ISSN: 2281-1346

Department of Economics and Management

DEM Working Paper Series

Probability Based Independence Sampler for Bayesian Quantitative Learning in Graphical Log-Linear Marginal Models

Ioannis Ntzoufras
(Athens University of Economics and Business)

Claudia Tarantola
(Università di Pavia)

Monia Lupparelli
(Università di Bologna)

 # 149 (01-18)

Via San Felice, 5
I-27100 Pavia
economiaweb.unipv.it

January 2018

Probability Based Independence Sampler for Bayesian Quantitative Learning in Graphical Log-Linear Marginal Models

Ioannis Ntzoufras

Department of Statistics, Athens University of Economics and Business, Greece

Claudia Tarantola *

Department of Economics and Management, University of Pavia, Italy

Monia Lupparelli

Department of Statistical Sciences, University of Bologna, Italy

December 21, 2017

Abstract

Bayesian methods for graphical log-linear marginal models has not been developed in the same extend as traditional frequentist approaches. In this work, we introduce a novel Bayesian approach for quantitative learning for such models. They belong to curved exponential families that are difficult to handle from a Bayesian perspective. Furthermore, the likelihood cannot be analytically expressed as a function of the marginal log-linear interactions, but only in terms of cell counts or probabilities. Posterior distributions cannot be directly obtained, and MCMC methods are needed. Finally, a well-defined model requires parameter values that lead to compatible marginal probabilities. Hence, any MCMC should account for this important restriction. We construct a fully automatic and efficient MCMC strategy for quantitative learning for graphical log-linear marginal models that handles these problems. While the prior is expressed in terms of the marginal log-linear interactions, we build an MCMC algorithm which employs a proposal on the probability parameter space. The corresponding proposal on the marginal log-linear interactions is obtained via parameter transformations. By this strategy, we achieve to move within the desired target space. At each step we directly work with well-defined probability distributions. Moreover, we can exploit a conditional conjugate setup to build an

*Address for correspondence: Claudia Tarantola, Department of Economics and Management, University of Pavia, Pavia, Italy.E-mail: claudia.tarantola@unipv.it

efficient proposal on probability parameters. The proposed methodology is illustrated by a simulation study and a real dataset.

Keywords: Graphical Models, Marginal Log-Linear Parameterisation, Markov Chain Monte Carlo Computation.

1 Introduction

Statistical models which impose restrictions on marginal distributions of categorical data have received considerable attention especially in social and economic sciences; see, for example, in Bergsma *et al.* (2009). A particular appealing class is that of log-linear marginal models introduced by Bergsma and Rudas (2002), that includes as special cases log-linear and multivariate logistic models. The marginal log-linear interactions are estimated using the frequencies of appropriate marginal contingency table, and expressed in terms of log-odds ratios. This setup is important in cases where information is available for specific marginal associations via odds ratios (i.e. marginal log-linear interactions) or when partial information (i.e. marginals) is available.

Log-linear marginal models have been used to provide parameterisations for discrete graphical models; see Lupparelli *et al.* (2009), Rudas *et al.* (2010) and Evans and Richardson (2013). In particular, Lupparelli *et al.* (2009) used them to define a parameterisation for discrete graphical models of marginal independence represented by a bi-directed graph. The absence of an edge in the bi-directed graph indicates marginal independence, and the corresponding marginal log-linear interactions (i.e. the corresponding log-odds ratio) are constrained to zero.

Despite the increasing interest in the literature for graphical log-linear marginal models, Bayesian analysis has not been developed as much as traditional methods. Some context specific results have been presented by e.g. Silva and Ghahramani (2009a), Bartolucci *et al.* (2012) and Ntzoufras and Tarantola (2013). For graphical log-linear marginal models, no conjugate analysis is available. Therefore, Markov chain Monte Carlo (MCMC) methods must be employed. Nevertheless, the likelihood of the model cannot be analytically expressed as a function of the marginal log-linear interactions. This creates additional difficulties on the implementation of MCMC methods since, at each step, an iterative procedure needs to be applied in order to calculate the cell probabilities and consequently the model likelihood. Moreover, in order to have a well-defined model of marginal independence, we need to construct an algorithm which generates parameter values that lead to a joint probability distribution with compatible marginals. To achieve this, we need an MCMC scheme which moves within the restricted space of parametrisations satisfying the conditions induced by the compatibility of marginal distributions.

In this paper we construct a novel, fully automatic, efficient MCMC strategy for quantitative learning

for graphical log-linear marginal models that handles the problems previously discussed. We assign a suitable prior distribution on the marginal log-linear parameter vector, while the proposal is expressed in terms of the probability parameters. The proposal distribution of marginal log-linear interactions is constructed by simply transforming generated candidate values of probability parameters. The corresponding proposal density is directly available by implementing standard theory about functions of random variables. The advantages of this strategy are clear: the joint distribution factorises under certain conditional independence models, and the likelihood can be directly expressed in terms of probability parameters. Furthermore, efficient proposal distributions can be constructed applying the conditional conjugate approach of Ntzoufras and Tarantola (2013), that exploit the representation of the model in terms of a Direct Acyclic Graph (DAG). Assigning prior distribution on the marginal log-linear interactions rather than on the probability parameters represents a novel approach and is particularly handy in the presence of informative prior about odds for specific marginal associations. For instance, symmetry constraints, vanishing high-order associations or further prior information about the joint and marginal distributions can be easily specified by setting linear constraints on marginal log-linear terms instead of non-linear multiplicative constraints on the probability space.

The plan of the paper is as follows. In Section 2, we introduce discrete graphical models of marginal independence and the marginal log-linear parameterisation. In Section 3, we describe the considered prior set-up. Section 4 is devoted to the proposed MCMC strategies. The methodology is illustrated in Section 5 which presents a simulation study and a real data analysis. In Section 6, we conclude with a brief discussion and ideas for future research.

2 Model Specification and Parameterisation

In this section we briefly introduce discrete graphical models of marginal independence, the related notation and terminology, and the corresponding marginal log-linear parameterisation.

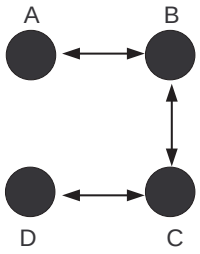
A bi-directed graph $G = (\mathcal{V}, E)$, is a graph with vertex set $\mathcal{V}$, and edge set E, such that $(v, u) \in E$ if and only if $(u, v) \in E$. Following Richardson (2003) edges are represented via bi-directed arrows. An alternative representation, proposed by Cox and Wermuth (1993), is by undirected dashed edges.

We consider a set of random variables $X_{\mathcal{V}} = \big(X_v,\ v \in \mathcal{V}\big)$, each one taking values $i_v \in \mathcal{I}_v$; where $\mathcal{I}_v$ is the set of possible levels for variable v. The cross-tabulation of variables $X_{\mathcal{V}}$ produces a $|\mathcal{V}|$-way contingency table with cell frequencies $\boldsymbol{n} = \big(n(i),\ i \in \mathcal{I}\big)$ where $\mathcal{I} = \times_{v \in \mathcal{V}} \mathcal{I}_v$. We further assume that $\boldsymbol{n} \sim Multinomial\big(\boldsymbol{p}, N\big)$ with $\boldsymbol{p} = \big(p(i),\ i \in \mathcal{I}\big)$; $p(i)$ is the joint probability for cell $i \in \mathcal{I}$, and $N = \sum_{i \in \mathcal{I}} n(i)$.

A bi-directed graph G is used to represent marginal independencies between variables $X_{\mathcal{V}}$ which are

expressed as non-linear constraints over the set of the joint probabilities $\boldsymbol{p}$. The list of independencies implied by a bi-directed graph can be obtained using the pairwise Markov property (Cox and Wermuth, 1993) and the connected set Markov property (Richardson, 2003). For discrete variables the connected set Markov property implies the pairwise Markov property, whereas the converse is not generally true. Following Drton and Richardson (2008), we define a discrete graphical model of marginal independence as the family of probability distributions for $X_{\mathcal{V}}$ that satisfy the connected set Markov property. For example the bi-directed graph in Figure 1 encodes the marginal independencies $X_{\{A,B\}} \perp\!\!\!\perp X_D$ and $X_A \perp\!\!\!\perp X_{\{D,C\}}$ under the connect set Markov property.

Figure 1: Example of bi-directed graph



The marginal log-linear parameterisation for bi-directed graphs has been proposed by Lupparelli (2006) and Lupparelli *et al.* (2009); it is based on the class of log-linear marginal models of Bergsma and Rudas (2002). According to Bergsma and Rudas (2002) the parameter vector $\boldsymbol{\lambda}$, containing the marginal log-linear interactions, can be obtained as

$$\boldsymbol{\lambda} = \boldsymbol{C} \log\left(\boldsymbol{M}\boldsymbol{P}\right) \text{ with } \boldsymbol{P} = \text{vec}(\boldsymbol{p}), \tag{1}$$

where vec($\boldsymbol{p}$) is a vector of dimension $|\mathcal{I}|$ obtained by rearranging the elements $\boldsymbol{p}$ in a reverse lexicographical ordering of the corresponding variable levels, with the level of the first variable changing first. Each marginal log-linear interaction satisfies identifiability constraints (here sum-to-zero constraints), and this is achieved via an appropriate contrast matrix $\boldsymbol{C}$. Each interaction is calculated from a specific marginal table identified via the marginalisation matrix $\boldsymbol{M}$. More precisely, $\boldsymbol{M}$ specifies from which marginal each element of $\boldsymbol{\lambda}$ is calculated. Each interaction is described by two sets of the variables, one set that refers to the marginal table in use and a second set (a subset of the first one) that identifies which variables are involved in this specific interaction. Finally, the first order interactions correspond to the main effects. Details for the construction of $\boldsymbol{C}$ and $\boldsymbol{M}$ are available in Appendices B and and C.

A graphical model of marginal independence is defined by zero constraints on specific marginal log-linear interactions. More precisely, we apply the following procedure presented by Lupparelli (2006) and Lupparelli *et al.* (2009): (i) define a hierarchical ordering (see Bergsma and Rudas, 2002) of the marginals

corresponding to disconnected sets of the bi-directed graph; (ii) append the marginal corresponding to the full table at the end of the list if it is not already included; (iii) for every marginal table estimate all interactions that have not been already obtained from the marginals preceding it in the ordering; (iv) for every marginal table corresponding to a disconnected set of G, restrict the highest order log-linear interaction to zero. Hence, the graphical structure imposes constraints of the type

$$\boldsymbol{K} \log \Big(\boldsymbol{M} \boldsymbol{P} \Big) = 0$$

with $\boldsymbol{K}$ being the sub-matrix of $\boldsymbol{C}$ for which the corresponding elements of $\boldsymbol{\lambda}$ are restricted to zero.

Note that, this parameterisation depends on the ordering of the marginals selected in step (i). Furthermore, it does not always satisfy variation independence; see Lupparelli *et al.* (2009), Rudas *et al.* (2010), and Evans and Richardson (2013). If the marginal selected in step (i) are order decomposable then variation independence is guaranteed (Bergsma and Rudas, 2002).

3 Bayesian Model Set-up

3.1 Prior Specification for Marginal Log-linear Interactions

The model in equation (1) can be rewritten in the following extended form

$$\begin{pmatrix} \boldsymbol{\lambda}^{M_1} \\ \vdots \\ \boldsymbol{\lambda}^{M_m} \\ \vdots \\ \boldsymbol{\lambda}^{|\mathcal{M}|} \end{pmatrix} = \operatorname{diag}\left(\boldsymbol{C}_1, \cdots, \boldsymbol{C}_m, \cdots, \boldsymbol{C}_{|\mathcal{M}|} \right) \begin{pmatrix} \log \boldsymbol{P}^{M_1} \\ \vdots \\ \log \boldsymbol{P}^{M_m} \\ \vdots \\ \log \boldsymbol{P}^{|\mathcal{M}|} \end{pmatrix}, \tag{2}$$

where $\mathcal{M} = \{M_1, \ldots, M_m \ldots, M_{|\mathcal{M}|}\}$ is the set of marginals under consideration, $\boldsymbol{\lambda}^{M_m}$ is the parameter vector obtained from the marginal probability table $\boldsymbol{p}^{M_m}$ which is re-arranged to a vector denoted by $\boldsymbol{P}^{M_m}$ for all $m = 1, 2, \ldots, |\mathcal{M}|$. The contrast matrix $\boldsymbol{C}$ is a block diagonal matrix with elements $\boldsymbol{C}_m$. Each sub-matrix $\boldsymbol{C}_m$ is obtained by inverting the design matrix $\mathbf{X}_{M_m}$ of the saturated model fitted on marginal M_m, and deleting rows corresponding to interactions that are not estimated from that specific marginal table; see Appendix C for details. From (2), we directly obtain that

$$\boldsymbol{\lambda}^{M_m} = \boldsymbol{C}_m \log \boldsymbol{P}^{M_m} \text{ for all } M_m \in \mathcal{M}.$$

Every $\boldsymbol{\lambda}^{M_m}$ may contain interactions that are constrained to zero due the graphical structure G and the induced contrast matrix. In the following we focus only on non-zero elements of $\boldsymbol{\lambda}$, on which we

assign a suitable prior distribution. We denote by $\vec{\boldsymbol{\lambda}}$ the set of elements of $\boldsymbol{\lambda}$ not restricted to zero, that is

$$\vec{\boldsymbol{\lambda}} \;=\; \left(\vec{\boldsymbol{\lambda}}^{M_m}; M_m \in \mathcal{M}\right) \text{ with } \vec{\boldsymbol{\lambda}}^{M_m} = \left(\lambda_j^{M_m} : \lambda_j^{M_m} \neq 0, j = 1, \ldots, \mathrm{R}_{C_m}\right),$$

where R_{C_m} is the number of rows of the contrast matrix $\boldsymbol{C}_m$ for marginal $M_m \in \mathcal{M}$.

When no information is available about $\vec{\boldsymbol{\lambda}}$, we can work separately on each element of $\vec{\boldsymbol{\lambda}}$, assigning suitable normal prior distributions with large variance to express ignorance, i.e.

$$f(\vec{\lambda}_j) \sim N(0, \sigma_j^2) \text{ for } j = 1, 2, \ldots, d_{\vec{\lambda}},$$

where $d_{\vec{\lambda}}$ is the number of elements of $\vec{\boldsymbol{\lambda}}$. Under ordered decomposability of marginals, we consider as the prior on $\vec{\boldsymbol{\lambda}}$ the product of the previous quantities. Otherwise, when ordered decomposability is not met, the prior on $\vec{\boldsymbol{\lambda}}$ may be specified proportional to the above product of independent normal distributions with support on the corresponding restricted parameter region.

A more sophisticated approach can be based on the prior suggestion of Dellaportas and Forster (1999) for standard log-linear models of the form $\log \boldsymbol{\mu} = \boldsymbol{X}\boldsymbol{\beta}$. They suggested the following prior distribution

$$\boldsymbol{\beta} \sim N\Big(\boldsymbol{\theta},\, 2|\mathcal{I}|\big(\boldsymbol{X}^T\boldsymbol{X}\big)^{-1}\Big) \text{ with } \boldsymbol{\theta} = (\log \overline{\boldsymbol{n}},\, 0, \ldots,\, 0)^T, \tag{3}$$

where $\boldsymbol{\mu}$ is the vector of the expected number of cell frequencies, $\boldsymbol{X}$ is a design matrix and $|\mathcal{I}| = \prod_{v \in \mathcal{V}} |\mathcal{I}_v|$ is the number of cells of the contingency table; see Knuiman and Speed (1988) for an earlier recommendation.

This prior was suggested as a default choice for model comparison of log-linear models and it arises naturally for log-linear marginal models since within each marginal we essentially work by fitting standard log-linear models. It was obtained after matching the prior moments of other default priors used for the probability parameters in conjugate analysis of graphical models such as the Jeffreys and the Perks prior distributions. Moreover, the prior distribution induced for the log-odds ratios remains the same even if extra, marginally independent, categorical variables are added in the model. Finally, such prior arises naturally in our context since the adopted parameterisation allow us to work by fitting standard log-linear models within each marginal. These are the main reasons why we recommend to adopt this prior here.

The prior vector $\boldsymbol{\theta}$ has all its elements equal to zero except the first one which is equal to the logarithm of the average number of observations per cell $\overline{\boldsymbol{n}}$. Under sum-to-zero constraints, $\boldsymbol{X}^T\boldsymbol{X}$ is a block diagonal matrix resulting to a set of independent priors for interactions referring to different set of variables.

In order to construct the prior distribution on $\vec{\boldsymbol{\lambda}}$, we work separately on each single set $\boldsymbol{\lambda}^{M_m}$ obtained from marginal M_m and we proceed as follows. Let $\boldsymbol{\lambda}_S^{M_m}$ be the parameter vector for the saturated model that can be estimated from marginal M_m; by construction, it coincides with the parameter vector of the

saturated standard log-linear model obtained from this marginal. In terms of Dellaportas and Forster (1999) parametrisation, $\boldsymbol{\lambda}_S^{M_m}$ can be written as $\boldsymbol{\lambda}_S^{M_m} = \boldsymbol{\beta}^{M_m} - \log(N)\,\mathbf{X}_{M_m}^{-1}\mathbf{1}$. Hence, from (3), the default prior for $\boldsymbol{\lambda}_S^{M_m}$ is given by

$$\boldsymbol{\lambda}_S^{M_m} \sim N\Big(\boldsymbol{\theta} - \log(N)\,\mathbf{X}_{M_m}^{-1}\mathbf{1}\ ,\ 2|\mathcal{I}_{M_m}|\left(\boldsymbol{X}_{M_m}^T\boldsymbol{X}_{M_m}\right)^{-1}\Big). \tag{4}$$

Finally, the prior of $\vec{\boldsymbol{\lambda}}^{M_m}$ is obtained via marginalisation from (4) since $\vec{\boldsymbol{\lambda}}^{M_m}$ is a subset of $\boldsymbol{\lambda}_S^{M_m}$. When using sum-to-zero constraints, then $\boldsymbol{X}_{M_m}^{-1}\mathbf{1} = (1, 0, \ldots, 0)^T$ resulting to a prior mean equal to $\theta_\jmath = 0$ for all $\vec{\boldsymbol{\lambda}}_\jmath^{M_m}$ except for the intercept for which the prior mean is given by $\log \overline{\boldsymbol{n}} - \log(N)$. This prior is greatly simplified to a product of independent $N(0, 2)$ for all marginal log-linear interactions (except for the intercept) in the case of binary variables. Finally, the prior for the full parameter vector $\vec{\boldsymbol{\lambda}}$ is obtained as a product of the priors on $\vec{\boldsymbol{\lambda}}^{M_m}$.

3.2 Likelihood Specification and Posterior Inference

The likelihood cannot directly be expressed in terms of $\boldsymbol{\lambda}$ (or equivalently $\vec{\boldsymbol{\lambda}}$) but only as a function of the probability parameter

$$\begin{aligned} f(\boldsymbol{n}|\boldsymbol{\lambda}) &= \frac{\Gamma(N+1)}{\prod_{i\in\mathcal{I}}\Gamma\big(n(i)+1\big)}\prod_{i\in\mathcal{I}}\wp_i(\boldsymbol{\lambda})^{n(i)}, \\ \text{where } \wp_i(\boldsymbol{\lambda}) &\equiv \Big\{p(i) : \boldsymbol{\lambda} = \boldsymbol{C}\log\Big(\boldsymbol{MP}\Big)\Big\},\ \text{for all } i \in \mathcal{I}\ . \end{aligned} \tag{5}$$

Unfortunately, in order to obtain $\wp_i(\boldsymbol{\lambda})$, or equivalently $\boldsymbol{P}$, from (1), we need to implement an iterative algorithm, and therefore the likelihood cannot be written in a closed form expression, see, for example, Bergsma and Rudas (2002), and Lupparelli *et al.* (2009). As a consequence of this, the corresponding posterior distribution of λ (or equivalently $\vec{\boldsymbol{\lambda}}$), cannot be evaluated straight away. Hence, MCMC methods are needed for posterior inference of $\vec{\boldsymbol{\lambda}}$.

Although simple Metropolis-Hastings schemes on the marginal log-linear interactions can be implemented to estimate the posterior distributions of the parameter vector, such algorithmic strategy will be inefficient. In fact, in every MCMC iteration the joint probabilities corresponding to the proposed values of $\boldsymbol{\lambda}$ need to be calculated using an iterative algorithm, and this will slow down the MCMC sampler and increase the autocorrelation. Moreover, there could be problems when calculating the joint probability parameters especially if the proposed set of marginal log-linear interactions are not variation independent. If variation independence is not satisfied the resulting set of probabilities are not compatible, i.e. the obtained probabilities may not add to one or be outside the zero-one interval (Qaqish and Ivanova, 2006). For these reasons, in the following section, we propose alternative MCMC strategy based on the probability representation of the model.

For comparative purposes, we have implemented a "vanilla" random walk algorithm MCMC on $\vec{\boldsymbol{\lambda}}$ (referred to as RW-λ) which proposes to change each log-linear parameter vector independently for every single marginal. Nevertheless, this algorithm does not ensures automatically that a well-defined joint probability distribution is always obtained. This problem can be avoided by imposing, at each step of the random walk algorithm, cumbersome restrictions on the conditional posterior distributions that will considerably delay the MCMC sampler.

4 Probability Based MCMC Samplers

4.1 Initial Set-up and Data Augmentation for MCMC

Following the notation of Ntzoufras and Tarantola (2013), we can divide the class of graphical log-linear marginal models in two major categories: homogeneous and non-homogeneous models. Both of them can be described by a DAG Markov equivalent over the observed margins; see Pearl and Wermuth (1994), Drton and Richardson (2008) and Silva and Ghahramani (2009a,b). Nevertheless, while homogeneous models can be represented via a DAG with the same vertex set, non-homogeneous ones requires the inclusion of some additional latent variables. The advantage of the DAG representation is that the joint probability over the augmented variable space (including both observed and latent variables) can be written using the standard factorisation. An example of the DAG representation is provided in Figure 2. The model can then be parameterised in terms of a minimal set of marginal and conditional probability parameters denoted here with $\boldsymbol{\Pi}$ ($\boldsymbol{\Pi}^D$ in the original paper) on which we can implement conjugate analysis based on products of Dirichlet distributions.

We approach Bayesian inference for non-homogenous bi-directed graph models exploiting the class of equivalence in terms of independence between bi-directed graphs and DAGs with latent variables; see for instance Pearl and Wermuth (1994). However, it is worth noticing that we do not account for additional non-independence constraints (in general inequality constraints in the probability space) which may arise in marginal DAGs ignoring some latent variables. Nowadays defining the exact class of marginal DAGs still represents an open issue; see Evans (2016) who discusses possible approximations. Moreover, these inequality constraints are not of great interest as they do not have a straightforward interpretation. Nevertheless, it is worth noting that if non-independence constraints are included after marginalising over the latent variable, the resulting likelihood is a close approximation of the true one.

Using the methodology of Ntzoufras and Tarantola (2013), we construct an MCMC sampler based on proposing values in terms of joint probabilities, avoiding compatibility problems. If the model is homogeneous the dimension of $\boldsymbol{\Pi}$ is the same as the dimension of $\vec{\boldsymbol{\lambda}}$. This is not true for non-homogeneous

Figure 2: Bi-directed 4-chain graph and the corresponding Markov equivalent DAG over the observed margin.

(a) Bi-directed graph (b) DAG representation

models since the dimension of $\mathbf{\Pi}$ is greater than the dimension of $\vec{\boldsymbol{\lambda}}$. In this case we need to augment the parameter space in order to implement Metropolis-Hastings algorithm.

In the following we denote with $\boldsymbol{\lambda}^{\mathcal{A}}$ the augmented set of marginal log-linear interactions; $\boldsymbol{\lambda}^{\mathcal{A}} = (\vec{\boldsymbol{\lambda}}, \boldsymbol{\xi})$ with $\boldsymbol{\xi}$ having dimension equal to $\dim(\mathbf{\Pi}) - \dim(\vec{\boldsymbol{\lambda}})$. If the model is homogeneous $\boldsymbol{\lambda}^{\mathcal{A}} = \vec{\boldsymbol{\lambda}}$.

More precisely, for any graphical log-linear marginal model G we can obtain a Markov equivalent DAG over the observed margins, denoted by D_G, with augmented vertex set $\mathcal{A} = \{\mathcal{V}, \mathcal{L}\}$, where $\mathcal{L}$ is the set of additional latent variables; if G is homogeneous then $\mathcal{L} = \emptyset$ and $\mathcal{A} = \mathcal{V}$. Under this approach, the joint probabilities can be written as

$$p(i) = \sum_{i_{\mathcal{L}} \in \mathcal{I}_{\mathcal{L}}} p^{\mathcal{A}}(i, i_{\mathcal{L}}) \text{ for } i \in \mathcal{I}_{\mathcal{V}}, \tag{6}$$

$$p^{\mathcal{A}}(i, i_{\mathcal{L}}) = p^{\mathcal{A}}(i^*) = \prod_{v \in \mathcal{A}} \pi_{v|pa(v)} \left(i^*_v | i^*_{pa(v)} \right) \text{ for } i^* = (i, i_{\mathcal{L}}) \in \mathcal{I}_{\mathcal{A}}, \tag{7}$$

and the probability parameter set is given by

$$\mathbf{\Pi} = \text{vec}\Big(\pi_{v|pa(v)}(i^*_v | i^*_{pa(v)});\ i^*_v \in \mathcal{I}_v \setminus \Big\{ |\mathcal{I}_v| \Big\}, i^*_{pa(v)} \in \mathcal{I}_{pa(v)}, v \in \mathcal{A} \Big),$$

where $p^{\mathcal{A}}(i^*) = P\big(X_{\mathcal{V}} = i^*_{\mathcal{V}}, X_{\mathcal{L}} = i^*_{\mathcal{L}}\big)$ is the joint probability for the observed variables $X_{\mathcal{V}}$ and the latent variables $X_{\mathcal{L}}$, $pa(v)$ stands for the parents set of v in graph D_G, $\pi_{v|pa(v)}\big(i^*_v | i^*_{pa(v)}\big) = P\big(X_v = i^*_v | X_{pa(v)} = i^*_{pa(v)}\big)$ is the conditional probability of each variable X_v given variables $X_{pa(v)}$ in the parent set $pa(v)$ of v. By using the induced augmented likelihood representation, we are able to construct a Gibbs sampler based on the conditional conjugate Dirichlet prior distributions on $\mathbf{\Pi}$ (Ntzoufras and Tarantola, 2013).

4.2 The general algorithm

The posterior distribution of the augmented set of marginal log-linear interactions is given by

$$f(\boldsymbol{\lambda}^{\mathcal{A}}|\boldsymbol{n}) \quad \propto \quad f\Big(\boldsymbol{n}|_{\wp}(\boldsymbol{\lambda})\Big) f(\vec{\boldsymbol{\lambda}}) f(\boldsymbol{\xi}),$$

where $f(\boldsymbol{\xi})$ is a pseudo prior used for the additional parameters.

We consider a Metropolis-Hastings algorithm which can be summarised by the following steps:

For $t = 1, \ldots, T$, repeat the following steps:

1. Propose a new vector $\boldsymbol{\Pi}'$ from $q(\boldsymbol{\Pi}'|\boldsymbol{\Pi}^{(t)})$; where $\boldsymbol{\Pi}^{(t)}$ are the values of $\boldsymbol{\Pi}$ at t iteration.

2. From $\boldsymbol{\Pi}'$, calculate the proposed joint probabilities $\boldsymbol{p}'$ (for the observed table) using equations (6) and (7).

3. From $\boldsymbol{p}'$, calculate $\boldsymbol{\lambda}'$ using (1) and then obtain the corresponding non-zero elements $\vec{\boldsymbol{\lambda}}'$.

4. Set $\boldsymbol{\xi}' = \boldsymbol{\Pi}'_{\xi}$; where $\boldsymbol{\Pi}'_{\xi}$ is a pre-specified subset of $\boldsymbol{\Pi}'$ of dimension $\dim(\boldsymbol{\Pi}) - \dim(\vec{\boldsymbol{\lambda}})$.

5. Accept the proposed move with probability $\alpha = \min(1, A)$ with

$$\begin{aligned} A &= \frac{f\Big(\boldsymbol{n}|_{\wp}(\boldsymbol{\lambda}')\Big) f(\vec{\boldsymbol{\lambda}}') f\left(\boldsymbol{\xi}'\right) q\big(\vec{\boldsymbol{\lambda}}^{(t)}, \boldsymbol{\xi}^{(t)}|\vec{\boldsymbol{\lambda}}', \boldsymbol{\xi}'\big)}{f\Big(\boldsymbol{n}|_{\wp}\big(\boldsymbol{\lambda}^{(t)}\big)\Big) f\big(\vec{\boldsymbol{\lambda}}^{(t)}\big) f(\boldsymbol{\xi}^{(t)}) q\big(\vec{\boldsymbol{\lambda}}', \boldsymbol{\xi}'\big|\vec{\boldsymbol{\lambda}}^{(t)}, \boldsymbol{\xi}^{(t)}\big)} \\ &= \frac{f(\boldsymbol{n}|\boldsymbol{\Pi}') f(\vec{\boldsymbol{\lambda}}') f(\boldsymbol{\xi}') q(\boldsymbol{\Pi}^{(t)}|\boldsymbol{\Pi}')}{f(\boldsymbol{n}|\boldsymbol{\Pi}^{(t)}) f\big(\vec{\boldsymbol{\lambda}}^{(t)}\big) f(\boldsymbol{\xi}) q(\boldsymbol{\Pi}'|\boldsymbol{\Pi}^{(t)})} \times \text{abs}\left(\frac{\mathcal{J}\Big(\boldsymbol{\Pi}^{(t)}, \vec{\boldsymbol{\lambda}}^{(t)}, \boldsymbol{\xi}^{(t)}\Big)}{\mathcal{J}\Big(\boldsymbol{\Pi}', \vec{\boldsymbol{\lambda}}', \boldsymbol{\xi}'\Big)}\right), \end{aligned} \tag{8}$$

   where abs$(\cdot)$ stands for the absolute value, $\boldsymbol{\Pi}_{\xi} = \boldsymbol{\xi}$, and $\mathcal{J} = \mathcal{J}(\boldsymbol{\Pi}, \vec{\boldsymbol{\lambda}}, \boldsymbol{\xi})$ is the determinant of the jacobian matrix of the transformation $\boldsymbol{\Pi} = g(\vec{\boldsymbol{\lambda}}, \boldsymbol{\xi})$ specified by Equations (6), (7), and (1). Similarly to Step 1, $\vec{\boldsymbol{\lambda}}^{(t)}$, $\boldsymbol{\xi}^{(t)}$ and $\boldsymbol{\lambda}^{(t)}$ are used to denote the values of the corresponding parameters in the current iteration t of the algorithm.

6. If the move is accepted, then set $\boldsymbol{\Pi}^{(t+1)} = \boldsymbol{\Pi}'$, $\boldsymbol{\xi}^{(t+1)} = \boldsymbol{\xi}'$, and $\vec{\boldsymbol{\lambda}}^{(t+1)} = \vec{\boldsymbol{\lambda}}'$ otherwise set $\boldsymbol{\Pi}^{(t+1)} = \boldsymbol{\Pi}^{(t)}$ and $\vec{\boldsymbol{\lambda}}^{(t+1)} = \vec{\boldsymbol{\lambda}}^{(t)}$.

The pseudo-parameter vector $\boldsymbol{\xi}$ is used only to retain the dimension balance between the marginal log-linear parameterisation and the probability parametrisation used in Ntzoufras and Tarantola (2013). Furthermore, it is directly matched to specific probability parameters of bi-directed graph G. Hence, we can indirectly "eliminate" its effect on the algorithm by assuming that its elements are uniformly distributed in the zero-one interval. Under this view, we set $f(\xi_i) = I_{\{0<\xi_i<1\}}$ having as a result the

elimination of the ratio $f(\boldsymbol{\xi}')/f(\boldsymbol{\xi}^{(t)})$ from (8). In the following, we consider this choice in order to simplify all proposed algorithms.

A good choice of the proposal $q(\boldsymbol{\Pi}'|\boldsymbol{\Pi}^{(t)})$ will lead to high (close to one) acceptance rates. Therefore, an efficient proposal for the Metropolis Hastings scheme described in this section, intuitively seems to be the following

$$q(\boldsymbol{\Pi}'|\boldsymbol{\Pi}^{(t)}) = \sum_{n_{\mathcal{A}}} f_q\big(\boldsymbol{\Pi}'|\boldsymbol{n}'^{\mathcal{A}}\big) f\big(\boldsymbol{n}'^{\mathcal{A}}|\boldsymbol{\Pi}^{(t)}, \boldsymbol{n}\big), \tag{9}$$

where $f\big(\boldsymbol{n}^{\mathcal{A}}|\boldsymbol{\Pi}, \boldsymbol{n}\big)$ is the distribution of counts $\boldsymbol{n}^{\mathcal{A}}$ given the observed frequency table $\boldsymbol{n}$ and the probability parameter set $\boldsymbol{\Pi}$ of the augmented table induced by $\mathcal{A}$; and $f_q\big(\boldsymbol{\Pi}|\boldsymbol{n}^{\mathcal{A}}\big)$ is the conditional posterior distribution of the probability parameter vector $\boldsymbol{\Pi}$ given a proposed set of augmented data $\boldsymbol{n}^{\mathcal{A}}$. Considering all possible configurations in (9) within each MCMC iteration is cumbersome and time consuming. One solution can be obtained by using a random sub-sample of $\boldsymbol{n}^{\mathcal{A}}$. Hence, we construct our MCMC by employing just one realisation of $\boldsymbol{n}^{\mathcal{A}}$ which will play the role of intermediate nuisance parameter that facilitates the construction of a sensible proposal distribution. This approach corresponds to an MCMC scheme with a target posterior distribution given by

$$f(\boldsymbol{\lambda}^{\mathcal{A}}, \boldsymbol{n}^{\mathcal{A}}|\boldsymbol{n}) \quad \propto \quad f\big(\boldsymbol{n}|\wp(\boldsymbol{\lambda})\big) f(\vec{\boldsymbol{\lambda}}) f(\boldsymbol{\xi}) f(\boldsymbol{n}^{\mathcal{A}}).$$

The parameter vector $\boldsymbol{n}^{\mathcal{A}}$ plays the role of (nuisance) augmented data with $f(\boldsymbol{n}^{\mathcal{A}})$ playing the role of a pseudo-prior. In order to simplify the MCMC configuration, this pseudo-prior can be set specified to be the uniform distribution over all possible configurations of $\boldsymbol{n}^{\mathcal{A}}$. Under this formulation, the acceptance probability in the Metropolis-Hastings is equal to $\alpha = \min(1, A)$ with A given by

$$A \quad = \quad \frac{f\big(\boldsymbol{n}|\boldsymbol{\Pi}'\big) f\big(\vec{\boldsymbol{\lambda}}'\big) f_q\big(\boldsymbol{\Pi}^{(t)}|\boldsymbol{n}^{\mathcal{A}(t)}\big) f\big(\boldsymbol{n}^{\mathcal{A}(t)}|\boldsymbol{\Pi}', \boldsymbol{n}\big)}{f\big(\boldsymbol{n}|\boldsymbol{\Pi}^{(t)}\big) f\big(\vec{\boldsymbol{\lambda}}^{(t)}\big) f_q\big(\boldsymbol{\Pi}'|\boldsymbol{n}'^{\mathcal{A}}\big) f\big(\boldsymbol{n}'^{\mathcal{A}}|\boldsymbol{\Pi}^{(t)}, \boldsymbol{n}\big)} \times \text{abs}\left(\frac{\mathcal{J}\big(\boldsymbol{\Pi}^{(t)}, \vec{\boldsymbol{\lambda}}^{(t)}, \boldsymbol{\xi}^{(t)}\big)}{\mathcal{J}\big(\boldsymbol{\Pi}', \vec{\boldsymbol{\lambda}}', \boldsymbol{\xi}'\big)}\right). \tag{10}$$

In (10), the probability functions $f\big(\boldsymbol{n}'^{\mathcal{A}}|\boldsymbol{\Pi}, \boldsymbol{n}\big)$ and $f\big(\boldsymbol{n}^{\mathcal{A}(t)}|\boldsymbol{\Pi}', \boldsymbol{n}\big)$ are readily available from the model construction and the likelihood representation of the augmented table. We only need to specify $f_q\big(\boldsymbol{\Pi}'|\boldsymbol{n}'^{\mathcal{A}}\big)$ which is the first component of (9) and it has the form of a posterior conditionally on the frequencies of the augmented table. For this component, we can exploit the conditional conjugate approach of Ntzoufras and Tarantola (2013). In order to do so, we consider as a "prior" $f_q(\boldsymbol{\Pi})$ a product of Dirichlet distributions in order to obtain a conjugate "posterior" distribution $f_q(\boldsymbol{\Pi}'|\boldsymbol{n}'^{\mathcal{A}})$. Under this approach, and by further considering that $f\big(\boldsymbol{n}|\boldsymbol{\Pi}\big) f\big(\boldsymbol{n}^{\mathcal{A}}|\boldsymbol{\Pi}, \boldsymbol{n}\big) = f\big(\boldsymbol{n}^{\mathcal{A}}|\boldsymbol{\Pi}\big)$, then (10) is further simplified to

$$A \quad = \quad \frac{f\big(\boldsymbol{n}^{\mathcal{A}(t)}|\boldsymbol{\Pi}'\big) f\big(\vec{\boldsymbol{\lambda}}'\big) f_q\big(\boldsymbol{\Pi}^{(t)}|\boldsymbol{n}^{\mathcal{A}(t)}\big)}{f\big(\boldsymbol{n}'^{\mathcal{A}}|\boldsymbol{\Pi}^{(t)}\big) f\big(\vec{\boldsymbol{\lambda}}^{(t)}\big) f_q\big(\boldsymbol{\Pi}'|\boldsymbol{n}'^{\mathcal{A}}\big)} \times \text{abs}\left(\frac{\mathcal{J}\big(\boldsymbol{\Pi}^{(t)}, \vec{\boldsymbol{\lambda}}^{(t)}, \boldsymbol{\xi}^{(t)}\big)}{\mathcal{J}\big(\boldsymbol{\Pi}', \vec{\boldsymbol{\lambda}}', \boldsymbol{\xi}'\big)}\right). \tag{11}$$

In the following of this manuscript, we will refer to this approach as the *probability-based independence sampler* (PBIS).

4.3 Prior Adjustment Algorithm

As we have already stated, although PBIS simplifies the MCMC scheme, the parameter space is still considerably extended by considering the augmented frequency table $\boldsymbol{n}^{\mathcal{A}}$. We can further simplify PBIS by using the following two-step procedure:

Step 1: Run the Gibbs sampler of Ntzoufras and Tarantola (2013).

Step 2: Use the sample of step 1 (or sub-sample of it) as a proposal in the general Metropolis-Hastings algorithm with acceptance rate (8).

Since the sample of Step 1 is obtained from an MCMC algorithm, auto-correlation will be present. A random (independent) sub-sample from the posterior distribution can be obtained by following different strategies. The most common approach can be obtained by using thinning, where we keep one observation for every set of K iterations. The thinning interval K can be easily defined by monitoring autocorrelation function or using the convergence diagnostic of Raftery and Lewis (1992) which is available in R packages such as CODA or BOA. A drawback of this approach is that we may end up having a considerably lower number of simulated observations. For this reason, we suggest to consider a random permutation of the full MCMC sample to "destroy" the induced autocorrelation. We believe that the effect of this strategy will be minimal, since this sample is only used as a proposal in the a second MCMC procedure. As a referee pointed out, by following either of these approaches, the sample variance will get close to the independent variance asymptotically. Although intuitively this approach will lead to a sample from the target posterior distribution (or a close approximation of it), to our knowledge no mathematical proof is available. Indeed empirical comparisons (see Section 5.1) indicate no differences between the posterior distribution obtained by a correctly thinned MCMC sample and the re-ordered sample or even with the one-shot MCMC sampler of Section 4.2.

Using such sample (or sub-sample) as proposed values within the Metropolis-Hasting algorithm is equivalent to using the posterior distribution $f_q(\boldsymbol{\Pi}|\boldsymbol{n})$ as proposal in (8), that is $q(\boldsymbol{\Pi}'|\boldsymbol{\Pi}^{(t)}) = f_q(\boldsymbol{\Pi}'|\boldsymbol{n})$. Under this proposal, (8) now simplifies to

$$A = \frac{f(\vec{\boldsymbol{\lambda}}')f_q(\boldsymbol{\Pi}^{(t)})}{f(\vec{\boldsymbol{\lambda}}^{(t)})f_q(\boldsymbol{\Pi}')} \times \text{abs}\left(\frac{\mathcal{J}\left(\boldsymbol{\Pi}^{(t)}, \vec{\boldsymbol{\lambda}}^{(t)}, \boldsymbol{\xi}^{(t)}\right)}{\mathcal{J}\left(\boldsymbol{\Pi}', \vec{\boldsymbol{\lambda}}', \boldsymbol{\xi}'\right)}\right). \tag{12}$$

We will refer to this algorithm as the the *prior-adjustment* algorithm (PAA) due to its characteristic to correct for the differences between the prior distributions used under the two parameterisations.

The "prior" distribution $f_q(\boldsymbol{\Pi})$ is only used to build the proposal. Therefore, it can be considered as a pseudo-prior. It does not influence the target posterior distribution but only affects the convergence rates

of PAA. We can choose the parameters of this pseudo-prior in such a way that (12) is maximized such that an optimal acceptance rate is achieved. When a non-informative prior distribution for $\vec{\boldsymbol{\lambda}}$ is used, all Dirichlet parameters involved in $f_q(\boldsymbol{\Pi})$ can be set equal to one. Under this choice, the effect of the pseudo-prior is eliminated from the proposal, leaving the data-likelihood to guide the MCMC algorithm. PAA is less computationally demanding that the single-run MCMC algorithm introduced in Section 4.2, since in we avoid four additional likelihood evaluations at each iteration required in the later.

4.4 The Jacobian

We conclude this section by providing analytical expressions of the Jacobian required in the acceptance probabilities within each MCMC step in Sections 4.2 and 4.3; see Equations (11) and (12). Specifically, the Jacobian terms are given by $\mathcal{J}(\boldsymbol{\Pi}, \vec{\boldsymbol{\lambda}}, \boldsymbol{\xi}) = \left| \dfrac{\partial \boldsymbol{\Pi}}{\partial(\vec{\boldsymbol{\lambda}}, \boldsymbol{\xi})} \right|$ and are simplified to

$$\begin{aligned} \mathcal{J}^{-1} = \left| \frac{\partial(\vec{\boldsymbol{\lambda}}, \boldsymbol{\xi})}{\partial \boldsymbol{\Pi}} \right| = \left| \frac{\partial(\vec{\boldsymbol{\lambda}}, \boldsymbol{\xi})}{\partial(\boldsymbol{\Pi}_{\xi}, \boldsymbol{\Pi}_{\setminus \xi})} \right| &= \begin{vmatrix} \dfrac{\partial \vec{\boldsymbol{\lambda}}}{\partial \boldsymbol{\Pi}_{\xi}} & \dfrac{\partial \vec{\boldsymbol{\lambda}}}{\partial \boldsymbol{\Pi}_{\setminus \xi}} \\ \dfrac{\partial \boldsymbol{\xi}}{\partial \boldsymbol{\Pi}_{\xi}} & \dfrac{\partial \boldsymbol{\xi}}{\partial \boldsymbol{\Pi}_{\setminus \xi}} \end{vmatrix} \\ &= \begin{vmatrix} \dfrac{\partial \vec{\boldsymbol{\lambda}}}{\partial \boldsymbol{\Pi}_{\xi}} & \dfrac{\partial \vec{\boldsymbol{\lambda}}}{\partial \boldsymbol{\Pi}_{\setminus \xi}} \\ \boldsymbol{I} & \boldsymbol{0} \end{vmatrix} = - \left| \frac{\partial \vec{\boldsymbol{\lambda}}}{\partial \boldsymbol{\Pi}_{\setminus \xi}} \right|, \end{aligned}$$

where $\boldsymbol{\Pi}_{\setminus \xi}$ is obtained from $\boldsymbol{\Pi}$ excluding the elements of $\boldsymbol{\Pi}_{\xi}$.

The elements of the Jacobian matrix are given by

$$\frac{\partial \lambda_k}{\partial \Pi_{\jmath}} = \sum_{l=1}^{\mathrm{c}_C} \left\{ C_{kl} \left(\sum_{\imath=1}^{|\mathcal{I}|} M_{l\imath} P_{\imath} \right)^{-1} \sum_{\imath=1}^{|\mathcal{I}|} M_{l\imath} \Delta_{\imath\jmath} \right\} \text{ with } \Delta_{\imath\jmath} = \frac{\partial P_{\imath}}{\partial \Pi_{\jmath}}, \tag{13}$$

where $P_{\imath}$ denote the $\imath$ element of $\boldsymbol{P}$, and c_C is the number of columns of the contrast matrix $\boldsymbol{C}$. For the saturated model, the above equation simplifies to

$$\frac{\partial \lambda_k}{\partial P_{\jmath}} = \sum_{l=1}^{\mathrm{c}_C} \frac{C_{kl}(M_{l\jmath} - M_{l|\mathcal{I}|})}{\sum_{\imath=1}^{|\mathcal{I}|} M_{l\imath} P_{\imath}}$$

since $\boldsymbol{\Pi} = \boldsymbol{P}$, and $P_{|\mathcal{I}|} = 1 - \sum_{\imath=1}^{|\mathcal{I}|-1} P_{\imath}$. More details on the calculation of (13) are provided in Appendix A.

In order to complete the specification of (13), we need to calculate the derivative terms $\Delta_{\imath\jmath}$. Let us now denote by i' the index of vector $\boldsymbol{P}$ such that $P_{i'} \equiv p(i)$. Moreover, the index $\jmath$ corresponds to a variable $u_{\jmath} \in \mathcal{A}$ such that $\Pi_{\jmath} \equiv \pi_{u_{\jmath}|pa(u_{\jmath})}\big(j_{u_{\jmath}}|j_{pa(u_{\jmath})}\big)$ for a specific cell j of the augmented table $\mathcal{I}^{\mathcal{A}}$. Therefore, for any $\imath = i'$, terms $\Delta_{\imath\jmath}$ are given by

$$\Delta_{\imath\jmath} = \Delta_{i'\jmath} = \frac{\partial P_{i'}}{\partial \Pi_{\jmath}} = \frac{\partial p(i)}{\partial \pi_{u_{\jmath}|pa(u_{\jmath})}\big(j_{u_{\jmath}}|j_{pa(u_{\jmath})}\big)} \text{ for every } i' \mapsto i \text{ and } \jmath \mapsto (u_{\jmath}, j). \tag{14}$$

For the computation of each Δ_{ij} we consider two different cases: (A) $u_j \in \mathcal{V}$ and (B) $u_j \in \mathcal{L}$. In the following, to simplify notation, we denote u_j by u. Furthermore, we indicate with $\mathcal{L}_u = \mathcal{L} \cap pa(u)$ the latent variables that are parents of u, and with $\mathcal{A}_u = \mathcal{V} \cup \{u\} \cup \mathcal{L}_u$.

For **case A**, when u is an observed variable, we obtain

$$\frac{\partial p(i)}{\partial \pi_{u|pa(u)}\big(j_u|j_{pa(u)}\big)} = \delta(i,j)\frac{p^{\mathcal{A}_u}\big(i, j_{\mathcal{L}_u}\big)}{\pi_{u|pa(u)}\big(i_u|j_{pa(u)}\big)} \tag{15}$$

with

$$\delta(i,j) = \begin{cases} 1 & \text{if } i_u = j_u < |\mathcal{I}_u| \text{ and } i_{pa(u)\setminus\mathcal{L}} = j_{pa(u)\setminus\mathcal{L}} \\ -1 & \text{if } j_u \neq i_u = |\mathcal{I}_u| \text{ and } i_{pa(u)\setminus\mathcal{L}} = j_{pa(u)\setminus\mathcal{L}} \\ 0 & \text{if } j_u \neq i_u < |\mathcal{I}_u| \text{ or } \quad i_{pa(u)\setminus L} \neq j_{pa(u)\setminus\mathcal{L}} \end{cases}, \tag{16}$$

where

$$p^{\mathcal{A}_u}\big(i, j_{\mathcal{L}_u}\big) = \begin{cases} P\Big(X_{\mathcal{V}} = i, X_{\mathcal{L}\cap pa(u)} = j_{\mathcal{L}\cap pa(u)}\Big) & \mathcal{L}_u \neq \emptyset \\ p(i) & \mathcal{L}_u = \emptyset \end{cases}.$$

For **case B**, when u is a latent variable, $pa(u) = \emptyset$ due to the structure of the DAG representation. Hence, the derivative is given by

$$\frac{\partial p(i)}{\partial \pi_{u|pa(u)}\big(j_u|j_{pa(u)}\big)} = \frac{\partial p(i)}{\partial \pi_u\big(j_u\big)} = \frac{p^{\mathcal{A}_u}(i, j_u)}{\pi_u(j_u)} - \frac{p^{\mathcal{A}_u}\big(i, |\mathcal{I}_u|\big)}{\pi_u\big(|\mathcal{I}_u|\big)} \text{ for } j_u < |\mathcal{I}_u| \,. \tag{17}$$

Detailed derivation of expressions (15)–(17) are available in Appendix A.

5 Illustrative Examples

5.1 Simulation Study

In this section, we evaluate the performance of the proposed methodology via a simulation study. We generated 100 samples from the marginal association model represented by the bi-directed graph of Figure 1 and true log-linear interactions given in Table 1. This model encodes the marginal independencies $(AB) \perp\!\!\!\perp D$ and $A \perp\!\!\!\perp (CD)$ under the connect set Markov property.

We compare the methods introduced and discussed in this article in terms of acceptance rate, effective sample size (ESS) per second of CPU time and the Monte Carlo Error (MCE). In addition to the algorithms described in Section 4 (PBIS and PAA) we also consider random walks on marginal log-linear interactions λ and on logits of probability parameters π (RW-λ and RW-π respectively).

For all interactions and for each method under consideration, we calculated the ESS per second of CPU time using both coda and rstan packages in R.

Table 1: True Effect Values Used for the Simulation Study

Marginal	Active interactions	Zero interactions
AC	$\lambda_{\emptyset}^{AC} = -1.40, \lambda_{A}^{AC}(2) = -0.15, \lambda_{C}^{AC}(2) = 0.10,$	$\lambda_{AC}^{AC} = 0$
AD	$\lambda_{B}^{AD}(2) = 0.12,$	$\lambda_{BD}^{BD}(2,2) = 0$
BD	$\lambda_{D}^{BD}(2) = -0.09,$	$\lambda_{AD}^{AD}(2,2) = 0$
ACD	$\lambda_{CD}^{ACD}(2,2) = 0.20,$	$\lambda_{ACD}^{ACD}(2,2,2) = 0$
ABD	$\lambda_{AB}^{ABD}(2,2) = -0.15,$	$\lambda_{ABD}^{ABD}(2,2,2) = 0$
ABCD	$\lambda_{BC}^{ABCD}(2,2) = -0.30, \lambda_{ABC}^{ABCD}(2,2,2) = 0.15,$	
	$\lambda_{BCD}^{ABCD}(2,2,2) = -0.10, \lambda_{ABCD}^{ABCD}(2,2,2) = 0.07.$	

We calculated MCEs for the mean and standard deviation of all interactions via the batch mean method using 50 batches of equal size. The previous quantities have been adjusted for computational time by fixing the number of iterations of PAA and then considering as number of iterations for the remaining methods the one corresponding to the computational time of PAA.

In order to proceed with a more rigorous analysis, we first present the results for a single randomly selected sample (see Table 2 for the specific dataset under consideration) and then we discuss the main important findings for all samples.

Table 2: Simulated data

	D_1				D_2			
	C_1		C_2		C_1		C_2	
A	(B_1)	(B_2)	(B_1)	(B_2)	(B_1)	(B_2)	(B_1)	(B_2)
A_1	25	44	47	21	6	36	65	29
A_2	31	25	31	12	27	17	65	19

In terms of acceptance rate, PAA achieves a 50% rate which is satisfactory for an independence sampler and considerably higher than the PBIS algorithm ($\approx 15\%$). This could be interpreted by the fact that the later method uses a data augmentation based approach, which expands the parameter space by introducing the latent counts as proposed in Section 4.2. On the other hand, the RW algorithms (either for λ or for π) were tuned in order to achieve an acceptance rate close to 35%. The acceptance rates of the independence samplers (PBIS and PAA) and of the RW algorithms cannot be compared due to the different nature of the proposals.

Summary statistics of ESS per second of CPU time are presented in Table 3, the results for all interactions are depicted in Figure 3. These results indicate that PAA is the most efficient method followed by PBIS and RW-λ having similar values, while the RW-π appears as a rather inefficient way to approach the problem.

Table 3: Summary statistics of ESS per second of CPU time for the simulated data

	CODA					STAN				
Method	Min	Q_1	Median	Q_3	Max	Min	Q_1	Median	Q_3	Max
RW on π	1.6	2.7	3.2	5.3	7.0	1.2	2.6	3.1	5.3	6.5
RW on λ	25.3	32.2	33.7	34.8	37.4	23.9	28.6	31.8	34.6	35.8
PBIS	25.2	28.6	28.8	31.6	33.9	20.4	26.3	27.6	29.1	32.3
PAA	59.4	68.0	70.9	80.9	85.6	58.2	65.7	70.0	77.3	84.1

Min, Q_1, Median, Q_3, Max: Minimun, first quantile, Median, third quantile and maximun of acceptance rates over different interactions

Figure 3: ESS per second of CPU time for the simulated data

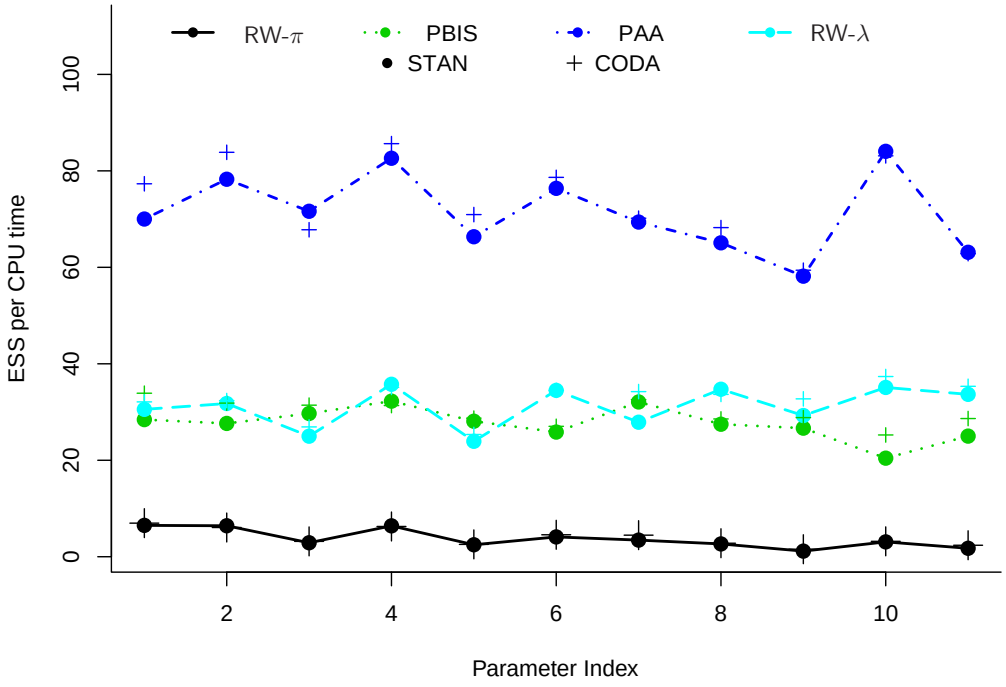
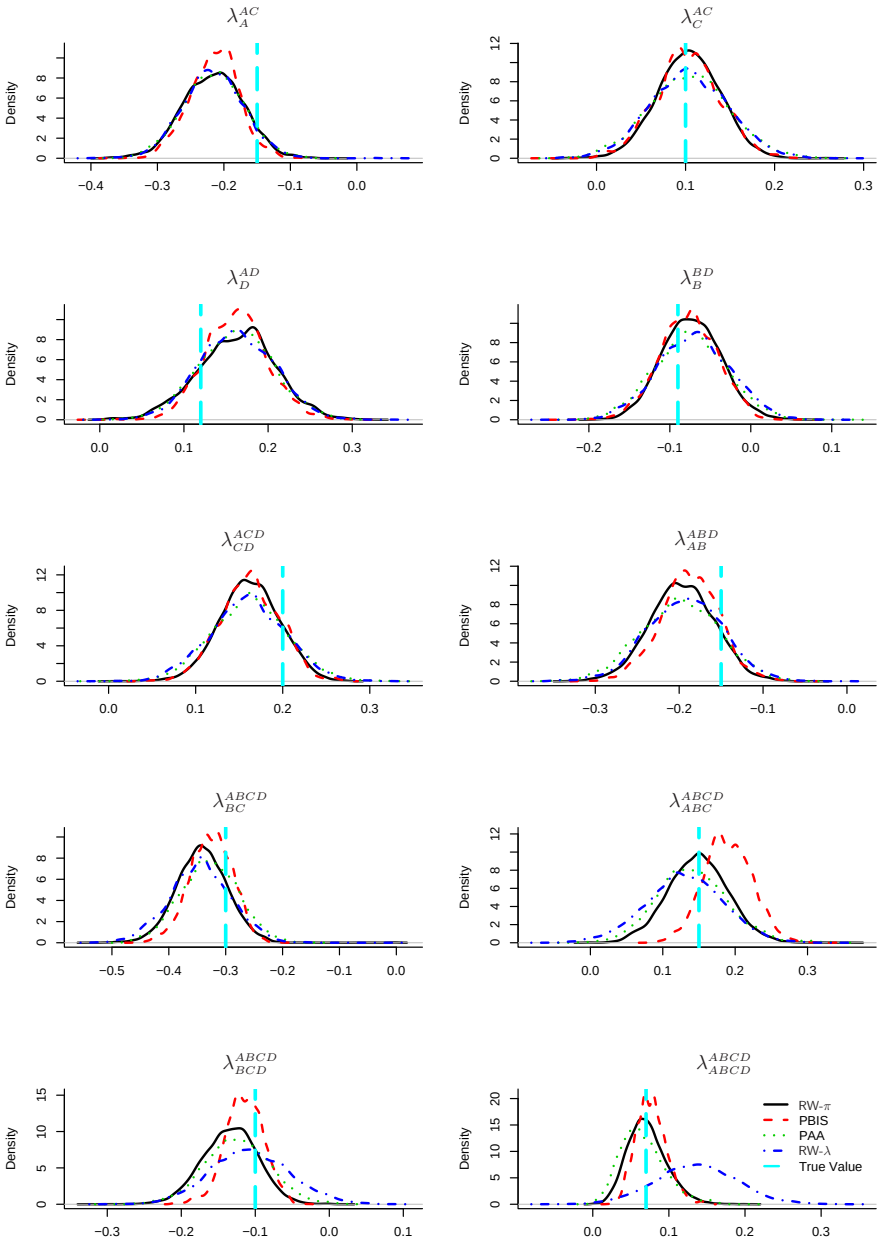


Figure 4 presents the estimated posterior distributions for all interactions. No major differences are observed in most interactions with the true values being in the centre (or near the centre) of the posterior distribution of interest. Some differences are observed in higher order interaction terms λ_{ABC}^{ABCD}, λ_{BCD}^{ABCD}

and λ^{ABCD}_{ABCD} estimated from the $ABCD$ marginal table. More specifically, for λ^{ABCD}_{ABC} only RW-π seems to estimates exactly the true value while PBIS slightly overestimates this effect and the rest of the methods underestimate it. For λ^{ABCD}_{BCD}, all methods provide similar posterior distributions with the PBIS having lower dispersion. Finally, for the four-way interaction, all methods correctly identify the true effect with the exception of RW-λ that overestimates the true value. Generally, the estimated posterior distribution obtained by our proposed method, PAA, seems that correctly identifies the true values for all interactions.

In Figure 5 for all methods and all marginal log-linear interactions we represent the time adjusted MCEs for the posterior means. We notice that PAA performs better than all competing methods, since the corresponding MCEs are lower for almost all interactions. In contrast, more dispersion is observed

Figure 4: Posterior densities for each parameter of the chain model estimated for the simulated data



for the posterior distribution of RW-π that is clearly performing worse than the other methods.

Regarding the posterior standard deviations, Figure 6 depicts time adjusted MCEs for all marginal log-linear interactions under all methods under consideration. PAA and PBIS demonstrate overall a better performance (in terms of Monte Carlo variability) in comparison to RW-λ and RW-π.

Regarding the simulation study, we present the ESS per second of CPU time and the posterior mean for each parameter over 100 generated datasets in Figures 7 and 8, respectively. From the distribution of the ESS per second, we confirm the results found in the single-sample analysis which indicate that PAA is clearly the most efficient between the four methods under consideration. PBIS and the RW on λ are equally efficient in terms of ESS/minute while the RW-π is the least efficient method. From Figure 8 we confirm that the estimated posterior means under all methods successfully identify (with minor

Figure 5: MCEs for posterior mean adjusted for time

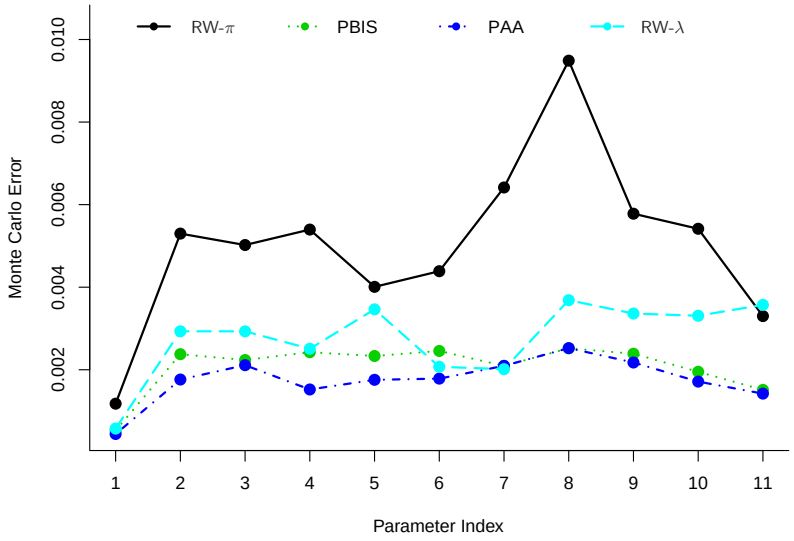
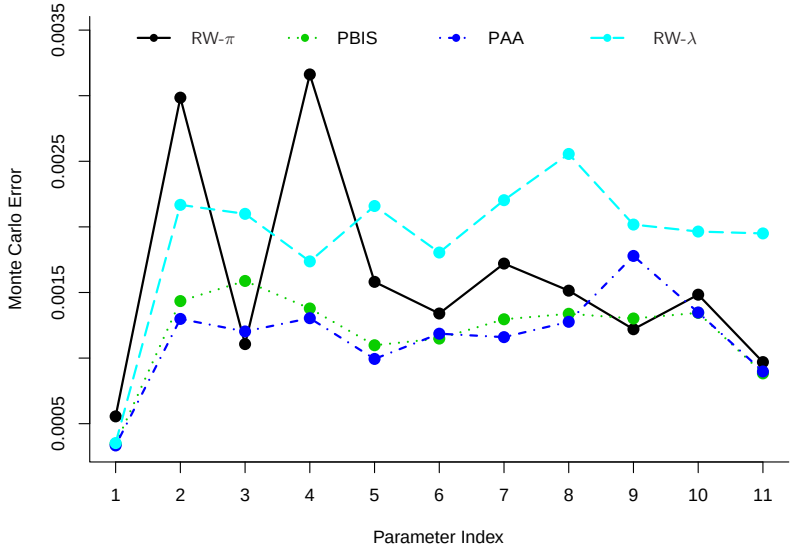
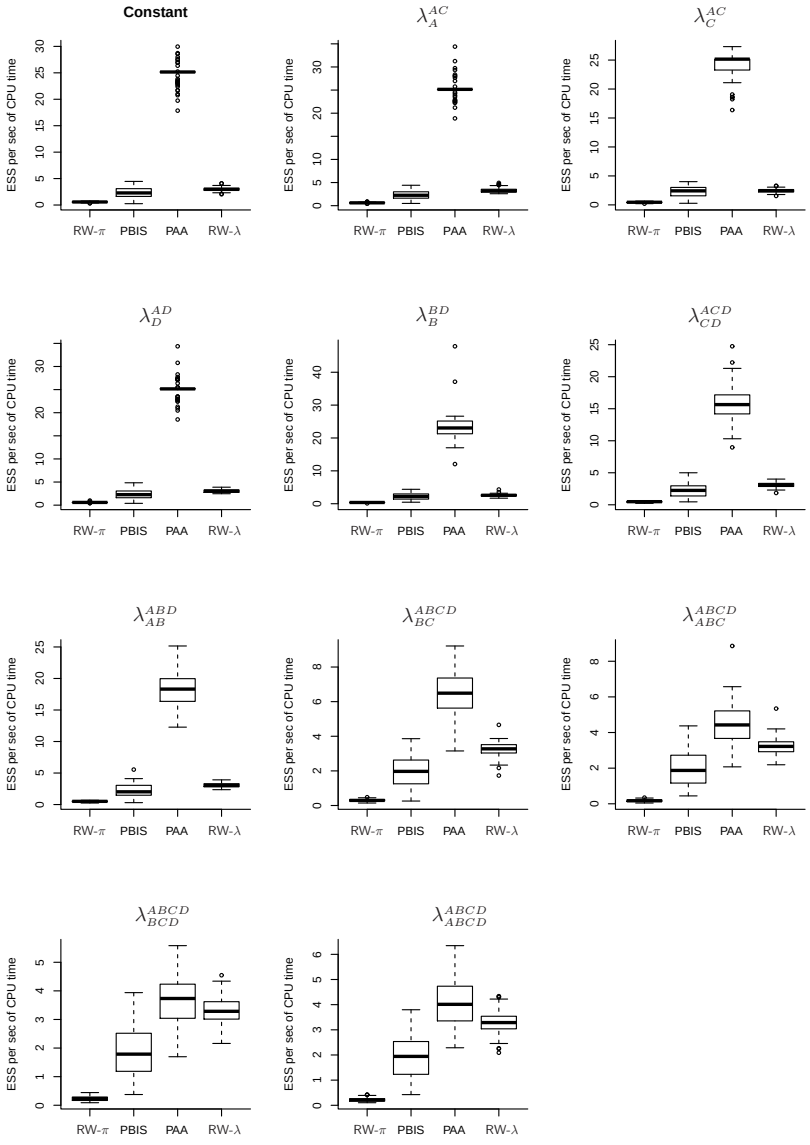


Figure 6: MCEs for posterior standard deviations adjusted for CPU time



deviances) the true parameter.

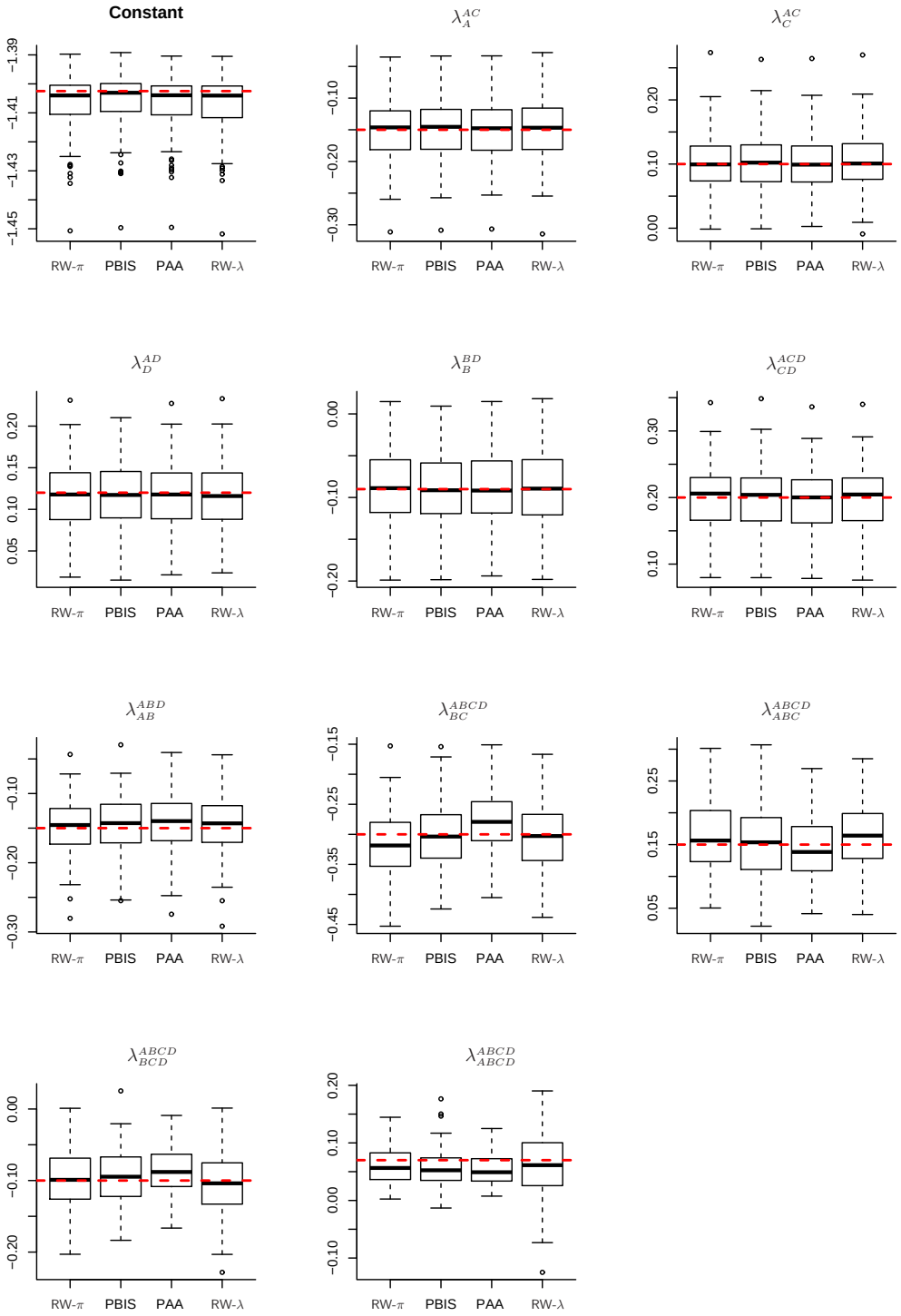
Figure 7: ESS per second of CPU time over 100 simulated datasets



Figures 9 and 10 present the 95% error bars of the average time adjusted MCEs for the posterior means and standard deviations. Each error bar represents the 2.5 and the 97.5 percentiles as well as the average of the quantities of interest (posterior means and standard deviations) for every interaction across all generated data sets. In most cases, PAA achieves values lower or at least of comparable size to the corresponding one of the other methods.

We conclude this section with a comparison of the dispersion of PAA and RW-λ. Figure 11 illustrates the distributions of the ratio of the posterior standard deviations of PAA versus RW-λ across all simulated datasets. We observe that for most interactions this ratio is distributed around one. For the last four interactions, where the latent is involved, we observe that PAA has systematically lower standard

Figure 8: Posterior means of marginal log-linear interactions over 100 simulated datasets



deviation.

Figure 9: MCEs for posterior Mean adjusted for CPU time for the simulation study

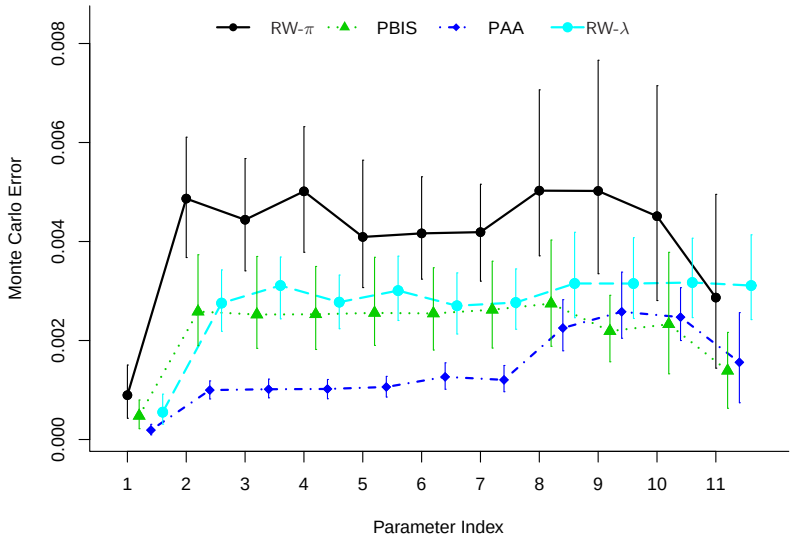


Figure 10: MCEs for posterior Standard Deviations adjusted for CPU time for the simulation study

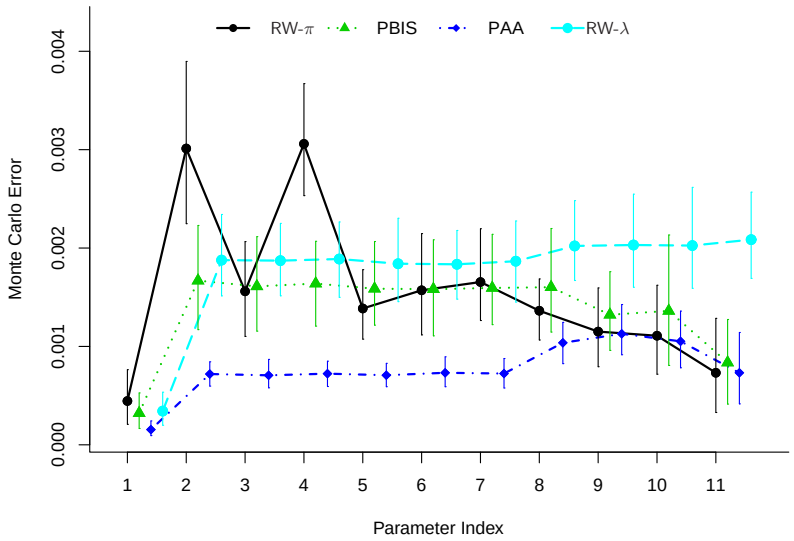
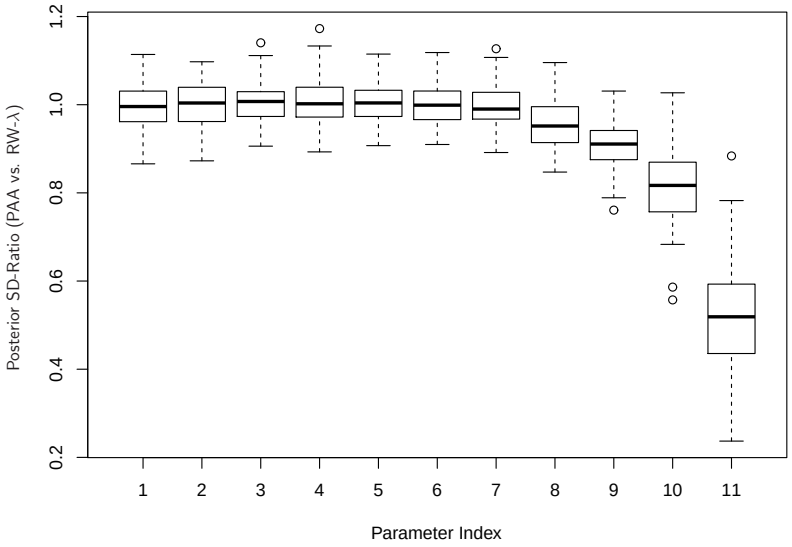


Figure 11: Boxplots of the Ratio of Posterior Standard Deviations of PAA vs. RW-λ for simulated 100 datasets



5.2 Torus Mandibularis in Eskimoid Groups Data

We illustrate the proposed methodology by using the dataset of Muller and Mayhall (1971) studying the incidence of the morphological trait torus mandibularis in different Eskimo groups. Torus mandibularis is a bony growth in the mandible along the surface nearest to the tongue. This morphological structure of the month is frequently used by anthropologist to study differences among populations and among groups within the same population. This data have been previously analysed by Bishop *et al.* (1975) via log linear models, and by Lupparelli (2006) via marginal log-linear graphical models.

For our analysis, we consider the data presented in Table 4, cross-classifying age (A), incidence of Torus Mandibularis (I), sex (S) and population (P). The dataset is a dichotomized version of the original data of Muller and Mayhall (1971). The examined Eskimo groups refers to different geographical regions, Igloolik and Hall Beach groups are from Foxe Basin area of Canada whereas Aleut are from Western Alaska. Furthermore, the data of the Aleuts group were collected by an investigator different from the one who collected the data for the first two groups, with a time difference between investigations of about twenty years. For the previous reasons we decided to reclassify the data in two groups: the first one including Igloolik and Hall Beach and the second one Aleut. Finally, variable age has been classified in two groups according to the median value.

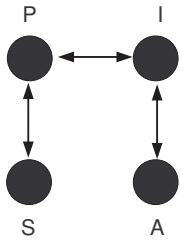
From the analysis of Lupparelli (2006) we know that the model represented in Figure 12 fits well the original data, hence, in the following, we concentrate on this specific four-chain graph.

Table 5 reports the posterior means and standard deviations for the marginal log-linear interactions

Table 4: Torus Mandibularis in Eskimo Populations

			Age Groups (A)	
Population (P)	*Sex* (S)	*Incidence* (I)	1-20	Over 20
Igloolik and Hall Beach	Male	Present	19	73
		Absent	103	38
	Female	Present	16	61
		Absent	87	36
Aleut	Male	Present	6	18
		Absent	19	14
	Female	Present	4	10
		Absent	17	20

Figure 12: Bi-directed graph for Torus data



obtained via 10000 iteration with a burn in of 1000 with the proposed PAA algorithm and the RW-λ. The maximum likelihood estimates (MLEs) and corresponding approximate standard errors are also reported for comparative purposes. MLEs are obtained implementing an algorithm for the optimization of the Lagrangian log-likelihood which includes a set of zero constraints satisfying the marginal independence model; see Lupparelli (2006) for details. Standard errors for the parameter estimates can be simply derived from the estimate of the Hessian matrix of the Lagrangian log-likelihood; see Aitchison and Silvey (1958).

From Table 5 we observe that the posterior estimates (for both MCMC methods) and the MLEs coincide for all interactions and main effects obtained by marginals where no latent variable is involved; also Figure D.1 in the Appendix. This is not the case for $\lambda_{IP}^{AIPS}(2,2)$, $\lambda_{IPS}^{AIPS}(2,2,2)$, $\lambda_{AIPS}^{AIPS}(2,2,2)$, where the latent variable is involved. More specifically, the posterior standard deviations are lower by 6.5%, 34% and 26%, respectively. This result is intuitively expected since PAA moves across the correct posterior distribution defined on the space of parameterisation with compatible marginal probabilities. On the other hand, both the RW-λ and the approximate MLEs standard errors are obtained without

Table 5: Posterior summaries and MLEs for marginal log-linear interactions for the Torus Mandibularis data

	PAA		RW-λ		ML	
	Mean	SD	Mean	SD	Estimate	SE
$\lambda_{\emptyset}^{AP}$	-1.391	0.004	-1.391	0.004		
$\lambda_{A}^{AP}(2)$	-0.001	0.042	-0.003	0.043	-0.002	0.043
$\lambda_{P}^{AP}(2)$	-0.072	0.043	-0.079	0.043	-0.072	0.043
$\lambda_{AP}^{AP}(2,2)$	0.000	0.000	0.000	0.000	0.000	
$\lambda_{S}^{AS}(2)$	-0.697	0.053	-0.695	0.055	-0.699	0.054
$\lambda_{AS}^{AS}(2,2)$	0.000	0.000	0.000	0.000	0.000	
$\lambda_{I}^{IS}(2)$	0.234	0.045	0.241	0.044	0.232	0.044
$\lambda_{IS}^{IS}(2,2)$	0.000	0.000	0.000	0.000	0.000	
$\lambda_{PS}^{APS}(2,2)$	0.004	0.053	-0.009	0.055	0.003	0.054
$\lambda_{APS}^{APS}(2,2,2)$	0.000	0.000	0.000	0.000	0.000	
$\lambda_{AI}^{AIS}(2,2)$	-0.509	0.051	-0.505	0.052	-0.507	0.051
$\lambda_{AIS}^{AIS}(2,2,2)$	0.000	0.000	0.000	0.000	0.000	
$\lambda_{IP}^{AIPS}(2,2)$	0.057	0.058	0.082	0.063	0.052	0.062
$\lambda_{AIP}^{AIPS}(2,2,2)$	0.132	0.068	0.049	0.065	0.151	0.062
$\lambda_{ISP}^{AIPS}(2,2,2)$	0.029	0.041	0.066	0.063	0.072	0.062
$\lambda_{AIPS}^{AIPS}(2,2,2)$	0.047	0.046	0.034	0.063	0.037	0.062

considering the restrictions imposed in order to obtain parameterisations leading to compatible marginal probabilities.

Finally, differences are also observed for interaction $\lambda_{AIP}^{AIPS}(2,2,2)$ where RW-λ provides posterior means far away from the corresponding MLEs with PAA being quite closely and standard deviance slightly higher than both the corresponding values of RW-λ and the MLEs standard errors.

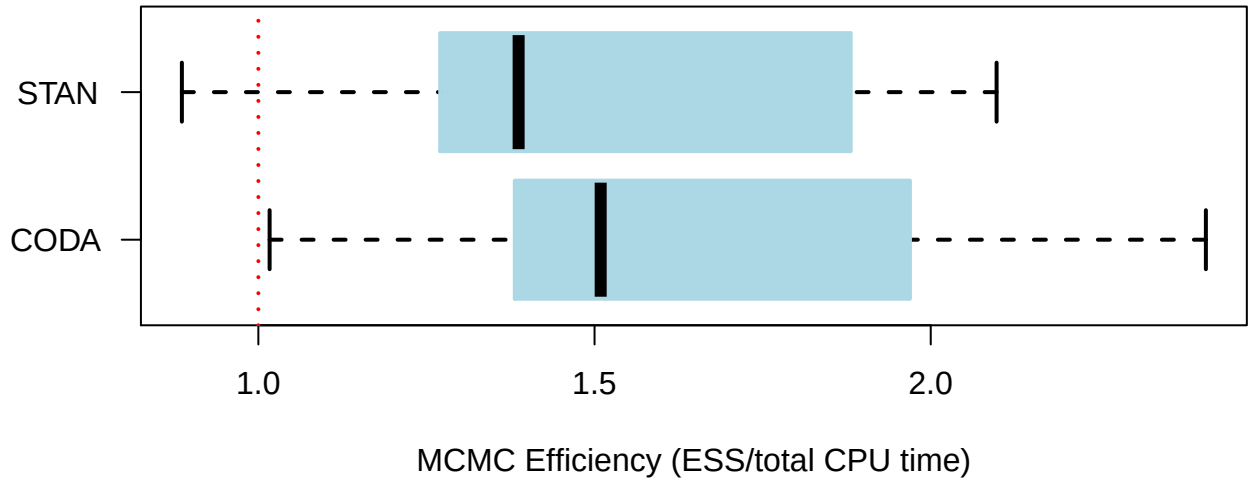
In terms of computational time, PAA was found to be faster with elapsed CPU time lower by 46% compared with the corresponding one for RW-λ (32.6 versus 60.8 seconds for 1000 iterations). Equivalently, the reported user CPU time was 48% lower for PAA than the one for RW-λ (29.9 versus 57.2 seconds for 1000 iterations). All runs were performed in a windows 10 PC Intel Core i7-5500U CPU 2.4GHz with 16GB memory; all timings were obtained with the system.time function in R.

For 11,000 iterations, the effective sample size (ESS) of the methods are of similar size ranging from

43% in favor of RW-λ (for S main effect) to 36% in favor of our method (for the four-way interation).

Taking into consideration also the computational time of the two algorithms, the MCMC efficiency (ESS/Elapsed CPU time) for the proposed algorithm is considerably higher for all interactions ranging from 2% increase to 141%. The average relative MCMC efficiency is higher for our method by 65% compared with the one of the random walk MCMC. The above statistics were calculated with the effectiveSize function of coda package in R. The overall picture is similar if we consider results using the monitor function of rstan package in R; see Figure 13 for a visual representation of the relative efficiency for all interactions.

Figure 13: Boxplots of the relative efficiency of Prior-Adjustment Algorithm (PAA) compared to the random walk MCMC obtained by CODA and STAN



Finally, the MCMC errors were lower for the majority of the interactions by using the naive estimator of CODA package with relative values ranging from 0.61 up to 1.13 while the corresponding relative values for the time series based estimator are ranging from 0.66 up to 1.30. The naive estimator of MCE is simply given by the usual standard error ignoring autocorrelation (i.e. the MCMC estimated posterior standard deviation over the square root of the number of iterations).

6 Discussion

A possible way to parameterise discrete graphical models of marginal independence is by using the log-linear marginal models of Bergsma and Rudas (2002). The marginal log-linear interactions are calculated from specific marginals of the original table, and independencies imply zero constraints on specific set of interactions in a similar manner as in conditional log-linear graphical models.

In this work we focus on the Bayesian estimation of the log-linear interactions for graphical models of marginal independence. In particular, the method we propose allows to assign prior directly on marginal log-linear interactions rather than on the probability interactions. This facilitates the incorporation of prior information since several models of interest can be specified by zero/linear constraints on log-linear terms. Bayesian analysis of such models is not widespread mainly due to the computational problems involved in the derivation of their posterior distribution. More specifically, MCMC methods need to be used since no conjugate analysis is available. Major difficulties arise from the fact that we need to sample from a posterior distribution defined on the space of parameterisations with compatible marginal probability distributions. In the proposed algorithm, we satisfy such restrictions by sampling from the probability space of the graphical model of marginal independence under consideration. Then we transform the probability parameter values to the corresponding marginal log-linear ones avoiding the iterative procedure needed for the evaluation of the likelihood. In order to achieve this, we exploit the augmented DAG representation of the model. This not only facilitates the prior elicitation but also the construction of the jacobian matrix involved in the acceptance probability of the induced Metropolis steps. Even if the derivation of the proposed algorithm is elaborate, it leads to an efficient and fully automatic setup. By this way we sample directly from the target posterior, and, on the same time, we avoid any time consuming and troublesome tuning of MCMC parameters.

For future research, the authors would like to exploit and study the connections between the prior and the posterior distributions for the two different parameterisations (probability versus marginal log-linear). Moreover, extension of the method to accommodate fully automatic selection, comparison and model averaging techniques is an intriguing topic for further investigation.

Supplementary Material

Supplementary Material: Appendix for the Paper "Probability Based Independence Sampler for Bayesian Quantitative Learning in Graphical Log-Linear Marginal Models" (available at `http://www.stat-athens.aueb.gr/~jbn/papers/files/2016_Ntzoufras_Tarantola_Lupparelli_Supplementary.pdf`). The supplementary material includes details for the Jacobian calculations (Appendix A) and details for the construction of $\mathbf{M}$ and $\mathbf{C}$ matrices (Appendices B and C respectively). Finally, some additional results for the illustrated examples of Section 5 are provided in Appendix D.

Acknowledgments

We would like to thank Giovanni Marchetti for providing us the R function inv.mlogit. This research was partially funded by the Research Centre of the Athens University of Economics and Business (Funding program for the research publications of the AUEB Faculty members) and by the Department of Economics and Management of University of Pavia.

References

Aitchison, J. and Silvey, S. D. (1958). Maximum likelihood estimation of parameters subject to restraints. *Annals of Mathematical Statistics* **29**, 813828.

Bartolucci, F., Scaccia, L. and Farcomeni, A. (2012). Bayesian inference through encompassing priors and importance sampling for a class of marginal models for categorical data. *Computational Statistics and Data Analysis*, **56**, 4067–4080.

Bergsma, W. P. and Rudas, T. (2002). Marginal log-linear models for categorical data. *Annals of Statistics*, **30**, 140-159.

Bergsma, W. O., Croon, M. A. and Hagenaars, J. A. (2009). *Marginal Models. For Dependent, Clustered, and Longitudinal Categorical Data*, Springer

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

Cox, D. R. and Wermuth, N. (1993). Linear dependencies represented by chain graphs (with discussion). *Statistical Science*, **8**, 204-218, 247-277.

Dellaportas, P. and Forster, J.J.(1999). Markov Chain Monte Carlo model determination for hierarchical and graphical log-linear models. *Biometrika*, **86**, 615-633.

Drton, M. and Richardson, T. S. (2008). Binary models for marginal independence. *Journal of the Royal Statistical Society*, Ser. B , **70**, 287-309.

Evans, R. J. (2016) Graphs for margins of Bayesian networks. *Scandinavian Journal of Statistics*, **43**, 625-648.

Evans, R. J. and Richardson, T. S. (2013) Marginal log-linear parameters for graphical Markov models. *Journal of the Royal Statistical Society*, Ser. B , **75**, 74368.

Knuiman, M. W. and Speed, T. P. (1988). Incorporating prior information into the analysis of contingency tables. *Biometrics*, **44**, 1061-1071.

Lupparelli, M. (2006). *Graphical models of marginal independence for categorical variables*. Ph. D. thesis, University of Florence.

Lupparelli, M., Marchetti, G. M., Bersgma, W. P. (2009). Parameterization and fitting of bi-directed graph models to categorical data. *Scandinavian journal of Statistics* **36**, 55976.

Muller, T.P. and Mayhall, J.T. (1971). Analysis of contngency table data on torus mandibularis using a loglinear model. *American Journal of Physical Anthropology*, **34**, 149-154.

Ntzoufras, I. and Tarantola, C. (2013). Conjugate and Conditional Conjugate Bayesian Analysis of Discrete Graphical Models of Marginal Independence. *Computational Statistics & Data Analysis*, **66**, 161-177.

Pearl, J. and Wermuth, N. (1994). When can association graphs admit a causal interpretation? In *Models and data, artificial intelligence and statistics iv*, Cheesman P. and Oldford,W., eds., Springer, New York, 205–214.

Plummer M., Best N., Cowles K. and Vines K. (2006). CODA: Convergence Diagnosis and Output Analysis for MCMC. *R News*, **6**, 7–11; `https://www.r-project.org/doc/Rnews/Rnews_2006-1.pdf`.

Qaqish, B. F. and Ivanova, A. (2006). Multivariate logistic models. *Biometrika* **93**, 1011-1017.

Richardson, T. S. (2003). Markov property for acyclic directed mixed graphs. *Scandinavian Journal of Statistics* **30**, 145-157.

Raftery, A.E. and Lewis, S.M. (1992). One long run with diagnostics: Implementation strategies for Markov chain Monte Carlo. Statistical Science, **7**, 493–497.

Rudas, T. and Bergsma, W. P. (2004). On applications of marginal models for categorical data. *Metron* LXII, 125.

Rudas, T., Bergsma, W. P. and Németh, R. (2010). Marginal log-linear parameterization of conditional independence models. *Biometrika*, **97**, 1006-1012.

Silva, R. and Ghahramani, Z. (2009a). The hidden life of latent variables: Bayesian learning with mixed graph models. *Journal of Machine Learning Research* 10, 1187–1238

Silva, R. and Ghahramani, Z. (2009b). Factorial mixture of Gaussians and the marginal independence model. *Proceedings of the 12th International Conference on Artificial Intelligence and Statistics (AISTATS)*, Clearwater Beach, Florida, USA.

Stan Development Team (2017). RStan: the R interface to Stan. R package version 2.16.2. `http://mc-stan.org`.

Supplementary Material: Appendix for the Paper "Probability Based Independence Sampler for Bayesian Quantitative Learning in Graphical Log-Linear Marginal Models"

Ioannis Ntzoufras, Athens University of Economics & Business, Greece
Claudia Tarantola, University of Pavia, Italy
and Monia Lupparelli, University of Bologna, Italy

A Jacobian calculations

Equation (13) can be rewritten as

$$\frac{\partial \lambda_k}{\partial \Pi_j} = \sum_{l=1}^{\mathrm{c}_C} Q_{klj} \quad \text{with} \quad Q_{klj} = C_{kl} \frac{\sum_{\imath=1}^{|\mathcal{I}|} M_{l\imath} \Delta_{\imath j}}{\sum_{\imath=1}^{|\mathcal{I}|} M_{l\imath} P_\imath},$$

where $\boldsymbol{M} = (M_{l\imath})$ is a $\mathrm{c}_C \times |\mathcal{I}|$ matrix, $\boldsymbol{\Delta} = (\Delta_{\imath j})$ is a matrix of dimension $|\mathcal{I}| \times d_\Pi$, and d_Π is the dimension of the parameter vector $\boldsymbol{\Pi}$.

A.1 Step-by-step computation of the Jacobian matrix

Once $\boldsymbol{\Delta}$ is obtained (see Appendix A.2), we can construct the Jacobian matrix using the following steps

1. construct $\boldsymbol{H} = \boldsymbol{M}\boldsymbol{\Delta}$ a matrix of dimension $\mathrm{c}_C \times dim(\Pi)$;
2. construct $\boldsymbol{\Gamma} = \boldsymbol{M}\boldsymbol{P}$ vector of dimension $\mathrm{c}_C \times 1$;
3. construct $\boldsymbol{\Gamma}' = \boldsymbol{\Gamma}\mathbf{1}_{d_\Pi}^T$, with $\mathbf{1}_d$ a $d_\Pi \times 1$ vector of ones, and $\boldsymbol{\Gamma}'$ a matrix of dimension $\mathrm{c}_C \times d_\Pi$ with all columns equal to $\boldsymbol{\Gamma}$;
4. set $\boldsymbol{H}' = \boldsymbol{H} \circ \boldsymbol{\Gamma}''$ where $\circ$ indicates the Hadamard product (element by element multiplication), and $\boldsymbol{\Gamma}''$ is a matrix with elements $\Gamma''_{\nu\kappa} = 1/\Gamma'_{\nu\kappa}$;
5. denote with $\boldsymbol{J} = \boldsymbol{C}\boldsymbol{H}'$ the Jacobian matrix dimension $d_\Pi \times d_\Pi$.

A.2 Computational details for the elements of $\boldsymbol{\Delta}$

We now describe how we can analytically obtain the elements of matrix $\boldsymbol{\Delta}$ defined in (14). Each element of this matrix is derivative of type $\dfrac{\partial p(i)}{\partial \pi_{u|pa(u)}\left(j_u|j_{pa(u)}\right)}$, $i \in \mathcal{I}$ and $j \in \mathcal{I}^{\mathcal{A}}$. We can separate the computations in two separate cases:

Case A: $u \in \mathcal{V}$, i.e. u is an observable variable.

Case B: $u \in \mathcal{L}$, i.e. u is a latent variable.

A.2.1 Case A: Computations for Expression (15)

For any variable $u \in \mathcal{V}$ and $j_u < |\mathcal{I}_u|$ we have that

$$\frac{\partial p(i)}{\partial \pi_{u|pa(u)}(j_u|j_{pa(u)})} = \frac{\partial \sum\limits_{i_{\mathcal{L}} \in \mathcal{I}_{\mathcal{L}}} p^{\mathcal{A}}(i, i_{\mathcal{L}})}{\partial \pi_{u|pa(u)}(j_u|j_{pa(u)})} = \sum_{i^*_{\mathcal{L}} \in \mathcal{I}_{\mathcal{L}}} \frac{\partial p^{\mathcal{A}}(i^*)}{\partial \pi_{u|pa(u)}(j_u|j_{pa(u)})}$$

since $i^* = (i, i_{\mathcal{L}}) \Leftrightarrow i^*_{\mathcal{V}} = i$ and $i^*_{\mathcal{L}} = i_{\mathcal{L}}$. From (7) we further obtain that

$$\begin{aligned}
&\frac{\partial p(i)}{\partial \pi_{u|pa(u)}(j_u|j_{pa(u)})} = \\
&= \sum_{i^*_{\mathcal{L}} \in \mathcal{I}_{\mathcal{L}}} \frac{\partial \left\{ \pi_{u|pa(u)}(i^*_u|i^*_{pa(u)}) \prod\limits_{v \in \mathcal{V} \cup \mathcal{L} \setminus u} \pi_{v|pa(v)}(i^*_v|i^*_{pa(v)}) \right\}}{\partial \pi_{u|pa(u)}(j_u|j_{pa(u)})} = \\
&= \sum_{i^*_{\mathcal{L}} \in \mathcal{I}_{\mathcal{L}}} \left\{ \left[\prod_{v \in \mathcal{V} \cup \mathcal{L} \setminus u} \pi_{v|pa(v)}(i^*_v|i^*_{pa(v)}) \right] \times \frac{\partial \pi_{u|pa(u)}(i^*_u|i^*_{pa(u)})}{\partial \pi_{j_u|pa(u)}(j_u|j_{pa(u)})} \right\} = \\
&= \sum_{i^*_{\mathcal{L}} \in \mathcal{I}_{\mathcal{L}}} \left\{ \left[\prod_{v \in \mathcal{V} \cup \mathcal{L} \setminus u} \pi_{v|pa(v)}(i^*_v|i^*_{pa(v)\setminus\mathcal{L}}, i^*_{\mathcal{L}_v}) \right] \times \frac{\partial \pi_{u|pa(u)}(i^*_u|i^*_{pa(u)\setminus\mathcal{L}}, i^*_{\mathcal{L}_u})}{\partial \pi_{u|pa(u)}(j_u|j_{pa(u)\setminus\mathcal{L}}, j_{\mathcal{L}_u})} \right\}, \qquad (18)
\end{aligned}$$

where $\mathcal{L}_u = \mathcal{L} \cap pa(u)$, is the set latent variables $\mathcal{L}$ that are parents of u. In the following we indicate with by $\overline{\mathcal{L}_u} = \mathcal{L} \setminus pa(u)$ the latent variables that are not parents of u.

Therefore, (18) can be rewritten

$$\begin{aligned}
&\frac{\partial p(i)}{\partial \pi_{u|pa(u)}(j_u|j_{pa(u)})} = \\
&= \sum_{i^*_{\overline{\mathcal{L}_u}} \in \mathcal{I}_{\overline{\mathcal{L}_u}}} \left\{ \sum_{i^*_{\mathcal{L}_u} \in \mathcal{I}_{\mathcal{L}_u}} \left[\prod_{v \in \mathcal{V} \cup (\mathcal{L}_u \cup \overline{\mathcal{L}_u}) \setminus u} \pi_{v|pa(v)}(i^*_v|i^*_{pa(v)\setminus\mathcal{L}}, i^*_{\mathcal{L}_v}) \times \frac{\partial \pi_{u|pa(u)}(i^*_u|i^*_{pa(u)\setminus\mathcal{L}}, i^*_{\mathcal{L}_u})}{\partial \pi_{u|pa(u)}(j_u|j_{pa(u)\setminus\mathcal{L}}, j_{\mathcal{L}_u})} \right] \right\} \\
&= \sum_{i^*_{\overline{\mathcal{L}_u}} \in \mathcal{I}_{\overline{\mathcal{L}_u}}} \left\{ \sum_{i^*_{\mathcal{L}_u} \in \mathcal{I}_{\mathcal{L}_u}} \left[\prod_{v \in \overline{\mathcal{L}_u}} \pi_{v|pa(v)}(i^*_v|i^*_{pa(v)\setminus\mathcal{L}}, i^*_{\mathcal{L}_v}) \right. \right. \\
&\qquad \left. \left. \times \prod_{v \in \mathcal{V} \cup \mathcal{L}_u \setminus u} \pi_{v|pa(v)}(i^*_v|i^*_{pa(v)\setminus\mathcal{L}}, i^*_{\mathcal{L}_v}) \times \frac{\partial \pi_{u|pa(u)}(i^*_u|i^*_{pa(u)\setminus\mathcal{L}}, i^*_{\mathcal{L}_u})}{\partial \pi_{u|pa(u)}(j_u|j_{pa(u)\setminus\mathcal{L}}, j_{\mathcal{L}_u})} \right] \right\} \\
&= \sum_{i^*_{\overline{\mathcal{L}_u}} \in \mathcal{I}_{\overline{\mathcal{L}_u}}} \left\{ \left[\prod_{v \in \overline{\mathcal{L}_u}} \pi_{v|pa(v)}(i^*_v|i^*_{pa(v)\setminus\mathcal{L}}, i^*_{\mathcal{L}_v}) \right] \right. \\
&\qquad \left. \times \sum_{i^*_{\mathcal{L}_u} \in \mathcal{I}_{\mathcal{L}_u}} \left(\prod_{v \in \mathcal{V} \cup \mathcal{L}_u \setminus u} \pi_{v|pa(v)}(i^*_v|i^*_{pa(v)\setminus\mathcal{L}}, i^*_{\mathcal{L}_v}) \times \frac{\partial \pi_{u|pa(u)}(i^*_u|i_{pa(u)\setminus\mathcal{L}}, i^*_{\mathcal{L}_u})}{\partial \pi_{u|pa(u)}(j_u|j_{pa(u)\setminus\mathcal{L}}, j_{\mathcal{L}_u})} \right) \right\}. \qquad (19)
\end{aligned}$$

We now concentrate on the second line of (19), that is

$$\sum_{i^*_{\mathcal{L}_u}\in\mathcal{I}_{\mathcal{L}_u}}\left(\prod_{v\in\mathcal{V}\cup\mathcal{L}_u\setminus u}\pi_{v|pa(v)}\big(i^*_v|i^*_{pa(v)\setminus\mathcal{L}},i^*_{\mathcal{L}_v}\big)\times\frac{\partial\pi_{u|pa(u)}\big(i^*_u|i^*_{\mathcal{L}_u},i^*_{pa(u)\setminus\mathcal{L}}\big)}{\partial\pi_{u|pa(u)}\big(j_u|j_{\mathcal{L}_u},j_{pa(u)\setminus\mathcal{L}}\big)}\right)=$$

$$=\begin{cases}\displaystyle\prod_{v\in\mathcal{V}\cup\mathcal{L}_u\setminus u}\pi_{v|pa(v)}\big(i^*_v|i^*_{pa(v)}\big) & \text{if } i^*_u=j_u<|\mathcal{I}_u| \text{ and } \ i^*_{pa(u)}=j_{pa(u)}\\ \displaystyle-\prod_{v\in\mathcal{V}\cup\mathcal{L}_u\setminus u}\pi_{v|pa(v)}\big(i^*_v|i^*_{pa(v)}\big) & \text{if } j_u\neq i^*_u=|\mathcal{I}_u| \text{ and } \ i^*_{pa(u)}=j_{pa(u)}\\ 0 & \text{if } j_u\neq i^*_u<|\mathcal{I}_u| \text{ or } \ i^*_{pa(u)}\neq j_{pa(u)}\end{cases}.$$

In the case where $i^*_u=j_u<|\mathcal{I}_u|$ and $i_{pa(u)\setminus\mathcal{L}}=j_{pa(u)\setminus\mathcal{L}}$ then

$$\begin{aligned}&\frac{\partial p(i)}{\partial\pi_{u|pa(u)}\big(j_u|j_{pa(u)}\big)}=\\ &=\sum_{i^*_{\mathcal{L}_u}\in\mathcal{I}_{\mathcal{L}_u}}\left\{\left[\prod_{v\in\mathcal{L}_u}\pi_{v|pa(v)}\big(i^*_v|i^*_{pa(v)}\big)\right]\prod_{v\in\mathcal{V}\cup\mathcal{L}_u\setminus u}\pi_{v|pa(v)}\big(i^*_v|i^*_{pa(v)}\big)I_{\{i^*_{pa(u)}=j_{pa(u)}\}}\right\}\\ &=\sum_{i^*_{\mathcal{L}_u}\in\mathcal{I}_{\mathcal{L}_u}}\left\{\prod_{v\in\mathcal{V}\cup\mathcal{L}\setminus u}\pi_{v|pa(v)}\big(i^*_v|i^*_{pa(v)}\big)I_{\{i^*_{pa(u)}=j_{pa(u)}\}}\right\}\\ &=\sum_{i^*_{\mathcal{L}_u}\in\mathcal{I}_{\mathcal{L}_u}}\left\{\frac{\pi_{u|pa(u)}\big(i^*_u|i^*_{pa(u)}\big)}{\pi_{u|pa(u)}\big(i^*_u|i^*_{pa(u)}\big)}\prod_{v\in\mathcal{V}\cup\mathcal{L}\setminus u}\pi_{v|pa(v)}\big(i^*_v|i^*_{pa(v)}\big)I_{\{i^*_{pa(u)}=j_{pa(u)}\}}\right\}\\ &=\sum_{i^*_{\mathcal{L}_u}\in\mathcal{I}_{\mathcal{L}_u}}\left\{\frac{p^{\mathcal{A}}(i,j_{\mathcal{L}_u},j_{\overline{\mathcal{L}_u}})}{\pi_{u|pa(u)}(i^*_u|j_{pa(u)})}\right\}=\frac{\displaystyle\sum_{i^*_{\mathcal{L}_u}\in\mathcal{I}_{\mathcal{L}_u}}p^{\mathcal{A}}(i,j_{\mathcal{L}_u},j_{\overline{\mathcal{L}_u}})}{\pi_{u|pa(u)}(i^*_u|j_{pa(u)})}.\end{aligned}$$

Finally we obtain that

$$\frac{\partial p(i)}{\partial\pi_{u|pa(u)}\big(j_u|j_{pa(u)}\big)}=\frac{p^{\mathcal{A}_u}(i,j_{\mathcal{L}_u})}{\pi_{u|pa(u)}(i^*_u|j_{pa(u)})}. \tag{20}$$

In a similar way, we find that

$$\frac{\partial p(i)}{\partial\pi_{u|pa(u)}(j_u|j_{pa(u)})}=-\frac{p^{\mathcal{A}_u}(i,j_{\mathcal{L}_u})}{\pi_{u|pa(u)}(i^*_u|j_{pa(u)})} \tag{21}$$

for $j_u\neq i^*_u=|\mathcal{I}_u|$ and $i_{pa(u)\setminus\mathcal{L}}=j_{pa(u)\setminus\mathcal{L}}$.

Finally, if $j_u\neq i^*_u<|\mathcal{I}_u|$ or $\ i_{pa(u)\setminus\mathcal{L}}\neq j_{pa(u)\setminus\mathcal{L}}$, (19) will become equal to

$$\frac{\partial p(i)}{\partial\pi_{u|pa(u)}(i^*_u|j_{pa(u)})}=0. \tag{22}$$

Since u is an observed variable $i_u\equiv i^*_u$, from equations (20)–(22), we obtain the final expressions

$$\frac{\partial p(i)}{\partial\pi_{u|pa(u)}\big(j_u|j_{pa(u)}\big)}=\delta(i,j)\frac{p^{\mathcal{A}_u}\big(i,j_{\mathcal{L}_u}\big)}{\pi_{u|pa(u)}\big(i_u|j_{pa(u)}\big)}$$

with

$$\delta(i,j) = \begin{cases} 1 & \text{if } i_u = j_u < |\mathcal{I}_u| \text{ and } i_{pa(u)\setminus\mathcal{L}} = j_{pa(u)\setminus\mathcal{L}} \\ -1 & \text{if } j_u \neq i_u = |\mathcal{I}_u| \text{ and } i_{pa(u)\setminus\mathcal{L}} = j_{pa(u)\setminus\mathcal{L}} \\ 0 & \text{if } j_u \neq i_u < |\mathcal{I}_u| \text{ or } \quad i_{pa(u)\setminus L} \neq j_{pa(u)\setminus\mathcal{L}} \end{cases},$$

where

$$p^{\mathcal{A}_u}\big(i, j_{\mathcal{L}_u}\big) = \begin{cases} P\Big(X_{\mathcal{V}} = i, X_{\mathcal{L}\cap pa(u)} = j_{\mathcal{L}\cap pa(u)}\Big) & \mathcal{L}_u \neq \emptyset \\ p(i) & \mathcal{L}_u = \emptyset \end{cases}.$$

A.2.2 Case B: Computations for Expression (17)

For every latent variable $u \in \mathcal{L}$ the parent set is empty. Hence $\pi_{u|pa(u)}\big(i_u^*|i_{pa(u)}^*\big) = \pi_u\big(i_u^*\big)$ for all $i^* \in \mathcal{A}$. Furthermore it holds that $p(i) = \sum_{i_{\mathcal{L}} \in \mathcal{I}_{\mathcal{L}}} p^{\mathcal{A}}(i, i_{\mathcal{L}}) = \sum_{i^*_{\mathcal{L}} \in \mathcal{I}_{\mathcal{L}}} p^{\mathcal{A}}(i^*)$.

Therefore for any parameter $\pi_u(j_u)$ with $j_u \neq |\mathcal{I}_u|$ and $u \in \mathcal{L}$, the derivative is given by

$$\begin{aligned} \frac{\partial p(i)}{\partial \pi_u(j_u)} &= \frac{\partial \sum_{i_{\mathcal{L}} \in \mathcal{I}_{\mathcal{L}}} p^{\mathcal{A}}(i, i_{\mathcal{L}})}{\partial \pi_u(j_u)} = \frac{\partial \sum_{i^*_{\mathcal{L}} \in \mathcal{I}_{\mathcal{L}}} p^{\mathcal{A}}(i^*)}{\partial \pi_u(j_u)} = \sum_{i^*_{\mathcal{L}} \in \mathcal{I}_{\mathcal{L}}} \frac{\partial p^{\mathcal{A}}(i^*)}{\partial \pi_u(j_u)} \\ &= \sum_{i^*_{\mathcal{L}\setminus u} \in \mathcal{I}_{\mathcal{L}\setminus u}} \left\{ \sum_{i_u^* \in \mathcal{I}_u} \frac{\partial \left\{ \pi_u(i_u^*) \prod_{v \in \mathcal{V}\cup\mathcal{L}\setminus\{u\}} \pi_{v|pa(v)}\left(i_v^*|i^*_{pa(v)}\right) \right\}}{\partial \pi_u(j_u)} \right\} \\ &= \sum_{i^*_{\mathcal{L}\setminus u} \in \mathcal{I}_{\mathcal{L}\setminus u}} \left\{ \sum_{i_u^* \in \mathcal{I}_u} \left[\frac{\partial \pi_u(i_u^*)}{\partial \pi_u(j_u)} \times \prod_{v \in \mathcal{V}\cup\mathcal{L}\setminus\{u\}} \pi_{v|pa(v)}\left(i_v^*|i^*_{pa(v)}\right) \right] \right\}. \end{aligned}$$

The derivatives involved in the above equation are given by

$$\frac{\partial \pi_u(i_u^*)}{\partial \pi_u(j_u)} = \begin{cases} 1 & \text{if } j_u = i_u^* < |\mathcal{I}_u| \\ -1 & \text{if } j_u \neq i_u^* = |\mathcal{I}_u| \\ 0 & \text{if } j_u \neq i_u^* < |\mathcal{I}_u| \end{cases}.$$

Hence, we obtain

$$\begin{aligned}
\frac{\partial p(i)}{\partial \pi_u(j_u)} &= \sum_{i^*_{\mathcal{L}\setminus u}\in\mathcal{I}_{\mathcal{L}\setminus u}} \Bigg\{ \prod_{v\in\mathcal{V}\cup\mathcal{L}\setminus\{u\}} \pi_{v|pa(v)}(i^*_v|i^*_{pa(v)}) I_{\{i^*_u=j_u\}} \\
&\qquad - \prod_{v\in\mathcal{V}\cup\mathcal{L}\setminus\{u\}} \pi_{v|pa(v)}(i^*_v|i^*_{pa(v)}) I_{\{i^*_u=|\mathcal{I}_u|\}} \Bigg\} \\
&= \sum_{i^*_{\mathcal{L}\setminus u}\in\mathcal{I}_{\mathcal{L}\setminus u}} \Bigg\{ \frac{\pi_u(i^*_u)}{\pi_u(i^*_u)} \prod_{v\in\mathcal{V}\cup\mathcal{L}\setminus\{u\}} \pi_{v|pa(v)}(i^*_v|i^*_{pa(v)}) I_{\{i^*_u=j_u\}} \\
&\qquad - \frac{\pi_u(i^*_u)}{\pi_u(i^*_u)} \prod_{v\in\mathcal{V}\cup\mathcal{L}\setminus\{u\}} \pi_{v|pa(v)}(i^*_v|i^*_{pa(v)}) I_{\{i^*_u=|\mathcal{I}_u|\}} \Bigg\} \\
&= \sum_{i^*_{\mathcal{L}\setminus u}\in\mathcal{I}_{\mathcal{L}\setminus u}} \left\{ \frac{p^{\mathcal{A}}(i^*_{\mathcal{V}}, j_u, i^*_{\mathcal{L}\setminus u})}{\pi_u(j_u)} - \frac{p^{\mathcal{A}}(i^*_{\mathcal{V}}, |\mathcal{I}_u|, i^*_{\mathcal{L}\setminus u})}{\pi_u(|\mathcal{I}_u|)} \right\} \\
&= \frac{p^{\mathcal{A}_u}(i^*_{\mathcal{V}}, j_u)}{\pi_u(j_u)} - \frac{p^{\mathcal{A}_u}(i^*_{\mathcal{V}}, |\mathcal{I}_u|)}{\pi_u(|\mathcal{I}_u|)} \\
&= \frac{p^{\mathcal{A}_u}(i, j_u)}{\pi_u(j_u)} - \frac{p^{\mathcal{A}_u}(i, |\mathcal{I}_u|)}{\pi_u(|\mathcal{I}_u|)},
\end{aligned}$$

where $i = i^*_{\mathcal{V}}$ and this concludes the computation of (17).

B Construction of Matrix M

Let $\mathcal{M} = \left\{M_1, M_2, \ldots, M_{|\mathcal{M}|}\right\}$ be the set of marginals under consideration by a given marginal log-linear parameterisation. Let $\mathbf{B}$ be a binary matrix of dimension $|\mathcal{M}| \times |\mathcal{V}|$ with elements B_{mv} indicating whether a variable v belongs to a specific marginal M_m. The rows of $\mathbf{B}$ correspond to the marginals in $\mathcal{M}$ whereas the columns to the variables. The variables follow a reverse ordering, that is column 1 corresponds to variable $X_{|\mathcal{V}|}$, column 2 to variable $X_{|\mathcal{V}|-1}$ and so on. Matrix $\mathbf{B}$ has elements

$$B_{mv} = \begin{cases} 1 & \text{if } v \in M_m \\ 0 & \text{otherwise.} \end{cases}$$

for every $v \in \mathcal{V}$.

The marginalisation matrix $\mathbf{M}$ can be obtained applying the following rules.

1. For each marginal M_m and each variable v we construct the following matrix

$$\mathbf{A}_{m,v} = \begin{cases} \mathbf{I}_{|\mathcal{I}_v|} & \text{if} \quad B_{mv} = 1 \\ \mathbf{1}^T_{|\mathcal{I}_v|} & \text{if} \quad B_{mv} = 0 \end{cases},$$

where $\mathbf{I}_{|\mathcal{I}_v|}$ is the identity matrix of dimension $|\mathcal{I}_v| \times |\mathcal{I}_v|$ and $\mathbf{1}_{|\mathcal{I}_v|}$ is a vector of dimension $|\mathcal{I}_v| \times 1$ with all elements equal to one.

The probability vector of the marginal table corresponding to M_m is given by $\mathbf{M}_m\pi$; where $\mathbf{M}_m$ is calculated as a Kronecker product of matrices $\mathbf{A}_{mv}$

$$\begin{aligned}\mathbf{M}_m &= \bigotimes_{q=0}^{|\mathcal{V}|-1} \mathbf{A}_{m,|\mathcal{V}|-q} \\ &= \mathbf{A}_{m,|\mathcal{V}|} \bigotimes \mathbf{A}_{m,|\mathcal{V}|-1} \bigotimes \cdots \bigotimes \mathbf{A}_{m,2} \bigotimes \mathbf{A}_{m,1},\end{aligned}$$

where $\bigotimes$ denotes the Kronecker product which is implemented in reverse lexicographical order (starting from the last on in $\mathcal{V}$).

2. Matrix $\mathbf{M}$ is constructed by stacking all the $\mathbf{M}_m$ matrices

$$\mathbf{M} = \begin{pmatrix} \mathbf{M}_1 \\ \vdots \\ \mathbf{M}_m \\ \vdots \\ \mathbf{M}_{|\mathcal{M}|} \end{pmatrix} .$$

C Construction of Matrix C

Firstly we need to construct the design matrix $\mathbf{X}_{\mathcal{V}}$ for the saturated model for the cross-classification of discrete variables $X_{\mathcal{V}}$ with sum to zero constraints. Firstly, for each variable v we construct the following matrix

$$\mathbf{J}_v = \begin{pmatrix} 1 & -\mathbf{1}^T_{(|\mathcal{I}_v|-1)} \\ \mathbf{1}_{(|\mathcal{I}_v|-1)} & \mathbf{I}_{(|\mathcal{I}_v|-1)} \end{pmatrix} ,$$

where $\mathbf{1}_\kappa$ is a vector of ones of length κ while $\mathbf{I}_\kappa$ is the identity matrix of dimension $\kappa \times \kappa$.

The design matrix of the satured model will be of dimension $\left(\prod_{v\in\mathcal{V}} |\mathcal{I}_v|\right) \times \left(\prod_{v\in\mathcal{V}} |\mathcal{I}_v|\right)$ and can be obtained as

$$\mathbf{X}_{\mathcal{V}} = \bigotimes_{q=0}^{|\mathcal{V}|-1} \mathbf{J}_{|\mathcal{V}|-q} .$$

The contrast matrix $\mathbf{C}$ can be constructed by using the following rules.

1. For each margin M_m, we construct the design matrix $\mathbf{X}_{M_m}$ corresponding to the saturated model (using sum-to-zero constraints) of the cross-classification of variables X_{M_m}. Then we consider its inverse $\mathbf{X}^{-1}_{M_m}$ in order to obtain the corresponding contrast matrix. Now, let $\mathbf{C}_m$ be a sub-matrix of the contrast matrix $\mathbf{X}^{-1}_{M_m}$ obtained by deleting rows corresponding to interactions that are not obtained from margin M_m.

2. The contrast matrix $\mathbf{C}$ is obtained as

$$\mathbf{C} = \bigoplus_{m:\ M_m \in \mathcal{M}} \mathbf{C}_m = \mathrm{diag}(M_1, \ldots, M_{|\mathcal{M}|}) \ ,$$

where $\bigoplus$ denotes the matrix direct sum.

D Additional Results

Figure D.1 depicts the posterior ergodic plots for the estimates of the elements of $\vec{\lambda}$ obtained using PAA and the corresponding approximate MLE based statistics for the real dataset of Section 5.2.

Figure D.1: Ergodic plots for marginal log-linear interactions using Prior-Adjustment Algorithm (PAA) compared with approximate maximum likelihood estimates for the Torus Mandibularis data