Algebraic algorithms for stochastic imputation of item nonresponse with edit restrictions

Mark van der Loo

Discussion paper (09037)

Statistics Netherlands

The Hague/Heerlen, 2009

Explanation of symbols

.	= data not available
*	= provisional figure
x	= publication prohibited (confidential figure)
–	= nil or less than half of unit concerned
–	= (between two figures) inclusive
0 (0,0)	= less than half of unit concerned
blank	= not applicable
2007-2008	= 2007 to 2008 inclusive
2007/2008	= average of 2007 up to and including 2008
2007/'08	= crop year, financial year, school year etc. beginning in 2007 and ending in 2008
2005/'06–2007/'08	= crop year, financial year, etc. 2005/'06 to 2007/'08 inclusive

Due to rounding, some totals may not correspond with the sum of the separate figures.

Publisher
Statistics Netherlands
Henri Faasdreef 312
2492 JP The Hague

Prepress
Statistics Netherlands - Grafimedia

Cover
TelDesign, Rotterdam

Information
Telephone .. +31 88 570 70 70
Telefax .. +31 70 337 59 94
Via contact form: www.cbs.nl/information

Where to order
E-mail: verkoop@cbs.nl
Telefax .. +31 45 570 62 68

Internet
www.cbs.nl

ISSN: 1572-0314

6008309037 X-10

Algebraic algorithms for stochastic imputation of item nonresponse with edit restrictions

Mark P.J. van der Loo

Summary:

Raw survey records often contain inconsistencies or missing items. To improve data quality, error correction and missing value imputation procedures are often included in the process of producing statistics. Imputation of missing values is often complicated by edit restrictions which reduce the number of value combinations that can be present in a record. The presence of edit restrictions also complicates estimating the amount of estimation uncertainty caused by an imputation step.

In this paper three algorithms are presented which can be used to impute missing datasets with categorical variables under edit restrictions. The algorithms are based on manipulating contingency tables rather than individual records. They are based on performing random walks on the set of all solutions to an imputation problem, a technique which has become available by recent progress made by Diaconis and Sturmfels (1998). The application to imputation problems as described here seems to be new.

A Metropolis-Hastings sampler and a Gibbs sampler have been implemented to draw random elements from a set of solutions to an imputation problem with edit restrictions. The probability associated with each solution can be modeled using familiar techniques from discrete data analysis such as log-linear or graphical models. A third algorithm, which finds a maximum likelihood solution under a specified probability model has been implemented as well.

The algorithms are tested on real survey datasets. It is concluded that the random walk algorithms offer a generic and fast way to impute categorical datasets and to study imputation variability. The efficiency of the different algorithms is also compared. The main conclusion is that although the Gibbs sampler needs less iterations than the Metropolis Hastings sampler, the total computational time necessary is about the same.

The current implementation is able to handle datasets where at most one item per record is missing. However, the theory described in this paper allows for generalisation to general missig data patterns without problems.

Keywords:

Stochastic imputation, categorical data, algebraic statistics, markov chain Monte Carlo

Contents

1 Introduction

One of the key tasks of a statistical institute is to transform raw data (microdata), either from surveys or administrative sources, into relevant statistical statements. It is therefore clear that the quality of microdata is of importance for the quality of the statements produced by an institute. It is well known however, that raw microdata is often plagued with errors or missing values. Since preventing all errors is either too costly or even impossible, statistical institutes include a data editing step in their statistical process to improve raw data quality. Data editing includes correcting inconsistencies and filling in missing values (imputation). It is desirable to automate this process as much as possible, and a lot of attention has been paid to develop algorithms and methods for data editing.

Many of the error correction methods are based on algorithms where a dataset is edited record by record. Examples are cold deck and hot deck imputation (Ford, 1983), nearest neighbour imputation, regression imputation or imputation based on the expectation maximization algorithm (Demster *et al.*, 1977; Wu, 1983). Error correction methods often follow the principle of Fellegi and Holt (1976) which basically states that as few fields as possible should be changed. One can show however that for systematic errors, this is not always the best choice. See for example Scholtus (2008).

Depending on the method, it can be difficult to estimate the amount of variance that imputation procedures add to estimated parameters. In general, analytical expressions for the variance are specific for an imputation model and can be cumbersome to derive.

The difference between this work and the methods mentioned above is that here instead of records, contingency tables based on the records are manipulated. In terms of contingency tables the set of solutions to a missing data problem is a finite set of nonnegative discrete vectors which obey certain linear conditions. Algorithms to draw random elements from such a set were proposed about a decade ago by Diaconis and Sturmfels (1998). The algorithms perform random walks on a finite convex set of discrete vectors. The main contribution of Diaconis and Sturmfels (1998) is a method to generate a set of steps (a markov basis) necessary to perform a random walk.

Here, an implementation of random walk algorithms in C and R is presented which facilitates the following two features.

1. Draw random solutions from the space of all solutions to an imputation problem according to a probability model. Metropolis-Hastings (Algorithm 1; p. 15) as well as Gibbs sampling (Algorithm 2; p. 16) algorithms

are implemented.

2. Perform maximum likelihood imputation based on a statistical model over the set of all solutions to the problem of imputing a dataset with missing items (Algorithm 3; p. 17).

The algorithms support edit restrictions which can be represented as structural zeros.

The methods are applicable to general m-way contingency tables, limited only by the practical ability to derive suitable probability models and (lack of) computational time. Also, the current implementation is suited only to deal with at most one missing item per record. A generalisation to include general missing data patterns is planned.

Apart from presenting the current implementation, this paper is aimed to give a short overview of the algebraic method for sampling contingency tables. Therefore, the rest of this paper is organized as follows: in the next section some basic properties of contingency tables are discussed and the problem of missing data is expressed in terms of contingency tables and their marginals. In section 3 a short introduction to multinomial probability models for contingency tables is given. In Section 4, the algorithms are listed and some implementation issues are discussed. Numerical results on the convergence properties of the implementation are presented in Section 5. In Section 6 the results are summarized and prospects for further development of the implementation are given. In Appendix A some basic properties of tensor products and direct sums of finite dimensional vector spaces are given. Appendix B gives some background on random walks and the Metropolis-Hastings sampler and in Appendix C it is shown explicitly that the markov basis used here is valid. As a reference, some frequently used notation is defined below.

Important notation. (Notation is also introduced in the text.) D: vector of discrete random variables D_i, x: contingency table, I: multi-index with nonnegative indices i_j with $1 \leq j \leq m$, referring to entries x_I in a contingency table. r, s and t are used as single indices of a contingency table, $d = \prod_{j=1}^{m} d_j$: dimension (number of cells) in a contingency table with d_j the number of levels of variable D_j. $\mathbb{Z}^k_{\geq 0}$: positive orthant[1] of $\mathbb{Z}^k$, $\vec{e}_I$: standard basis vector of $\mathbb{Z}^k$ with coefficients $e_{I'}$. The backslash ($\setminus$) is used to indicate set difference. A: linear map sending x to its marginals, $\mathfrak{I}$: set of edit constraints, $\mathcal{M}$, $\mathcal{U}$ multinomial and uniform distributions, Ω_X: space of valid contingency tables, $\mathbb{P}$: probability, E: expected value, $p(x)$: (pseudo) probability assigned x, p: vector of pseudoprobabilities with entries $p_x = p(x)$. $|\cdot|$: absolute value or set cardinality and $[k] \equiv \{0, 1, \ldots, k\}$.

Table 1. Three categorical variables, with value levels and indices.

Variable	value	index
Age		
	young	0
	middle	1
	old	2
Marital status		
	married	0
	unmarried	1
Gender		
	male	0
	female	1

2 Contingency tables and missing data

A contingency table is a list of counts of value combinations for several categorical variables. The most informative way to represent a contingency table is as a multidimensional array or tensor. Each index of the tensor corresponds to one variable on a questionaire, and each position in the array corresponds to one combination of variable values. For computational purposes, it can be more convenient to represent contingency tables as a one-dimensional array, or vector. Specifically, this representation allows the calculation of marginals to be represented as a simple matrix-vector multiplication.

The purpose of the following paragraphs is to describe the connection between the two representations, and to give an explicit matrix representation of the map which computes marginals of a contingency table. Next, the missing data problem is described in these terms. As a service to the reader, Appendix A lists some basic properties of tensor products and direct sums of finite dimensional vector spaces.

2.1 Contingency tables

Consider a vector $D = (D_1, D_2, \ldots, D_m)$ of m random categorical variables. Each variable D_j can take on d_j values, giving $d = \Pi_{j=1}^{m} d_j$ possible realizations. For reasons pointed out in the next subsection, it is convenient to label the possible values of D_j with an index i_j, taking values $0 \leq i_j \leq d_j - 1$. The possible realizations of D can be then identified with a multi-index $I = (i_1, i_2, \ldots, i_m)$. When every random variable D_j corresponds to a question in a questionary, every possible I corresponds to a combination of answers given by a single re-

Table 2. Properties of respondents s for the variables in Table 1. The corresponding multi-index I is given in the last column.

s	Age	Marital status	Gender	I
1	middle	married	female	$(1,0,1)$
2	old	unmarried	male	$(2,0,1)$
3	middle	married	female	$(1,0,1)$

spondent. As an example, consider the variables in Table 1. The variables Age, Marital status, and Gender give rise to a $3 \times 2 \times 2$ contingency table, with the indices indicated in the 3rd column. In this example, an old unmarried male corresponds to $I = (2, 1, 0)$.

To construct a contingency table for multiple respondents, one basically counts the ocurrences of different values of I. This can be done formally by associating with each possible realization of D, a basis vector $\vec{e}_I$ of the tensor product space $\otimes_{i=1}^{m} \mathbb{Z}^{d_i}$ in the standard representation. That is, $\vec{e}_I$ is a tensor with coefficients $e_{I'}$ where $e_{I'} = 1$ if $I = I'$ and zero otherwise. For example, the old unmarried male mentioned above is associated with the following basis tensor of $\mathbb{Z}^3 \otimes \mathbb{Z}^2 \otimes \mathbb{Z}^2$:

$$\vec{e}_{210} = \left[\begin{pmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 1 \end{pmatrix}, \begin{pmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{pmatrix} \right]. \tag{1}$$

Here, the commas and brackets are left out in the explicit subscript of $\vec{e}_I$ to avoid cluttering of symbols.

A contingency table y, corresponding to complete answers in a survey S is now constructed by adding the basis tensors corresponding to each respondent s:

$$y = \sum_{s \in S} \vec{e}_I(s) = \sum_I y_I \vec{e}_I. \tag{2}$$

The coefficients y_I denote the number of times each combination of answers has occurred in the survey: they are the entries of the contingency table. Explicitly,

$$y = \left[\begin{pmatrix} y_{000} & y_{010} \\ y_{100} & y_{110} \\ y_{200} & y_{210} \end{pmatrix}, \begin{pmatrix} y_{001} & y_{011} \\ y_{101} & y_{111} \\ y_{201} & y_{211} \end{pmatrix} \right]. \tag{3}$$

As an example, consider the records given in Table 2. The contingency table corresponding to the three respondents is calculated by:

$$y \;\; = \;\; \vec{e}_I(1) + \vec{e}_I(2) + \vec{e}_I(3)$$

$$
\begin{aligned}
&= \vec{e}_{101} + \vec{e}_{210} + \vec{e}_{101} \\
&= 2\vec{e}_{101} + \vec{e}_{210}. \qquad (4)
\end{aligned}
$$

Explicitly, this reads

$$
\begin{aligned}
y &= 2\left[\begin{pmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{pmatrix}, \begin{pmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 0 \end{pmatrix}\right] + \left[\begin{pmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 1 \end{pmatrix}, \begin{pmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{pmatrix}\right] \\
&= \left[\begin{pmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 1 \end{pmatrix}, \begin{pmatrix} 0 & 0 \\ 2 & 0 \\ 0 & 0 \end{pmatrix}\right]. \qquad (5)
\end{aligned}
$$

2.2 Marginals

A marginal is obtained by summing over one or more indices of a contingency table. Since in this work we will be concerned with item nonresponse where at most one item per record is missing (to be discussed in the next subsection), we will only consider marginals obtained by summing over a single index. For a contingency table of dimension $d_1 \times d_2 \times \ldots \times d_m$, summing over i_j yields a marginal which has dimension $d^{(j)} = d/d_j$. There are m such marginals $b^{(j)}$ with entries $b^{(j)}_{I\backslash j}$ given by:

$$
b^{(j)}_{I\backslash j} = \sum_{i_j=0}^{d_j-1} y_{i_1,i_2,\ldots,i_j,\ldots,i_m}, \qquad (6)
$$

where $I\backslash j = (i_1, i_2, \ldots, i_{j-1}, i_{j+1}, i_{j+2}, \ldots, i_m)$. For example, summing over a single index in the contingency table of Eq. (5) yields the marginals

$$
\begin{aligned}
b^{(1)} &= \left[\begin{pmatrix} b^{(1)}_{00} & b^{(1)}_{10} \end{pmatrix}, \begin{pmatrix} b^{(1)}_{01} & b^{(1)}_{11} \end{pmatrix}\right] = \left[\begin{pmatrix} 0 & 1 \end{pmatrix}, \begin{pmatrix} 2 & 0 \end{pmatrix}\right] \qquad (7) \\
b^{(2)} &= \left[\begin{pmatrix} b^{(2)}_{00} \\ b^{(2)}_{10} \\ b^{(2)}_{20} \end{pmatrix}, \begin{pmatrix} b^{(2)}_{01} \\ b^{(2)}_{11} \\ b^{(2)}_{21} \end{pmatrix}\right] = \left[\begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}, \begin{pmatrix} 0 \\ 2 \\ 0 \end{pmatrix}\right] \qquad (8) \\
b^{(3)} &= \begin{pmatrix} b^{(3)}_{00} & b^{(3)}_{01} \\ b^{(3)}_{10} & b^{(3)}_{11} \\ b^{(3)}_{20} & b^{(3)}_{21} \end{pmatrix} = \begin{pmatrix} 0 & 0 \\ 2 & 0 \\ 0 & 1 \end{pmatrix}. \qquad (9)
\end{aligned}
$$

It is possible to construct a linear map which computes the marginals of a table. To do so, note first that each marginal $b^{(j)}$ is an element of the tensor product $\otimes_{i\neq j}\mathbb{Z}^{d_i}$, which has basis vectors $\vec{e}_{I\backslash j}$. Using Eq. (2), is not difficult to show that the linear map $A^{(j)}$, sending $\vec{e}_I$ to $\vec{e}_{I\backslash j}$ computes marginal $b^{(j)}$ when applied to contingency table y. Namely

$$
A^{(j)}y = \sum_{i_1\ldots i_j\ldots i_m} y_{i_1\ldots i_j\ldots i_m} A^{(j)}\vec{e}_{i_1\ldots i_j\ldots i_m}
$$

$$= \sum_{i_1 \ldots i_{j-1}, i_{j+1}, \ldots i_m} \left(\sum_{i_j} y_{i_1 \ldots i_j \ldots i_m} \right) \vec{e}_{i_1 \ldots i_{j-1}, i_{j+1}, \ldots i_m}$$
$$= \sum_{I \backslash j} b^{(j)}_{I \backslash j} \vec{e}_{I \backslash j} = b^{(j)}. \tag{10}$$

Secondly, note that each realization of D contributes to every marginal. For example, the respondent in Eq. (1) adds to the marginals $b^{(1)}_{10}$, $b^{(2)}_{20}$ and $b^{(3)}_{21}$, which stand for unmarried males, old males, and old unmarried individuals respectively. Combining the marginals in a direct sum $b = \oplus_{j=1}^{m} b^{(j)}$, we can compute all marginals of y with:

$$Ay = b, \quad \text{where} \quad A\vec{e}_I = \oplus_{j=1}^{m} \vec{e}_{I \backslash j}. \tag{11}$$

A representation for A can be obtained by switching from the tensor representation of y and b to a vector representation so A becomes a matrix. The vector representation of y is obtained by regarding all entries y_I as elements y_t of a d-dimensional column vector. Here, t is the index obtained by ordering all possible indices I in reverse lexicographical order (first index running first), with $0 \leq t \leq d-1$. In the example this means that

$$y = \begin{pmatrix} y_0 \\ y_1 \\ \vdots \\ y_8 \\ \vdots \\ y_{11} \end{pmatrix} = \begin{pmatrix} y_{000} \\ y_{100} \\ \vdots \\ y_{201} \\ \vdots \\ y_{211} \end{pmatrix}. \tag{12}$$

The following equation relates the vector index t to the multi-index I:

$$t(I) = \sum_{j=1}^{m} i_j \Pi_{k=1}^{j-1} d_k, \quad \text{with} \quad \Pi_{k=1}^{0} d_k \equiv 1. \tag{13}$$

The inverse relation reads

$$I(t) = (i_1(t), i_2(t), \ldots, i_m(t)),$$

with

$$i_j(t) = \left(t \,\mathsf{div}\, \Pi_{k=1}^{j-1} d_k \right) \,\mathsf{mod}\, d_j. \tag{14}$$

Here, div and mod denote integer division and remainder upon division respectively. Equations (13) and (14) can be understood by thinking of the multi-index $I = (i_1, i_2, \ldots, i_m)$ as a positional number system, indexing the numbers $t = 0, 1, \ldots, d-1$. The form of these relations depends on indices t and i_j running from 0 up.

The vector representation of the complete marginal b is given by the direct sum of the vector representations of the marginals $b^{(j)}$. So in the example, $b^{(1)}_{00}$ corresponds to b_0 in the vector representation and $b^{(3)}_{11}$ corresponds to b_{15}. The equation that relates the entries $b^{(j)}_{I\backslash j} = b_t$ is given by:

$$t(I,j) = \sum_{k=1}^{j-1} d^{(k)} + t(I\backslash j), \quad \text{with} \quad \sum_{k=1}^{0} d^{(k)} \equiv 0. \tag{15}$$

Here, $d^{(k)}$ is the dimension of the kth marginal and $t(I\backslash j)$ is obtained by restricting Eq. (13) to $I\backslash j$. Using Eqs. (14) and (15) we can now state the explicit form of the matrix representation of A:

$$A_{kl} = \begin{cases} 1 & \text{if} \quad k = t(I(l),j) \quad \text{for any} \quad j \in \{1,2,\ldots,m\} \\ 0 & \text{otherwise.} \end{cases} \tag{16}$$

Here, k runs from 0 to $\sum_{j=1}^{m} d^{(j)}$, the number of entries in b, and $0 \leq l \leq d-1$.

Having defined relations (13) - (15), both single indexing t and multi-indexing I of tensorial objects will be used in the rest of the paper without mentioning.

2.3 Missing data, stochastic imputation, and edit restrictions

Consider a dataset with n^{tot} records, of which n^{full} are complete, and n lack precisely one item. Assume that this is the only missing data pattern, so $n^{\mathsf{full}} + n = n^{\mathsf{tot}}$. For the set of complete records, a contingency table y^{full} can be constructed. Denote the complete table corresponding with the unknown complete dataset by

$$y^{\mathsf{tot}} = y^{\mathsf{full}} + x, \tag{17}$$

and the corresponding complete marginal by b^{tot}. Here, x represents the missing part of the table.

The solution to the imputation problem is to find an estimate $\hat{x}$ so that y^{tot} can be estimated with

$$\hat{y}^{\mathsf{tot}} = y^{\mathsf{full}} + \hat{x} \tag{18}$$

Here, x is considered a realization of a random variable X, taking values in a probability space Ω_X. A reasonable imputation is then to choose the most probable value of x under a specified probability measure, which can be estimated from complete data for example. Alternatively, a random x can be drawn from Ω_X to reflect the variability of the incomplete sample. In the remainder of this paragraph Ω_X will be specified. The probability distribution and sampling algorithms are discussed in the subsequent sections.

First, since the number of partially filled records is known, there is a restriction

$$\sum_{t=0}^{d-1} x_t = n \quad \text{with} \quad x_t \geq 0, \tag{19}$$

which ensures that Ω_X is finite. Second, restrictions can be derived for the marginals of x. Using Eq. (17), the marginals for y^{tot} can be written as:

$$Ay^{\mathsf{tot}} = b^{\mathsf{full}} + Ax. \tag{20}$$

The second term cannot be computed, since the partially filled records cannot be used to construct the contingency table x. However, it is possible to construct partial marginals b^{part} based on the information in the partially filled records. For example, consider again the variables in Table 1. If there are 10 married males of unknown age in the dataset, this gives $b_{00}^{(1)\mathsf{part}} = 10$. Combining al the known parts of the records with missing items into b^{part}, the missing part of the marginals b^{miss} is defined by

$$Ax = b^{\mathsf{part}} + b^{\mathsf{miss}}. \tag{21}$$

Since $b^{\mathsf{miss}} \geq 0$ by Eqs. (16) and (19), the following set of inequalities is obtained:

$$Ax \geq b^{\mathsf{part}}, \tag{22}$$

where $\geq$ is interpreted elementwise.

Equations (19) and (22) are enough to define Ω_X in the case when there are no edit restrictions on value combinations. In the case of categorical variables, edit restrictions limit the set of allowed values for x (and y), which can be expressed as

$$x_I = 0, \quad \text{for} \quad I \in \mathfrak{I}, \tag{23}$$

with $\mathfrak{I}$ some set of indices. In this context edit restrictions are often refered to as structural zeros. For example, if the edit restriction "young people cannot be married" is imposed on the variables in Table 1, $\mathfrak{I}$ can be written as

$$\mathfrak{I} = \{(0,0,0),(0,0,1)\} \quad \text{or equivalently} \quad \mathfrak{I} = \{0,6\}. \tag{24}$$

Here, Eq. (13) was used to compute the single index representation from the multi-index notation on the left.

Combining the edit restrictions with the conditions in Eqs. (19) and (22), Ω_X is given by

$$\Omega_X = \{x \in \mathbb{Z}_{\geq 0}^d | \Sigma_{t=1}^d x_t = n \wedge Ax \geq b^{\mathsf{part}} \wedge x_t = 0 \ \forall t \in \mathfrak{I}\}. \tag{25}$$

The main problem now is to draw elements from Ω_X according to some probability distribution. Although the problem of counting the elements of Ω_X is not solved in general, it is understood that the number of elements becomes too large to list them in practice. For example, when $b^{\mathsf{part}} = 0$ and $\mathfrak{I} = \varnothing$, the number of distinct elements $|\Omega_X| = \binom{d+n-1}{n}$. In most practical situations this means that representation of Ω_X in computer memory is not an option. One way around this is to generate a possible solution $x(0)$ as starting point for a random walk on Ω_X. During the walk, new elements $x(1), x(2), \ldots$ are generated by taking randomly chosen elementary steps, while the correlation between the current value $x(\tau)$ and $x(0)$ decreases. After a sufficient number of steps (the burn-in time) are taken, $x(\tau)$ can be considered a random sample from Ω_X.

3 Statistical models

3.1 Distribution on Ω_X

Based on for example historical data or complete records, one can assign probabilities to all possible solutions x to the imputation problem. When there are no restrictions on the entries of x except the one given in Eq. (19), the probability distribution is given by the multinomial distribution

$$\mathcal{M}(x|\theta) = \frac{n!}{x_0!x_1!\cdots x_{d-1}!}\theta_0^{x_0}\theta_1^{x_1}\cdots\theta_{d-1}^{x_{d-1}}, \tag{26}$$

where $\theta_t = \mathbb{P}[D = \vec{e}_{I(t)}]$ are the cell probabilities. Remember that we associated each realization of D with a basis vector $\vec{e}_I$ in paragraph 2.1. When there are edit restrictions given by a set of indices $\mathfrak{I}$ as in Eq. (23), we get the restricted multinomial distribution $\mathcal{M}(x|\theta, \mathfrak{I})$ simply by deleting x_I and θ_I from Eq. (26) for all $I \in \mathfrak{I}$. For the distribution on Ω_X a truncated distribution can be used, given by

$$p(x) = \mathbb{P}(X = x|\theta, b^{\mathsf{part}}, \mathfrak{I}) = \begin{cases} \mathcal{M}(x|\theta, \mathfrak{I}) & \text{if} \quad Ax \geq b^{\mathsf{part}} \\ 0 & \text{otherwise.} \end{cases} \tag{27}$$

Note that $p(x)$ is not normalized; it is a pseudodistribution. This poses no problem since it can be shown (see Section 4) that only the ratios $p(x)/p(x')$ with $x, x' \in \Omega_X$ are needed to perform random walks. In general, the parameter vector θ is not known and needs to be estimated, so $p(x)$ is replaced by an estimate $\hat{p}(x)$ corresponding to an estimate $\hat{\theta}$.

3.2 Modes of the multinomial distribution

As stated in Section 2.3, a reasonable imputation $\hat{x}$ can be to take (one of) the mode(s) of $p(x)$. There is no explicit expression for the integer modes of the

multinomial distribution, although an algorithm which finds all the modes has been proposed by le Gall (2003).

The modes of $\mathcal{M}$ are those values of x which maximize $\mathcal{M}(x|\theta)$ for a given θ under the condition that $\sum_t x_t = n$. Analytical continuation of $\mathcal{M}$ such that $x \in \mathbb{R}^d_{\geq 0}$ gives the log-likelihood function:

$$\ell(x|\theta) = -\sum_{t=0}^{d-1} \ln \Gamma(x_t + 1) + \sum_{t=0}^{d-1} x_t \ln(\theta_t) \tag{28}$$

Imposing Eq. (19) yields the Lagrange function

$$L(x, \lambda) = \ell(x|\theta) + \lambda \left(\sum_{t=0}^{d-1} x_t - n \right). \tag{29}$$

Equalizing derivatives with respect to x_t and λ to zero yields

$$x_t = \psi^{-1}(\ln \theta_t + \lambda) - 1 \tag{30}$$

for $0 \leq t \leq d-1$. Here $\psi(z) = \Gamma'(z)/\Gamma(z)$ is the digamma function [see for example Abramowitz and Stegun (1972)] and $\psi^{-1}[\psi(z)] = z$ for nonnegative real z. The constant λ must be chosen so that

$$\sum_t \psi^{-1}(\ln \theta_t + \lambda) = n + d. \tag{31}$$

The integer modes are the $x \in \Omega_X$ which are nearest (in the Euclidian sense) to the real solution. Integer solutions need not be unique. Consider for example $d = 2$, $n = 1$ and $\theta = (1/2, 1/2)$. There are two possible states, namely $x = (1, 0)$ and $x = (0, 1)$, which both have probability $\mathcal{M}(x|\theta) = 1/2$ which is also the discrete maximum. For comparison, the solution on the real plane is given by $x = (0.5, 0.5)$, which has probability $\mathcal{M}(x|\theta) = \frac{1}{2}\Gamma(3/2)^{-2} \approx 0.64$.

As stated in Section 2.3 (see also Eq. 18), one reasonable imputation is to estimate x as the most probable value given a probability measure over its possible values. When there are multiple integer modes, in distribution (27) the statistical model does not distinguish a single maximum probability solution to the imputation problem. In that case any of the modes in Ω_X is equally acceptable.

In this work, an "upward random walk" is used to approach the modes of the distribution. That is, starting from a value $x(0) \in \Omega_X$ a random step is picked, and the step is taken only if it increases $p(x)$. After a while the average number of steps taken will decrease, indicating proximity to one of the modes.

4 Markov chain Monte Carlo and optimization

In the following subsection the central algorithms used to generate random walks are given. Some implementation issues concerning performance are also discussed. In appendices B and C more detail on the background of the algorithms is given.

4.1 Sampling algorithms

Two basic algorithms have been implemented to generate random samples from Ω_X. Both these algorithms perform random walks on Ω_X. They are based on repeating two steps: (1) starting from a value $x \in \Omega_X$, draw a (set of) vector(s) v uniformly such that $x + v \in \Omega_X$. (2) with a probability proportional to $p(x + v)$ [Eq. (27)] actually take the step.

In drawing the steps v, we have to take in to account the conditions in Eqs. (19), (22) and the possibility of edit restrictions. It is obvious that any vector v^{rs} with coefficients $v^{rs}_t = \delta_{rt} - \delta_{st}$ obeys

$$\sum_{t=0}^{d-1}(x + v^{rs})_t = \sum_{t=0}^{d-1} x_t = n, \tag{32}$$

so condition (19) is satisfied. Here, δ_{ij} is the kronecker delta defined by $\delta_{ij} = 1$ if $i = j$ and zero otherwise. Theorem 3.1 of Diaconis and Sturmfels (1998) ensures that the set of vectors v^{rs}, with $r \neq s$ and $0 \leq r, s \leq d-1$ are sufficient to produce irreducible markov chains on Ω_X (they form a markov basis, cf. Appendix B. The following algorithm is based on the Metropolis-Hastings sampler [Lemma 2.1 of Diaconis and Sturmfels (1998)], and it incorporates edit restrictions as well as condition (22).

Algorithm 1: Metropolis-Hastings sampler

Input: $x \in \Omega_X$, A, p, b^{part}

Output: N^{th} value x of a markov chain on Ω_X.

$n := 0$;
while $n < N$ **do**
    Draw (r, s) uniformly without replacement from $[d-1]\backslash\mathfrak{I}$;
    if $A(x + v^{rs}) \geq b^{\mathsf{part}} \wedge x + v^{rs} \geq 0$ **then**
        $n := n + 1$;
        $x := x + v^{rs}$ with probability $\min\{p(x + v^{rs})/p(x), 1\}$;

Here, we use the notation $[d-1] = \{0, 1, \ldots, d-1\}$. Drawing two indices r and s from $[d-1]\backslash\mathfrak{I}$ ensures that entries x_I remain unchanged for all $I \in \mathfrak{I}$. Thus, edit restrictions are included trivially here. In the current implementation edits

are encoded in θ, by setting $\theta_I = 0$ for all $I \in \mathfrak{I}$. The **if**-statement in line 5 ensures that the partial marginal conditions are included and no negative values for x_I can occur. The algorithm is set up so that steps which would lead to an x outside of Ω_X (invalid steps) are not counted. The ratio of valid *versus* invalid steps drawn depends in general on the values of p and the elements of b^{part}, see also section 5. It is also assumed here that the set of structural zeros $\mathcal{I}$ is small enough so that it can be stored explicitly in computer memory.

The Metropolis-Hastings algorithm was implemented as a C-routine for speed and called from the R statistical environment to facilitate analysis (see R-Dev). The routines were optimized for speed as much as possible. For example, to check the linear condition $A(x + v^{rs}) \geq b^{\mathsf{part}}$ we keep track of the current value for $b = Ax$ and only check and update the entries of x and b affected by the operation $x_r \mapsto x_r + 1$ and $x_s \mapsto x_s - 1$. Also note that the ratio $p(x+v^{rs})/p(x)$ can be calculated as:

$$\frac{p(x + v^{rs})}{p(x)} = \frac{x_s}{x_r + 1} \frac{\theta_r}{\theta_s}, \tag{33}$$

which significantly enhances numerical stability during computation.

The steps v^{rs} described here are the smallest possible, and rather large chain lengths ($N \sim 10^5$) are necessary to ensure convergence. It is not difficult to see that for integer k, the step kv^{rs} also conserves condition (19) and it was already shown in Lemma 2.2 of Diaconis and Sturmfels (1998) that a random walk based on these steps can be implemented too. The resulting algorithm is a Gibbs sampler and reads as follows.

1 **Algorithm 2:** Gibbs sampler

Input: $x \in \Omega_X$, A, p, b^{part}

Output: N^{th} value x of a markov chain on Ω_X.

2 $n := 0$;

3 **while** $n < N$ **do**

4 Draw (r, s) uniformly without replacement from $[d-1]\backslash\mathfrak{I}$;

5 **if** $x_r > 0 \vee x_s > 0$ **then**

6 $n := n + 1$;

7 Determine k^{min} and k^{max} such that $x + kv^{rs} \in \Omega_X$ for $k^{\mathsf{min}} \leq k \leq k^{\mathsf{max}}$;

9 Draw k from $\{k^{\mathsf{min}}, k^{\mathsf{min}} + 1, \ldots, k^{\mathsf{max}}\}$ with probability proportional to $q(x, k) = p(x + kv^{rs})/p(x)$;

10 $x := x + kv^{rs}$;

Here, the values of k^{min} and $k^{\max}$ are determined by the conditions in Eq. (22) and that $x \geq 0$. The only issue worth mentioning here is that in practice the set of ratios $q(x, k)$, $k^{\mathsf{min}} \leq k \leq k^{\mathsf{max}}$ suffers from numerical instabilities, and

therefore has to be limited further. The ratios are computed using the following recursion relation:

$$q(x,0) = 1 \quad \text{and} \quad q(x,k) = \frac{x_s - k + 1}{x_r + k} \frac{\theta_r}{\theta_s} q(x,k-1), \tag{34}$$

which can be derived from Eq. (27). In practice, the ratio x_s/x_r can be in the order of $10^{\pm 2}$. Combined with a large difference $k^{\mathsf{max}} - k^{\mathsf{min}}$, this leads to values $q(x,k)$ out of the range which can be represented by a computer (Inf or NaN). Large (or small) ratios θ_r/θ_s can have a similar effect. Moreover, to draw k (line 9), the cumulative distribution $Q(x,k) = \sum_{k=k^{\mathsf{min}}}^{k} q(x,k)/Q(x,k^{\mathsf{max}})$ is computed. The value of k is determined by drawing a value u from $\mathcal{U}(0,1)$ and determining the smallest value of k for which $Q(x,k) \geq u$. Here we have to take into account that on a 32-bit computer, double precision numbers are represented by 53 bits, so a uniform random number generator can produce a maximum of $2^{53} \approx 10^{16}$ different numbers between 0 and 1. It is therefore meaningless to have ratios $|^{10}\log[q(x,k^{\mathsf{max}})/q(x,k^{\mathsf{min}})]| \geq 16$. To prevent numerical instabilities, the range of stepsizes $k^{\mathsf{max}} - k^{\mathsf{min}}$ is limited further by demanding that $|^{10}\log q(x,k)| \geq \kappa$, where κ is an adjustable threshold value. To be absolutely safe (on a 32 bit system) a value of $\kappa = 8$ should be used. In practice a value of $\kappa = 16$ gives good results and allows somewhat larger stepsizes on average.

4.2 Upward random walk

To approximate one of the modes of the distribution over Ω_X, an algorithm based on the Metropolis-Hastings sampler is used.

1 **Algorithm 3:** Upward random walk

Input: $x \in \Omega_X$, A, p, b^{part}

Output: N^{th} value x of a random upward walk on Ω_X.

2 $n := 0$;

3 **while** $n < N$ **do**

4 $\quad n := n + 1$;

5 $\quad$ Draw (r,s) uniformly without replacement from $[d-1]\backslash\mathfrak{I}$;

6 $\quad$ Determine $\max_k\{p(x + kv^{rs})\}$ with $k \in \{0, \pm 1\}$ and $x + kv^{rs} \in \Omega_X$;

7 $\quad x := x + kv^{rs}$;

The algorithm randomly chooses a valid step v^{rs}. A step is taken if either $p(x + v^{rs})$ or $p(x - v^{rs})$ is larger than $p(x)$. This is certainly not the most efficient algorithm to approximate a mode although for large enough N it will converge to a mode. For moderately large N, a reasonable solution close to a maximum is obtained. The convergence behaviour as a function of N is

studied in Section 5. The efficiency of the walk decreases near the mode since the number of $p(x)$-increasing steps decreases near an optimum. The algorithm is adopted here for ease of implementation, postponing a better solution to future work.

5 Numerical examples

The algorithms of Sections 4.1 and 4.2 were tested on two different datasets. The first dataset is publicly available data from the US Labour Force Survey (USLFS), made available via the machine learning archive of the University of California-Irvine (UCI-MLR). The advantage of using a published dataset is that it facilitates (future) comparison of methods by different authors. The second dataset is a set of records from the Dutch Integrated System of Social Statistics (ISSS). In the next subsection convergence behaviour is studied by introducing item nonresponse in the USLFS data. In Section 5.2 convergence properties are studied as a function of several parameters, such as nonresponse fraction, number of variables and chain length.

5.1 Convergence properties with public USLFS data

The USLFS data set consists of 32 560 records with 15 categorical variables. Of these, four variables were chosen, namely

$$(\text{workclass})_8 \times (\text{marital status})_7 \times (\text{sex})_2 \times (\text{race})_5, \tag{35}$$

where the subscripts designate the number of levels d_i for each variable. The variable workclass actually has nine categories, but records with workclass="?" were omitted. Records with (multiple) item nonresponse were also removed, yielding a full dataset of $n^{\mathsf{tot}} = 30\,724$ complete records. Of these records 3 073 records were drawn randomly, and in each of the drawn records, a single item was set empty. Thus we have $n^{\mathsf{full}} = 27\,651$, n =3 074, and the dimension of the table is $d = 8 \cdot 7 \cdot 2 \cdot 5 = 560$.

The probability distribution of Eq. (27) was parameterized with a simple model based on complete-record frequencies:

$$\hat{\theta}_t = \frac{y_t^{\mathsf{full}} + \eta}{\sum_{t=0}^{d-1}(y_t^{\mathsf{full}} + \eta)}. \tag{36}$$

Here, $\eta = 10^{-3}$ is a small number, added to have nonzero probability for cells without any observation. In this model it is assumed that there are no structural zeros.

To study convergence properties of the Metropolis-Hastings and the Gibbs sampler, random solutions were generated by taking the Nth value $x(N)$ of a

markov chain. markov chains of various lengths ($N = 10^k$, $k = 1, 2, \ldots, 6$) were generated, all starting from the same pre-imputed startvalue $x(0)$. The startvalue was generated by sequential random imputation of the records with item nonresponse and computing the contingency table. A markov chain based on the Gibbs sampler of Algorithm 2 was used to generate $x(0)$ from this table.

To study convergence properties, consider a distance function on Ω_X, based on the L^1-norm:

$$\mathsf{dist}(x, x') = \frac{1}{2n}||x - x'||_1 = \frac{1}{2n}\sum_{t=0}^{d-1}|x_t - x'_t|. \tag{37}$$

where $n = \sum_{t=0}^{d-1} x_t = \sum_{t=0}^{d-1} x'$. It can be seen that $\mathsf{dist}(x, x')$ scales between 0 and 1.

If a chain is long enough, producing multiple drawings should average out to the expected value $\mathsf{E}(x)$. Therefore, a measure of convergence for a markov chain of length N is given by $\mathsf{dist}(\mathsf{E}[x(N)], \mathsf{E}[x])$. Unfortunately, neither $\mathsf{E}[x(N)]$ nor $\mathsf{E}[x]$ can be calculated precisely. Remember that $\mathsf{E}[x] = \sum_{x\in\Omega_X} xp(x)$, which is not a feasible calculation. However, since $p(x)$ is nearly a multinomial distribution, we will use the approximation $\mathsf{E}[x] \approx n\hat{\theta}$. The expectation of $x(N)$ is estimated by averaging over a number of samples: $\hat{\mathsf{E}}[x(N)] = \bar{x}(N)$. We are now able to define a convergence parameter C_N as:

$$C_N = \mathsf{dist}[\bar{x}(N), n\hat{\theta}]. \tag{38}$$

This parameter will converge to $\mathsf{dist}(\mathsf{E}[x], n\hat{\theta})$ as the number of samples and the chain length increases.

In Figure 1 the convergence parameter is shown as a function of number of samples (horizontal axis) and the markov chain length. It is clear that the Gibbs sampler (left panel) needs shorter chains to reach convergence than the Metropolis-Hastings sampler (right panel). A markov chain generated by 10^4 Gibbs steps ($\Box$) has the same level of convergence as the Metropolis-Hastings sampler at 10^5 steps ($\Diamond$). For the Gibbs sampler, increasing the chain length to 10^5 or 10^6 has little effect on convergence. The same holds for increasing the chain length of the Metropolis-Hastings sampler from 10^5 to 10^6.

Apart from chain lengts, the actual time it takes to generate the markov chains should be taken into account. In Table 3 the average relative runtimes for generating the markov chains are listed. The second and third column show runtime scaled to the MH-sampler for markov chain length $N = 10^3$. For comparison: on a AMD64 laptop at 1.8GHz running R under Linux, this calculation took 0.15 seconds (measured as user time by R).

Although the Gibbs sampler is about 10 times more efficient in terms of chain length, it is also roughly about 10 times slower. The reason is that for every

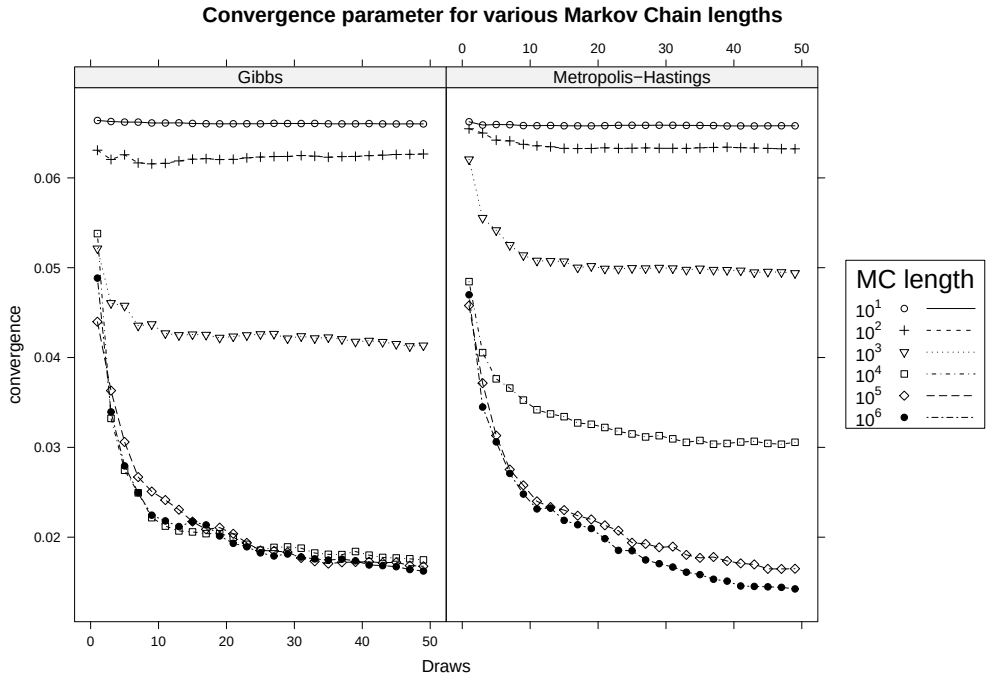

Figure 1. Convergence behaviour for drawing random samples to impute USLFS data at various markov chain lengths. The horizontal axis designates the number of drawings used to determine the convergence parameter of Eq. (38). The left panel shows results for the Gibbs sampler and the right panel shows results for the Metropolis-Hastings sampler. In terms of chain length, the Gibbs sampler is obviously more efficient.

step in the Gibbs sampler, a whole series of probability ratios $q(x, k)$ must be calculated, where for the Metropolis-Hastings algorithm only one probability ratio $p(x + v^{rs})/p(x)$ is computed at each step. Here, these effects cancel each other in terms of computational time. Taking 10^4 Gibbs steps takes about the same time as generating 10^5 Metropolis-Hastings steps.

It must be noted that the runtime also depends on the particular values of $\hat{\theta}_t$. For example, the USLFS data used here has 289 cells with no observations in y^{full} and these cells obtain a low probability for occupancy. This means that at runtime, a fair amount of cells are probably empty. Every step that would decrease the value of these cells is therefore invalid. The more cells are empty, the higher the chance is that invalid steps are drawn which have to be rejected, thereby increasing the time it takes to take a fixed number of MH-steps. This effect can be circumvented by setting $\eta = 0$ in Eq. (36), so $\theta_t = 0$ and x_t will be treated as a structural zero. Setting $\eta = 0$ does alter the model assumptions and therefore the interpretation of the imputation results.

When $\eta = 10^{-3}$, as in the previous example, the average number of invalid Metropolis-Hastings steps drawn before a markov chain with $N = 10^4$ is gen-

Table 3. Relative timings of the Metropolis-Hastings and the Gibbs sampler. Timing of the MH-sampler with $N = 1\,000$ is used as reference.

MC length N	MH	Gibbs	ratio
10	0.60	0.60	1.00
10^2	0.60	1.07	1.78
10^3	1.00	5.73	5.73
10^4	5.00	58.80	11.76
10^5	52.7	960	18.21
10^6	974	11233	11.53

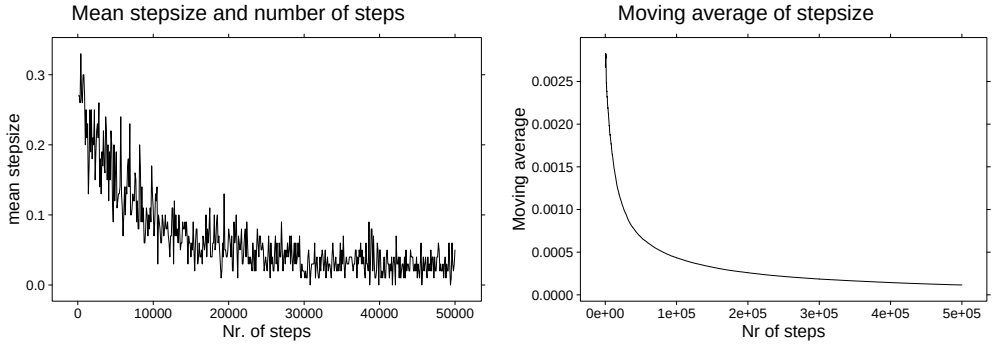

Figure 2. Convergence of the optimization algorithm as a function of the number of steps taken. The left panel shows the local mean stepsize over 100 Metropolis-Hastings steps (stepsize can vary between 1 and 0) for the first $5 \cdot 10^4$ steps. The right panel shows the moving average over the first $5 \cdot 10^5$ steps.

erated equals about 20 000. Twice as much as the length of the chain. The average stepsize over valid steps is 0.31 (the average stepsize is given by the number of steps taken divided by the number of valid steps drawn). When $\eta = 0$, all cells without observations are treated as structural zeros, and care must be taken to make sure that $x(0) \in \Omega_X$. In this case, the MH algorithm draws about 5 000 invalid steps and the average stepsize doubles to about 0.63. The increase in average stepsize is due to the lower number of empty cells.

Finally, convergence of the upwards random walk algorithm of Section 4.2 was tested. To test convergence, starting from a random point $x(0) \in \Omega_X$, every 100 steps the average stepsize (number of steps taken/100) was computed. The result is shown in the left panel of Figure 2. The average stepsize decreases with the length of the walk and approaches zero near a mode. In the right panel the moving average is shown over a larger number of steps. After $5 \cdot 10^5$ steps virtually no more steps are taken, indicating proximity to one of the modes.

5.2 Convergence properties with Dutch ISSS data

Data of the Dutch Integrated System of Social Surveys (ISSS[2]) was used to study convergence properties, timings and imputation quality of the algorithms described above. The ISSS is a large file containing data from administrative sources and (household) surveys. The following variables were selected from the file:

$$\begin{aligned}&(\text{educational level})_7 \times (\text{internet})_2 \times (\text{health})_5 \\ &\quad \times (\text{smoke2001})_2 \times (\text{church})_5 \times (\text{ethnic group})_7,\end{aligned} \tag{39}$$

where subscripts denote the number of levels for each variable. The same dataset has been used extensively by Cobben (2009) to study nonresponse effects. The complete file consists of 36 515 records. After omitting all records where the value of one of the above variables is "unknown" or missing, 6 476 records are left, which is what we use to test our algorithms. In all tests, either 617 ($\approx 10\%$) or 1 619 ($\approx 25\%$) of the records are chosen randomly and equipped with one missing item each, also at random.

5.2.1 Convergence and timing

Convergence of sampling algorithms was measured as described in Section 5.1. That is, missings are generated, and a simple imputation model is chosen, similar to the one in Eq. (36) with $\eta = 10^{-3}$. Convergence is measured with Eq. (38), where the number of variables, markov chain length, the number of missings and the sampling method are varied. markov chains were created for data with 2 variables: $(\text{educational level})_7 \times (\text{internet})_2$, 3 variables: $(\text{educational level})_7 \times (\text{internet})_2 \times (\text{health})_5$ and so on up to all six variables. The corresponding dimensions of the contingency tables are d =14, 70, 140, 700, 4900. The results are plotted in Fig. 3.

The end of a markov chain can be considered a random sample from (Ω_X, p) when convergence parameter C_N does not decrease anymore as a function of chain length N. It can be seen in Fig. 3 that larger chain lengths are necessary when the number of variables (and hence the dimension d of the table) increases. For small table dimensions, the Gibbs sampler clearly converges faster as function of chain length than the Metropolis-Hastings sampler. For larger table dimensions the difference between the two samplers is smaller.

In Fig. 4 and 5 the runtime as a function of chain length is plotted for different numbers of variables. The time is given in seconds, but the value will naturally depend on the hardware (here: a virtual pc running Windows 2000). As can be expected, both the Metropolis-Hastings sampler and the Gibbs sampler have

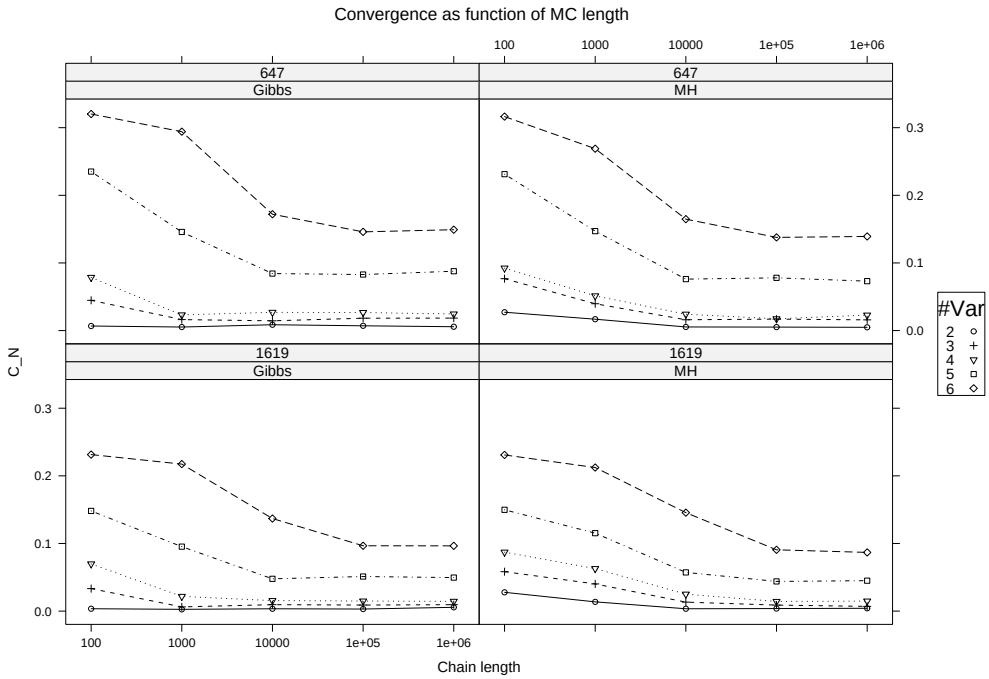

Figure 3. Convergence [See Eq. (38)] to $\mathsf{E}(x)$ *as a function of MC length, for different numbers of variables. 2 Variables relates to* $(\text{educational level})_7 \times (\text{internet})_2$*, 3 variables to* $(\text{educational level})_7 \times (\text{internet})_2 \times (\text{health})_5$ *and so on, see Eq. (39). The table dimensions d are 14, 70, 140, 700 and 4900.*

linear asymptotic time complexity as function of chain length. Apart from the dependence on chain length there is a constant overhead, needed to allocate memory, detect structural zeros, and creating a lookup table which holds the relation between indices in the contingency table and the marginals. This overhead increases with table dimension, which causes the deviation from linear behaviour at small chain lengths.

In Fig. 5 the same data as in Fig. 4 is shown, only now with the table dimension d as horizontal axis. The dependence of runtime on d is more complex. The Gibbs sampler shows a minimum runtime at $d = 700$ (5 variables), while the Metropolis-Hastings sampler increases monotonically with chain length (when no points are shown, the elapsed time is less than a millisecond). The minimum in the runtime of the Gibbs sampler is the consequence of two opposing effects. As mentioned in Section 5.1, the rate determining step for the Gibbs sampler is the calculation of the probability ratios $q(x, k)$. In this numerical experiment, the total number of observations is constant while the dimension of the table is varied. For small table dimensions (i.e. small number of variables), large stepsize ranges $k^{\min}$–$k^{\max}$ are possible since all observations are distributed over a small number of cells. As the table dimension increases, observations are distributed over more cells and smaller stepsize ranges are possible. Hence,

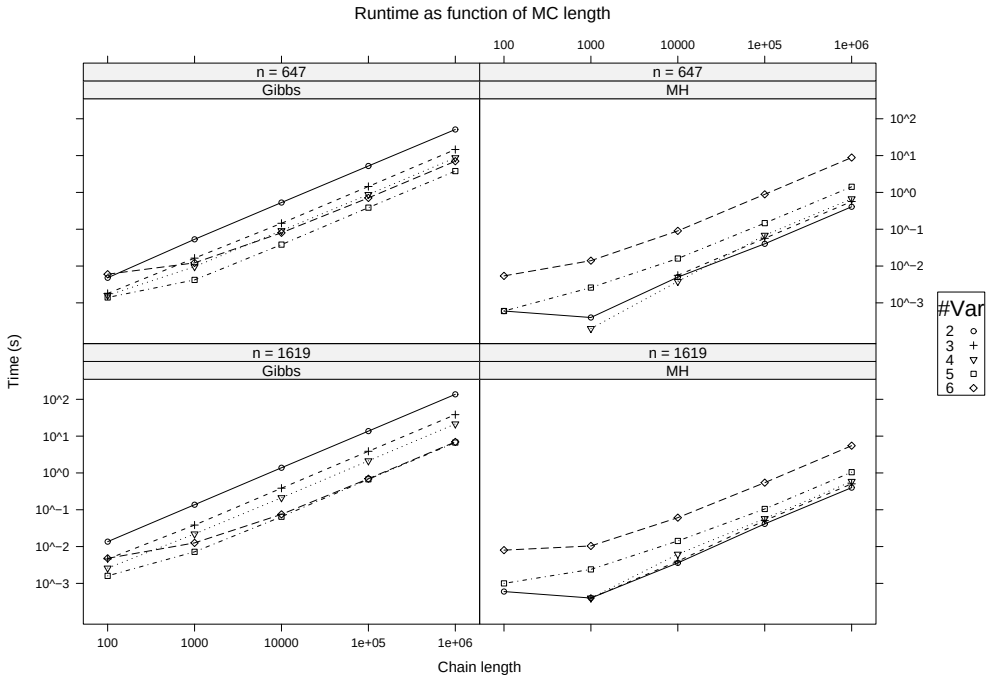

Figure 4. Runtime as function of chain length and number of variables. Missing points indicate a runtime shorter than a millisecond (which are rounded to zero).

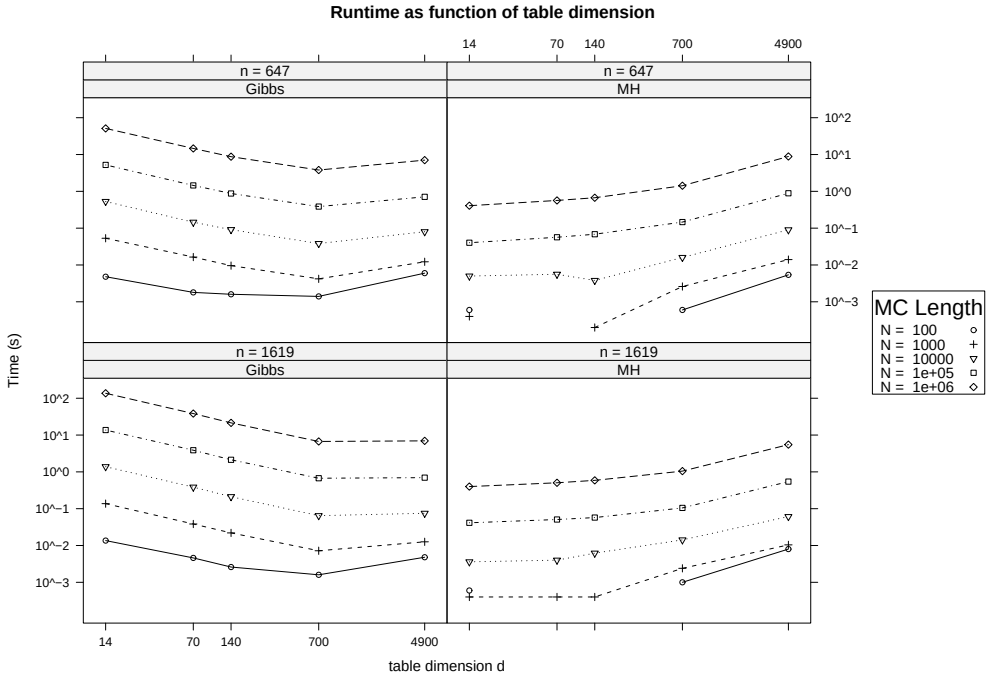

Figure 5. Runtime as function of table dimension. Same data as in Fig. 4.

Table 4. Table dimension d, number of observed empty cells ξ and number of invalid steps divided by markov chain length

		Invalid steps / chain length			
		n= 647		n=1619	
d	ξ	MH	Gibbs	MH	Gibbs
14	0	0.001	0.000	0.000	0.000
70	5	0.480	0.095	0.288	0.041
140	15	0.768	0.193	0.506	0.109
700	270	3.026	1.152	1.869	0.663
4900	4127	23.929	10.645	14.277	6.339

calculating $q(x, k)$ takes less time. The opposing effect is caused by an increasing number of observed empty cells as the table dimension increases. For the Gibbs sampler, any pair of empty cells $x_r = x_s = 0$ corresponds to an invalid step v^{rs}, so the probability of drawing invalid steps increases with table dimension, causing increased overhead.

In Table 4 the number of empty cells (ξ) is shown together with the number of invalid steps per chain length for the Metropolis-Hastings and the Gibbs sampler. It can be seen that the Gibbs sampler draws much less invalid steps than the Metropolis-Hastings sampler. This can be explained by a closer examination of Algorithms 1 and 2. Given a table x in the markov chain, the probability that the Metropolis-Hastings sampler draws an invalid step, is given by $\mathbb{P}[x_s = 0 \vee A(x + v^{rs}) < b^{\mathsf{part}}]$ while that same probability for the Gibbs sampler is given by $\mathbb{P}[x_r = x_s = 0 \vee A(x + v^{rs}) < b^{\mathsf{part}}]$. The latter is clearly smaller than or equal to the former.

5.2.2 Imputation quality

In this section, the upward random walk algorithm [See Algorithm 3] is used to generate (near) maximum likelihood imputations. Again, a number of $n = 647$ or $n = 1619$ records were chosen randomly, and equipped with one missing item. All six variables from Eq. (39) were taken into account. Next, the statistical model of Eq. (36) was parameterized using the remaining complete data and $\eta = 10^{-3}$.

To gain insight in the imputation quality, the real contingency table x^{real} (the table before the introduction of missing values) was also stored. In Fig. 6 the relative probability $p[x(\tau)]/p[x^{\mathsf{real}}]$ is shown as a function of the number of steps τ taken by the upward random walk algorithm. The relative probability converges in about 25 000 steps to a value much larger than one. This means that

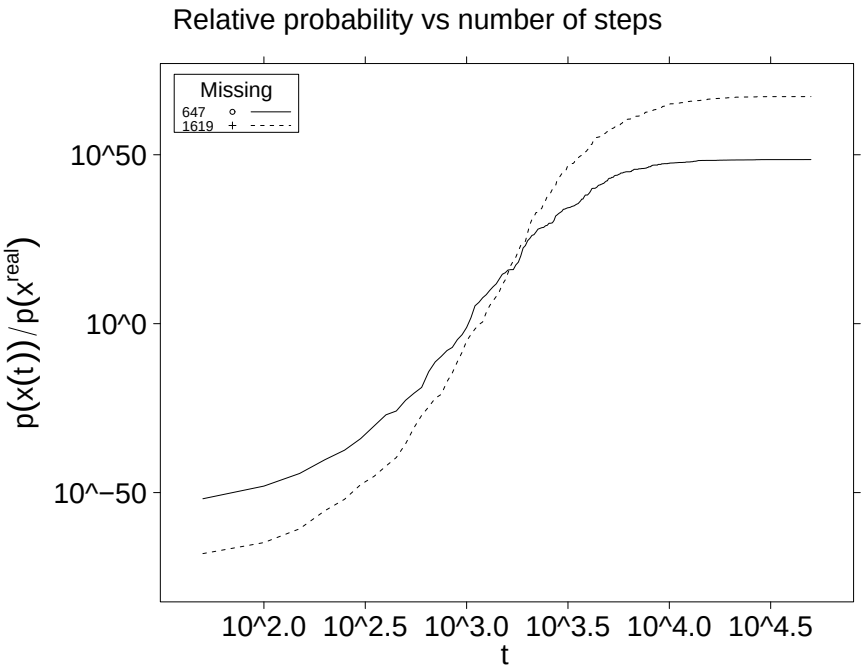

Figure 6. Relative probability of $x(\tau)$ with respect to the real value as a function of the number of steps in Algorithm 3.

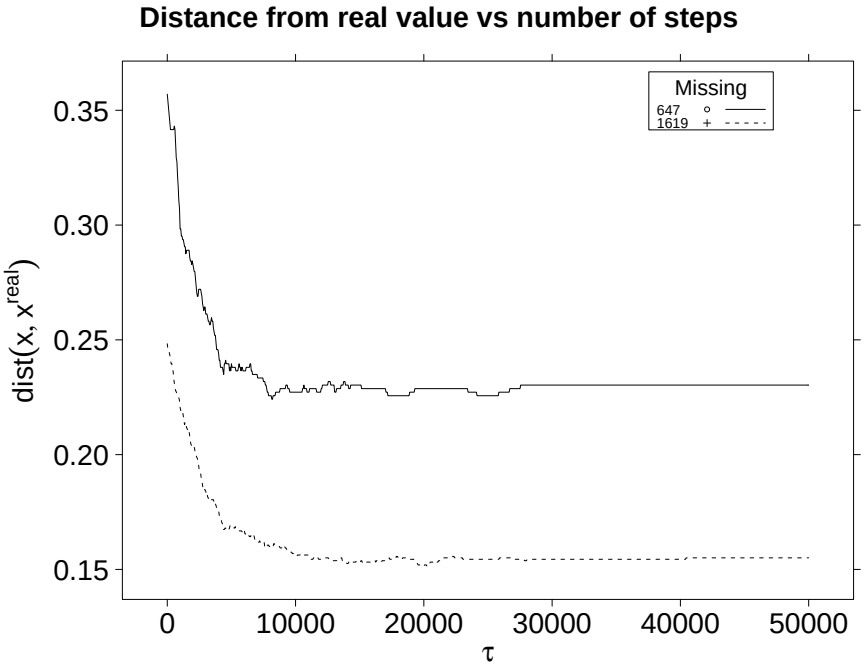

Figure 7. Distance between $x(\tau)$ and real value as a function of the number of steps in Algorithm 3.

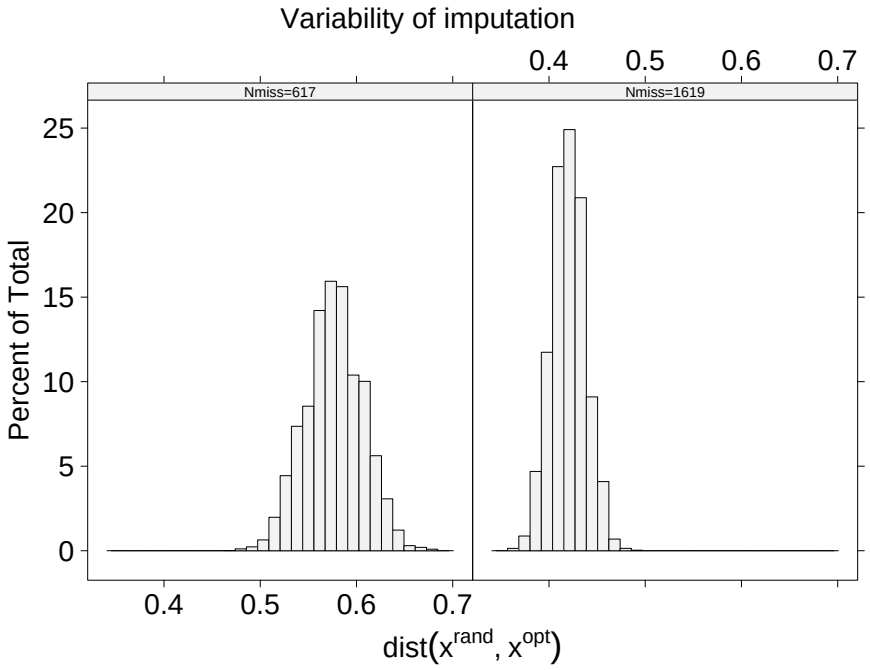

Figure 8. Distribution of distance between random and optimal imputation for 10^4 *random imputations.*

the most probable value, given the incomplete data, does not equal x^{real}. Even if the complete data was used, this can be the case, since for the multinomial distribution, the expected value in general does not equal the mode. In Fig. 7 the distance $\mathsf{dist}[x^{\mathsf{real}}, x(\tau)]$ as a function of the number of steps τ is shown (see Eq. 37). Note that the distance for the calculation with $n = 647$ cannot be compared with the distance for $n = 1619$, since the distances are defined on different probability spaces Ω_X. However, this measure could be used to compare different statistical models at the same number of missings, for example. The distance between the real and estimated value decreases initially as τ increases, and oscillates before converging to a constant value. Again, the final distance does not converge to zero since the expected value and mode of the multinomial distribution need not be the same.

Finally, as a demonstration of the power of this imputation method, 10^4 random imputations x^{rand} were generated by Gibbs sampling, and for every imputation the distance $\mathsf{dist}(x^{\mathsf{rand}}, x^{\mathsf{opt}})$ from the optimum value x^{opt} was computed. Here, x^{opt} was determined with the upward random walk algorithm. The results are shown in Fig. 8 and can be interpreted as a measure of variability of the imputation model of Eq. (36). Also, the location of the histogram is an indication of the distance between the expected value and the mode.

6 Conclusions and outlook

The methods described in this paper can be used to find maximum likelihood or random imputations for categorical missing data problems. By focusing on contingency tables the well-understood machinery of parameterizing statistical models over sets of contingency tables (*e.g.* log-linear models) becomes available. The algorithms described here yield a powerful tool to investigate and compare such imputation models since large numbers of random imputations can be generated quickly. Important information such as estimation of imputation variance therefore becomes readily available.

The numerical tests on survey data show that the implementations work fast and reliable. The tests also show that although the Gibbs sampler converges faster in terms of markov chain length, this does not allways mean that it takes less time to compute. The behaviour depends largely on the presence of empty cells (which are not structural zeros).

The current implementation is limited to just one missing value per record, and takes into account one equality restriction, namely that the sum over entries in the contingency table is constant. To increase the utility of our implementation, the following extensions could be implemented.

1. Extension to general missing data patterns. This is a simple extension to the theory as described in Section 2. The major generalisation is that more marginals have to be taken into account.

2. Extension to general markov bases. As described in Appendix C, other linear equality restrictions than $\sum_t x_t = n$ can be implemented, although the markov bases involved can be nontrivial. One solution could be to store a library of markov bases for different tables which can be read by our program. Bases can be generated with for example 4ti2 (4ti2 team) or Macaulay 2 (Grayson and Stillman). Generating a library seems a reasonable option since computing bases can take extremely long. Also, over time more bases are becoming available in literature via the progress made in the underlying theory.

3. Incorporation into an EM-like algorithm. In the numerical examples demonstrated in this paper, the distribution on the set of all solutions to an imputation problem is parameterized using data from complete records. In order to use the information available in the incomplete data, the upward random walk algorithm can be interpreted as a Maximization step in some form of the Expectation Maximization algorithm.

4. Error correction. The method to walk through the solution space Ω_X can be used for error correction. Define Ω'_X, similar to Ω_X, except that

edit restrictions are not incorporated. Then, $\Omega_X \subseteq \Omega'_X$. Starting with a table x in $\Omega'_X \backslash \Omega_X$, it is possible to walk to the region of edit-obeying tables Ω_X. There are several types of algorithms thinkable which could perform this task. It could also be used as a general method to generate the startvalue $x(0)$ of a markov chain.

5. Generalization to continuous data. It is an open question wether the method described in this work can be extended to continuous numerical data.

Implementing the first suggestion will already yield a complete and ready-to-use imputation program which can be called from the R environment. The program should then be easy to use by statistical researchers to impute categorical datasets and to evaluate statistical models on categorical datasets with missing data. The 4^{th} suggestion is particularly usefull for generating startvalues. Evaluating statistical models under the various missing data mechanisms [M(C)AR, NMAR[3]] should be interesting, especially when the EM algorithm becomes available for this method.

Acknowledgements I would like to thank the members of the imputation study group: Sander Scholtus, Leander Kuijvenhoven, Nino Mushkudiani, Jeroen Pannekoek, Jaco Daalmans and Jan van der Laan, and Ton de Waal for carefully reading the manuscript and for useful suggestions. Any remaining errors are on my account.

Notes

[1]In the relevant literature on markov bases for contingency tables, the notation $\mathbb{Z}^k_{\geq 0}$ seems more common than $\mathbb{N}^k_0$ which is why it is adopted here.

[2]Dutch: *Permanent onderzoek leefsituatie (POLS).*

[3]Missing (Completely) at Random, Not Missing at Random

References

4ti2 team. 4ti2—a software package for algebraic geometric and combinatorial problems of linear spaces. Available at www.4ti2.de.

M. Abramowitz and I.A. Stegun, editors. *Handbook of mathematical functions: with formulas graphs and mathematical tables.* Dover Publications Inc., New York, ninth edition, 1972.

S. Aoki and A. Takemura. The list of indispensible moves of the unique minimal Markov basis for $3 \times 4 \times K$ and $4 \times 4 \times 4$ contingency tables with fixed two-diminsional marginals. Technical Report METR 2003-38, Dpt. of Mathematical Informatics, university of Tokyo, 2003a.

S. Aoki and A. Takemura. Minimal basis for connected Markov Chain over $3 \times 3 \times K$ contingency tables with fixed two-dimensional marginals. *Aust. NZ. J. Stat.*, 45:229–249, 2003b.

F. Cobben. *How to deal with nonresponse in sample surveys: Methods for analysis and adjustment.* PhD thesis, University of Amsterdam, Faculty Economics and Business, Roeterstraat 11, 1018 WB Amsterdam, 2009. To appear.

D. A. Cox, J. B. Little, and D. O'Shea. *Ideals, varieties, and algorithms.* Springer, New York, 3rd edition, 2007.

A. Demster, N. Laird, and D. Rubin. Likelihood from incomplete data via the EM algorithm. *J. Roy. Stat. Soc. B. Met.*, 39:1–39, 1977.

P. Diaconis and B. Sturmfels. Algebraic algorithms for sampling from conditional distributions. *The Annals of Statistics*, 26:363, 1998.

M. Drton, B. Sturmfels, and S. Sullivant. *Lectures on algebraic statistics*. Oberwolfbach lectures. Birkhäusen, Basel, 2008.

I. Fellegi and D. Holt. A systematic approach to automatic imputation. *J. Am. Stat. Assoc.*, 71:353, 1976.

B.L. Ford. *An overview of hotdeck procedures*, volume II. Academic Press, New York, 1983.

D. R. Grayson and M. E. Stillman. Macaulay 2, a software system for research in algebraic geometry. Available at www.math.uiuc.edu/Macaulay2/.

D. Hilbert. Über die Theorie der algebraischen Formen. *Ann. Math.*, 36:473–534, 1890.

F. le Gall. Determination of the modes of a multinomial distribution. *Statist. Probab. Lett.*, 62:325–333, 2003.

R-Dev. *Writing R-extentions*. R Developement core team, October 2008. Version 2.8.0, available from http://cran.r-project.org/manuals.html.

S. Scholtus. Algorithms for correcting obvious inconsistencies and rounding errors in business data. Technical Report 08015, Statistics Netherlands, Den Haag, 2008.

B. Sturmfels. *Gröbner bases and convex polytopes*, volume No. 8 of *Univ. lecture series*. American Mathematical Society, Providence, Rhode Island, 1996.

S. M. Sullivant. *Toric ideals in algebraic statistics*. PhD thesis, University of California, Berkely, 2005.

UCI-MLR. University of California-Irvine Machine Learning Repository. http://archive.ics.uci.edu/ml/machine-learning-databases/adult/. The UCI MLR is a collection of databases, domain theories, and data generators that are used by the machine learning community for the empirical analysis of machine learning algorithms.

C.F.J. Wu. On the convergence properties of the EM algorithm. *Ann. Stat.*, 11:95–103, 1983.

A Tensor products and direct sums of vector spaces

As a service to the reader the definition and some elementary properties of finite dimensional linear spaces are given below.

A vector space V is a set satisfying the following properties

1. If v and w are elements of V, then so is $v + w$, and $v + w = w + v$.
2. There is a neutral element 0 for which $v + 0 = v$, for every $v \in V$.
3. For every element v, there is an inverse element $\bar{v}$, such that $v + \bar{v} = 0$.
4. If $v \in V$ and λ is a scalar, then λv is also in V.

Here, we assume that λ is from a field or ring, which assures that for two scalars λ and μ, we have $\lambda\mu = \mu\lambda$. The standard example of a field is the set of real numbers $\mathbb{R}$ and the standard example of a ring is the set of integers $\mathbb{Z}$. Also, scalar multiplication distributes over addition.

Elements of V are called *vectors*. A set of abstract vectors $B = \{\mathbf{e}_1, \mathbf{e}_2, \ldots\}$ which has the property that every $v \in V$ can be uniquely expressed as

$$v = \sum_i v_i \mathbf{e}_i, \tag{A.1}$$

with the v_i scalar is called a *basis* of V. The v_i are called coefficients of v in basis B. A vector space with a finite basis is called a finite dimensional vector space. Here, we are concerned only with such spaces. In general, the choice of basis is not unique, but given a finite basis any vector can be represented as $v = (v_1, v_2, \ldots, v_n)$. Note that we use the same notation v for the abstract vector in Eq. (A.1) as for it's representation. Furthermore, from Eq. (A.1) it follows that addition of vectors is represented by elementwise addition of the coefficients, scalar multiplication distributes over the coefficients, the zero vector is represented by all coefficients equal to zero and the additive inverses are obtained by multiplying all coefficients with -1. All bases of a vector space have the same number of elements, called the *dimension* of V, or $\dim V$. Each basis vector $\mathbf{e}_i$ has a *standard representation* $\vec{e}_i$ with coefficients $e_j = \delta_{ij}$.

A *linear map* $A : V \to W$, with V and W vector spaces has the property

$$A(\lambda v + \mu v') = \lambda A v + \mu A v', \tag{A.2}$$

with λ and μ scalars and $v, v' \in V$. The action of a linear map on a vector is determined its action on the basis vectors:

$$A\mathbf{e}_i^V = \sum_j \mathbf{e}_j^W A_{ji}, \tag{A.3}$$

where A_{ji} are elements of the matrix representation of A and $\mathbf{e}_i^V$ and $\mathbf{e}_j^W$ are basis vectors of V and W respectively.

Given two finite vector spaces V and W. It is possible to construct a new finite vector space $U = V \otimes W$, called the *tensor product* space as follows. If $\mathbf{e}_i^V$ and $\mathbf{e}_j^W$ are basis vectors of V and W, we construct a basis for $V \otimes W$ by defining the abstract basis vectors

$$\mathbf{e}_{i,j} = \mathbf{e}_i^V \otimes \mathbf{e}_j^W, \tag{A.4}$$

which obey the properties

1. Symmetry: $\mathbf{e}_i^V \otimes \mathbf{e}_j^W = \mathbf{e}_j^W \otimes \mathbf{e}_i^V$.
2. Homogeneity: $(\lambda \mathbf{e}_i^V) \otimes \mathbf{e}_j^W = \lambda(\mathbf{e}_i^V \otimes \mathbf{e}_j^W)$.
3. Additivity: $\mathbf{e}_i^V \otimes (\mathbf{e}_j^W + \mathbf{e}_k^W) = \mathbf{e}_i^V \otimes \mathbf{e}_j^W + \mathbf{e}_i^V \otimes \mathbf{e}_k^W$.

Together, these properties assure that the tensor product is *bilinear* (linear in both arguments). It also follows that, using standard representations for V and W, the coefficients $u_{i,j}$ for $u = (v \otimes w) \in U$ are given by

$$u_{i,j} = (v \otimes w)_{i,j} = v_i w_j, \tag{A.5}$$

for $v \in V$ and $w \in W$. When elements from V are represented as row vectors and elements from W as column vectors, the tensor product is just the $\dim V \times \dim W$ matrix containing products of the coefficients. Also, the dimension $\dim(V \otimes W) = \dim V \cdot \dim W$.

Given two vector spaces V and W, it is possible to construct a new vector space $Z = V \oplus W$ as follows. If $\mathbf{e}_i^V$ and $\mathbf{e}_j^W$ are basis vectors of V and W, we construct a basis for $V \oplus W$ by defining the abstract basis vectors

$$\mathbf{e}_k = \mathbf{e}_i^V \oplus \mathbf{e}_j^W, \tag{A.6}$$

which obey the properties

1. Symmetry: $\mathbf{e}_i^V \oplus \mathbf{e}_j^W = \mathbf{e}_j^W \oplus \mathbf{e}_i^V$.
2. Distributivity of scalar multiplication: $\lambda(\mathbf{e}_i^V \oplus \mathbf{e}_j^W) = (\lambda \mathbf{e}_i^V) \oplus (\lambda \mathbf{e}_j^W)$.
3. Distributivity of the sum: $(\mathbf{e}_i^V + \mathbf{e}_{i'}^V) \oplus (\mathbf{e}_j^W + \mathbf{e}_{j'}^W) = (\mathbf{e}_i^V \oplus \mathbf{e}_j^W) + (\mathbf{e}_{i'}^V \oplus \mathbf{e}_{j'}^W)$.

These properties assure that every vector in $z \in Z$ can be uniquely expressed as $v \oplus w$ with $v \in V$ and $w \in W$. A representation of z is obtained simply by concatenating the representations of v and w:

$$z = (v, w) = (v_1, v_2, \ldots, v_{\dim V}, w_1, w_2, \ldots, w_{\dim W}), \tag{A.7}$$

and we have $\dim(V \oplus W) = \dim V + \dim W$.

B Some background on the sampling algorithms

Consider again the finite set Ω_X of Eq. (25). A markov process on Ω_X is a process where elements of Ω_X are chosen sequentially and randomly with replacement, and the probability of selecting an element $x(\tau+1)$ is conditional only on the previous selection $x(\tau)$. The sequence $\{x(\tau)\,|\,\tau = 0, 1, \ldots\}$, with $x(0)$ some chosen value, generated in this way is called a markov chain or random walk, and it can be thought of as a sequence of realisations of random variables X_τ. Consequently, a $|\Omega_X| \times |\Omega_X|$ transition matrix T can be constructed of which the elements elements are the conditional probabilities

$$T_{xx'} = \mathbb{P}[X_{\tau+1} = x' | X_\tau = x]. \tag{B.8}$$

Obviously all elements of T are nonnegative and $\sum_{x'} T_{xx'} = 1$. Furthermore, the transition matrix (and the corresponding markov chain and Markov process) is called irreducible if there is no permutation matrix U such that $U'TU = T_1 \oplus T_2$. Irreducibility means that for every x and $x' \in \Omega_X$ there is at least one finite markov chain containing x and x'. From here, all mentioned markov chains are assumed irreducible.

It is a standard result from markov chain theory that regardless of the start-value $x(0)$, the distribution of elements in a markov chain converges to a fixed distribution p over Ω_X as τ increases. Here, $p = \{p_x = p(x)\,|\,x \in \Omega_X\}$ is the unique left eigenvector of T with unit eigenvalue, or

$$pT = p. \tag{B.9}$$

Thus, a random sample from (Ω_X, p) can be obtained by generating a sufficiently long markov chain and taking the final element as a drawing.

Given a probability distribution on Ω_X, such as the one given in Eq. (27), there are in general many choices for T. The Metropolis-Hastings algorithm uses a particularly simple choice for T, which can be derived as follows. First, write T as $T = T^{(0)} + T^{(1)}$, where $T^{(0)}$ has zeros on the diagonal and $T^{(1)}$ contains only the diagonal elements of T. The condition in Eq. (B.9) then yields

$$p_{x'} = \sum_{x \in \Omega_X} p_x T_{xx'} = \sum_{x \in \Omega_X} p_x T^{(0)}_{xx'} + p_{x'} T^{(1)}_{x'x'}. \tag{B.10}$$

Using $T^{(1)}_{x'x'} = 1 - \sum_x T^{(0)}_{x'x}$, we get

$$\sum_{x \in \Omega_X} p_{x'} T^{(0)}_{x'x} = \sum_{x \in \Omega_X} p_x T^{(0)}_{xx'}. \tag{B.11}$$

One particular choice for which the above condition holds is when T obeys the microreversibility conditions

$$p_{x'} T_{x'x} = p_x T_{xx'}. \tag{B.12}$$

It is not difficult to check that the choice

$$T_{xx'} = \min\left\{\frac{p_{x'}}{p_x}, 1\right\} \tag{B.13}$$

obeys these conditions. An important consequence of the above is that the distribution p does not need to be normalized, since only the ratio $p(x')/p(x)$ needs to be computed. In order to derive an algorithm to generate a random walk, the elements of T can be written as:

$$T_{xx'} = \min\left\{\frac{p_{x'}}{p_x}, 1\right\} = r(x'|x)q(x, x'), \tag{B.14}$$

where $r(x'|x)$ is the probability that element x' is generated from point x and $q(x, x')$ is the probability that the step is actually taken. There is some liberty in choosing r and q. In particular, if $r(x'|x) = r(x|x')$ we have $q(x, x') = \min\{p(x')/p(x), 1\}$ from the microreversibility conditions. Instead of directly generating x', a vector v can be generated so that $x' = x + v$. The step vector v must be drawn from a set of vectors large enough to ensure irreducibility of the markov chain. This introduces the concept of a markov basis, which is the subject of the next section. The markov basis with elements v^{rs} for Ω_X is given explicitly in Eq. (C.18). In effect, the Metropolis-Hastings sampler is defined by $T_{xx'} = \mathcal{U}\{v^{rs}\}q(x, x + v^{rs})$ if $x' = x + v^{rs}$ and zero otherwise. Here $\mathcal{U}\{\cdot\}$ is the probability of drawing v^{rs} under the uniform distribution over the markov bases. It is a property of Markov bases that they do not need to be minimal in order to obtain the convergence property in Eq. (B.9). In that sense, the Gibbs sampler is just an extension of the Metropolis-Hastings algorithm, and is characterized by $T_{xx'} = \mathcal{U}\{v^{rs}\}q(x, x + kv^{rs})$ for all k for which $x' = x + kv^{rs} \in \Omega_X$.

C A proof for the markov basis

This section is aimed to show that the set of v^{rs} steps of section 4.1 is indeed a valid markov basis. It also serves as a glance at the formulation used in the calculation of markov bases for more general problems, for example when certain marginals of a contingency table are fixed.

The most important property of elements in Ω_X is that the sum $\sum_t x_t = n$ for all $x \in \Omega_X$. It will be convenient to define the following linear map:

$$\pi : \otimes_{i=1}^{m} \mathbb{Z}_{\geq 0}^{d_i} \to \mathbb{Z}_{\geq 0},\ \pi x = \sum_{t=0}^{d-1} x_t. \tag{C.15}$$

We are now able to define a markov basis.

Definition C.1. A set B_X of vectors $v \in \mathbb{Z}^d$ is a markov basis for Ω_X if

$$\pi(x+v) = \pi x \quad \forall x \in \Omega_X, \tag{C.16}$$

and for all x and $x' \in \Omega_X$, there is a finite sequence of pairs $(v(\tau), \varepsilon(\tau))$ with $\varepsilon(\tau) \in \{\pm 1\}$ so that

$$x' = x + \sum_\tau \varepsilon(\tau) v(\tau). \tag{C.17}$$

The first demand makes sure that no steps outside Ω_X are allowed and the second demand ensures irreducibility.

A markov basis for Ω_X is given explicitly by

$$B_X = \{v^{rs} \in \mathbb{Z}^d \,|\, v^{rs}_t = \delta_{rt} - \delta_{st},\ 0 \le t \le d-1\}. \tag{C.18}$$

Before we can proof this, we need to introduce two notions from the theory of polynomial rings. A ring of polynomes $\mathbf{k}[z_0, z_2, \ldots, z_{d-1}]$ is the set of all polynomes generated by taking finite products and linear combinations of abstract variables z_i. Every element $f \in \mathbf{k}[z_0, z_1, \ldots, z_{d-1}]$ can be written as

$$f = \sum_\alpha a_\alpha z^\alpha, \tag{C.19}$$

where we use the shorthand notation $z^\alpha = z_0^{\alpha_0} z_1^{\alpha_1} \cdots z_{d-1}^{\alpha_{d-1}}$, $\alpha \in \mathbb{Z}^d_{\ge 0}$ and the linear coefficients $a_\alpha \in \mathbf{k}$ where $\mathbf{k}$ is any field, for example the complex numbers. The second notion is that of an ideal. An ideal J is a subset of $\mathbf{k}[z_0, z_1, \ldots, z_{d-1}]$ which is closed under finite linear combinations and multiplication. It was shown by Hilbert (1890) [but see Cox *et al.* (2007)] that every ideal has a finite set of generators $g_1, g_2, \ldots, g_k$ such that every $h \in J$ can be written as a finite sum

$$h = \sum_{i=1}^k g_i f_i, \tag{C.20}$$

where all $f_i \in \mathbf{k}[z_0, z_1, \ldots, z_{d-1}]$. In particular, if J is generated by $g_1, g_2, \ldots, g_k$, we write

$$J = \langle g_1, g_2, \ldots, g_k \rangle. \tag{C.21}$$

Now, consider the lifting of π defined by

$$\hat{\pi} : \mathbf{k}[z_0, z_1, \ldots, z_{d-1}] \to \mathbf{k}[w], \quad \hat{\pi}(f) = \sum_\alpha a_\alpha w^{\pi\alpha}, \tag{C.22}$$

with f as in Eq. (C.19). Thus, lifting just means that we let π act on the vectors of exponents of a polynomial, and change the number of variables of

the polynomial accordingly. It follows that $\hat{\pi}(f+g) = \hat{\pi}(f) + \hat{\pi}(g)$ and $\hat{\pi}(fg) = \hat{\pi}(f)\hat{\pi}(g)$. We also define the kernel of $\hat{\pi}$, given by all $h \in \mathbf{k}[z_0, z_1, \ldots, z_{d-1}]$ for which $\hat{\pi}(h) = 0$. It follows that $\ker \hat{\pi}$ is an ideal. Note that the demand in Eq. (C.16) shows that every $v^{rs} \in B_X$ must be in $\ker_{\mathbb{Z}} \pi$, the integer kernel of π. It was shown by Diaconis and Sturmfels (1998) that the second demand [Eq. (C.17)] is obeyed if and only if there is a one-to-one correspondence between the generators of the kernel of $\hat{\pi}$ and the elements of the markov basis. Here we specialize their theorem to our case.

Theorem C.2 (Specialization of Thm. 3.1 of Diaconis and Sturmfels (1998))**.** Consider the set B_X of Eq. (C.18) and write v^{rs} as the vector difference $v^{rs} = \alpha_+^r - \alpha_-^s$. Vector α_+^r has unit rth coefficient and is zero elsewhere, and likewise for α_-^s. Also define $\Omega_\pi = \{x \in \mathbb{Z}_{\geq 0}^d | \pi x = n\}$. The set B_X is a markov basis for Ω_X if and only if the ideal

$$J_{\hat{\pi}} = \langle z^{\alpha_+^r} - z^{\alpha_-^s} \,|\, 0 \leq r, s \leq d-1 \rangle = \ker \hat{\pi}. \tag{C.23}$$

Proof. The proof of this theorem is stated in the reference and will not be repeated here. □

Proposition C.3. The set B_X, defined in Eq. (C.18) is a markov basis for Ω_π.

Proof. To show that B_X is indeed a markov basis, we need to show that $J_{\hat{\pi}}$ obeys the equality in Eq. (C.23). First we will show that $J_{\hat{\pi}} \subseteq \ker \hat{\pi}$. The action of $\hat{\pi}$ on any generator yields

$$\hat{\pi}(z^{\alpha_+^r} - z^{\alpha_-^s}) = w^{\pi\alpha_+^r} - w^{\pi\alpha_-^s} = 0, \tag{C.24}$$

since $\pi\alpha_+^r = \pi\alpha_-^s = 1$. Next, we show that $\ker \hat{\pi} \subseteq J_{\hat{\pi}}$. Suppose $h = \sum_\beta a_\beta z^\beta \in \ker \hat{\pi}$, then

$$\hat{\pi}(h) = \sum_\beta a_\beta w^{\pi\beta} = 0. \tag{C.25}$$

This can only happen when all nonzero a_β occur in pairs a_{β_+}, a_{β_-} with $a_{\beta_+} = a_{\beta_-} \equiv a_{\beta_\pm}$ and $\pi\beta_+ = \pi\beta_-$, so Eq. (C.25) has terms

$$a_{\beta_+} w^{\pi\beta_+} - a_{\beta_-} w^{\pi\beta_-} = a_{\beta_\pm} \hat{\pi}(z^{\beta_+} - z^{\beta_-}) = 0. \tag{C.26}$$

We now need to show that $(z^{\beta_+} - z^{\beta_-}) \in J_{\hat{\pi}}$. First divide out common factors so $(z^{\beta_+} - z^{\beta_-}) = z^\gamma (z^{\beta'_+} - z^{\beta'_-})$. It follows that $\pi\beta'_+ = \pi\beta'_-$ and $\beta'_+ \perp \beta'_-$. To show that $z^{\beta'_+} - z^{\beta'_-} \in J_{\hat{\pi}}$, write

$$\begin{aligned} z^{\beta'_+} - z^{\beta'_-} &= z_r g_+ - z_s g_- \\ &= g_+(z_r - z_s) + z_s(g_+ - g_-), \end{aligned} \tag{C.27}$$

with z_r and z_s factors of $z^{\beta'_+}$ and $z^{\beta'_-}$ respectively. The first term is obviously an element of $J_{\hat{\pi}}$ and the binomial $g_+ - g_-$ is of lesser degree than $z^{\beta'_+} - z^{\beta'_-}$. The above procedure can be applied recursively to the second term because the exponent vectors of g_+ and g_- inherit the properties of β'_+ and β'_- under action of π. This shows that indeed $h \in J_{\hat{\pi}}$. □

Remember that the set of valid contingency tables [Eq. (25)] can be written as

$$\Omega_X = \Omega_\pi \cap \{x \in \mathbb{Z}^d_{\geq 0} | x_t = 0 \text{ when } t \in \mathcal{I}\} \cap \{x \in \mathbb{Z}^d_{\geq 0} | Ax \geq b\}. \qquad \text{(C.28)}$$

Corollary C.4. B_X contains a markov basis for Ω_X.

Proof. The inclusion of edit restrictions $\mathcal{I}$ reduces d to $d - |\mathcal{I}|$, and does not change the conditions of the proposition. The restrictions $Ax \geq b$ imply that Ω_X is the intersection of Ω_π with a collection of half-spaces. Ω_X is thus a convex subset of Ω_π. Since B_X is a markov basis for Ω_π, (a subset of) B_X is a markov basis for $\Omega_X \subseteq \Omega_\pi$. □

Remarks. As stated before, Thm. (C.2) is a specialization of Thm. 3.1 in Diaconis and Sturmfels (1998). The specialization concerns the map π. In the general form, π can be any linear map $\mathbb{Z}^d_{\geq 0} \mapsto \mathbb{Z}^k_{\geq 0}$ with $k \leq d$, where the most important example is the map which computes marginals of x. The formulation in terms of ideals has as advantage that general algorithms such as the Buchberger algorithm are available to compute the generating set. See for example Cox *et al.* (2007) for an accessible introduction. A good introduction to the theory of markov bases for contingency tables can also be found in Sullivant (2005) or Drton *et al.* (2008). The ideal $J_{\hat{\pi}}$ of Eq. (C.23) is called a binomial ideal or a toric ideal. The theory of of toric ideals is described thoroughly in Sturmfels (1996).

In principle, markov bases can be computed for any linear map π although the Buchberger algorithm can have extremely high computational time- and memory complexity as a function of the dimension d of the contingency table. Software implementing (improved versions of) the Buchberger algorithm is available for free, for example in the 4ti2 package (4ti2 team) or in the Macaulay 2 environment (Grayson and Stillman). Among the most studied cases are those where π fixes all marginals of a table. In that case, even the calculation of markov bases of for example $4 \times 4 \times 4 \times 4$ contingency tables seems computationally not feasible. Some special cases are known however. Diaconis and Sturmfels (1998) use an earlier version of Macaulay to compute the basis for the $3 \times 3 \times 3$ case with fixed line sums (sums over one index at the time). They also give the general result for $2 \times J \times K$ tables. Aoki and Takemura (2003b) have determined the general basis for $3 \times 3 \times K$ tables with fixed 2D marginals. It was

already shown by Diaconis and Sturmfels (1998) that markov bases for tables with fixed marginals can be described in terms of several types of indispensable moves. Aoki and Takemura (2003a) list indispensible moves for $3 \times 4 \times K$ and $4 \times 4 \times 4$ contingency tables with fixed 2D marginals.