

CONFERENCE OF EUROPEAN STATISTICIANS

UNECE Work Session on Statistical Data Editing
(27-29 May 2002, Helsinki, Finland)

Topic (iv): Impact of new technologies on statistical data editing

**IMPUTING MISSING VALUES IN TWO-WAY CONTINGENCY TABLES USING
LINEAR PROGRAMMING AND MARKOV CHAIN MONTE CARLO**

Contributed Paper

Submitted by National Center for Health Statistics, CDC/DHHS, United States¹

Abstract: Observations on two categorical variables are to be cross-classified in a contingency table \mathbf{B} . Some units are classified by both row and column, comprising a subtable \mathbf{A} of \mathbf{B} . However, due to nonsampling error, the remaining units are classified only by row, or only by column, or not at all. The problem is to impute classifications for the remaining units, viz., add them to appropriate counts of \mathbf{A} , resulting in an unbiased estimate $\hat{\mathbf{B}}$ of the true but unobserved table \mathbf{B} . A classical approach to related problems is iterative proportional fitting (IPF) which produces MLEs for the entries of \mathbf{B} , based on known marginals. Here, the relevant \mathbf{B} -marginals are unknown. Also, under IPF some MLE's may be smaller than the original entries of \mathbf{A} , sample zeroes are fixed at zero, imputation variance is ignored, and MLE's are likely to be noninteger. We present an alternative approach, based on linear programming and MCMC, that overcomes these drawbacks and in addition is capable of constraining imputations to within specified sampling variability. Our approach reveals a new method for MCMC simulation in an important class of multi-dimensional contingency tables.

I. INTRODUCTION

1. A sample of observations on two categorical variables is to be cross classified into a contingency table \mathbf{B} . Some sample units are classified *completely*, viz., by both row and column, and define a contingency table \mathbf{A} . Some of the remaining units can be classified only by row, others only by column, and others not at all. The problem is to use available information to augment \mathbf{A} by assigning the remaining units to the cross-classified categories, resulting in an unbiased estimate $\hat{\mathbf{B}}$ of the true but unobserved contingency table \mathbf{B} . Greene et al. (2001) describe an application involving classification of fires, and evidently the underlying problem is of general interest and applicability.

2. Greene et al. (2001) attempt a solution to this problem based on iterative proportional fitting (*IPF*), a classical method for contingency table imputation. Unfortunately, our problem involves variable row and column totals whereas in classical IPF these are fixed. Greene et al. (2001) handle this by scaling the row and column marginals of \mathbf{A} up to \mathbf{B} . This leads to several problems, including the possibility that some estimated counts (for \mathbf{B}) are less than known counts (from \mathbf{A}). To avoid specifying marginals for \mathbf{B} , Schenker (2001) suggests replacing IPF with the *E-M algorithm*. IPF has additional drawbacks, discussed in Section II, and in particular IPF and E-M do not assure integer estimates. We present an exact approach based on networks, a specialized linear program, and Markov chain Monte Carlo (*MCMC*). Our solution reveals a new method for MCMC simulation based on networks in an important class of contingency tables.

¹ Prepared by Lawrence H. Cox (LCOX@CDC.GOV)

3. Section II provides a mathematical formulation of the problem, based on mathematical networks, and presents the major steps in the solution, based on Markov chain Monte Carlo simulation. Section III identifies an important class of multi-dimensional contingency tables with specified marginals that can be represented as networks and presents a new method for MCMC simulation on a network. This class subsumes the class of all two-dimensional tables with fixed one-dimensional marginals and all three-dimensional tables with fixed two-dimensional marginals. Section IV contains concluding comments.

II. AN IMPUTATION PROBLEM IN TWO-DIMENSIONAL CONTINGENCY TABLES

II.1 The Imputation Problem

4. A survey is conducted based on a sample of size b_{++} IID population units. Survey managers want to cross-classify sample units by two exclusive and exhaustive sets of categories, viz., want to represent the cross-classification in the form of an $r \times c$ contingency table $\mathbf{B} = (b_{ij})$ of nonnegative counts totaling to a full sample of b_{++} observations. Due to nonsampling error, for $1 \leq i \leq r$, $1 \leq j \leq c$, only $a_{ij} \leq b_{ij}$ observations are classified completely by row and by column, for a total of $\sum_{i,j} a_{ij} = a_{++} < b_{++}$ completely

classified units. These units define an $r \times c$ contingency table \mathbf{A} involving a_{++} units. Among the remaining $m = b_{++} - a_{++}$ units, $m_{i, c+1}$ can be classified only by row (are *missing by column*), $m_{r+1, j}$ can be classified only by column (are *missing by row*), and, $m_{r+1, c+1}$ cannot be classified by neither row nor column (are *missing by row and column*). These relationships are summarized in Table 1. The inner box apportions the b_{++} individual units; the outer boxes present the row, column and grand totals. A special case is when both \mathbf{A} and m are zero, viz., the *no two-way interaction* log-linear model whose sufficient statistics are the one-dimensional marginals. Here, one-way interactions are replaced by lower bounds. The need to repeatedly impute tables consistent with the model arises, e.g., to assess goodness-of-fit of the model to a specific table.

$\begin{matrix} (a_{i,j}) & (m_{i,c+1}) \\ (m_{r+1,j}) & m_{r+1,c+1} \end{matrix}$	$\begin{matrix} (a_{i+} + m_{i,c+1}) \\ \sum_{j=1}^{c+1} m_{r+1,j} \end{matrix}$
$\begin{matrix} (a_{+j} + m_{r+1,j}) & \sum_{i=1}^{r+1} m_{i,c+1} \end{matrix}$	$\begin{matrix} b_{++} \\ = \\ a_{++} + m \end{matrix}$

Table 1: Contingency Table B With Observed (a) and Missing (m) Counts

5. Absent additional information, it is reasonable to assume that observations are *missing at random*, i.e., missingness by row or column and missingness by row and column are random within column, row and table. The problem is to estimate **B** based on the model specified in Table 1, viz., to distribute the m unclassified units among the entries of **B** in a manner that is unbiased and consistent with Table 1.

II.2 Drawbacks to A Classical Approach to the Problem

6. *Iterative proportional fitting* (Bishop et al. 1975, pp. 82-101) is a classical method for imputation in two- (and multi-) dimensional contingency tables based on known marginals. Unfortunately, IPF presents several drawbacks in this situation. First, *classical proportional fitting* (Deming and Stephan 1940) fits internal entries in a two-dimensional table to known one-dimensional marginal totals by iteratively *raking* interim values to each set of marginals in turn. The procedure converges to maximum likelihood estimates (MLEs) of the internal entries under a no two-way interaction log-linear model (Bishop et al. 1975). The drawbacks of IPF for the current problem are as follows. First, the one-dimensional marginals of **B** are not known. Second, raking can result in $b_{ij} < a_{ij}$ for some (i, j), which is infeasible (see, e.g., Greene et al. 2001). Sometimes this is a consequence of the first drawback. Third, raking preserves sample zeroes, which are not necessarily true (population) zeroes. Fourth, raking does not assure an integer solution (although an approximate integer solution can be obtained by applying base 1 *unbiased controlled rounding* to the raked table; see Cox 1987). Fifth, raking produces a single table of MLEs close to the mean and ignores imputation variance—a randomly selected table is preferred. This requires an algorithm for randomly selecting an estimate $\hat{\mathbf{B}}$ of **B** from among all candidates satisfying Table 1.

II.3 A Solution Involving Linear Programming and MCMC

7. Linear programming formulations are worth exploring to avoid these drawbacks. Denote by $x_{ij} \geq 0$ the amount by which a_{ij} must be increased to yield an (unbiased) estimate \hat{b}_{ij} of b_{ij} . Table 1 is equivalent to the system of *integer linear constraints*:

$$\begin{aligned} x_{ij} &\geq 0, x_{ij} \text{ integer}, i = 1, \dots, r; j = 1, \dots, c \\ \sum_{i,j} x_{ij} &= m \\ x_{i+} &= \sum_j x_{ij} \geq m_{i,c+1} \\ x_{+j} &= \sum_i x_{ij} \geq m_{r+1,j} \end{aligned} \tag{1}$$

The latter two conditions are *capacity constraints*. Any solution $\mathbf{X} = \{x_{ij}\}$ of (1) avoids the second drawback. As \mathbf{X} does not depend on $\mathbf{A} = (a_{ij})$, the third drawback is avoided. By virtue of the missing-at-random assumption, it suffices to assume that \mathbf{A} and $\mathbf{X}|\mathbf{A}$ are random samples (of size a_{++} and $m = b_{++} - a_{++}$, respectively) drawn from the distribution $\boldsymbol{\pi}$ of all contingency tables \mathbf{B} of size b_{++} satisfying the constraints of Table 1. The distributional form of $\boldsymbol{\pi}$ is determined by the sampling and nonresponse mechanisms. For example, assuming that sample units are exchangeable and that the $p_{ij} = P\{\text{randomly selected unit has characteristics } (i, j)\}$ are Poisson, then $\boldsymbol{\pi}$ is a multinomial distribution over rxc tables of size b_{++} generated from a complete independence model subject to the constraints of Table 1.

8. The linear system (1) defines the constraint system for a specialized form of linear program--a *network*. See Ahuja et al. (1993) for details on networks. Networks can be solved with extreme speed and efficiency, even for large problems, by specialized continuous methods available in standard commercial software. Networks assure that the same methods yield integer solutions whenever constraints are integer. This avoids the fourth drawback, and assures that problems of arbitrary size are computable.

9. The network corresponding to equation system (1) is constructed in the form of a diagram comprising *nodes* and directed *arcs*, as follows. A *source node* is drawn on the left of the diagram. The source node is assigned a *node requirement* equal to $-m$, signifying that this node instantiates a *flow* of m units to the network. Each row i of the table is represented by a node called a *row node*. These are arranged vertically to the immediate right of the source node. Each row total x_{i+} is represented by a directed arc from the source node to row node i . This arc is assigned *lower capacity* equal to $m_{i, c+1}$, signifying the minimum flow that this arc will carry. Each column j of the table is represented by a node called a *column node*. These are arranged vertically to the right of the row nodes. Each internal entry x_{ij} is represented by a directed arc from row node i to column node j . By convention, all arcs in a network have lower capacity at least zero, and equal to zero if a capacity is not specified. This is the case for all (i, j) arcs here. By convention, all nodes in a network have node requirement zero unless otherwise stated. This is the case for all row nodes i and column nodes j here. Finally, to the right of the column nodes is drawn a *sink node*. The sink node has node requirement equal to m , signifying that it must receive m units of flow. Each column total x_{+j} is represented by a directed arc from column node j to the sink node. This arc has lower capacity equal to $m_{r+1, j}$. Denote this network N . Imputed contingency tables \mathbf{B} thus correspond one-to-one with *feasible integer solutions* $\mathbf{X} = \mathbf{n}$ of the network via $\mathbf{B} = \mathbf{A} + \mathbf{n}$.

10. An initial feasible integer solution $n(0)$ to (1) is obtained by performing a *maximum flow* computation on the network (Ahuja et al. 1993, pp. 69-70). MCMC simulation offers a method for producing a randomly selected feasible table n , provided that a computationally feasible algorithm for generating the chain is available: Start the chain at $n(0)$, run the chain (the *Markov step*), and, after allowing sufficient steps to assure that the limiting distribution of the chain has been reached (*burn-in*), stop the chain at a randomly selected step. *Multiply imputed* replicates of the table (Rubin 1987) can be obtained via multiple random draws, provided that procedures (e.g., from Bunea and Besag 2000) are in place to avoid the effects of serial correlation induced by the choice of moves. MCMC is discussed in detail in Section 2.4. Contingent on the existence of a computationally feasible algorithm for generating the chain, the fifth drawback is avoided. Section 2.4 examines difficulties in computing the Markov chain. Section 3 presents an efficient algorithm for doing so in an important class of tables.

II.4 MCMC Simulation in Contingency Tables

11. The Markov step requires two mechanisms: (a) a procedure for randomly selecting a local “move” from the current feasible integer solution to a neighboring one and (b) a procedure for deciding whether or not to move that preserves π as the limiting distribution of the chain (*Metropolis step*). The procedure in (a) must be *symmetric*, meaning that the probability of moving back from the second solution to the first equals the probability of the original forward move, and in addition the procedure must ensure that the chain is *irreducible* (Robert and Casella 1993), meaning that, starting at any feasible solution, all other feasible solutions eventually can be reached. For practical purposes, both mechanisms need to be computationally efficient. Irreducibility holds for chains based on two-dimensional tables with fixed one-dimensional marginals (Diaconis and Sturmfels 1998), and, as demonstrated in Section 3, also for our problem and any k-dimensional table subject to (k-1)-dimensional marginals representable as a network.

12. The Markov step in MCMC is usually based on a single *simple move*, viz., moving one unit around a subset of the entries of a current integer solution of (1) to reach a second, nearby, integer solution. This presents significant theoretical and computational challenges in general multi-dimensional tables, as follows. Ignoring for the nonnegativity and capacity constraints of (1), the set of all integer moves between solutions of (1) forms a mathematical *ring* \mathbf{R} over the integers \mathbf{Z} , viz., the sum, difference and integer multiple of any move is a move. A minimal subset of \mathbf{R} from which all moves in \mathbf{R} can be generated by addition, subtraction and integer multiplication is called an *integer basis* of \mathbf{R} . Mathematical theory provides a means for computing a basis, via the theory of *Gröbner bases* (see Adams and Loustaunau 1994), but these computations have to date been carried out for special categories of tables or for only a few general tables of small size and dimensions (Diaconis and Sturmfels 1998). In essence, for multi-dimensional tables the problem can be posed and a solution sketched but not computed.

13. The situation in two-dimensions is quite special. For fixed one-dimensional marginals, a basis for the ring of all moves consists of all “squares” drawn between four table entries $\{ (i, j), (i, k), (l, k), (l, j) : i \neq l, j \neq k \}$. Simple moves correspond to assigning values +1 or -1 to these entries in an alternating manner, viz., $\{ +1, -1, +1, -1 \}$. All moves between solutions of (1) are generated by this set of simple basic moves, called “moves of four”. Similarly, if only the grand total is fixed, then a basis for the ring of moves is given by all “moves of two,” viz., $\{ (i, j), (l, k) : (i, j) \neq (l, k) \}$. Our imputation problem lies somewhere in between: only the grand total is fixed, but lower bounds are prescribed on each one-dimensional marginal total. The only clear approach to construct the chain using standard methods would be to construct $n(0)$, randomly generate moves of two, and reject infeasible moves. This is likely to be inefficient and furthermore is limited to the imputation problem. An improved, much more general approach is offered in Section 3 which exploits the network structure to construct a Markov chain comprising only feasible moves.

14. There are two actions associated with the decision whether or not to move to a new solution. First, if the simple move selected is not a *feasible move*, viz., a move to a feasible table, then the move is not made. Second, the decision must be made whether to remain at the current step (feasible integer solution $n(t)$) or to move to the resulting step (feasible integer solution $n(t+1)$ reached by making the move). This decision, the Metropolis step, is made based on π : the move is made with probability $\min \{ 1, \pi(n(t+1))/\pi(n(t)) \}$. This assures that π is the limiting distribution of the chain and, consequently, stopping the chain at a random step after burn-in is equivalent to randomly selecting a feasible integer table from π . Note that, although π involves an unknown normalizing constant, the ratio is computable, e.g., for π multinomial.

II.5 Preserving Sampling Variability

15. It may be desirable to avoid selecting imputed table(s) from the distributional tails of π . This can be accomplished as follows. Denote by $\hat{\mathbf{B}}$ a matrix of MLE for internal entries b_{ij} of \mathbf{B} and by $\hat{\boldsymbol{\varepsilon}}$ a matrix of estimated standard errors for the MLE estimates. All contingency tables $\mathbf{B} = \mathbf{A} + \mathbf{X}$ that lie within specified sampling variability of $\hat{\mathbf{B}}$ can be characterized by imposing *capacity constraints* on network (1):

$$U((\hat{\mathbf{B}} - \mathbf{A}) - \mathbf{p}\boldsymbol{\varepsilon}) \leq \mathbf{X} \leq D((\hat{\mathbf{B}} - \mathbf{A}) + \mathbf{p}\boldsymbol{\varepsilon}) \quad (2)$$

\mathbf{p} is a vector of constants specifying, in units defined by their standard deviations, how far solutions can vary from the MLEs, e.g., $\mathbf{p} = \mathbf{2}$. The functions U and D specify integer round-up and round-down operations, respectively, and are applied to assure that solutions to the capacitated network (1), (2) are integer. To assure, e.g., that all imputed tables lie within two standard errors of one another, set $\mathbf{p} = \mathbf{1}$.

16. How to obtain the estimated MLEs $\hat{\mathbf{B}}$? Using only the fixed grand total b_{++} , IPF results in the MLE estimate $(b_{++}/a_{++})\mathbf{A}$ used by Greene et al. (2001). This estimate, based on only a_{++} observations and a *no effects* model, is likely to be imprecise. More important, this estimate fails to account for all of the constraints of Table 1, viz., patterns of missingness specified by the m -constraints. Subtables \mathbf{C} of \mathbf{B} that take this information into account are characterized by Table 2. An unbiased $\hat{\mathbf{B}}$ results from applying IPF to \mathbf{C} , resulting in $\hat{\mathbf{C}}$, and scaling up to \mathbf{B} , viz., $\hat{\mathbf{B}} = (b_{++}/(b_{++} - m_{r+1,c+1}))\hat{\mathbf{C}}$. Indeed, the network-MCMC approach notwithstanding, this step alone improves upon Green et al. (2001).

$(c_{i,j})$	$(a_{i+} + m_{i,c+1})$
$(a_{+j} + m_{r+1,j})$	$b_{++} - m_{r+1,c+1}$

Table 2: Subtables of \mathbf{B} That Account for Patterns of Missingness Specified in Table 1

17. To summarize our method: A starting solution (integer feasible table) n_0 is obtained by solving the capacitated network (1), (2). MCMC simulation is run starting at this solution and stopping at a future random step, yielding a random feasible integer table, corresponding to an unbiased estimate $\hat{\mathbf{B}}$ of \mathbf{B} . Multiple imputation can be accomplished by stopping the chain at multiple random steps provided procedures are employed to avoid serial correlation in the chain. In the next section, we introduce a specialized method for performing MCMC on a network and discuss its advantages.

III. A METHOD FOR MCMC SIMULATION ON MULTI-DIMENSIONAL TABLES REPRESENTABLE AS NETWORKS

III.1 A Network Method for Constructing the Markov Chain

18. We present a new method for imputation in contingency tables based on networks and MCMC simulation that is applicable to any multi-dimensional contingency table with specified marginals representable as a network. This class contains the imputation problem of Table 1. It also contains the class of all k -dimensional contingency tables of size $r \times c \times 2^{k-2}$ with fixed $(k-1)$ -dimensional marginal totals, and in particular all two-dimensional tables with fixed one-dimensional marginals and three-dimensional tables of size $r \times c \times 2$ with fixed two-dimensional marginals (Cox 2000). The latter case is important, e.g., for the *Rasch model* (see Bunea and Besag 2000).

19. Networks N corresponding to tables in the preceding paragraph are as follows. The network N for imputation problem (1) was described previously. For a two-dimensional table of size $r \times c$ with fixed one-dimensional marginals, simply remove the source and sink nodes and their arcs from N and instantiate node requirements as follows: at row node i , the node requirement is $-b_{i+}$, and at column node j , the node requirement is b_{+j} . The result is a simple *bipartite network*. For an $r \times c \times 2$ three-dimensional table with fixed two-dimensional marginals, N is given in Figure 1. For a k -dimensional table of size $r \times c \times 2^{k-2}$ with fixed $(k-1)$ -dimensional marginal totals, N is built iteratively from dimension $(k-1)$ to dimension k , as indicated in Figure 2. Cox (2000) provides detailed proofs and constructions.

20. Networks for performing MCMC simulation in covered situations are constructed from the preceding networks as follows. Begin with N , the network corresponding to a covered table and marginals of interest (from the preceding paragraph). The marginal total constraints can be represented in matrix form as $\mathbf{AX} = \mathbf{f}$. Any two solutions of $\mathbf{AX} = \mathbf{f}$ differ by an element of the *kernel of A*, $\text{Ker}(\mathbf{A}) =$ the set of vectors \mathbf{y} for which $\mathbf{Ay} = \mathbf{0}$. Note that these vectors are not nonnegative. $\text{Ker}(\mathbf{A})$ is a vector space, and consequently each vector in $\text{Ker}(\mathbf{A})$ is uniquely expressible as sums of constant multiples of vectors comprising a *vector space basis* for $\text{Ker}(\mathbf{A})$. We are only interested in those vectors in the kernel representing *feasible (integer) moves* from one feasible integer table to another feasible integer table. Such vectors must be integer-valued, and in addition must obey further constraints ensuring that, when added to an appropriate feasible table, a second feasible table is obtained. Careful construction below of the network ensures both conditions.

21. Let n denote the current feasible integer table in the MCMC simulation. The network M_n used to generate feasible moves from n is defined as follows. M_n has precisely the same nodes and arcs as N , but, in addition, corresponding to each original arc s there is a *reverse arc* s^* . All nodes have node requirement equal to zero. Randomly select a positive integer $n_* \leq m$. Forward arcs s are assigned upper capacity equal to n_* . Reverse arcs s^* are assigned upper capacity equal to the value of the corresponding entry in n . (In particular, arcs corresponding to entries with $n_s = 0$ are assigned upper capacity zero.) In our imputation problem, also assign lower capacity on reverse arcs s^* corresponding to row or column totals equal to n_s minus the lower capacity for the total, and capacitate original or reverse arcs corresponding to internal entries so as to force compliance with (2).

22. The equation system is thus: $(\mathbf{A}, -\mathbf{A})(s, s^*)' = \mathbf{0}$, $s, s^* \geq 0$. From linear programming theory, linear objective (cost) functions are minimized at *extreme points* (vertices) of the polyhedron defined by the linear constraints. From network theory, integer-valued constraint coefficients assure integer-valued extreme points. Thus, coordinates of M_n -extreme points $\{s, s^*\}$ are nonnegative integers.

23. Randomly assign costs h_s equal to either -1 or +1 to the forward M_n -arcs s . For each reverse arc s^* , randomly assign cost h_{s^*} equal to either 0 or $-h_s$. Run the network to minimize the resulting cost function h . An optimal solution corresponds to a M_n -feasible integer move d . The Metropolis step is based on the potential move from $n(t) = n$ to $n(t+1) = n + d$. Because all moves are possible—not just simple moves—this procedure will reduce serial correlation and require a smaller sample, thereby speeding computation. This procedure is both symmetric and irreducible.

24. Consequently, based on the network structure of the constraint system, an algorithm for MCMC simulation on a network representation of a covered table with appropriate marginals fixed is as follows.

ALGORITHM

0. Execute the network to obtain a *starting solution* $n(0)$
 1. Go to the *current solution*, denoted n and its corresponding network M_n
 2. Randomly assign costs -1 or +1 to the forward network arcs s
Randomly assign costs 0 or the opposite cost to their reverse arcs s^*
Compute a minimum cost network flow, denoted $d = \{e, e^*\}$
If $d = 0$, GOTO 2.
Metropolis step:
 With probability $\min\{1, \pi(n+d)/\pi(n)\}$:
 Replace current starting solution n by new current solution $n + d$,
 move to $n + d$, and include $n + d$ in the sample
 Otherwise, maintain current solution, location, and sample and GOTO 2
STOP when a sufficiently large sample of solutions has been obtained
GOTO 1.
- END

III.2 Illustration: Thin Three-Dimensional Tables and the Rasch Model

25. Bunea and Besag (2000) examine MCMC for size $r \times c \times 2$ three-dimensional tables with fixed two-dimensional marginal totals. Their method is based on moves among only eight table cells—the minimum number theoretically possible in three dimensions. In two dimensions, analogous moves (“moves of four”) form an integer basis, but in three dimensions “moves of eight” do not. The authors are concerned only with binary tables, viz., all entries 0 or 1, for which sums along the k -dimension are positive. This includes, the authors note, a popular formulation of the Rasch model for which the $k = 2$ table is the reverse image of $k = 1$, forcing marginal totals in the k -direction to equal 1. For such tables, the authors devise an algorithm based on moves of eight that may wander outside the feasible region, but returns after one additional move, and on that basis are able to successfully perform MCMC.

26 Bunea and Besag (2000) present a table problematic from this perspective, viz., not all totals positive, illustrated here as Table 3 and denoted n . This table permits no feasible moves of eight. Nevertheless, without specialized techniques, our network method reveals that there is precisely one basic integer move d , given in Table 4, and further that there is precisely one other feasible integer table, namely $n + d$, given in Table 5.

1	0	0
0	0	1
0	1	0

0	1	0
1	0	0
0	0	1

Table 3: 3x3x2 Table of Bunea and Besag (2000, Equation 3.2)

-1	1	0
1	0	-1
0	-1	1

1	-1	0
-1	0	1
0	1	-1

Table 4: Unique Integer Basic Move for Table 3

0	1	0
1	0	0
0	0	1

1	0	0
0	0	1
0	1	0

Table 5: Only Other Feasible Solution Subject to Marginals of Table 3

IV. CONCLUDING COMMENTS

27. We have presented two methods. The first is an approach to an imputation problem of interest and generality. We demonstrate its solution using networks and MCMC simulation. This method successfully connects edit constraints, viz., Table 1, with the imputation problem. Although, as we indicate, it is possible to solve the imputation problem without a new approach to MCMC simulation based on networks, there exists a larger class of multi-dimensional contingency tables subject to fixed marginals for which the details of MCMC computation are not clear, and may be complex and computationally inefficient. This motivates our solution of a second problem—a network method for efficient MCMC simulation—addressing this issue for a large and important class of multi-dimensional tables.

28. The network method offers improvement in several areas. First, the simple (and other) moves selected are guaranteed to be feasible, thereby avoiding unnecessary computation. Second, beyond two dimensions, characterization and enumeration of the simple moves may be difficult or complex. These are embodied in the network using our method. Third, at minuscule additional effort we are able to incorporate moves of more than a single unit and of greater complexity than simple moves. This is likely to reduce serial correlation, require smaller samples, and speed the algorithm. Fourth, computation can be performed using standard commercial software in fast time. The network structure thus guarantees feasibility, integrality, irreducibility and computational efficiency.

References

- Adams, W and P Lounstaunau (1994). An Introduction to Gröbner Bases. Providence, RI: American Mathematical Society.
- Ahula, R, T Magnati and J Orlin (1993). Network Flows: Theory, Algorithms, and Applications. Upper Saddle River, NJ: Prentice Hall.
- Bunea, F and J Besag (2000). MCMC in IxJxK contingency tables. *Fields Institute Communications* 26, 25-36.
- Bishop, Y, S Fienberg and P Holland (1975). Discrete Multivariate Analysis—Theory and Practice. Cambridge, MA: MIT Press.
- Cox, LH (1987). A constructive procedure for unbiased controlled rounding. *Journal of the American Statistical Association* 82, 520-524.
- _____ (2000). On properties of multi-dimensional contingency tables. Submitted.
- Diaconis, P and B Sturmfels (1998). Algebraic algorithms for sampling from conditional distributions. *Annals of Statistics* 26, 363-397.
- Greene, M, L Smith, M Levenson, S Hiser and J Mah (2001). Raking fire data. Presented: Federal Committee on Statistical Methodology Research Conference 2001. Available: <http://www.fcsm.gov>
- Robert, C and G Casella (1999). Monte Carlo Statistical Methods. New York: Springer.
- Rubin, D (1987). Multiple Imputation for Nonresponse in Surveys. New York: John Wiley.
- Schenker, N (2001). Discussion of three papers on treatment of missing data. Overhead transparencies from invited discussion at: Federal Committee on Statistical Methodology Research Conference 2001. Available: <http://www.fcsm.gov>

Figure 1

NETWORK FOR RXCX2 TABLE SUBJ. TO 2-DIM MARGINALS

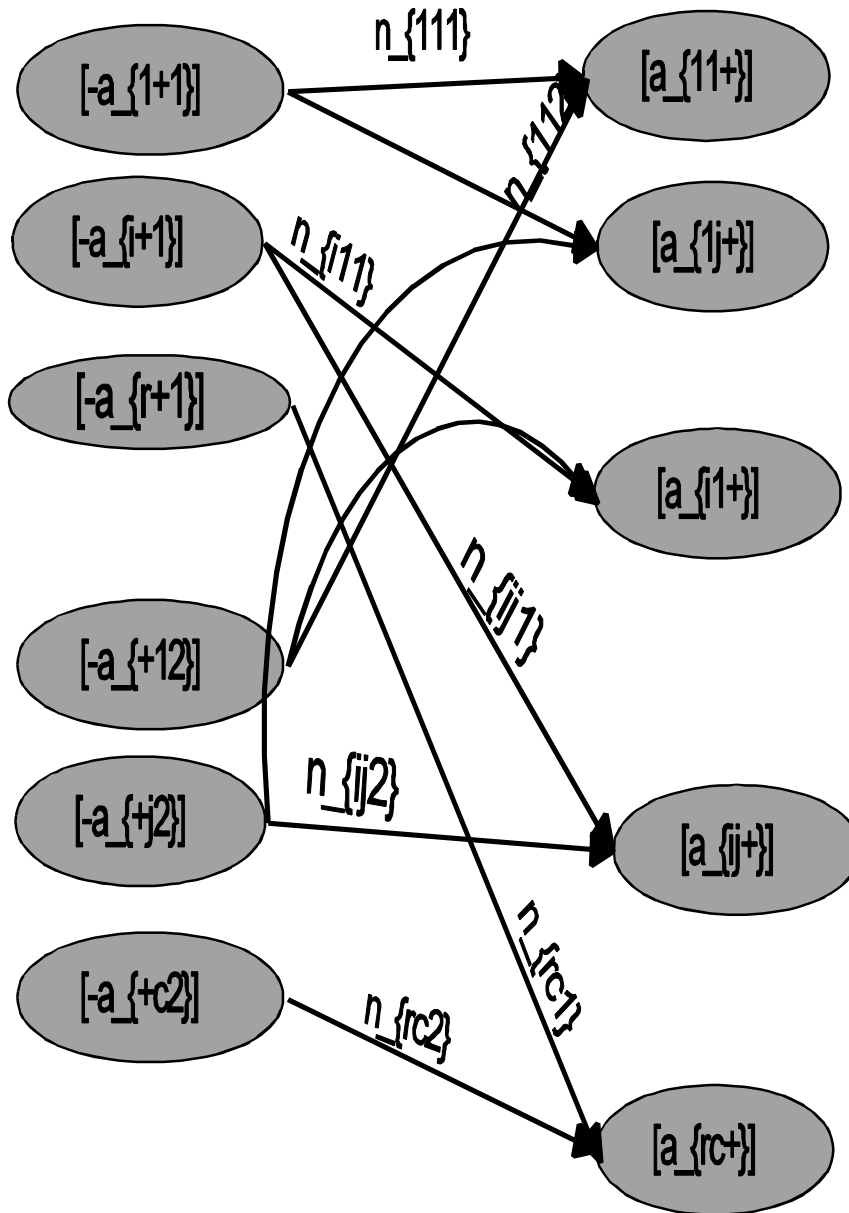


Figure 2

CREATING $rxcx2^L$ NETWORK FROM $rxcx2^{(L-1)}$ NETWORK

Original $rxcx2^{(L-1)}$ Network



New $rxcx2^L$ Network

