Statistica Sinica **17**(2007), 1371-1393

CONTINGENCY TABLES OF NETWORK TYPE: MODELS, MARKOV BASIS AND APPLICATIONS

Lawrence H. Cox

National Center for Health Statistics

Abstract: Contingency tables are a staple of quantitative science. Statistical science has paid continuing attention to developing theory, analytical methods, and models for contingency tables that in turn require theoretically verifiable, computationally efficient algorithms. Despite recent advances, there remain theoretical and computational obstacles to developing such algorithms, in some cases for tables with relatively few cells or in low dimensions. We define and investigate a class of multi-dimensional contingency tables — *tables of network type* — that overcome these limitations and enjoy strong theoretical properties and efficient computational algorithms. We demonstrate that tables in this class are abundant and familiar, including 2-dimensional tables, the Rasch model, log-linear models involving summation over mostly dichotomous variables, and tables of these types subject to structural zeroes. We describe ways to collapse non-network tables into network tables. We construct a Markov basis for tables of network type based on moves involving only coefficients -1, 0, $+1$. We provide theoretical models and efficient algorithms for solving three important statistical problems over tables of network type — sampling contingency tables subject to specified marginals, imputation and analysis for item and unit nonresponse subject to edit constraints, and obtaining exact bounds on entries in partially specified tables for purposes including statistical disclosure limitation. We relate our results to the De Loera-Onn formulation of all integer linear programs as slim 3-dimensional contingency tables.

Key words and phrases: Data confidentiality, exact bounds, imputation, integer optimization, log-linear model, Markov chain Monte Carlo, nonresponse, probability sampling.

1. Introduction

Contingency tables are a staple of quantitative science. Statistical science has paid continuing attention to developing analytical methods and models for contingency tables over several decades (Bishop, Fienberg and Holland (1975)). In turn, implementation of these methods requires theoretically verifiable, computationally efficient algorithms. As data providers such as national statistical offices seek to meet rapidly increasing and changing demands for data and for new and more flexible data formats and access modalities, attention has been given to the concept of a public use statistical data base query system (SDBQS)

to answer such count queries as those arising from a national census (60-80 categorical variables). A SDBQS, in essence, is defined over an extremely large, high-dimensional contingency table and consequently verifiable, efficient methods for statistical computing on tables of this magnitude and scope are needed (Karr, Dobra, Sanil and Fienberg (2002) and Cox (2004)). An important demand is to construct a (large) probability sample of tables conditional on a specified log-linear model, viz., a probability sample from the set of all contingency tables exhibiting the minimal sufficient statistics (MSS) of the log-linear model. A *partially specified contingency table* is the set of nonnegative integer solutions of the system of linear constraints defined by fixing the values of a set of MSS.

There is evidence that efficient methods may be out of reach for many contingency tables, at least in the near term. This paper defines and mathematically characterizes a class of contingency tables of practical use for which important statistical problems can be modeled theoretically and solved efficiently. These tables are based on mathematical networks (Nemhauser and Wolsey (1988, Part III), a class of linear optimization problems well-known in operations research. We refer to this class as *tables of network type*, identify important and familiar partially specified tables of network type, and illustrate the power of the characterization by solving three problems over tables of network type: sampling tables subject to specified MSS, imputation in tables for item and unit nonresponse, and obtaining exact bounds on entries in partially specified tables.

Early work on networks and statistics focused on statistical applications of networks for 2-dimensional tables: Cox and Ernst (1982), Kelly, Golden, Assad and Baker (1990) (controlled rounding), Causey, Cox and Ernst (1985) (2-way sample stratification), Cox (1987) (unbiased controlled rounding) and Cox (1995) (complementary cell suppression). Other work deals with network structure (or lack thereof) for specific table types: Cox and George (1989) investigate how network structure of 2-dimensional tables is affected by subtotal constraints; Cox (2003) studies failure of feasibility and integrality in general tables, and shows that thin k-dimensional tables subject to $(k-1)$-dimensional marginals are network. Here we present the first effort to identify tables of network type as a generic class to be studied, explore membership and properties of tables in this class, develop efficient solutions over tables in this class of important statistical problems that are not as easily solved over general tables (if at all), and relate this classification to recent theoretical research on tables in general.

Section 2 describes the three statistical problems and issues affecting theory and efficient solution over general contingency tables, and it presents preliminaries on mathematical networks. Section 3 characterizes a large class of important, familiar tables as network, and illustrates related classes that are not. In Section 4 we construct a Markov basis for contingency tables of network type involving

only $\{-1, 0, +1\}$-moves, and provide algorithms for efficiently solving the three problems over network tables. Our technique can avoid creating infeasible solutions, a theoretical improvement and speed up of the Diaconis-Sturmfels method. In Section 5, we examine network tables in the context of all tables, in two ways. We introduce techniques for collapsing non-network tables into network tables, and demonstrate how this can be used to obtain useful partial information for non-network tables. We relate network tables to the De Loera-Onn isomorphic mapping of all integer linear programs to slim 3-dimensional contingency tables, and provide a refined mapping that distinguishes network from non-network tables. Section 6 provides concluding comments.

2. Statistical Computing over Partially Specified Contingency Tables

2.1. Three problems in statistics

In this section, we discuss three statistical problems that in Section 4 are modeled and solved over tables of network type by means of network structure and algorithms. Absent network or other specialized structure, these problems in general require combinatorial optimization, are NP-hard, and often cannot be completely solved.

First Problem: Draw a probability sample from a partially specified contingency table

The first problem is drawing a probability sample from a partially specified contingency table of network type, where each feasible table has nonzero probability of selection. This is the subject of a seminal paper of Diaconis and Sturmfels (1998) that illustrates the elegance and power of algebraic statistics, specifically *Gröbner bases* (Sturmfels (1996)). At that time, computational obstacles to computing the mathematical objects that define the underlying Markov chain were formidable for tables as small as $3 \times 3 \times 3$. Meaningful, continuing progress has been made, particularly in low dimensions, but high dimensional and large tables remain challenging. Our approach is related not to dimension but structure.

We now outline the Diaconis-Sturmfels method, referring the reader to the original paper for details, to Lang (1965, Chap. II) for algebraic preliminaries, to Sturmfels (1996, Chap. 2) for Gröbner bases, and to Pistone, Riccomagno and Wynn (2001, Chap. 2) for application of rings, ideals and Gröbner bases to statistics.

A partially specified contingency table is the set of nonnegative integer solutions to a system of integer linear equations $An = b$, $n \geq 0$, where A is a coefficient matrix defined by linear constraints imposed by the MSS, and b denotes fixed integer values of the MSS. Given a nonnegative integer solution $n^{(0)}$, all other solutions are of the form $n^{(i)} = n^{(0)} + k^{(i)}$ with $k^{(i)}$ integer, $Ak^{(i)} = 0$ and

$k^{(i)} \geq -n^{(0)}$. $Ker_Z(A, n^{(0)})$, the *integer kernel of A at* $n^{(0)}$, is the set of all such $k^{(i)}$, also called the set of *feasible moves from* $n^{(0)}$. The set $Ker_Z(A, n^{(0)})$ and the set of all feasible moves $\bigcup_n Ker_Z(A, n)$ are subsets of $Ker_Z(A) = \{k : Ak = 0\}$. A *chain* of solutions $\{n^{(0)}, n^{(1)}, \ldots, n^{(i-1)}, n^{(i)}, \ldots\}$ can be constructed via successive addition of feasible moves.

As each integer feasible solution has nonzero probability of selection, a procedure for constructing a probability sample must be capable of constructing any nonnegative integer solution $n^{(i)}$ of $An = b$. This can be accomplished if a Markov basis for the set of all feasible moves is available.

Definition 1. A *Markov basis* $M(A)$ for the set of all feasible moves is a subset of $Ker_z(A)$ with the property that, given any two nonnegative integer solutions $n^{(i)}$, $n^{(k)}$ of $An = b$, there exists a sequence of elements $f_j^{(i)} \in M(A)$ such that

$$n^{(k)} = n^{(i)} + \sum_{j=1}^{j(i)} \varepsilon_j f_j^{(i)} \quad \text{with} \quad \varepsilon_j = \pm 1 \quad \text{and}$$

$$n^{(i)} + \sum_{j=1}^{l} \varepsilon_j f_j^{(i)} \quad \text{is a feasible solution of} \quad \mathbf{An} = \mathbf{b}, \quad 1 \leq l \leq j(i).$$

If all coordinates of $f^{(j)}$ equal -1, 0, $+1$, $f^{(j)}$ is a $\{-1, 0, +1\}$-*move*.

Diaconis and Sturmfels (1998) associate a Markov basis to elements of a reduced Gröbner basis for a polynomial ideal (see Sturmfels (1996, Chap. 1 and 4)). The Markov basis is used with a hypergeometric stationary distribution $\pi(n)$ for the Markov chain in a Metropolis step to generate random samples. This elegant method is dependent on the computability of a Gröbner basis. The general algorithm for constructing Gröbner bases — the Buchberger algorithm — is doubly exponential (Sturmfels (1996)), but research on algorithms for tables is ongoing and algorithms at worst exponential are available (e.g., Hemmecke and Malkin (2006)). In Section 4, we provide a theoretically verified, computationally efficient algorithm for constructing a Markov chain of $\{-1, 0, +1\}$-moves for tables of network type. In addition, we demonstrate how to eliminate the feasibility check in the Diaconis-Sturmfels algorithm.

Second Problem: Imputation and analysis for an incomplete contingency table

The second set of problems is in the realm of imputation and analysis of incomplete contingency tables. A 2-dimensional example illustrates the problem. A sample of n_{++} respondents is asked two questions. The first question is categorized in r categories, the second in c categories. m_{ij} responses are recorded for category (i, j), corresponding to $m_{++} = \sum_{i,j} m_{ij}$ complete responses. In

addition, $m_{+,c+1} = \sum_{i=1}^{r} m_{i,c+1}$ respondents answer only the first question, with $m_{i,c+1}$ responses in first question category i, and $m_{r+1,+} = \sum_{j=1}^{c} m_{r+1,j}$ respondents answer only the second question, with $m_{r+1,j}$ responses in second question category j (*item nonresponse*), and $m_{r+1,c+1}$ answer neither question (*unit nonresponse*). As discussed in Section 4.3, several statistical problems including nonresponse problems can be examined in this framework.

Third Problem: Exact bounds for entries of a partially specified contingency table

The third problem is that of obtaining exact bounds for entries of a partially specified contingency table of network type. Data confidentiality provides one motivation for this problem. Small counts — defined by a specific numeric threshold — and complements of small counts represent unacceptable disclosure risk and cannot be released or inferred. One solution is to release selected marginal totals in lieu of counts. The released marginals must of course lie above the threshold, but this is not sufficient to ensure that an attacker could not infer that a value is below threshold. Prior to data release, the releaser must compute exact bounds on selected table entries (Cox (2002)). As discussed in Section 4.4, recent theoretical results further complicate this problem.

2.2. Networks

Two sets of marginal totals are *consistent* if they produce the same values for all shared lower dimensional marginal totals, including the grand total. Consider the familiar problem of obtaining maximum likelihood estimates of entries n_{ij} in a 2-way table conditional on specified row and column totals n_{i+}, n_{+j}. If the marginals are consistent, viz., a unique grand total n_{++} exists, then MLE exist for each n_{ij}, namely $n_{i+}n_{+j}/n_{++}$.

This property should not be taken for granted: three consistent sets of 2-dimensional marginals $n_{ij+} = \begin{pmatrix} 1 & 0 \\ 1 & 0 \end{pmatrix}$, $n_{i+k} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$, $n_{+jk} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$, for a partially specified $2 \times 2 \times 2$ 3-dimensional contingency table subject to a no 2-factor effects log-linear model fail to define even one table of nonnegative values, and consequently no integer tables or MLE exist (Cox (2003, p.255, Example 1(a))).

The MLE $n_{i+}n_{+j}/n_{++}$ for the 2-dimensional problem are not necessarily integers. In applications such as 2-way stratified sampling, it is desirable to replace the MLE by nearby integer values, if such exist (Cox (1987)). A second property of 2-dimensional tables is that consistent integer marginals assure the existence of at least one *integer solution* (Cox (2003)); so, integer estimates of the MLE do exist.

These and other strong properties of 2-dimensional tables hold because a 2-dimensional table can be modeled as a network.

Definition 2. A *directed graph* is a pair (V, E) where V is a set of objects called *nodes* and E is a subset of $V \times V$ of objects called *directed arcs*. If directed arc $(u, v) \in E$, u is the *from-node* of (u, v), v is the *to-node*, and (u, v) is *directed from u to v*.

We are interested only in directed graphs without *loops*–arcs (u, u) are not permitted.

Definition 3. A *network* $Ax = b$, $x \geq 0$ is a system of linear equations in nonnegative variables $\mathbf{x}$ defined over a directed graph without loops, with the following properties:
- network equations (rows of $\mathbf{A}$) and graph nodes correspond one-to-one
- network variables x_i and graph arcs (u, v) correspond one-to-one
- a network variable has coefficient -1, 0 or $+1$ in any equation.

Thus, each variable appears in at most two equations (one may assume precisely two). Each network equation e is expressible: $\sum_{i \in I_e} x_i = \sum_{j \in O_e} x_j + b_e$, where I_e is the set of to-arcs for node e, O_e is the set of its from-arcs, and b_e is its *node requirement*, a constant that can be positive, negative or zero. Instantiations (values) of variables are *flows*. Arc flows may be *capacitated*, viz., restricted by lower and/or upper bounds. Flows are nonnegative. Terminology equations/nodes and variables/arcs is used interchangeably.

If in addition the network satisfies the requirement that the set of equations can be partitioned into two disjoint subsets such that: for any variable, one of its equations is in the first subset, the other in the second, then the network is *bipartite*. In Section 3, we will see that our tables correspond to bipartite networks. If we represent the network by selecting $\mathbf{A}$ to be the node-arc incidence matrix of the directed graph, it easy to show that $\mathbf{A}$ is *totally unimodular*, viz., every square submatrix of $\mathbf{A}$ has determinant -1, 0 or $+1$.

Adjoining a linear optimization $\min\{c(x)\}$ to the network defines a linear program (LP): $\min\{c(x)\}$ subject to: $Ax = b$, $x \geq 0$. Total unimodularity of the network matrix $\mathbf{A}$ is a very strong property fundamental to integer optimization, as follows.

The *simplex algorithm* solves a linear program in two stages: identification of an extreme point of the LP polyhedron (*feasible region*), and a steepest descent algorithm for moving from one extreme point to a neighboring extreme point exhibiting equal or lesser value of the objective. The simplex transforms $\mathbf{A}$ to row-reduced echelon form, viz., row reduction followed by an invertible matrix $\mathbf{B}$ to achieve: $(I, N)x = Bb$ with $\mathbf{I}$ an identity matrix. Each $\mathbf{B}$ corresponds to an

extreme point: x-coordinates corresponding to rows of $\mathbf{I}$ (the *basic variables*) are set equal to the corresponding values of $\mathbf{Bb}$ and *nonbasic variables*, corresponding to columns of $\mathbf{N}$, are set to zero.

Total unimodularity of $\mathbf{A}$ assures that $\mathbf{B}$ is integer, ensuring that the basic coordinates $(Bb)_i$ of all polyhedral extreme points are integer. Consequently, continuous linear programming can be used to solve discrete combinatorial integer programming problems over networks. Algorithms for network optimization are cubic and, in some cases, quadratic in the inputs (Kennington and Helgason (1980) and Garey and Johnson (1979, Sec. A2.4)). We present two standard, important network theorems without proof.

Theorem 1. *A capacitated network is a network.*

Theorem 2. *The extreme points of a network with integer node requirements and integer arc capacities are integer.*

Two-dimensional tables are network, viz., the row and column totals comprise the bipartite nodes. By contrast, in three dimensions the coefficient matrix $\mathbf{A}$ of the set of $3 \times 3 \times 3$ tables partially specified by MSS $(n_{ij+}) = (n_{i+k}) = \begin{pmatrix} 1 & 1 & 0 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}$, $(n_{+jk}) = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 0 \end{pmatrix}$ exhibits submatrices of determinant ± 2, and noninteger extreme points, see Example 1.

$$n_{ij1} = \begin{pmatrix} 0.5 & 0.5 & 0 \\ 0 & 0.5 & 0.5 \\ 0.5 & 0 & 0.5 \end{pmatrix}, \quad n_{ij2} = \begin{pmatrix} 0.5 & 0.5 & 0 \\ 0.5 & 0 & 0.5 \\ 0 & 0.5 & 0.5 \end{pmatrix}, \quad n_{ij3} = \begin{pmatrix} 0 & 0 & 0 \\ 0.5 & 0.5 & 0 \\ 0.5 & 0.5 & 0 \end{pmatrix}.$$

Example 1. Noninteger Extreme Point for 3-Dim.Table Subject to 2-Dim. Marginals

3. Contingency Tables of Network Type

We describe important, familiar classes of partially specified contingency tables that are of network type, and related tables that are not. In Section 5.1, we describe approaches to collapsing non-network tables to network tables so as to yield partial information for non-network tables that may otherwise be unavailable.

Two-dimensional tables subject to all 1-dimensional marginals (row and column totals), or to a unique 0-dimensional marginal (grand total), are of network type. Two-dimensional tables subject in addition to row or column subtotal constraints are of network type but, if subjected to both row and column subtotals,

are not (Cox and George (1989)). By virtue of Example 1, partially specified 3-dimensional tables as small as $3 \times 3 \times 3$ subject to MSS = all 2-dimensional marginals (*no* 3-*factor effects* log-linear model) are not network. Example 2, $(n_{i++}) = (n_{+j+}) = (n_{++k}) = (1, 1)$, shows that partially specified 3-dimensional tables as small as $2 \times 2 \times 2$ subject to MSS = all 1-dimensional marginals (*complete independence* model) are not network.

$$n_{ij1} = \begin{pmatrix} 0 & 0.5 \\ 0.5 & 0 \end{pmatrix}, \quad n_{ij2} = \begin{pmatrix} 0.5 & 0 \\ 0 & 0.5 \end{pmatrix}.$$

Example 2. Non-integer Extreme Point for $2 \times 2 \times 2$ 3-Dim. Complete Independence Model.

We present two theorems specifying a large class of important, familiar, partially specified contingency tables that are of network type.

Theorem 3. *If a partially specified contingency table is defined by at most two configurations of MSS, then it is of network type.*

Proof. If there is only one configuration of MSS, for purposes of proof (only) assume there are two copies of it, so that we need only consider the case of precisely two MSS.

Each MSS is defined by separating the variables into two classes: *summation variables* over which the MSS perform summation; and *index variables* that label the MSS. For example in $\{n_{ij+}\}$, i and j are the index variables and k is the summation variable. We define network nodes in terms of the index variables and network arcs in terms of the summation variables. Reference to a specific example will be helpful: the 3-dimensional two two-factor effects absent model, MSS = $\{n_{ij+}\}$, $\{n_{++k}\}$, of size $I \times J \times K$.

Comments marked $(*)$ refer to the example.

Create a bipartite network:

- create a left node set corresponding to index variables of the first MSS
  - $*$ IJ nodes labeled (i, j)
- assign corresponding left node requirements
  - $*$ left node requirements: $-n_{ij+}$
- create a right node set corresponding to index variables of second MSS
  - $*$ K nodes labeled (k)
- assign corresponding right node requirements
  - $*$ right node requirements: n_{++k}
- construct arcs y connecting corresponding left and right nodes
  - $*$ arc y_{ijk} connects left node (i, j) to right node (k).

This construction completes the proof.

Theorem 4. *Ignore all MSS that sum over only a single dimension of size two* (*a* thin *dimension*). *If the remaining partially specified table is of network type, then so is the original partially specified table.*

Proof. Cox (2003, Sec. 5) proved this for the no k-factor effects model over partially specified k-dimensional tables of size $b \times c \times 2^{k-2}$ (*thin tables*). The general proof follows.

Ignore MSS that sum over only a single thin dimension; assume there are p of these. Let I index all the non-excluded dimensions, and $s(I)$ denote the product of the sizes of these dimensions. None of these MSS can sum over the excluded thin dimensions, otherwise the minimality of the set of MSS configurations would be violated. The non-excluded MSS restricted to the $(k-p)$-dimensional space specified by I define a network N, by assumption.

Select one of the p ignored MSS that sums over only a thin dimension; denote this dimension l. Re-incorporating dimension l into the model defined by the non-excluded MSS results in two components: one copy of N each for $l = 1$ and $l = 2$, with arcs denoted $y_{I,l=1}$, $y_{I,l=2}$, respectively. The theorem is proved if we can show how to reincorporate the summation over dimension l into the two component network while preserving network structure, because then we can apply the construction recursively (p times) to recover all excluded variables and MSS. The recursive step of the network construction is as follows.

Define $s(I)$ new nodes, each with node requirement $-n_{I,+}$. These are from-nodes, with two arcs emanating from each: $y_{I,l=1}$, $y_{I,l=2}$ — each arc taken from the corresponding copy of N. The first arc of the pair ($l = 1$) connects to the to-node of $y_{I,l=1}$ in the first copy, and that node maintains its node requirement. The second arc of the pair ($l = 2$) connects to the from-node for $y_{I,l=2}$ in the second network, and this node is assigned a new node requirement equal to the sum of $n_{I,+}$ over all arcs now connected to the node minus the original requirement of the node. The remaining nodes — the from-nodes of the first network copy and the to-nodes of the second — represent equations that are linearly dependent on equations represented by the new network, and may be ignored.

An example illustrates the proof. Consider a $3 \times 3 \times 2$ table subject to MSS = $\{n_{ij+}\}$, $\{n_{i+k}\}$, $\{n_{+jk}\}$ (no 3-factor effects model over a thin table). Ignoring the (thin) first MSS, we obtain two identical bipartite networks based on connecting the first and second elements of MSS = $\{n_{+j1}, n_{+j2}\}$, $\{n_{i+1}, n_{i+2}\}$ by variables y_{ij1}, y_{ij2}. I enumerates (i, j) and $s(I) = s(i)s(j)$. The final network has left node set enumerated by $I = (i, j)$ with node requirements $-n_{ij+}$. Its right node set involves $s(i) + s(j)$ nodes. The first $s(i)$ right nodes have node requirement n_{i+1} and are connected to the left node set by arcs y_{ij1}; the second $s(j)$ have

node requirement $\sum_j n_{ij+} - n_{+j2}$ and are connected to the left node set by arcs y_{ij2}. Redundancy of the remaining nodes is demonstrated via relationships among the node requirements: $n_{i+2} = n_{i++} - n_{i+1} = \sum_j n_{ij+} - n_{i+1}$ and $n_{+j1} = n_{+j+} - n_{+j2} = \sum_i n_{ij+} - n_{+j2}$.

In summary, familiar tables that are of network type include:
- 2-dimensional tables
- 2-dimensional tables subject to row or column subtotal constraints; in particular, defined by a hierarchical variable along (only) one of its dimensions
- no k-factor effects models over thin k-dimensional tables, e.g., over dichotomous tables
- 3-dimensional one 2-factor effect and two 2-factor effects absent models
- log-linear models with precisely two configurations of MSS
- multiple, algebraically independent copies of any of the above
- any of the above subject to feasible structural zeroes.

Note that there are no restrictions on the dimension of tables of network type or, excepting thin tables, on their size.

Tables of network type provide analysts with computational efficiency and a firm theoretical foundation. Tables of network type provide table designers and analysts with a new set of options, sometimes the only option, as discussed in Section 5.

4. A Markov Basis for Tables of Network Type and Solutions for the Three Statistical Problems

4.1. Markov basis

We represent $Ker_Z(A) = \{n : An = 0\}$ as a network, as follows. Each row of **A** corresponds to an equation involving coefficients -1, 0, $+1$, and each variable occurs in precisely two equations. Define one node for each equation. Define two arcs corresponding to each variable x and the pair of equations g, h in which it occurs: the first arc, the *positive arc*, denoted y^+ goes from g to h; the second arc, the negative arc, denoted y^-, goes from h to g; each arc is referred to as the *opposite arc* of the other. Each node has node requirement equal to 0; arc flows are, as usual, nonnegative. This network, the *kernel network*, corresponds to the system of equations $A(y^+ - y^-) = 0$, and maps many-to-one onto the kernel of **A**. We restrict the kernel network to the unit hypercube by imposing capacity constraints $0 \leq y^+,\ y^- \leq 1$, which by Theorems 1 and 2 preserve network structure and integrality of extreme points. Call the restriction the *restricted kernel network*. Corresponding to restricted network extreme points $\{y_s^+, y_s^-\}$ are restricted network *extreme point moves* $y = \{y_s\} = \{y_s^+ - y_s^-\}$. The following is clear.

Lemma 1. *Extreme point moves* $\mathbf{y}$ *of the restricted kernel network correspond one-to-one with* $\{-1, 0, +1\}$*-moves of* $Ker_Z(A)$*.*

Theorem 5. *A Markov basis for* $Ker_Z(A)$ *based on* $\{-1, 0, +1\}$*-moves can be computed from the set of restricted kernel network extreme point moves.*

Proof. Select nonzero $k \in Ker_Z(A)$. Define a sub-network of the restricted kernel network as follows. For each variable appearing in k with nonzero coefficient: if the coefficient is positive, include its positive arc y^+ in the sub-network; if negative, include its negative arc y^-. For each selected arc, include the two nodes connected by the arc. Capacitate all sub-network arcs between 0 and 1. Each sub-network node is incident to at least one from-arc and at least one to-arc. Pick any sub-network node and one of its to-arcs. This arc connects to a second node distinct from the first (no loops). Pick a to-arc of the second node. This arc connects to a third node distinct from the second but also distinct from the first (sub-network contains no pair of opposite arcs). Continue in this manner. This sequence of arcs eventually enters a node exited previously, thereby defining a nontrivial circuit of arcs. A maximum flow of 1 unit can be instantiated along this circuit. The maximum flow defines a restricted network extreme point, which in turn defines an extreme point move (coordinates -1, 0, $+1$), and in turn a $\{-1, 0, +1\}$-move f. Replace k by $k - f$ (also in the kernel) and repeat the steps of this paragraph. The reduced k eventually becomes 0, resulting in a Markov basis decomposition of the original k into $\{-1, 0, +1\}$-moves $k = \sum_{j=1}^{w} \varepsilon_j f^{(j)}$.

4.2. Sampling contingency tables subject to specified marginals

The Diaconis and Sturmfels (1998) method constructs an aperiodic, irreducible, connected Markov chain by MCMC sampling from a reduced Gröbner basis for a polynomial ideal. This is a sophisticated and excellent methodology. Two aspects of the method are important here. First, if the Gröbner basis is unavailable, the sample cannot be created. Second, at each iteration the proposed solution must be examined for feasibility, increasing computation. Our contributions are as follows. First, we offer an alternative theory and a computationally efficient algorithm for constructing the Markov chain for any table of network type. Second, we are able to avoid infeasibility.

The original Diaconis-Sturmfels algorithm is as follows:

- *Initialize*: Generate an integer solution $n^{(0)}$ of $An = b$, $n \geq 0$
- Set: $i = 1$
- Count: If $i >$imax, QUIT
- *Select*: Uniformly random selection of a member $g^{(i)}$ of the reduced Gröbner basis
- Compute: $n^{(i)} = n^{(i-1)} + g^{(i)}$

- *Feasibility*: If $n^{(i)}$ is infeasible, return to Select
- *Metropolis Step*: Move to $n^{(i)}$ with probability $\min\{1, \pi(n^{(i)})/\pi(n^{(i-1)})\}$
- Increment: i to $i+1$
- GOTO: Count.

After sufficiently many initial iterations, a sample is constructed by including either $n^{(i)}$ or $n^{(i-1)}$ in the sample, based on the Metropolis step. The hypergeometric stationary distribution of the Markov chain is denoted above by $\pi(n)$. These probabilities involve an unknown normalizing constant and are not computable but as the "numerators" are available the ratio of two probabilities (Metropolis step) is computable.

For tables of network type, we replace Select with our *Proposal Algorithm*:

- For each pair of network arcs y_s^+, y_s^- and $M >> 0$, set $d(y_s^+) = -M, 0, M$ with uniformly random probabilities $= 1/3$.
  Set $d(y_s^-) = -d(y_s^+)$, and refer to this function as $d(y) = d(y^+ - y^-)$
- Set the network cost function $c(y_s^+, y_s^-) = d(y) + \sum_s (y_s^+ + y_s^-)$
- Return the network to its Start position (see below)
- Create a $\{-1, 0, +1\}$-move $g^{(i)}$ by minimizing $c(y_s^+, y_s^-)$ over the network.

Lemma 2. *If the Proposal Algorithm yields extreme point move $y^{(0)}$ from network minimization of randomly generated cost function $c(y^+, y^-) = d(y) + \sum_s (y_s^+ + y_s^-)$, then it will yield $-y^{(0)}$ from network minimization of $c^*(y^+, y^-) = -d(y) + \sum_s (y_s^+ + y_s^-)$.*

Proof. Minimization of $c(y_s^+, y_s^-)$ over the network yields a solution that (1) minimizes $d(y)$ over the network and (2) comprises the fewest possible nonzero arcs among all $d(y)$-minimizing solutions. The network constraints are: $Ay = (A, -A)\binom{y^+}{y^-} = 0$, $0 \le y^+, y^- \le 1$. We refer to this as the *Start position* for the network. Now, $\arg\min\{-d(y)\} = \arg\min\{d(-y)\} = \arg\min\{d(y^- - y^+)\}$, so the optimization

$\min\{-d(y) : (A, -A)\binom{y^+}{y^-} = 0,\ 0 \le y^+,\ y^- \le 1\}$ is equivalent to

$\min\{d(y) : (A, -A)\binom{y^-}{y^+} = 0,\ 0 \le y^+,\ y^- \le 1\}$. Consequently,

$\min\{c^*(y_s^+, y_s^-) : (A, -A)\binom{y^+}{y^-} = 0,\ 0 \le y^+,\ y^- \le 1\}$ is equivalent to

$\min\{c(y_s^+, y_s^-) : (A, -A)\binom{y^-}{y^+} = 0,\ 0 \le y^+, y^- \le 1\}$. Network computations are performed on the coefficient matrix $(\mathbf{A}, -\mathbf{A})$, the cost vector $\mathbf{c}$, and the right-hand side (which is $\mathbf{0}$), and not on the column vector of variables. As the network initializes at the Start position for each proposal, as the capacity constraints are symmetric, and as network computations are deterministic, then the sequences of computations for the two optimizations $c(y_s^+, y_s^-)$ and $c^*(y_s^+, y_s^-)$ are identical and yield two solutions which are negative to each other, viz., solutions in which y^+, y^- are reversed.

Computing the initial solution, which can be difficult for some problems, is trivial on a network: simply optimize any convenient objective function over the original network. If in addition we capacitate to zero all arcs y_s^- in the restricted kernel network corresponding to variables that are zero in $n^{(i-1)}$, then no coordinate of $n^{(i)}$ can be negative and consequently $n^{(i)}$ is feasible, eliminating the Feasibility check.

This methodology may be applied with great advantage to other statistical problems, particularly smaller problems not requiring massive iteration. For example, Buena and Besag (2000) are concerned with $I \times J \times K$ tables subject to MSS $= \{n_{ij+}\}$, $\{n_{i+k}\}$, $\{n_{+jk}\}$, and with computing solutions to the $I \times J \times 2$ Rasch model. They present a specialized method for moving between solutions in $I \times J \times 2$ tables that may move outside the feasible region, but at the next move returns inside. The network method supercedes this approach. Given the $3 \times 3 \times 2$ Rasch model of Example 3 (left), Buena and Besag (2000) seek to move to all other solutions. This is easily done using the network, which reveals that the only integer feasible move from Example 3 (left) is Example 3 (center), resulting in a second, and only other, Rasch model — Example 3 (right).

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} + \begin{pmatrix} -1 & 1 & 0 \\ 1 & 0 & -1 \\ 0 & -1 & 1 \end{pmatrix} \begin{pmatrix} 1 & -1 & 0 \\ -1 & 0 & 1 \\ 0 & 1 & -1 \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}.$$

Example 3. Rasch model of Buena and Besag (2000).

It remains to verify two important conditions for our revision of the Diaconis-Sturmfels algorithm. Namely, that (a) the Proposal Algorithm-which replaces a uniformly random Select step–meets the conditions for Metropolis-Hastings, and, (b) the Proposal Algorithm is capable of constructing any member of the Markov basis with nonzero probability. Both issues are addressed in terms of the relationship between the randomly generated cost function and extreme point moves (Markov basic elements).

Our randomly generated cost functions correspond many-to-many with extreme point moves of the kernel network: each of our cost functions is minimized at an extreme point move but possibly more than one, and, conversely, each extreme point move may minimize multiple cost functions. A *minimal extreme point move* is a move whose set of nonzero coordinates does not strictly contain that of another. For (b), it suffices to show that the Proposal Algorithm can generate any minimal extreme point move $y^{(0)}$ with nonzero probability. This holds if each $y^{(0)}$ uniquely minimizes a cost function $c(y_s^+, y_s^-)$ of the specified type. That cost function is given by: for each coordinate s with $y_s^{(0)} = 1$, set

$d(y_s^+) = -M$, $d(y_s^-) = M$; if $y_s^{(0)} = -1$, set $d(y_s^+) = M$, $d(y_s^-) = -M$; otherwise, $d(y_s^+) = d(y_s^-) = 0$. Thus, (b) holds.

Regarding (a), the issue is whether the Proposal Algorithm for generating *proposals* (potential sample units) for the Markov chain, combined with the Diaconis-Sturmfels Metropolis step, assures that the chain converges to $\pi(n)$. Diaconis-Sturmfels employ uniformly random probabilities. Proposal Algorithm cost functions are selected in a uniformly random manner, but do not correspond one-to-one with extreme point moves. Computation of move probabilities is complex, and affected by factors such as order of the columns of $\mathbf{A}$. Rather than attempt to refine the Proposal Algorithm to achieve uniformly random selection, instead we demonstrate that the proposal density $q(n^{(i+1)}; n^{(i)})$ for the Proposal Algorithm is *symmetric*, which is sufficient for convergence of the Markov chain to the limiting distribution $\pi(n)$. Thus, (a) must hold.

Theorem 6. *The proposal density $q(n^{(i+1)}; n^{(i)})$ for the Proposal Algorithm is symmetric.*

Proof. Given two tables $n^{(i)}$ and $n^{(i+1)}$, if there exists a member of the Markov basis $g(i)$ such that $n^{(i+1)} = n^{(i)} + g^{(i)}$, then $n^{(i)} = n^{(i+1)} + (-g^{(i)})$. Based on Lemma 2, under any conditions for which the Proposal Algorithm creates the proposal move $g(i)$ to $n^{(i+1)}$ from $n^{(i)}$ based on selecting cost function $c(y_s^+, y_s^-)$, that is, $g^{(i)} = \arg\min\{c(y^+, y^-)\}$, it will create the proposal move $-g(i)$ to $n^{(i)}$ from $n^{(i+1)}$ based on selecting the equi-probable cost function $c^*(y)$, viz., $-g^{(i)} = \arg\min\{c^*(y^+, y^-)\} = \arg\min\{c(y^-, y^+)\}$. Thus, $q(n^{(i+1)}; n^{(i)}) = q(n^{(i)}; n^{(i+1)})$ and the proposal density is symmetric.

Whenever the coordinates of zero entries of $n^{(i+1)}$ and $n^{(i)}$ differ, if we attempt to enforce feasibility by setting the corresponding negative arc capacities to zero, the networks over which the two optimizations are performed will be different. This could result in different sequences of computations and affect proposal symmetry. Consequently, in practice it is necessary to choose between avoiding infeasible moves and assuring an everywhere symmetric proposal density. This problem does not arise if all upper capacities remain at 1 (e.g., when neither table contains zeroes).

4.3. Adjusting incomplete information for nonresponse

An important problem is to estimate complete count data for a 2-dimensional table given observed partial counts and marginals. Incomplete data methods based on fixed marginals such as iterative proportional fitting break down for this problem because, with the exception of the grand total n_{++}, marginal totals are not fixed. Similarly, ratio estimation, viz., estimating cell counts n_{ij} for the complete table by $m_{ij}(n_{++}/m_{++})$, can produce estimates that are less than

observed partial marginals $m_{i+} = \sum_{j=1}^{c+1} m_{i,c+1}$, $m_{+j} = \sum_{i=1}^{r+1} m_{r+1,j}$, possibly to a degree greater than can be accounted for by measurement error. Similarly, as attempted by Greene, Smith, Levenson, Hiser and Mah (2001), applying ratio estimation to the partial marginals followed by iterative proportional fitting based on these estimates also can result in estimates significantly less than observed partial counts. Each of these approaches fails to completely condition estimates of complete data on observed data. The EM algorithm overcomes these limitations (Little and Rubin (2002, Chap. 13)), and may be applied to this problem.

This problem and other problems can be addressed if a probability sample of integer feasible solutions is available. A framework for estimation and analysis based on the sample size and inequality constraints imposed by observed counts is required. The constraint system for the 2-dimensional problem is network over $ij + i + j$ variables:

$$\sum_{j=1}^{c} n_{ij} = n_{i+}, \quad \sum_{i=1}^{r} n_{ij} = n_{+j}, \quad \sum_{i=1}^{r} n_{i+} = \sum_{j=1}^{c} n_{+j} = n_{++},$$

$$i=1,\ldots,r; \quad j=1,\ldots,c: \quad n_{ij} \in Z, \; n_{ij} \geq 0, \quad n_{ij} \geq m_{ij}, \quad n_{i+} \geq m_{i+}, \quad n_{+j} \geq m_{+j}.$$

The restricted kernel network (Section 4.1) is constructed from this network. Several statistical applications can be addressed by this model and the method of Section 4.2.

Based on the restricted kernel network and Section 4.2, generate a random sample of tables from this model. Each sample table k has an associated hypergeometric probability $\pi(k) = d_k/N$ involving an unknown normalizing constant N. Only the "numerator" d_k is used in the Metropolis step. Numerators are computed from the rc probabilities that a randomly selected sample individual falls in table cell (i, j). For example, if item nonresponse is attributed to a missing completely at random assumption in each row and column, and similarly if unit nonresponse is completely at random, then the probability that a randomly selected individual falls in cell (i, j) is $(rcm_{ij}+cm_{i,c+1}+rm_{r+1,j}+m_{r+1,c+1})/rcn_{++}$.

The table of expected values of entries based on this missing data model can be estimated from the sample, as follows. The hypergeometric probabilities for the sample tables remain unknown. The limiting distribution for all tables is $\pi(n)$, and probabilities p_k, p_l for sample tables T_k, T_l are in the same proportions as exhibited within the set of all tables, viz., d_k/d_l. As all tables consistent with the model exhibit grand total n_{++}, then so must the expected table, and consequently the sum of the combination weights for estimating the expected

value estimate is 1. The estimated expected value table is:

$$\frac{\sum_k p_k T i}{\sum_k p_k} = \frac{\sum_k \frac{d_k}{N} T_k}{\sum_k \frac{d_k}{N}} = \frac{\sum_k d_k T_k}{\sum_k d_k}.$$

These weights can also be used to compute a variety of other sample-based estimates. Such estimates typically are not integer. As controlled rounding can be performed on network tables (Cox and Kim (2006)), then base $B = 1$ controlled rounding applied to an estimated table yields a nearby integer table.

A second application is to use the sample to test the hypothesis that an imputed table is consistent with a particular incomplete data mechanism. Based on the missing data cell probabilities, one constructs a sample of tables and numerators d_k, computes the numerator d_0 for the imputed table, and compares d_0 with the empirical d-distribution to obtain an exact test p-value (Buena and Besag (2000)).

4.4. Exact bounds for table entries

Exact bounds on unknown table entries subject to fixed MSS are useful for a variety of reasons. For concreteness, we focus on statistical disclosure limitation (Federal Committee on Statistical Methodology (1994)). Small counts, meaning counts from 1 to some threshold t, represent unacceptable risk of disclosure of the identity and confidential data pertaining to survey respondents. One solution is to withhold cell counts from public release and instead release selected marginal totals. The issue is then whether released marginals can be combined to reconstruct or estimate small counts.

This exact bounding problem is equivalent to a possibly large set of integer linear programs over a typically large set of variables and constraints, viz., for each small withheld count s: $\min\{n_s\}$, $\max\{n_s\}$ subject to the released marginals. For tables of network type, these reduce to network computations. In tables so large that the number of optimizations is overwhelming, the method of Sections 4.2 could be used to create a probability sample from which exact bounds for selected entries could be estimated from weak bounds and an empirical distribution based on sample values for the entry.

There is an even more significant advantage of network structure for bounding entries in contingency tables. De Loera and Onn (2004) demonstrated that arbitrary gaps can exist in integer solution sets of integer programs. Solution sets for a linear program are *interval*, meaning that subject to linear constraints all values in the range $[\min\{x\}, \max\{x\}]$ are feasible for each LP-variable x. De

Loera-Onn showed that solution sets for integer linear programs are not necessarily interval, and indeed can exhibit arbitrary *gaps* between consecutive feasible integer values for a particular variable. This fundamentally challenges disclosure limitation theory and practice for count data, as the notion of a safe threshold t for small counts is based implicitly on the number of possible values the count can take. De Loera-Onn showed, in effect, that an arbitrary small count could have integer feasible set $\{1, t\}$ which disclosure practitioners would regard as too risky, but which is permitted by the t-threshold rule. Networks have a strong advantage over most other tabular systems in this regard, as follows.

Theorem 7. *Integer solution sets for a network are interval.*

Proof. Choose an arc and any integer between its integer minimum and maximum. Capacitate the arc below and above by this value. LP is convex, and therefore interval, so the capacitated network is LP-feasible. The network has only integer extreme points, each of which exhibits the selected integer value for the arc.

Thus, disclosure limitation theory for counts over tables of network type remains sound.

Another set of tables that is interval, and for which a simple formula for exact bounds is available, correspond to decomposable graphical models (Dobra and Fienberg (2000)). The most familiar log-linear models in this class are complete independence models, viz., 1-dimensional MSS. Example 2 implies that decomposable and network models define different classes of partially specified tables. Combined, these two classes offer the practitioner a rich, varied set of theoretically verified, computable tables.

5. Network Tables in the Context of All Tables

Section 3 provided a list of important, familiar tables that are of network type, and similarly demonstrated that many other tables are not network. In the latter case, the polyhedron defined by the linear programming feasible region of the partially specified table often exhibits non-integer extreme points. This means that in most cases integer-valued contingency table problems cannot be solved by continuous network or linear programming methods, and may require combinatorial optimization, which is NP-hard. In many cases a Gröbner basis is not available, and related techniques cannot be applied.

In Section 5.1, we will examine the utility of techniques for collapsing non-network tables into network tables in order to obtain partial information that otherwise may be unavailable. De Loera and Onn (2004) demonstrated that any integer linear program is isomorphic to a partially specified 3-dimensional contingency table of size $r \times c \times 3$ (a *slim table*) subject to all 2-dimensional marginal

totals; the mapping preserves polyhedral extreme points and integer points. Theorem 4 showed that a partially specified 2-dimensional tables of size $r \times c \times 2$ (a thin table) subject to all 2-dimensional marginal totals is network. In Section 5.2, we will examine our results in terms of the De Lorera-Onn mapping and will present a refined mapping separating network and non-network tables.

5.1. Collapsing non-network tables to network tables

Section 3 demonstrated that a thin partially specified m-dimensional table subject to $(m-1)$-dimensional marginals along the thin dimensions is network, and so too is a partially specified table subject to at most two configurations of MSS. Conversely, Examples 1 and 2 indicate that many other partially specified tables are not network. The question arises as to whether we can apply network methods to achieve useful partial information for non-network problems that may otherwise be intractable. One approach is to replace the original problem by a closely related (collapsed) problem that is network.

The first approach is to collapse categories along individual dimensions. For k-dimensional tables subject to $(k-1)$-dimensional marginal totals, one can collapse all but two of the dimensions into two subgroups, viz., along each of the $k-2$ dimensions, assign each original category to one of two subgroups and aggregate original category counts into subgroup counts. The resulting table is then thin, hence network, and can be analyzed efficiently and completely.

Collapsing categories is particularly useful in the context of our third problem — bounding entries — as bounds resulting from analysis of the collapsed table, while potentially weaker, are nevertheless valid bounds. In a disclosure limitation setting, this partial information may be crucial as, if any weak bound is unsafe, the cell is not protected and the table (collapsed or uncollapsed) cannot be released. In our first and second problems, involving inference and estimation based on a probability sample, collapsing categories may mean the difference between having a sample or not.

The second approach is to collapse configurations of MSS. One method is to collapse individual MSS until only two configurations remain. For example, $MSS = \{n_{ij+}\}, \{n_{i+k}\}, \{n_{+jk}\}$ can be collapsed to $MSS = \{n_{i++}\}, \{n_{+jk}\}$. This enables efficient and valid analysis or inference, albeit coarser. Another method is to consider the MSS in pairs, perform efficient analysis on each pair, and compare results. The utility of this approach for inference is not altogether clear, but once again it will yield useful and perhaps valuable information for bounding problems. The point to keep in mind is that collapsing is not to be used in lieu of full analysis if full analysis can be performed by other means,

but rather to provide an alternative to the data provider or analyst in situations where no efficient analytical method is available for the full problem.

5.2. Thin and slim contingency tables

We have shown that partially specified thin 3-dimensional tables subject to 2-dimensional marginals are network. De Loera and Onn (2004) demonstrated that any integer linear programming problem can be represented as a partially specified slim 3-dimensional contingency table subject to 2-dimensional marginals (line sums). The mapping, defined over the linear programming relaxation, is isomorphic, so that feasible points, integer points, and extreme points of the original and slim problems correspond one-to-one. Each rational integer linear program is mapped to a network biflow which, after including slack, is slim. The mapping does not distinguish networks from non-networks. Inspired by their result, we presented a mapping of a partially specified k-dimensional table subject to fixed marginals (MSS) to a slim 3-dimensional table subject to 2-dimensional MSS that maps the networks of Theorems 3 and 4 to thin 3-dimensional tables and maps non-networks to slim-but-not-thin tables.

Our mapping preserves the coefficient matrix of the original k-dimensional table, assuring that the polyhedron is feasible, integer and extreme points correspond one-to-one. Following De Loera-Onn, we introduce a fresh set of variables for each constraining MSS, viz., we map each variable in the original system to an appropriate number of distinct variables in the slim table, and we employ complementary variables, viz., for variable x in the slim table, we introduce variable $\overline{x}$ such that $x + \overline{x} = a$, where a is a uniform upper bound on all variables. We use complements to ensure that all unique variables mapped from the same original variable in the original constraint system are forced to be equal. The details follow.

To simplify notation, we illustrate the mapping for $k = 3$. Each original variable occurs in at most three constraint equations, and maps to (at most) three variables x, y, z in the slim table. Consider the partial slim 3-dimensional table of Figure 1.

$$\begin{pmatrix} x & 0 & 0 \\ 0 & \overline{y} & 0 \\ 0 & 0 & \overline{z} \\ \overline{x} & y & 0 \\ 0 & 0 & z \end{pmatrix} \quad \begin{pmatrix} 0 & 0 & 0 \\ 0 & y & 0 \\ 0 & 0 & 0 \\ 0 & \overline{y} & z \\ 0 & 0 & \overline{z} \end{pmatrix} \quad \begin{pmatrix} \overline{x} & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & z \\ x & 0 & \overline{z} \\ 0 & 0 & 0 \end{pmatrix}$$

Figure 1. Partial Slim 3-Dimensional Table.

We say "partial" because the full table has many more rows and columns in each of its three planes. All columns above are extended downwards with zeroes; how rows are extended and additional rows added is described below.

Begin with any original variable. All column equations are complementarity equations: a slim table variable plus its complement equals a. The first thee rows of the left plane specify the three (or fewer) constraint equations in which the original variable appears; the slim table variables corresponding to the original variable are x, y, z. The first equation appears in its original form. The second and third equations are re-written in terms of complementary variables. The fourth row equation in this plane has zeroes in all other entries except the two shown and by setting the constant value of this equation to a, forces $x = y$. The fifth row is simply the original third constraint equation.

The first and third rows of the middle plane are all zeroes. The second row is the second equation. The fourth is set to a and forces $y = z$. The fifth is the third equation, rewritten in complementary form. The first row of the right plane is the first equation, rewritten in complementary form. The second is all zeroes. The third is the third equation. The fourth is set equal to a and forces $x = z$. The fifth is all zeroes. All non-zero equations running along the slim dimension are complementarity equations, viz., the sum of two variables is a.

The procedure is iterated as follows. The next original table variable is represented below and to the right of the preceding one. First, any constraint equations involving this variable not already represented in the slim table system are written into the next row(s) of the left plane. Note that only zeroes appear to the left of the target variable in these new equations. The corresponding rows of the middle and right planes and the next two (or fewer) rows in all three planes are filled in the same manner as for the preceding original variables. The result is a very large, sparse, partially specified, slim 3-dimensional table subject to 2-dimensional MSS whose duplicated variables x, y, z are forced equal to a common original variable, and whose polyhedron corresponds one-to-one with that of the original partially specified k-dimensional table.

One advantage of this representation is that if the original k-dimensional table is represented as a bipartite network (Theorem 3), the image of the mapping collapses naturally to a thin 3-dimensional table, represented in Figure 2.

$$\begin{pmatrix} x & 0 & 0 \\ 0 & \overline{y} & 0 \\ \overline{x} & y & 0 \end{pmatrix} \quad \begin{pmatrix} \overline{x} & 0 & 0 \\ 0 & y & 0 \\ x & \overline{y} & 0 \end{pmatrix}$$

Figure 2. Partial Thin 3-Dimensional Table for Networks of Theorems 3 & 4.

Similarly, if the original table is a network comprising two identical networks connected by a line sum over a thin dimension (Theorem 4), then the

mapping may be revised to collapse to a thin 3-dimensional table, as follows. First, map only the original variables from the first thin plane. Second, replace complementary variables $\overline{x}$ and equations $x + \overline{x} = a$, $x + \overline{y} = a$, $\overline{x} + y = a$ by corresponding thin line equations $x_{ij1} + \overline{x}_{ij2} = n_{ij+}$, etc., resulting again in Figure 2. Conversely, if the original partially specified k-dimensional table is not network, then it cannot map to a thin 3-dimensional table, and instead must map to a slim-but-not-thin table.

6. Concluding Comments

We have modeled a large class of familiar, important log-linear models (a.k.a. partially specified contingency tables) as mathematical networks, and have provided theory and algorithms for computing three important, general problems over these tables: probability sampling of contingency tables, imputing for item and unit nonresponse subject to edit constraints, and exact bounding of entries. We have constructed Markov bases for partially specified contingency tables of network type comprising $\{-1, 0, +1\}$-moves. We show how to avoid creating infeasible solutions. Theoretical and computational problems for the vast majority of problems outside our class are formidable. We show how to collapse tables that are not network to network tables, enabling efficient partial analysis in these cases. We relate our classification to that of De Loera-Onn for general integer linear programs, present a refined mapping that separates network from non-network tables, and show that network tables overcome the integer gap limitation. Our results argue that table designers and analysts can benefit from creation and use of tables within a well-understood class as developed here to facilitate accurate and efficient analysis, particularly in situations where full analysis is not possible or practical.

Acknowledgement

We acknowledge with thanks the careful review and an observation from a referee that Theorem 5 generalizes from network matrices to totally unimodular matrices. We acknowledge with appreciation the meticulous review and suggestions from an associate editor, and an invitation to submit this paper to a special issue from an editor.

Disclaimer

This paper represents the work of the author and should not be interpreted as representing the policies or practices of the Centers for Disease Control and Prevention or any other organization.

References

Bishop, Y. M., Fienberg, S. E. and Holland, P. W. (1975). *Discrete Multivariate Analysis: Theory and Practice*. MIT Press, Boston.

Buena, F. and Besag, J. (2000). MCMC in $I \times J \times K$ contingency tables. *Fields Inst. Commun.* **26**, 25-36.

Causey, B. D., Cox, L. H. and Ernst, L. R. (1985). Applications of transportation theory to statistical problems. *J. Amer. Statist. Assoc.* **80**, 903-909.

Cox, L. H. (1987). A constructive procedure for unbiased controlled rounding. *J. Amer. Statist. Assoc.* **82**, 520-524.

Cox, L. H. (1995). Network models for complementary cell suppression. *J. Amer. Statist. Assoc.* **90**, 1453-1462.

Cox, L. H. (2002). Bounds on entries in 3-dimensional contingency tables subject to given marginal totals. In *Inference Control in Statistical Databases, Lecture Notes in Computer Science* 2316 (Edited by J. Domingo-Ferrer), 21-33. Springer-Verlag, Berlin.

Cox, L. H. (2003). On properties of multi-dimensional statistical tables. *J. Statist. Plann. Inference* **117**, 251-273.

Cox, L. H. (2004). Inference control problems in statistical database query systems. In *Research Directions in Data and Applications Security* (Edited by C. Farkas and P. Samarati), 1-13. Kluwer, Boston.

Cox, L. H. and Ernst, L. R. (1982). Controlled rounding. *INFOR* **20**, 423-432.

Cox, L. H. and George, J. A. (1989). Controlled rounding for tables with subtotals. *Ann. Oper. Res.* **20**, 141-157.

Cox, L. H. and Kim, J. J. (2006). Effects of rounding on the quality and confidentiality of statistical data. In *Privacy in Statistical Databases* 2006, *Lecture Notes in Computer Science* 4302 (Edited by J. Domingo-Ferrer and L. Franconi), 48-56. Springer-Verlag, Heidelberg.

De Loera, J. and Onn, S. (2004). All rational polytopes are transportation polytopes and all polytopal integer sets are contingency tables. In *IPCO* 2004*, Lecture Notes in Computer Science* 3064 (Edited by D. Bienstock and G. Nemhauser), 338-351. Springer-Verlag, Berlin.

Diaconis, P. and Sturmfels, B. (1998). Algebraic algorithms for sampling from conditional distributions. *Ann. Statist.* **26**, 363-397.

Dobra, A. and Fienberg, S. (2000). Bounds for cell entries in contingency tables given marginal totals and decomposable graphs. *Proc. Nat. Acad. Sci.* **97**, 11885-11892.

Federal Committee on Statistical Methodology (1994). Statistical Policy Working Paper 22: Report on Statistical Disclosure Limitation Methodology. Office of Management and Budget, Washington, DC (NTIS PB94-165305). Available: www.fcsm.gov

Garey, M. R. and Johnson, D. S. (1979). *Computers and Intractability: A Guide to the Theory of NP-Completeness*. WH Freeman and Company, New York.

Greene, M. A., Smith, L. E., Levenson, M. S., Hiser, S. and Mah, J. C. (2001). Raking fire data. Proceedings of 2001 FCSM Research Conference. Federal Committee on Statistical Methodology, Office of Management and Budget, Washington, DC: Available: http://www.fcsm.gov/events/papers2001.html

Hemmecke, R. and Malkin, P. (2006). **4ti2**. Available: www.4ti2.de

Karr, A. F., Dobra, A., Sanil, A. P. and Fienberg, S. E. (2002). Software systems for tabular data releases. *International Journal on Uncertainty, Fuzziness and Knowledge- based Systems* **10**, 529-544.

Kelly, J. P., Golden, B. L., Assad, A. A. and Baker, E. K. (1990). Controlled rounding of tabular data. *Oper. Res.* **38**, 760-772.

Kennington, JL and Helgason, RV. (1980). Algorithms for Network Programming. Wiley, New York.

Lang, S. (1965). *Algebra*. Addison-Wesley, Reading, MA.

Little, R. and Rubin, D. (2002). *Statistical Analysis with Missing Data*. Wiley Interscience, New York.

Nemhauser, G. L. and Wolsey, L. A. (1988). *Integer and Combinatorial Optimization*. Wiley, New York.

Pistone, G., Riccomagno, E. and Wynn, H. P. (2001). *Algebraic Statistics: Computational Commutative Algebra in Statistics*. Chapman & Hall, New York.

Sturmfels, B. (1996). *Gröbner Bases and Complex Polytopes*. American Mathematical Society, Providence.

National Center for Health Statistics, 3311 Toledo Road, Hyattsville, MD 20782, U.S.A.

E-mail: lcox@cdc.gov

(Received August 2006; accepted April 2007)