

# Parametric families on large binary spaces

Christian Schäfer \*

February 24, 2011

## Abstract

In the context of adaptive Monte Carlo algorithms, we cannot directly generate independent samples from the distribution of interest but use a proxy which we need to be close to the target.

Generally, such a proxy distribution is a parametric family on the sampling spaces of the target distribution. For continuous sampling problems in high dimensions, we often use the multivariate normal distribution as a proxy for we can easily parametrise it by its moments and quickly sample from it.

Our objective is to construct similarly flexible parametric families on binary sampling spaces too large for exhaustive enumeration. The binary sampling problem seems more difficult than its continuous counterpart since the choice of a suitable proxy distribution is not obvious.

## 1 Parametric families and Monte Carlo

### 1.1 Adaptive Monte Carlo

A Monte Carlo algorithm is said to be adaptive if it is able to adjust, sequentially and automatically, its sampling distribution to the problem at hand. Precisely, the algorithms are able to incorporate information obtained from past simulations to improve the sampling distribution  $q$  in terms of nearness to the target distribution  $\pi$ .

Some important classes of adaptive Monte Carlo algorithms are Adaptive Importance Sampling (e.g. Cappé et al., 2008), Adaptive Markov chain Monte Carlo (e.g. Andrieu and Thoms, 2008), Sequential Monte Carlo (Del Moral et al., 2006) and the Cross-Entropy method (Rubinstein and Kroese, 2004).

For the sampling distribution, we usually select a suitable parametric family  $q = q_\theta$  and adjust its parameter  $\theta$  during the course of the adaptive algorithm. In continuous sampling spaces, good results are often achieved using normal distributions  $\mathcal{N}(\mu, \Sigma)$ , for they reproduce the marginals and covariance structure of the target.

### 1.2 Data from the target distribution

In the sequel, let  $d > 0$  denote the dimension of the binary space  $\mathbb{B}^d = \{0, 1\}^d$ . Adaptive Monte Carlo algorithms are generally able to produce a, not necessarily independent and possibly weighted, sample

$$\mathbf{w} = (w_1, \dots, w_n) \in [0, 1]^n, \quad \mathbf{X} = (\mathbf{x}_1, \dots, \mathbf{x}_n)^\top \in \mathbb{B}^{n \times d}$$

from the target distribution  $\pi$  we want to emulate using a binary model. We define the index set  $D = \{1, \dots, d\}$  and denote by

$$\bar{x}_i \stackrel{\text{def}}{=} \sum_{k=1}^n w_k x_{k,i}, \quad \bar{x}_{i,j} \stackrel{\text{def}}{=} \sum_{k=1}^n w_k x_{k,i} x_{k,j}, \quad i, j \in D \quad (1)$$

the weighted first and second sample moments. We further define by

$$r_{i,j} \stackrel{\text{def}}{=} \frac{\bar{x}_{i,j} - \bar{x}_i \bar{x}_j}{\sqrt{\bar{x}_i(1 - \bar{x}_i)\bar{x}_j(1 - \bar{x}_j)}}, \quad i, j \in D. \quad (2)$$

the weighted sample correlation.

### 1.3 Suitable parametric families

We first frame some properties making a parametric family suitable as sampling distribution in adaptive Monte Carlo algorithms.

- (a) For reasons of parsimony, we want to construct a family of distributions with at most  $\dim(\theta) \leq d(d+1)/2$  parameters.
- (b) Given a sample  $\mathbf{X} = (\mathbf{x}_1, \dots, \mathbf{x}_n)^\top$  from the target distribution  $\pi$ , we need to estimate  $\theta^*$  such that the binary model  $q_{\theta^*}$  is close to  $\pi$ .
- (c) We need to generate samples  $\mathbf{Y} = (\mathbf{y}_1, \dots, \mathbf{y}_m)^\top$  from the model  $q_\theta$ . We need the rows of  $\mathbf{Y}$  to be independent.
- (d) For some algorithms, we need to evaluate the probability  $q_\theta(\mathbf{y})$ . For instance, we need  $q_\theta(\mathbf{y})$  to compute importance weights or acceptance ratios in the context of Importance Sampling or Markov chain Monte Carlo, respectively.
- (e) Analogously to the multivariate normal, we need our calibrated binary model  $q_{\theta^*}$  to reproduce the marginals and covariance structure of  $\pi$ .

\*CREST and Université Paris Dauphine · christian.schafer@ensae.fr

## 2 Distributions on binary spaces

Before we embark on the discussion of binary models, we make some observations which hold true for every binary distribution. The notation and results introduced in this section will be used throughout the rest of this work. Here, we denote by  $\pi$  some generic distribution on  $\mathbb{B}^d$

*Moments* We use the short notation,

$$u_I(\gamma) \stackrel{\text{def}}{=} \prod_{i \in I} \gamma_i, \quad I \subseteq D,$$

for the product of all components index by  $I$  with  $\prod_{i \in \emptyset} = 1$ . Since  $u_I(\gamma) = 1$  iff  $\gamma_i = 1$  for all  $i \in I$ ,  $u_I$  is the indicator function for the unit vector  $\mathbf{1}_{|I|}$ . We can characterize every distribution on  $\mathbb{B}^d$  by  $2^d - 1$  full probabilities

$$p_I \stackrel{\text{def}}{=} \mathbb{P}_\pi(\gamma_I = 1, \gamma_{D \setminus I} = 0), \quad I \subseteq D$$

or by  $2^d - 1$  cross-moments, that is marginal probabilities,

$$m_I \stackrel{\text{def}}{=} \mathbb{E}_\pi(u_I(\gamma)) = \mathbb{P}_\pi(\gamma_I = \mathbf{1}), \quad I \subseteq D.$$

In the following, we assume that  $m_i \in (0, 1)$  for all  $i \in D$ , since for  $m_i \in \{0, 1\}$ , the component  $\gamma_i = m_i$  is constant and therefore not part of the sampling problem.

For the product of components normalized to have zero mean and unit variance, we write

$$v_I(\gamma) \stackrel{\text{def}}{=} \prod_{k \in I} (\gamma_k - m_k) / \sqrt{m_k(1 - m_k)}, \quad I \subseteq D.$$

Note that  $\mathbb{E}_\pi(v_{i,j})$  is the correlation between  $\gamma_i$  and  $\gamma_j$ . Therefore, we call

$$c_I \stackrel{\text{def}}{=} \mathbb{E}_\pi(v_I(\gamma))$$

the correlation of order  $|I|$ .

*Marginals* We use the notation

$$\pi_I(\gamma_I) = \sum_{\xi \in \mathbb{B}^{d-|I|}} \pi(\gamma_I, \xi), \quad I \subseteq D.$$

for the marginal distributions. Note the connection to the cross-moments

$$\begin{aligned} \pi_I(\mathbf{1}_{|I|}) &= \sum_{\xi \in \mathbb{B}^{d-|I|}} \pi(\mathbf{1}_{|I|}, \xi) = \sum_{\gamma \in \mathbb{B}^d} u_I(\gamma) \pi(\gamma) \\ &= m_I. \end{aligned} \quad (3)$$

*Representations* For any function  $f: (0, 1) \rightarrow \mathbb{R}$ , we can write

$$f(\pi(\gamma)) = \sum_{I \subseteq D} \prod_{i \in I} \gamma_i \prod_{i \in D \setminus I} (1 - \gamma_i) f(p_I).$$

If  $f$  has an inverse  $f^{-1}$ , there are thus coefficients  $a_I$  such that

$$\pi(\gamma) = f^{-1}(\sum_{I \subseteq D} a_I u_I(\gamma)). \quad (4)$$

*Constraints* The general constraints on binary data are

$$(\sum_{i \in I} m_i - |I| + 1) \vee 0 \leq m_I \leq \min\{m_K \mid K \subseteq I\}, \quad (5)$$

where the upper bound is the monotonicity of the measure, and the lower bound follows from

$$\begin{aligned} |I| - 1 &= \sum_{\gamma \in \mathbb{B}^d} (|I| - 1) \pi(\gamma) \\ &\geq \sum_{\gamma \in \mathbb{B}^d} (\sum_{i \in I} \gamma_i - u_I(\gamma)) \pi(\gamma) \\ &= \sum_{i \in I} m_i - m_I. \end{aligned}$$

In fact,  $m_I$  is a  $|I|$ -dimensional copula with respect to the expectations  $m_i$  for  $i \in I$ , see [Nelsen \(2006, p.45\)](#), and the inequalities (5) correspond to the Fréchet-Hoeffding bounds.

*Sampling* For sampling from a binary distribution  $\pi$ , we apply the chain rule factorization

$$\begin{aligned} \pi(\gamma) &= \pi_{\{1\}}(\gamma_1) \prod_{i=2}^d \pi_{\{1:i\}}(\gamma_i \mid \gamma_{1:i-1}) \\ &= \pi_{\{1\}}(\gamma_1) \prod_{i=2}^d \pi_{\{1:i-1\}}(\gamma_{1:i-1}) / \pi_{\{1:i\}}(\gamma_{1:i}), \end{aligned} \quad (6)$$

which permits to sample a random vector component-wise, conditioning on the entries we already generated. We do not even need to compute the full decomposition (6), but only the conditional probabilities  $\pi_{\{1:i\}}(\gamma_i = 1 \mid \gamma_{1:i-1})$  defined by

$$\frac{\pi_{\{1:i\}}(\gamma_{1:i-1}, 1)}{\pi_{\{1:i\}}(\gamma_{1:i-1}, 1) + \pi_{\{1:i\}}(\gamma_{1:i-1}, 0)}. \quad (7)$$

The full probability  $\pi(\gamma)$  is then computed as a by-product of the sampling Procedure 1.

---

### Procedure 1 Sampling via chain rule factorization

---

```

y = (0, ..., 0), p ← 1
for  $i = 1 \dots, d$  do
   $r \leftarrow \pi_{\{1:i\}}(\gamma_i = 1 \mid \gamma_{1:i-1})$ 
  sample  $y_i \sim \mathcal{B}_r$ 
   $p \leftarrow \begin{cases} p \cdot r & \text{if } y_i = 1 \\ p \cdot (1 - r) & \text{if } y_i = 0 \end{cases}$ 
end for
return y, p

```

---

## 3 Product models

The simplest non-trivial distributions on  $\mathbb{B}^d$  are certainly those having independent components.

### 3.1 Definition

For a vector  $\mathbf{m} \in (0, 1)^d$  of marginal probabilities, we define the product model

$$\begin{aligned} q_{\mathbf{m}}(\gamma) &\stackrel{\text{def}}{=} \prod_{i \in D} m_i^{\gamma_i} (1 - m_i)^{1 - \gamma_i} \\ &= \prod_{i \in D} (1 - m_i) \exp(\sum_{i \in D} \logit(m_i) \gamma_i). \end{aligned} \quad (8)$$

The second representation using the logit function

$$\text{logit}: (0, 1) \rightarrow \mathbb{R}, \quad \text{logit}(p) = \log p - \log(1 - p) \quad (9)$$

is useful to identify the product model as special case of more complex models. Later, we rather write  $\mathcal{B}_{\mathbf{m}}$  instead of  $q_{\mathbf{m}}$  for the product model, since (8) is the generalization of the Bernoulli distribution to  $d$  dimensions.

### 3.2 Properties

We check the requirement list from Section 1.3:

- (a) The product model is parsimonious with  $\dim(\theta) = d$ .
- (b) The maximum likelihood estimator  $\mathbf{m}^*$  is the sample mean (1).
- (c) We easily sample from  $q_{\mathbf{m}}$ , since (6) holds trivially.
- (d) We easily evaluate the probability of a product of independent components.
- (e) **The model  $q_{\mathbf{m}}$  does not reproduce any dependencies we might observe in the data  $\mathbf{X}$ .**

The last point is a weakness which makes this simple model impractical when adaptive Monte Carlo algorithms are applied to challenging sampling problems. The product model  $q_{\mathbf{m}}$  is often not flexible enough to come sufficiently close to the target distribution  $\pi$ . Therefore, the rest of this paper deals with ideas on how to sample binary vectors with a given dependence structure.

### 3.3 Beyond the product model

There are, to our knowledge, two main strategies to produce binary vectors with correlated components.

- (1) We can construct a generalized linear model which permits computation of its marginal distributions. We apply the chain rule factorization (6) and write  $q_{\theta}$  as

$$q_{\theta}(\gamma) = q_{\theta}(\gamma_1) \prod_{i=2}^d q_{\theta}(\gamma_i | \gamma_{1:i-1}), \quad (10)$$

which allows us to sample vectors component-wise.

- (2) We sample from a multivariate auxiliary distribution  $h_{\theta}$  and map the samples into  $\mathbb{B}^d$ . We call

$$q_{\theta}(\gamma) = \int_{\tau^{-1}(\gamma)} h_{\theta}(\mathbf{v}) d\mathbf{v} \quad (11)$$

a copula model, although we refrain from working with explicit uniform marginals (Mikosch, 2006).

In the following, we first study a few generalized linear models and then review a some copula approaches.

## 4 Linear models

Taking  $f$  the identity mapping in (4), we obtain a full linear representation

$$\pi(\gamma) = \sum_{I \subseteq D} a_I u_I(\gamma).$$

However, we cannot give a useful interpretation of the coefficients  $a_I$ . Bahadur (1961) derived the following representation:

**Proposition 1.** *Define the index set*

$$\mathcal{I} \stackrel{\text{def}}{=} \cup_{k \in D} \{I \subseteq D \mid |I| = k\}.$$

*Then we can write any binary distribution as*

$$\pi(\gamma) = \mathcal{B}_{\mathbf{m}}(\gamma) (1 + \sum_{I \in \mathcal{I}_k} v_I(\gamma) c_I),$$

where  $\mathbf{m} = (m_1, \dots, m_d)$  are the marginal probabilities.

*Proof.* For convenience, we give the proof proposed by Bahadur in Appendix 10.1.  $\square$

This decomposition, first discovered by Lazarsfeld, is a special case of a more general interaction theory (Streitberg, 1990) and allows for a reasonable interpretation of the parameters. Indeed, we have a product model times a correction term  $1 + \sum_{I \in \mathcal{I}_k} v_I(\gamma) c_I$  where the coefficients are higher order correlations.

### 4.1 Definition

We can try to construct a more parsimonious model by removing higher order interaction terms. For additive approaches, however, we face the problem that a truncated representations do not necessarily define probability distributions since they might not be non-negative.

Still, for a symmetric matrix  $\mathbf{A}$ , we define the  $d(d+1)/2$  parameter model

$$q_{\mathbf{A}, a_0}(\gamma) = \mu(a_0 + \gamma^T \mathbf{A} \gamma), \quad (12)$$

where  $\mu$  is a normalization constant and we set  $a_0 = -(\min_{\gamma \in \mathbb{B}^d} \gamma^T \mathbf{A} \gamma \wedge 0)$ . Since  $a_0$  is the solution of an NP hard quadratic unconstrained binary optimisation problem, this definition is of little practical value.

### 4.2 Moments

In virtue of the linear structure, we can derive explicit expressions for the cross-moments and marginal distributions, explicit meaning that the complexity is polynomial in  $d$ . The proofs are basic but rather tedious, so we moved them to the appendix section.

Next, we give a general formula yielding all cross-moments, including the normalization constant.

**Proposition 2.** For a set of indices  $I \subseteq D$ , we can write the corresponding cross-moment as

$$m_I = \frac{1}{2^{|I|}} + \frac{\sum_{i \in I} \left[ 2 \sum_{j \in D} a_{i,j} + \sum_{j \in I \setminus \{i\}}^d a_{i,j} \right]}{2^{|I|} (4a_0 + \mathbf{1}^\top \mathbf{A} \mathbf{1} + \text{tr}[\mathbf{A}])}.$$

For a proof see Appendix 10.2

**Corollary 1.** The normalization constant is

$$\mu = 2^{-d+2} (4a_0 + \mathbf{1}^\top \mathbf{A} \mathbf{1} + \text{tr}[\mathbf{A}])^{-1},$$

and the expected value is

$$\mathbb{E}_{q_{\mathbf{A},a_0}}(\gamma) = \frac{1}{2} + \frac{\sum_{k=1}^d a_{i,k}}{4a_0 + \mathbf{1}^\top \mathbf{A} \mathbf{1} + \text{tr}[\mathbf{A}]}.$$

The mean  $m_i$  is close to  $1/2$  unless the row  $\mathbf{a}_i$  dominates the matrix. Therefore, if  $\mathbf{A}$  is non-negative definite, the marginal probabilities  $m_i$  can hardly take values at the extremes of the unit interval.

### 4.3 Marginals

For the marginal distributions

$$q_{\mathbf{A},a_0}^{(1:k)}(\gamma_{1:k}) = \sum_{\xi \in \mathbb{B}^{d-(k+1)}} q_{\mathbf{A},a_0}(\gamma_{1:k}, \xi)$$

there are explicit and recursive formulae. Hence, we can compute the chain rule decomposition (6) which in turn allows to sample from the model.

**Proposition 3.** For the marginal distribution holds

$$q_{\mathbf{A},a_0}^{(1:k)}(\gamma_{1:k}) = \mu 2^{d-k-2} s_k(\gamma_{1:k}),$$

where

$$s_k(\gamma_{1:k}) = 4a_0 + \sum_{i=1}^k \gamma_i \left( \sum_{j=1}^k \gamma_j a_{i,j} + \sum_{j=k+1}^d a_{i,j} \right) + \sum_{i=k+1}^d \sum_{j=k+1}^d a_{i,j} + \sum_{i=k+1}^d a_{i,i}.$$

For a proof see Appendix 10.3

Recall the connection between marginal distributions and moments we observed in (3). For  $\gamma_I = \mathbf{1}$  we obtain

$$\begin{aligned} s_I(\mathbf{1}_k) &= 4a_0 + 4 \sum_{i \in I} \left( \sum_{j \in I} a_{i,j} + \sum_{j \in I^c} a_{i,j} \right) \\ &\quad + \sum_{i \in I^c} \sum_{j \in I^c} a_{i,j} + \sum_{i \in I^c} a_{i,i} \\ &= 4a_0 + \sum_{i \in D} \sum_{j \in D} a_{i,j} + \sum_{i \in D} a_{i,i} + 3 \sum_{i \in I} \sum_{j \in I} a_{i,j} \\ &\quad + 2 \sum_{i \in I} \sum_{j \in I^c} a_{i,j} - \sum_{i \in I} a_{i,i} \\ &= 4a_0 + \mathbf{1}^\top \mathbf{A} \mathbf{1} + \text{tr}[\mathbf{A}] + \\ &\quad \sum_{i \in I} \left[ 2 \sum_{j \in D} a_{i,j} + \sum_{j \in I \setminus \{i\}} a_{i,j} \right], \end{aligned}$$

and  $\pi_I(\mathbf{1}_k) = \mu 2^{d-|I|-2} s_I(\mathbf{1}_k)$  is indeed the expression for the cross-moments in Proof of Proposition 2.

### 4.4 Fitting the model

Given a sample  $\mathbf{X} = (\mathbf{x}_1, \dots, \mathbf{x}_n)^\top \sim \pi$  from the target distribution, we can determine  $a_0$  and a matrix  $\mathbf{A}$  such that the model  $q_{\mathbf{A},a_0}$  fits the first and second sampling moments

$$\bar{x}_{\{i,j\}} = n^{-1} \sum_{k=1}^n x_{k,i} x_{k,j}, \quad i, j \in D$$

by solving a linear system of dimension  $d(d+1)/2 + 1$ . We first use the bijection

$$\tau: D \times D \rightarrow \{1, \dots, d(d+1)/2\}, \quad \tau(i, j) = i(i-1)/2 + j$$

to map symmetric matrices into  $\mathbb{R}^{(d+1)d/2}$ . Precisely, for the matrices  $\mathbf{A}$  and  $\bar{\mathbf{X}}$ , we define the vectors

$$\hat{\mathbf{a}}_{\tau(i,j)} \stackrel{\text{def}}{=} a_{i,j}, \quad \hat{\mathbf{x}}_{\tau(i,j)} \stackrel{\text{def}}{=} \bar{x}_{i,j}$$

and the design matrix

$$\hat{\mathbf{S}}_{\tau(i,j), \tau(k,l)} \stackrel{\text{def}}{=} 2^{\mathbb{1}_{\{i,j\}(k)} + \mathbb{1}_{\{i,j,k\}(l)}}.$$

Note that  $|\hat{\mathbf{a}}| = \mathbf{1}^\top \mathbf{A} \mathbf{1} + \text{tr}[\mathbf{A}]$ . We then equate the distribution moments to the sample moments and normalize such that

$$2^{d-2} (\mathbf{I} a_0 + 1/4 \hat{\mathbf{S}} \hat{\mathbf{a}}) = \hat{\mathbf{x}}, \quad 2^{d-2} (4a_0 + |\hat{\mathbf{a}}|) = 1. \quad (13)$$

The solution of the linear system

$$\begin{pmatrix} \hat{\mathbf{a}}^* \\ a_0^* \end{pmatrix} = 2^{-d+2} \begin{bmatrix} 1/4 \hat{\mathbf{S}} & \mathbf{1} \\ 4 \mathbf{1}^\top & 1 \end{bmatrix}^{-1} \begin{pmatrix} \hat{\mathbf{x}} \\ 1 \end{pmatrix}$$

is finally transformed back into a symmetric matrix  $\mathbf{A}^*$ . Since the design matrix does not depend on the data, fitting several models to different data on the same space  $\mathbb{B}^d$  is extremely fast.

### 4.5 Properties

We check the requirement list from Section 1.3:

- The linear model is sufficiently parsimonious having dimension  $\dim(\theta) = d(d+1)/2$ .
- We can fit the parameters  $\mathbf{A}$  and  $a_0$  via method of moments. However, the fitted function  $q_{\mathbf{A}^*, a_0^*}(\gamma)$  is usually not a distribution.
- We can sample via chain rule factorization.
- We can evaluate  $q_{\mathbf{A},a_0}(\mathbf{y})$  via chain rule factorization while sampling.
- The model  $q_{\mathbf{A},a_0}$  reproduces the mean and correlations of the data  $\mathbf{X}$ .

Since in applications, the fitted matrix  $\mathbf{A}^*$  is hardly ever positive definite, we cannot use the linear model in an adaptive Monte Carlo context. As other authors (Park et al., 1996; Emrich and Piedmonte, 1991) remark, additive representations like Proposition 1 are instructive but we cannot derive practical models from them.

## 5 Log-linear models

If  $\pi(\gamma) > 0$  for all  $\gamma \in \mathbb{B}^d$ , we can use  $f = \log$  in (4) and obtain a full log-linear representation

$$\pi(\gamma) = \exp\left(\sum_{I \subseteq D} a_I u_I(\gamma)\right).$$

Note that we assume the probability mass function  $\pi$  is assumed to be log-linear in the parameters  $a_I$ . In the context of contingency tables the term ‘‘log-linear model’’ refers to the assumption that the marginal probabilities  $m_I$  are log-linear in the higher order marginals.

*Remark* Contingency table analysis is a well studied approach to modeling discrete data (Bishop et al., 1975; Christensen, 1997). For binary data, the underlying sampling distribution is assumed to be multinomial which requires an enumeration of the state space we want to avoid. Gange (1995) uses the Iterative Proportional Fitting algorithm (Haberman, 1972) from log-linear interaction theory to construct a binary distribution with given marginal probabilities. The fitting procedures, however, require storage of all configurations  $\pi_I(\gamma_I)$  and the construction of the joint posterior from the fitted marginal probabilities. The method is powerful and exact but computationally infeasible even for moderate dimensions.

### 5.1 Definition

Removing higher order interaction terms, we can construct a  $d(d+1)/2$  parameter model

$$q_{\mu, \mathbf{A}}(\gamma) \stackrel{\text{def}}{=} \mu \exp(\gamma^\top \mathbf{A} \gamma), \quad (14)$$

where  $\mathbf{A}$  is a symmetric matrix. We immediately recognize the product model (8) as the special case  $\mu = \prod_{i \in D} (1 - m_i)^d$  and  $\mathbf{A} = \text{diag}[\text{logit}(\mathbf{m})]$ . Cox and Wermuth (1994) refer to this version of the log-linear model as quadratic exponential model.

### 5.2 Marginals

The moments or marginal distributions of  $q_{\mathbf{A}}$  are sums of exponentials which, in general, do not simplify to expressions that are polynomial in  $d$ . Therefore, we cannot perform a chain rule factorization (6) to sample from the model.

Cox and Wermuth (1994) proposed the following second degree Taylor approximations to the marginal distributions which are again of the form (14).

**Proposition 4.** *We write the parameter  $\mathbf{A}$  as*

$$\mathbf{A} = \begin{pmatrix} \mathbf{A}' & \mathbf{b}^\top \\ \mathbf{b} & c \end{pmatrix}, \quad (15)$$

and define the parameters

$$\begin{aligned} \tilde{\mathbf{A}}_{d-1} &= \mathbf{A}' + (1 + \tanh(\frac{c}{2})) \text{diag}[\mathbf{b}] + \frac{1}{2} \text{sech}^2(\frac{c}{2}) \mathbf{b} \mathbf{b}^\top, \\ \tilde{\mu}_{d-1} &= \mu(1 + \exp(c)) \end{aligned}$$

Then  $q_{\tilde{\mathbf{A}}_{d-1}}(\gamma_{1:d-1})$  is the second degree Taylor approximation to the marginal distribution  $q_{\mathbf{A}_{1:d-1}}(\gamma_{1:d-1})$ . For a proof see Appendix 10.4.

If we recursively compute  $q_{\tilde{\mathbf{A}}_{d-1}}, \dots, q_{\tilde{\mathbf{A}}_1}$ , we can derive approximate conditional probabilities using (7). Precisely, we have

$$q_{\tilde{\mathbf{A}}_i}(\gamma_i = 1 \mid \gamma_{1:i-1}) = \text{logit}^{-1}(\tilde{c}_i + \tilde{\mathbf{b}}_i^\top \gamma_{1:i-1}), \quad (16)$$

where  $\text{logit}^{-1}(x) = (1 + \exp(-x))^{-1}$  and  $\tilde{c}_i, \tilde{\mathbf{b}}_i$  are parts of the matrix  $\tilde{\mathbf{A}}_i$  according to the notation introduced in (15). In particular, (16) is a logistic regression. We come back to this class of models in the following Section 6. We can sample from the proxy

$$\tilde{q}_{\tilde{\mathbf{A}}}(\gamma) \stackrel{\text{def}}{=} \prod_{i \in D} q_{\tilde{\mathbf{A}}_i}(\gamma_i \mid \gamma_{1:i-1}) \approx q_{\mathbf{A}}(\gamma),$$

which is close to the original log-linear model. The goodness of the approximation might be improved by judicious permutation of the components. The approximation error is hard to control, however, since we repeatedly apply the second degree approximation and propagate initial errors.

### 5.3 Fitting the model

As in section 4.4, we use the bijection

$$\tau: D \times D \rightarrow \{1, \dots, d(d+1)/2\}, \quad \tau(i, j) = i(i-1)/2 + j$$

to map symmetric matrices into  $\mathbb{R}^{(d+1)d/2}$ . Precisely, for the matrices  $\mathbf{A}$  and  $\bar{\mathbf{X}}$ , we define the vectors

$$\hat{a}_{\tau(i,j)} \stackrel{\text{def}}{=} a_{i,j}, \quad \hat{x}_{\tau(i,j)} \stackrel{\text{def}}{=} \bar{x}_{i,j}.$$

We let  $y_k = \log \pi(\mathbf{x}_k)$  for  $k = 1, \dots, n$  and fit the model solving the least square problem

$$\min_{\hat{\mathbf{a}} \in \mathbb{R}^{(d+1)d/2}} \|\hat{\mathbf{X}} \hat{\mathbf{a}} - \mathbf{y}\|_2$$

which yields the parameters

$$a_{i,j}^* = [(\hat{\mathbf{X}}^\top \hat{\mathbf{X}})^{-1} \hat{\mathbf{X}}^\top \mathbf{y}]_{\tau(i,j)}.$$

Note that in most adaptive Monte Carlo algorithms that involve importance sampling or Markov transitions, the probabilities  $\pi(\mathbf{x}_k)$  of the target distribution are already computed such that the fitting procedure is rather fast.

### 5.4 Properties

We check the requirement list from Section 1.3:

- (a) The log-linear model is sufficiently parsimonious with  $\dim(\theta) = d(d+1)/2$ .
- (b) We can fit the parameter  $\mathbf{A}$  via minimum least squares.

- (c) We can sample from an approximation  $\tilde{q}_{\tilde{\mathbf{A}}}(\boldsymbol{\gamma}) \approx q_{\mathbf{A}}(\boldsymbol{\gamma})$  to the log-linear model. However, we cannot control the approximation error.
- (d) We can evaluate  $q_{\mathbf{A},a_0}(\mathbf{y})$  up to the normalisation constant  $\mu$  which suffices for most adaptive Monte Carlo methods.
- (e) The model  $q_{\mathbf{A},a_0}$  reproduces the mean and correlations of the data  $\mathbf{X}$ .

## 6 Logistic models

In the previous section we saw that even for a rather simple non-linear model we cannot derive closed-form expressions for the marginal probabilities. Therefore, instead of computing the marginals for a  $d$ -dimensional model  $q_{\theta}(\boldsymbol{\gamma})$ , we directly fit univariate models

$$q_{\mathbf{b}_i}(\gamma_i = 1 \mid \gamma_{1:i-1}), \quad i \in D$$

to the conditional probabilities  $\pi(\gamma_i = 1 \mid \gamma_{1:i-1})$  of the target function. Precisely, we postulate the logistic relation

$$\logit(\mathbb{P}_{\pi}(\gamma_i = 1)) = b_{i,i} + \sum_{j=1}^{i-1} b_{i,j} \gamma_j, \quad i \in D$$

for the marginal probabilities of the target distribution  $\pi$ . We defined the logit function in (9).

### 6.1 Definition

For a  $d$ -dimensional lower triangular matrix  $\mathbf{B}$ , we define the logistic model as

$$\begin{aligned} q_{\mathbf{B}}(\boldsymbol{\gamma}) &\stackrel{\text{def}}{=} \prod_{i \in D} \mathcal{B}_{p(b_{i,i} + \mathbf{b}_{i,1:i-1}^{\top} \boldsymbol{\gamma}_{1:i-1})}(\gamma_i) \\ &= \exp\left(\sum_{i \in D} (\gamma_i - 1)(b_{i,i} + \mathbf{b}_{i,1:i-1}^{\top} \boldsymbol{\gamma}_{1:i-1})\right) \\ &\quad - \log(1 + \exp(b_{i,i} + \mathbf{b}_{i,1:i-1}^{\top} \boldsymbol{\gamma}_{1:i-1})) \end{aligned} \quad (17)$$

where  $\mathcal{B}_p$  is the Bernoulli distribution and

$$p(x) = \logit^{-1}(x) = (1 + \exp(-x))^{-1}$$

the inverse-logit function. We identify the product model  $\mathcal{B}_{\mathbf{m}}$  as the special case  $\mathbf{B} = \text{diag}[\logit(\mathbf{m})]$ . The logistic model is not a log-linear model.

Note that there are  $d!$  possible logistic models and we arbitrarily pick one while there should be a permutation  $\sigma(D)$  of the components which is optimal in a sense of nearness to the data. In practice, however, changing the parametrisation does not seem to have a noticeably impact on the quality of the adaptive Monte Carlo algorithm.

### 6.2 Sparse logistic regressions

The major drawback of all multiplicative models is the fact that they do not have closed-form likelihood-maximizers

such that the parameter estimation requires costly iterative fitting procedures. Therefore, we construct a sparse version of the logistic regression model which we can estimate faster than the saturated model.

Instead of fitting the saturated model  $q(\gamma_i \mid \gamma_{1:i-1})$ , we preferably work with a more parsimonious regression model like  $q(\gamma_i \mid \gamma_{L_i})$  for some index set  $L_i \subseteq \{1, \dots, i-1\}$ , where the number of predictors  $\#L_i$  is typically smaller than  $i-1$ .

We solve this nested variable selection problem using some simple, fast to compute criterion. For  $\varepsilon$  about  $1/100$ , we define the index set

$$I \stackrel{\text{def}}{=} \{i = 1, \dots, d \mid \bar{x}_i \notin (\varepsilon, 1 - \varepsilon)\},$$

which identifies the components which have, according to the data, a marginal probability close to either boundary of the unit interval.

We do not fit a logistic regression for the components  $i \in I$ . We rather set  $L_i = \emptyset$  and draw them independently, that is we set  $b_{i,i} = \logit(\bar{x}_i)$  and  $\mathbf{b}_{i,-i} = \mathbf{0}$  which corresponds to logistic model without predictors. The reason is twofold. Firstly, interactions do not really matter if the marginal probability is excessively small or large. Secondly, these components are prone to cause complete separation in the data or might even be constant.

For the conditional distribution of the remaining components  $I^c = D \setminus I$ , we construct parsimonious logistic regressions. For  $\delta$  about  $1/10$ , we define the predictor sets

$$L_i \stackrel{\text{def}}{=} \{j = 1, \dots, i-1 \mid \delta < |r_{i,j}|\}, \quad i \in I^c,$$

which identifies the components with index smaller than  $i$  and significant mutual association.

### 6.3 Fitting the model

Given a sample  $\mathbf{X} = (\mathbf{x}_1, \dots, \mathbf{x}_n)^{\top} \sim \pi$  from the target distribution we regress  $\mathbf{y}^{[i]} = \mathbf{X}_i$  on the columns  $\mathbf{Z}^{[i]} = (\mathbf{X}_{1:i-1}, \mathbf{1})$ , where the column  $\mathbf{Z}_i^{[i]}$  yields the intercept to complete the logistic model.

We maximise the log-likelihood function  $\ell(\mathbf{b}) = \ell(\mathbf{b} \mid \mathbf{y}, \mathbf{Z})$  of a weighted logistic regression model by solving the first order condition  $\partial \ell / \partial \boldsymbol{\beta} = \mathbf{0}$ . We find a numerical solution via Newton-Raphson iterations

$$-\frac{\partial^2 \ell(\mathbf{b}^{[r]})}{\partial \mathbf{b} \mathbf{b}^{\top}} (\mathbf{b}^{[r+1]} - \mathbf{b}^{[r]}) = \frac{\partial \ell(\mathbf{b}^{[r]})}{\partial \mathbf{b}}, \quad r > 0, \quad (18)$$

starting at some  $\mathbf{b}^{[0]}$ ; see Procedure 2 for the exact terms. Other updating formulas like Iteratively Reweighted Least Squares or quasi-Newton iterations should work as well.

**Procedure 2** Fitting the weighted logistic regressions**Input:**  $\mathbf{w} = (w_1, \dots, w_n)$ ,  $\mathbf{X} = (x_1, \dots, x_n)^\top$ ,  $\mathbf{B} \in \mathbb{R}^{d \times d}$ **for**  $i \in I^c$  **do** $\mathbf{Z} \leftarrow (\mathbf{X}_{L_i}, \mathbf{1})$ ,  $\mathbf{y} \leftarrow \mathbf{X}_i$ ,  $\mathbf{b}^{[0]} \leftarrow \mathbf{B}_{i, L_i \cup \{i\}}$ **repeat** $p_k \leftarrow \text{logit}^{-1}(\mathbf{Z}_k \mathbf{b}^{[r-1]})$  **for all**  $k = 1, \dots, n$  $q_k \leftarrow p_k(1 - p_k)$  **for all**  $k = 1, \dots, n$  $\mathbf{b}^{[r]} \leftarrow (\mathbf{Z}^\top \text{diag}[\mathbf{w}] \text{diag}[\mathbf{q}] \mathbf{Z} + \varepsilon \mathbf{I}_n)^{-1} \times$   
 $(\mathbf{Z}^\top \text{diag}[\mathbf{w}]) (\text{diag}[\mathbf{q}] \mathbf{Z} \mathbf{b}^{[r-1]} + (\mathbf{y} - \mathbf{p}))$ **until**  $|b_j^{[r]} - b_j^{[r-1]}| < 10^{-3}$  for all  $j$  $\mathbf{B}_{i, L_i \cup \{i\}} \leftarrow \mathbf{b}$ **end for****return**  $\mathbf{B}$ 

Sometimes, the Newton-Raphson iterations do not converge because the likelihood function is monotone and thus has no finite maximizer. This problem is caused by data with complete or quasi-complete separation in the sample points (Albert and Anderson, 1984). There are several ways to handle this issue.

- We just halt the algorithm after a fixed number of iterations and ignore the lack of convergence. Such proceeding, however, might cause uncontrolled numerical problems.
- Firth (1993) proposes to use a Jeffrey's prior on  $\mathbf{b}$ . The penalized log-likelihood does have a finite maximizer but requires computing the derivatives of the Fisher information matrix.
- We just add a simple quadratic penalty term  $\varepsilon \boldsymbol{\beta}^\top \boldsymbol{\beta}$  to the log-likelihood to ensure the target-function is convex and does not cause numerical problems.
- As we notice that some terms of  $\mathbf{b}_i$  are growing beyond a certain threshold, we move the component  $i$  from the set of components with associated logistic regression model  $I^c$  to the set of independent components  $I$ .

In practice, we recommend to combine the approaches (c) and (d). In Procedure 2, we did not elaborate how to handle non-convergence, but added a penalty term to the log-likelihood, which causes the extra  $\varepsilon \mathbf{I}_n$  in the Newton-Raphson update. Since we solve the update equation via Cholesky factorizations, adding a small term on the diagonal ensures that the matrix is indeed numerically decomposable.

## 6.4 Properties

We check the requirement list from Section 1.3:

- The logistic regression model is sufficiently parsimonious with  $\dim(\theta) = d(d+1)/2$ .

- We can fit the parameters  $\mathbf{b}_i$  via likelihood maximisation for all  $i \in D$ . The fitting is computationally intensive but feasible.

- We can sample  $\mathbf{y} \sim q_{\mathbf{B}}$  via chain rule factorization.

- We can exactly evaluate  $q_{\mathbf{B}}(\mathbf{y})$ .

- The model  $q_{\mathbf{B}}$  reproduces the dependency structure of the data  $\mathbf{X}$  although we cannot explicitly compute the marginal probabilities.

## 7 Gaussian copula models

In the preceding sections, we discussed three approaches based on generalized linear models. Now we turn to the second class of models we call copula models.

Let  $h_\theta$  be a family of auxiliary distributions on  $\mathcal{X}$  and  $\tau: \mathcal{X} \rightarrow \mathbb{B}^d$  a mapping into the binary state space. We can sample from the copula model

$$q_\theta(\boldsymbol{\gamma}) = \int_{\tau^{-1}(\boldsymbol{\gamma})} h_\theta(\mathbf{v}) d\mathbf{v}$$

by setting  $\mathbf{y} = h(\mathbf{v})$  for a draw  $\mathbf{v} \sim h_\theta$  from the auxiliary distribution.

## 7.1 Definition

Apparently, non-normal parametric distributions  $s_\theta$  with at most  $d(d-1)/2$  dependence parameters either have a very limited dependence structure or rather unfavourable properties (Joe, 1996). Therefore, the normal distribution

$$h_\Sigma(\mathbf{v}) = (2\pi)^{-d/2} |\Sigma|^{-1/2} \exp(-1/2 \mathbf{v}^\top \Sigma^{-1} \mathbf{v}),$$

with mapping  $\tau: \mathbb{R}^d \rightarrow \mathbb{B}^d$

$$\tau_\mu(\mathbf{v}) = (\mathbb{1}_{(\infty, \mu_1]}(v_1), \dots, \mathbb{1}_{(\infty, \mu_d]}(v_d)),$$

appears to be the natural and almost the only option for  $h_\theta$ . This model has already been discussed repeatedly in the literature (Emrich and Piedmonte, 1991; Leisch et al., 1998; Cox and Wermuth, 2002).

## 7.2 Moments

For  $I \subseteq D$ , the cross-moment or marginal probabilities is

$$m_I = \sum_{\boldsymbol{\gamma} \in \mathbb{B}^d} q_{\mu, \Sigma}(\mathbf{1}_I, \boldsymbol{\gamma}_{D \setminus I}) = \int_{\cup_{\boldsymbol{\gamma} \in \mathbb{B}^d} \{\tau_\mu^{-1}(\mathbf{1}_I, \boldsymbol{\gamma}_{D \setminus I})\}} h_\Sigma(\mathbf{v}) d\mathbf{v}$$

$$= \int_{\times_{i \in I} \{\tau_{\mu_i}^{-1}(1)\}} h_\Sigma(\mathbf{v}) d\mathbf{v} = \int_{\times_{i \in I} (-\infty, \mu_i]} h_\Sigma(\mathbf{v}) d\mathbf{v},$$

where we used (3) in the first line. Thus, the first and second moment of  $q_{(\mu, \Sigma)}$  are

$$m_i = \Phi_1(\mu_i), \quad m_{i,j} = \Phi_2(\mu_i, \mu_j; \sigma_{i,j})$$

where  $\Phi_1(v_i)$  and  $\Phi_2(v_i, v_j; \sigma_{i,j})$  denote the cumulative distribution functions of the univariate and bivariate normal distributions with zero mean, unit variance and correlation coefficient  $\sigma_{i,j} \in [-1, 1]$ .

### 7.3 Sparse Gaussian copulae

We can speed up the parameter estimation and improve the condition of  $\Sigma$ , if we work with a parsimonious Gaussian copula. We can apply the same criterion we already introduced for the sparse logistic regression model. For  $\varepsilon$  about  $1/100$ , we define the index set

$$I \stackrel{\text{def}}{=} \{i = 1, \dots, d \mid \bar{x}_i \notin (\varepsilon, 1 - \varepsilon)\}.$$

which identifies the components which have a marginal probability close to either boundary of the unit interval.

We do not fit any correlation parameters for the components in  $I$  but set  $\sigma_{i,j} = 0$  for all  $j \in D \setminus \{i\}$ . Firstly, the correlation does not really matter if the marginal probability is excessively small or large. Secondly, we fit the parameter  $\Sigma$  by separately adjusting the bivariate correlations  $\sigma_{i,j}$ , and components with high correlations and extreme marginal probability lower the chance that  $\Sigma$  is positive definite.

For the remaining components  $I^c = D \setminus I$ , we construct parsimonious Gaussian copula. For  $\delta$  about  $1/10$ , we define the association set

$$A \stackrel{\text{def}}{=} \{\{i, j\} \in I^c \times I^c \mid \delta < |r_{i,j}|, i \neq j\}$$

which identifies the components with significant correlation. For  $i, j \in D \times D \setminus L$  we also set  $\sigma_{i,j} = 0$  to accelerate the estimation procedure.

### 7.4 Fitting the model

We fit the model  $q_{(\mu, \Sigma)}$  to the data by adjusting  $\mu$  and  $\Sigma$  to the sample moments. Precisely, we solve the equations

$$\Phi_1(\mu_i) = \bar{x}_i, \quad i \in D \quad (19)$$

$$\Phi_2(\mu_i, \mu_j; \sigma_{i,j}) = \bar{x}_{i,j}, \quad (i, j) \in A \quad (20)$$

with sample mean  $\bar{x}_i$  and  $\bar{x}_{i,j}$  as defined in (1). We easily solve (19) by setting

$$\mu_i = \Phi_1^{-1}(\bar{x}_i), \quad i \in D.$$

The difficult task is computing a feasible correlation matrix from (20). Recall the standard result (Johnson et al., 2002, p.255)

$$\frac{\partial \Phi_2(y_1, y_2; \sigma)}{\partial \sigma} = h_\sigma(y_1, y_2), \quad (21)$$

where  $h_\sigma$  denotes the density of the bivariate normal distribution. We obtain the following Newton-Raphson iteration

$$\alpha_{r+1} = \alpha_r - \frac{\Phi_2(\mu_i, \mu_j; \alpha_r) - \bar{x}_{i,j}}{h_{\alpha_r}(\mu_i, \mu_j)}, \quad (i, j) \in A, \quad (22)$$

starting at some  $\alpha_0 \in (-1, 1)$ . We use a fast series approximation (Drezner and Wesolowsky, 1990; Divgi, 1979) to evaluate  $\Phi_2(\mu_i, \mu_j; \alpha)$ . These approximations are critical when  $\alpha_r$  comes very close to either boundary of  $[-1, 1]$ . The

Newton iteration might repeatedly fail when restarted at the corresponding boundary  $r_0 \in \{-1, 1\}$ . This is yet another reason why it is preferable to work with a sparse Gaussian copula. In any event,  $\Phi_2(y_1, y_2; \sigma)$  is monotonic in  $\sigma$  since (21), and we can switch to bi-sectional search if necessary.

---

#### Procedure 3 Fitting the dependency matrix

---

**Input:**  $\bar{x}_i, \bar{x}_{i,j}$  for all  $i, j \in D$

$\mu_i = \Phi_{-1}(\bar{x}_i)$  for all  $i \in D$

$\Sigma = \mathbf{I}_d$

**for**  $(i, j) \in A$  **do**

**repeat**

$$\sigma_{i,j}^{[r+1]} \leftarrow \sigma_{i,j}^{[r]} - \frac{\Phi_2(\mu_i, \mu_j; \sigma_{i,j}^{[r]}) - \bar{x}_{i,j}}{h_{\sigma_{i,j}^{[r]}}(\mu_i, \mu_j)}$$

**until**  $|\sigma_{i,j}^{[r]} - \sigma_{i,j}^{[r-1]}| < 10^{-3}$

**end for**

**if not**  $\Sigma \succ 0$  **then**  $\Sigma \leftarrow (\Sigma + |\lambda| \mathbf{I}_d) / (1 + |\lambda|)$

**return**  $\mu, \Sigma$

---

A rather discouraging shortcoming of the Gaussian copula model is that locally fitted correlation matrices  $\Sigma$  might not be positive definite for  $d \geq 3$ . This is due to the fact that an elliptical copula, like the Gaussian, can only attain the bounds (5) for  $d < 3$ , but not for higher dimensions.

We propose two ideas to obtain an approximate, but feasible parameter:

- (1) We replace  $\Sigma$  by  $\Sigma^* = (\Sigma + |\lambda| \mathbf{I}) / (1 + |\lambda|)$ , where  $\lambda$  is the smallest eigenvalue of the dependency matrix  $\Sigma$ . This approach evenly lowers the local correlations to a feasible level and is easy to implement on standard software. Alas, we make an effort to estimate  $d(d-1)/2$  dependency parameters, and in the end we might not get more than an product model.
- (2) We can compute the correlation matrix  $\Sigma^*$  which minimizes the distance  $\|\Sigma^* - \Sigma\|_F$ , where  $\|\mathbf{A}\|_F^2 = \text{tr}[\mathbf{A}\mathbf{A}^\top]$ . In other words, we construct the projection of  $\Sigma$  into the set of correlation matrices. Higham (2002) proposes an Alternating Projections algorithm to solve nearest-correlation matrix problems. Yet, if  $\Sigma$  is rather far from the set of correlation matrices, computing the projection is expensive and, according to our experience, leads to troublesome distortions in the correlation structure.

### 7.5 Properties

We check the requirement list from Section 1.3:

- (a) The Gaussian copula model is sufficiently parsimonious with  $\dim(\theta) = d(d+1)/2$ .
- (b) We can fit the parameters  $\mu$  and  $\Sigma$  via method of moments. **The parameter  $\Sigma$  is not always be positive definite which might require additional effort it feasible.**

- (c) We can sample  $\mathbf{y} \sim q_{(\mu, \Sigma)}$  using  $\mathbf{y} = \tau_{\mu}(\mathbf{v})$  with  $\mathbf{v} \sim h_{\Sigma}$ .
- (d) **We cannot evaluate  $q_{\mathbf{B}}(\mathbf{y})$  since this requires computing a high-dimensional integral expression.**
- (e) The model  $q_{(\mu, \Sigma)}$  exactly reproduces the mean and correlation structure of the data  $\mathbf{X}$ .

We cannot use the Gaussian copula model in the context of Importance Sampling or Markov chain Monte Carlo, since evaluation of  $q_{(\mu, \Sigma)}(\mathbf{y})$  is not possible. This model can be quite useful, however, in other adaptive Monte Carlo algorithms, for instance the Cross-Entropy method (Rubinstein, 1997) for combinatorial optimisation.

## 8 Poisson reduction models

Let  $N = \{1, \dots, n\}$  denote another index set with  $n \gg d$ . Approaches to generating binary vectors that do not rely on the chain rule factorization (6) are usually based on combinations of independent random variables

$$\mathbf{v} = (v_1, \dots, v_n) \sim \otimes_{k \in N} h_{\theta_k}.$$

We define index sets  $\mathcal{M} = \{S_i \in N \mid i \in D\}$  and generate the entry  $y_i$  via

$$\tau_i: \mathcal{X}^{|S_i|} \rightarrow \{0, 1\}, \quad \tau_i(\mathbf{v}) = f(\mathbf{v}_{S_i}), \quad i \in D.$$

In the context of Gaussian copulae, the auxiliary distributions  $h_{\theta_k} = h_{\theta}$  are  $d$  independent standard normal variables. Park et al. (1996) propose the following model based on sums of independent Poisson variables.

### 8.1 Definition

We define a Poisson model  $q_{(\mathcal{S}, \lambda)}$  with auxiliary distribution

$$h_{\lambda}(\mathbf{v}) = \prod_{k \in N} (\lambda_k^{v_k} e^{-\lambda_k}) / v_k!$$

and mapping  $\tau: \mathbb{N}_0^n \rightarrow \mathbb{B}^d$

$$\tau_{\mathcal{S}}(\mathbf{v}) = (\mathbb{1}_{\{0\}}(\sum_{k \in S_1} v_k), \dots, \mathbb{1}_{\{0\}}(\sum_{k \in S_d} v_k)).$$

### 8.2 Moments

For an index set  $I \in D$ , the cross-moments or marginal probabilities are

$$m_I = \mathbb{P}(\forall i \in I: \sum_{k \in S_i} v_k = 0) = \exp(-\sum_{k \in \cap_{i \in I} S_i} \lambda_k).$$

Therefore, fitting via method of moments is possible.

**Proposition 5.** For  $\gamma \in \mathbb{B}^d$ , define the index sets

$$D_0 = \{i \in D \mid \gamma_i = 0\}, \quad D_1 = \{i \in D \mid \gamma_i = 1\},$$

and the families of subsets  $\mathcal{S}_t = \{I \in D_1 \mid |I| = t\}$ . We can write the mass function of the Poisson model as

$$q_{(\mathcal{S}, \lambda)}(\gamma) = \sum_{\mathbf{v} \in \tau^{-1}(\gamma)} h_{\lambda}(\mathbf{v}) = m_{D_0} \left[ 1 - \sum_{t=1}^{|D_0|} (-1)^{t-1} \sum_{I \subseteq \mathcal{S}_t} \exp(-\sum_{k \in \cap_{i \in I} S_i \cup_{j \in D_1} S_j} \lambda_k) \right].$$

For a proof see Appendix 10.5.

### 8.3 Fitting the model

We need to determine the family of index sets  $\mathcal{M}$  and the Poisson parameters  $\lambda = (\lambda_1, \dots, \lambda_n)$  such that the resulting model  $q_{(\mathcal{S}, \lambda)}$  is optimal in terms of distance to the mean and correlation. Obviously, we face a rather difficult combinatorial problem. Park et al. (1996) describe a greedy algorithm, based on convolutions of Poisson variables, that finds at least some feasible combination of  $\mathcal{S}$  and  $\lambda$ .

### 8.4 Properties

We check the requirement list from Section 1.3:

- (a) The Poisson reduction model is not necessarily parsimonious. The number of parameters  $\dim(\theta)$  is determined by the fitting algorithm.
- (b) We fit the model via method of moments using a fast but non-optimal greedy algorithm.
- (c) We sample  $\mathbf{y} \sim q_{(\mathcal{S}, \lambda)}$  using  $\mathbf{y} = \tau_{\mathcal{S}}(\mathbf{v})$  with  $\mathbf{v} \sim h_{\lambda}$ .
- (d) **We cannot evaluate  $q_{(\mathcal{S}, \lambda)}(\mathbf{y})$  since it requires summation of  $2^{d-|\mathbf{y}|} - 1$  terms using an inclusion-exclusion principle which is computationally not feasible.**
- (e) The model  $q_{(\mathcal{S}, \lambda)}$  reproduces the mean and certain correlation structures of the data  $\mathbf{X}$ . **We cannot sample negative correlations.**

Since the model is limited to certain patterns of non-negative correlations, we cannot use it as general-purpose model in adaptive Monte Carlo algorithms. It might be useful, however, if we know that the target distribution  $\pi$  has strictly non-negative correlations.

## 9 Archimedean copula models

Genest and Neslehova (2007) discuss in detail the potentials and pitfalls of applying copula theory, which is well developed for bivariate, continuous random variables, to multivariate discrete distribution. Yet, there have been earlier attempts to sample binary vectors via copulae: Lee (1993) describes how to construct an Archimedean copula, more precisely the Frank family, (see e.g. Nelsen (2006, p.119)), for sampling multivariate binary data.

Unfortunately, most results in copula theory do not easily extend to high dimensions. Indeed, we need to solve a non-linear equation for each component when generating a random vector from the Frank copula, and Lee (1993) acknowledges that this is only applicable for  $d \leq 3$ . For low-dimensional problems, however, we can just enumerate the solution space  $\mathbb{B}^d$  and draw from an alias table (Walker, 1977), which somewhat renders the Archimedean copula approach an interesting exercise, but without much practical value in Monte Carlo applications.

## References

- Albert, A. and Anderson, J. A. (1984). On the existence of maximum likelihood estimates in logistic regression models. *Biometrika*, (72):1–10.
- Andrieu, C. and Thoms, J. (2008). A tutorial on adaptive MCMC. *18(4)*:343–373.
- Bahadur, R. (1961). A representation of the joint distribution of responses to  $n$  dichotomous items. In Solomon, H., editor, *Studies in Item Analysis and Prediction*, pages pp. 158–68. Stanford University Press.
- Bishop, Y., Fienberg, S., and Holland, P. (1975). *Discrete multivariate analysis: Theory and Practice*. Cambridge, MA: MIT Press.
- Cappé, O., Douc, R., Guillin, A., Marin, J., and Robert, C. (2008). Adaptive importance sampling in general mixture classes. *Statistics and Computing*, 18(4):447–459.
- Christensen, R. (1997). *Log-linear models and logistic regression*. Springer Verlag.
- Cox, D. and Wermuth, N. (1994). A note on the quadratic exponential binary distribution. *Biometrika*, 81(2):403–408.
- Cox, D. and Wermuth, N. (2002). On some models for multivariate binary variables parallel in complexity with the multivariate Gaussian distribution. *Biometrika*, 89(2):462.
- Del Moral, P., Doucet, A., and Jasra, A. (2006). Sequential monte carlo samplers. *Journal of the Royal Statistical Society: Series B(Statistical Methodology)*, 68(3):411–436.
- Divgi (1979). Computation of univariate and bivariate normal probability functions. *Annals of Statistics*, (7):903–910.
- Drezner, Z. and Wesolowsky, G. O. (1990). On the computation of the bivariate normal integral. *Journal of Statistical Computation and Simulation*, (35):101–107.
- Emrich, L. and Piedmonte, M. (1991). A method for generating high-dimensional multivariate binary variates. *The American Statistician*, 45(4):302–304.
- Firth, D. (1993). Bias reduction of maximum likelihood estimates. *Biometrika*, (80):27–38.
- Gange, S. (1995). Generating Multivariate Categorical Variates Using the Iterative Proportional Fitting Algorithm. *The American Statistician*, 49(2).
- Genest, C. and Neslehova, J. (2007). A primer on copulas for count data. *Astin Bulletin*, 37(2):475.
- Haberman, S. (1972). Algorithm AS 51: Log-linear fit for contingency tables. *Applied Statistics*, pages 218–225.
- Higham, N. J. (2002). Computing the nearest correlation matrix — a problem from finance. *IMA Journal of Numerical Analysis*, (22):329–343.
- Joe, H. (1996). Families of  $m$ -variate distributions with given margins and  $m(m-1)/2$  bivariate dependence parameters. *Lecture Notes-Monograph Series*, 28:120–141.
- Johnson, N., Kotz, S., and Balakrishnan, N. (2002). *Continuous Multivariate Distributions, volume 1, Models and Applications*. New York: John Wiley & Sons,.
- Lee, A. (1993). Generating Random Binary Deviates Having Fixed Marginal Distributions and Specified Degrees of Association. *The American Statistician*, 47(3).
- Leisch, F., Weingessel, A., and Hornik, K. (1998). On the generation of correlated artificial binary data. *preparation*.
- Mikosch, T. (2006). Copulas: Tales and facts. *Extremes*, 9(1):3–20.
- Nelsen, R. (2006). *An introduction to copulas*. Springer Verlag.
- Park, C., Park, T., and Shin, D. (1996). A Simple Method for Generating Correlated Binary Variates. *The American Statistician*, 50(4).
- Rubinstein, R. Y. (1997). Optimization of computer simulation models with rare events. *European Journal of Operations Research*, 99:89–112.
- Rubinstein, R. Y. and Kroese, D. P. (2004). *The Cross-Entropy Method: A Unified Approach to Combinatorial Optimization, Monte-Carlo Simulation, and Machine Learning*. Springer-Verlag.
- Streitberg, B. (1990). Lancaster interactions revisited. *Annals of Statistics*, 18(4):1878–1885.
- Walker, A. (1977). An efficient method for generating discrete random variables with general distributions. *ACM Transactions on Mathematical Software*, 3(3):256.

## 10 Appendix

### 10.1 Proposition 1

*Proof.* Recall that  $\mathcal{I} = \cup_{k \in D} \{\{i_1, \dots, i_k\} \subseteq D \mid i_1 < \dots < i_k\}$  and  $v_I(\gamma) = \prod_{i \in I} [(\gamma_i - m_i) / \sqrt{m_i(1 - m_i)}]$  with  $m_i > 0$  for all  $i \in D$ . We define an inner product

$$(f, g) \stackrel{\text{def}}{=} \mathbb{E}_{\mathcal{B}_m} (f(\gamma)g(\gamma)) = \sum_{\gamma \in \mathbb{B}^d} f(\gamma)g(\gamma) \prod_{i \in D} m_i^{\gamma_i} (1 - m_i)^{1 - \gamma_i}$$

on the vector space of real-valued functions on  $\mathbb{B}^d$ . The set  $S = \{v_I(\gamma) \mid I \in \mathcal{I}\}$  is orthonormal, since

$$(v_I, v_J) = \prod_{i \in I \cap J} \mathbb{E}_{\mathcal{B}_m} \left( \frac{(\gamma_i - m_i)^2}{m_i(1 - m_i)} \right) \prod_{i \in (I \cup J) \setminus (I \cap J)} \mathbb{E}_{\mathcal{B}_m} \left( \frac{\gamma_i - m_i}{\sqrt{m_i(1 - m_i)}} \right) = \begin{cases} 0 & \text{for } I \neq J \\ 1 & \text{for } I = J, \end{cases}$$

There are  $2^d - 1$  elements in  $S$  and  $\mathcal{B}_m(\gamma) > 0$  which implies that  $S \cup \{1\}$  is an orthonormal basis of the real-valued function on  $\mathbb{B}^d$ . It follows that each function  $f: \mathbb{B}^d \rightarrow \mathbb{R}$  has exactly one representation as linear combination of functions in  $S \cup \{1\}$  which is  $f = (f, 1) + \sum_{I \in \mathcal{I}} v_I(f, v_I)$ . Since

$$(\pi / \mathcal{B}_m, v_I) = \sum_{\gamma \in \mathbb{B}^d} (\pi(\gamma) / \mathcal{B}_m(\gamma)) v_I(\gamma) \mathcal{B}_m(\gamma) = \mathbb{E}_\pi (v_I(\gamma)) = c_I,$$

we obtain  $\pi(\gamma) / \mathcal{B}_m(\gamma) = 1 + \sum_{I \in \mathcal{I}} v_I(\gamma) c_I$  for  $f = \pi / \mathcal{B}_m$  which concludes the proof.  $\square$

### 10.2 Proposition 2

*Proof.* We first derive two auxiliary results to structure the proof.

*Lemma 1.* For a set  $I \subseteq D$  of indices it holds that

$$\sum_{\gamma \in \mathbb{B}^d} \prod_{k \in I \cup \{i, j\}} \gamma_k = 2^{d - |I| - 2 + \mathbb{1}_I(i) + \mathbb{1}_{I \cup \{i\}}(j)}.$$

For an index set  $M \subseteq D$ , we have the sum formula  $\sum_{\gamma \in \mathbb{B}^d} \prod_{k \in M} \gamma_k = 2^{d - |M|}$ . If we have an empty set  $M = \emptyset$  the sum equals  $2^d$  and each time we add a new index  $i \in D \setminus M$  to  $M$  half of the addends vanish. The number of elements in  $M = I \cup \{i, j\}$  is the number of elements in  $I$  plus one if  $i \notin I$  and again plus one if  $i \neq j$  and  $j \notin I$ . Written using indicator function, we have  $|I \cup \{i, j\}| = |I| + \mathbb{1}_{D \setminus I}(i) + \mathbb{1}_{D \setminus (I \cup \{i\})}(j) = |I| + 2 - \mathbb{1}_I(i) - \mathbb{1}_{I \cup \{i\}}(j)$  which implies Lemma 1.  $\square$

*Lemma 2.*

$$\sum_{i \in D} \sum_{j \in D} 2^{\mathbb{1}_I(i) + \mathbb{1}_{I \cup \{i\}}(j)} a_{i,j} = \mathbf{1}^\top \mathbf{A} \mathbf{1} + \text{tr}[\mathbf{A}] + \sum_{i \in I} [2 \sum_{j \in D} a_{i,j} + \sum_{j \in I \setminus \{i\}} a_{i,j}]$$

Straightforward calculations:

$$\begin{aligned} 2^{\mathbb{1}_I(i) + \mathbb{1}_{I \cup \{i\}}(j)} &= (1 + \mathbb{1}_I(i))(1 + \mathbb{1}_{I \cup \{i\}}(j)) = (1 + \mathbb{1}_I(i))(1 + \mathbb{1}_I(j) + \mathbb{1}_{\{i\}}(j) - \mathbb{1}_{I \cap \{i\}}(j)) \\ &= 1 + \mathbb{1}_I(i) + \mathbb{1}_I(j) + \mathbb{1}_I(i)\mathbb{1}_I(j) + \mathbb{1}_{\{i\}}(j) + \mathbb{1}_I(i)\mathbb{1}_{\{i\}}(j) - \mathbb{1}_{I \cap \{i\}}(j) - \mathbb{1}_I(i)\mathbb{1}_{I \cap \{i\}}(j) \\ &= 1 + \mathbb{1}_{\{i\}}(j) + \mathbb{1}_I(i) + \mathbb{1}_I(j) + \mathbb{1}_{I \times I}(i, j) - \mathbb{1}_{I \cap \{i\}}(j), \end{aligned}$$

where we used  $\mathbb{1}_I(i)\mathbb{1}_{\{i\}}(j) = \mathbb{1}_I(i)\mathbb{1}_I(i)\mathbb{1}_{\{i\}}(j) = \mathbb{1}_I(i)\mathbb{1}_I(j)\mathbb{1}_{\{i\}}(j) = \mathbb{1}_I(i)\mathbb{1}_{I \cap \{i\}}(j)$  in the second line. Thus, we have

$$\begin{aligned} \sum_{i \in D} \sum_{j \in D} 2^{\mathbb{1}_I(i) + \mathbb{1}_{I \cup \{i\}}(j)} a_{i,j} &= \sum_{i \in D} \sum_{j \in D} (1 + \mathbb{1}_{\{i\}}(j) + \mathbb{1}_I(i) + \mathbb{1}_I(j) + \mathbb{1}_{I \times I}(i, j) - \mathbb{1}_{I \cap \{i\}}(j)) a_{i,j} \\ &= \sum_{i \in D} \sum_{j \in D} a_{i,j} + \sum_{j \in D} a_{j,j} + \sum_{i \in I} \sum_{j \in D} a_{i,j} + \sum_{i \in D} \sum_{j \in I} a_{i,j} + \sum_{i \in I} \sum_{j \in I} a_{i,j} - \sum_{i \in I} a_{j,j} \\ &= \mathbf{1}^\top \mathbf{A} \mathbf{1} + \text{tr}[\mathbf{A}] + \sum_{k \in I} [2 \sum_{l \in D} a_{k,l} + \sum_{l \in I} a_{k,l} - a_{k,k}] \\ &= \mathbf{1}^\top \mathbf{A} \mathbf{1} + \text{tr}[\mathbf{A}] + \sum_{k \in I} [2 \sum_{l \in D} a_{k,l} + \sum_{l \in I \setminus \{k\}} a_{k,l}] \end{aligned}$$

The last line is the assertion of Lemma 2.  $\square$

Using the two Lemmata above, we find a convenient expression for the cross-moment

$$\begin{aligned}
m_I &= \sum_{\gamma \in \mathbb{B}^d} (\prod_{k \in I} \gamma_k) \mu(a_0 + \gamma^\top \mathbf{A} \gamma) \\
&= \mu \left[ \sum_{\gamma \in \mathbb{B}^d} a_0 + \sum_{\gamma \in \mathbb{B}^d} (\prod_{k \in I} \gamma_k) \sum_{i \in D} \sum_{j \in D} \gamma_i \gamma_j a_{i,j} \right] \\
&= \mu \left[ 2^{d-|I|} a_0 + \sum_{i \in D} \sum_{j \in D} a_{i,j} \sum_{\gamma \in \mathbb{B}^d} (\prod_{k \in I \cup \{i,j\}} \gamma_k) \right] && \text{(Lemma 1)} \\
&= \mu \left[ 2^{d-|I|} a_0 + \sum_{i \in D} \sum_{j \in D} 2^{d-|I \cup \{i,j\}|} a_{i,j} \right] \\
&= \mu 2^{d-|I|-2} \left[ 4a_0 + \sum_{i \in D} \sum_{j \in D} 2^{\mathbb{1}_I(i) + \mathbb{1}_{I \cup \{i\}}(j)} a_{i,j} \right] && \text{(Lemma 2)} \\
&= \mu 2^{d-|I|-2} \left[ 4a_0 + \mathbf{1}^\top \mathbf{A} \mathbf{1} + \text{tr}[\mathbf{A}] + \sum_{i \in I} \left[ 2 \sum_{j \in D} a_{i,j} + \sum_{j \in I \setminus \{i\}} a_{i,j} \right] \right]
\end{aligned}$$

Since  $m_\emptyset = 1$  by definition, we the normalization constant is  $\mu = 2^{-d+2} (4a_0 + \mathbf{1}^\top \mathbf{A} \mathbf{1} + \text{tr}[\mathbf{A}])^{-1}$ , which allows us to write down the normalized cross-moments

$$m_I = \frac{1}{2^{|I|}} + \frac{\sum_{i \in I} \left[ 2 \sum_{j \in D} a_{i,j} + \sum_{j \in I \setminus \{i\}} a_{i,j} \right]}{2^{|I|} (4a_0 + \mathbf{1}^\top \mathbf{A} \mathbf{1} + \text{tr}[\mathbf{A}])}.$$

The proof is complete.  $\square$

### 10.3 Proposition 3

*Proof.* We easily margin out the last component  $d$ , having  $I = \{1, \dots, d-t\}$ ,

$$\begin{aligned}
q_{\mathbf{A}, a_0}^{(d-1)}(\gamma_I) \mu^{-1} &= \left( q_{\mathbf{A}, a_0}^{(d)}(\gamma_I, 1) + q_{\mathbf{A}, a_0}^{(d)}(\gamma_I, 0) \right) \mu^{-1} = 2a_0 + (\gamma_I, 1)^\top \mathbf{A} (\gamma_I, 1) + (\gamma_I, 0)^\top \mathbf{A} (\gamma_I, 0) \\
&= 2a_0 + \text{tr}[\mathbf{A} [(\gamma_I, 1)(\gamma_I, 1)^\top + (\gamma_I, 0)(\gamma_I, 0)^\top]] = 2a_0 + \text{tr} \left[ \mathbf{A} \begin{bmatrix} 2\gamma_I \gamma_I^\top & \gamma_I \\ 2\gamma_I^\top & 1 \end{bmatrix} \right]
\end{aligned}$$

Iterating the argument, we obtain for  $I = \{1, \dots, d-t\}$  and  $I^c = D \setminus I$

$$q_{\mathbf{A}, a_0}^{(d-t)}(\gamma_I) \mu^{-1} = 2^t a_0 + 2^{t-2} \text{tr} \left[ \mathbf{A} \begin{bmatrix} 4\gamma_I \gamma_I^\top & 2\gamma_I \mathbf{1}_t^\top \\ 2\mathbf{1}_t \gamma_I^\top & \mathbf{1}_t \mathbf{1}_t^\top + \mathbf{I}_t \end{bmatrix} \right]$$

Straightforward calculations:

$$\begin{aligned}
\text{tr} \left[ \mathbf{A} \begin{bmatrix} 4\gamma_I \gamma_I^\top & 2\gamma_I \mathbf{1}_t^\top \\ 2\mathbf{1}_t \gamma_I^\top & \mathbf{1}_t \mathbf{1}_t^\top + \mathbf{I}_t \end{bmatrix} \right] &= \text{tr}[\mathbf{A} [(2\gamma_I, \mathbf{1}_t)(2\gamma_I, \mathbf{1}_t)^\top + \text{diag}[\mathbf{0}_t, \mathbf{1}_t]]] = [(2\gamma_I, \mathbf{1}_t)^\top \mathbf{A} (2\gamma_I, \mathbf{1}_t) + \text{tr}[\mathbf{A} \text{diag}[\mathbf{0}_t, \mathbf{1}_t]]] \\
&= [4 \sum_{i \in I} \sum_{j \in I} \gamma_i \gamma_j a_{i,j} + 4 \sum_{i \in I} \sum_{j \in I^c} \gamma_i a_{i,j} + \sum_{i \in I^c} \sum_{j \in I^c} a_{i,j} + \sum_{i \in I^c} a_{i,i}] \\
&= [4 \sum_{i \in I} \gamma_i (\sum_{j \in I} \gamma_j a_{i,j} + \sum_{j \in I^c} a_{i,j}) + \sum_{i \in I^c} \sum_{j \in I^c} a_{i,j} + \sum_{i \in I^c} a_{i,i}]
\end{aligned}$$

The proof is complete.  $\square$

### 10.4 Proposition 4

*Proof.* For convenience of notation, let  $\gamma_- = (\gamma_1, \dots, \gamma_{d-1})$ . Note that  $q_{\mathbf{A}}(\gamma) = \mu \exp(\gamma_-^\top \mathbf{A}' \gamma_- + \gamma_d (2\mathbf{b}^\top \gamma_- + c))$ . The marginal distribution is therefore

$$\begin{aligned}
\pi(\gamma_-) &= \mu \exp(\gamma_-^\top \mathbf{A}' \gamma_-) (1 + \exp(2\gamma_-^\top \mathbf{b} + c)) \\
&= \mu \exp(\gamma_-^\top \mathbf{A}' \gamma_- + \gamma_-^\top \mathbf{b} + c/2) (\exp(-\gamma_-^\top \mathbf{b} - c/2) + \exp(\gamma_-^\top \mathbf{b} + c/2)) \\
&= \mu \exp(\gamma_-^\top \mathbf{A}' \gamma_- + \gamma_-^\top \mathbf{b} + c/2) 2 \cosh(\gamma_-^\top \mathbf{b} + c/2).
\end{aligned}$$

The marginal log mass function is thus

$$\log \pi(\gamma_-) = \log(2\mu) + c/2 + \gamma_-^\top \mathbf{A}' \gamma_- + \gamma_-^\top \mathbf{b} + \log \cosh(\gamma_-^\top \mathbf{b} + c/2).$$

For logcosh we can use a Taylor approximation

$$\log \cosh(\gamma_-^\top \mathbf{b} + c/2) \approx \log \cosh(c/2) + \gamma_-^\top \mathbf{b} \tanh(c/2) + 1/2 (\gamma_-^\top \mathbf{b})^2 \operatorname{sech}^2(c/2)$$

to obtain

$$\log \pi(\gamma_-) \approx \log(2\mu \cosh(c/2)) + c/2 + \gamma_-^\top \mathbf{A}' \gamma_- + (1 + \tanh(c/2)) \gamma_-^\top \mathbf{b} + 1/2 \operatorname{sech}^2(c/2) (\gamma_-^\top \mathbf{b})^2$$

Since  $\gamma_-$  is a binary vector, we have  $\gamma_-^\top \mathbf{b} = \gamma_-^\top \operatorname{diag}[\mathbf{b}] \gamma_-$  and can thus rewrite the inner products as

$$\begin{aligned} \gamma_-^\top \mathbf{A}' \gamma_- + \gamma_-^\top \mathbf{b} + (\gamma_-^\top \mathbf{b})^2 &= \operatorname{tr} [\mathbf{A}' \gamma_- \gamma_-^\top + \operatorname{diag}[\mathbf{b}] \gamma_- \gamma_-^\top + \mathbf{b} \mathbf{b}^\top \gamma_- \gamma_-^\top] \\ &= \gamma_-^\top (\mathbf{A}' + \operatorname{diag}[\mathbf{b}] + \mathbf{b} \mathbf{b}^\top) \gamma_-. \end{aligned}$$

We let denote

$$\mu^* = 2\mu \cosh(c/2) \exp(c/2) = \mu (\exp(-c/2) + \exp(c/2)) \exp(c/2) = \mu (1 + \exp(c))$$

and

$$\mathbf{A}^* = \mathbf{A}' + (1 + \tanh(c/2)) \operatorname{diag}[\mathbf{b}] + 1/2 \operatorname{sech}^2(c/2) \mathbf{b} \mathbf{b}^\top$$

to form the approximation  $\pi(\gamma_-) \approx \mu^* \exp(\gamma_-^\top \mathbf{A}^* \gamma_-)$  which completes the proof. □

## 10.5 Proposition 5

*Proof.* Straightforward calculations using the inclusion-exclusion principle for the union of events:

$$\begin{aligned} q_{(\mathcal{S}, \lambda)}(\gamma) &= \sum_{\mathbf{v} \in \tau^{-1}(\gamma)} h_\lambda(\mathbf{v}) = \mathbb{P}_{h_\lambda} (\cap_{i \in \mathcal{D}} \{\mathbf{1}_{\{0\}} \sum_{k \in S_i} v_k = \gamma_i\}) \\ &= \mathbb{P}_{h_\lambda} (\cap_{i \in \mathcal{D}_1} \cap_{k \in S_i} \{v_k = 0\}, \cap_{i \in \mathcal{D}_0} \cup_{k \in S_i} \{v_k > 0\}) \\ &= \mathbb{P}_{h_\lambda} (\cap_{i \in \mathcal{D}_1} \cap_{k \in S_i} \{v_k = 0\}) \mathbb{P}_{h_\lambda} (\cap_{i \in \mathcal{D}_0} \cup_{k \in S_i \setminus \cup_{j \in \mathcal{D}_1} S_j} \{v_k > 0\}) \\ &= \mathbb{P}_{q_{(\mathcal{S}, \lambda)}} (\gamma_{\mathcal{D}_1} = \mathbf{1}) \left( 1 - \mathbb{P}_{h_\lambda} (\cup_{i \in \mathcal{D}_0} \cap_{k \in S_i \setminus \cup_{j \in \mathcal{D}_1} S_j} \{v_k = 0\}) \right) \\ &= m_{\mathcal{D}_0} \left[ 1 - \sum_{t=1}^{|\mathcal{D}_0|} (-1)^{t-1} \sum_{I \subseteq \mathcal{S}_t} \mathbb{P}_{h_\lambda} (\cap_{i \in I} \cap_{k \in S_i \setminus \cup_{j \in \mathcal{D}_1} S_j} \{v_k = 0\}) \right] \\ &= m_{\mathcal{D}_0} \left[ 1 - \sum_{t=1}^{|\mathcal{D}_0|} (-1)^{t-1} \sum_{I \subseteq \mathcal{S}_t} \exp(-\sum_{k \in \cap_{i \in I} S_i \setminus \cup_{j \in \mathcal{D}_1} S_j} \lambda_k) \right]. \end{aligned}$$

The proof is complete. □