The Bayesian analysis of contingency table data using the bayesloglin R package

Matthew Friedlander

Keywords. Bayesian model selection, Bayesian inference, MC3, hierarchical, graphical, decomposable, log-linear models, Gibbs sampling, hyper Dirichlet prior, Bayesian iterative proportional fitting, Czech autoworkers data, igraph, bayesloglin.

1 Introduction

Data in the form of a contingency table arise when individuals are cross classified according to a finite number of criteria. Log-linear modeling (see e.g., [1], [4], or [3]) is a popular and effective methodology for analyzing such data enabling the practitioner to make inferences about dependencies between the various criteria. For hierarchical log-linear models, the interactions between the criteria can be represented in the form of a graph; the vertices represent the criteria and the presence or absence of an edge between two criteria indicates whether or not the two are conditionally independent [11]. This kind of graphical summary greatly facilitates the interpretation of a given model.

For log-linear analysis, we can use the conjugate prior of [14] to work in the Bayesian paradigm. With this prior, the MC3 algorithm of [13] allows for exploration of the space of models to try to find those with the highest posterior probability. Once top models have been identified, a block Gibbs sampler can be constructed to sample from the posterior distribution and to estimate parameters of interest. Our aim in this paper, is to introduce the bayesloglin R package [17] to carry out these tasks.

The outline of this paper is as follows: In section 2, we develop the notation for hierarchical log-linear models which is based on [12]. In section 3, we give the conjugate prior for hierarchical models under Poisson sampling. In section 4, we describe the MC3 algorithm for searching the spaces of hierarchical, graphical, and decomposable models. Section 5 deals with Gibbs sampling and section 6 gives some exact results for the normalizing constant and the mean and variance of the log-linear parameters for decomposable models. In section 7 we illustrate the use of the bayesloglin package for analyzing the often studied Czech autoworkers data from [9].

2 Preliminaries

The notation in this section is adapted from [12] with some minor changes to move from multinomial to Poisson sampling. Let V be a finite set of indices representing $|V|$ criteria. We assume that the criterion labeled by $v \in V$ can take values in a finite set $\mathcal{I}_v$. The resulting counts are gathered in a contingency table such that

$$\mathcal{I} = \prod_{v \in V} \mathcal{I}_v$$

is the set of cells $i = (i_v, v \in V)$. The vector of cell counts is denoted $n = (n(i), i \in \mathcal{I})$ with corresponding mean $m(i) = E(n) = (m(i), i \in \mathcal{I})$. For $D \subset V$,

$$\mathcal{I}_D = \prod_{v \in D} \mathcal{I}_v$$

is the set of cells $i_D = (i_v, v \in D)$ in the D-marginal table. The marginal counts are $n(i_D) = \sum_{j : j_D = i_D} n(j)$ with $m(i_D) = E\left(n(i_D)\right)$.

Let $\mathcal{D}$ be a family of subsets of V such that $D \in \mathcal{D}$ and $D_1 \subset D$ implies that $D_1 \in \mathcal{D}$. We will assume that $\cup_{D \in \mathcal{D}} D = V$. The hierarchical log-linear model generated by $\mathcal{D}$ is

$$\log m(i) = \sum_{D \in \mathcal{D}} \lambda_D(i)$$

where $m(i)$ is assumed positive and $\lambda_D(i)$ is a real valued function that depends on i only through i_D.

We now select a select a special element in each $\mathcal{I}_v$. For convenience, we denote it 0. We also denote 0 in i the cell with all its components equal to 0. The choice of special element 0 in each $\mathcal{I}_v$ is arbitrary. If $i \in \mathcal{I}$, the support of i is the subset of V defined as $S(i) = \{v \in V, i_v \neq 0\}$. We let $J = \{j \in \mathcal{I}, S(j) \in \mathcal{D}\}$ and define the notation $j \lhd i$ for $i \in \mathcal{I}$ and $j \in J$ to mean that $j_{S(j)} = i_{S(j)}$. By convention, we say that $0 \lhd i$ for any $i \in \mathcal{I}$. For any $a \in \mathcal{D}$, we also define the sub-model $J_a = \{j \in J : S(j) \subseteq a\}$.

Let $(e_j, j \in J)$ be the canonical basis of R^J. For all $i \in \mathcal{I}$, we define $f_i \in R^J$ by

$$f_i = \sum_{j \in J, j \lhd i} e_j.$$

The baseline constrained hierarchical log-linear model generated by $\mathcal{D}$ has the unique representation

$$\log m(i) = \sum_{j \in J : j \lhd i} \theta_j = \langle f_i, \theta \rangle$$

for $i \in \mathcal{I}$ and $\theta = (\theta_j, j \in J) \in R^J$. In matrix notation, we have

$$\log m = X\theta$$

where X is an $\mathcal{I} \times J$ design matrix of full column rank with rows $\{f_i, i \in \mathcal{I}\}$. It is worth noting that X is a binary 0/1 matrix with a first column that is all 1's.

3 Prior distribution under Poisson sampling

We assume that the components of n are independent and follow a Poisson distribution. The sufficient statistic $t = X^T n$ has a probability distribution in the natural exponential family

$$f(t) = \exp\left(\langle \theta, t \rangle - \sum_{i \in \mathcal{I}} \exp\left(\langle f_i, \theta \rangle\right)\right) \nu\left(dt\right)$$

with respect to a discrete measure ν that has convex support

$$C^p = \left\{\sum_{i \in \mathcal{I}} y(i) f_i, y(i) \geq 0, i \in \mathcal{I}\right\} = \text{cone}\left\{f_i, i \in \mathcal{I}\right\}$$

i.e. the convex cone generated by the rows of the design matrix X. The Diaconis and Ylvisaker [7] conjugate prior with respect to the Lebesgue measure for the log-linear parameters is

$$f(\theta) = I(r,\alpha)^{-1} \exp\left(\alpha \langle r, \theta\rangle - \alpha \sum_{i\in\mathcal{I}} \exp\langle f_i, \theta\rangle\right)$$

where

$$I(r,\alpha) = \int_{\theta\in R^J} \exp\left(\alpha \langle r, \theta\rangle - \alpha \sum_{i\in\mathcal{I}} \exp\langle f_i, \theta\rangle\right) d\theta$$

and is proper when $\alpha > 0$ and $r = X^T y$ for some $y > 0$ i.e. r is in the relative interior of C^p. The Bayes factor for comparing two models J_1 and J_2 is

$$B_{12} = \frac{P(t|J_1)}{P(t|J_2)} = \frac{I_1\left(\frac{t+\alpha r}{1+\alpha}, 1+\alpha\right)/I_1(r,\alpha)}{I_2\left(\frac{t+\alpha r}{1+\alpha}, 1+\alpha\right)/I_2(r,\alpha)}$$

4 The MC3 algorithm for model selection

The Bayesian paradigm to model selection involves choosing models with high posterior probability from a set $\mathcal{M}$ of competing models. We associate with each model $J \in \mathcal{M}$ a neighbourhood $\text{nbd}(J) \subset \mathcal{M}$. The MC^3 algorithm proposed by [13] constructs an irreducible Markov chain with state space $\mathcal{M}$ and and equillibrium distribution $\{p(J|n) : J \in \mathcal{M}\}$ where $P(J|t)$ is the posterior probability of J. We assume that all models are apriori equally likely; hence $P(J|t)$ is proportional to the marginal likelihood $P(t|J) = I(t+r, \alpha+1)/I(r,\alpha)$.

If the chain is in state J at we draw a candidate model J' from a uniform distribution on nbd(J). The chain moves to J' with probability

$$\min\left\{1, \frac{P(t|J)/\#\text{nbd}(J)}{P(t|J')/\#\text{nbd}(J')}\right\}$$

where #nbd(J) denotes the number of neighbours of J. Otherwise the chain does not move. The evaluation of the marginal likelihoods and the specification of model neighbourhoods is done with respect to the particular properties of the set of candidate models considered.

1. **Hierarchical log-linear models**. We calculate the marginal likelihood through the Laplace approximation to the normalizing constants for the prior and posterior distribution of the log-linear model parameters. The neighbourhood of a hierarchical model J consists of the hierarchical models obtained from J by adding one of its dual generators (i.e. minimal terms not present in the model) or deleting one of its generators (i.e. maximal terms present in the model). For details see [9] and [6].

2. **Graphical log-linear models**. We evaluate the marginal likelihood using the Laplace approximation to the normalizing constants as we do in the hierarchical case. The neighbourhood of a graphical model with corresponding graph G consists of those models whose independence graphs are obtained from G by adding or removing one edge.

3. **Decomposable log-linear models**. In this case, the marginal likelihood can be obtained explicitly. See Section 6 for the formula. The neighbourhood of a decomposable model with corresponding graph G consists of those models whose independence graphs are decomposable and are obtained by adding or deleting one edge from G.

5 Gibbs sampling

Our aim in this section is to develop a blocked Gibbs sampler to sample from the posterior distribution and to estimate parameters of interest. We begin by partitioning the cells and the prior into blocks. For $a \in \mathcal{D}$ we define the sets $B_{i_a} = \{j \in \mathcal{I} : j_a = i_a\}$ for $i_a \in \mathcal{I}_a$. These sets are disjoint and partition $\mathcal{I}$. Define the vectors $\chi_{i_a}, i_a \in \mathcal{I}_a$ with

$$\chi_{i_a}(i) = \begin{cases} 1 & i \in B_{i_a} \\ 0 & \text{otherwise} \end{cases}$$

and the matrix χ with columns $x_{i_a}(i)$. We can then write: $f_i = f_{(i_a, i_{\mathrm{a}}^{\mathrm{c}})} = f_{(i_a,0)} + f_{(0,i_{\mathrm{a}}^{\mathrm{c}})}$ and $\theta = (\theta_a, \theta_{a^c})$ where $\theta_a = (\theta_j : S(j) \subseteq a)$ and $\theta_{a^c} = (\theta_j : S(j) \not\subseteq a)$.

The marginal counts $m(i_a), i_a \in \mathcal{I}_a$ follow a log-linear model with

$$\begin{aligned} m\left(i_a\right) &= \sum_{i \in B_{i_a}} \exp\left(\langle f_i, \theta \rangle\right) \\ &= \exp\left(\left\langle f_{(i_a,0)}, \theta \right\rangle\right) \sum_{i \in B_{i_a}} \exp\left(\left\langle f_{(0,i_{a^c})}, \theta \right\rangle\right) \end{aligned}$$

and, taking logs,

$$\log m\left(i_a\right) = \sum_{i \in B_{i_a}} \left\langle f_{(i_a,0)}, \theta \right\rangle + \log\left(\sum_{i \in B_{i_a}} \exp\left(\left\langle f_{(0,i_{a^c})}, \theta \right\rangle\right)\right)$$

Let $m_a = (m(i_a), i_a \in \mathcal{I}_a)$ and partition the matrix X such that $X = [X_a, X_{\bar{a}}]$ where X_a is a matrix made up of those columns of X corresponding to j such that $S(j) \subseteq a$ and $X_{\bar{a}}$ is a matrix with all the other columns. Then, in matrix notation,

$$\log m_a = \left(\frac{\chi^T X_a}{\left|\mathcal{I} \backslash \mathcal{I}_a\right|}\right) \theta_a + \log\left(\chi^T \exp\left(X_{\bar{a}} \theta_{\bar{a}}\right)\right)$$

Returning to the prior, parametrized temporarily in terms of m, we can partition f as

$$\begin{aligned} f(m|J) &\propto \exp\left(\alpha \left\langle y, \log m \right\rangle - \alpha \sum_{i \in \mathcal{I}} m(i)\right) \\ &= \left\{ \prod_{i_a \in \mathcal{I}_a} \prod_{i \in B_{i_a}} \left(\frac{m(i)}{m\left(i_a\right)}\right)^{\alpha y(i)} \right\} \left\{ \prod_{i_a \in \mathcal{I}_a} m\left(i_a\right)^{\alpha y(i_a)} \exp\left(-\alpha m\left(i_a\right)\right) \right\} \\ &= f\left(m_{\bar{a}}\right) f\left(m_a | m_{\bar{a}}\right) \end{aligned}$$

and we see that $f\left(m_a | m_{\bar{a}}\right)$ is the product of independent $\text{Gamma}\left(1 + \alpha y\left(i_a\right), 1/\alpha\right), i_a \in \mathcal{I}_a$ distributions. Since it is easy to generate from $f\left(m_a | m_{\bar{a}}\right)$ for each $a \in \mathcal{D}$, a blocked Gibbs sampler [10] is feasible to sample from $f(\theta)$. Following [8], we begin by choosing an arbitrary initial value of $\theta^{(0)}$. For a given value of $\theta^{(k)}$, we update as follows:

1. Generate independent $m\left(i_a\right) \sim \text{Gamma}\left(\alpha y\left(i_a\right), 1/\alpha\right)$ random variables for all $a \in \mathcal{D}$ and $i_a \in \mathcal{I}_a$.

2. For each $a \in \mathcal{D}$, in any arbitrary order set,

$$\theta_a^{(k)} = \left(\frac{\chi^T X_a}{|\mathcal{I} \backslash \mathcal{I}_a|}\right)^{-1} \left(\log\left(m_a\right) - \log\left(x^T \exp\left(X_{\bar{a}} \theta_{\bar{a}}\right)\right)\right)$$

using the most recent value of $\theta_{\bar{a}}$ available.

After a suitable burn-in, the resulting samples come from $f(\theta)$. We note that the above Gibbs sampler is also known as the Bayesian Iterative Proportional Fitting algorithm. See [8],[2],[15], and[16] for more details.

6 Some exact results for decomposable models

For decomposable models, some exact results exist for the normalizing constant and the mean and variance of the log-linear parameters. Let us reconsider the prior defined in section 3 as

$$f(\theta) = I(r, \alpha)^{-1} \exp\left(\alpha \langle r, \theta \rangle - \alpha \sum_{i \in \mathcal{I}} \exp \langle f_i, \theta \rangle\right)$$

with

$$I(r, \alpha) = \int\limits_{\theta \in R^J} \exp\left(\alpha \langle r, \theta \rangle - \alpha \sum_{i \in \mathcal{I}} \exp \langle f_i, \theta \rangle\right) d\theta$$

where $\alpha > 0$ and $r = (r_j, j \in J) \in \mathrm{ri}\,(\mathrm{C_p})$. Then

$$\mathrm{E}\left(\alpha\theta\right) = \frac{\partial \log I(r, \alpha)}{\partial r}$$

and

$$\mathrm{Cov}(\alpha\theta) = \frac{\partial^2 \log I(r, \alpha)}{\partial r^2}$$

In the case of log-linear models where m is Markov with respect to a decomposable graph $G = (V, E)$, with vertex set V and edge set E, an explicit formula exists for $I(r, \alpha)$. Let C denote the set of cliques and S the set of minimal vertex separators. For a given $s \in S$, let $V_1, V_2, ..., V_p$ be the connected components of the subgraph $G_{V \backslash s}$ and q be the number of $j = 1, 2, ..., p$ such that s is not a clique of $S \cup V_j$. Then $\nu(s) = q - 1$ is called the multiplicity of s and $\sum_{s \in S} \nu(s) = |C| - 1$ [11]. Based on proposition 4.2 of [14], adapted to Poisson sampling, we have

$$I(r, \alpha) = \alpha^{-\alpha \sum_{i \in \mathcal{I}} y(i)} \frac{\prod_{c \in C} \prod_{i_c \in \mathcal{I}_c} \Gamma\left(\alpha y(i_c)\right)}{\prod_{s \in S} \prod_{i_s \in \mathcal{I}_s} \left\{\Gamma\left(\alpha y(i_s)\right)\right\}^{\nu(s)}}$$

Taking logs and differentiating with respect to r gives

$$E(\theta) = -\frac{d \sum_{i \in \mathcal{I}} y(i)}{dr} \log \alpha + \sum_{c \in C} \sum_{i_c \in \mathcal{I}_c} \psi\left(\alpha y(i_c)\right) \frac{dy\left(i_c\right)}{dr} - \sum_{s \in C} \sum_{i_s \in \mathcal{I}_s} \nu(s) \psi\left(\alpha y(i_s)\right) \frac{dy\left(i_s\right)}{dr}$$

where ψ is the digamma function. Note that the derivatives in the right hand side of $\mathrm{E}(\theta)$ are vectors. In particular, $d \sum_{i \in \mathcal{I}} y(i)/dr = (1, 0, ..., 0)^T$ since $r_0 = \sum_{i \in \mathcal{I}} y(i)$. Differentiating once more we have

$$\mathrm{Cov}\left(\theta\right) = \sum_{c \in C} \sum_{i_c \in \mathcal{I}_c} \psi_1\left(\alpha y(i_c)\right) \frac{dy\left(i_c\right)}{dr} \frac{dy\left(i_c\right)}{dr}^T - \sum_{s \in S} \sum_{i_s \in \mathcal{I}_s} \nu(s) \psi_1\left(\alpha y(i_s)\right) \frac{dy\left(i_s\right)}{dr} \frac{dy\left(i_s\right)}{dr}^T$$

with ψ_1 being the trigamma function. We note that for decomposable models, the subgraphs $G_c, c \in C$ and $G_s, s \in S$ are all complete and we have a saturated model on those subgraphs. For $a \in C \cup S$, it is easy to find $d\left(y(i_a)\right)/dr, i_a \in \mathcal{I}_a$ by inverting the design matrix for the model J_a.

7 The bayesloglin R package.

The bayesloglin package includes the 2^6 Czech autoworkers data from [9]. This cross-classification of 1841 men gives six potential risk-factors for coronary thrombosis: (a) smoking, (b) strenuous mental work, (c) strenuous physical work, (d) systolic blood pressure, (e) ratio of beta and alpha lipoproteins and (f) family anamnesis of coronary heart disease. Currently, bayesloglin only allows choice of the hyperparameter α and sets $y(i) = 1/|\mathcal{I}|$ for each $i \in \mathcal{I}$. Consequently, $r_0 = \sum_{i \in \mathcal{I}} y(i) = 1$. The required R code to search for the top decomposable, graphical, and hierarchical log-linear models is:

```
> data(czech)
> s1 <- MC3 (init = NULL, alpha = 1, iterations = 5000, replicates = 1,
              data = czech, mode = "Decomposable")
> s2 <- MC3 (init = NULL, alpha = 1, iterations = 5000, replicates = 1,
              data = czech, mode = "Graphical")
> s3 <- MC3 (init = NULL, alpha = 1, iterations = 5000, replicates = 1,
              data = czech, mode = "Hierarchical")
```

The top models in terms of posterior probability are

```
> head(s1, n = 5)
                  formula   logPostProb
1    [a,c,e][b,c][d,e][f]   5271.975
2  [a,c,e][a,d,e][b,c][f]   5271.103
3    [a,c,e][a,d][b,c][f]   5271.077
4 [a,c][b,c][b,e][d,e][f]   5270.549
5  [a,c,e][b,c][b,f][d,e]   5270.394

> head(s2, n = 5)
                         formula  logPostProb
1      [a,c][a,d,e][b,c][b,e][f]  7122.398
2   [a,c][a,e][b,c][b,e][d,e][f]  7121.580
3    [a,c][a,d,e][b,c][b,e][b,f]  7121.374
4   [a,c][a,d][a,e][b,c][b,e][f]  7120.683
5 [a,c][a,e][b,c][b,e][b,f][d,e]  7120.556

> head(s3, n = 4)
                                 formula  logPostProb
1      [a,c][a,d][a,e][b,c][c,e][d,e][f]  7125.171
2      [a,c][a,d][a,e][b,c][b,e][d,e][f]  7124.704
3 [a,c][a,d][a,e][b,c][b,e][c,e][d,e][f]  7124.229
4    [a,c][a,d][a,e][b,c][b,f][c,e][d,e]  7124.147
```

These results match those obtained by the same methods in [14]. Consider the top hierarchical model $[a,c][a,d][a,e][b,c][c,e][d,e][f]$. We can use the function `gibbsSampler` to sample from the posterior and obtain estimates of the mean and variances of the log-linear parameters. We use a burn-in of 5000 iterations.

```
> formula <- freq ~ a*c + a*d + a*e + b*c + c*e + d*e + f
> s <- gibbsSampler (formula, alpha = 1, data = czech,
                    nSamples = 15000, verbose = T)
> postMean <- colSums(s[5000:15000,]) / 10000
> postCov <- cov(s[5000:15000,])
> postVar <- diag(postCov)
```

The values of `postMean` and `postVar` are

```
> postMean
(Intercept)           a1           c1           b1           d1           e1
  3.0915633  -0.4150080   1.0199107   0.9010453  -0.2877865  -0.4890538
         f1        a1:c1        b1:c1        a1:d1        a1:e1        c1:e1
 -1.8057132   0.5409632  -2.8017859  -0.3542662   0.4871123  -0.4479492
      d1:e1
  0.3784125

> postVar
(Intercept)          a1          c1          b1           d1           e1
0.006921940 0.008033988 0.008498167 0.005310040  0.005564232  0.008184625
         f1       a1:c1       b1:c1       a1:d1        a1:e1        c1:e1
0.004433024 0.009185728 0.015035403 0.009219168  0.009280780  0.009133959
      d1:e1
0.009298324
```

We now consider the decomposable model $[a,c,e][b,c][d,e][f]$. The `findPostMean` and `findPostCov` functions can compute the posterior mean and covariance matrix, which for decomposable models, is available in closed form. In R we have

```
> formula <- freq ~ a*c*e + b*c + d*e + f
> postMean <- findPostMean (formula, alpha = 1, data = czech)
> postCov <- findPostCov(formula, alpha = 1, data = czech)
> postVar <- diag(postCov)

> postMean
(Intercept)          b1           c1           a1           e1           d1
 3.1561271    0.9002899   1.0149757  -0.5565110  -0.4621862  -0.4387784
        f1        b1:c1        a1:c1        a1:e1        c1:e1        d1:e1
-1.8051306   -2.8012942   0.5494842   0.4645452  -0.4380842   0.3412027
  a1:c1:e1
-0.0194745
```

```
> postVar
(Intercept)          b1          c1          a1          e1          d1
0.006563014 0.005252849 0.009530313 0.008807288 0.009375078 0.003956279
         f1       b1:c1       a1:c1       a1:e1       c1:e1       d1:e1
0.004478660 0.014932109 0.015834157 0.018016838 0.018531263 0.009099995
   a1:c1:e1
0.037264994
```

The reader can verify that the Gibbs sampler gives a close approximation to the exact values for this model.

Acknowledgements. The author is grateful to the creators of the `igraph` R package [5] which was used extensively for manipulating graphs. Special thanks also to Adrian Dobra, whose C++ code for representing hierarchical log-linear models, was adapted for use in R.

References

[1] A. Agresti. *Categorical Data Analysis.* John Wiley & Sons, Hoboken, NJ, 2 edition, 1990.

[2] A. Agresti. *Gelman, A. and Carlin, J. B. and Stern, H. S. and Rubin D.* Chapman and Hall, 2 edition, 2004.

[3] Y. M. M. Bishop, S. E. Feinberg, and P. W. Holland. *Discrete Multivariate Analysis.* MIT Press, Cambridge, MA, 1975.

[4] R. Christensen. *Log-linear Models and Logistic Regression.* Springer Verlag, 1997.

[5] G. Csardi and T. Nepusz. The igraph software package for complex network research. *InterJournal*, Complex Systems:1695, 2006. URL `http://igraph.org`.

[6] P. Dellaportas and J. J. Forster. Markov chain monte carlo determination for hierarchical and graphical log-linear models. *Biometrika*, 86:615–633, 1999.

[7] P. Diaconis and D. Ylvisaker. Conjugate priors for exponential families. *Annals of statistics*, 7:269–281, 1979.

[8] A. Dobra and H. Massam. The mode oriented stochastic search (moss) algorithm for log-linear models with conjugate priors. *Statistical Methodology*, 7:204–253, 2010.

[9] D. E. Edwards and T. Havranek. A fast procedure for model search in multidimensional contingency tables. *Biometrika*, 72:339–351, 1985.

[10] C. S. Jensen and A. Kong. Blocking gibbs sampling for linkage analysis in large pedigrees with many loops. *American Journal of Human Genetics*, 65:885–901, 1999.

[11] S. F. Lauritzen. *Graphical Models.* Oxford University Press, NY, 1996.

[12] G. Letac and H. Massam. Bayes factors and the geometry of discrete hierarchical log-linear models. *Annals of statistics*, 40:861–890, 2012.

[13] D. Madigan and J. York. Bayesian methods for estimation of the size of a closed population. *Biometrika*, 84:19–31, 1997.

[14] H. Massam, J. Liu, and A. Dobra. A conjugate prior for discrete hierarchical log-linear models. *Annals of Statistics*, 37:3431–3467, 2009.

[15] M. Piccioni. Independence structure of natural conjugate densities to exponential families and the gibbs sampler. *Scandinavian Journal of Statistics*, 27:111–127, 2000.

[16] J. L. Schafer. *Analysis of Incomplete Multivariate Data*. Chapman and Hall, London, 1997.

[17] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2016. URL `http://www.R-project.org/`.