Ann Inst Stat Math (2017) 69:673–704
DOI 10.1007/s10463-016-0560-2

Goodness of fit for log-linear network models: dynamic Markov bases using hypergraphs

Elizabeth Gross[1] · **Sonja Petrović**[2] · **Despina Stasi**[2]

Received: 27 January 2015 / Revised: 4 February 2016 / Published online: 5 April 2016

Abstract Social networks and other sparse data sets pose significant challenges for statistical inference, since many standard statistical methods for testing model/data fit are not applicable in such settings. Algebraic statistics offers a theoretically justified approach to goodness-of-fit testing that relies on the theory of Markov bases. Most current practices require the computation of the entire basis, which is infeasible in many practical settings. We present a dynamic approach to explore the fiber of a model, which bypasses this issue, and is based on the combinatorics of hypergraphs arising from the toric algebra structure of log-linear models. We demonstrate the

E. Gross is supported by the NSF Postdoctoral Research Fellowship, NSF award #DMS-1304167. S. Petrović and D. Stasi acknowledge partial support from Grants #FA9550-12-1-0392 and #FA9550-14-1-0141 from the U.S. Air Force Office of Scientific Research (AFOSR) and the Defense Advanced Research Projects Agency (DARPA). Some of the computations were performed on a cluster provided by an NSF-SCREMS Grant to IIT. Part of this work was completed while D. Stasi was a postdoc at Pennsylvania State University Statistics Department.

Electronic supplementary material The online version of this article (doi:10.1007/s10463-016-0560-2) contains supplementary material, which is available to authorized users.

✉ Sonja Petrović
Sonja.Petrovic@iit.edu

Elizabeth Gross
elizabeth.gross@sjsu.edu

Despina Stasi
despina.stasi@gmail.com

[1] Department of Mathematics and Statistics, San José State University, One Washington Square, San Jose, CA 95192, USA

[2] Department of Applied Mathematics, Illinois Institute of Technology, Rettaliata Engineering Center room 208, 10 West 32nd Street, Chicago, IL 60616, USA

approach on the Holland–Leinhardt p_1 model for random directed graphs that allows for reciprocation effects.

Keywords Algebraic statistics · Markov basis · Hypergraph · Toric ideal · Contingency table · Network model · Random graph · Sampling algorithm

1 Introduction

Network data often arise as a single sparse observation of relationships among units, for example, individuals in a network of friendships, or species in a food web. Such a network can be naturally represented as a contingency table whose entries indicate the presence and type of a relationship, and whose dimension depends on the complexity of the model. This representation makes networks amenable to analysis by standard categorical data analysis tools and, in particular, it brings to bear the log-linear models literature, e.g., Bishop et al. (1975). However, given that often only a small sample or even just a single observation of the network is all we have access to, or that the data are sparse, several problems remain. In particular, in the case of network models, since quantitative methods are essentially nonexistent, goodness-of-fit testing is usually carried out qualitatively using model diagnostics. Namely, the clustering coefficient, triangle count, or another network characteristic is used for a heuristic comparison between observed and simulated data. In Hunter et al. (2008), the authors offer a systematic approach for comparing structural statistics between an observed network and networks simulated from the fitted model and point out some of the difficulties of fitting the ERGMs. More recently, Goldenberg et al. (2009) review various network models and discuss modeling and fitting challenges that remain.

Even for linear exponential families, the problem of determining goodness of fit is a difficult one for network data. When standard asymptotic methods, such as χ^2 approximations, are deemed unreliable (see Haberman 1981), or when the observed data are sparse, one may want to use exact conditional tests. In such tests, the observed network (or table) u with sufficient statistic vector $S(u)$ is compared to the reference set, called the *fiber* $\mathcal{F}_S$, defined to be the space of all realizations of the network under the given set of constraints S. Unfortunately, the size and combinatorial complexity of the fiber are the main obstacle for complete fiber enumeration, so that even in small problems (see Slavković et al. 2015, Sect. 4), determining the exact distribution is often unfeasible. Moreover, fiber enumeration and sampling is crucial not only for goodness-of-fit testing but also for data privacy considerations (see Slavković 2010).

The theory of Markov bases provides a possible solution to the problem of sampling the fibers for any log-linear model. Namely, a Markov basis is a set of "moves" that, starting from any point in a fiber, allows one to perform a random walk on the fiber and visit every point with positive probability. Therefore, the standard Metropolis–Hastings algorithm provides a way to carry out exact tests and as argued by Diaconis and Sturmfels (1998), this procedure yields *bona fide* tests for goodness of fit. Furthermore, every log-linear model comes equipped with a non-unique but finite Markov

basis. The existence and finiteness of the basis is a consequence of the main result of Diaconis and Sturmfels (1998), what is now often called the Fundamental Theorem of Markov Bases in the algebraic statistics literature. However, two main computational challenges remain open to make this theory useful for network and large table data in practice. We describe these challenges broadly next and, then, address them in the remainder of this manuscript.

The first computational challenge is in determining the Markov basis itself. The fact that a Markov basis for a model guarantees to connect *every* one of its fibers makes it a highly desirable object to obtain. Unfortunately, the fastest algorithms for computing the moves for an arbitrary model (these algorithms exploit the toric structure of the model) are not fast enough. Even for some basic log-linear network models, it can take hours to find all Markov moves for networks with less than ten nodes. This motivates a structural study of Markov bases for a given fixed family of models. To this end, the literature provides many examples, including Aoki and Takemura (2003, 2005), Develin and Sullivant (2003), Dobra (2003), Dobra and Sullivant (2004), Hara et al. (2010), Hara et al. (2009b, a), Haws et al. (2014), Král et al. (2010), Norén (2015), Rapallo and Yoshida (2010), Sturmfels and Welker (2012) and Yamaguchi et al. (2013). Researchers in applied fields may also be interested in a non-Markov-basis method by Hara et al. (2012) which relies on generating moves that are random combinations of lattice basis elements. As lattice bases are easy to calculate even for complicated models, this method is promising. However, it does not take into account observed data and will thus result in a large number of the moves generated being non-applicable. Since our example of interest is a network model with inherent sampling constraints, we should also note that such constraints can compound the issue of computing a set of moves guaranteed to connect each fiber. Sampling constraints restrict the fiber, and in fact, if one is interested in sampling a restricted fiber, Ogawa et al. (2013) and Aoki et al. (2012) show that one needs a larger set of moves, for example a *Graver basis*, to guarantee connectivity. A Graver basis (see Drton et al. (2009), Sect. 1.3; Aoki et al. (2012), Sect. 4.6 for definition and discussion) is a particular Markov basis and generally contains more moves than a minimal Markov basis (where minimal is defined with respect to set inclusion).

The second computational challenge comes from the fact that knowing an entire Markov basis for a model may still not be sufficient to run goodness-of-fit tests efficiently. Namely, Markov bases are data-independent (see Dobra et al. 2008, Problem 5.5.). To paraphrase Aoki et al. (2012): since a Markov basis is common for every fiber $\mathcal{F}_S$ (that is, for all values that the vector S of sufficient statistics can take), the set of moves connecting the particular fiber of the observed data $u \in \mathcal{F}_{S(u)}$ will usually be significantly smaller than the entire basis for the model. To handle this issue, Dobra (2012) suggests generating only moves needed to complete one step of the random walk, that is, only *applicable* moves. Dobra refers to the set of moves generated in this way as a *dynamic Markov basis*, since the full basis is not generated ahead of time. An example of this strategy is found in Ogawa et al. (2013), where the authors present an algorithm for generating a random element of the Graver basis for the beta model. The beta model is a basic generalization of the Erdös – Rényi random graph model: an ERGM for simple undirected random graphs where the degrees of the nodes form the sufficient statistics. In fact,

this work can be cast within a more general framework of sampling from the space of contingency tables with fixed properties. A commonly fixed set of table properties are marginals of the table: they represent sufficient statistics of many—but not all—log-linear models. The paper Dobra (2012) focuses on log-linear models whose sufficient statistics are fixed marginals. There, the Markov moves are obtained through a sequential adjustment of cell bounds, a method that appears in sequential importance sampling (SIS); see, for example, Chen et al. (2005) and Dinwoodie and Chen (2011). In contrast, we build a dynamic Markov basis by exploiting the combinatorics of the model. This allows us to complement Dobra's methodology to log-linear models whose sufficient statistics are not necessarily table marginals and extend that of Ogawa, Hara and Takemura to include also directed graph models.

In this manuscript, we explore the problem of performing goodness-of-fit tests for log-linear models when sufficient statistics are not necessarily table marginals, and in the presence of sampling constraints. In this case, there is *no* general methodology for obtaining the part of the Markov bases which is relevant for the observed data. In this work, we address the issues raised above from the point of view of algebraic statistics and combinatorial commutative algebra. We propose the use of *parameter hypergraphs* to generate *Graver* moves that are data-dependent and, therefore, applicable to the observed network (or table). Using Graver bases ensures connectivity of restricted fibers, while respecting sampling constraints. Furthermore, as Petrović and Stasi (2014) frame the Graver basis determination problem in terms of combinatorics of hypergraphs, we add this combinatorial ingredient to the recipe which allows us to generate the moves in a dynamic fashion, based on the observed table or network. The sufficient statistics for the model need not be table marginals; the only assumption we impose, mostly for simplicity, is that the model parametrization is squarefree in the parameters (see Sect. 2 for details). The random walk associated with the moves we produce in this way is irreducible, symmetric and aperiodic, and so we may use the Metropolis-Hastings algorithm (see Robert and Casella 1999, Sect. 7) to implement a Markov chain whose stationary distribution is equal to the conditional distribution on the fiber. This allows us to sample from the fiber of an observed network or table as desired.

We illustrate our methodology and apply dynamically generated Markov bases to the p_1 model from Holland and Leinhardt (1981) (see also discussion in Fienberg and Wasserman 1981); specifically because previous methods are not applicable to this model directly. In particular, the sufficient statistics of the p_1 model are not table marginals; instead they are of two types: one is a sub-table marginal, and the other is a subtable sum. This can be seen easily when representing the model in contingency table form following Fienberg and Wasserman (1981). Holland and Leinhardt proposed to model a random directed graph by parametrizing propensity of nodes to send and receive links as well as reciprocate edges, where dyads are independent of each other. Petrović et al. (2010) and Fienberg et al. (2010) study the algebra and geometry of these models and derive structural results for their Markov bases. Remarkably, the moves can be obtained by a direct computation only for networks with less than 7 nodes, using 4ti2 (2008), currently the fastest software capable of producing such bases. Thus testing model fit for larger networks is not feasible using the traditional Metropolis-Hastings

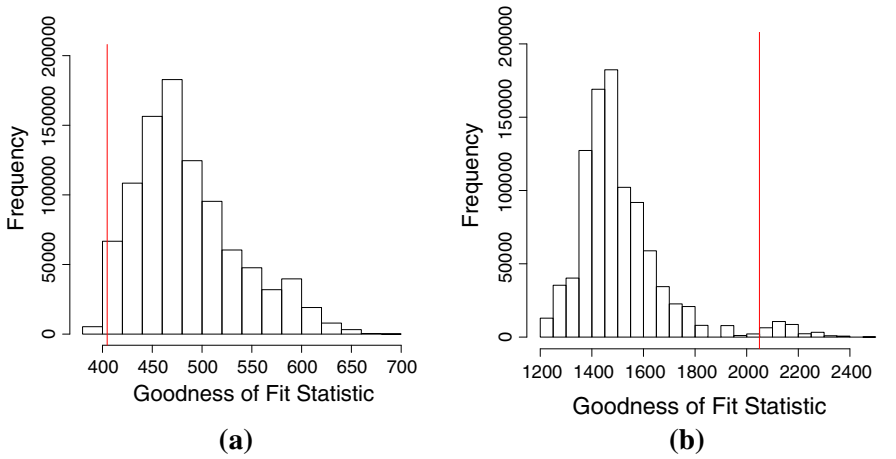

Fig. 1 Sampling distribution of the chi-square statistic: histograms from simulation running Algorithm 2 for the p_1 model with edge-dependent reciprocation. **a** Affinity network derived from Sampson's monastery data set in Sampson (1968). Observed chi-square value: 404.7151. $p = 0.986$. **b** Chesapeake food web data set derived from Baird and Ulanowicz (1989). Observed chi-square value: 2049.403. $p = 0.03459$

algorithm. Using a straightforward implementation of Algorithm 2 in R (2005), which we have made available in supplementary material (Gross et al. 2014), we test several familiar network data sets. Figure 1a shows the histogram of the values of the chi-square statistics for 1,000,000 steps in the chain, including 50,000 burn-in steps), obtained from the Sampson (1968) monastery study. The horizontal axis represents chi-square statistic values. The observed value of the chi-square statistic for the monk dataset is represented by the vertical red line, giving a visual representation of the large p-value of 0.986 and thus a pretty good model fit. A similar histogram in Fig. 1b shows that the p_1 model does not fit the Chesapeake Bay food web data so well: the estimated p-value is 0.03459 after 1,000,000 moves.

This paper is organized as follows: Section 2 develops the combinatorial approach to the construction of Markov bases dynamically and provides the necessary mathematical background. Section 3 illustrates the developed methodology for the Holland and Leinhardt's p_1 model. Examples and simulations are given in Sect. 4. Specifically, further discussion and analyses of the model fit for the directed networks arising from the monk and food web data can be found in Sects. 4.4 and 4.5. Sections 4.3 and 4.2 provide studies of mobile money networks of a Kenyan family and of four networks simulated from the p_1 distribution, respectively. Finally, simulations on a small synthetic network in Sect. 4.1 indicate good mixing times and quick convergence of the p-value estimate (e.g., see Fig. 7b). As this is best illustrated when the entire fiber has been determined exactly, we also consider a small 591-network fiber for an undirected graph on 8 nodes from Sect. 5.1 in Ogawa et al. (2013). Our walk explores the entire fiber in as little as 15,000 moves and the total variation distance from the uniform distribution is below 0.25 after 10,000 moves. This improves the fiber discovery rate and could be due to the fact that the steps in the simulated walks are longer than minimal Markov moves would suggest, since we are generating a superset of the squarefree applicable part of the Graver basis in our algorithm.

2 Parameter hypergraphs of log-linear models: dynamic moves for Metropolis-Hastings

Here we introduce the mathematical construction that allows us to dynamically generate applicable moves for sampling fibers of general log-linear models. Specifically, Theorem 1 and Corollary 1 show that random walks on fibers of log-linear models are equivalent to sampling sub-hypergraphs of the parameter hypergraph with a fixed degree sequence, revealing the combinatorial nature of the applicable move construction problem. We end this section by demonstrating our theory and our method for sampling fibers in a small example in Example 3. Our main application in this paper, however, is the subject of Sects. 3 and 4, where we explain how to sample elements from p_1 fibers.

2.1 Markov bases: fundamentals

Consider a log-linear model on an $m_1 \times \cdots \times m_u$ contingency table U with sufficient statistic S. Let $u \in \mathbb{Z}_{\geq 0}^{m_1 \times \cdots \times m_u}$ be a realization of the table U, with observed sufficient statistic $S(u)$. The fiber of u, which we will denote $\mathcal{F}_{S(u)} \subset \mathbb{Z}_{\geq 0}^{m_1 \times \cdots \times m_u}$ (or simply $\mathcal{F}_S$ if the observed table u is implied from the context), is the space of all realizations v of the table whose sufficient statistic is the same as that of u, i.e., $S(v) = S(u)$. For two tables in the same fiber $u, v \in \mathcal{F}_S$, the entry-wise difference $u - v$ is called the *move* from table v to table u. This move $u - v$ is another r-way table with entries equal to zero in the cell $(i_1, \ldots, i_u)$ if $u_{i_1,\ldots,i_u} = v_{i_1,\ldots,i_u}$, a positive integer in the $(i_1, \ldots, i_u)$ cell if $u_{i_1,\ldots,i_u} > v_{i_1,\ldots,i_u}$, and a negative integer in the $(i_1, \ldots, i_u)$-cell otherwise. Note that, by definition, S is linear; thus the sufficient statistic of any move connecting two tables in the same fiber, $S(u - v)$, is zero. In particular, adding a move to a contingency table does not change the value of the sufficient statistic vector. We will call any table $m \in \mathbb{Z}^{m_1 \times \cdots \times m_u}$ such that $S(m) = 0$ a *Markov move* on $\mathcal{F}_S$. Thus, to discuss walks on a fiber, we may either specify the start and target tables v and u, or the Markov move $m = u - v$.

A *Markov basis* B is a set of Markov moves such that for any fiber $\mathcal{F}_S$ and any two contingency tables $u, v \in \mathcal{F}_S = \mathcal{F}_{S(u)}$, there exists a sequence of moves $m_1, \ldots, m_k \in B$ such that v is reachable from u by the corresponding walk on the fiber $\mathcal{F}_{S(u)}$, i.e., $u = v + \sum_{i=1}^{k} m_i$ and each partial sum $u_l = v + \sum_{i=1}^{l} m_i$, $l < k$, is a table in the fiber $\mathcal{F}_{S(u)}$ (that is, u_l has nonnegative entries). The existence and finiteness of a Markov basis is guaranteed by the Fundamental Theorem of Markov bases from Diaconis and Sturmfels (1998), which states that the moves correspond to generators of an algebraic object (namely, the toric ideal) associated with each log-linear model. Equipped with a set of moves, one can perform a random walk on the fiber $\mathcal{F}_S$. A priori, the resulting Markov chain need not be irreducible; however, if the set of moves is a Markov basis, then irreducibility is guaranteed. Moreover, a Metropolis-Hastings algorithm can be used to adjust the transition probabilities, returning a chain whose stationary distribution is exactly the conditional distribution on the given fiber.

In the remainder of this section, we discuss dynamically constructing arbitrary elements of a Markov basis B for log-linear models using *the parameter hypergraph* of

the model. For simplicity, we restrict ourselves to log-linear models with 0/1 design matrices (that is, parameters do not appear with multiplicities in the model parametrization), although the definition and construction could be extended to a more general case. As mentioned in the introduction, this method will be particularly useful in several cases: when B cannot be computed in its entirety. For example, this can be the case when the model is not decomposable, meaning that the divide-and-conquer strategy of Dobra and Sullivant (2004) cannot be applied, or when sufficient statistics of the model are more complex than table marginals and the table is large. To that end, we define the main tool of our construction.

2.2 From tables to hypergraphs

Let $\mathcal{M} := \mathcal{M}_S$ be any log-linear model for discrete random variables $Z_1, \ldots, Z_m$ with sufficient statistic S. Suppose that the joint probabilities of the model are such that the parameters $\theta_1, \ldots, \theta_n$ appear without multiplicities (that is, S can be obtained from the table in a linear fashion).

Definition 1 The model $\mathcal{M}$ is encoded by a hypergraph $H_{\mathcal{M}}$ on the vertex set $\theta_1, \ldots, \theta_n$, which is constructed as follows: $\{\theta_j\}_{j \in J}$ is an edge of $H_{\mathcal{M}}$ if and only if the index set J describes one of the joint probabilities in the model, i.e., there exist values $i_1, \ldots, i_m$ such that, up to the normalizing constant, $\text{Prob}(Z_1 = i_1, \ldots, Z_m = i_m) \propto \prod_{j \in J} \theta_j$. The hypergraph $H_{\mathcal{M}}$ is called the *parameter hypergraph of the model* $\mathcal{M}$.

Notation 1 For convenience we gather the notational conventions we will use throughout the manuscript. Log-linear models will be denoted by $\mathcal{M}_S$ with sufficient statistic S, or simply $\mathcal{M}$ when S is clear from context. The parameter hypergraph $H_{\mathcal{M}} = (V, E)$ has vertex set V and edge set E. Edges in the hypergraph are written as products of parameters instead of the usual lists, e.g., $\theta_1 \cdots \theta_k$ will represent the edge $\{\theta_1, \ldots, \theta_k\}$.

The easiest way to understand $H_{\mathcal{M}}$ is to view it as depicting the structure of parameter interactions. Since vertices of the hypergraph represent parameters of the model, edges in $H_{\mathcal{M}}$ collect all the parameters that appear in a joint probability under the model. There is a one-to-one map between the contingency table cell labels and edges in the parameter hypergraph. Let us illustrate on two simple but familiar examples.

Example 1 (Two independent random variables) Consider the independence model of two discrete random variables Z_1 and Z_2, taking a and b values, respectively. Denote the marginal probabilities $\text{Prob}(Z_1 = i)$ and $\text{Prob}(Z_2 = j)$ by x_i and y_j, respectively. Since the independence model for Z_1 and Z_2 is specified by the formula $P_{ij} := \text{Prob}(Z_1 = i, Z_2 = j) = x_i y_j$, we see that the parameter hypergraph $H_{Z_1 \perp\!\!\!\perp Z_2}$ has $a+b$ vertices: $x_1, \ldots, x_a, y_1, \ldots, y_b$ and an edge between every x_i and y_j. Thus, in this case, the hypergraph is the complete bipartite graph on $\{x_1, \ldots, x_a\} \sqcup \{y_1, \ldots, y_b\}$, depicted in Fig. 2a.

Example 2 (Quasi-complete independence) For a $l \times m \times n$ table, the quasi-complete independence model is a complete independence model with structural zeros. If the

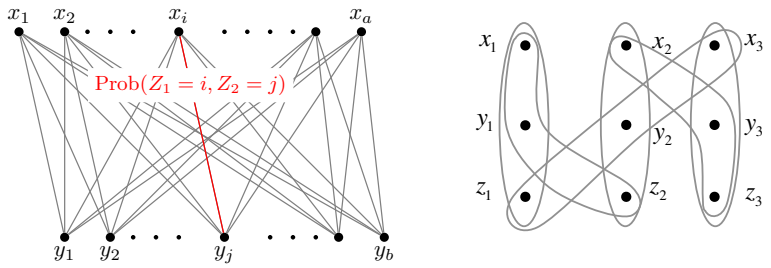

Fig. 2 Two examples of parameter hypergraphs. **a** Independence model: Example 1. **b** Quasi independence model: Example 2

cell (i, j, k) is a structural zero, then $\text{Prob}(Z_1 = i, Z_2 = j, Z_3 = k) = 0$; otherwise, $\text{Prob}(Z_1 = i, Z_2 = j, Z_3 = k) = x_i y_j z_k$ where $x_i = \text{Prob}(Z_1 = i)$, $y_j = \text{Prob}(Z_2 = j)$ and $z_k = \text{Prob}(Z_3 = k)$ are marginal probabilities.

To obtain the parameter hypergraph for the quasi-complete independence model, we start with the complete 3-partite hypergraph with vertex partition V_1, V_2 and V_3 such that $\#V_1 = l, \#V_2 = m$ and $\#V_3 = n$ and then remove every edge that corresponds to a cell with a structural zero. The hypergraph in Fig. 2b is the parameter hypergraph for the quasi complete independence model on a $3 \times 3 \times 3$ table where all cells are structural zeros except (1, 1, 1), (1, 1, 2), (2, 2, 2), (2, 3, 3), (3, 2, 1) and (3, 3, 3).

In the next section (Definition 5) we will see a more complex example in H_{p_1}, the parameter hypergraph for the edge-dependent reciprocation version of the p_1 model.

A crucial observation about the parameter hypergraph is that it not only encodes the parameter interactions, but also that any observed table can be viewed as a subset of its edges, with multiplicities if the model allows them. Specifically, suppose the table u has an entry 1 in the cell $(i_1, \ldots, i_u)$. If the model postulates $\text{Prob}(X_1 = i_1, \ldots, X_u = i_u) \propto \theta_{j_1} \cdots \theta_{j_k}$, then the $(i_1, \ldots, i_u)$-cell entry is represented by the edge $\theta_{j_1} \cdots \theta_{j_k}$. A larger entry (say, 3) in the table would be represented by an edge with multiplicities (the edge $\theta_{j_1} \cdots \theta_{j_k}$ would have multiplicity 3). Multiplicities are recorded with a function $\mu : E \to \mathbb{Z}$ (e.g., $\mu(\theta_{j_1} \cdots \theta_{j_k}) = 3$).

Definition 2 The list of edges

$$\{\theta_{j_1} \cdots \theta_{j_k} : u_{(i_1,\ldots,i_u)} > 0 \text{ and } \text{Prob}(X_1 = i_1, \ldots, X_u = i_u) \propto \theta_{j_1} \cdots \theta_{j_k}\},$$
$$\text{where edge } \theta_{j_1} \cdots \theta_{j_k} \text{ appears } \mu(\theta_{j_1} \cdots \theta_{j_k}) = u_{(i_1,\ldots,i_u)} \text{ times},$$

will be denoted by $e(u)$. It is the multiset of edges representing the table u and has support in the edge set E of the parameter hypergraph. Finally, for the multiset of edges $\mathcal{E}$ of the parameter hypergraph such that $e(u) = \mathcal{E}$, we will write $e^{-1}(\mathcal{E})$ to denote the table u.

Next, notice that the sufficient statistic $S(u)$ can be calculated from the hypergraph edges $e(u)$, since the vertices covered by $e(u)$ represent those natural parameters that affect the computation of $S(u)$. In the independence model example (cf. Example 1),

if u is the 2×2 table with 1 in cell $(1, 2)$ and a 2 in the cell $(2, 1)$, then $e(u) = \{x_1y_2, x_2y_1, x_2y_1\}$. The entries of the sufficient statistic of the table under $Z_1 \perp\!\!\!\perp Z_2$ are the row and column sums; the first row having sum 1 means that x_1 appears once in the set of edges $e(u)$; in other words, the degree of the vertex x_1 is 1. The first column having sum 2 means that y_1 has degree 2 in $e(u)$. Therefore, the sufficient statistic vector $S(u)$ equals the *degree vector* of the multi-hypergraph $(V, e(u))$. It is obtained by simply counting the number of edges incident to each vertex in $e(u)$ and setting the degree of all other vertices in V to zero.

Finally, we describe how to construct and explore the fiber $\mathcal{F}_{S(u)}$. Preserving the value of the vector $S(u)$ means finding another edge set $e(v)$ such that the degree vector of $e(v)$ is the same as that of $e(u)$. If we view the edges $e(u)$ as colored red and $e(v)$ blue, then the move $v - u$ corresponds to a collection of edges $(e(v), e(u))$, where each vertex appears in the same number of blue and red edges.

We have thus shown the following is an equivalent way to view the fiber $\mathcal{F}_{S(u)}$ and its connecting moves:

Theorem 1 *Recall that an observed table u is represented by a multiset $e(u)$ of edges on the hypergraph $H_{\mathcal{M}}$.*

(a) *The fiber $\mathcal{F}_{S(u)}$ consists of all multisets of edges of $H_{\mathcal{M}}$ with degree vector equal to $S(u)$.*

(b) *Any move $v - u$ in the Markov basis connecting u to some $v \in \mathcal{F}_{S(u)}$ is represented by the edge sets $(e(v), e(u))$ over the parameter hypergraph $H_{\mathcal{M}}$ such that the degree vector of $e(v)$ is the same as that of $e(u)$.*

In particular, exact testing for a log-linear model $\mathcal{M}$ reduces to finding all sub-hypergraphs of $H_{\mathcal{M}}$ with a given degree sequence.

Theorem 1 implies that sampling the fiber of any log-linear model is equivalent to sampling the space of sub-hypergraphs of the parameter hypergraph with fixed degree sequence and can thus be used to approximate the exact p-value. We call a collection $(e(v), e(u))$ from Theorem 1 a *(color-)balanced* edge set; balanced edge sets are defined and discussed in more detail in Petrović and Stasi (2014). It is shown in Petrović and Stasi (2014) that the collection of all such sets constitute a Markov (and in fact, the Graver) basis. Complexity of minimal Markov moves to connect a given (unrestricted) fiber is studied in Gross and Petrović (2013).

For convenience, let us summarize here the hypergraph notation we will use in the following section:

Notation 2 For an observed table u, the (multi)set of red (observed) hyperedges $e(u)$ will be denoted by $\mathcal{R}$, and any blue (multi)set that balances the vertices covered by $\mathcal{R}$ will be denoted by $\mathcal{B}$. Note that every $\mathcal{B}$ corresponds to a table $v \in \mathcal{F}_{S(u)}$. The move $v - u$ will be denoted as $\mathcal{W} = (\mathcal{B}, \mathcal{R})$.

Remark 1 By abuse of notation, we will also denote by $(\mathcal{B}, \mathcal{R})$ only those edges over $H_{\mathcal{M}}$ representing the non-zero entries of the move $v - u$. Indeed, if a cell has the same value in both tables, the move directly connecting the tables does not affect that cell; thus the corresponding edge need not be recorded in $(\mathcal{B}, \mathcal{R})$. If it is included in this set, then the move simply subtracts and adds 1 to the cell in the table, that is, it removes and then adds back the particular edge in $e(u)$.

2.3 Sampling constraints and applicable moves

As mentioned briefly in the introduction, a Markov basis will connect all table realizations in a fiber that are subject to the constraint that each table entry is non-negative. However, in the presence of table cell bounds or structural zeros (e.g., Bishop et al. 1975, Sect. 5.1), Markov moves will inevitably produce tables whose cell entries exceed these bounds. These sampling constraints often arise in real-world data. In the network modeling case, a structural zero means a certain relation or edge can never be observed, while a cell bound puts a restriction on how many times an edge between two nodes can be observed in any instance of the network. In fact, most (simple) network models begin with a basic assumption that allows only one edge per dyad, for example, the p_1 model from Holland and Leinhardt (1981) (see also Fienberg et al. 2010) and the beta model from Chatterjee et al. (2011). This introduces another problem for running random walks on fibers: at any given step, the table or network produced may not be observable, and so many of the steps in the walk will be rejected. In fact, these rejections are likely to occur because the usual Markov bases are blind to data and sampling constraints. To compound this problem, a Markov basis only guarantees that the fiber of non-negative table realizations is connected. It is quite reasonable to expect that there exist two tables in the same fiber such that every path connecting them traverses a table that does not satisfy the additional cell bounds. In this sense, the sampling constraints have suddenly *disconnected* the fiber $\mathcal{F}_S$! With this in mind, we will differentiate between the usual fiber $\mathcal{F}_S$ and what we call the observable fiber $\overline{\mathcal{F}}_S$:

Definition 3 The *observable fiber* $\overline{\mathcal{F}}_S \subsetneq \mathcal{F}_S$ is the set of all realizations u of the contingency table $U \in \mathbb{Z}_{\geq 0}^{m_1 \times \cdots \times m_u}$ with nonnegative entries and sufficient statistic S that respect the sampling constraints of the model, i.e., integer bounds on cells or structural zeros.

For example, in the p_1 model, the observable fiber $\overline{\mathcal{F}}_S$ contains only *simple* directed graphs, which means each cell in the contingency table representing the directed graph is either a 0 or a 1. Naturally, there is a corresponding condition on the hypergraph: no edge in $e(u)$ representing the table u can have multiplicity larger than 1. Thus any move $(\mathcal{B}, \mathcal{R})$ applied to $e(u) \supseteq \mathcal{R}$ must be such that in the resulting set of edges, $(e(u) \setminus \mathcal{R}) \cup \mathcal{B} \subseteq H_{\mathcal{M}}$, every edge appears at most once.

For the case of $0/1$ contingency tables, that is, tables with cell bound of 1 everywhere, Hara and Takemura (2010) study the observable fibers and show in Proposition 2.1 that the squarefree part of the Graver basis will connect *any* fiber $\overline{\mathcal{F}}_S$ *respecting* $0/1$ sampling constraints. Here, "squarefree part" simply means that each entry in the table representing the move $u - v$ is either 0 or 1; we will say that such a move *respects the* $0/1$ *sampling constraint*. Their result is, in fact, more general, and applies to higher integer cell bounds and structural zeros as well.

Proposition 1 (Hara and Takemura 2010) *The elements of the Graver basis which respect the sampling constraints suffice to connect the observable fiber in all cases where sampling constraints are integer bounds on cells.*

The proof relies on an algebraic fact that moves correspond to binomials in a toric ideal, and every binomial in the ideal can be written as a conformal sum of Graver basis elements. We will not go into technical details of this result here; the reader is referred to Sturmfels (1996) and recent text Aoki et al. (2012).

In general, there are more squarefree moves in the Graver basis than there are in a minimal Markov basis, though we should be clear that the latter need not be a subset of the former. In particular, the set of squarefree Graver elements almost never equals the squarefree moves from a minimal basis. Moreover, the Graver basis is notoriously difficult to compute, providing another reason against pre-computing the moves for the given model, and instead, generating dynamically only those moves that can be applied to the observed table or network and remain in the observable fiber $\overline{\mathcal{F}}_S$.

Definition 4 A move $v-u$ is said to be *applicable* to a point u in the fiber (equivalently, to the network represented by a table u) if it produces another point v in the observable fiber $\overline{\mathcal{F}}_S$, respecting the sampling constraints of the model at hand.

In terms of the hypergraph edges, the move $v-u$, represented as $(\mathcal{B}, \mathcal{R})$, is applicable if $(e(u)\backslash\mathcal{R}) \cup \mathcal{B} = e(v)$ for some table $v \in \overline{\mathcal{F}}_S$.

We now extend Theorem 1 to characterize applicable Graver moves in terms of the parameter hypergraph. By Theorem 2.8 in Petrović and Stasi (2014) and the Fundamental Theorem of Markov bases, *any* move corresponds to a balanced edge set of $H_{\mathcal{M}}$. Furthermore, moves in the Graver bases correspond to the *primitive* balanced edge sets of $H_{\mathcal{M}}$. We can summarize applicable Graver moves in terms of $H_{\mathcal{M}}$ in the following way:

Corollary 1 *Adopt Notation 2. Any move $v-u$ in the* Graver *basis that is applicable to u is a balanced edge set $(\mathcal{B}, \mathcal{R})$ of the parameter hypergraph $H_{\mathcal{M}}$ such that*

1. $\mathcal{R} \subseteq e(u)$,
2. $(e(u)\backslash\mathcal{R}) \cup \mathcal{B} = e(v)$ *for some table* $v \in \overline{\mathcal{F}}_S$ *and*
3. *there exists no move* $(\mathcal{B}', \mathcal{R}')$ *such that* $\mathcal{B}' \subset \mathcal{B}$ *and* $\mathcal{R}' \subset \mathcal{R}$.

In the result above, (1) ensures non-negativity of the resulting table v, (2) ensures the move is applicable and (3) ensures the move is a Graver basis element. That the moves connect the observable fiber of tables u is a corollary of Proposition 1. In practice, checking condition (3), primitivity, is a non-trivial task; instead, an algorithm that produces each Graver move with positive probability suffices for goodness-of-fit testing purposes. For example, in Sect. 3 we run walks on fibers using Graver basis elements along with larger applicable moves.

In summary, Corollary 1 gives us the exact recipe we need to dynamically generate applicable moves.

2.4 Metropolis-Hastings using parameter hypergraphs

Now that we have the language to describe applicable moves in terms of the parameter hypergraph, we describe a Metropolis algorithm where the moves are generated dynamically with respect to the parameter hypergraph. In particular, we embed the

combinatorial idea from Corollary 1 within the Metropolis–Hastings algorithm to perform random walks on fibers. We refer to it as the *Metropolis–Hastings using parameter hypergraphs* to distinguish it from the Metropolis–Hastings algorithm stated, for example, in (Aoki et al. 2012, Algorithm 2.1) and (Drton et al. 2009, Algorithm 1.1.13). The latter requires a Markov basis as input, while the algorithm proposed below constructs moves dynamically from the current state.

Algorithm 1: Metropolis–Hastings using parameter hypergraphs

input : $u \in \mathcal{T}(n)$, a contingency table (or $G = g$, a network represented by u),
$S(u)$, the sufficient statistic for the model $\mathcal{M}$,
N the number of steps,
$f(\cdot|S(u))$ conditional probability distribution,
$GF(\cdot)$, test statistic

output: Estimate of p-value

1 Compute the MLE $\tilde{p}$.
2 Set $GF_{\text{observed}} := GF(u)$.
3 Randomly select a multiset of hyperedges $\mathcal{R}$ from $e(u)$.
4 Find a multiset of hyperedges $\mathcal{B}$ from $H_{\mathcal{M}}$ that balances $\mathcal{R}$, ensuring that each Graver move $(\mathcal{B}, \mathcal{R})$ has positive probability of being constructed.
5 Set $m = e^{-1}(\mathcal{R}) - e^{-1}(\mathcal{B})$.
6 $q = \min\left\{1, \frac{f(U=u+m|t)}{f(U=u|t)}\right\}$.
7 $u = \begin{cases} u+m, & \text{with probability } q \\ u, & \text{with probability } 1-q \end{cases}$
8 **if** $GF(u) > GF_{observed}$ **then**
9 $\quad k = k + 1$.
10 Repeat Steps 3-9 N times.
11 Output $\frac{k}{N}$.

To carry out the specific implementation of Steps 3 and 4 for a given log-linear model, one should take advantage of the model's specific structure revealed by its parameter hypergraph $H_{\mathcal{M}}$. Namely, the observed table u should be interpreted as a multiset of edges $e(u)$ on $H_{\mathcal{M}}$, as explained in Theorem 1(a). Then, these edges should be rearranged into another set of edges on $H_{\mathcal{M}}$ with the same degree sequence: this is what the move $(\mathcal{R}, \mathcal{B})$ represents. Of course, we cannot determine the specifics of the procedure of finding $\mathcal{B}$ given $\mathcal{R}$ for every possible model at once, because the structure of the hypergraph dictates which edges are allowed and which are not. This is a difficult problem, but there is hope that specific models used in practice and, therefore, their associated $H_{\mathcal{M}}$s, are generally well structured.

In order to achieve convergence in Algorithm 1, attention needs to be paid to the proposal for $\mathcal{B}$ in Step 4. Symmetry and aperiodicity of the Markov chain produced will suffice though; irreducibility is automatic as we require the set of moves we generate to be a superset of the Graver basis. Indeed, if the procedure for finding $\mathcal{B}$ in Step 4 is symmetric and aperiodic for any choice of $\mathcal{R}$, then Algorithm 1 is in fact

a Metropolis–Hastings algorithm and, as $N \to \infty$, the output k/N will converge to $P(GF(U) \geq GF(u) \,|U \in \mathcal{F}_{S(u)})$ (Drton et al. 2009; Robert and Casella 1999).

In Sect. 3, we implement Steps 3 and 4 of Algorithm 1 to dynamically produce dynamically applicable moves for the Holland–Leinhardt p_1 model by relying on the nice structure of its parameter hypergraph, while ensuring that our proposal is symmetric and aperiodic. A smaller example illustrating the process on the independence model, a model that should be familiar to the reader, appears in Example 3.

The use of $H_{\mathcal{M}}$ allows us to bypass two crucial issues of the usual chain (Diaconis and Sturmfels (1998)), which relies on precomputing a minimal Markov basis, and which are summarized in the last paragraph of Dobra (2012). First, Algorithm 1 does not require computing the full Markov basis, or the full Graver basis as may be required due to sampling constraints. Second, the number of rejections in the usual Metropolis-Hastings are significantly reduced, since rejections are due to the fact that most moves drawn from the full Markov basis will be non-applicable to the current table and our method will never generate a move that violates the lower bounds of the entry-constrains. This, in turn, should have positive impact to the mixing time of the chain.

Example 3 (Steps 3 and 4 of Algorithm 1 for the independence model) Suppose we observe a 5×5 contingency table all of whose entries are 0 except the $(1, 1)$ and $(2, 2)$ entries, which are 1. There are 200 moves in a minimal Markov basis for the independence model $Z_1 \perp\!\!\!\perp Z_2$. However, only one of those is applicable, namely

$$\begin{array}{|c|c|c|c|c|}\hline -1 & 1 & 0 & 0 & 0 \\ \hline 1 & -1 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 \\ \hline \end{array},$$

or, written in terms of the parameter hypergraph, $\mathcal{W} = (\mathcal{B}, \mathcal{R})$ where $\mathcal{B} = \{x_1y_2, x_2y_1\}$ and $\mathcal{R} = \{x_1y_1, x_2y_2\}$. This move replaces the entries $(1, 1)$ and $(2, 2)$ by 0, and entries $(1, 2)$ and $(2, 1)$ by 1. Any other move will produce negative entries in the table and thus move outside the fiber. A more interesting example can be similarly constructed on a k-way table that is either sparse or has many non-zero entries but $\overline{\mathcal{F}_S}$ allows only $0/1$ entries.

Next, suppose the observed table is

$$u = \begin{array}{|c|c|c|c|c|}\hline 3 & 2 & 0 & 1 & 0 \\ \hline 1 & 0 & 0 & 0 & 1 \\ \hline 0 & 0 & 0 & 2 & 0 \\ \hline 0 & 1 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 \\ \hline \end{array}.$$

The table u is represented by the multiset of edges

$$e(u) = \{x_1y_1, x_1y_1, x_1y_1, x_1y_2, x_1y_2, x_1x_4, x_2y_1, x_2y_5, x_3y_4, x_3y_4, x_4y_2\}$$

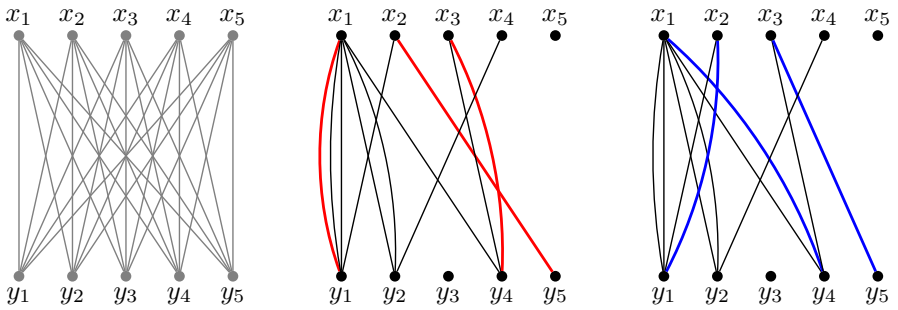

Fig. 3 Example 3: parameter hypergraph (*left*); observed tables $e(u)$ and $e(v)$ with applicable move $(\mathcal{B}, \mathcal{R})$ highlighted (*center* and *right*)

from the independence model (hyper)graph illustrated in Fig. 2a. Denote the bipartite (hyper)graph in Fig. 2a as G. It is known that any Markov move for the independence model corresponds to a collection of closed even walks on G, and any Graver move corresponds to a primitive closed even walk on G. For a detailed account of the correspondence between primitive balanced edge sets of G and primitive closed even walks (see Villarreal 2000). Due to this correspondence, a natural procedure for performing Step 4 in Algorithm 1 is to randomly select a set of edges from $e(u)$, say, $\mathcal{R} = \{x_1y_1, x_2y_5, x_3y_4\}$, and then complete a closed even walk on $\mathcal{R}$, so that the new edges form $\mathcal{B} = \{x_2y_1, x_3y_5, x_1y_4\}$. Notice $\mathcal{R}$ and $\mathcal{B}$ have the same degree vector and $(\mathcal{B}, \mathcal{R})$ is applicable to u. This move is depicted in Fig. 3. The first figure is the parameter hypergraph. The second represents the observed table $e(u)$, with edges in $\mathcal{R}$ highlighted. The third is the edge set $e(v)$ with $\mathcal{B}$ highlighted. The resulting table is

$$v = \begin{array}{|c|c|c|c|c|}\hline 2&2&0&2&0\\\hline 2&0&0&0&0\\\hline 0&0&0&1&1\\\hline 0&1&0&0&0\\\hline 0&0&0&0&0\\\hline\end{array}.$$

We note that if every multiset of $e(u)$ of the same size is drawn with equal probability while selecting $\mathcal{R}$, then the procedure described above would be symmetric and aperiodic, and thus, Algorithm 1 would converge as desired.

3 Goodness-of-fit testing for the p_1 model

In a seminal 1981 paper (Holland and Leinhardt 1981), Holland and Leinhardt introduced what they referred to as the p_1 model for dyadic relational data in a social network summarized in the form of a directed graph. Their model, which is log-linear in form (Fienberg and Wasserman 1981), allows for effects due to differential attraction (popularity) and expansiveness, as well as an additional effect due to reciprocation. For each dyad, a pair of nodes (i, j), the parameter α_i describes the effect of an outgoing edge from i, and β_j the effect of an incoming edge pointed towards j, while ρ_{ij} corresponds to the added effect of reciprocated edges. The parameter θ quantifies

the average "density" of the network, i.e., the tendency of having edges, and λ_{ij} is a normalizing constant to ensure that the probabilities for each dyad (i, j) add to 1.

Given a directed graph, each dyad (i, j) can occur in one of the four possible configurations: no edge, edge from i to j, edge from j to i and a pair of reciprocated edges between i and j. The model postulates that, for each pair (i, j), the probabilities of observing the four possible configurations, in that order, satisfy the following equations:

$$\begin{aligned}
p_{ij}(0,0) &= \exp[\lambda_{ij}] \\
p_{ij}(1,0) &= \exp[\lambda_{ij} + \alpha_i + \beta_j + \theta] \\
p_{ij}(0,1) &= \exp[\lambda_{ij} + \alpha_j + \beta_i + \theta] \\
p_{ij}(1,1) &= \exp[\lambda_{ij} + \alpha_i + \beta_j + \alpha_j + \beta_i + 2\theta + \rho_{ij}].
\end{aligned}$$

where

$$\sum_i \alpha_i = \sum_j \beta_j = 0.$$

We will focus on the *edge-dependent* version of the reciprocation parameter, where $\rho_{ij} = \rho_i + \rho_j + \rho$.

Making the following substitutions

$$\alpha_i' = e^{\alpha_i + \theta}, \quad \beta_i' = e^{\beta_i}, \quad \rho_i' = e^{\frac{1}{2}\rho + \rho_i}, \quad \lambda_{ij}' = e^{\lambda_{ij}}$$

and ignoring the superscripts for convenience, we arrive at the following simplified equations to describe the probability of observing each configuration for a pair (i, j):

$$\begin{aligned}
p_{ij}(0,0) &= \lambda_{ij} \\
p_{ij}(1,0) &= \lambda_{ij}\alpha_i\beta_j \\
p_{ij}(0,1) &= \lambda_{ij}\alpha_j\beta_i \\
p_{ij}(1,1) &= \lambda_{ij}\alpha_i\beta_j\alpha_j\beta_i\rho_i\rho_j.
\end{aligned}$$

While normalizing constants are usually ignored, we will follow Petrović et al. (2010) and treat λ_{ij} as a model parameter. The advantage of this technique is that, given an observable network g, these extra parameters ensure that the sampling constraint of a dyad (pair) $\{i, j\}$ being observed in one and only one state is satisfied for all networks in $\mathcal{F}_{S(g)}$. Effectively this implies that every point in the fiber $\mathcal{F}_{S(u)}$ of a network u is observable. So while in this case we do not need to distinguish between the observable fiber and the fiber, we state our main result regarding the observable fiber to emphasize how this approach generalizes to other network models.

Definition 5 (The parameter hypergraph of the p_1 model) We will denote the parameter hypergraph of the p_1 model as H_{p_1}. Recall that the hyperedges of H_{p_1} are determined by the parameters appearing in the joint probabilities of the model. Thus, for the p_1 model with edge reciprocation there are three types of hyperedges: singletons (corresponding to $p_{ij}(0, 0)$ for each dyad (i, j)), hyperedges of size 3 (corresponding to $p_{ij}(1, 0)$ and $p_{ij}(0, 1)$) and hyperedges of size 7 (corresponding to $p_{ij}(1, 1)$).

More formally, $H_{p_1} = (V_p, E)$, where $V_p = \{\alpha_i, \beta_i, \rho_i : 1 \le i \le n\} \cup \{\lambda_{ij} : 1 \le i < j \le n\}$, and $E = E_1 \cup E_3 \cup E_7$, with $E_1 = \{\lambda_{ij} : 1 \le i < j \le n\}$, $E_3 = \{\alpha_i \beta_j \lambda_{ij} : 1 \le i \ne j \le n\}$, and $E_7 = \{\alpha_i \alpha_j \beta_i \beta_j \rho_i \rho_j \lambda_{ij} : 1 \le i < j \le n\}$.

By definition, the sufficient statistics of the p_1 model are the in- and out- degrees of every node in the network and, in the case of edge-dependent reciprocation, the counts of reciprocated edges incident to each node. Interpreting the network as a contingency table in a natural way as in Fienberg and Wasserman (1981) reveals that these statistics are not table marginals; instead they are subtable sums and marginals of a specific lower-dimensional subtable.

3.1 Markov moves for the p_1 model

Here we describe the form of a Markov move $\mathcal{W} = (\mathcal{B}, \mathcal{R})$ for the p_1 model with edge-dependent reciprocation in terms of the parameter hypergraph H_{p_1} given in Definition 5. The moves can be described in terms of balanced edge sets on a graph obtained by contracting hyperedges in H_{p_1}. Note that by definition, balanced edge sets on graphs reduce to collections of closed even walks.

Let A_n be the undirected bipartite graph on $2n$ vertices with vertex set

$$V(A_n) = \{\alpha_i \mid 1 \le i \le n\} \cup \{\beta_i \mid 1 \le i \le n\}$$

and edge set

$$E(A_n) = \{\alpha_i \beta_j \mid 1 \le i \ne j \le n\}.$$

Let K_n be the undirected complete graph on the n vertices $\{\rho_j \mid 1 \le j \le n\}$. The graphs A_n and K_n can be constructed from H_{p_1} as follows: To construct A_n from H_{p_1}, simply consider all hyperedges of size 3 in H_{p_1}. Each of these hyperedges has vertices $\alpha_j, \beta_k, \lambda_{j,k}$ for some $1 \le j \ne k \le n$. The contracted edges $\alpha_j \beta_k$ (with $\lambda_{j,k}$ deleted) are precisely the edges in A_n. To construct K_n, consider all hyperedges of size 7 in H_{p_1}. Note that each of these edges corresponds to an edge in H_{p_1} that has vertices $\alpha_j, \alpha_k, \beta_j, \beta_k, \lambda_{j,k}, \rho_j, \rho_k$ for some $1 \le j \ne k \le n$. Deleting all the vertices except ρ_j, ρ_k from each hyperedge of size 7 contracts them to size 2, and the result is the complete graph on the n vertices ρ_j, $j = 1, \ldots, n$.

Let $H_{p_1}|_{(3,7)}$ be the subhypergraph of H_{p_1} where $\mathcal{V}(H_{p_1}|_{(3,7)}) = \mathcal{V}(H_{p_1})$ and $E(H_{p_1}|_{(3,7)}) = \{e \in E(H_{p_1}) \mid \#e = 3 \text{ or } \#e = 7\}$. The previous two paragraphs describe a bijection between the edge sets of $A_n \cup K_n$ and $H_{p_1}|_{(3,7)}$:

$$\begin{aligned} \phi : E(A_n \cup K_n) &\to E(H_{p_1}|_{(3,7)}) \\ \alpha_i \beta_j &\mapsto \alpha_i \beta_j \lambda_{ij} \\ \rho_i \rho_j &\mapsto \alpha_i \alpha_j \beta_i \beta_j \lambda_{ij} \rho_i \rho_j. \end{aligned}$$

For a simple balanced edge set $W = (B, R)$ of $A_n \cup K_n$, the set $(\phi(B), \phi(R))$ may not be balanced. However, it can become balanced by appending edges of the form $\{\lambda_{ij}\}$ to the sets $\phi(R)$ and $\phi(B)$. Thus, we define a lifting operation that grows W to a simple balanced edge set of H_{p_1} in this manner:

$$\begin{aligned}\text{lift}\, W := (\mathcal{B}, \mathcal{R}), \text{ where}\\ \mathcal{B} = \phi(B) \cup \{\lambda_{ij} \mid \deg_{\phi(R)}(\lambda_{ij}) > \deg_{\phi(B)}(\lambda_{ij})\} \text{ and}\\ \mathcal{R} = \phi(R) \cup \{\lambda_{ij} \mid \deg_{\phi(B)}(\lambda_{ij}) > \deg_{\phi(R)}(\lambda_{ij})\}.\end{aligned}$$

Let $H_{p_1}|_{(7)}$ be the subhypergraph of H_{p_1} that contains all the hyperedges of H_{p_1} of size 7. Let $H_{p_1}|_{(3)}$ be the subhypergraph of H_{p_1} that contains all the hyperedges of H_{p_1} of size 3. If $\mathcal{W} = (\mathcal{B}, \mathcal{R})$ is a balanced edge set of H_{p_1}, then each ρ_i in the hyperedges of size 7 of $\mathcal{W}$ must be color-balanced. This implies that the α's and β's are color-balanced with respect to $H_{p_1}|_{(7)}$. Thus, it follows that the α's and the β's are color-balanced in $H_{p_1}|_{(3)}$. These observations are noted in Petrović et al. (2010), but in algebraic terms using the binomials of the ideal of the hypergraph $I_{H_{p_1}}$.

Since a balanced edge set $\mathcal{W} = (\mathcal{B}, \mathcal{R})$ on H_{p_1} is a move between two observable networks only if $\deg_{\mathcal{R}}(\lambda_{ij}) = \deg_{\mathcal{B}}(\lambda_{ij}) \in \{0, 1\}$, we arrive at the following proposition:

Proposition 2 *A move between two observable networks g_1 and g_2 in the same fiber is of the form* lift$W$ *such that W is a balanced edge set on $A_n \cup K_n$ and* $\deg_{\mathcal{R}}(\lambda_{ij}) = \deg_{\mathcal{B}}(\lambda_{ij}) \in \{0, 1\}$.

Corollary 2 *For the p_1 model with edge-dependent reciprocation, the set of all $\mathcal{W} = (\mathcal{B}, \mathcal{R})$ such that $\mathcal{W} =$* lift(W) *and W is a balanced edge set of $A_n \cup K_n$ and* $\deg_{\mathcal{R}}(\lambda_{ij}) = \deg_{\mathcal{B}}(\lambda_{ij}) \in \{0, 1\}$ *connects the observable fiber $\overline{\mathcal{F}_S}$ for every possible value of the sufficient statistic S.*

Remark 2 The set of moves described in Corollary 2 is a superset of the applicable square-free Graver basis, as stated in Algorithm 1.

3.2 Generating an applicable move

Now that we have described the general form of the Markov moves for the p_1 model, we give an algorithm for generating an applicable move. Let $g = g_u \cup g_d$ be an observable network written as the union of its reciprocated[1] part g_u and its unreciprocated part g_d. For a directed graph $G = (V, E)$, let undir(G) be the edges of the skeleton[2] of G and let recip$(G) =$ (undir(G), recip(E)), where recip(E) contains both directions of an edge if at least one direction is in E. The following is a general algorithm for generating applicable moves for the p_1 model with edge-dependent reciprocation. It uses the fact that every balanced edge set of a graph corresponds to a set of closed even walks on that graph. The output is either an element of the Graver basis, or an applicable combination of several Graver moves, which themselves need not be applicable. Since the hyperedges of a balanced edge set on H_{p_1} each correspond to a dyadic configuration

[1] Recall that a directed edge (u, v) is called *reciprocated* if (v, u) is also in the network. Otherwise, (u, v) is called *unreciprocated*. In subsequent figures we sometimes draw reciprocated edges as undirected to reduce clutter.

[2] The skeleton of a graph $G = (V, E)$ is the graph obtained by replacing the directed edges in E with their undirected counterparts and then removing multiple edges.

realizable in the network, we will return moves in the form (b, r) where r are the edges to be removed from the network and b are the edges to be added.

Algorithm 2: Generating applicable moves for the p_1 model.

input : $g = g_u \cup g_d$, a directed graph,
c_1, the probability of choosing a Type 1 Move that alters only g_u,
c_2, the probability of choosing a Type 2 Move that alters only g_d,
c_3, the probability of choosing a Type 3 Move that alters both types of edges jointly,
where $c_1 + c_2 + c_3 = 1$.
output: (b, r), an applicable move.

Generate c, a random number between 1 and 3 chosen with probabilities (c_1, c_2, c_3) (weighted coin).
if $c = 1$ **then**
  Use Algorithm 3 to select a Type 1 move. Only reciprocated edges are removed and added. A move of this type corresponds to a set of closed even walks on K_n.
else if $c = 2$ **then**
  Use Algorithm 4 to select a Type 2 move. Only unreciprocated edges are removed and added. A move of this type corresponds to a set of closed even walks on A_n.
else if $c = 3$ **then**
  Use Algorithm 5 to select a Type 3 move. Both types of edges are removed and added. A move of this type corresponds to a set of closed even walks on A_n and a set of closed even walks on K_n.
end

Example 4 Figure 4 illustrates the process of generating a Type 2 move. First the edges (x_2, x_1), (x_3, x_4) and (x_5, x_6) from a network g are chosen. These will be the edges that are removed from g in the move. We consider these edges as edges of A_n. A walk is completed on A_n by adding the blue edges $\{\alpha_2, \beta_6\}$, $\{\alpha_3, \beta_1\}$ and $\{\alpha_5, \beta_4\}$. The blue edges are then interpreted in terms of pairs and dyadic configurations in g. These are the edges that are added to g in the move.

Remark 3 Notice that in each of the above algorithms, it is possible that the trivial move is returned. This means the walk in Algorithm 1 would stay in the same place at that step. While this does not affect the stationary distribution of the Markov chain, it can have a negative impact on mixing times if too many trivial moves are returned. However, this is the problem also with the usual Metropolis-Hastings algorithm, as mixing time questions are generally open. Section 4 shows some indication that the chain seems to be mixing well. In the case of the p_1 model, the probability of returning the trivial move in any of the above algorithms depends on the in and out-degree sequences of the unreciprocated edges and the reciprocated edges. One direction for further research is to understand and try and reduce the output of trivial

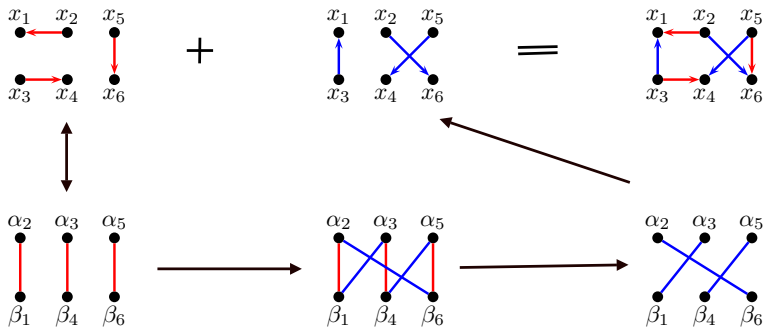

Fig. 4 An example of generating a Type 2 move

moves. Even understanding which networks result in a high probability of a trivial move being returned in Algorithms 3, 4, 5 would be an interesting combinatorial problem.

Algorithm 3: Generating a Type 1 Move

input : g_u, the reciprocated part of a directed graph,
g_d, the unreciprocated part of a directed graph.
output: (b, r), a Type 1 (reciprocated-only) applicable move.

Choose a random subset r_0 of edges from undir(g_u) of size at least two.
for *each edge* $e \in r_0$ **do**
  Randomly direct the edge e and denote the directed edge as a_e.
end
Randomly order the list of directed edges $\{a_e \mid e \in r_0\}$: call the resulting sequence $\mathbf{a}$.
Randomly generate $(\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_k)$ to be a partition of the sequence $\mathbf{a}$ such that each part $\mathbf{a}_j$ contains at least two elements.[3] Note that k, the number of subsequences, is also random.
for $1 \le j \le k$ **do**
  Denote the i_{th} edge in $\mathbf{a}_j$ by a_{e_i}, and let m be the size of the subsequence $\mathbf{a}_j$. Generate the set of directed edges b_j by joining the tail of $a_{e_{i+1}}$ to the head of a_{e_i} for i from 1 to $m-1$ and joining the tail of a_{e_1} to the head of a_{e_m}. In symbols:
  $b_j := \{\, (a_{e_{i+1}}(1), a_{e_i}(2)) \mid 1 \le i < m-1 \,\} \cup \{\, (a_{e_1}(1), a_{e_m}(2)) \,\}$.
  if b_j *is not simple*[4] **then**
    return the trivial move $(\emptyset, \emptyset)$.
  end
end
Let $b = \uplus_{j=1}^{k} \text{recip}(b_j)$. Here $\uplus$ is the multiset union symbol.
Let $r = \text{recip}(r_0)$.
if b *is not simple* **or** $b \cap (E(g_u) \setminus r) \neq \emptyset$ **or** $b \cap E(g_d) \neq \emptyset$ **then**
  return the trivial move $(\emptyset, \emptyset)$.
else
  return (b, r).
end

[3] Choosing this partition is equivalent to choosing a combinatorial composition σ of the number $\#r_0$. The composition σ should be chosen according to a known but arbitrary distribution $P_{\#r_0}(\sigma)$ with full support.

[4] A simple directed graph does not contain directed edges with multiplicity more than one or loops; reciprocated edges are allowed.

Algorithm 4: Generating a Type 2 Move

input : g_d, the unreciprocated part of a directed graph,
g_u, the reciprocated part of a directed graph.
output: (b, r), a Type 2 (non-reciprocated-only) applicable move.

Choose a random subset r of edges from g_d of size at least two.
Randomly order the list of directed edges $\{e \mid e \in r\}$: call the resulting sequence $\mathbf{a}$.
randomly generate $(\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_k)$ to be a partition of the sequence $\mathbf{a}$ such that each part $\mathbf{a}_j$ contains at least two elements. Note that k, the number of subsequences, is also random.
for $1 \leq j \leq k$ **do**
    Denote the i_{th} edge in $\mathbf{a}_j$ by e_i, and let m be the size of the subsequence $\mathbf{a}_j$. Generate the set of directed edges b_j by joining the tail of e_{i+1} to the head of e_i for i from 1 to $m-1$ and joining the tail of e_1 to the head of e_m. In symbols:

$$b_j := \{\, (e_{i+1}(1), e_i(2)) \mid 1 \leq i < m-1 \,\} \cup \{\, (e_1(1), e_m(2)) \,\}.$$

end
Let $b = \uplus_{j=1}^{k} b_j$.
if *b is not simple or contains reciprocated edges,* **or** $b \cap (E(g_d) \setminus r) \neq \emptyset$ **or** $b \cap E(g_u) \neq \emptyset$ **then**
    return the trivial move $(\emptyset, \emptyset)$.
else
    return (b, r).
end

Algorithm 5: Generating a Type 3 Move

input : $g = g_u \cup g_d$, a directed graph
output: (b, r), a Type 3 (mixed) applicable move.

Use Algorithm 3 with input g_u to obtain (b_u, r_u).
Use Algorithm 4 with input g_d to obtain (b_d, r_d).
if $b_u \cap b_d \neq \emptyset$, **or** $(b_u \cup b_d) \cap (E(g) \setminus (r_u \cup r_d)) \neq \emptyset$ **then**
    return the trivial move $(\emptyset, \emptyset)$.
else
    return $(b_u \cup b_d, r_u \cup r_d)$.
end

Proposition 3 *Every move outputted by Algorithms 3, 4, 5 is an applicable Markov move of the form* lift$W$ *such that W is a balanced edge set on $A_n \cup K_n$ and* $\deg_{\mathcal{R}}(\lambda_{ij}) = \deg_{\mathcal{B}}(\lambda_{ij}) \in \{0, 1\}$. *Moreover, on input g_1, if $g_2 \in \mathcal{F}_{S(g)}$ and $g_1 \neq g_2$, Algorithm 2 has a non-zero probability of returning the move $g_2 - g_1$.*

Proof Algorithm 3 chooses a set of edges r_0 from $\text{undir}(g_u)$ and completes k closed even walks on K_n. We will denote the balance edge set of A_n corresponding to this set of closed even walks as W. Step 7 checks that $\text{lift}W = (\mathcal{B}, \mathcal{R})$ satisfies $\deg_{\mathcal{R}}(\lambda_{ij}) = \deg_{\mathcal{B}}(\lambda_{ij}) \in \{0, 1\}$. If the condition is not satisfied, then the trivial move is returned. Otherwise, (b, r), outputted by Algorithm 3, is of the form specified. Applicability of (b, r) follows from the fact that r is a subset of g_u and $\deg_{\mathcal{R}}(\lambda_{ij}) = \deg_{\mathcal{B}}(\lambda_{ij}) \leq 1$. Moves outputted from Algorithms 4, 5 can be analyzed in a parallel fashion.

For the second part of the statement, Proposition 2 states that the move between two networks g_1, g_2 in the same fiber is of the form $\text{lift}W$ where $W = (R, B)$ is a balanced edge set on $A_n \cup K_n$. Assume that R is contained entirely in K_n. Denote the closed even walks on K_n that correspond to W as $W_1, \ldots, W_k$. The move $g_2 - g_1$ will be returned if 1 is chosen in Algorithm 2, the edges of g_1 corresponding to R are chosen at Step 1 of Algorithm 3, and Steps 3 and 4 result in a sequence $\mathbf{a} = (\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_k)$ such that $\mathbf{a}_i$ corresponds to a cyclic permutation of the odd edges of W_i. If R is contained entirely in A_n or contains edges from both A_n and K_n, then a similar argument follows. □

Algorithm 2 and its subroutines, Algorithms 3, 4 and 5, describe the procedure for finding $(\mathcal{B}, \mathcal{R})$ in Steps 3 and 4 of Algorithm 1 (the Metropolis–Hastings algorithm using parameter hypergraphs). Thus, for the p_1 model, each move in the underlying Markov chain of Algorithm 1 proceeds as follows: select a set of reciprocated and unreciprocated edges from the current network in the chain that correspond to a set of hyperedges $\mathcal{R}$ from the parameter hypergraph H_{p_1}; flip a coin and either construct a move on the reciprocated edges (Algorithm 3), on the unreciprocated edges (Algorithm 4), or on both (Algorithm 5); the resulting new edges correspond to a set of hyperedges $\mathcal{B}$ from H_{p_1} with the same degree sequence as $\mathcal{R}$, and finally apply the move with probability q as described in Step 6 of Algorithm 1; this modification guarantees that the desired stationary distribution is attained.

Theorem 2 *Let g be an observable network with more than two edges and with sufficient statistic $S(g)$. The Markov chain, $(\mathcal{G}_t)_{t=0}^{\infty}$, where the step from $\mathcal{G}_i$ to $\mathcal{G}_{i+1}$ is given by Algorithm 2 is an irreducible, symmetric and aperiodic random walk on $\mathcal{F}_S$.*

Proof Irreducibility follows from Proposition 3.

To show symmetry, let $g_1 = g_{1_u} \cup g_{2_d}$ and $g_2 = g_{2_u} \cup g_{2_d}$ be two simple networks with reciprocated parts g_{1_u}, g_{2_u} and unreciprocated parts g_{1_d}, g_{2_d}. The move (b, r) from g_1 to g_2 is the combination of moves (b_u, r_u) from g_{1_u} to g_{2_u} and (b_d, r_d) from g_{1_d} to g_{2_d} where $b = b_u \cup b_d$ and $r = r_u \cup r_d$. The move (b_u, r_u) on the network corresponds to a balanced edge set $W_u = (B_u, R_u)$ on the parameter (hyper)graph K_n, which forms a set of primitive closed even walks on K_n. The probability of choosing r_u in Step 1 of Algorithm 3 is dependent only on the number of edges in g_{1_u}, which is equal to the number of edges in g_{2_u}. Step 5 in Algorithm 3 completes walks on sequences of edges from r_u by connecting heads to tails. Thus, given that r_u was chosen in Step 1, the probability of choosing an ordering of the vertices, an ordering of the edges and a composition in Steps 2–4 such that Step 5 will output b_u is dependent only on the structure of W_u (the primitive walks in W_u, the length of these walks and which of these walks share a vertex). So, since W_u is the same regardless whether

we are moving from g_{1_u} to g_{2_u} or from g_{2_u} to g_{1_u}, the probabilities of making these moves in a single step are equal. A similar situation occurs between the reciprocated parts of g_1 and g_2.

For aperiodicity, notice that every non-diagonal entry of the transition matrix P of $(\mathcal{G}_t)_{t=0}^{\infty}$ is greater than zero. Therefore, since g contains more than two edges, $P^n(i, j) > 0$ for all $n \geq 2$. □

Corollary 3 *If g has more than two edges, then with probability one, as the number of steps $N \to \infty$, the output of Algorithm 1 with steps 3 and 4 implemented according to Algorithm 2 converges to $P(\chi^2(\mathcal{G}) \geq \chi^2(g) \ : \mathcal{G} \in \mathcal{F}_{S(g)})$.*

Algorithm 2 and its subroutines Algorithms 3, 4, 5 are implemented in `R`; the code is available in the supplementary material on Gross et al. (2014). The examples in Sect. 4 that compute estimated p-values use the function `Estimate.p.Value`. It takes an observed network and implements Algorithm 1 using an iterative proportional scaling algorithm (Holland and Leinhardt 1981, p. 40) to compute the MLE, and Algorithm 2 for Step 4. We chose to use the chi-square statistic for the goodness-of-fit statistic.

Our implementation makes use of the `R` package `igraph` Csardi and Nepusz (2006), and in particular its graph data structure and methods for producing graph unions and graph intersections. Each of these methods has complexity linear in the sum of the cardinalities of the edge sets and vertex sets of the input. As a result the complexity of the algorithm is at worst $O((|V|+|E|)^2)$, where V and E are the vertex and edge sets, respectively.

4 Simulations

We apply Algorithms 1 and 2 and run goodness-of-fit tests in `R` on several real-world network datasets as well as simulated networks under the p_1 model. In what follows, the number of steps in the chain along with the initial burn-in is reported. Our statistic of choice for $GF(u)$ is the chi-square statistic, directly measuring the distance of the network u from the MLE. For each simulation, we report the estimated p-value returned on line 11 of Algorithm 2 and the sampling distribution of $GF(u)$.

4.1 A small synthetic network

We begin with a test case to check how Algorithm 2 explores the fiber. In (Ogawa et al. 2013 Sect. 5.1), the authors sample the fiber of an undirected graph H_0 on 8 nodes, depicted in Fig. 5, under the beta model. By enumeration they have determined that the size of the fiber is 591. Considering this graph as a directed network all of whose edges are reciprocated, we can test the fit of the p_1 model as well and study its fiber similarly. The fibers of H_0 under the two models are the same, since in both cases, the fiber consists of all undirected (or reciprocated-edge) graphs with the same (in- and out-) degree vector as H_0.

We ran Algorithm 2 and stored all graphs discovered in the run. Starting from H_0, after 1000 steps, 232 points in the fiber were discovered. After 5000 steps, 538

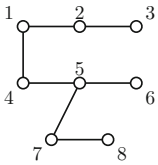

Fig. 5 The graph H_0 from Fig. 13 in Ogawa et al. (2013)

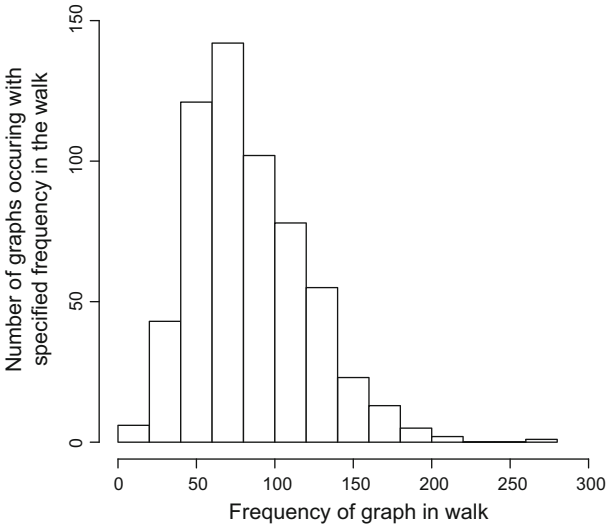

Fig. 6 Histogram from sampling

graphs were discovered, and the entire fiber of 591 graphs was reached after less than 15,000 steps in the chain. At this point, the chain samples the fiber almost uniformly, as the total variation distance between the sampling distribution and the uniform distribution on the fiber is calculated to be 0.2088025 (at the 15,000th step). For comparison purposes, the TV-distance is 0.1703418 after 50,000 steps. Figure 6 shows the histogram of graphs sampled in the 50,000-move walk. Therefore, running a Markov chain of at least 50,000 steps should be sufficient for testing purposes for this example.

A run of Algorithm 1 for 450,000 steps, after 50,000 burn-in steps, produced the values of the chi-square statistics in Fig. 7a and the p-value estimate of 0.86. The estimates of the p-value from the simulation are plotted in Fig. 7b against the step number of the Markov chain and give further evidence of convergence.

4.2 Networks simulated from the p_1 distribution

Consider the four digraphs on 10 nodes that Holland and Leinhardt simulated from the p_1 distribution (see Holland and Leinhardt 1981, Fig. 3). The networks are depicted in Fig. 8.

For each network, chains of length 200,000 provide expected results. The estimated p-values are 0.284774, 0.7185896, 0.4673885 and 0.7432897, respectively. The his-

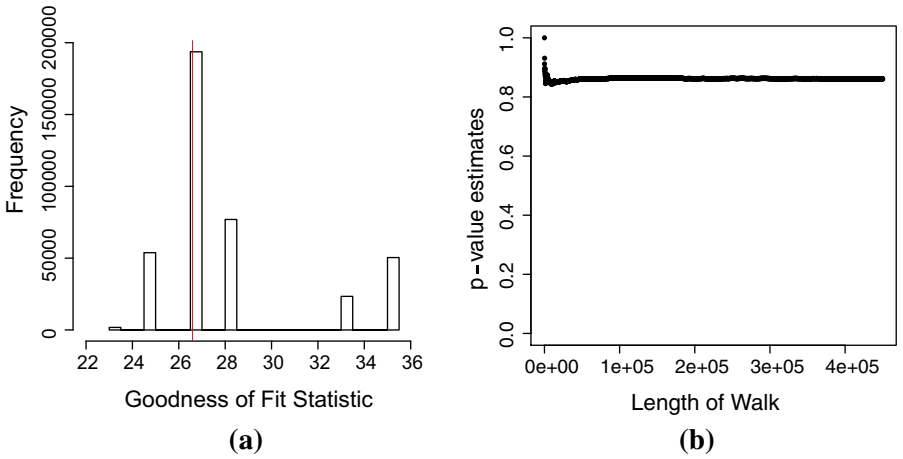

Fig. 7 Simulation results for graph H_0: chain of length 500,000 including 50,000 burn-in steps. **a** Histogram of chi-square statistic. **b** Plot of p-value estimates from a typical run

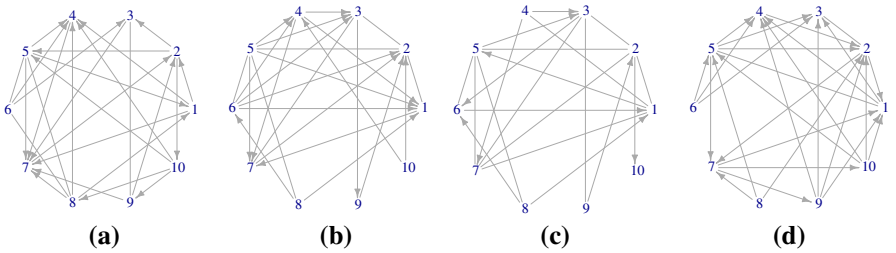

Fig. 8 Four digraphs simulated from the p_1 distribution from Holland and Leinhardt 1981, Figure 3. For clarity, the reciprocated edges are drawn as undirected. **a** Graph 1. **b** Graph 2. **c** Graph 3. **d** Graph 4

tograms of the sampling distribution of the chi-square statistics from the 220,000-step simulation (with 20,000 burn-in steps) are shown in Fig. 9. The p-values reach their estimated value in approximately 25,000 steps after burn in.

4.3 Mobile money networks

Figure 10 is a directed graph on 12 vertices with 13 unreciprocated edges and 15 reciprocated edges. The data are from Kushimba et al. (2013) and were collected through a survey conducted in Bungoma and Trans-Nzoia Counties in Kenya, and among Kenyans living in Chicago, Illinois in the summer of 2012. Vertices represent members of an extended family. An edge from vertex v_i to vertex v_j represents that v_i had sent money to v_j using a mobile money transfer. Since the network depicted in Fig. 10 is a social network and the individuals are social actors, it is reasonable to suspect transitive effects are present. In such a setting, it is expected the p_1 model would not fit this data very well, and, in fact, Holland and Leinhardt (1981) suggest the p_1 model as a realistic null model in such cases.

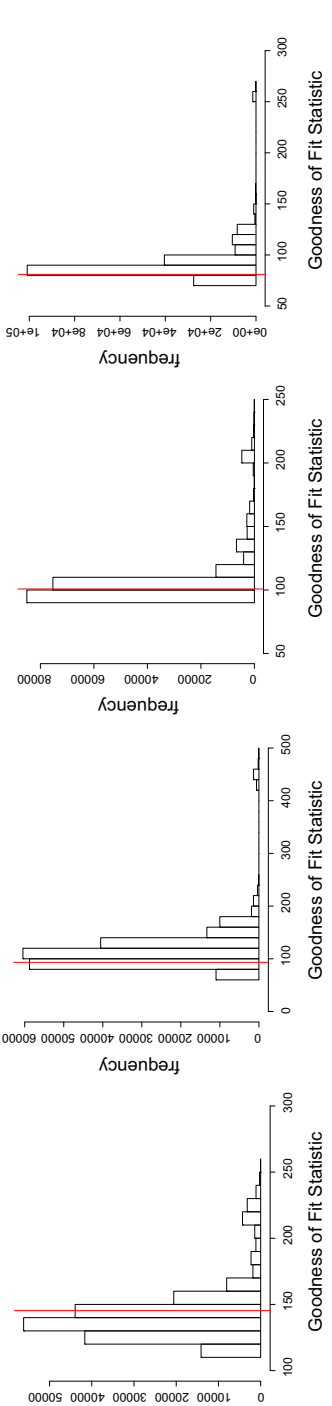

Fig. 9 Histograms of chi-square statistics from sampling the fibers with 220,000 steps (20,000 burn-in steps) for the four digraphs in Fig. 8. **a** Graph 1. p-value: 0.284774. **b** Graph 2. p-value: 0.7185896. **c** Graph 3. p-value: 0.4673885. **d** Graph 4. p-value: 0.7432897

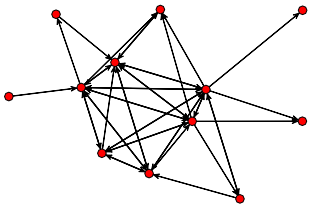

Fig. 10 Mobile money transfers between members of an extended family

Running Algorithm 1 for 300,000 steps after an initial burn-in of 30,000 steps returns an estimated p-value of 0.06024261, which would suggest that the p_1 model with edge-dependent reciprocation is indeed a poor fit for this data, and in fact, if the significance level is set to less than 0.1 we would reject the model. Figure 11a shows the histogram of the sampling distribution of the chi-square statistics with the chi-square statistic for the observed network marked in red. Figure 11b shows the estimated p-value plotted against the step number of the Markov chain and gives evidence of convergence.

4.4 Chesapeake Bay ecosystem

In their 1989 paper (Baird and Ulanowicz 1989), the authors constructed trophic networks for specific regions of the Chesapeake Bay using extensive data gathered from 1983 to 1986. Their work used highly sophisticated estimation methods, relying on a multitude of different sources. Due to their profound detail, Ulanowicz and Baird's food webs have been extensively analyzed over the past 25 years. Often for statistical model-fitting purposes, the edges are considered as undirected. This choice, however, has been largely motivated by the scarcity of tools available to analyze directed networks. Other than heuristic methods, procedures for performing goodness-of-fit testing for directed network models have not existed.

The data set on which we test the p_1 model are depicted in Fig. 12 (see also Baird and Ulanowicz 1989, Figure 2). The list of edges of this directed network was downloaded from Pajek (2004a) and represents the Web 34 Chesapeake Bay Mesohaline Ecosystem. The graph has 39 vertices and 176 edges. The majority of vertices represent species in a Chesapeake Bay food web, with a directed edge $u \to v$ indicating that species u eats species v. we note that other elements that are not species, such as passive carbon storage compartment are also included as vertices. There are 6 reciprocated edges in the graph.

We expect a block structure in food networks that do not naturally occur in p_1-model generated networks. In fact, the estimated p-value is 0.03459158, indicating that the p_1 model with edge-dependent reciprocation is not a good for this data. If the significance level is set to less than 0.05, we would reject this model. The histogram of a simulation with 1,000,000 steps is shown in Fig. 1b.

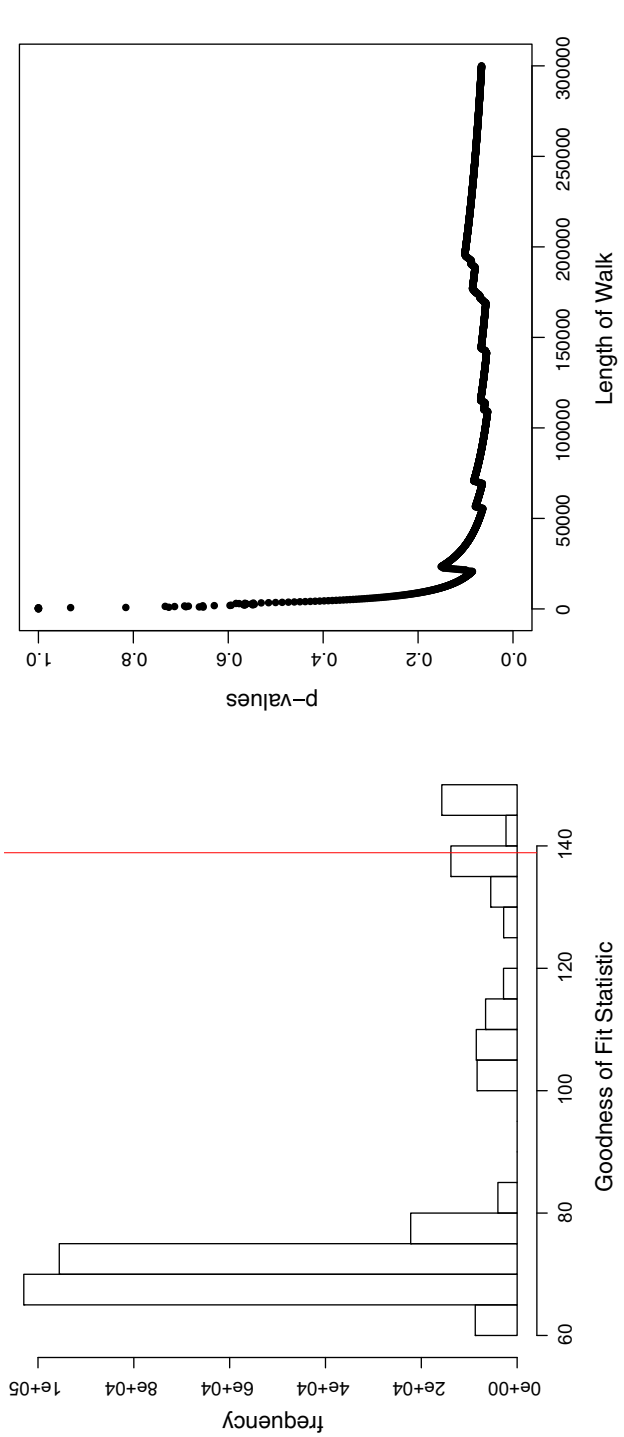

Fig. 11 Simulation results for the mobile money network data from Kushimba et al. (2013): chain of length 330,000 with 30,000 burn-in steps. **a** Histogram for sampling distribution of chi-square statistic, with indicated observed value. **b** Plot of p-value estimates from a typical run

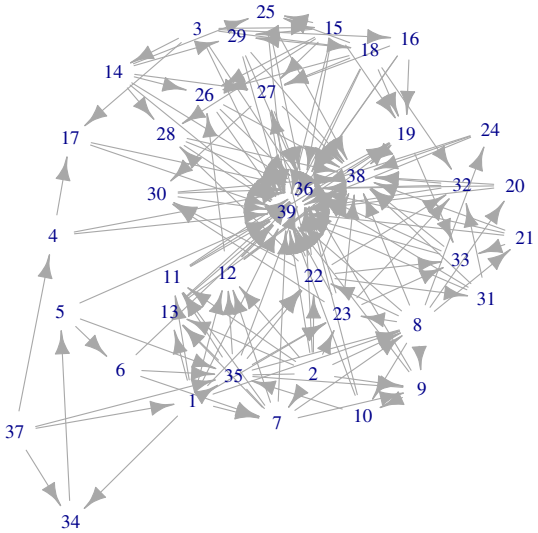

Fig. 12 The directed network representing the food web relationships in Chesapeake Bay data from Pajek (2004a)

4.5 Sampson's monastery study

Sampson (1968) conducted an ethnographical study of social interactions between novices in a New England monastery in the mid 1960s. Sampson observed 25 novices over a period of 2 years, gathering social relations data at four time points, and on multiple relationships. This has been a favorite example for analysis by sociologists, statisticians and others, and was used in original p_1 model studies. At the fourth time point ($T4$), there were 18 monks, and the social network had 54 directed edges representing the top three answers to the question "whom do you like" for each novice. We consider the directed graph in Fig. 13 representing the relationships derived from this affinity sociometric data. The list of edges in the graph was downloaded from Pajek (2004b).

Perhaps not unsurprisingly, the p_1-model with edge-dependent reciprocation seems to fit these data remarkably well. The chi-square statistic for the observed network is 404.7151, which is very close to the minimum chi-square statistic that was returned during a 1,000,000 step walk (see Fig. 14a). The estimated p-value for this data is 0.9863126. The random walk seems to be exploring the fiber broadly, discovering about 8800 new networks every 50,000 steps, though we do not know the exact size of the fiber.

5 Conclusion

The central motivation for this work is the scarcity of tools available to analyze directed networks. Other than heuristic methods, procedures for performing goodness-of-fit testing for directed network models have not existed. In the usual setting, the Metropolis–Hastings algorithm for sampling from conditional distributions requires

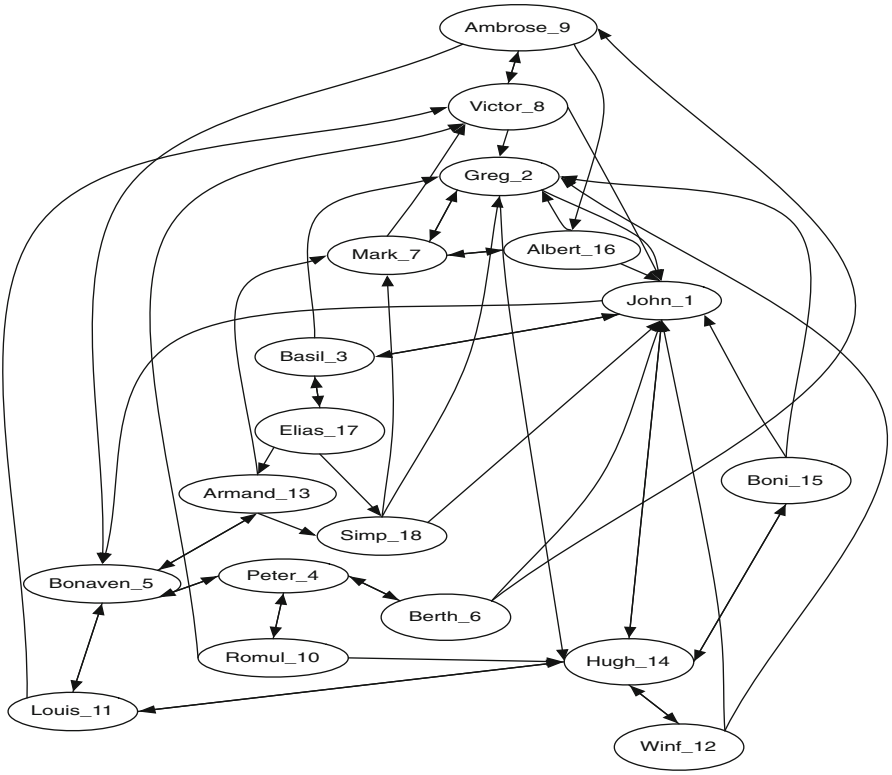

Fig. 13 Network derived from the monk dataset at time $T4$ in Sampson (1968) (Goldenberg et al. 2009, Figure 2.1)

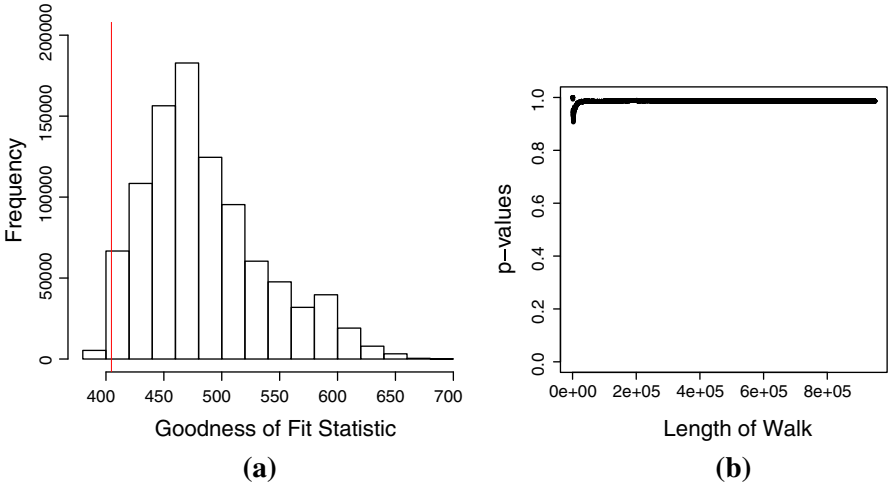

Fig. 14 Results of simulation with 1 million steps (50,000 burn-in steps) for Sampson's monastery data from time period $T4$. **a** Histogram of chi-square values from the simulation. **b** The p-value estimates

a Markov basis for a given model to be precomputed. By definition, however, Markov bases are data-independent, thus presenting a computational problem that becomes both wasteful and infeasible for network models on as few as 7 nodes. In addition, sampling constraints (e.g., one edge per dyad in a network or cell bounds in a contingency table) have presented problems for algebraic statistics as the restricted (observable) fibers cannot always be connected with a minimal set of Markov moves. Instead, a knowledge of a much larger set of moves, such as the Graver basis, is required for sampling. Since Graver bases are notoriously difficult to compute except for (notable) special cases (e.g., where a divide-and-conquer strategy applies, as in decomposable models), being able to dynamically generate one applicable move at a time is essentially the only hope for ever being able to utilize the algebraic statistics idea in practice.

Using the work by Dobra (2012) and by Ogawa et al. (2013) as our main motivation, we propose a methodology for dynamically generating moves and combinations of moves from the Graver basis (and thus a Markov basis) that guarantee to connect observable fibers for networks or contingency tables where sufficient statistics are not necessarily table marginals. This approach allows for a data-oriented algorithm, providing a dynamic exploration of any fiber without relying on an entire Markov basis. It produces only a relatively small subset of the moves—which could still be a large subset indeed—sufficient to connect the observable points in the fiber.

In contrast with previous approaches, our proposed modification uses moves that are constructed by understanding the balanced edge sets of the parameter hypergraph of the given model. Drawing upon the classical literature in combinatorial commutative algebra and recent work in algebraic statistics, we show how, in principle, one can construct applicable moves using the parameter hypergraph of any log-linear model and any observed network. Thus, the goodness-of-fit testing problem is translated into the problem of finding sub-hypergraphs of the parameter hypergraph with fixed degree sequences. Whereas this is a hard problem in general, it can be solved for specific models, which allows Algorithm 1 to be used for goodness-of-fit testing. As an example, we have described the entire procedure on the p_1 model with edge-dependent reciprocation. For the p_1 model, we (1) derive the structure of such the Markov moves in relation to the parameter hypergraph and (2) implement an algorithm to generate them dynamically. We hope this technique of analyzing the parameter hypergraph to construct dynamic Markov bases will be used for other log-linear models and spurs new ideas for goodness-of-fit testing for exponential random graph models in general.

Acknowledgements The authors are grateful to Alessandro Rinaldo and Stephen E. Fienberg for their support at the inception of this project. The authors would also like to thank two anonymous referees for their very thoughtful comments and suggestions which improved this manuscript.

References

Aoki, S., Takemura, A. (2003). Minimal basis for a connected Markov chain over $3 \times 3 \times k$ contingency tables with fixed two-dimensional marginals. *Australian & New Zealand Journal of Statistics*, *45*(2), 229–249.

Aoki, S., Takemura, A. (2005). Markov chain Monte Carlo exact tests for incomplete two-way contingency tables. *Journal of Statistical Computation and Simulation*, *75*(10), 787–812.

Aoki, S., Hara, H., Takemura, A. (2012). *Markov bases in algebraic statistics*. Springer Series in Statistics. New York: Springer.

Baird, D., Ulanowicz, R. (1989). The seasonal dynamics of the Chesapeake Bay ecosystem. *Ecological Monographs*, *59*, 329–364.

Bishop, Y. M., Fienberg, S. E., Holland, P. W. (1975). *Discrete multivariate analysis: Theory and practice*. New York: Springer.

Chatterjee, S., Diaconis, P., Sly, A. (2011). Random graphs with a given degree sequence. *Annals of Applied Probability*, *21*(4), 1400–1435.

Chen, Y., Dinwoodie, I. H., Sullivant, S. (2005). Sequential importance sampling for multiway tables. *Annals of Statistics*, *34*, 523–545.

Csardi, G., Nepusz, T. (2006). The igraph software package for complex network research. *International Journal of Complex Systems*, *1695*.

Develin, M., Sullivant, S. (2003). Markov bases of binary graph models. *Annals of Combinatorics*, *7*(4), 441–466.

Diaconis, P., Sturmfels, B. (1998). Algebraic algorithms for sampling from conditional distribution. *Annals of Statistics*, *26*(1), 363–397.

Dinwoodie, I. H., Chen, Y. (2011). Sampling large tables with constraints. *Statistica Sinica*, *21*, 1591–1609.

Dobra, A. (2003). Markov bases for decomposable graphical models. *Bernoulli*, *9*(6), 1093–1108.

Dobra, A. (2012). Dynamic Markov bases. *Journal of Computational and Graphical Statistics*, *21*(12), 496–517.

Dobra, A., Sullivant, S. (2004). A divide-and-conquer algorithm for generating Markov bases of multi-way tables. *Computational Statistics*, *19*, 347–366.

Dobra, A., Fienberg, S. E., Rinaldo, A., Slavković, A., Zhou, Y. (2008). Algebraic statistics and contingency table problems: Log-linear models, likelihood estimation and disclosure limitation. *Emerging applications of algebraic geometry* (pp. 63–88). IMA. Volumes in Mathematics and its Applications, vol. 149, New York: Springer Verlag.

Drton, M., Sturmfels, B., Sullivant, S. (2009). Lectures on algebraic statistics, Oberwolfach Seminars, vol 39. Springer, Basel. doi:10.1007/978-3-7643-8905-5.

Fienberg, S. E., Wasserman, S. S. (1981). Discussion of Holland, P. W. and Leinhardt, S. An exponential family of probability distributions for directed graphs. *Journal of the American Statistical Association*, *76*, 54–57 (1981).

Fienberg, S.E., Petrović, S., Rinaldo, A. (2010). Algebraic statistics for p_1 random graph models: Markov bases and their uses. *Looking Back. Proceedings of a Conference in Honor of Paul W. Holland*, chapter 1, *Lecture Notes in Statistics—Proceedings*, vol.202, New York: Springer.

Goldenberg, A., Zheng, A. X., Fienberg, S. E., Airoldi, E. M. (2009). A survey of statistical network models. *Foundations and Trends in Machine Learning*, *2*(2), 129–233.

Gross, E., Petrović, S. (2013). Combinatorial degree bound for toric ideals of hypergraphs. *International Journal of Algebra and Computation*, *23*(6), 1503–1520.

Gross, E., Petrović, S., Stasi, D. (2014). Goodness of fit for log-linear network models: supplementary material. http://math.iit.edu/~spetrov1/DynamicP1supplement/. Accessed 18 Mar 2016.

Haberman, S. J. (1981). An exponential family of probabilty distributions for directed graphs: Comment. *Journal of the American Statistical Association*, *76*(373), 60–61.

Hara, H., Takemura, A. (2010). Connecting tables with zero-one entries by a subset of a Markov basis. In M. Viana, H. Wynn (Eds.), *Algebraic methods in statistics and probability II, contemporarymathematics* (Vol. 516, pp. 199–213)., American Mathematical Society: Providence.

Hara, H., Takemura, A., Yoshida, R. (2009a). Markov bases for two-way subtable sum problems. *Journal of Pure and Applied Algebra*, *213*(8), 1507–1521.

Hara, H., Takemura, A., Yoshida, R. (2009b). A Markov basis for conditional test of common diagonal effect in quasi-independence model for square contingency tables. *Computational Statistics & Data Analysis*, *53*(4), 1006–1014.

Hara, H., Aoki, S., Takemura, A. (2010). Minimal and minimal invariant Markov bases of decomposable models for contingency tables. *Bernoulli*, *16*(1), 208–233.

Hara, H., Aoki, S., Takemura, A. (2012). Running Markov chain without Markov basis. In T. Hibi (Ed.), *Harmony of Gröbner bases and the modern industrial society*. Singapore: World Scientific.

Haws, D., Martin del Campo, A., Takemura, A., Yoshida, R. (2014). Markov degree of the three-state toric homogeneous Markov chain model. *Beiträge zur Algebra und Geometrie/Contributions to Algebra and Geometry*, *55*, 161–188.

Holland, P. W., Leinhardt, S. (1981). An exponential family of probability distributions for directed graphs (with discussion). *Journal of the American Statistical Association*, *76*(373), 33–65.

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

Král, D., Norine, S., Pangrác, O. (2010). Markov bases of binary graph models of K_4-minor free graphs. *Journal of Combinatorial Theory, Series A*, *117*(6), 759–765.

Kushimba, S., Chaggar, H., Gross, E., Kunyu, G. (2013). Social networks of mobey money in Kenya. In: *Working Paper 2013-1*, Institute for Money, Technology, and Financial Inclusion, Irvine.

Norén, P. (2015). The three-state toric homogeneous Markov chain model has Markov degree two. *Journal of Symbolic Computation*, *68*(2), 285–296.

Ogawa, M., Hara, H., Takemura, A. (2013). Graver basis for an undirected graph and its application to testing the beta model of random graphs. *Annals of Institute of Statistical Mathematics*, *65*(1), 191–212.

Pajek (2004a). Food webs. http://vlado.fmf.uni-lj.si/pub/networks/data/bio/foodweb/foodweb.htm. Accessed 18 Mar 2016.

Pajek (2004b). Sampson's monastery dataset. http://vlado.fmf.uni-lj.si/pub/networks/data/esna/sampson.htm. Accessed 18 Mar 2016.

Petrović, S., Stasi, D. (2014). Toric algebra of hypergraphs. *Journal of Algebraic Combinatorics*, *39*(1), 187–208.

Petrović, S., Rinaldo, A., Fienberg, S.E. (2010). Algebraic statistics for a directed random graph model with reciprocation. In: M. A. G. Viana, H. Wynn (Eds.), *Algebraic Methods in Statistics and Probability II, Contemporary Mathematics, vol. 516*, American Mathematical Society.

R DCT (2005). R: a language and environment for statistical computing. http://www.R--project.org. Accessed 18 Mar 2016.

Rapallo, F., Yoshida, R. (2010). Markov bases and subbases for bounded contingency tables. *Annals of the Institute of Statistical Mathematics*, *62*(4), 785–805.

Robert, C., Casella, G. (1999). *Monte Carlo statistical methods*. In: *Springer Texts in Statistics*. New York: Springer.

Sampson, S.F. (1968). A novitiate in a period of change: an experimental and case study of relationships. PhD thesis, Department of Sociology, Cornell: Cornell University.

Slavković, A. B. (2010). Partial information releases for confidential contingency table entries: Present and future research efforts. *Journal of Privacy and Confidentiality*, *1*(2).

Slavković, A. B., Zhu, X., Petrović, S. (2015). Fibers of multi-way contingency tables given conditionals: relation to marginals, cell bounds and markov bases. *Annals of the Institute of Statistical Mathematics*, *67*(4), 621–648.

Sturmfels, B. (1996). *Gröbner bases and convex polytopes*., University Lecture Series. Providence: American Mathematical Society.

Sturmfels, B., Welker, V. (2012). Commutative algebra of statistical ranking. *Journal of Algebra*, *361*, 264–286.

Villarreal, R. H. (2000). *Monomial algebras*., Monographs and Research Notes in Mathematics. Boca Raton: Chapman and Hall/CRC.

Yamaguchi, T., Ogawa, M., Takemura, A. (2013). Markov degree of the Birkhoff model. *Journal of Algebraic Combinatorics*, *38*(4), 1–19.

4ti2 T (2008) 4ti2: a software package for algebraic, geometric and combinatorial problems on linear spaces combinatorial problems on linear spaces. http://www.4ti2.de. Accessed 18 Mar 2016.