z for which the level set $\{T_z = T_z(G)\} = \{G\}$, and therefore implies avoiding all invalid z and some valid z . Assuming z is valid, fix $a \in z$ and consider the vector $e := e(a)$. Letting $s := (+1, -1, \dots, +1, -1)$, any graph in $\{T_z = T_z(G)\}$ must assign weights $c_G(e) + s\Delta$ to e and for some Δ for which the resulting weights are non-negative. We denote the range of Δ by $[\Delta_l, \Delta_u]$. If $\Delta_u = \Delta_l = 0$ then the only graph satisfying this is G . This will happen if there exists i odd and j even such that $c_G(e_i) = c_G(e_j) = 0$. If G is sparse then only a small proportion of z can avoid this. Moreover, which z avoid this depends on the current state and so any state-independent w will be inefficient. Our state-dependent w , which we now define, is designed to limit this.

Occasionally the sampling strategy will fail to sample a kernel index z . Let *id* refer to an arbitrarily chosen invalid z^* , to be chosen by default if this happens. We start by redefining the vertex sets $N_G(u)$ and $M_G(u)$ from Section 3 as

$$N_G(u) := \{v \in V : uv \in E, uv \notin \mathcal{F}\},$$

$$M_G(u) := \{v \in V : uv \notin \mathcal{F}\}.$$

Given the current state G we sample z as follows. Let $n = 0$, $a_{-1} = *$ and sample a_0 uniformly from the set of vertices v for which $N_G(v)$ is non-empty. Repeat the following until termination.

1. If $N_G(a_n) \setminus \{a_{n-1}\}$ is empty return *id*, else sample a_{n+1} uniformly from this set.
2. If $M_G(a_{n+1}) \setminus \{a_n\}$ is empty return *id*, else if a_0 is in this set then return $[a_0 \dots a_{n+1} a_0]$. Otherwise sample a_{n+2} uniformly from this set and let $n = n + 2$.

The chain cannot move if the above procedure returns *id* or an invalid $[a]$. The former is rare and occurs in cases of extreme sparsity. The latter will be more likely with a large set of fixed edges.

4.3 The Kernel Set

We now derive the kernel set. First define $\tilde{\pi}$ as in (2), as an iterated integral of non-negative measurable functions on $\mathcal{Z} \times \mathcal{G}$. Each K_z will take the form of (3). That is, we sample *directly* from the conditional probability of the joint $\tilde{\pi}$ given T_z and the coordinate $Z(z, G) = z$. Therefore if z is invalid then K_z must be the identity kernel. However if z is valid we saw in Section 4.2 that the update can be parameterised by Δ taking values in $[\Delta_l, \Delta_u]$. Therefore, it suffices to derive the distribution of Δ . This is the focus of this section. Throughout we let $G_{\Delta'}$ refer to the graph obtained from the current state at $\Delta = \Delta'$.

Suppose that we sample a vertex sequence a and $z := [a]$ is valid. Since π is the uniform distribution, $\Pr[\Delta = \Delta']$ is proportional to $w_{G_{\Delta'}}(z)$ for each $\Delta' \in [\Delta_l, \Delta_u]$. It is often not possible to sample a from G_{Δ_l} , or to sample a^r from G_{Δ_u} . This is why we collapse a and a^r into z ; doing so ensures we can always sample z from each $G_{\Delta'}$ for $\Delta' \in [\Delta_l, \Delta_u]$. Suppose $\Delta_u - \Delta_l > 1$. Fix any $\Delta_l < \Delta' < \Delta_u$ and define $\alpha := w_{G_{\Delta'}}(z)$. This is interpreted as the probability of sampling z from $G_{\Delta'}$. Inspecting the kernel selection law defined in Section 4.2 we see that $w_{G_1} = w_{G_2}$ for any G_1 and G_2 with the same topology. Since $E(G_{\Delta'})$ is the same for any $\Delta_l < \Delta' < \Delta_u$, it follows that α does not depend on the specific Δ' chosen. If on the other hand $\Delta_u - \Delta_l \leq 1$ we give α an arbitrary finite value. Then

$$\text{Law}(\Delta) = \frac{1}{A} \left[w_{G_{\Delta_l}}(z) \delta_{\Delta_l} + w_{G_{\Delta_u}}(z) \delta_{\Delta_u} + \sum_{\Delta_l < \Delta' < \Delta_u} \alpha \delta_{\Delta'} \right], \quad (4)$$

where $A := w_{G_{\Delta_l}}(z) + w_{G_{\Delta_u}}(z) + \alpha \max(\Delta_u - \Delta_l - 1, 0)$. $w_{G_{\Delta_l}}(z)$, $w_{G_{\Delta_u}}(z)$ and α are easily computed by following the details of Section 4.1.

4.4 Summary

Algorithm 2 gives pseudo-code for one iteration of the sampler. By construction, the chain is π -invariant. Proposition 4.1 holds by additionally showing the chain is connected.

Proposition 4.1. *The chain defined by (K, w) has limiting distribution π .*

Q can be readily adapted to sample distributions whose density is known up to a normalizing constant by using Metropolis-Hastings to sample Δ . The computation cost of Algorithm 2 is proportional to the number of edges updated.

Algorithm 2 One Iteration of the Weighted Graph Sampler (WGS)

Require: G, \mathcal{F}

- 1: $\text{sign}(uv) \leftarrow 0$; $\text{edges} \leftarrow \{\}$; $(\Delta_l, \Delta_u) \leftarrow (-\infty, \infty)$; $a_{-1} \leftarrow *$; $n \leftarrow 0$
- 2: $a_0 \sim U(\{v \in V : N_G(v) \neq \emptyset\})$
- 3: **repeat**
- 4: **if** $N_G(a_n) \setminus \{a_{n-1}\} = \emptyset$ **then return** G
- 5: **else** $a_{n+1} \sim U(N_G(a_n) \setminus \{a_{n-1}\})$
- 6: **if** $a_0 \in M_G(a_{n+1}) \setminus \{a_n\}$ **then** $a_{n+2} \leftarrow a_0$
- 7: **else if** $M_G(a_{n+1}) \setminus \{a_n\} = \emptyset$ **then return** G
- 8: **else** $a_{n+2} \sim U(M_G(a_{n+1}) \setminus \{a_n\})$
- 9: **if** $\text{edges} \cap \{a_{n+1}a_n, a_{n+1}a_{n+2}\} \neq \emptyset$ **then return** G
- 10: $\text{edges} \leftarrow \text{edges} \cup \{a_{n+1}a_n, a_{n+1}a_{n+2}\}$
- 11: $\text{sign}(a_{n+1}a_n) \leftarrow +1$; $\text{sign}(a_{n+1}a_{n+2}) \leftarrow -1$
- 12: $n \leftarrow n + 2$
- 13: **until** $a_n = a_0$
- 14: **for each** $\text{edge} \in \text{edges}$ **do**
- 15: **if** $\text{sign}(\text{edge}) = +1$ **then** $\Delta_l \leftarrow \max(\Delta_l, -c_G(\text{edge}))$
- 16: **else** $\Delta_u \leftarrow \min(\Delta_u, c_G(\text{edge}))$
- 17: Sample Δ according to (4)
- 18: **for each** $\text{edge} \in \text{edges}$ **do**
- 19: $c_G(\text{edge}) \leftarrow c_G(\text{edge}) + \text{sign}(\text{edge})\Delta$

return G

5 Simulation Study

Methods used in this section, and in Section 6, were programmed in C++ and wrapped to R. The algorithms were run on an Intel Core i5-6360U 2GHz CPU. In Section 5.1 we investigate the effect of graph density and size on the performance of the proposed samplers, while Section 5.2 looks at the effect of fixed edges/non-edges. An R-package implementing the new algorithms is available in the supplemental materials.

5.1 Effect of Size and Sparsity

Consider the Erdős-Rényi model $G(n, \theta)$ for directed graphs with self-loops. The parameter n denotes the number of vertices, and each ordered vertex pair is an edge with probability θ , independent of all other edges. If $G \sim G(n, \theta)$ then the conditional distribution of G given its degree sequences is uniform over all graphs with the same degrees. This observation provides us with a useful strategy for assessing convergence of samplers in the unweighted graph setting, which we now detail.

Fixing a particular value of n and θ , we first simulate N independent graphs $G_i \sim G(n, \theta)$. For each G_i , we construct a new graph G_i^0 with the same degrees as G_i using a maximum flow algorithm. The algorithm we use is adapted from that described in Gandy and Veraart (2016), Appendix A. To test the performance of a given sampler we use it to simulate N Markov chains. The i^{th} chain is given initial state G_i^0 and run to obtain samples G_i^1, \dots, G_i^M . If the chain has converged to its target distribution after t iterations, then the distribution of G_i^t and G_i should be statistically indistinguishable. Moreover, if we have access to a statistic T then we can compare, for each t , the empirical distribution of $T(G_1^t), \dots, T(G_N^t)$ to the distribution of $T(G)$. If the sampler mixes rapidly, we expect these to be similar for small t .

The same approach can be used in the weighted graph setting. An analog to the Erdős-Rényi model for weighted and directed graphs with self-loops assigns edge weights to G according to a geometric distribution, i.e. the event $c_G(uv) = k$ occurs with probability $\theta^k(1 - \theta)$ independent of all other edge weights. Conditioning on G 's vertex strengths then yields the uniform distribution over all graphs with the same strengths.

It remains to specify T in the unweighted and weighted settings. In the former we estimate

reciprocity. Letting X denote the adjacency matrix of G , then we use $T = T_u$ with $T_u(G) := \sum_{i < j} X_{ij}X_{ji}$. This statistic is interpreted as the total number of mutual dyads in the graph. In the latter case $T = T_w$ with

$$T_w(G) = \frac{\sum_{u \neq v} \min(c_G(uv), c_G(vu))}{\sum_{u \neq v} c_G(uv)}.$$

This is a measure of reciprocity for weighted graphs, first proposed in Squartini et al. (2013).

Recall that for each t we wish to compare the empirical distribution of $T(G_1^t), \dots, T(G_N^t)$ to that of $T(G)$. $T_u(G)$ is distributed $\text{Bin}(n(n-1)/2, \theta^2)$, and so in the unweighted setting we undertake M Chi-squared tests and record the sequence of p-values p_1, \dots, p_M . This allows us to formally assess convergence of the samplers. The distribution of $T_w(G)$ is not known analytically, and so for the weighted setting we draw 10^5 samples from this distribution and undertake two-sample Kolmogorov-Smirnov tests instead.

We repeat the above procedure for various combinations of n and θ to uncover the effect of graph size and density on the statistical efficiency of the samplers. Algorithm 2 (WGS) is compared to the Diaconis & Sturmfels chain (DS) detailed in Section 3. The DS chain operates on the adjacency matrix of the graphs. Algorithm 1 (UGS) is compared to a simple and widely used randomization procedure that works as follows. At each stage select two edges uv and wx at random from the current graph G . If $ux \notin E(G) \cup \mathcal{F}$ and $wv \notin E(G) \cup \mathcal{F}$ then replace uv and wx with ux and wv , else do not change G . We refer to this randomization procedure as the Switch chain. In each setting, we use $M = 10^4$ and $N = 500$. Thinning used for each chain was chosen to make the computation time per sample comparable.

The results are displayed in Table 1. The efficiency of the Switch chain deteriorates relative to UGS as θ increases. DS becomes inefficient compared to WGS in sparse graphs, as depicted in Figure 4. The proposed methods perform comparatively well across all graph sizes and densities considered.

5.2 Incomplete Tables

UGS and WGS are irreducible in the face of arbitrary fixed edges/non-edges. Here we investigate the ability of the samplers to traverse the state space under such constraints. We do this in the context of incomplete binary and contingency tables, which are tables with some entries fixed at

Table 1: Results of the simulations outlined in Section 5.1. We record $t^* := \min\{t : p_t > 0.1\}$, i.e. the smallest t for which the null that $T(G_1^t), \dots, T(G_N^t)$ is drawn from the distribution of $T(G)$ was not rejected at the 10% level. Mixing rate is the estimated proportion of sampling steps that changed the state of the chain. mean ESS/s reports the mean time-normalized effective sample size across all M chains.

Setup	Unweighted						Weighted					
	t^*		Mixing Rate		mean ESS/s		t^*		Mixing Rate		mean ESS/s	
	UGS	Switch	UGS	Switch	UGS	Switch	WGS	DS	WGS	DS	WGS	DS
$n = 20 \text{ \& } \theta =$												
0.1	5	4	1.00	0.53	7.01×10^4	5.60×10^4	4	189	0.49	0.00	2.60×10^4	3.48×10^2
0.5	6	11	1.00	0.13	2.15×10^4	1.21×10^4	79	427	0.58	0.14	3.50×10^3	7.57×10^2
0.9	3	30	1.00	0.00	3.51×10^4	1.43×10^3	244	460	0.86	0.58	1.11×10^3	3.98×10^2
$n = 50 \text{ \& } \theta =$												
0.1	10	8	1.00	0.72	1.83×10^4	1.62×10^4	21	2232	0.51	0.01	4.19×10^3	3.70×10^1
0.5	29	46	1.00	0.21	6.81×10^3	5.33×10^3	473	2385	0.57	0.16	2.45×10^2	2.00×10^1
0.9	6	193	1.00	0.006	1.50×10^4	5.99×10^2	1531	2603	0.84	0.59	2.64×10^1	9.51
$n = 100 \text{ \& } \theta =$												
0.1	20	15	1.00	0.77	1.45×10^3	1.32×10^3	64	7433	0.51	0.01	7.87×10^2	2.30
0.5	65	111	1.00	0.23	5.97×10^2	5.95×10^2	1600	10,359	0.59	0.05	1.47×10^1	2.64
0.9	13	534	1.00	0.002	1.36×10^3	6.00×10^1	4992	11218	0.84	0.57	8.62	3.12

zero. They arise in several contexts. In the contingency table setting, particular combinations of the two variables may be impossible, forcing zero entries in the corresponding cells. Alternatively there may be missing observations or in some contexts, researchers may wish to fit composite models by partitioning the cells into subsets, and fitting a separate log-linear model for each group (Goodman, 1963, 1968; Fienberg, 1969). See Bishop and Fienberg (1969) for extensive examples of incomplete tables.

We construct 10^3 6×6 incomplete contingency tables using the following procedure. To construct the i^{th} table, we randomly place half of the table coordinates into the fixed set \mathcal{F}_i . We then use a maximum flow algorithm to construct a table x_i^0 in the set \mathcal{X}_i , which consists of all 6×6 tables x with all margins equal to 3, and additionally satisfying $x_{\mathcal{F}_i} = 0$. If \mathcal{X}_i is empty then \mathcal{F}_i is re-sampled until it is not. We use the LattE software (?) to count the number of tables in \mathcal{X}_i . 10^5 samples are obtained using WGS and DS with thinning of 10 and with initial state x_i^0 , and we record the proportion of tables in \mathcal{X}_i visited by each sampler. This is repeated for incomplete binary tables to compare UGS and Switch, however using table margins equal to 1. The results are shown in Figure 5.

Figure 5 provides empirical evidence that WGS and UGS can traverse the state space. DS and Switch appear to be reducible for particular patterns of fixed entries. Diaconis and Sturmfels (1998) propose an alternative chain which uses techniques from computational algebra to compute

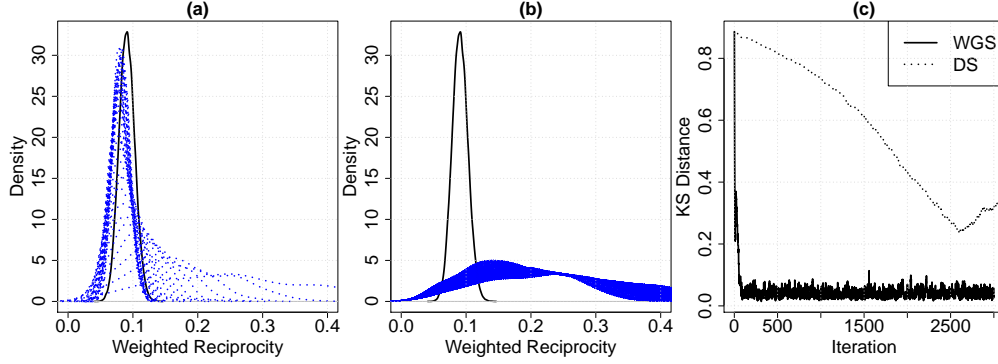


Figure 4: In (a) and (b) the black line is the density of $T_w(G)$ estimated using 10^5 samples when $n = 10^2$ and $p = 0.1$. (a) dotted lines show the evolution of the empirical density of $\{T(G_j^l)\}_{j=1, \dots, N}$ for $l \leq 40$ using WGS. (b) this quantity using DS for $l \leq 10^3$. (c) the Kolmogorov-Smirnov distance between samples $\{T(G_j^l)\}_{j=1, \dots, N}$ and Monte Carlo samples for each l using both WGS and DS.

a Markov basis for the state space. Other approaches using computational algebra include Aoki and Takemura (2005) and Rapallo (2006). Unfortunately the cost of computing the basis is exponential in the size of the table, and these methods are feasible only for tables with only a few rows and columns. Chen (2007) propose SIS algorithms for uniform sampling of incomplete binary and contingency tables. The authors provide an implementation of their method for binary tables, and we use this to test their sampler (labelled SIS_CP1). In each of the 10^3 cases, we collected 10^5 samples and found their method visited all graphs in the state space. However, we show in Section 6.1 that the algorithm is not always reliable.

6 APPLICATIONS

We consider the comparative performance of the new samplers on real datasets. In Section 6.1 we use Algorithm 1 to detect compartmentalization in an ecological network, and in Section 6.2 we use Algorithm 2 to investigate nestedness in a large affiliation network.

Reported standard errors were computed using spectral methods from R's coda package. These estimates were compared to those obtained using batch means, and where feasible, bootstrapping. These latter estimates are not reported as there was little discernible difference from those obtained by spectral methods. Thinning used in each method was set to approximately equate the resulting standard errors. We used burn-in equivalent to 20% of samples obtained.

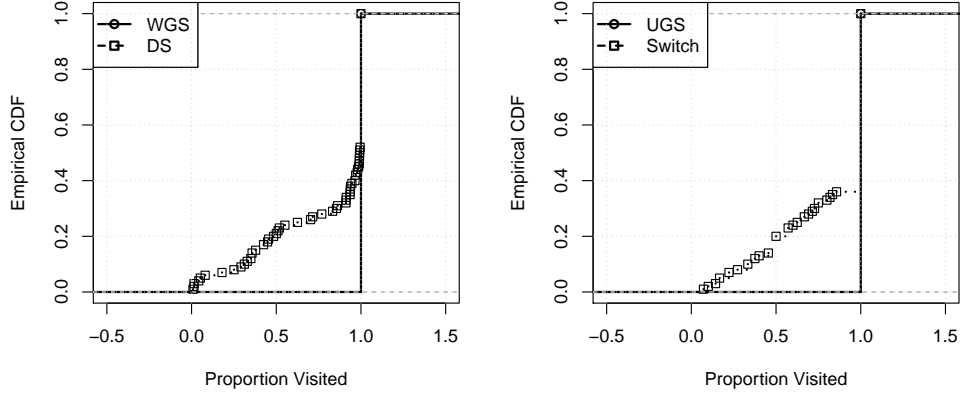


Figure 5: Empirical CDF of proportion of tables visited by each sampler over the 10^3 runs. Left: results for incomplete contingency tables. Right: results for binary tables.

6.1 Ecological Networks

A food web is a digraph encoding predator-prey relationships within a group of species. Each species is a node and a link exists from species A to species B if and only if B consumes A.

Ecologists wish to identify and explain structural patterns in observed food webs including motifs, diet contiguity, intervality, connectance and compartmentalization. We will focus on assessing the tendency towards compartmentalization in food webs. Compartmentalization describes the extent to which species can be partitioned into distinct groups such that linkage density *within* groups is greater than that *between* groups (Girvan and Newman, 2002; Krause et al., 2003).

Figure 6 depicts the food web of 33 species in the Chesapeake bay in the summer. The data was collected by Baird and Ulanowicz (1989). Pimm and Lawton (1980) proposed a statistic \bar{C} to measure the level of compartmentalization in a food web. Here we describe a directed analogue of this statistic. Let G represent a food web of n species, and i and j be two species. Let c_{ij} be the number of shared predators of species i and j as a proportion of the total number of predators of i and j . \bar{C} is then the mean of the off-diagonal elements of (c_{ij}) , defined by

$$\bar{C} := \frac{1}{n(n-1)} \sum_{i=1}^n \sum_{j=1, j \neq i}^n c_{ij}.$$

\bar{C} takes values in $[0, 1]$ and higher values are associated with greater levels of compartmentalization.

web is unusual under the null distribution. Our method of determining trophic structure is crude, and a closer analysis of the food web is warranted before drawing any conclusions.

SIS-CP1 took 24 seconds to run and over 97% of the samples produced were discarded as invalid, leaving only 3,069 to be used for estimation. The estimated p-value was $0.0558 \pm 6.3 \times 10^{-3}$. Using alternative methods to calculate the species' trophic levels gives rise to other sets of forbidden edges. For some such patterns, SIS-CP1 was unable to construct a single valid sample.

6.2 Affiliation Networks

In social network analysis, an affiliation network represents membership or participation data between a set of actors and a set of groups. For example, a link may indicate participation of an actor in an event. Dyadic data of this type include board membership (eg. Mizruchi, 1983), participation in online forums (eg. Allatta, 2003) and authorship of research articles (eg. Newman et al., 2001). Affiliation networks are bipartite graphs, and can therefore be represented as a contingency table, with rows corresponding to actors and with columns denoting the groups.

Social scientists are interested in detecting network structure through particular metrics. Example patterns of interest include 'small-world effects', clustering and degree distributions. Here we focus on detecting nestedness in data collected by Opsahl (2013) on messages sent by users of an online social platform to online forums. This data is a 899 by 522 contingency table whose $(i, j)^{\text{th}}$ entry is the number of messages posted by user i in forum j . Informally speaking, nestedness is the degree to which neighbors of nodes with low degree are a subset of the neighbours of nodes with higher degree. Nestedness has been detected in a number of network systems including ecological interaction networks, social media information networks and socio-economic networks, and has been shown to have important implications for the robustness and stability of a system.

Several measures for nestedness have been developed for integer contingency tables. Galeano et al. (2009) propose to use the weighted-interaction nestedness (WIN) estimator, which is a metric based on a weighted Manhattan distance and takes values between 0 and 1. Higher values indicate higher levels of nestedness. We will test whether the observed WIN statistic is unusually high when compared to a suitable null distribution.

We assume under the null hypothesis that the table is a uniform draw from the set of all tables

with the same margins. This procedure of fixing the margins is widely used in both binary and integer matrices (Connor and Simberloff, 1979; Gotelli and Entsminger, 2001; Ulrich and Gotelli, 2007). Alternative null models are available; for example fixing one margin or only satisfying the observed margins in expectation.

We obtain 10^3 samples using WGS and DS with thinning set to 10^6 . WGS estimated the average WIN distance of the sampled tables at $0.0539 \pm 7.3 \times 10^{-6}$. This estimated standard error is equivalent to that from 10^3 independent samples, indicating good mixing. DS estimated $0.056 \pm 1.7 \times 10^{-3}$ and has a high correlation between successive samples, giving an effective sample size of 12. The estimate exhibits high bias because the chain shows non-stationary behaviour for the first 300 iterations. The observed statistic was 0.157, and so both methods give a p-value of 0. Taken at face value this indicates strong evidence for nestedness. However, this is more likely down to misspecification of the null model. WIN is sensitive to overall matrix density and the sampled tables are systematically denser than the observed table. It would be instructive therefore to consider alternate null distributions which better preserves this property.

Eisinger and Chen (2017) develop efficient SIS methods for sampling tables uniformly over all tables with given margins. The authors provide code for a cell-by-cell method labelled SIS-G* (coded in C). Using SIS-G* on this example we were unable to produce a valid sample. It appears the method is not scalable to large tables.

7 DISCUSSION

This article has developed new MCMC samplers for two important problems. First, for sampling from the set of unweighted graphs respecting prescribed vertex degrees. Second, for sampling from the set of weighted graphs respecting prescribed vertex strengths. The samplers work when conditioning on the presence or absence of a set of edges. We have shown examples where alternative MCMC methods are infeasible as they rely on computing a Markov basis, and where existing SIS methods perform poorly. In contrast, our methods do not require computing a Markov basis, and are orders of magnitude more efficient in these examples.

State-dependent mixing of Markov kernels is a general concept, and the specific implementation of our samplers is not unique. The technique could be used to develop alternative samplers

specialized to particular setting. The samplers can be readily extended to sample from arbitrary distributions known up to a normalization constant. For example, they can be adapted to carry out Bayesian network tomography in the case of a star network topology. In contrast, SIS methods are not readily adaptable to more general distributions. A theoretical analysis of the mixing times of the new samplers is beyond the scope of this article, and is left for future work.

8 ACKNOWLEDGEMENTS

The work of the first author was supported by an EPSRC Research Studentship. The authors thank the editor, associate editor, and two anonymous referees for their comments that have substantially improved the article.

9 SUPPLEMENTARY MATERIALS

R-package: “cgsampr” is a package containing an implementation of Algorithms 1 and 2, and the data used in Sections 6.1 and 6.2. Please refer to the readme file for an introduction to the package. (GNU zipped tar file)

References

- Agresti, A. (1992), “A Survey of Exact Inference for Contingency Tables,” *Statistical Science*, 7, 131–153.
- Agresti, A. (2013), *Categorical Data Analysis*, 3 ed, Hoboken: John Wiley & Sons.
- Allatta, J. T. (2003), “Structural Analysis of Communities of Practice: An Investigation of Job title, Location, and Management Intention,” in *Communities and Technologies*, Dordrecht: Springer Netherlands, pp. 23–42.
- Aoki, S., and Takemura, A. (2005), “Markov Chain Monte Carlo Exact Tests for Incomplete two-way Contingency Tables,” *Journal of Statistical Computation and Simulation*, 75, 787–812.
- Baird, D., and Ulanowicz, R. E. (1989), “The Seasonal Dynamics of the Chesapeake Bay Ecosystem,” *Ecological Monographs*, 59, 329–364.
- Bayati, M., Kim, J. H., and Saberi, A. (2010), “A Sequential Algorithm for Generating Random Graphs,” *Algorithmica*, 58, 860–910.

- Bezáková, I., Sinclair, A., Štefankovič, D., and Vigoda, E. (2012), “Negative Examples for Sequential Importance Sampling of Binary Contingency Tables,” *Algorithmica*, 64, 606–620.
- Bishop, Y. M. M., and Fienberg, S. E. (1969), “Incomplete Two-Dimensional Contingency Tables,” *Biometrics*, 25, 119–128.
- Blitzstein, J., and Diaconis, P. (2011), “A Sequential Importance Sampling Algorithm for Generating Random Graphs with Prescribed Degrees,” *Internet Mathematics*, 6, 489–522.
- Chen, Y. (2007), “Conditional Inference on Tables With Structural Zeros,” *Journal of Computational and Graphical Statistics*, 16, 445–467.
- Chen, Y., Diaconis, P., Holmes, S. P., and Liu, J. S. (2005), “Sequential Monte Carlo Methods for Statistical Analysis of Tables,” *Journal of the American Statistical Association*, 100, 109–120.
- Connor, E. F., and Simberloff, D. (1979), “The Assembly of Species Communities: Chance or Competition?,” *Ecology*, 60, 1132–1140.
- Cormen, Thomas H and Leiserson, Charles E and Rivest, Ronald L and Stein, C. (2009), *Introduction to Algorithms*, 3 ed, Cambridge, Massachusetts: MIT press.
- Diaconis, P., and Sturmfels, B. (1998), “Algebraic Algorithms for Sampling from Conditional Distributions,” *Annals of Statistics*, 26, 363–397.
- Eisinger, R. D., and Chen, Y. (2017), “Sampling for Conditional Inference on Contingency Tables,” *Journal of Computational and Graphical Statistics*, 26, 79–87.
- Fienberg, S. E. (1969), “Preliminary Graphical Analysis and Quasi-Independence for two-way Contingency Tables,” *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 18, 153–168.
- Galeano, J., Pastor, J. M., and Iriando, J. M. (2009), “Weighted-Interaction Nestedness Estimator (WINE): A New Estimator to Calculate over Frequency Matrices,” *Environmental Modelling and Software*, 24, 1342–1346.
- Gandy, A., and Veraart, L. A. M. (2016), “A Bayesian Methodology for Systemic Risk Assessment in Financial Networks,” *Management Science*, 63, 4428–4446.
- Girvan, M., and Newman, M. E. J. (2002), “Community Structure in Social and Biological Networks,” *Proceedings of the National Academy of Sciences*, 99, 7821–7826.
- Goodman, L. A. (1963), “Statistical Methods for the Preliminary Analysis of Transaction Flows,” *Econometrica*, 31, 197–208.
- Goodman, L. A. (1968), “The Analysis of Cross-Classified Data: Independence, Quasi-Independence, and Interactions in Contingency Tables with or without Missing Entries,” *Journal of the American Statistical Association*, 63, 1091–1131.

- Gotelli, N., and Entsminger, G. (2001), “Swap and Fill Algorithms in Null Model Analysis: Rethinking the Knight’s Tour,” *Oecologia*, 129, 281–291.
- Guimerà, R., Stouffer, D. B., Sales-Pardo, M., Leicht, E. A., Newman, M. E. J., and Amaral, L. A. N. (2010), “Origin of Compartmentalization in Food Webs,” *Ecology*, 91, 2941–2951.
- Gustafsson, J.-E. (1980), “A Solution of the Conditional Estimation Problem for Long Tests in the Rasch model for Dichotomous Items,” *Educational and Psychological Measurement*, 40, 377–385.
- Holland, P. W., and Leinhardt, S. (1981), “An Exponential Family of Probability Distributions for Directed Graphs,” *Journal of the American Statistical Association*, 76, 33–50.
- Krause, A. E., Frank, K. A., Mason, D. M., Ulanowicz, R. E., and Taylor, W. W. (2003), “Compartments Revealed in Food-Web Structure,” *Nature*, 426, 282–285.
- McDonald, J. W., Smith, P. W. F., and Forster, J. J. (2007), “Markov chain Monte Carlo Exact Inference for Social Networks,” *Social Networks*, 29, 127–136.
- Milo, R., Shen-Orr, S., Itzkovitz, S., Kashtan, N., Chklovskii, D., and Alon, U. (2002), “Network Motifs: Simple Building Blocks of Complex Networks,” *Science*, 298, 824–827.
- Mizruchi, M. S. (1983), “Who Controls Whom? An Examination of the Relation between Management and Boards of Directors in Large American Corporations,” *The Academy of Management Review*, 8, 426–435.
- Newman, M. E. J., Strogatz, S. H., and Watts, D. J. (2001), “Random Graphs with Arbitrary Degree Distributions and their Applications,” *Physical Review E*, 64, 1–17.
- Opsahl, T. (2013), “Triadic Closure in Two-Mode Networks: Redefining the Global and Local Clustering Coefficients,” *Social Networks*, 35, 159–167.
- Pimm, S. L., and Lawton, J. H. (1980), “Are Food Webs Divided into Compartments?,” *The Journal of Animal Ecology*, 49, 879–898.
- Rao, A. R., Jana, R., and Bandyopadhyay, S. (1996), “A Markov Chain Monte Carlo Method for Generating Random (0, 1) Matrices with Given Marginals,” *Sankhya: The Indian Journal of Statistics, Series A (1961-2002)*, 58, 225–242.
- Rapallo, F. (2006), “Markov Bases and Structural Zeros,” *Journal of Symbolic Computation*, 41, 164–172.
- Rasch, G. (1960), “Probabilistic Models for some Intelligence and Achievement Tests,” *Copenhagen: Danish Institute for Educational Research*, .
- Rezende, E. L., Albert, E. M., Fortuna, M. A., and Bascompte, J. (2009), “Compartments in a Marine Food Web Associated with Phylogeny, Body Mass, and Habitat Structure,” *Ecology Letters*, 12, 779–788.

- Roberts, J. M. (2000), “Simple Methods for Simulating Sociomatrices with Given Marginal Totals,” *Social Networks*, 22, 273–283.
- Ryser, H. J. (1963), *Combinatorial Mathematics* Mathematical Association of America.
- Snijders, T. A. B. (1991), “Enumeration and Simulation Methods for 0-1 Matrices with Given Marginals,” *Psychometrika*, 56, 397–417.
- Squartini, T., Picciolo, F., Ruzzenenti, F., and Garlaschelli, D. (2013), “Reciprocity of Weighted Networks,” *Scientific Reports*, 3.
- Ulrich, W., and Gotelli, N. J. (2007), “Null Model Analysis of Species Nestedness Patterns,” *Ecology*, 88, 1824–1831.
- Verhelst, N. D. (2008), “An Efficient MCMC Algorithm to Sample Binary Matrices with Fixed Marginals,” *Psychometrika*, 73, 705–728.
- Wasserman, S., and Faust, K. (1994), *Social Network Analysis: Methods and Applications*, Structural Analysis in the Social Sciences Cambridge University Press.
- Williams, R. J., and Martinez, N. D. (2004), “Limits to Trophic Levels and Omnivory in Complex Food Webs: Theory and Data,” *The American Naturalist*, 163, 458–468.
- Zhang, J., and Chen, Y. (2013), “Sampling for Conditional Inference on Network Data,” *Journal of the American Statistical Association*, 108, 1295–1307.

APPENDIX: PROOFS

Proof of Lemma 2.2. Without loss of generality (and for notational simplicity) assume π and w are dominated by one-dimensional Lebesgue measure. We show

$$\int_A \pi(x)Q(x, B) dx = \int_B \pi(x)Q(x, A) dx \quad \text{for all } A, B \in \mathcal{B}.$$

Fix any A and B in \mathcal{B} , and define the densities $\{f_x\}$ as in Definition 2.1. Then

$$\begin{aligned} \int_A \pi(x)Q(x, B) dx &= \int_A \int_{\mathcal{Z}} \int_B \pi(x)K_z(x, y)f_x(z) dy dz dx \\ &= \int_A \int_{\mathcal{Z}} \int_B \pi(y)K_z(y, x)f_y(z) dy dz dx = \int_B \pi(x)Q(x, A) dx \end{aligned}$$

as required. In the first step we have expressed the integral using densities. In the second, we use π -reversibility of each K_z , and the fact that for each $z \in \mathcal{Z}$ and $x \in \mathcal{X}$, $f_x(z) = f_y(z)$ for $K_z(x, \cdot)$ -a.e. $y \in B$. A simple change of variables then yields the result. \square

Proof of Proposition 3.1. Fix G in \mathcal{G} . Define the second-order Markov chain $(Y_n)_{n \geq 0}$, where $Y_n := (a_{n-1}, a_n, G_n)$ and G_n is defined as follows. Let $G_0 = G$, otherwise if n is odd, let G_n be the graph obtained after $a_n a_{n-1}$ is removed from $E(G_{n-1})$. If n is even, G_n is the graph obtained after $a_{n-1} a_n$ is added to $E(G_{n-1})$. Define \mathcal{Y} as the set of points reachable from $(*, a_0, G)$ for some a_0 in $\{v : N_G(v) \neq \emptyset\}$. Let $D := (\mathcal{Y}, E)$ be the digraph underlying this chain and define A as the subset of points (v, u, G') in \mathcal{Y} for which $G' \in \mathcal{G}$. Let $T := \inf\{n \geq 1 : Y_n \in A\}$ be the first passage time of A . Proposition 3.1 is equivalent to showing $\mathbb{E}(T) < \infty$. The following holds true, and will be shown at the end of this proof.

$$\text{From any } (v, u, G') \in \mathcal{Y}, \text{ there exists a simple path to } A. \quad (5)$$

We can bound the probability of traversing each edge in D from below by some constant $p > 0$. Let N denote the size of \mathcal{Y} . Suppose the chain is at some state $y \notin A$. By (5), this implies the probability of hitting A within the next N steps is bounded from below by p^N . Hence,

$$\mathbb{E}(T) = \sum_{n=1}^{\infty} n \Pr[T = n] \leq N \sum_{k=1}^{\infty} k \Pr[k-1 < T/N \leq k] \leq N \sum_{k=1}^{\infty} k p^N (1 - p^N)^{k-1} = N p^{-N}.$$

It remains to show (5). Note that for any $(v_1, u_1, G_1)(v_2, u_2, G_2) \in E(D)$:

$$(v^*, u_2, G_2)(u_2, u_1, G_1) \in E(D) \text{ if and only if } (v^*, u_2, G_2) \in \mathcal{Y} \text{ and } v^* \neq v_2. \quad (6)$$

By definition, there exists a point $y_0 := (*, u_0, G)$ and a walk $y_0 \dots y_k$ in D such that $y_k = (v, u, G')$. Given that $V(D)$ is finite, continuing an arbitrary walk along D from y_k implies we must eventually either return to A , or visit a graph already seen along the walk. Denote the vertex visited at the l th step of this walk by $y_l = (u_{l-1}, u_l, G_l)$. If we revisit A we are done, otherwise define

$$n := \inf\{l > k : G_l = G_m \text{ for some } m < l\}.$$

The condition $G_n = G_m$ implies that $u_n = u_m$. Additionally $u_{n-1} \neq u_{m-1}$, otherwise this would imply $G_{n-1} = G_{m-1}$, which contradicts the definition of n . By (6), $(u_{m-2}, u_{m-1}, G_{m-1})(u_{m-1}, u_m, G_m) \in E(D)$ implies that $(u_{n-1}, u_n, G_n)(u_m, a_{m-1}, G_{m-1}) \in E(D)$. Thus we can traverse to (u_m, u_{m-1}, G_{m-1}) . Iteratively applying (6) (which we can do as $u_{l+1} \neq u_{l-1}$ for all $l \geq 0$) implies we can reach a state with graph $G = G_0$, which must be in A , completing the proof of (5). \square

Proof of Lemma 3.3. We first show that (K, w) is a symmetric decomposition. Fix any $G \in \mathcal{G}$ and any $z \in \mathcal{Z}$, and let a be a representative of z . Let p refer to the statement ‘ $w_G(z) = w_{G^*}(z)$ for all G^* for which $K_z(G, G^*) > 0$ ’. It suffices to show that p is true.

Consider a Markov chain with kernel K_z and current state G . Suppose the chain remains unchanged after one iteration of Algorithm 1. Then p is true trivially. Without loss of generality, suppose the swaps corresponding to a are viable, and the chain moves to some $G^* \in \mathcal{G}$. Remark 3.2 implies swaps corresponding to a^r are not viable. Since the swaps corresponding to a sampled vertex sequence must be viable, $w_G(z)$ is the probability of sampling a given the chain is at G .

At G^* , the swaps corresponding to a^r are viable. By an analogous argument, it follows that $w_{G^*}(z)$ is the probability of sampling a^r given the chain is at G^* . One can deduce from Algorithm 1 that the probability of sampling w given the chain is at G is equal to the probability of sampling a^r given the chain is at G^* . This holds because the degree sequence is the same for either state.

We now show that each $K_z \in K$ is reversible with respect to the uniform distribution. This is implied by detailed balance. Specifically, for each $K_z \in K$ we show

$$K_z(G, G^*) = K_z(G^*, G) \quad \text{for all } G, G^* \in \mathcal{G}.$$

Fix any G and G^* . $K_z(G, G^*) = 1$ if and only if $K_z(G^*, G) = 1$, because applying two iterations of a Markov chain with kernel K_z from some current state G' , returns G' . The result follows by additionally observing that $K_z(G, G^*)$ can only be zero or one. \square

Proof of Proposition 3.4. Lemma 3.3 and connectedness of the chain suffice. Fix any $G, G' \in \mathcal{G}$, and suppose the current state of the chain is G . Form a digraph H as follows. For each vertex pair uv , if $uv \in E(G)$ and $uv \notin E(G')$, add a *red* edge uv to $E(H)$. If $uv \notin E(G)$ and $uv \in E(G')$, add a *blue* edge uv to $E(H)$. Define an *alternating cycle* as a cycle whose edges are alternately red and blue. G and G' are equivalent if and only if H has no edges.

Then H is the union of a finite sequence of edge-disjoint alternating cycles. Fix any such cycle $v_0v_1\dots v_kv_0$, ordered so that the v_0v_1 is red. The chain can sample $v_0v_1\dots v_kv_0$ with positive probability, yielding a new graph G'' , whilst removing all edges in H corresponding to this cycle. Iterate until H has no more edges. Hence the chain is connected. \square

Proof of Proposition 3.5. For a given \mathcal{F} , the map from \mathcal{G} to $B_{\mathcal{G}}$ is injective, so the sampler can be thought of as a Markov chain ergodic with respect to the uniform distribution on $B_{\mathcal{G}}$.

We briefly describe how to view the Markov chain as operating on $B_{\mathcal{G}}$. An initial vertex v_j is sampled from V . The chain now samples u_i from the out-neighborhood of v_j and replaces the edge $v_j u_i$ with $u_i v_j$. If G is undirected, additionally switch $v_i u_j$ with $u_j v_i$. Continue walking along the vertices of the graph in this manner until the sampler returns to the initial vertex for the first time.

Without loss of generality, suppose G is directed. Partition B 's vertex set into strongly connected components S_1, \dots, S_K . Fix $u_i \in S_k$ and $v_j \in S_l$. If no edge is incident to u_i and v_j then $ij \in \mathcal{F} \subseteq \tilde{\mathcal{F}}$. Otherwise if $k \neq l$, edges between S_k and S_l are uniformly in one direction; say from S_k to S_l . Suppose the chain on B traverses $u_i v_j$, replacing it with $v_j u_i$. Returning to the initial vertex requires traversal of $v_j u_i$. Hence, $u_i v_j$ can be flipped only an even number of times, and the direction is unchanged. By Lemma 3.4, $ij \in \tilde{\mathcal{F}}$. If $k = l$, $u_i v_j$ can be switched odd number of times, so $ij \notin \tilde{\mathcal{F}}$. The undirected case holds analogously. \square

Proof of Proposition 4.1. It suffices to show connectedness. Define

$$d(G, G') := \sum_{u \in V} \sum_{v \in V} |c_G(uv) - c_{G'}(uv)| \quad \text{for all } G, G' \in \mathcal{G}.$$

Then (\mathcal{G}, d) is a metric space. Fix any two distinct graphs $G, G' \in \mathcal{G}$, and suppose the current state of the chain is G . It suffices to show that one can construct a sampling step yielding a new graph *strictly* closer to G' in this metric space.

Let $n_{uv} := c_G(uv) - c_{G'}(uv)$ for each vertex pair uv . We form a multi-graph H as follows. If $n_{uv} > 0$, add n_{uv} *red* copies of the *direction reversed* edge vu to $E(H)$, while if $n_{uv} < 0$, add $-n_{uv}$ *blue* copies of uv to $E(H)$. The graphs G and G' are equivalent if and only if H has no edges. Define an *alternating cycle* in H as a cycle whose edges are alternately *red* and *blue*.

H can be expressed as the union of a finite number of edge-disjoint alternating cycles. Fix any such alternating cycle $v_0 v_1 \dots v_l v_0$ in H . Order the cycle so that $v_0 v_1$ is red. Letting \mathbb{O}_n denote the set of odd natural numbers less than or equal to n , we define

$$k := \inf\{n \in \mathbb{O}_{l-2} : v_n v_{n+1} \notin \mathcal{F}\}$$

where we let $\inf \emptyset := l$.

Under Algorithm 2, there is a positive probability of sampling the vertex sequence $v_0 v_1 \dots v_k v_0$ given the chain is at G . Sampling $\Delta = -1$ along this vertex sequence returns a new graph G'' , removing at least three edges from H whilst adding at most one. Hence $d(G'', G') \leq d(G, G') - 2$. □